JEU 2D UNITY

Epic Adventure

M1 DEDI
STEPHANE DERICK KUEVIDJIN
FLORENTINE PASBECQ

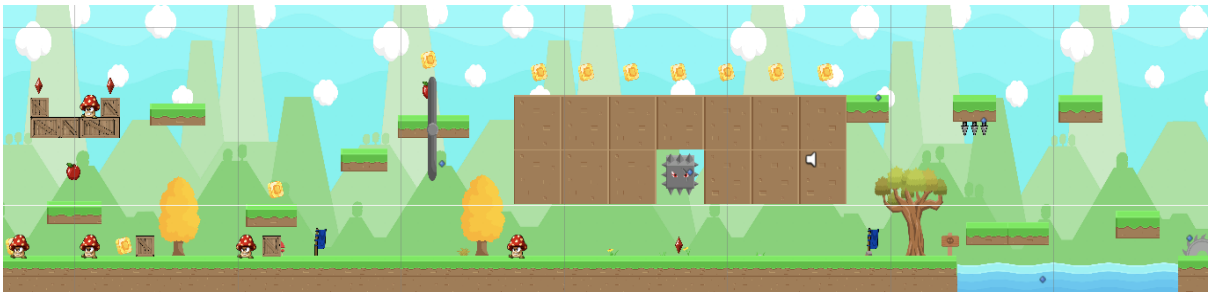
Sommaire

Partie 1 : Présentation de l'idée

Notre idée de départ était d'avoir un vaisseau qui tire ou un personnage qui évolue dans les différents niveaux de plateforme 2d.
Nous sommes finalement partis sur l'idée d'un personnage du type super mario qui nous permettait d'avoir plus de possibilités pour les niveaux.
Le personnage doit avancer en ramassant des pièces et des diamants au cours du jeu tout en évitant les ennemis pour passer au niveau suivant.
Nous avons créé un jeu 2D sur plusieurs niveaux avec des univers différents pour chaque niveau.
Etant donné que le personnage ramassera des pièces et des diamants au cours de son voyage, nous avons décidé de nommer le jeu **Epic Adventure** (aventure épique en français).

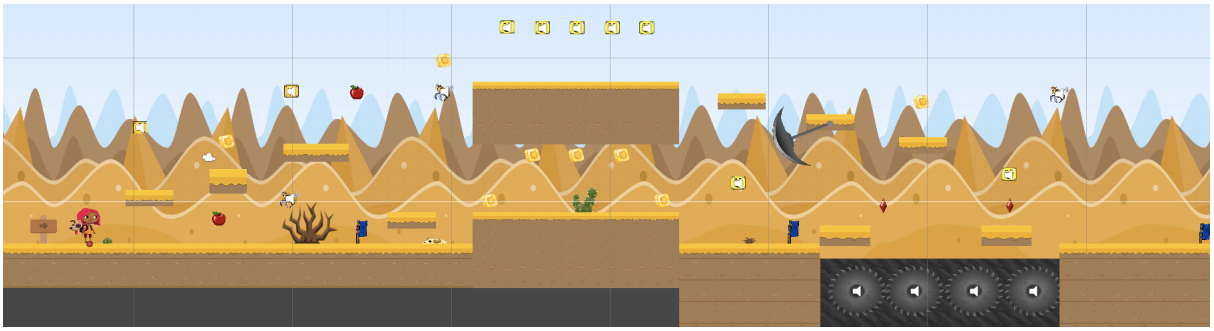
Présentation des univers et du personnage :

- Niveau 1



Le niveau 1 est axé sur la nature avec beaucoup de végétation, des fleurs, un cours d'eau, un souterrain et une musique qui colle à son univers. Les ennemis y sont généralement des **champignons** que le personnage écrase s'il veut survivre.

- Niveau 2



Le niveau 2 est axé sur le désert avec de la végétation différentes, des blocs d'une autre couleur et une musique différente de l'autre niveau. Les ennemis de ce niveau sont des **abeilles** qui suivent le personnage lorsqu'il est à une certaine distance et qui essayeront de le piquer.

Schéma de conception du niveau 1 :

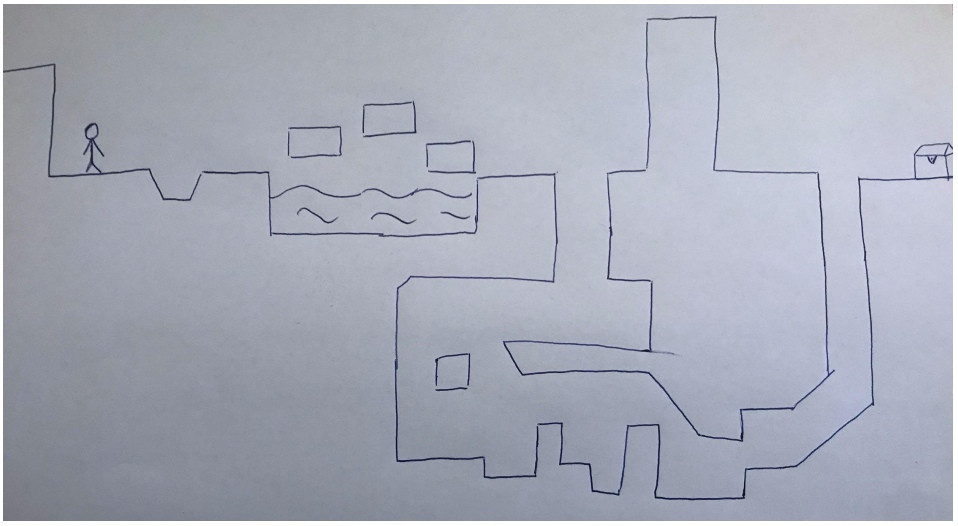
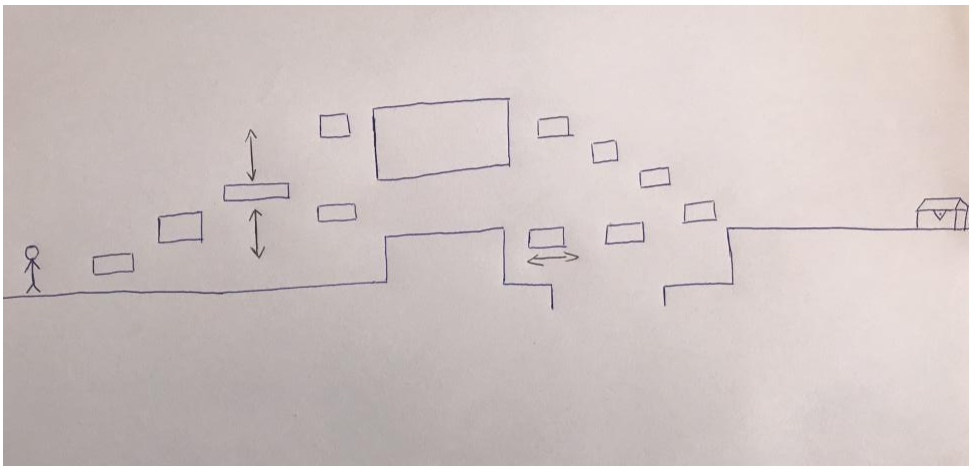


Schéma de conception du niveau 2 :



- Personnage

Le personnage que nous avons choisi a l'apparence d'une petite aventurière avec un sac à dos en forme de tête de chien.

Partie 2 : Gameplay (fonctionnalités du jeu)

a) Histoire

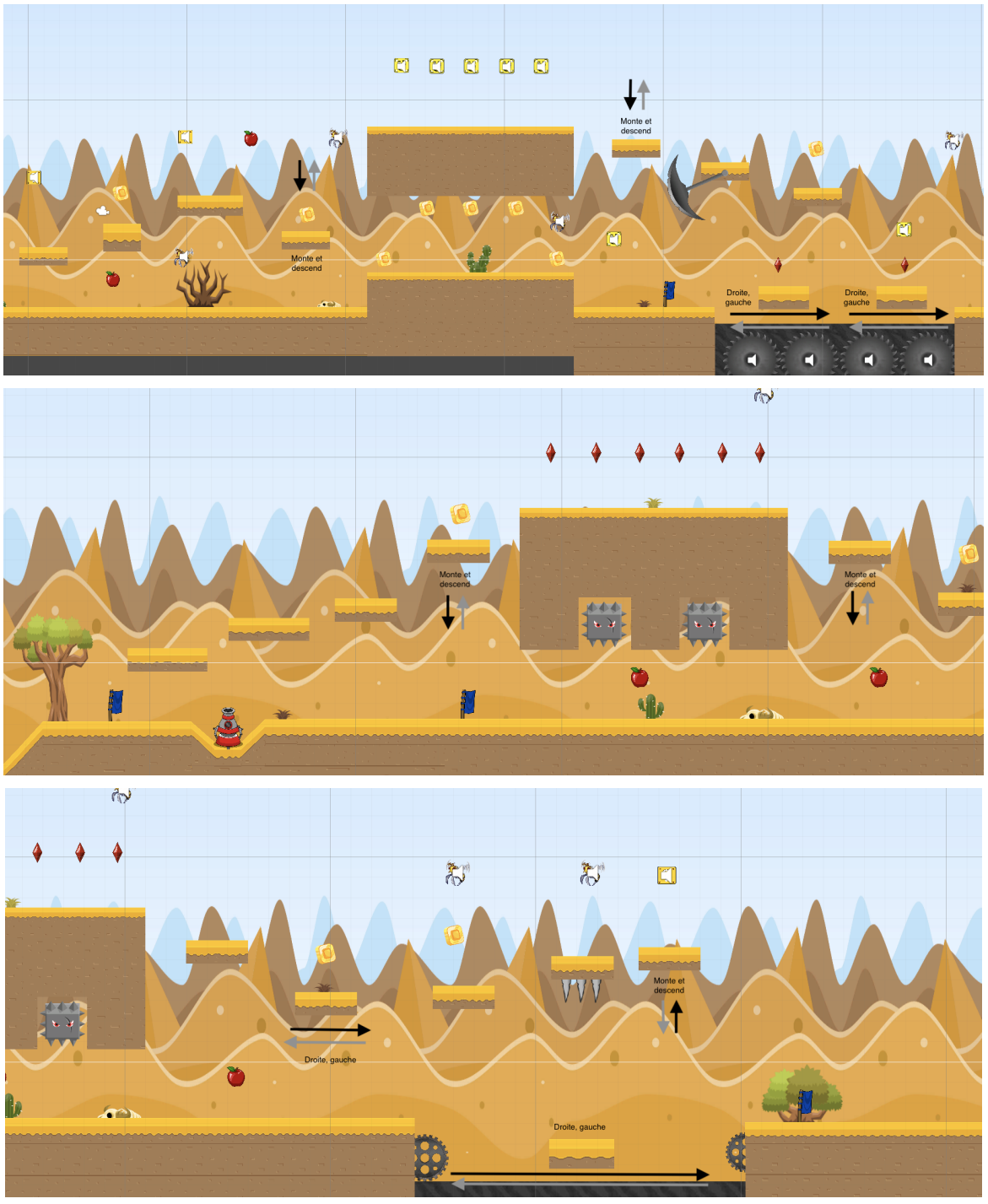
Le personnage du joueur est une petite fille qui se balade puis elle finit par se perdre mais pour retrouver son chemin, elle doit survivre et gagner tous les niveaux.

b) Niveau 1

Lorsque le joueur lance le jeu, il se retrouve dans le premier niveau. Un panneau lui indique d'avancer.
Pour ce premier niveau, nous avons pensé à laisser du temps au joueur pour s'adapter dans le jeu (saut, double-saut , déplacement…) puis il a une épreuve à traverser : un labyrinthe souterrain.
Si le joueur tombe ou touche un ennemi, il perd une vie et réapparaît. Le personnage dispose de 4 vies. Une vie est représentée par 3 cœurs. Si le personnage epuise toute ses vies et meurt, c'est la fin du jeu (Game Over).
Donc le jeu offre une fonctionnalité de réapparition au dernier point de sauvegarde représenté par un drapeau bleu.
Le but du niveau étant de s'adapter au jeu, de passer les différents ennemis et épreuves pour arriver à finir le niveau 1 et accéder au niveau suivant.

c) Niveau 2

Le niveau 2 reprend le même fonctionnement que le niveau 1 mais dans un environnement désertique.
Le jour y affronte des abeilles qui veulent le piquer. De plus, il y a plusieurs types de pièges notamment des lames ou des piques qui sortent de la terre que le personnage devra éviter s'il veut s'en sortir.
Le joueur doit avancer et sauter sur les plateformes en mouvement (verticales, horizontales) ou non. L'épreuve finale du jeu est l'épreuve de **l'araignée géante** .



À la fin du niveau 2, le joueur affronte une araignée géante en l'évitant et en lui lançant des pommes jusqu'à ce qu'elle soit battue. À ce moment, une plateforme apparaît, permettant au personnage de sortir du trou.

d) Objets et récompenses

Le personnage ramasse différents items et récompenses comme des :

- Pommes rouges (à lancer sur les ennemis).
- Pièces d'or.
- Diamants rouges.

Tous ces éléments peuvent se trouver n'importe ou dans le jeu ; le joueur devra parfois les dénicher en cassant des caisses et des briques.

Partie 3 : Scripts

Pour réaliser ce jeu, nous avons dû réaliser plusieurs scripts. Chaque script a été commenté (dans la limite de l'utile), dans le but d'en expliquer le fonctionnement.

a) Déplacements

Pour le déplacement du personnage, nous avons pensé à un mode de déplacement normal et à un mode vitesse qui permet au personnage d'accélérer. Nous avons réalisé les scripts :
PlayerController.cs pour l'ensemble des déplacements du personnage. Nous avons simulé une gravité et une vélocité permettant au joueur de se mouvoir dans les directions gauche et droite. Concernant la vitesse , nous avons mis en place un multiplicateur qui permet de régler la vitesse du personnage afin que celle-ci ne s'incrémente pas à l'infini (**Run Multiplier**). Le personnage a un rigidbody.
Concernant le saut du personnage, nous avons pensé à une fonctionnalité supplémentaire qui est celle du double saut. Ainsi grâce à la méthode **GroundCheck**, le personnage détecte s'il est sur GameObject de layer "ground" et peut effectuer un saut. Ainsi lorsque le personnage est en l'air et qu'il n'a pas encore touché le sol , le doubleJump qui était initialement à False passe à true lui permettant de rebondir dans les airs. Nous avons également effectué un contrôle sur le saut, Le joueur ne peut par exemple pas faire des sauts successifs à l'infini, de même que la vitesse des sauts (**Jump Speed**) a été réglée de manière à ce que le personnage ne saute pas hors du niveau.

b) Objets, récompenses et ennemis

Les différents objets et récompenses sont centralisés dans un GameObject de type Canvas. Grâce à une incrémentation, nous pouvons voir le nombre de pièces et de diamants augmenter au fur et à mesure que le personnage les collecte.
Pour cela nous avons utilisé deux scripts appelés : **CoinController.cs** et **DiamondController.cs** qui permettent d'affecter un nombre de points aux récompenses ramassées (via la fonction **Point to Add**) et de donner la valeur d'une pièce ou d'un diamants (**Coin Value**).
Le joueur peut également ramasser des pommes qui lui seront utiles pour se défendre durant le jeu. Grâce au script **Ammo_item.cs** que nous avons écrit, il y'a une incrémentation du nombre de pommes à chaque fois qu'il ramasse une pomme rouge. C'est-à-dire que le personnage ramassera 9 pommes, puis 10 pommes, puis 11 pommes etc...

Pour les ennemis que ce soit pour les blocs, les scies, les champignons, les abeilles ou encore l'araignée différents scripts ont été ajoutés pour leurs permettrent d'avoir des déplacements.

Concernant les blocs ennemis (Mace), nous avons créé des animations pour les déplacements hauts, bas. Sur chaque bloc qui touche le joueur, ce dernier se voit perdre une partie ou toutes ses vies, cela est dû à un Script quennous avons créé et qui s'appelle **HurtPlayer.cs** soustrait des vies au joueur.

c) Plateformes mobiles

Que ce soit pour le niveau 1 ou pour le niveau 2, nous avons utilisé des plateformes statiques et mobiles.
Pour ces dernières, il a fallu faire des animations pour créer le mouvement et régler leur vitesse, leur point de départ, leur point d'arrivée et les faire tourner en boucle.

d) Réapparition

Afin de simuler la mort du personnage et sa réapparition quand il touche un ennemi ou qu'il tombe, nous avons dû réaliser un script de résurrection du personnage par rapport au dernier point de sauvegarde.
Pour que le personnage réapparaisse après chaque chute ou erreur durant le niveau, des Checkpoints en formes de drapeaux ont été installés pour la réapparition tout le long des niveaux.
Nous avons pour cela écrit un script appelé **CheckPointController.cs** qui gère les points de sauvegarde et de réapparition du personnage.

Pour que le script fonctionne, nous avons placé une **Box Collider** avec l'option **is Trigger** en contrebas des trous et sur les ennemis pour pouvoir gérer le mécanisme.

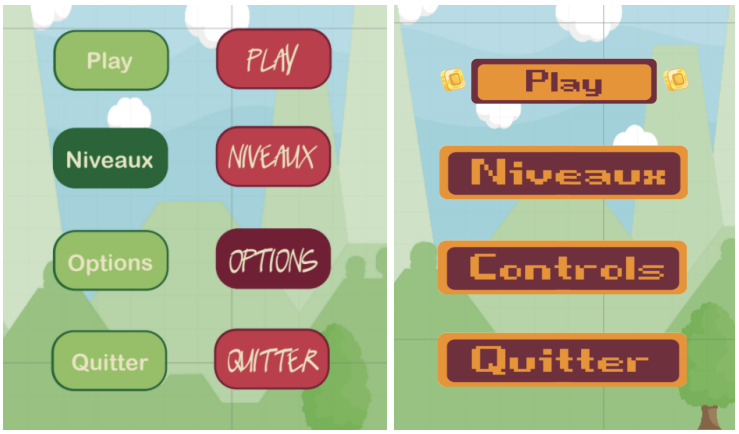
Ainsi quand le joueur entre en collision avec ses ennemis et meurt, il ne réapparaîtra qu'au dernier point de sauvegarde.

e) Fin de niveau

Lorsque le joueur atteint un drapeau à la fin du niveau, un script permet d'indiquer au joueur que le niveau est terminé et d'arrêter le jeu automatiquement en renvoyant vers la carte des niveaux.
Pour se faire, nous avons pensé à un script qui load la scène souhaitée et nous l'avons mis sur un drapeau avec un **Box Collider 2D** ainsi que l'option **is Trigger** activée.
Exemple : la fin du niveau 1 load le niveau 2 par le biais de la carte.

f) Menu principal

Pour le menu principal, nous avons dans un premier temps chercher un design pour les boutons.



Nous avons donc choisi la dernière version qui était plus aboutie.

Il est indispensable d'avoir un menu principal pour centraliser les commandes du jeu.
Nous avons créé une nouvelle scène, puis ajouté les différents boutons : "Play", "Controls", "Quitter".
Les boutons "Play" et "Quitter" mènent à des actions simples :

- Le bouton "Quitter" ferme l'application : Application.Quit();
- Le bouton "Play" mène au premier niveau du jeu grâce à une fonction StartGame qui charge le niveau 1 comme ci-dessous :
  SceneManager.LoadScene(firstLevel); ici FirstLevel étant notre niveau 1.

Ensuite, le bouton "Controls" a le même fonctionnement que celui du "Play", parce qu'il charge la scène "controls".
Voici un aperçu du code de chargement des contrôles :

```
public void Controls()
  {
     SceneManager.LoadScene(controls);
  }
```

g) Menu pause

Après avoir créé le menu principal, nous avons pensé à un menu pause accessible par la touche echap qui permet de mettre le jeu en pause sans perdre de vie et aussi pour quitter la partie en cours.
Le script **PauseMenu.cs** permet de figer l'arrière-plan et de ne rendre actif que les boutons "Continuer" et "Retour au Menu". Le joueur a également la possibilité de régler le volume grâce à un slider associé à un composant audioSource.

h) Canvas

Nous avons créé un gameObject de type **Canvas** pour réaliser le menu en haut et en bas de l'écran, permettant de voir les vies qu'il reste au joueur, le nombre de pièces et d'ennemis qu'il tue et le score total. Le script utilisé est le **FileManager.cs** qui a permis de faire le décompte de la vie du personnage jusqu'à ce qu'il soit game Over.

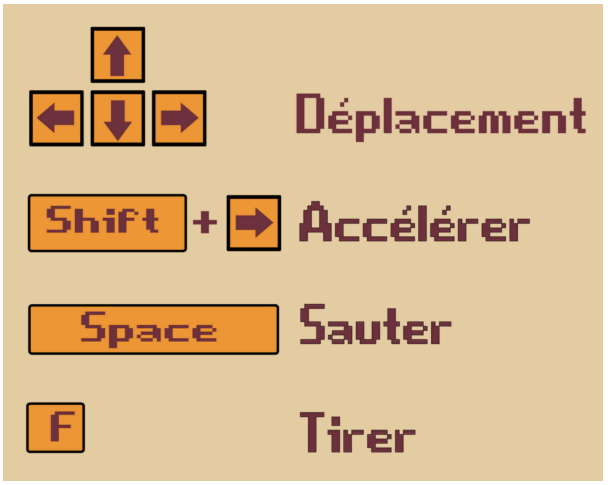
i) Problèmes rencontrés

Au départ du projet, nous avons eu de gros problèmes de synchronisation au niveau du projet unity ce qui provoquer des conflits entre les deux ordinateurs. Nous avons résolu ce problème en revenant sur une ancienne sauvegarde du projet et un changement de version d'unity.

Nous avons également rencontré des problèmes d'affichage pour les icônes du score pièce et du score diamant, car nous n'avons pas les mêmes dimensions d'écran l'un étant sur windows et l'autre sur mac.

Partie 4 : Notice utilisation des touches

Voici le fonctionnement des touches :

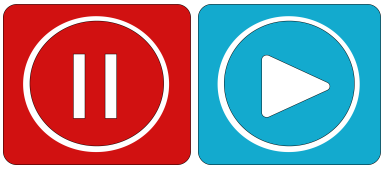


Pour accélérer sur certains claviers, il faut remplacer Shift + → par Maj + →.

Il y a aussi la touche **echap** qui permet d'accéder à un menu pendant un niveau en cours, le menu pause.

Quand le joueur est sur la carte des niveaux pour passer du niveau 1 au 2 ou l'inverse, il doit changer avec les flèches gauche et droite. Et pour valider le niveau auquel il va jouer, il doit utiliser la touche espace.

Pour le mobile, les déplacements se font au tactile et pour naviguer dans les menus ou pour mettre en pause pendant le jeu, il y a une touche play/pause qui a été ajoutée.



Nous avons aussi rajouté des boutons de joystick pour les déplacements et le saut du personnage.

Partie 5 : Éléments utilisés

a) Assets

Les assets ont été trouvés sur le store d'Unity et sur gameart2d.
Nous avons créé les décors des trois niveaux à partir de différents assets.

- Pour le niveau 1 :
  - Assets/Assets_utilises/Free Platform Game Assets/Tiles.
  - Assets/Assets_utilises/Free Platform Game Assets/Background.
  - Assets/Assets_utilises/Free Platform Game Assets/GUI (Update 1.7)/Environments.
  - Assets/Assets_utilises/Free Platform Game Assets/Enemies.
  - Assets/Assets_utilises/asset_niv_1.
  - Assets/Assets_utilises/Platformer Tileset/Props.

- Pour le niveau 2 :
  - Assets/Assets_utilises/deserttileset/png.
  - Assets/Assets_utilises/Free Platform Game Assets/Tiles.

- Pour le personnage : Assets/Assets_utilises/2DPlatformerController.

- Pour les pièces : Assets/Assets_utilises/Free Platform Game Assets/ Coin animation.

Nous avons surtout utilisé dans les assets les éléments de décor comme les blocs pour créer le sol et la végétation ou des récompenses, ennemis.

b) Musiques

Nous avons choisi plusieurs sons et musiques pour augmenter l'immersion dans le jeu. En voici les sources :

- Une musique pour le niveau 1 et les menus :
  Assets/Audio/sons/load.
  https://www.melodyloops.com/tracks/funny-adventure/

- Une musique d'ambiance pour le souterrain niveau 1 :
  Assets/Audio/music/musiques utilisées/
  eerie-cave-ambience-background-effect.
  https://www.youtube.com/watch?v=I_EH6nqQK5Y

- Un musique pour le niveau 2 :
  Assets/Audio/music/musiques utilisées/desert-sound-sound-effect.

https://www.youtube.com/watch?v=rWSWHkljU0c

- Un son pour les pièces :
  Assets/Audio/sons/Coin.wav
  TMK | Downloads | Sounds & Music | Sound Clips | Super Mario World (SNES) (themushroomkingdom.net)

- Un son pour les pommes (récupération) :
  Assets/Audio/sons/item_found.
  TMK | Downloads | Sounds & Music | Sound Clips | Super Mario World (SNES) (themushroomkingdom.net)

- Un son pour les pommes (lancer) :
  Assets/Audio/sons/Pop Gun.wav
  SoundFX - OneDrive (live.com)

- Un son pour le saut :
  Assets/Audio/sons/Player Jump.
  TMK | Downloads | Sounds & Music | Sound Clips | Super Mario World (SNES) (themushroomkingdom.net) (kick.wav)

- Un son pour écraser les ennemis ou taper sur les blocs :
  Assets/Audio/sons/smw_kick.wav.

- Un son pour les abeilles :
  Assets/Audio/sons/wasp.wav.
  SoundFX - OneDrive (live.com)

- Un son pour l'araigné :
  Assets/Audio/sons/spider.mp3.
  Freesound - "Angry spider monster" by Darsycho

- Un son quand le personnage s'affaiblit :
  Assets/Audio/sons/playerhurt.wav.
  Freesound - "Hurt2 - "It hurts!"" by lemonjolly

- Un son quand le perso meurt :
  Assets/Audio/sons/player_died.
  SoundFX - OneDrive (live.com)

- Un son pour les drapeaux de sauvegarde, pour les diamants :
  Assets/Audio/sons/Checkpoint.wav.
  SoundFX - OneDrive (live.com)
- Un son de victoire pour la fin des niveaux :

Assets/Audio/sons/victory.
Freesound - "victoryff.swf.mp3" by rezyma

- Un son de gameover :
  Assets/Audio/sons/Playerdied.
  Freesound - "Game Over Arcade" by myfox14

- Une musique pour quand on clique sur les boutons (retour, play) :
  Assets/Audio/sons/menu selection.

- Un son quand on est sur la carte des niveaux :
  Assets/Audio/sons/Level Select.
  Music - OneDrive (live.com)

- Un son pour le générique :
  Assets/Audio/sons/Intro.