

Treasure Hunt



UbiHard

23 Juin 2021

Table des matières

1	Présentation de l'équipe	3
2	Rappel du cahier des charges modifié	5
2.1	Présentation du projet	5
2.1.1	Le monde	5
2.1.2	Le personnage	6
2.1.3	L'environnement	6
2.1.4	Les ennemis	6
2.1.5	Les objets	9
2.1.6	Les pièges	10
2.1.7	Les consommables	10
2.1.8	Les énigmes	11
2.1.9	L'interface	12
2.2	Multijoueur	13
2.3	I.A	13
2.4	Objet de l'étude	13
2.5	Etat de l'art	14
2.5.1	Histoire	14
2.5.2	Exemples	15
2.6	Tâches	16
2.6.1	Planification des tâches	16
2.6.2	Description des tâches	18
2.6.3	Bonus	19
2.7	Technologie et méthodologie	19
2.7.1	Technologie	19
2.7.2	Méthodologie	20
2.7.3	Cheat codes	20
2.8	Économie	21
2.9	Sources	21
3	Rappel de la Première Soutenance	22
3.1	Résumé	22
3.2	Le site Web	23
3.3	Le Personnage	23
3.4	Les Ennemis	25
3.5	Les pièges	26
3.6	Les objets	27

3.7	L'Interface	29
3.8	Level Design	30
3.9	Le Multijoueur	30
3.10	Les Textures	31
3.11	Les Sons	31
4	Rappel de la Deuxième soutenance	32
4.1	Résumé	32
4.2	Le site Web	34
4.3	Les Ennemis	34
4.4	Les pièges	36
4.5	Les objets	37
4.6	L'Interface	41
4.7	Le Multijoueur	42
4.8	Les Sons	42
5	Troisième et dernière soutenance	42
5.1	Les cheat code.	42
5.2	Amélioration de certains niveau	43
5.3	Ennemis	44
5.4	Multijoueur	44
5.5	Interface	44
5.6	Site Web	44
5.7	Les sons	45
6	Revue du projet	45
6.1	Nos peines et nos joies	45
6.2	Avis personnels	45
6.3	Conclusion du Projet	46

“Vous avez trouvé une carte menant au trésor de la Buse, se trouvant près de la Réunion. Vous et votre équipage décidez de partir à sa recherche mais votre quête ne sera pas sans embûches. Des créatures étranges abritent cette île et vous empêcheront d’atteindre le trésor tant convoité.”

Pour notre premier projet nous allons réaliser un jeu 2D de réflexion et d’aventure qui pourra se jouer seul ou avec des amis, allant jusqu’à 4 joueurs dans une même session. Le jeu serait jouable en clavier/souris et le joueur devra affronter des ennemis, faire du parkour et résoudre des énigmes pour progresser dans l’aventure

Concernant l’univers dans lequel évoluera le joueur, nous avons décidé qu’il incarnera un pirate à la recherche d’un trésor. Il s’agit du trésor du célèbre La Buse, un pirate français du XVIII^e siècle. Pour ce faire il devra triompher de plusieurs niveaux pour atteindre le niveau final, au centre de l’île, où se trouve le trésor.

La particularité de notre jeu est que toute mort est définitive („Permadeath”). C’est-à-dire que dès lors qu’un joueur meurt ses coéquipiers devront continuer l’aventure sans lui. C’est pourquoi le joueur devra être très vigilant quant à ses actions dans le jeu.

Pour réaliser ce projet nous l’avons découpé en plusieurs tâches : La création d’un monde, les déplacements du joueur et son interaction avec le monde, l’intelligence artificielle des ennemis et le multijoueur. Nous les décrirons plus précisément dans la suite du cahier des charges.

Enfin pour être sûr d’atteindre nos objectifs nous avons réalisé un planning d’avancement montrant quelles tâches devront être finis pour la première soutenance, ainsi qu’une prévision plus large en vue de soutenances 2 et 3.

1 Présentation de l’équipe

Quentin SEFRIN (Chef de projet)

Quand j’étais petit, j’étais très intéressé par les sujets scientifiques. En grandissant, cela n’a pas changé, même si je m’intéresse plus à l’informatique désormais. Pour ce

projet, je n'avais pas d'idée précise. Pour moi, son intérêt principal est de me créer une première expérience enrichissante en matière de projet informatique. Cependant, l'idée proposée par les autres membres du groupe m'a bien plus, et j'ai accroché au projet. Avec ce jeu, j'espère développer des compétences de travail en groupe, ainsi qu'apprendre à gérer plus efficacement mon temps (alterner les cours et les séances de projets). Cela me permettra aussi de mieux connaître mes capacités, par exemple le temps qu'il me faut pour réaliser telle ou telle tâche.

Nicolas SCHMITT

Le domaine de l'ingénierie est une de mes passions depuis mon plus jeune âge. Mais, ce n'est que récemment que je me suis rendu compte que l'informatique m'intéressait tout autant. Ce projet est pour moi l'occasion de mêler deux de mes passions, à savoir la programmation et les jeux vidéo. Mais c'est aussi l'occasion pour moi de prouver ma valeur en tant que membre d'un groupe. J'entends par là ma capacité à travailler en groupe et à réaliser un projet sur le long terme ce qui nécessite une certaine rigueur et un certain sérieux que j'espère avoir, ou du moins développer.

Nicolas WITTWE

Passionné par les jeux vidéo depuis longtemps, j'ai voulu en savoir plus sur les coulisses de ce domaine. Mais j'ai eu peu d'occasion d'en découvrir plus sur l'informatique avant la terminale. C'est grâce à la spécialité ISN que j'ai donc découvert plus en détail l'informatique dans laquelle nous avons dû réaliser un projet et j'ai tout de suite accroché. J'ai donc décidé de continuer l'informatique dans mes études supérieures et j'ai choisi l'EPITA. J'en ai déjà beaucoup appris depuis le début de l'année et je pense que ce projet a encore plein de choses à m'apprendre. Ce projet va donc me permettre d'apprendre plusieurs capacités en alliant deux de mes passions : les jeux vidéo et l'informatique.

Joan ZASEMPA

Je suis très heureux de faire partie de ce projet ambitieux. Depuis quelques années l'informatique me passionne et plus particulièrement le jeu vidéo durant mes loisirs. Cette passion a évolué avec mon entrée à EPITA. C'est pour moi le premier projet aussi concret avec ma première programmation, et ceci en groupe pour pouvoir partager les succès comme les échecs auxquels nous allons devoir faire face. Travailler en groupe est un réel défi, nous allons pouvoir mettre en commun nos qualités et apprendre à travailler avec les défauts de chacun pour en tirer le meilleur parti. Je pense que notre

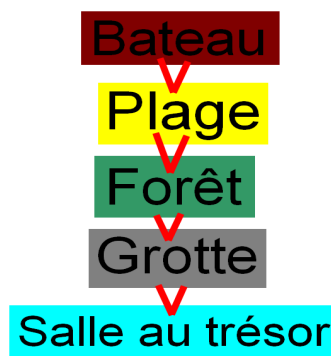
équipe va réussir à mener à bien ce beau projet en mettant en valeur nos acquis mais aussi en cherchant d'autres fondamentaux.

2 Rappel du cahier des charges modifié

2.1 Présentation du projet

2.1.1 Le monde

L'île sur laquelle s'aventure le joueur est composée d'un lobby et de 3 biomes (bateau, plage, jungle, grotte), ainsi que d'une salle au trésor. Le joueur devra alors quitter son navire et traverser l'île pour rejoindre le trésor. Cependant, des énigmes bloqueront l'accès à la suite du jeu, et il devra les résoudre. Chaque biome sera composé de 3 énigmes importantes et de plusieurs petites énigmes bonus. Pour une partie en multijoueur, l'équipe ne pourra pas se retrouver séparée dans différents niveaux, représentés par un biome différent, et il sera impossible de revenir dans un niveau déjà terminé. Durant tout le jeu, les éléments tels que les joueurs, les ennemis, etc, seront soumis au moteur physique 2D de Unity. Afin de réaliser certaines énigmes, le joueur devra aller dans des zones plus petites. Cela nous permet de placer des indices à retenir. Ces zones feront parties du niveau mais ne seront accessibles que via des portails. De plus, le joueur devra forcément aller au bout de la zone pour en sortir, le point d'arrivée ne permettant pas de sortir.



2.1.2 Le personnage

Vous incarnez un personnage qui peut réaliser diverses actions dans le jeu et qui possède des caractéristiques.

Caractéristiques :

- Marqueur indiquant le personnage qu'on joue
- Points de vie : 3 coeurs
- Points d'attaque corps à corps : 2
- Points d'attaque corps à distance : 1

Les action :

- Se déplacer : le joueur se déplace vers la gauche ou vers la droite en utilisant respectivement les touches Q et D.
- Sauter : le joueur saute avec la touche Espace. Il ne peut pas sauter s'il est déjà en l'air.
- S'accroupir : le joueur peut s'accroupir avec la touche C pour réduire sa taille et ainsi esquiver des projectiles ou passer dans un passage étroit.
- Attaquer au corps-à-corps : le joueur peut donner un coup de sabre pour tuer les ennemis avec le clic souris gauche. Un cooldown est mis en place pour ne pas rendre le jeu trop facile.
- Attaquer à distance : le joueur utilise un pistolet pour tirer dans la direction où il regarde (gauche ou droite) avec le clic souris droit. Un cooldown est également mis en place. L'attaque à distance est plus lente que l'attaque au corps à corps.
- Interagir avec le monde : le joueur peut interagir avec des objets du monde (coffres, échelles, portails, ...) avec la touche E.
- Les consommables : il ne s'agit pas d'une interaction directe sur le monde, mais cela permet d'agir sur la vie du personnage. Plus de précisions à la subsection Consommables.

2.1.3 L'environnement

Le joueur devra, durant son aventure, faire face à un environnement varié.

2.1.4 Les ennemis

Tout d'abord, il y a des ennemis avec différentes forces et faiblesses, que le joueur doit vaincre ou tromper pour atteindre son but, le trésor.

Les ennemis possèdent un nom, des points de vie, des points d'attaques, une vitesse d'attaque et de déplacement, une portée ainsi qu'un champ de vision.

- Ennemi 1 :
 - Nom : Guerrier squelette
 - Points de vie : 4
 - Points d'attaque : 1 coeur
 - Vitesse d'attaque : lent
 - Comportement : Cet ennemi patrouille entre des points prédéfinis. Après être arrivé à un point, il se dirige ensuite au point suivant. Quand un joueur est à sa portée, il lance une attaque au corps-à-corps.
 - Caractéristiques spéciales :
 - Points représentant son parcours de patrouille. Il y a 2 points.

- Ennemi 2 :
 - Nom : Archer squelette (pas implémenté)
- Ennemi 3 :
 - Nom : Chauve-souris
 - Points de vie : 1
 - Points d'attaque : 1 demi - coeur
 - Moyen de déplacement : vole
 - Vitesse d'attaque : rapide
 - Comportement : cet ennemi patrouille entre plusieurs points. Après être arrivé à un point, il attend quelques secondes et se dirige ensuite au point suivant. Il ne poursuit pas le joueur, mais lui inflige des dégâts s'il entre dans sa portée.
 - Caractéristiques spéciales :
 - Points représentant son parcours de patrouille. Il peut y avoir plus de 2 points.

- Ennemi 4 :
 - Nom : Loup
 - Points de vie : 2
 - Points d'attaque : 1
 - Vitesse de déplacement : spécifique
 - Vitesse d'attaque : rapide
 - Comportement : Cet ennemi se repose à un point prédéfini. Quand un joueur entre dans son champ de vision, il le poursuit. Pendant la poursuite, le loup court, et quand il retourne à son point de repos, il marche.
 - Caractéristiques spéciales :

- Point de repos : il s'agit d'un point où le mob attend. Pour le loup, il s'assoit et attend.
 - Distance de poursuite : le mob ne peut pas s'éloigner de son point de repos d'une distance supérieure à celle-ci.
-
- Ennemi 5 :
 - Nom : Golem
 - Points de vie : 10
 - Points d'attaque : 21 coeur
 - Vitesse d'attaque : lent
 - Animations :
 - Comportement : Cet ennemi est gros et puissant, mais lent. Il attend à une position précise et poursuit le joueur dès qu'il entre dans son champ de vision et l'attaque quand il est à sa portée. Cependant, il ne laisse pas la possibilité au joueur de passer par la zone qu'il garde.
 - Caractéristiques spéciales :
 - Point de repos : il s'agit d'un point où le mob attend
 - Distance de poursuite : le mob ne peut pas s'éloigner de son point de repos d'une distance supérieure à celle-ci.



2.1.5 Les objets

Les objets sont placés à travers les niveaux et permettent de rendre le jeu intéressant en ajoutant de la difficulté, de l'aléatoire et un moyen de 'communiquer' plus facilement les énigmes du monde.

- Échelle : un objet qui permet au joueur de monter ou descendre quand il interagit avec. On utilise une texture d'échelle ou de corde.
- Coffre : un objet fermé qui ne s'ouvre qu'une seule fois : quand le joueur interagit avec. Quand il s'ouvre, le joueur reçoit un consommable aléatoire.
- Totem coffre : un objet qui ne s'ouvre qu'une seule fois et qui donne un totem de résurrection au joueur. Il n'y en a qu'un seul dans chaque zone, souvent bien caché.
- Levier : cet objet permet au joueur de déclencher une action. Certains leviers avec une texture différente ne seront actifs que durant un certains temps après leur activation.
- Digicode : un affichage où le joueur doit rentrer n chiffres dans un certain ordre, et une énigme permettra de trouver le code. Chaque chiffre rentré dans le digicode sera gardé en mémoire, ainsi le joueur peut rentrer le code en plusieurs fois.
- Cible : un objet qui agit comme un levier, cependant le joueur ne peut pas l'actionner. Pour cela, il faut qu'un objet 'balle' tirée entre en contact avec la cible.
- Bloc mobile : il s'agit d'un élément agissant comme un mur, mais que le joueur peut pousser.
- Torche : cet objet ne sert qu'aux énigmes. Il possède 2 états : allumé ou éteint. Le joueur devra allumer et/ou éteindre les torches d'une énigme en fonction de celle-ci.
- Panneau : il s'agit d'un cadre d'affichage dans lequel nous mettrons des informations utiles pour résoudre les énigmes. Par exemple, s'il faut retenir une série de symboles, ceux-ci seront affichés via ce panneau.
- Porte : une porte bloque l'accès à certaines zones du niveau. Elle peut être ouverte ou fermée, et peut être activée par un levier ; une cible, etc...
- Portail : permet de se retrouver dans une zone fermée plus petite. Il agit comme un téléporteur emmenant le joueur dans une zone inaccessible autrement.
- Flèche : il s'agit du projectile tiré par les ennemis. Elle avance tout droit dans la direction où elle a été tirée et disparaît après avoir rencontré un objet solide. Cet objet est implémenté mais n'est pas utilisé car l'archer squelette n'a pas été implémenté.

- Balle : il s'agit du projectile tiré par le joueur. Elle avance tout droit dans la direction où elle a été tirée et disparaît après avoir rencontré un objet solide, elle peut activer les cibles et faire des dégâts aux ennemis.

2.1.6 Les pièges

Chaque niveau est rempli de pièges en plus des ennemis. Certains inflige des dégâts au joueur, tandis que d'autres le gêne simplement.

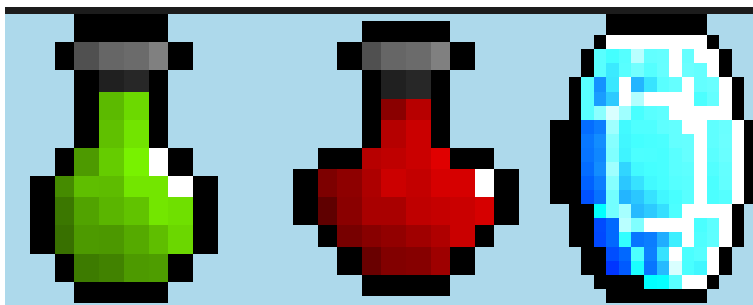
- Zone mortelle : il s'agit d'endroits où le joueur meurt instantanément. Dans la forêt, c'est de l'eau (la majorité des pirates ne savait pas nager), tandis que dans les grottes il s'agit de lave.
- Pics : un bloc qui cause 1 demi-cœur de dégât au joueur. Nous avons gardé un seul skin de pic.
- Objet qui tombe : un bloc qui cause 1 demi-cœur de dégât au joueur. Il se trouve en hauteur et tombe quand un joueur passe en dessous. Sur la plage, il s'agit de noix de coco, tandis que dans les grottes, il s'agit d'un rocher.
- Zone ralentissante : une tuile de sol qui ralentit légèrement le joueur et qui l'empêche de sauter. Sur la plage, il s'agit de sables mouvants.

2.1.7 Les consommables

Afin de rendre le jeu plus intéressant, nous avons introduit des consommables. il s'agit d'objets récupérables dans des coffres de manière aléatoire ou prédéfini. Certains coffres spécifiques auront en effet un loot prédéfini, tandis que d'autres lâcheront un objet aléatoire.

- Potion inférieure : soigne un demi-cœur de vie au joueur qui l'utilise (raccourci clavier : I)
- Potion supérieure : soigne 2 demi-cœur de vie au joueur qui l'utilise (raccourci clavier : O)
- Totem de résurrection : quand un joueur meurt, s'il possède un totem de résurrection il est ressuscité.

Les objets possédés par le joueur seront représentés par des compteurs, un pour chaque type d'objet. L'affichage de l'inventaire est précisé dans l'onglet Interface.



2.1.8 Les énigmes

Afin de progresser dans les niveaux, le joueur devra résoudre des énigmes. Ces énigmes sont variées : réflexion, mémorisation, jump / parkour, ... Nous utiliserons les 'panneaux' pour afficher le contenu des énigmes nécessitant une mémorisation. Certaines énigmes sont des bonus qui donnent au joueur des consommables supplémentaires, cependant, fournir une mauvaise réponse à certaines énigmes infligera des dégâts au joueur. Toutes les énigmes n'ont pas encore été créées, mais nous avons déjà quelques exemples.

Voici une liste des différentes énigmes implémentées :

Enigme grotte 1 : le joueur arrive face à des piliers au dessus de la lave. Il doit sauter sur les piliers pour ne pas tomber dans la lave. Mais, sur un panneau, il y a une chaîne de caractères composée de '-' et de 'X', il y a autant de caractères que de piliers, chaque caractère représente un pilier. Les 'X' représentent les faux piliers et les '-' représentent les vrais piliers. Quand le joueur saute sur un faux pilier il le traverse et tombe dans la lave. Le joueur doit retenir lesquels sont des faux pour ainsi compléter le parcours.

Enigme grotte 2 : le joueur est face à une porte fermée par un digicode à n chiffres. À côté se trouve n portails de couleurs différentes. Le joueur doit rentrer dans chacun d'eux pour obtenir un chiffre, et ensuite remettre les chiffres dans l'ordre à l'aide d'un code couleur. Le digicode sera prédéfini pendant la conception du niveau.

Enigme Forêt 1 : le joueur doit résoudre un système (mathématiques) avec des variables représentées par des fruits. la valeur de chaque fruit lui permet ensuite d'entrer un code dans un digicode qui ouvre une porte. Le joueur débloquent ainsi un objet mobile (tonneau), en le mettant sur une zone de non saut il pourra accéder à la suite du niveau.

Enigme Foret 2 : Un cible est caché dans le niveau, si le joueur tire dessus, une porte s'ouvre donnant accès à un coffre.

Enigme Foret 3 : Il y a deux arbres, un avec des leviers, un autre avec des portes, les leviers du premieres arbres permettent d'ouvrir les portes du second arbre. Le joueur doit donc comprendre quelle doit etre ouverte et faire des allez - retours entre les 2 arbres pour comprendre quelle levier ouvre quelle porte.

Enigme salle au trésor 1 : Parcour avec un timer.

Enigme salle au trésor 2 : Le joueur doit résoudre un système mathématique qui permet d'ouvrir une porte grace à une combinaison rentré dans un digicode.

Enigme salle au trésor 3 : Plusieurs torches sont cachés dans le niveau, le joueur doit toutes les allumer pour ouvrir une porte.

Enigme salle au trésor finale : Il y a 5 activités dans le niveau qui permettent d'ouvrir 5 des 6 portes devant la salle du coffre. Chacune de ces activités est associée à un chiffre. Pour ouvrir la sixieme porte, il faut rentrer ces chiffres en fonction de l'ordre d'ouverture des portes en fonctions de l'activité à laquelle elle correspond.

2.1.9 L'interface

L'interface est un point important dans la conception d'un jeu. Nous allons mettre en place un système de HUD qui affichera la vie et les objets du joueur. En multijoueur, le HUD affichera également la vie des autres joueurs, ainsi qu'un consommable supplémentaire.

Les sauvegardes

Les sauvegardes ne sont pas manuelles, mais automatiques. Au début de chaque niveau, le jeu sauvegarde les données dans un fichier. Les éléments sauvegardés sont

donc :

- Le mode de jeu (solo ou multijoueur)
- La vie du joueur
- Le nombre d'objets possédés
- La position du joueur (le niveau et la position dans celui-ci)

2.2 Multijoueur

Nous gérons le multijoueur grâce à Photon, qui permet gratuitement de connecter jusqu'à 20 joueurs en même temps.

Le jeu permet d'intégrer une équipe de 4 joueurs. Un joueur crée la partie et lui donne un nom. Les autres joueurs peuvent rejoindre la partie en rentrant ce nom. Chaque joueur dispose d'un écran propre afin de faciliter les déplacements dans un même niveau. En revanche, tous les joueurs doivent être dans le même niveau, même s'ils peuvent se trouver dans différentes zones accessibles via portail. Pour changer de niveaux, tous les joueurs vivants doivent être présents dans une petite zone à la fin de celui-ci. Quand un joueur est mort, sa caméra se fixe sur celle de l'un des autres joueurs encore en vie.

Les sauvegardes en multijoueurs se feront de la même manière qu'en solo, le fichier est sauvegardé sur la machine du joueur qui a lancé la partie. Quand il choisit le mode multijoueur, le joueur peut sélectionner 'lancer une partie' ou 'rejoindre une partie'. Le joueur peut rejoindre n'importe quelle partie tant qu'il a le nom de la salle. S'il veut lancer une partie, il peut soit lancer une nouvelle partie, soit lancer une partie à partir d'une sauvegarde existante sur son PC. Les sauvegardes indiqueront si le jeu a été lancé en multijoueur ou en solo, est-il ne sera pas possible de charger une partie d'un mode vers l'autre. Dans les sauvegardes, les personnages non utilisés (s'il n'y a que 2 joueurs par exemple), seront sauvegardés comme des joueurs ayant toute leur vie.

2.3 I.A

Notre jeu doit comporter une certaine difficulté, il est donc impensable de ne pas avoir d'intelligence artificielle ayant pour objectif de vaincre le joueur.

Chaque ennemi aura sa propre I.A, basée sur ses caractéristiques. Par exemple, le guerrier squelette patrouille et attaque le joueur, tandis que la chauve-souris se déplace sans lui prêter attention.

2.4 Objet de l'étude

Ce projet a un but éducatif ainsi que le développement pour notre passion pour l'informatique, en appliquant tout ce que l'on a appris, mais aussi en faisant des recherches

afin d'approfondir nos connaissances.

Tout d'abord il permet de découvrir ou d'approfondir la notion de projet qui est une méthode de travail très utilisé dans le monde du travail et en particulier dans l'informatique. Ce projet permet aussi de renforcer nos compétences :

- En programmation, notamment en C#, car ce jeu sera développé en C#. Il y aura aussi un peu d'HTML et de CSS pour le site web, ainsi que des bases en LaTeX.
- En communication. En effet le travail de groupe est un des piliers de ce projet. Il faut donc savoir communiquer avec les autres. De plus, lors des soutenances il faut être capable de présenter notre projet et de le vendre, d'une certaine façon.
- En organisation (cahier des charges, attribution des tâches, respect d'une deadline)

2.5 Etat de l'art

2.5.1 Histoire

Aujourd'hui le jeu vidéo est très populaire chez les jeunes mais il reste encore mal vu par beaucoup d'adultes car pour eux le jeu vidéo provoque une addiction et rend les jeunes asociaux. Mais c'est plus que ça. Le jeu vidéo nous permet de nous évader dans un autre monde. C'est comme un film dans lequel nous sommes l'acteur grâce à notre clavier ou notre manette. Le jeu vidéo peut être éducatif ou nécessiter de la réflexion. Il peut aussi faire appel à des capacités de coopération.

Tout d'abord, voici un petit peu d'histoire sur les jeux vidéos en général. L'histoire du jeu vidéo débute dans les années 1950, où l'idée du jeu vidéo naît au sein des universités lors de recherches sur l'informatique. Les jeux vidéos ne se font connaître du grand public qu'à partir des années 1970 avec la commercialisation des premières bornes d'arcade ainsi que de consoles de jeu vidéo (portables et ensuite de salon). Les premiers jeux vidéos étaient très rudimentaires et avaient peu de mécaniques. Par exemple *Pong*, qui est le premier jeu vidéo commercialisé en 1972, était très simple. Il s'agissait de bouger une barre qui faisait office de raquette, et de renvoyer une balle à notre adversaire.

Le jeu vidéo s'est ensuite développé avec de nouveaux types de jeux et de mécaniques. C'est surtout à l'arrivée de la 3D qu'il a commencé à devenir très prisé. Le jeu vidéo a continué à se populariser au fil des années, jusqu'à aujourd'hui, où il est très populaire surtout chez les jeunes.

Notre jeu vidéo est un mélange de deux types de jeu vidéo : les jeux d'aventure et les jeux de réflexion. Voici un bref résumé de leur histoire :

Le jeu vidéo de réflexion, jeu de puzzle ou puzzle game, est un genre de jeu vidéo

centré sur la réflexion. Le jeu de réflexion amène le joueur à résoudre des casse-têtes, des énigmes, ou à naviguer à travers des lieux complexes comme des labyrinthes. Il fait appel chez les joueurs à des aptitudes comme la logique, la reconnaissance de formes ou encore l'habileté de construction visuo-spatiale. Plus généralement, il oblige le joueur à réfléchir. Les premiers jeux de réflexion sont apparus dans les années 1970 : il s'agissait le plus souvent d'adaptations de jeux de réflexion traditionnels (exemples : Les jeux de *Nim*, le *Pendu* ou le encore *Mastermind*). Le genre s'est développé dans les années 1980, avec l'apparition de jeux d'arcade mêlant réflexion et habileté. Sorti en 1985, *Tetris* est sans aucun doute le plus connu des jeux de réflexion. C'est d'ailleurs lui qui a popularisé le jeu de puzzle.

Le jeu d'aventure est un genre de jeu vidéo dont l'intérêt prédominant se focalise sur la narration plutôt que sur les réflexes et l'action. Plus précisément, les jeux d'aventure mettent le plus souvent l'accent sur l'exploration, les dialogues, la résolution d'énigmes (celles-ci peu présentes dans la plupart et n'obligeant pas le joueur à réfléchir autant que dans un jeu de réflexion). Les premiers jeux d'aventure étaient textuels, c'est à dire qu'il n'y avait pas de graphisme. C'était des jeux où l'on pouvait entrer des commandes influençant sur le reste du jeu, tout en gardant un affichage en texte.

Ensuite ils se sont améliorés avec l'apparition de graphismes. Aujourd'hui, le jeu d'aventure est l'un des types de jeux les plus joués, notamment des titres comme *Red Dead Redemption 2*, ou encore la saga *Assassin's Creed*, qui ont eu un grand succès.

2.5.2 Exemples

Portal est un jeu d'action/réflexion jouable en coopération et dont la principale particularité est que les énigmes se résolvent grâce à des portails.

Valiant Hearts : The Great War est un jeu développé par Ubisoft Montpellier. C'est un jeu d'aventure et de réflexion en 2D se déroulant pendant la 1ère guerre mondiale. Il possède une narration et des énigmes intéressantes. Ses graphismes sont des images de style BD .

Thimbleweed Park est un jeu d'aventure en 2D. Sa particularité est que c'est un point & click. C'est-à-dire que c'est un jeu qui se joue principalement à la souris.

2.6 Tâches

2.6.1 Planification des tâches

Tâche	Responsable	Planification
Joueur mouvement	Nicolas Schmitt	1°S : fini
Joueur autres actions	Nicolas Schmitt	1°S : fini
Joueur sons	Nicolas Schmitt	1°S : fini
Ennemi 1 implémentation	Nicolas Schmitt	1°S : fini
Ennemi 1 mouvement	Nicolas Schmitt	1°S : fini
Ennemi 1 sons	Joan Zasempa	1°S : fini
Ennemi 1 I.A	Quentin Sefrin	1°S : fini
Ennemi 2 implémentation	Joan Zasempa	1°S : fini
Ennemi 2 mouvement	Joan Zasempa	1°S : fini
Ennemi 2 sons	Joan Zasempa	1°S : fini
Ennemi 2 I.A	Quentin Sefrin	2°S : fini
Ennemi 3 implémentation	Quentin Sefrin	2°S : fini
Ennemi 3 mouvement	Quentin Sefrin	2°S : fini
Ennemi 3 sons	Joan Zasempa	2°S : fini
Ennemi 3 I.A	Quentin Sefrin	2°S : fini
Ennemi 4 implémentation	Quentin Sefrin	2°S : fini
Ennemi 4 mouvement	Quentin Sefrin	2°S : fini
Ennemi 4 sons	Joan Zasempa	2°S : fini
Ennemi 4 I.A	Quentin Sefrin	2°S : fini
Ennemi 5 implémentation	Joan Zasempa	2°S : fini
Ennemi 5 mouvement	Joan Zasempa	2°S : fini
Ennemi 5 sons	Joan Zasempa	2°S : fini
Ennemi 5 I.A	Quentin Sefrin	2°S : fini

Tâche	Responsable	Planification
Création niveau test	Nicolas Schmitt	1°S : fini
Level design 'Lobby'	Nicolas Schmitt	2°S : fini
Level design Plage	Joan Zasempa	2°S : fini
Level design Forêt	Quentin Sefrin	2°S : fini
Level design Grotte	Nicolas Schmitt	2°S : fini
Level design Salle finale	Nicolas Wittwe	2°S : fini
Piège Oneshot	Nicolas Wittwe	1°S : fini
Piques vers le haut	Nicolas Wittwe	2°S : fini
Piques vers le bas	Nicolas Wittwe	2°S : fini
Ralentissement	Nicolas Wittwe	2°S : fini
HUD vie	Nicolas Wittwe	1°S : fini
HUD objets	Nicolas Wittwe	1°S : fini
Menu principal	Quentin Sefrin	1°S : fini
Menu solo	Nicolas Wittwe	1°S : fini
Menu multi	Quentin Sefrin	2°S : fini
Site Web	Nicolas Wittwe	1°S : fini puis màj
Échelle	Nicolas Wittwe	2°S : fini
Coffre	Nicolas Wittwe	1°S : fini
Levier	Nicolas Wittwe	1°S : modèle sans timer fini
Cible	Nicolas Wittwe	2°S : fini
Digicode	Nicolas Wittwe	2°S : fini
Bloc mobile	Nicolas Wittwe	2°S : fini
Flèche	Nicolas Schmitt	1°S : fini
Balle	Nicolas Schmitt	1°S : fini
Torche	Nicolas Wittwe	2°S : fini
Panneau	Nicolas Schmitt	2°S : fini
Porte	Nicolas Wittwe	1°S : fini
Portail	Nicolas Schmitt	2°S : fini
Consommable 1	Joan Zasempa	1°S : fini
Consommable 2	Joan Zasempa	1°S : fini
Consommable 3	Joan Zasempa	1°S : fini

Tâche	Responsable	Planification
Musique Lobby	Joan Zasempa	2°S : fini
Musique Plage	Joan Zasempa	2°S : fini
Musique Forêt	Quentin Sefrin	2°S : fini
Musique Grotte	Joan Zasempa	2°S : fini
Musique Fin	Joan Zasempa	2°S : fini
Bruits de l'environnement Lobby	Joan Zasempa	2°S : fini
Bruits de l'environnement Plage	Joan Zasempa	2°S : fini
Bruits de l'environnement Forêt	Quentin Sefrin	2°S : fini
Bruits de l'environnement Grotte	Joan Zasempa	2°S : fini
Bruits de l'environnement Fin	Joan Zasempa	2°S : fini
Sons objets	Joan Zasempa	3°S : fini
Sauvegards	Quentin Sefrin	2°S : fini
Multijoueur	Quentin Sefrin	1°S : implémenté

1° Soutenance : pour la première soutenance, nous aurons implémenté le multijoueur ainsi qu'un niveau test. Nous aurons également ajouté 3 ennemis, dont 1 avec son I.A, ainsi que le joueur et quelques objets / consommables. Certains sons seront déjà dans le jeu, afin de vérifier le fonctionnement de Unity pour les sons. En parallèle, nous aurons chercher des textures et des sons pour les autres niveaux.

2° Soutenance : un niveau de test pour chaque niveau permettra de tester les énigmes après la 1° Soutenance. Tous les niveaux du jeu seront implémentés en grande partie. Les sons et les textures seront tous ajoutés au jeu, de même que les ennemis, les objets et les pièges. De nombreuses énigmes auront été créées et implémentées dans le niveau test. Le système de sauvegarde sera opérationnel.

3° Soutenance : le jeu sera téléchargeable via le site web. Nous aurons tester les niveaux afin de régler la difficulté, en modifiant des caractéristiques du joueur ou des ennemis. Nous aurons réaliser un trailer, et nous ajouterons des énigmes et des textures supplémentaires en fonction du temps restant.

2.6.2 Description des tâches

1. Site Web :
 - Créer l'accueil du site web
 - Mettre à jour régulièrement
 - Ajouter l'option de téléchargement du jeu
2. Personnage :
 - Implémenter le personnage avec toutes ses caractéristiques
 - Implémenter la gestion en multijoueur (couleur, ...)

- Implémenter toutes les fonctions du personnages
- 3. Level design :
 - Réfléchir aux textures nécessaires à la mise en place du niveau
 - Avec ces textures, les ennemis et les objets, imaginer le niveau petit à petit
 - Avant d'implémenter une énigme dans le niveau, l'essayer dans le niveau test
 - vérifier la faisabilité du niveau
- 4. Création des objets / pièges :
 - Définir les propriétés utiles de l'objet (son état, une position, ..) s'il y en a
 - Implémenter l'objet / piège avec les fonctions qui permettront l'update en jeu
- 5. Ennemis :
 - Créer les mouvements de l'ennemi
 - Créer ses fonctions (attaque, ...)
 - implémenter son I.A
- 6. Multijoueur :
 - Implémenter le multijoueur
- 7. Interfaces :
 - Implémenter un menu permettant de quitter le jeu
 - implémenter le menu avant le lancement d'une partie
 - ajouter le HUD
- 8. Audios :
 - Trouver les sons liés aux biomes, musique et ambiance
 - Trouver les sons du joueur et des ennemis
 - Trouver les sons des objets et des pièges

2.6.3 Bonus

Il s'agit de notre premier projet, notre capacité à planifier nos réalisations dans le temps est plutôt limitée. Nous avons donc décider de prévoir ici quelques fonctionnalités qui pourront être rajoutées si notre planification des tâches était vraiment mal pensée (dans le bon sens) :

Aucun bonus n'as été implémenté du à un retard sur certaines tâches.

2.7 Technologie et méthodologie

2.7.1 Technologie

Tout d'abord, afin de réaliser ce cahier des charges ainsi que les futurs comptes-rendus, nous utilisons le logiciel TeXworks, qui utilise le langage LaTeX sous Windows.

Nous devons créer et héberger un site Web, nous avons donc choisi d'utiliser Netlify. Il s'agit d'un service d'hébergement en ligne en partie gratuit et lié à Git, ce qui permet une mise à jour directement après le push du site. Pour un site comme celui que nous allons faire, l'offre gratuite est suffisante.

Pour la réalisation du jeu en lui-même, nous utiliserons Unity, qui est un outil adapté à la création d'un jeu 2D.

Nous avons eu besoins de textures pour notre monde, nos décorations, nos personnages, ... Nous avons principalement utilisé des assets obtenus sur des sites tels que l'asset store Unity ou encore Itch.io. Au besoin, pour la création de variation de certaines textures, comme la couleur des joueurs, nous avons utilisé un logiciel de traitement d'image (Gimp).

Pour la réalisation du trailer, nous avons utilisé un logiciel de montage vidéo intitulé DaVinci Resolve 17.

Nous avons utilisé, afin de mettre de l'ambiance dans le jeu, des musiques et des bruits d'ambiance. Nous avons choisis des sons gratuits et évidemment libre d'utilisation.

Dans la même idée, nous avons utilisé un Google Docs afin de lister précisément toutes les fonctions de chacun de nos objets. Ce document sert à communiquer plus rapidement des informations simples, plutôt que de se perdre avec de trop nombreux messages.

Afin de partager plus efficacement les dossiers et fichiers hors projets, tels les textures ou les sons, nous avons utilisé Discord.

2.7.2 Méthodologie

Pour pouvoir réaliser ce projet, nous avons dû faire preuve d'organisation et de bonne communication. En effet, durant le développement de notre jeu vidéo, certaines tâches nécessitent de se mettre d'accord avec les personnes en charge d'autres tâches. Par exemple, afin de réaliser efficacement une I.A fonctionnelle pour nos ennemis, nous devons prévoir à l'avance toutes les fonctions nécessaires à la définition des ennemis.

Les aspects créatifs de notre projet sont aussi liés à plusieurs tâches. Nous devons donc décider ensemble de certains aspects du jeu, comme la pertinence d'une musique dans tel niveau, ...

2.7.3 Cheat codes

Nous avons imaginé des raccourcis clavier permettant de tricher dans le jeu. En voici la liste :

- Alt + F : déverrouiller la porte la plus proche

- Alt + B : passer au niveau suivant
- Alt + M : avancer au niveau final
- Alt + I : rend le joueur insensible aux dégâts
- Alt + J : se réanimer soi-même
- Alt + L : tuer les ennemis dans le niveau
- Alt + G + (& ou é ou ``) : régénère la vie

2.8 Économie

Nous avons utilisé uniquement des logiciels, des images et des sons gratuits, afin de ne pas dépenser d'argent durant la réalisation du projet.

2.9 Sources

Wikipédia - Jeu de réflexion
Wikipédia - Jeu d'aventure

3 Rappel de la Première Soutenance

3.1 Résumé

Pour la première soutenance, nous avons implémenter les mécaniques de bases de notre jeu, à savoir le personnage, les coffres et les pièges.

C'était aussi pour nous l'occasion de découvrir le logiciel Unity, au début nous avons eu beaucoup de problème avec git, notamment des problèmes de conflit sur la scène.

Nous avons aussi implémenté une première version du multijoueur qui n'était pas totalement fonctionnel mais qui nous a permis d'avoir une idée plus claire de sa conception.

Tâche	Responsable	État
Joueur mouvement	Nicolas Schmitt	Terminé
Joueur autres actions	Nicolas Schmitt	Terminé sauf mort
Joueur sons	Joan Zasempa	Presque tous terminés
Ennemi 1 implémentation	Nicolas Schmitt	Terminé
Ennemi 1 mouvement	Nicolas Schmitt	Terminé
Ennemi 1 sons	Joan Zasempa	Pas terminé
Ennemi 1 I.A	Quentin Sefrin	Pas terminé
Ennemi 2	Joan Zasempa	Pas terminé
Création niveau test	Nicolas Schmitt	Terminé
Piège Oneshot	Nicolas Wittwe	Terminé
H.U.D vie	Nicolas Wittwe	Terminé
H.U.D objets	Nicolas Wittwe	Terminé
Menu principal	Quentin Sefrin	Terminé
Menu solo	Nicolas Wittwe	Terminé
Site Web	Nicolas Wittwe	Déployé
Coffre	Nicolas Wittwe	Terminé
Levier	Nicolas Wittwe	Terminé
Flèche	Nicolas Schmitt	Pas terminé
Balle	Nicolas Schmitt	Terminé
Porte	Nicolas Wittwe	Terminé
Consommable 1	Joan Zasempa	Terminé
Consommable 2	Joan Zasempa	Terminé
Consommable 3	Joan Zasempa	Pas de résurrection
Multijoueur	Quentin Sefrin	Implémenté, mais bugué

3.2 Le site Web

Notre site Web était déjà en place. Il est en langage HTML et CSS. Il n'était pas encore complet, mais présente notre projet dans son ensemble. Il est composé de quatre pages. La première présente le jeu aux joueurs intéressés.

Son adresse est <https://ubihard-projet.netlify.app/>



3.3 Le Personnage

Nous avons implémenté le personnage avec certaines de ses fonctionnalités. Il peut, pour l'instant :

- Se déplacer horizontalement avec les touches Q et D. Pour cela, on utilise la méthode `Input.GetButtonDown("Horizontal")`, qui renvoie -1 si Q est pressé, 1 si D est pressé et 0 si les deux sont pressés. Cela permet ensuite, avec une méthode `Move()` créée, de déplacer le personnage en fonction de sa vitesse et de sa direction.
- Sauter à l'aide de la touche ESPACE. Avec la méthode `Input.GetButtonDown("Jump")`

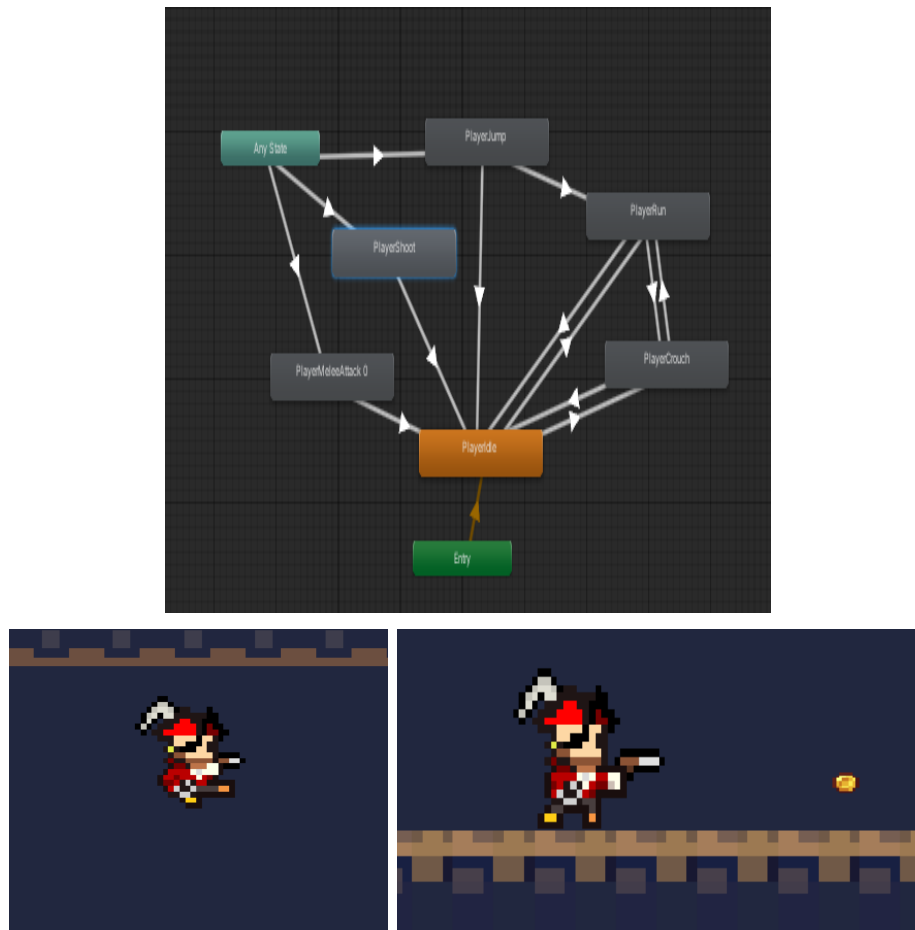
et si le personnage n'est pas au sol, on applique une force positive verticale au personnage. Pour détecter s'il est au sol, on utilise un point placé aux pieds du personnage, et on vérifie s'il est en contact avec le sol.

- S'accroupir avec la touche S, ce qui réduit sa taille et sa vitesse, lui permettant d'esquiver les flèches et de passer dans des endroits étroits. Le personnage possède un circle collider 2D sur sa moitié inférieure et un box collider 2D sur sa moitié supérieure. Quand il s'accroupit, la box collider 2D est désactivée, ce qui réduit la zone de collision du personnage.
- Attaquer au corps à corps avec clic gauche. Un point est défini sur l'arme du personnage. On calcule à l'aide de physics 2D, au moment de l'attaque, si un ennemi est présent dans un rayon autour de ce point. Si un ennemi est présent, la méthode TakeDamage() de l'ennemi est appelée. Un cooldown a été mis en place, afin de laisser l'animation du personnage se dérouler et de ne pas le rendre surpuissant face aux ennemis.
- Attaquer à distance avec clic droit. Un point est défini sur l'arme du personnage, la balle tirée par le joueur part de ce point. l'objet balle est préfabriqué et instancié au moment du tir. La balle avance en ligne droite et quand elle rencontre un obstacle, elle se détruit. Si l'obstacle est une ennemi, la méthode TakeDamage() de l'ennemi est appelée. Un cooldown sera mis en place plus tard, afin de laisser l'animation du personnage se dérouler et de ne pas le rendre surpuissant face aux ennemis. De plus, pour être plus réaliste, un pistolet pirate ne tire pas en rafale.

Pour les timers ou cooldowns de notre projet, nous avons décidé de les incrémenter dans la fonction FixedUpdate() plutôt qu'Update(), afin que la puissance de l'ordinateur n'influence pas ceux-ci. Au niveau des animations du personnage, nous en avons 6 pour l'instant :

- Idle
- Run
- Jump
- Crouch
- MeleeAttack
- Shoot

Quand on lance le jeu, le joueur est d'abord en animation d'Idle, s'il se met à courir, il passe de l'animation d'Idle à l'animation de Run et inversement quand il arrête de courir. Plusieurs paramètres ont été créés dans l'animateur afin de gérer les transitions entre les différentes animations. L'animation Jump peut se lancer depuis n'importe quelle animation si le joueur saute, de même que les animations MeleeAttack et Shoot quand le joueur attaque. Une fois qu'elle est finie, on peut retourner vers l'animation de Run ou d'Idle en fonction de la vitesse du personnage.

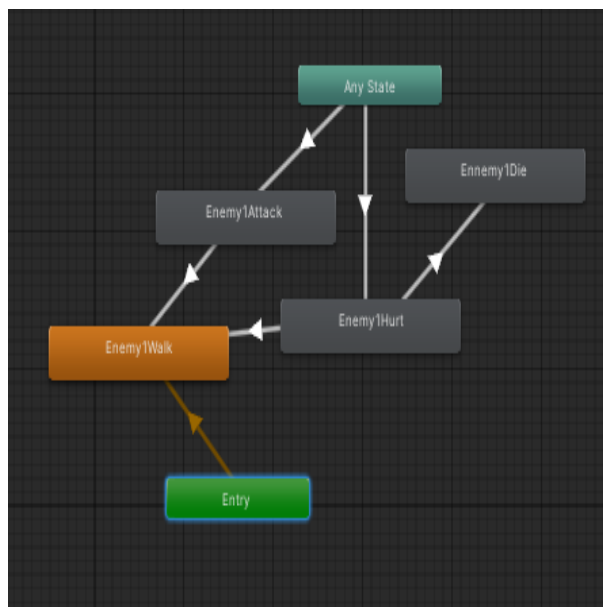
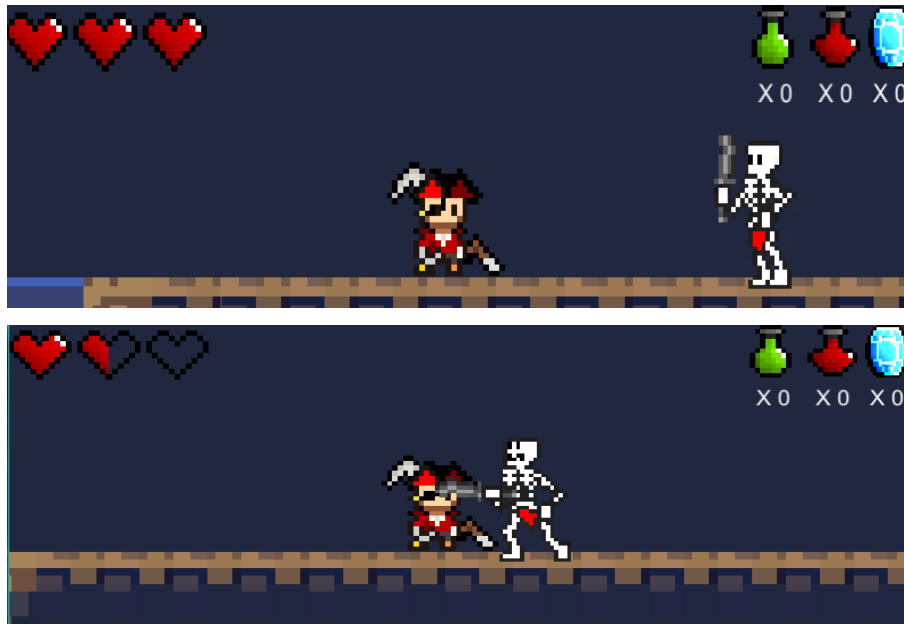


3.4 Les Ennemis

Pour cette première soutenance, nous avons créé un ennemi : le guerrier squelette. Il peut faire des allers-retours entre deux points et attaquer un joueur devant lui. De plus, il peut prendre des dégâts et mourir. L'ennemi possède pour l'instant 3 animations :

- Walk
- Attack
- Hurt
- Die

Quand la vie du squelette est ≤ 0 la fonction `Die()` de ce dernier est appelée, puis il ne reste plus qu'un tas d'os sur le terrain. Chaque fois la fonction `TakeDamage()` du squelette est appelée, l'animation `Hurt` est jouée.

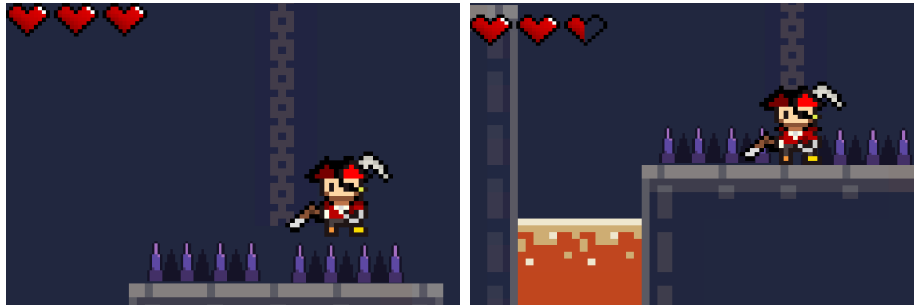


3.5 Les pièges

La création des pièges est à jour par rapport à nos prévisions. Le piège mortel, qui enlève toute la vie du joueur, est en effet implémenté et fonctionnel. Pour cela, si le joueur est en contact avec le piège, il appelle la méthode `TakeDamage(int n)`, qui soustrait

à la vie actuelle du joueur l'entier n en argument. Il prend la forme de lave ou d'eau.

Nous avons implémenté un peu en avance les piques vers le haut. Ceux-ci infligent 1 de dégât soit un demi-cœur à tous les joueurs avec qui ils sont en contact. Ils fonctionnent donc de la même manière que le piège mortel. Cependant, un timer après avoir infligé des dégâts l'empêche de blesser à nouveau le joueur immédiatement. Autrement, il perdrait toute sa vie instantanément.



3.6 Les objets

Nous avons déjà terminer la création de plusieurs objets.

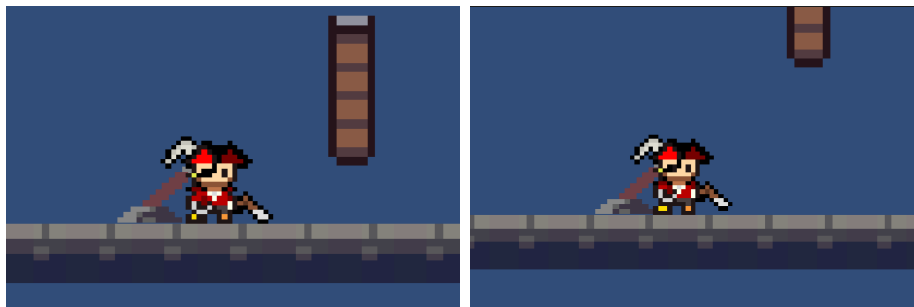
Le premier est le coffre. Nous avons créé un coffre qui s'ouvre quand un joueur à proximité utilise la touche d'interaction. Les coffres donnent au joueur un objet consommable. Au départ, nous avons prévu de ne créer qu'un seul coffre, et de modifier les consommables qu'il donnent en fonction du mode de jeu (solo ou multijoueur). Cependant, après réflexion, nous avons décidé de créer 2 coffres différents : le premier donne au joueur une potion aléatoire tandis que le deuxième lui donne toujours un totem de résurrection. Les coffres fonctionnent ainsi : quand un joueur est en contact avec un coffre fermé (détecté grâce à un box collider 2D), si le joueur interagit avec la touche E, alors le coffre donnera au joueur une potion au hasard. Le deuxième fonctionne de la même manière, mais donnera uniquement un totem de résurrection (il ne sera présent qu'en multijoueur).





Le deuxième est une porte. Pour l'afficher de manière cohérente dans un monde 2D, nous avons choisi une texture ressemblant à un mur. Ainsi, pour son ouverture, nous avons choisi de la faire "glisser" vers le haut ou vers le bas. Pour cela, nous avons défini 2 points : un d'état fermé, l'autre d'état ouvert. Nous appliquons ensuite une force vers ces points en fonctions de l'état de la porte et du levier.

Le troisième est un levier. Il permet d'ouvrir ou de fermer les portes. Nous avons prévu 2 types de levier, cependant un seul a été complètement réalisé pour cette soutenance. En effet, le deuxième type de levier n'est pas encore associé à l'objet porte. Nous avons créé le premier type de levier. Il change de sens quand le joueur interagit avec en étant à proximité. Le sens du levier détermine la direction de la porte, vers un état ouvert ou fermé. Pour l'instant, il est possible de stopper le mouvement d'une porte grâce au levier. Cela ne semble pas être un mécanisme gênant, nous l'avons donc conservé, mais il pourrait être retiré plus tard lors de l'équilibrage du jeu. Nous avons implémenté le deuxième type de levier celui avec un timer. Celui-ci reprend sa position initiale (ferme la porte) au bout d'un certain temps. Ce temps sera variable en fonction de l'endroit où sera placé le levier et sa porte. En effet, ils seront utilisés dans des énigmes où le joueur devra parcourir une certaine distance dans un temps limité.



Les quatrièmes sont les consommables :

- Potion de soin inférieure : elle rend au joueur 1 point de vie. Cela modifie le H.U.D du joueur, au niveau des points de vie et du nombre de consommables

- possédés.
- Potion de soin supérieure : elle agit de la même manière que la potion inférieure, mais rend 2 points de vie.
 - Totem de résurrection : il ramène un joueur mort à la vie. Il n'a pas encore d'effet de résurrection, mais le joueur peut l'obtenir et l'utiliser aussi en solo, ce qui rend toute sa vie au joueur. Le système de résurrection sera implémenté avec le système de mort. A terme, il ne sera pas disponible en mode solo.

3.7 L'Interface

Nous avons déjà créé un menu en partie fonctionnel. Pour l'instant, il ne permet pas de jouer à partir d'une sauvegarde. Il est cependant fonctionnel et permet de démarrer une nouvelle partie, en solo ou en multijoueur. Il reste cependant à modifier afin de le rendre attrayant aux yeux des joueurs.



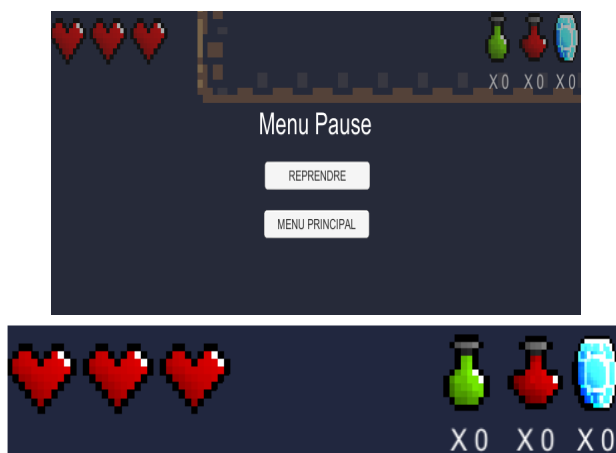
Nous avons implémenté le H.U.D solo. Dans celui-ci se trouvent la vie du joueur et son inventaire. Pour la vie nous avons fait un système de barre de vie avec un skin de cœur. Pour cela nous avons deux images superposées : une avec les cœur vide et une avec les cœurs rouges. Les cœur vide sont tous le temps affiche mais l'image cœur rouge s'affiche en fonction du pourcentage de vie du joueur. Nous avons défini les PV max du joueur à 6 (2 fois le nombre de cœur) pour que nous puissions avoir soit des cœurs pleins ou des demi-cœurs.

Pour l'inventaire nous avons créé 3 images qui représente chacune un objet et sous chacune d'elle un texte qui affiche son nombre. En solo il n y a pas de totem de résurrection mais nous l'avons quand même mis dans le H.U.D (pour faire des tests sur le coffre totem par exemple). Pour cela nous avons une variable entière pour chaque objet qui représente le nombre d'items que le joueur possède. Ces variables sont incrémentées dans leur méthode attitrée et sont converties en chaîne de caractère pour être affichées. Les méthodes sont ensuite utilisées dans d'autre objet qui donne les consommable. Par exemple les coffres font appel à ces méthodes pour incrémenter le nombre d'objet du

joueur. Bien sûr le nombre d'item de chaque objet est initialisé à 0 à chaque début de partie.

Nous avons également implémenté un menu pause solo qui permet au joueur en appuyant sur la touche « escape » d'ouvrir ce menu. Dans ce menu il y a deux boutons : le premier « Reprendre » permet de reprendre la partie. A noter également que quand le menu est ouvert on peut le refermer en appuyant une deuxième fois sur « escape ». Le deuxième « Retour au menu » permet de retourner au menu principal. Ce menu pause n'arrête pas le temps car c'est un jeu multijoueur mais il désactive le joueur pendant qu'il est ouvert.

De plus, nous sommes en avance sur ce point, car nous avons déjà créé un menu pause en partie fonctionnel en multijoueur.



3.8 Level Design

Au début du projet, il est inutile de designer les niveaux du jeu, il nous faut d'abord créer les éléments nécessaires à la création du monde. Cependant, nous avons créé un niveau de test afin de réaliser nos éléments. Il était prévu de ne créer qu'un seul niveau pour nos tests, mais pour les rendre plus faciles, nous avons décidé de créer un niveau pour chacun d'entre nous.

Nous avons tout de même réalisé une scène spéciale pour la présentation de la soutenance.

3.9 Le Multijoueur

Pour cette soutenance, le système multijoueur est assez basique. Il permet de créer une partie ou d'en créer une, sans plus d'options. Pour le moment, seul le déplacement

des joueurs est synchronisé, sans aucune animation. Pour cela, nous avons utilisé le package PUN 2, qui permet de se connecter sur un serveur en multijoueur avec la méthode `ConnectUsingSettings()`. Par la suite, dans le menu du jeu, le joueur décide s'il souhaite rejoindre une partie ou en créer une. Pour le moment, il est seulement possible de rejoindre une partie aléatoirement à l'aide de la méthode `JoinRandomRoom`, mais par la suite, nous permettrons au joueur de voir en avance une liste des différentes parties en cours, et de rejoindre celle qu'il souhaite. De même, pour créer une partie, le joueur n'est pas libre. La partie créée avec la méthode `CreateRoom()` a toujours le même nom et la même capacité de joueurs. Cependant, il sera possible par la suite de choisir le nom de la partie, ainsi que le nombre de joueurs maximum, compris entre 2 et 4. Durant la mise en place du multijoueur, il a fallu faire face à certains problèmes. Premièrement, en plus

3.10 Les Textures

Nous avons déjà commencé à rechercher des textures à utiliser dans notre jeu pendant l'écriture du cahier des charges. Nous ne savons pas encore toutes les textures que nous allons utiliser, mais nous en avons déjà trouvées certaines. Nous avons déjà décidé des textures du personnage, de chaque ennemi, des objets, ainsi que des décors de certains niveaux.

3.11 Les Sons

Nous avons prévu d'implémenter quelques sons pour la première soutenance, cependant, nous n'avons pas pu ajouter tous les sons que nous voulions incorporer pour la première soutenance. Notamment, les sons de saut du joueur et d'utilisation des consommables ont été ajoutés au jeu. Pour cela, nous avons placé dans le script, on initialise une source audio, ainsi qu'un fichier audio. Ensuite, au moment d'effectuer une action, on utilise la méthode `source.AudioOneShot(son)` afin de jouer le son depuis la source. Il a donc fallu ajouter un composant `Audio Source` dans Unity. Toutefois, nous avons déjà trouvé certains sons que nous allons utiliser, notamment la musique d'introduction et les sons des squelettes.

Pour cette première soutenance, nous avons prévues un peu trop de tâches, ainsi nous ne sommes pas à jour sur tous les points dans l'avancement de notre projet. En effet, nous avons sous-estimé les difficultés d'implémentation des ennemis et du multijoueur. Cependant, le but de la première soutenance était d'implémenter les bases de notre jeu, ce qui est le cas, nous avons créé tous les éléments de bases de notre projet. Nous avons donc réussi à prendre en main plusieurs aspects de l'éditeur Unity, et nous serons plus efficaces durant la deuxième période de la création de notre projet.

De plus, malgré quelques retards, nous sommes tout de même en avance sur la création des pièges et des objets, nous sommes donc plutôt confiants quant à la suite du projet.

4 Rappel de la Deuxième soutenance

4.1 Résumé

Pour la deuxième soutenance nous avons implémenté la totalité des pièges ainsi qu'une première version des différents niveaux à savoir le lobby, la plage, le foret, la cave et la salle au trésor.

Nous avons aussi réalisé la plupart des ennemis, même si nous avons du retard.

Tâche	Responsable	État
Joueur sons	Nicolas Schmitt	Terminé
Ennemi 2 implémentation	Joan Zasempa	Terminé, non animé
Ennemi 3 implémentation	Nicolas Schmitt	Terminé
Ennemi 4 implémentation	Quentin Sefrin	Terminé, non animé
Ennemi 5 implémentation	Joan Zasempa	Terminé, non animé
Level design 'Lobby'	Nicolas Schmitt	Commencé
Level design Plage	Joan Zasempa	Commencé
Level design Forêt	Quentin Sefrin	Commencé
Level design Grotte	Nicolas Schmitt	Commencé
Level design Salle finale	Nicolas Wittwe	Terminé
Piques vers le haut	Nicolas Wittwe	Terminé
Piques vers le bas	Nicolas Wittwe	Terminé
Ralentissement	Nicolas Wittwe	Terminé
Menu multi	Quentin Sefrin	Terminé
Site Web	Nicolas Wittwe	Mise à jour
Échelle	Nicolas Wittwe	Terminé
Cible	Nicolas Wittwe	Terminé
Digicode	Nicolas Wittwe	Terminé
Bloc mobile	Nicolas Wittwe	Terminé
Flèche	Nicolas Schmitt	Terminé
Torche	Nicolas Wittwe	Terminé
Panneau	Nicolas Schmitt	Terminé
Portail	Nicolas Schmitt	Terminé
Consommable 3	Joan Zasempa	Terminé
Musique Lobby	Joan Zasempa	Terminé
Musique Plage	Joan Zasempa	Terminé
Musique Forêt	Quentin Sefrin	Terminé
Musique Grotte	Joan Zasempa	Terminé
Musique Fin	Joan Zasempa	Terminé
Bruits de l'environnement Lobby	Joan Zasempa	Terminé
Bruits de l'environnement Plage	Joan Zasempa	Terminé
Bruits de l'environnement Forêt	Quentin Sefrin	Terminé
Bruits de l'environnement Grotte	Joan Zasempa	Terminé
Bruits de l'environnement Fin	Joan Zasempa	Terminé
Sauvegardes	Quentin Sefrin	Pas terminé
Multijoueur	Quentin Sefrin	Manque les ennemis et certains objets

Dans le processus de création du jeu, nous avons intervertis certaines tâches. Ainsi, les bruits et musique de la forêt ont été fait par Joan, l'ennemi 4 et l'échelle par Nicolas Schmit, et le panneau par Nicolas Wittwe.

4.2 Le site Web

Notre site Web était déjà en place à la première soutenance. Nous l'avons légèrement modifié depuis, notamment en réglant quelques problèmes d'affichage et en y ajoutant le contenu de la deuxième soutenance. Nous avons également commencé à y ajouter les liens vers les différents sites où nous avons trouvé les assets que nous utilisons.

Son adresse est <https://ubihard-projet.netlify.app/>

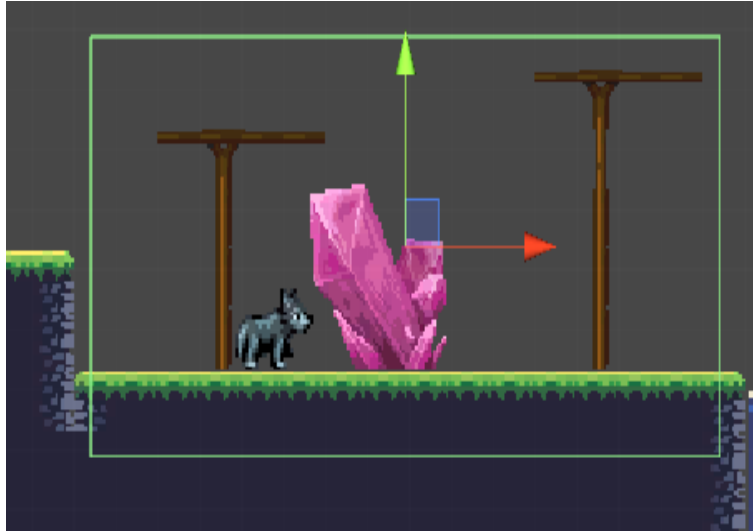


4.3 Les Ennemis

Pour cette soutenance intermédiaire, nous avons mis en place 4 ennemis supplémentaires :

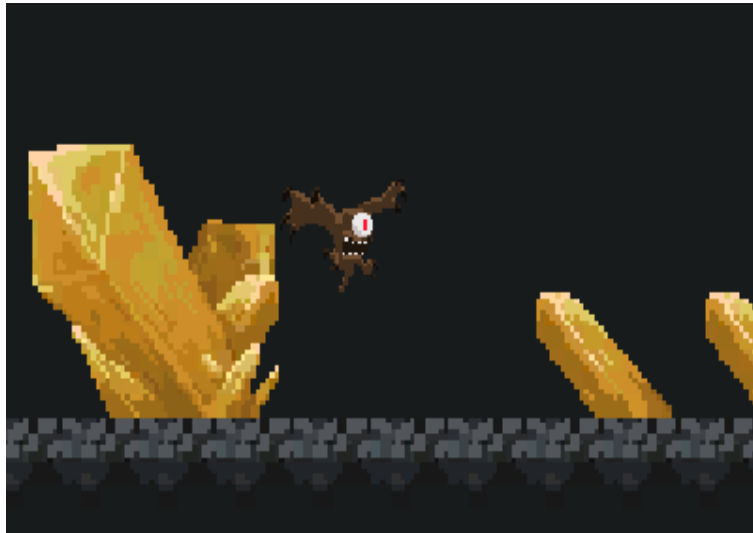
Loup

Le loup est un ennemi au sol qui patrouille comme le guerrier squelette. Cependant, il s'arrête parfois pour se reposer et il attaquera tout joueur qui entre sur son territoire. Le 'territoire' est en fait une box collider que nous réglons manuellement pour chaque loup.



Chauve-souris

La chauve – souris est un ennemi qui vole, elle se déplace entre plusieurs points et prend parfois des pauses ou elle ne se déplace plus. Si le joueur entre en contact avec la chauve-souris, elle lui infligera 1/2 coeur de dégâts.



4.4 Les pièges

Nous avons implémenté les piques vers le bas, qu'on l'on peut assimiler à un objet qui tombe. Quand le joueur rentre en contact avec une position située sous le piège, on déclenche `Active Rigidbody2D` (c'est à dire que l'on active la gravité pour l'objet) de l'objet ce qui entraine la chute de l'objet sur un axe verticale. Lorsque l'objet rentre en contact avec quelque chose il est détruit. De plus si l'objet rentre en contact avec le joueur il lui inflige des dégâts grâce a l'appelle de la fonction `TakeDamage`. A noter que les skins de cet objet dépendront du biome.



Enfin, le dernier piège de notre jeu est la zone de non-saut, où le joueur ne peut pas sauter et où sa vitesse est réduite. Celui-ci fonctionne comme suit : quand le joueur est dans cette zone l'attribut `runSpeed` du `Player` est divisé par deux et sa force de saut est mise à 0

4.5 Les objets

Nous avons déjà terminé la création de tous les objets de notre jeu. Pour cette soutenance, il s'agissait de :

Tout d'abord, modifier le fonctionnement des portes. Nous avons juste changé l'endroit où se trouve le booléen qui permet l'activation de la porte. Aux lieux de le définir dans chaque levier et de devoir créer une porte pour chacun, le booléen est maintenant créé dans `door` et est appelé par chaque actionneur (levier, cible...). Cela facilite l'utilisation des portes.

Ensuite, nous avons créé les torches. Chaque torche peut être allumée quand le joueur est proche et qu'il presse `E`. En plus celle-ci peuvent être utilisées pour des énigmes. En effet si certaines sont allumées (une combinaison sans ordre) alors elle pourra déclencher l'ouverture d'une porte. Cette combinaison est stockée dans une liste de booléens.



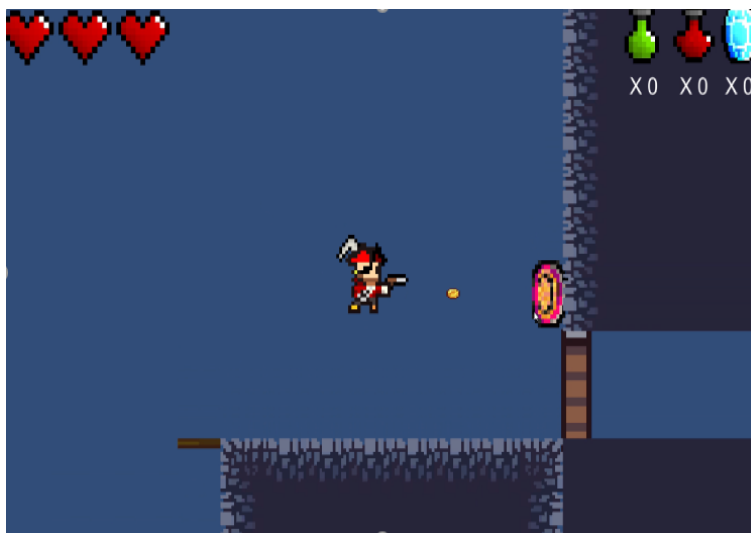
Nous avons également mis en place le panneau. Quand le joueur est proche du panneau et qu'il presse E, alors un affichage apparaît sur l'écran. Sur cet affichage il peut y avoir différentes informations, tels que des indices, des énigmes ou encore juste un peu d'histoire. Si le joueur s'éloigne du panneau celui-ci se refermera automatiquement.

Les panneaux seront utiles pour résoudre des problèmes, notamment pour trouver des codes. C'est là qu'intervient le digicode. Le digicode permet d'ouvrir une porte quand le joueur entre le bon code. Celui-ci fonctionne comme suit : quand le joueur s'approche de celui-ci et appuie sur E, l'affichage du digicode s'ouvre et le joueur peut rentrer un code, le valider ou le supprimer grâce aux boutons de l'interface. Si le joueur s'éloigne du digicode celui-ci se refermera automatiquement. De plus le code qu'il rentre est affiché sur l'écran. Quand le joueur rentre le bon code le digicode se ferme et la porte s'ouvre. Si le code est faux celui-ci s'affichera en rouge. De plus dans certains niveaux un code faux pourra infliger des dégâts au joueur. Le digicode peut être fermé en appuyant sur la touche E.



Toujours pour ouvrir une porte, nous avons ajouté la cible. Lorsqu'une balle ou une

flèche entre en contact avec la cible grâce à un box collider2D, celle-ci va renvoyer un booléen pour permettre d'ouvrir la porte qui lui est liée. Dans le futur, les cibles ne seront pas utilisables avec les balles du joueur.



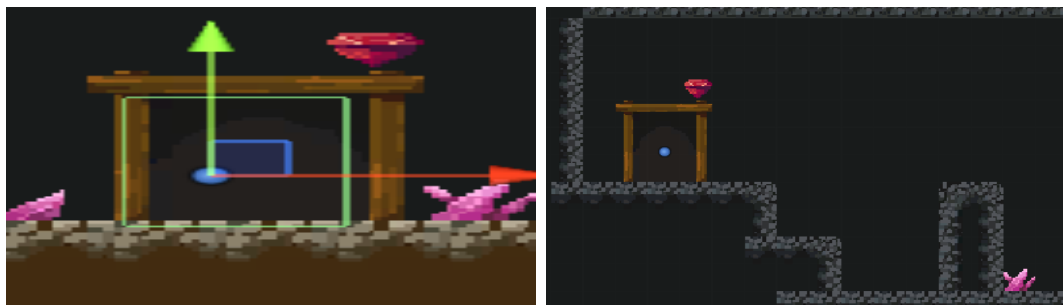
Un objet qui peut être utile pour certaines épreuves de parcours : le bloc mobile. Il s'agit tout simplement d'un bloc que le joueur peut pousser afin de passer certains obstacles ou atteindre des endroits trop hauts pour être atteint en sautant. Techniquement c'est un objet avec un vecteur de déplacement horizontal mais sans vitesse. Donc la vitesse lui est donnée par tout autre objet étant en contact avec lui.



Nous avons également implémenté l'échelle, qui nous donne une plus grande liberté dans la création des niveaux. L'échelle permet au joueur de se déplacer vers le haut. Elle consiste principalement en une box collider. Nous modifions la taille de cette box en fonction du niveau, cela permet de varier la taille des échelles. Quand le joueur est en contact avec la box collider et qu'il appuie sur E, il ne subit plus de gravité et gagne une force verticale vers le haut. Cela permet de faire monter le joueur.

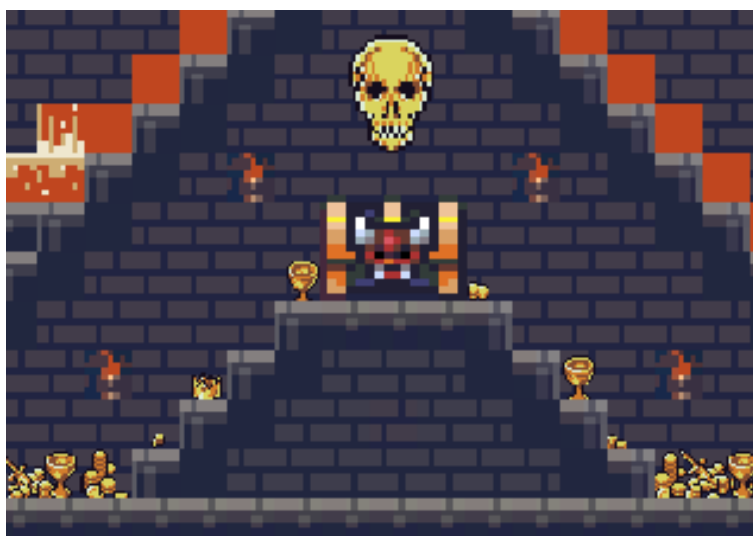


Nous avons également implémenté les portails. Le portail permet au joueur de se téléporter à un autre portail. Pour l'utiliser le joueur doit être suffisamment proche de ce dernier et appuyer sur la touche E. Une fois cela fait il se téléporte au portail de destination. Chaque portail possède un script « teleport » et dans ce script une variable « destination » de type portail qui représente le portail de destination. Pour permettre au joueur de se déplacer entre 2 portails, il faut donc ajouter deux portails dans la scène, puis assigner le premier à la variable destination du deuxième et inversement.



Pour passer d'un niveau à l'autre, nous avons créé une 'zone finale'. Quand le joueur est dans cette zone et qu'il appuie sur E, il sera envoyé au niveau suivant. Cette zone n'a pas de texture : celle-ci dépendra du niveau et de l'endroit où elle se trouve.

Enfin, nous avons rajouté un objet qui n'était pas prévu, le coffre final. Une fois celui-ci ouvert il lance la fermeture de la dernière porte de la salle afin de bloquer les joueurs. Pendant ce temps une coulée de lave tombe dans la salle au trésor et va tuer les joueurs. Et ensuite l'écran de fin se lance.



4.6 L'Interface

Pour cette soutenance, l'interface a peu évolué. Maintenant, le menu du jeu, ainsi que les boutons du menu pause, ont chacun leur texture finale.



4.7 Le Multijoueur

Le multijoueur a légèrement évolué depuis la première soutenance.

4.8 Les Sons

Nous avons implémenté les musiques de tous les niveaux, ainsi que leur sons d'ambiance. Par exemple, dans le niveau plage, on entend des mouettes. En revanche, les sons de certains ennemis et objets ne sont pas implémentés.

5 Troisième et dernière soutenance

Pour cette soutenance, nous avons finalisé l'implémentation du multijoueur. Nous avons supprimé certaines tâches du cahier des charges du à une raison familiale chez Joan Zasempa.

5.1 Les cheat code.

Nous avons décidé de changer les touches utilisées pour les cheat codes.

Next level (LShift + N) : Ce cheat code permet de passer au niveau suivant. En multijoueur, il fait passer tous les joueurs au niveau suivant

Final level (LShift + F) : Ce cheat code permet de passer de passer au niveau final. En multijoueur, il fait passer tous les joueurs au niveau final Treasure Room.

Unlock (LShift + U) : Permet d'ouvrir les portes proches du joueur qui l'utilise.

Invincible (LShift + M) : Permet de rendre le joueur invincible, il ne pourra plus subir de dégâts.

Nuke (LShift + K) : Permet de tuer tous les ennemis dans la zone actuel.

Give me life (LShift + G) : Permet de récupérer toute sa vie.

Relive (LShift + R) : Permet de ressusciter un joueur mort. En solo, permet au joueur d'activer ou de désactiver un mode "totem infini", dans lequel il réapparaît au début du niveau en cas de mort.

5.2 Amélioration de certains niveau

Dans cette soutenance nous avons revu certain niveau qui était trop court.

Cave :

Ajout d'épreuve des saut.

Le joueur doit obtenir un code qu'il doit rentrer dans un digicode. Le joueur doit aller dans 4 salles pour récupérer un chiffre, certaines de ces salles ont été améliorées.

Ajout d'une énigme : Avant d'accéder à la zone finale le joueur doit effectuer un parkour dans un temps limite.

Plage :

Le niveau de la plage est le premier niveau, il est donc le plus simple, ce niveau est linéaire et est un simple parcours, il permet de s'habituer avec les mécaniques du jeu.

Lobby :

Ce niveau a été refait complètement par Nicolas Wittwe. Il s'agit maintenant d'un bateau pirate. Le joueur commence dans la cale pirates et doit quitter le bateau par l'avant en sautant sur la plage et en passant à la zone plage. Il pourra trouver tout au long de son chemin des panneaux qui lui fourniront des indications sur le jeu en général. Il s'agit en fait d'un tutoriel.

Forêt :

Ajout d'une énigme bonus qui permet d'obtenir une coffre.

Mise en place de l'énigme principale du niveau : le joueur doit descendre à l'intérieur d'un arbre géant pour accéder au niveau suivant. Dans cet arbre, il y a de nombreuses portes qui permettent de continuer ou de gagner des bonus, mais pour ouvrir ces portes, le joueur devra mémoriser les fruits présents près des portes, car ils correspondent à un levier qui est placé dans un autre arbre en amont du niveau.

Treasure room :

Ce niveau n'a pas beaucoup évolué, étant déjà fini a la soutenance 2, mais il y a quand même eu de petit ajouts et ajustements.

5.3 Ennemis

Les ennemis 4 et 5 sur lesquels nous avons du retard ont été terminés. En effet, il ne possédait pas encore leur I.A complète durant la deuxième soutenance. En revanche, l'ennemi 2 a été abandonné.

5.4 Multijoueur

Le multijoueur a été finalisé durant cette troisième partie du projet, mais il n'est pas totalement fonctionnel. En effet, nous avons un problème qui n'est pas résolu. Le premier joueur d'une salle de 2 joueurs ou plus est déconnecté au bout d'un certain temps.

5.5 Interface

Les sauvegardes et la mort du joueur ont été mises en place :

Au début de chaque niveau, les données du joueur sont sauvegardées. Le joueur pourra donc démarrer une partie au dernier point de sauvegarde atteint durant sa dernière partie.

Quand le joueur meurt, la partie est terminée et le joueur retourne au menu principal. Cependant, s'il possède un ou plusieurs Totems de résurrection, il reprendra au début du niveau et perdra un totem. son avancement dans le niveau est conservé.

La procédure d'installation a également été mise en place.

5.6 Site Web

Le site est maintenant complet. On peut y télécharger le jeu et le guide d'utilisation, et la bande-annonce du jeu s'y trouve également.

Nous avons réalisé la bande annonce avec le logiciel Davinci Resolve 17 pour faire le montage. Voici la timeline de la vidéo :

Tout au long de la vidéo il y a une musique. Le montage a été réalisé en fonction de celle-ci

- Introduction du nom de l'équipe : UbiHard
- Premier clip du jeu
- Introduction du multi avec clip du jeu a 4 joueurs

- Introduction des énigmes avec clip du digicode
- Introduction des pièges avec clip d'un parcours piège a 3 joueurs
- Introduction des ennemis avec clip des chauve-souris et des squelettes
- Défilement de plusieurs images du jeu en rythme sur la musique
- Annonce du jeu et date

5.7 Les sons

Les derniers sons des objets, ainsi que ceux de tous les ennemis ont été implémentés.

6 Revue du projet

6.1 Nos peines et nos joies

Joie : Tout d'abord il y a la fierté d'avoir créé notre propre jeu vidéo et de pouvoir le partager à notre entourage. Le travail en groupe nous a permis de développer nos capacités de communication et de leadership pour certains. C'était aussi l'opportunité d'apprendre à gérer des crises tel que certains bug très handicapants. Cela nous a aussi permis de développer de nombreuses capacités comme le game design, le montage vidéo, la programmation en C# ou en html.

Peine : nous avons eu beaucoup de problème avec git, certains conflits nous ont fait perdre des choses que nous avions implémentés. Et nous avons eu des gros bugs qui nous ont fait perdre beaucoup de temps lors de la préparation des soutenances. Nous avons eu aussi beaucoup de problèmes avec le multijoueur qui était la partie la plus compliqué à implémenter. Au final, nous pensons que nous avons été trop ambitieux sur certaines parties de notre jeu, notamment les I.A. et la diversité des options prévues pour le multijoueur. Cela nous permettra de mieux appréhender nos capacités pour de futurs projets.

6.2 Avis personnels

Nicolas Wittwe

Ce projet m'a plus car en plus de pouvoir m'améliorer en programmation, j'ai amélioré mes capacités d'organisation et de communication. J'ai aussi découvert de nouvelles disciplines telles que le montage vidéo. Je suis aussi content qu'on ait pu réaliser notre propre jeu et de pouvoir le présenter à d'autres personnes.

Joan Zasempa

Je suis très fier du résultat final, j'ai aimé travailler et communiqué en groupe, et j'ai également acquis des connaissances que je ne pensais pas acquérir au début de l'année.

Nicolas Schmitt

Ce projet fut l'occasion pour moi de développer mes compétences de travail en équipe et d'apprendre à respecter des deadlines. Je suis aussi fier du résultat final que nous avons réussi à produire.

Quention Sefrin

Durant ce projet, j'ai appris à prendre sur moi durant l'implémentation du multi-joueur, qui m'a donné du fil à retordre. Cependant, j'ai acquis quelques compétences sur l'utilisation de serveurs en ligne dans un jeu. Enfin, le résultat de notre projet n'est pas celui auquel je m'étais attendu, mais aux vues de nos efforts et des difficultés rencontrés, je suis satisfait que l'on ait réussi à produire ce jeu.

6.3 Conclusion du Projet

Les objectifs que nous nous étions fixés au début de ce projet se sont révélés être trop élevés pour notre groupe. Nous avons donc du revoir certains aspects de notre jeu : nous avons simplifié les I.A des ennemis et diminué les options disponible pour le joueur, en mode solo comme en multijoueur. Cependant, nous avons réussi à finir ce projet et à présenter notre jeu, même incomplet.