

Projet Unity :

Sauver le Royaume de Neuvi

@neuvi_doesstuff sur instagram



Sommaire

Projet Unity :.....	1
Sommaire.....	2
Introduction :.....	3
Partie 1 : Création et animation de Neuvi.....	3
Partie 2 : Création de la map.....	4
Partie 3 : Coder le jeu.....	5

Introduction :

Notre projet Unity est en version 2022.3.36, récupéré des PC de l'IUT. Il utilise un kit d'assets pour la map +1 objet importé, et il seront mentionnés ci dessous. Le reste (pièce, tonneau, Neuvi, trône, téléporteur...) à été importé de nos projets Blender ou créés à partir d'objets Unity.

Le projet est disponible sur github ici :

https://github.com/SALOMON-Clement-2225077aa/Neuvi_Game_Unity

Répartition des tâches :

- Clément →
 - Création de toute la map Unity et de Neuvi.
 - Mise en place des pièces, clés, bombes, et leur fonctionnement avec l'UI.
 - Scripts prenant en charge les téléporteurs, les bombes, la mort du joueur, l'animation de la porte, les déplacements du boss...
- Nathan →
 - Déplacement et animations de Neuvi
 - Création des 3 menus du jeu
 - Finalisation du combat contre le boss
 - Mise en place de tous les sounds effects

Partie 1 : Création et animation de Neuvi

Avant même de créer Neuvi et son trône sur Blender, nous avons déjà commencé à créer et à animer Neuvi sur Unity.

Il est donc plus rudimentaire, moins détaillé, et deviendra notre personnage jouable.

J'ai utilisé l'animationController de unity pour gérer les animations de neuvi, notamment celui de sa marche.

L'animationController permet de gérer les différents états d'animations et les transitions entre elles.

Partie 2 : Création de la map

Une fois Neuvi capable de se déplacer, la deuxième étape a été de construire la map. Il fallait quelque chose de pas trop grand, mais qui comportait tous les éléments demandés : téléporteur, clés, ennemis, pièces, tonneau...

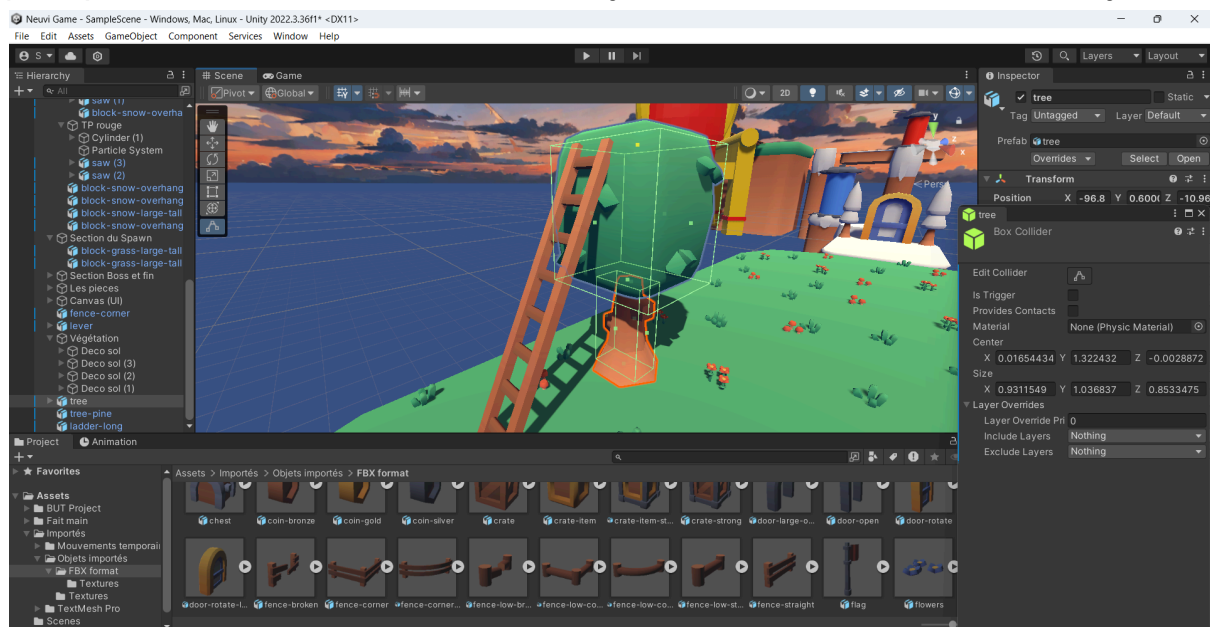
Ne sachant pas trop par où commencer, j'ai cherché de l'inspiration sur Youtube, et j'ai trouvé cette courte vidéo :

https://youtu.be/ulFc6p3hQzQ?si=TR5TkDdU_dDY6bJs

qui présente comment faire une map facilement (en utilisant des tilemaps, ce que je n'ai pas fait) mais en partant d'un 'Platformer Kit' :

<https://kenney.nl/assets/platformer-kit>.

Grâce à lui, j'ai pu rapidement placer mes premiers blocs, faire de la décoration, et avoir une map qui ressemblait à quelque chose. C'était très pratique, mais il a tout de même fallu rajouter les collisions à tous les objets.



La map est donc assez simple : une zone initiale, où Neuvi nous explique la situation, nous de récupérer les bombes et la clé pour aller battre la terrible grenouille Jacob. Une fois fait, Neuvi peut retourner sur son trône et récupérer sa couronne.

La couronne est le seul objet que nous avons importé en dehors du platformer kit. Je l'ai trouvé ici →

<https://free3d.com/3d-model/crown-iron-v2--346892.html>

Les images des menus ont été générées par IA pour avoir une cohérence entre elles et une bonne qualité à moindre coût.

Partie 3 : Coder le jeu

Une fois la map créée, j'ai voulu commencer par mettre en place le téléporteur, pour pouvoir accéder à la partie de la map. Le code est assez simple, il suffit de demander en header un objet Unity qui fait office de point d'arrivée, et de téléporter le joueur sur ses coordonnées. Une fois cela fait, on glisse le code sur le téléporteur, le paramètre 'on trigger', et assigne un objet vide comme destination dans l'inspecteur.

Après ça, j'ai importé ma pièce de Blender, et j'en ai placé quelques-unes un peu partout. Pour permettre à Neuvi de les récupérer, il a fallu d'abord mettre en place une interface avec un compteur. Puis, quand Neuvi rentre en collision avec une des pièces, cela incrémente le compteur de l'UI et supprime la pièce.

J'ai ensuite fait un système similaire pour ramasser la clé, mais cette fois-ci c'est un booléen : soit j'ai la clé et je l'affiche dans l'inventaire, soit je ne l'ai pas et elle n'est donc pas affichée. Ce booléen me permet de savoir si, quand j'entre à proximité de la porte (box collider du cadenas de la porte étendue à une grande zone devant la porte), elle doit s'ouvrir ou non.

Il était demandé d'avoir un coffre et une animation d'ouverture avec la clé, mais nos objets sont cachés dans les tonneaux Blender à exploser avec la bombe. J'ai donc fait l'animation sur la porte → Le cadenas saute et la porte s'ouvre.

En parlant des bombes voici leur fonctionnement : une fois récupéré, le booléen hasBomb est mis sur vrai, et le jeu détecte quand le joueur clique sur la touche 'E' du clavier. Quand c'est le cas, il crée un objet bombe devant le joueur qui est d'office supprimé 2 secondes après sa création.

("Destroy(placedBomb, 2f);").

Au moment où elle est détruite, j'invoque un 2e objet créé préalablement qui est une grande sphère rouge représentant l'explosion, auquel j'ai collé un script qui exécute un script de mort et le fait arriver à l'écran de game over (donc le tue).

Ce même script de mort est aussi utilisé pour la zone mortelle et le boss.

On peut choisir de tuer le boss ou non, dans tous les cas, on peut détruire la barrière empêchant d'arriver à la fin, mais selon cette décision la fin sera différente.