

Grégory Gosselin De Bénicourt

Unreal Engine 4

Guide des Mises à Jour



Plus de 200 fonctionnalités à la loupe

Unreal Engine 4 – Guide des mises à jour (4.20)

Copyright Éditions Graziel

Dépôt Légal: Août 2018

ISBN 979-10-93846-09-5

Éditions Graziel

9, chemin des Barroutiers

81300 Graulhet

Tel: 09 52 05 40 15

Fax: 09 57 05 40 15

RCS: Castres B 801 370 800

courriel: infos@graziel.com

site web: www.graziel.com

Auteur: Grégory GOSSELLIN DE BENICOURT

Collection «Les petits guides »

Unreal Engine 4 – Guide des mises à jour (4.20)



Ce livre vous est **offert**,
mais il n'est ni **libre** ni **gratuit**

Le logo d'Unreal Engine est une marque déposée par EPIC Games, 620 Crossroads Blvd., Cary, NC 27518 - USA

Toutes les autres marques citées ont été déposées par leur éditeur respectif.

La loi du 11 mars 1957 n'autorisant aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective», et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayant cause, est illicite» (alinéa 1^{er} de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contre-façon sanctionnée par les articles 425 et suivants du Code Pénal.

Sommaire

POURQUOI CE LIVRE?	14
Présentation d'Unreal Engine.....	14
De Unreal Engine 1.0 à 4.20.....	17
Faisons un peu connaissance.....	23
Pour la lecture.....	25
RENDU, MATÉRIAUX, POST-PROCESS	26
Amélioration du précalcul de materials (expérimental).....	26
Composure Compositing Plugin.....	26
Material Shader Code.....	28
Brouillard volumétrique.....	28
Brouillard volumétrique.....	28
Éclairage précalculé sur le brouillard volumétrique.....	36
Opacité et masque d'opacité pour Material Flattening.....	37
Améliorations Mesh Editor Matériaux et Sections Panel.....	38
Outil Mesh Paint Tool amélioré.....	38
Detect Material on Mesh Surfaces.....	39
Curve Atlas dans les matériaux.....	39
Profondeur de haute qualité du terrain et Tone Mapping.....	40
Nouvel algorithme de profondeur de champ appelé "Circle DOF".....	40
Qualité supérieure du DOF.....	41
Cinematic Depth of Field.....	41
Support expérimental de Direct X 12.....	45
Area Shadows (stationary lights).....	46
Ambient Occlusion Material Mask.....	47
Amélioration de la Distance Field Ambient Occlusion.....	48
Mesh Distance Field Materials.....	50
Post Process Blending.....	51
Runtime Asset Cache.....	52
Volume Decals.....	53
Shading visage et vêtements.....	56
Shading réaliste des visages, cheveux, yeux, peau et tissus.....	56
Shading peau en haute-résolution.....	59
Dithered opacity mask.....	61
Capsules d'ombrage.....	62
Builds plus rapides (Intel embree support).....	64
Lightmass portals.....	64
Amélioration de la directionalité des skylights.....	66
Canaux d'éclairage.....	66
Éclairage indirect multi-rebond des puits de lumière.....	67
Rectangular Area Lights.....	68
Objets translucides.....	70
Per-vertex translucent lighting.....	70
Amélioration des lumières sur les objets translucides.....	71
Optimisations pour DIRECTX 12.....	72
Flux de développement sur GitHub.....	73
Metal remplace OpenGL pour Mac OSX.....	73
Réflexions.....	74
Réflexion planaire en temps réel.....	74
Réflexions de haute qualité.....	75
Réflexions: surfaces translucides.....	76
Mobile Skylight Reflections.....	77
Bent Normal Maps and Reflection Occlusion: amélioration de l'éclairage.....	78
Modèle d'éclairage spéculaire mobile amélioré.....	80
Ombrages avec plusieurs cartes normales.....	81
Écrire directement en Blueprint.....	82
La mise en cache de la carte des Ombres "SHADOW MAP CACHING".....	85

Unreal Engine 4 – Guide des mises à jour (4.20)

Mesh preview scenes.....	87
Mesh Decals.....	88
Le physical light units.....	89
Apparition des volumetric lightmaps.....	90
Nouveau moteur de rendu d'ombrages.....	93
Ombres de contact (Contact Shadows).....	94
Les scenarii d'éclairage.....	95
Amélioration de la qualité de la "Reflection Capture".....	97
Material attribute nodes.....	99
Vector Noise dans les matériaux.....	100
Améliorations du tonemapper.....	102
Amélioration de la gestion du streaming des textures.....	103
Material editor.....	104
Déplacement de nœuds dans le material editor.....	104
Support de prévisualisation avancé de Material Editor.....	104
Instance enfant ou fratrie dans l'éditeur de matériaux.....	105
Material Layering (expérimental).....	107
Sortie HDR (High Dynamic Range).....	107
Color Grading controls.....	109
Outils de visualisation de texture streaming debugging.....	110
Mathematics material nodes.....	110
Previous Frame Switch node.....	111
Pre-Skinned Local Normals material Node.....	112
Une nouvelle méthode de sur-échantillonnage temporel (Temporal Upsampling).....	112
ANIMATIONS / SEQUENCER.....	116
Amélioration du media framework.....	116
Improved Animation Preview Scenes.....	116
Morph Target Exageration.....	117
Création d'Animations à partir de la physique.....	118
Animations Multi-threaded.....	119
Configurable Animation Notify Triggers.....	119
Motion Blur et antialiasing pour l'animation de Vertex dans les matériaux.....	120
Interface de distribution séquentielle d'Element.....	121
Non-Linear Animations Blending.....	121
Contrôleurs Bone animations.....	123
Améliorations de la fonctionnalité "Two bones IK".....	125
Animation Blueprint.....	126
Amélioration des blueprints d'animation.....	126
Make Dynamic Additive Node pour Animation Blueprints.....	127
Live editing d'animation Blueprints (expérimental).....	127
Blueprint animation par défaut.....	128
Hordes de personnages avec Blueprint d'animation.....	129
Sequencer animations en Blueprints (expérimental).....	129
Animation Editing System.....	130
Baking Pose lors de la suppression de Bones.....	131
Partage du paramétrage LOD des skeletal mesh.....	132
Aperçu Additional Meshes avec CopyPose.....	133
Filtres des dossiers dans Anim Asset Browser.....	133
Règles de transition des animations.....	134
Meilleure gestion des assets de remplacement des animations.....	134
Jouer l'animation la plus pertinente.....	134
Mélange personnalisé des graphes d'animation.....	135
Animation Curve Evaluation Change.....	135
Animation Asset Metadata Support.....	136
Anim dynamics (simulation physique pour les personnages).....	136
Enregistrement d'animations depuis le GAMEPLAY.....	138
Animation posing en copiant à partir de meshes.....	139
Marked-based animation syncing.....	140
Curve Blending for Animation Montages.....	141
Animation Montage Time Stretch Curve.....	141
Sequencer.....	142
Apparition du Sequencer.....	142

Unreal Engine 4 – Guide des mises à jour (4.20)

Sequencer : cinématographie haut de gamme.....	144
Amélioration de la Camera Rig Crane.....	146
Enregistrer de l'audio dans une séquence.....	147
Sequencer animations blending.....	147
Sequencer events.....	148
Animer des objets dynamiques avec Sequencer.....	148
Amélioration de l'interface graphique de Sequencer.....	149
Shot Tracks.....	149
Les Animate Material Parameter Collections.....	149
Amélioration de l'Interface.....	150
Auto-key simplifié.....	150
Sequencer edits et Level Edits.....	151
Divers.....	151
Intégration dans le World Outliner.....	152
Mélanger les sections du sequencer.....	152
Image Plate Actor and Sequencer Track (Experimental).....	154
Améliorations du sequencer: imports FBX.....	154
Compiler à la volée.....	155
Dupliquer des pistes.....	155
Améliorations de Frame Accuracy pour Sequencer.....	156
Media Track for Sequencer.....	157
Sequencer Curve Editor et améliorations.....	158
Variables d'animation sur des instances anim dans Sequencer.....	159
Import / Export XML Final Cut Pro 7 dans Squenceur.....	160
Améliorations des enregistreurs de séquence.....	160
4.20.....	160
Level Sequence Dynamic Transform Origin.....	161
Sequencer Anim BP Weight Control.....	161
Twist Corrective node.....	162
Embedeed composite animations.....	163
Procedural Mesh slicing (TOP).....	163
Low level cloth simulation.....	164
Animations Vertex : l'importateur Alembic (expérimental).....	165
Streaming GeomCache et importateur Alembic amélioré (Expérimental).....	166
Animations des Visages.....	166
Animation pose assets.....	166
Animations faciales.....	167
Amélioration du partage d'animations faciales en utilisant des Curve only animations.....	168
Digital Human Improvements.....	168
Pose driver animation node.....	170
Animation node pose watching.....	171
Animation viewer curve.....	172
Éditeur d'animation Revamp.....	173
Fenêtre courbe d'animation.....	174
Modifier les nœuds des courbes d'animation.....	174
Relier des courbes aux bones.....	175
Pre-skinned local position.....	175
Améliorations du nœud d'animation RigidBody.....	177
mouvements en "Local Space".....	177
Os virtuels.....	178
Corps physiques avec des parents simulés.....	178
Amélioration pose driver.....	179
Child Animation Montages.....	180
Morph Target debug View Mode.....	181
Pose snapshotting.....	182
Amélioration 'Look At' Animation Node.....	182
Spline IK Solver.....	183
Amélioration de l'Animation export.....	183
Amélioration: éditeur Animation Blend Space.....	184
Animation Modifiers (Early Access Preview).....	184
Amélioration des outils d'animation.....	185

PHYSIQUE ET PARTICULES.....	186
-----------------------------	-----

Unreal Engine 4 – Guide des mises à jour (4.20)

Éditeur d'effets visuels Niagara vient compléter (et remplacer à terme) Cascade ! (Early Access).....	186
Améliorations de la conception et de la création d'effets.....	187
Interface utilisateur mise à jour.....	188
Nouveaux modules.....	189
Simulation GPU.....	189
Simulation CPU.....	190
Amélioration automatique Convex Collision Meshes.....	192
Full Scene Particle Collision avec Mesh Distance Fields.....	192
Actor Encroachment Detection.....	194
Profondeur de champ de particules.....	195
Découpages de particules (flipbook rapide des particules pour le rendu).....	196
Physical component animations.....	197
Upgrade de PhysX en 3.4.....	198
Optimisations PhysX.....	198
Nouvel éditeur d'assets physiques.....	199
Prise en charge des véhicules améliorée.....	200
Mass properties debugging.....	201
Le support de véhicules est déplacé sous forme de plugin.....	202
Outils Vêtements - Cloth.....	203
Amélioration du cloth skinning.....	203
"clothing Paint Tool" (Experimental).....	204
Outil vêtement mis en place.....	204
Anim Drive sur l'outil Cloth.....	205
Améliorations au niveau des vêtements.....	206

PROGRAMMATION BLUEPRINTS ET C++.....207

Blueprint Merge Tool.....	207
Blueprintable Components.....	208
Full Engine C++ Source Code.....	209
C++ Classes dans le Content Browser.....	210
Visual Logger (AI Visualization Tool).....	211
Mode Debug.....	212
Blueprint Watch Window.....	213
Problème des Blueprint Cycliques.....	214
Améliorations concernant l'Unreal C++.....	214
Nœuds de retour multiples dans les Blueprints.....	215
Construire des objets personnalisés dans Blueprints.....	216
Valeur par défaut de la classe Blueprint.....	216
Level Blueprint Communication.....	217
Blueprint Asset Ids.....	218
Actor Tick Intervals.....	218
Support Visual Studio.....	219
Automatic Blueprint Node Arrangement.....	221
Recherche de Blueprints.....	222
L'inspecteur de Pixels.....	223
Blueprints en C++.....	223
Integer vers Bitmask.....	224
Nouveau: line trace renvoie les coordonnées de texture.....	225
Amélioration de l'édition spline.....	226
Sous-animation Blueprint.....	227
Amélioration de l'accès pour les scripts aux Static Meshes pour les LOD et les collisions.....	228
Cooking blueprints en C++.....	229
Compilateur C++ plus rapide.....	229
Stockage de packs d'objets en Blueprint.....	231
Array Reordering.....	233
Play montage.....	234
Analyse statique avec PVS-Studio.....	235
Blueprint Compilation Manager.....	235
Registre des assets dans Blueprints.....	235
Signets Blueprint.....	236
Blueprint Property Accessors.....	236
Nouveau: Orphan Pins.....	237
Nouveau: Sobol Blueprint and Material Nodes.....	238

Unreal Engine 4 – Guide des mises à jour (4.20)

Texture-based Importance Sampling Blueprint nodes.....	240
PAYSAGE / FOLIAGE / GÉNÉRATION PROCÉDURALE.....	242
Realistic Foliage Lighting.....	242
Pour des mondes gigantesques (Open Worlds).....	243
Rendu de l'herbe pour le paysage.....	243
Générateur procédural de Feuillage (expérimental) et Refonte.....	245
Niveau hiérarchique du Système LOD.....	247
Procedural Mesh Component (expérimental).....	248
LOD basé sur la taille de l'écran.....	249
PROXY LOD.....	250
Expérimental.....	250
Proxy LOD prêt pour la production.....	251
Combler les "vides".....	252
LOD hiérarchique - Prévisualisation rapide et mise en cluster.....	253
Contrôle Normal Amélioré: Fendu Hard Edge Normal.....	254
LOD Clustering personnalisable (HLOD Volumes).....	255
Landscape Mirror Tool.....	256
Dithered LOD Crossfades.....	256
Improved Hierarchical LOD (HLOD).....	258
LOD et réduction de Bones !.....	259
Herbe et feuillage évolutifs.....	260
Collisions.....	261
Selective LOD pour collision mesh.....	261
Génération automatique de LOD.....	262
Auto Compute LOD Distances.....	263
Outil de rotation de paysages.....	264
MENUS ET WIDGETS (UMG).....	265
UMG / Slate Clipping 2.0.....	265
Améliorations de la zone de sécurité UMG.....	265
UI.....	267
UI: Curseur Personnalisé.....	267
UI: interface utilisateur améliorée.....	268
Unreal Motion Graphics: le nouveau Widget Binding System.....	268
Curve Editing pour l'interface utilisateur Unreal Motion Graphics.....	269
Widgets.....	269
Fast CreateWidget.....	269
Widget Depth Ordering.....	270
Widget interaction component.....	270
Sprites in UMG Widgets.....	271
Fontes en outline.....	272
Interface Blur rects.....	273
Mise en place d'un asset store pour les fontes.....	274
RÉSEAU & MULTIJOUEURS EN LIGNE.....	275
Replays des jeux en réseau.....	275
Répliquations.....	276
Character movement speed hack protection.....	276
SONS ET MUSIQUES.....	277
Moteur Audio d'origine.....	277
Niveaux de qualité sonore.....	277
Courbes d'atténuation personnalisées pour l'audio.....	278
Décompression audio Asynchrone Real-time.....	279
Spatialisation stereo.....	279
Sound focus.....	280
Sound occlusion.....	281
Sound concurrency.....	282
UE4 Sound Mixes.....	282
Audio localization.....	283
Nouveau plugin MIDI.....	284

Unreal Engine 4 – Guide des mises à jour (4.20)

Pour le séquencer.....	284
Stereo Layers Unified Across Platforms.....	284
Tirage au sort de l'ordre des layers.....	285
Steam Audio.....	285
Early Access.....	285
Steam Audio Beta 10.....	285
Nouveau moteur Audio.....	286
Unreal Engine Audio (Early Access Preview).....	286
Synthesis Plugin (Early Access).....	286
Améliorations de l'architecture de plug-in audio.....	287
Source Bus.....	287
Nouveaux effets de sous-mixage.....	288
Ajout du plugin audio Resonance.....	288
Fichiers Ambisonics.....	288
Améliorations VOIP native de UE4.....	289
Nouveau composant de capture de microphone.....	289
Submix Audio Recording.....	289
PAPER 2D.....	291
Tilesmap et Tileset.....	291
Amélioration de l'édition de Layer Tile Map.....	291
TileMaps plus accessibles.....	292
Flipbooks.....	293
Collision Detection.....	293
Sprites.....	294
Peindre directement sur des Sprite.....	294
Support de multiples textures.....	294
Interactive Extract Sprites Dialog.....	295
Grouped Sprite Components.....	296
Pas d'évolution?.....	296
RÉALITÉ AUGMENTÉE.....	298
De la création de spectacles à la mise en scène pour le e-sport !.....	298
Capture de réalité mixte (accès anticipé).....	298
nDisplay Flexible, Rendu multi-écrans.....	299
Magic Leap One™ Early Access Support.....	300
ARKit (iOS) Support.....	301
Experimental.....	301
Apple ARKit iOS 11.....	302
Apple ARKit 2.0 Support.....	303
ARCore (Android) Support.....	303
Google Plugin pour Tango UE4 (expérimental).....	303
Prise en charge de Google ARCore Developer Preview.....	304
Un Framework unifié pour la réalité augmentée.....	306
Google ARCore 1.2 Support.....	307
RÉALITÉ VIRTUELLE.....	308
Support SteamVR / HTC Vive.....	309
Prise en charge de SteamVR sur Mac.....	310
Support Vive Pro.....	310
Playstation VR.....	311
PlayStation 4 Morpheus VR avec 120Hz Reprojection.....	311
Support du contrôleur Playstation VR.....	312
Social Screens (Early Access Preview).....	312
Direct Multiview Support pour Daydream.....	312
Support de la plate-forme Daydream.....	313
Support OSVR.....	313
VR Motion Controller Support.....	314
VR Motion Controller Support.....	314
Live Link Motion Controller Assistance.....	315
Visualisation des composants de Motion Controller.....	315
Low-latency VR Motion Controllers.....	316
Physic simulation en VR.....	316

Unreal Engine 4 – Guide des mises à jour (4.20)

Rendu VR.....	317
VR rendu stereo.....	318
Améliorations VR head mounted display camera.....	318
VR stereo layers.....	319
VR chargement de films.....	319
Champ éloigné en vision monoscopique pour mobile VR.....	319
Amélioration du rendu VR sur mobile.....	320
Editeur VR.....	321
Peindre directement en VR.....	323
La mise en place d'arbres en automatique en VR.....	324
Color picker en VR.....	325
Editeur de paysage en VR.....	326
Menu VR editor amélioré.....	326
VR Editor Numpad.....	327
Support écran VR Spectator pour Oculus, Vive et amélioration pour PSVR.....	327
VR mode camera Previews.....	329
INTELLIGENCE ARTIFICIELLE.....	330
AI Perception System: Blueprint Scripting.....	330
Dynamic Navigation Mesh pour mondes ouverts.....	331
AI: Support Blueprint Support pour EQS.....	332
Améliorations globales.....	332
MOBILES, CONSOLES ET HTML5.....	333
HTML5/WebGL.....	333
Support Expérimental (Windows seulement).....	333
Amélioration de HTML5.....	334
Android.....	335
Développement Android sur Mac.....	335
Desktop-Quality Rendering on Android (OpenGL ES 3.1 w/ AEP).....	336
Loading movies et startup movies fonctionne désormais pour Android (à partir de la version 4).....	337
Support de Contenu téléchargeable (DLC).....	338
Support de contenu pour les patches.....	338
Support pour les grands Jeux Android.....	338
Manettes Gamepad pour Android.....	339
Widget Web Browser sur Android.....	340
Test automation for android.....	340
Vulkan.....	341
Support Android sur linux.....	342
Android graphics debugueur.....	342
Virtual Keyboards.....	343
Support for Runtime Permissions on Android.....	344
Android Toolchain et SDK mis à jour.....	345
Réduction de la taille de l'exé Mobile sur iOS et Android.....	345
Mobile Separate Translucency.....	346
sRGB sur Android haut de gamme.....	346
Android Aspect Ratio Support ultra-grand écran.....	347
Ajout du support audio Unreal sur de nouvelles plateformes (Early Access).....	347
Gradle Android.....	348
Plugin de caméra Android.....	348
iOS.....	349
Faire des projets iOS sur Windows.....	349
Révision du projet Xcode.....	349
Prise en charge améliorée de Metal sur macOS et iOS.....	350
Streaming audio sous ios.....	351
Notifications à distance pour ios.....	351
Simplification des certificats et signatures sur iOS.....	351
Desktop-Class Forward Renderer sur iOS (expérimental).....	351
signature automatique de code.....	352
Tous mobiles confondus.....	352
Ombres dynamiques sur personnages pour mobiles.....	353
Dynamic Point Lights pour Mobile.....	354
Traces (decals) sur mobiles.....	354

Unreal Engine 4 – Guide des mises à jour (4.20)

Effets Refraction pour mobiles.....	355
Material Quality system.....	356
Mobile Provisioning Profile Selector.....	358
Post traitement de haute qualité.....	358
Amélioration des ombrages pour les mobiles.....	359
Les particules.....	360
Assistant packaging pour mobiles.....	360
Custom Post-Process materials.....	361
Lighting channels sur Mobiles.....	361
Support de profondeur personnalisée sur mobiles.....	362
Bibliothèque blueprint pour mobiles.....	363
Amélioration du composant Scene Capture.....	363
Aperçu de rendu (opengl es 3.1, metal, vulkan).....	364
Options mobile shader de réduction de la mémoire.....	365
Service de localisation pour les données GPS sur Android et ios.....	365
Support pour Nvidia SLI Alternate Frame Rendering (AFR).....	365
Custom Stencil sur mobiles.....	365
Résolution dynamique.....	366
Synchronisation de frame à faible latence.....	367
Hardware and Software Occlusion Queries.....	367
Platform Material Stats.....	368
Amélioration du débogage sur Android.....	369
Amélioration du mode paysage sur mobile.....	369
Améliorations mobiles diverses.....	370
Les SDK des différentes plates-formes ont été mises à jour.....	371
Support switch Nintendo.....	373
Support Linux ARM64.....	375

ÉDITEUR ET TEMPLATES..... 376

In-World Component Editing.....	376
Feature Packs and Content Packs.....	376
Tutorials for New Project Templates.....	378
New Thumbnail Look and Feel.....	378
Project Launcher.....	379
Localization Dashboard.....	379
Nouvel Exemple de personnage Unreal.....	381
Amélioration de Animation Retarget Manager.....	382
Asset size Mapper.....	383
Assistant de création de Plugin.....	383
Navigateur de contenu.....	384
Recherche avancée.....	384
Filtre pour trouver les assets récents dans le navigateur de contenu.....	385
Amélioration des collections.....	386
Collections imbriquées.....	386
Collections dynamiques.....	387
Quick Asset Management.....	387
Nouveau: Asset Management Framework Production.....	387
Asset Management Framework (Early Access Preview).....	389
Asset Audit Window (Early Access Preview).....	389
Amélioration des rapports d'état.....	390
Amélioration du curve editor: éditeur de courbes.....	390
Performance et multithreading.....	391
Garbage collector.....	393
Éditeur de lecteur média.....	394
Création de Static Mesh à partir de n'importe quel actor.....	395
Child actor templates.....	395
Amélioration du composant Câble.....	396
Live GPU profiler.....	397
Le Blendspace Editor.....	398
Nouveau look et optimisation le workflow.....	398
Améliorations de la convivialité de l'éditeur Blendspace.....	399
Workflow.....	399
dossiers favoris.....	399

Unreal Engine 4 – Guide des mises à jour (4.20)

Optimisation de l'espace disque.....	400
Étiqueter les couleurs enregistrées dans le sélecteur de couleurs.....	400
Scripted Extensions for Actor and Content Browser Context Menu.....	401
Encryption/Signing Key Generator.....	401
Mesh Description Format.....	403
PLUGINS ET LOGICIELS TIERS.....	405
Authentication Steam.....	405
Shotgun Integration (Early Access).....	405
Importation en général.....	407
Importation Automatique des Assets.....	407
OpenEXR: Textures High Dynamic Range.....	407
Full Scene Importation.....	408
Actor fusion.....	408
Leap Motion.....	409
Autodesk Maya.....	410
Live Link Maya.....	410
Nvidia Ansel.....	411
Versioning et Travail collaboratif.....	412
Git Source Control Plugin (Beta).....	412
ArchViz.....	413
Character Controls.....	413
Technologie DataSmith.....	413
Pour les plugins.....	415
Global Shader Support pour Plugins.....	415
Les dépendances entre les plug-ins.....	416
Replication Driver / Replication Graph.....	417
Expérimental.....	417
CONCLUSION.....	418
Le mot de la fin.....	418
Livres en français sur Unreal engine 4.....	420
Tome 1.....	420
Tome 2.....	423
Tome 3.....	426
Tome 4.....	428
Tome 5.....	430
Conseils de lecture.....	433
Remerciements.....	435

A propos de cette édition

J'ai procédé à une refonte partielle, un regroupement des thèmes a été réalisé, mais il est encore en cours.

Il y a encore un gros travail à réaliser pour regrouper les thèmes et faire un peu de nettoyage dans les mises à jour plus anciennes.

Certaines sections ont une chronologie un peu anarchique, c'est dû au regroupement qui est en cours, veuillez m'en excuser, ce document est en constante évolution.

Plusieurs vidéos ont été ajoutées avec un logo facilement identifiable, il suffira de cliquer dessus pour avoir ainsi plus d'infos en vidéo.

Merci pour vos [nombreux retours positifs](#) qui m'encouragent à mettre à jour ce livre régulièrement.

POURQUOI CE LIVRE?

Présentation d'Unreal Engine



Unreal Engine est certainement le moteur de 3D temps-réel le plus abouti de notre époque. Il est accompagné d'un **éditeur** à la fois **simple** et **puissant**, qui le rend accessible à un public non-développeur: artistes en tous genres, architectes et ingénieurs, étudiants. A l'origine, il était orienté vers les **grosses productions de jeux vidéo**. Mais il s'est doté dernièrement d'un **nouveau système de licence** très attractif qui le rend accessible à tous, pour des projets qui dépassent de loin le secteur du jeu: présentations interactives temps-réel, cinématiques, ... là où il y a une histoire à raconter, un produit ou un concept à présenter, voir un concert «Next Gen» dans un univers virtuel en utilisant un casque de réalité virtuelle !

Unreal Engine est composé d'un éditeur offrant tous les outils nécessaires à la création d'un jeu vidéo moderne, ainsi qu'un moteur de jeu redistribuable permettant de faire tourner ce jeu sur de nombreuses plates-formes: ordinateurs de bureau (Windows, Linux, OSX), mobiles (Android, iOS), navigateurs web (WebGL/HTML5), ainsi que consoles (Microsoft XBOX, Sony Playstation) et même Oculus Rift.

Quand j'ai commencé à écrire les premiers tomes de la série des cahiers d'Unreal Engine, je savais que progressivement des moteurs de jeux comme Unity, CryEngine ou le Source Engine de Valve allait

Unreal Engine 4 – Guide des mises à jour (4.20)

progressivement céder leur place à cet outil puissant, à moins de se spécialiser dans des secteurs particuliers.

Je savais aussi que l'arrivée de la réalité virtuelle et de la réalité augmentée allaient créer une forte demande en direction de moteurs robustes et doté d'un rendu exceptionnel.

Toutefois, ce qui m'avait échappé à l'époque, c'est qu'un moteur de jeu est avant tout un **moteur de 3D temps-réel** et que **l'industrie, le secteur du marketing, de l'architecture, de la santé...** bref, quasiment tous les secteurs de l'économie allaient vouloir créer des applications et des outils de présentation de leurs produits et services en utilisant la 3D temps-réel, comme fin des années 90 elles s'engouffraient dans les vitrines web.

Le 22 Avril 2016 fut l'occasion d'un twitch d'Epic sur Unreal Engine et l'entreprise. Il fait suite à cette présentation du configurateur de McLaren présenté à la GDC: Le résultat est magnifique – je comprends que les grandes marques s'orientent de plus en plus vers le temps réel vu le rendu ! Et cela ouvre tellement d'interactions possibles avec le public !

Il y a une sorte d'**échange transversal** qui s'opère progressivement entre le monde du jeu vidéo et celui de l'industrie, des marques, etc. Et pas seulement dans un sens, c'est un peu là la nouveauté. Ce qui est développé d'un côté est utile aux autres et vice et versa. N'oublions pas qu'UE4 est **open source** et que les nombreuses features que l'on voit à chaque version apparaître proviennent de la communauté. Et cette communauté n'est pas composée uniquement de barbus adeptes du libre (d'ailleurs UE4 n'est pas libre) – mais des ingénieurs, des industriels, de gros organismes comme l'aérospatial... tous ces acteurs intègrent leurs technologies dans le moteur pour mieux en profiter après: **c'est du gagnant-gagnant pour tous.**

Unreal Engine possède une **documentation abondante**, voir impressionnante. De très nombreux sujets y sont traités, mais cette pléthore d'informations rend aussi la découverte de l'outil très difficile, tant **la couverture fonctionnelle** du produit **semble interminable**. Et quand on se penche sur un thème précis, on se retrouve inévitable-

Unreal Engine 4 – Guide des mises à jour (4.20)

ment en manque d'informations, ce qui peut paraître paradoxal. C'est pourquoi plusieurs tutoriels vidéo ont été réalisés et l'apprentissage de l'outil passe inévitablement par l'étude de ces derniers. Il est aussi nécessaire d'étudier les projets fournis en exemple. Toutefois, il faut déjà un certain niveau de connaissances pour pouvoir profiter de ces exemples. *«Autant ouvrir un livre écrit en latin en espérant pouvoir acquérir les bases du langage par la pratique!»*

C'est la raison principale qui m'a conduit à écrire des livres sur Unreal Engine 4: d'une part, c'était une façon pour moi d'assurer une veille du produit, de regrouper un certain nombre de recettes de base que j'apprenais ou que je développais au fur et à mesure.



Et je crois que la mission est remplie: je pense qu'avec les 3 premiers tomes, nous couvrons un enseignement "de base" (avec parfois des recettes avancées issues de ma propre expérience). L'approche est "progressive" et "pratique" – pas facile à décrire, mais l'important c'est que c'est mon approche personnelle et qu'elle semble correspondre à une majorité de lecteurs.

Bon, l'objectif de ce livre n'est pas de faire de la publicité pour mes cahiers, car la plupart d'entre vous sont déjà des lecteurs.

Unreal Engine 4 – Guide des mises à jour (4.20)

Non, je voulais attirer votre attention sur un élément important: **Unreal Engine est constamment en évolution** – c'est un monstre de fonctionnalités, et parfois un abîme pour ceux qui se lancent dans l'aventure. Une fois lus mes livres sur ce moteur, il va vous falloir travailler dur pour assurer une veille du produit, tester les nouvelles fonctionnalités, lire les tutoriels suivants. Et cela va à un rythme infernal au point où un expert pourrait passer la moitié de son temps à se former. Pour ma part, je suis abonné à des dizaines de flux RSS sur UE4, je surveille toutes les vidéos de tutos qui sortent, je lis tous les livres sur le sujet, je consulte presque tous les posts sur le forum d'Unreal Engine... Mais c'est normal, et cela doit être ainsi: j'ai choisi de me spécialiser sur ce moteur et je dois consacrer une bonne partie de mon temps à faire de la recherche et de la formation.

De Unreal Engine 1.0 à 4.20



Vous vous souvenez probablement du premier jeu **"Unreal"** d'Epic Games. On peut dire qu'Unreal Engine est né à peu près à cette époque. L'histoire commence par James Schmalz, fondateur de Digital Extremes, un studio de jeux vidéo canadien qui fait une démonstration de son dernier prototype à Tim Sweeney, le PDG et fondateur d'Epic Games. Il s'agissait de la visite d'un château médiéval en 3D. Sweeney se montre alors très intéressé par son projet et se joint à lui pour la programmation du moteur. Il va s'occuper de la partie éditeur de niveau, d'implémenter un moteur de physique et va s'occuper aussi des optimisations concernant le rendu. C'était l'époque de **Doom**, créé par John Carmack, une référence en ma-

Unreal Engine 4 – Guide des mises à jour (4.20)

tière de jeux 3D. Tim, grand admirateur de Carmack, décide de pousser encore plus loin le moteur et s'engage dans une course aux améliorations technologiques... Après 18 mois de travail acharné et sept refontes du code source, il parvient enfin à obtenir une version stable et performante du moteur. Réalisant les possibilités offertes par cette technologie, Schmalz et Sweeney décident dès 1994 de l'utiliser pour créer un jeu vidéo qu'ils baptisent Unreal et qui donnera son nom au moteur de jeu.

Voilà pour la petite histoire, mais c'est certainement avec la version 3 que le moteur commence véritablement à obtenir ses lettres de noblesses et se fait connaître du grand public avec l'**Unreal Development Kit (UDK)**. Il s'agit d'un kit de développement de jeux vidéo, basé sur Unreal Engine 3, introduisant déjà une notion de ce que seront plus tard les Blueprints au travers de Kismet, possédant un éditeur de cinématique qu'il s'appelaient déjà Matinee...

Bref, l'outil était puissant et si vous avez déjà joué à **Life Is Strange** qui a été conçu avec UE3, vous avez remarqué que l'on peut déjà produire un jeu très tendance aux graphismes excellents. Oui, mais non – Tim n'était pas entièrement satisfait – le sera-t-il un jour? Je crois qu'il est au monde du virtuel et de la 3D Temps-réel ce que Musk est à la voiture électrique ou au voyage sur Mars: un précurseur, toujours féru de progrès, d'améliorations ... toujours l'esprit 10 ans minimum dans le futur à prévoir pour devenir un acteur du changement. Non, je n'ai pas cité Steve Jobs qui n'est pas pour moi de la même trempe – même s'il faut lui reconnaître un talent tout particulier pour faire adopter par le grand public des technologies existantes qui étaient restées sur le banc de touche. Je pense à l'ipod, à la tablette ... Bref, je m'égare.

En août 2005, alors qu'aucun jeu utilisant Unreal Engine 3 n'était encore sorti, Epic Games annonce que l'équipe travaillait depuis deux ans sur la version 4 ! Ce n'est pas très marketing tout ça... ce moteur graphique cible la huitième génération de consoles. Et devinez: la seule personne à travailler sur le code source de l'Unreal Engine 4 était Tim Sweeney ! Et ce n'est qu'en Mai 2012 que la toute première version stable du moteur sortira, la 4.0! La version 4.1 mettra quant à elle près de 2 ans à arriver, en Avril 2014. Puis la 4.2, 2 mois après...

Unreal Engine 4 – Guide des mises à jour (4.20)

Et depuis, on enchaîne les versions au rythme d'une version tous les 2 mois environ.

Mais au-delà de ce rythme, c'est surtout **le passage en 2015 à l'Open Source qui a fait que la communauté grandissante de contributeurs a permis au moteur de passer à une vitesse supraluminique!** (oui, j'aime bien les formulations un peu dithyrambiques).

Epic a annoncé **la gratuité** de son éditeur lors de la Game Developer Conference qui s'est déroulée du 2 au 6 mars 2015 à San Francisco. Avant, les développeurs devaient souscrire un abonnement mensuel de 19\$/mois, cela a duré moins d'un an. Auparavant, il était payant et seules les grosses sociétés pouvaient se le permettre. C'était déjà une première approche réussie vers la démocratisation de l'outil. Depuis l'annonce par Epic, plusieurs autres éditeurs ont annoncé le passage à la gratuité également: Unity 5 qui n'est «quasiment» pas bridé (à part le splashscreen) dans la «personnal edition» et Source 2 de Valve. Une bataille fait rage chez les éditeurs et les gagnants sont les studios de création de jeux vidéo qui vont pouvoir proposer des jeux de qualité AAA sans devoir investir de grosses sommes dans l'outil de développement.

Tim Sweeney explique ce passage au gratuit «par une volonté de **libérer la créativité** et de permettre au jeu vidéo, en tant qu'art, de continuer à se développer». Il résume cette opération par «*If you love something, set it free*» (littéralement, «si vous aimez quelque chose, rendez-le libre»).

Alors voilà, j'aurais pu vous décrire toutes les évolutions du moteur depuis sa création, au moins en version 4 car il a été entièrement réécrit. D'ailleurs, certaines fonctions existant sous UE3 ont mis beaucoup de temps à apparaître sous UE4... c'est ainsi !

Mais je pense que les évolutions les plus importantes se produisent depuis 2015. Et puis, si vous avez étudié les premiers tomes des cahiers d'Unreal Engine qui sont basés sur la 4.7, ça vous mettra vos informations à jour, même s'il vous faudra un travail personnel pour acquérir la maîtrise de ces nouvelles fonctionnalités.

Unreal Engine 4 – Guide des mises à jour (4.20)

Juste pour la culture, voici quelques éléments à retenir:

- **4.1:** support de la Playstation 4, de la Xbox one, de SteamOS et de Linux. C'est à cette période qu'Epic a introduit la fantastique "Elemental Demo". Apparaît également un éditeur de localisation pour les jeux. Nouveau template pour ce qui sera plus tard le 6DOF (déplacement dans les 6 directions de l'espace).
- **4.2:** Nouveau template pour la gestion de véhicules, et notamment au niveau physique avec de nombreuses courbes de paramétrage. Reprise du système d'animation de caméra d'UE3 avec un contrôle via Matinee et les Blueprints. Apparitions de la "structure" dans les Blueprints (enfin !) et la possibilité de créer des bibliothèques de fonctions personnalisées. FABRIK: Nouveau solveur de cinématique inverse fonctionnant sur les bones dans les Blueprints d'animations. Nouveau nœud permettant d'entrer des expressions mathématiques telles qu'elles dans les Blueprints. Ajout de fonctions pour le débogage d'animations. Le support du FBX passe en version 2014.
- **4.3:** plus de 500 mises à jours, mais les plus importantes sont: l'amélioration du rendu (Skylights, Distance Field AO), l'intégration du plugin Paper 2D comme système principal pour les animations 2D, les Behaviour Trees comme arbres décisionnels pour l'Intelligence Artificielle, la gestion des courbes de beziers directement sous l'éditeur mais aussi dans les Blueprints, la possibilité de paramétrer les meshes de collision directement sous l'éditeur,... Bref, du lourd aussi dans cette mise à jour.
- **4.4:** Apparition de l'UMG, le nouveau système de gestion des widgets et de menus, fonctionnant à la fois en 2D et en 3D. Amélioration de nouvelles fonctionnalités présentées précédemment, dont l'éditeur de behaviour trees.
- **4.5:** Beaucoup d'améliorations et d'optimisations au niveau du rendu, avec l'apparition des Screen-Space Subsurface Scattering (simuler la "transparence" de la peau par exemple). L'éditeur de widgets est grandement amélioré avec de nouveaux types. Une meilleure intégration du C++ sans avoir à

Unreal Engine 4 – Guide des mises à jour (4.20)

refermer l'éditeur après avoir compilé de nouveau un objet. Ajout d'un système de retargeting d'animation fort utile quand on veut adapter une animation à plusieurs squelettes différents. Ajout de la possibilité de lire des vidéos dans les textures (avec quelques lacunes encore aujourd'hui d'ailleurs).

- **4.6:** Comme toujours, des améliorations au niveau du rendu, mais pas seulement: la possibilité d'enregistrer des parties en réseau pour faire un replay, la prévisualisation sous l'éditeur comme sur le mobile avec le support d'OpenGL ES2, un nouveau widget permettant d'afficher le contenu d'une page web et d'interagir avec, la gestion de sessions Steam pour le multijoueur, la possibilité d'utiliser jusqu'à 128 textures dans un seul material, amélioration du système d'animations notamment avec le Root Motion, etc.
- **4.20 :** Vous pouvez maintenant construire des personnages numériques réalistes et des mondes crédibles avec un réalisme inégalé et en prime utiliser la nouvelle technologie Digital Humans qui alimente les démos «Meet Mike» et «Siren».

<https://youtu.be/9owTAISsvwk>

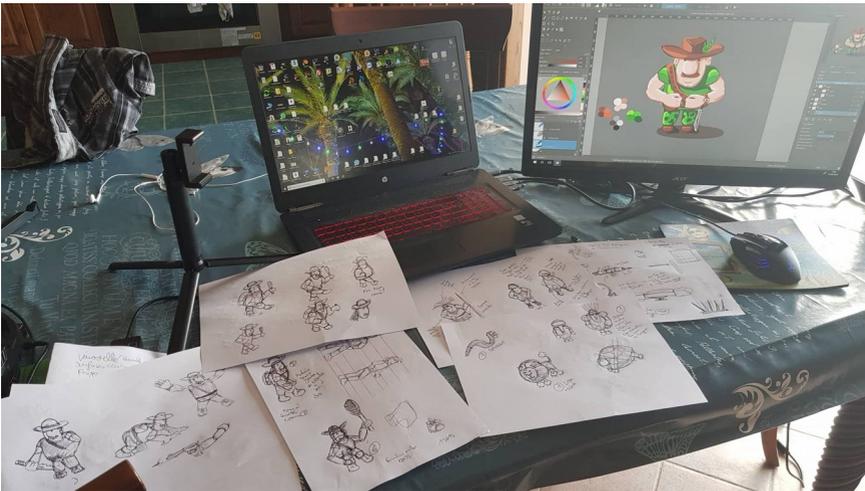
Vous pourrez donc créer des personnages, des environnements réalistes que vous pourrez utiliser dans différents domaines comme les jeux, la réalité virtuelle, la réalité augmentée, la réalité mixte, le cinéma ou la télévision ou encore pour des applications d'entreprise. Avec les nouvelles bibliothèques d'édition de scripts et d'automatisation, vous pouvez vous créer des outils personnalisés et développer vos propres workflows. De nombreuses modifications ont été apportées pour la création sur mobiles grâce à **Fortnite**, le jeu créé par Epic Games et dont le dernier tournoi a [fait parler de lui](#).

J'avais déjà traduit sur mon blog les différentes mises à jour du moteur, les "release notes" d'Epic Games. Ici, j'ai eu envie de vous compiler tout cela, en l'organisant en plusieurs sections et en présentant d'une autre façon ces dernières. J'y ai aussi inséré plusieurs notes personnelles, des réflexions, des choses qui pourront vous être utiles.

Unreal Engine 4 – Guide des mises à jour (4.20)

Je n'envisage pas dans l'immédiat de mise à jour des premiers tomes des cahiers car ils n'ont qu'un an (version 4.7 et 4.10) et que les informations qu'ils contiennent sont valables à 99%. Si je les met à jour à terme, ce sera surtout pour ajouter de nouvelles choses, mais si le format livre est un bon support pour commencer sa formation au moteur, il n'est pas le meilleur vecteur pour assurer une veille technologique dans la durée. Et les tutoriels en vidéo restent souvent hésitantes, manquent d'explication, et dès que l'interface change, c'est l'horreur !

Je vous annonce également la sortie prochaine d'une série d'ateliers (en vidéo en grande partie) sur la création de jeux vidéo, sur l'archviz, la création d'applications de réalité virtuelle et augmentée, le tout autour d'Unreal Engine.



En attendant, vous pouvez suivre le devlog du jeu créé dans le cadre d'une formation sur la création de jeux 2D avec Unreal Engine, à partir de [ma chaine youtube](#).

Faisons un peu connaissance



Avant de passer à la présentation des nouvelles fonctions, pour ceux qui ne me connaissent pas, faisons un peu connaissance. Je m'appelle Greg (mon nom complet est sur la couverture), et je programme depuis l'âge de 8 ans... et j'en ai... bref, ça fait plus de 30 ans et je n'ai jamais arrêté. Je resterai probablement un éternel drogué... mais vu l'âge moyen de mes lecteurs, je ne suis pas le seul dans ce cas il semblerait.

Je ne vais pas vous raconter ma vie car ce n'est pas l'objet de ce livre, mais ce que je peux vous dire c'est que j'ai passé une grande partie de cette dernière à m'intéresser et à concevoir divers moteurs 3D pour le jeu vidéo. Mon premier moteur était en turbo pascal... mon second en assembleur 386, et les suivants se basaient sur OpenGL et directX. J'ai réalisé [mes premiers jeux](#) édités aux USA quand j'étais étudiant. J'ai travaillé dans plusieurs sociétés de jeux vidéo, mais mon parcours est vraiment atypique de ce côté là – je m'intéressais à trop de choses au final.

Bref, avant d'étudier en détail l'UDK, j'étais plutôt orienté Ogre3D, Crystal Space, Irrlicht et Panda3D. Ce dernier me plaisait beaucoup car il était très simple d'utilisation et on pouvait faire facilement un jeu avec un seul script! Le Game Engine de Blender me plaît beaucoup également de par sa proximité avec Blender et du fait qu'il constitue

Unreal Engine 4 – Guide des mises à jour (4.20)

pour moi le meilleur point d'entrée pour comprendre comment fonctionne un moteur de jeu, et plus particulièrement un moteur 3D. Mais c'est une autre histoire également.

Et puis est venu la version 4 d'Unreal Engine et là ce fut un **coup de foudre** immédiat! Depuis, je n'ai plus jamais aligné de ligne de code pour concevoir un moteur 3D. UE4 répondait à toutes mes attentes et bien au-delà. Et lorsque je rencontrais un manque, alors son côté Open Source faisait que je pouvais le faire évoluer!

Bref, je me suis spécialisé dans ce moteur, j'ai écrit des livres sur le sujet, j'y consacre une partie de mon blog, de réalise des formations dans le monde francophone (France, Canada, Suisse et Belgique principalement) et je collabore étroitement avec des studios de Jeux vidéo pour la création de certains modules et parfois même pour du Gameplay, histoire de rester en contact avec le terrain.

J'ai eu l'immense honneur de remporter en 2016 [un prix de 8000\\$ pour mes travaux](#) sur Unreal Engine lors des **Unreal Educational Grants** organisés par Epic Games. Je tiens d'ailleurs à remercier à nouveau Epic Games pour leur confiance et leur soutien, et plus particulièrement Tim Sweeney qui reste à l'écoute des autres développeurs et avec qui j'ai pu discuter plusieurs fois. C'est admirable qu'un homme de sa trempe se rende à ce point accessible.

Voilà, vous savez le principal de moi, le reste ne vous intéresse probablement pas.

Maintenant que vous me connaissez un peu plus, j'aurais aimé en savoir un peu plus sur vous, vos compétences en 3D, vos motivations, vos attentes... Bref, si vous voulez bien prendre 5 minutes pour m'envoyer un petit email à greg@benicourt.com et vous présenter, cela récompensera mon travail pour créer ce livre, vous l'offrir et le mettre à jour régulièrement.

Pour me suivre:

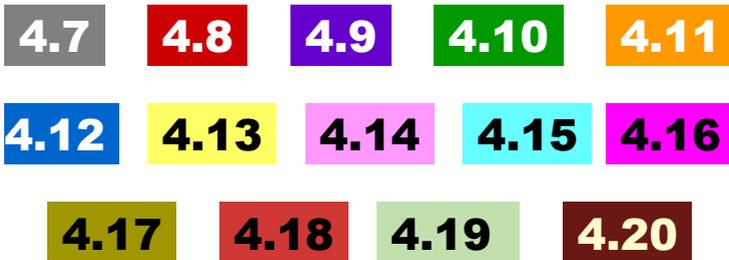
- Mon blog www.benicourt.com
- Mon compte [Facebook](#), [Twitter](#) et [Google+](#)

Unreal Engine 4 – Guide des mises à jour (4.20)

- [Contact pro](#) via LinkedIn
- Ma chaîne [Youtube](#)

Pour la lecture

Les versions sont indiquées en respectant un code couleur :



En fonction de la version à laquelle vous vous êtes arrêtés, cela peut vous permettre d'aller rapidement à l'essentiel.

RENDU, MATÉRIAUX, POST-PROCESS

Amélioration du précalcul de materials (expérimental)

4.17

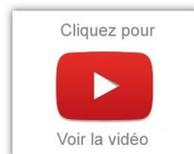
Il est désormais possible d'effectuer un certain nombre de précalculs (baking) sur les Materials des Static et des Skeletal Meshes.

Note: nécessite l'activation de l'**Asset Material Baking** setting dans la partie expérimentale dans l'Editor Preferences. Je n'ai pas encore testé, j'avoue que cela me laisse perplexe... quel type de précalcul peut-être réalisé ici et qui ne l'était pas avant ?

Composeure Compositing Plugin

4.17

À la suite de la démonstration de "The Human Race" à la GDC 2017, en partenariat avec The Mill et General Motors, Unreal Engine 4.17 inclut ce nouveau plugin, fonctionnant à la fois en C++ et Blueprint, et conçu pour **créer facilement une composition temps réel complexe**. Cela permet de **mélanger facilement de la vidéo (temps-réel ou non) avec des objets 3D**. Ça sniffie bon la réalité augmentée tout ça ;) - Oui, c'est exactement ça, regardez la vidéo suivante:



Unreal Engine 4 – Guide des mises à jour (4.20)

Un véhicule "réel" est équipé de "QR Codes" permettant d'identifier précisément ses différentes surfaces. Grâce à cela, UE4 peut positionner précisément les objets virtuels sur cet objet réel. Bon, ça reste du "Early Access", ce n'est donc pas à utiliser en production de suite.

C'est encore une démonstration de la volonté d'Epic Games de faire en sorte que son moteur ne soit pas exclusivement utilisé par le monde du jeu vidéo. Je vous rappelle [cet article de l'an dernier](#) sur la question. Ici, cela peut permettre de tourner une publicité... interactive (ou on peut changer la couleur, le modèle, etc.) ou bien de préparer un film... On imagine même les applications dans une émission en direct où le public peut faire divers choix en appelant un numéro ou en twittant ! Bon, je ne vais pas donner trop de conseils gratuits sinon on ne va plus me payer mon consulting ;)

Caractéristiques du plugin:

- Crée une liaison entre un Player Controller et un Texture Render Target 2D pour remplacer le pipeline de rendu du moteur par défaut
- Import des cinématiques du Sequencer, permettant des modifications "après coup".
- Parametric lens distortion: générer une UV displacement map depuis [OpenCV standard camera calibration model](#) .
- UVMap déforme des textures selon une UV displacement map et des matrices UV avec flexibilité du matériau.
- Stand-alone bloom: recréer des effets de composition comme des halos lumineux autour des objets.
- Correction des couleurs, Chromatic aberration

Voilà, et le tout est accompagné d'un projet exemple et de helpers pour les Blueprints.

Le moteur a également évolué pour apporter plus de souplesse au plugin, ces fonctionnalités pouvant également être utilisées par ailleurs comme le Support expérimental du canal alpha dans la chaîne de post-traitement du moteur: Circle DOF, TemporalAA, Motion Blur et Bloom.. Si cela vous intéresse, je vous invite à [consulter le Thread d'origine](#).

Material Shader Code

4.16

Vous pouvez maintenant activer un emplacement de stockage partagé pour tous les codes de shaders en utilisant le **Material Shader Code**, un seul exemplaire étant stocké pour les matériaux ou les instances de matériaux qui génèrent le même code shaders.

Brouillard volumétrique

Brouillard volumétrique

4.16

Vous pouvez intégrer le brouillard volumétrique dans une scène quelle que soit sa taille, permettant ainsi de rendre des effets de brouillard et de fumées plus réalistes.

Le brouillard peut être affecté par un certain nombre de lumières, une seule lumière directionnelle (Cascaded Shadow Maps ou static shadowing), un certain nombre de lumières avec ombrage dynamique ou statique (Cast Volumetric Shadow), un puits de lumière (Distance Field Ambient Occlusion) ou encore des particules de lumière (Volumetric Scattering Intensity>0).

Pour la configuration du brouillard volumétrique, vous pouvez consulter la doc.

Y-a eu un twitch hier d'Epic sur la question: vous pouvez [le revoir ici](#) (je mettrai plus tard le lien youtube, mais pas encore dispo). ça dure 1H, donc je vais essayer de vous résumer cette fonctionnalité: il s'agit d'améliorer la gestion du Bloom, c'est à dire d'un effet de brillance, d'HDR, de reflet de la lumière - c'est le coté halo lumineux des objets, mais aussi le coté étincelant (lens flare) d'un objet reflétant fortement la lumière. Regardez les diapos suivantes pour avoir une illustration du procédé.



L'amélioration est apportée en utilisant les [transformation rapides de fourrier \(FFT\)](#). Habituellement, c'est utilisé pour numériser au plus proche un signal analogique par exemple. Là, je n'entrerai pas dans le détail, mais les FFT sont utilisées, via les GPU dans le cadre d'un shader pour apporter la modélisation la plus réaliste possible de cet effet. La vidéo précédente (le twitch dédié) vous apprend dans un second temps à mettre en place le système.

Si vous le souhaitez, vous pouvez aussi voir ici les diapos qui ont été passées lors de la démo:

Bloom As Weak Scattering

Weak Scattering: Some Causes

- Haze
- Car Windshield (*mine is always a little dirty*)
- Eyelash Diffraction
- Inner-Camera Effects
- Fraunhofer Diffraction
 - Physically-Based Glare Effects for Digital Images (1996, Spencer et al)
 - Glare Generation Based on Wave Optics (2004, Kakimoto et al, Computer Graphics Forum)

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom As Weak Scattering

Weak Scattering

A single pin point remains mostly focused

Model Assumptions:

- Every point of light is scattered the same
- For most sources the scattered amount is imperceptible
- The scattering from bright sources creates “bloom”



In Camera scattering of a point

00:24:46 #UE4 LATEST FOLLOWER UNREAL ENGINE 4

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom: Standard Game-Style

Weak Scattering

Sum of Gaussian Filter

$$\sum_i^N \omega_i \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{\vec{x} - \vec{x}_i}{\sigma}\right)^2\right)$$


Standard Bloom in UE4: Recommended for game use

- Image down-sampled multiple times and Gaussian Blur applied
- Resulting images are summed and added to original
- Every pixel scatters light in a symmetric way to its neighbors
- Very Fast!
- But...

00:25:19 #UE4 LATEST FOLLOWER UNREAL ENGINE 4

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom As Weak Scattering

The camera response to a single bright source:

Camera scattering of point
Appears to be a very small star

Close Up of center
Center pixel about 10,000 times brighter than any other point in this .exr

Full image, adjusted contrast
Detail extents far across image!

00:27:14 #UE4
LATEST FOLLOWER

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom Scatter As Gather: Convolution

$$\hat{v}(\vec{x}) = (v * f)(\vec{x}) = \iint_{-\infty}^{\infty} v(\vec{r})f(\vec{x} - \vec{r})d\vec{r} = \iint_{-\infty}^{\infty} v(\vec{x} - \vec{r})f(\vec{r})d\vec{r}$$

Original Value
Filtered Value
Filter

Each Pixel Scatters to all the others according to the filter.

View this as a weighted sum, here in one dimension :

$$\hat{v}_i = \sum v_j f_{i-j} = \sum v_{i-j} f_j$$

For Example, simple smoothing
 $f = \{0, \dots, 0, 1/4, 1/2, 1/4, 0, \dots, 0\}$

$$\hat{v}_i = v_{i-1}/4 + v_i/2 + v_{i+1}/4$$

00:28:37 #UE4
LATEST FOLLOWER

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom Scatter As Gather: Convolution

Convolution:

$$\hat{v}(\vec{x}) = (v * f)(\vec{x}) = \iint_{-\infty}^{\infty} v(\vec{\tau}) f(\vec{x} - \vec{\tau}) d\vec{\tau} = \iint_{-\infty}^{\infty} v(\vec{x} - \vec{\tau}) f(\vec{\tau}) d\vec{\tau}$$

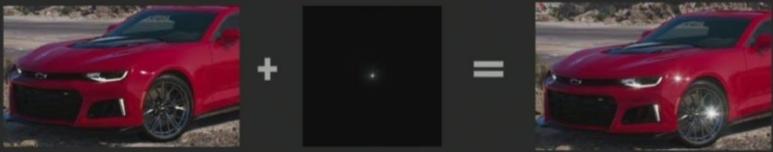
- Every point is weighted sum *of all other points*
- N points gather from all N points. $O(N^2)$
- We need a faster way! -> *FFT*

00:11:45 #UE4 LATEST FOLLOWER 01:02:46

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom Scatter As Gather: Convolution

Convolution With Fast Fourier Transform:



1. Image_Frequencies = FFT(Image)
2. Filter_Frequencies = FFT(Filter)
3. Convolved_Frequencies = Image_Frequencies x Filter_Frequencies
4. Convolved_Image = InverseFFT(Convolved_Frequencies)

00:11:20 #UE4 LATEST FOLLOWER 01:02:46

Unreal Engine 4 – Guide des mises à jour (4.20)

Unreal Engine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom Scatter As Gather: Convolution

Original Value
Filtered Value
Filter

$$\hat{v}(\vec{x}) = (v * f)(\vec{x}) = \iint_{-\infty}^{\infty} v(\vec{\tau}) f(\vec{x} - \vec{\tau}) d\vec{\tau} = \iint_{-\infty}^{\infty} v(\vec{x} - \vec{\tau}) f(\vec{\tau}) d\vec{\tau}$$

View this as a weighted sum, here in one dimension :

$$\hat{v}_i = \sum v_j f_{i-j} = \sum v_{i-j} f_j$$

For Example, simple smoothing
 $f = \{0, \dots, 0, 1/4, 1/2, 1/4, 0, \dots, 0\}$

$$\hat{v}_i = v_{i-1}/4 + v_i/2 + v_{i+1}/4$$

Each Pixel Scatters to all the others according to the filter.



00:11:01 #UE4 LATEST FOLLOWER 01:40:01

Unreal Engine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

OrionGame [DebugGame] ++Orion+Dev-General



World Outliner

Label	Type
Origin (Editor)	World
Post_Agora_Dawn	PostProcessVolume
Post_JungleEyeAdapt	PostProcessVolume
Post_ProcessMaterialF	PostProcessVolume
Post_ProcessVolume	PostProcessVolume
Post_ProcessVolume_De	PostProcessVolume

Showing 5 of 2,846 actors (1 selected)

Details: PostProcessVolume

Image Effects

- Bloom: Method: Standard, Intensity: 0.575, Threshold: 0.9
- Convolution Kernel: [Image]

Advanced



00:30:34 LATEST FOLLOWER 01:40:01

Unreal Engine 4 – Guide des mises à jour (4.20)

Unreal Engine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom: What is Fourier Transform?

Any finite length signal can be expressed as a sum of sines and cosines

$$v_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} V_k \exp\left(-i2\pi k \frac{n}{N}\right) \quad k, n \in [0, N-1] \quad \exp\left(i2\pi k \frac{n}{N}\right) = \cos\left(2\pi k \frac{n}{N}\right) + i \sin\left(2\pi k \frac{n}{N}\right)$$

↓ Signal ↓ Amplitude of each frequency ↘ Sine & Cos

Inverse computes amplitudes of each frequency

$$V_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} v_n \exp\left(i2\pi k \frac{n}{N}\right) \quad k, n \in [0, N-1]$$


Low-Pass filter: only used first few V_k

09:28:14 #UE4 LATEST FOLLOWER 01:49:46

Unreal Engine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom: Fast Fourier Transform?

Sines and Cosines have symmetries that can be exploited speeding the transform

$$V_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} v_n \exp\left(i2\pi k \frac{n}{N}\right) \quad k, n \in [0, N-1] \quad \exp\left(i2\pi k \frac{n}{N}\right) = \cos\left(2\pi k \frac{n}{N}\right) + i \sin\left(2\pi k \frac{n}{N}\right)$$

Split the sum into even and odd terms, and symmetry saves work!

$$V_k = \text{Even}_k + \exp\left(i2\pi k \frac{1}{N}\right) \text{Odd}_k \quad \text{Even}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N/2-1} v_{2n} \exp\left(i2\pi k \frac{n}{N/2}\right)$$

$$V_{k+N/2} = \text{Even}_k - \exp\left(i2\pi k \frac{1}{N}\right) \text{Odd}_k \quad \text{Odd}_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N/2-1} v_{2n+1} \exp\left(i2\pi k \frac{n}{N/2}\right) \quad \text{for } 0 \leq k < N/2$$

Now recurse, each small transform (*Even*, *Odd*) and be treated as new transforms And subdivided. Due to this splitting FFT does best with power_2 signals.

09:28:07 #UE4 LATEST FOLLOWER 01:49:46

Unreal Engine 4 – Guide des mises à jour (4.20)

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom : Parallel FFT for GPU

GPU: Highly parallel, but with thread-communication limitations.
Similar restriction are found in supercomputing architectures.

- Stockham Formulation: Parallel formulation
 - Used for Vector Computers (1987 D Baily, Journal of Supercomputer Applications)
 - Applied to GPUs (2008 Govindaraju, Proceedings of ACM/IEEE on Supercomputing)
 - Our implementation largely follow this
- Each Scanline
 - Transformed independently
 - Assigned to a thread group with group shared memory

00:29:04 #UE4 LATEST FOLLOWER 01:40:49

UnrealEngine
Unreal Engine - Image-Based (FFT) Convolution for Bloom

Bloom: Compute Shaders

FFT implemented as compute shaders

- Pass
 - Forward Horizontal Transform
 - Uses 2-for-1 trick to transform (r,g) and (b,a)
- Pass
 - Forward Vertical Transform
 - Convolution, i.e. Multiply, with cached pre-transformed Kernel
 - Inverse Vertical Transform
- Pass
 - Inverse Horizontal Transform
 - Inverts 2-for-1 trick

FFT: Two-For-One Forward Vertical of size 1024 of buffer 1053 x 738
FFT: Apply Forward Horizontal Transform, Multiply Texture, and Inverse Transform
FFT: Two-For-One Inverse Vertical of size 1024 of buffer 1053 x 1026
From GPUvisualizer

00:27:51 #UE4 LATEST FOLLOWER 01:40:49

En gros, la petite étoile que vous voyez sur la carrosserie de la voiture ci-dessous, c'est ça (en partie).



Éclairage précalculé sur le brouillard volumétrique

4.18

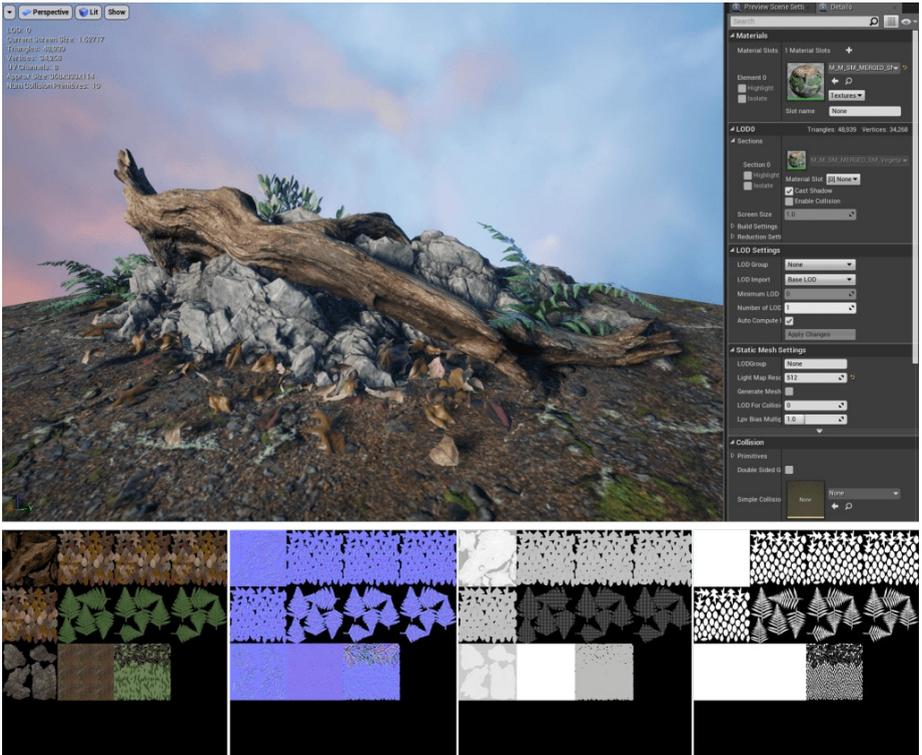
Le Volumetric Fog interpole désormais l'éclairage précalculé sur sa position dans l'espace à l'aide des Volumetric Lightmaps, ce qui permet aux lumières stationnaires - dont l'éclairage indirect est stocké dans des lightmaps - d'affecter le brouillard. Les lumières statiques et émissives affectent le brouillard sans rien coûter, puisqu'elles sont toutes fusionnées dans la volumetric lightmap. La lumière du ciel suit maintenant les couleurs de la journée (lever/coucher de soleil).



Opacité et masque d'opacité pour Material Flattening:

4.16

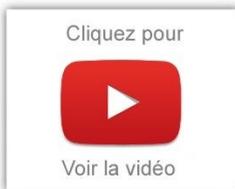
Ont été ajoutées le support pour le baking des valeurs d'opacité lorsque vous utilisez le Actor Merge Tool ou un LOD hiérarchique. Le matériau résultant (instancié) utilise le mode de fusion configuré pour assurer qu'il suit le chemin de rendu correct. Exemple:



Améliorations Mesh Editor Matériaux et Sections Panel

4.17

L'interface pour l'édition des matériaux et des sections Mesh a été condensée pour permettre aux artistes d'être plus productifs lors de l'édition des meshes. Ce changement affecte à la fois le statique Mesh Editor et Skeletal Mesh Editor.



Le défilement a été réduit en fusionnant le contrôle de Lods dans le même panel permettant de travailler sur un LOD, la synchronisation du LOD modifié avec la fenêtre LOD. Cela permet à l'utilisateur de voir toujours le LOD en cours de modification dans la fenêtre.

Outil Mesh Paint Tool amélioré

4.16

il est plus simple et la fonctionnalité peut être utilisée dans d'autres parties de l'éditeur.



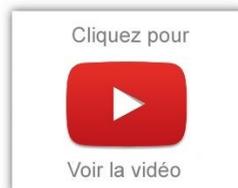
Les outils de peinture peuvent maintenant être utilisés sur des meshes de squelettes! C'est ce qu'on avait vu déjà dans [mon article sur les textures volumétriques](#) (même s'il n'y a pas de lien direct, le système utilise cette fonctionnalité). La peinture est appliquée directement sur

l'asset du mesh du squelette.

Detect Material on Mesh Surfaces:

4.16

Une nouvelle fonction «Get Material From Face Index» a été ajoutée pour les composants, elle permet de récupérer le matériau appliqué à un composant après avoir effectué un Line Trace. Ceci est pris en charge pour Static Meshes, Procedural Mesh Components et BSP.



Curve Atlas dans les matériaux

4.20

Les matériaux peuvent désormais utiliser un Curve Atlas pour stocker et accéder aux données de courbe de couleur linéaire avec un support supplémentaire fourni par Blueprint. L'Atlas de courbes utilise la même couleur de courbe linéaire que précédemment, sauf que vous pouvez utiliser autant de courbes de couleurs linéaires que la taille de votre Atlas spécifié.

Pour créer un nouvel Atlas de courbes, utilisez le navigateur de contenu pour sélectionner Add New > Miscellaneous et sélectionnez Curve Atlas

Lorsque vous ouvrez un éditeur d'assets Curve, vous pouvez ajuster les pinces de teinte, de saturation, de luminosité, de vibrance et d'alpha de n'importe quelle courbe individuelle. De plus, les aperçus dans le navigateur de contenu affichent le dégradé défini par la courbe.

Pour plus d'informations, voir [Curve Atlases](#) dans les matériaux.



Profondeur de haute qualité du terrain et Tone Mapping

Nouvel algorithme de profondeur de champ appelé "Circle DOF"

4.8

Si vous ne savez vraiment pas ce qu'est un DoF, je vous conseille d'aller [bidouiller cette page](#), un cas pratique vaut mieux que mille mots. Une fois fait, nous allons pouvoir parler technique.



- **Nouvel algorithme de profondeur de champ appelé "Circle DOF"**: cercle bokeh et agréables transitions de focus, on contrôle par les paramètres physiques de l'appareil photo:
 - fstop,
 - longueur focale,
 - distance focale,
 - floupour simuler la diffusion de la lumière à travers l'atmosphère
- égal au système de codage couleur Académie (ACES), nouveaux étalonnages des couleurs, balance des blancs, baked en 32x32x32 LUT, 50% plus rapide qu'auparavant.

Qualité supérieure du DOF

4.11

Nous pouvons augmenter le nombre d'échantillons (samples) du "Circle DoF" pour augmenter la qualité en réduisant le bruit. En concret, cela donne ça :



Sur l'image ci-dessus, les éléments sont éloignés – l'image est meilleure à droite car il y a moins de bruit parasite au niveau du rendu.

Cinematic Depth of Field

4.20

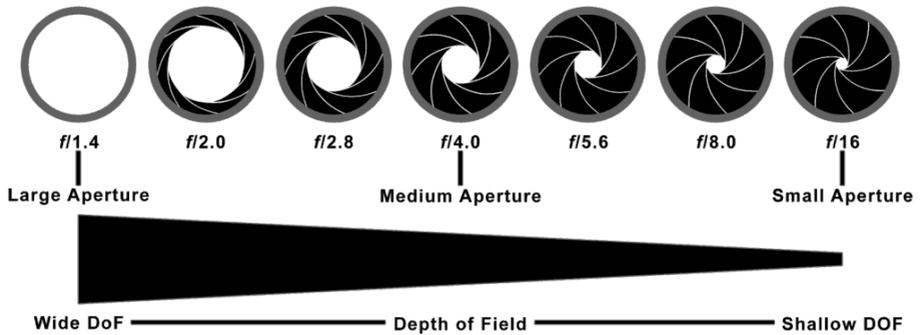
La [nouvelle Cin-DoF](#), vous permet de rendre **des scènes de qualité cinéma dans un environnement en temps réel!** Cette nouvelle méthode est conçue pour remplacer la méthode **Circle DoF** et est plus rapide que la plupart des autres méthodes DoF, telles que **Bokeh**. Pour plus d'info sur le DoF, je vous renvoie à [la doc officielle](#).

Avec Cin-DoF, l'effet de profondeur de champ est plus propre, grâce à l'utilisation d'une méthode appelée "**Procedural Diaphragm Simulation**" (**PDS**). Pfff, là, va falloir que je vous explique. Un diaphragme de lentille est contrôlé sur un appareil photo par des lames.



La PDS permet aux artistes de configurer le nombre de lames utilisées, d'ajuster le diaphragme et de configurer la courbure des lames, tel que:

Unreal Engine 4 – Guide des mises à jour (4.20)



L'idée, c'est que certaines parties de la scène sont peut-être hors focus, mais restent importantes. Il faut donc pouvoir l'ajouter au focus en quelques sortes, même si ce n'est pas naturel. C'est pour mettre en effet un objet à l'horizon ou pour un FX particulier. Bref, ça laisse une latitude de réglage. Je ne sais pas en quoi on peut qualifier cela de procédural... peut-être de par le fait qu'il faut continuellement ajuster l'exposition lors de l'animation comme dans une véritable caméra. Je ne sais pas, si vous avez une meilleure explication, je suis preneur.

Cette nouvelle implémentation DoF prend également en charge la couche alpha, la stabilité de la résolution dynamique et inclut des paramètres pour les projets pour console.



Unreal Engine 4 – Guide des mises à jour (4.20)



1 - Profondeur cinématique du champ activée; 2 - Profondeur de champ désactivée

Cinematic Depth of Field est activé par défaut en lieu et place de **Circle DoF** dans les paramètres **Camera** et **Post Process**.

- Cinematic DoF prend en charge les plates-formes suivantes:
 - D3D11 SM5, D3D12 SM5, Vulkan SM5, PlayStation 4, Xbox One et Mac.
- La simulation Bokeh procédurale prend en charge les fonctionnalités suivantes:
 - Configuration du nombre de blades pour Diaphragm.
 - Configuration de la courbure des blades directement avec la plus grande ouverture de l'objectif (F-stop minimal).
 - Commandes configurables disponibles dans les paramètres de la **caméra** du post-traitement volum, de l'actor camera et du Cine Camera Actor.
- De nombreux paramètres d'évolutivité personnalisables utilisant les variables de la console r.DOF. * Pour l'adapter aux besoins de votre projet sur du matériel avec des ressources finies.

Pour plus d'informations, une documentation sur [cette nouvelle approche est en ligne](#).

Support expérimental de Direct X 12

4.9



Sous Windows 10, on pouvait essayer d'utiliser Direct X 12 en faisant fonctionner le moteur avec "-DX12" sur la ligne de commande. Les ingénieurs de Microsoft ont ajouté le support de Direct X 12 à UE4, et l'équipe d'Unreal Engine a travaillé avec eux pour intégrer les changements vers 4.9. A noter que sous 4.12, cela reste une fonction expérimentale.

Allez voir [cette vidéo](#) qui compare Direct X 11 et 12 avec Unreal Engine et une GTX 770.

Area Shadows (stationary lights)

4.9

Vous n'êtes plus limité à des ombres nettes avec des tailles uniformes. Le **Lightmass ray-tracer** prend désormais en charge les **zones d'ombre pour les lumières fixes!**



Activez l'option pour utiliser cette fonctionnalité "Use Area Shadows for Stationary Light". Le "Light Source Angle" contrôle l'adoucissement de ces ombres.

Les ombres uniformes sont meilleures avec de faibles résolutions de lightmaps. Elles sont contrôlées par la propriété 'Light Source Angle' comme les zones d'ombre.

Il y a aussi une optimisation pour les zones de cartes qui sont touchées par une seule lumière directionnelle, les ombres sont dans un seul "channel texture", pour 1/4 du coût mémoire par rapport aux 4 canaux précédents.

Ambient Occlusion Material Mask

4.9

Par exemple pour ajouter des effets et de la saleté dans des domaines où il serait accumulés.



La capture d'écran ci-dessus utilise un masque AO pour mélanger automatiquement une couche de saleté dans les coins de la scène.

Pour utiliser un masque AO, vous aurez besoin de permettre à la fois l'utilisation de 'Use Ambient Occlusion' et de 'Generate Ambient Occlusion Material Mask' avec World Settings -> Lightmass settings, puis build lighting.

Les autres contrôles AO comme Max Occlusion Distance peuvent être utiles pour peaufiner le look. (Veillez à régler Direct et Indirect Occlusion Fraction à 0, de sorte que cette AO ne soit pas appliquée à l'éclairage général.)

Amélioration de la Distance Field Ambient Occlusion

4.16

Distance Field Ambient Occlusion et **Ray Traced Distance Field Shadows** sont maintenant **30-50% plus rapides** sur les dernières consoles et le PC de moyen gamme. **Distance Field Generation** est, lui **2,5 fois plus rapide!** Je ne veux pas entendre que les versions précédentes avaient dû être codées avec les pieds ! Non, mais, vous vous rendez-compte de la tonne d'ingénierie qu'il y a sous le capot d'UE4 ?!



4.9

Sky occlusion offre de **belles ombres douces autour des objets éclairés naturellement**, ce qui est particulièrement important quand il y a des nuages, un cas d'utilisation traditionnellement difficile à rendre en haute qualité sans éclairage pré calculé.

Distance Field Ambient Occlusion résout donc ce problème pour

Unreal Engine 4 – Guide des mises à jour (4.20)

les jeux entièrement dynamiques où le moment de la journée et l'environnement peuvent changer à tout moment. Il est maintenant utilisable sur un PC moyen gamme et sur PlayStation 4.



Les calculs AO sont maintenant très rapides. Il faut toutefois faire attention aux dédoublement.



Dans les cas utilisant une caméra statique et des surfaces plates, la nouvelle méthode est **1,6x plus rapide**.

Dans les scènes complexes et une caméra en mouvement rapide, la nouvelle méthode est **5.5x plus rapide**.

Mesh Distance Field Materials

4.9

Cette fonctionnalité permet aux matériaux de déterminer la distance du plus proche solide. Il y a deux nouveaux nœuds pour accéder aux propriétés globales de terrain à distance dans l'éditeur de matériaux.



- `DistanceToNearestSurface` renvoie une distance. Notez que la distance est toujours proche de 0 sur la surface d'un maillage opaque.
- `DistanceToNearestSurface` peut être utilisée pour éviter d'avoir des effets d'occlusion:

L'utilisateur UE4 Roel Bartstra a déjà trouvé quelques utilisations intéressantes: une pour obtenir un look de corps mou, et une autre pour créer des cartes de flux qui évitent automatiquement les surfaces.

Comme avec toutes les fonctionnalités de champ de distance, cela nécessite que "Generate Mesh Distance Fields" soit activé dans les paramètres de rendu du projet.

Post Process Blending

4.9

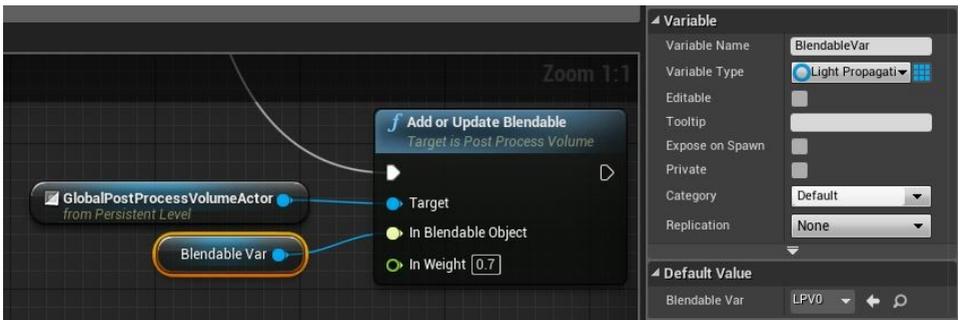
Les Post-process "Blendables" sont utilisés pour faire des transitions post-process entre différents états.

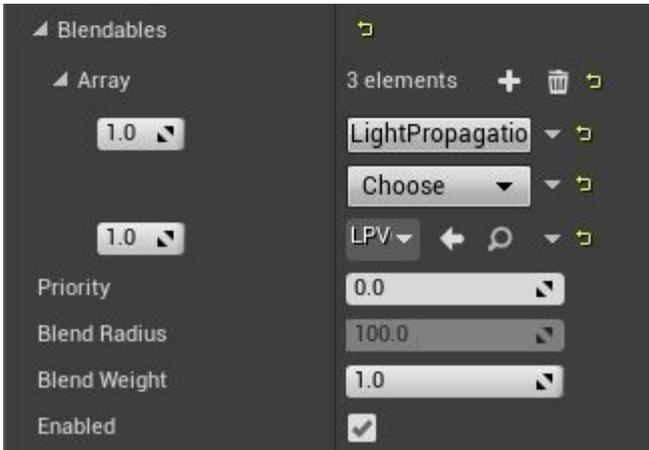
Il est possible de régler le poids de chaque état en Blueprints ou en code C++.

Vous pouvez opter pour la création d'un asset Blendable que vous pouvez partager entre de nombreux objets.

La définition des paramètres personnalisés est désormais plus facile.

Voici un exemple d'utilisation des Blueprints pour modifier le % de mélange pour un lancement:

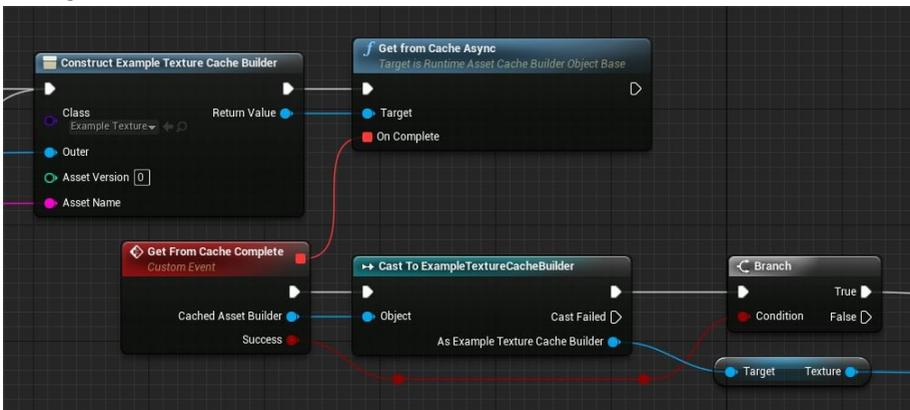




Runtime Asset Cache

4.9

Certains jeux ont besoin de copies locales des assets (ex: publicité graphique). Le nouveau Runtime Asset Cache est un **système de cache d'assets générique** qui peut être utilisé pour stocker des données générées pendant l'exécution dans des assets buckets configurables.



Unreal Engine 4 – Guide des mises à jour (4.20)

Le Runtime Asset Cache a les caractéristiques suivantes:

- **Buckets multiples**, stocker séparément les assets par utilisateur.
- Chaque bucket a une **taille configurable**.
- Possibilité de faire des **versions d'assets**, permettant de les reconstruire quand il deviennent obsolètes.
- Interrogation du cache Synchrones et asynchrones.
- Lorsque le cache grossit et dépasse la taille spécifiée, les entrées les plus anciennes sont retirées.

Volume Decals

4.9

3D function field rendering est une **alternative aux polygones pour décrire la géométrie**. Il est mieux adapté à la résolution de problèmes avec un rendu difficile tels que des ombres, des opérations booléennes, le lissage, l'occlusion ambiante, le ray tracing, et la déformation.

Les fonctions de distance 2D et 3D sont déjà utilisés dans divers endroits dans le moteur, comme les polices, lightmaps, et les ombres sur le terrain à distance/l'ambient occlusion.

La nouvelle fonctionnalité expérimentale (en 4.9) Volume Decals permet à une fonction de distance de faire un rendu "objets opaques" dans le GBuffer, en utilisant toutes les informations reportées, y compris l'ambient occlusion, les réflexions, l'éclairage, et la profondeur de champ.

Unreal Engine 4 – Guide des mises à jour (4.20)



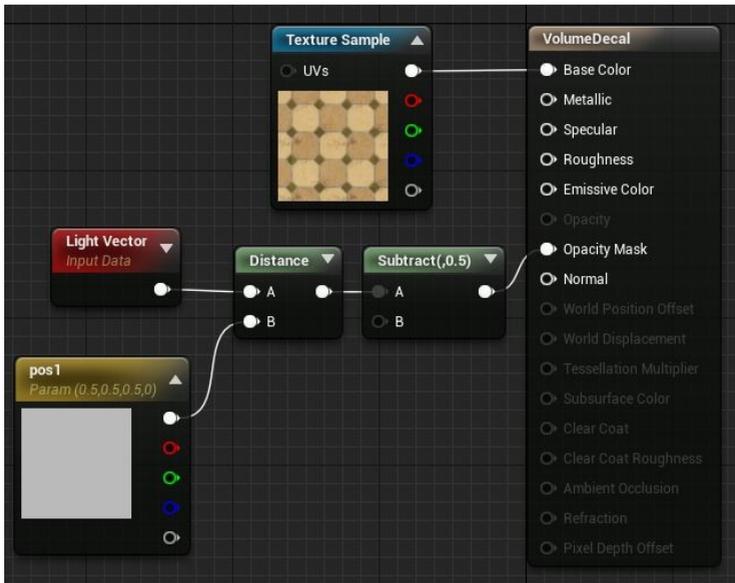
Le bol dans l'image ci-dessous est un Volume decal.

L'objet a été créé (voir image ci-dessous, de gauche à droite) à partir de deux sphères en faisant des intersections et des soustractions booléennes.



La distance est définie ainsi:

Unreal Engine 4 – Guide des mises à jour (4.20)



La fonction est inachevée. Les projection d'ombres, le support de carte, et un éclairage indirect manquent, et la performance et la qualité des optimisations ne sont pas encore au top. En outre, la fonction est gourmande en GPU (la fonction de distance doit être évaluée des centaines de fois pour chaque pixel). C'est en cours d'amélioration.

Shading visage et vêtements

Shading réaliste des visages, cheveux, yeux, peau et tissus

4.11

Ajout d'un modèle d'ombrage pour les cheveux.



Pour utiliser cette fonction, il suffit de choisir des cheveux ("Hair") dans la liste des modèles d'ombrage dans l'éditeur de matériaux.



Le modèle d'ombrage des yeux utilise une version simplifiée du

Unreal Engine 4 – Guide des mises à jour (4.20)

subsurface scattering au travers de la rétine.

Ces modèles montrent aussi la réfraction à travers la cornée, l'assombrissement de l'anneau limbique, etc.

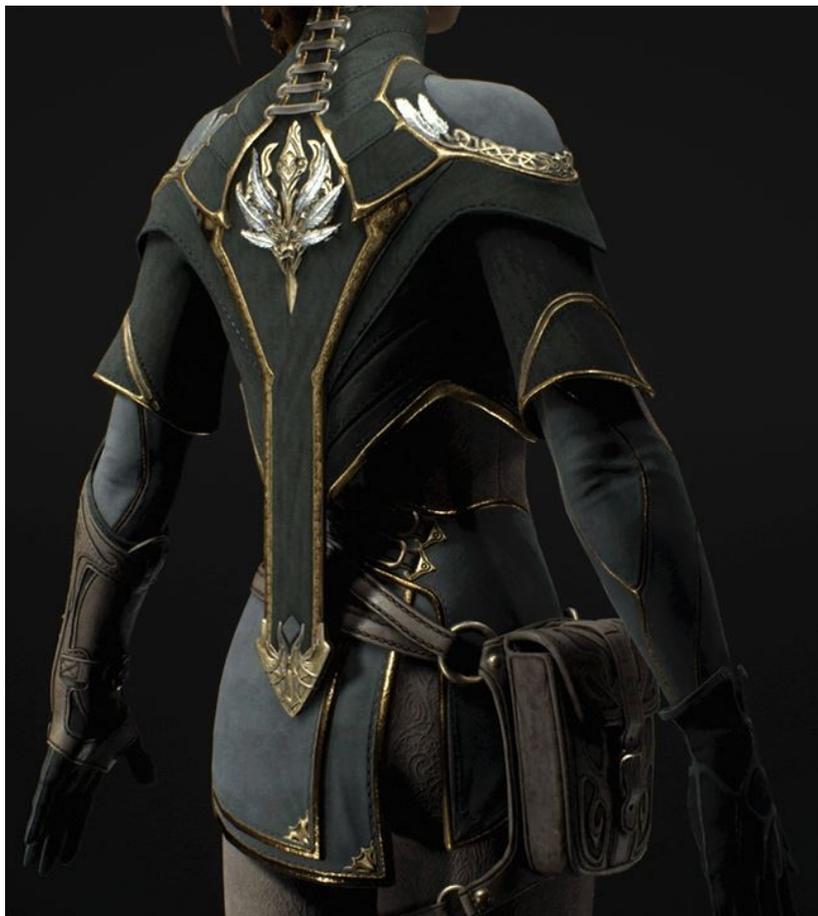


Le modèle nécessite moins de mémoire GPU.

La diffusion est indépendante de la résolution et il n'y a plus de changement de couleur sur les bords des objets.

La Texture et l'éclairage des détails sont mieux préservés en stockant l'éclairage diffus et spéculaire séparément dans un motif en damier plutôt que dans un seul pixel.

Les tissus bénéficient d'un nouveau physically based shading model. Regardez ce velours recouvert de dorures !



C'est grâce au travail réalisé sur le jeu Paragon que sont apparues ces nouveautés.

Paragon c'est le dernier **MOBA** (Multiplayer Online Battle Arena) des Studios Epic Games en version tir à la troisième personne.

Si vous voulez le tester, il y a [une communauté française](#) qui s'est formée.

Shading peau en haute-résolution

4.14

UE4 prend désormais en charge l'**ombrage de la peau en haute-résolution** pour le modèle d'ombrage **Subsurface Profil**. Ceci permet d'obtenir un très bon éclairage pour les détails comme les pores et les rides.

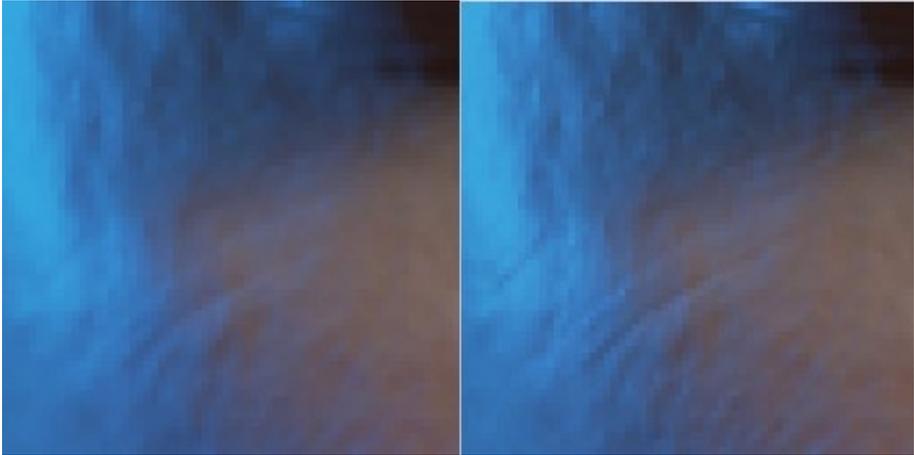


Ci-dessus, éclairage obtenue avec le motif damier utilisé auparavant, et ci-après le nouvel affichage:



Unreal Engine 4 – Guide des mises à jour (4.20)

Bon, j'avoue, avec cette image et sans possibilité de bouger autour de l'objet, c'est pas évident de voir la différence. Un petit zoom pour mieux voir:



Nouvel éclairage où chaque pixel contient des informations d'éclairage diffus et spéculaire dans un encodage RGBA. Cela nécessite au moins un format de couleur scène 64 bits avec un canal alpha complet.

Ce comportement peut être modifié en utilisant la variable console ***r.SSS.Checkerboard***.

Les valeurs possibles pour cela sont: 0: Damier désactivé (résolution complète), 1: Damier activé (ancien comportement), 2: Automatique (par défaut)

Dithered opacity mask

4.11

Pour **simuler une surface translucide** en utilisant un matériau opaque:



Case à cocher “Dithered Opacité Mask” dans l’éditeur de matériaux.

Capsules d'ombrage

4.11

Unreal Engine supporte désormais les **ombrages indirects doux** (soft indirect shadows) produits par la capsule du personnage. Avant, il n'étaient disponibles qu'au travers du mécanisme de screen-space ambient occlusion et ne produisait pas de très bons résultats.



Dans les jeux, c'est surtout pour les **points de contact des personnages et véhicules avec le sol** que c'est intéressant. Il est possible de les utiliser pour les ombrages directs, permettant une grosse optimisation du temps de calcul de l'ombrage (au prix d'une approximation grossière comme dans l'illustration ci-dessous).

Unreal Engine 4 – Guide des mises à jour (4.20)



Le coût du GPU est proportionnel au nombre de capsules et du nombre de pixels concernés par l'ombre portée.

Voici comment activer les Ombres Capsule:

- Créer un nouvel asset en utilisant uniquement les Sphyl bodies (capsules). Les sphères fonctionnent également, mais ne sont pas aussi flexibles. Les capsules doivent se chevaucher légèrement au niveau des joints. Les capsules de pied sont les plus importantes pour que le personnage semble relié à la terre.
- Attribuer l'asset à l'ombre Physique de l'Asset Skeletal Mesh Asset
- Activer les capsules d'ombres indirectes sur le composant Mesh Skeletal

Builds plus rapides (Intel embree support)

4.11

La bibliothèque “Intel’s Embree ray tracing” est intégrée dans Lightmass et permet des calculs plus rapides.

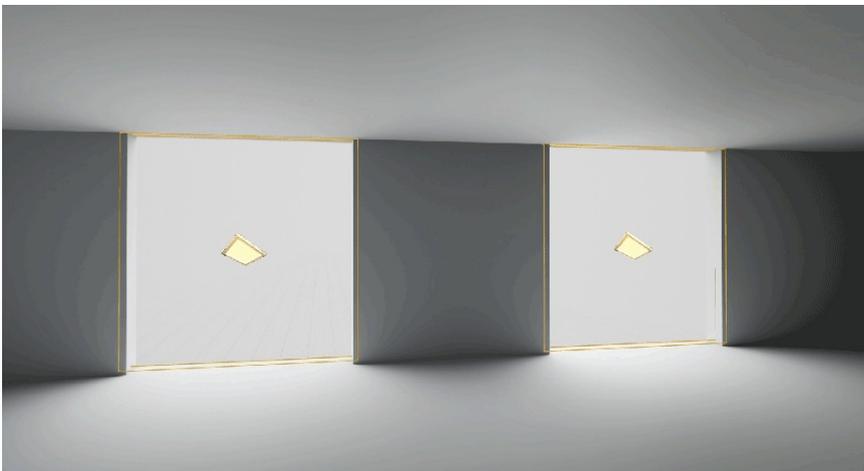
Intel propose aux développeurs les sources d'un rendu photoréaliste basé sur le lancer de rayon.

La particularité d'**Embree** est d'être optimisée pour les instructions des processeurs Intel : SSE et AVX. Pour cela, il vous faut disposer d'un **processeur Intel** pour profiter pleinement des optimisations (pour le build, pas pour le jeu au final). A l'origine, Embree a été pensée comme un outil de base destiné à être inséré dans un moteur de rendu. C'est chose faite sous UE4.

Comme exemple, le “Sun Temple” level lighting est **2.4x plus rapide à builder en utilisant Embree**.

Lightmass portals

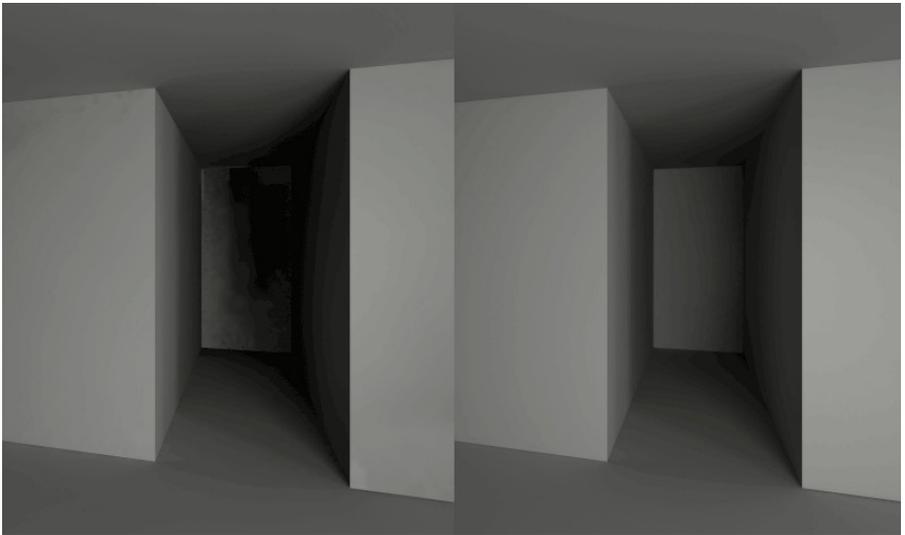
4.11



Unreal Engine 4 – Guide des mises à jour (4.20)

Les rendus en intérieur des skylights (qui passent par les ouvertures de fenêtres, portes, etc.) peuvent être massivement améliorés en utilisant des Lightmass Portals.

Ils permettent de focaliser les rayons de la lumière dans la direction de l'ouverture. (Ci – dessous à gauche : sans portail – ci – dessous. Droite : avec portails)



Amélioration de la directionnalité des skylights

4.18

Un cubemap filtré est utilisé avec une résolution par défaut beaucoup plus élevée. Le mip approprié est choisi en fonction de la taille des rayons pour éviter l'aliasing.



Canaux d'éclairage

4.11

Les canaux d'éclairage permettent à des lumières dynamiques d'affecter uniquement les objets lorsque leurs canaux d'éclairage se chevauchent.

Nous avons maintenant un support pour jusqu'à 3 canaux d'éclairage.



Vraiment **utile pour les cinématiques**: permet par exemple d'éclairer uniquement un personnage pour le mettre en valeur, sans toucher à son environnement !

Éclairage indirect multi-rebond des puits de lumière

4.18

Support pour les multiples rebonds d'éclairage indirect provenant des Skylights statiques et fixes grâce à un nouveau paramètre «Num Sky Lighting Bounces» dans World Settings -> Lightmass.



Les matériaux doivent avoir une valeur diffuse élevée pour que les multiples rebondissements soient visibles dans la plupart des cas.

Rectangular Area Lights

4.20

Elles vous permettent de rendre l'éclairage plus réaliste pour les environnements contenant de grandes sources de lumière, telles que les plafonniers fluorescents, les téléviseurs, les enseignes allumées, et plus encore! Les voyants de zone rectangulaire sont accessibles depuis le panneau Modes avec les autres types de voyants.

La Rectangular Area Light agit principalement comme un point lumineux, sauf qu'elle a la largeur et la hauteur de la source pour contrôler la lumière d'émission.

Jusqu'à maintenant, on se débrouillait avec un material émissif, mais c'était beaucoup moins facile à paramétrer et cela permettait moins de choses.

Unreal Engine 4 – Guide des mises à jour (4.20)



Actuellement, seul le Deferred Renderer est pris en charge. Quid de l'antialiasing par exemple ? A voir. Si vous ne connaissez pas la différence avec le Forward Renderer, disons que c'est une façon de réduire la complexité du rendu et d'améliorer les performances. On parle de différé car aucun shading n'est réalisé en première passe des vertex et pixel shaders. Cela n'interviendra qu'à la seconde passe. Allez [voir ici](#) pour comprendre le fonctionnement.

L'ombrage pour le mode statique et stationnaire fonctionne comme une area light source alors que la version mobile, fonctionne plus comme une point light.

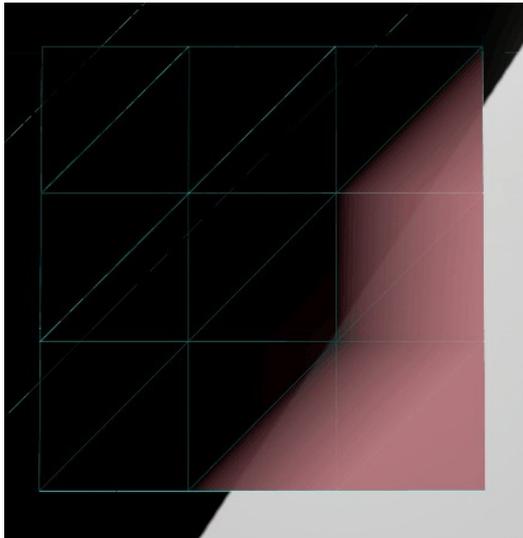
Coté performance, rien pour le type statique (puisque précalc). Après, dès qu'il s'agit de type mobile avec ombrage, c'est plus couteux globalement que les lumières de type point light ou spot (sauf pour l'ombrage, c'est pareil). Pour les type stationnaires, ça va dépendre de la plateforme cible et du scaling de résolution. Ça peut doubler...à surveiller donc.

Objets translucides

Per-vertex translucent lighting

4.11

Le côté translucide de la lumière peut maintenant être rendu beaucoup plus rapidement à l'aide de paramètres per-vertex translucent lighting.



Il y a deux nouveaux modes d'éclairage de translucidité disponibles dans l'éditeur de matériaux qui calculent l'éclairage par vertex

Amélioration des lumières sur les objets translucides

4.14

Dans le rendu différé, la nouvelle fonctionnalité d'ombrage peut être utilisée sur des surfaces translucides pour obtenir reflets spéculaires de multiples lumières.



Optimisations pour DIRECTX 12

4.11

Permettre une meilleure utilisation du processeur tout en générant des commandes de rendu en parallèle.

Nous apprenons que les ingénieurs de Microsoft ont intégré Direct X 12 à la Xbox One (y-avait des rumeurs à ce sujet depuis le début). Cela reste expérimental, mais on peut déjà l'activer sous UE4:

- paramétrer `bBuildForD3D12` à « true » dans la section `XboxOneRuntimeSettings` de `BaseEngine.ini`
- paramétrer `D3D12_ROOT_SIGNATURE` à « 1 » dans `XboxOneShaderCompiler.cpp`
- Décommenter `GetSamplePosition` dans `PostProcessSelectionOutline.usf`
- Rebuild et relancer

Direct X 12 offre, pour ce faire, un accès plus direct, plus bas niveau, aux composants matériels, ce qui dispense de **coûteuses transactions avec le CPU**.

Une démonstration a notamment été réalisée lors de la GDC sur le jeu [Lionhead](#) qui tourne sous Unreal Engine 4.

Direct X 12 est effectivement l'équivalent chez Microsoft de [Mantle](#) chez AMD. Si ce n'est que contrairement à cette dernière, l'API de Microsoft est interopérable, **elle peut fonctionner avec n'importe quelle carte graphique**. Cette nouvelle API fonctionne donc à bas niveau sur les GPU, un peu comme [Vulkan pour OpenGL](#).

Pour en savoir plus sur Direct X 12, j'ai fait [un point sur les principales fonctions](#) de la librairie sur mon blog.

4.16

DirectX 12 est maintenant le moteur de rendu par défaut pour **Xbox One** avec des améliorations de performances. On s'en doute, comme l'API est plus proche du matériel. Voir [cet article](#) pour + d'infos.

Vous pouvez développer des jeux HTML5 en utilisant [WebAssembly](#) **(wasm)** et [WebGL 2](#). Unreal Engine utilise le dernier [emscripten](#) toolchain Mozilla (de v1.37.9).

Il s'agit d'une nouvelle technologie pas forcément prise en charge par tous les navigateurs, elle est considérée comme une caractéristique Early Access et nécessite un accès GitHub. Vous pouvez activer wasm et WebGL 2.0 dans la section emscripten des paramètres du projet HTML5. Si vous avez besoin du soutien du navigateur le plus large possible, continuer à utiliser asm.js et WebGL 1.

Flux de développement sur GitHub

4.20

Les flux de développement d'Unreal Engine sont maintenant mis à jour en direct sur GitHub. Si vous souhaitez disposer de la dernière version du code de développement, vous pouvez désormais extraire ces flux directement.

Pour en savoir plus, consultez l'[article de blog d'EPIC](#) .

Metal remplace OpenGL pour Mac OSX

4.11

Metal pour OSX est l'équivalent de Direct X 12 de Microsoft, Mantle d'AMD et Vulkan. C'est une API permettant d'accéder à la programmation bas niveau (plus directe) des GPU.

Voilà, rien à redire, les aficionados de la pomme ne sont pas en reste. On leur promettait depuis longtemps, c'est désormais une **réalité**.

Plus encore, il y a aussi un support expérimental pour les Shader Model 5 features, on peut tester en ajoutant en ligne de commande: "-metalsm5".

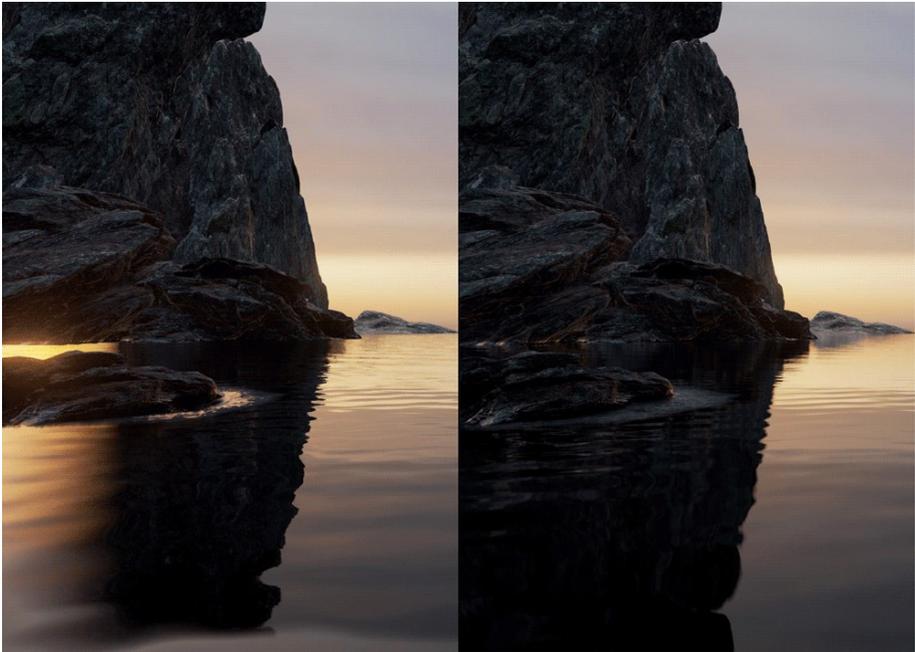
Réflexions

Réflexion planaire en temps réel

4.12

Il y a un coût en calculs pour utiliser cette fonction, cependant, la qualité de la réflexion est superbe.

Pour l'utiliser, il faut autoriser: "Support global clip plane for Planar Reflections" dans Rendering project settings et relancer l'éditeur.



Les matériaux réfléchissants à proximité seront automatiquement affectés par la réflexion plane et leurs normales seront utilisées pour modifier la réflexion, ce qui permet des **ondes simulées**.

Réflexions de haute qualité

4.12

Possibilité de **coder les vecteurs de normales et tangentes en 16 bits par canal** pour un compromis entre la qualité et la mémoire supplémentaire nécessaire.

Cette option, “Use High Precision Tangent Basis”, est dans la section Build Settings de l’éditeur Static Mesh.



Résolutions personnalisées pour “Sky cubemaps” et “Reflection Probes”. Possibilité de choisir la résolution qui convient le mieux en fonction de la performance, la mémoire et les compromis de qualité.

“Cubemaps” personnalisés pour Reflection Probes pour détailler plus ou moins une partie.

“High Precision GBuffer Normal Encoding”: possibilité de sélectionner un format de codage GBuffer normal de haute précision.

Réflexions: surfaces translucides

4.8

Les surfaces translucides peuvent tenir compte dynamiquement des autres objets sur l'écran en activant les matériaux translucides avec l'option '**Screen space reflections**'.

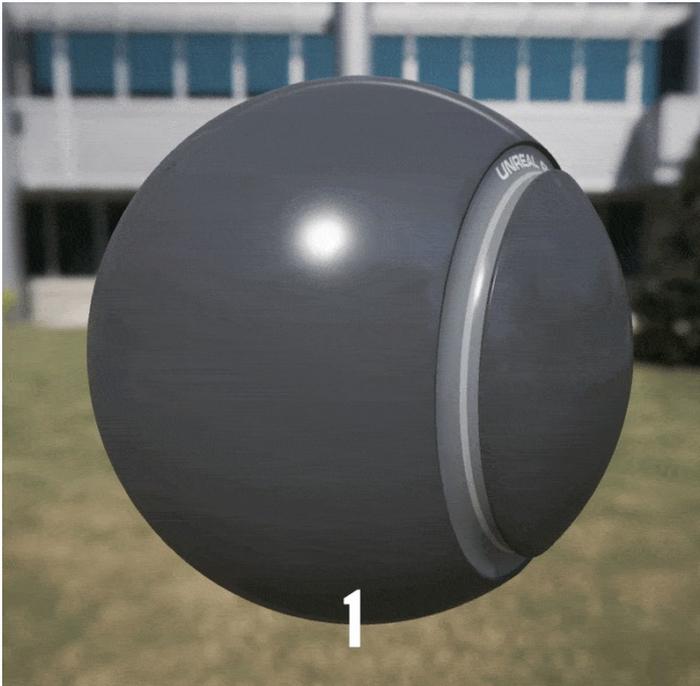


Plus besoin de shader maison, c'est supporté en standard désormais.

Mobile Skylight Reflections

4.20

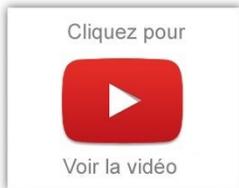
Le moteur de rendu mobile utilise maintenant un Skylight Cubemap pour les réflexions spéculaires lorsqu'aucune capture de réflexion n'est pertinente.



1 - Mobile, no reflection captures ; 2 - PC, no reflection captures

Bent Normal Maps and Reflection Occlusion: amélioration de l'éclairage

4.17



Les Bent normals, chékoédéo? Littéralement, des normals (perpendiculaires aux faces) tordues! Bon, ça n'explique rien je suis d'accord.

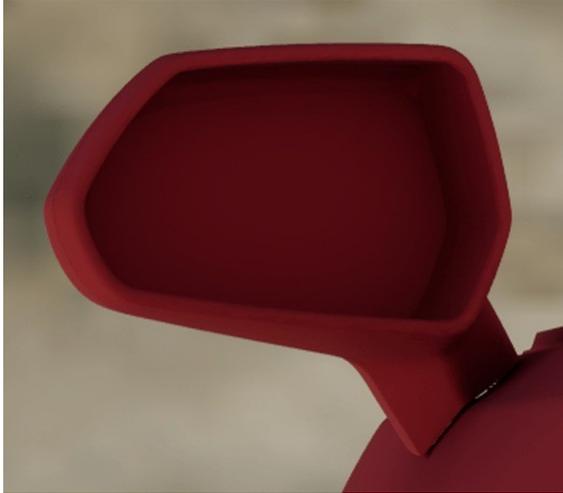
En fait, on parle surtout d'une map supplémentaire qui est utilisée pour le calcul de l'occlusion ambiante (AO). Je ne vais pas revenir sur ce qu'est l'AO, mais cela correspond grosso modo à une partie de l'ombrage que l'objet projette sur lui-même. C'est plus large que cela, mais cela donne une idée à ceux qui ne maîtrisent pas cet aspect. En quoi la Bent Normal Map apporte t-elle du réalisme ? On se rapproche d'avant du modèle d'illumination globale (GI) tout en conservant de bonnes performances. En fait, les bents normals sont utilisées à la place des normals donc il n'y a aucune perte de performance, tout est précalculé. Un algorithme qualifié d'un peu magique par Epic va tordre ces normals pour lui donner des détails supplémentaires améliorant le rendu indirect. Le plus gros impact étant que l'on perd moins de lumière durant l'occlusion.

Ces informations sont aussi utilisées pour **l'occlusion de la réflexion**. L'AO est traditionnellement une occlusion la lumière diffuse indirecte. La Reflection occlusion est l'occlusion pour lumière indirecte de **type spéculaire**. Cela fonctionne en coupant le lobe spéculaire visible avec le cône ou un cône représentant la partie non occluse de l'hémisphère décrit par la bent normal comme axe du cône et la quantité AO comme angle de cône.... Oulala, moi je traduis comme je peux, mais ces calculs d'illumination dépassent mes connaissances en lancé de rayon acquise à l'université ! Personnellement, je préfère implémenter l'algo correspondant à une formule plutôt que d'écrire cette dernière. Si cela vous intéresse, je vous conseille d'étudier le code source ! Ce qu'il faut retenir, c'est que cela peut réduire les fuites de lumière spéculaires de manière significative, en particulier lorsque les données SSR pour la réflexion ne sont pas disponibles.

Unreal Engine 4 – Guide des mises à jour (4.20)

Enfin, Epic a ajouté une approximation multi-rebond pour l'occlusion. Au lieu de l'AO ou d'un ombrage d'occlusion de réflexion au premier rebond, on utilise une approximation de ce qu'entraîneraient beaucoup de rebonds. Les matériaux brillants reçoivent moins de noir et les matériaux colorés sont plus saturés.

Avant:



Après:



Modèle d'éclairage spéculaire mobile amélioré

4.20

La réponse spéculaire mobile a été modifiée pour utiliser le modèle d'éclairage GGX par défaut. Si vous ne connaissez pas ce modèle, je vous renvoie [ce papier du Siggraph 2013](#) présenté par Brian Karis.

Cela améliore la qualité spéculaire mobile et correspond mieux au Shader Model 5 ([SM5](#)), mais ajoute un petit coût au temps de traitement des shaders.



1 - 4.20 spéculaire GGX par défaut; 2 - 4.19 Spéculaire sphérique gaussienne

Le modèle Spherical Gaussian Specular précédent est toujours accessible via l'option de projet 'Use legacy shading mode' et peut être trouvé sous Rendering > Mobile.

Ombrages avec plusieurs cartes normales

4.12

Possibilité d'utiliser plusieurs textures normales avec des matériaux de revêtement transparent pour obtenir un **ombrage réaliste pour des surfaces comme la fibre de carbone**.



Gauche: ex UE4 Clear Coat ,
Centre: nouveau UE4 Clear Coat avec fond normal,
Droite: Photo réelle – on en est pas loin non ?

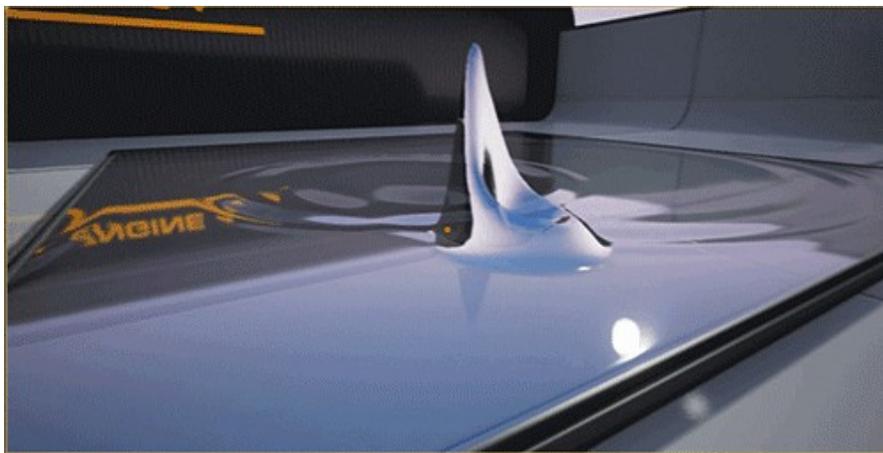
Cette option est activée via les paramètres du projet: Rendu -> Clear Coat Use Dual Normal.

Il faut placer un *Clear Coat Bottom Normal*.

Écrire directement en Blueprint

4.13

(donc sans programmation C++) ses propres matériaux et effets

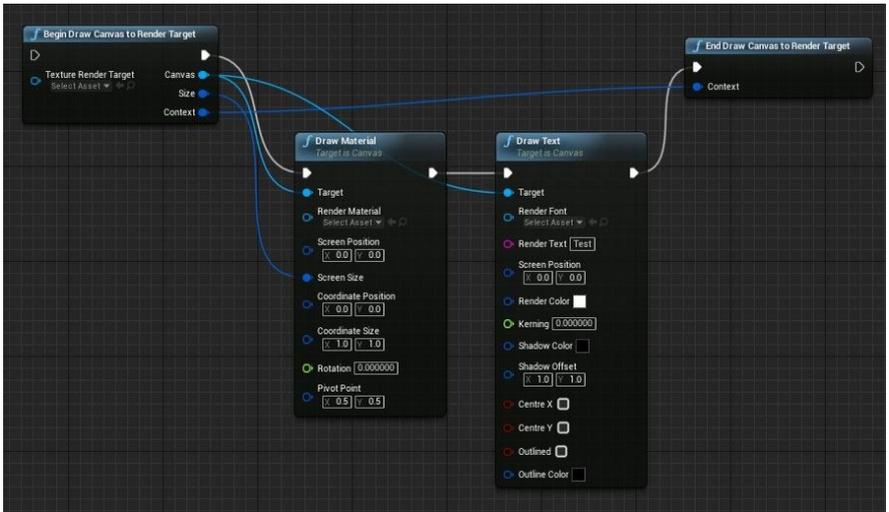


Exemple d'utilisation: Simulation de surface de fluide créée entièrement grâce aux Blueprints et matériaux. Les personnages et les projectiles influent sur le fluide.

La nouvelle fonction Blueprint **Draw Material to Render Target** pour les entrées de matériaux avec l'entrée **Emissive Color**.



Unreal Engine 4 – Guide des mises à jour (4.20)



Pour un travail plus avancé, il faut utiliser Begin Draw Canvas to Render Target et End Draw Canvas to Render Target comme ci-dessous:

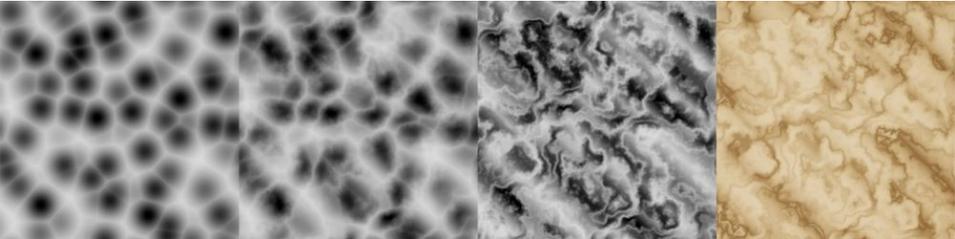
- La possibilité d'utiliser le GPU (\geq Shader Model 5) pour calculer le morphing de meshes
- l'amélioration de la réfraction de haute qualité,
- **les fonctions optimisées de bruit** permettent de générer des matériaux de manière aléatoire, y compris le bruit de **Voronoi** pour la génération procédurale et le marbre ou du sable:



Unreal Engine 4 – Guide des mises à jour (4.20)



Cet exemple utilise un bruit Voronoi avec une technique appelée «cartographie gradient» pour obtenir un look de marbre.



De gauche à droite:

1. La norme Voronoi Noise, 1 octave
2. Voronoi avec bruit "dégradé" ajouté à la position d'entrée à 0,05
3. le bruit Gradient multiplié par 0,3 avant d'ajouter à la position d'entrée Voronoi
4. En utilisant l'étape 3 comme coordonnées de texture pour un pavage texture aléatoire

Le bruit Voronoi comporte quatre niveaux de qualité, avec la diminution des artefacts mais avec une augmentation significative du temps d'ombrage.

La mise en cache de la carte des Ombres "SHADOW MAP CACHING"

4.13

Elle permet de mettre plus de lumières dynamiques, les calculs de lumière peuvent être réutilisés d'une image à l'autre, permettant d'améliorer les performances.



L'image ci-dessus montre 33 points dynamiques d'ombre avec des calculs minimes.

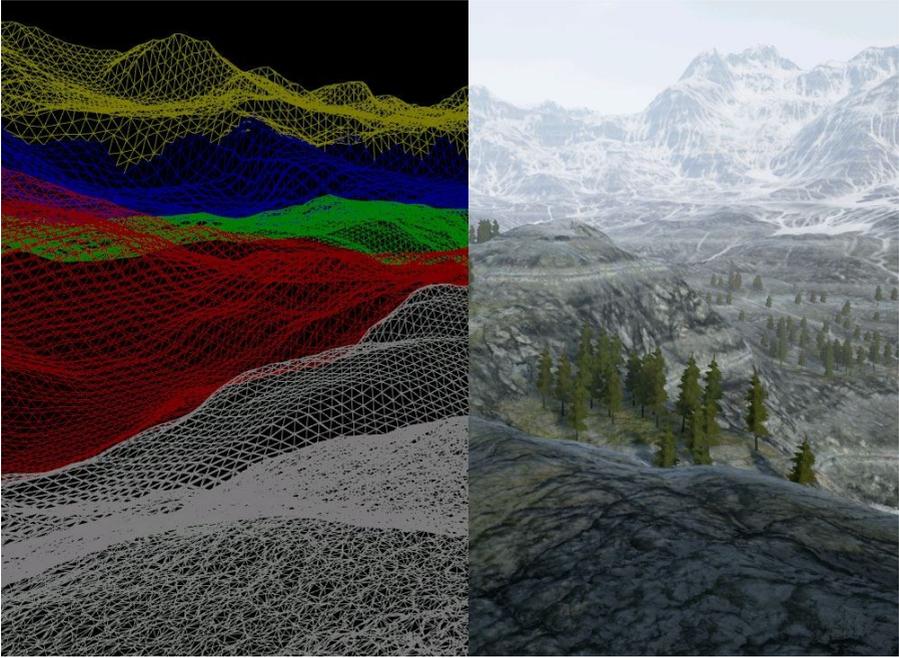
Par défaut, la mise en cache ne peut se produire lorsque:

- Primitives sont réglées en mobilité statique ou stationnaire
- Les matériaux utilisés n'utilisent pas la Position Offset du monde
- La lumière est un point ou une tache

Les matériaux qui utilisent une Tessellation d'animation ou une profondeur de Pixel Offset peuvent provoquer des artefacts.

Unreal Engine 4 – Guide des mises à jour (4.20)

Amélioration du **landscape tessellation** pour un LOD plus rapide et donc la possibilité de gérer des paysages encore plus grands !



Dans le même ordre d'idée, les composants de type scene captures ont été améliorés (comme le support des projections orthographiques).

Mesh preview scenes

4.13

Pour prévisualiser les meshes statiques et squelettiques.



Un nouveau panneau **Preview Scene Settings** permet de configurer plusieurs profils (scènes) pour prévisualiser vos meshes, et les profils permettent de changer:

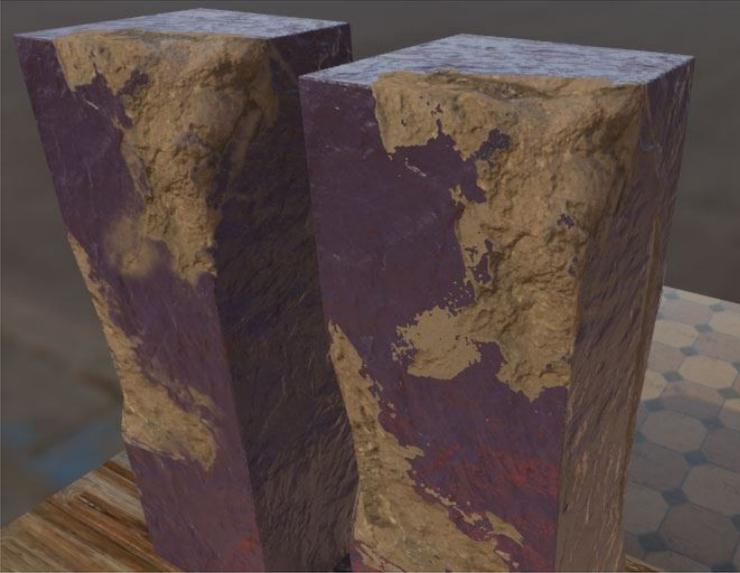
- Lumière directionnelle (couleur, intensité, rotation)
- lumière Sky (HDRi environnement map, intensité, rotation)
- paramètres de traitement



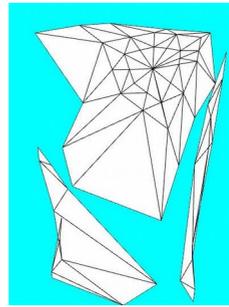
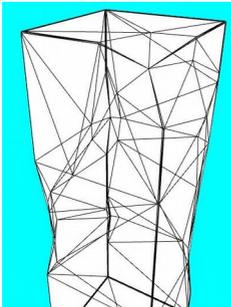
Mesh Decals

4.13

Permet de coller des matériaux sur les maillages statiques, et donc de superposer différents matériaux.



Les piliers ci-dessus sont créés en maillage de base superposée avec un maillage unique détaillé. Le pilier gauche utilise un decal de maillage pour permettre les mélanges de couleur, normals et rugosité. Le plus à droite pilier est un matériau masqué, juste pour la comparaison.



Le physical light units

Il s'agit de pouvoir gérer toutes les sources de lumières en utilisant des unités physiques (lumens et candelas) - certaines sources étaient bien définies, mais d'autres utilisaient des unités spécifiques à UE4.

La compatibilité ascendante est assurée, mais désormais, ces caractéristiques physiques sont disponibles sur chaque source de lumière, ainsi que sur l'éclairage global de la scène.



Apparition des volumetric lightmaps

4.18

Les surface lightmaps servent au pré-calcul de l'éclairage indirect sur les objets de type statiques, c'est à dire, n'étant pas en mouvement. Mais ces dernières ignorent les objets dynamiques comme les personnages, les effets de particules, ou la végétation brassée par le vent...

Pour cela, UE a introduit les **volumetric Lightmaps**.

Évidemment, on ne peut pas utiliser ces cartes de la même façon, car on ne connaît pas la position des objets avant l'exécution. De fait, ces volumes stockent une information similaire, mais dans une sorte de cache qui sera utilisé à l'exécution pour réaliser des interpolations entre valeurs, selon la position des différents objets.

Comme il ne s'agit plus de surface, mais d'avoir une information "en tous points de l'espace", on parle de volume... ou plus précisément de briques composées de 4x4x4 cellules. En réalité, c'est un peu plus complexe, car cela va dépendre du fait que l'objet existait ou non à l'origine.

Un plus grand nombre de briques sont instanciées lorsqu'elles sont à proximité des objets statiques. Mais le principe est là.

Comparaison détaillée:

Volumetric Lightmap	Volumes de lumière précalculés / cache d'éclairage indirect
Interpolation efficace sur GPU	Interpolation coûteuse sur le processeur
Interpolation par pixel	Interpolation par objet, même sur des composants instanciés
Appliqué au brouillard volumétrique	Ne peut pas affecter le brouillard volumétrique
Placé avec une densité	Placé avec une densité élevée

Unreal Engine 4 – Guide des mises à jour (4.20)

Volumetric Lightmap	Volumes de lumière précalculés / cache d'éclairage indirect
élevée autour de toutes les surfaces statiques	seulement sur les surfaces statiques, fuite fréquente des échantillons à faible densité
Les voxels frontaliers sont étirés pour couvrir la zone extérieure Importance Volume	Noir indirect Extérieur Importance Volume
Ne prend pas actuellement en charge le streaming de niveau	Prend en charge le streaming de niveau

Par l'ancienne méthode:



Unreal Engine 4 – Guide des mises à jour (4.20)

Par la nouvelle:



Si vous voulez faire vos propres tests, n'oubliez pas de cocher la case "Dynamic Indirect Lighting" sur votre source lumineuse ;)

Nouveau moteur de rendu d'ombrages

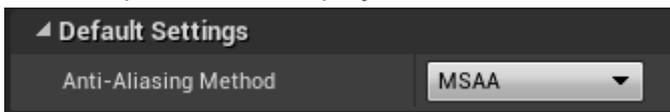
4.14

(optimisé pour la VR entre autres)

Il permet de combiner les fonctionnalités d'éclairage avec l'antialiasage à plusieurs échantillons (MSAA), le MSAA est préférable **pour les projets VR**.



Pour utiliser MSAA, définissez la valeur par défaut Anti-Aliasing Méthode dans les paramètres du projet de rendu:



Les **fonctionnalités** gérées sont par exemple:

- les lumières stationnaires en prenant en compte les objets en mouvement (ombres dynamiques)
- les réflexions multiples et planaires
- Toutes les lumières pré calculées (lightmaps)
- les ombres capsules

Ombres de contact (Contact Shadows)

4.14

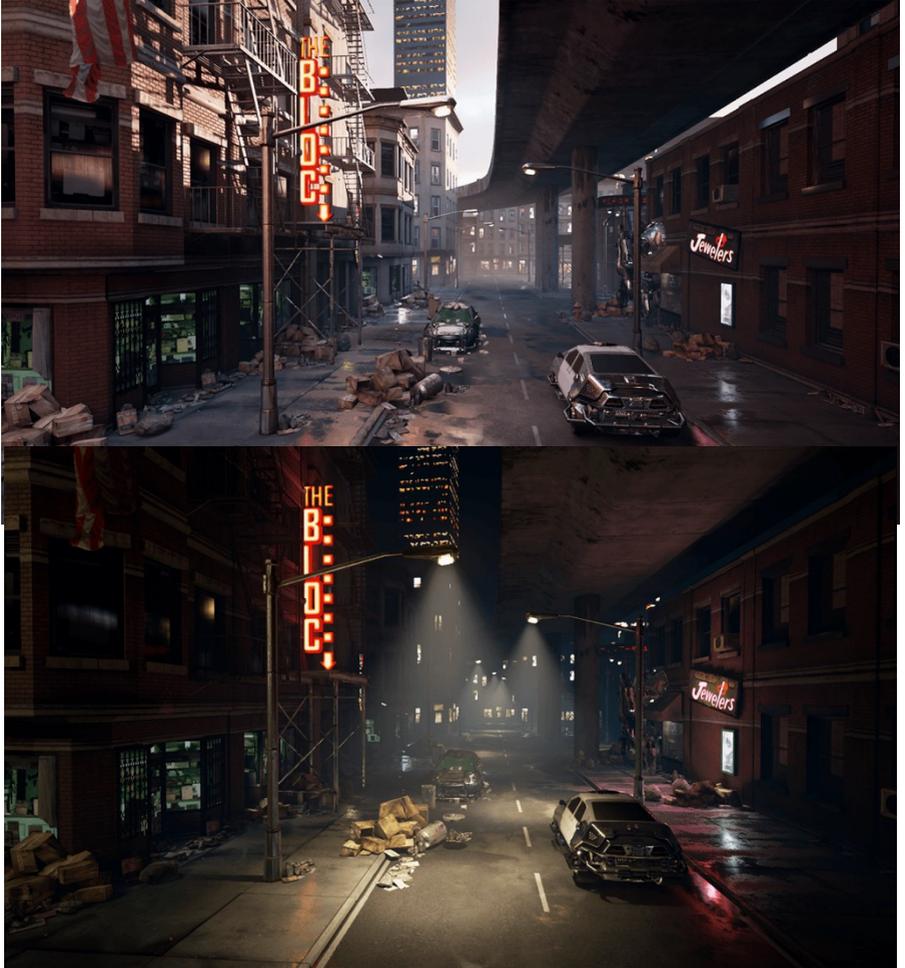
Les nouvelles fonctionnalités **Ombres de contact** (Contact Shadows) permettent un excellent détail des ombres dynamiques des objets complexes.



Les scenarii d'éclairage

4.14

L'éclairage précalculé pour de multiples configurations d'éclairage avec la même géométrie est maintenant possible.



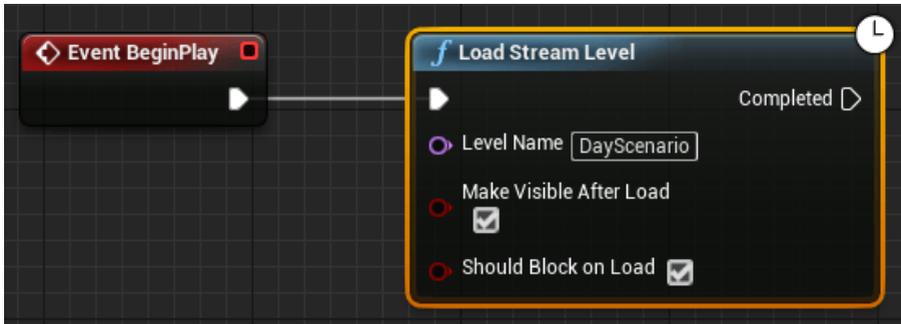
Dans l'exemple ci-dessus, la lumière directionnelle, la lumière du ciel et skybox ont été placés dans un scénario d'éclairages appelé

Unreal Engine 4 – Guide des mises à jour (4.20)

DayScenario. Les lampadaires ont été placés dans **NightScenario.** Vous imaginez les gains de productivité de votre workflow quand il s'agit de pouvoir utiliser les mêmes lieux à différents moments de la journée ?

Pour utiliser les scénarios d'éclairage:

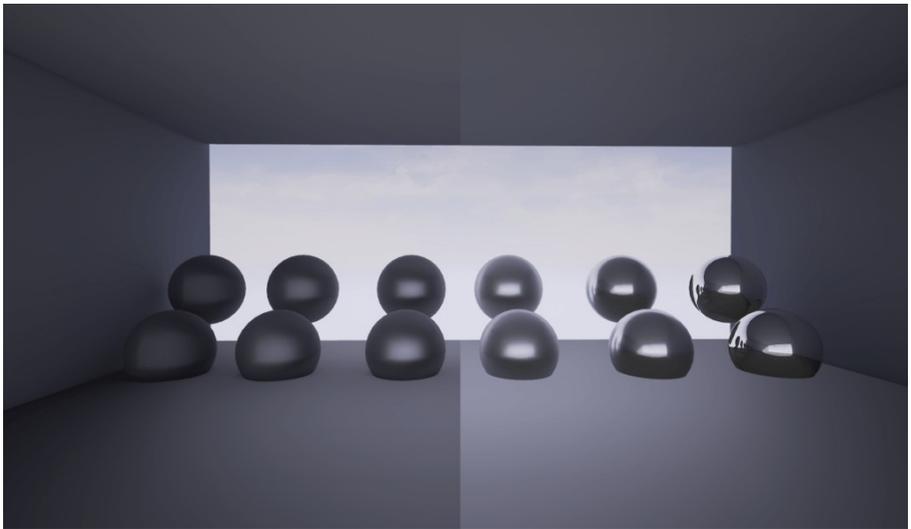
- clic droit sur un sous-niveau dans la fenêtre level windows et changer pour Lighting Scenario. Quand un niveau de scénario est rendu visible, ses lightmaps sont appliquées dans le monde.
- Changer la méthode de diffusion de niveau Blueprint au niveau du scénario d'éclairage
- Placer les meshes et les lumières dans ce niveau et construire l'éclairage
- Dans le BeginPlay du niveau du level blueprint du niveau persistant, exécuter un Load Stream Level au niveau du scénario d'éclairage.



Amélioration de la qualité de la "Reflection Capture"

4.14

Quand on utilise les **"Reflection Capture"** (voir tome 3 des cahiers d'unreal Engine), le moteur réalise un mix entre l'émission spéculaire provenant de la capture avec la lumière indirecte émise par la lightmap.



A gauche, on utilise le mix, et pas à droite. Cela donne de bons résultats sur les surfaces rugueuses, mais pas sur les surfaces lisses. De fait, le mixage est désormais désactivé pour toute surface ayant un roughness $> .3$, puis un gradient (fade) sera appliqué jusqu'à $.0$. Cela donne ainsi de meilleurs résultats.

Et cela a d'ailleurs un impact sur l'environnement direct comme illustré ci-dessous:



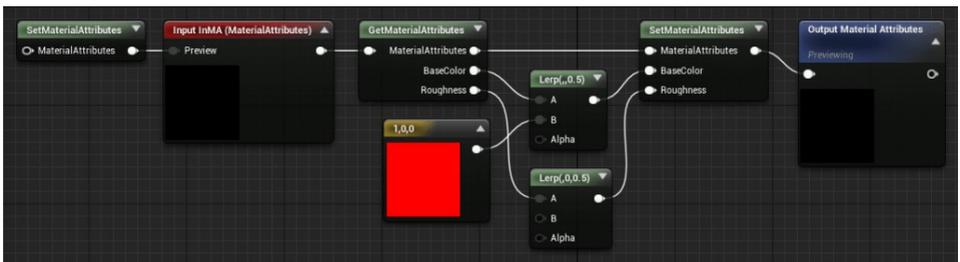
Material attribute nodes

4.14

Travailler avec les **Material Attributes** est maintenant plus facile.

- **GetMaterialAttributes** - Ce nœud est un remplacement compact pour BreakMaterialAttributes
- **SetMaterialAttributes** - Ce nœud est un remplacement compact pour MakeMaterialAttributes
- **BlendMaterialAttributes** - nouveau nœud pour permettre de faire plus facilement le mélange des matières Attributs structures.

La sélection d'un nœud montre la liste des pins actuelles dans le panneau de détails qui peuvent être étendues ou supprimés. Par exemple, la fonction ci-dessous prend un ensemble d'attributs mélange ensuite la couleur de base et Roughness à une surface rouge brillant.



Vector Noise dans les matériaux

4.14



| Cellnoise | Vector Noise | Gradient | Curl | Voronoi |

Le nouveau nœud "vector noise" ajoute plusieurs fonctions avec des vecteurs 3D ou 4D.

Tout ou partie du calcul être précalculé dans une texture en utilisant le Blueprint **Draw Material to Render Target** introduit en 4.13.

Ces nœuds fournissent une alternative à la création de textures générées avec un outil externe. Les nouvelles fonctions sont les suivantes :

1. **Cellnoise** : Renvoie une couleur aléatoire pour chaque cellule dans une grille 3D. Les résultats peuvent donc fournir un moyen fiable pour ajouter un effet aléatoire à un matériau.

2. **Perlin 3D Noise**: Calcule une version de Perlin Noise Simplex avec sortie vecteur 3D. Chaque composant de sortie est compris entre -1 et 1.

3. **Perlin Gradient**: Calcule le gradient 3D analytique d'un scalaire bruit Perlin Simplex. La sortie est sur quatre canaux, où les trois premiers (RGB) sont le gradient, et le quatrième (A) est le bruit scalaire. Ceci est utile pour les bosses et pour les flow maps sur une surface

4. **Perlin Curl**: Calcule la boucle 3D d'un vecteur de bruit Perlin Simplex. La sortie est un vecteur 3D. Ceci est utile pour l'écoulement de fluides ou de particules.

Unreal Engine 4 – Guide des mises à jour (4.20)

5. **Voronoi**: Calcule le même bruit Voronoi que le nœud de matériau scalar Noise. Le bruit de Voronoi scalaire éparpille les points dans l'espace 3D et renvoie la distance la plus proche. La version vectorielle du bruit renvoie l'emplacement du point le plus proche en RGB, et la distance en A. Surtout couplé avec Cellnoise, cela peut permettre à un comportement aléatoire par cellule de Voronoi.

Ci-dessous, un matériau pierre réalisé en utilisant le composant à distance du vecteur de bruit / Voronoi pour moduler quelques bosses de surface et se fondre dans la mousse dans les fissures, et la position de la graine avec Vector Noise / Cellnoise pour changer la couleur et hauteur des bosses par roche.



Encore des éléments qui ajoutent aux possibilités de génération procédurale d'UE4, afin d'ajouter plus d'aléatoire dans les matériaux.

Améliorations du tonemapper

4.15



Le tonemapper fournit des couleurs et des tons plus semblables à un ceux d'un film.

Le tonemapper a maintenant l'[Academy Color Encoding System \(ACES\)](#) par défaut. Ce tonemapper permet également au moteur de cibler des types d'affichage multiples, y compris l'affichage **HDR**.

Vous pouvez revenir à l'ancienne version en ajoutant `TonemapperFilm=0` dans votre fichier `DefaultEngine.ini`. Tout du moins, pour l'instant.

Amélioration de la gestion du streaming des textures

4.15



Permet de réduire l'utilisation du CPU, de la mémoire (réduction de 40%) et les temps de chargements (2 à 3 fois) tout en éliminant les artefacts de basse résolution (calculés par matériau et non plus par maille et prenant en compte les LOD).

C'est pour moi **l'une des améliorations les plus importantes** car elle touche fondamentalement nos développements.

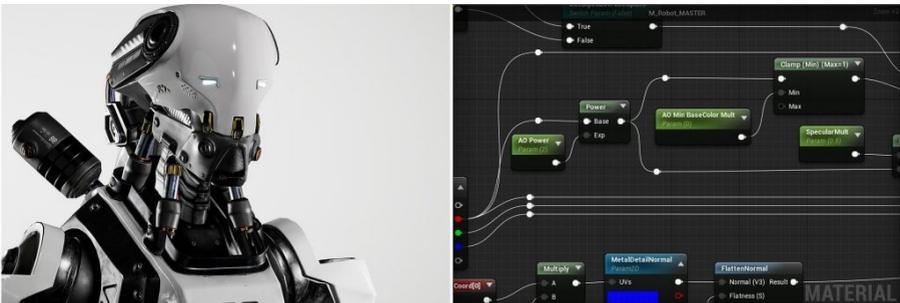
Material editor

Déplacement de nœuds dans le material editor

4.15

Réacheminer les nœuds est maintenant possible dans l'éditeur de matériaux pour permettre une meilleure organisation des graphiques matériaux.

On pouvait le faire dans les Blueprint, vous savez ces petits noeuds de reroutage - c'ets juste pour le visuel, mais ça permet d'être plus propre au final.



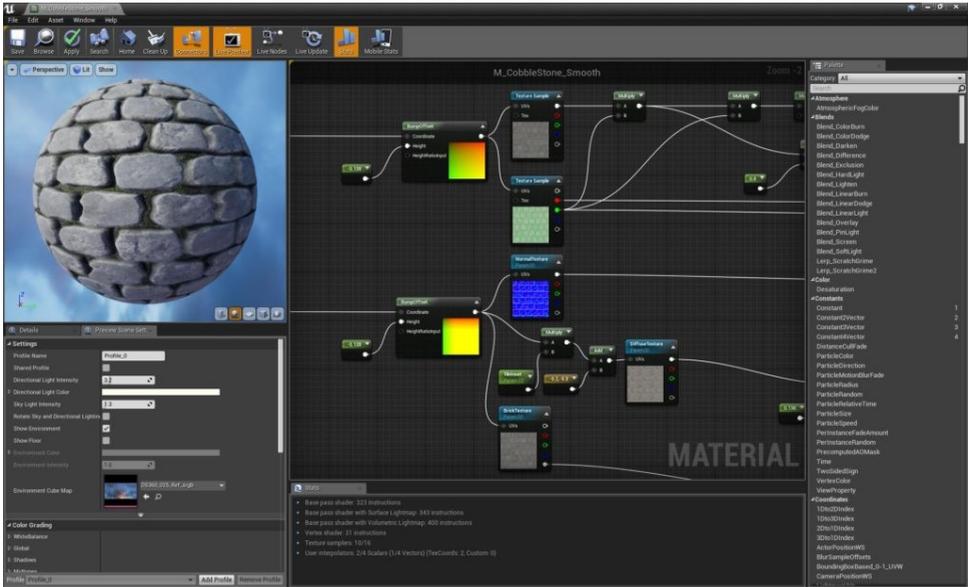
Pour ajouter des nœuds dans Material Editor: Dans la Palette, Faites glisser "Add Reroute Node" au graphique, ou, double-cliquez sur une spline de connexion pour insérer une nouvelle reroute au nœud.

Support de prévisualisation avancé de Material Editor

4.18

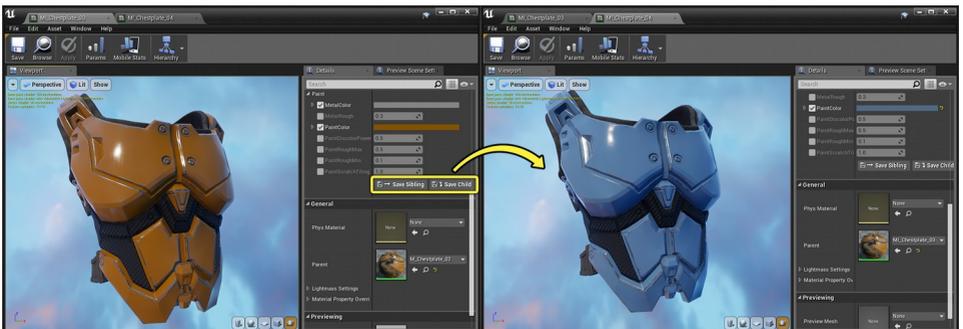
L'éditeur de matériaux et l'éditeur d'instance de matériaux ont maintenant un système de prévisualisation avancée "Advanced Preview Scenes" qui permet de modifier les paramètres de la scène tels que l'éclairage, les environnement cube maps, ainsi que la possibilité de faire pivoter l'environnement autour du preview mesh.

Unreal Engine 4 – Guide des mises à jour (4.20)



Instance enfant ou fratrie dans l'éditeur de matériaux

Ainsi que dans l'éditeur d'instance de matériau... Vous pouvez maintenant enregistrer les valeurs des paramètres dans une nouvelle instance enfant ou dans une nouvelle instance de fratrie dans l'éditeur de matériaux et dans l'éditeur d'instance de matériau!

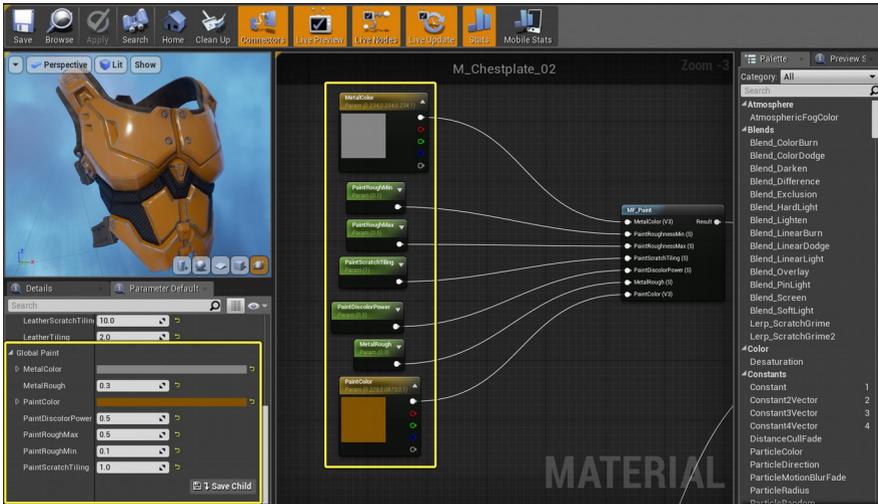


Unreal Engine 4 – Guide des mises à jour (4.20)

"**Save to Sibling Instance**" permet d'enregistrer les paramètres et les remplacements actuels dans une nouvelle instance Material avec le même matériau parent.

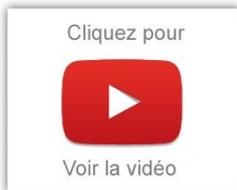
Save to Child Instance dans l'instance enfant permet d'enregistrer les paramètres et les substitutions en cours dans une nouvelle instance de matériau prenant l'instance actuelle comme matériau parent.

Pour modifier les valeurs par défaut



Material Layering (expérimental)

4.19



La superposition de matériaux permet de combiner les matériaux dans une pile, en utilisant les nouveaux assets Material Layer et Material Layer Blend! Cela permet de créer un graphe matériel correct sans faire manuellement des sections de nœuds. Cette fonctionnalité est similaire à Material Functions, mais prend en

charge la création d'instances enfant.

Pour l'activer: Project Settings > Rendering > Experimental en définissant sur vrai Support Material Layers

Sortie HDR (High Dynamic Range)

4.15

pour Windows **Direct3D 11** (**NVIDIA** uniquement), et Mac en profitant de fonctionnalités comme le contraste plus élevé et une plus large gamme de couleurs.

Les images suivantes sont des images simulées et sont uniquement à des fins de démonstration -

Je suppose que la news s'applique sur la photo de gauche qui est, en effet, plus contrastée. Pour ma part, je ne vais pas pouvoir, car je suis sur une Radeon - je sais, moi aussi je préfère Nvidia, que cela soit pour les **outils de gaming** ou pour **CUDA**- mais j'ai fait un **compromis qualité/Prix pour la VR**.

Unreal Engine 4 – Guide des mises à jour (4.20)

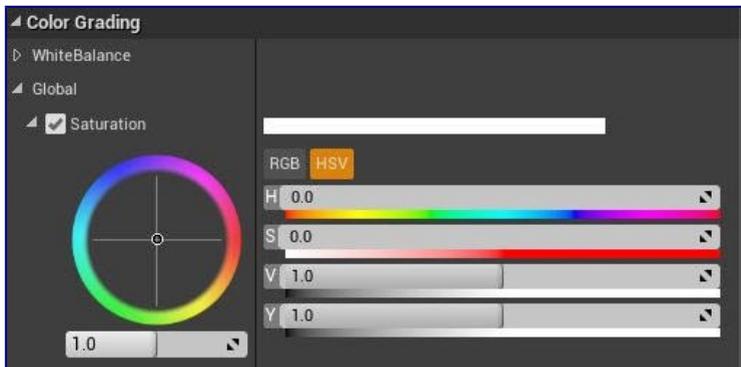


D'autres dispositifs sont à venir dans les prochaines versions (hé ben oui, **Ars Longa Vita Brevis**- ne vous attendez pas à vous reposer sur vos lauriers). L'implémentation actuelle gère le traitement complet de la scène rendue par la norme ACES. Le support de sortie HDR est détecté au lancement et signalé seulement si un périphérique le prenant en charge est connecté. La sortie HDR peut être modifiée par les nouvelles variables de la console ou par GameUserSettings.

Color Grading controls

4.16

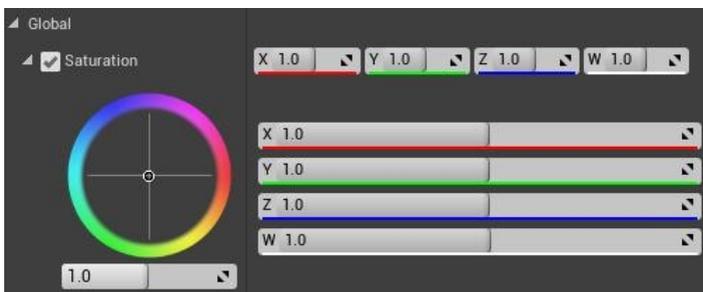
L'interface utilisateur Color Grading est devenue plus facile à utiliser:



Un nouveau mode HSV a été ajouté, ce qui permet aux graphistes plus habitués à ce mode de choisir plus facilement la couleur. Il est possible de modifier dynamiquement la valeur min / max des curseurs en fonction de leur type en utilisant les touches Ctrl + Slider Drag. Un nouveau bouton de remise à zéro a été ajouté pour réinitialiser une catégorie de classement couleur ensemble. (i.e. global, ombres, démons, Highlights)

4.15

nouveaux contrôles pour les propriétés Color Grading dans Post process settings pour une expérience plus intuitive et un contrôle plus précis.



Outils de visualisation de texture streaming debugging

4.15

3 nouveaux view modes sont disponibles pour s'assurer que le système streaming de texture fonctionne de manière optimale pour le jeu.



1. **Primitive Distance Accuracy:** montre la précision de la distance calculée par le streamer par rapport à la distance calculée par le GPU.
2. **Mesh UV Densities Accuracy:** montre l'effet des densités UV utilisées par le streamer pour chaque mesh. Si les densités UV sont fausses, les textures sur les meshes risquent d'être floues ou sur-streamées.
3. **Material Texture Scales Accuracy:** Certaines textures utilisées dans un matériau sont échantillonnées en utilisant une valeur à l'échelle de l'un des canaux mesh UV. Cela tente de calculer les coordonnées de la texture à utiliser et l'échelle à appliquer à chaque texture échantillonnée.

Mathematics material nodes

4.15

ajout de quelques nœuds de matériaux couramment demandés pour éviter l'utilisation de fonctions ou de noeuds personnalisés pour les opérations courantes. Les nœuds marqués du tag "Fast" exécutent des **approximations au lieu des instructions réelles**. C'est top d'y avoir pensé, ne trouvez-vous pas ? Ils améliorent la performance de matériaux plus complexes, mais ont des restrictions d'entrée et les

Unreal Engine 4 – Guide des mises à jour (4.20)

compromis de précision. A nous de voir si visuellement ça change vraiment la donne... et au pire, les utiliser sur mobiles par exemple ! J'aime bien qu'on nous laisse un peu de boulot aussi à nous développeurs !

Ça rappelle [le calcul de la racine carrée inverse](#) dans Quake 3 arena (et plus vieux que ça d'ailleurs). Ça me rappelle aussi les tables de précalcul (sinus et autres fonctions) qu'on utilisait en [assembleur](#) pour faire des démos et éviter des calculs ! Bon, ça manquait cruellement de précision toutefois ;)



Previous Frame Switch node

4.15

Si vous voulez faire des animations de type "motion blur", vous savez le fait de voir encore en partie la forme précédente pour simuler un effet de vitesse, et ceci en exploitant la persistance rétinienne... hé bien, il serait fort utile d'accéder aux informations de la trame précédente, non ? Ceci est rendu possible par ce nouveau noeud. Je ne sais pas encore quelles informations il fournit, mais je ne vais pas tarder à le savoir. Ce n'est pas l'image à proprement parler, mais un ensemble de paramètres. [A tester...](#)

Pre-Skinned Local Normals material Node

4.15

Vous pouvez maintenant utiliser cette [fonction](#) pour obtenir des effets comme la création automatique des maps des UV sur les skeletal meshes en utilisant des matériaux tri-planaires alignés.

Cela fonctionne d'une manière similaire à **Pre-Skinned Local Position**, mais renvoie la normale à la surface locale pour Skeletal Meshes et statiques Meshes. Cela ouvre la porte à plus d'effets possibles dans les matériaux.

Une nouvelle méthode de sur-échantillonnage temporel (Temporal Upsampling)

4.19

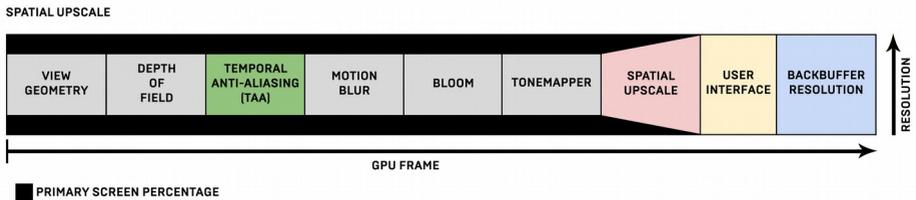
Pour bien comprendre ce dont il s'agit, je vais devoir vous introduire le **Screen Percentage** : c'est une technique utilisée pour faire du rendu à une résolution inférieure ou supérieure à celle qui est présentée. On parle de % par rapport au rendu affiché.

Avant la version 4.19, il s'agissait de modifier ce coefficient. Maintenant, il y a deux types de coefficients :

1. Le primary Spatial Upscaling : On effectue un rendu à une résolution plus faible, puis on **effectue un upscale avant que l'UI** ne soit affichée. C'est un agrandissement, un upsampling (de sample, échantillon, et up pour sur-échantillonnage).
2. Le secondary Spatial Upscaling : on réalise une seconde passe d'Upscale, mais indépendamment de la première.

Unreal Engine 4 – Guide des mises à jour (4.20)

Si on effectue un rendu avec un coefficient supérieur à 100%, on diminue la résolution de l'image pour l'affichage, c'est ce qu'on appelle du **super-sampling**.



Les différents calculs dans le pipeline de rendu sont réalisés juste avant l'User Interface. Cela signifie que le Bloom, le Motion Blur, etc. - tout cela est réalisé sur une image de plus basse résolution, et c'est l'ensemble qu'on "grossit" pour l'affichage.

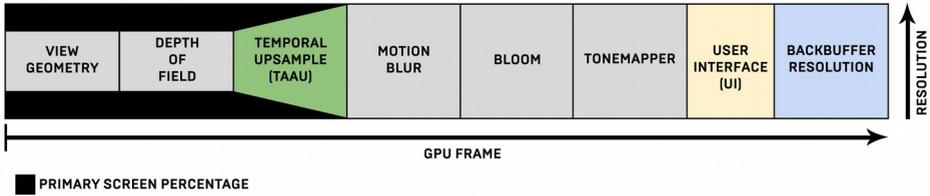
La qualité de l'Upscale se règle avec la variable Spatial Upscale Quality, correspondant à différents niveaux (0=Nearest Filtering, 1=Simple Bilinear, ... 5 =36-tap Gaussian-filtered unsharp mask). Le niveau 5 étant très couteux en ressources, mais donnant de bons résultats quand les % sont assez faibles - on parle alors d'extrem upsampling.

Bon pour l'instant, on ne parle que de "spatial", c'est à dire de résolution, de dimension d'espace. Alors, c'est quoi ce temporel dans le titre ?

Introduisons le "Temporal Anti-Aliasing Upsample" (TAA Upsample). On est toujours dans la passe primaire. Au final, plutôt que de travailler dans la résolution inférieure, on va faire le upscale et travailler dans la résolution supérieure. Cela donne:

Unreal Engine 4 – Guide des mises à jour (4.20)

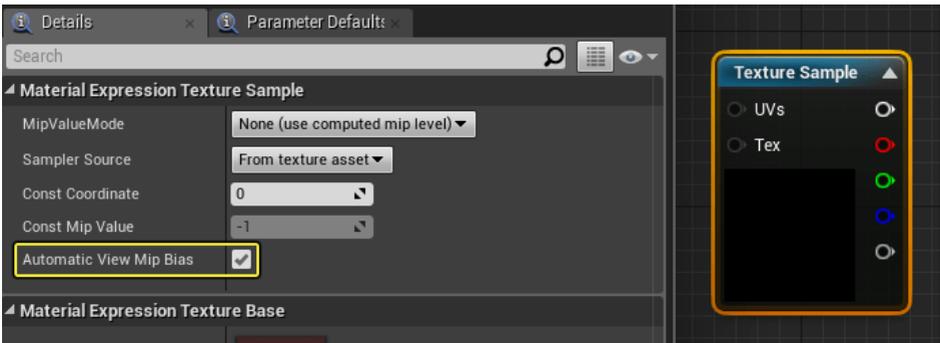
SPATIAL AND TEMPORAL ANTI-ALIASING UPSAMPLE



De fait, l'antialiasing, le Motion Blur, le Bloom, le tonemapper... tout cela bénéficie de plus d'éléments pour le calcul et effectue donc un rendu de meilleure qualité.



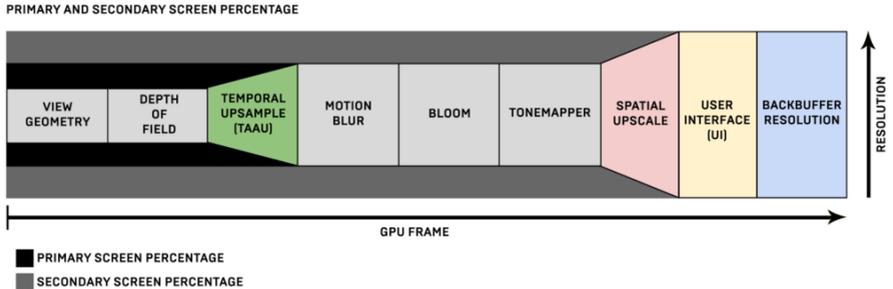
Pour que l'upsampling puisse donner de bons résultats, il faut que l'upsample puisse accéder à une meilleure résolution de texturation de l'objet - et cela, c'est le Mip Mapping, c'est à dire le fait que plus un objet est loin et petit, et plus on va utiliser une texture de petite résolution - tout cela est automatique bien entendu. Pour qu'un material puisse automatique switcher de résolution lors du TAAU, il faudra que cette case soit cochée:



Mais pour des textures dites "haute-fréquence", c'est à dire à changement rapide de contraste (comme le grillage ci-dessus), cela peut donner un effet de scintillement et il est alors recommandé de décocher cette case.

Unreal Engine 4 – Guide des mises à jour (4.20)

Ceci m'amène à vous parler du Secondary Spatial Upscale. Il est réalisé après la passe "TAAU". On comprend mieux avec le schéma:



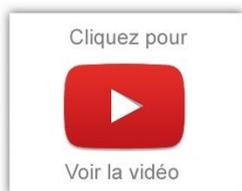
La résolution de tout ce qui suit la TAAU n'est pas obligatoirement celle qui sera utilisée au final. C'est une façon d'améliorer, mais en ayant un curseur.... plus on s'approche de la résolution d'affichage et plus l'effet sera joli, mais plus il sera coûteux. Ainsi, le second Spatial Upscale intervient après le Tonemapper, juste avant l'UI.

A cela, il faudrait ajouter la densité de pixel pour la VR, mais je vous renvoie pour cela à [cette page](#).

ANIMATIONS / SEQUENCER

Amélioration du media framework

4.18



Le Framework Media prend désormais en charge la composition vidéo en temps réel, la synchronisation temporelle, les workflows simplifiés et la capture vidéo (matérielle et VoIP). Sa stabilité et ses performances ont été également améliorées, ce qui n'est pas plus mal car je ne sais pas si vous avez essayé avant de passer des vidéos 4K type 360° plaquées dans une sphère... c'était pas top. Espérons que cela aille mieux désormais ;)

La lecture audio est maintenant intégrée à la nouvelle API Audio Mixer et utilise de nouveaux composants Sound Media (remplace Media Sound Waves) qui sont construits sur des composants de synthèse pouvant être attachés à des acteurs de niveau et utilisés dans des widgets UMG.

Improved Animation Preview Scenes

4.16

Plusieurs améliorations pour un aperçu des scènes d'animation: **Preview scene settings** a été déplacé dans l'onglet paramètres existant, ajout d'un raccourci pour basculer rapidement preview mesh de la barre d'outils principale.

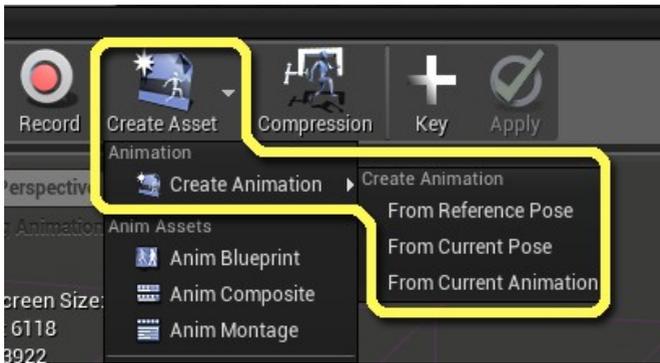
Lors de l'édition des scènes de prévisualisation, vous n'avez plus à créer un asset « preview scene collection » juste pour un aperçu de mailles supplémentaires. Si vous êtes satisfait de votre configuration de maille, vous pouvez maintenant l'enregistrer à un actif.

Morph Target Exaggeration

4.7

Il est désormais possible d'exagérer le morphing: une sorte d'interpolation au delà des limites prescrites, ce qui peut être bien pratique par moment.

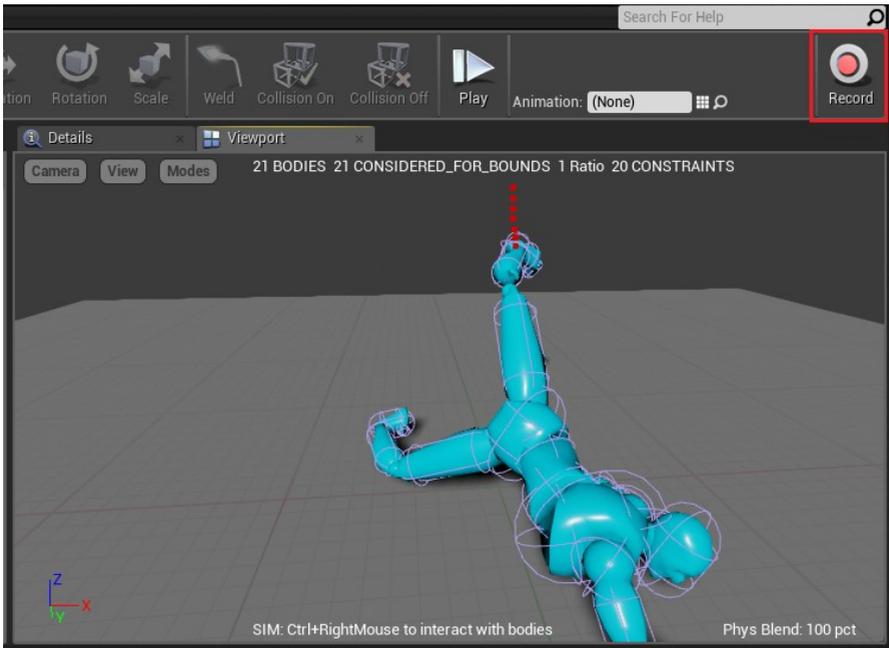
Et cela marche dans les deux sens:



Création d'Animations à partir de la physique

4.7

L'éditeur de physique permet maintenant d'exporter les simulations sous forme d'animation!



Il suffit de cliquer sur "Record" dans la toolbar, d'interagir avec l'objet dans la viewport pour créer une simulation intéressante! Après, on clique sur "Stop" pour sauver la simulation en animation.

On peut aussi le faire à partir de l'éditeur de niveau: on clique sur "Simulate" dans la toolbar (play), on cherche les Skeletal Mesh qui sont en mode simulation dans la scène, clic droit sur le mesh et sélectionnez "Keep Simulated Changes" (ou touche 'K') pour sauver sous forme d'animation.

Animations Multi-threaded

4.7

C'est quand même un point vachement important: les animations de type skeletal demandent un certain temps de calcul à cause des interpolations et des différentes transformations. Avant, toutes les animations étaient gérées dans une seule tâche: il n'était donc pas possible de les répartir entre les processeurs.

Mais aujourd'hui, tout le monde dispose d'un ordinateur avec plusieurs CPU et plusieurs cœurs! Je dirais que ce n'était donc pas trop tôt !

Configurable Animation Notify Triggers

4.8

Nouveaux paramètres sur tous les notificateurs :



Unreal Engine 4 – Guide des mises à jour (4.20)

- **Notify Trigger Chance** : nombre compris entre 0,0 et 1,0 pour indiquer le % de déclenchements (0.0 = Jamais, 1.0 = Toujours)

Pour l'émission de particules par exemple, pas besoin de passer par un BP avec un random.

- **Notify Filter Type**- Pour filtrer quand il faut notifier, 2 choix:
 - Pas de filtrage - notifications toujours actives (en supposant qu'il passe le paramètre Trigger Chance)
 - LOD - déclenchement si le maillage propriétaire est en dessous de la limite de détection.

Motion Blur et antialiasing pour l'animation de Vertex dans les matériaux

4.8

Les matériaux utilisant le World Position Offset pour animer des vertex génèrent des "motion vectors".

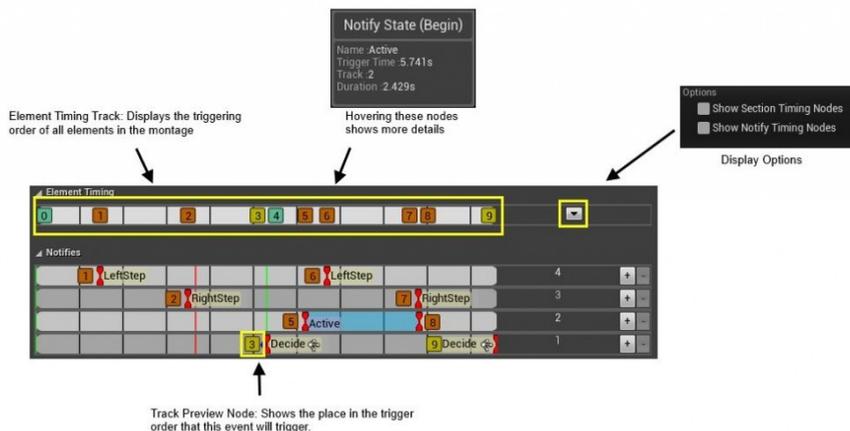
Ces derniers permettent un motion blur per-pixel animé de façon procédurale.

Anti-Aliasing et Distance Field Ambient Occlusion peuvent être utilisés en sus.

Interface de distribution séquentielle d'Element

4.9

Un nouveau panneau à l'éditeur de Montage aide à contrôler l'ordre séquentiel des événements lors de vos animations.



Non-Linear Animations Blending

4.9

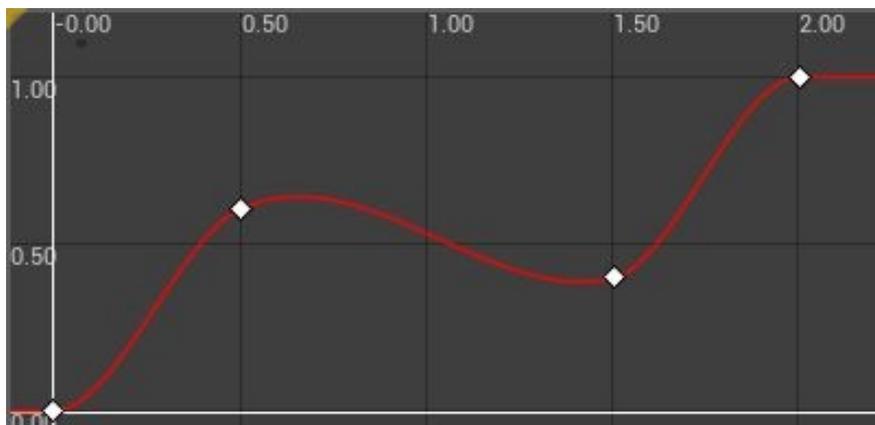
De nombreuses fonctions de mélange sont supportées entre différentes animation et transitions!

Auparavant, seul le mélange linéaire ou cubique était pris en charge. Maintenant, vous pouvez choisir parmi de nombreuses fonctions de mélange différentes: linéaire, cubique, Hermite cubique, sinusoïdale, quadratique, circulaire, exponentielle et personnalisée.

L'option «Custom» exige un asset «Float Curve».

Unreal Engine 4 – Guide des mises à jour (4.20)

Après en avoir créé un, vous pouvez l'affecter dans le panneau de détails pour les nœuds concernés :



Cela permet de spécifier à peu près tout mélange que vous souhaitez effectuer.

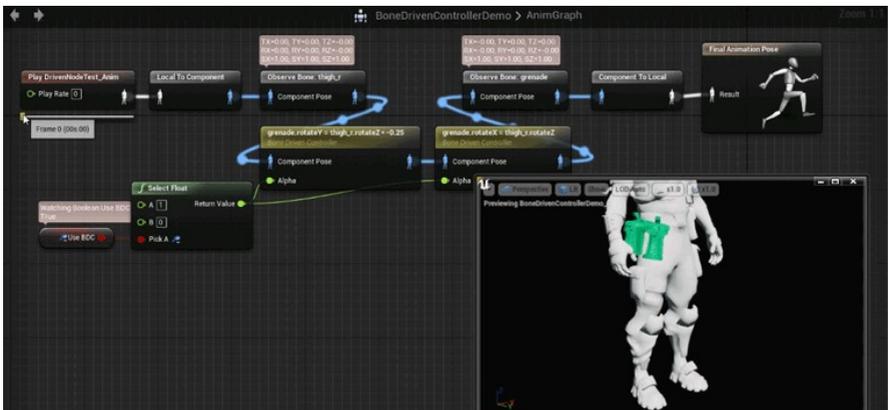
La longueur de la courbe est normalisée et réduite afin de respecter le délai de mélange spécifié et toutes les valeurs qui sont en dehors de la gamme 0-1 sont ajustées. Si une courbe de mesure n'est pas spécifiée, le système retombe à un mélange linéaire.

Contrôleurs Bone animations

4.9

Cela permet à un Bone "Driver ou Pilote" d'affecter dynamiquement le mouvement d'un ou plusieurs bone(s) "Driven".

Ce qui est **utile pour les personnages avec des accessoires attachés!** Vous pouvez éviter la géométrie d'intersection lors d'une animation, même si beaucoup de mélange sont utilisés.



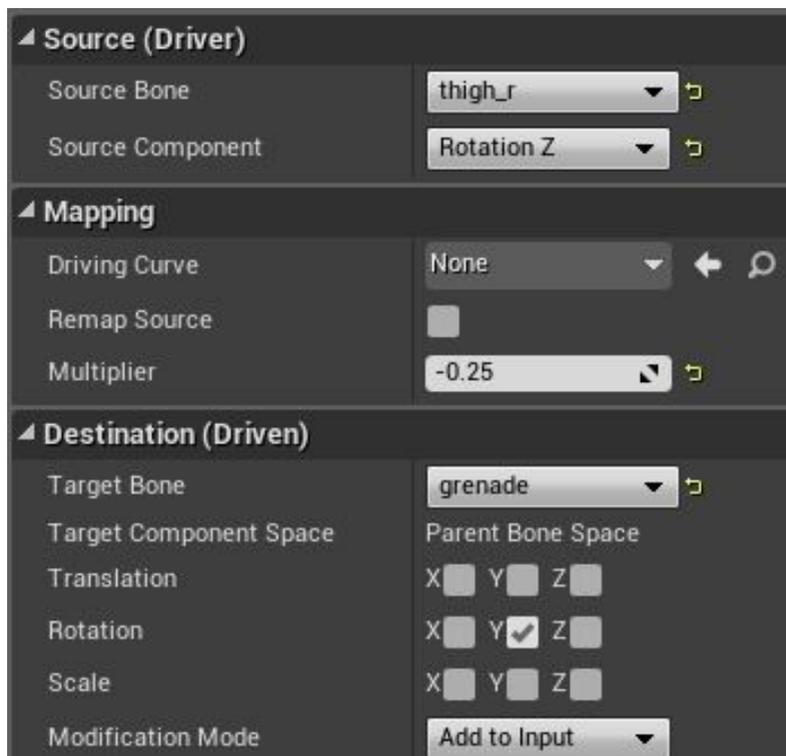
Dans l'exemple ci-dessus, l'accessoire fixé (vert) n'a aucune animation reliée, il est entraîné dans les deux axes en fonction du bone de la cuisse du personnage. Tout cela est calculé lors de l'exécution.

Vous pouvez soit définir la valeur "Driver" directement avec un multiplicateur, le redéfinir complètement, ou tout simplement utiliser un asset Curve pour conduire l'animation.

Utiliser une courbe est généralement la meilleure approche car elle permet de définir la réponse naturelle et de peaufiner interactivement les points / tangentes, et de voir les changements en temps réel.

Unreal Engine 4 – Guide des mises à jour (4.20)

Voici les nouveaux paramètres pour les contrôleurs d'animation axés sur le bone:



Améliorations de la fonctionnalité "Two bones IK"

4.17

Vous avez maintenant un meilleur contrôle sur les articulations contrôlées par deux os IK avec la nouvelle option « AllowTwist » qui vérifie si les joints au milieu peuvent se tordre. Cela peut être particulièrement utile pour les configurations de bras mécaniques.



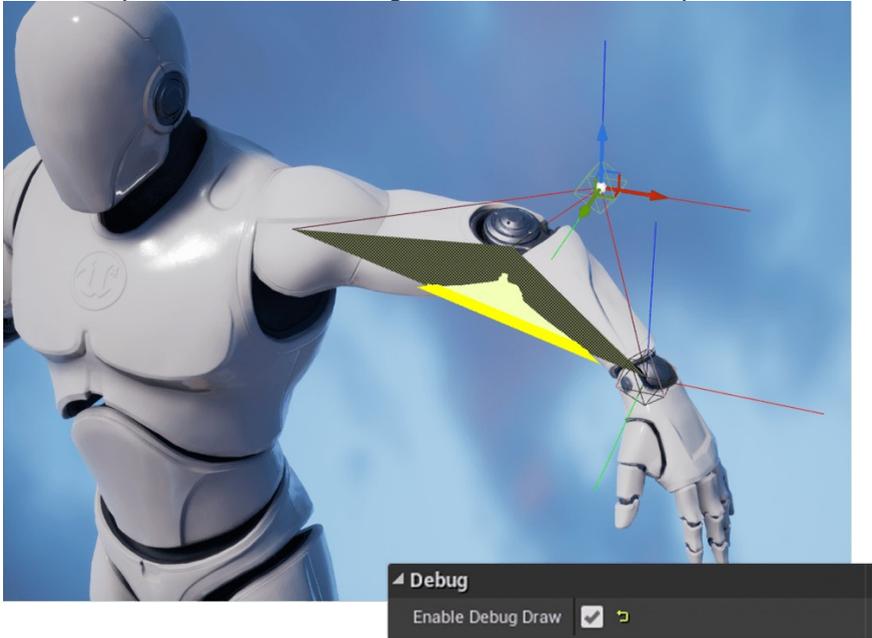
Allow Twist Desactivé



Allow Twist Activé

Unreal Engine 4 – Guide des mises à jour (4.20)

L'option « Maintain Effector Relative Rotation » est complétée par une nouvelle option « Enable Debug Draw », comme indiqué ci-dessous:



Animation Blueprint

Amélioration des blueprints d'animation

4.19

- Possibilité de contrôler si l'instance de post-traitement s'exécute ou non sur un composant de maillage squelettique à la fois dans Blueprints et dans la suite de l'éditeur d'animations
- Le nœud **copy pose** prend maintenant en charge les courbes de copie ainsi que la transformation de bone - Pour utiliser cette option, il faut activer `bCopyCurves`, il va copier les courbes existantes dans le squelette actuel. Si la liste de courbes a changé, le système d'animation doit être réinitialisé.

Unreal Engine 4 – Guide des mises à jour (4.20)

- Ajout d'un avertissement lors de la lecture d'une animation sur une pose additive: quand une animation non additive est connectée à la broche additive d'un noeud d'animation additif (comme Apply Additive), un avertissement s'affiche dans le journal des messages avec Animation et Nom du Blueprint.
- Sélection Native AnimInstance sur SkeletalMeshComponent - Autorise les instances d'animation natives à être définies sur les composants maillés squelettiques. Auparavant, seuls les plans d'animation pouvaient être définis sur les composants de maillage squelettiques.

Make Dynamic Additive Node pour Animation Blueprints

4.17

Un nouveau nœud "Make Dynamic Additive" dans l'animation Blueprint prend une pose de base et une pose cible comme entrées et crée la pose additive lors de l'exécution.

Cela peut être particulièrement utile lorsque vous travaillez avec des **performances live**: la capture de mouvement en live, ou d'autres situations où les données d'animation sont transmises au moteur Unreal lors de l'exécution.

Il peut également vous permettre d'utiliser un seul asset de séquence d'animation comme remplacement ou additif, économisant la mémoire disque pour un faible coût d'exécution.

Live editing d'animation Blueprints (expérimental)

4.17

Les Animation Blueprints peuvent maintenant être édités en direct et recompilés tout en jouant dans l'éditeur, ce qui permet des itérations

Unreal Engine 4 – Guide des mises à jour (4.20)

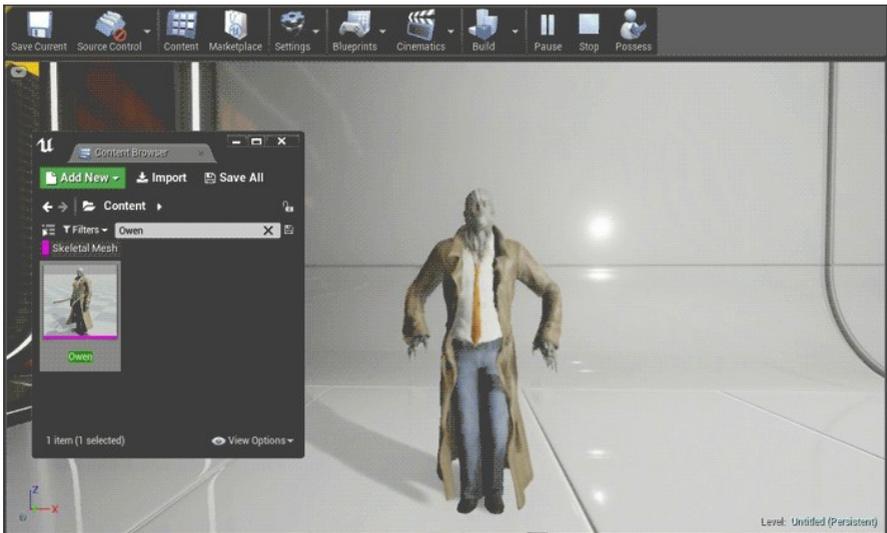
beaucoup plus rapides. C'est même transparent, on ne se rend pas compte qu'il y a [recompilation au final](#).

Lors du choix d'une instance Blueprint d'animation pour déboguer dans le niveau, la fenêtre de l'éditeur Blueprint d'animation est maintenant «connectée» à cette instance et affiche un miroir de l'état d'animation de l'instance.

Blueprint animation par défaut

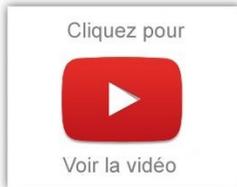
4.14

Permet d'attribuer un Animation Blueprint à un skeletal mesh qui sera toujours en exécution quelque soit l'outil utilisé (dans le niveau, dans le sequencer, etc.).



Hordes de personnages avec Blueprint d'animation

4.16



Lightweight RigidBody Simulation permet de créer des hordes de personnages avec la possibilité d'utiliser un Blueprint d'animation utilisant l'API PhysX. Les personnages réalisés par cette simulation peuvent générer une collision avec une géométrie statique du monde.

C'est probablement une des fonctions les plus impressionnantes. De quoi faire péter du zombi en veux-tu, en voilà ! (et je ne veux pas entendre de bombe claymore à la sortie d'un concert... :().

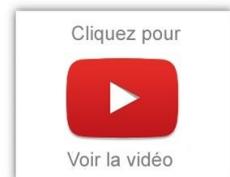
Sequencer animations en Blueprints (expérimental)

4.15



Le nouveau plugin **Actor Sequence** permet d'intégrer des séquences dans les Actor Blueprints. Cela permet la réutilisation des séquences en liant les animations à l'instance de Blueprint et de les déclencher à l'aide du graphique de l'événement.

Vous pouvez également ajouter un composant Actor Sequence à tout acteur pour ajouter des animations à une seule instance d'un acteur.

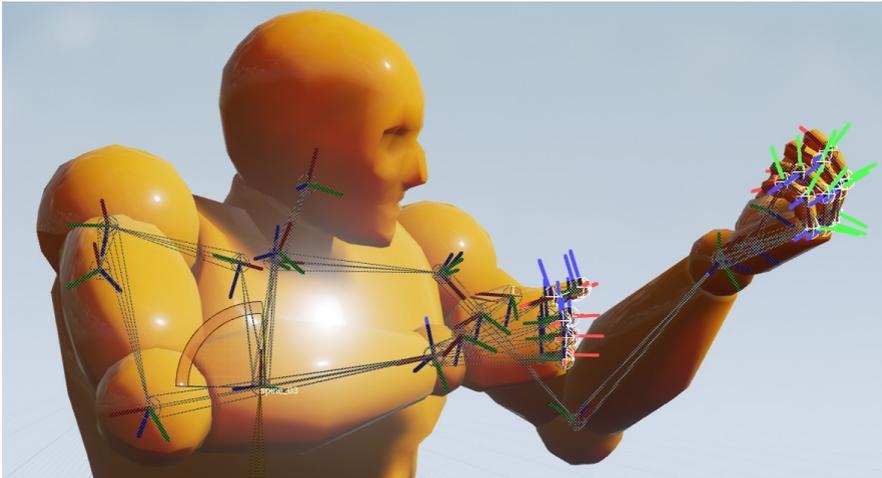


Animation Editing System

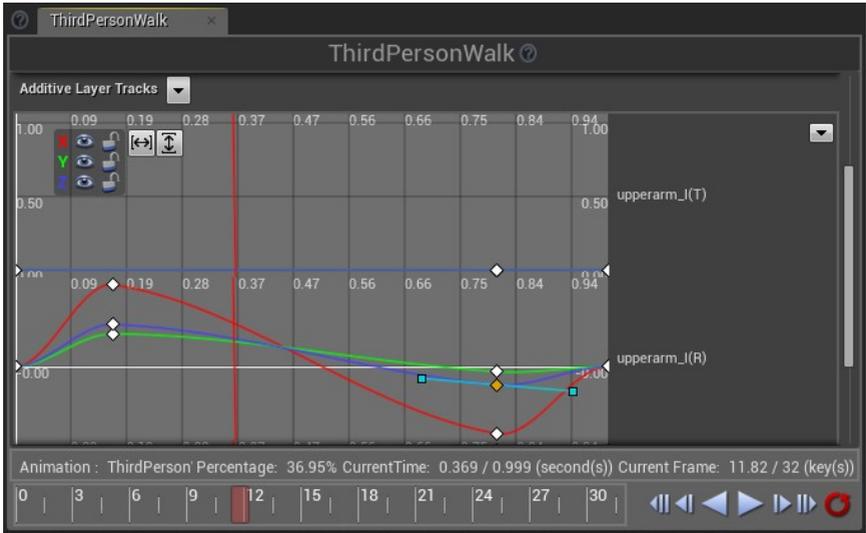
4.7

Plus besoin de passer par un logiciel de type modéleur/animateur pour créer vos animations.

Bougez le curseur de temps, modifiez la position des bones, leur rotation en utilisant le gizmo et créez une image clé (touche 'S'). Répétez l'opération avec une autre position...



Grâce à la courbe, vous pourrez paramétrer le mouvement de façon aisée. On peut aussi utiliser une “additive layer” qui permettra de conserver les anims, même si on réimporte le modèle et ses animations.



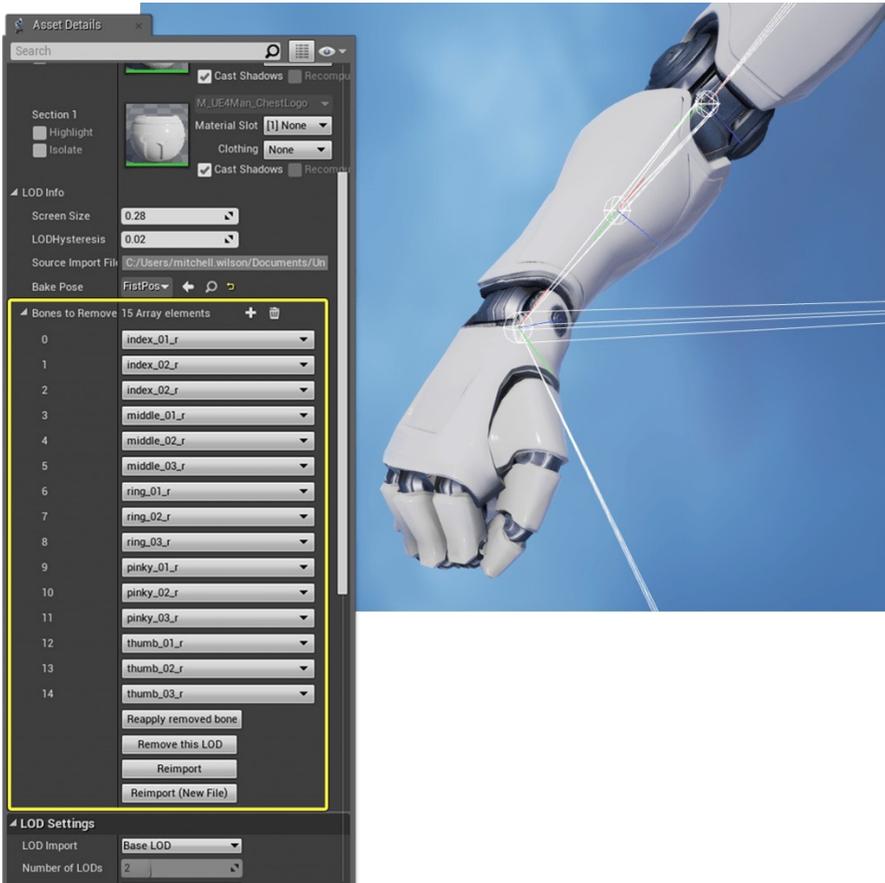
Il est même possible de sauver l'animation comme un Asset et ainsi, pouvoir la réutiliser dans de multiples projets.

Baking Pose lors de la suppression de Bones

4.17

Possibilité de spécifier une animation Pose à utiliser pour reskinning les vertices dans Skeletal Mesh LOD pour lesquels les os d'origine sont supprimés, au lieu de les pondérer le plus proche parent d'os.

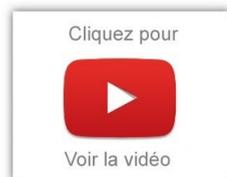
Unreal Engine 4 – Guide des mises à jour (4.20)



Partage du paramètre LOD des skeletal mesh

4.20

Définissez les paramètres LOD une seule fois pour les réutiliser sur plusieurs ressources Skeletal Mesh à l'aide du nouvel élément LOD Settings! Dans le panneau Détails de l'actif d'un maillage squelettique, sous Paramètres de LOD,



Unreal Engine 4 – Guide des mises à jour (4.20)

vous pouvez maintenant sélectionner un actif de paramètres LOD à utiliser ou générer un nouvel actif en fonction des paramètres actuels.

Une [page relative aux détails des assets](#) est à votre disposition. Vous pouvez également affecter le paramètre LOD et régénérer les LOD à partir de Blueprint à l'aide d'une Blutility.

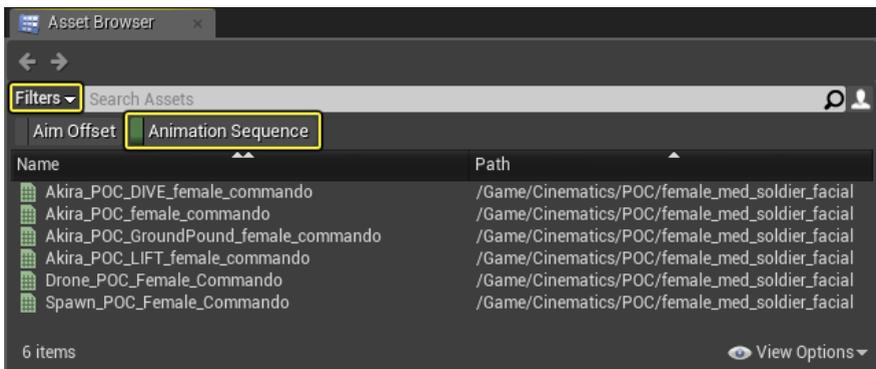
Aperçu Additional Meshes avec CopyPose

4.17

Dans les outils d'animation, "Additional Meshes" qui permet d'afficher un aperçu de meshes modulaire utilise désormais le nœud «Copy Pose» au lieu de «MasterPose». Cela signifie qu'il peut prendre en charge les meshes qui ne partagent pas un squelette et les os sont copiés par nom, ce qui rend plus souple.

Filtres des dossiers dans Anim Asset Browser

4.17



Unreal Engine 4 – Guide des mises à jour (4.20)

Le navigateur d'Animation peut désormais filtrer sur la base d'un certain nombre de dossiers sélectionnés. Le nombre de filtres disponibles est contrôlé par un paramètre dans la préférence de l'éditeur.

Règles de transition des animations

4.9

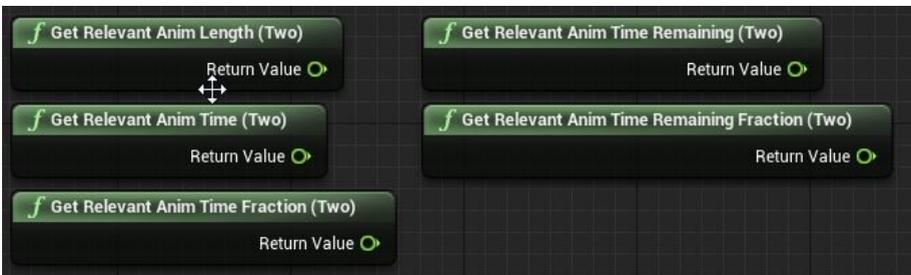
Amélioration de l'arbre de morphing d'animation (animation blending trees), particulièrement dans les transitions.

Meilleure gestion des assets de remplacement des animations

Les animations référencées par les règles de transition ne doivent plus être de la même taille dans chaque enfant "Animations Blueprint".

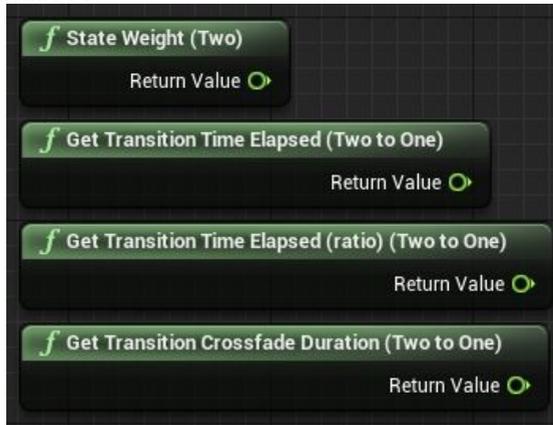
Jouer l'animation la plus pertinente

Afin de rendre les machines d'état plus utilisables, vous pouvez maintenant utiliser une nouvelle classe de "getter" qui capte toujours l'animation pondérée la plus élevée. Les nœuds suivants sont maintenant disponibles:



Mélange personnalisé des graphes d'animation

Des informations complémentaires sont disponibles dans les graphiques de mélanges personnalisés pour leur permettre d'être utilisés pour une animation de transition personnalisé one-shot. Les nœuds suivants sont maintenant disponibles dans les graphes de fusion personnalisés:

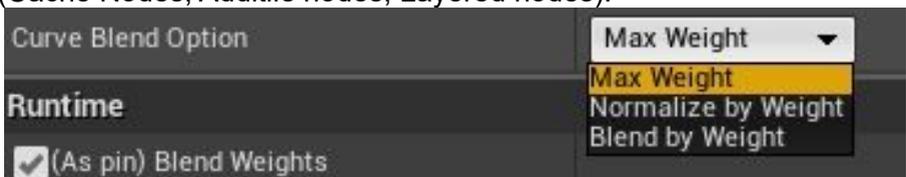


Animation Curve Evaluation Change

4.9

La phase d'évaluation propose les avantages suivants:

Le poids de la courbe est correctement évalué avec des mélanges. (Cache Nodes, Additifs nodes, Layered nodes).



Curve evaluation est placée en multi-threading si vous utilisez l'évaluation parallèle

Cela permet de créer un nœud qui peut driver la courbe de la transformation d'un bone et vice-versa

Si vous avez besoin d'une courbe sans évaluation (retard), vous ne

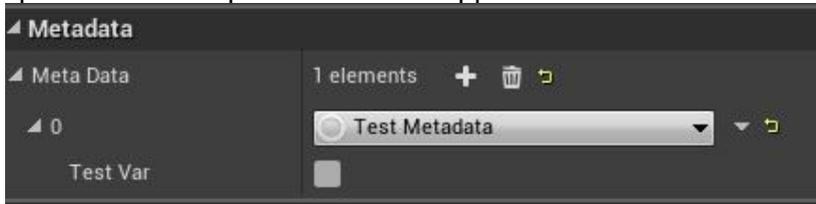
Unreal Engine 4 – Guide des mises à jour (4.20)

pourrez pas obtenir les données mises à jour. Les données de la courbe sont traitées comme des transformations de bones maintenant.

Animation Asset Metadata Support

4.9

Des métadonnées peuvent être ajoutées à l'asset d'animation. Les métadonnées sont une classe Blueprintable dérivée de la classe Anim Meta Data. Cela permet d'ajouter des métadonnées personnalisées à des actifs d'animation. Anim Séquence, Anim séquentiel, Anim Composite et Blendspace sont tous supportés.



Vous pouvez interroger ces données à partir d'un asset d'animation en C++ en appelant `GetMetaData()`, ou en utilisant `GetSectionMetaData()` Anim Montages.

Anim dynamics (simulation physique pour les personnages)

4.11

Anim Dynamics se présente comme un nouveau node du Blueprint d'animation, permettant d'ajouter des mouvements dynamiques aux skeletal meshes, sans avoir à passer par des solutions comme celles de Nvidia par exemple.

Voici ce qu'on peut faire et simplement:



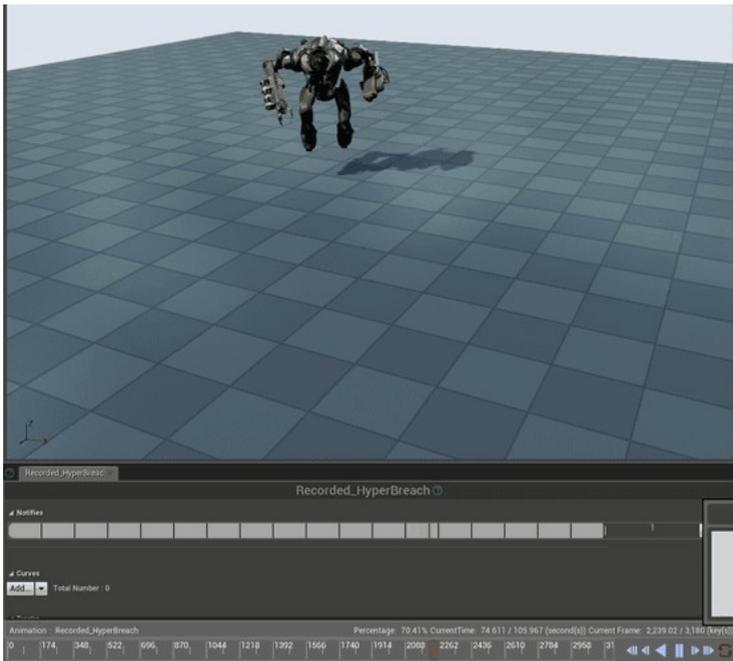
Principales caractéristiques:

- **Simulation de corps rigide simplifié**; pas de collision et utilisation de boxes pour les calculs d'inertie
- **Contraintes de corps rigides**: angular, cone, prismatic, planar (voir thread d'origine pour plus d'infos).
- **Chaînes**: modèle simplifié pour les cordes, chaînes et autres objets similaires
- **spring targets**: Utilisation de ressorts linéaires ou angulaires pour des effets de rebonds plus intenses
- **Vent**: permet de partager les mêmes données qu'APEX concernant la source du vent et ainsi obtenir un effet plus homogène, et donc, plus réaliste.
- **Adaptive sub-stepping**: l'animation peut utiliser le normal tick des physics settings du projet ou bien utiliser des sous-étapes pour une simulation plus réaliste. Le tout peut être paramétré dans un node.
- **Visualisation**: permet un debug plus facile (Angular, Prismatic et Planar limits, Planar exclusion methods, etc.)

Enregistrement d'animations depuis le GAMEPLAY

4.11

Nous pouvons maintenant enregistrer l'animation d'un skeletal mesh en live et le sauvegarder sous la forme d'une Anim Sequence!



Cet asset peut être utilisé dans le moteur ou exporté comme FBX.

Comment utiliser cette fonction:

- Pour enregistrer une animation, ouvrez la console et tapez: `RecordAnimation [Path for new asset]`
- Pour arrêter l'enregistrement, tapez: `StopRecordingAnimation`
- Donc, on peut le faire à n'importe quel moment dans le jeu.

Animation posing en copiant à partir de meshes

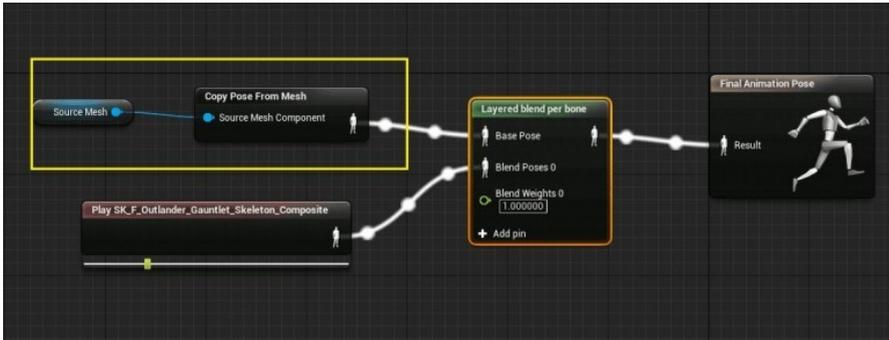
4.11

Nous disposons d'un nouveau nœud dans le Blueprint d'animation pour mélanger (blend) une animation avec un skeletal mesh externe dans une pose particulière.



Dans l'exemple ci-dessus, le Gant utilise le nœud "Copy Pose from Mesh" dans son propre Blueprint Anim pour copier les transformations de la main et du bras à partir d'un Source Mesh, le personnage féminin.

Unreal Engine 4 – Guide des mises à jour (4.20)



Pour cela, il faut que le nom des bones correspondent bien entendu, les autres bones étant ignorés.

Marked-based animation syncing

4.11

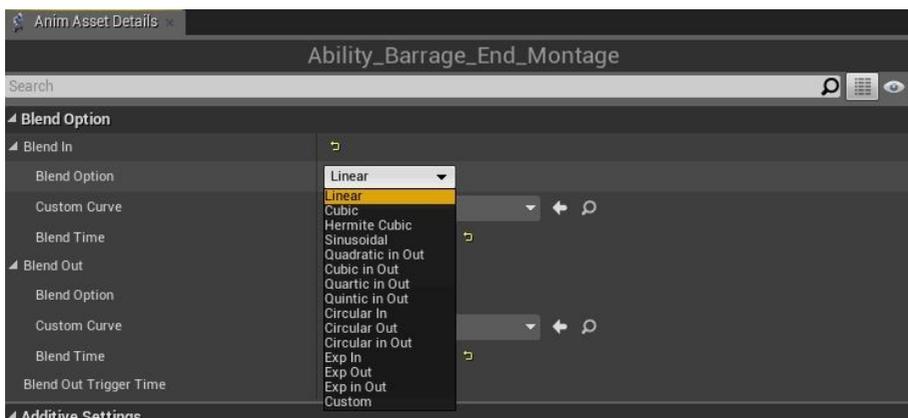


Les animations peuvent maintenant être synchronisés en utilisant des marqueurs (notifies) dans les données d'animation. Vous pouvez ajouter des données de marquage par un clic droit sur une piste et en sélectionnant Add Notify->New Sync Marker. Effectivement, c'est beaucoup plus simple quand vous devez synchroniser deux personnages par exemple: l'un qui saute dans vos bras, l'autre qui rattrape.

Curve Blending for Animation Montages

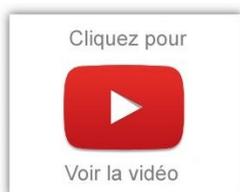
4.11

Les montages supportent maintenant la courbe de mélange. Les Blend In et Out contrôlent la façon dont le Montage doit être mélangé au moment de la lecture.



Animation Montage Time Stretch Curve

4.18



Il est possible d'ajouter une piste Time Stretch Curve à un montage d'animation pour contrôler la vitesse de lecture afin de maintenir la vitesse de lecture normale du Montage tout en ralentissant sélectivement d'autres parties du Montage.

Pour utiliser la courbe Time Stretch, créer une courbe flottante nommée MontageTimeStretchCurve dans le montage et ajouter des clés pour définir la courbe.

Sequencer

Apparition du Sequencer

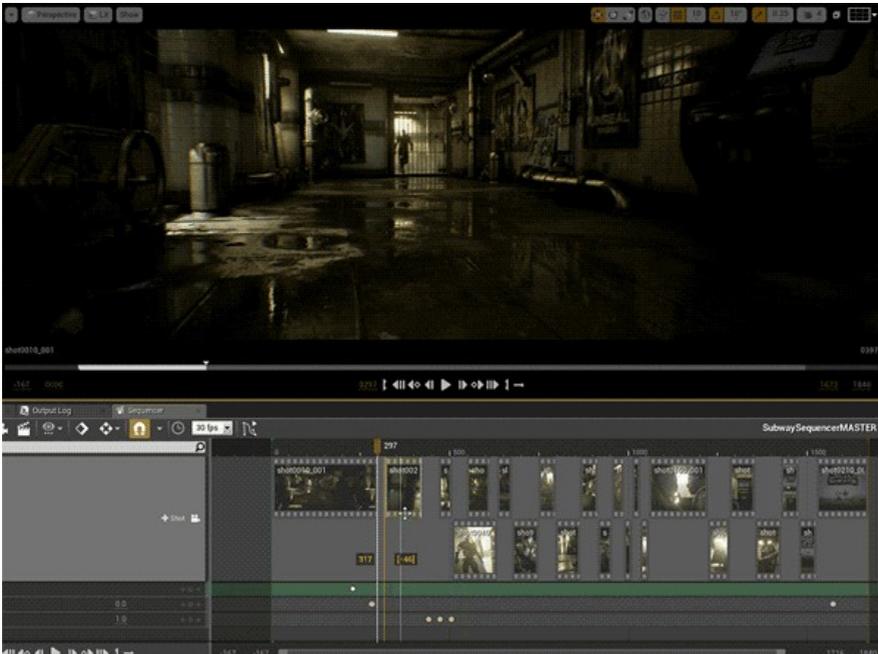
4.11

4.12

Sequencer est le nouveau système d'animation cinématographique non linéaire, encore en cours de développement. Je vous en avais déjà parlé précédemment dans les cahiers d'Unreal Engine.

Il est destiné à remplacer à terme Matinee qui souffre d'une vieille architecture héritée d'UE3 et qui provoque des lourdeurs d'utilisation dans UE4. Bon, il est toujours en phase expérimentale, mais dispose de plus de fonctions au fur et à mesure... Il a été d'autant plus introduit dans cette version.

Il possède de nombreuses fonctionnalités comme le keyframing automatique, les caméras cinématographiques, et l'enregistrement de gameplay en direct.



Unreal Engine 4 – Guide des mises à jour (4.20)

Il est prévu pour combiner la puissance d'éditeurs non linéaires avec des techniques d'animation 3D.

Il permet aussi de faciliter la collaboration sur les grands projets d'animation grâce à un flux de travail non-linéaire.

L'interface est très simple d'utilisation et sur les bases de la majorité des logiciels 3D et de Matinee. Pour ceux qui connaissent bien Matinee, ce dernier souffrait beaucoup de son intégration partielle au moteur de génération 4 – il était surtout un portage de la version 3.

Qui n'a pas déjà pesté en devant placer un actor Matinee sur la scène? Là, on hérite des propriétés habituelles en termes de Timeline et autres outils communs aux autres fonctions d'UE4. Pour un nouveau venu, c'est beaucoup plus logique !

De nombreux outils de caméras ont été améliorés: focales, rigcamera et viewport, ... Cela facilite la création de plans plus réalistes.

Sequence recording vous permet d'enregistrer une partie du jeu et d'y ajouter des éléments ou de les modifier.



[La documentation est en ligne ici.](#)

Sequencer : cinématographie haut de gamme

4.13

Le **Séquenceur**, l'éditeur de cinématographique non linéaire, a été mis à jour pour la cinématographie haut de gamme.

L'enregistrement d'une partie de jeu en direct a été améliorée, il ne s'agit plus d'une simple vidéo car sons et effets sont enregistrés de façon séparée et peuvent être modifiés. Pour en voir plus, [c'est ici](#).



Il est possible de créer des groupes de caméra, de suivre un personnage si la caméra est mise en enregistrement automatique.

Sequencer peut maintenant **importer et exporter des fichiers au format EDL** (CMX).

Unreal Engine 4 – Guide des mises à jour (4.20)



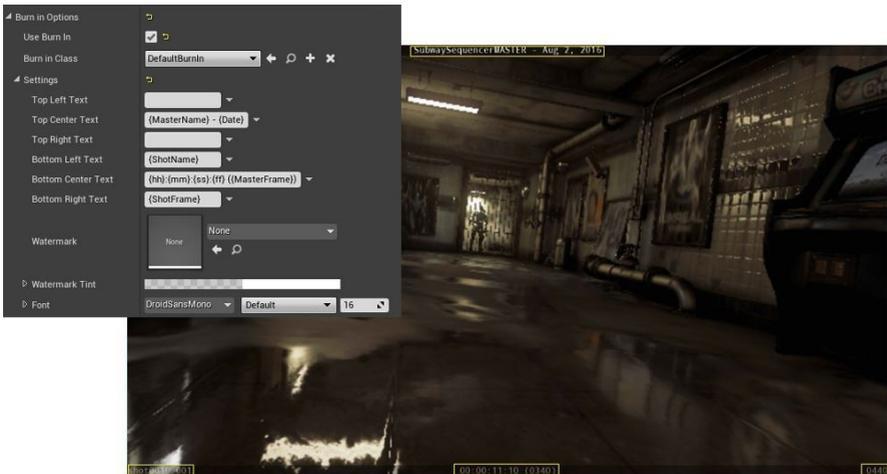
La photo montre une séquence qui a été exportée vers Adobe Premiere Pro. Toute modification de séquençage faite sous Premiere peut ensuite être réimportée dans le Séquenceur de UE4 ! Super non ?

Sequencer burn-ins

4.13

Lors du rendu, vous pouvez maintenant configurer un "burn-ins" pour les images exportées pour l'identification et le suivi des plans, c'est à dire la possibilité d'ajouter du texte dynamique (comme le n° de frame, la date, etc.):

Unreal Engine 4 – Guide des mises à jour (4.20)

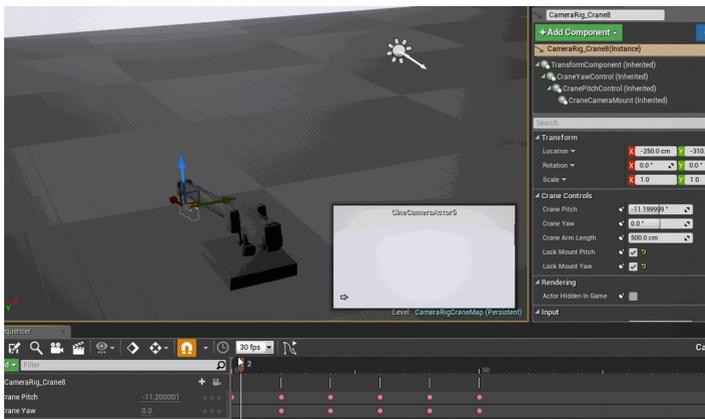


Amélioration de la Camera Rig Crane

4.14

Pour le tournage de films, on utilise des sortes de grues sur lesquelles on fixe une caméra. Pour les besoins du sequencer, [un même dispositif](#) a été développé il y a quelques temps. Existe également le [système de rail](#) sur lequel se déplace la caméra.

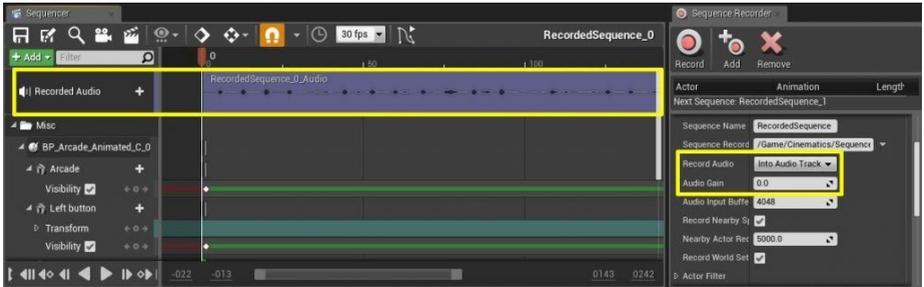
Le comportement de la grue imite maintenant parfaitement celui d'une véritable grue :



Enregistrer de l'audio dans une séquence

4.14

Vous pouvez maintenant enregistrer de l'audio à partir d'un microphone pendant l'enregistrement dans une séquence.



Sequencer animations blending

4.15

Sequencer prend désormais en charge le mélange pondéré entre plusieurs animations permettant des séquences d'animation complexes construites à partir d'animations existantes. Toute animation peut être ajoutée à un moment donné.

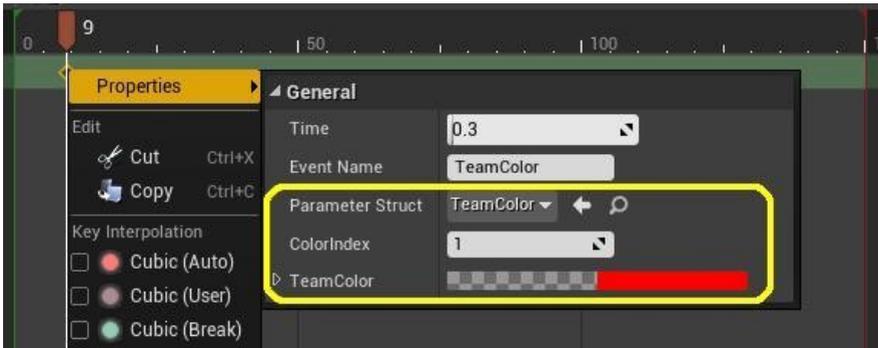
[Bon la vidéo](#) parle d'elle-même (c'est chiant quelques fois de mettre des mots sur des concepts).

Vous pouvez contrôler le poids d'une animation en élargissant la piste et en saisissant la valeur dans le montage. Pour permettre à votre Skeletal Mesh d'utiliser plusieurs poses, un nouveau mélange multi par le nœud Alpha est disponible. Vous pouvez ajouter de nouvelles poses dans le menu contextuel.

Sequencer events

4.15

peut désormais contenir des payloads de données (traduire par lot de données supplémentaires - ou surcharge) pour déclencher des événements suivant le contexte. Pour déclencher un événement avec un payload, créer une structure qui définit les paramètres pour l'événement et l'appliquer à la propriété parameter struct:



Animer des objets dynamiques avec Sequencer

4.15

en utilisant des Blueprints pour appliquer des Sequences aux objets lors de l'exécution.

Cela signifie qu'on **peut animer un autre objet que celui pour lequel la séquence a été écrite**. Pour extraire les identificateurs d'une séquence, utiliser le nouveau nœud `GetSequenceBindings`.

Amélioration de l'interface graphique de Sequencer

4.15

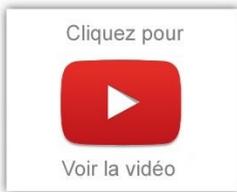
un nouvel [outil de transformation](#) Touches / sections permet de repositionner et redimensionner la sélection actuelle. Il est possible de copier/coller des pistes sur des objets compatibles.

La totalité de la séquence est à l'heure relative, un plan peut être isolé en réglant "Evaluate Sub Sequences in Isolation".

Les composants clés peuvent être sélectionnés individuellement.

Shot Tracks

4.16



le Sequencer connaît des améliorations pour la **cinématique et la création de films**.

- Hierarchical bias par tir**: pour que les cinéastes construisent un pipeline proche de ce qu'ils utilisent habituellement (voir vidéo)

- Propriété «When Finished» pour toutes les pistes**

- Pre/post roll**: concept général pour toutes les pistes.

Les Animate Material Parameter Collections

4.16

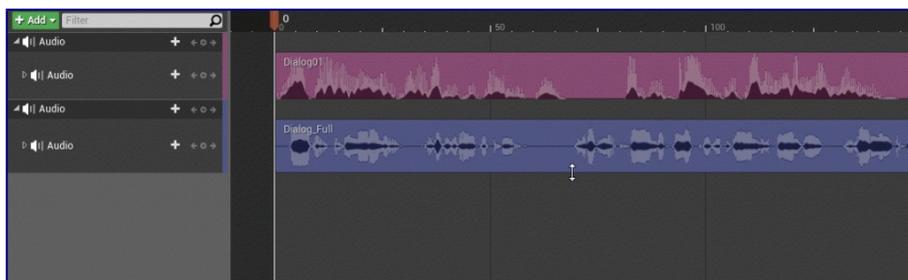


peuvent être animées avec un contrôle total sur les paramètres d'animation - scalaire et vectoriel. Plus besoin d'animer les valeurs des paramètres individuels sur chaque instance matérielle pour partager l'animation.

Amélioration de l'Interface

4.16

Les vignettes pour des pistes audio rendent maintenant les échantillons de crête avec une courbe de RMS intérieure (lissée). Les pistes audio peuvent également être redimensionnées verticalement!



Auto-key simplifié

4.17

Auto-key a été simplifié pour devenir un seul état basculable afin d'améliorer la convivialité et la clarté.

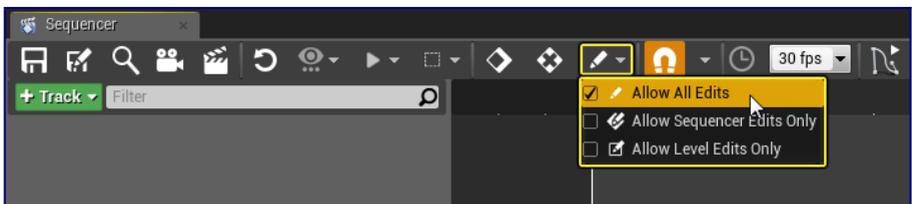


Sequencer edits et Level Edits

4.17

"*Allow Sequencer Edits Only*" permet d'isoler les changements réalisés. Cela ne va créer de nouvelles tracks que s'il n'existe pas déjà de keyframe - et cela n'entrera pas en conflit avec d'autres keyframes existants. Pas de propagation par exemple.

"*Allow Level Edits Only*" permet de travailler sur les propriétés par défaut des objets.



Divers

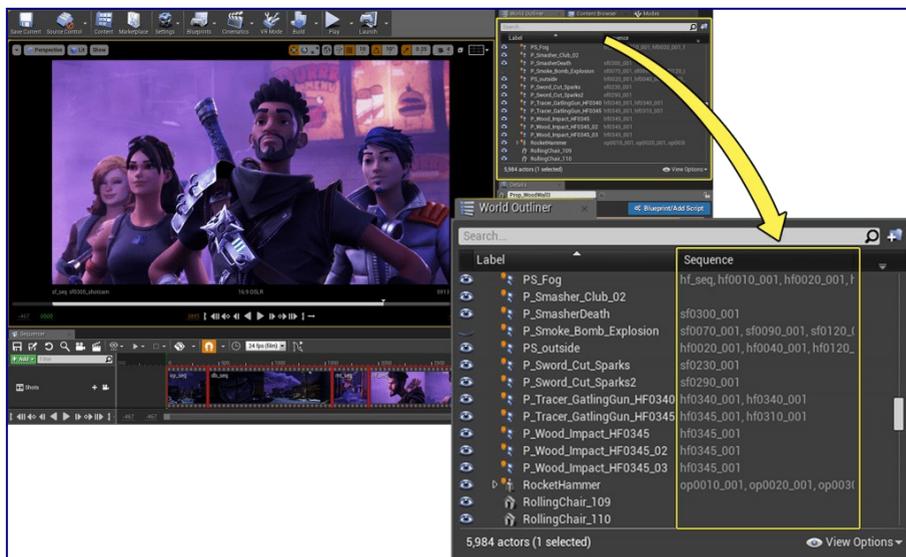
4.17

- Implémentation des "Attenuation settings" pour les audio attachés à des actors,
- possibilité de spécifier des règles pour attacher ou détacher des sections,
- plus d'options de cinématique dans le Level Sequence Player,
- Key Time ajoutés aux propriétés Vector, Transform, Color key,
- raccourcis clavier "étape suivante/ précédente" tir (Shift + ",", and Shift + "."), etc.

Intégration dans le World Outliner

4.17

World Outliner montre l'ensemble des séquences auxquelles un acteur est lié. Et ça c'est top pour avoir une vue d'ensemble.



Mélanger les sections du sequencer

4.17

Pour créer des transitions en douceur avec facilité (2D / 3D transforms, vecteurs, couleurs, floats, etc.):

- Toutes les sections supportées se mélangent "gracieusement" (j'ai voulu harder la traduction stricte) quand il y a un chevauchement pour produire une valeur unique appliquée à l'objet.
- Les sections qui se chevauchent sur la même la ligne sont interpolées automatiquement lorsque c'est possible.
- Les pistes d'animation sont interpolées automatiquement

Unreal Engine 4 – Guide des mises à jour (4.20)

- De nombreuses fonctions sont disponibles en standard (linéaire, polynomiale, facilité exp in / out, etc.). Vous pouvez également spécifier un asset courbe ou mettre en œuvre vos propres fonctions de mélange personnalisé à l'aide de l'interface IMovieSceneEasingFunction (disponible dans BP).
- Les "Transform sections" peuvent désormais fonctionner sur une combinaison de canaux sans affecter le reste de la transformation (soit seulement Localisation et rotation).

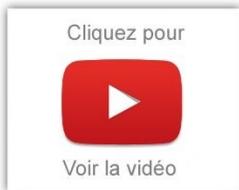


Si l'animation des personnages, des animaux et des objets du monde est déjà considéré comme un métier à part entière, avec des spécialisations, c'était souvent réalisé directement dans le modeleur, sous 3DSMax, Maya, Blender, etc.

Mais quand il s'agit de travailler une cinématique complète en 3D Temps-réel, les animateurs étaient souvent frustrés par les outils proposés. Avec le Sequencer, il devient possible de réaliser des cinématiques époustouflantes grâce à des outils qu'on ne trouvait jusqu'à présent qu'en précalcul. De fait, on peut vraiment dire qu'il **y a une spécialisation à prendre**, dès aujourd'hui, pour les animateurs qui veulent travailler dans le monde du jeu vidéo ou dans tous les domaines requérant le temps-réel. C'est même un métier à part entière car la cinématique n'est plus figée puisqu'elle peut prendre en compte de nombreuses interactions.

Image Plate Actor and Sequencer Track (Experimental)

4.17



Cela permet l'affichage de séquences précises d'images en plein écran lorsque l'actor est relié à une caméra. Bon, sur le coup, je n'ai pas tout compris.

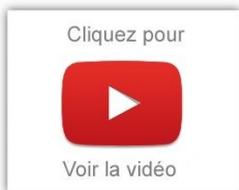
Heureusement, [un peu de documentation](#) est venu compléter tout ça.

En gros, **on peut utiliser des images (issues de vidéos ou non) pour constituer le background de l'actor**. En bougeant la caméra, cela bouge automatiquement l'image qui est utilisée comme background, donnant ainsi une impression d'immersion dans une scène alors qu'il ne s'agit que d'une image.

Voir [cette vidéo](#). Cette forme ne permet pas le temps-réel, mais pourquoi ne pas imaginer à l'avenir que la séquence d'images proviennent d'une webcam? Bref, encore un effort vers l'utilisation de la **réalité Augmentée** par Epic Games.

Améliorations du sequencer: imports FBX

4.18



L'importation des FBX a été améliorée en ajoutant la possibilité d'importer des transformations statiques, d'ajouter des caméras cinématiques et l'option de forcer l'axe x avant, les pistes sont mieux dessinées et les touches ayant une double fonction apparaissent en rouge, le support du glisser/déposer a été ajouté et la possibilité de rejouer les scripts de construction lors du nettoyage.

4.20

En important un fichier FBX dans Unreal, toutes les données FbxProperty enregistrées du fichier sont importées. Ces métadonnées sont accessibles dans les scripts Blueprint ou Python que vous exécutez dans l'éditeur Unreal. Cela peut vous aider à personnaliser vos propres pipelines de gestion des assets pour Unreal.

Pour plus de détails, voir [Pipeline de métadonnées d'actifs FBX](#).

Compiler à la volée

4.19

Sequencer est maintenant capable de compiler des modèles d'évaluation partiellement ou complètement périmés à partir des données sources, permettant une compilation beaucoup plus efficace.

Dupliquer des pistes

4.19



Il est désormais possible de copier / coller / dupliquer des pistes d'objet et leurs pistes enfants à partir du menu contextuel. Il est maintenant possible de copier d'une animation UMG à une autre.

Améliorations de Frame Accuracy pour Sequencer

4.20

Nous avons déjà traité de ces améliorations dans [cet article](#). Pour résumer, un meilleur contrôle du frame-accuracy, l'apparition de Media Tracks, des courbes encore plus paramétrables, la possibilité d'animer des variables sur des Instances Anim via des possessables et le support du format XML de Final Cut Pro 7 en import et en export (fonctionnant avec Adobe Premiere Pro également, et d'autres).

Sequencer stocke désormais toutes les données internes de temps sous forme d'entiers permettant une solide prise en charge de la frame-accuracy dans les situations où cela est nécessaire. Les Keys, section bounds et d'autres données sont maintenant toujours verrouillées à la résolution de séquence sous-jacente contrôlable par l'utilisateur; ceci peut être aussi fin ou aussi grossier que voulu. Des résolutions très élevées favoriseront une plus grande fidélité du placement des clés et des sous-frames, tout en réduisant la plage de séquence globale.



Mises à jour principales:

- Le curseur de temps dans Sequencer est maintenant représenté comme un bloc qui couvre toute la plage de la Frame actuellement évaluée, montrant très clairement quelles touches sont évaluées et lesquelles ne sont pas pour une trame donnée.

Unreal Engine 4 – Guide des mises à jour (4.20)

- La lecture “Force Fixed Frame Interval” a été rebaptisée “Frame Locked”, en réglant le FPS du moteur sur le taux d’affichage de la séquence et en verrouillant le temps sur des numéros entiers (pas d’interpolation de sous-trame)
- L’évaluation des sous-frames reste totalement prise en charge pour les situations dans lesquelles la précision de trame n’est pas prise en compte (comme l’animation UMG).
- Différentes sources de temps sont désormais prises en charge pour l’évaluation de l’exécution, telles que l’horloge du moteur (world-pause), l’horloge audio et l’horloge de la plate-forme.
- L’interface utilisateur peut maintenant être visualisée en code temporel Non Drop Frame (NDF) et en Drop Frame (DF). NDF est disponible pour toutes les fréquences d’images et convertit directement le nombre d’images en heures, minutes, secondes et images restantes. DF Timecode est uniquement pris en charge sur les NTSC Rates (23.976, 29.97, 59.94). Le format d’affichage peut être modifié avec la combinaison de touches Ctrl + T ou avec le menu de l’interface utilisateur.

Reportez-vous à la nouvelle page [Sequencer Time Refactor notes](#) pour plus d’informations.

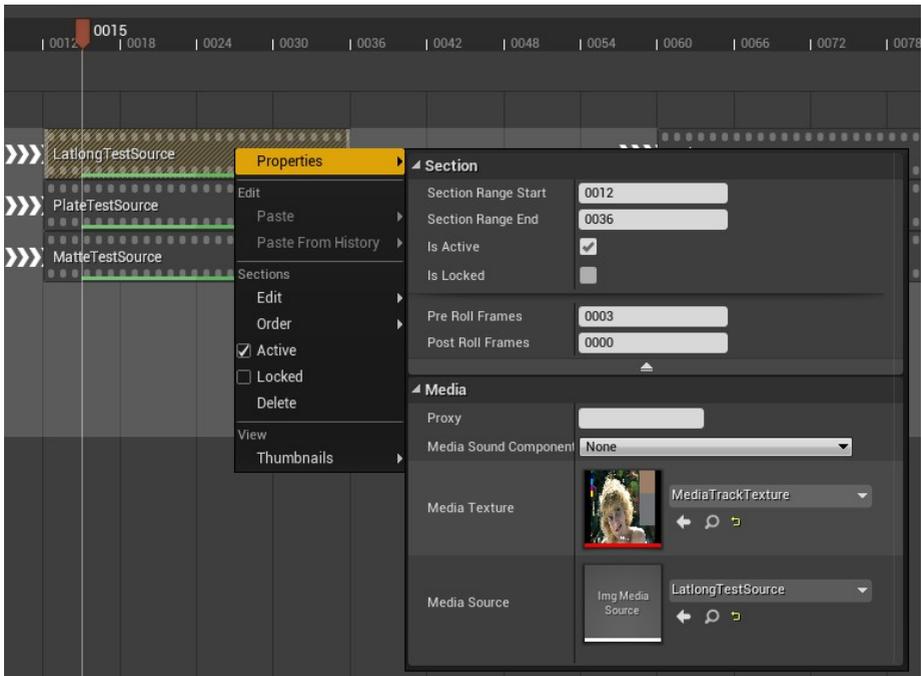
Media Track for Sequencer

4.20

Sequencer a une nouvelle piste pour lire les sources multimédia. **C’est comme la piste audio, mais pour les films.**

Il suffit de glisser-déposer un élément Media Source dans l’affichage des pistes ou de créer une piste multimédia dans le menu Ajouter une piste. Cette fonctionnalité fonctionne actuellement mieux avec les séquences d’images, en particulier **EXR**. Les séquences d’images dans la piste média synchroniseront précisément les images avec une sortie rendue.

Unreal Engine 4 – Guide des mises à jour (4.20)



Consulter la page [Using Media Tracks](#) pour plus d'informations.

Sequencer Curve Editor et améliorations

4.20

Plusieurs améliorations ont été apportées à l'éditeur de courbes et à l'évaluation dans Sequencer, notamment:

Les tangentes pondérées sont maintenant supportées sur les courbes flottantes.

Ajout du support pour les changements continus d'angle d'Euler lors de la modification des rotations. Les angles d'Euler ne sont plus limités à -180,180, ce qui est nécessaire pour éviter les retournements dans l'animation.

Unreal Engine 4 – Guide des mises à jour (4.20)



Utilisation de courbes pondérées dans l'éditeur de courbe du séquenceur

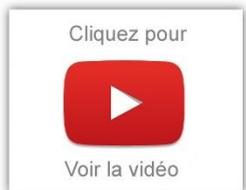
Vous pouvez maintenant activer la rotation de quaternions dans une section de transformation 3D via le menu Propriétés de la piste pour utiliser l'interpolation de quaternion pour interpoler en douceur entre deux rotations. Ceci est similaire à la fonctionnalité précédemment disponible en Matinee. Le grand remplacement n'est pas terminé!

Variables d'animation sur des instances anim dans Sequencer

4.20

Il est maintenant possible d'animer des variables sur des **Anim Instances** via des possessibles, permettant un contrôle direct des variables, des fonctions et d'autres contenus d'Anim Blueprint. Pour ajouter une liaison Inst Instance à Sequencer, recherchez son nom dans le bouton [+ Track] des composants de skeletal animation.

Toutes les variables exposées aux cinématiques seront affichées sur le sélecteur de piste.



Reportez-vous à la page [Contrôler les instances animées](#) avec le séquenceur pour plus d'informations.

Import / Export XML Final Cut Pro 7 dans Sequencer

4.20

Les données de scène du Sequencer peuvent maintenant être exportées et importées depuis le format XML de Final Cut Pro 7. Cela peut être utilisé pour renvoyer des données à Adobe Premiere Pro et à d'autres logiciels d'édition prenant en charge FCP 7 XML. Vous pouvez découper et décaler les photos dans le logiciel d'édition et les recadrer automatiquement dans le séquenceur lors de l'importation.

Remarque: L'audio n'est pas supporté pour le moment.

<https://youtu.be/lq5uhsZXidc>

Améliorations des enregistreurs de séquence

4.20

Sequence Recorder prend désormais en charge un système de profil stocké dans le niveau Persistant. Les profils vous permettent de stocker les acteurs que vous souhaitez enregistrer et leurs paramètres, ainsi que le chemin de sortie pour stocker les données enregistrées. Sequence Recorder prend également en charge l'enregistrement de plusieurs prises pour chacun des acteurs sélectionnés.

Veuillez consulter la page [Enregistreur de séquence](#) pour plus d'informations.

Level Sequence Dynamic Transform Origin

4.19



Ceci permet de changer dynamiquement le World Offset, l'origine d'une animation dans le séquenceur et de pouvoir réutiliser des séquences dans d'autres espaces de coordonnées, dans une autre scène par exemple. Pour activer cette possibilité, il faut activer "Override Instance Data then assign a Transform

Origin Actor". Le fonctionnement est illustré dans cette courte vidéo;

Chaque fois que nous appuyons sur une touche (voir ci-dessus), le cube se déplace et notre actor continue de marcher le long d'un chemin, à partir de l'emplacement de l'actor d'origine de transformation.

Sequencer Anim BP Weight Control

4.19



le Sequencer weight blending fonctionne maintenant avec les Animations Blueprints. Vous pouvez utiliser le même emplacement pour l'animation et contrôler la taille par une courbe. Affectez le même Slot Name dans "Properties" pour chaque animation à fusionner. Puis effectuer un keyframe sur les valeurs de poids. Dans

l'anim BP, utiliser le Slot Node. L'AnimBP utilisera toutes les valeurs de poids fournies lors de la fusion.

Twist Corrective node

4.12

Le nouveau nœud Twist permet d'**utiliser les valeurs de la courbe comme cibles de morphing**.

C'est basé sur la torsion d'un os par rapport à l'autre. Par exemple, quand le cou est trop tourné dans un sens, vous pouvez appliquer une cible Morph au cou afin de créer un look plus naturel.



Cette fonction peut s'avérer fort utile également en réalité virtuelle si vous asservisiez les avant-bras d'un personnage aux motions controllers.

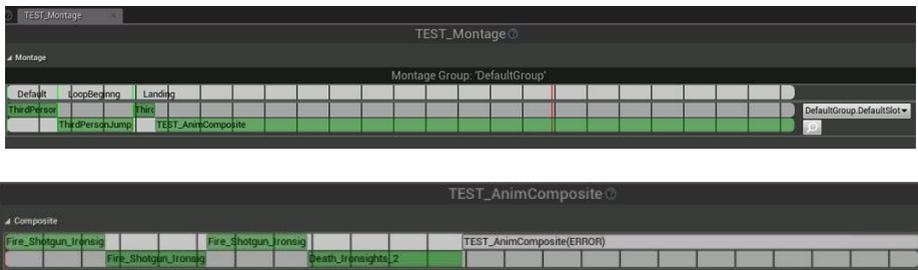
Des torsions peuvent apparaître et elles peuvent être visuellement compensées par ce système.

Embeded composite animations

4.12

Vous pouvez maintenant intégrer un AnimComposite dans une autre AnimMontage ou AnimComposite, vous permettant de créer votre propre asset d'animation de façon plus flexible.

Dans l'image ci-dessous, TEST_Montage contient TEST_AnimComposite, qui contient lui-même un certain nombre d'autres séquences d'animation.



Procedural Mesh slicing (TOP)

4.13

Nouvel utilitaire dans le Procedural Mesh Component pour **"trancher"**. Qui supporte les simples collisions, et la simulation physique. Un utilitaire permet de copier des données à partir d'un maillage statique à un maillage procédural.

Alors là, je trouve cela vraiment super: non seulement, on va pouvoir très simplement apporter plus de réalisme aux scènes en utilisant le moteur de physique, mais en plus, cela laisse envisager une utilisation pour creuser des terrains par exemple, sans passer par les Voxels.



Low level cloth simulation

4.16



Low level cloth simulation permet de donner plus de contrôle sur la simulation de tissus avec le **NvCloth** de **NVIDIA** (remplaçant **APEX**).

Animations Vertex : l'importateur Alembic (expérimental)

4.13

Alembic permet d'importer des animations vertex complexes. La composante d'animation physique permet aux personnages de réagir de manière réaliste aux forces physiques en réalisant l'animation squelettique par le biais des moteurs.



L'importation peut être faite de différentes façons:

- **Mesh statique:** Une seule image de l'animation Alembic est importée
- **Geometry Cache:** L'animation Alembic sera lue comme un flipbook.
- **Mesh Skeletal** Lors de l'importation, la séquence d'animation est compressée en utilisant un schéma PCA

Jean Meli nous a réalisé un [petit tuto en français](#) pour exporter une animation Physique de Blender vers UE4 en utilisant Alembic

Streaming GeomCache et importateur Alembic amélioré (Expérimental)

4.20

Epic continue à améliorer la stabilité et la performance du système de cache géométrique.

Les vertex animation frames individuelles sont maintenant compressées en utilisant un codec intra-frame basé sur le codage Huffman. Les données compressées sont streamées à partir du disque, ce qui permet la lecture d'une séquence plus longue avec un plus faible volume de mémoire. La nouvelle implémentation est encore très expérimentale et n'est pas prête à être utilisée en production.

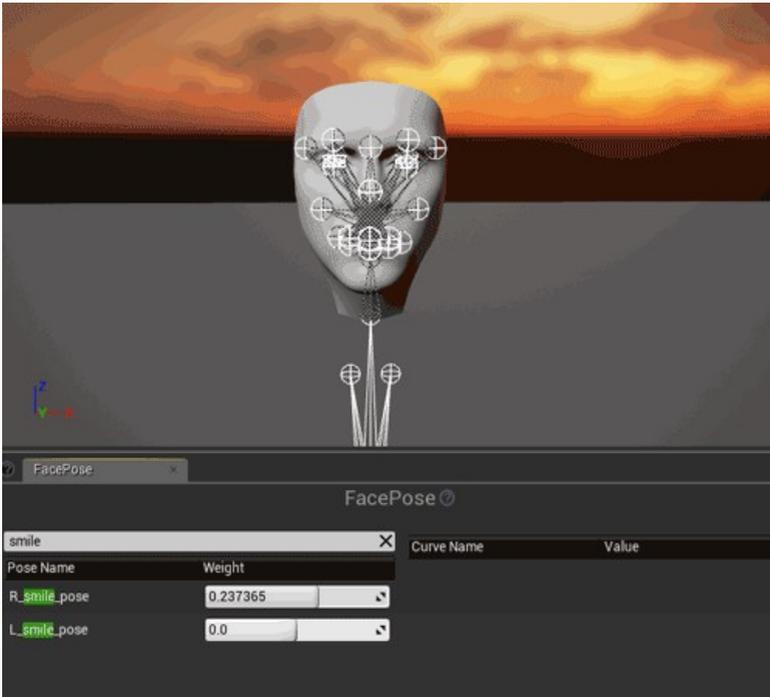
L'importateur Alembic a été modifié pour importer de manière itérative des images plutôt que d'importer toutes les images en masse. Cela devrait améliorer le pipeline PCA et la stabilité et la vitesse globales.

Animations des Visages

Animation pose assets

4.13

Un nouveau type d'animation d'assets appelé Pose Asset contient un ensemble de poses d'os nommés, que vous pouvez mélanger.



Permet l'**animation faciale FACS** (Facial Action Coding System) ou l'utilisation des [courbes viseme](#) pour l'adaptation du visage au dialogue. Bref, c'est vraiment une fonctionnalité très intéressante à plus d'un titre !

Animations faciales

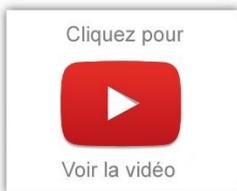
4.15

Un nouveau système pour les **animations faciales** a été ajouté, mais reste au stade expérimental. Nous n'avons que peu d'infos pour le moment, mais cela suggère l'arrivée d'un outil pour cela dans les versions ultérieures !

Amélioration du partage d'animations faciales en utilisant des Curve only animations

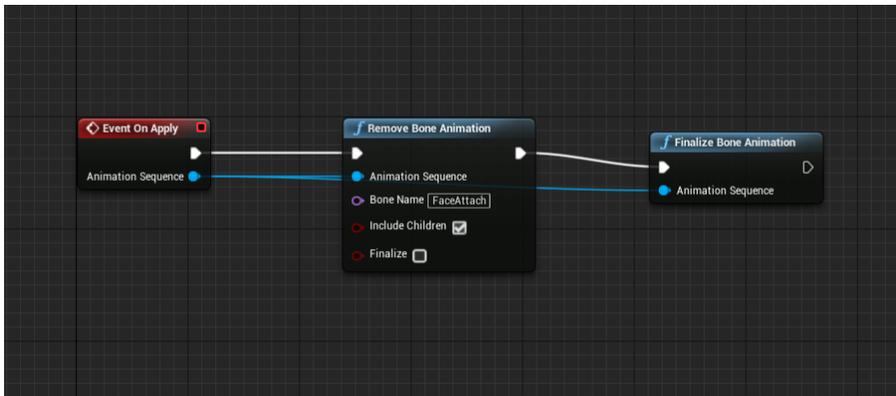
4.19

Le partage d'animations faciales a été simplifié. Il est possible d'animer les visages en utilisant une courbe et cela, sans avoir à faire une transformation du squelette. Il est donc possible d'utiliser une même animation faciale pour plusieurs personnages :



Pour y accéder, option "Remove All Bone Tracks" du menu Asset sous Animation.

Le modificateur d'animations peut être utile pour supprimer certaines articulations afin de partager des animations de corps. L'ajout des fonctions Remove Bone Animation supprime une courbe d'animation par nom de la séquence d'animation donnée.



Digital Human Improvements

4.20

De nombreuses améliorations de rendu ont été faites pour développer des personnages réalistes qui prennent vie.

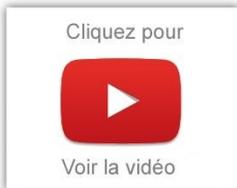


Les améliorations de rendu suivantes ont été apportées pour la peau, les yeux, l'éclairage et le Subsurface Scattering:

- Ajout d'un nouveau modèle spéculaire avec la méthode **Double Beckman Dual Lobe**. Je n'ai pas trouvé de papier à ce sujet, si jamais vous avez un lien...
- Transmission de la lumière en utilisant la rétrodiffusion pour les profils de subsurface.
- Meilleure ombrage de contact pour la diffusion subsurface avec la couleur de fond perdu.
- Illumination globale dynamique à courte distance à travers les matériaux post-process.
- Ajout de détails pour les yeux en utilisant une normal map distincte pour l'iris.

Pose driver animation node

4.16



Améliorations concernant les Pose Driver: permet de faire des drive blend shapes /bones de façon procédurale, en comparant la pose actuelle à un set de références cibles. C'est particulièrement utile pour les épaules et les bras.

Vous pouvez maintenant sélectionner plusieurs bones comme «entrées» pour lire une pose, voir quels bones serait modifiés par le noeud, spécifier une «courbe personnalisée» pour la façon dont chaque cible doit être activée, choisir de driver les courbes (morphes, matériel) directement, au lieu d'avoir un Pose Asset.

L'interface utilisateur a été améliorée pour permettre la création / édition de poses cibles, des barres pour montrer l'activation cible, etc.

Les Target Location peuvent être sélectionnés directement par simple click dans le viewport.



4.13

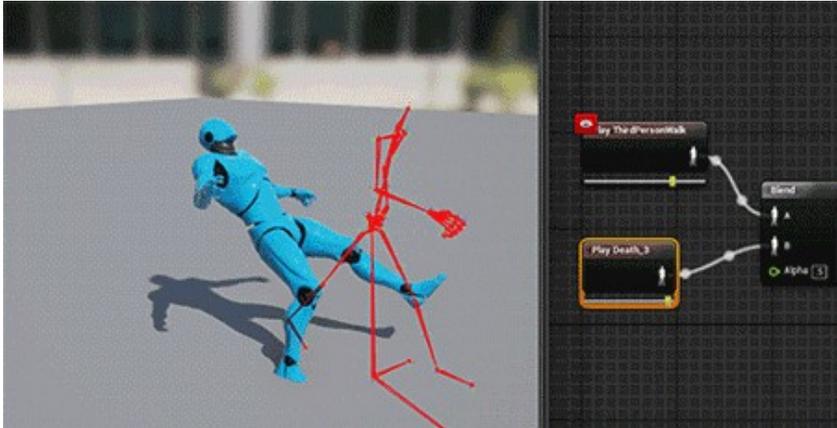
Un nouveau node permet d'utiliser une courbe sur le mouvement d'un os.

Celui-ci utilise un RBF (Radial Basis Function) pour interpoler les valeurs en fonction de l'orientation d'un os cible.

Animation node pose watching

4.13

Le graphe des animations peut être suivi dans Persona.



Cela vous permet de voir une représentation de la pose générée à un point quelconque dans le graphique anim dynamiquement.

Des moments multiples peuvent être actifs à la fois vous permettant de comparer des poses à différents points et de trouver le point exact où les erreurs dans votre pose actuelle sont introduites.

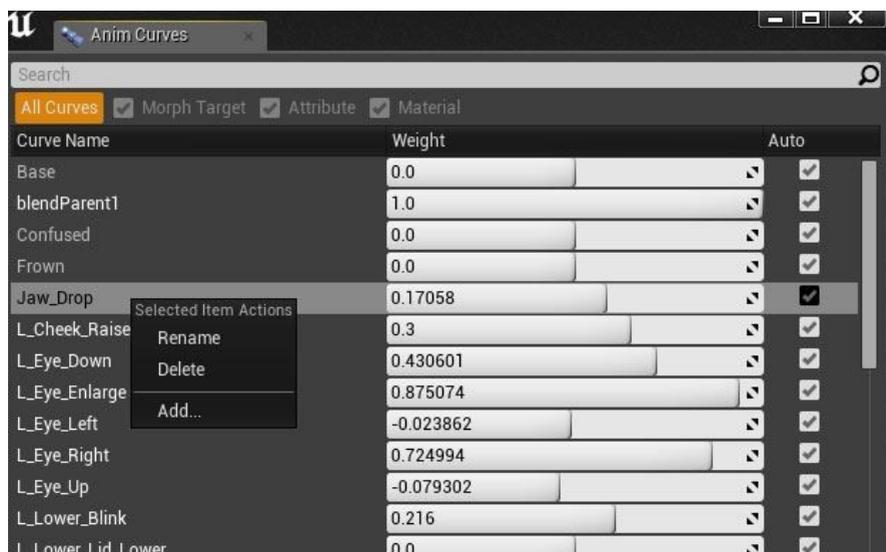
Cela peut être très utile pour le débogage des Blueprints d'animation complexes.

Animation viewer curve

4.13

L'onglet Squelette Curve de Persona a été supprimé et cette fonctionnalité a été déplacée dans l'onglet Animation Curves. Vous pouvez maintenant renommer et supprimer des courbes, ainsi que la prévisualisation des données de courbe.

On homogénéise le tout en quelques sortes.



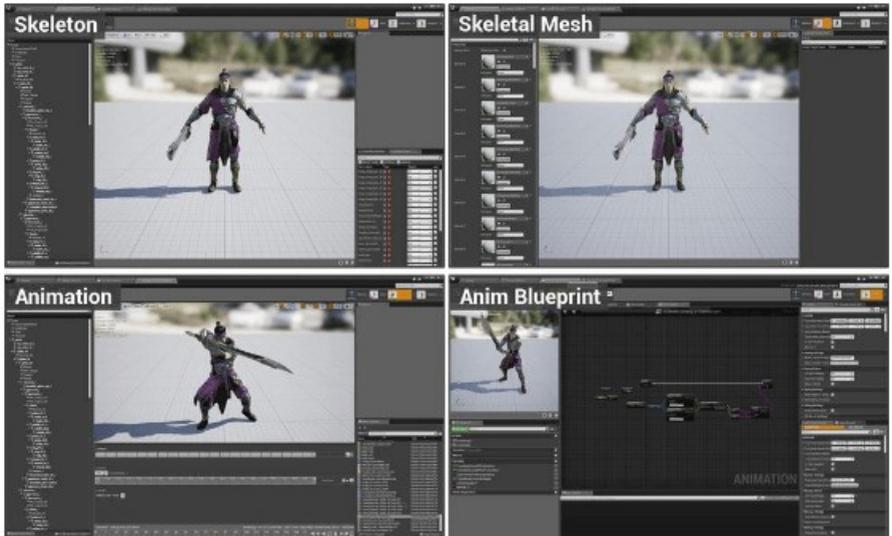
Vous pouvez voir toutes les courbes qui appartiennent au squelette actuel ou les courbes actuellement actives de prévisualisation asset. Vous pouvez également filtrer par type spécifique de courbe pour ne voir que les courbes actives.

Pour modifier la valeur de la courbe, vous pouvez soit désactiver l'option de case à cocher Auto ou tapez simplement la valeur.

Éditeur d'animation Revamp

4.14

Les outils liés à l'animation ont été divisés en éditeurs d'assets distincts plutôt que d'utiliser un éditeur avec plusieurs modes.



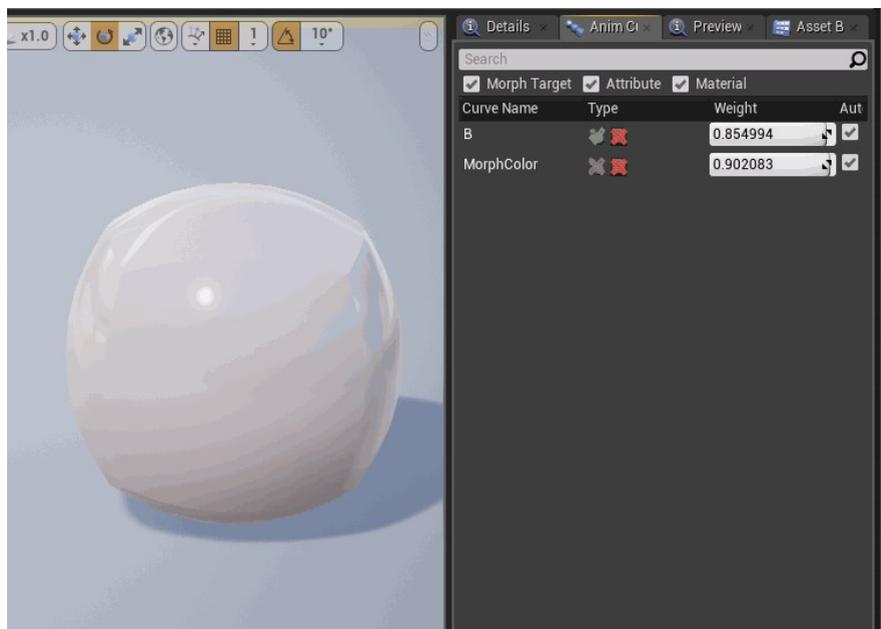
Beaucoup d'autres améliorations ont été apportées sur l'éditeur Skeletal Mesh, l'éditeur Skeleton, l'éditeur d'animation et l'éditeur de Blueprint animation.

Fenêtre courbe d'animation

4.14

Vous pouvez maintenant modifier la courbe animations grâce à la nouvelle fenêtre dédiée.

Les courbes sont visualisés en direct que vous les modifiez.



Modifier les nœuds des courbes d'animation

4.15

les courbes d'animation peuvent désormais être modifiées en utilisant la logique arbitraire dans l'animation graphique pour permettre des animations plus dynamiques. Après avoir ajouté le nouveau noeud Modify Curve , vous pouvez faire un clic droit dessus et choisir les courbes à modifier. Grâce à cela, on peut ajouter un peu de hasard

dans la courbe d'animation et obtenir des séquences encore plus réalistes visuellement.

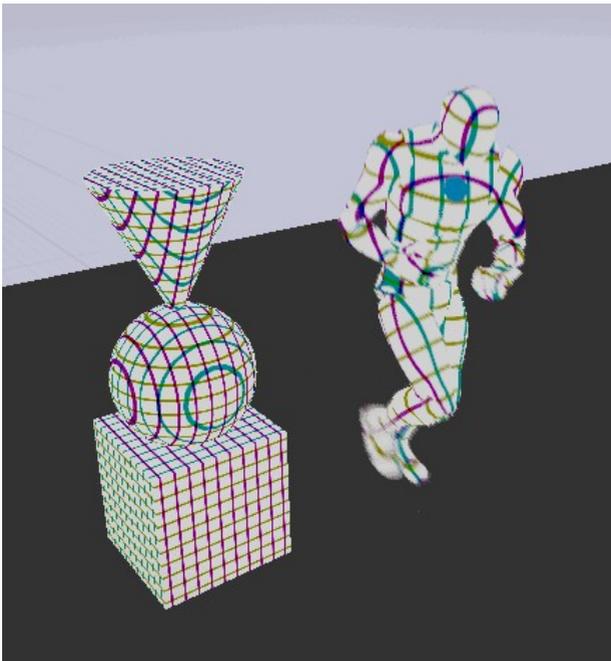
Relier des courbes aux bones

4.15

les courbes d'animation peuvent désormais être reliées à un os du Squelette afin d'être éliminées avec l'os lors du changement de LOD, particulièrement utile pour les animations faciales.

Pre-skinned local position

4.14

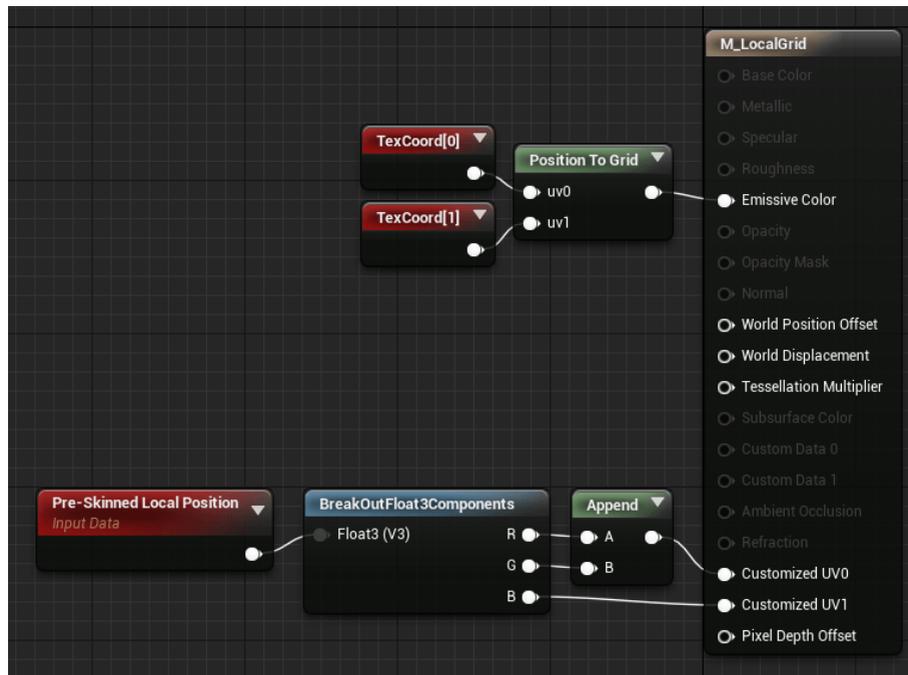


Unreal Engine 4 – Guide des mises à jour (4.20)

Les matériaux ont maintenant accès à la position de référence d'un skeletal mesh pour une utilisation dans les sorties per-vertex.

Cela permet des effets localisés sur un personnage animé. Le nœud peut être partagé pour les maillages statiques pour lesquels il renvoie la position locale.

L'exemple ci-dessous crée un motif en grille dans l'espace local, qui reste relatif à la maille du squelette pendant l'animation:



Améliorations du nœud d'animation RigidBody

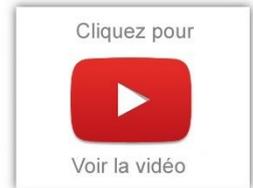
mouvements en "Local Space"

4.20

Vous pouvez désormais effectuer des mouvements en "Local Space" sur des corps simulés lorsque vous déplacez le Skeletal Mesh Component.

Des options permettent de voir la vitesse linéaire et l'accélération du composant dans l'espace et les appliquer à la simulation en espace local (à l'échelle et clampées).

Une option permet à toute articulation (joint) d'être la base de la simulation, et il est plus facile désormais de réinitialiser la simulation.



Os virtuels

4.14

Ont été ajoutés des «os virtuels» à un squelette.

Les os virtuels ne sont pas skinnables, mais contraints entre deux os existants sur le squelette. Par exemple, vous pouvez ajouter un joint, enfant d'une main, mais contraint à un joint de paume.



Corps physiques avec des parents simulés

4.16

Les bones enfants entraînés par les données d'animation du personnage alors que leurs parents sont aussi pilotés par des données de simulation physique.

Unreal Engine 4 – Guide des mises à jour (4.20)



Ainsi, le corps peut réagir plus facilement avec son entourage.

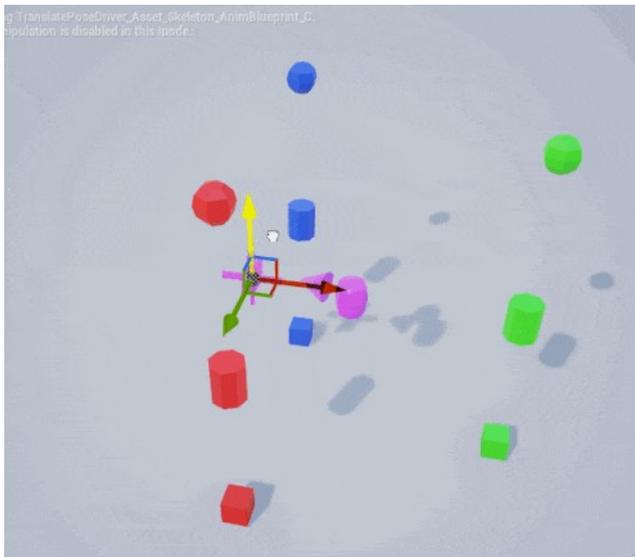
Imaginez le résultat: un impact de balle dans l'épaule par exemple... un comportement encore plus réaliste qui peut-être programmé procéduralement.

Amélioration pose driver

4.14

Le nœud Pose driver permet à un bone de conduire d'autres aspects de l'animation, basé sur un jeu d'exemples.

Il peut maintenant être une cible de morphing, par exemple la conduite d'un os épaule pour la rotation du bras.

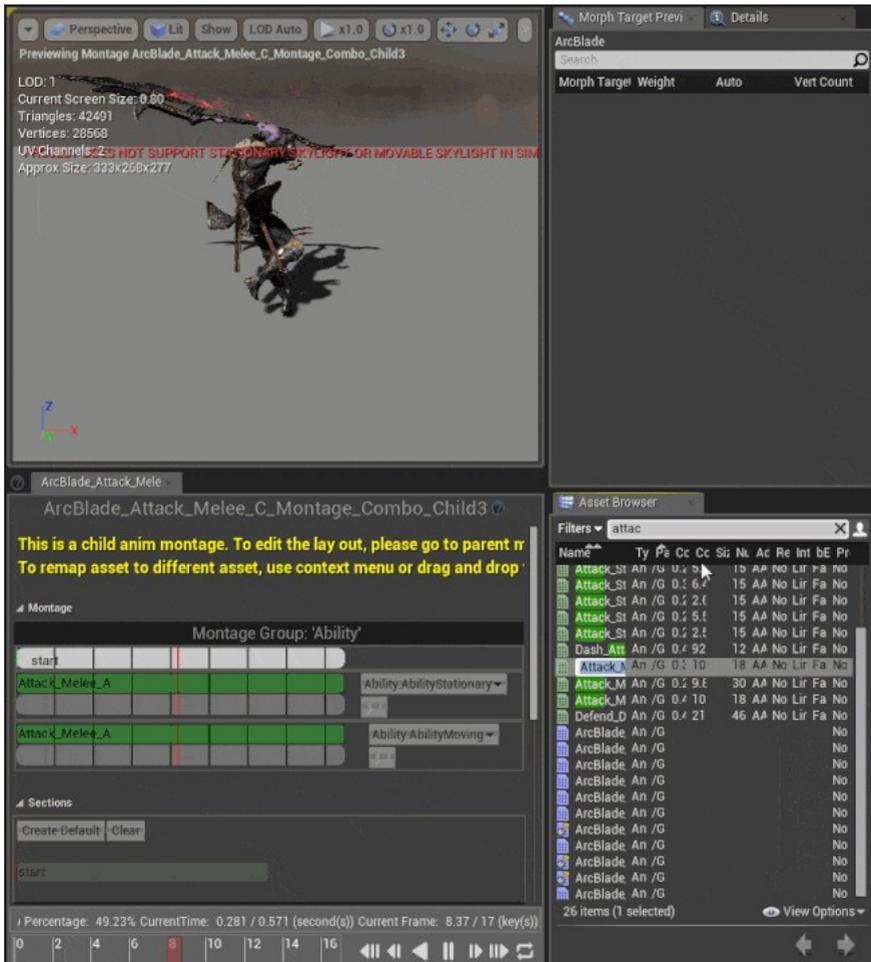


Child Animation Montages

4.14

Créer un Montage enfant basé sur un Montage parent, permet de remplacer des clips d'animation.

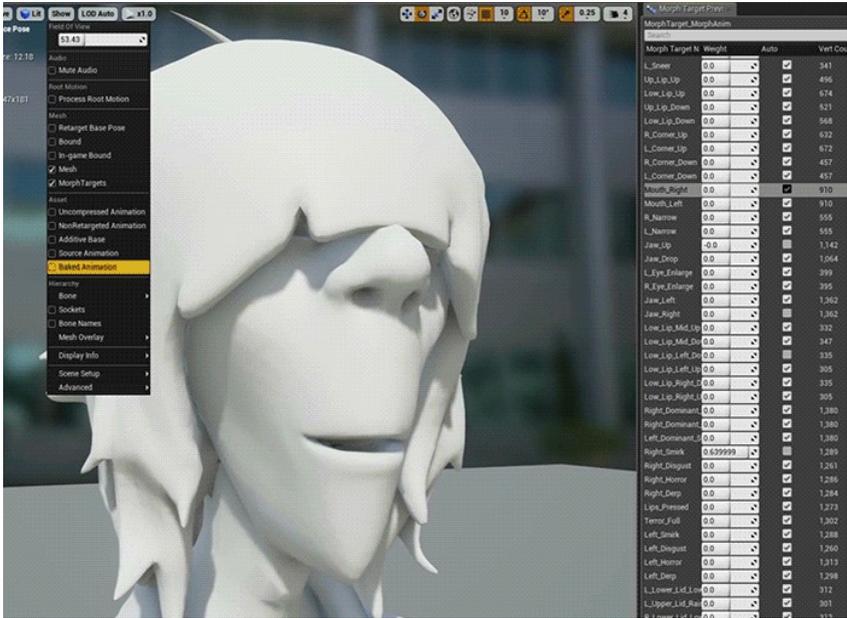
Utile pour ajouter des variations à un mouvement tout en garantissant qu'il n'aura pas d'incidence sur le gameplay.



Morph Target debug View Mode

4.14

Permet de voir quels sommets sont affectés par chaque cible de morphing.



Pose snapshotting

4.15

Possibilité de capturer une pose au runtime du Skeletal Mesh, et ce, en Blueprints pour une utilisation dans le Plan Anim!



Amélioration 'Look At' Animation Node

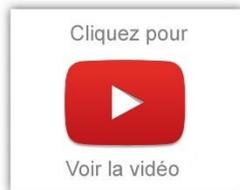
4.16



Il peut maintenant être utilisé par rapport à un bone ou un socket. La visualisation de regard sur les contrôles est améliorée. Bon, on pouvait faire sans, mais c'est tout aussi bien de ne pas avoir à passer par un empty par exemple.

Spline IK Solver

4.16



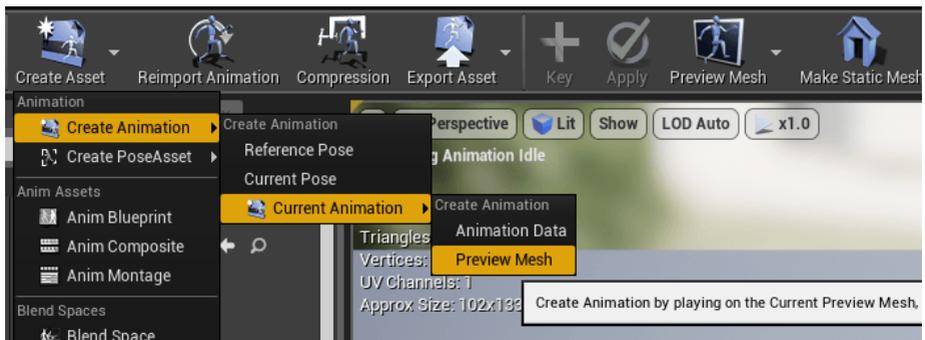
Ce nœud IK spline est utile pour le contrôler des noeuds racines des personnages (spline, ce sont les os de la colonne vertébrale).

Amélioration de l'Animation export

4.16

Un support pour la création et l'exportation des animations a été ajouté en incluant des données d'animation supplémentaires générées à partir d'un diagramme de post-traitement affecté à la maille du squelette, tel que Anim Dynamics pour la simulation de la physique.

Pour inclure ces données supplémentaires, choisissez les menus Preview Mesh de Create Animation ou Export Animation.

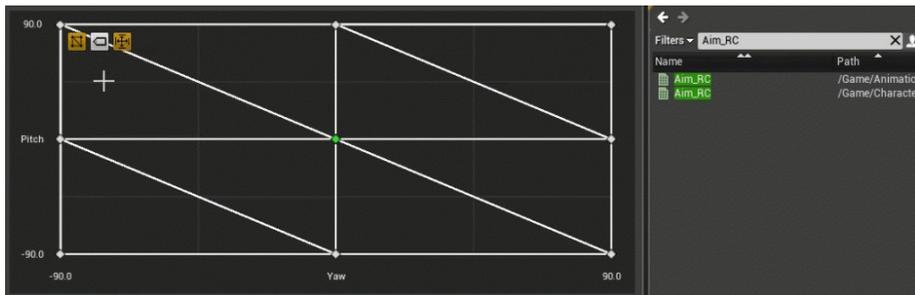


Réflexion faite, l'animation des personnages devient vraiment très complexe - mais aussi de plus en plus réaliste. Je vois déjà le désarroi sur le visage de certains de mes lecteurs qui se plaignent déjà d'avoir du mal à suivre les évolutions du logiciel.

Amélioration: éditeur Animation Blend Space

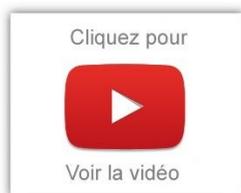
4.16

Il permet maintenant d'afficher les noms d'animation pour chaque échantillon en utilisant le bouton **Show Animation Names** dans la grille. Vous pouvez maintenant faire glisser et déposer des animations sur les échantillons existants pour les remplacer.



Animation Modifiers (Early Access Preview)

4.16



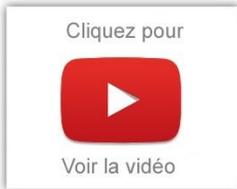
Les **Animation Modifiers** permettent d'appliquer une séquence d'actions à une séquence d'animation donnée ou un squelette, comme le pointage sur le pied droit sur le sol et l'ajout de marqueurs d'animation pour la synchronisation avec le bone ball_r à son point le plus bas (sol).

Une nouvelle série de fonctions pour accéder aux données d'animation spécifiques sont disponibles dans la bibliothèque de fonctions de l'Animation Blueprint. Accéder et appliquer un Animation Modifier se fait à travers un nouvel onglet dans Skeleton Editor et Animation Editor. Les Animation Modifiers peuvent être ajoutés à un

squelette ou une séquence d'animation. Pour les séquences d'animation, l'Animation Modifier est appliqué uniquement à la séquence elle-même. Appliqué à un squelette, il est appliqué à toutes les séquences d'animation en lien avec le squelette.

Amélioration des outils d'animation

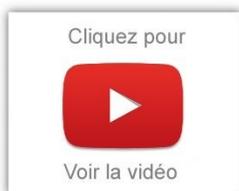
4.19



- Ajout de la liste de commandes Pinnable aux éditeurs d'animation: cela permet un accès plus facile aux commandes fréquemment utilisées. Les listes de commandes des éditeurs d'animations peuvent être attachées à la vue ;
- Ajout de 4 fenêtres d'affichage distinctes à toutes les fenêtres de l'éditeur d'animations - chacune avec ses propres paramètres.
- Ajout de la possibilité de s'accrocher et tourner autour d'un os en particulier
- Informations supplémentaires sur le débogage des composants de maillage squelettique
- Possibilité de changer la section visible d'un squelette à la volée: Exposed ShowMaterialSection
- Possibilité de "Masquer les broches non connectées" pour les nœuds BreakStruct de bouton
- Possibilité de nommer des formes de collision

PHYSIQUE ET PARTICULES

Éditeur d'effets visuels Niagara vient compléter (et remplacer à terme) Cascade ! (Early Access)



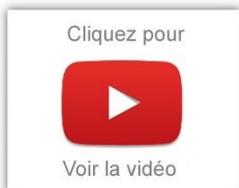
Niagara vient semble-t-il remplacer Cascade, le système de gestion des particules, un peu comme Sequencer est venu remplacer Matinee pour les animations. Nous avons réalisé [un article complet dessus](#).

4.20

Niagara vient semble-t-il remplacer Cascade, le système de gestion des particules, un peu comme Sequencer est venu remplacer Matinee pour les animations. Bon, ça ne veut pas dire que l'ancien sera supprimé, mais peut-être que tous les efforts seront concentrés sur cette nouvelle mouture, qui est pour le moment, en "Early Access". Il n'est donc pas conseillé de l'utiliser en production.



Allons voir cela d'un peu plus près !



Niagara est-il meilleur que Cascade? Sa force réside dans sa capacité à laisser l'utilisateur scripter le comportement de ses particules sans devoir intervenir coté code pour créer de nouvelles fonctionnalités.

Unreal Engine 4 – Guide des mises à jour (4.20)

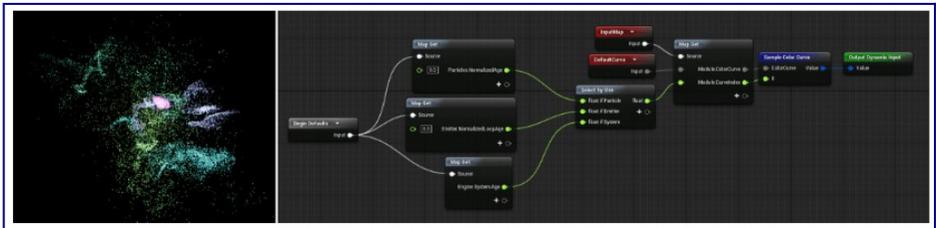
Il utilise un système de nœuds visuels que les artistes peuvent utiliser facilement. On peut toutefois se poser des questions quant aux performances de Niagara / Cascade. A cela, Epic répond que le code généré par les blueprints est interprété par un système de bas niveau permettant de hautes performances.

Toutefois, **cela fonctionne sur CPU**, et sur GP, mais pas sur toutes les plateformes (voir un peu plus loin).

Pour un aperçu de Niagara, vous pouvez visionner la présentation [VFX programmable de GDC 2018 avec Niagara de Unreal Engine](#) et lire la documentation de [Niagara](#).

Améliorations de la conception et de la création d'effets

4.20



Gauche - Système de particules utilisant un module d'entrée dynamique; Droite - Module d'entrée dynamique

- **Skeletal Meshes** peuvent spécifier leur émission à partir de la surface, soit par le nom du matériau, soit par une région d'influence de l'os nommée.
- La spécification des valeurs par défaut dans les modules a été améliorée, permettant à de nombreux comportements d'appeler des fonctions à l'aide d'entrées dynamiques par défaut.
- Mesh particules supportent maintenant la vitesse angulaire.
- La prise en charge des beams a été ajoutée au rendu Ribbon avec les nouveaux modules correspondants.

Unreal Engine 4 – Guide des mises à jour (4.20)

- Les dépendances entre modules peuvent maintenant être définies, permettant à l'utilisateur d'être informé quand il place la pile dans une mauvaise configuration. En outre, les utilisateurs ont des options de **correction automatique**.
- De nombreuses améliorations ont été apportées à merging System Emitters et Base Emitters,, améliorant ainsi la stabilité globale.
- Les modules peuvent maintenant être activés / désactivés dans la pile. Cela fonctionnera également pour l'héritage.
- **La prise en charge du séquenceur et Blueprint** pour la définition des variables d'espace de nom d'utilisateur Niagara a été ajoutée.
- Vous pouvez générer des paramètres à l'aide d'**expressions HLSL** personnalisées, d'**entrées dynamiques (graph snippets)**, de liens vers d'autres **variables** ou **par valeur**.
- En option, les particules peuvent maintenant avoir un **identifiant persistant**
- Plusieurs moteurs de rendu de chaque type peuvent être appliqués à un émetteur. Un émetteur peut avoir deux moteurs de rendu, l'un tirant sa position de la position d'une particule et l'autre tirant sa position de la position de décalage d'une particule.
- Le **plugin Niagara Extras** contient également un matériau de débogage qui achemine divers paramètres par particule vers un affichage de type dialogue.
- importateur CSV

une grande variété de fonctionnalités pour Niagara a été ajoutée sous le **système de test** automatisé.

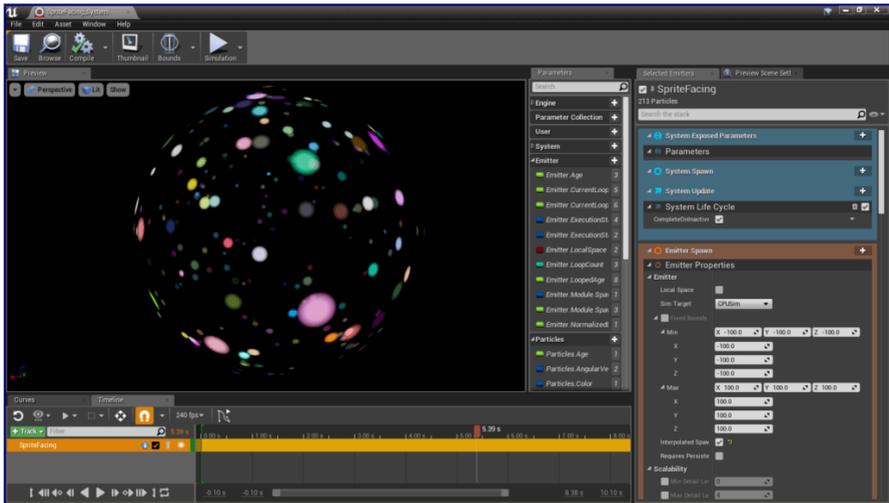
Interface utilisateur mise à jour

4.20

L'interface Niagara a été conçue pour rendre les effets complexes intuitifs à créer. Elle utilise une "pile" comme principale méthode de combinaison de morceaux de logique de script. Dans la pile, vous trouverez une **chronologie** pour contrôler les aspects de l'effet au fil du temps, un **panneau de paramètres** pour un accès facile aux

Unreal Engine 4 – Guide des mises à jour (4.20)

variables disponibles dans l'effet, une **feuille de calcul d'attributs** pour trouver rapidement et réagir à l'information que l'effet est en cours d'exécution.



Nouveaux modules

4.20

Tous les modules de Niagara ont été mis à jour ou réécrits pour prendre en charge les comportements couramment utilisés. De nouvelles fonctionnalités de l'interface utilisateur ont également été ajoutées pour la pile Niagara qui imitent les options des développeurs avec UProperties en C++, permettant l'activation / désactivation en ligne ou l'affichage des variables en fonction de l'état d'une autre variable.

Simulation GPU

4.20

Niagara prend maintenant en charge la simulation **GPU** lorsqu'il est utilisé sur les plates-formes **DX11**, **PS4**, **Xbox One**, **OpenGL (ES3.1)** et **Metal**. Il est prévu que **Vulkan** et **Switch** prennent en charge la

Unreal Engine 4 – Guide des mises à jour (4.20)

simulation GPU **dans une future version**. Les limitations actuelles et les problèmes connus avec la simulation GPU sont décrits ci-dessous:

- Le support complet de Niagara nécessite la capacité de relire les données du GPU. Actuellement, seules nos interfaces de rendu **DX11** et **PS4** supportent cette fonctionnalité, et **OpenGL** et **Metal** sont en cours.
- **Les champs Collision, Curves et Curl Noise** sont pris en charge sur le GPU. **Les maillages**, les **maillages dépouillés**, les **composants de spline** et d'autres interfaces de données spécialisées ne sont pas encore pris en charge. L'API permettant aux shaders GPU d'interagir avec **UnigaraDataInterfaces** a également été repensée.
- Les rendus **Sprite** et **Instanced Static Mesh** à partir de particules est pris en charge sur les simulations GPU. À l'heure actuelle, la **génération de lumière** à partir de particules et de rubans à partir de particules ne fonctionne pas sur le GPU.
- Les événements ne fonctionnent que sur le processeur et subiront des modifications importantes après Unreal Engine 4.20.

Simulation CPU

4.20

La simulation CPU Niagara fonctionne maintenant sur **PC, PS4, Xbox One, OpenGL (ES3.1)** et **Metal**. Pour le moment, **Vulkan** et **Switch** ne sont pas supportés.

- La machine virtuelle du processeur (VM) compile maintenant son contenu au DDC sur un thread d'arrière-plan, ce qui améliore considérablement la vitesse de compilation globale et l'efficacité de l'équipe. Un travail supplémentaire est nécessaire pour que l'étape d'optimisation VM finale se produise dans ShaderCompileWorker car elle dépend de bibliothèques non thread-safe. Les dépendances de compilation sont correctement suivies à travers les modules.

Unreal Engine 4 – Guide des mises à jour (4.20)

- La simulation de physique sur l'unité centrale devrait modéliser correctement les valeurs **physiques du matériau** pour la friction et la restitution (rebondissement).
- Les émetteurs vont maintenant simuler en parallèle sur les threads de travail.

Voilà, je vous propose une petite présentation réalisée par [Meletou](#) qui vous présentera de façon pratique une première utilisation de Niagara:



Amélioration automatique Convex Collision Meshes

4.8

Avec le Static Mesh Editor pour générer automatiquement un maillage de collision pour le modèle, le **maillage généré sera de bien meilleure qualité et terminé plus rapidement!**



Full Scene Particle Collision avec Mesh Distance Fields

4.9

Unreal Engine 4 – Guide des mises à jour (4.20)

Possibilité d'**entrer en collision avec toute la scène**. Cela peut être utilisé, par exemple, pour des étincelles qui entrent en collision avec la scène.



Pour d'autres effets comme la neige s'accumulant au sol.

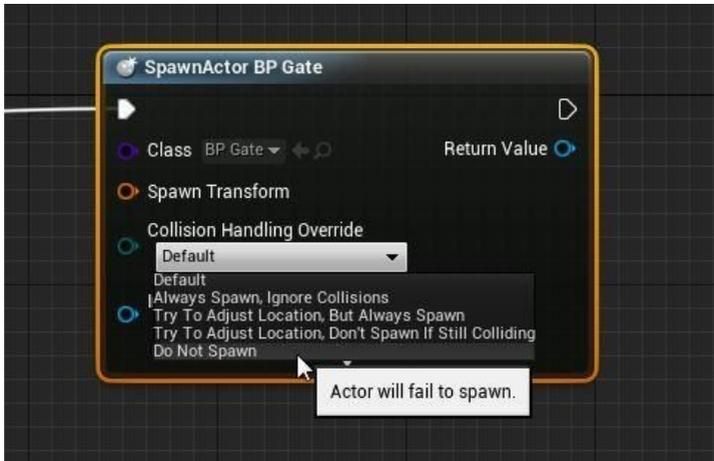


Dans le mode collision, il est possible de spécifier la distance du champ: "distance field", «Generate Mesh Distance Fields" doit être activé dans les paramètres .

Actor Encroachment Detection

4.9

La **détection d'empiètement** fonctionne dans plus de situations. De nouvelles options de personnalisation sont disponibles.



L'option "Spawn Even If Colliding" ("bNoCollisionFail" en C++) lorsque la création d'un actor est faite pour supporter les quatre méthodes de collision:

- Le créer de toute façon.
- Le créer à un emplacement approprié le plus proche possible, sinon le créer là de toute façon.
- Ne le créer que si un emplacement approprié peut être trouvé à proximité, sinon ne pas le créer.
- Ne pas le créer.

Les Blueprints existants seront mis à jour automatiquement pour choisir l'emplacement correct.

Les classes d'actors ont un nouveau membre appelé Spawn Collision Handling Method qui contrôle la façon de résoudre les collisions de création par défaut.

Pour les actors avec une composante de mouvement, on suppose que le composant mis à jour est la Spawn Collision Handling Method et sera le seul composant testé pour les empiètements.

Pour les autres actors, tous les composants avec les paramètres de

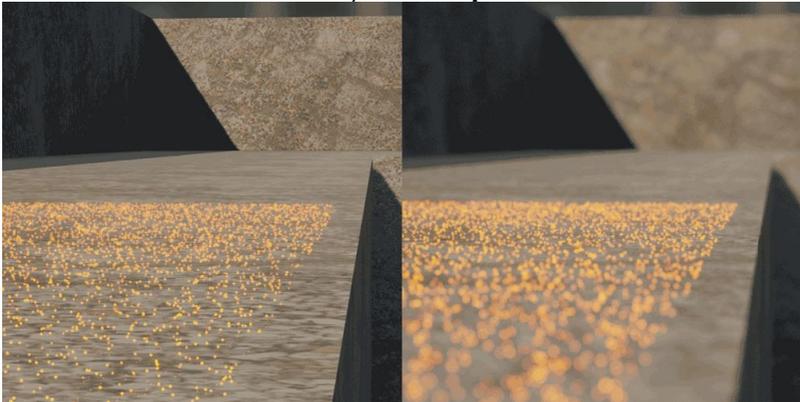
collision appropriés seront désormais vérifiés.

Vérifiez vos paramètres de collision et la méthode de gestion dans tous les cas pour éviter les problèmes.

Profondeur de champ de particules

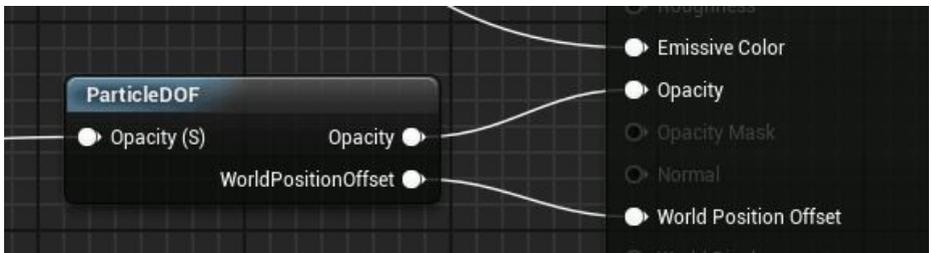
4.11

Les nouvelles "material functions" permettent d'**utiliser le Depth of Field** (profondeur de champ – le fait qu'on puisse faire un focus sur une zone et rendre le reste flou) **sur les particules**.



L'image de gauche montre une scène simple, avec beaucoup de particules placées sur le terrain.

L'image de droite utilise un cercle profondeur de champ activé et la nouvelle fonction de matériau (ParticleDOF):



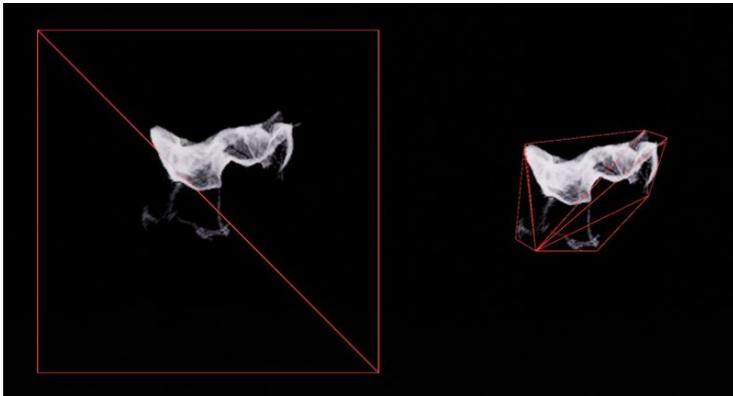
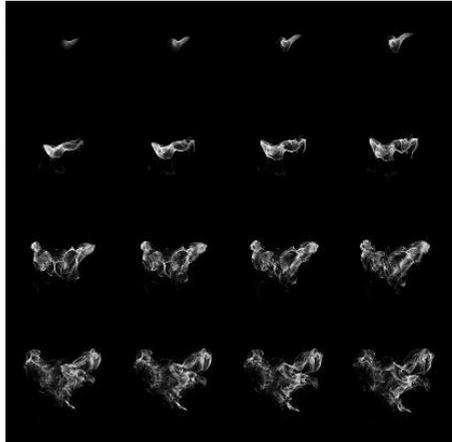
Découpages de particules (flipbook rapide des particules pour le rendu)

4.11

Les particules utilisant des animations flipbook ont tendance à gaspiller des zones.

A titre d'exemple, ci-dessous, la texture est essentiellement constituée de pixels transparents.

Maintenant, les particules ont une délimitation beaucoup plus serrée et cela gaspille moins de ressources:

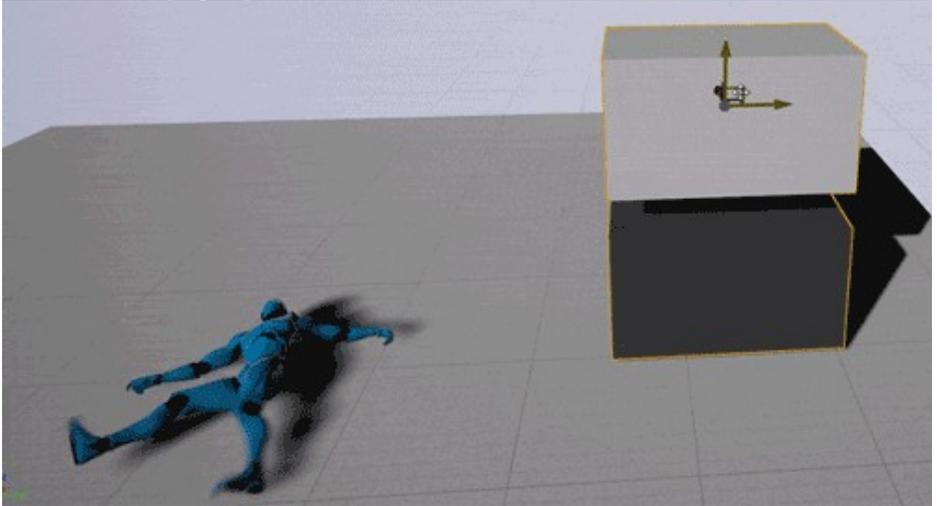


C'est encore un **compromis entre économie de mémoire pour le stockage et vitesse d'exécution**: ce qui est très lié au final. En effet, si la carte graphique vient à manquer de mémoire, il y aura des aller-retours plus fréquents et donc une perte de performance.

Physical component animations

4.13

Permet de faire facilement l'animation du maillage du squelette par des moteurs physiques.



Le composant permet de définir l'utilisation de profils d'animation physique pré-configurés qui peuvent être créés et modifiés à l'intérieur de PhAT. La nouvelle fonctionnalité "**Physical Animation Profiles**" fournit un moyen de personnaliser la physique dans différents contextes, ainsi que le peaufiner les animations spéciales. Bon, difficile de vous décrire plus cela, il faudra qu'on se trouve une vidéo qui parle de ça...

Upgrade de PhysX en 3.4

4.14

Le moteur physique inclus, PhysX, a été mis à jour vers la dernière version, ce qui améliore fortement la performance ; de nouvelles fonctionnalités du moteur physique seront intégrées dans les versions à venir du moteur de jeu.



Cette version de PhysX permet la détection continue de Collision (CCD) sur les objets cinématiques, ce qui permet des **collisions précises entre corps rigides en mouvement rapide**. C'est important si vous tirez avec un projectile comme une balle de fusil dans un mur par exemple.

Optimisations PhysX

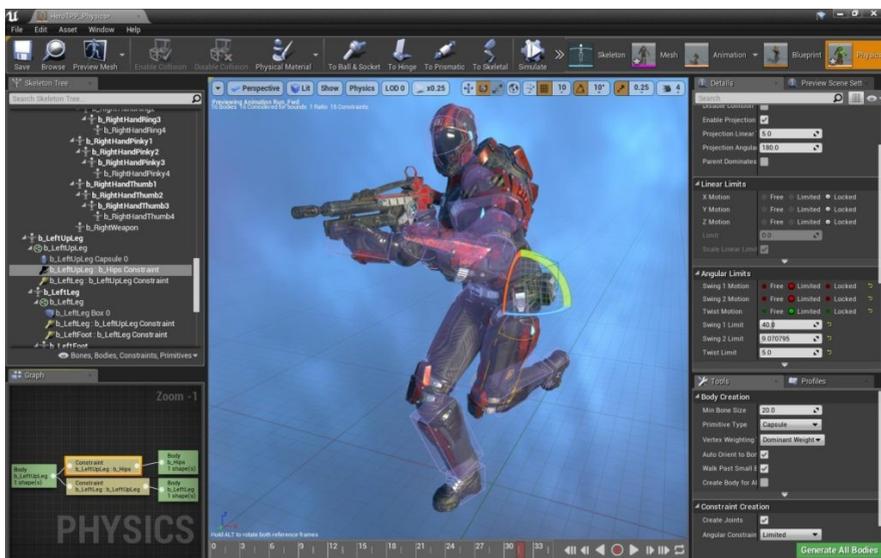
4.17

Les tâches PhysX sont maintenant regroupées ensemble pour réduire la charge, avec `p.BatchPhysXTasksSize`. Ce compromis de parallélisme doit être réglé en fonction de votre jeu.

Nouvel éditeur d'assets physiques

4.18

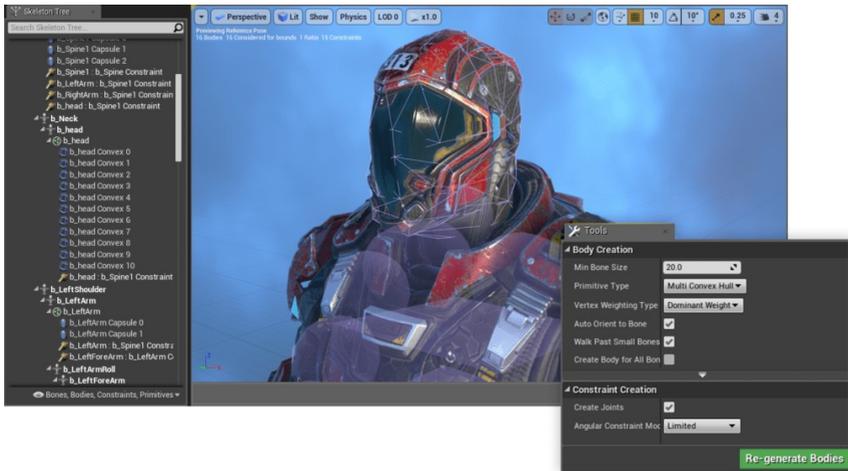
Le Physics Asset Editor (anciennement connu sous le nom de PhAT) a été entièrement remanié. L'interface utilisateur permet une vue améliorée du Skeleton Tree avec un meilleur filtrage et affichage de tous les éléments de la scène et une nouvelle vue pour configurer visuellement les contraintes entre les corps en utilisant des nœuds. Vous pouvez même créer des boucles de contraintes.



Les animations peuvent maintenant être choisies à tout moment, mises en pause, nettoyées, etc.

Il est dorénavant possible de générer automatiquement des corps physiques en utilisant plusieurs convex hulls par os. Cela permet d'obtenir une approximation de la géométrie non convexe à l'aide de plusieurs coques convexes ! Il fallait y penser...

Unreal Engine 4 – Guide des mises à jour (4.20)



Prise en charge des véhicules améliorée

4.14

Les développeurs ont changé la façon dont les forces sont appliquées au niveau des pneus. Auparavant, elles étaient appliquées au barycentre du véhicule.

Les déplacements des véhicules n'étaient pas toujours gérés de manière très réaliste : les forces qui s'appliquaient sur les pneus étaient considérées sur le véhicule lui-même et non sur le pneu. Ainsi, le moteur n'arrivait pas à simuler un véhicule qui tanguait.

Simple Wheeled Vehicle Movement Component permet la suspension de la roue et la friction des pneus sans s'ennuyer avec les complexités du moteur et la simulation de la transmission.

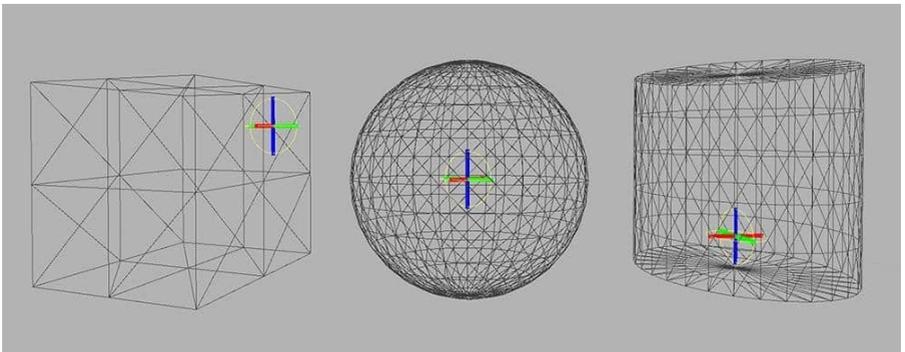


Ce composant vous permet d'appliquer facilement le couple aux pneus.

Mass properties debugging

4.15

possibilité de visualiser le centre de la masse et de l'inertie associée à des objets physique. A ben, en voilà une bonne idée !



Pour visualiser les propriétés de masse, autoriser showflag (Show > Advanced > Mass Properties). L'épaisseur de chaque axe indique le moment d'inertie le long de l'axe.

Le support de véhicules est déplacé sous forme de plugin

4.15

Le support PhysX véhicule est maintenant en option pour économiser de l'espace disque et de mémoire aux utilisateurs qui n'emploient pas ce support. Il y a également ajout de plusieurs points d'extension physique utiles au moteur (par exemple `OnPhysSceneInit/Term`, `OnPhysSceneStep`) afin de rendre plus facile pour les autres développeurs l'écriture de leurs propres systèmes similaires. Une étape manuelle est nécessaire si vous souhaitez migrer des assets anciens 'TireType' (déconseillé dans le moteur) pour les nouveaux assets 'TireConfig' (définis dans le nouveau plug-in PhysXVehicles). Une nouvelle commande de console disponible dans l'éditeur UE4 le fait pour vous: il suffit de convoquer la console et entrez 'ConvertTireTypes'.

Cela va créer de nouveaux assetsTireConfig, copier les propriétés et les affecter à des Blueprints VehicleWheel. Les anciens assets de TireType peuvent alors être enlevés.

Je ne sais pas quoi penser du fait qu'ils retirent les véhicules et les placent en plugin... probablement ont-ils jugé trop léger cette fonctionnalité et préfèrent-ils laisser la possibilité à d'autres plugins de se greffer à la place de ce dernier.

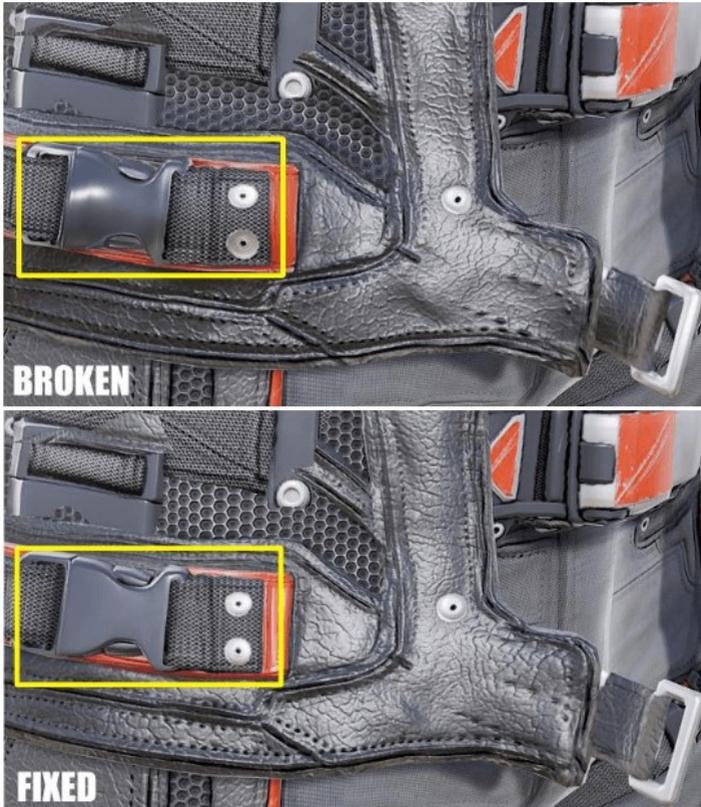
Ce n'est pas toujours évident de comprendre les choix architecturaux réalisés derrière un moteur de cette importance !

Outils Vêtements - Cloth

Amélioration du cloth skinning

4.14

Il est possible de calculer ses propres données **mesh-to-mesh skinning** pour les vêtements, donc plutôt que d'utiliser les données de rendu exportées dans un fichier .apx ou .apb, nous utilisons maintenant ce que UE4 a déjà depuis l'asset APEX-exporté et Reskin. Cela corrige des soucis comme les normals qui pouvaient être mal appliqués après import comme ci-dessous :



"clothing Paint Tool" (Experimental)

4.17

Le framework "clothing Paint Tool" a été repensé pour permettre d'ajouter plus de fonctionnalités à l'avenir. Si vous ne connaissez pas l'outil, je vous engage à aller voir [cette page](#).

C'est vrai qu'il s'agit d'outils assez récents (4.16) que je n'ai pas encore eu l'occasion personnellement d'utiliser.

La simulation peut-être faite à postero et le changement d'asset ne pose pas de soucis (par rapport à l'approche "one shot" précédente)

Support de "masks" de vêtements: permet de définir plusieurs masques que l'on peut échanger au cours du développement tout en testant différentes configurations.

Framework outils remanié pour peindre le tissu:

- Ajout peinture de vêtements lisses
- Ajout remplissage de vêtements
- Ajout du panneau d'édition de configuration à l'onglet de vêtements pour éviter d'avoir à sortir de l'onglet 'Asset Details'.

Outil vêtement mis en place

4.18

Un tout nouvel outil "built-in" a été mise en place sous l'éditeur pour gérer les vêtements. Il n'est donc plus nécessaire de passer par Apex de Nvidia.

Une fenêtre "Vêtements" permet d'éditer un ensemble de propriétés spécifiques aux vêtements qui fournissent plus de contrôle sur les contraintes d'habillement, l'élimination sélective des sommets d'auto-collision, l'échelle d'inertie.

Unreal Engine 4 – Guide des mises à jour (4.20)

Les outils de peinture améliorés permettent de peindre "la déformabilité du vêtement" directement sur le maillage. Les nouveaux outils de peinture incluent: Brosse, Dégradé, Lisse, Remplissage. Les paramètres peints peuvent être désactivés au lieu d'être écrasés pour faciliter l'itération.

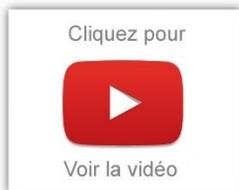


Les outils d'habillement ont été intégrés aux assets physiques pour la création de collisions. Il est également plus facile de créer des données de vêtements dans l'éditeur de maillage squelette en cliquant avec le bouton droit sur la section de maillage à convertir.

Anim Drive sur l'outil Cloth

4.19

La nouvelle fonction "Anim Drive" ajoute un contrôle via les Blueprints pour créer des interactions avec la physique des vêtements, permettant ainsi, par exemple, de simuler des ressorts qui tirent les vêtements.



- **Prise en charge du contrôle de la gravité** pour les simulations de vêtements - L'utilisation de l'interactor object sur un composant de maillage squelettique permet de spécifier une priorité de gravité arbitraire.
- Ajout de la fonctionnalité de **auto-range feature** aux plages de vue de tissu - Ajout d'un moyen aux outils pour étendre ces plages si nécessaire
- **Visualisation de vêtements améliorée** - activé lorsque le mode Peinture de vêtements est actif.
- Le menu Clothing Create Menu utilise par défaut l'élément physique de maillage squelettique

Améliorations au niveau des vêtements

4.20

Les asset physiques supportent maintenant les capsules coniques pour la collision dans la simulation de vêtements.



Remarque: elles ne sont pas prises en charge pour les collisions dans des simulations de corps rigides. Wtf ?

Vous pouvez également copier les couleurs de vertex Skeletal Mesh dans n'importe quel Clothing Parameter Mask sélectionné.

PROGRAMMATION BLUEPRINTS ET C++

Blueprint Merge Tool

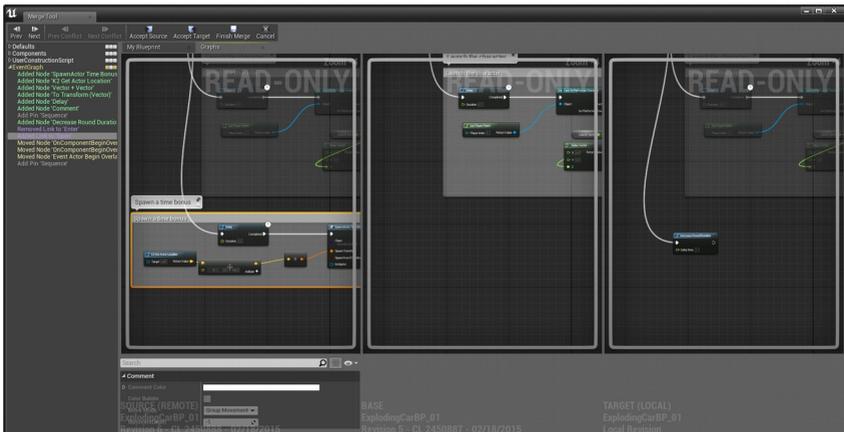
4.7

Il est maintenant possible de fusionner deux Blueprints.

L'outil permet de visualiser les différences et aide pour la combinaison des deux.

Il y a 3 façons d'utiliser cet outil:

- **File -> Merge** dans le Blueprint Editor
- **Clic-droit** sur un asset et select **Merge**
- Ou en ligne de commande: **UE4Editor.exe -diff [options] <File A> <File B> <Common Base File> <Output File>**



Là encore, chapeau... Avant j'hésitais beaucoup plus entre l'approche C++ et Blueprint, par peur justement de ne pas pouvoir facilement fusionner deux Blueprints.

Unreal Engine 4 – Guide des mises à jour (4.20)

Il fallait recréer les variables, les fonctions, copier-coller chaque sous-graphe dans les Blueprints... avec la possibilité d'avoir des liens qui sautent, etc.

Donc, oui, grand soulagement ici.

Blueprintable Components

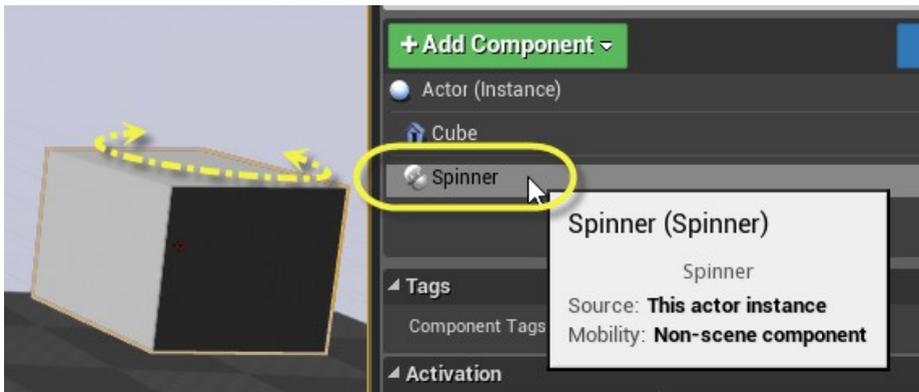
4.7

Il est désormais possible de **créer des composants "indépendants", de les programmer et de les ajouter à la volée à un acteur de la scène**. Imaginons un composant "Fait Tourner", vous le programmez pour qu'il tourne automatiquement et hop, vous l'ajoutez à un acteur de la scène. Plutôt cool non ?

Pour cela: Cliquez sur **"Add Component"**, puis sélectionnez **"New Blueprint Script Component..."**

On choisit ensuite la **Parent Class** pour le nouveau composant. Et voici la partie la plus cool: ajouter un Blueprint au composant.

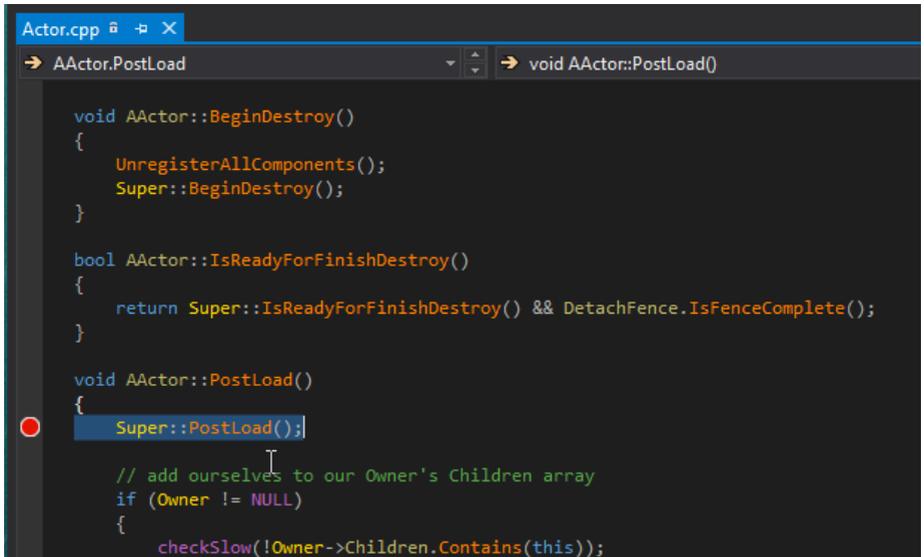
Il est maintenant possible d'ajouter ce composant à n'importe quel acteur:



Full Engine C++ Source Code

4.7

Unreal Engine inclut désormais le code source C++ complet full engine du moteur et symboles pour le débogage.



```
Actor.cpp [?] [?] [X]
→ AActor.PostLoad [?] [?] [?] → void AActor::PostLoad()

void AActor::BeginDestroy()
{
    UnregisterAllComponents();
    Super::BeginDestroy();
}

bool AActor::IsReadyForFinishDestroy()
{
    return Super::IsReadyForFinishDestroy() && DetachFence.IsFenceComplete();
}

void AActor::PostLoad()
{
    Super::PostLoad();

    // add ourselves to our Owner's Children array
    if (Owner != NULL)
    {
        checkSlow(!Owner->Children.Contains(this));
    }
}
```

Pour les novices, cela permet progressivement de passer des Blueprints au C++ sans qu'il y ait besoin d'une procédure complexe de téléchargement, de recompilation de l'éditeur et de configuration.

En même temps, en cas de crash (et oui, ça arrive souvent quand même), on a accès à un rapport plus complet.

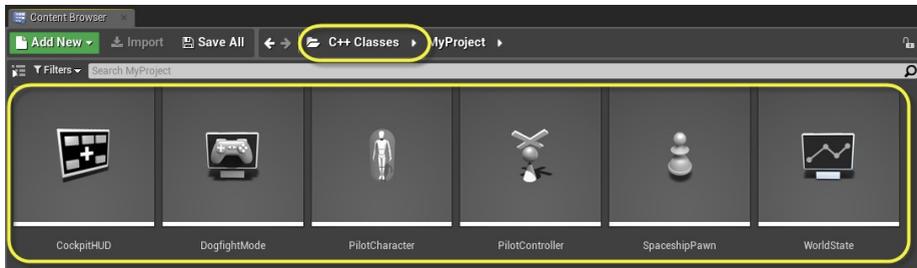
Il est toujours possible de recompiler l'éditeur.

Pour cela, il suffit de se rendre sur le [GitHub](#). C'est nécessaire pour, par exemple, utiliser Nvidia Gameworks.

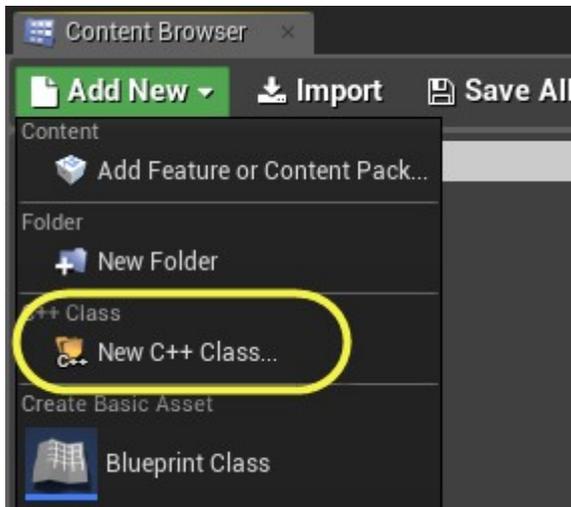
C++ Classes dans le Content Browser

4.7

Les classes C++ apparaissent maintenant dans le Content Browser!



On peut les créer directement à partir de là !

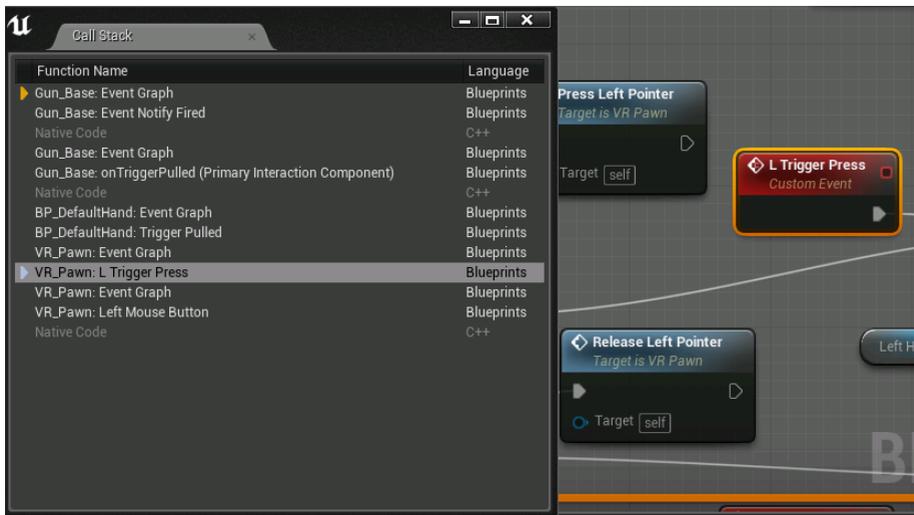


Mode Debug

4.19

Affichage de la pile d'appels (Call Stack) pour le débogage Blueprint

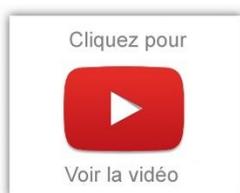
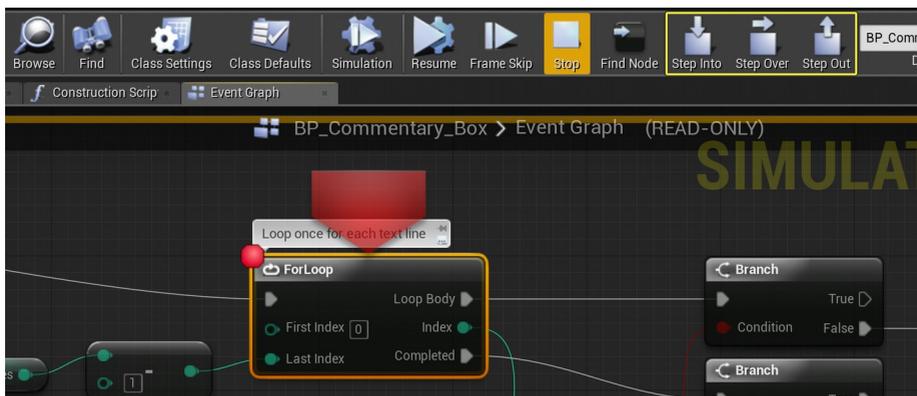
Un contrôle affichant la pile d'appels Blueprint actuelle a été ajouté au menu Outils de développement. Un double-clic sur les entrées de la pile d'appels focalise le nœud correspondant dans l'éditeur de plan.



Améliorations uniques du Blueprint Debugging

Le débogueur Blueprint du moteur prend désormais en charge les opérations Step Into (F11), Step Over (F10) et Step Out (Alt + Maj + F11) lorsqu'il est arrêté via un Breakpoint, ce qui rend les sessions de débogage plus rapides et plus faciles.

Unreal Engine 4 – Guide des mises à jour (4.20)



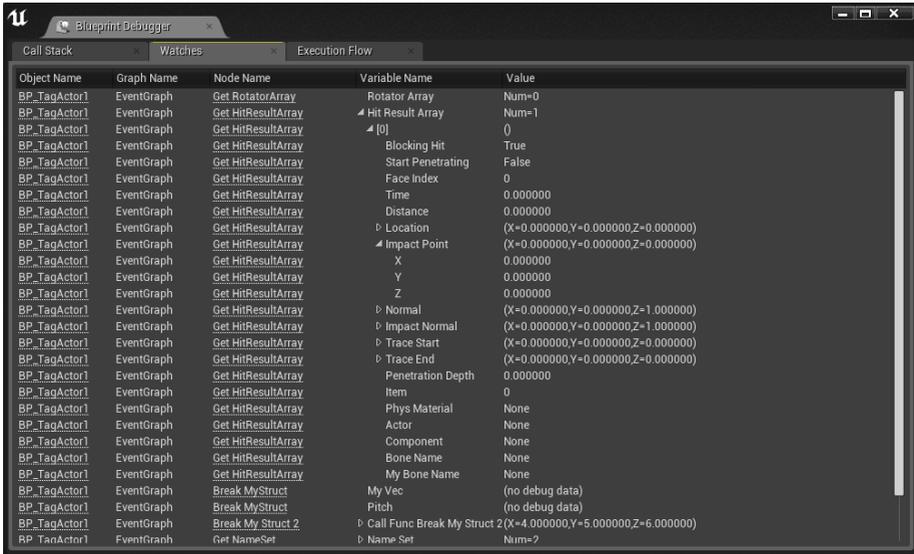
Il est aussi maintenant possible de casser les liens avec les macros et les nœuds de graphe.

Blueprint Watch Window

4.20

Cette fenêtre est conçue pour accélérer le débogage en vous donnant accès aux variables et aux nœuds que vous souhaitez regarder, même sur plusieurs Blueprints. Les données de chaque Blueprint que vous ouvrez dans l'éditeur, et qui fait partie de la pile d'appels en cours, seront regroupées dans une liste unique, vous permettant d'inspecter les variables et les sorties de fonction. En outre, vous pouvez changer de Blueprints avec facilité. Vous pouvez cliquer sur une entrée dans la colonne "Nom du nœud" pour accéder au nœud nommé dans un Blueprint, tandis que la sélection des entrées dans la colonne "Nom de l'objet" sélectionnera l'instance de l'objet associé à cette entrée. Les tableaux, ensembles, cartes et autres structures de données peuvent être étendus, ce qui permet un examen détaillé des données qu'ils contiennent rapidement et facilement.

Unreal Engine 4 – Guide des mises à jour (4.20)



Problème des Blueprint Cycliques

4.7

Certains Blueprints faisant appel à d'autres Blueprints pouvait planter si ces autres faisaient eux-même appel à ces premiers Blueprints: c'est le problème de la boucle infinie.

L'éditeur sait maintenant repérer ce genre de problèmes et évite des plantages à l'exécution.

Améliorations concernant l'Unreal C++

4.7

Le code C++ utilisé par Unreal est rempli de macros permettant de l'utiliser sous forme scriptée (ou assimilée). Il n'est pas évident d'entrer rapidement dans ce genre de code même quand on maîtrise

le C++.

Il y a donc un effort continu pour simplifier l'approche et rendre le code plus "familier" (comprendre: "plus standard")

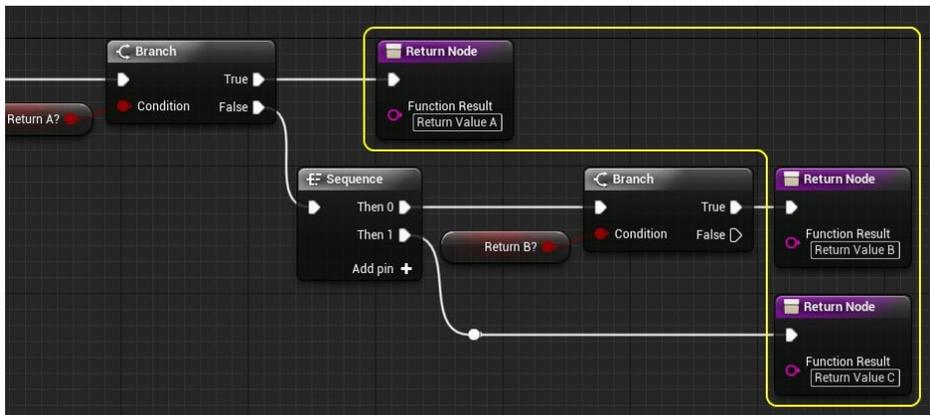
Donc, plus besoin de constructeur particulier (utilisant un argument de type Object Initializer). Les sous-objets ont une syntaxe plus simple. On peut avoir plusieurs classes dans un seul header C++.

On peut déclarer des types Unreal (comme les Unreal Enums ou delegates) dans un header sans avoir à passer par une classe spéciale, etc.

Bref, les programmeurs C++ apprécieront.

Nœuds de retour multiples dans les Blueprints

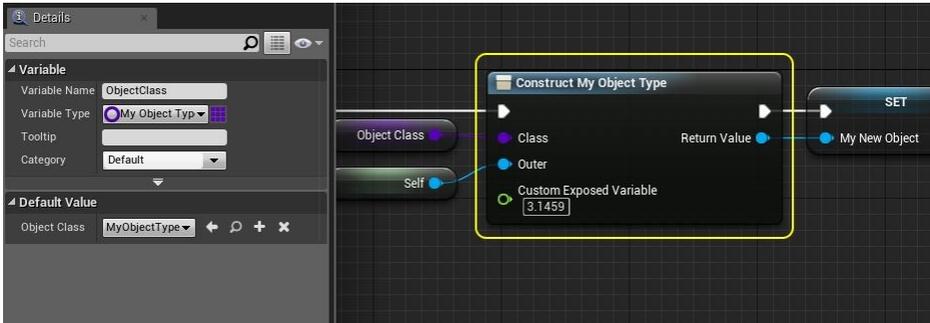
4.9



Une fonction peut maintenant avoir plusieurs nœuds de retour. Dès que l'exécution atteint un retour de l'exécution, elle est terminée. Ceci peut être utilisé pour revenir au début d'une fonction de Blueprint, ou pour avoir des valeurs de retour différentes. C'est une fonctionnalité qui se faisait attendre...

Construire des objets personnalisés dans Blueprints

4.9

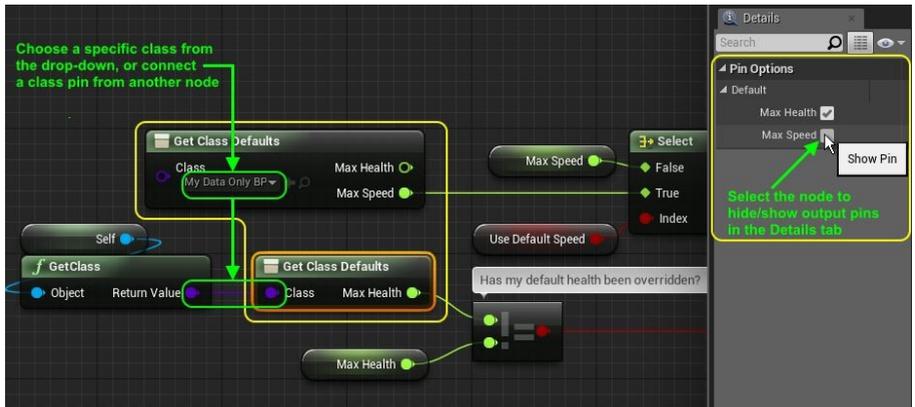


Une fonction qui m'aurait bien servi: pouvoir créer des objets autres que des acteurs (spawn actor from class) - ben voilà, maintenant c'est fait.

Valeur par défaut de la classe Blueprint

4.9

Dans les Blueprints, il est maintenant possible d'accéder aux valeurs par défaut d'une classe à travers le nouveau nœud "Get Class Defaults".



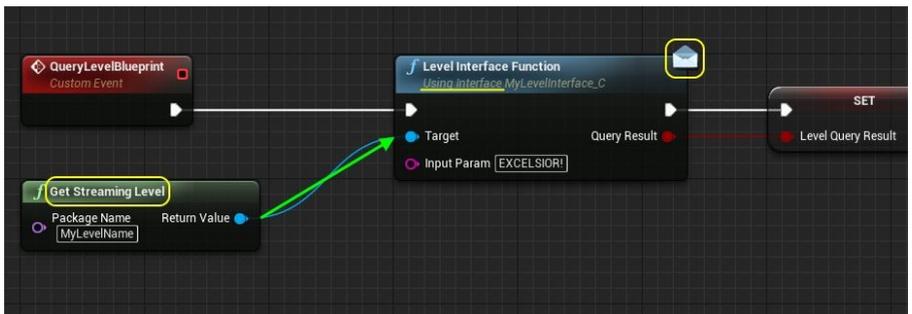
Cela peut être utile lorsque vous travaillez avec des Blueprints de données seulement, vous donnant accès à ces valeurs de Blueprint sans avoir à créer d'instance.

Level Blueprint Communication

4.9

Les Level Blueprints peuvent désormais être accessibles par le biais d'interfaces.

Ils peuvent adopter et appliquer des fonctions de l'interface comme la classe Blueprints le peut.



Après qu'une interface ait été ajoutée à Level Blueprint, les appels peuvent être faits au travers des nœuds d'interface (en utilisant un

niveau de streaming comme cible).

Utilisez le nœud "Get Streaming Level" pour pouvoir communiquer. Quand vous utilisez "Get Streaming Level" cela ne fonctionne que pour les sous-niveaux.

Blueprint Asset Ids

4.9

Les Blueprints peuvent charger des assets à la demande et non automatiquement. Il y a deux nouveaux types de données: "Asset ID" et "Classe d'assets ID".

Un Asset ID représente un asset, et peut être déterminé en objet en utilisant le nœud "Resolve Asset". Si l'asset n'a pas été chargé, alors le nœud "Resolve Asset" retourne un objet non valide. Vous pouvez charger l'asset en utilisant le nœud "Load Asset".

De même, une "Classe d'assets ID" permet de résoudre dans une classe, et peut être chargé avec le nœud "Load Asset Class".

Actor Tick Intervals

4.9

Vous pouvez maintenant spécifier **un intervalle pour l'exécution d'une fonction tick** qui sera exécutée pour réduire les temps d'exécution.

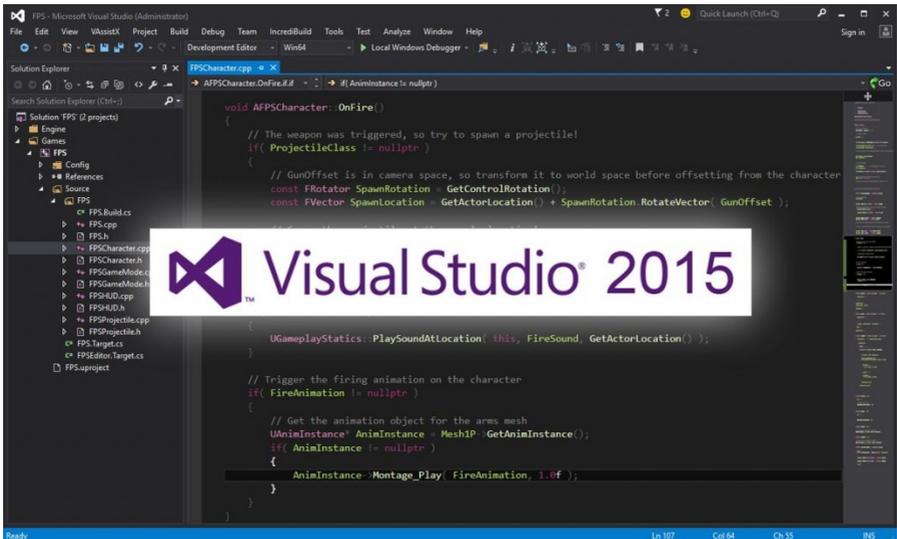
Accessible via «Actor and Component default properties» et pour les programmeurs, il peut être réglé en C++ comme toute autre propriété de la fonction Tick.

Support Visual Studio

4.10

Le nouveau compilateur permet de détecter plus d'erreurs dans votre code. Ce n'est pas forcément une bonne nouvelle pour les exportations! Bref, mieux vaut détecter les problèmes avant, même si cela fonctionnait souvent sans avoir besoin de rigueur supplémentaire.

Si vous utilisez l'assistant "Nouveau projet" dans Unreal Engine, vous êtes accompagnés dans le téléchargement et l'installation de Visual Studio 2015.



Si vous avez plusieurs versions de Visual Studio, vous pouvez sélectionner celle que vous voulez utiliser par section «Code source» dans «Préférences Editor».

Lors de la compilation de votre premier projet, soyez à l'affût de nouvelles erreurs de compilation signalées, le nouveau compilateur C++ est plus strict.

Unreal Engine 4 – Guide des mises à jour (4.20)

Si vous utilisez Visual Studio 2013, vous devrez compiler vous-même avec GitHub.

4.16

Visual Studio 2013 n'est plus pris en charge sur Windows avec cette version. Il faut Visual Studio 2015 ou Visual Studio 2017.

4.18

Il est possible d'utiliser Visual Studio Code pour écrire du code sur toutes les plates-formes UE4. Pour l'utiliser, sélectionner "Visual Studio Code" comme IDE préféré dans la boîte de dialogue des préférences de l'éditeur, ou ajouter l'argument `-vscode` lors de la génération de fichiers de projet sur la ligne de commande.

Toutes les plates-formes doivent également avoir installé les environnements d'exécution .NET Core 2.0, qui peuvent être obtenus à partir du [site Web Microsoft .NET Core](#).

Pour utiliser Visual Studio Code pour générer et déboguer tous les types de projets, certaines extensions supplémentaires sont requises. Sur toutes les plates-formes, assurez-vous que l'extension Microsoft C / C ++ et l'extension C # sont installées.

Sous Linux et Mac, l'extension "Mono Debug" est nécessaire pour déboguer les projets C #, et l'extension "LLDB Debugger" est nécessaire pour déboguer les projets C ++.

En outre, afin de déboguer des projets C #, le runtime mono doit être installé:

- On OS X: *brew install mono*
- Sous Linux: *sudo apt-get installation mono-complète*

4.20

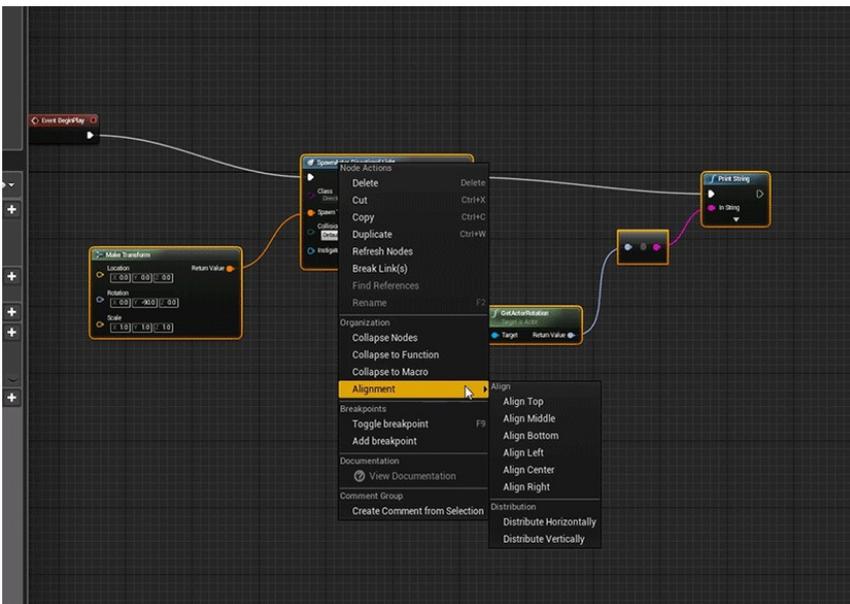
UE4 utilise maintenant le compilateur Visual Studio 2017 et le moteur génère des fichiers de projet pour Visual Studio 2017 par défaut. Il était temps non ? Visual Studio 2015 est toujours pris en charge, mais nécessite une certaine configuration. En outre, nous avons ajouté le support pour le SDK Windows 10.

Voir les [spécifications matérielles et logicielles](#) pour plus d'informations.

Automatic Blueprint Node Arrangement

4.10

Ah ben je l'attendais celle là! Ajout d'un ensemble complet de routines de mise en page pour l'organisation rapide et automatique de graphes de Blueprint. C'est pas trop tôt .



Cliquer à droite sur les nœuds sélectionnés et choisir "Alignement"

Unreal Engine 4 – Guide des mises à jour (4.20)

pour organiser automatiquement ces nœuds.

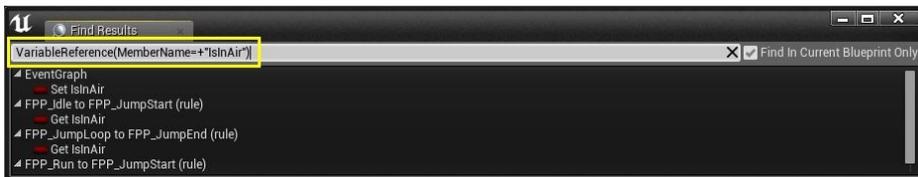
Voici ce que vous pouvez faire avec cette fonctionnalité:

- Alignement des nœuds sélectionnés en fonction de leurs limites (Haut / Moyen / Bas, Gauche / Centre / Droite).
- Distribuer les nœuds sélectionnés le long d'un axe horizontal / vertical.
- Remettre des connexions entre les nœuds sélectionnés.
- Remettre un lien spécifique relié à un particulier pin

Recherche de Blueprints

4.11

L'outil de recherche de Blueprints a été mis à jour pour permettre des recherches plus avancées (résultats plus ciblés).



On peut maintenant mettre un flag “Development only” sur les fonctions et faire gueuler le compilateur si on essaye de compiler un projet pour le shipping en utilisant ces fonctions.

Pas mal de bugs liés à l'éditeur ont été identifiés et corrigés dans cette release – il n'y a qu'à voir la longue série de notes à ce sujet dans le thread d'origine.

4.18

La fonctionnalité de recherche globale *Find in Blueprints* permet de trouver rapidement quelque chose dans plusieurs Blueprints, et ce, sans avoir à ouvrir préalablement une fenêtre de recherche Blueprint.

Unreal Engine 4 – Guide des mises à jour (4.20)



Rappelez vous quand vous faites une recherche dans un BP, et que vous étendez à tous les BP... vous êtes toujours dépendant de cette première fenêtre... si vous la fermez, vous perdez votre recherche.

Là, c'est bien une fenêtre de recherche globale qu'on peut appeler avec CTRL-MAJ-F (en mode BP Edition).

L'inspecteur de Pixels

4.12

Le nouvel outil Inspecteur de Pixel permet de cliquer sur un pixel dans la scène et de visualiser le matériel et de l'état de rendu qui a contribué à sa couleur.

Pour le lancer: "Developer Tools" dans la barre d'outils et cocher "Enable Pixel Inspector".

Blueprints en C++

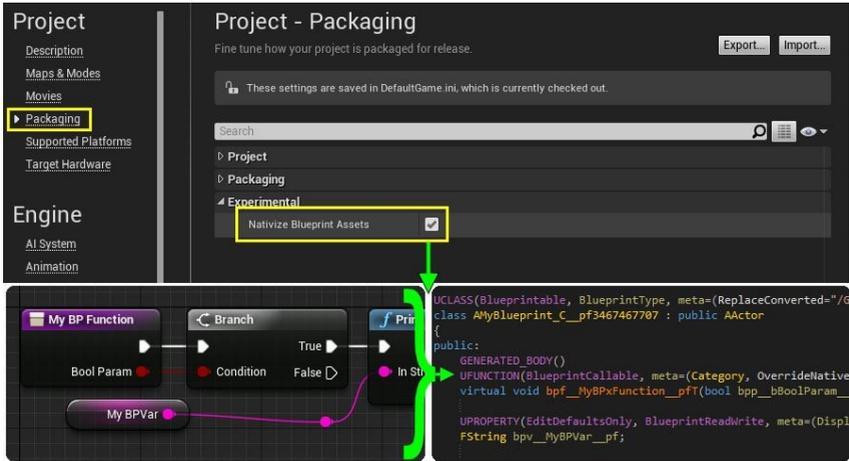
4.12

Pour améliorer l'exécution des Blueprints, une fonctionnalité permet de packager les Blueprints en code source natif.

Pour le permettre: Packaging Settings: Experimental => Nativize Blueprint Assets

Il faudrait faire des tests de performance pour voir s'il y a un changement significatif.

Unreal Engine 4 – Guide des mises à jour (4.20)



J'avais cru comprendre que les Blueprints étaient déjà convertis en C++, puis compilés pour le Build final de l'application.

Mais dans ce cas, à quoi sert cette fonction ?

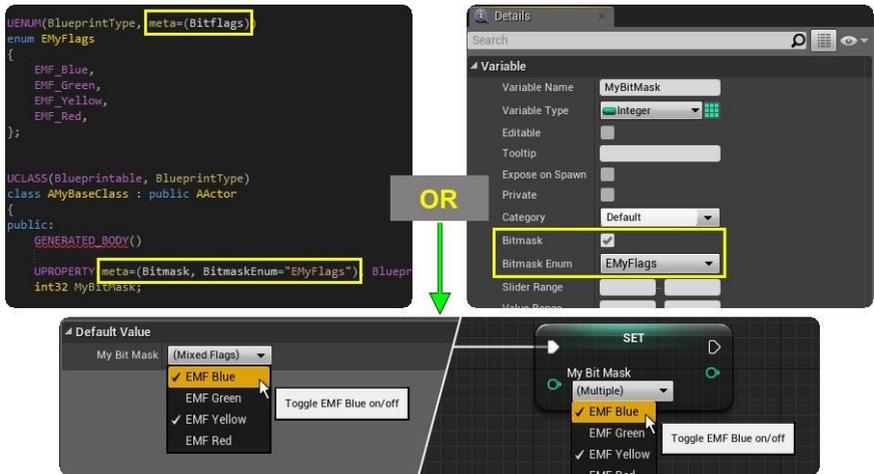
Integer vers Bitmask

4.12

Des entiers et des bitmask ! Si on coche bitmask sur un entier, on peut alors gérer individuellement les éléments – ça fonctionne autant en Blueprint qu'en C++.

Cela permet surtout de pouvoir plus facilement s'interfacer avec des API externes ou optimiser certaines choses avec les matériaux par exemple.

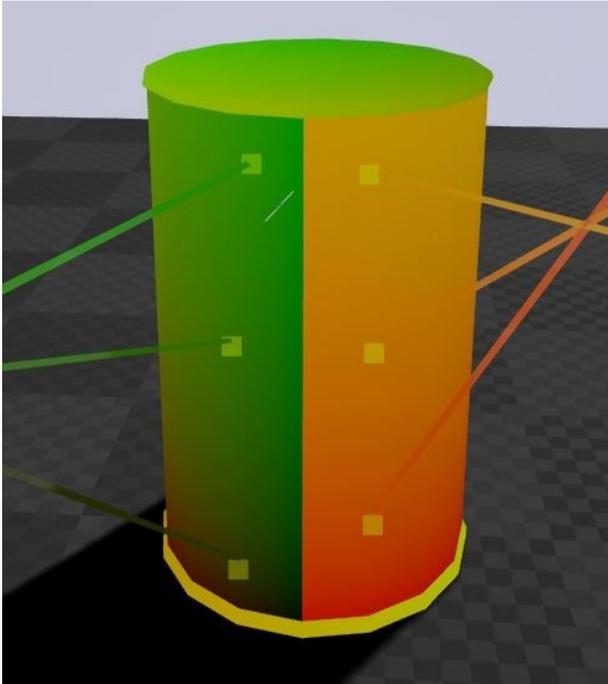
Unreal Engine 4 – Guide des mises à jour (4.20)



Nouveau: line trace renvoie les coordonnées de texture

4.13

Permet de renvoyer à partir d'un line trace, non seulement le point d'impact, mais aussi la coordonnées UV de la texture ! Je vois déjà plein d'applications pratiques, comme de simplifier le fait de pointer sur une mappemonde par exemple.



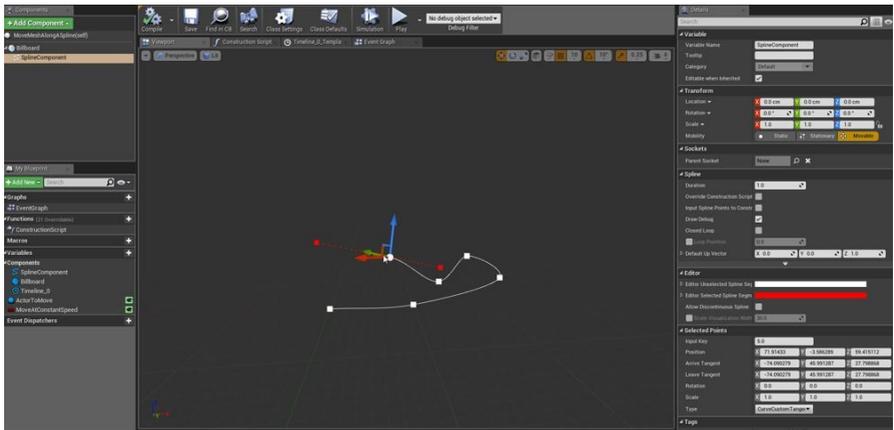
L'option est sous Paramètres du projet -> Physique -> Optimisations. Lorsque cette option est activée, vous pouvez utiliser la fonction "Find Collision UV" pour obtenir un Hit Result. L'activation de cette fonction fait utiliser de la mémoire supplémentaire, comme une copie de l'information UV doit être stocké dans la mémoire du CPU.

Amélioration de l'édition spline

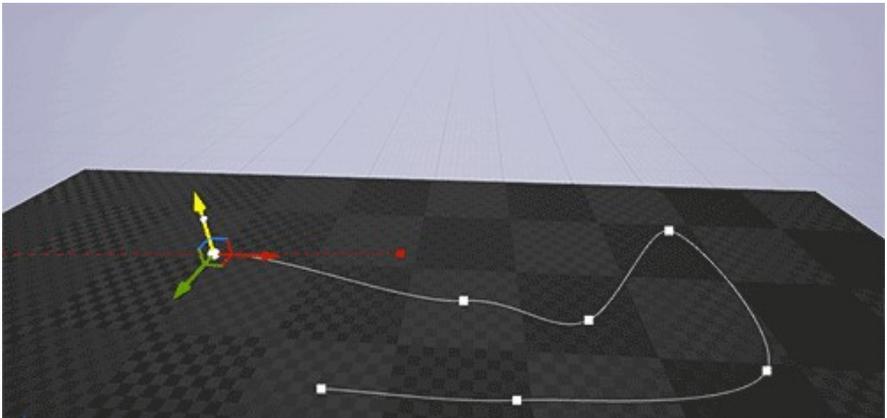
4.13

Il est possible de définir des points de composants Spline dans l'éditeur de Blueprint en utilisant le visualiseur standard. De nouvelles instances du Blueprint seront créées avec les valeurs par défaut, bien que celles-ci peuvent être substituées sur un niveau une fois placé.

Unreal Engine 4 – Guide des mises à jour (4.20)



Le "Reset to Default" établit une instance inverse vers le Blueprint par défaut. Toutes les modifications apportées au Blueprint défaut sont propagées à une instance dont les points spline n'ont pas été ultérieurement modifiés.



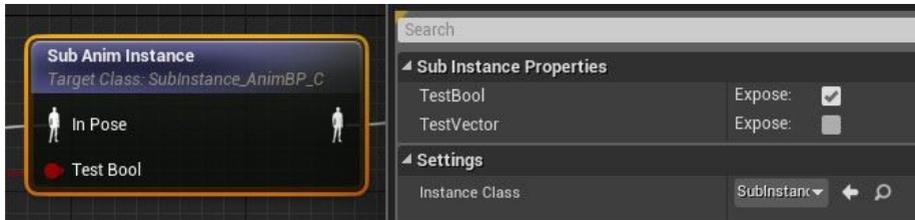
Sous-animation Blueprint

4.13

La logique d'animation peut être partagée à l'aide d'un noeud "Sub Anim instance" au sein de l'animation Blueprint pour une autre animation Blueprint Sub. Cela permet également de séparer les

Unreal Engine 4 – Guide des mises à jour (4.20)

grandes animations Blueprints dans des assets distincts en «locomotion» et «parties de la physique».



Amélioration de l'accès pour les scripts aux Static Meshes pour les LOD et les collisions

4.20

Les scripts Blueprint et Python que vous exécutez dans l'éditeur Unreal peuvent désormais modifier davantage de propriétés de vos ressources Static Mesh. Cela vous permet d'automatiser certains des outils offerts par l'interface utilisateur de l'éditeur de maillage statique.

Vous pouvez désormais **générer automatiquement des niveaux de détail (LOD) pour votre géométrie**, ce qui augmente les performances de rendu de la scène en utilisant progressivement des versions moins détaillées de la géométrie lorsque la distance entre le point de vue de la caméra et la géométrie augmente. [Voir Création de niveaux de détail dans Blueprints et Python](#)

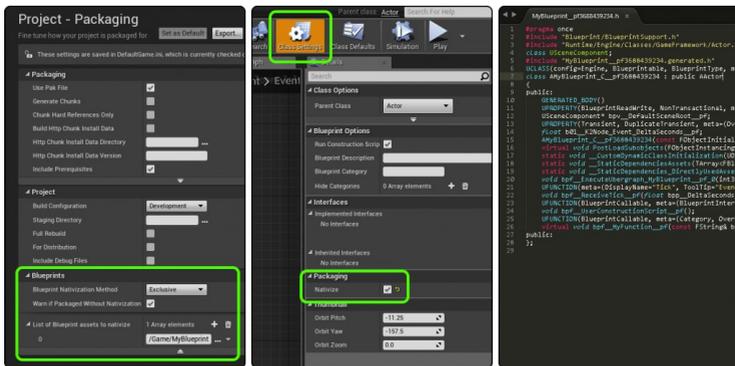
Vous pouvez maintenant générer automatiquement des maillages de collision qui représentent vos assets Static Mesh dans la simulation physique. Voir [Configuration des collisions avec des maillages statiques dans Blueprints et Python](#)

Encore de chouettes améliorations permettant de travailler en procédural.

Cooking blueprints en C++

4.15

Les blueprints sont packagés en code source natif. Fonction jadis expérimentale dont nous avons parlé mais qui est aujourd'hui passé en mode production. De fait, programmer en Blueprint ou C++ ne devrait pas avoir d'impact... Reste que quelques fois, il est plus facile de programmer directement en C++, mais cela n'enlève rien au fait que les Blueprints sont plus faciles à comprendre et à maintenir dans le temps. Voir la [notice](#).

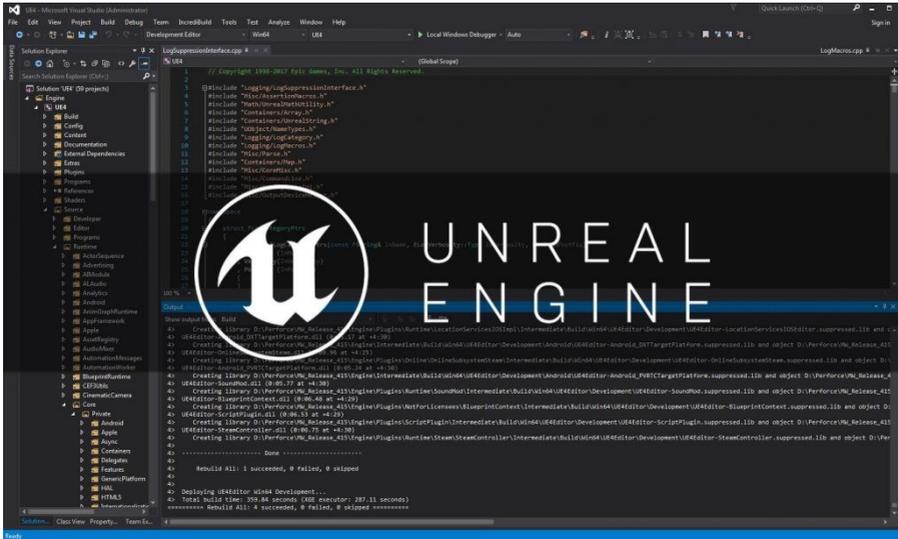


Compilateur C++ plus rapide

4.15

Unreal Engine 4 codebase compile désormais jusqu'à 50% plus rapidement. Euh oui, mais pourquoi ??? C'est sorti du chapeau avec le lapin qui court plus vite ? Non, c'est ce qui a été permis grâce à de nouvelles conventions.

Unreal Engine 4 – Guide des mises à jour (4.20)



Les conventions pour le nouveau modèle sont les suivantes:

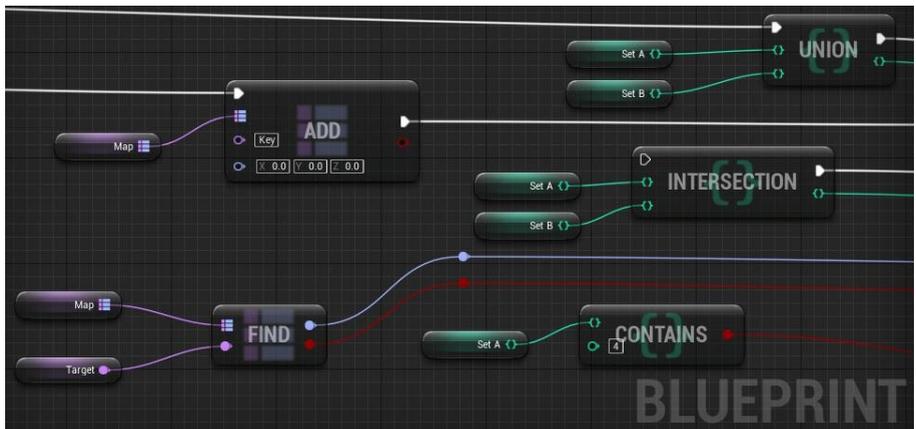
1. Chaque tête comprend maintenant tout ce qu'il faut pour compiler - un en-tête de CoreMinimal.h contient un ensemble de types ubiquitaires de base (par exemple FString, FName, TArray, FVector, etc)
2. Chaque fichier .cpp comprend son fichier .h pour valider que chaque tête contient tout ce qu'il faut. UnrealBuildTool vérifiera que les fichiers source adhèrent à cette convention.
3. Aucun code de moteur ne comprend un en-tête monolithique, comme Engine.h ou UnrealEd.h plus - Ils existent encore pour la compatibilité avec les projets de jeu et ne produisent pas des avertissements quand ils sont déjà inclus.
4. Aucun code de moteur inclut un en-tête pré-compilé.

Voilà, rien de magique, juste du savoir-faire et un peu de logique. C'est vrai que le code source du moteur n'est pas évident à maîtriser pour le profane - y-a pas mal de macros et cela ne respecte pas toujours les conventions C/C++. Mais cela progresse aussi de ce côté, même le code source progresse en maturité et les résultats sont là: une compilation beaucoup plus rapide !

Stockage de packs d'objets en Blueprint

4.15

Des collections d'objets peuvent maintenant être stockées dans les Blueprints, et sans forcément passer par des tableaux ou des datatables. Il s'agit des **Maps** et de **Sets**. [On en avait déjà parlé](#) il me semble, alors c'est peut-être que c'est passé en mode production maintenant, je ne sais plus. Ces structures de données garantissent qu'ils ne contiennent que des articles uniques, **sans doublon**; Ils sont optimisés en ce sens.



La différence entre map et set ? Dans un **set**, il n'y a pas vraiment de notion d'ordre. En contre-partie, c'est plus léger et plus rapide d'accès qu'une map.

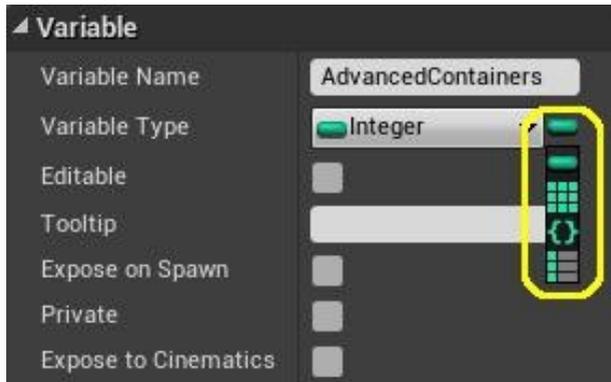
La **Map** non plus remarquez, mais il y a une clé d'accès qui garanti un accès très rapide également !

Unreal Engine 4 – Guide des mises à jour (4.20)



La nouveauté, c'est surtout le contrôle de type au niveau de la variable qui est maintenant dans un menu déroulant, vous permettant de sélectionner 'Single Variable', 'Array', 'Set', ou 'Map'. Lorsque «carte» est sélectionnée, un deuxième menu déroulant pour le type «valeur» apparaît.

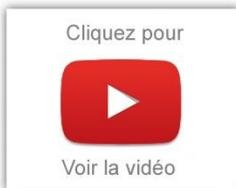
Unreal Engine 4 – Guide des mises à jour (4.20)



De même, les variables de type Map et Set peuvent maintenant être exposés du C++ vers les Blueprints.

Array Reordering

4.18



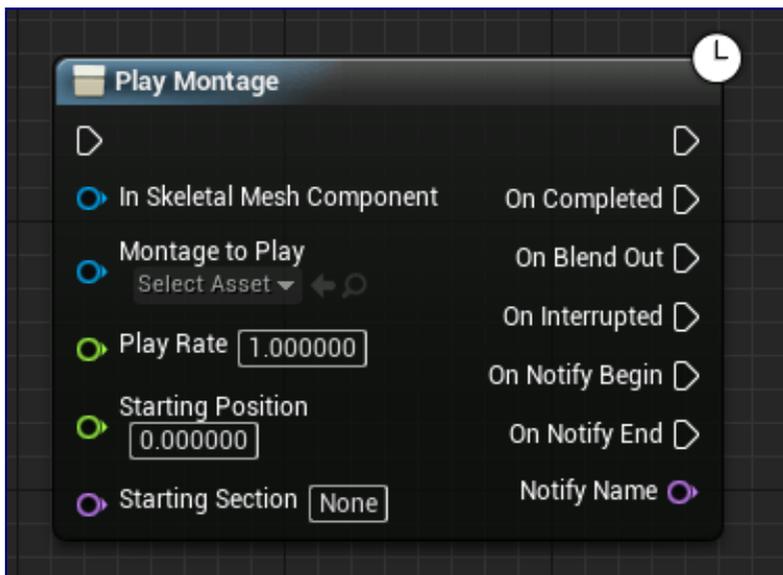
Glisser et déposer des éléments pour les réorganiser dans un tableau: Toutes les arrays configurées via Blueprints et C ++ sont réordonnables par défaut.

Pour désactiver le comportement par défaut et créer un tableau d'ordre fixe en C ++, utiliser les métadonnées `EditFixedOrder`.

Play montage

4.16

Nouveau nœud asynchrone qui peut être utilisé dans tout Blueprint pour jouer des montages d'animations. Il offre un accès facile à certains événements de rappel, vous permet de déclencher d' autres nœuds quand un montage se fond sur, est interrompu, etc ...



- *OnCompleted* est appelé quand le Montage finit de jouer et le mélange fait.
- *OnBlendOut* est appelé quand le montage commence à mélanger, que ce soit parce qu'il est arrêté automatiquement ou manuellement.
- *OnInterrupted* est appelé si le montage commence à Mélanger, mais parce qu'il a été interrompu par un autre Montage.
- *OnNotifyBegin* et *OnNotifyEnd* sont des callbacks utilisant 'Play Montage Notify' ou 'Play Montage Notify Window' Anim Avertit dans l'actif de Montage. Ces AnimNotifies peuvent

Unreal Engine 4 – Guide des mises à jour (4.20)

renvoyer un «Notify Name» de faire la différence entre plusieurs callbacks du même Montage.

Analyse statique avec PVS-Studio

4.17

UE4 prend désormais en charge l'analyseur statique **PVS-Studio** (disponible séparément).

Pour le permettre, ajoutez `-StaticAnalyzer = PVSSstudio` à la ligne de commande `UnrealBuildTool`. C'est assez intéressant si vous souhaitez améliorer votre code, repérer les séquences inutiles, voir des erreurs directement grâce à cet analyseur statique.

Blueprint Compilation Manager

4.17

Il réduit le temps de compilation de blueprints de 40 à 50%! Voilà, c'est tout, sortez, y-a plus rien à voir ! Voilà, on ne sait pas forcément pourquoi, mais cela a été amélioré. Remarque: Vous pouvez désactiver le nouveau gestionnaire de compilation dans les paramètres du projet s'il provoque des problèmes avec un projet existant.

Registre des assets dans Blueprints

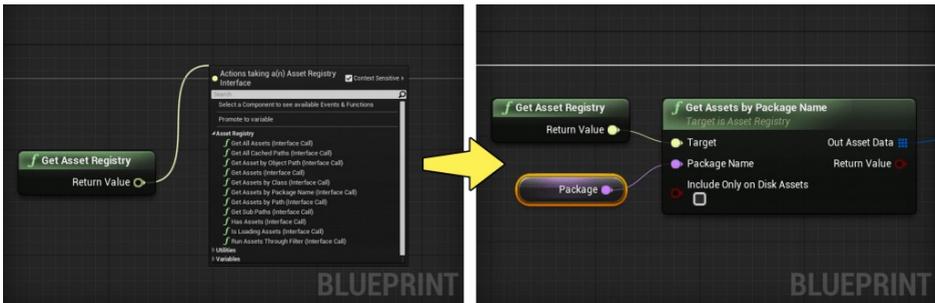
4.17

Le Registre des assets peut maintenant être utilisé dans Blueprints pour trouver rapidement des informations sur les actifs!

Signets Blueprint

4.20

La fonctionnalité Blueprint Bookmarks permet de créer des signets dans le Blueprint Editor. Les signets créés sont listés dans une nouvelle fenêtre de l'interface utilisateur, vous pourrez cliquer dessus pour restaurer la position et le niveau de zoom de la fenêtre. En plus des signets que vous créez, vous pouvez également accéder rapidement à n'importe quel nœud Commentaire de votre Blueprint en le sélectionnant dans une liste distincte. Les signets sont stockés localement sur votre machine, de sorte qu'ils n'affectent pas les Blueprints eux-mêmes, et la synchronisation du contenu ne remplacera pas vos signets avec ceux d'un autre utilisateur.



Blueprint Property Accessors

4.17

Les variables natives peuvent maintenant utiliser le balisage UProperty pour être utilisées en Blueprints via UFunction accessors, même si la propriété avait été exposée directement à l'aide BlueprintReadOnly ou balisage BlueprintReadWrite.

```
private:
    UPROPERTY(EditAnywhere, BlueprintSetter=ApplySkin)
    UCharacterSkinDescription* ActiveSkin;

public:
    UFUNCTION(BlueprintSetter)
    void ApplySkin(UCharacterSkinDescription* InSkin)
    {
        if (InSkin != ActiveSkin)
        {
            ActiveSkin = InSkin;

            USkeletalMeshComponent* MyMeshComponent = GetMesh();
            MyMeshComponent->SetSkeletalMesh(ActiveSkin ? ActiveSkin->Mesh : nullptr);

            MyMeshComponent->OverrideMaterials.Reset();
            if (ActiveSkin != nullptr)
            {
                for (int32 Index = 0; Index < ActiveSkin->Materials.Num(); ++Index)
                {
                    MyMeshComponent->SetMaterial(Index, ActiveSkin->Materials[Index]);
                }
            }
        }
    }
}
```

Nouveau: Orphan Pins

4.17

Une nouvelle fonctionnalité "Orphan Pin" a été ajoutée pour éviter des bugs mystérieux comme le résultat du retrait de pins!



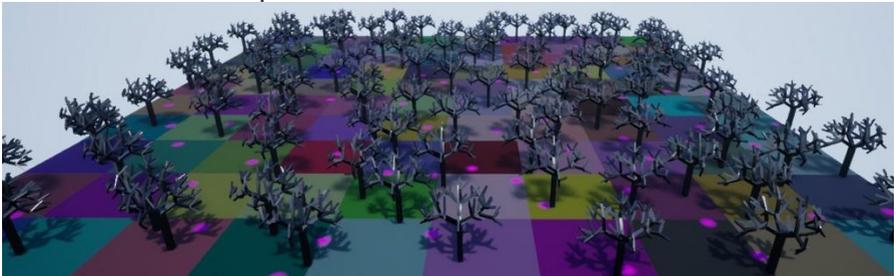
Chaque fois qu'un pin connecté, en cours d'utilisation, est retiré d'un nœud, plutôt que de la faire disparaître tranquillement, brisant toutes les connexions, elle sera désormais retenue comme orpheline, et une erreur ou d'un avertissement sera émis.

Nouveau: Sobol Blueprint and Material Nodes

4.17

Plusieurs nœuds ont été ajoutés pour générer des emplacements de points en utilisant la séquence **quasi-aléatoire Sobol**.

La séquence Sobol peut générer des points avec un grand nombre de dimensions indépendantes. Les nœuds Blueprints Random Sobol Float donnent la valeur d'une seule dimension. Ceux-ci peuvent être combinés ensemble pour créer des distributions 2D ou 3D.

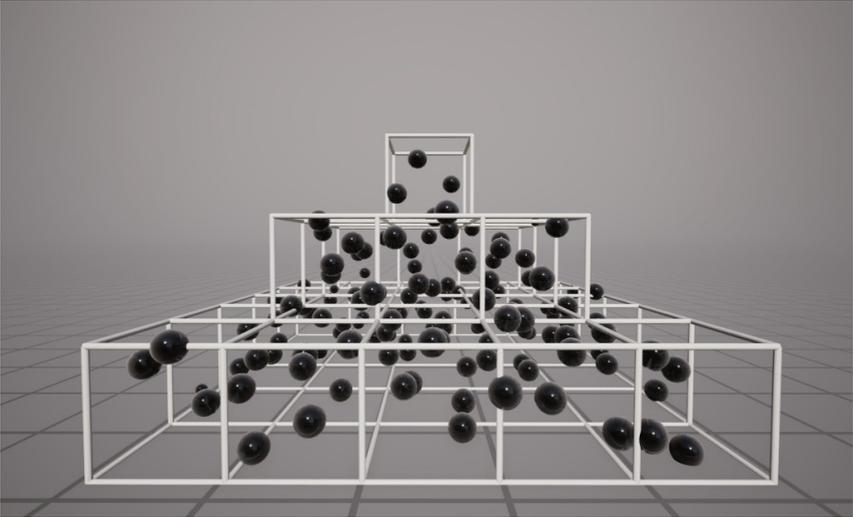


Les nœuds Blueprints Random Sobol Cell 2D placent des points 2D Sobol dans les cellules d'une grille 2D. Cela permet de contrôler la densité en plaçant un nombre différent de points dans chaque cellule. Le nœud Sobol de matériau fournit les mêmes emplacements de points 2D que le nœud Random Sobol 2D avec une grille de 256x256 cellules fixe.

Cet exemple place un nombre variable d'arbres dans chaque cellule, avec un matériau sur le terrain qui montre les cellules et dessine un point sous le premier arbre placé dans chaque cellule.

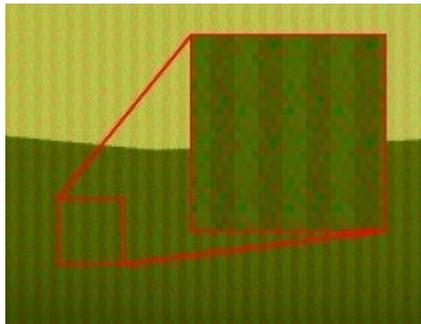
Le nœud Blueprint Random Sobol Cell 3D fait la même chose pour les points en 3D dans une grille 3D. Cet exemple met quatre sphères dans chaque cellule à l'intérieur de la pyramide.

Unreal Engine 4 – Guide des mises à jour (4.20)

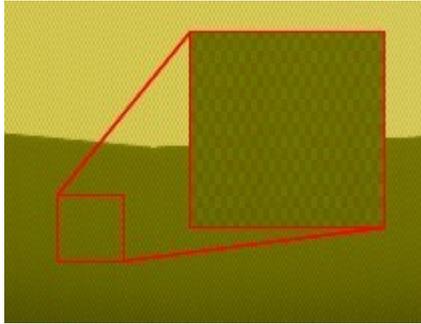


Le noeud Temporal Sobol utilise une autre séquence 2D Sobol au niveau de chaque pixel, cyclant à travers un autre ensemble de points de chaque trame.

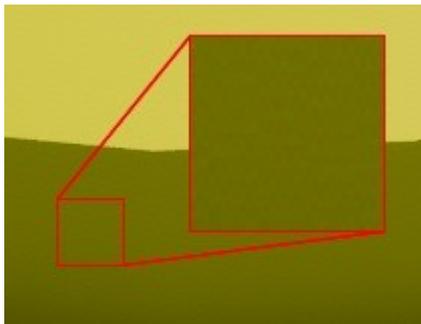
Cela peut être utile pour les effets faits pour se fondre dans plusieurs cadres avec anticrénelage temporel



1 échantillon par pixel, sans TAA



4 échantillons par pixel, sans TAA



4 échantillons par pixel, avec TAA

Texture-based Importance Sampling Blueprint nodes

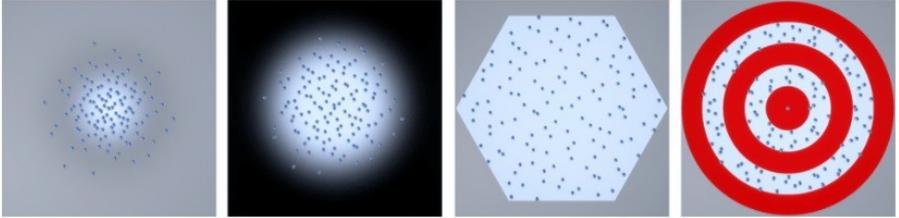
4.17

Importance sampling place des points d'échantillonnage selon une fonction de densité donnée. Des nœuds Blueprints ont été ajoutés pour permettre le placement flexible en 2D, utilisant une texture. Le Blueprint Make Importance Texture traite la texture pour l'utilisation et le Blueprint Importance Sample donne les emplacements des points entraînés par la densité prévue.

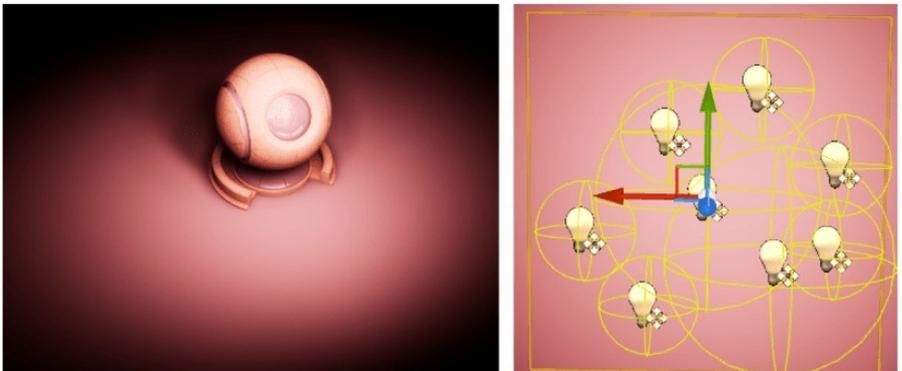
Voici quatre exemples de placement de sphères en fonction de la luminance de quatre textures différentes.

Unreal Engine 4 – Guide des mises à jour (4.20)

- La première varie en douceur, dense au centre, moins dense vers l'extérieur.
- Le second est uniforme dans le centre et moins dense le long du bord.
- Le troisième est totalement uniforme dans l'hexagone.
- Le quatrième est dense dans les anneaux blancs et moins dense dans les anneaux rouges.



Ci-dessous, huit sources lumineuses sont placées en fonction de la texture cible rouge et blanc. La taille et la couleur de la source est proportionnelle à la densité locale.



PAYSAGE / FOLIAGE / GÉNÉRATION PROCÉDURALE

Realistic Foliage Lighting

4.7

Le nouveau système permet à la lumière d'être transmise au travers de l'herbe, des feuilles, du papier ou tout autre matériau.



L'éclairage diffus sur le côté opposé de la surface devient un éclairage transmissif sur le côté ombré:

- L'éclairage transmissif est un mélange de retour de lumière diffuse dépendant de la vue pour la rétrodiffusion
- La couleur de subsurface filtre cet éclairage et peut être utilisée pour représenter une épaisseur variable ainsi (par exemple, les nervures des feuilles)
- Notez que l'utilisation d'opacité avec ce modèle d'ombrage n'affectant que la distance de atténuation de l'ombre

Bref, un algorithme complexe, mais optimisé pour le temps-réel et qui donne des visuels impressionnants.

Pour des mondes gigantesques (Open Worlds)

4.7

Le système de foliage system a été optimisé pour supporter de larges Open Worlds.



Le moteur utilise désormais un système hiérarchique pour gérer et effectuer le culling qui peut aller jusqu'à plusieurs millions d'instances! Aussi, une transition en douceur se fait entre les niveaux de détail (LOD) avec un fondu temporel pour éliminer le bruit !

Rendu de l'herbe pour le paysage

4.8

Nouveau système pour l'herbe et autres couvertures de sol avec un **rendu dynamique sur le paysage**.

Chaque type d'herbe est décrit par un nouvel asset Landscape Grass Type, et chaque type d'herbe peut spécifier plusieurs maillages statiques pour permettre la variation.

Unreal Engine 4 – Guide des mises à jour (4.20)



- Quand le joueur se déplace à travers le monde, de nouvelles instances sont ajoutées, et les instances qui ne sont plus utilisées sont supprimées (du clipping quoi...)
- Modifier la densité de l'herbe pour adapter la mise en place de l'herbe pour correspondre à la texturation de terrain
- Les instances herbe sont générées en tâche de fond selon la carte de densité de l'herbe.
- L'herbe est efficacement rendue par un système d'instanciation hiérarchique. Les transitions LOD sont effectuées pixel par pixel et assez transparentes .

Une carte de la densité de l'herbe est générée à partir du matériau de la paysage.

Créer des règles de placement plus complexes telles que la suppression automatique de l'herbe sur la base normale de surface ou de l'altitude ou de l'utilisation de plusieurs couches de paysage.

Générateur procédural de Feuillage (expérimental) et Refonte

4.8

Alors ça c'est top: Fonctionnalité expérimentale pour générer d'énormes (ou minuscules, ou moyennes) régions de feuillage rapidement.



- Type de feuillage actif: propriété pour personnaliser la façon dont ils sont dispersés et leur taille.
- Différents types de feuillage pour simuler toute une forêt!

Amélioration de la convivialité, possibilité de travailler plus facilement avec un grand nombre de types de feuillage dans la palette à la fois.

Unreal Engine 4 – Guide des mises à jour (4.20)

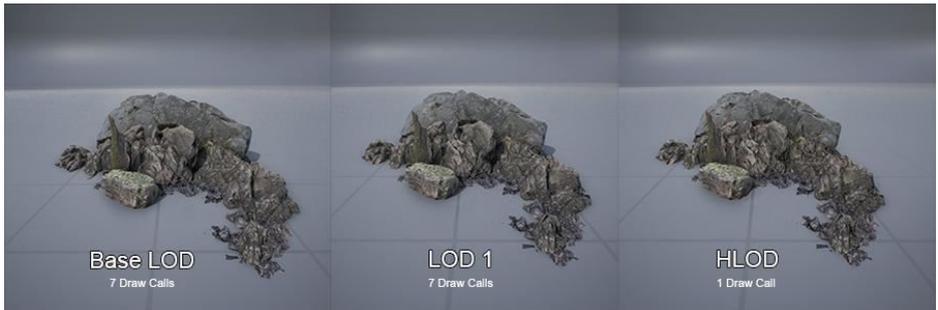


- Créer facilement de nouveaux types de feuillage à partir de la vue du feuillage.
- Les vignettes de feuillage ont des boutons pour modifier la visibilité et la sauvegarde de l'actif, ainsi que le nombre d'instances au total.
- Activer ou désactiver les types de feuillage multiples à la fois avec la multi-sélection.

Niveau hiérarchique du Système LOD

4.8

Fonctionnalité de LOD hiérarchique qui crée automatiquement des LODs pour des groupes de meshes. **Réduit considérablement les besoins en dessins et le nombre de polygones quand ils sont visibles de loin.** Idéal pour les jeux en monde ouvert!



- Les groupes de meshes à combiner sont trouvés automatiquement. Si Simplygon est disponible, il sera utilisé pour générer le maillage bas-détail pour un groupe. Sinon, la version LOD de chaque mesh sera combinée.
- On peut retirer des personnages en désactivant leur option Auto LOD génération.

Procedural Mesh Component (expérimental)

4.8

Pour créer des maillages à l'exécution à l'aide de C++ ou des Blueprints.



Multiples sections (matériaux), soutien de collision, possibilité de spécifier normales, tangentes, UV et couleurs par vertex.

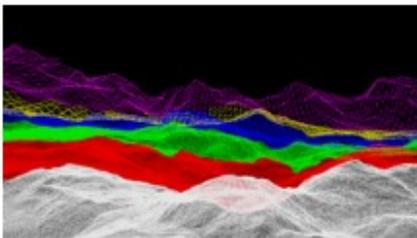
Bibliothèque d'utilitaires et une fonction pour générer les normales lisses et tangentes avec les positions maillage de vertex et UV. j'en parle un peu plus dans cet article.

Reste à voir quels modèles on peut utiliser à la base maintenant que l'outil est en phase XP.

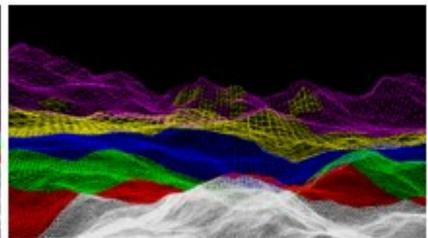
LOD basé sur la taille de l'écran

4.19

Créez des panoramas à couper le souffle dans les grands mondes ouverts grâce aux optimisations du rendu Paysage. Le rendu des paysages a été optimisé. Le niveau de détail (LOD) n'est plus basé sur la distance, mais sur la taille à l'écran pour déterminer les détails d'un composant, comme le fonctionnement du système LOD Stell Mesh.



Ancienne approche



Nouvelle Approche

Vous pouvez visualiser les LOD de paysage en allant dans Viewport> Visualizers> LOD.

PROXY LOD

Expérimental

4.19

Un nouveau plugin (expérimental et uniquement sous Windows) de production de LOD Low Poly est intégré, il permet de produire des modèles avec peu de polygone, avec des matériaux précalculés pour chaque niveau de LOD. Cela offre une alternative "intégrée" à Simplygon dont nous avons déjà parlé.

Pour activer l'outil, sélectionner "Hierarchical LOD Mesh Simplification" sous Editeur dans la boîte de dialogue Paramètres du projet. À partir du plugin LOD Mesh Reduction Plugin, sélectionner le nouveau module "ProxyLODMeshReduction". Après le redémarrage de l'éditeur invité, le plugin remplacera l'outil Simplygon tiers pour les LOD de fusion de maillage statiques. Ce nouveau plugin est accessible de deux façons: le HLOD Outliner et la boîte de dialogue Merge Actors.



Source Geometry: 11 geometry pieces, 38,360 tris, and 27,194 verts | Result Geometry Proxy: 1 geometry piece, 8,095 tris, and 5,663 verts

Proxy LOD prêt pour la production

4.20

Amélioration des performances en réduisant les coûts de rendu dus au nombre de polygones, aux appels et à la complexité du matériel. Cet outil fournit **une alternative au package tiers Simplygon** dont nous avons déjà parlé précédemment.

L'outil Proxy LOD produit une représentation plus simple en créant un proxy sous la forme d'un seul maillage paramétré low-poly et de textures associées. Ce proxy peut ensuite être affiché au moment de l'exécution lorsqu'une réduction de la qualité du modèle est acceptable - par exemple, lorsque la géométrie n'occupe qu'un petit nombre de pixels à l'écran.

Remarque: L'outil Proxy LOD n'est actuellement disponible que sous Windows.



L'image montre les bâtiments et les parcs utilisant l'outil Proxy LOD.

Vous avez, par rapport à la version d'essai 4.19, un meilleur contrôle sur les normales sur la géométrie de proxy et la possibilité de générer des mandataires beaucoup plus simples en utilisant le remplissage d'espace pour fermer automatiquement les portes et les fenêtres.

Comblers les "vides"

4.20

Avec la géométrie "fermée", le système Proxy élimine automatiquement toutes les structures inaccessibles (par exemple, les murs intérieurs ou les meubles dans une maison fermée).

Pour des résultats idéaux, la géométrie de la source devrait être construite ou modifiée dans cette optique, mais en raison des contraintes de production du jeu, ce n'est pas toujours possible.

Pour faciliter la génération de LOD Proxy efficace, l'outil Proxy LOD peut utiliser les techniques de dilatation et d'érosion basées sur les niveaux pour fermer les espaces. Le cas d'utilisation prévu est principalement pour des portes et des fenêtres dans des bâtiments éloignés.



1 - mesh originale; 2 - Pas de comblement ; 3 – Remplissage

Unreal Engine 4 – Guide des mises à jour (4.20)

L'image ci-dessus montre l'effet de l'utilisation de Gap Filling. Toutes les images ont été contraintes d'utiliser une petite quantité fixe d'espace de texture.

L'image 2 est le résultat du proxy LOD sur un bâtiment sans utiliser Gap Filling, auquel cas le LOD inclut l'intérieur du bâtiment (au prix de triangles et de texels invisibles).

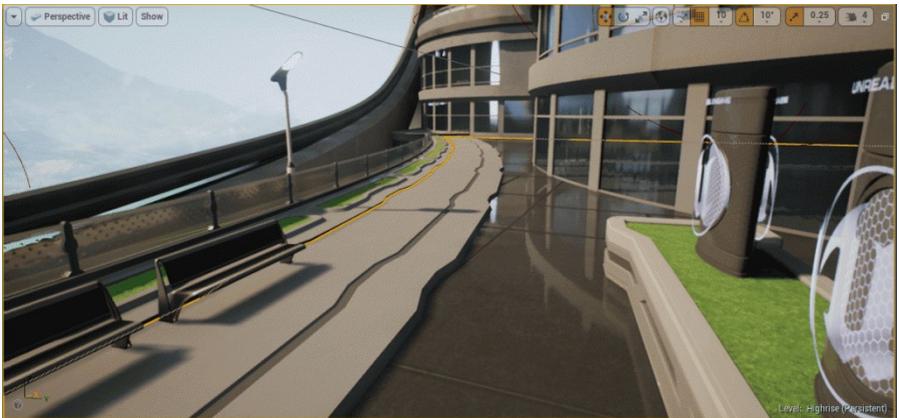
L'image 3 est le même bâtiment avec Gap Filling pour fermer automatiquement les portes et les fenêtres des bâtiments --> moins de triangles totaux et une meilleure utilisation de la ressource de texture limitée.

LOD hiérarchique - Prévisualisation rapide et mise en cluster

4.9

Hierarchical Level-of-Detail (HLOD) permet d'avoir de bons détails sur certaines parties lors d'une simplification via LOD.

La prévisualisation rapide (Fast HLOD Preview) vous permet de **voir seulement les parties désirées sans calculer l'ensemble**, les limites sont fonction du réglage Draw Distance dans les paramètres du système LOD.



Contrôle Normal Amélioré: Fendu Hard Edge Normal

4.20

Le développement de Fortnite a nécessité une utilisation efficace des LOD. Pour la plupart des proxies, de très petites textures de couleur de base sont générées et aucune normal map n'est utilisée, cette approche nécessite les normales de qualité les plus élevées possibles sur le maillage proxy lui-même.



1 - Angle du bord dur = 80; 2 - Angle du bord dur = 0

Unreal Engine 4 – Guide des mises à jour (4.20)

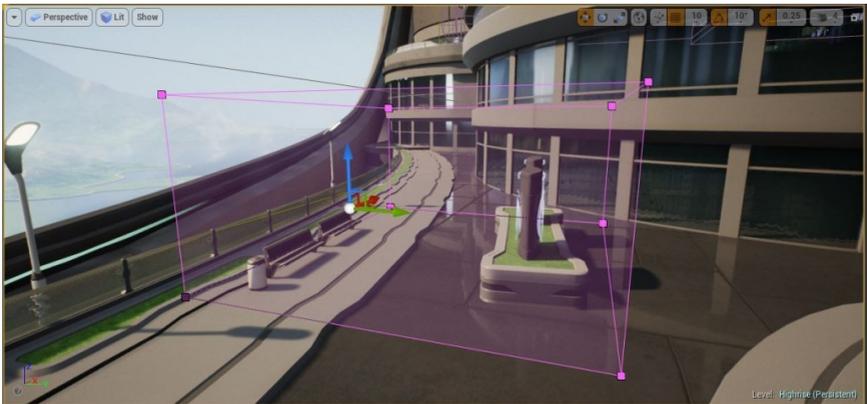
L'image ci-dessus montre l'effet de la division hard-edge pour les normales de vertex. L'image 2 montre des normales de sommet lisse, comme dans la version 4.19 expérimentale du plugin - les régions sombres près du fond de la maison sont indicatives des lacunes. L'image 1 montre la division normale du sommet du bord dur avec un angle de coupure du bord dur fourni par l'utilisateur.

En plus de l'angle de coupure du bord dur, l'utilisateur peut maintenant spécifier la méthode utilisée pour calculer le vertex normal, en sélectionnant **Angle pondéré**, **Pondéré par zone** et **Pondéré à parts égales**.

LOD Clustering personnalisable (HLOD Volumes)

4.9

Vous pouvez maintenant placer des volumes Hierarchical LOD qui définiront un cluster pour les acteurs qu'il encapsule. Utilisez cette option pour définir manuellement les domaines et les objets qui devraient être regroupés dans un niveau.



Landscape Mirror Tool

4.10

Grâce au nouveau “Mirror Tool“, vous pouvez facilement faire des **paysages symétriques**.



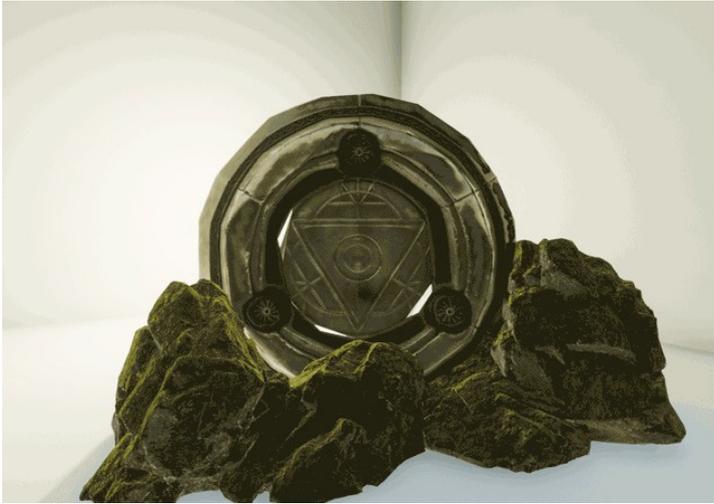
Doublez les heightmaps, material layers et le paysage, herbe. Doublez dans les deux sens sur l’axe X ou Y.

Dithered LOD Crossfades

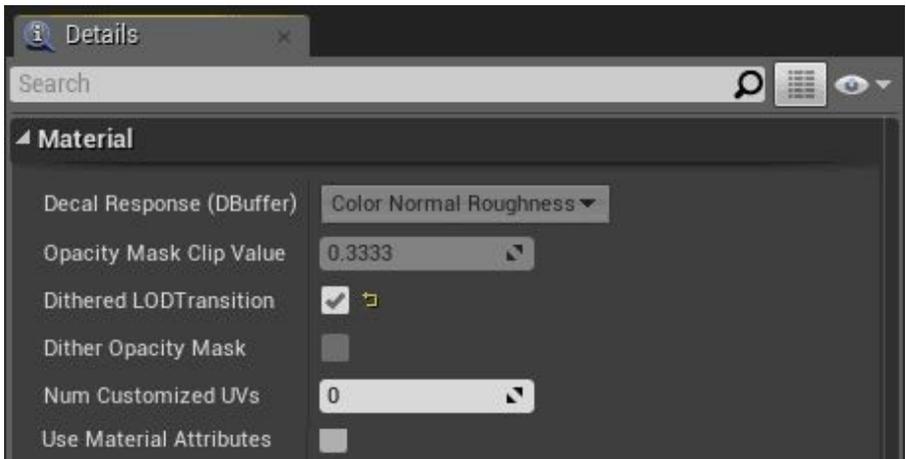
4.11

Les static meshes sont maintenant dotés d’un **morphing de type “crossfade”** lors de leur passage d’un niveau de Lod à un autre.

Unreal Engine 4 – Guide des mises à jour (4.20)



Cette fonction doit être activée sur le matériau car il y a un petit coût de performance. Mais cela peut s'avérer très sympathique sur des objets sur lesquels le joueur a tendance à fixer son regard (bâtiment au loin à atteindre, etc.).



Improved Hierarchical LOD (HLOD)

4.11

Le HLOD peut **remplacer automatiquement beaucoup de meshes détaillées par quelques meshes simples quand ils sont au loin**. Cela permet d'atteindre des niveaux de qualité beaucoup plus élevés lors de l'affichage des objets de près, et une plus grande rapidité.



L'exemple est celui de l'Agora de Paragon: 2.56 Million triangles, 5690 draw calls (à partir de 3.94 million, 7060 draw calls)

Afin de tirer le meilleur profit de HLOD, vous aurez besoin du [SDK Simplygon](#) (qui va faire la simplification en automatique). Il peut être utilisé gratuitement dans tout projet commercial (ou non bien sûr) dont le "budget" ne dépasse pas 1M€.

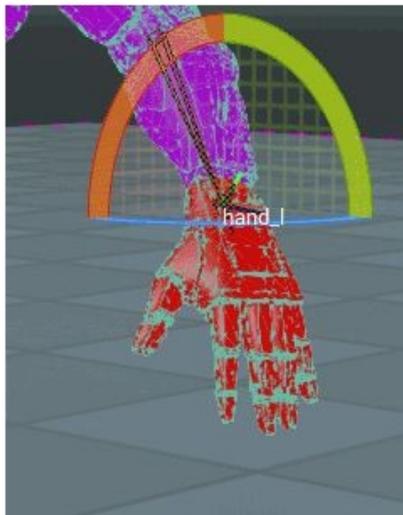
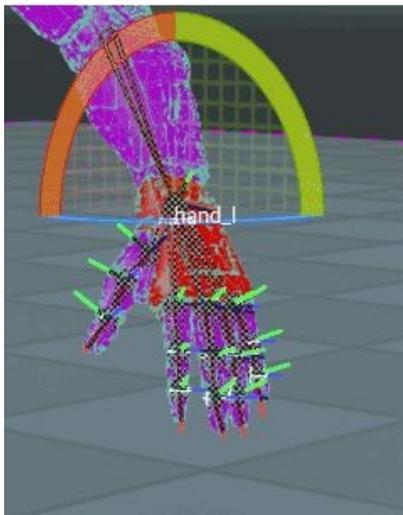
Sinon, vous pouvez aussi faire vos modèles vous-mêmes sous votre logiciel préféré (Blender, Maya, etc.) et créer vos LODs... mais coté workflow, c'est quand même mieux de passer par ce nouveau système.

LOD et réduction de Bones !

4.11

Oui, vous l'avez compris, vous avez certainement vu l'image ci-dessous qui parle pour elle-même!

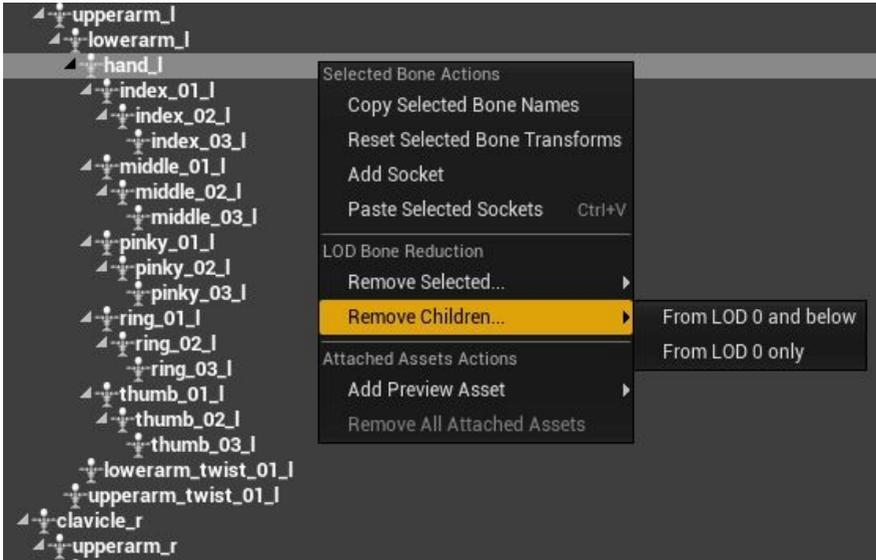
Le LOD intègre la notion de **réduction des bones** (comme pour les doigts de la main), **tout en conservant le weightpainting!**



Nous allons pouvoir encore plus travailler salement et UE4 va gommer notre souillonnerie!

Non, désolé, cela m'a échappé... je me dis juste que les optimisations semi-automatiques comme celle-ci sont vraiment impressionnantes, et qu'au final, les graphistes auront de moins en moins besoin de connaissances "métier" et pourront se focaliser sur la création artistique indépendamment de la machine cible – c'est mieux dit hein comme ça... plus politiquement correct ?

Utilisez la nouvelle option 'Remove Children' dans l'arbre du squelette. C'est vraiment trop simple:



Herbe et feuillage évolutifs

4.12

Un curseur permet de régler les niveaux de détails.



Ceci par l'option "Enable Density Scaling" dans la section Scalability

de FoliageType.

La mise à l'échelle de densité à appliquer à l'herbe et au feuillage à chaque niveau est configuré avec le "grass.DensityScale" et "foliage.DensityScale" dans les sections [FoliageQuality] de BaseScalability.ini et peut être redéfinie dans DefaultScalability.ini de votre jeu.

Collisions

4.12

Le paysage prend en charge la collision simplifiée du joueur.

La "Simple Collision Mip Level setting" vous permet de choisir le niveau LOD de paysage à utiliser pour la collision simplifiée.

La valeur Collision Mip Level est toujours utilisée pour spécifier le niveau LOD du paysage à utiliser pour une collision détaillée.

La collision peut maintenant être générée pour les Spline meshes créés lors de l'exécution.

Selective LOD pour collision mesh

4.12

Possibilité de choisir le LOD d'un Static Mesh à utiliser pour une collision «complexe» (per-poly).

Cela permet d'importer ou de générer une représentation de collision plus simple pour un maillage, pour réduire l'utilisation de mémoire sur des mesh high-poly.



Génération automatique de LOD

4.14

Unreal Engine réduit maintenant automatiquement le nombre de polygones de vos maillages statiques pour créer LODs.



l'animation ci-dessus montre cinq LODs qui ont été générés automatiquement. Chacun d'eux est la moitié du nombre de triangles comme le précédent.

Ce simplificateur de maille maintient les UV et donc **les mêmes matériaux** peuvent être utilisés, et les LOD peuvent partager la même lightmap.

Auto Compute LOD Distances

4.14

paramètre important parce que l'algorithme connaît la différence visuelle maxi possible pour déterminer le niveau d'erreur acceptable. Il va donc automatiquement calculer la taille de l'écran à utiliser pour chaque LOD.

Outil de rotation de paysages

4.14

L'outil miroir de paysage (landscape mirror tool) peut maintenant inverser la géométrie parallèlement au plan de miroir, pour créer des cartes multijoueur diagonalement opposées.



MENUS ET WIDETS (UMG)

UMG / Slate Clipping 2.0

4.17

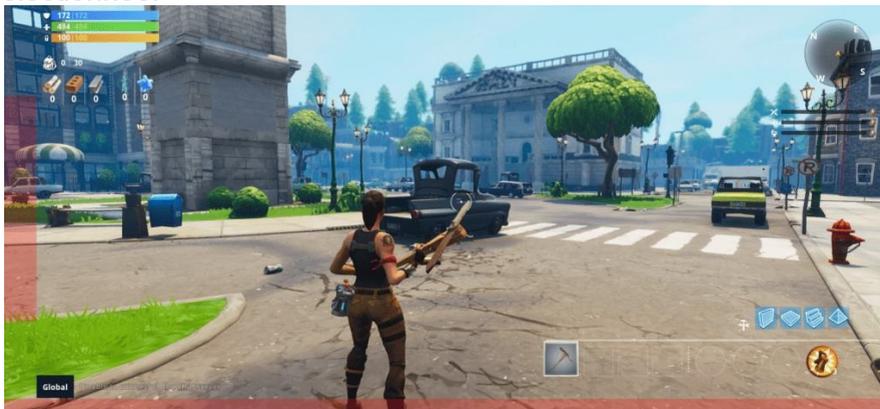


Slate a un nouveau système de clipping qui prend en charge le découpage en layers complexes avec des transformations arbitraires. Pour mieux comprendre, voir cette vidéo:

Améliorations de la zone de sécurité UMG

4.20

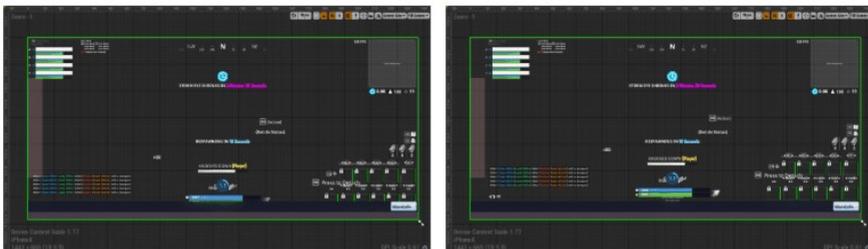
Les tailles d'écran que vous sélectionnez dans les paramètres UMG et Play-In-Editor (PIE) sont maintenant liées aux profils de périphériques, qui prennent également en compte le facteur d'échelle de contenu mobile, ce qui signifie que la résolution finale taille sélectionnée.



Unreal Engine 4 – Guide des mises à jour (4.20)

Les améliorations suivantes ont été apportées au flux de travail UMG Safe Zone:

- La prévisualisation de la zone de sécurité est maintenant automatiquement activée pour la zone de sécurité du titre de débogage lorsque vous utilisez une valeur inférieure à 1 pour tester la taille de l'écran des téléviseurs et des moniteurs.
- La commande `r.MobileContentScaleFactor` fonctionne pour mettre à l'échelle les résolutions de téléphone et de tablette dans les aperçus UMG et les modes PIE.
- Les zones sécurisées non-uniformes sont désormais prises en charge pour les appareils tels que l'iPhoneX, où certaines parties de l'écran sont inaccessibles.
- Les zones de sécurité, les boîtes d'échelle et les widgets de bordures communes réagissent correctement aux zones de sécurité non uniformes et aux tailles UMG Designer.
- UMG affiche maintenant le périphérique sélectionné, sa taille d'écran et son facteur de mise à l'échelle uniforme pour faciliter la consultation dans le graphique Designer.
- Utilisez `r.MobileContentScaleFactor` pour adapter les résolutions de téléphone et de tablette aux modes UMG et PIE.



Pour plus d'informations, voir [Zones de sécurité UMG](#).

UI

UI: Curseur Personnalisé

4.7

On peut créer des custom Widgets qu'on utilise comme des Curseurs. Ainsi, on peut **ajouter du texte, une barre de vie, et tout ce qu'on veut au curseur**, comme par exemple:



Les "Software cursors" sont utiles dans plusieurs cas:

- Ajouter des curseurs graphiques à du matériel ne possédant pas de curseur habituellement, comme par exemple, une souris virtuelle sur une console.
- Ajouter des curseurs aux fonctionnalités avancées, avec des animations, ou possédant des interactions complexes.
- Avoir des curseurs identiques d'une plate-forme à une autre (PC, Mac, Linux, etc.)

A noter que les curseurs logiciels sont mis à jour à chaque frame, alors que les curseurs hardware bénéficient de la vitesse de rafraîchissement du système d'exploitation.

UI: interface utilisateur améliorée

4.16

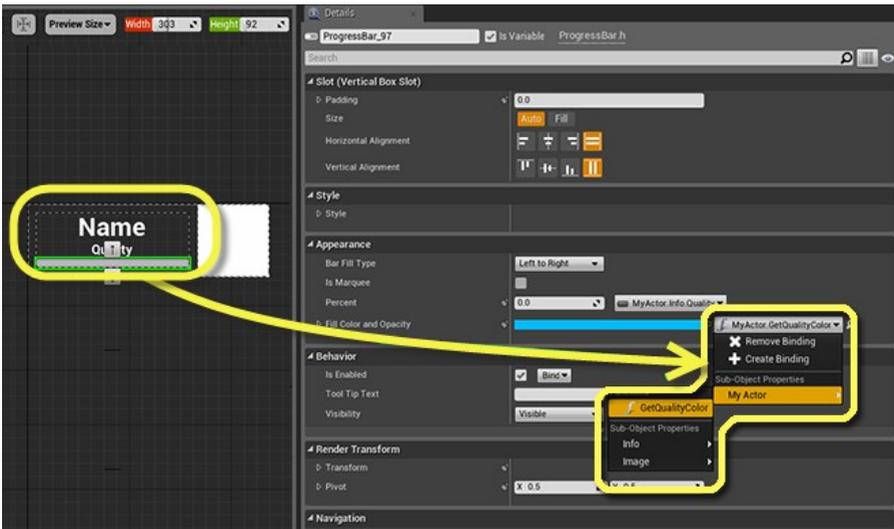
Les jeux qui utilisent des **Invalidation Panels** (je connaissais les box, pas les panels) ont maintenant une option pour **mettre en cache** des éléments (uniquement widget), leur permettant de bénéficier d'une **texture batching améliorée** et réduit les temps de manière significative.

Le résultat est un gain important de performances sur les appareils mobiles!

Unreal Motion Graphics: le nouveau Widget Binding System

4.7

Il est possible d'ajouter des fonctions et bindings pour chaque sous-propriété d'un widget:



Les conversions de base sont gérées automatiquement (comme

Unreal Engine 4 – Guide des mises à jour (4.20)

Linear Color vers Slate Color, ou Texture2D vers Slate Brush).

"Binding to properties" est désormais 10X plus rapide grâce à ce nouveau système.

Curve Editing pour l'interface utilisateur Unreal Motion Graphics

4.8

UMG a maintenant un éditeur de courbes. UMG, c'est le machin pour **faire les interfaces, les menus, les boîtes à cocher, voir tout ce qui est overlay**. Ben maintenant, ça **supporte l'animation des widgets!**



- sélectionner les pistes de valeurs numériques dans une animation widget
- animer grâce à l'éditeur de courbe.
- jouer, rembobiner et boucler l'animation lors de la prévisualisation avec les boutons de contrôle de transport.

Widgets

Fast CreateWidget

4.16

La création de widget UMG est maintenant jusqu'à **3 fois plus rapide**. Bon, perso, cela ne m'avait jamais vraiment gêné. Le compilateur

Unreal Engine 4 – Guide des mises à jour (4.20)

widget génère maintenant un modèle de widget qui est stocké dans le même pack que la Blueprint Class générée et utilisée lors de la création de widget.

Widget Depth Ordering

4.9

Les composants Widget à l'écran sont maintenant **triés automatiquement suivant leur distance par rapport au spectateur**, en veillant à ce que la plus proche soit toujours au-dessus des autres composants.



Widget interaction component

4.13

Pour simuler des événements venant du matériel avec des composants de widget dans la scène.

Unreal Engine 4 – Guide des mises à jour (4.20)



Vous pouvez joindre comme pointeur laser toute manette, souris ou autre matériel afin d'interagir avec les widgets.

Je peux vous confirmer que c'était bien une fonctionnalité manquante jusqu'à présent ! Dans l'un de mes derniers projets de VR on devait pouvoir interagir avec des menus réalisés en widget 3D. Il fallait émuler des clics de souris et utiliser une bidouille pour y arriver. C'était chiant et pas top, mais on y arrivait. Maintenant, ce sera beaucoup plus simple. Avant ça, on ne pouvait pas vraiment utiliser les widgets 3D en VR sans bidouiller...

Sprites in UMG Widgets

4.13

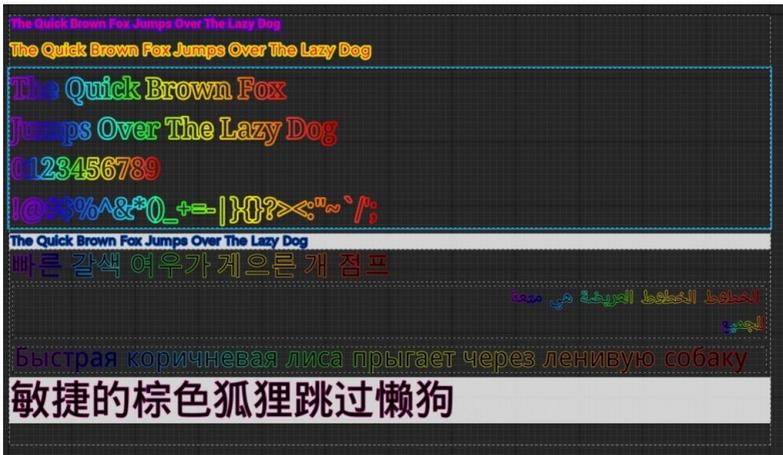
On peut maintenant utiliser des sprites provenant des brushes paper 2D dans les widgets 3D (UMG).



Fontes en outline

4.14

Les fontes peuvent maintenant être utilisées en version "outline" dans l'UI:



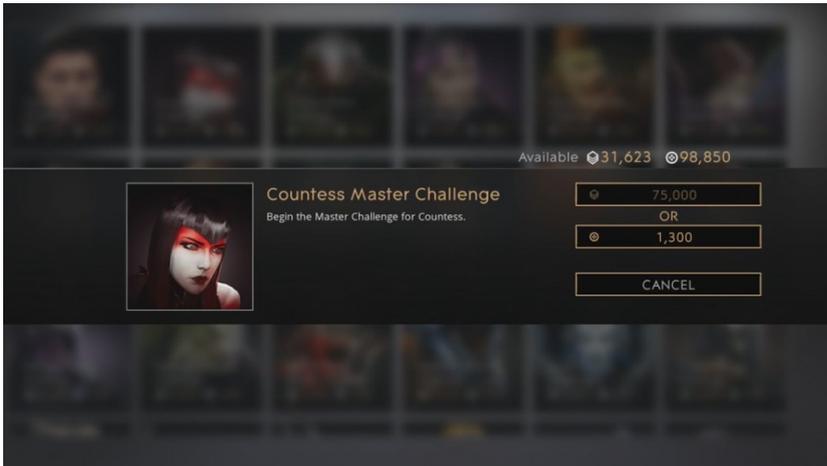
Tous les widgets possèdent donc un "outline size", mais aussi un "outline material" pouvant être utilisé pour réaliser divers effets comme:



Interface Blur rects

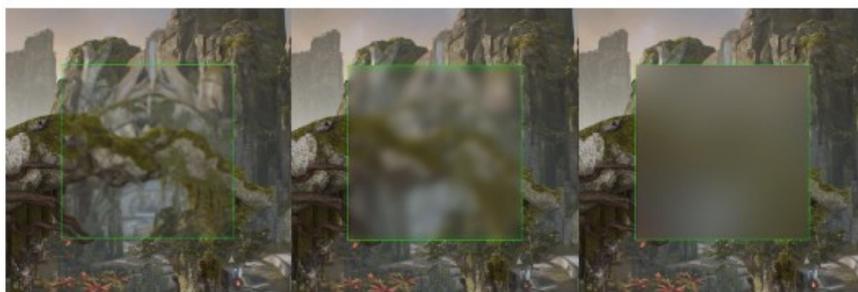
4.15

pour brouiller l'arrière plan dans l'interface utilisateur ou dans la scène 3D.



Une valeur de résistance définissant le flou d'arrière-plan va de 0 (off) à 100 (pleine puissance) et peut être réglée pour modifier le pourcentage de flou au fil du temps.

Unreal Engine 4 – Guide des mises à jour (4.20)



Force = 2

Force = 10

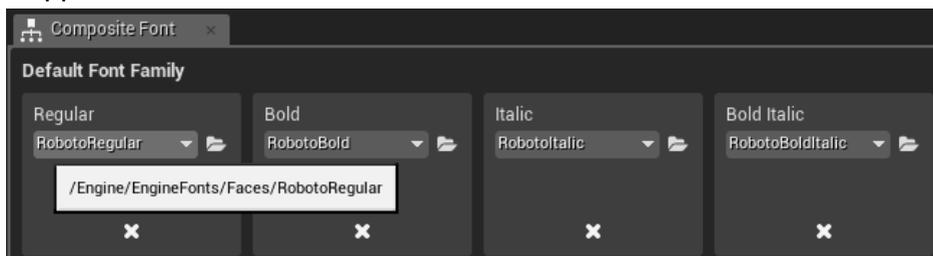
Force = 50

Ce n'est pas si gadget que ça: en fait, j'avais développé ma propre fonction pour ça en utilisant les filtres 2D, mais si c'est tout prêt, c'est encore mieux :)

Mise en place d'un asset store pour les fontes

4.15

qui réduit la consommation de mémoire et l'améliore la stabilité de l'application.

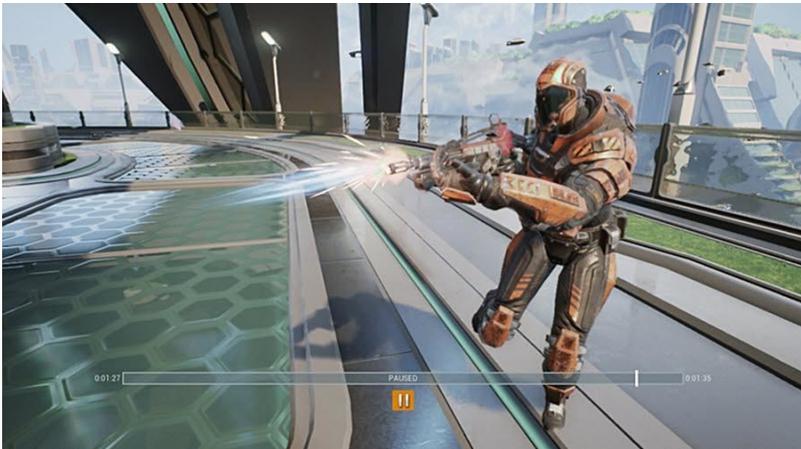


RÉSEAU & MULTIJOUEURS EN LIGNE

Replays des jeux en réseau

4.8

Enregistrement complet du jeu pour relecture! Je vous présente cette fonctionnalité dans le tome 5 des cahiers d'Unreal Engine.



Quand vous avez fini votre partie, lancez le replay et revoyez votre partie.

Bon, ça doit pas être évident quand même. Si on prend juste l'exemple du "Root Motion", ça ne fonctionne pas automatiquement en réseau, alors j'imagine que là, ça doit pas forcément couler de source. Mais c'est à voir et à tester avant de jaser...

Lors de l'affichage d'une rediffusion:

- La molette de la souris contrôle la vitesse de lecture
- Il y a un bouton à l'écran pour mettre en pause / play
- Vous pouvez cliquer n'importe où sur la barre de temps pour choisir le moment

Réplifications

4.12

Coté réplication, des optimisations significatives ont été réalisées, sauvant ainsi plusieurs cycles de CPU.

C'est surtout au niveau de la **surveillance de ce qui change** que le système à été optimisé, évitant ainsi de mettre à jour via le réseau des propriétés inchangées.

On annonce une amélioration "massive" des performances, rien de moins !

Character movement speed hack protection

4.12

Les jeux en réseau sont souvent la cible de hackers qui tentent d'obtenir des avantages par les **exploits**.

Les jeux conçus en utilisant la fonctionnalité non modifiée de déplacement de personnages étaient sensibles à cet exploit. Une détection de ce style d'action a été mise en place de façon à limiter ces actions.

SONS ET MUSIQUES

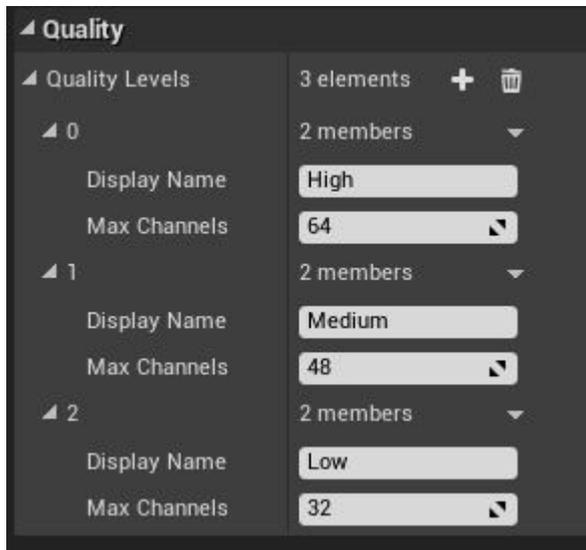
Moteur Audio d'origine

Niveaux de qualité sonore

4.9

Possibilité de gérer l'utilisation de la mémoire audio et la performance, notamment pour les mobiles.

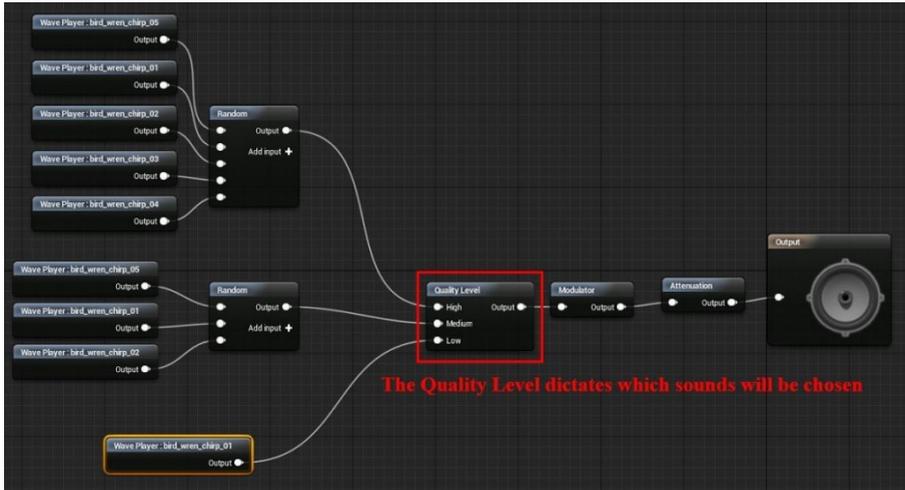
Les niveaux de qualité sonore disponibles sont définis dans la section Audio des paramètres du projet. Chaque niveau de qualité peut actuellement spécifier le nombre maximum de canaux audio créés par le dispositif audio (bien que certaines plates-formes, notamment Android, peut encore limiter cette valeur un nombre inférieur).



Dans un Sound Cue, vous pouvez placer un nœud de niveau de qualité sonore donnant une broche d'entrée pour chaque niveau de

Unreal Engine 4 – Guide des mises à jour (4.20)

qualité défini dans les paramètres du projet afin de réduire le nombre de variations qui sont chargées ou d'indiquer des fichiers de faible qualité sonore.



Quand on lance un jeu, le niveau de qualité est spécifié via les paramètres de jeu de l'utilisateur.

La valeur par défaut peut être réglée via le fichier `GameUserSettings.ini` approprié et le réglage, dans le jeu, via leur interface de réglages.

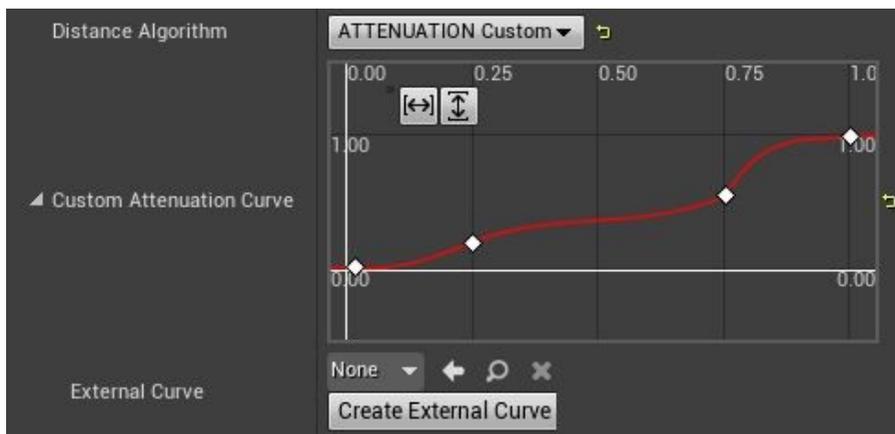
Actuellement, le jeu doit être redémarré pour que les changements prennent effet.

Pour Jouer dans l'éditeur, vous pouvez spécifier quel niveau de qualité à utiliser via les Paramètres de lecture de l'éditeur de niveau. Changer cette valeur ne fonctionne alors que la session de jeu est actif et de nouveaux sons utilisera le niveau de qualité actuel.

Courbes d'atténuation personnalisées pour l'audio

4.9

Vous pouvez maintenant **définir votre propre courbe personnalisée pour atténuer les sons.**



Dans "attenuation settings" fixez le Distance Algorithm à "custom" et vous verrez la section "custom curve" apparaître.

Vous pouvez ensuite spécifier soit une courbe d'assets extérieurs ou définir une courbe personnalisée.

Décompression audio Asynchrone Real-time

4.9

Les plates-formes qui prennent en charge en temps réel ogg-Vorbis basé sur la décompression audio (PC, Xbox One, Mac, Android) effectuent maintenant la **décompression de manière asynchrone** quand c'est possible pour éviter le blocage des opérations.

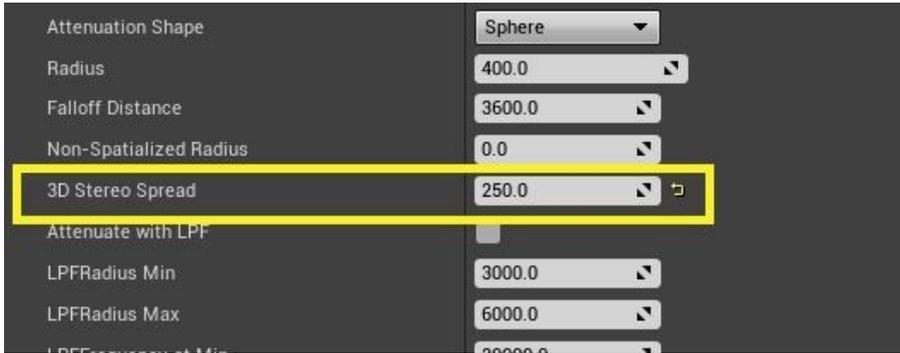
Spatialisation stereo

4.11

La spatialisation 3D est maintenant possible pour les **assets stéréo** sur PC, Xbox One et les plates-formes PS4. Elle spatialise essentiellement chaque canal de source d'entrée comme si elles étaient mono-sources.

Les positions des canaux gauche et droit sont déterminées par la position de l'émetteur du son par un 3D Stereo Spread, un nouveau paramètre dans Sound Attenuation Settings.

Unreal Engine 4 – Guide des mises à jour (4.20)

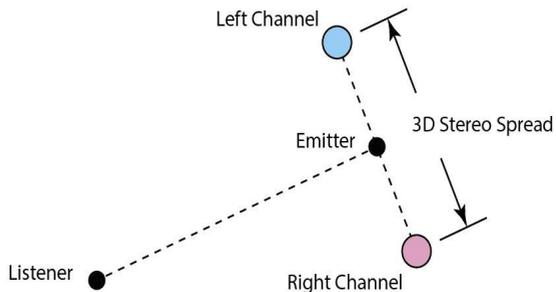


Le paramètre 3D Stereo Spread définit la distance en unités de jeu entre les canaux gauche et droit et le long d'un vecteur perpendiculaire au vecteur d'auditeur-émetteur.

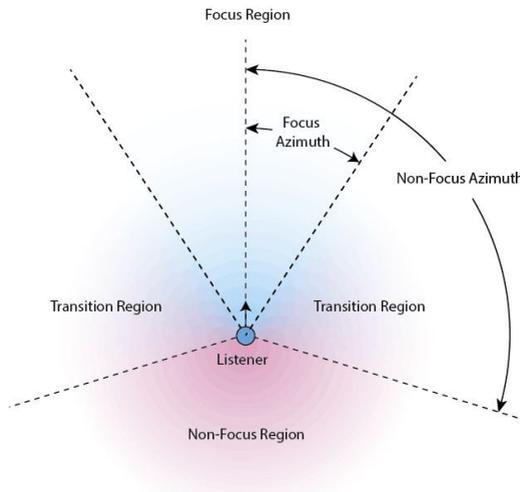
Sound focus

4.11

Sound Attenuation prend désormais en charge le **focus sonore**, une fonctionnalité qui permet aux concepteurs sonores de contrôler divers paramètres automatiquement en fonction de la direction du son par rapport à l'auditeur.



Le schéma suivant illustre les paramètres d'azimut:



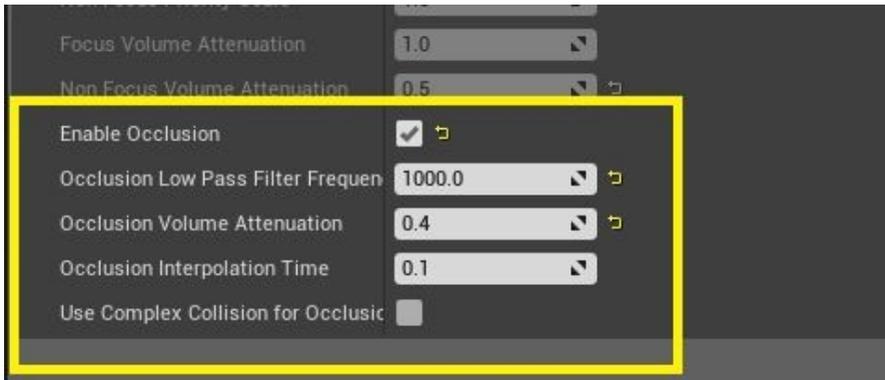
A l'utilisation, ça reste coton, mais ça permet de créer en effet une zone où le son est plus fort, ce que l'on pouvait déjà faire avant, mais plus précisément maintenant.

Sound occlusion

4.11

Pour permettre des calculs d'occlusion sur un son, il suffit de spécifier dans l'asset Sound Attenuation Settings comme indiqué dans le tableau suivant:

Unreal Engine 4 – Guide des mises à jour (4.20)



En gros, cela utilise un raycast pour savoir si le son entre en contact avec un objet proche et réaliser une occlusion, un peu à l'image de l'ambient occlusion pour la lumière.

Je n'ai pas encore testé ça, mais je ne sais pas si j'aurais l'oreille pour ça... En tous cas, cela devrait permettre une meilleure immersion dans les lieux confinés par exemple – si le designer son est attentif aux réglages...

Sound concurrency

4.11

Cela permet de gérer une sorte de scale sur les sons en concurrence, en fonction de l'émetteur du son. Par exemple, si le personnage émet 3 sons, dont la voix, et que l'on souhaite que sa voix soit plus importante s'il y a concurrence de son, on peut le faire pour lui, indépendamment des autres acteurs de la scène.

Avant, on pouvait déjà le faire, mais sans prendre en compte la source il me semble. Bref, si vous voulez en savoir plus, reportez vous au thread d'origine.

UE4 Sound Mixes

4.12

UE4 Sound Mixes permet d'**ajuster dynamiquement les propriétés de classe sonores**.

Unreal Engine 4 – Guide des mises à jour (4.20)

Avec 4.12, les mélanges sonores remplacent dynamiquement les ajustements de classe sonore de Blueprints en utilisant une nouvelle fonction de Blueprint, “Set Sound Mix Class Override”.

Audio localization

4.12

Le support intégré est amélioré pour la localisation audio, entendez ici la traduction quand on parle de localisation.

Le texte parlé est maintenant rassemblé dans des assets et inclus dans vos fichiers PO pour la traduction avec votre autre texte reposant sur l’asset.

L’enregistrement et l’importation de l’audio sont facilités pour vos assets d’onde de dialogue.

Dans le tome 4 des cahiers d’Unreal Engine, je pointais déjà du doigt cette lacune.



Nouveau plugin MIDI

4.14

Permet de connecter directement son synthétiseur par exemple et de dialoguer avec lui via des ordres "midi", ces derniers étant accessibles par les Blueprints via un **MIDI Device Manager**.



Il suffit ensuite de faire un break sur la structure et d'accéder à un tas de variables.

Pour le sequencer

4.15

ce qui permet aux concepteurs de mettre en place fade in, fade out, et d'autres effets. Ben oui, certaines fonctions évidentes ne sont pas toujours en place - ça comble un manque ici... N'oublions pas que le sequencer est relativement récent et qu'il est amené à remplacer totalement matinee à terme (c'est déjà le cas en vérité).

Stereo Layers Unified Across Platforms

4.17

Toutes les plateformes VR prennent désormais en charge les couches stéréo out-of-the-box par UE4.

Vous pouvez maintenant utiliser des layers stéréo à travers toute plate-forme, sans devoir faire attention aux différences fonctionnelles entre plates-formes.

Tirage au sort de l'ordre des layers

4.17

Pour fournir un comportement similaire pour la commande de Stereo Layer sur toutes les plateformes. Pour les plates-formes qui le supportent, il est possible de revenir à l'ancien comportement en réglant les `vr.StereoLayers.bMixLayerPriorities` à 1.

Steam Audio

Early Access

4.16

Epic et [Valve](#) se sont associés pour mettre en œuvre une implémentation entièrement intégrée du SDK steam audio utilisant les nouvelles fonctionnalités de Unreal Engine Audio (spatialisation, occlusion, réverbération). C'est exactement ce dont je vous parlais dans [cet article](#).

C'est fou ce que l'API audio évolue rapidement depuis quelques versions.

Steam Audio Beta 10

4.19

- Prise en charge Android pour la spatialisation et l'occlusion
- Mode Editeur pour un accès facile à la fonctionnalité Steam Audio
- Boutons pour ajouter et supprimer des composants de géométrie Phonon à tous les acteurs de maillage statique
- Bouton pour exporter la scène vers `.obj` pour le débogage
- Acteur Phonon Scène enlevé en faveur de l'exportation vers le fichier

Nouveau moteur Audio

Unreal Engine Audio (Early Access Preview)

4.16

disponible au début sur PC, Mac, iOS, Android et Switch. Il comprend un **mélangeur audio multi-plateformes** avec soutien backwards-compatible, y compris un nouvel égaliseur multi - plateformes et les effets Reverb master. De plus, le nouveau moteur audio Unreal introduit de nouvelles fonctionnalités comme un graphique Submix, effets Submix, effets source, synthèse en temps réel, et un meilleur support de plug - in audio.

Le nouveau moteur audio Unreal est pas activé par défaut dans 4.16, pour l'activer, utilisez l'argument de ligne de commande «-audiomixer».



Pour y voir plus clair, je vous invite à visualiser les dernières vidéo de Meletou à ce sujet.Un grand merci à lui pour cette présentation.

Synthesis Plugin (Early Access)

4.16

Le nouveau plug-in contient deux nouveaux synthétiseurs temps réel utilisant la nouvelle Unreal classe «**SynthComponent**» du moteur audio pour mettre en œuvre un synthétiseur soustractif ainsi qu'un granulator temps réel. Ces nouveaux synthétiseurs ne sont pas seulement des outils utiles pour la musique procédurale et la conception sonore, mais ils servent d'exemple pour fabriquer un plug-in tiers et même les concepteurs sonores pourraient l'utiliser. Le plug-in contient également de nombreux effets source DSP et de

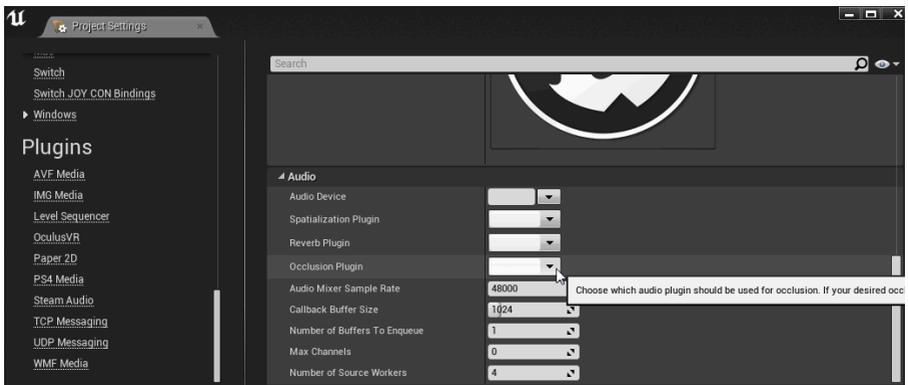
Unreal Engine 4 – Guide des mises à jour (4.20)

prémixage : Source Effets (Stereo Delay, bit crusher, processeur dynamique, Enveloppe Follower, Filtre EQ, filtre analogique virtuel (variable Ladder / État), Vague Shaper, Chorus, Phaser), Prémixage Effets: Reverb, EQ, Dynamics Processeur

Améliorations de l'architecture de plug-in audio

4.18

La spatialisation des sources binaurales basées sur les objets, l'occlusion de la source et la réverbération principale sont mieux réalisées grâce à une meilleure intégration dans le moteur. Ces effets nécessitent un support de plug-in audio spécial.



Source Bus

4.18

Le source bus permet à plusieurs sources sonores d'envoyer et de former une seule source en temps réel, d'acheminer l'audio vers une source audio 3D.

Nouveaux effets de sous-mixage

4.18

- Delay: Cet effet simule un délai de diffusion simple sur l'audio envoyé au sous-mixage.
- Tap Delay: Effet de retard multicanal complexe qui permet une quantité arbitraire de taps ("échos") pouvant être ajoutés, modifiés et supprimés indépendamment via une API Blueprint puissante pendant l'exécution.
- Flexiverb: Algorithme de réverbération monocanal peu coûteux et optimisé pour optimiser la densité d'écho avec le moins de multiplications possibles. Cet effet est le mieux adapté pour émuler une réverbération de petite taille sur des plates-formes où les ressources du processeur sont limitées.

Ajout du plugin audio Resonance

4.19

Il prend en charge la spatialisation binaurale, la réverbération, la directivité sonore et la lecture ambisonique sur Windows, iOS, Android, Linux et Mac. Disponible uniquement sur le nouveau moteur audio.

Fichiers Ambisonics

4.19

Les fichiers Ambisonics peuvent maintenant être importés dans des projets Unreal et lus à l'aide de la sélection Spatialization plugins. First-Order Ambisonics est un format spherical surround. Ceci est utile pour la réalité virtuelle où les utilisateurs verront des changements dans les champs sonores verticaux en inclinant la tête ou pour toute autre expérience interactive où l'orientation de l'utilisateur permet des changements dans la verticale.

Améliorations VOIP native de UE4

4.19

Meilleure cohérence, notamment une meilleure détection du silence sur Windows. L'intégration de sources VOIP dans la scène permet maintenant:

- Spatialisation
- Atténuation de distance
- Effets de source personnalisés
- Réverbération

Tout est contrôlé à l'aide du composant de scène UvoipTalker. Disponible uniquement sur le nouveau moteur audio.

Nouveau composant de capture de microphone

4.19

Le composant de capture de microphone permet:

- Alimentation audio directement dans le jeu à utiliser avec le mélangeur audio
- Traitement audio avec effets de source ou effets de submixage.

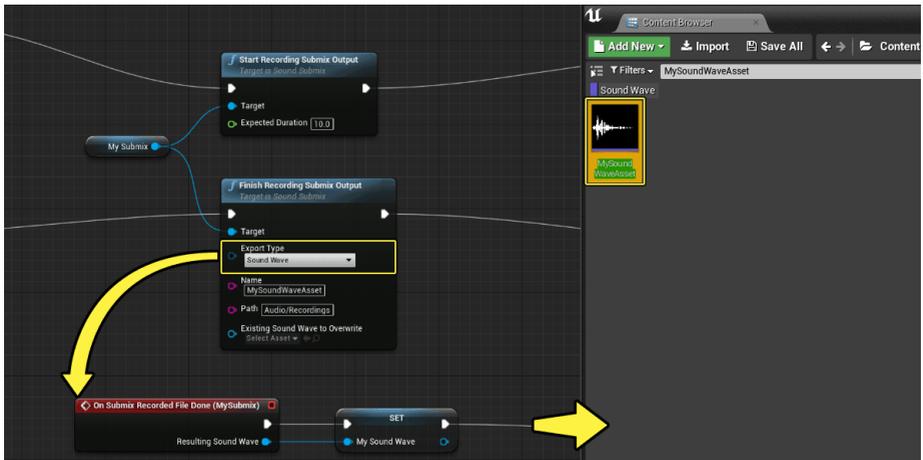
Bien qu'il ne prenne pas en charge l'enregistrement des sorties micro directement sur les assets pour le moment, il existe des possibilités pour les développeurs de plugins tiers de tirer parti des données d'une entrée microphone. Windows uniquement.

Submix Audio Recording

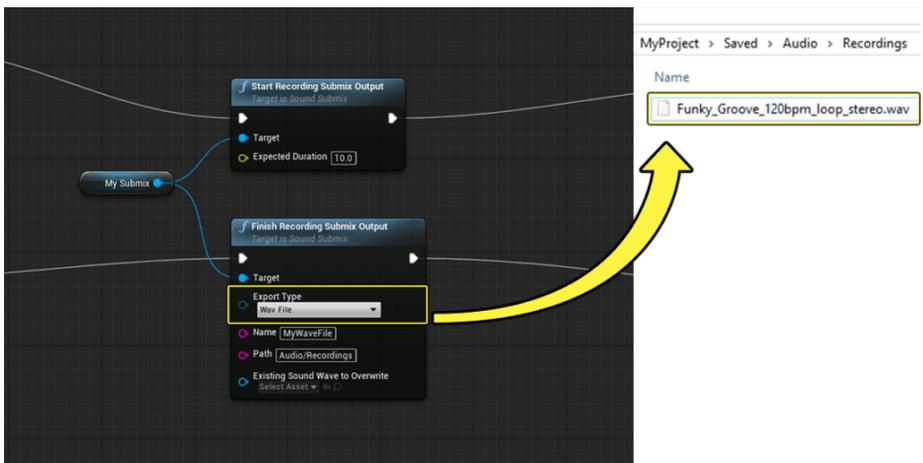
4.20

Dans le nouveau moteur audio, la possibilité d'enregistrer la sortie du moteur - ou la sortie de n'importe quel Submix individuel - dans un fichier * .wav ou SoundWave Asset a été ajoutée.

Unreal Engine 4 – Guide des mises à jour (4.20)



Exportation d'une sortie Submix vers un SoundWave Asset.



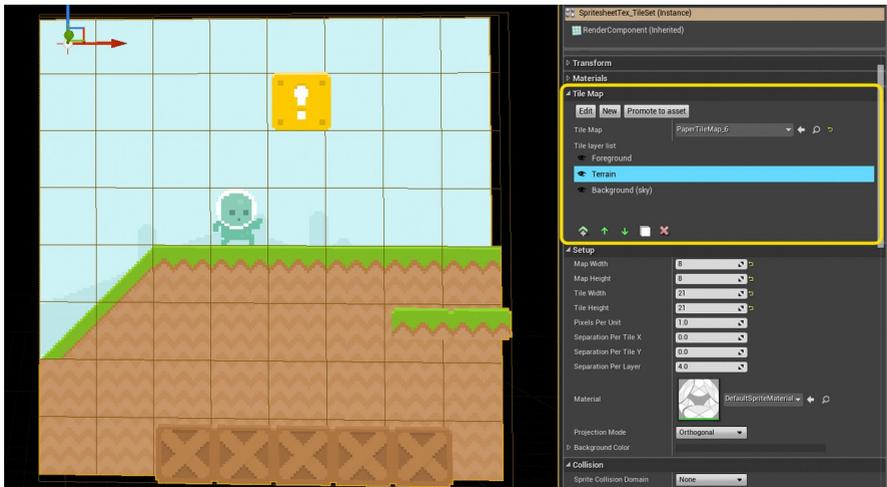
*Exportation de la sortie du Submix vers un fichier *.wav.*

PAPER 2D

Tilesmap et Tileset

Amélioration de l'édition de Layer Tile Map

4.7



- Les Layers peuvent être réordonnés
- On peut sélectionner directement le layer à peindre
- Le layer sélectionné reste actif d'une session à une autre
- On peut dupliquer les layers
- Les Tile Maps sont désormais des assets réutilisables ou des instances du level (avec la possibilité de les transformer en asset)
- Etc.

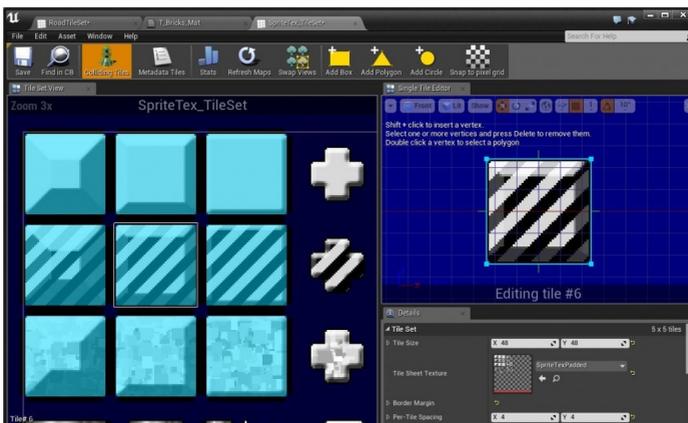
A noter que les Tiles Maps sont toujours une fonction expérimentale du moteur.

TileMaps plus accessibles

4.8



- Programmation avec Tile Map Components de Blueprints entièrement pris en charge
- Tile map color peut maintenant être contrôlé à la fois par couche et à travers la carte de tuiles entière
- Staggered Isometric et Staggered Hexagonal projections pris en charge, et les grid/cursor previews sont améliorés dans tous les modes
- Ajout du support pour la rotation des tuiles
- Géométrie de collision et une clé de métadonnées personnalisés peuvent désormais être spécifiés

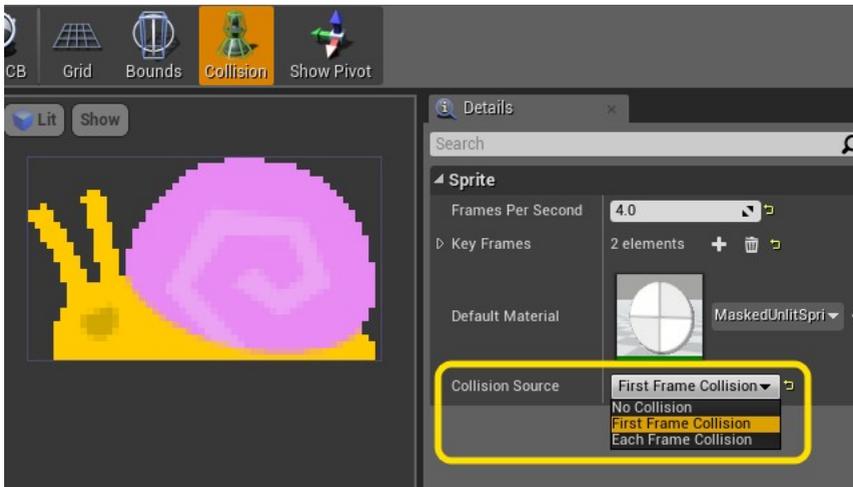


Flipbooks

Collision Detection

4.7

Flipbook, c'est ce qui permet de créer des animations 2D à partir d'une série d'images (sprites 2D animés, planche d'images, etc.). Maintenant, il est possible de **définir des propriétés de collisions au niveau des frames de l'animation**.



Ainsi:

- No Collision: Pas de collision
- First Frame Collision: Une collision est dispo uniquement pour la première image (elle sera identique partout)
- Each Frame Collision: La collision est définir pour chaque image

Sprites

Peindre directement sur des Sprite

4.8

Les composants Sprite peuvent être sélectionnés et peints en mode Mesh Paint



La texture source ou des textures supplémentaires peuvent être sélectionnées et peintes.

Remarque: la peinture de vertex ne fonctionne pas actuellement avec des composants sprite.

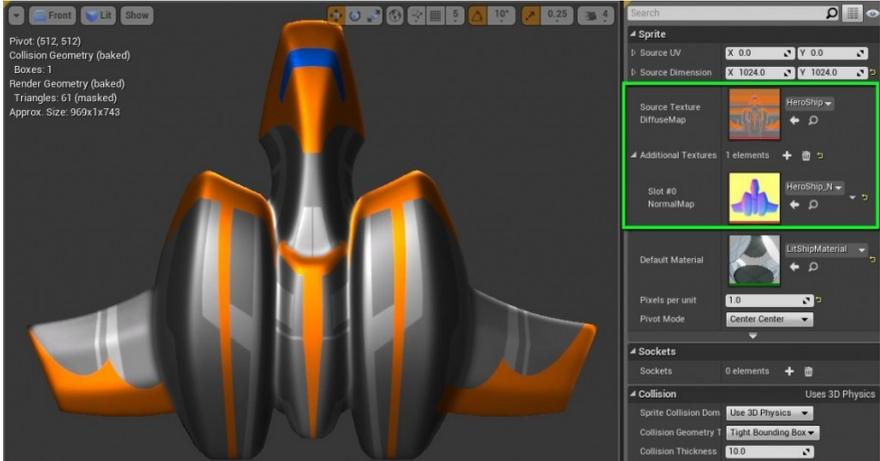
Support de multiples textures

4.8

Définir des texture supplémentaires «créneaux».

Pour réaliser des matériaux personnalisés, un sprite éclairé par une carte normale, ou passer d'une texture translucide en deux textures opaques (l'un contenant des données alpha) pour la compression ETC1 sur Android.

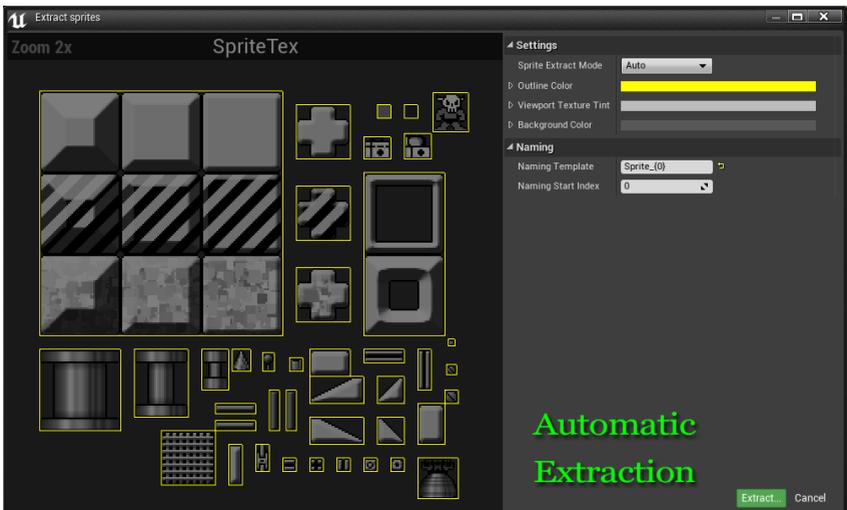
Unreal Engine 4 – Guide des mises à jour (4.20)



Interactive Extract Sprites Dialog

4.8

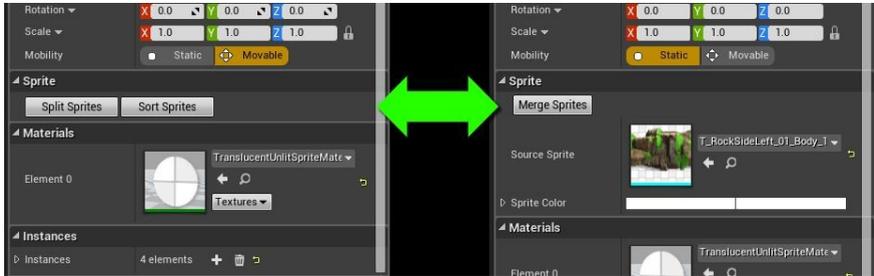
La boîte de dialogue pour extraire des sprites Extraire permet plus de contrôles sur la façon dont les sprites sont divisés à partir d'une texture, la rendant plus rapide et plus facile pour importer / préparer des assets pour une utilisation avec Paper2D



Grouped Sprite Components

4.8

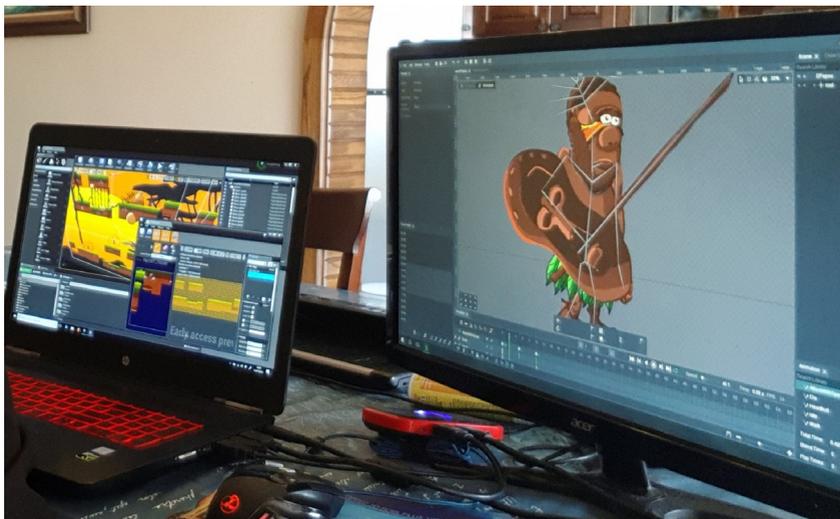
Permet de travailler facilement avec des groupes de sprites.



Pas d'évolution?

Pour terminer, Paper2D n'a quasiment pas évolué depuis la version 4.8 mais propose un système relativement simple et stable pour la création de jeux 2D. On attend toujours la possibilité d'activer une physique uniquement 2D, et la gestion d'un squelette pour l'animation 2D avec blending (on peut utiliser des plugins tiers comme Creature), ainsi que la gestion de sprites vectoriels.

Unreal Engine 4 – Guide des mises à jour (4.20)



Mon prochain jeu utilise Paper2D, mais je lui ajoute pas mal de fonctionnalités. Voir [démonstration en cours](#).

RÉALITÉ AUGMENTÉE

De la création de spectacles à la mise en scène pour le e-sport !

On sent que coté Epic Games, il y a une réelle volonté de passer du support "écran" et casques VR à un mode plus "spectacle" où plusieurs personnes peuvent assister et participer. J'imagine déjà des ambiances terribles dans des boites de nuit, des concert ou des expositions inoubliables. Et le tout, avec de l'interactivité et des effets à couper le souffle comme sait le faire UE4. On entre vraiment progressivement dans une autre dimension où réel et imaginaire vont très vite se confondre au quotidien. La magie va t-elle prendre naissance? Non, pour de la vraie magie, il faudra attendre la téléportation et la naissance des holodecks de startrek. Mais visuellement... on va vraiment y être et c'est pas pour dans 10 ans !

Capture de réalité mixte (accès anticipé)

4.20

Créez des expériences de spectateurs captivantes pour les applications de réalité mixte en utilisant la nouvelle fonctionnalité de capture de réalité mixte, qui facilite le mélange de vrais joueurs dans un espace de jeu virtuel !

Le support de l'accès anticipé à la capture de réalité mixte a trois composantes:

- l'entrée vidéo,
- l'étalonnage
- et la composition en jeu.

Unreal Engine 4 – Guide des mises à jour (4.20)

Une liste de webcams prises en charge et de périphériques de capture HDMI qui vous permettent d'extraire la vidéo du monde réel dans Unreal Engine à partir de diverses sources est à disposition.

Si vous avez un Vive Tracker ou un dispositif de suivi similaire, Mixed Reality Capture peut faire correspondre l'emplacement de votre caméra à l'appareil photo intégré au jeu pour rendre les prises de vues plus dynamiques et intéressantes. La configuration et l'étalonnage sont effectués à l'aide d'un outil d'étalonnage autonome qui peut être réutilisé sur les titres Unreal Engine 4. Une fois que vous avez configuré votre lieu de tournage, vous pouvez l'utiliser dans toutes les applications.

Alors que le support des fonctionnalités est en phase d'accès anticipé, Epic est impatiente de recevoir des commentaires tout en continuant à améliorer le système. Vous trouverez plus d'informations sur la configuration de la capture de réalité mixte dans la [documentation](#) sur le développement de la réalité mixte .

Cela a l'air fort prometteur, j'ai bien envie de me pencher là dessus, mais je crains qu'il soit nécessaire d'avoir pas mal de matos pour ça... Dites moi si vous avez l'occasion de tester de votre coté.

nDisplay Flexible, Rendu multi-écrans

4.20

Créez, sans effort, des murs vidéo pour de grandes installations de visualisation en utilisant le nouveau système nDisplay ! Lancement automatique d'un nombre illimité d'instances d'Unreal Engine, verrouillées fermement, avec un contenu déterministe et une synchronisation temporelle précise, sur un nombre illimité d'ordinateurs hôtes, chaque instance pilotant son propre projecteur ou moniteur. Utilisez le rendu stéréoscopique actif ou passif pour améliorer le sentiment d'immersion du spectateur dans la scène 3D et la prise en charge VRPN intégrée pour piloter le système à partir des contrôleurs VR mobiles.



De quoi satisfaire les organisateurs de spectacles voulant créer des expériences visuelles artistiques et immersives !

Magic Leap One™ Early Access Support

4.20

À la GDC, le support en Early Access pour Magic Leap One a été annoncé avec Creator Edition, un software pour le développement précoce d'expériences pour Magic Leap, dans le cadre d'un partenariat plus large entre les deux sociétés. Depuis Unreal Engine 4.20, vous pouvez développer pour Magic Leap One™ en utilisant la version entièrement supportée d'Unreal Engine.

- contrôle de caméra,
- maillage du monde,
- contrôleurs de mouvement
- et rendu en différé.



Les développeurs peuvent télécharger le kit de développement logiciel et le simulateur Magic Leap sur developer.magicleap.com . Pour les développeurs ayant accès au matériel, Unreal Engine 4.20 peut se déployer et s'exécuter sur le périphérique, en plus de prendre en charge les flux de travail d'itération zéro via l'éditeur Play In.

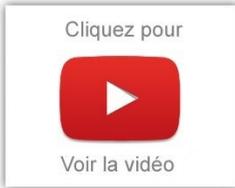
ARKit (iOS) Support

Experimental

4.17

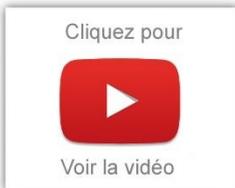
Support de ARKit pour iOS en version expérimentale. Cette technologie d'Apple est compatible avec les appareils équipés au minimum du processeur A9 et d'iOS 11. de nombreuses [utilisations sont envisageables](#).

Unreal Engine 4 – Guide des mises à jour (4.20)



Ce plugin est basé sur les travaux de l'équipe de Wingnut, mis en valeur lors de la conférence de presse d'Apple WWDC. Il permet aux développeurs d'utiliser les fonctionnalités avant le support officiel d'Apple dans iOS 11.

C'est assez impressionnant quand on voit la stabilité du bouzin, en comparaison de technos comme OpenCV. Ce qui est dommage, c'est qu'on est limité à Apple et que cela ne correspond pas aux habitudes d'Epic de ne proposer des fonctionnalités que sur une seule plateforme. Doit-on s'attendre sous peu à un implémentation de ces fonctionnalités sur PC avec un autre système ? Je l'espère pour ma part... Dans tous les cas, on voit bien qu'Epic s'investit de plus en plus dans la **Réalité Augmentée**, après avoir obtenu de super résultats en Réalité virtuelle. Normalement, il est prévu le support binauculaire des téléphones Google et de [la technologie Tango](#)...



Mais ça va donner quoi avec le monoculaire des téléphones traditionnels ? Et pour l'instant, il n'y a que peu de smartphones qui sont compatibles... Bref, affaire à suivre. En tous cas, chez Ekstra Reality, on fait le pari de l'ARKit:

Apple ARKit iOS 11

4.18

Unreal Engine 4.18 inclut un support en production de l'ARKit d'Apple dans iOS11.



Les flux de travail ont été simplifiés en utilisant les composants de l'infrastructure existante, une bonne gestion de la caméra passthrough a été ajoutée et une amélioration des performances et de prédiction. Téléchargez le projet [ARSample](#) pour voir ce qui a changé pour 4.18. Les projets

Unreal Engine 4 – Guide des mises à jour (4.20)

ARKit créés en utilisant 4.17 devront être mis à jour pour fonctionner en 4.18.

Apple ARKit 2.0 Support

4.20

- meilleure qualité de suivi,
- **détection de plans verticaux** (ahah, depuis qu'on l'attendait)
- suivi du visage,
- détection d'images 2D,
- détection d'objets 3D
- expériences AR persistantes et expériences AR partagées.

La prise en charge de ces nouvelles fonctionnalités vous permet de placer des objets AR sur plusieurs surfaces, de suivre la position et l'orientation d'un visage, de reconnaître et de donner vie à des images 2D, de détecter des objets 3D et de faciliter de nouveaux types d'expériences collaboratives.

ARCore (Android) Support

Google Plugin pour Tango UE4 (expérimental)

4.17

Unreal Engine 4.17 inclut un support expérimental pour le projet Google Tango sur Android. On en parlait ci-avant justement. Le plugin prend en charge le suivi de mouvement suivant six axes, l'apprentissage de la zone, la reconstruction de l'environnement et d'autres fonctionnalités de Google Tango.

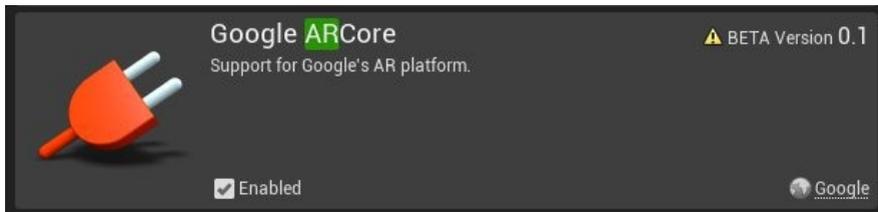
Remarque: l'API et les interfaces peuvent changer dans les versions ultérieures du moteur.

Prise en charge de Google ARCore Developer Preview

4.18

Pour le moment, il est possible d'utiliser la plateforme **Android** et **iOS** de façon **quasi-unifiée** (on verra qu'il y a quelques différences, mais on s'en sort au final). Quand je dis unifié, je veux dire qu'on peut facilement réaliser une application qui sera packagée pour l'une ou l'autre des plateformes, en changeant peu de code (ou en intégrant la différence directement dans le code).

En 4.18, la gestion de l'AR passe par des plugins: **Google ARCore** et **Apple ARKit**.



L'équipe d'Epic nous propose ce [Sample Project](#). L'intérêt, c'est qu'on dispose dedans des Blueprints nécessaire pour créer rapidement une application d'AR.

Pour le tester, vous devez vous assurer d'avoir un smartphone compatible. Même si AR Core est prévu pour fonctionner sur de nombreux smartphones Android "à terme", le SDK Preview [ne fonctionnera que](#) sur un Samsung Galaxy S8 ou un Google Pixel (+XL, 2, XL). Cela devrait arriver pour les autres [courant 2018](#). Probablement une conséquence [du projet Tango](#).

Unreal Engine 4 – Guide des mises à jour (4.20)



Dans project settings, on a une nouvelle option qui permet de déclencher au lancement le mode AR:



Si on ouvre ARPlayerPawn, on trouve un traitement dissocié, si c'est Android ou iOS.

Il reste encore des différences dans l'implémentation des 2 plateformes. Notamment, sur **la façon de détecter l'ensemble des plans** (ARKit est un peu plus opaque – normal, c'est Apple). Mais l'objectif d'Epic est d'obtenir au final un framework unifié d'accès à l'AR (ce qu'on sera obtenu en 4.19).

A voir: Twitch d'Epic sur la [Réalité Augmentée](#)

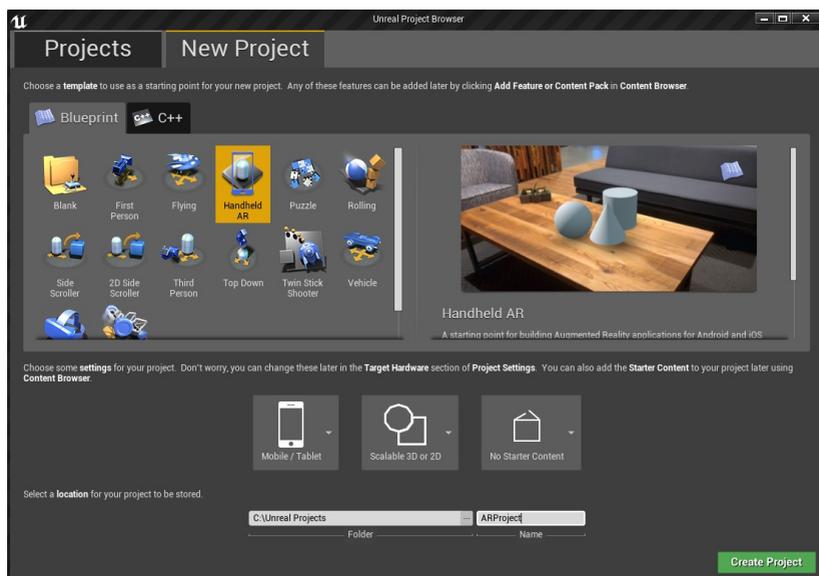
Un Framework unifié pour la réalité augmentée

4.19

Vous savez que sur Android, l'utilisation de la RA passe par Google ARCore. Mais sur iOS, c'est ARKit d'Apple. Les deux frameworks se programment différemment et n'ont pas exactement les mêmes fonctions, ni le même comportement. De fait, on est heureux de pouvoir compter un framework unifié coté UE4 qui nous masque ces différences justement, et nous permet ainsi de développer des applis et des jeux en réalité augmentée qui vont tourner sur les deux environnements avec très peu de modifications.

L'ensemble des outils pour développer ce type d'applications passe donc désormais par une librairie unique. Cela inclut les fonctions supportant l'alignement, les effets de lumière, le pinning, le tracking, etc.

Un nouveau template Blueprint **HandheldAR** a aussi été mis en place. Il fournit un exemple complet de projet démontrant les nouvelles fonctionnalités.



Unreal Engine 4 – Guide des mises à jour (4.20)

Il y a [un livestream](#) qui en parle, j'y reviendrai peut-être dans un prochain article ou une vidéo.

Google ARCore 1.2 Support

4.20

- prise en charge de la détection de **plan vertical**,
- images augmentées
- ancrés de nuage.

La prise en charge de ces nouvelles fonctionnalités vous permet de placer des objets AR sur davantage de surfaces, de reconnaître et de donner vie aux images et de faciliter de nouveaux types d'expériences de collaboration AR.

J'en connais qui vont pouvoir se passer de Vuforia et donc d'Unity maintenant que ces nouvelles fonctionnalités sont disponibles ;)

Entre nous, il était temps !

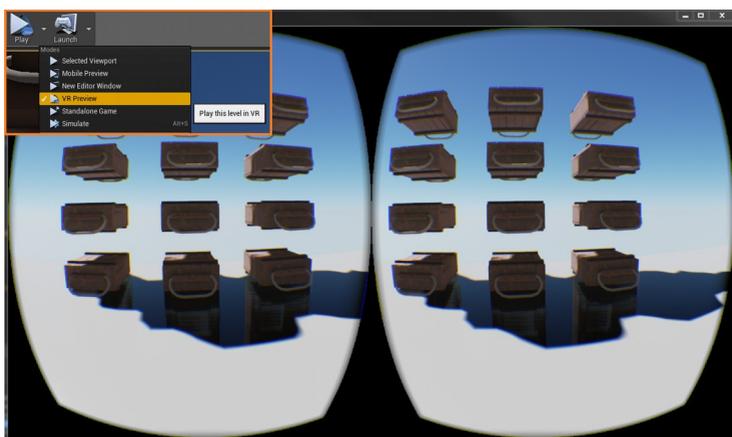
RÉALITÉ VIRTUELLE

Depuis les version 4.8, et ce, jusqu'à aujourd'hui, Epic Games a mis un point d'honneur à améliorer sans cesse le support des nouveaux casques de réalité virtuelle du marché. On sent que le paquet a été mis de ce coté là, et ayant travaillé, durant ce temps, sur un jeu multijoueurs utilisant la réalité virtuelle, je peux vous assurer que les mises à jour ont été fort appréciées. Il est aujourd'hui possible de réaliser de grosses productions utilisant la VR avec Unreal Engine.

4.7

Il est possible de lancer un preview directement dans l'éditeur. Si vous êtes les heureux possesseurs d'un casque de réalité virtuelle, vous allez pouvoir tester cela dans l'éditeur.

Il faudra pour cela rapprocher votre bouille de l'écran et utilisez votre main pour faire séparer... euh, pas sûr que cela soit terrible... Non, je déconne, cela envoi directement vers l'appareil.



C'est très utile, on ne peut pas s'en passer d'ailleurs. Toutefois, quand on commence à travailler en multijoueurs (réseau) avec des casques

Unreal Engine 4 – Guide des mises à jour (4.20)

de réalité virtuelle... c'est l'horreur... Mais peut-on imaginer une autre approche? Oui, la simulation du casque – c'est à dire avoir l'affichage à l'écran, même si le casque n'est pas branché. Mais ce n'est pas encore d'actualité.

Support SteamVR / HTC Vive

4.8

4.9

4.10

- Aperçu du jeu instantané sur HTC Vive à l'aide du bouton "Aperçu VR"!
- Toutes les fonctions de contrôleur de mouvement HTC Vive sont prises en charge et possibilité de lier les boutons de commande / claviers
- Support complet du casque – pas de la caméra avant toutefois.



Unreal Engine a maintenant un support complet pour l'API SteamVR de Valve et le matériel HTC Vive!

Bon, rien à redire de ce côté. C'est le casque que j'utilise et cela fonctionne très bien... Sauf si vous coupez "SteamVR"... même l'éditeur est killé! Et vous n'aurez même pas le temps de sauvegarder.

La version 4.9 apporte aussi des corrections et améliorations:

- La correction des problèmes de latence apparus dans la 4.8 2 +

Unreal Engine 4 – Guide des mises à jour (4.20)

- Le support pour l'abstraction de standard motion controller, donc deux contrôleurs peuvent guider le même joueur
- Les projets C++ sont maintenant bien supportés par le plugin API

Bon, la version 4.10 offre un support plus complet, comme le fait de pouvoir **utiliser les motion controllers directement comme des composants**. Cela devient ainsi un jeu d'enfant d'associer une main virtuelle à un contrôleur par exemple.

Prise en charge de SteamVR sur Mac

4.18

UE4 supporte officiellement la VR sur Mac via SteamVR. Toutes les interfaces auxquelles vous êtes habitué sur PC sont disponibles sur Mac, il est donc facile de transférer votre projet sur Mac pour le lancer. La mise à jour s'accompagne de la prise en charge de Metal 2 et de nombreuses autres optimisations d'exécution.

Support Vive Pro

4.19

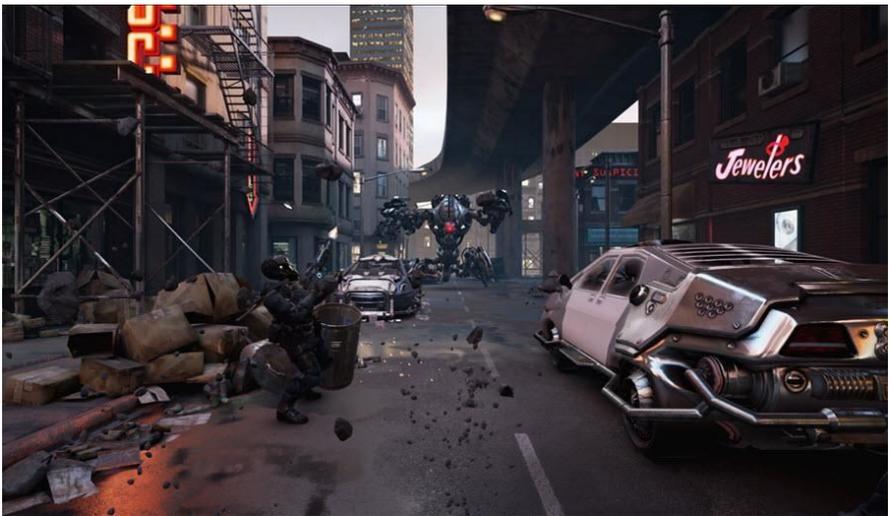
Le contenu UE4 existant prenant en charge SteamVR est dorénavant compatible avec Vive Pro! Aucune modification n'est nécessaire pour utiliser le Vive Pro avec tout projet UE4 existant, mais il faut vérifier les performances de chaque projet UE4 avec le Vive Pro et optimiser en conséquence car le Vive Pro fonctionne à une résolution plus élevée.

Playstation VR

PlayStation 4 Morpheus VR avec 120Hz Reprojection

4.8

Sony Morpheus est désormais pris en charge, avec un rendu 120 Hz. Purée, j'aimerais bien voir ça, mais j'ai demandé un kit de dev à Sony, et je crois que je peux encore attendre...



- Unreal Engine prend désormais en charge les deux kits de développement Morpheus 60hz et 120Hz
- La lecture audio 3D en VR a également été grandement améliorée pour cette version.

Support du contrôleur Playstation VR

4.15



Pour l'activer, il suffit de changer la valeur "Hand" à "Gun" sur votre Component Motion Controller. Les boutons de la PlayStation®VR Aim Controller carte les mêmes boutons sur la manette Dual Shock.

Social Screens (Early Access Preview)

4.16

PSVR Social Screen preview permet d'avoir un support des écrans séparés entre le casque et l'écran - fonction qui existe déjà sur PSVR, mais qui vient d'être supportée par UE4. Plus précisément, cette techno permet de transformer la vue "3DVR" du casque en une vue "2D" que les spectateurs peuvent regarder sur l'écran. Enfin, c'est ce que j'ai lu, j'ai pas encore testé.

Direct Multiview Support pour Daydream

4.17

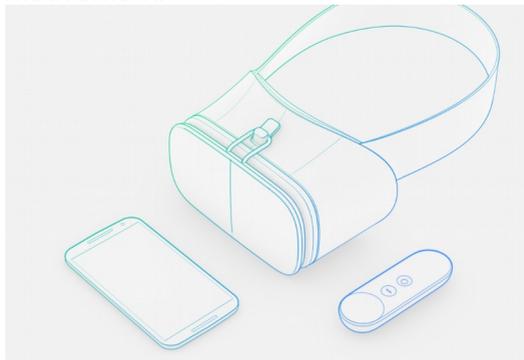
MultiView Direct peut maintenant être utilisé sur [la plate-forme Daydream](#) (google VR). Direct multiview travaille avec le compositeur

Daydream pour supprimer une copie plein écran, ce qui améliore les performances et réduit la surcharge de la mémoire de la fonction.

Support de la plate-forme Daydream

4.12

Daydream de Google est une nouvelle plate-forme VR pour mobiles. UE4 inclut un support complet pour cette plate-forme, y compris le contrôleur de mouvement.



Pour activer le support, allez sur Plugins dans l'éditeur, et activer "Google VR" et "Controller Google VR."

Le code reste portable et peut-être utilisé avec le Vive, l'oculus, le Gear VR ou d'autres supports.

Si vous implémentez des fonctions Daydream et que votre téléphone ne les prend pas en compte, il n'y aura donc pas pour autant de plantage: juste un perte de confort et d'optimisation.

Support OSVR

4.12

UE4 supporte maintenant pleinement OSVR, la plate-forme de réalité Open Source qui est au cœur du casque de VR de Razer.

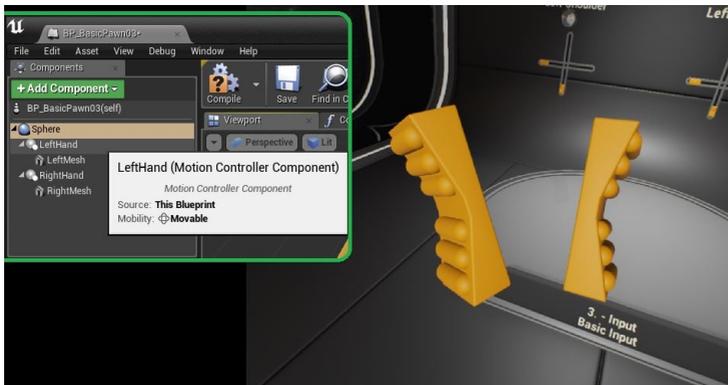
Pour plus d'infos: <http://www.osvr.org/>



VR Motion Controller Support

VR Motion Controller Support

4.9



Les motion controller components sont maintenant accessibles par une interface commune.

Le controller HTC Vive est mieux supporté (comme le Mouvement PlayStation).

Les boutons et triggers peuvent maintenant être acheminés à un seul joueur à partir de contrôleurs multiples utilisant l'abstraction Motion

Controller key.

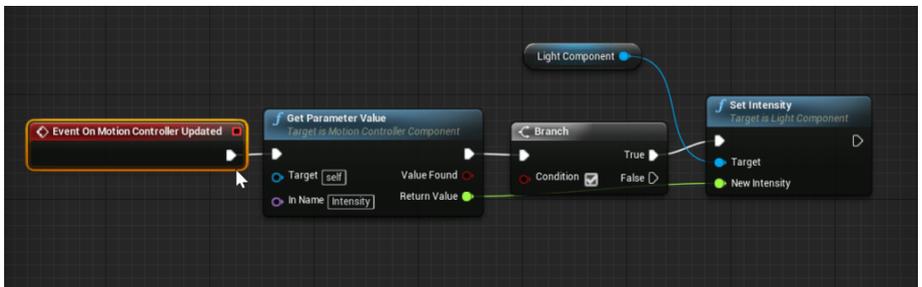
De plus, **vous pouvez ajouter le motion tracking** (recherche de mouvement) à **votre projet en ajoutant simplement un composant motion controller au personnage**, pour mettre à jour automatiquement la position de tout ce qui est rattaché au controller!

Live Link Motion Controller Assistance

4.19

Live Link peut maintenant être utilisé avec les **Motion Controllers**. La position du composant Motion Controller est gouvernée par la première transformation du sujet sélectionné dans Live Link. Il faut dériver un nouveau Blueprint à partir de MotionControllerComponent et de surcharger la fonction OnMotionControllerUpdated.

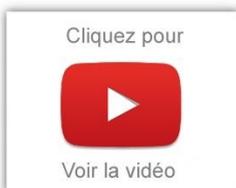
Voici un exemple de façon possible de piloter une intensité de composants légers à partir d'un paramètre Intensité dans Live Link:



Visualisation des composants de Motion Controller

4.19

Les contrôleurs de mouvement ont maintenant une nouvelle visualisation afin de leur ajouter rapidement un modèle d'affichage Static Mesh.

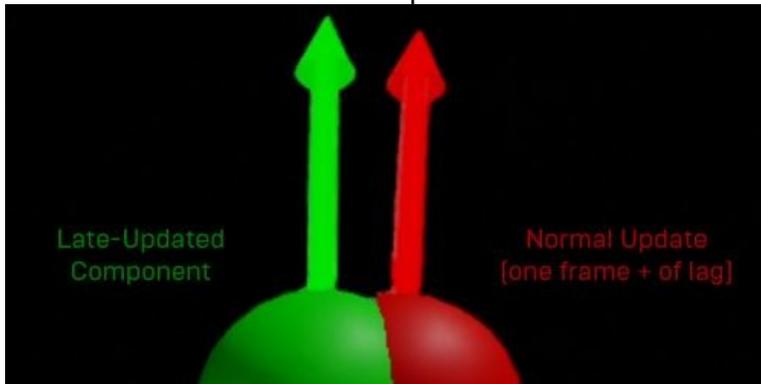


Par défaut, le système tente de charger un modèle de maillage statique correspondant au périphérique qui pilote le Motion Controller.

Low-latency VR Motion Controllers

4.10

Mise à jour des composants du contrôleur de mouvement. Vous pouvez voir la différence avec l'exemple ci-dessous.



C'est la technique utilisée pour **réduire la latence avec visiocasques**.

ils mettent à jour maintenant le Motion Controller Component positions deux fois par frame: une fois pendant la "input tick", et une fois juste avant le rendu.

Les contrôles sont plus réactifs et reflètent mieux ce que le joueur est en train de faire dans le monde réel. Cette fonction est activée par défaut.

Physic simulation en VR

4.16

utilisation des contrôleurs de mouvement pour interagir avec les objets.

Et le mode snapping qui permet de fixer plus facilement les objets en utilisant l'ancrage

Unreal Engine 4 – Guide des mises à jour (4.20)

Cette fonctionnalité est actuellement disponible uniquement en mode VR, mais le support sera ajouté pour le montage de bureau dans une version future.

Rendu VR

4.10

Vos jeux VR deviennent plus rapides grâce à de nouvelles optimisations de rendu pour visiocasques.

Une nouvelle “Hidden and Visible Mesh optimization” a été mise en place pour réduire la surcharge de GPU avec le rendu VR.



La “hidden area mesh” fonctionne comme une pré-passe, nous prenons un maillage représentant la zone cachée de la vue, et nous rejetons tout derrière le mesh. Pour l'étape de post-traitement, nous utilisons l'inverse de ce maillage qui représente les pixels visibles.

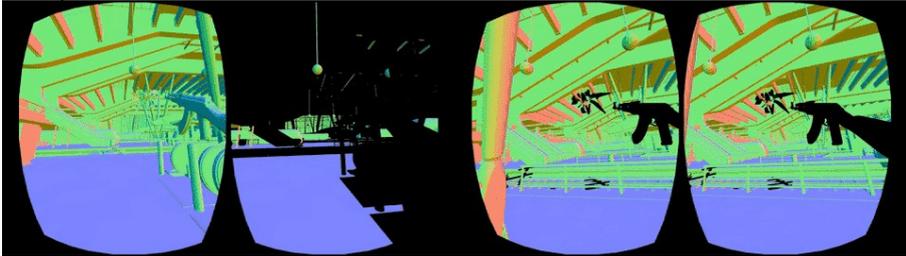
Sur “[Bullet Train](#),” nous voyons des économies d'environ 0,25 ms sur le PC quand le frame time total est de seulement 11ms. Sur les plateformes de console, vous pouvez vous attendre à voir beaucoup plus de gains de performance!

VR rendu stereo

4.11

“Instanced Stereo Rendering” est une **optimisation de rendu des images stéréoscopiques pour casques VR**, les deux yeux sont rendus en même temps.

Voici en gros comment cela fonctionne (à gauche, l’ancien, à droite, l’ISR):



L’utilisation de Bullet Train montre une amélioration de 14% sur le temps CPU, et une amélioration de 7% sur le GPU.

Pour activer cette fonctionnalité dans votre projet: paramètres de projet dans l’éditeur, et cocher la case “Instanced Stereo”.



Améliorations VR head mounted display camera

4.11

Le composant de la caméra active est compensé dans le moteur de la même manière que le casque dans le monde réel est décalé par rapport à son origine.

Vous pouvez facilement calculer la position du casque VR dans votre monde, joindre des mailles et d’autres objets directement, et de simplifier le système de contrôle de votre jeu VR.

Dans le twitch, ils appelaient cela [“Camera Refactor”](#).

VR stereo layers

4.11

Cela vous permet de **faire une interface utilisateur plus lisible et moins déformée**.

Actuellement, cette fonctionnalité est implémentée uniquement pour le casque Oculus Rift, mais viendra bientôt à d'autres plates-formes!

Un [twitch a été réalisé spécifiquement](#) pour les nouveautés VR de la 4.11.

VR chargement de films

4.14

Le moteur prend désormais en charge les films sur Oculus, GearVR, SteamVR et PSVR. Pour utiliser l'écran de démarrage, vous pouvez définir une texture en utilisant le noeud "Splash Screen Set" ou choisir de voir automatiquement apparaître une carte avec le noeud "Activer Auto Loading Splash Screen".

Support de [VRWorks 2.1](#): VR SLI, Lens Matched Shading (Pascal), Multi Res Shading (Maxwell) et Single Pass Stereo sont aussi intégrées et permettent de gagner pas mal de performances (jusqu'à 30% rien qu'avec le LMS/MRS).

Champ éloigné en vision monoscopique pour mobile VR

4.15

peut fournir un gain important de performances que de rendre les objets éloignés une fois. La divergence stéréo diminue à mesure que les objets sont plus éloignés de la caméra. A une certaine distance, le rendu stéréo des objets éloignés est indiscernable d'un rendu de monoscopique. D'où cette optimisation qu'on aurait aimé réalisée plus tôt tant cela parait évident, mais bon !

Unreal Engine 4 – Guide des mises à jour (4.20)

C'est pris en charge **uniquement sur les plates-formes mobiles** pour l'instant.

[Pour l'activer](#), sélectionnez la case à cocher dans Project Settings -> Rendering -> VR. Les multi-vues mobile et le champ lointain monoscopique ne sont pas supportés simultanément et mobile HDR doit être désactivé.

Amélioration du rendu VR sur mobile

MultiView Direct est maintenant pris en charge pour Samsung Gear VR, avec **amélioration de la performance globale** et réduction de l'utilisation de la mémoire.

Monoscopic far field peut maintenant être utilisé avec MultiView activé sur les engins VR assurant un rendu stéréo de votre scène optimal.

Google Daydream supporte le MultiView mobile standard avec un support direct dans une version future.

Editeur VR

4.12

L'éditeur VR est disponible pour vous permettre de **naviguer librement dans la scène en VR grâce aux contrôleurs de mouvement tout en éditant.**

Vous pouvez sélectionner et déplacer les objets, en placer de nouveaux et même accéder à l'interface utilisateur de l'éditeur, tout en étant immergé dans la VR.

La bibliothèque de contenus est aussi accessible, avec les outils de placement précis.



Cette fonction nécessite actuellement soit un Vive HTC ou un Oculus touch comme matériel de développement. Pour l'essayer, il suffit de mettre sur ON "Enable VR Editing" dans la section expérimentale des Préférences de l'éditeur.

Unreal Engine 4 – Guide des mises à jour (4.20)

Cette fonction est encore expérimentale mais offre déjà de belles possibilités.

Personnellement, je l'ai testée et je trouve que c'est assez pratique de pouvoir naviguer en VR dans son level, de déplacer tel ou tel objet... bref, pour composer une scène c'est génial. Mais dès qu'il faut interagir avec une fenêtre comme la fenêtre "Details" par exemple, ça devient l'horreur car il faut se coller à la fenêtre pour arriver à lire. C'est que la police d'écriture, la taille de cette dernière sur les écrans de VR est quasiment illisible.. Bref, c'est un peu comme un menu desktop qu'il faudrait adapter à un mobile. Attendons que ces fenêtres soient redéveloppées pour la VR et là on aura véritablement un outil très intéressant. Il faut laisser le temps au temps...

[Pour en savoir plus, c'est ici.](#)

4.13

Les jeux VR peuvent maintenant utiliser les ombres de type *capsule*, et l'instanciation stéréo a été optimisée (encore une fois).

Vous pouvez entrer et sortir du mode édition VR en mettant ou retirant le casque - c'est un gain de temps appréciable pour le développeur!



Il est possible de jouer tout de suite en activant "Enable VR Editing" dans la section des Préférences Editeur. Appuyez sur le bouton "Play" du menu rapide pour commencer à jouer en VR! Pour revenir à

Unreal Engine 4 – Guide des mises à jour (4.20)

l'éditeur de VR, maintenez les boutons de poignée sur les deux contrôleurs et pressez les deux boutons de déclenchement.

Peindre directement en VR

4.13

Vous pouvez maintenant peindre sur des textures ou maillage des sommets en utilisant l'éditeur de VR.



Cela permet d'utiliser des contrôleurs de mouvement pour peindre en étant immergé dans VR. Pour utiliser cette fonction, ouvrez la fenêtre "Modes" en VR, puis cliquez sur l'onglet "Mesh Paint". Maintenant, sélectionnez un objet dans le monde, puis visez et peignez! La sensibilité à la pression sur la gâchette du contrôleur est prise en charge et vous pouvez utiliser le bouton «Modifier» pour effacer.

Unreal Engine 4 – Guide des mises à jour (4.20)

Bon, j'avoue que ça me paraît gadget, mais cela va dans le sens de la création d'un niveau en mode VR - donc au final, c'est top de pouvoir avoir cette possibilité.

La mise en place d'arbres en automatique en VR

4.13

Vous pouvez utiliser les contrôleurs de mouvement pour mettre en place les arbres dans la VR.



3

Sélectionnez un type d'arbres, visez et appuyez sur la gâchette pour les installer dans le monde, avec le bouton de modification, vous pouvez effacer les arbres. Comme pour la peinture, la sensibilité à la

Unreal Engine 4 – Guide des mises à jour (4.20)

pression est prise en charge. En outre, l'éditeur **Foliage** a été amélioré.

Color picker en VR

4.13

La fenêtre Color Picker est maintenant disponible en VR, de sorte que vous pouvez modifier les propriétés de couleur sur les lumières et les autres assets du niveau.

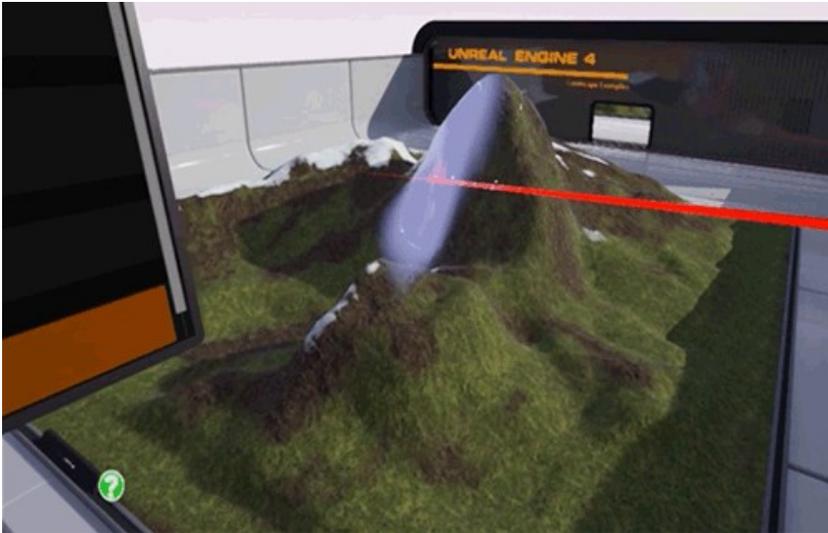


Coté VR, un petit gadget rigolo: le VR EDITOR FLASHLIGHT - ben oui, en mode éditeur, si c'est trop sombre et que vous auriez bien besoin d'une torche... elle existe maintenant !

Editeur de paysage en VR

4.14

Vous pouvez maintenant créer et sculpter des matériaux de paysage terrain et de peinture à l'aide des contrôleurs de mouvement en VR!



Menu VR editor amélioré

4.15

Les menu rapide et menu radial dans l'éditeur de VR ont [été mis à jour](#) avec un nouveau look et pour améliorer la convivialité et la fonctionnalité.

Cliquez sur le touchpad pour appeler / fermer le menu radial, sélectionnez les options avec une direction sur le touchpad, puis sélectionnez avec la gâchette. Vous pouvez également [sélectionner des éléments](#) avec le laser, mais le pavé tactile remplace le laser.

VR Editor Numpad

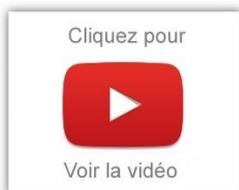
4.15

entrer rapidement des chiffres (0-9), ainsi que des valeurs négatives et décimales dans les champs de texte dans l'éditeur de VR en utilisant la nouvelle interface. Utiliser le bouton de déclenchement pour sélectionner les numéros, la touche de modification pour revenir en arrière. **Pour retirer le pavé numérique**, cliquer ailleurs dans la scène, ou sur le pavé tactile, tout comme pour le menu radial standard.



Support écran VR Spectator pour Oculus, Vive et amélioration pour PSVR

4.17



Le Spectator Screen est la sortie "connectée à un appareil de VR, souvent une télé ou un écran, permettant à d'autres personnes de voir ce que le joueur voit dans son casque... mais sans la stéréo ! Ainsi, quand un utilisateur utilise le HMD, d'autres personnes peuvent observer ou d'interagir avec l'expérience.

Unreal Engine 4 – Guide des mises à jour (4.20)

Vous pouvez maintenant dessiner n'importe quelle texture, y compris des rendus de cibles ou des interfaces utilisateur, à l'écran Spectator, en plus ou à la place d'une vue de ce qui est affiché en VR. Un système simple de mise en page est accessible aux Blueprints. La plupart des modes de débogage sont maintenant disponibles sur toutes les plateformes.

Pour plus d'informations sur l'ajout du support VR Spectator à votre projet, consultez la doc [Virtual Reality Spectator](#) Screen.

4.16



Les **interfaces de travail** pour la VR (tout est maintenant accessible à partir d'un menu radial, photo ci-dessous), l'animation, Sequencer,

Unreal Engine 4 – Guide des mises à jour (4.20)

maintenant disponible en VR, sont améliorées. Bienvenue donc aux séquences cinématiques en VR!

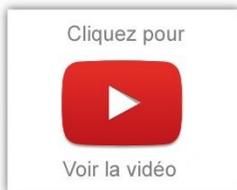
Teleport a été mis à jour afin que vous puissiez instantanément passer d'un emplacement à un autre et redimensionner pour voir ce que verra le joueur.

C'est vraiment top. L'interface est bien plus intuitive et ressemble à ce qu'on peut trouver déjà dans certains jeux.

Quel aisance de pouvoir utiliser le sequencer de cette façon, en se déplaçant dans la scène, en arrêtant de le temps, en procédant à des manipulations directement en 3D...[regardez ça](#).

VR mode camera Previews

4.17

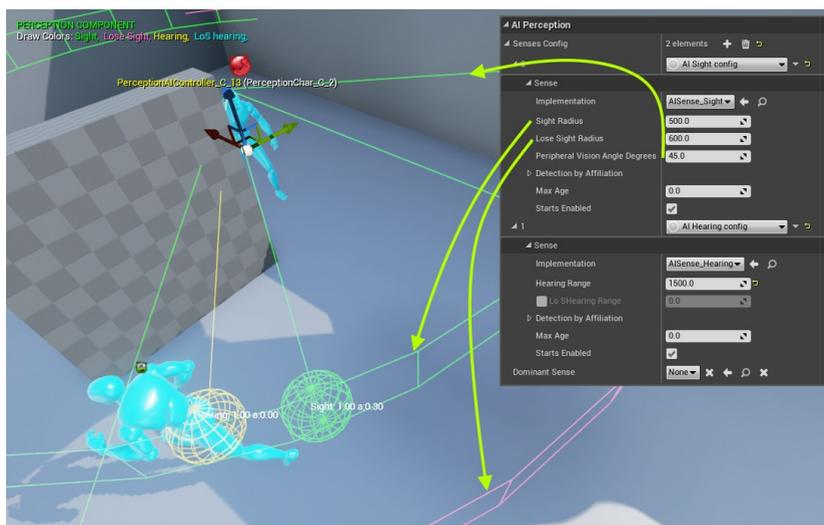


Pour vous aider à faire du level design en mode VR, c'est à dire en immersion directe, un mode preview de caméra a été mis en place sous l'éditeur. La sélection d'une caméra dans une scène ou d'un actor qui contient une caméra comme un personnage, vous donnera automatiquement un aperçu dans le monde de ce que la caméra voit. Vous pouvez épingler cet aperçu comme l'aperçu de la caméra de bureau, et déplacer le panel dans le monde.

INTELLIGENCE ARTIFICIELLE

AI Perception System: Blueprint Scripting

4.7



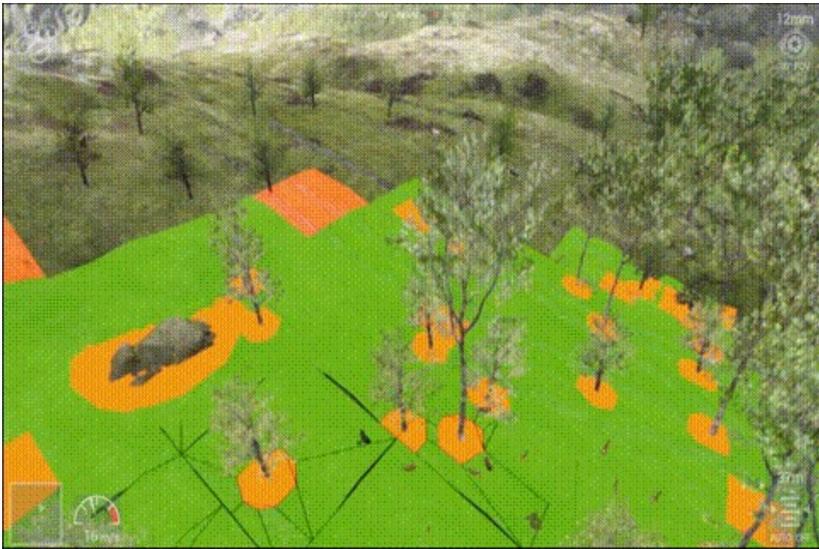
Le système **AI Perception** faisant partie normalement du pawn sensing est désormais **configurable via les Blueprints**.

Dynamic Navigation Mesh pour mondes ouverts

4.8

Tout nouveau système pour supporter les NavMeshes dynamiques dans les grands mondes ouverts.

Le moteur génère désormais dynamiquement le NavMesh à la demande. Il suffit de configurer le système de navigation du projet pour ce mode et d'ajouter Navigation Invoker Component pour les acteurs concernés.



Pour la génération procédurale de mondes en Open World, c'est vraiment une fonctionnalité attendue.

J'en profite pour ajouter que coté navmesh et pathfinding, il y a une sacré liste de mises à jour et de corrections. Si c'est un point qui vous intéresse particulièrement, allez voir la rubrique "navigation" de [cette page](#).

AI: Support Blueprint Support pour EQS

4.8

Utiliser l'Environment Querying System (EQS) des Blueprints. En exécutant une requête, on obtient une 'query instance', qui se déclenche quand la requête est terminée. Il y a également des fonctions pour accéder aux résultats de la requête.

Améliorations globales

4.11

De nouvelles fonctionnalités liées à l'EQS ([Environment Query System](#)), aux Behaviour Trees, à la navigation... et de nombreux bugs en moins surtout !

J'ai vu également un Blueprint de type controller pour la gestion des foules (**crowd**). Cela semble prometteur, même si la gestion de troupes demanderait une meilleure gestion du déplacement en formation.

Enfin, il y a un plugin qui est disponible (AI Character Template ou quelque chose comme cela) qui devrait permettre de paramétrer plus facilement des PNJ quand on part de zéro, mais je n'ai pas encore trouvé ce qui se cache derrière ce plugin. A vos plumes si vous savez !

MOBILES, CONSOLES ET HTML5

Pour la partie réalité virtuelle et augmentée des mobiles, consultez les chapitres respectifs.

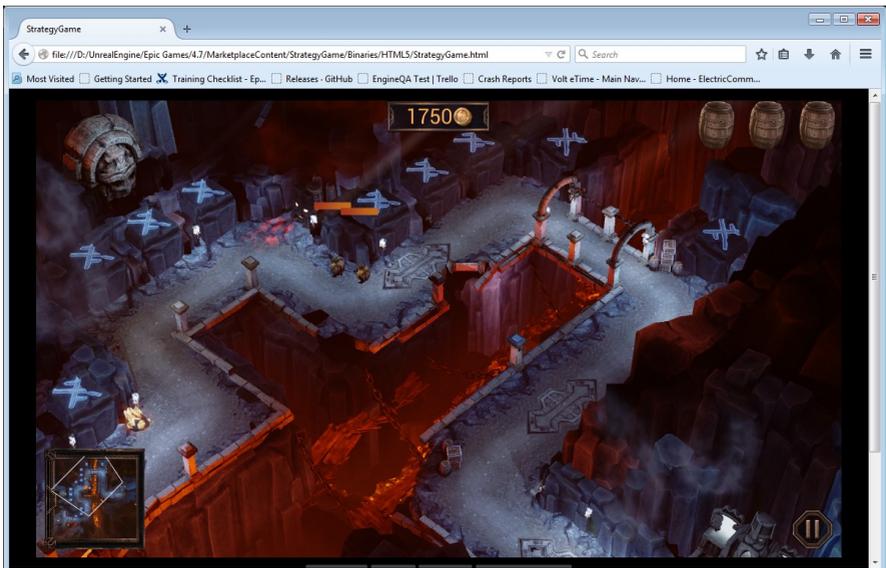
HTML5/WebGL

Support Expérimental (Windows seulement)

4.7

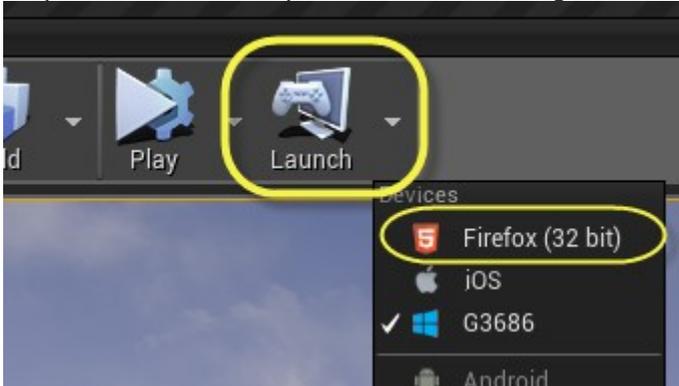
C'est, pour moi, une des fonctions les plus importantes. [Je n'avais pas réussi à faire fonctionner le bouzin](#) avec la version 4.6.1, même en recompilant l'éditeur et en suivant à la lettre la procédure.

Maintenant, je constate que c'est réglé - c'est directement accessible dans l'éditeur : je n'ai eu qu'à installer le [Emscripten SDK](#), et fermer et relancer l'éditeur, rien de plus !



Unreal Engine 4 – Guide des mises à jour (4.20)

Ensuite, on peut même voir la preview dans le navigateur :



Bon, pour le développement, il faut avoir un navigateur compatible: Firefox, Safari, ou Chrome 64-bit. L'export de mon jeu a pris pas mal de temps (plus d'une heure, j'ai l'ai retrouvé en me levant ce matin), et certaines textures ne passent pas. Par contre, les prochains exports seront plus rapides.

Amélioration de HTML5

4.9

Vous ne devez plus installer d'autres supports pour commencer le développement sous HTML5 et empacker votre jeu pour un navigateur web!

Pour le package, il suffit de sélectionner HTML5 dans le menu Fichier -> Package.

Les autres améliorations comprennent:

- Un petit serveur Web est inclus avec le jeu pour manipuler des fichiers compressés pour les tests en local.
- HTML5 networking est désormais pris en charge!
- Soutient de l'upload Amazon S3. Une fois les informations d'identification pour AWS remplis, l'appli est uploadée sur S3 et le jeu accessible sur S3.

4.18

WebAssembly est désormais activé par défaut pour les projets HTML 5, permettant de créer des jeux complexes pour le Web, qui se chargent rapidement et fonctionnent correctement (Euh....).
A noter que le multithreading n'est pas encore disponible sur WebAssembly.

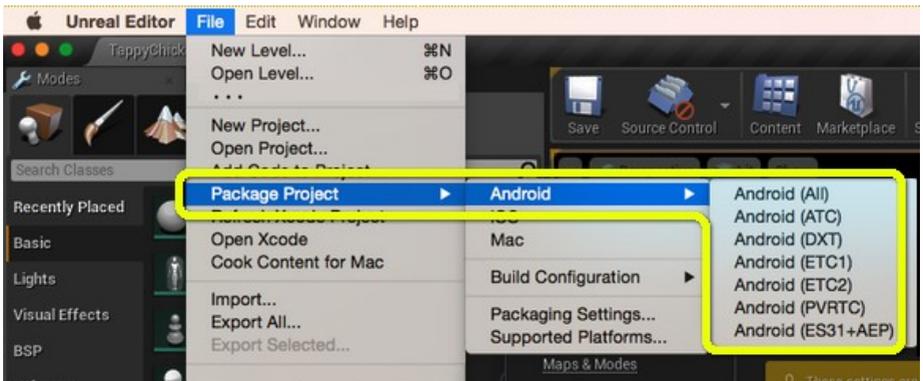
Android

Développement Android sur Mac

4.7

La **version OS X** d'Unreal Engine supporte désormais pleinement le développement sous **Android** ! Dommage que l'ouverture soit toujours dans un seul sens. Essayez de développer pour iOS avec un PC! Il vous faudra quand même un mac au final!

Les utilisateurs de Mac peuvent builder et cooker les packages pour Android, tout comme les déployer sur les machines à partir de l'éditeur ou en ligne de commande.



Unreal Engine 4 – Guide des mises à jour (4.20)

- installer [Java for OS X](#) et l'**Android SDK** (qui peut être installé en utilisant le TADP installer : Engine/Extras/Android).
- Après installation, on relance l'éditeur et on configure les chemins du SDK dans la nouvelle section "Android SDK" de "Project Settings".

Desktop-Quality Rendering on Android (OpenGL ES 3.1 w/ AEP)

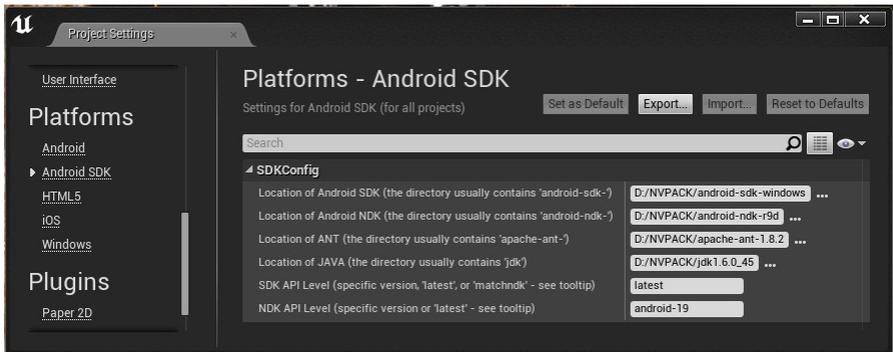
4.7

Une version bêta d'OpenGL ES 3.1 (combinée à Android Extension Pack) **permet d'obtenir une qualité de rendu proche de celui qu'on a sur son desktop!**

Il faut sélectionner "ES3.1+AEP" comme target au moment du building et cooking.

Recompiler l'éditeur pour bénéficier des fonctionnalités.

Unreal Engine 4 – Guide des mises à jour (4.20)

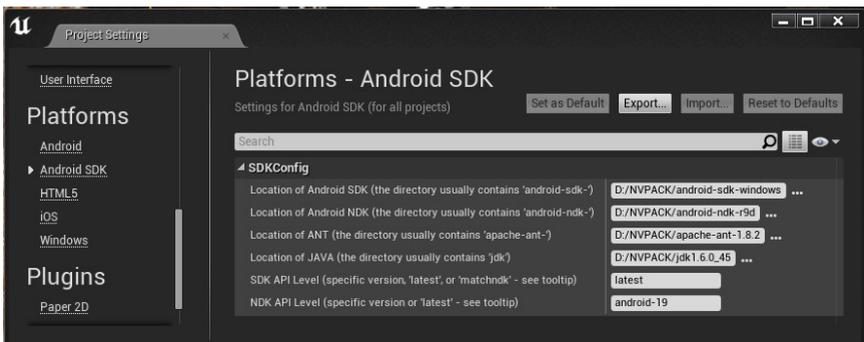


Loading movies et startup movies fonctionne désormais pour Android (à partir de la version 4).

4.7

A noter que l'appareil Android doit pouvoir lire des vidéos au format MP4.

A configurer dans les Project's Settings



Les paths vers le SDK peuvent aussi être configurés directement dans l'éditeur (plus la peine de passer par des variables d'environnement).

Support de Contenu téléchargeable (DLC)

4.8

Pour créer des fichiers Pak qui peuvent être distribués aux utilisateurs plus tard...

Support de contenu pour les patches

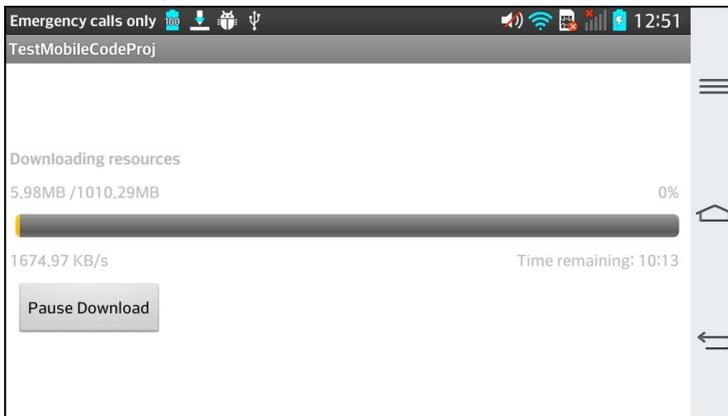
4.8

Pour générer des **fichiers de correctifs** avec seulement le contenu des fichiers changés depuis une version précédente! (un Diff quoi!)

Support pour les grands Jeux Android

4.8

Support pour le téléchargement .obb des fichiers du Google Play Store. A bon entendre...



Manettes Gamepad pour Android

4.10

Plusieurs manettes sont maintenant supportées sur Android. Les manettes suivantes sont automatiquement détectées.

- Amazon Fire Game Controller and Amazon Fire TV remote
- NVIDIA SHIELD Controller and SHIELD Portable
- Samsung Game Pad EI-GP20

D'autres vont s'adapter avec les correspondances génériques.

Autres nouvelles fonctionnalités pour les contrôleurs Android:

- Mappings maintenant détectés automatiquement suivant le nom du dispositif de commande
- Plusieurs manettes désormais prises en charge simultanément
- Nous nous occupons maintenant à la fois des mappings Z / RZ et RX / RY des axes de joystick droit
- Gâchette gauche / Frein et gâchette droite axes / accélération fonctionnent correctement
- Commutateur Hat X / Y permet maintenant de diriger



Appuyez sur un bouton sur une manette non affecté associe l'ID du périphérique avec le premier numéro du contrôleur non assigné. Le

Unreal Engine 4 – Guide des mises à jour (4.20)

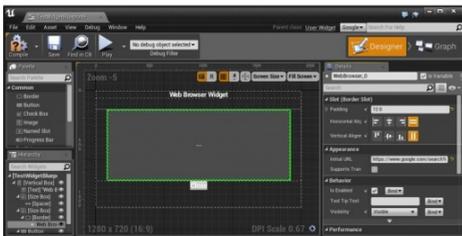
Gamepad peut également être utilisé avec 3 nouveaux Blueprint nodes.

Widget Web Browser sur Android

4.10

Android **prend désormais en charge les widgets Navigateur Web** créés dans un Widget Blueprint ou en C++ en utilisant Slate.

Ceci peut être utilisé pour afficher un contenu hébergé n'importe où sur le web, par exemple de nouveaux messages sur le texte d'aide.



Sur Android, le contrôle web natif de la plate-forme est utilisé alors il n'y a pas d'augmentation de la taille de l'exécutable.

Le Web Browser widget sera porté sur iOS dans une version future.

Test automation for android

4.13

Le lanceur de projet peut packager et lancer votre projet sur plusieurs appareils Android simultanément. L'application en cours d'exécution sur chaque appareil communiquera au PC hôte via le câble USB et apparaîtra dans la fenêtre Session Frontend.

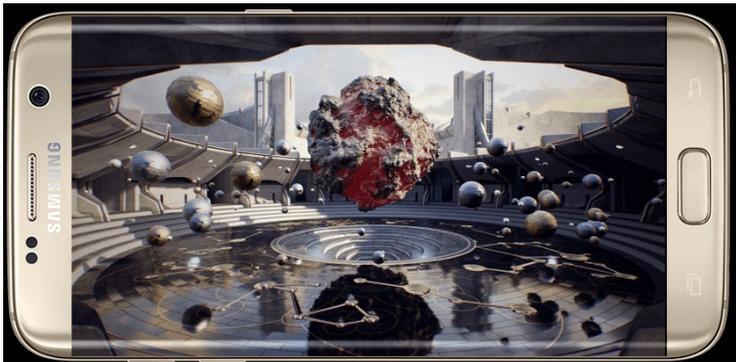
Vulkan

4.14

Vulkan est maintenant géré sur les plateformes Android

Cela était déjà possible avec quelques manipulations sous Windows par exemple, mais avec la version 14, c'est intégré pour Android.

- UE4 supporte les appareils sous Android 7 (Nougat) avec pilotes Vulkan, ainsi que le Galaxy S7 Samsung avec mise à jour OTA récente.
- De nombreux problèmes de rendu ont été corrigés.
- Le rendu passera automatiquement à OpenGL ES s'il est lancé sur un appareil Android non compatible Vulkan.



Unreal Engine 4 – Guide des mises à jour (4.20)



Vous pouvez alors lancer des tests automatisés sur tous les appareils et voir les résultats de la session Frontend.

Support Android sur linux

4.14

Codeworks pour Android de NVIDIA pour Linux est la meilleure façon de mettre en place les outils de NDK et SDK nécessaires. En outre, OpenJDK 1.8 doit être installé; set JAVA_HOME pointer vers votre installation.

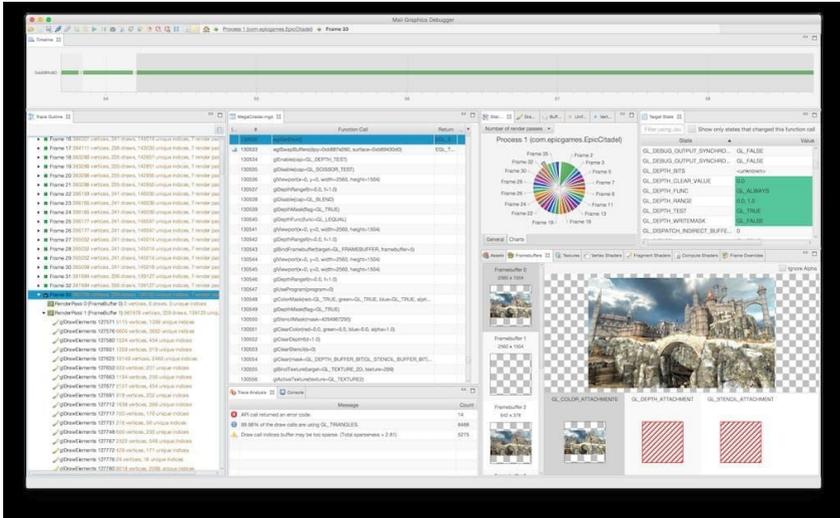
Vulkan Android sur Linux n'est pas encore pris en charge.

Android graphics debuggeur

4.15

les applications Android peuvent être débuggées avec le [Mali Graphics Debugger](#) ou [Adreno Profiler](#) selon le GPU de l'appareil. Les options de débogueur Graphics peuvent être trouvées dans Project Settings -> Android. Vous devez d'abord télécharger et installer ces débogueurs sur le site Web du fournisseur de GPU.

Unreal Engine 4 – Guide des mises à jour (4.20)

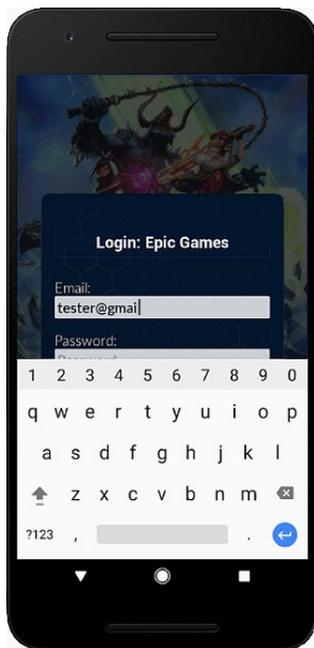


Virtual Keyboards

4.16

Sur mobile, le **Virtual Keyboards** (clavier virtuel) **Android** est pris en charge. La **taille des exécutables a été réduite** pour les applications mobiles! (Early Access Preview). Ouf... car ça me gonflait vraiment de voir ces gros APK traîner sans raison...

L'utilisation du clavier virtuel est activé en cochant la case à cocher sous Project Settings > Platforms > Android > APK Packaging.

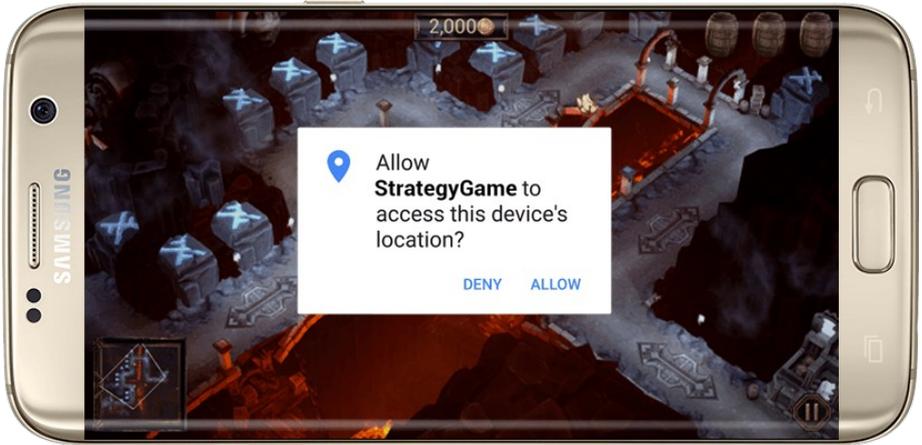


Support for Runtime Permissions on Android

4.16

Unreal Engine 4 prend désormais en charge les [autorisations](#) d'exécution requis par Android 23. Ceux-ci incluent l'accès aux contacts, les photos, l'état du téléphone, stockage externe, appareil photo, et les services de localisation. Ainsi, on se rapproche du mode habituel d'une application Android:

Unreal Engine 4 – Guide des mises à jour (4.20)



4.16 nécessite Android SDK 23 ou +, vous pouvez trouver le programme d'installation CodeWorksforAndroid-1R6u1 dans votre répertoire Engine/Extras/AndroidWorks.

Android Toolchain et SDK mis à jour

4.17

Android toolchain supporte maintenant les applications NDK 14b et NDK 15 en expérimental. Google Play Services a été mis à niveau vers 9.8.0 ainsi que les bibliothèques de support Android nécessaires.

A partir de cette version Gradle est pris en charge comme option expérimentale. Cette [option est activée](#) par projet dans les paramètres du projet Android avec la case à cocher «Enable Gradle instead of Ant [Experimental]».

Réduction de la taille de l'exé Mobile sur iOS et Android

4.17

Unreal Engine 4 – Guide des mises à jour (4.20)

Pour Android, on utilise désormais le drapeau -ICF qui permet d'économiser environ 5 Mo non compressés

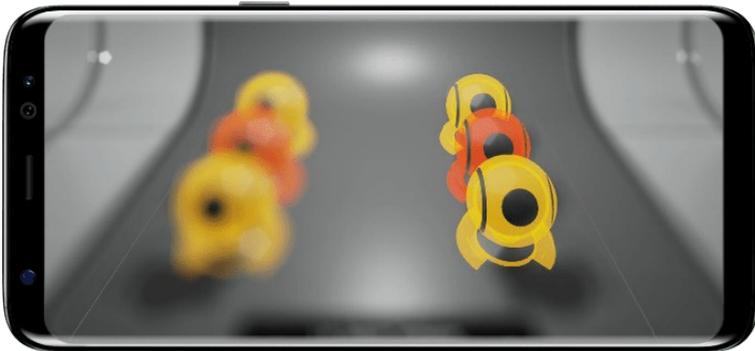
Pour iOS, des exceptions pour ObjectiveC par défaut sont désactivées entraînant une économie d'environ 6,5 Mo de la taille de l'exécutable. Il est aussi possible de désactiver le code inline pour iOS builds.

Mobile Separate Translucency

4.17

Les objets translucides marqués comme «Mobile Separate Translucency» seront rendus séparément après le Blom et la profondeur de champ. Cette option se trouve dans la Translucency section de l'éditeur de matériaux.

La translucidité séparée nécessite que l'option MobileHDR soit activée et MSAA désactivée, sinon elle sera rendue comme translucidité normale.



sRGB sur Android haut de gamme

4.17

Unreal Engine 4 – Guide des mises à jour (4.20)

Les applications Android conçues avec OpenGL ES3.1 ou Vulkan utiliseront désormais un échantillonnage sRGB natif au lieu de l'imiter --> meilleure résolution des couleurs de perception des textures sur les appareils.

Android Aspect Ratio Support ultra-grand écran

4.17



Unreal Engine 4,17 prend en charge des ratios ultra-large, pour supporter des périphériques grand écran tels que le Samsung Galaxy S8.

Par défaut, le ratio maximal supporté est de 2,1 (largeur/ hauteur). Le format par défaut peut être modifié dans la section Project Settings .

Ajout du support audio Unreal sur de nouvelles plateformes (Early Access)

4.17

Le nouveau système audio Unreal est supporté sur Android, avec l'amélioration des performances et des solutions pour le stuttering. Il y a aussi support pour PlayStation 4, Xbox One, iOS et Mac OS.

Gradle Android

4.18

Gradle activé par défaut pour Android pour l'emballage des APK Android.

Plugin de caméra Android

4.18

Unreal Engine a maintenant un plugin optionnel pour donner accès aux caméras avant et arrière sur les appareils Android. L'activation du plug-in ajoute un nouveau panneau de Paramètres du projet. Top !



iOS

Faire des projets iOS sur Windows

4.8

Support pour faire des projets en code iOS en utilisant SSH.

Exigences:

- Connexion à un Mac avec Xcode installé
- Certificat de signature iOS dans le trousseau système
- DeltaCopy ou autre logiciel SSH et Rsync pour Windows

Révision du projet Xcode

4.10

Epic a amélioré la façon dont les projets Xcode sont générés pour votre jeu!

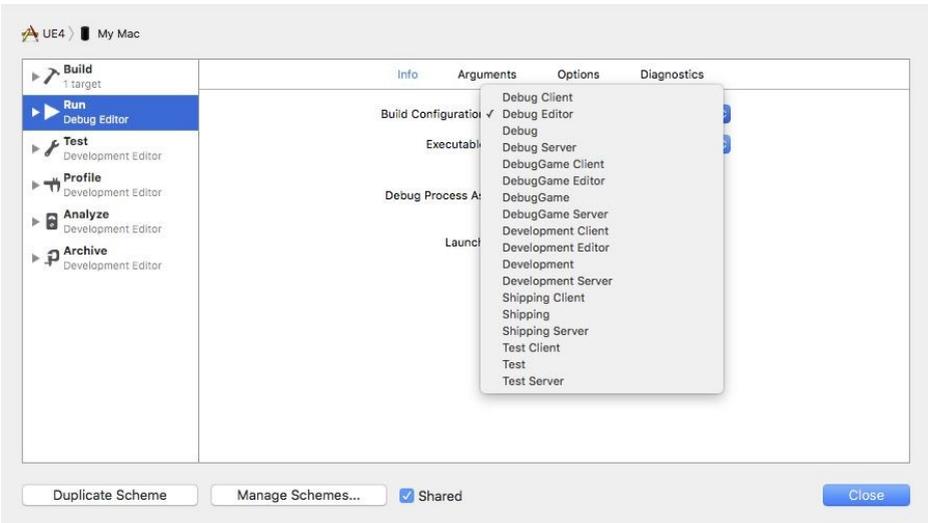
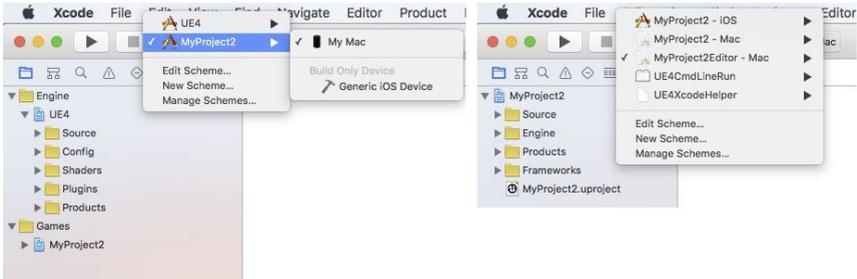
Vous avez maintenant **accès en direct au Xcode et aux caractéristiques “Fix it”**, et l’indexation est beaucoup plus rapide.

Il est plus facile de basculer entre les configurations!

Au lieu d’un projet contenant plusieurs objectifs, nous avons maintenant un espace de travail avec plusieurs projets. Vous ouvrez maintenant UE4.xcworkspace au lieu de UE4.xcodeproj dans Xcode. Chaque projet dans l’espace de travail crée un schéma unique utilisé pour construire diverses cibles de UE4.

Sélection des cibles UE4 dans Xcode:

Unreal Engine 4 – Guide des mises à jour (4.20)



Xcode choisit automatiquement la bonne application quand vous changez de configuration. (changer de configuration ne met pas à jour l'exécutable dans l'éditeur). Il utilisera toutefois la bonne application lorsque vous exécuterez à partir de Xcode.

Prise en charge améliorée de Metal sur macOS et iOS

4.15

[le support](#) permet d'utiliser un grand nombre de nouvelles API et les fonctionnalités de langage shader ajoutées par Apple sur mac OS 10.12 Sierra et iOS 10.

Unreal Engine 4 – Guide des mises à jour (4.20)

Mac et iOS prennent désormais en charge la version standard Metal Shader comme paramètre de projet, permettant aux développeurs de choisir si ils souhaitent adopter de nouvelles fonctionnalités ou maintenir une compatibilité ascendante. Il y a aussi un support expérimental pour le rendu HDR sur Mac avec un affichage approprié intégré, tel que le Retina iMac, ou attaché via Thunderbolt.

Streaming audio sous ios

4.15

le Streaming audio peut réduire à la fois l'utilisation de la mémoire et les temps de chargement de niveau. Les assets déjà marqués pour le streaming seront automatiquement diffusés sur d'autres plates-formes.

Notifications à distance pour ios

4.15

Le jeu peut enregistrer des callbacks pour gérer la notification. L'éditeur fournit des outils pour mettre en place des notifications pour l'application.

Simplification des certificats et signatures sur iOS

4.18

Il n'est plus nécessaire d'installer votre clé privée ou vos certificats à l'aide de Keychain Access sur le Mac distant.

Desktop-Class Forward Renderer sur iOS (expérimental)

4.18

Unreal Engine 4 – Guide des mises à jour (4.20)

Unreal Engine 4.18 contient le Desktop-Class Forward Renderer utilisé dans la démo de Wingnut AR à la WWDC 2017! (appareils iOS dotés d'un processeur A9 ou version ultérieure exécutant iOS 10.3 ou version ultérieure, avec une version A10X).



signature automatique de code

4.19

La signature automatique de code (ACS) lors de l'utilisation de Xcode a été introduite. Cela fonctionne également quand il n'y a aucune disposition de mobile pré-installée sur le Mac mais le Mac doit contenir les certificats.

Tous mobiles confondus

4.9

Cette sortie contient beaucoup de fonctions pour vous aider à créer des applications mobiles de pointe.

De nouveaux compléments incluent des ombres dynamiques

Unreal Engine 4 – Guide des mises à jour (4.20)

performantes, des lumières mobiles et des traces (decals).

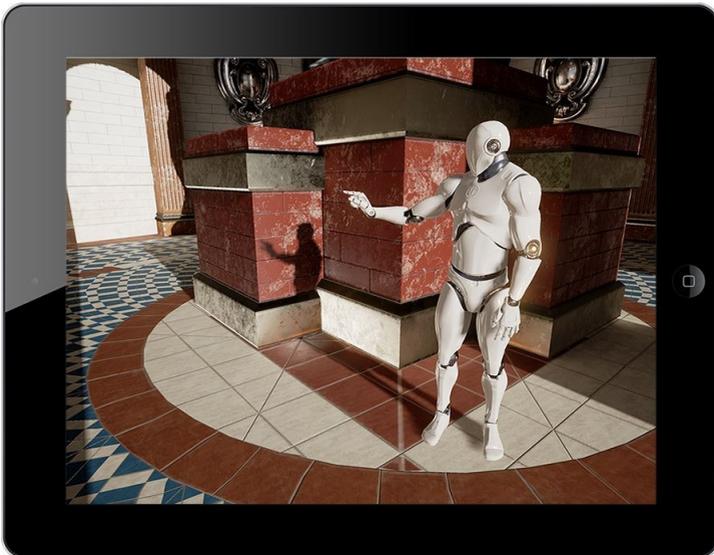
Dans cette version, la fonctionnalité d'achat iOS In-App a été améliorée et des Notifications de mise en avant et le support pour [CloudKit](#) ont été ajoutés.

Pour Android, ont été ajoutés OpenGL ES 3.1, le support de [Immersive Mode](#), la décompression audio asynchrone et de meilleures fonctions d'orientation.

[NVIDIA AndroidWorks](#) est maintenant aussi inclus.

Ombres dynamiques sur personnages pour mobiles

4.9



Des ombres modulées dynamiques du personnage à partir de lumières directionnelles sont maintenant supportées sur les dispositifs mobiles!

Unreal Engine 4 – Guide des mises à jour (4.20)

Vous pouvez utiliser des ombres modulées dynamiques même si votre scène est statiquement éclairée.

Pour permettre cette fonction, lancez l'option "Cast Modulated Shadows" sur une lumière directionnelle.

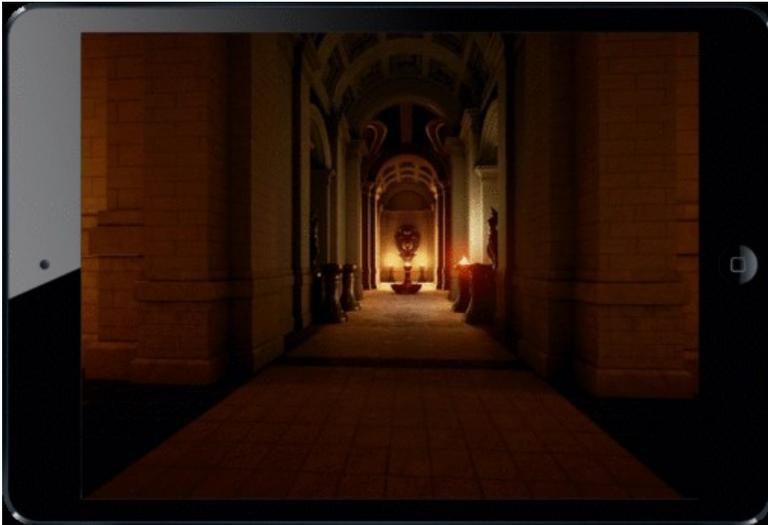
Dynamic Point Lights pour Mobile

4.9

Dynamic point lights est maintenant supporté sur appareils mobiles et HTML5.

Ils peuvent être utilisés pour **améliorer des effets visuels de courte durée** comme des explosions, des projectiles, des effets de particule, etc.

Jusqu'à 4 dynamic point light sont supportés sur chaque objet illuminé.



Traces (decals) sur mobiles

4.9



Les modes translucides, additifs et modulés sont supportés.
Note: seules les traces "non éclairées" sont supportés sur les mobiles et le web actuellement.

A screenshot of the Amazon S3 configuration screen in Unreal Engine 4. The screen is dark grey with white text. It features a toggle switch for "Upload builds to Amazon S3 when pack" and four input fields for "Amazon S3 Key ID", "Amazon S3 Secret Access Key", "Amazon S3 Bucket Name", and "Nested Folder Name".

Effets Refraction pour mobiles

4.10

Activé par défaut sur iPhone 5S et plus et appareils Android tels que le Nexus 5, Galaxy Note et Galaxy 4 S6, le mode peut aussi être activé sur d'autres appareils en réglant le "r.RefractionQuality" dans les paramètres de config.



Material Quality system

4.10

Il permet à votre jeu de **s'adapter automatiquement à la résolution et aux capacités de votre appareil**, même pour les appareils bas de gamme, et ce, sans sacrifier la qualité sur des appareils plus modernes!

Vous pouvez maintenant générer des versions de qualité faible, moyenne et haute pour tous les matériaux! A chaque étape, vous pouvez désactiver les rendering features pour augmenter les performances sur les appareils bas de gamme.

UE4 détecte automatiquement le niveau de qualité optimal par défaut pour de nombreux appareils.

Forward Rendering Overrides			
Quality Option	Low	Medium	High
Enable Quality Override	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Force Fully Rough	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Force Non-metal	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Disable Lightmap directionality	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Unreal Engine 4 – Guide des mises à jour (4.20)

Galaxy S6 Edge (matériaux de haute qualité)



Haute qualité de matériau sélectionnée automatiquement

Réflexions spéculaires, rugosité, métaux, normal maps et éclairage directionnel.

Galaxy S III (qualité des matériaux revus à la baisse):



Qualité des matériaux diminuée automatiquement pour une performance optimale.

Réflexions spéculaires, variation de la rugosité, métaux, normal maps et éclairage directionnel désactivés.

Unreal Engine 4 – Guide des mises à jour (4.20)



Mobile Provisioning Profile Selector

4.10

Les utilisateurs sont désormais en mesure de choisir un mobile et le certificat dans les paramètres pour un projet iOS.



Post traitement de haute qualité

4.12

“filmic tonemapper” fonctionne sur les appareils mobiles.

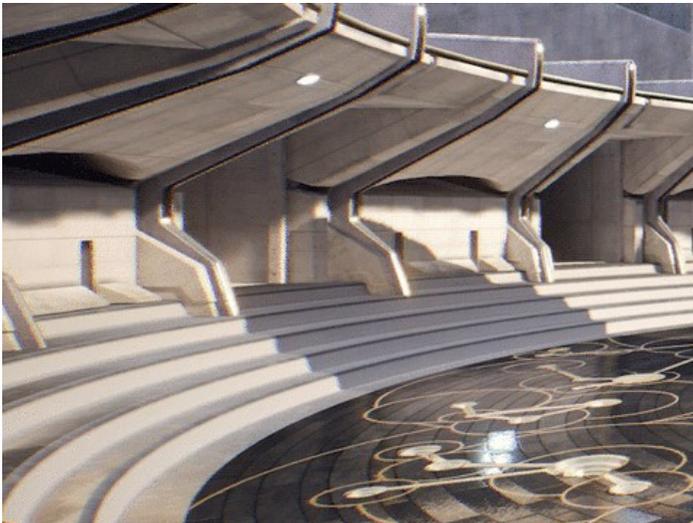
Meilleure qualité de profondeur de champ gaussien sur les appareils mobiles avec le soutien de ES3.



Amélioration des ombrages pour les mobiles

4.12

Les **maps d'ombre en cascade** pour une seule lumière fixe sont désormais prises en charge dans le moteur de rendu mobile!



La fonction utilise un texture sampler supplémentaire, mais peut être désactivée via les paramètres du projet: Rendering -> Mobile ->

Unreal Engine 4 – Guide des mises à jour (4.20)

Combined Static and CSM Shadowing qui libérera l'échantillonneur pour l'utilisation des matériaux.

Les particules

4.12

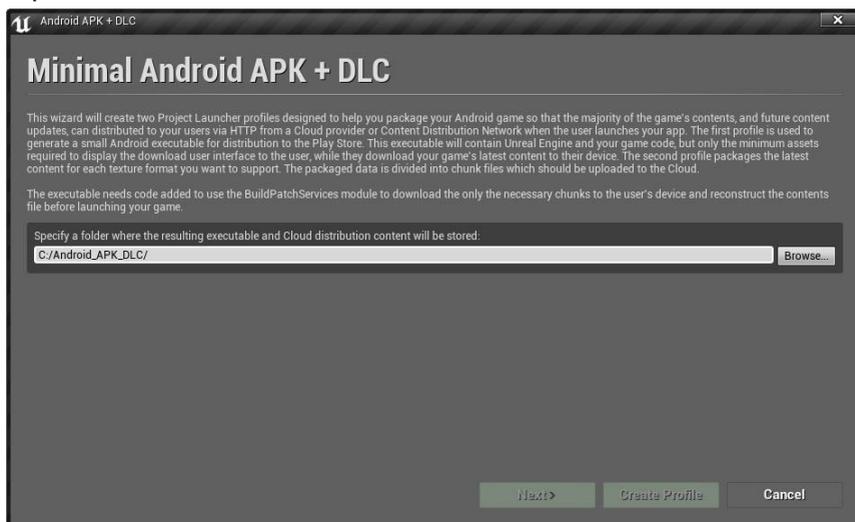
Les particules GPU avec champs de vecteurs fonctionnent pour les mobiles.

Il faut que l'appareil prenne en charge 32 bits à virgule flottante cibles de rendu avec le soutien MRT (Apple avec A8 + (iPhone 6, iPad Air 2, iPad Mini 4, et plus récent, Android 6.0 avec Adreno 4xx et Mali-T7xx GPU (Galaxy Note 4, note 5, S6, S7, Nexus 6P et autres).

Assistant packaging pour mobiles

4.13

Un Assistant packaging pour mobile a été ajouté pour aider au téléchargement sur un App Store et le reste du contenu est téléchargé à partir du Cloud.



Custom Post-Process materials

4.13

Les matériaux personnalisés Post-process peuvent désormais être utilisés sur les appareils mobiles.



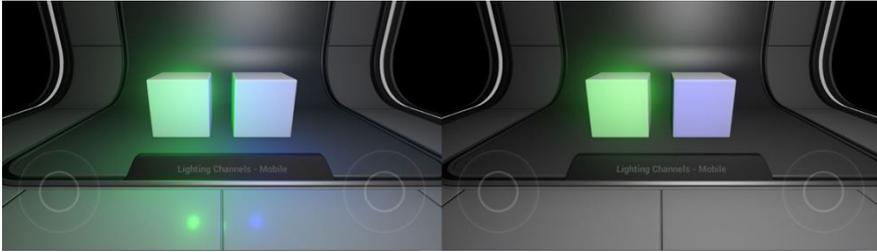
- Cette fonction nécessite que l'option «**HDR mobile**» soit activée dans les paramètres du projet et nécessite des appareils récents
- Prend en charge la récupération de PostProcessInput0 (Scène de couleur)

Lighting channels sur Mobiles

4.13

Les canaux d'éclairage fonctionnent maintenant dans le renderer mobile! Ceux-ci vous permettent d'éclairer de manière sélective des objets.

Unreal Engine 4 – Guide des mises à jour (4.20)

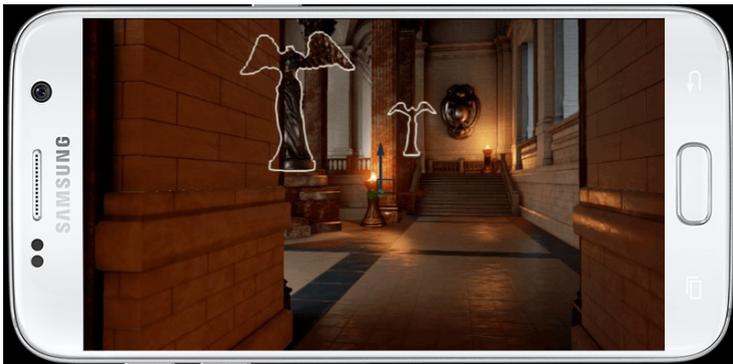


- lumières directionnelles multiples prises en charge dans différents canaux.
- Chaque primitive peut être affectée par une seule lumière directionnelle. Le premier canal d'éclairage situé sur une primitive détermine quelle lumière directionnelle aura une incidence sur elle.
- les shadow maps en cascade (CSM) de lumières directionnelles fixes ou mobiles exprimés uniquement sur des primitives avec des canaux correspondant d'éclairage.
- Les dynamic point lights soutiennent pleinement les canaux d'éclairage.

Support de profondeur personnalisée sur mobiles

4.14

Depuis Scene Depth, Custom Depth ou Scene Color.



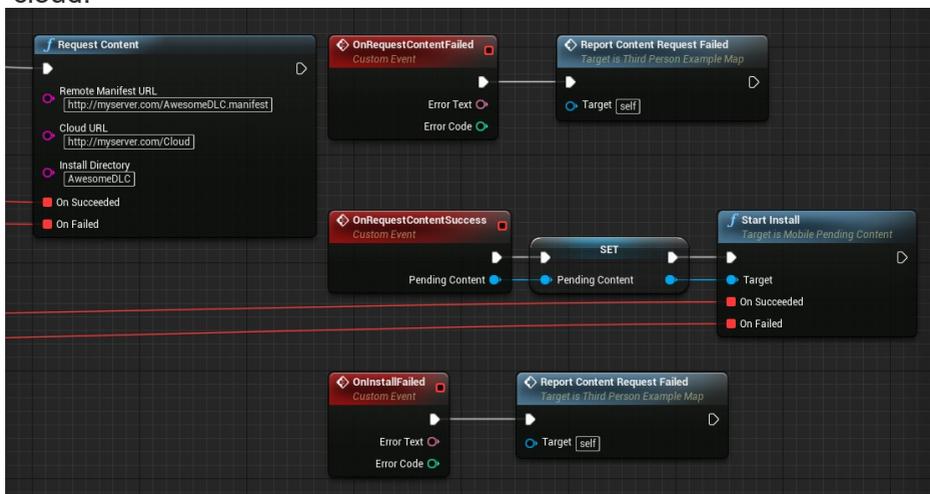
Unreal Engine 4 – Guide des mises à jour (4.20)

Comme ça nécessite des post-traitements, Mobile HDR doit être activé et la fonction ne fonctionne pas actuellement pour tout mobile avec MSAA activé.

Bibliothèque blueprint pour mobiles

4.14

La nouvelle bibliothèque Mobile Patch Utilities Blueprint library contient toutes les fonctionnalités requises pour permettre un jeu sur mobile de télécharger et installer le contenu et les correctifs à partir du cloud.



Amélioration du composant Scene Capture

4.14

Unreal Engine 4 – Guide des mises à jour (4.20)



Aperçu de rendu (opengl es 3.1, metal, [vulkan](#))

4.15

Vous pouvez obtenir un aperçu des fonctionnalités pour mobiles haut de gamme dans la fenêtre de l'éditeur en utilisant **Mobile Preview**. Ce mode émule l'ensemble des fonctionnalités disponibles pour iOS Metal, les appareils Android OpenGL ES 3.1 et Android Vulkan.



Options mobile shader de réduction de la mémoire

4.15

Pour réduire de manière significative l'utilisation de la mémoire en désactivant les configurations d'éclairage non utilisées.

Les paramètres suivants disponibles sous Project Settings -> Rendering peuvent être modifiés pour réduire l'utilisation de la mémoire et diminuer la taille du paquet app.

Service de localisation pour les données GPS sur Android et ios

4.15

L'accès aux données GPS est maintenant disponible et les Blueprint nodes sont fournis sous Services-> Mobile-> Location. Ah ben ça c'est chouette pour faire de la réalité augmentée ou toute autre application basée sur la géolocalisation.

Support pour Nvidia SLI Alternate Frame Rendering (AFR)

4.15

pour fournir des performances supérieures lorsque plusieurs cartes graphiques sont détectées. Les développeurs devront travailler avec NVIDIA pour tester leur jeu. Quid du Radeon Crossfire ? Je crois qu'il ne faut pas compter dessus car Epic et Nvidia, c'est la grande histoire d'amour.

Custom Stencil sur mobiles

4.15

Les appareils mobiles peuvent désormais utiliser 'Custom Stencil' dans les matériaux de post-traitement permettant la description d'objet, la visualisation des objets occlus, ou d'autres effets. Cette fonction nécessite MobileHDR activée. Pour activer cette fonctionnalité dans votre projet aller à Project Settings -> Rendering -

Unreal Engine 4 – Guide des mises à jour (4.20)

> Postprocessing et réglez 'Custom Depth-Stencil Pass' sur 'Enabled with Stencil'.



Pour cela: Désactivation Allow Static Lighting, de Support Combined Static et CSM Shadowing, de Support Distance Field Shadows, de Support Movable Directional Lights, et Réglage Max Movable Point Lights à 0. Puis activation de Use Shared Movable Point Light Shaders.

Résolution dynamique

4.19

La **résolution dynamique** pour consoles (PS4 et Xbox One, autres dans une version ultérieure). Le principe reste simple, pour assurer un framerate, c'est à dire, un nombre d'images par secondes, fixe, la résolution des images est modifiée dynamiquement.

On peut l'activer ou non via:

```
GEngine→GetDynamicResolutionState()→SetEnabled(true/false)
```

Synchronisation de frame à faible latence

4.19

Toujours coté consoles (Xbox One et PS4), le mode de **synchronisation de frame à faible latence** modifie la façon dont la synchronisation des threads est effectuée entre le jeu, le rendu, les threads RHI et le GPU- ce qui permet de réduire la latence coté inputs.

Les effets de la synchronisation de trames à faible latence peuvent être contrôlés à l'aide de la variable de console `r.GTSyncType`. Cette variable autorise les valeurs suivantes:

- 0 - Le thread du jeu se synchronise avec le thread de rendu (ancien comportement, et par défaut)
- 1 - Le thread du jeu se synchronise avec le thread RHI (équivalent à UE4 avant le rendu parallèle)
- 2 - Le fil du jeu se synchronise avec la chaîne d'échange présente +/- un décalage en millisecondes (disponible uniquement sur les plates-formes supportées, et ne fonctionne que si la synchronisation est activée).

Hardware and Software Occlusion Queries

4.20

Elles sont prises en charge par les appareils mobiles haut de gamme sur iOS et Android (ES 3.1 ou Vulkan)

Software Occlusion Queries est une fonctionnalité expérimentale qui utilise le processeur pour éliminer les composants primitifs de la scène. Parce qu'il utilise une approche prudente, il peut être utilisé sur n'importe quel appareil mobile.

Avec software d'occlusion - Rendu montrant les parties occluses

Pour activer les requêtes d'occlusion logicielle, procédez comme suit:

Unreal Engine 4 – Guide des mises à jour (4.20)

1. Activer `r.Mobile.AllowSoftwareOcclusion` 1.
2. Désactiver `r.AllowOcclusionQueries` 0.
3. Permettre à n'importe quelle primitive d'être un occluteur en définissant **LOD for Occluder Mesh** true dans l'éditeur Static Mesh.



Vous pouvez visualiser les résultats dans [Mobile Previewer](#) en utilisant **High-End Mobile**, puis activer `r.SO.VisualizeBuffer` 1.

Platform Material Stats

4.20

Visualisez et optimisez rapidement vos matériaux en utilisant la nouvelle fenêtre Platform Stats dans l'éditeur de matériaux! Vous pouvez maintenant voir les statistiques de plusieurs plate-formes de shaders et niveaux de qualité. Pour les plates-formes mobiles, nous utilisons un compilateur de shaders hors ligne pour donner des instructions plus précises et des informations sur l'utilisation de la texture.

Unreal Engine 4 – Guide des mises à jour (4.20)

charge, à condition qu'il y ait suffisamment de sampleurs de texture disponibles.

Vous pouvez maintenant utiliser les noeuds Material Material Switch Material dans Landscape Materials, ce qui vous permet de **créer un seul matériau Landscape pour toutes les plates-formes**.



1 - Paysage sur mobile; 2 - Paysage sur poste fixe

Améliorations mobiles diverses

4.20

- Minimum Static Mesh LOD par plateforme
- Minimum Skeletal Mesh LOD par plateforme
- Améliorations de l'occlusion matérielle
- Outils HLOD et optimisations de flux de travail
- Noeud de qualité audio
- Variation audio
- Sous-échantillonnage audio par plate-forme
- Qualité de compression audio par plate-forme
- Shading modèle tweaks pour mieux correspondre PC
- Correction de la luminosité de la capture de réflexion

Unreal Engine 4 – Guide des mises à jour (4.20)

- Prise en charge du paysage pour quatre couches
- Améliorations de la tessellation du paysage
- Pas de coût de mémoire pour les LOD non utilisés
- Paramètres basés sur la mémoire de l'appareil
- Réduction de la mémoire matérielle
- Scriptabilité de l'éditeur pour les modifications d'actifs en masse
- Partage de composants de particules

Les SDK des différentes plates-formes ont été mises à jour

On ne conservera ici que la toute dernière mise à jour. Pour avoir les autres, [rendez-vous ici](#).

4.19

- **IDE Version the Build farm compiles against**
 - **Visual Studio:** Visual Studio 2017 v15.6.3 toolchain (14.13.26128) and Windows 10 SDK (10.0.12699.0)
 - Minimum supported versions
 - Visual Studio 2017 v15.6
 - Visual Studio 2015 Update 3
 - **Xcode:** Xcode 9.4
- **Android:**
 - NDK 12b (New CodeWorks for Android 1r6u1 installer will replace previous CodeWorks for Android 1R5 before release, still on NDK 12b)
- **HTML5:** Emscripten 1.37.19
- **Linux:** v11_clang-5.0.0-centos7
- **Lumin:** 0.12.0
- **Steam:** 1.39
- **SteamVR:** 1.39
- **Oculus Runtime:** 1.25
- **Switch:**
 - SDK 4.5.0 + optional NEX 4.2.1 (Firmware 4.1.0-1.0)
 - SDK 5.3.0 + optional NEX 4.4.2 (Firmware 5.0.0-4.0)

Unreal Engine 4 – Guide des mises à jour (4.20)

- Supported IDE: VS 2015 / 2017
- **PS4:**
 - 5.508.031
 - Firmware Version 5.530.011
 - Supported IDE: Visual Studio 2015, Visual Studio 2017
- **Xbox One (XB1, XB1-S, XB1-X):**
 - XDK: April 2018
 - Firmware Version: April 2018 (version 10.0.17133.2020)
 - Supported IDE: Visual Studio 2017
- **macOS:** SDK 10.13
- **iOS:** SDK 11
- **tvOS:** SDK 11



Support switch Nintendo

4.15



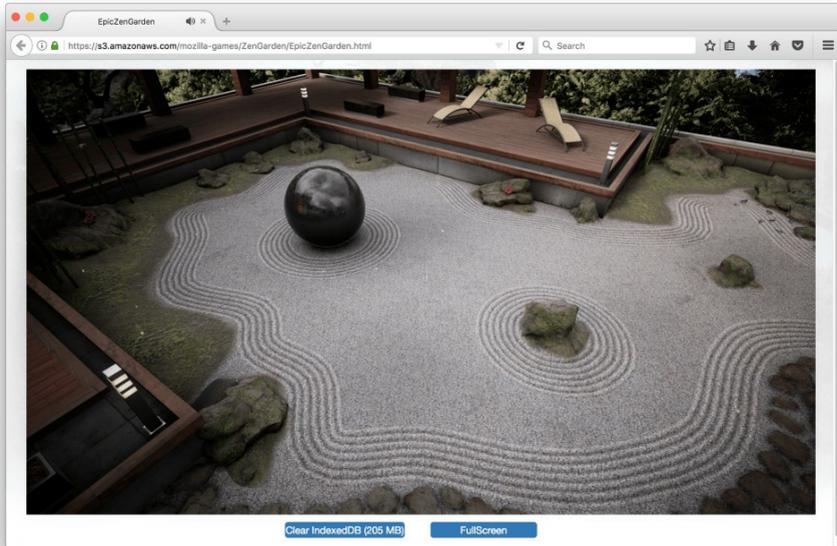
La version est encore expérimentale mais sera définitive lors de la prochaine mise à jour. Bon, ça c'est la nouvelle qui a fait le tour du web ! C'est aussi ce sur quoi [l'équipe avec laquelle j'ai travaillé il y a plus d'un an](#) maintenant attendait (nous en avons eu vent, mais pas de façon officielle).

Personnellement, je ne sais pas trop quoi penser de cette console. L'idée est assez bonne, mais je ne me sens pas trop concerné en tant que joueur... j'ai une vieille Wii parce que c'est une console familiale et qu'il y avait des accessoires sympas... après je suis plus console NexGen type Xbox one/PS4 même si j'en suis encore à jouer avec ma vieille xbox 360. En tous cas, je ne sais pas encore comment on va exporter vers la console, si cela va être simple ou une grosse usine à gaz (car l'export vers console d'UE4, c'est quand même pas très pratique). Si c'est simple et direct, je me prendrai probablement une [Switch](#). A suivre...

4.16

Le support [Nintendo Switch](#) est désormais complet, prêt pour le portage simple de jeux et même en réseau, suite au travail commun avec Nintendo. Ahaha!!!! A vos souris les amis, y-a plein de jeux à créer sur cette nouvelle console.

Unreal Engine 4 – Guide des mises à jour (4.20)



Bon, je ne m'étendrai pas sur le sujet car je vous en ai déjà parlé à plusieurs reprises sur ce blog.

4.20

- Prise en charge de la résolution dynamique et du suréchantillonnage temporel (Temporal Upsampling)
- Synchronisation de trame à faible latence pour l'entrée du contrôleur
- Importantes optimisations de rendu du processeur
- Améliorations au filetage
- Meilleure compression de texture
- Prise en charge du profilage de la mémoire
- support de 1080p lorsque la console est sur son dock
- Et beaucoup d'autres corrections!



Support Linux ARM64

4.15

support pour les périphériques (ARM64 AArch64) sous Linux, ces dispositifs doivent avoir OpenGL 3.2 ou version ultérieure pour faire tourner le moteur. Je me demande ce que ça donne avec le [Raspberry Pi 3](#)... c'est un ARM64 bits sous Linux, mais je crois savoir que l'implémentation de l'Api 3D n'est pas encore au top. Si vous avez des infos là dessus, je suis preneur.

ÉDITEUR ET TEMPLATES

In-World Component Editing

4.7

Les composants peuvent maintenant être ajoutés directement à des acteurs placés dans le niveau (c'est à dire, sans passer par l'asset d'origine).

Après avoir créé un ensemble de composants, on peut cliquer sur **“Blueprint / Add Script”** pour le sauver comme une classe réutilisable de Blueprint ou de C++. L'acteur sera automatiquement remplacé par une instance de cette classe.

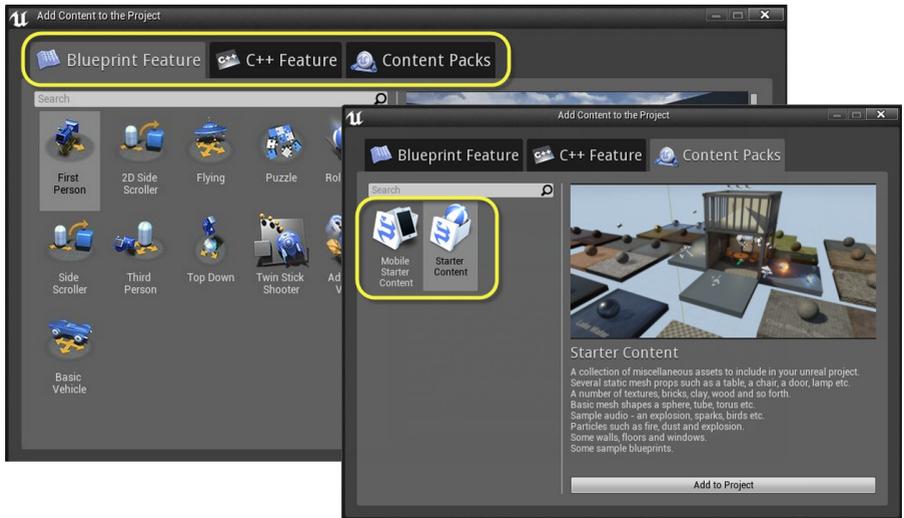
C'est vraiment très pratique à l'utilisation, surtout lorsqu'on avance pas à pas et qu'on structure son code au fur et à mesure. C'est vrai que dans l'idéal, on devrait tout prévoir à l'origine: les objets, les classes, les composants... mais souvent on avance à tâtons, on améliore, et c'est là que cette nouvelle fonctionnalité devient fort utile.

Feature Packs and Content Packs

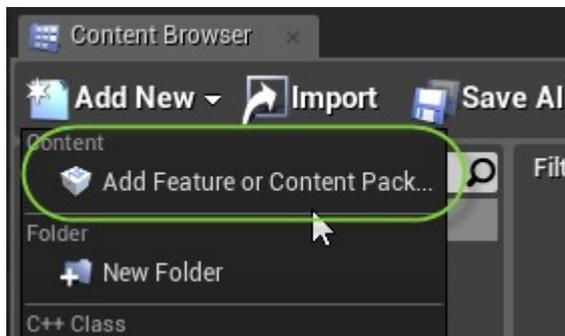
4.7

Plus besoin de créer un nouveau projet et/ou de migrer les assets vers votre projet actuel pour disposer des fonctionnalités d'un template. On peut désormais "ajouter" la fonctionnalité au projet courant:

Unreal Engine 4 – Guide des mises à jour (4.20)



Imaginez que vous ayez bossé sur un FPS et que vous voulez ajouter un template Vehicle. Rien de plus simple avec cette nouvelle approche. J'imagine que cela va booster également le marketplace.



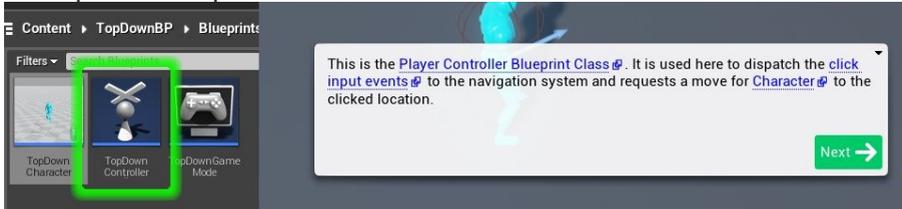
On peut également ajouter le **Starter Content** à tout moment, avec une version **optimisée pour les mobiles**.

Je me rappelle encore du temps où je copiais le template directement sous l'explorateur de fichiers, ou que j'utilisais la fonction Migrate... c'est quand même bien plus simple ainsi. D'autant plus que les templates devraient apparaître en nombre au fil du temps.

Tutorials for New Project Templates

4.7

Chaque nouveau projet (pour l'instant, il n'y a que TopDown) utilisant un template a maintenant son propre Overview Tutorial qui fournit un descriptif de ce qu'il fait et des classes associées.

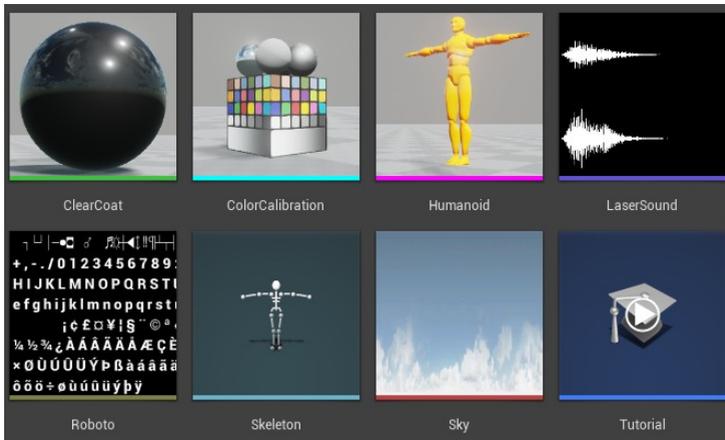


Et il y a bien entendu l'ensemble de l'API gérant cela qui est ouverte aux Blueprints pour créer vos propres tutoriaux. C'est d'autant plus intéressant si vous souhaitez partager votre travail ou vendre des modules sur le Marketplace.

New Thumbnail Look and Feel

4.7

Les classes C++ et les Blueprint profitent dans la mesure du possible d'un nouveau système d'affichage des Thumbnails.



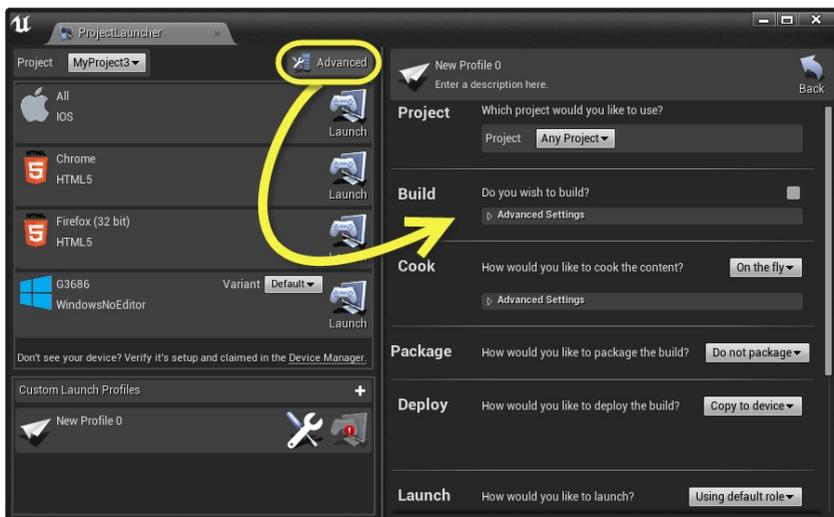
Unreal Engine 4 – Guide des mises à jour (4.20)

J'aime beaucoup également la possibilité de prendre le snapshot avec l'angle et le zoom souhaité pour améliorer la présentation de ce dernier.

Project Launcher

4.7

La nouvelle interface "Project Launcher" permet de lancer plus facilement un projet avec plus de contrôles.



Cela permet de tester son jeu sur de multiples plate-formes et configurations plus simplement, tout comme lancer simultanément plusieurs sessions.

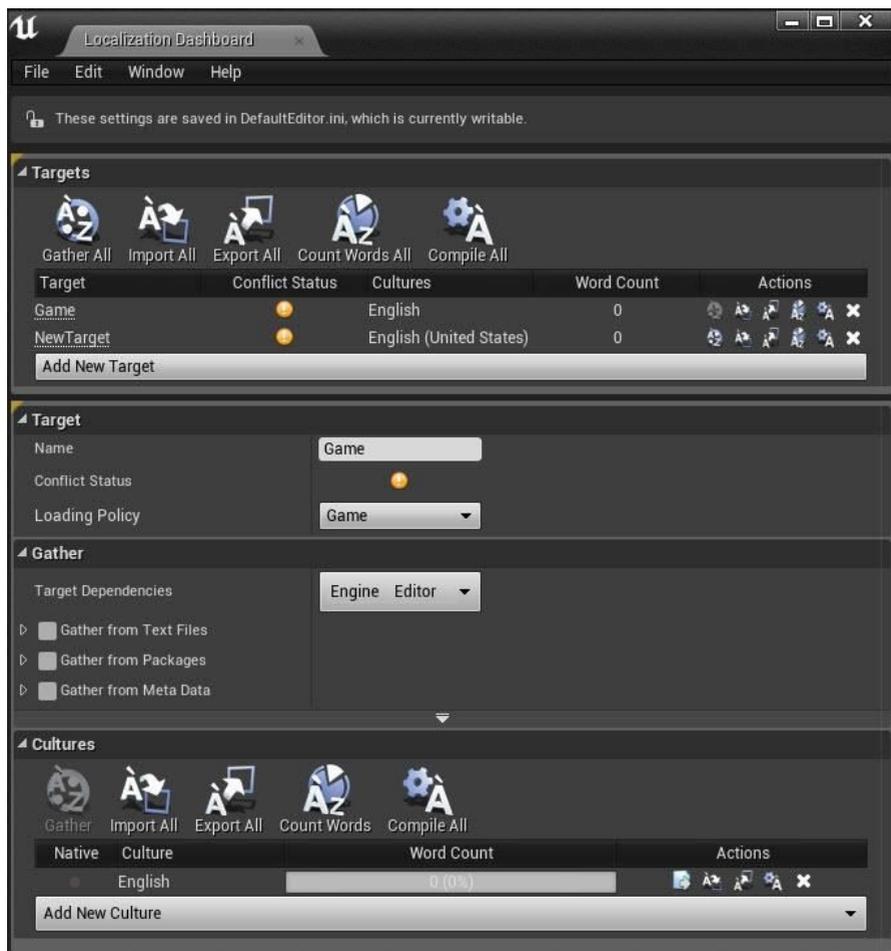
Localization Dashboard

4.8

Le tableau de bord localisation est maintenant utilisable comme une fonctionnalité expérimentale. Et c'est encore le cas en 4.12

Unreal Engine 4 – Guide des mises à jour (4.20)

Il permet le changement de langage et bien d'autres encore. Vous retrouverez tous ces éléments dans le tome 3 des cahiers d'Unreal Engine.



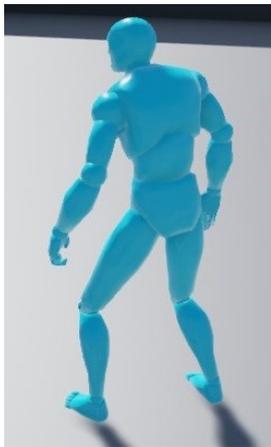
Nouvel Exemple de personnage Unreal

4.8

Nouvelle version, **haute et basse résolution** (pour les mobiles) du personnage "exemple" Unreal (le bonhomme bleu pour les intimes qui ne maîtrisent pas le retargetting...). Ce modèle est inclus dans l'ensemble des modèles. Il est importé automatiquement lorsque vous utilisez le "ThirdPerson" Template.

Oui, le nouveau est plus beau, mieux conçu au niveau du squelette, etc. Mais c'est un peu la poisse car les squelettes ne sont pas compatibles à 100%, même en retargeting. De fait, certaines animations ne fonctionnent plus comme il faut. J'aurais préféré qu'ils laissent l'ancien et ajoutent le nouveau, sans changer non plus les classes d'objets correspondants.

Avant



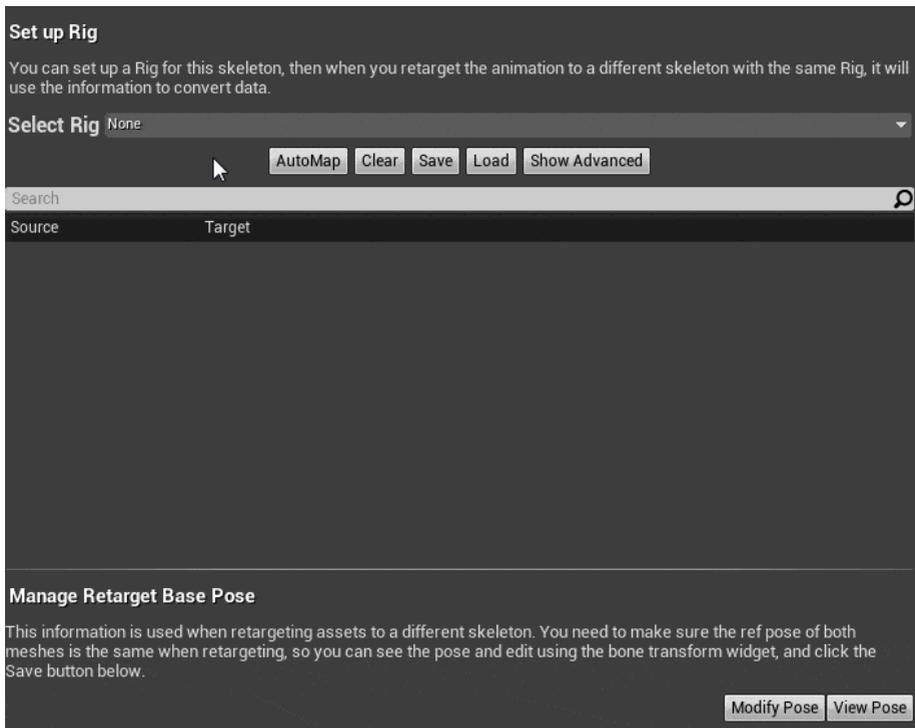
Après



Amélioration de Animation Retarget Manager

4.20

Animation Retarget Manager prend désormais en charge l'enregistrement et le chargement des données de mappage, ce qui permet de sauvegarder et de réutiliser les données de mappage sur plusieurs meshes. Vous pouvez également enregistrer rapidement plusieurs données de rig pour différentes animations et les réutiliser avec cette fonctionnalité. Ca va faciliter le travail des animateurs qui travaillent avec des skeletal différents du modèle Unreal... ça me rappelle de longues soirées...



Consultez la page [Retarget Manager](#) pour plus d'informations.

Asset size Mapper

4.8

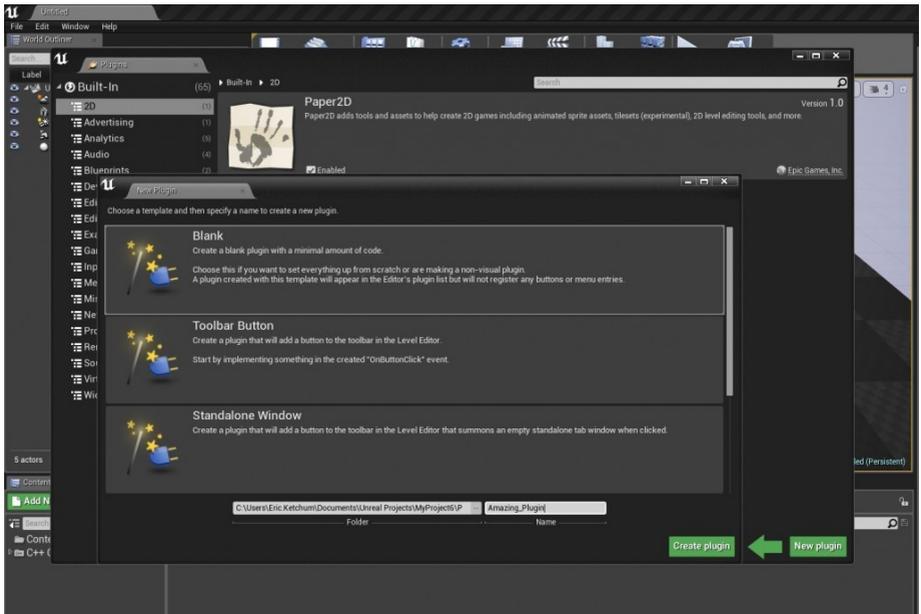
"Size Mapper" permet de voir la **mémoire utilisée par un asset avec tout ce qui dépend de lui** sous forme d'arbre interactif. Je vous en avais déjà parlé, mais je crois qu'il n'y avait pas cette notion de taille prise en mémoire il me semble.

Bon, ça reste un peu difficile à utiliser à cause de la récursivité de l'affichage, mais ça donne une bonne idée tout de même.

Assistant de création de Plugin

4.9

Il est maintenant très facile de **créer de nouveaux plugins avec c++**.



Unreal Engine 4 – Guide des mises à jour (4.20)

Le navigateur de Plug-ins (accessible depuis le menu Window) permet désormais de créer un nouveau plugin!

Vous serez en mesure de donner à votre nouveau plugin un nom et de sélectionner quelques types de plugin prédéfinis (Blank, barre d'outils, fenêtre autonome).

Cela va créer tous les fichiers initiaux dont vous avez besoin pour votre plugin pour compiler et charger dans l'éditeur.

Bon, ça reste un assistant de création, mais il évite de tout refaire à la main et d'oublier certaines choses.

N'oubliez pas que les plugins peuvent être déposés dans le répertoire du projet et pas seulement dans le répertoire du moteur (ce que je vous déconseille si vous voulez gérer plusieurs versions). Vous pouvez aussi ajouter le code source pour une recompilation en cas de changement de version. Je vous préparerai quelque chose de complet sur la question. Nous en reparlerons.

Navigateur de contenu

Recherche avancée

4.9

Le Navigateur de contenu prend désormais en charge la **syntaxe de recherche avancée**.

Unreal Engine 4 est utilisé sur de nombreux grands projets, dont certains avec des équipes composées de centaines de développeurs et les bibliothèques de contenu avec des centaines de milliers d'actifs! Ces nouvelles fonctionnalités de recherche permettent d'améliorer le flux de travail pour trouver votre contenu sur des projets de toute taille.

Unreal Engine 4 – Guide des mises à jour (4.20)

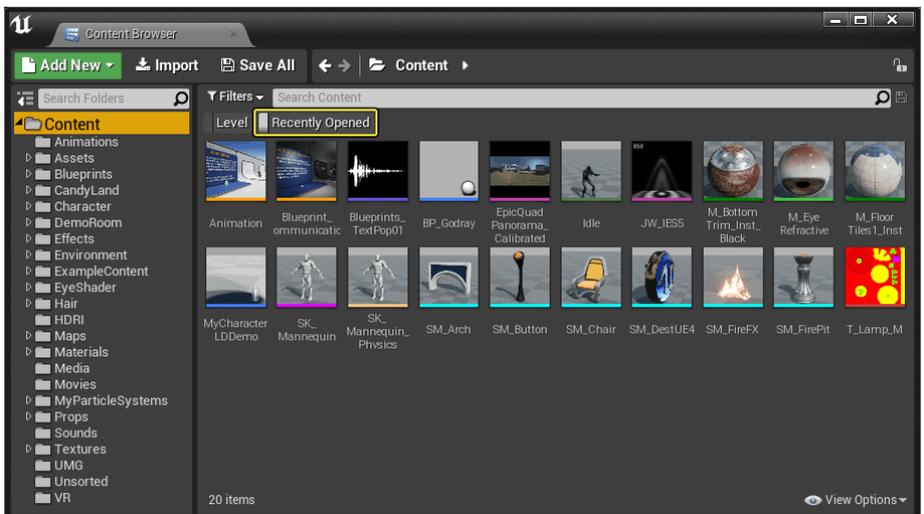


Cela augmente la syntaxe de recherche existante et les touches spéciales suivantes:

- Nom - Teste le nom de l'actif.
- Path - Teste le chemin de l'actif.
- Classe (alias: Type) - Essaie la classe d'actifs.
- Collection (alias: TAG) - Teste les noms de toutes les collections qui contiennent l'actif.

Filter pour trouver les assets récents dans le navigateur de contenu

Retrouvez rapidement les assets récemment ouverts dans le navigateur de contenu à l'aide du nouveau filtre Recently Opened ! Ce filtre répertorie les 20 derniers assets ouverts.



Unreal Engine 4 – Guide des mises à jour (4.20)

Vous pouvez trouver le filtre Recently Opened dans la liste Filters sous Other Filters. Vous pouvez modifier le nombre d'assets récemment ouverts répertoriés dans Editor Preferences > Content Browser avec le nombre d'assets à conserver "Number of Assets to Keep in the Recently Opened Filter".

Pour plus d'informations, voir [Content Browser Filters](#).

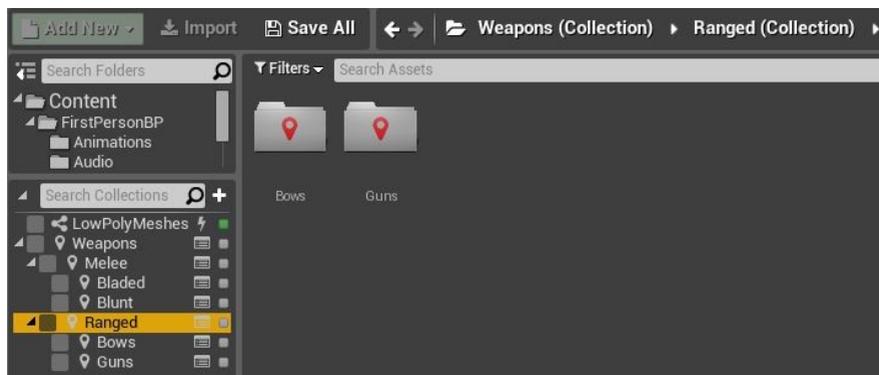
Amélioration des collections

4.9

Les collections permettent de vous aider à gérer les actifs dans votre jeu.

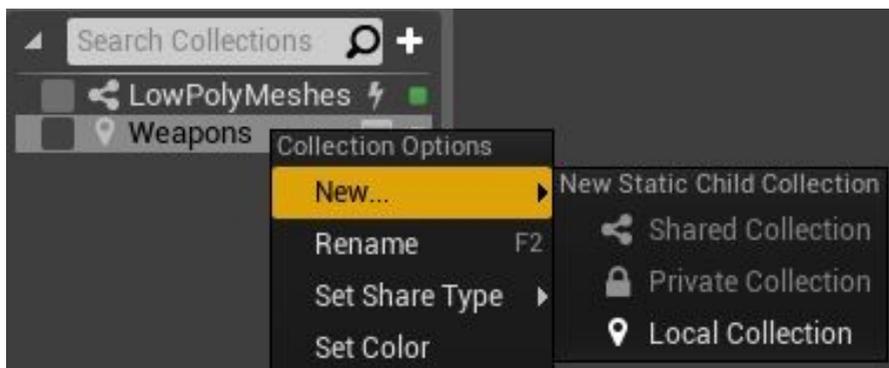
Collections imbriquées

Vos collections d'assets peuvent maintenant être **organisées en hiérarchie**.



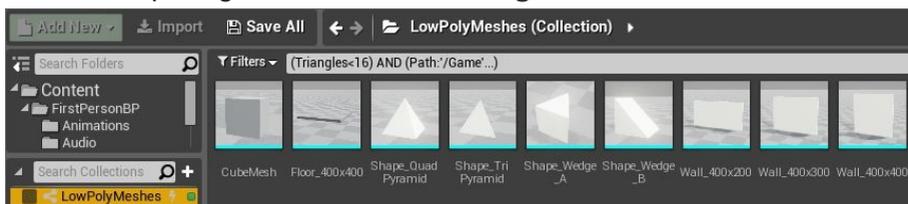
Child collections peuvent être créées par le choix de l'option "New..." dans un menu collection contexte donné.

Alternativement, les collections peuvent être parentées par glisser-déposer.



Collections dynamiques

Les collections dynamiques exploitent la puissance de l'avancée de la recherche de contenu dans le Navigateur pour vous permettre de créer et de partager des **filtres de navigateur de contenu**.



Je dois avouer que je n'ai pas encore beaucoup utilisé cette fonction... à tort probablement car je suis resté un peu figé dans le mode "organisation par répertoire".

En même temps, cela fonctionne bien. Il va falloir que je penche sérieusement sur la question.

Quick Asset Management

Nouveau: Asset Management Framework Production

4.17

Unreal Engine 4 – Guide des mises à jour (4.20)

Pour utiliser le gestionnaire d'assets, appeler des fonctions sur la classe native UAssetManager (qui peut être sous-classée pour votre jeu), ou appeler des fonctions Blueprints de la catégorie d'Asset Manager, comme Async Load Primary Asset.

4.16

Asset Management Framework (Early Access Preview)

L'Asset Manager est un nouvel objet global qui peut être utilisé pour découvrir, charger et analyser des cartes et assets spécifiques au jeu dans l'éditeur **ou à l'exécution**. C'est un point vachement important, car jusqu'à présent, c'était assez galère de lister les assets d'un projet. Il est facile de lister des actors, mais pour les assets, c'était une autre paire de manches! A l'exécution je veux dire...

Il fournit un cadre pour rendre plus facile la création des choses comme des quêtes, des armes ou des héros et de les charger sur demande. **Il est encore en cours de développement**. Je suppose qu'on devrait pouvoir ajouter également des assets de l'extérieur et pendant l'exécution pour faciliter par exemple, le téléchargement d'une nouvelle arme dans un jeu, ou d'un nouveau bouclier, etc.

L'onglet Asset Manager dans les paramètres du projet peut être utilisé pour mettre en place les règles de votre jeu: les Types d'assets primaires qui sont scannés par l'Asset Manager peut être questionnés lors de l'exécution avant leur chargement, et peuvent ensuite être chargés de manière asynchrone sur demande. Dans la documentation de progression pour cette fonction est disponible sur [AnswerHub](#).

4.16

Asset Audit Window (Early Access Preview)

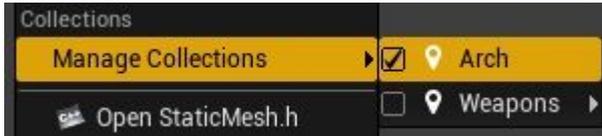
Construit sur l'Asset Management Framework, la fenêtre de vérification de l'asset peut être utilisée pour vérifier la taille du disque, utilisation de la mémoire, et les propriétés d'assets générales.

4.9

Unreal Engine 4 – Guide des mises à jour (4.20)

Vous pouvez maintenant **ajouter rapidement un asset dans de nombreuses collections à la fois** en utilisant les nouvelles cases à cocher Quick Asset Management.

Elles sont disponibles dans la vue de collection elle-même ou dans le menu contextuel de chaque asset.



Amélioration des rapports d'état

4.9

Chaque collection a un **indicateur d'état à droite**:

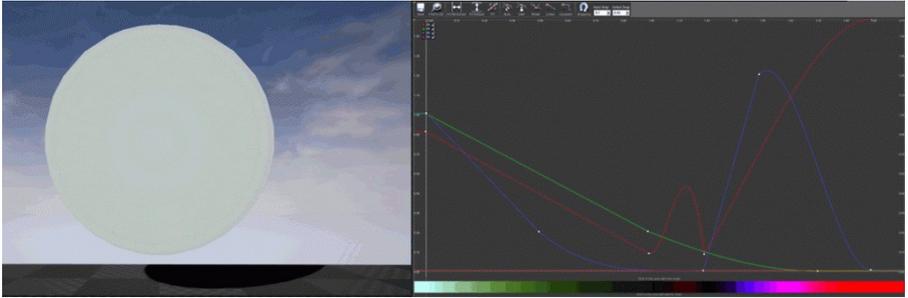
- Rouge - collection avec un problème ou en lecture seule. Voir l'encadré d'aide pour savoir comment résoudre le problème.
- Orange - collection non mise à jour avec la dernière version de contrôle de code source.
- Bleue - collection avec des modifications locales non enregistrées. (sauvegarde ou auto-check échoué, ou collection avec redirecteurs suivi).
- Vert - collection non vide et mise à jour.
- Grise - collection vide.

Amélioration du curve editor: éditeur de courbes

4.9

De nombreuses améliorations ont été apportées à l'éditeur de courbes pour rendre **plus facile à manipuler les touches de courbe et les tangentes**.

Unreal Engine 4 – Guide des mises à jour (4.20)



- Le Bouton du milieu de la souris déplace désormais des images clés sélectionnées ou des tangentes sélectionnées
- Il y a maintenant un support pour la sélection et la manipulation de plusieurs tangentes
- "Sélection" respecte maintenant le contrôle (toggle) et le décalage (ajout à la sélection existante)
- SHIFT restreint le mouvement à un axe
- Navigation: ALT + bouton milieu de la souris va au milieu de la vue et ALT + bouton droit de la souris zoome la vue
- Nouvelles options d'affichage de la tangente:
 - Toujours afficher les tangentes,
 - montrer les tangentes pour les points sélectionnés,
 - ne jamais montrer tangentes
- Nouvelles options pré-post infini extrapolation pour les courbes
- Créez des touches additionnelles plus facilement
 - Les clés sont ajoutées uniquement quand il y a seulement une courbe à modifier
 - Lorsque plusieurs courbes sont disponibles, cliquer sur une courbe ajoute une clé à cette courbe, et cliquer aucune courbe ajoute des clés à toutes les courbes

Je pense que l'éditeur de courbes a atteint un bon stade de maturité depuis cette version. Il va être utilisé par de nombreux composants de l'éditeur, dont le fameux "Sequencer editor" qui va remplacer Matinee à terme.

Performance et multithreading

4.11

Unreal Engine 4 – Guide des mises à jour (4.20)

L'**architecture multithread** (pouvoir exécuter plusieurs tâches en parallèle, optimiser en fonction des cœurs, des CPU, des GPU, etc.) a été améliorée de plusieurs manières :

- réduction du coût de création des tâches,
- ajout du support des tâches prioritaires
- et suppression des points de synchronisation.

Le jeu Paragon a offert de nouveaux défis à l'équipe, il a fallu soutenir l'animation de 10 héros à la fois, et de plus de 120 PNJ, beaucoup de FX, tout en rendant une belle carte détaillée à 60fps sur PC et PS4.



- **La parallélisation** est cruciale pour atteindre des cadences élevées sur les PC et les consoles modernes, donc l'architecture de filetage a dû être améliorée.
- **Performances de rendu:** meilleur équilibre des tâches de travail, suppression des points de synchronisation
- **Simulation des tissus (Cloth)** beaucoup "beaucoup" **plus rapide** et permettant une meilleure utilisation du multithreading. Appel de l'APEX solveur directement pour chaque asset pour mieux planifier.
- **Multithread animation:** c'est un point important – le Blueprint d'animation peut maintenant bénéficier de l'utilisation de plusieurs cœurs et CPU.

Unreal Engine 4 – Guide des mises à jour (4.20)

- **Possibilité de pré calculer** (bake) les animations additives, permettant à ces dernières de tourner 3x plus vite.
- **Fast Garbage collection:** Ici, le cache est traité de façon plus cohérente, permettant de réduire le temps de traitement d'un facteur 9x.

Tout cela fait partie des améliorations qui ne sont pas visibles directement. Mais si vous avez déjà travaillé sur un gros projet, tenté de chasser les pertes de performances... vous voyez qu'un grand pas a été franchi ici. C'est d'autant plus vrai que mêmes les architectures mobiles possèdent maintenant plusieurs processeurs et cœurs et vont bénéficier de ces améliorations. Et ne parlons pas des consoles!

Vous remarquerez, au fil du temps, que c'est souvent en lançant de gros projets (l'enfant et son Kite pour les open worlds, Paragon pour les animations et le rendu des personnages, etc.) qu'Epic a amélioré telle ou telle partie de son moteur et de son éditeur.

Garbage collector

4.16

Le Garbage collector, c'est la façon dont on gère la mémoire quand des éléments sont supprimés, est désormais 2x plus rapide de façon à améliorer grandement le multithreading.

4.20

La performance de Garbage Collection a été optimisée en réduisant jusqu'à 13x certaines opérations! Plus précisément, Epic a apporté les améliorations suivantes.

La phase "Mark" a été optimisée et est maintenant multithread. Sur les machines à plusieurs cœurs, le coût de marquage des objets comme inaccessibles a été réduit de 8 ms à 0,6 ms pour environ 500 000 objets .

La phase "BeginDestroy" (Unhashing Objects) est maintenant exécutée sur plusieurs images, en n'utilisant pas plus de 2 ms par image. Le coût de démontage des objets ne sera plus inclus dans la même trame que la phase "Mark" et l'analyse d'accessibilité.

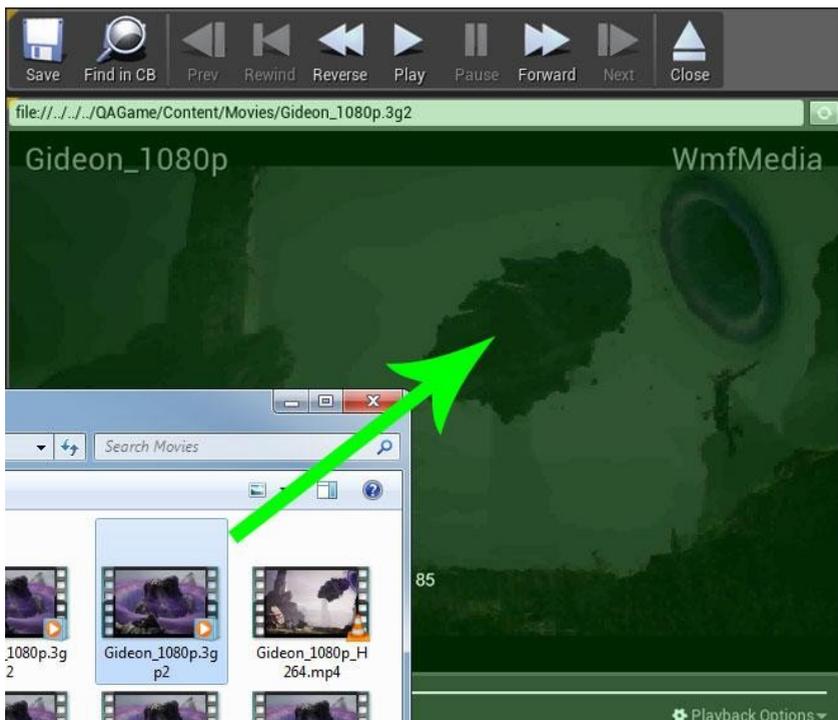
Unreal Engine 4 – Guide des mises à jour (4.20)

La vérification d'hypothèse de Garbage Collection, qui s'exécute dans les versions de développement, utilise désormais le même code multithread que la collecte de référence. Par conséquent, les versions de développement verront une amélioration des temps de récupération des ordures. Dans les tests d'Epic, les temps d'échantillonnage pour environ 500 000 objets ont été réduits de plus de 320 ms à moins de 80 ms .

Éditeur de lecteur média

4.14

Vous pouvez maintenant glisser déposer des fichiers depuis l'ordinateur dans la fenêtre du lecteur multimédia pour visualiser les fichiers vidéo sans devoir créer un asset FileMediaSource.



Création de Static Mesh à partir de n'importe quel actor

4.14

Il suffit de faire un clic droit sur un actor dans le level, puis de cliquer sur "Convert Actor to Static Mesh" comme dans [cette vidéo](#). cela fonctionne aussi avec les personnages skinnés: ainsi on peut prendre une pose et "extraire" le static mesh correspondant (pour faire une statue par exemple).

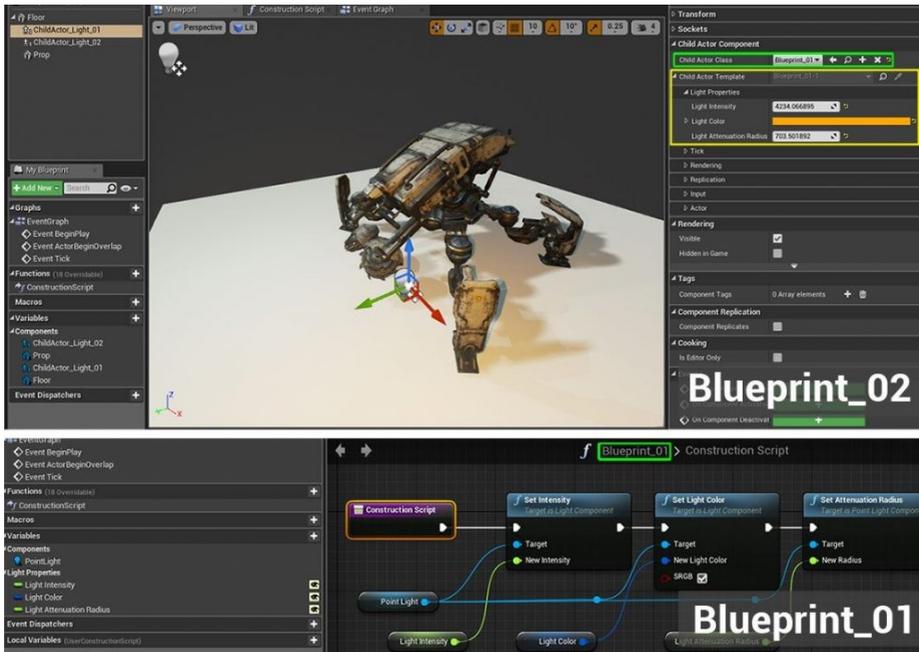
J'ai encore du mal à imaginer les applications de cette fonctionnalités, mais bon je suppose que cela doit être assez pratique. Je pense notamment à la génération procédurale et la possibilité d'exporter ainsi facilement ses créations. Mais il doit y avoir encore d'autres besoins auxquels je ne pense pas forcément.

Child actor templates

4.14

Les composants ajoutés aux Child Actor Components peuvent avoir leurs propriétés personnalisées via Child Actor Templates.

Unreal Engine 4 – Guide des mises à jour (4.20)



Une fois que vous ajoutez un Child Actor Component, vous verrez un modèle extensible dans le panneau Détails de l'éditeur de Blueprint. De là, vous pouvez accéder à toutes les propriétés du Child Actor y compris les variables publiques.

Amélioration du composant Câble

4.14

Les câbles peuvent maintenant avoir un rendu nettement plus réaliste : le moteur de jeu considère leurs collisions avec l'environnement (y compris avec friction) et permet d'y attacher des objets et des effets.



Live GPU profiler

4.14

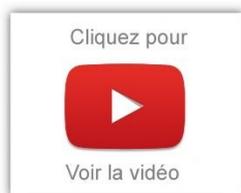
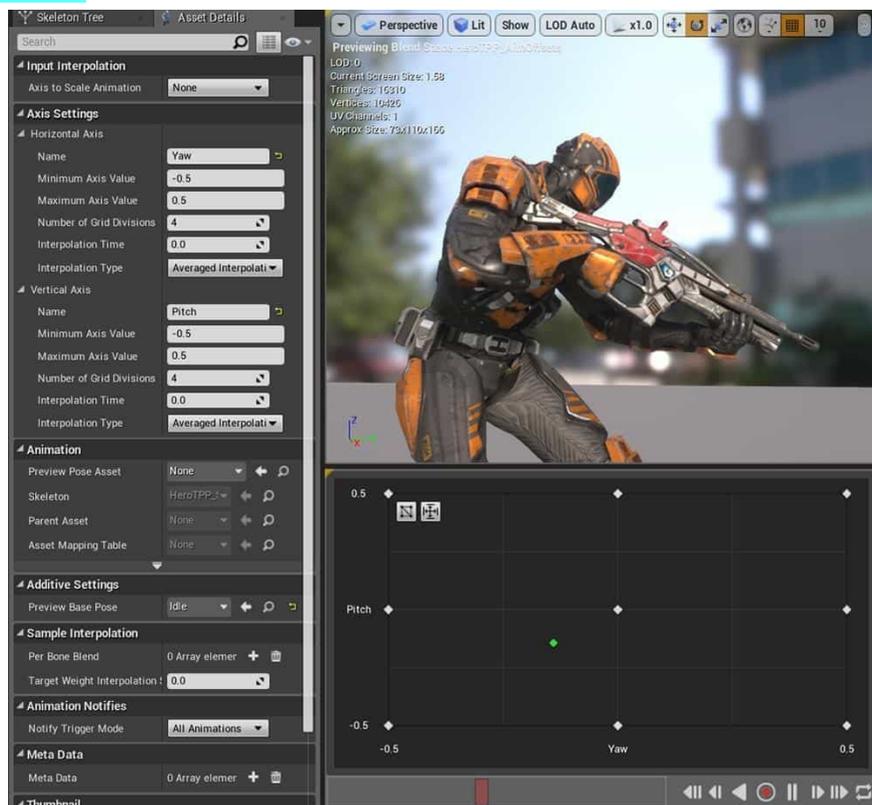
Il fournit des statistiques pour les principales catégories de rendu. Pour l'utiliser, entrez la commande de la console 'stat gpu' ou allez dans l'éditeur via le sous-menu 'Stat' dans le menu déroulant Options dropdown.

	Average	Max
GPU [STATGROUP_GPU]		
Quakes		
TOTAL	1576	1430
Basepass	382	425
Postprocessing	252	289
Prorass	0.95	1.35
Reflection Environment	115	152
Capsule Shadows	0.85	1.24
RTS	0.45	0.83
Slate 3D	0.65	0.91
Signs	0.67	1.07
Translucency	0.55	0.97
[unaccounted]	0.45	0.84
Slate UI	0.34	0.75
Composition BeforeBasePass	0.25	0.28
Render Velocities	0.23	0.55
Particle Simulation	0.21	0.23
ShadowProjection	0.16	0.18
Distortion	0.16	0.55
Shadow Depth	0.14	0.23
Composition PreLighting	0.15	0.55
Custom Depth	0.09	0.24
Composition PostLighting	0.00	0.04

Le Blendspace Editor

Nouveau look et optimisation le workflow

4.15



- De multiples façons de modifier les propriétés de l'échantillon: glisser-déposer, menu contextuel et spin-boxes pour modifier la valeur de l'échantillon ou d'une séquence d'animation.
- Aperçu du blendspace en maintenant **Maj (Shift)** enfoncée

Unreal Engine 4 – Guide des mises à jour (4.20)

- Glisser et déposer des opérations - Validation et aperçu des glisser-déposer depuis le Animation Tools Asset Browser.
- Étirer pour ajuster la grille du plus petit au plus grand axe, etc.

Améliorations de la convivialité de l'éditeur Blendspace

4.18



Il est possible de faire glisser la punaise de prévisualisation (verte) sur la grille tout en maintenant la touche Maj enfoncée.

Workflow

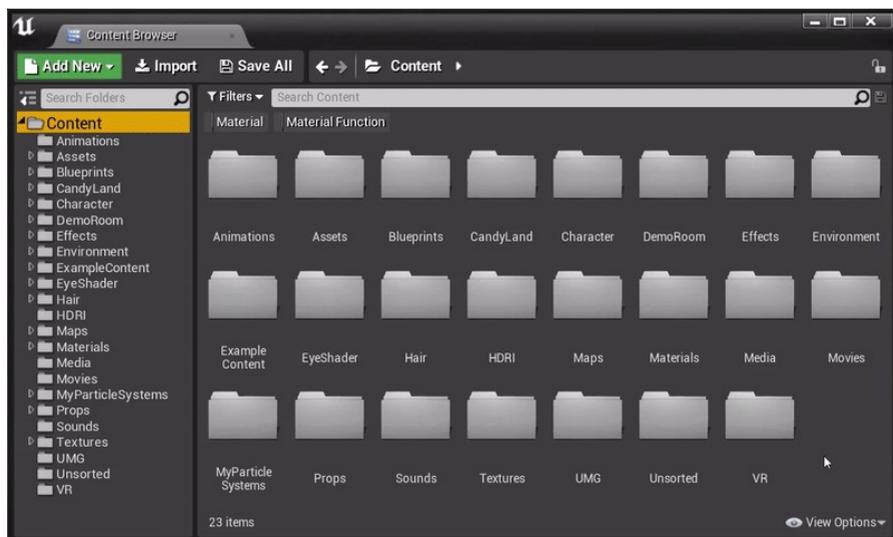
dossiers favoris

4.19

Un dossier particulier peut être marqué dans votre navigateur de contenu en tant que «favori» pour le rendre facilement accessible!

La section "Favoris" du navigateur de contenu peut être activée / désactivée à l'aide des options d'affichage pour Afficher les favoris. Cette fonction est activée par défaut.

Unreal Engine 4 – Guide des mises à jour (4.20)



Optimisation de l'espace disque

4.19

Plusieurs outils ont été mis en place pour permettre d'analyser plus efficacement les tailles des packages.

Étiqueter les couleurs enregistrées dans le sélecteur de couleurs

4.20

Les couleurs enregistrées dans votre barre de thème ou menu de thème peuvent maintenant avoir des étiquettes à des fins d'identification! Les étiquettes peuvent être facilement définies en cliquant avec le bouton droit sur l'échantillon de couleur enregistré et en entrant un nom pour la couleur enregistrée.

<https://youtu.be/qCtrqIFCYHA>

Pour plus d'informations, voir [Color Picker](#).

Scripted Extensions for Actor and Content Browser Context Menu

4.20

Créez facilement des outils contextuels et des améliorations de flux de travail sans écrire une ligne de code. On peut étendre les menus contextuels des acteurs et des contenus dans le navigateur à l'aide de Blueprint Utilities ou **Blutilities**. C'est chouette comme petit nom ?

Créez une nouvelle Blutility en utilisant l'une des nouvelles classes parentes - AssetActionUtility (pour les extensions Content Browser) ou ActorActionUtility (pour les extensions Actor).

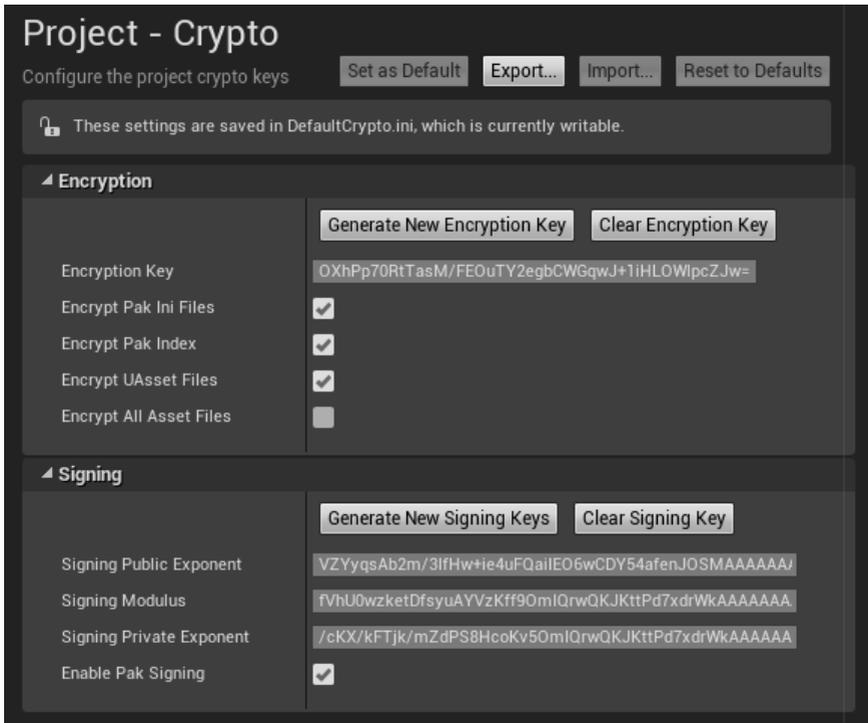
Vous pouvez spécifier les types d'acteurs ou d'assets auxquels les actions s'appliquent avec la fonction GetSupportedClass. Ajoutez une logique dans les événements (ou fonctions) sans valeur de retour, en les marquant comme "Call In Editor" afin qu'ils apparaissent dans le menu contextuel, et une boîte de dialogue s'affichera lorsque l'événement est déclenché pour remplir des valeurs pour tous les paramètres vous définissez sur vos événements

Encryption/Signing Key Generator

4.19

Il est dorénavant possible de configurer complètement la signature et le cryptage des fichiers .pak à partir de l'éditeur via un seul panneau de paramètres unifiés. Ces fonctionnalités peuvent désormais être contrôlées dans Paramètres du projet, sous la section Projet, dans le panneau Crypto.

Unreal Engine 4 – Guide des mises à jour (4.20)



À partir de ce panel, il est possible de générer ou de supprimer des clés pour la signature ou le cryptage, ainsi que d'activer ou désactiver les différents types de signature et de cryptage disponibles. Les fonctionnalités suivantes peuvent être activées ou désactivées à partir de ce menu:

- Encrypt Pak .ini Files - Crypte tous les fichiers .ini présents dans les fichiers .pak du projet. Cela empêche l'extraction facile ou la falsification des données de configuration pour votre produit, à un coût d'exécution minimale.
- Encrypt Pak Index - Crypte l'index de fichier pak, ce qui empêche UnrealPak d'ouvrir, de visualiser et de décompresser les fichiers pak de votre produit, à un coût d'exécution minimale.
- Crypter les fichiers UAsset- Crypte les fichiers uasset dans le fichier pak. Les fichiers uasset contiennent des informations d'en-tête sur les actifs à l'intérieur, mais pas sur les données réelles elles-mêmes. Le chiffrement de ces données fournit une sécurité supplémentaire à vos données, mais ajoute un

Unreal Engine 4 – Guide des mises à jour (4.20)

petit coût d'exécution et une augmentation de l'entropie des données, ce qui peut avoir un impact négatif sur la taille des correctifs.

- Crypter les ressources- Crypte complètement tous les biens dans le fichier pak. Notez que ce paramètre a un effet mesurable sur les performances d'E / S des fichiers d'exécution et augmente la quantité d'entropie dans vos données finales, ce qui rend le système de correction de distribution moins efficace.

Mesh Description Format

4.20

UE4 propose un nouveau format avec une abstraction de plus haut niveau qui peut représenter n'importe quel type de mesh dans le moteur. C'est un processus graduel qui améliorera le flux de travail.

Tous les meshes (statique, skeletal et autres meshes potentiels tels que le terrain et BSP) peuvent avoir **la même représentation interne** avec une certaine interchangeabilité, dans une certaine mesure. Même si ça reste de haut niveau, car on se doute bien qu'une opération booléenne sur un BSP ne peut pas être "stockée" et "traitée" comme un TriMesh. Mais pouvoir les manipuler à un haut niveau d'abstraction de la même façon, c'est top.

La plupart des outils de géométrie UE4 fonctionneront sur tout type de maillage basé sur le format de géométrie.

Tout Mesh utilisant le nouveau format peut être examiné et modifié à l'aide d'une API standard permettant **une modification à l'exécution, native ou scriptée**, ouvrant de nombreuses possibilités pour le contenu généré de manière **procédurale**. Et ça, c'est probablement ce qui me plaît le plus ici... Depuis quelques versions, on commence à avoir une marge assez importante pour faire du procédurale, ce qui devrait aboutir à d'extraordinaires productions dans les années à venir. Personnellement, c'est un terrain qui m'intéresse beaucoup,

Unreal Engine 4 – Guide des mises à jour (4.20)

mais il faudrait que je penche d'avantage sur la question. Le moment est peut-être venu...

Les maillages seront importés directement dans ce format avec la possibilité de **préserver les représentations de maillage de niveau supérieur**, telles que les quads ou la dureté des arêtes. Jusqu'à aujourd'hui, ceux-ci sont perdus lors de l'importation d'un maillage statique ou squelettique.

Le nouveau format de maillage est structuré en interne de sorte que **les modifications peuvent être effectuées en temps réel**, même pour les maillages les plus complexes.

Dans cette version, seul le mesh statique est converti pour utiliser le nouveau format de maillage. Les utilisateurs ne remarqueront aucune différence dans leur flux de travail quotidien et les assets eux-mêmes ne changeront pas. Actuellement, les nouvelles données sont automatiquement créées à partir de l'ancien format et mises en cache dans le DDC.

PLUGINS ET LOGICIELS TIERS

Authentification Steam

4.20

L'authentification Steam a été ajoutée! Les jeux peuvent désormais ajouter un composant de gestionnaire de paquets qui s'interface avec les API d'authentification de Steam, permettant d'annoncer correctement leurs serveurs, de gérer les interdictions VAC / éditeur et de fournir une meilleure validation des clients. S'il est activé, les clients rejoignant un serveur doivent maintenant être authentifiés par Steam avant d'être autorisés à jouer. Par défaut, les clients qui échouent l'authentification sont expulsés du serveur.

Shotgun Integration (Early Access)

4.20

[Shotgun](#) est un outil de gestion de production très populaire utilisé par de nombreux studios de cinéma, de télévision et de jeux pour diriger leur management d'artistes en termes d'assets à produire et les dépendances.

Il y a vraiment pas mal de fonctionnalités intéressantes, je vous laisse découvrir tout cela dans cette vidéo:

<https://vimeo.com/208865883>

Sous UE4, on peut désormais directement faire appel au Shotgun Panel comme ci-dessous:

Unreal Engine 4 – Guide des mises à jour (4.20)



Ainsi, on peut vérifier directement l'activité du projet sur lequel on travaille. On peut importer directement les assets dans le Content Browser. Cela fonctionne également directement dans le menu contextuel des assets (clic droit sur l'actor par exemple).

Source: [Thread Original](#)

Importation en général

Importation Automatique des Assets

4.7

Que ce soit pour ajouter des modèles, des textures ou des sons, il suffit maintenant de les mettre dans le répertoire "Content" du projet pour qu'ils soient automatiquement "importés" dans le projet. Les conversions nécessaires seront automatiquement effectuées.

C'est d'autant plus intéressant quand on est amené à faire des modifications sur ses assets avec un éditeur externe. Plus besoin de devoir réimporter.

Notons qu'il est possible de désactiver cette fonctionnalité dans les préférences de l'éditeur ("Loading & Saving" > **Monitor Content Directories**).

OpenEXR: Textures High Dynamic Range

4.7

Les textures 2D **High Dynamic Range** (HDR) peuvent maintenant être importées sous la forme de fichiers OpenEXR (disponible précédemment uniquement pour les Cube Maps).

OpenEXR
Open Source Plug-Ins for After Effects

On peut stocker des images avec des canaux allant jusqu'à 16 bits.



Full Scene Importation

4.12

Avec Full Scene Importation vous pouvez **importer une scène complète réalisée avec Max / Maya / Blender** dans l'éditeur. Rien que ça !!!!

Un nouvel asset Scène FBX est créé pour permettre de ré-importer.

La version 4.12 permet d'importer des scènes qui incluent des skeletal meshes.

Pour essayer la fonctionnalité, allez dans File > Import Scene, et sélectionnez un environnement FBX à importer. Rien que ça, c'est une super avancée !



Actor fusion

4.12

Vous pouvez maintenant facilement **fusionner les mesh actors ensemble** à droite dans la fenêtre de niveau.

Cela permet d'obtenir de meilleures performances.

Leap Motion

4.7

Alors là, trop cool. Vous connaissez cet appareil? Sinon, allez faire un tour sur www.leapmotion.com

Par défaut, l'éditeur supporte le plug in permettant de s'interfacer avec. L'appareil ne coûte que 90€ et c'est vraiment épatant ce qu'on peut faire avec.



Cela passe par un plugin, mais l'API est entièrement exposée vers les Blueprints et le C++.

J'en ai acheté un pour ma part, et c'est vraiment simple à programmer sous UE4. Cela peut même se coupler avec votre casque de réalité virtuelle (avec une extension USB). Ce n'est clairement pas un gadget!

Autodesk Maya

Live Link Maya

4.19

Diverses améliorations sur le système de live link, avec notamment des plug-ins live link Maya et plugin Motionbuilder.

Les modifications réalisées sous Maya sont ainsi directement visible sous l'éditeur UE4 comme [en témoigne cette vidéo](#).

La fenêtre de l'interface utilisateur Maya Live Link peut être activée via la console MEL avec la commande **MayaLiveLinkU**. Une fois activé, il faut copier les fichiers binaires associés à votre version de Maya dans le dossier Maya Plug-ins et l'activer via le gestionnaire de plug-ins Maya.

Stream Active Camera: Même les manipulations sur la caméra active sous Maya auront une incidence sur celle de l'éditeur. Certains bogues ont été également corrigés (concernant le rig facial par exemple).

Nvidia Ansel

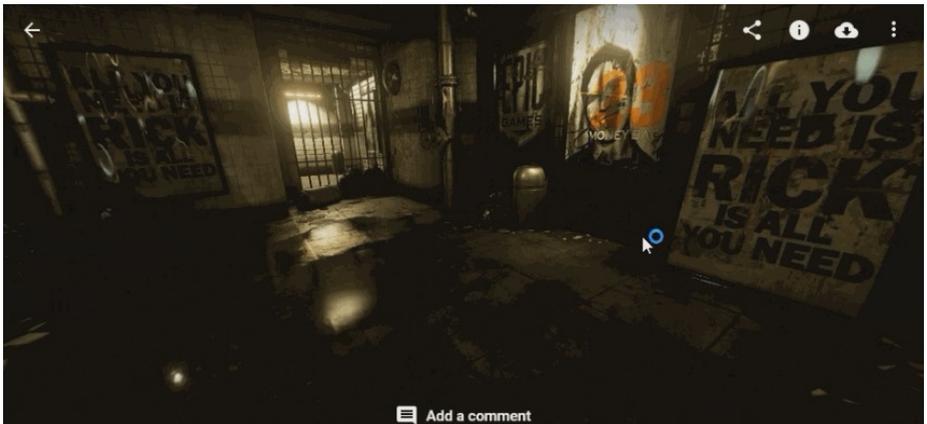
4.14

ANSEL



Prendre des screenshots en Super résolution ou des photos à 360°, et les retoucher comme un pro... c'est possible grâce à Nvidia Ansel - je vous conseille la lecture de [cet article](#).

Sous UE4, cela se matérialise par un plugin qui permet d'accéder à l'outil dans une session standalone. On peut aussi le contrôler via les blueprints grâce à la classe **Player Camera Manager**.

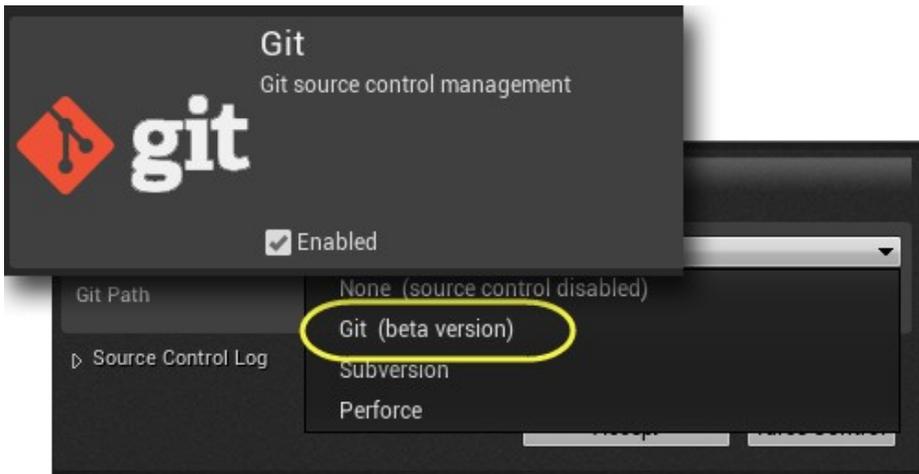


Versioning et Travail collaboratif

Git Source Control Plugin (Beta)

4.7

Merci à [Sébastien Rombauts](#) pour son excellente contribution sur le [plugin Git source control](#) pour Unreal Engine!



Bon, c'est très bien pour faire du versioning quand on travaille seul sur un projet. Toutefois, pour bosser en équipe, je vous conseille de passer par Perforce qui est l'outil conseillé également par Epic Games.

ArchViz

Character Controls

4.9

Ce plugin permet d'ajouter des contrôles plus poussés pour des applications d'architecture afin de créer des visites plus réalistes. Si c'est votre domaine, je vous conseille donc d'essayer ce plugin.



Technologie DataSmith

4.17

Pour l'**architecture et le design**, la boîte à outils simplifiée et la technologie [datasmith](#) est beaucoup plus pratique, comme cela a été démontré lors du [SIGGRAPH 2017](#). Datasmith aide artistes et concepteurs à simplifier l'importation de données dans Unreal Engine depuis plus de 20 sources de CAO / DCC, y compris Autodesk 3ds Max. Il en résulte un gain de temps significatif.

Voici un avant/après de ce qu'il est possible de faire:

Unreal Engine 4 – Guide des mises à jour (4.20)



Pour les plugins

Global Shader Support pour Plugins

4.17

Vous pouvez maintenant **ajouter de nouveaux shaders à l'intérieur des Plugins et des projets** (Plugin et modules du projet avec shaders globaux nécessitent l'utilisation de la PostConfigInnit en phase de chargement).

Plugins et projets peuvent fournir leurs propres Shaders. Vous pouvez regarder le plugin LensDistortion:
(Engine/Plugins/Compositing/LensDistortion).

Personnellement, je n'ai pas rencontré le problème, mais je me souviens que mon collègue Marien vociférait contre Epic à cause de cela, car pour un plugin ça n'allait pas, il fallait recompiler le moteur il me semble...

Enfin, bref, c'est dispo maintenant, il suffit de respecter les éléments suivants:

- Les fichiers USF doivent être situés dans le répertoire privé.
- Les appels à IMPLEMENT_SHADER_TYPE, doivent inclure le chemin complet du fichier de shaders USF (comme «Engine/Private/Foo.usf» ou «plugin/FooBar/Private/MyComputeShader.usf»).
- Les fichiers C ++ générés doivent être mis en correspondance dans le répertoire /Engine/Generated/ virtual directory.
- Shaders /directory nécessite au moins un répertoire privé ou public. Le répertoire généré / ne devrait pas exister.

Unreal Engine 4 – Guide des mises à jour (4.20)

Structure du fichier du répertoire Shader

Les fichiers de shaders ont été divisés en USF et une nouvelle extension de fichier USH.

- Les fichiers USH sont utilisés pour les fichiers d'en-tête de shaders qui ne contiennent pas un point d'entrée de shaders et qui ne devraient contenir que d'autres fichiers de shaders.
- les fichiers USF seraient les seuls fichiers ne pouvant être inclus qui ont les points d'entrée de shaders.

De plus, les fichiers de shaders ont été séparés sous les dossiers publics et privés. Tout comme avec le code source du moteur, l'API publique doit être placée dans des fichiers dans un dossier public, alors que les implémentations sont placés dans le dossier privé.

Les dépendances entre les plug-ins

4.17

Les plug-ins peuvent maintenant déclarer des dépendances sur d'autres plug-ins que le moteur utilisera pour activer automatiquement et charger ces plug-ins.

Les dépendances peuvent être répertoriées dans un .uplugin en utilisant la même syntaxe que les fichiers .uproject.

Replication Driver / Replication Graph

Expérimental

4.20

Le plugin **Replication Graph** fournit un système de réplication optimisé pour les jeux avec un grand nombre d'acteurs et de joueurs. Ahah, c'est [pour Fornite ça](#) ? Le système fonctionne en construisant une série de nœuds personnalisés qui peuvent centraliser les données et le calcul. Ces nœuds persistent sur plusieurs trames et peuvent être partagés par les connexions clientes, ce qui réduit le travail CPU redondant et permet de regrouper les acteurs dans des nœuds en fonction de règles de mise à jour spécifiques au jeu. Il est utilisé dans Fortnite Battle Royale.

CONCLUSION

Le mot de la fin

Bon, ça fait beaucoup de choses à assimiler tout ça? Souvent je reçois des courriers de lecteurs qui expriment à la fois leur joie de constater que le moteur se dote de nouvelles fonctionnalités, mais aussi leur frustration de ne pas pouvoir suivre le changement...

Je comprends cette peur, **elle est naturelle**: nous vivons dans un monde de progrès en constante accélération – nous pourrions disserter sur la nature philosophique du progrès, mais il n'en reste pas moins qu'il faut chaque jour s'adapter, apprendre de nouvelles choses, et en désapprendre d'autres.

Au bout d'un moment, on s'y perd et Epic Games n'a probablement pas trouvé la recette miracle pour que l'on puisse suivre ces changements dans les meilleures conditions. Je pense notamment à tous ceux qui sont en production, qui doivent développer et qui ont déjà du mal à tenir leur délai. Comment trouver le temps dans ces moments là de suivre le dernier Twitch d'Unreal, d'extraire de la documentation en ligne les nouvelles fonctions, d'étudier les derniers projets et templates, ou les tutoriels en vidéo...

Cette collection de savoir éparse ressemble à ce qu'on voit dans un kaléidoscope: il faut longtemps et beaucoup d'expérience pour relier chaque morceau les uns aux autres.

Malheureusement, la solution n'est ni dans un livre, ni dans une série de tutos, ni dans une formation de 15 jours où vous ne pourrez que survoler les choses de façon statique. Cela fait des mois que cela me trotte en tête: comment vous aider d'avantage à acquérir une expertise sur ce moteur, vous permettre de facturer vos développements ou votre consulting comme il se doit ?

Unreal Engine 4 – Guide des mises à jour (4.20)

Nous manquons cruellement de développeurs, d'artistes, de gens qualifiés sous Unreal Engine tout simplement. Et la demande est de plus en plus importante. Certains clients me demandent de les aider à recruter pour leurs prochains projets. Mais, et c'est d'autant plus vrai dans le monde francophone, s'il y a pléthore de personnes formées à Unity, il y a un manque manifeste sous Unreal Engine – et c'est encore vrai aujourd'hui en 2018.

En attendant, merci d'avoir pris le temps de lire ce livre et n'hésitez pas à le conseiller autour de vous et me faire des commentaires en retour.

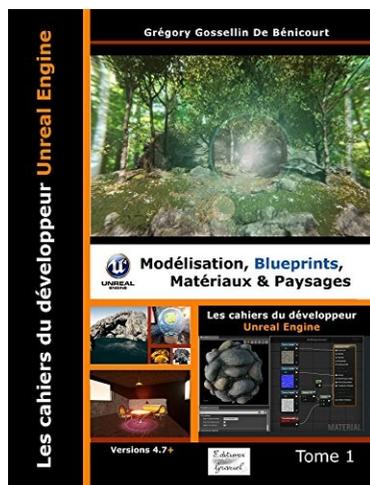
@très vite!



Livres en français sur Unreal engine 4

Si vous n'avez pas lu tous les tomes de la collection "Les cahiers d'Unreal Engine", voici un résumé de chaque tome:

Tome 1



Pour commencer cette série sur Unreal Engine, il fallait inévitablement commencer par les bases. Dans ce premier tome, nous allons vous aider à faire vos **premiers pas sous l'éditeur**.

Mais plutôt que de vous lister une à une chaque fonction, nous allons vous les faire découvrir par la pratique, en créant **votre premier jeu** sous UE4. Ce jeu sera complété par d'autres tomes au fil de vos lectures.

Tout d'abord, nous allons apprendre à utiliser les fonctions de **modélisation** de l'éditeur qui nous permettent de créer des objets à partir de formes de base (les «brushes») et leur appliquer des matériaux. C'est l'un des points forts de cet éditeur: il possède son propre outil de modélisation. Certes, il n'a pas la puissance d'outils comme Blender, Maya ou 3DSMax, mais sa simplicité et son intégration permettent de concevoir très rapidement la structure générale d'un niveau de jeu. Ce sera l'occasion de créer un premier bâtiment et son escalier, à aménager l'intérieur de ce bâtiment avec des objets externes. Ces objets peuvent être téléchargés, importés à partir d'autres projets ou modélisés par vos soins avec vos outils préférés.

Unreal Engine 4 – Guide des mises à jour (4.20)

Puis, nous allons créer nos premiers **Blueprints**: sortes de composants alliant objets 3D et programmation visuelle. Grâce aux Blueprints, on peut associer un «comportement», une fonctionnalité à un objet. Ainsi, nous allons créer plusieurs objets que nous placerons dans le bâtiment et qui auront un comportement particulier: boutons d'allumage de lumières, spots avec détecteurs de présence, système d'ouverture de portes, etc. Nous en profiterons pour vous montrer un petit aperçu de «Matinee», l'éditeur de cinématiques qui permet, entre autres, de créer des séquences complexes d'animation.

Ensuite, nous explorerons un domaine assez important: la **création de matériaux**. En effet, Unreal Engine possède un système très abouti de programmation des matériaux. Si vous êtes graphiste et que vous modélisez vous-même vos propres objets, il faudra en passer par là. Le Rendu Physique Réaliste ou PBR (Physically Based Rendering) est un modèle d'illumination pour créer des rendus 3D qui respectent les lois de la physique. Le rendu est époustouflant et se rapproche d'avantage de la synthèse d'images. C'est un domaine très vaste, mais au travers de la création de plusieurs matériaux, nous allons découvrir les principales fonctionnalités associées à ce domaine: les textures, les normal maps, la transparence et les masques, les matériaux émissifs, les matériaux animés et dynamiques, les déformations via le World Displacement, les Subsurface colors, la réfraction, etc. Nous créerons également nos propres fonctions de matériaux, ainsi que nos instances à partir de modèles.

Un chapitre très important de ce premier tome est la **modélisation de paysages** en utilisant les fonctions internes de l'éditeur. Pour notre jeu, nous allons donc modéliser un vaste paysage et y intégrer notre bâtiment. Nous explorerons toutes les fonctions de modelage du paysage (le sculpt): les différentes outils, brosses, comment créer des chemins en utilisant les outils de plates-formes et rampes d'accès. Nous verrons également comment «copier-coller» des parties du paysage. Puis, nous aborderons la «peinture» du paysage en utilisant plusieurs calques (layers). Grâce aux connaissances que nous avons pu acquérir sur les matériaux, nous allons créer un matériau très particulier ayant la capacité de s'adapter à l'inclinaison du terrain. Ainsi, des rochers pourront surgir des dunes de sable, juste en élevant le niveau du terrain avec les outil de sculpting. En créant ce

Unreal Engine 4 – Guide des mises à jour (4.20)

genre de matériaux, la modélisation d'un paysage devient un délice et permet de décupler sa créativité.

Enfin, nous apprendrons à utiliser un autre outil fantastique qui fait la richesse de l'éditeur: l'**outil «foliage»**. On l'utilise pour ajouter au paysage des éléments comme la végétation, les roches, ou tout objet que l'on souhaite avoir en grande quantité mais avec lesquels on n'entre pas directement en interaction dans le jeu (autrement que par la collision par exemple). Cet outil extraordinaire permet de peindre un paysage avec des arbres comme on le ferait avec un pinceau et une toile. Il est hautement paramétrable et permet des optimisations spectaculaires, permettant ainsi de gérer des centaines milliers d'instances dans votre scène, tout en restant à un niveau acceptable de jouabilité. Il n'y qu'à voir cette démonstration «**Open Worlds**» présentée lors de la Game Developer Conference de 2015. Nous aurions pu assister à un film en images de synthèse où un cerf volant parcourt un vaste paysage... mais non, il s'agissait bien d'une animation temps-réel. Nous verrons comment créer de l'herbe ou importer des arbres et des rochers, et comment les associer au paysage.

Pour terminer, nous intégrerons un **personnage** au paysage – avatar que nous allons contrôler. Nous n'entrerons pas dans les détails de l'animation du personnage ou dans la création et l'importation de modèles existants, car ce sera l'objet du second tome, mais nous verrons comment gérer les entrées clavier et souris par exemple, et comment profiter des «**Templates**» disponibles en nous les appropriant.

Tome 2



Vous trouverez les sources du premier projet sur le site accompagnant ce livre. Vous n'êtes pas obligé d'utiliser ce projet, vous pouvez démarrer directement à partir d'un autre paysage que vous aurez vous-même modélisé ou importé.

Toutefois, si nous avons été très **progressif** dans notre approche de l'éditeur au travers du **premier tome**, nous considérons que **toutes les notions abordées** sont aujourd'hui **acquises**.

Dans un premier temps, nous allons revenir à la toute dernière partie du tome 1, l'importation d'un «Template» de type «Third Person Shooter» (TPS). C'est un modèle doté d'une caméra qui le suit dans tous ses déplacements et qui permet de voir le personnage à l'écran. L'objectif de ce second tome est de se concentrer sur **le personnage**: l'importation et l'animation de modèles externes, ses animations, etc.

A cet effet, nous allons nous plonger dans l'étude de l'**éditeur de mesh** **et** **d'animations**.

Nous en profiterons pour voir **les différents types d'animation**, du Shape Key (morphing) au skeletal bone (déformation via un squelette), en passant par la création de séquences, de Blend Spaces, d'animations composites, de montage, etc.

Puis, nous rentrerons dans le cœur de l'animation: le **Blueprint d'animation** qui utilise une machine d'états. Enfin, nous étudierons un Blueprint particulier, «**character**», qui est un modèle hautement paramétrable et dont on peut se servir dans notre jeu pour créer tout type de personnage.

Unreal Engine 4 – Guide des mises à jour (4.20)

Dans une seconde partie, nous allons **créer notre propre modèle de personnage**. Vous ne vous sentez pas l'âme d'un infographiste? Ce n'est pas un soucis: nous allons utiliser **MakeHuman**, un logiciel gratuit et open source permettant de créer des personnages avec un nombre impressionnant de paramètres.

Nous allons voir comment l'importer dans notre jeu et le substituer à notre «Template». L'approche est délicate et demande de suivre scrupuleusement la procédure indiquée, c'est à votre portée, même si vous n'avez jamais utilisé un outil de modélisation de votre vie.

Nous explorerons les capacités d'Unreal Engine à **simuler des objets souples** («clothing») au travers de son moteur physique **PhysX**. Grâce au **SDK Apex**, nous animerons la chevelure de notre personnage qui se mettra à voler au vent et à suivre chacun de nos mouvements. De la même façon, nous pourrions animer une robe, une cape, une corde, un rideau, voire un drapeau ou une nappe. La technique reste la même.

En important notre personnage, nous allons hériter de toutes les animations déjà présentes dans le modèle, mais aussi de toutes celles que nous pourrions être amené à importer ou télécharger à partir du Market Place. D'ailleurs, nous allons profiter d'un pack gratuit pour **ajouter des animations de tir** à notre personnage. Nous vous enseignerons comment lui ajouter une arme, la ranger dans le dos et la déployer au moment du tir. Nous verrons comment **tirer une balle** et gérer son impact via le système de collision. Puis nous intégrerons l'ensemble dans la machine d'état. Ensuite, nous verrons comment appliquer les différents **dommages**, comment **détruire des objets** et les voir tomber en morceaux progressivement, comment aussi gérer la **mort d'un personnage**. Nous utiliserons également un système basique d'inventaire pour gérer les cartouches et ramasser celles qui traînent dans le décor (comme celles des personnages morts).

Nous aurions pu nous arrêter là, mais c'est un peu ennuyeux non? Nous allons donc ajouter quelques **Personnages Non Joueurs (PNJ)**, des ennemis. Bien sûr, nous allons pouvoir exploiter tout ce qui a été créé jusqu'à présent, avec de légères modifications puisque ces

Unreal Engine 4 – Guide des mises à jour (4.20)

PNJ se déplacent sans que vous ayez besoin de prêter un clavier à votre voisin de pallier.

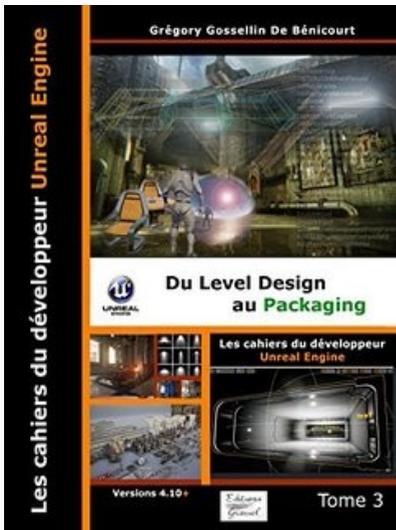
Le PNJ a sa **propre intelligence**: Ils se promènent dans le décor, font un tour de garde, repèrent le joueur à une certaine distance et sous un certain angle et déclenchent le branle-bas de combat! Là, il faudra courir ou être plus rapide qu'eux au tir! Ils vont vous traquer sans relâche, jusqu'à ce que vous soyez mort ou que vous trouviez un moyen de vous échapper (sauter d'une falaise sans se tuer par exemple).

Vous devez penser que tout ceci est très difficile à programmer: ce n'est pas faux. Mais heureusement, Unreal Engine fournit à cet effet un outil très intéressant: le **Behaviour Tree**, littéralement l'arbre comportemental. A l'image de la machine d'état utilisée pour l'animation, les BT permettent de paramétrer des comportements complexes et ce... assez simplement! Ouf! Ce sera tout de même la partie la plus complexe de ce second tome, mais c'est aussi l'une des plus passionnante.

Nous commençons à avoir tous les ingrédients d'un bon jeu. Pour se changer les idées, nous vous proposons de plonger au cœur de la **gestion des particules** en créant des balles traçantes et en laissant des impacts de balle sur les roches (ces fameux «**decals**» dont nous vous parlions dans le premier tome). De la même façon, vous pouvez créer des tâches de sang sur les objets lorsqu'une balle transperce un personnage, laisser des traces de pas dans le sable quand vous marchez ou faire en sorte que votre personnage prenne feu quand une bonbonne de gaz explose – tout cela relève du même principe: particules et decals.

Nous espérons que tout cela vous a motivé. Accrochez-vous bien car à la fin de ce second tome, vous allez déjà pouvoir jouer à votre propre jeu, vous devriez être capable d'ajouter de nombreuses fonctionnalités que nous n'avons pas abordées ici.

Tome 3



Ce troisième tome vient **conclure** la partie qui a pour but de vous initier à Unreal Engine dans le cadre de la création d'un jeu vidéo.

Dans un premier temps, nous revenons sur le **Level Design**. Dans le tome 1, nous avons appris à utiliser l'éditeur pour modéliser un paysage et un bâtiment, mais nous n'avons qu'effleuré ce qu'est la gestion d'un niveau de jeu : passer d'un niveau à un autre, utiliser le «**Level Streaming**» et conserver des informations en passant d'un niveau à un autre (**données**

persistantes). Nous allons nous focaliser également sur la **gestion de l'illumination** et étudier les techniques de pré-calcul que sont les **Lightmass** et les **Lightmaps**. Ensuite, nous ferons un point sur les réflexions: les **Screen Space Reflection**, ainsi que les composants de type «**Scene Capture**» et «**Reflection Capture**». Puis, nous nous attaquerons aux effets spéciaux hors particules: les **post-process**. Nous ferons bien entendu un tour complet des différents effets, et nous en profiterons pour en créer 3 nouveaux au travers de Post-process Material. Enfin, nous nous attarderons sur la **gestion des caméras**, le passage d'une caméra à une autre avec un effet de fondu via les Blueprints et la mise en place d'une **contrôle relatif caméra du personnage**. Nous vous montrerons comment animer une caméra le long d'une courbe «spline» et effectuer des transitions sous Matinee. Nous en profiterons pour vous présenter le nouveau venu destiné à remplacer Matinee : Le **Level Sequence Editor**.

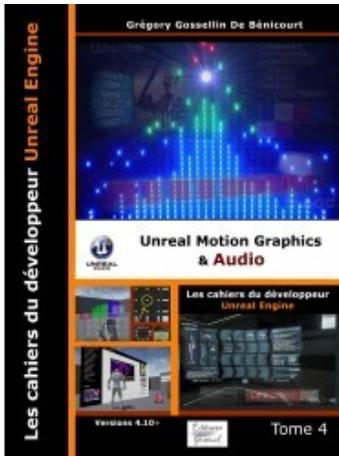
Mais créer un jeu vidéo, c'est d'avantage que de s'occuper du level design et programmer un **gameplay** : il faut aussi créer des **menus**, permettre de faire des **sauvegardes** des parties, gérer des **sons**, des **musiques** et des **voix**. Il peut être important d'assurer la **localisation** (traduction) de votre jeu si vous souhaitez pouvoir l'éditer dans

Unreal Engine 4 – Guide des mises à jour (4.20)

plusieurs langues. Enfin, et surtout, vous devrez passer par le **packaging** afin de créer une version distribuable de votre jeu, ainsi que des **patches** correctifs, et ce, vers chaque plate-forme ciblée. Nous en profiterons également pour réaliser un export en **HTML5** et le tester avec un serveur web local.

Ce qui vous attend est relativement dense, d'autant plus que nous considérons que vous n'êtes plus débutant. Nous passons donc la seconde vitesse: préparez-vous.

Tome 4



Les 3 premiers tomes de la collection nous ont permis de prendre en main Unreal Engine et d'**apprendre à créer un jeu de A à Z**.

Nous avons privilégié une approche à la fois **progressive** et **kaléidoscopique** du sujet, passant quelques fois sur certains détails pour nous concentrer sur d'autres, comme on le ferait lors d'un séminaire de formation.

Ce **4ième tome** n'est pas entièrement dédié au jeu vidéo, même s'il contient de nombreuses références à ce domaine. Mais il nous paraît important de rappeler qu'Unreal Engine n'est pas seulement un outil pour créer des jeux vidéo – il est tellement plus que cela !

La première partie est orientée vers la maîtrise de l'**Unreal Motion Graphics (UMG)** qui est le nouveau système d'Unreal Engine pour **créer des interfaces utilisateurs**: du HUD à la création de menus, en passant par les composants de type widgets 3D. Vous apprendrez à utiliser le «**Widget Designer**», un outil puissant pour créer des interfaces: chaque type de widget sera examiné à la loupe, ainsi que la programmation Blueprint correspondante. Nous vous montrerons également comment utiliser le système d'animation des widgets, très proche de Matinee que vous connaissez déjà.

La seconde partie se concentre sur la **gestion de l'audio**: les musiques, les sons et les voix. Vous verrez comment intégrer des sons 8 pistes de type 7.1 ou utiliser n'importe quel autre son dans une scène 3D, en gérant son atténuation, la réverbération du milieu, etc. Les termes techniques comme «Sound Wave», «Sound Cue», «Sound Class» ou encore «Sound Mix» n'auront plus de secret pour vous.

Unreal Engine 4 – Guide des mises à jour (4.20)

La lecture des 3 premiers tomes est un plus car il y a de nombreuses notions que nous avons abordées dans ces derniers, et sur lesquels nous ne reviendrons pas dans ce tome. Ce 4ième tome est le premier d'une nouvelle série attaquant Unreal Engine sous un angle plus «spécialisé». Nous avons essayé de vous fournir une information la plus complète possible, même là où la documentation officielle faisait défaut.

Êtes-vous prêt à maîtriser les arcanes des interfaces utilisateurs et de l'audio sous Unreal Engine ?

Tome 5

Le **cinquième tome** complète la formation à Unreal Engine par la **mise en réseau de l'application créée**.

L'ajout de fonctionnalités «réseau» à un jeu ou à une application est réellement quelque chose de stimulant. Quand on pense à cette fonctionnalité pour un **jeu vidéo**, on sait de quoi il s'agit: le multijoueurs en ligne.

Toutefois, on ne pense pas assez aux autres possibilités dans les domaines de **l'enseignement**, de **l'immobilier**, de **la santé**, de **l'industriel**, du **marketing**, voire du **militaire**: pouvoir travailler, visiter ou expérimenter à plusieurs un même procédé, une même simulation.

Travailler sur une combinaison de réalité virtuelle permettant littéralement de «voir son corps» en virtuel et surtout de pouvoir se déplacer, saisir des objets virtuels, etc., c'est déjà fascinant en soi. Mais lorsque vous enfilez la combinaison, qu'un autre gars fait de même à l'autre bout de la planète et que vous vous dirigez vers lui pour une première poignée de mains... c'est une sensation indescriptible. On dit qu'**un rêve partagé par plusieurs personnes est déjà une forme de réalité**. Tout cela n'aurait pu être possible sans l'aspect «réseau» justement.

Maintenant, imaginez... Un chirurgien dirige son opération en pilotant un robot sur un autre continent. Ou encore, un participant à une conférence dans une immense salle virtuelle, avec des personnes de tout pays réparties dans toute la pièce. Ou bien, la visite d'un appartement par un prospect étranger accompagné de l'agent immobilier sans qu'aucun des deux n'ait à se déplacer. Autre exemple, un mécanicien «en ligne» va montrer à son client comment changer les durites de son véhicule... les possibilités sont infinies. Unreal Engine 4 est un outil fabuleux et lorsqu'on maîtrise ses capacités réseau, c'est une **porte ouverte** vers un **océan de possibilités**.



Unreal Engine 4 – Guide des mises à jour (4.20)

Disposer d'un **framework** «clé en main», simple à maîtriser et à faire évoluer est en partie l'objet de ce livre. Mais il faut comprendre toutes les subtilités du développement réseau. Ces subtilités sont rendues tangibles au travers d'exemples simples et progressifs.

Un exemple très simple vous montre qu'en **5 minutes**, vous pouvez créer un jeu et y connecter plusieurs joueurs via d'autres postes. C'est un exemple trompeur car tout semble très simple, mais c'est aussi que le travail est vraiment mâché. Cela a au moins le mérite de vous mettre directement dans le bain, d'épuiser une partie de votre curiosité afin que vous puissiez focaliser votre attention sur la suite. Car il y a beaucoup de **concepts à comprendre** et un peu de **savoirs-faire à accumuler** avant de se lancer dans un projet de création d'un jeu ou d'une application «en ligne». C'est donc ce que nous allons étudier dans un second temps.

- **bases d'un framework multijoueurs en ligne**, avec la création d'un lobby, c'est à dire d'un hall permettant de créer une session serveur, de rechercher des parties et de joindre l'une d'elle en tant que client. Nous allons souvent utiliser les mots de jeu en ligne, de joueurs... mais au final, **lorsqu'on crée une application, on parle d'utilisateur et de clients**. Mise en place un certain nombre d'éléments pour s'affranchir des limitations des Blueprints et créer un **framework avancé**, avec une meilleure gestion des joueurs.
- développer un système de **chat textuel et vocal**,
- utiliser l'«instant replay» pour **filmer des parties réseau** et rejouer l'intégralité ou une séquence de la partie.
- utilisation des outils de débogage et de mesure de performance du jeu sur le réseau.
- configurer un **serveur dédié** sous **Windows, OSX et Linux**. La mise en place d'un tel serveur n'est pas aussi simple qu'on pourrait le penser au préalable. Cela nécessite, par exemple, une re-compilation complète de l'éditeur mais également, une configuration plus avancée de votre projet.
- joindre une partie nécessite de connaître l'adresse du serveur qui héberge la partie.
 - passer par des plates-formes tierces comme **Steam**, son utilisation et comment configurer un projet pour prendre en charge cette plate-forme.
 - nous avons également développé un **serveur sur mesure en PHP/JSON**, hébergé chez vous ou avec votre site web, que vous pourrez réutiliser et améliorer pour vos projets. Nous allons d'ailleurs le développer ensemble !

Unreal Engine 4 – Guide des mises à jour (4.20)

- pour développer un MMO ou avoir des services supplémentaires, vous pouvez passer par des plates-formes **tierces**, autres que Steam, comme **Photon Engine** pour le temps-réel ou **XtraLife BaaS** pour le tour par tour.
- le multijoueurs sur **Android, iOS, Amazon, Facebook, PS4, Xbox One, Nintendo Switch** etc. qui utilisent les mêmes fonctions que celles étudiées dans ce tome.

Conseils de lecture

Au niveau francophone, je vous conseille 2 excellentes sites, ceux de Jean Charpentier et de Jean Meli (les "2 Jeans"). Ils animent également une chaîne YouTube de tutoriels assez bien fournie. Voici les liens:

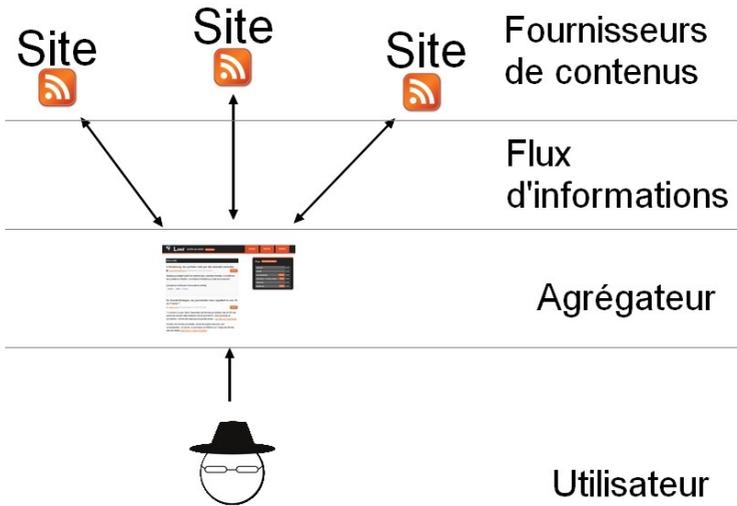
- **Jean Charpentier**: son site creerunjeuvideo.fr et sa chaîne YouTube.
- **Jean Meli, dit "Meletou"**: son site passion3D et sa chaîne YouTube.



Pour suivre une chaîne YouTube, le mieux est de créer un compte YouTube et de vous abonner à la chaîne. Ensuite, régulièrement, vous allez dans [cette section](#) et vous aurez la liste de toutes les nouvelles vidéos produites par ces chaînes, classées par ordre de publication inverse (LIFO pour les intimes).

Pour suivre plusieurs blogs, je vous conseille de passer par les flux RSS. si vous ne connaissez pas les flux RSS, sachez que la plupart des sites web utilisent un moteur de blog ou tout autre système qui lie les nouveaux articles à un flux (qui est un fichier **XML**). Vous pouvez paramétrer certaines messageries pour automatiquement rapatrier les éléments de ces flux ou utiliser un "reader", c'est à dire, une application ou un site (comme [feedly](#) qui remplace Google reader) qui va centraliser tous vos flux RSS. On appelle cela un **agrégateur de flux**. Après, libre à vous de l'utiliser avec votre pc de bureau, votre tablette ou votre mobile – ou les 3 à la fois !

Unreal Engine 4 – Guide des mises à jour (4.20)



Remerciements

Tout d'abord j'aimerais remercier **Tim Sweeney** et toute son équipe pour avoir créé un aussi merveilleux outil et l'avoir partagé avec le monde.

Merci à tous mes amis et contacts qui suivent régulièrement mon blog et qui m'encouragent dans ce domaine vaste et passionnant de la création de jeux.

Et enfin, plus particulièrement, un grand merci à **ma compagne** pour son travail de relecture, pour ses idées et son soutien inconditionnel.