



DEV&CO

RAPPORT DE SOUTENANCE 1

---

## Objectif Lune !

---

# Objectif Lune !



melody.huang  
chloe.eap

hengrui.ye  
nael.hulice-mencle

15/01/2024

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Présentation du jeu</b>	<b>2</b>
<b>3</b>	<b>Présentation du planning de réalisation</b>	<b>2</b>
3.1	Répartition des tâches . . . . .	2
3.2	Tâches communes . . . . .	4
<b>4</b>	<b>Présentation des tâches qui ont été réalisées</b>	<b>4</b>
4.1	Mélody HUANG . . . . .	4
4.2	Chloé EAP . . . . .	8
4.3	Hengrui YE . . . . .	12
4.4	Nael HULIC-MENCLE . . . . .	16
<b>5</b>	<b>Avancement du projet</b>	<b>18</b>
<b>6</b>	<b>Ressenti du projet</b>	<b>19</b>
<b>7</b>	<b>Conclusion</b>	<b>19</b>
<b>8</b>	<b>Source</b>	<b>20</b>

## 1 Introduction

Le but de ce rapport de soutenance est de montrer notre avancement sur le projet, nos difficultés et la répartition des tâches. Pour rappel, “Objectif Lune!” est un jeu de plateforme et d'aventure en 2D en C#. La réalisation du jeu vidéo était initialement prévue sur Godot. Nous avons finalement changé et sommes passés sur Unity avant le début de la réalisation du projet. En effet, Unity dispose d'une plus grande ressource de documentations et de différentes options qui nous permettra d'avoir le choix et de mieux comprendre. De nombreuses fonctionnalités que nous avions prévues au départ ont par ailleurs été modifiées pour que le projet puisse être réalisé et fonctionnel dans les temps. Les explications des changements ainsi que leurs remplacements seront expliqués dans la présentation des tâches réalisées par les membres du groupe.

Initialement, la suppléante du réseau était Chloé. Suite à une meilleure compréhension de la différence entre le multijoueur et le réseau, Hengrui devient le suppléant pour le réseau et Chloé la suppléante pour le multijoueur. Ce changement fait suite à une meilleure logique et à une meilleure compréhension sur le sujet. Cela n'empêche pas l'entraide au sein du groupe.

## 2 Présentation du jeu

L'histoire d’”Objectif Lune!” se déroule en 2550. Notre héros reçoit alors un message de son père seulement composé d'une position géographique. Curieux, le personnage principal décide de s'y rendre. Son père lui aurait-il préparé une surprise après toutes ces années d'absence ? Arrivé sur place, notre héros n'y trouva rien d'autre qu'une mystérieuse trappe. “Où peut-elle bien mener ?” N'ayant à peine le temps de se poser cette question qu'une énorme explosion retentit. Pris de peur et de surprise, notre héros se précipite à l'intérieur de la trappe. La nouvelle pièce était plongée dans le noir complet jusqu'à ce qu'une vidéo se lance. C'est une vidéo de la part de son père lui indiquant que, durant ces dernières années, ce dernier était membre d'une équipe de chercheurs dans le nucléaire dont l'expérience ratée a provoqué l'explosion. L'enregistrement continue son cours en expliquant que les survivants, transformés en mutants, errent à l'extérieur et que notre héros doit maintenant construire une fusée pour s'échapper vers la Lune. Heureusement pour notre personnage principal, le bunker contient tout le nécessaire pour survivre et se préparer : de la nourriture, de quoi se défendre, et bien d'autres choses à découvrir.

## 3 Présentation du planning de réalisation

### 3.1 Répartition des tâches

Dans le but d'obtenir un jeu jouable à la fin de l'année, nous avons réparti les différentes tâches dans le groupe.

Voici le tableau des répartitions des différentes tâches afin de réaliser le jeu vidéo. Chaque tâche dispose d'un responsable et d'un suppléant qui aidera et complètera la tâche.

Noms/Tâches	Mélody HUANG	Chloé EAP	Hengrui YE	Nael HULICE-MENCLE
Graphisme et Jaquette	Suppléant	Responsable		
Base du jeu	Suppléant		Responsable	
Son			Suppléant	Responsable
Code du personnage			Responsable	Suppléant
Code des niveaux	Responsable	Suppléant		
Code du bunker		Responsable		Suppléant
Site	Responsable			Suppléant
Intelligence Artificielle			Suppléant	Responsable
Multijoueur		Suppléant	Responsable	
Réseau	Responsable		Suppléant	

Tout d'abord, un rôle crucial : le code du personnage. Cette tâche concerne le développement du personnage principal que le joueur incarnera tout au long du jeu. Le code du personnage implique ses différents mouvements, c'est-à-dire le saut, se déplacer et attaquer. Sans ce personnage, aucune interaction ne peut avoir lieu. Cette tâche a été confiée à Hengrui aidé par Nael.

Secondement, nous avons le graphisme qui permet d'avoir un visuel sur le jeu. Les différents graphismes ont été dessinés et réalisés par une étudiante en école d'animation, Coline EAP. Celle-ci suit nos précisions et nos attentes concernant nos exigences. Chloé se charge donc de diriger les graphismes dans les détails avec l'aide de Mélody.

Troisièmement, le son permet de s'immerger dans le jeu. En effet, différents sons sont utilisés dans le jeu qui plonge le joueur dans un environnement propre au jeu. Chaque événement possède un son en particulier : un son pour les combats et un autre pour le bunker par exemple. Les différents sons sont réalisés par une passionnée par la musique, Alexandra TOURE suivant les directives de Nael avec l'aide de Hengrui.

Par ailleurs, un jeu ne peut pas se réaliser sans défis. Nous avons donc décidé d'implémenter différents niveaux avec des difficultés et des objectifs différents pour ajouter du challenge au jeu. Cette tâche est réalisée par Mélody aidée par Chloé.

Puis, notre jeu se déroule dans un monde apocalyptique d'où la présence d'un bunker, stratégique pour "Objectif Lune !". En effet, il ne s'agit pas d'un simple bunker. Il s'agit d'un point fondamental possédant différentes fonctionnalités : un ordinateur pour la sauvegarde, la carte du monde pour la sélection des niveaux mais aussi la fusée en elle-même à construire.

Le bunker est réalisé par Chloé avec l'aide de Nael.

Par la suite, nous avons le site internet. Le site internet permet de rendre notre jeu accessible à tous. Il regroupe également les avancements du projet ainsi que les détails du jeu. Le site, réalisé en HTML et CSS, est fait par Mélody avec l'aide de Nael.

Ensuite, nous avons les bases du jeu regroupant les systèmes de drop des différents matériaux. En effet, chaque matériau pourra être obtenu à la fin d'un niveau si l'objectif est atteint. Ces bases sont réalisées par Hengrui avec le soutien de Mélody.

De plus, l'intelligence artificielle nous permet de rendre le jeu plus automatique. Chaque ennemi se déplacera selon le contexte. L'ennemi commencera à attaquer le joueur s'il est proche et se déplacera seulement là où il peut. Cette manipulation est gérée par Nael avec le soutien de Hengrui.

Le jeu sera disponible en mode solo mais également en multijoueur. L'importance du multijoueur est la possibilité de réaliser un niveau à plusieurs. Il permet de tisser des liens avec une personne tierce. Il nécessitera une communication importante ainsi qu'un travail en équipe. La tâche est réalisée par Hengrui avec l'aide de Chloé.

Et enfin, le réseau. Il a pour but de permettre le multijoueur sur un même réseau internet. Le réseau est réalisé par Mélody et aidé par Hengrui.

### 3.2 Tâches communes

Chacun effectue de son côté les tâches dont il a hérité pour réaliser le projet. Pour que le jeu puisse fonctionner à la fin, nous travaillons actuellement sur github en créant chacun une branche afin d'envoyer son travail. Lorsque chacune de nos tâches sera finalisée, nous allons rassembler les différentes branches afin de pouvoir travailler sur une seule et unique branche. Le groupe a donc pour devoir de faire le travail à temps pour obtenir une avancée linéaire du projet et de régler les problèmes à temps.

De plus, chaque membre du groupe se doit de non seulement communiquer son avancée mais aussi de se mettre à jour sur celle des autres pour ainsi aider ses coéquipiers et donner son avis sans oublier de partager son ressenti.

## 4 Présentation des tâches qui ont été réalisées

### 4.1 Mélody HUANG

Je suis donc responsable du code des niveaux, du site internet et du réseau. Les niveaux ont un rôle crucial sur le jeu. En effet, ils permettent de captiver l'attention et l'envie au joueur de continuer à jouer. Pour rappel, il existe 12 niveaux.

Tout d'abord, nous avons le niveau d'introduction qui permet d'introduire l'objectif et les différentes fonctionnalités du jeu. Ce niveau est très simple avec une petite histoire racontant le synopsis du jeu. Ensuite, 10 niveaux seront consacrés à la recherche de matériaux pour construire la fusée. En effet, accomplir un niveau, nous permet d'obtenir des matériaux selon un système de drop réalisé par Hengrui. Les défis sont donc variés, il peut s'agir de tuer tous les ennemis, atteindre la porte de sortie sans se faire toucher par des ennemis, ou encore un labyrinthe d'où il faut s'enfuir. Un défi peut être repris pour un autre niveau mais avec une difficulté en plus, des ennemis plus puissants, ou un champ de vision restreint. Puis, un niveau secret se débloque après l'obtention de tous les matériaux et après le premier essai de décollage de la fusée. Une pièce sera manquante, il faudra donc éliminer l'ennemi final. Il s'agit donc du dernier niveau pour obtenir la pièce manquante.

Difficulté :

- Facile = peu de monstres
- Moyen = plus de monstres
- Difficile = plus de monstres et vision restreinte, c'est-à-dire qu'il ne peut voir qu'autour de lui et non au loin.

Explication des différents niveaux :

- Niveau 1 : difficulté facile, l'objectif est de tuer tous les monstres pour atteindre la sortie.
- Niveau 2 : difficulté facile, ne pas se faire toucher par un ennemi et s'enfuir. Si le joueur est touché par un ennemi, il devra recommencer le niveau.
- Niveau 3 : difficulté moyenne, il s'agit d'un labyrinthe, il faut trouver la sortie.
- Niveau 4 : difficulté moyenne, l'objectif est de tuer tous les monstres pour atteindre la sortie.
- Niveau 5 : difficulté moyenne, il s'agit de nouveau d'un labyrinthe mais les pièces sont à récupérer directement dans celui-ci.
- Niveau 6 : difficulté moyenne, ne pas se faire toucher par un ennemi et s'enfuir. Si le joueur est touché par un ennemi, il devra recommencer le niveau.
- Niveau 7 : difficulté moyenne, l'objectif est de tuer tous les monstres pour atteindre la sortie.
- Niveau 8 : difficulté difficile, ne pas se faire toucher par un ennemi et s'enfuir. Si le joueur est touché par un ennemi, il devra recommencer le niveau.
- Niveau 9 : difficulté difficile, labyrinthe d'où il faut trouver la sortie.
- Niveau 10 : difficulté difficile, ne pas se faire toucher par un ennemi et s'enfuir. Si le joueur est touché par un ennemi, il devra recommencer le niveau. Ajout de piège.

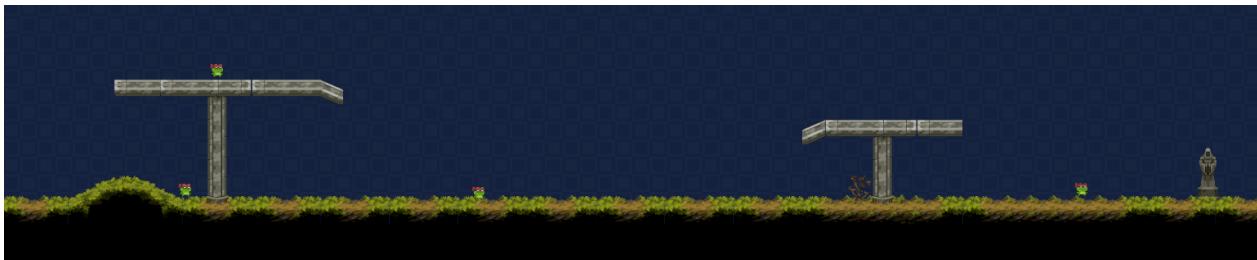
Le niveau 11 aura pour objectif de tuer le boss final.

Pour réaliser ces niveaux, j'ai d'abord développé une scène simple d'où le personnage principal se téléportera automatiquement vers le bunker lorsqu'il arrivera à la sortie du niveau. Une fenêtre apparaîtra également pour afficher les matériaux obtenus s'il a réussi. Sinon, une autre s'affichera demandant si le joueur souhaite retourner dans le bunker ou recommencer le niveau.

Je me suis donc principalement concentrée sur la partie fonctionnelle des niveaux et

non du visuel. En effet, les graphismes n'étant pas finis, nous avons pris un personnage vert sur Internet, libre de droit. Ce personnage représente l'emplacement des ennemis. Lorsque les graphismes seront terminés, il sera remplacé. Ce sera également le cas pour le fond et le personnage principal, qui, ici n'est pas représenté. Le personnage principal sera situé tout à gauche au début des niveaux. De plus, la statue ici présente, représente la sortie du niveau. Le personnage ne pourra sortir qu'une fois l'objectif atteint.

Voici un “visuel” rapide :



La difficulté majeure lors de la création des différents niveaux est le changement de niveau. J'ai eu du mal à comprendre Unity et son fonctionnement. J'ai donc pour l'instant créé différents niveaux qui existeront. Pour résoudre le problème de changement de niveau, j'ai suivi différents tutoriels et essayé d'implémenter.

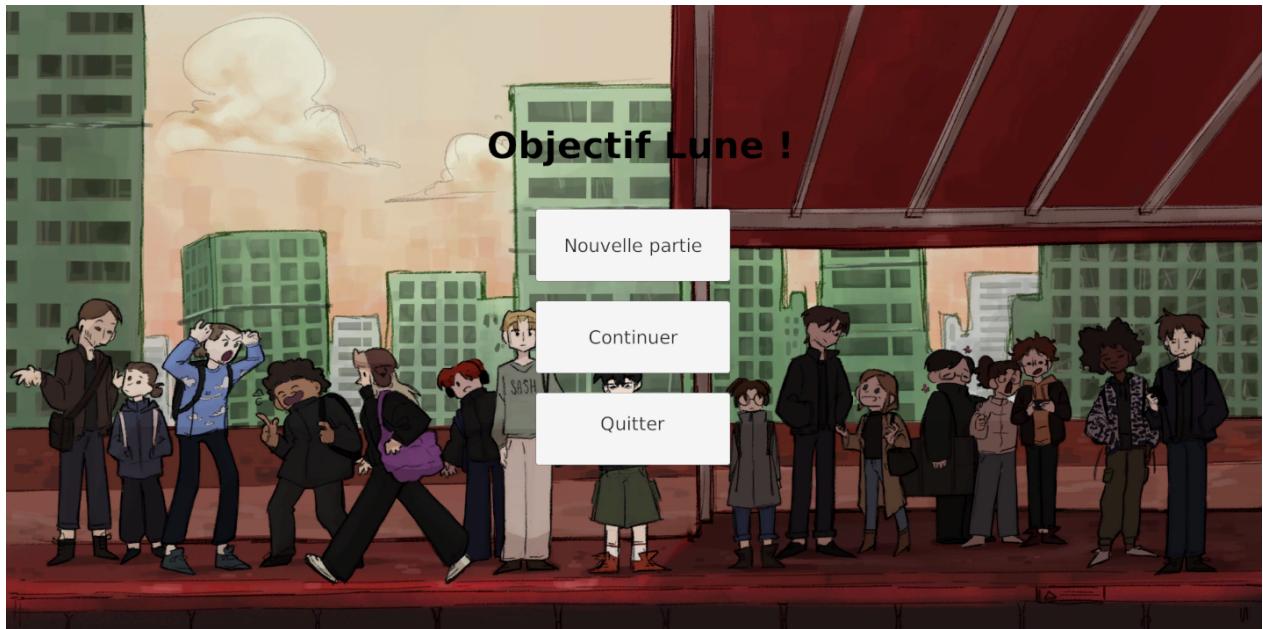
J'ai également implémenté un visuel lorsque le joueur mourra. Le joueur aura alors le choix de recommencer le niveau ou de retourner au bunker. Pour le moment, le changement de niveau n'est pas tout à fait fonctionnel de même pour la création du choix, une fois le niveau terminé. Je privilégie donc cette étape qui est indispensable pour la suite du jeu. Son visuel est donc le suivant :



Cette fonctionnalité n'est pas parfaite. En effet, lorsque nous lançons ce niveau, la fenêtre s'ouvre automatiquement sans pouvoir faire autre chose. Pour faire face à ce problème, j'ai décidé de créer une autre scène et de la superposer par-dessus la scène. Cette solution

n'est pas évidente et risque de générer des erreurs. Je trouverai une nouvelle solution pour faire face à ce problème.

De plus, j'ai créé l'interface du menu du jeu. Le visuel n'est pas parfait et sera amélioré pour la prochaine soutenance. Voici le visuel actuel :



Ensuite, j'ai donc développé le site internet. Pour l'instant, il n'est pas hébergé mais il sera prochainement accessible via un lien. Le site a été développé en HTML et en CSS. J'ai eu la chance d'avoir eu auparavant des stages qui m'ont initiée en HTML et en CSS. Malgré tout, je ne me rappelais plus exactement leur fonctionnement. J'ai donc suivi un tuto pour débuter puis j'ai fait quelques recherches sur internet pour trouver comment faire certaines fonctionnalités.

Voici sa page d'accueil :



Nous avons pensé à ajouter un logo de jeu en haut à gauche, d'où le vide et le carré. Quand nous passons la souris sur une des catégories en haut, celle-ci se blanchit. De plus, lorsque nous passons d'une page à l'autre, le nom concerné se blanchit et son écriture devient noire. Elle permet d'indiquer sur quelle page est actuellement l'utilisateur.



Cependant, le contenu des autres pages n'a pas encore été réalisé.

Étant responsable du réseau, je n'ai, pour l'instant, pas touché à cette partie. En effet, dans notre groupe, nous n'arrivons pas à faire la différence entre multijoueur et réseau qui était identique pour nous. Cependant, nous avons seulement réussi à faire la différence après que Hengrui ait commencé le multijoueur sans se rendre compte qu'il faisait le réseau. C'est pour cela que Hengrui a commencé le réseau, qui sera expliqué dans sa partie. Puis je reprendrai son travail pour le terminer à la prochaine soutenance.

## 4.2 Chloé EAP

Pour rappel, je suis la responsable du graphisme et du code du bunker.

Au niveau des graphismes, nous en avons terminé une grande majorité : celui du personnage principal, de l'ensemble du bunker, et des niveaux. Autrement dit nous nous sommes mis d'accord sur ce que doit dessiner Coline et nous savons à quoi va ressembler notre jeu

visuellement, la jaquette ainsi que l'animation du premier lancement du jeu. Cependant, nous n'avons pas encore le propre de tous ces éléments ce qui constitue un léger retard.

Apparence du personnage principal :



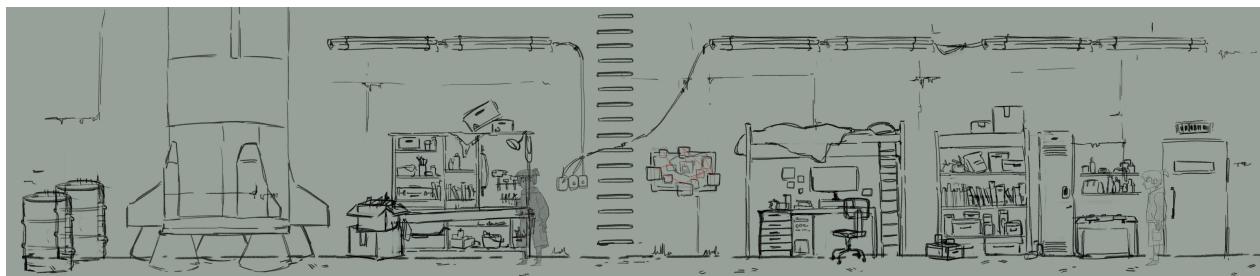
Apparence des mutants :



Le personnage principal porte une tenue civile puisque l'histoire se déroule en 2550 et qu'il a été pris au dépourvu par l'explosion. Les couleurs de ses habits sont plutôt sombres pour représenter le côté effacé et fatigué du personnage principal mais aussi pour qu'il puisse se fondre dans le décor du jeu et ainsi éviter de se faire repérer par les ennemis. Les couleurs principales du jeu sont le rouge pour représenter le danger et le vert pour contraster avec le rouge.

Les mutants ont été imaginés pour porter des éléments humains mais aussi des éléments d'animaux pour représenter la nature punissant l'homme après son échec, l'explosion nucléaire. Le mutant papillon de nuit est le boss final. Celui-ci possède deux formes, la première invulnérable, la deuxième dévoilant ses points faibles : ses yeux.

Plan du bunker et détails de la carte du monde :



Le bunker a été pensé pour être le plus minimaliste possible, il n'est censé contenir que

le nécessaire à la survie du personnage principal mais aussi le nécessaire pour les fonctionnalités du jeu. La carte du monde a, quant à elle, été pensée pour imiter une enquête menée par le personnage principal. Elle évoluera en fonction de l'avancée du joueur en dévoilant petit à petit les différents niveaux.

Visuel de la jaquette :



La jaquette du jeu reprend l'une des couleurs principales du jeu : le rouge. Nous y verrons le personnage principal dos à nous, face à la lune et équipée de son arme fétiche, une batte, entouré d'une ville en ruine.

Comme le suggère l'illustration, le titre du jeu sera écrit dans l'ombre du personnage principal.

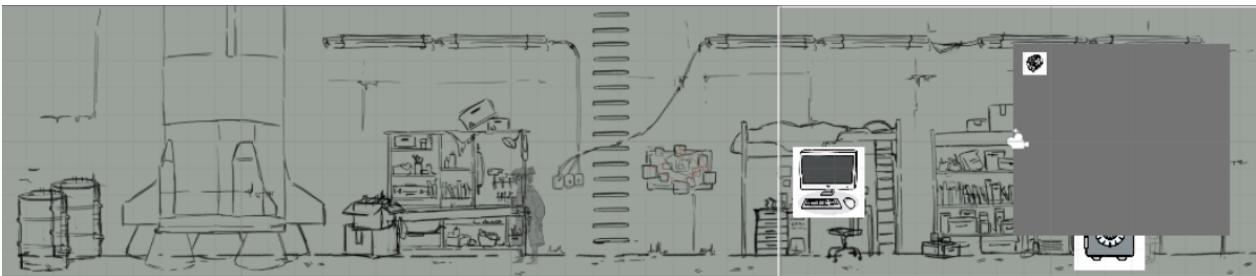
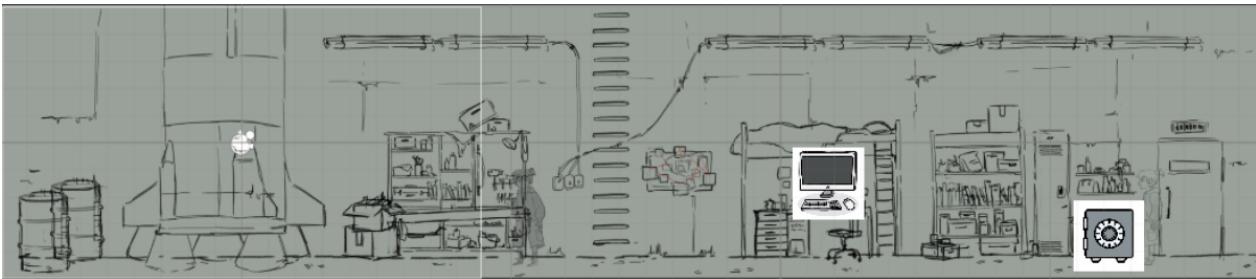
Toujours au niveau du graphisme, il nous reste donc à réfléchir au déroulement de l'animation qui clôturera le jeu, c'est-à-dire le décollage de la fusée, ainsi que l'apparence des composants de la fusée et celui de la fusée en elle-même. Les plateformes qui constituent les niveaux seront reprises d'un site en libre de droit. Pour la prochaine soutenance, l'idéal serait d'avoir les éléments du bunker au propre ainsi que l'arrière-plan des niveaux.

Au niveau du bunker, nous avons fait beaucoup de modifications afin de simplifier le jeu, pensant qu'il était à l'origine trop peu réalisable. Nous avons donc enlevé le point de synthèse et la bibliothèque. Le point de synthèse servait à synthétiser des matériaux en composant pour la construction de la fusée. Cependant il nous semblait trop compliqué d'implémenter

tous les matériaux, leurs taux de drop et le point de synthèse dans les temps. La bibliothèque servait quant à elle à entreposer des livres liés au système d'intelligence du personnage principal, que nous avons également enlevé. (Les raisons sont expliquées dans la partie d'Hengrui.)

Ne possédant pas les dessins officiels du bunker, j'ai créé un prototype de bunker à partir d'images auxquelles j'ai attaché du code. J'ai donc commencé le code de l'inventaire et du point de sauvegarde sans pour autant les avoir finalisés. Théoriquement, il n'est pas difficile de les programmer. La difficulté principale a résidé dans la synchronisation avec l'interface graphique : j'ai eu du mal à trouver comment faire interagir le personnage principal avec les éléments, de même pour comment faire apparaître les différentes fenêtres après l'interaction.

Voici à quoi ressemble les prototypes du bunker ainsi que celui de l'inventaire :



L'idéal serait de finaliser le point de sauvegarde et l'inventaire pour la prochaine soutenance. Il ne me restera alors que la carte du monde à programmer, la fusée ainsi que les dessins officiels à ajouter.

### 4.3 Hengrui YE

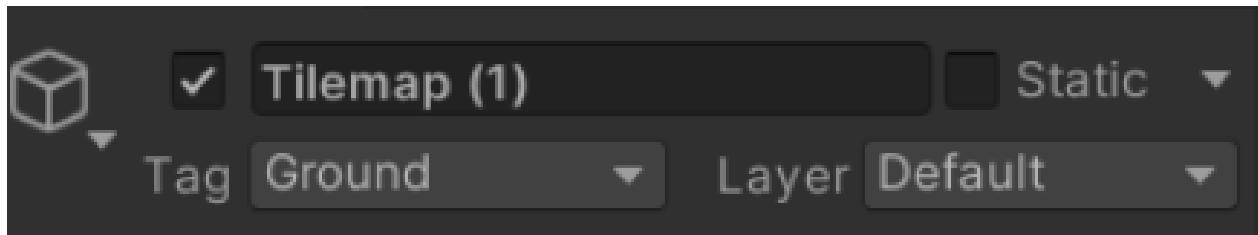
Pour ce projet, je suis celui qui s'occupe du code du personnage, des bases du jeu mais aussi du multijoueur.

Pour le code du personnage, j'ai déjà réalisé les mouvements les plus basiques pour notre personnage principal, c'est-à-dire les mouvements horizontaux (de gauche à droite), mais aussi les sauts. Durant la réalisation de ces mouvements, le saut fut un mouvement plus compliqué que les deux autres. C'est-à-dire que les mouvements horizontaux n'avaient aucune contrainte difficile : il suffit que le personnage puisse se déplacer lorsqu'on appuie sur

la touche dédiée, alors que le saut possède une contrainte qui le rend plus complexe : le saut doit pouvoir être réalisé seulement si le personnage principal est au sol. Et pour réaliser ceci, je me suis servi d'un composant de la classe GameObject qui est le composant tag.

```
0 references
private void Update()
{
    float horizontalInput = Input.GetAxis("Horizontal");
    rb.velocity=new Vector2(horizontalInput*movespeed,rb.velocity.y);
    if(Input.GetKey(KeyCode.Space)&&grounded)
    {
        Jump();
    }
}
1 reference
private void Jump()
{
    rb.velocity=new Vector2(rb.velocity.x,Jumpingforce);
    grounded=false;
}
0 references
private void OnCollisionEnter2D(Collision2D col)
{
    if(col.gameObject.tag=="Ground")
    {
        grounded=true;
    }
}
```

Comme le montre cette image, les deux premières lignes de code permettent au personnage de se déplacer horizontalement. Le reste constitue le code pour le saut. La fonction “OnCollisionEnter2D” est l’élément crucial faire sauter le personnage principal seulement lorsqu’il est au sol. Ce fonctionnement permet de transformer le booléen “grounded” seulement lorsque le personnage principal est en contact avec un élément qui possède le tag “Ground”. De ce fait, si on attribue le tag “Ground” au sol, alors le personnage principal ne pourra sauter que s’il est en contact avec le sol.



En ce qui concerne le code du personnage, nous avons aussi décidé de supprimer le système de force physique ou encore le système d'intelligence. Il n'y aura donc plus de système de synthèse, et le personnage principal pourra obtenir ses composants en tant que récompense de niveau. Nous avons décidé d'enlever ces deux systèmes car leurs présences a pour conséquence l'ajout de beaucoup d'autres fonctionnalités non primordiales ayant une grande complexité au niveau du code. Puisque nous sommes des débutants dans le domaine, nous avons donc préféré supprimer ces systèmes et privilégier les fonctionnalités fondamentales pour que le jeu fonctionne parfaitement. Pour les armes, il n'existe désormais qu'une arme principale, une batte, et une arme secondaire requise pour le niveau final. La batte est l'arme par défaut, fournie au début du jeu tandis que le briquet doit être ramassé dans un des niveaux. Le nombre de vie, initialement lié au système d'intelligence est maintenant attribué en fonction du niveau lancé et de l'avancée du joueur.

Au niveau du code de personnage, il me reste plusieurs fonctionnement à terminer, par exemple sa barre de vie, ramasser les armes qui sont présentes dans les niveaux, ou encore les mouvements d'attaques. Pour la prochaine soutenance, mon but est de terminer tous les fonctionnements, et pour la dernière soutenance, j'essaierai de corriger tous les problèmes qui peuvent toujours être présents dans les différents fonctionnements.

Au niveau de la base du jeu, je n'ai pas encore réalisé de code concret écrit en C# du système de taux de drop. Pour l'instant, le code de ce dernier repose uniquement sur quelques idées qui n'ont pas encore été concrétisées. Ce système sera constitué d'une variable qui va compter le nombre de monstres tués par le personnage lors d'un niveau. Ce nombre pourrait augmenter une autre variable représentant le pourcentage de drop. Ce n'est qu'à la fin du niveau que ce pourcentage sera utile pour déterminer le nombre de composants obtenus.

Au niveau du réseau, sa réalisation n'est pas encore terminée. Avec l'aide du Netcode for GameObject, un système proposé par Unity, la réalisation du début pour le réseau fut sans grosse difficulté. C'est-à-dire qu'à l'aide du Netcode for GameObject de nombreuses tâches sont directement réalisées par Unity, comme la synchronisation des données entre plusieurs joueurs.

Pour l'instant, le jeu peut fonctionner sur plusieurs personnages sur différentes fenêtres, cependant les différentes fenêtres doivent être lancées sur un même ordinateur.

```

public class NetworkManagerUI : MonoBehaviour
1 reference
{[SerializeField] private Button Serverbtn;
1 reference
[SerializeField] private Button hostbtn;
1 reference
[SerializeField] private Button Clientbtn;
0 references
private void Awake(){
    Serverbtn.onClick.AddListener(()=>{
        NetworkManager.Singleton.StartServer();
    });
    hostbtn.onClick.AddListener(()=>{
        NetworkManager.Singleton.StartHost();
    });
    Clientbtn.onClick.AddListener(()=>{
        NetworkManager.Singleton.StartClient();
    });
}
}

```

Ceci est un code simple qui permet au joueur d'accéder à l'hôte, ou au client ou au serveur lorsqu'il appuie sur l'un des trois boutons. Pour l'instant, il ne sert seulement qu'à tester le fonctionnement du jeu sur différentes fenêtres, mais ce type de code va aussi beaucoup aider par la suite, notamment pour les boutons de choix du mode solo ou multijoueur.

Pendant la réalisation du réseau, j'ai aussi rencontré quelques problèmes, mais qui ont pu être résolus à l'aide de quelques composants qui sont déjà présents dans Unity.

Le premier problème que j'ai rencontré est le déplacement du personnage, c'est-à-dire que lorsqu'on cherche à déplacer un personnage, l'autre personnage se déplace aussi. Ce problème est dû au fait que dans le script du mouvement, nous n'avons pas différencié les joueurs. Pour résoudre ce problème, j'ai trouvé une propriété déjà présente dans Unity.Netcode qui est le booléen "Is Owner", qui permet de dire si l'action demandée provient du joueur local.

```

private void Update()
{
    if(!IsOwner) return;
    float horizontalInput = Input.GetAxis("Horizontal");
    rb.linearVelocity = new Vector2(horizontalInput * movespeed, rb.linearVelocity.y);
    if(Input.GetKey(KeyCode.Space) && grounded)
    {
        Jump();
    }
}

```

Lorsqu'on ajouté cette propriété comme dans l'image ci-dessus, le problème est donc résolu.

Le deuxième problème est la synchronisation des données. C'est-à-dire que lorsqu'un des personnages se déplace sur l'une des deux fenêtres, sur l'autre fenêtre, le mouvement n'est pas abouti, ce qui pose donc un problème majeur. Cependant grâce à un autre composant de Unity, qui est le Client NetWork Transform, ce problème a été résolu facilement. Car avec l'ajout de ce composant dans le préfab du joueur, les données de l'action sont directement envoyées à l'autre client, ce qui permet donc la synchronisation des données et donc des mouvements.

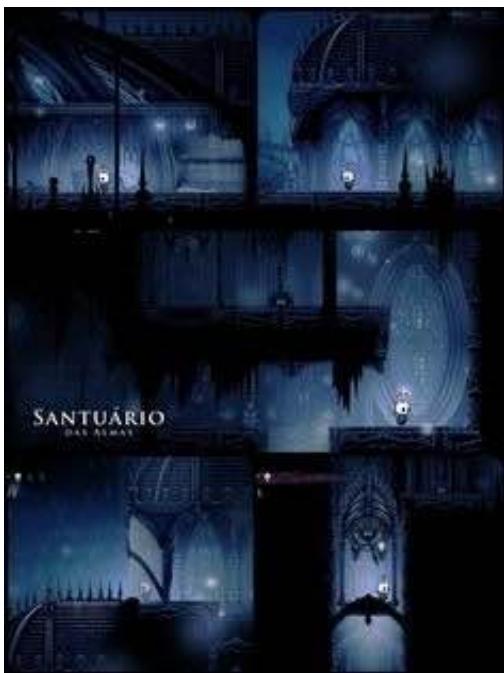
Concernant le multijoueur, nous laisserons le choix au joueur de jouer soit seul, soit à deux. Lorsque deux joueurs seront sur le jeu, l'inventaire ainsi que le but du jeu resteront identiques. Chaque joueur disposera d'une arme et il ne sera pas possible de gagner 2 fois plus de matériaux. Cependant, lors de l'entrée sur un niveau, les deux joueurs devront accepter pour lancer le niveau. Un niveau sera impossible seul. De plus, lorsqu'un des deux joueurs meurt, l'autre pourra continuer et terminer le niveau malgré tout. Mais s'il meurt lui aussi, le niveau devra recommencer.

#### 4.4 Nael HULIC-MENCLE

Pour ma part je me suis occupé de toute la partie sur les IAs et sur la partie de la bande sonore que nous allons utiliser pour notre jeu. Pour détailler ce que j'ai fait, je me suis principalement centré sur la documentation des outils que nous allons utiliser afin d'implémenter efficacement une IA dans notre jeu. Pour ce faire plusieurs méthodes peuvent être mises en œuvre mais nous avons opté pour l'utilisation de capteurs et de calcul de distance entre l'ennemi et le personnage. Vous l'aurez compris, la partie en cours de code sera donc centrée sur le suivi du joueur durant les niveaux ainsi que les attaques que peuvent subir le joueur. Ce qui a été le plus dur concernant la partie sur l'IA a été de comprendre ce qu'il fallait réellement faire. Par la suite cela a été de réussir à trier les idées et à commencer à voir pour la réalisation de celle-ci. Cela a été plutôt fastidieux et a pris du temps d'où le peu de codes qui a été pour l'heure produits. Cependant, à force de me documenter sur le sujet plusieurs idées sont devenues envisageables.

Pour la partie des bandes sonores, nous avons eu l'aide d'Alexandra, une passionnée de la musique. Étant responsable des sons, j'ai indiqué nos attentes et nos idées. Nous avons donc opter pour l'utilisation de 4 principales bandes sonores : la première sera pour l'écran d'accueil, la seconde sera dans le menu de sélection de niveaux ainsi durant l'ouverture de la carte, la troisième sera pour l'intérieur de l'habitat de notre personnage ainsi le bunker, les quatrièmes seront pour les niveaux. Afin de diversifier les émotions véhiculées par les musiques, celles-ci ont été créées spécialement pour notre jeu. Nous nous sommes basés principalement sur les couleurs, le jeu des couleurs est assez important pour le choix des morceaux à implémenter dans le jeu. Étant donné l'environnement apocalyptique du jeu, les musiques qui seront présentées durant le déroulement du jeu ne peuvent pas être joyeuses. C'est pourquoi la plupart des bandes sonores seront plutôt calmes et rappelleront un environnement en ruine.

Le choix des musiques est important dans un jeu pour transmettre des émotions différentes selon le moment. En termes d'inspiration, nous avons opté pour une inspiration assez diversifiée, celle-ci est basée tout d'abord sur les musiques d'animés (japonaises), sur "Geometry Dash", et "Undertale", un jeu sorti en septembre 2015. Le style de musique pour lequel nous avons opté est du type électronique. Concernant maintenant l'avancement du son, plusieurs bandes sonores sont quasiment prêtes mais méritent tout de même quelques retouches afin d'être finalisées. Nous espérons avoir fini ces bandes sons prochainement afin d'avoir le temps de les implémenter dans notre jeu vidéo. Ainsi plusieurs variations de bandes sons sont en production afin de varier les émotions du joueur. Les bandes sons déjà créées ont été inspirées de ces images :



## 5 Avancement du projet

Pour rappel, voici le tableau détaillé de l'avancement qui a été convenu. A chaque soutenance, un objectif est fixé afin de terminer le projet à temps.

Noms/Tâches	Soutenance méthodologie	Soutenance 1	Soutenance 2	Soutenance 3
Graphisme et jaquette	20%	40%	60%	100%
Base du jeu	20%	50%	80%	100%
Son	0%	40%	100%	100%
Code du personnage	20%	50%	80%	100%
Code des niveaux	5%	20%	70%	100%
Code du bunker	10%	40%	70%	100%
Site	0%	30%	50%	100%
Intelligence Artificielle	0%	20%	50%	100%
Multijoueur	0%	20%	100%	100%
Réseau	0%	20%	70%	100%

Ci-dessus vous retrouverez le tableau d'avancement initial. Cependant, certaines tâches avancent moins rapidement que prévu.

Le graphisme et la jaquette sont toujours en cours de création mais prennent plus de temps que prévu. La fin est toujours attendue avant la dernière soutenance.

Les bases du jeu n'ont été développées que artificiellement, avec quelques calculs qui ont été réalisés mais pas sous la forme de code. Elles devraient être très prochainement mises en place.

L'intelligence artificielle a pris un peu de retard dû à un manque de ressources. Cependant, dès la prochaine soutenance, elle devrait être implémentée.

Certaines tâches ont, quant à elles, été réalisées dans les temps.

Chaque contexte aura un son, les sons n'ont pas été encore choisi définitivement et n'ont pas été ajoutés dans le jeu. Son suivi suit toujours le tableau d'avancement.

Les niveaux ont été fixés, c'est-à-dire que nous savons ce que chaque niveau fera. L'implémentation des différents niveaux sera, par ailleurs, plus longue à développer. Seul un niveau "test" a été implémenté pour tester ces différentes fonctionnalités.

Le code du personnage avance : il ne possède cependant pas encore toutes les différentes fonctionnalités (l'attaque) et l'animation du personnage n'a pas encore été ajoutée. En revanche, il se déplace sans soucis, ne se cogne pas contre des murs.

Le code du bunker, soit le point de sauvegarde, l'inventaire, prend forme petit à petit, se développe chaque jour. Son avancement est dans les temps.

Le multijoueur et le réseau ne sont pas des notions assez évidentes. Ces tâches sont dans les temps. Cependant, le réseau aura un léger retard. Pour résoudre ce retard, une focalisation sur cette tâche sera réalisée pour permettre d'avoir une application directe sur le projet dès la prochaine soutenance.

Quant au site internet, il a été réalisé en avance. Sa finalisation n'est pas encore faite mais il obtiendra des mises à jour avant chaque soutenance.

## 6 Ressenti du projet

Nous n'avons pas imaginé à quel point la réalisation du projet allait être compliquée. Nous avons tous déjà réalisé des projets de groupe auparavant et tout se passait bien. La difficulté de ce projet est la nécessité d'une grande communication autour des différentes tâches. En effet, chaque tâche possède des particularités qui empêchent d'avancer et de la réaliser. Une tâche non réalisée empêche l'avancement d'une autre. L'entraide est donc un point clé. Certains membres du groupe avaient déjà légèrement codé auparavant, puis d'autres qui n'avaient pas touché à du code avant. La compréhension de chaque notion est importante. Nous avons un sentiment de stress de ne pas réussir à faire le projet à temps et de répondre aux attentes que nous avions fixées sur le jeu.

## 7 Conclusion

Pour la prochaine soutenance, nous espérons mettre en place différents niveaux, un multijoueur fonctionnel. Mais également un bunker avec son point de sauvegarde et son inventaire fonctionnels. Toutes les autres tâches avanceront également. Nous prêterons plus d'attention sur le réseau, qui n'est pas une notion évidente que nous n'avons pas encore mis en place.

Pour conclure, le projet a été mis en marche depuis novembre avec ces différentes recherches, mise en accord sur les perspectives du jeu et son objectif. Le programmation du code qui a débuté en décembre et qui continue d'évoluer.

Nous sommes partis de rien pour atteindre, nous l'espérons, le sommet. Il s'agit d'un long chemin avec des obstacles qui nous barrent la route.

## 8 Source

**Graphisme** : Coline EAP, étudiante en école d'animation

**Contact** :

- Instagram : meimeii.draws
- LinkedIn : Coline EAP

**Sons** : Alexandra TOURE

**Contact** :

- toure.alexandra@yahoo.com
- 0652792744

**Plateforme** : <https://anokolisa.itch.io/moon-graveyard>