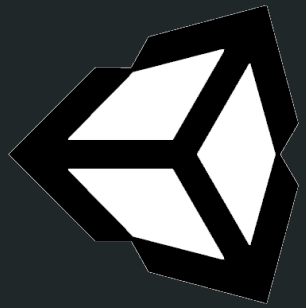
Introduction à Unity

La physique



Comment on fait ça ?

Ici qu’est ce qu’on voit ?

1. La gravité (Rigidbody)
2. La pomme rentre en contact avec un objet et ne le traverse pas (Collider)
3. La force réciproque qui provoque le rebond de la pomme (physical Materials)

La gravité : le rigidbody

- Prenez un objet (un cube) et placez le dans la scène comme ça : il ne tombe pas

- L’élément principal de la physique des objets en mouvement est le rigidbody

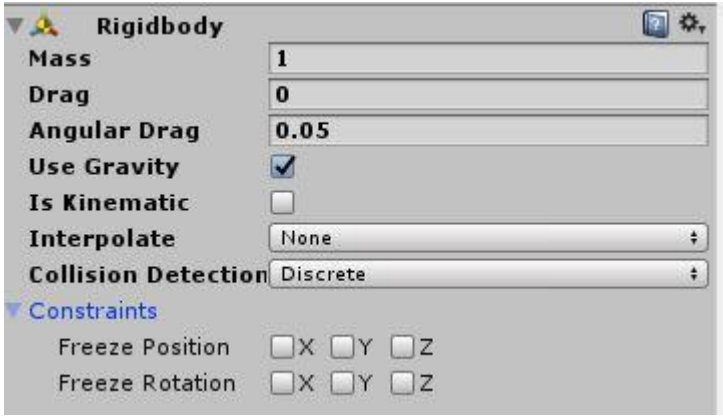
- Ajouter le rigidbody via AddComponent sur votre objet

- Relancer la simulation : l’objet tombe cette fois !

Le RigidBody

- OK, c’est bien mais concrètement ?

- Le composant rigidbody va enregistré en tant qu’objet physique l’objet (cf. Cours 2 update physique)



- Mass : définie le poids de l’objet
- Drag et Angular drag : le coefficient de frottement
- IsKinematic : l’objet doit-il être influencé par la physique
- Interpolate : permet de changer le mode de pré-calcul des prochains changements physiques

Pourquoi IsKinematic ?

On pourrait se demander pourquoi mettre un rigidbody si on coche ensuite isKinematic ? (on enregistre l’objet pour la physique mais on fait en sorte que ce dernier ne soit pas influencé par la physique)

Les colliders : intro

- Prenez un cube mettez lui un rigidbody
- (décochez le collider mit par défaut)
- Placez un plan en dessous
- Lancez la simulation

- Le Cube traverse le plan !

- Recochez le collider, relancez : le cube reste sur le plan !

Les colliders : Qu’est ce que c’est ?

- Ils définissent la forme et donc le type de collisions que vont subir les objets

- C’est la zone (“Hitbox”) de contacte avec l’objet

- Ils peuvent être bloquant ou déclencheur : IsTrigger
  - Attention un collider IsTrigger ne prendra plus en compte les collisions

- dans les scripts :
  - void OnCollisionEnter(Collision Col) : détecte deux collider qui se touche
  - void OnTriggerEnter(Collider other) : détecte quand un collider entre dans un autre collider IsTrigger

Les colliders : les performances

Du plus performant au moins performant :

1. Sphères
2. Capsules
3. Box
4. Mesh : très précis donc beaucoup plus difficiles à calculer

https://forum.unity3d.com/threads/capsule-vs-box-colliders.34254/

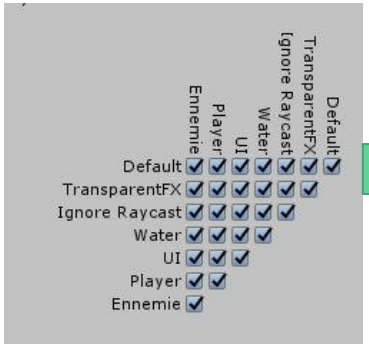
Les layers

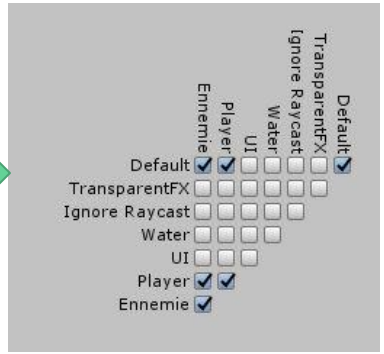
- Les layers sont des octets qui définissent l’appartenance des colliders à un groupe

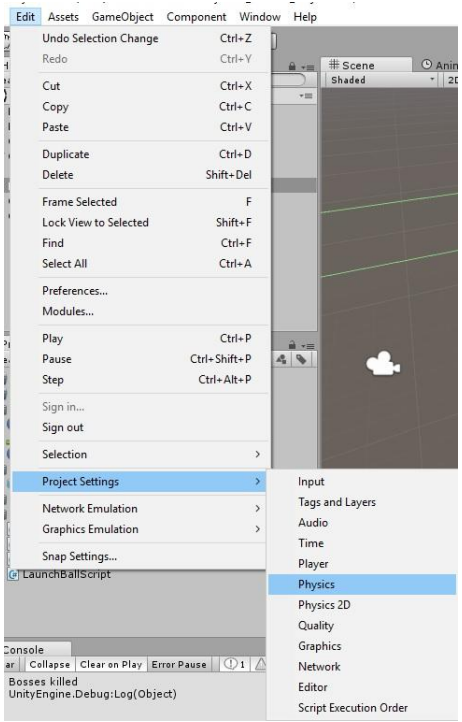
- Ils permettent de catégoriser les interactions entre colliders
    - Par exemple : vous ne pouvez pas attaquer vos alliés mais les ennemies peuvent les attaquer

Matrice Physique

- Edit ➔ Project Settings ➔ Physics
- Ouvre un panneau ou l'on peut modifier la gravité
- Mais aussi les types de layers et leur interaction
- Faire une matrice custom permet de gagner énormément en performance et de maîtriser les interactions !







Matrice complète

Matrice Custom

Les tags

- A n’utiliser qu’en cas de vrai nécessiter

- Les tags sont des chaînes de caractères donc coûteuse en temps de calcule

- Ils permettent d’ajouter un niveau de différenciation au layer
  - Par exemple :
    - Layer : Ennemie & Tag : Boss
    - Layer : Ennemie & Tag : Sbire

Interaction Souris - Monde

- Comment transformer des coordonnées 2D (écran) en coordonnées 3D ?

- La fonction a utiliser est : MaCamera.ScreenPointToWorld(position de ma souris)
  - Mais attention :
    - la position de la souris se trouve en faisant : Input.mousePosition
    - Seulement cette position est exprimé en 2D (même si le vecteur retourné est un Vector3
    - Il faut donc lui donner un Z approximatif (souvent camera.forward + 1)

- Optionnel : pour aller plus loin il existe plusieurs matrice de transformation qui servent à définir la transformation espace 2D ➜ 3D (OpenGL)
  - MatriceView qui définit l'espace 2D de l'écran
  - MatriceTransform qui définit la matrice de passage 2D ➜ 3D
  - MatriceObjet (ou monde) qui définit la position dans l'espace 3D

Raycast : lancé de rayon

- Le raycast (ou lancé de rayon) sert à tester si un objet se trouve sur la trajectoire du rayon

- Attention cette opération est EXTRÊMEMENT coûteuse ! A ne pas faire dans un update.

- Elle va servir notamment à savoir si le joueur a cliqué sur un objet ou non.

- La fonction utilisée est : Physics.Raycast()

- https://docs.unity3d.com/ScriptReference/Physics.Raycast.html

Raycast : et ses copains

- le raycast lance un rayon mais bien évidemment il existe d'autre type :
  - CapsuleCast : qui envoi à une distance donnée un rayon en forme de capsule (pour l'image on pourrait dire que c'est un collider IsTrigger mais ce n'est pas tout à fait ça)
  - BoxCast : envoie un cube
  - SphereCast : qui envoie une sphère à un endroit donnée

- Attention : il est possible de spécifier les layers que doit toucher votre raycast pour éviter de toucher un mur avant le joueur ou autre.

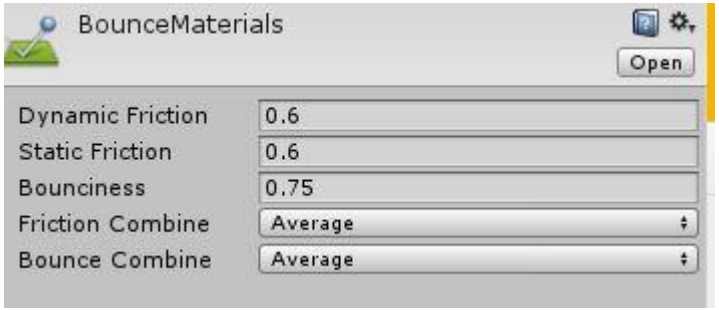
Physical Materials

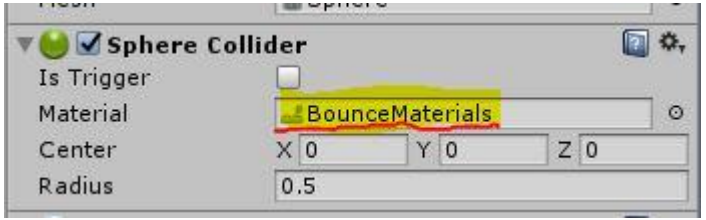
- Se créer dans les dossiers ➜ clic droit ➜ Create ➜ Physics Materials

- Permet facilement de définir les données physique comme le coefficient de frottement, l'élasticité etc

- Se met dans Material du collider

- Cf. projet de cours avec les balles qui rebondissent





Best Practices

https://unity3d.com/learn/tutorials/topics/physics/physics-best-practices