



Introduction à Unity
Deuxième partie
Version 4.6

Guillaume Levieux

Conservatoire National des Arts et Métiers

28 janvier 2015

Déroulé

- 1 L'interface d'Unity
 - Scene window
 - Hierarchy window
 - Inspector window
 - Project window
 - Game window
- 2 Les objets sous Unity
 - Importer un objet sous unity
 - Instancier un objet sous unity
 - Les liens entre objets

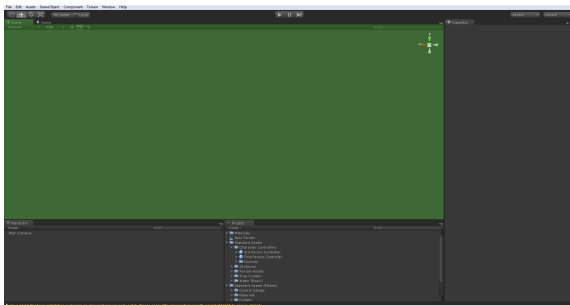
L'interface d'Unity

L'interface d'Unity



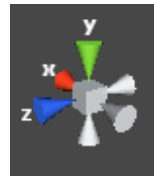
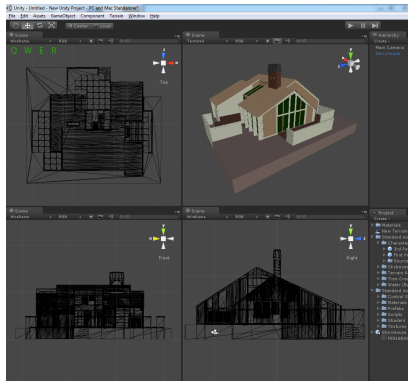
Scene window

- Rendu de la scène en cours de création
- Permet le placement des objets dans l'espace
- Permet d'observer la scène sous différents aspects
- A savoir : maximise toute fenetre par "Espace"



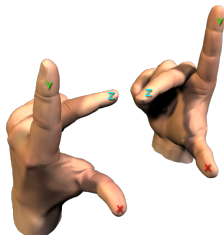
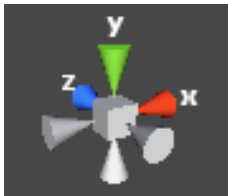
Scene window

- Mode 4 vues
- Une perspective et top left right
- Toutes configurables : point de vue sur le gizmo



Scene window

- Repère main gauche
- Z = profondeur
- Y = hauteur



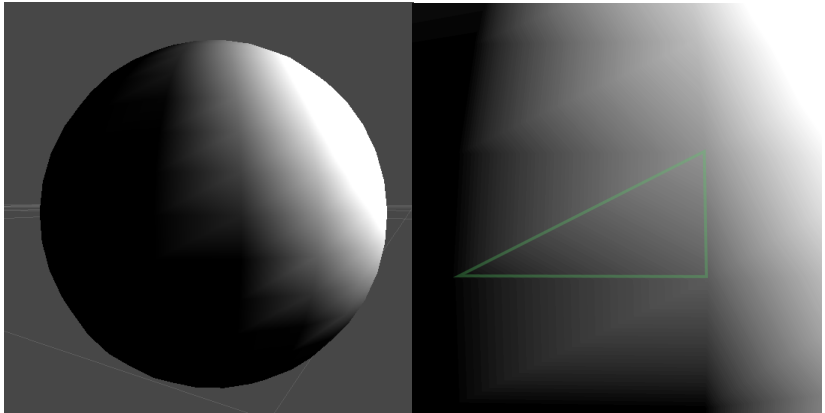
Draw Modes

- Textured : objets texturés
- Wireframe : rendu fil de fer
- Tex+Wire : rendu texturé et fil de fer
- Renderpath : vert pour deffered, jaune pour forward et rouge pour vertex lit
- lightmap resolution : Affiche une grille pour évaluer la définition de la lightmap

Vertex Lit

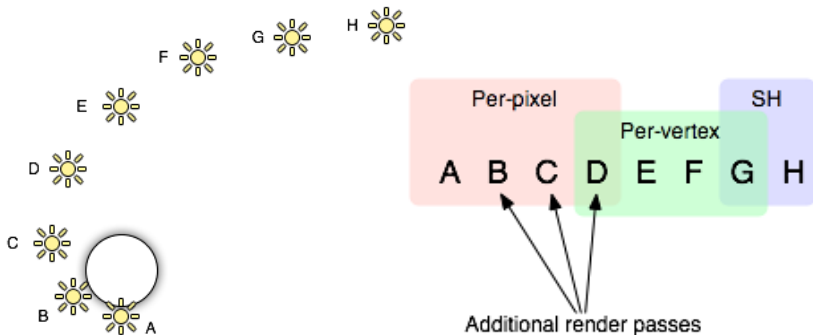
- Illumination calculée pour toutes les lumières, pour chaque sommet.
- Méthode la plus rapide et supportée par le plus de hardwares
- Aucun effet "au pixel" :
 - ombres projetées,
 - normal mapping,
 - light cookies,
 - reflets spéculaires détaillés (au niveau de la facette uniquement)

Vertex Lit



Forward

- Un certain nombre des lumières les plus brillantes en per-pixel
- Les 4 suivantes en vertex lit
- Le reste en Spherical Harmonics

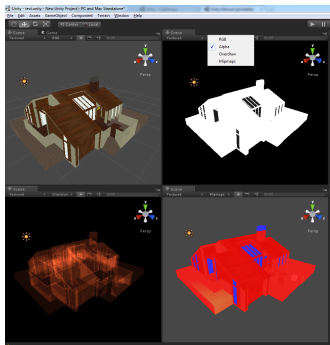


Deferred

- Base Pass : objets rendus pour créer des buffers screen-space de la profondeur, des normales et de la puissance de la spéculaire. C'est une passe standard, mais on garde le z buffer, puis on encode la normale dans le RGB, et la spéculaire dans l'alpha.
- Lighting pass : on utilise les buffers précédents pour calculer la lumière, dans l'espace image ! Les point et les sport lights sont rendues comme des objets 3D, et on calcule comme ça l'illumination de chaque pixel. Chaque lumière ne coute que le nombre de pixels qu'elle affecte.
- Final pass : on rend a nouveau les objets, qui utilisent les calculs précédents pour les mixer avec leur couleur de texture, et tenir compte de l'ambiante et de l'emissive.

Render Modes

- RGB : couleurs normales
- Alpha : rendu composante alpha (du material)
- Overdraw : parties de l'écran les plus dessinées (sans zbuffer)
- Mipmaps : rouge > tex trop grande bleu > tex trop petite



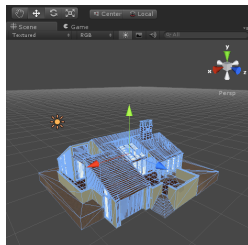
Déplacer la vue

- Clic droit : rotation vue (centre vue)
- alt + clic gauche : rotation vue (centre objet)
- Bouton milieu souris : translation vue
- Molette : zoom
- Gizmo : se ramener à une vue de base
- Double clic dans hierarchy window centre cam sur objet



Maniupler un objet

- Clic gauche : selection objet
- Bouton 1 : déplace vue
- Bouton 2 : déplace objet
- Bouton 3 : rotation objet
- Bouton 4 : agrandir / réduire



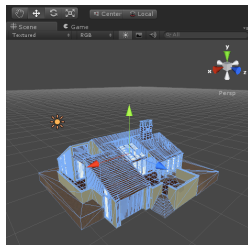
Modifier le pivot

Bouton 1 : center / pivot

- Center : centre de transformation au centre de la selection
- Pivot : centre de transformation au point de pivot du mesh

Bouton 2 : local / global

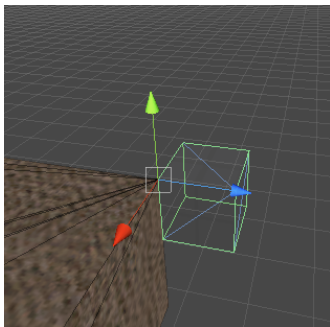
- local : axes alignés sur le repère local
- global : axes alignés sur dans le repère global



Vertex snapping

Appuyer sur V en mode translation (shift + V = toggle)

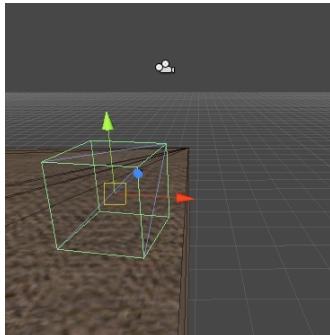
- Permet de choisir un vertex
- Permet de snaper ce vertex sur un autre



Surface snapping

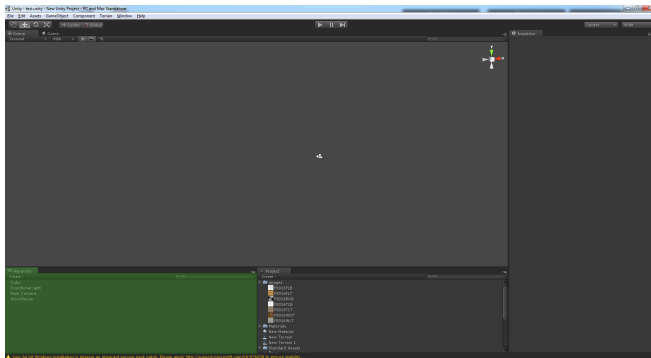
Si deux objets sont équipés de colliders alors appuyer sur ctrl + shift permet de :

- tradater un objet pour le placer au contact d'un autre
- faire tourner un objet autour de l'axe normal à la surface de contact



Hierarchy window

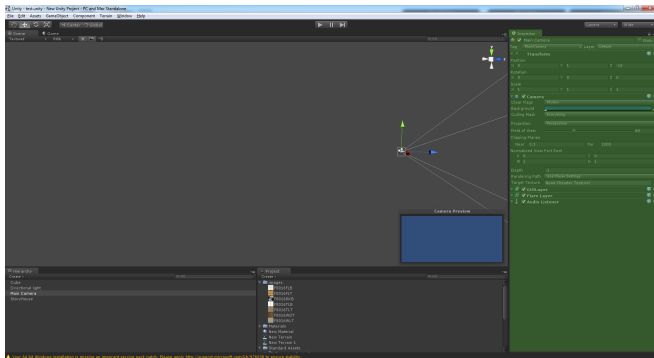
- Liste les objets de la scène en cours de création
- Uniquement les objets instanciés
- Double clic (ou selection puis F) sur un objet = focus



Inspector window

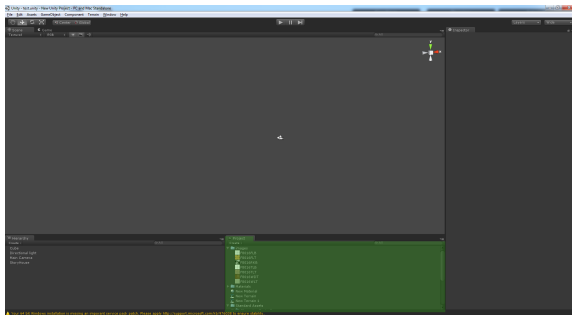
Fenêtre d'édition des paramètres :

- Paramètres d'un objet sélectionné
- Paramètres du projet ou de rendu si choisit dans menu "Edit"



Project window

- Vue du répertoire assets du projet
- Permet de manipuler le contenu de ce répertoire
- Réserve d'objets, pas leurs instances



Project window

A savoir sur la Project Window

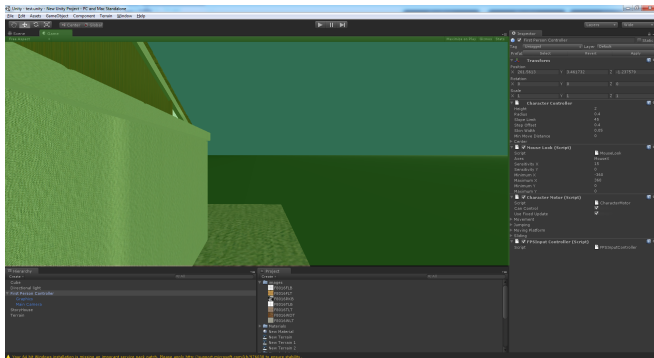
- Comme java, Unity s'appuie sur la structure des répertoires
- Reps : Assets + Library (+ Temp si unity tourne)
- Toute modification du rep Assets automatiquement prise en compte

Attention ! SAUF :

- Nommer, déplacer les fichiers **uniquement** par la project window
- Risque de corrompre les données du projet.
- Tips : "ctrl + d" duplique un objet

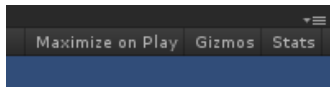
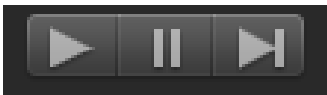
Game window

- Permet de lancer le jeu (boutons play)
- A coté des fenêtres de scène

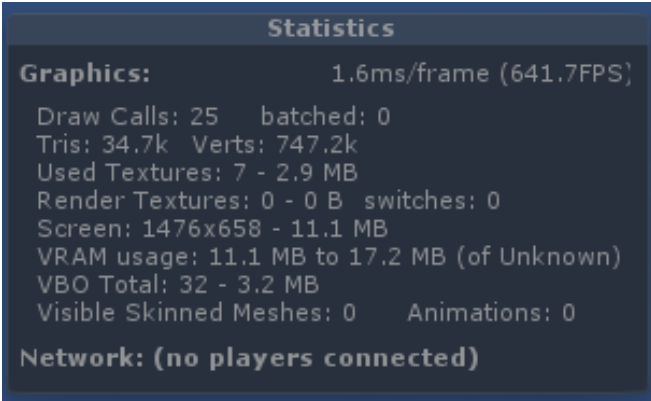


Game window

- Trois boutons : play, pause et step
- Attention : toute modification en mode jeu est **temporaire**
- Visible car interface plus foncée
- Activer "Maximize on play" évite les malentendus
- Fenetre de stats utile.



Fenetre de stats



Statistics

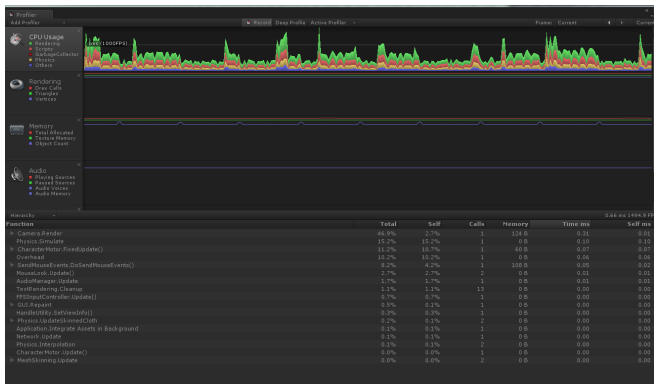
Graphics: 1.6ms/frame (641.7FPS)

Draw Calls: 25 batched: 0
Tris: 34.7k Verts: 747.2k
Used Textures: 7 - 2.9 MB
Render Textures: 0 - 0 B switches: 0
Screen: 1476x658 - 11.1 MB
VRAM usage: 11.1 MB to 17.2 MB (of Unknown)
VBO Total: 32 - 3.2 MB
Visible Skinned Meshes: 0 Animations: 0

Network: (no players connected)

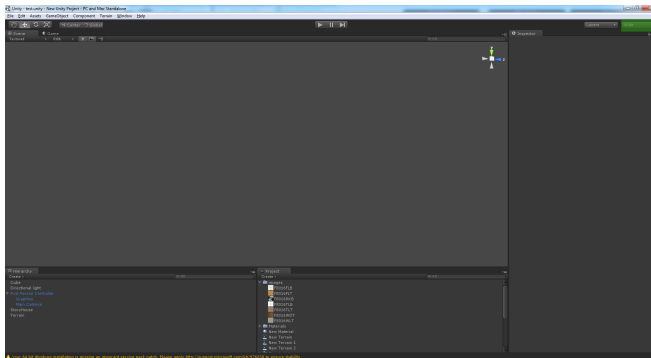
Profiler window

- Stats avancées
- Windows -> Profiler (ctrl+7)
- Permet d'étudier la répartition de la charge de calcul



Layout Manager

- On perd rapidement des fenetres, surtout en bi-ecran
- Créer et sauver son layout au début, puis le rappeler au besoin
- plusieurs layout par défaut



Les objets sous Unity

Les objets sous Unity



Importation d'objet

Importation très efficace

- Une des grandes qualités d'Unity
- Simplement glisser déposer dans la Project Window
- Maya, Cinema 4D, 3ds Max, Cheetah3D, Modo, Lightwave, Blender
- .FBX, .dae, .3DS, .dxf et .obj
- Utiliser au maximum la project window
- Peut être difficile de merger deux projets unity (copier coller repertoires...)

Importation d'objet

Attention : si asset texturé ou autres fichier externes

- Un des seuls cas où l'on évite la Project Window
- Créer directement un dossier dans le répertoire Assets
- Tout placer à l'intérieur
- Respecter la structure de fichiers de l'asset importé
- Unity met tout à jour automatiquement

Modification d'un objet

- Second cas ou l'on bypass la Project Window
- Aller directement dans le dossier de l'asset
- Ouvrir sous le logiciel de modif (3ds, photoshop)
- Modifier et sauver
- Unity met à jour automatiquement

Creation d'un objet de base

Unity permet de créer des assets de base

- Bouton "create" de la project window
- Creation répertoire
- Renommage avec F2
- Creation de scripts, shaders, animations, materials, texture de rendu, font, GUI skin.

Instancié un objet importé

- Sélectionner l'objet dans la project window
- Faire glisser dans la scène (scene window)
- Peut être placé de manière absolue grâce à l'inspector (Transform)
- Manipuler ensuite l'objet avec les commandes de la scene window

Instancier un objet de base

Unity permet de créer un certain nombre d'objets de base

- Bouton create de la hiérarchy window ou "Game Object -> Create"
- Système de particules, Camera, primitives géométriques, zone de vent, zone de reverb audio...
- Les terrains sont gérés à part dans le menu terrain.

Liens entre objets

- Un objet peut faire référence à un autre objet
- Par ex : un material pointe sur des textures
- Glisser déposer l'objet dans la variable de l'inspector.
- Ces variables peuvent être personnalisées : script

Les composants

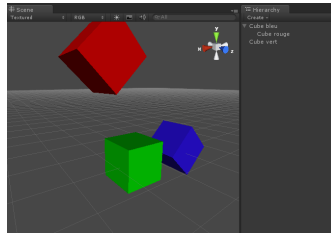
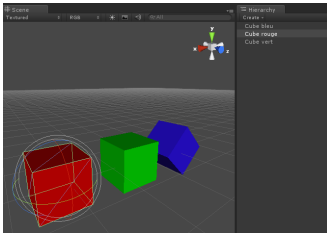
Unity s'appuie sur la notion de composants

- Un objet est une agrégation de composants
- Par ex : ajouter la physique à un objet = composant
- Par ex : un script est un composant
- Chaque composant modifie les capacités et le comportement de l'objet.
- Sélectionner un objet et ajouter le composant
- "Component -> xxx"

La notion de parents

Les objets peuvent être organisés hiérarchiquement

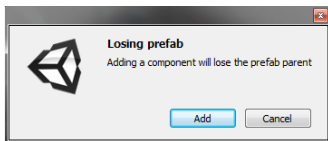
- Couramment utilisé pour changer le repère local
- Faire glisser un objet sur un autre le placer comme fils
- Les objets parents ont une flèche



Les prefabs

Les préfabs peuvent être vus comme la création d'une classe

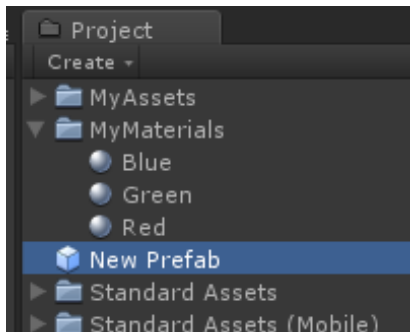
- Définit un type d'objets
- Tout objet instancié à partir d'une préfab partage ses propriétés
- Permet de modifier plusieurs objets en même temps
- Tout objet ou composant ajouté à un prefab est ajouté à ses instances
- Si on ajoute un composant ou un objet fils à un objet instancié, il perd son lien préfab



Créer prefabs

Pour créer un prefab

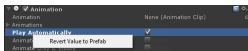
- Créer un prefab vide dans la project window
- Faire glisser un objet pour le rendre générique
- Modifier ensuite la prefab comme un objet
- (ajouter des composants, modifier des valeurs)



Gérer une instance de prefab

Une instance de prefab peut avoir ses propres données

- Toute propriété modifiée sur une instance devient spécifique ("en gras")
- Cette propriété devient indépendante de la valeur dans la prefab
- Mais impossible de modifier la structure (composant ou fils)
- Possibilité de "dé-spécifier" une propriété
- (clic droit -> "revert value to prefab")



Gérer une instance de prefab

Une instance de prefab est toujours liée a sa prefab

- Meme si lien cassé par ajout de composant
- Select : selectionne la prefab
- Revert : retour à la valeur prefab
- Apply : applique la valeur locale à la prefab
- Reconnect : recrée le lien avec la prefab (si lien cassé)

