

Rapport du jeu et des modèles à réaliser sous Unity et Blender.

Auteur : Maxime THIEL

Date : début janvier 2025

Le décor

J'avais initialement pour but de réaliser la carte en utilisant des pavés : un pour le sol, plusieurs pour l'eau, les plateformes. J'ai rapidement abandonné cette idée, je ne la trouvais pas satisfaisante pour avoir une belle map avec du relief.

J'ai créé quatre terrains. J'ai pu mettre du relief avec l'outil Raise or Lower Terrain, et peindre différentes textures avec Paint Texture selon l'endroit : de l'herbe en général, mais aussi de la terre pour le chemin, de la roche pour les montagnes et de la neige pour les sommets. J'ai ajusté les tilings pour avoir des répétitions de textures optimales (ni trop grandes, ni trop petites).

J'ai creusé un grand trou avec Paint Holes pour y mettre l'eau. En partant d'un cube, j'ai fait un pavé auquel j'ai appliqué une texture d'eau et mis de la transparence. Un autre pavé plus petit existe, cette fois-ci plus haut en Y, dans une partie creusée de la montagne, de façon à piéger le joueur qui s'y aventurerait. Un rondin avec une texture de bois est posé pour traverser la grande mare.

Les arbres n'étant pas évidents à réaliser sous forme de modèle 3D, j'ai pour cela importé un asset de l'Unity Asset Store : **Tree Collection Pack 2017**. J'en ai dispersé quelques uns pour décorer.

Pour donner un peu plus de vie au terrain, j'ai appliqué de l'herbe qui bouge avec le vent, avec l'outil Paint Details (à partir d'une image transparente d'herbe).

Les scripts

Objets clonés et objets uniques

- `CoinsController.cs` : Permet de générer toutes les pièces (il y en a cinq-cents), positionnées de manière aléatoire en X et Z, tout en recherchant le Y du

terrain (étant donné qu'il y a du relief). Toutes les pièces tournent en même temps, avec application de la gravité. On compte les pièces collectées. Pour ce script de clonage comme pour ceux qui suivent, on attache un script de comportement individuel à chaque clone (Coin.cs, AKey.cs, AChest.cs, MonsterBehaviour.cs).

- **Coin.cs** : Gère la collision de la pièce avec le tag Body (player), et joue le son correspondant une seule fois, avant d'être désactivée.
- **KeysAndChestsCloning.cs** : Crée trente clés, coffres et marqueurs. Les clés (avec couleurs différentes) et coffres sont placés à des positions aléatoires (comme CoinsController.cs) et sont renseignés dans des listes. Les marqueurs se placent au dessus des coffres et sont désactivés. J'ai également mis une fonction pour ajouter à une troisième liste (obtainedKeys) l'indice d'une clé obtenue, ce qui pourra servir pour l'activation des marqueurs.
- **AKey.cs** : Fonctionne comme pour Coin.cs.
- **AChest.cs** : Fonctionne comme pour Coin.cs. Détecte la présence ou non du joueur dans sa sphère de collision. Si obtainedKeys contient l'identifiant du coffre, il y a activation du marqueur vert pour avertir le joueur au loin. On pourra alors ouvrir ou fermer le coffre (rotation du couvercle).
- **MonsterSpawner.cs** : Même principe que pour les précédents cloneurs.
- **MonsterBehaviour.cs** : J'ai créé des IEnumerator pour le mouvement autonome (MoveMonster()) et la rotation lors des coups que le monstre donne (RotatePattern()). Dans Update(), plusieurs conditions sont vérifiées pour faire perdre de la vie au monstre (joueur a une épée, est à proximité et presse F).
- **Sword.cs** : Place l'épée à la main droite du joueur quand elle celle-ci est touchée. On peut la faire bouger avec la touche F.
- **Trophy.cs** : Trophée qui tourne sur lui-même et change le gameState à "victory" quand il est touché par le joueur.

Gestion de l'UI

- **UIManager.cs** : Active ou désactive les différents éléments selon gameState : "menu", "inGame", "dead" ou "victory".
- **ButtonHover.cs** : Permet aux sprites 2D Play, Parameters, Quit et Exit to menu de s'agrandir au survol, et quand ils sont cliqués de changer le gameState

("menu" ou "inGame").

- `CounterController.cs` : Modification des TextMeshPro des nombres de pièces collectées et de monstres tués.
- `Hearts.cs` : Gestion des cœurs avec liste d'images, changement de l'image source et modification de la largeur.

Mousy et la caméra orbitale

- Scripts donnés dans le dossier Mousy (+ script généré pour l'Input Asset), avec modifications sur plusieurs.
- `PlayerStepsParticles.cs` : Émission de particules de fumée (avec position et rotation du joueur). Émet quand le joueur marche, cesse quand il est statique.

Autres

- `PortalTeleport.cs` : Permet la téléportation entre deux portails. Chacun à deux champs pour lier les deux objets.
- `WaterCollision.cs` : Quand l'eau est touchée, le joueur perd un demi-cœur.

L'interface utilisateur

Graphismes réalisés avec le logiciel de dessin Krita.



Play *Parameters* *Quit*



Dead.
Victory!

Pour mettre mon nom, les variables de score de monstres tués et de pièces, j'ai utilisé TextMeshPro.

Le joueur

Au tout début de la conception du jeu, j'ai créé un GameObject parent Body (personnage rudimentaire avec des formes) muni de deux scripts : un pour le déplacement du personnage, et l'autre pour le mouvement orbital de la caméra, pour pouvoir rapidement me déplacer sur la carte et l'aménager. J'ai ensuite remplacé cela par le personnage animé Mousy de Mixamo, donné dans le Starter Pack.

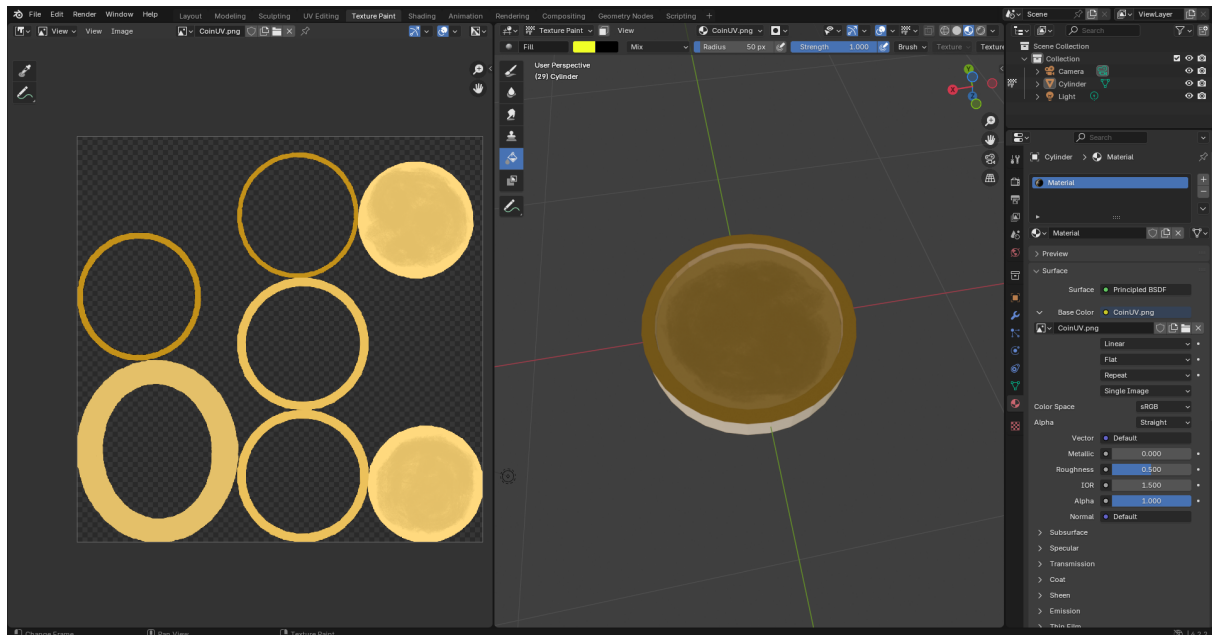
Pour pouvoir mettre en place Mousy, j'ai utilisé, modifié et mis en place :

- Différents scripts pour le mouvement, la rotation orbitale, un Scriptable (paramétrage vitesse...), un Input Action Asset (contrôles clavier, souris d'ordinateur...), un Animator Controller, le fbx de Mousy avec ses différentes animations.
- Animator, Player Input, Character Controller, les scripts assignés PlayerMovement.cs et AnimationSimple.cs, Rigidbody et Audio Source dans l'inspecteur de Mousy.
- La transition entre deux animations (Idle et Walking), en faisant le lien entre le script du mouvement et l'Animator Controller avec une variable.

Les modèles créés avec Blender

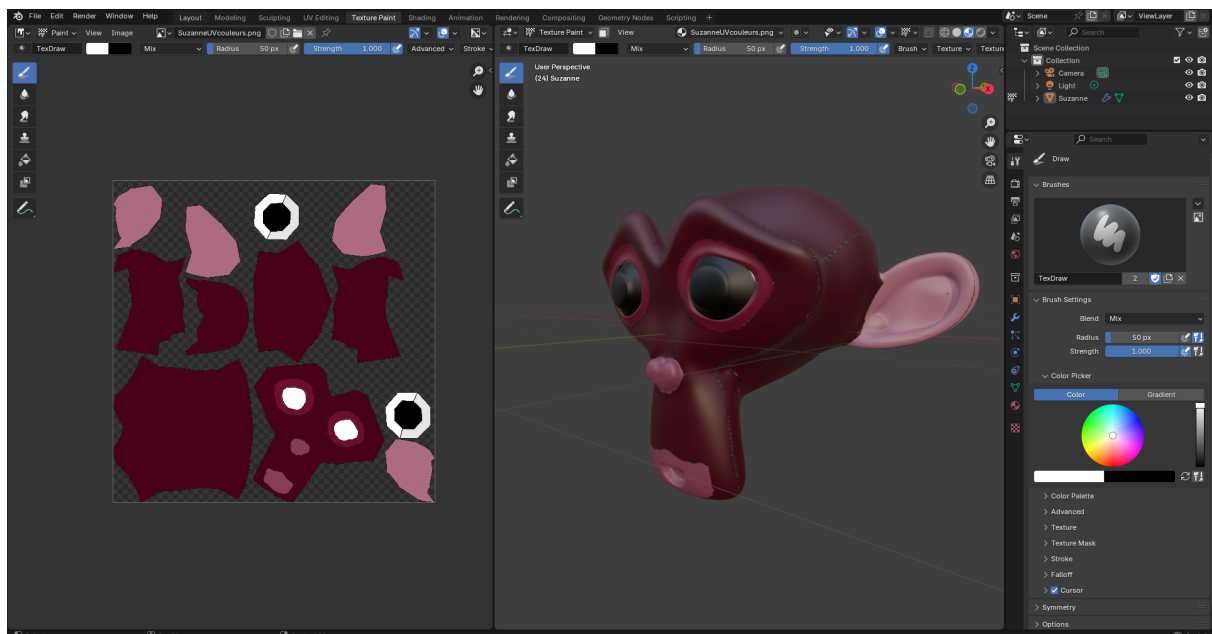
La colorisation des UV a été faite avec Krita (mark seam avec un raccourci sur les différentes lignes pour avoir des UV (patterns) propres → unwrapping UV pour mettre à jour l'UV).

La pièce (coin.fbx)



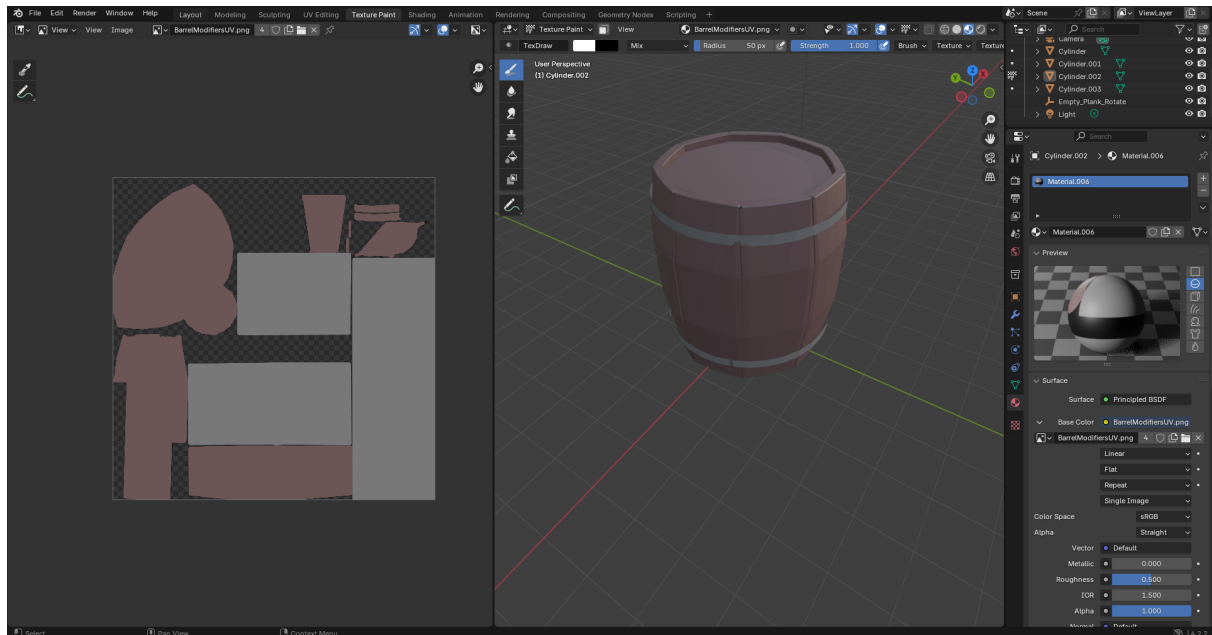
Création d'un cylindre avec un certain nombre de côtés → Scale pour obtenir l'épaisseur souhaitée → Inset Faces pour creuser chaque intérieur avec bordure (sur les deux faces).

Le singe Suzanne (suzanne.fbx)



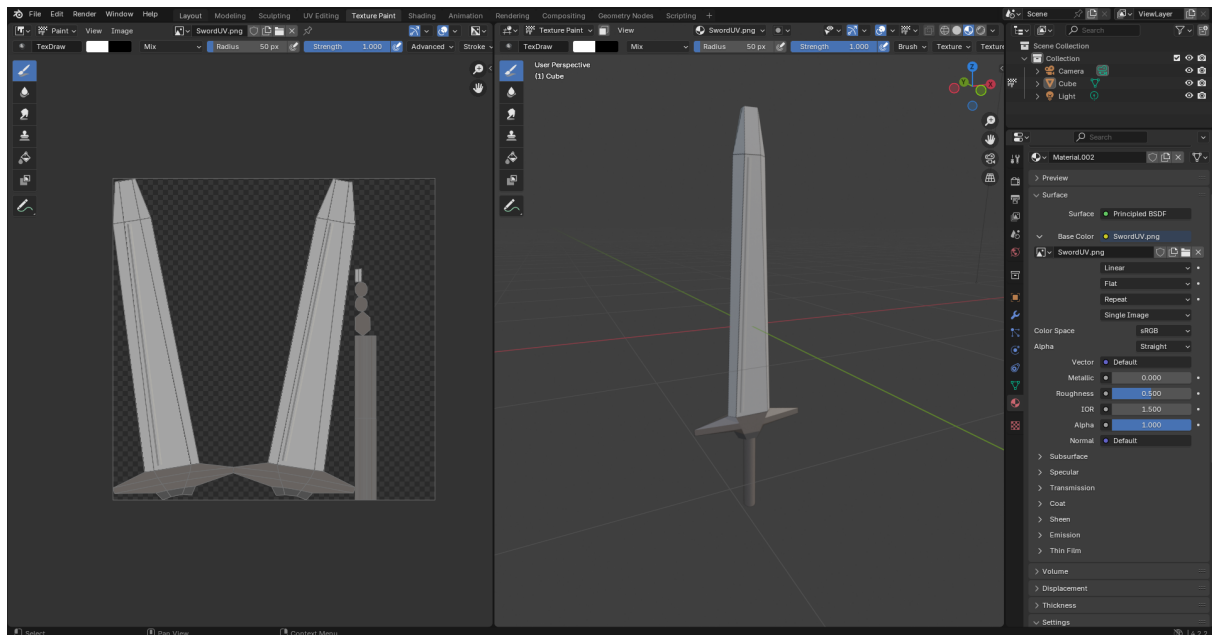
Utilisation du modèle de base « Suzanne ». J'ai juste eu à faire l'UV.

Le tonneau (barrel with modifiers.fbx)



Création d'un cube → Scale pour obtenir une planche (pavé) verticale → Sélection de points avec Select Box et utilisation de Poly Build pour déformer et avoir une forme de trapèze → Décalage de la planche hors du centre avec Move sur l'axe X (en arrière) → Ajout et application d'un modifier pour multiplier et décaler les planches entre elles, et régler la distance par rapport au centre → Un nouveau cylindre où l'on applique Scale et Move pour obtenir le bas → Duplication pour obtenir le haut → La même pour les cercles, en plus larges et avec les intérieurs vidés.

L'objet de mon choix : une épée (sword.fbx)



Création d'un cube → Scale pour obtenir une baguette → Loop Cut pour séparer (sommet, partie centrale, bas) → Sélection de points avec Select Box et déformations avec Poly Build → Ajout d'un cylindre en bas (Scale et Move).

Contrôles

- Cliquer sur Play ou presser "K" pour jouer.
- Cliquer sur Quit pour arrêter le jeu.
- En plein, cliquer sur le triangle gris en haut à gauche ou presser "Échap" pour revenir au menu.
- Presser "ZQSD" pour bouger, "Espace" pour sauter et "Shift" pour accélérer.
- Collecter les pièces, clés et épée au contact (en y allant dessus).
- Téléportation au contact des rectangles roses transparents.
- Donner des coups d'épée en pressant "F", et se mettre proche des monstres pour les tuer (on les tue en sept coups).
- Marqueurs verts au dessus de coffres qu'on peut ouvrir après collecte de clés. Presser "O" pour les ouvrir et "C" pour les refermer.

Sources

Textures

sol :

<https://i.pinimg.com/originals/e8/99/bf/e899bf31761509f5644d7b0c31e2a1b9.jpg>

eau :

https://img.freepik.com/free-photo/closeup-pool-water-texture_181624-59065.jpg

pierre :

<https://www.deviantart.com/agf81/art/Stone-Texture-Seamless-197981741>

herbe :

<https://rawcatalog.com/fr/asset/741/>

herbe verticale :

<https://www.anyrgb.com/en-clipart-2dkvj>

neige :

<https://everytexture.com/wp-content/uploads/2018/12/everytexture.com-stock-nature-snow-00002-1024x768.jpg>

skybox : Starter Pack

Assets

Mousy (Starter Pack) et les monstres Warrok : Mixamo

arbres : <https://assetstore.unity.com/packages/3d/vegetation/trees/tree-collection-pack-2017-76974>

clés : <https://assetstore.unity.com/packages/3d/props/rust-key-167590>

trophée : <https://assetstore.unity.com/packages/3d/props/3d-low-poly-trophy-and-cup-pack-created-with-fastmesh-asset-293357>

coffres : <https://assetstore.unity.com/packages/3d/props/3d-low-poly-chest-240360>

Sons

bruits de pièce et de pas : Starter Pack

bruit clé : <https://pixabay.com/sound-effects/search/key/> → Key get

bruit trophée :

<https://mixkit.co/free-sound-effects/win/> (Achievement bell)

bruit épée :

<https://pixabay.com/sound-effects/search/swoosh/> (Swoosh Sound Effect for Fight Scenes or Transitions - 2)

bruit coup de poing envoyé : <https://pixabay.com/sound-effects/search/punch/> (Punch 2)

musique de jeu :

<https://www.youtube.com/watch?v=P3nENGcXAXc> (Royalty free RPG Music - Cinematic Orchestral Fantasy Music by WOW Sound)