

Formation Unity 3D

Initiation à Unity 3D

- Découverte de Unity 3D
- Prise en main de l'interface

Unity

Unity / Unity 3D / Unity for Games

Créé en 2005 par Unity Technologies

Unity 5 en 2015 puis depuis 2017 versioning
selon années (ex: 2021.1)

Moteur de jeu multiplateforme (windows, mac,
linux, web, console, mobile iOS et Android,
casque VR, ...)

Interface contrôlable à la souris et scripts

C#, Boo Script et Unity Script => C#

- Jeux vidéo
- Effets spéciaux de films
- Architecture

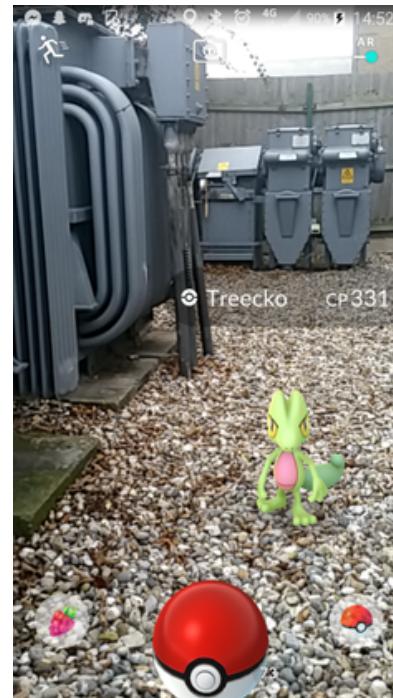
- 2D
- 3D
- AR
- VR

Made with Unity

Pokemon GO

Réalité Augmentée

Mobile



Made with Unity

Cuphead

Console/PC

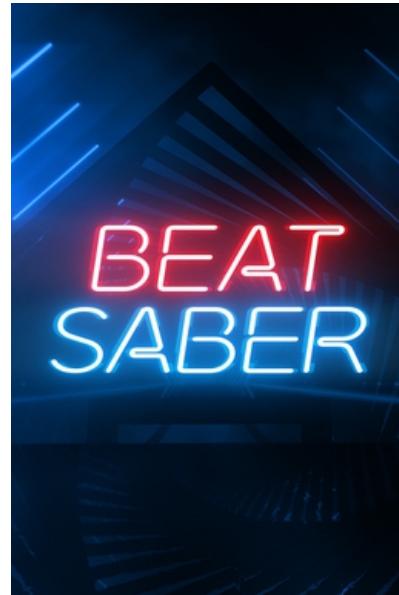


By May be found at the following website: <https://www.polygon.com/cuphead-guide/2017/10/3/16391920/captain-brineybeard-shootin-n-lootin>, Fair use, <https://en.wikipedia.org/w/index.php?curid=55845121>

Made with Unity

Beat Saber

Casque VR

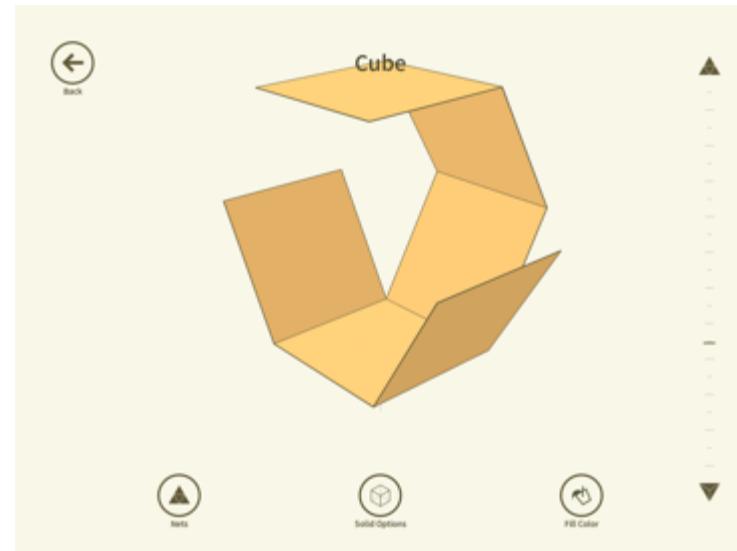


By Hyperbolic Magnetism - https://store.steampowered.com/app/620980/Beat_Saber/, Fair use, <https://en.wikipedia.org/w/index.php?curid=57376336>

Made with Unity

Shapes

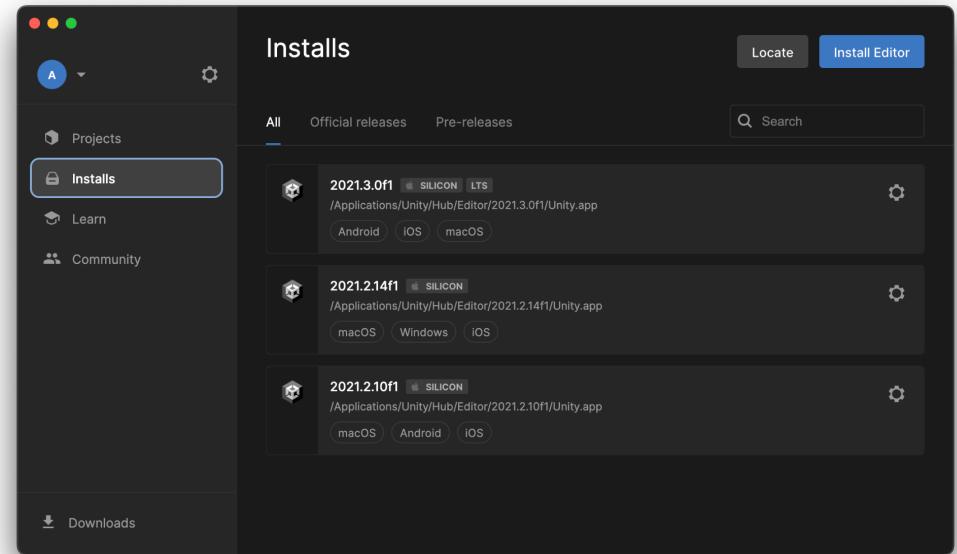
Apprentissage des formes géométriques



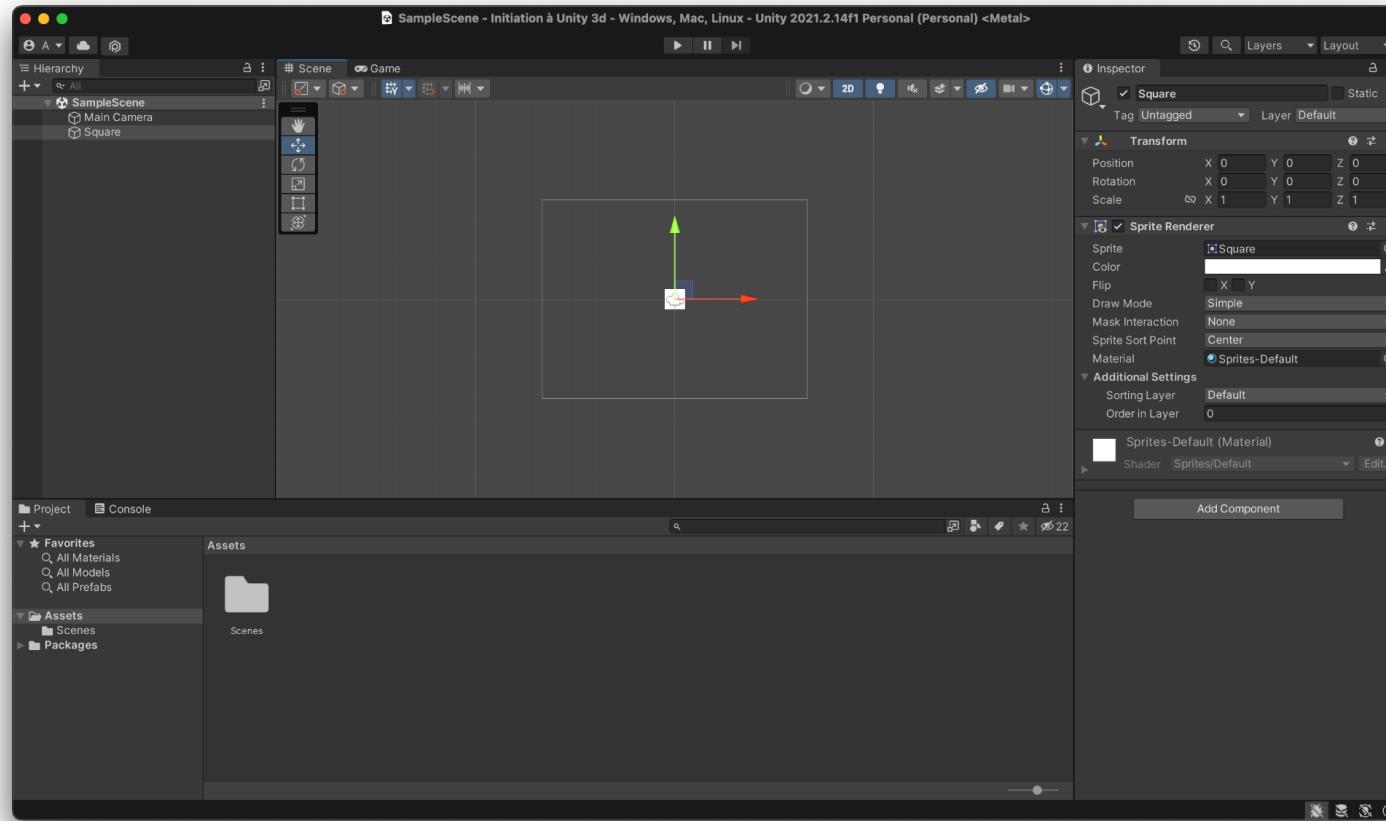
Source : <https://shapes.learnteachexplore.com/shapes-3d-geometry-learning/>

Unity Hub

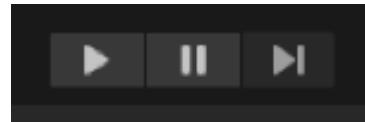
- Lanceur Unity
- Plusieurs versions de Unity avec les modules associés



Interface Unity 3D



Play mode/Edit mode



- Édition : Modification des objets sur le disque dur
- Lecture : Instanciation d'une scène et des objets associés en mémoire RAM

Découverte interface

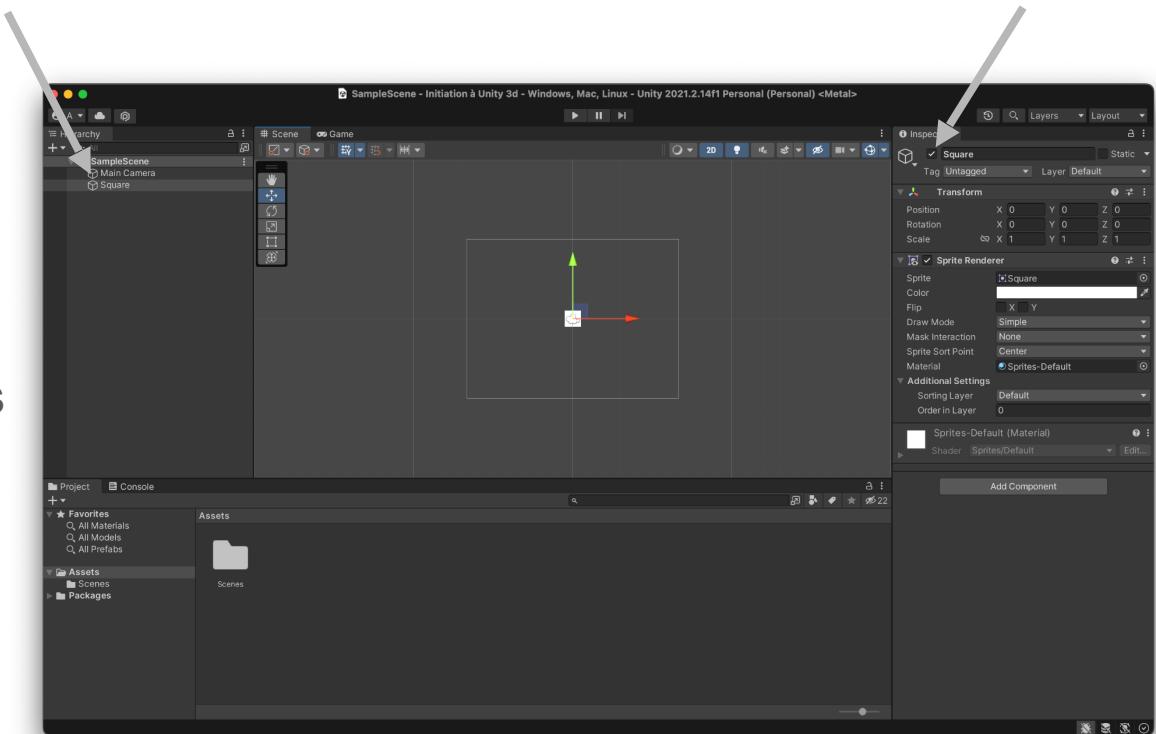
- Créer un projet 3D
- Ajouter un cube
- Manipuler la vue “éditeur”
- Manipuler la caméra pour voir le cube
- Changer taille cube
- Changer taille et déplacer objet parent

Concepts importants

GameObject

Entité de base

- Nom
- Actif ou non
- Possède des composants



Component

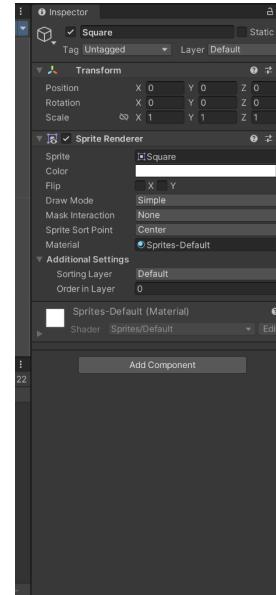
Comportement réutilisable qui peut être ajouté à un GameObject

Contient :

- des attributs
- des méthodes

Exemples :

- Transform
- Sprite Renderer
- AudioSource
- AutoRotate



Prefab

→ GameObject réutilisable

Drag and Drop dans un dossier pour créer un fichier

Modification du fichier → Modification de tous les GameObjects issus de ce prefab

Scène

- Correspond souvent à un écran d'un jeu (accueil, choix du niveau, jeu)
 - Ensemble de GameObjects
 - Fichier
 - Lecture → Instanciation de la scène et de tous les GameObjects associés
 - Modification lors de la lecture → pas de sauvegarde

Camera

GameObject avec un composant “Camera”

Position et rotation

Éléments du jeu affichés selon la caméra

Plusieurs caméras

2D/3D Orthogonal/Perspective

Exercice/Démonstration : Jeu de plateforme

1. Création du projet et des dossiers
2. Importation des images
3. Création de la plateforme (GameObject->Ajout de composant, prefab)
4. Création de la balle (Drag and drop)
5. Déplacement de la balle (Accès à un composant depuis un autre)
6. Saut
7. Étoile (collider, trigger, Destroy)
8. Balle rebondissante
9. Respawn (bas écran + étoile)
10. ScriptableObject Balle

Concepts avancés

Scriptable Object

Objet non instancié dans une scène

Global

Peut être utilisé par exemple pour :

- Partager des données entre différents GameObjects (Inventaire)
- Servir de base de données pour des objets du jeu (ex: Pokemon)

Création d'une classe héritant de ScriptableObject

Création d'une ou plusieurs instances de cette classe sous forme de fichier

Collision

Collider

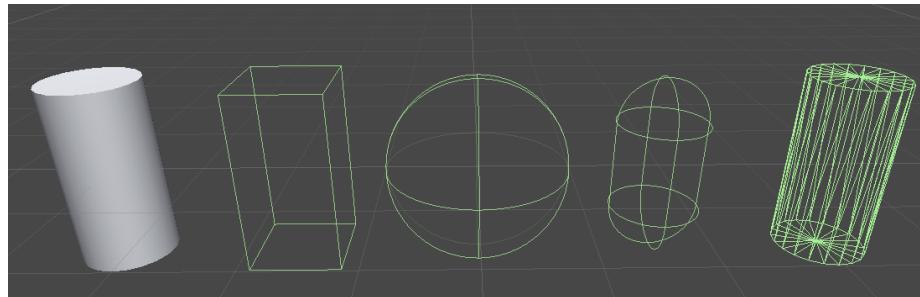
- Détermine la zone de l'objet qui peut être en collision avec autre chose

Rigidbody

- Comportement physique (gravité, frottement, force, ...)

Un rigidbody attaché minimum

Comportement automatique, méthodes appelées par Unity (OnCollisionEnter, OnCollisionExit, ...)



Physics.Raycast

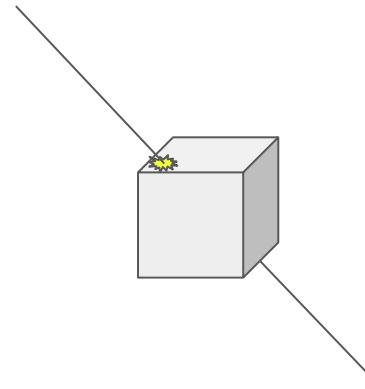
Lancé de rayons depuis un script

Depuis un point de l'espace

Vers une direction

Détecte un “collider” ou plusieurs

Retourne l'objet touché et le point de contact



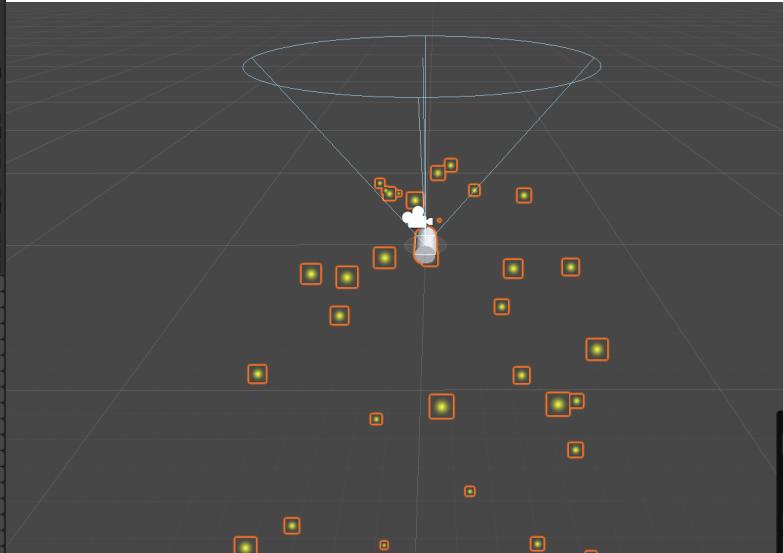
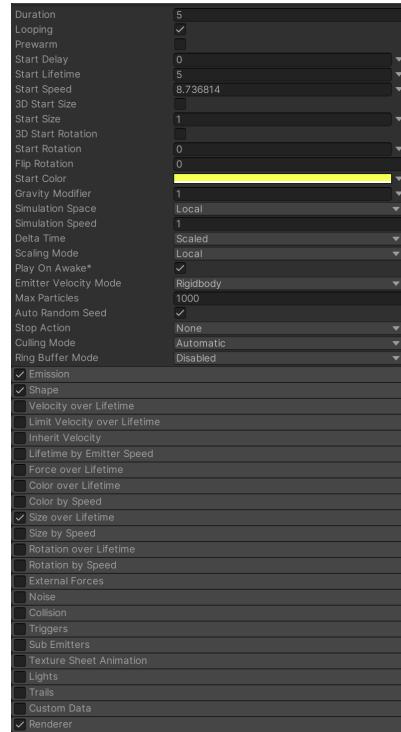
ParticleSystem

Génération de particules

Nombreux paramètres

Exemples :

- Explosion (plusieurs combinés)
- Eau qui sort d'un robinet
- Feu

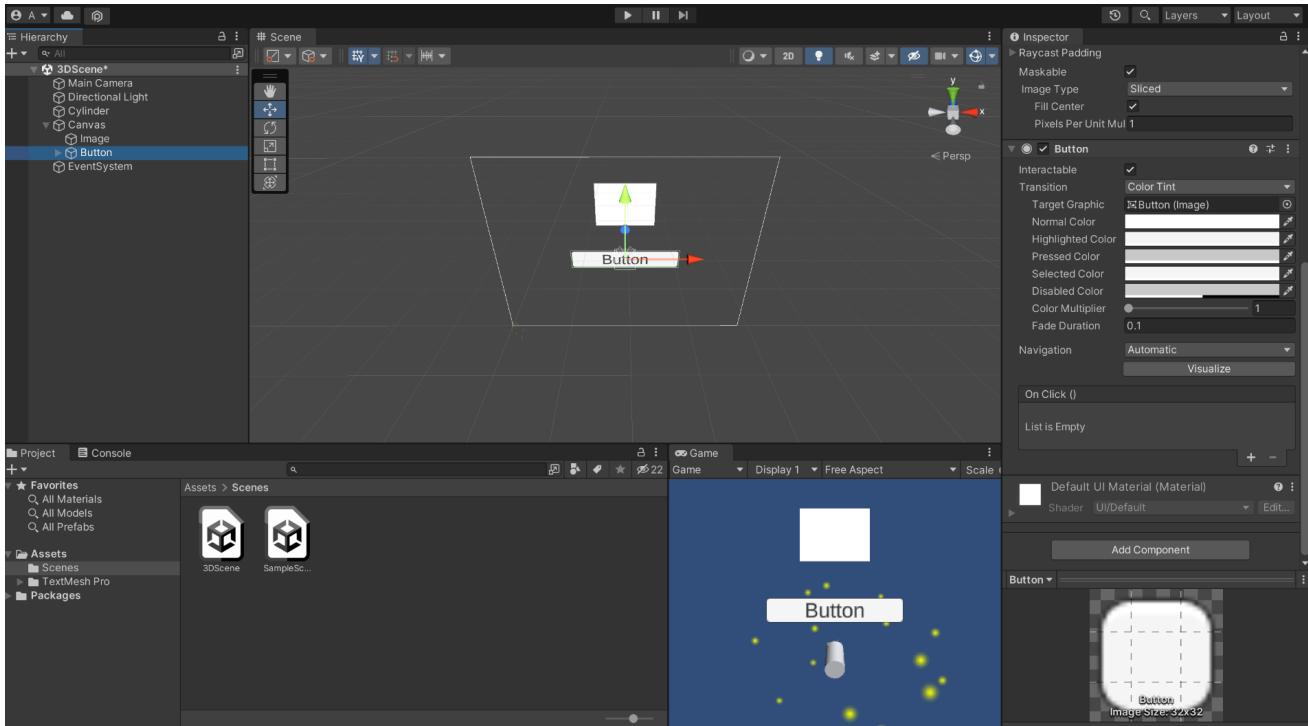


UI

Canvas

Image

Button

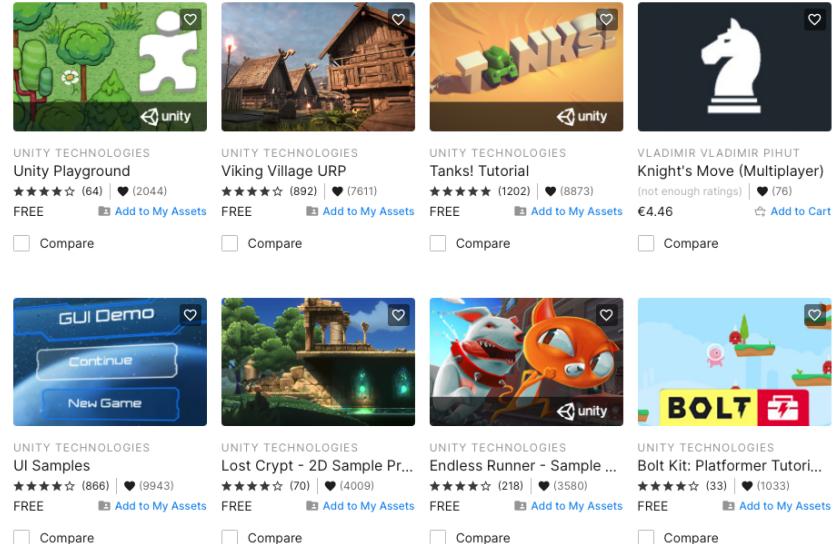


Asset store

<https://assetstore.unity.com>

Librairie en ligne

- Code
- Modèles 3D
- Illustrations
- Projets complets
- ...



Shader

Décide de la manière dont sont affichés les pixels

- Transparence
- Bruit
- Contour
- Cel/Toon Shading

Code ou Shader graph editor

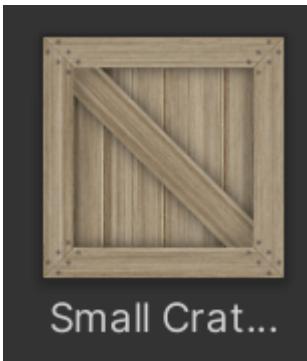


Crédit : Nintendo



Crédit : Rockstar Games

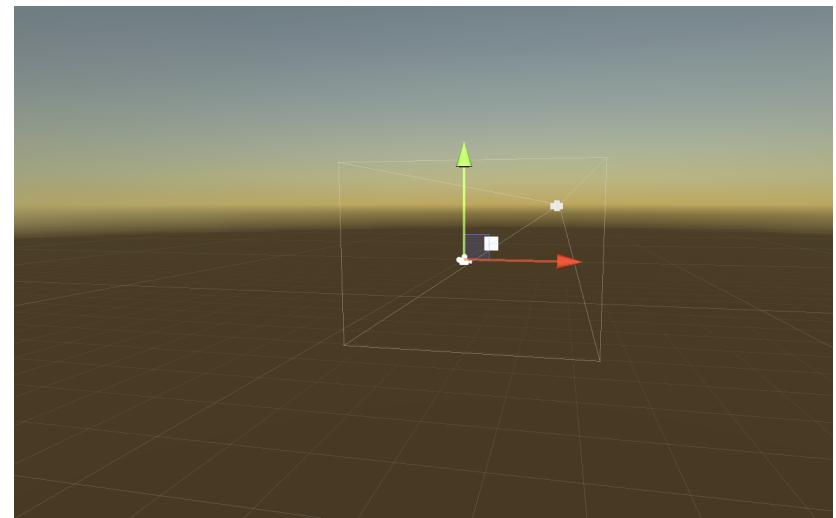
Material



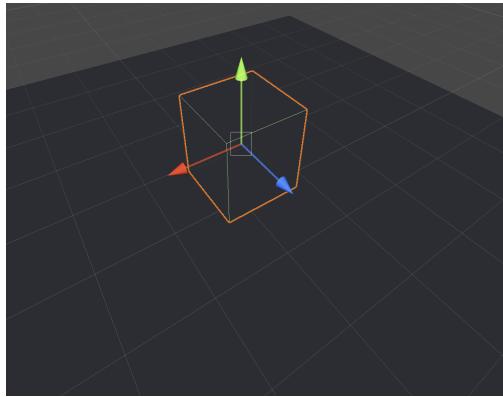
Skybox

Par défaut, couleur unie

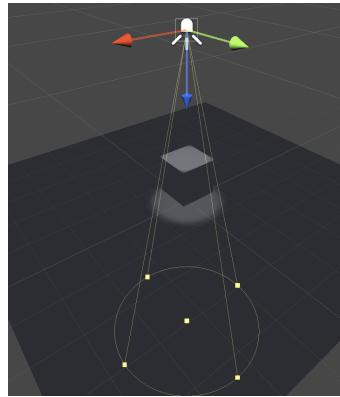
Permet d'utiliser un “Material” en fond de caméra



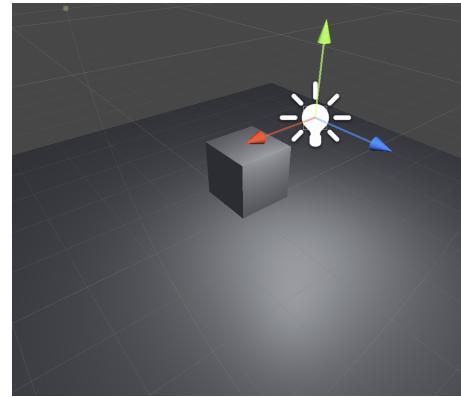
Lumière 3d



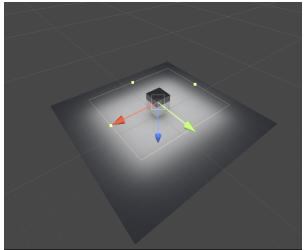
Sans lumière



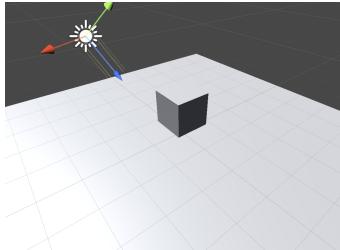
Spot light



Point Light

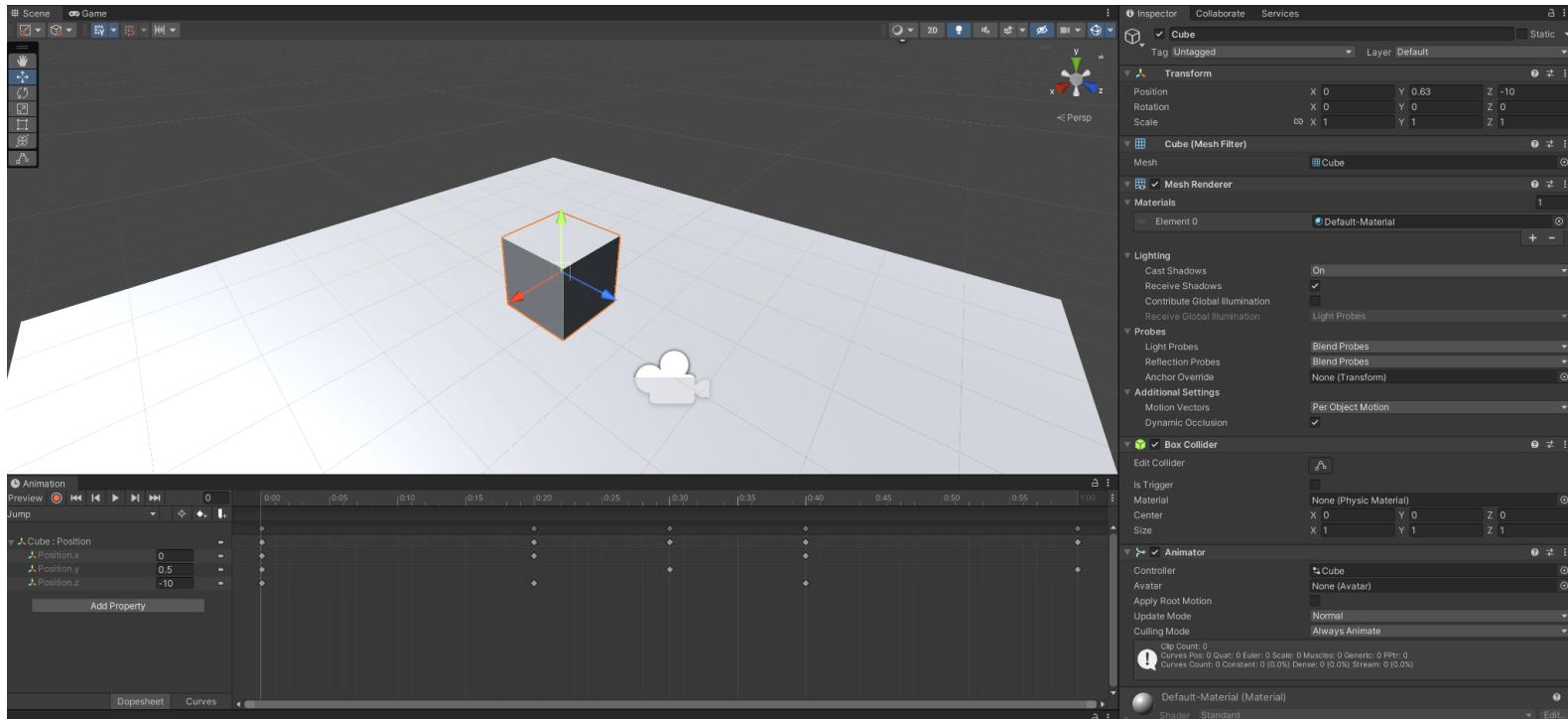


Area Light



Directional Light

Animation

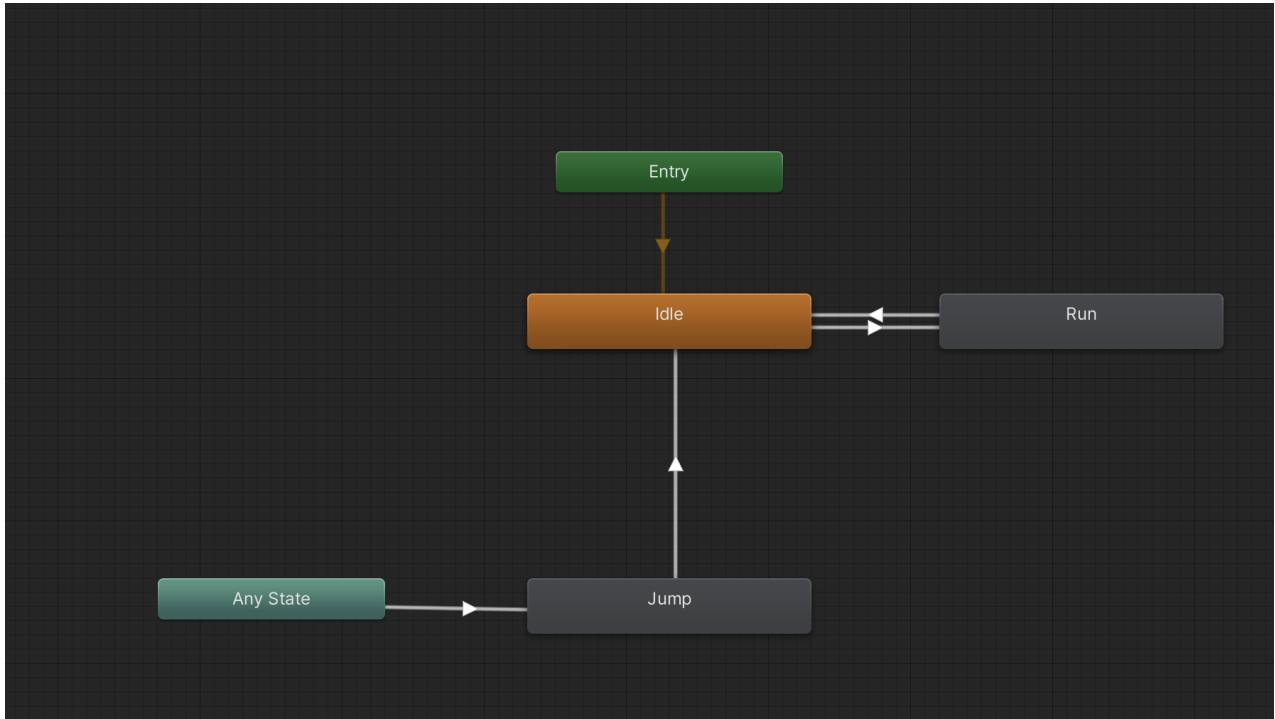


Animator

Machine à états

Transitions:

- Temps
- Trigger
- Valeur



Composant sur-mesure

Nombreux composants disponibles
(SpriteRenderer, Rigidbody, Collider, ...)

Possibilité de rajouter des nouveaux
composants :

- Sélectionner un GameObject puis **Add Component** puis **New Script**

Classe qui étend **MonoBehavior**

Plusieurs méthodes à surcharger :

Start

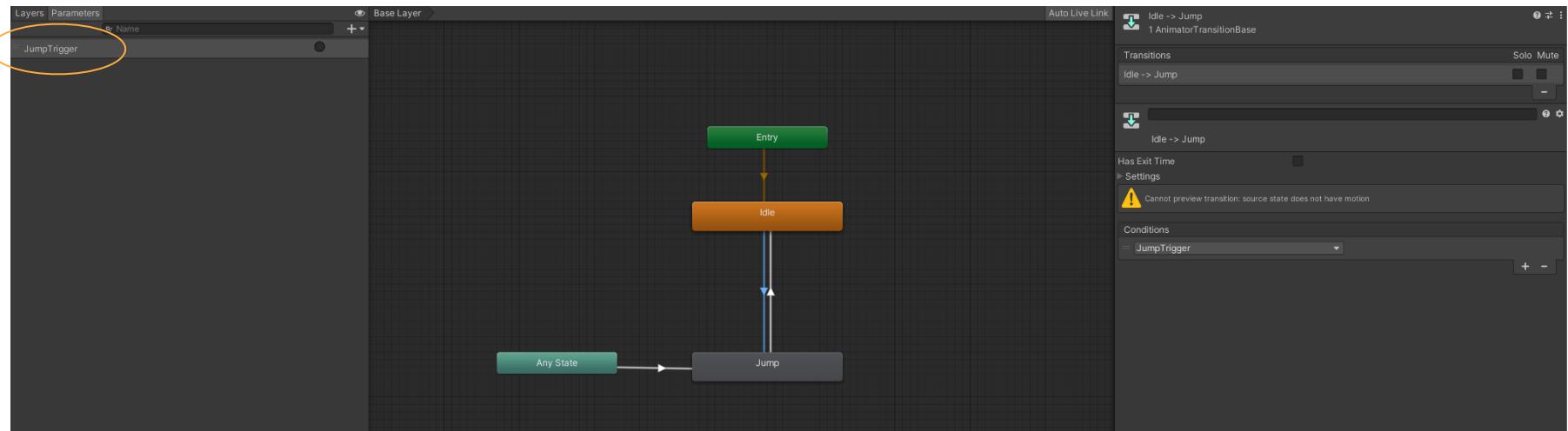
Update

OnEnable/OnDisable

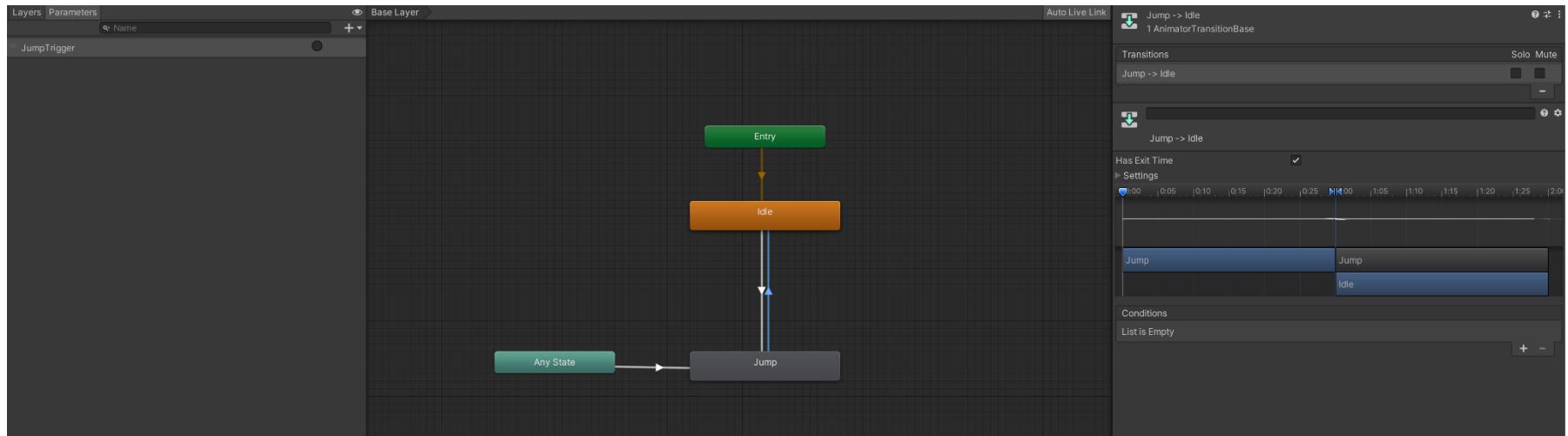
OnCollisionEnter/OnCollisionExit

OnMouseUp/OnMouseDown

Composant sur-mesure

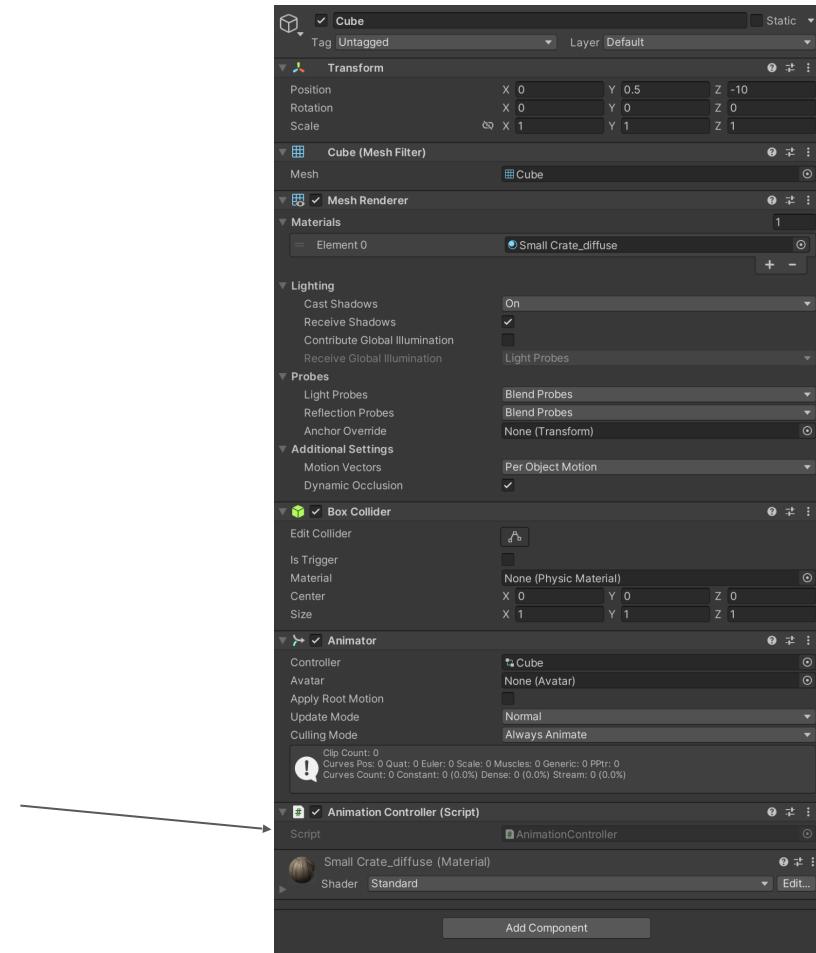


Composant sur-mesure



Composant sur-mesure

```
C# AnimationController.cs ×  
1  using System.Collections;  
2  using System.Collections.Generic;  
3  using UnityEngine;  
4  
5  public class AnimationController : MonoBehaviour  
6  {  
7      void Update()  
8      {  
9          if (Input.GetKeyDown(KeyCode.Space))  
10         {  
11             GetComponent<Animator>().SetTrigger(name: "JumpTrigger");  
12         }  
13     }  
14 }  
15
```



QCM (Plusieurs réponses possibles)

1. Si je crée un objet que je souhaite pouvoir réutiliser, je dois :

- a. Le transformer en Prefab
- b. Le transformer en Scène
- c. L'ajouter dans une Scène
- d. Écrire le code qui va le créer

2. Un GameObject :

- a. Est une classe C# instanciée
- b. Est un objet qui a une position, une taille et une rotation sur lequel on peut rajouter des composants
- c. Est forcément affichée par une caméra
- d. Ne peut pas contenir de sous GameObjects

3. Une scène :

- a. Est un GameObject avec un composant Scène
- b. Est une collection de GameObject positionnés
- c. Contient l'ensemble du jeu
- d. Est 2d ou 3d mais pas les deux

4. Un composant :

- a. Est représenté par une classe qui étend MonoBehavior
- b. Possède forcément une position, une taille et une rotation
- c. Est composé de sous composants

QCM (Plusieurs réponses possibles)

5. **Si je veux créer un cube animé, je dois :**
 - a. Ajouter un composant Animation à mon cube
 - b. Ajouter un composant Animator à mon cube
 - c. Scripter l'animation dans un composant
6. **Alors que le jeu ne tourne pas, un cube est placé à une position A. Je lance le jeu et je déplace à la souris le cube vers une position B depuis la vue Edition. J'arrête le jeu. À quelle position sera le cube ?**
 - a. Position A
 - b. Position B
 - c. Position A mais reviendra à B dès que je cliquerai à nouveau sur Play
7. **Comment ajouter une Caméra dans une scène ?**
 - a. Créer un GameObject et ajouter un composant Camera
 - b. La caméra n'est pas un GameObject et n'a pas besoin d'être ajoutée
8. **Comment détecter une collision sur un objet ?**
 - a. Créer un script qui va vérifier la position de tous les objets pour vérifier s'ils sont en collision
 - b. Implémenter la méthode OnCollisionEnter sur le composant qui subit la collision (ex : cible de tir à l'arc)
 - c. Implémenter la méthode OnCollisionEnter sur le composant qui inflige la collision (ex : flèche)
 - d. Implémenter la méthode OnCollisionEnter sur le composant qui inflige ou celui qui subit la collision selon ce que l'on souhaite faire

QCM (Plusieurs réponses possibles)

9. Qu'est ce qu'un Shader ?

- a. Un système d'affichage d'ombres
- b. Une skybox
- c. Script qui décide la manière dont sont affichés les pixels

10. Qu'est ce que je peux trouver gratuitement sur Unity Asset Store ?

- a. Le code source de jeux
- b. Des modèles 3D et images 2D
- c. Des sons
- d. Des composants

11. Je crée un Prefab et je l'ajoute à deux scènes :

- a. Si je modifie une valeur d'un de ses composants dans une scène, le changement sera propagé automatiquement à l'autre scène
- b. Si je modifie une valeur d'un de ses composants dans une scène, le changement sera propagé automatiquement au fichier du Prefab
- c. Si je modifie une valeur d'une de ses composants dans le fichier du prefab, le changement sera propagé automatiquement aux deux scènes

12. Qu'est ce qui est représenté par un fichier sur le disque dur ?

- a. Les scènes
- b. Les Prefabs
- c. Les GameObjects
- d. Les composants

13. Si je veux vérifier que mon personnage touche le sol, je dois :

- a. Mettre un objet avec un collider sous ces pieds et vérifier s'il est en collision avec le sol ou non
- b. Faire un lancé de rayon physique vers le sol
- c. Comparer la position de mon personnage comparé à celui du sol
- d. Implémenter la méthode OnCollisionEnter

QCM (Plusieurs réponses possibles)

14. Un jeu créé sur Unity peut être compilé pour

- a. Mac/Windows/Linux
- b. Certaines consoles
- c. Android/iPhone
- d. Navigateur Web

15. Un GameObject

- a. A un nom
- b. Peut contenir d'autres GameObjects
- c. Peut être actif ou inactif
- d. Peut ne pas avoir de position

16. La méthode Update d'un composant :

- a. Est appelé à chaque rafraîchissement de l'écran
- b. Est appelé à interval régulier
- c. Peut contenir des traitements longs

17. Comment accéder au composant ComposantA d'un GameObject depuis le ComposantB sur ce même GameObject ?

- a. this.gameObject.GetComponent<ComposantA>()
- b. this.composantA
- c. GetComponent<ComposantA>()
- d. this.GetComponent<ComposantA>()

Architecture d'un projet

Classement par fonctionnalité

- **Assets/**
 - **_Project/**
 - Player/
 - **player.png**
 - **PlayerController.cs**
 - Map/
 - **map.png**
 - **MapRenderer.cs**
 - **MapMaterial.mat**
 - Plugin 1/
 - Plugin 2/

Classement par type

- **Assets/**
 - **_Project/**
 - Images/
 - **player.png**
 - **map.png**
 - Scripts/
 - **PlayerController.cs**
 - **MapRenderer.cs**
 - Material/
 - **MapMaterial.mat**
 - Plugin 1/
 - Plugin 2/

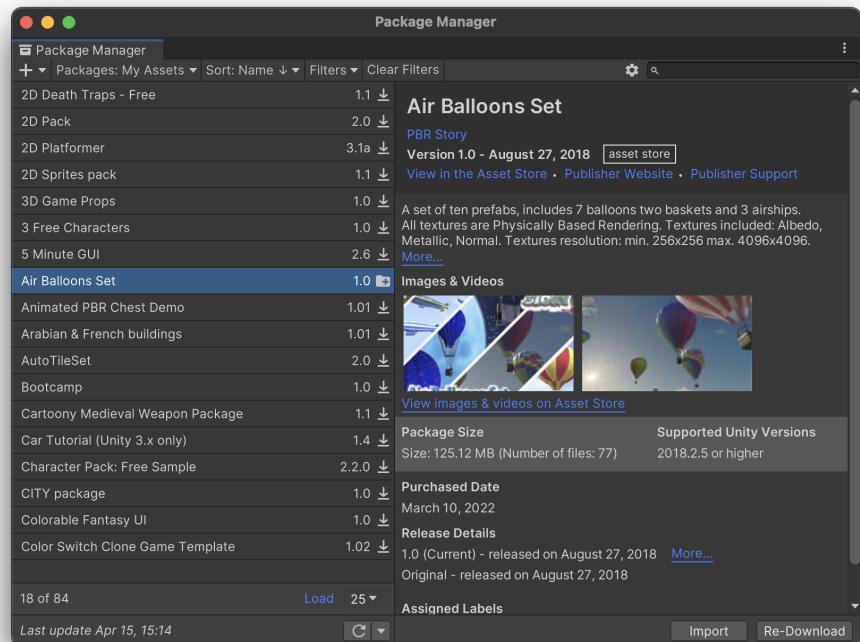
Plugins additionnels

À placer à la racine du projet (Dossier Assets)

Ajout manuel

ou

Ajout via Asset Store (Asset store -> Package manager -> Importation)



Git

Scène, Prefab, ScriptableObject → Problème merge

Component → OK

Ticket pour autorisation de modifier un élément

Exercice : Jeu Pong 3d

1. Création de la balle
2. Ajout des raquettes
3. Comptabilisation et affichage des points
 - a. ScriptableObject avec event pour affichage de score
4. Ajout IA basique

Réglages :

- Edit→Project Settings→ Physics →
Bounce threshold : 0
- Physics Material : Friction 0 Minimum,
Bounce : 1 maximum

