
Compte Rendu du TP 2

Réalité Virtuelle / Réalité Augmentée

Jérémy DAFFIX

Le code source et les fichiers du projet sont disponibles sur GitHub :

<https://github.com/jeremydaffix/TP1-VR>

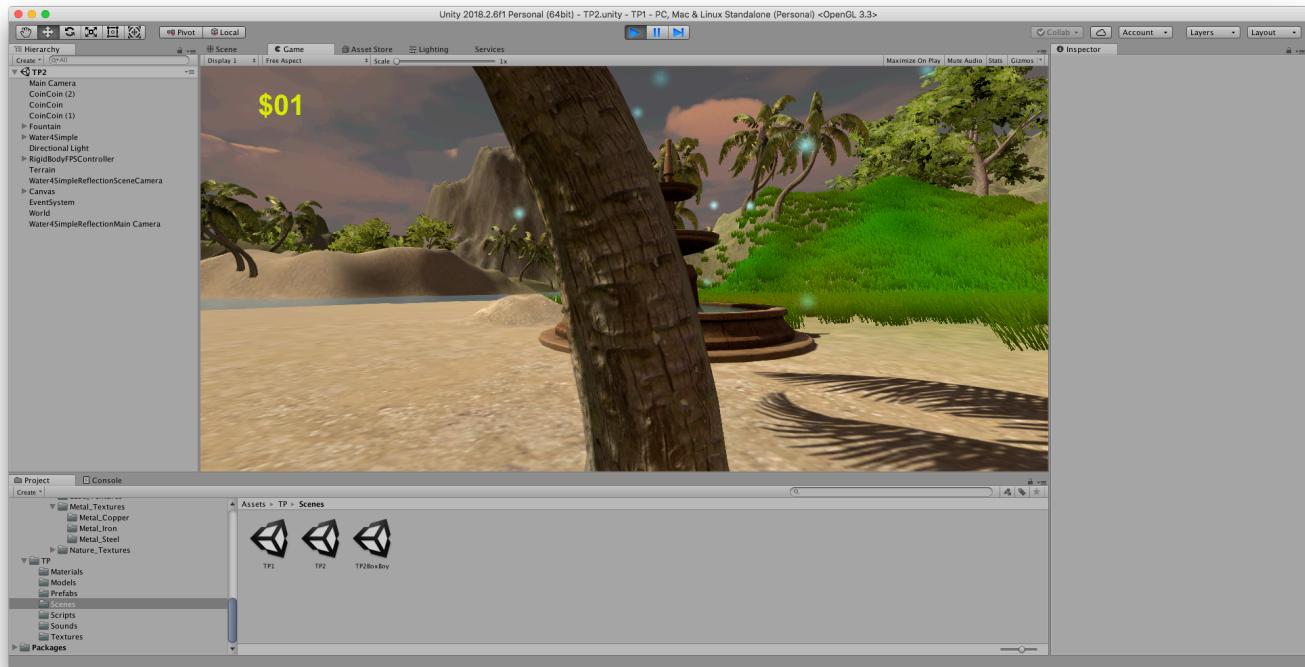
Assets utilisés :

Ressources fournies sur le Moodle du module

Textures supplémentaires :

<https://assetstore.unity.com/packages/2d/textures-materials/abstract/early-prototyping-material-kit-51761>

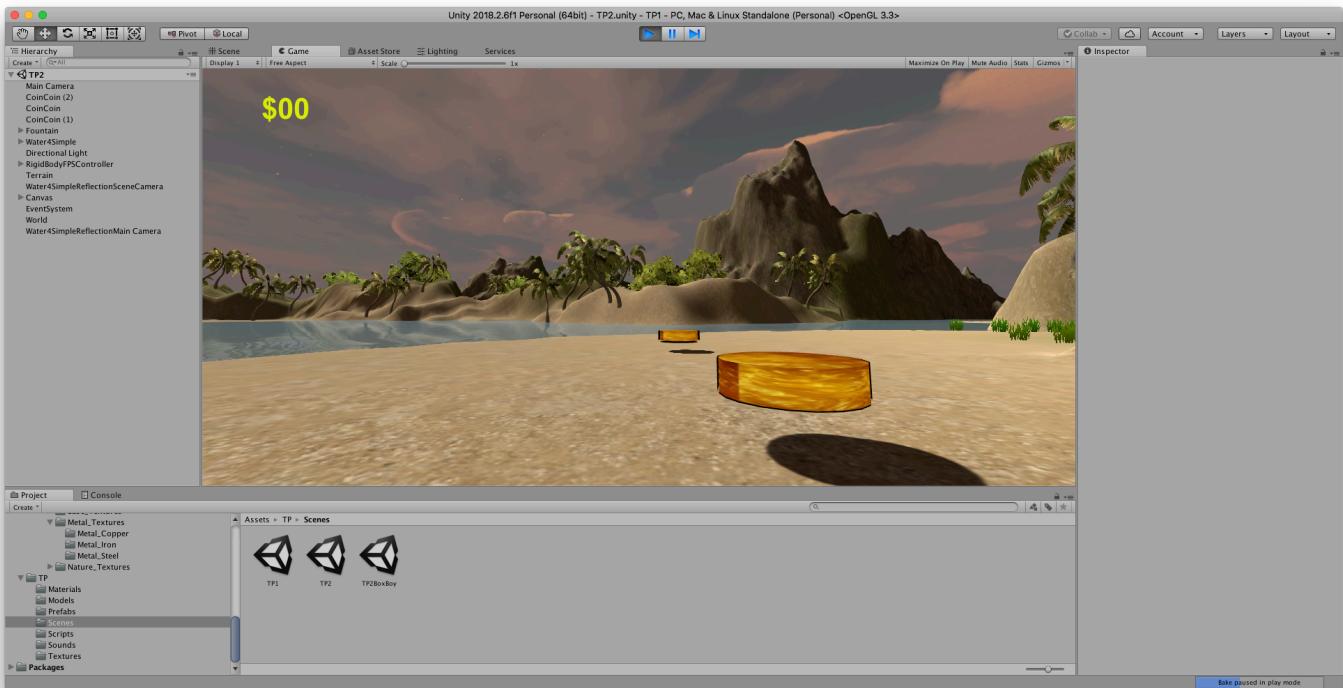
Captures d'écran et avancement



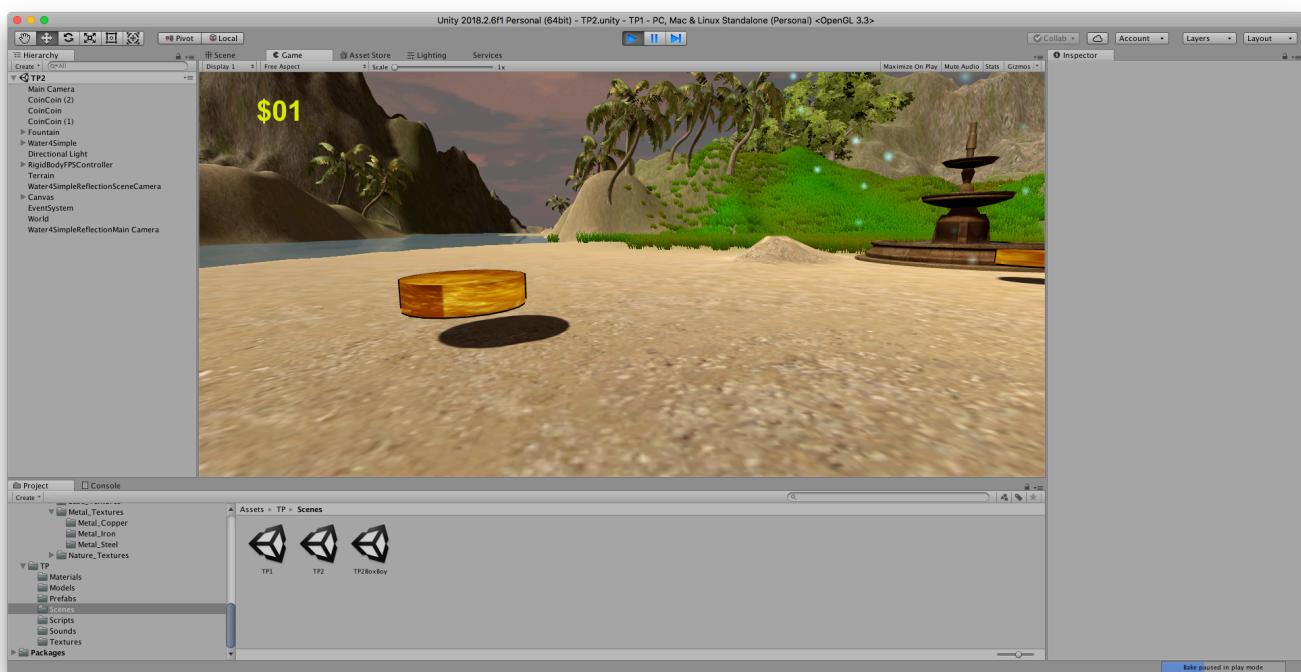
Les arbres empêchent le RigidBodyFPSController de passer.

Note :

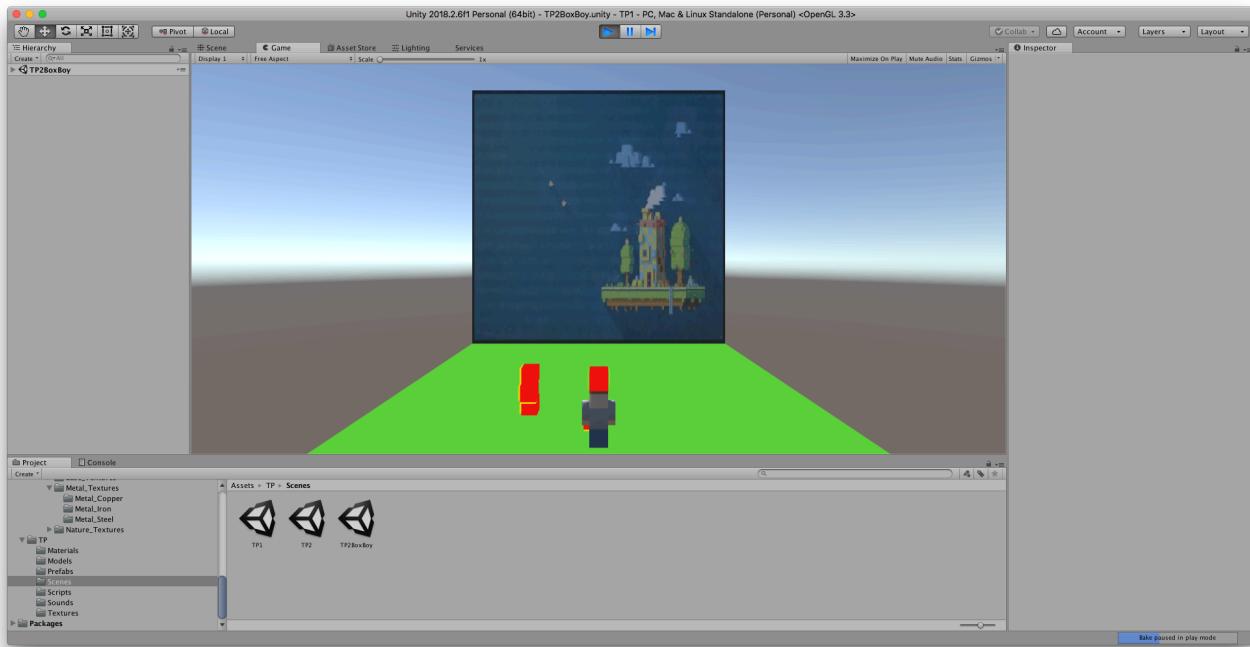
Le système de trees de Unity fournit déjà un système de colliders, pas besoin de rajouter un capsulecollider supplémentaire sur les arbres.



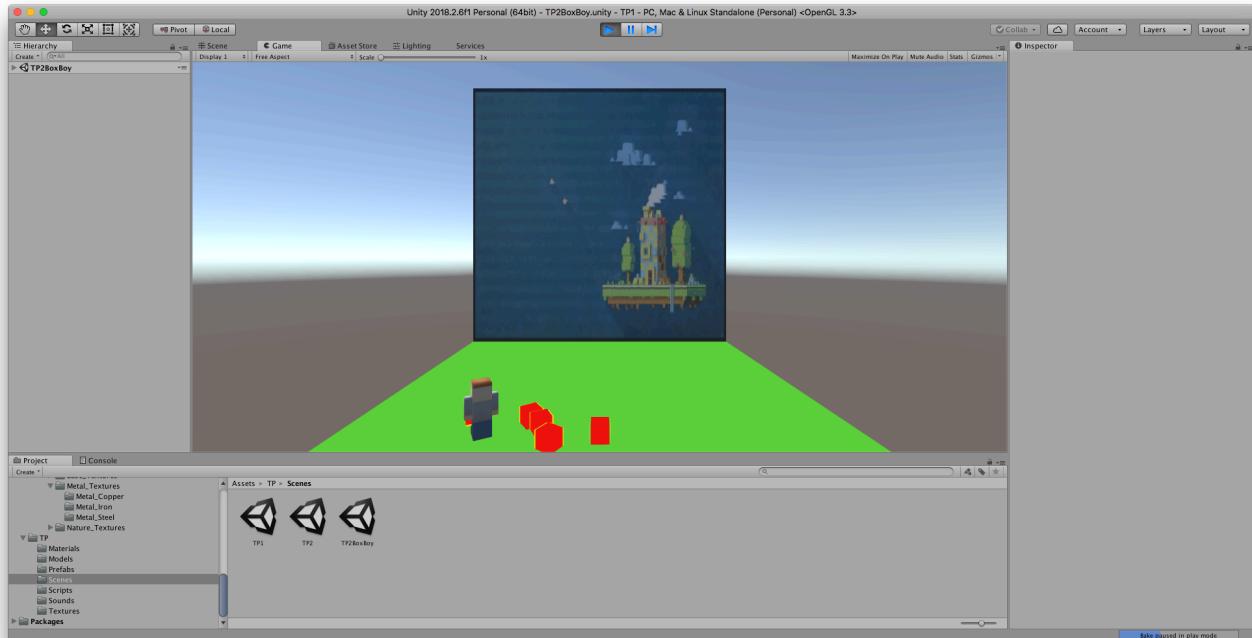
Les pièces, avec un shader « Toon / Basic Outline » et une texture Or.



Le texte se met à jour et un son est joué lorsque le joueur ramasse une pièce.



La scène du BoxBoy. Un plan « ground », un plan « wall » avec une texture, 4 cubes (dont un sur sa tête) soumis aux lois de la physique.



BoxBoy peut se déplacer et casser la pile de cubes. Vilain BoxBoy.

Architecture du projet

Les Assets "extérieurs" viennent s'installer automatiquement dans *Assets/*.

Pour éviter que tout soit mélangé, créer un répertoire spécialement pour les fichiers du projet en lui-même (ici, *Assets/TP*).

Assets/TP/Scenes -> les scènes

Assets/TP/Scripts -> le code C#

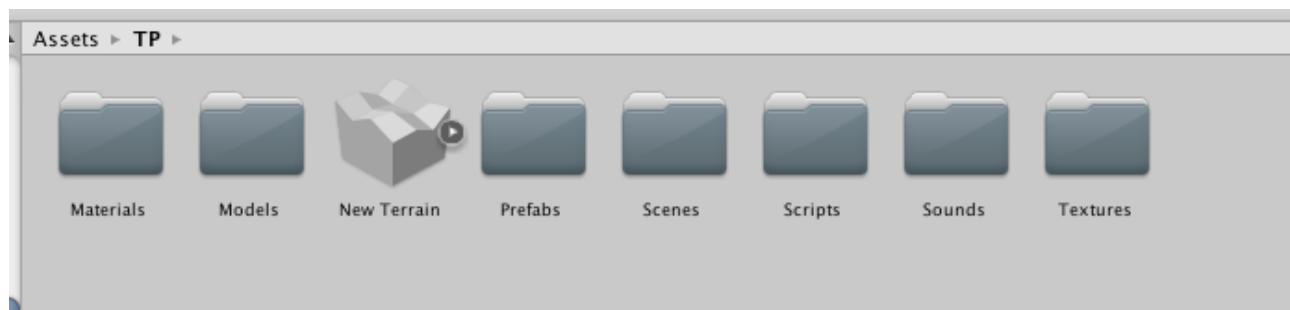
Assets/TP/Materials -> les matériaux

Assets/TP/Textures -> les images de textures

Assets/TP/Sounds -> les sons

Assets/TP/Prefabs -> les prefabs

Assets/TP/Models -> les modèles 3D



L'architecture du projet.

Notes diverses

Seul un modèle 3D .fbx est fourni pour BoxBoy. Il faut donc créer un Prefab utilisant ce modèle dans le dossier *Prefabs/* .

Utilisation du shader « Toon / Basic Outline » pour les pièces et les cubes, ce qui donne un effet graphique particulier.

Plutôt que d'utiliser des *GameObject.Find("nomObjet")* pour trouver les différents éléments GameObject dont on a besoin, on peut mettre des propriétés publiques et glisser dessus l'objet voulu à partir de l'éditeur.

Eviter de modifier directement les fichiers des assets téléchargés, plutôt effectuer une copie des fichiers modifiés : cela permettra éventuellement de les mettre à jour plus tard à partir de l'Asset Store sans écraser nos modifications.

Ajout d'un rigidbody et d'un collider à BoxBoy et aux plans, en plus des cubes. Les plans sont marqués comme "kinematic" pour qu'ils ne bougent pas.

Afin que tout soit correctement géré par le moteur physique (cela peut amener des bugs), il vaut mieux utiliser la méthode *MovePosition()* du RigidBody à la place de *transform.Translate()* :

```
rb.MovePosition(rb.transform.position + new Vector3(x, 0, -y));
```