

EPITA

PROJET S2

LES BARDES

GAMMA

RAPPORT DE SOUTENANCE 1



Table des matières

Introduction	2
1 Level Design & build	2
1.1 Premier niveau	2
1.2 Deuxième niveau	5
2 Game Design & gameplay	6
2.1 Téléporteur	6
2.2 Timer	7
2.3 Score	7
2.4 Effets de plateformes	8
2.5 Respawn, End Game, Déplacements	8
3 Multijoueur	9
4 Gestion des collisions & physique	10
5 Sound Design	11
6 Modélisation 3D	12
6.1 Personnages	12
6.2 Collectables & co	13
7 Site Internet	14
7.1 Création	14
8 Menus	15
9 Intelligence Artificielle	16
10 Avancement prévu pour la 2^{ième} soutenance	16
Conclusion	17

Introduction

Nous sommes le groupe des Bardes, composé de Romain HERMARY, Laure MARCHERON, Gaël GAUTIER et Louis LE GATT. Nous travaillons actuellement à la réalisation de notre projet de jeu vidéo : GAMMA. Il s'agit d'un jeu de plateformes musical, s'inspirant d'un mélange entre les univers de la série des Mario, de Portal, mais aussi de jeux musicaux. GAMMA sera composé de plusieurs niveaux jouables en multijoueur mais aussi en solo. Comme dit dans le cahier des charges, nous souhaitons un jeu accessible au plus grand nombre, avec des mécaniques de jeu simples à appréhender et à maîtriser. Pour parvenir à obtenir le jeu souhaité dans le temps qui nous est imparti, nous nous sommes réparti le travail en fonction de nos envies principalement, et de nos connaissances personnelles dans certains domaines comme la musique, par exemple. Nous avons commencé le projet très peu de temps après la validation de notre cahier des charges pour nous laisser un maximum de temps, et éviter les fameux "rush projet" mauvais pour la santé (mentale). Nous avons passé beaucoup de temps à discuter du projet, dans le but d'éclaircir certains points du projet, de définir ce que chacun devait faire, de la manière de nous organiser pour partager le travail. Nous avons décidé de nous voir tous les lundi au moins 2 heures pour que chacun explique ce qu'il a fait durant la semaine passée et ce qu'il compte faire pour la suite. C'est également la période durant laquelle nous avons essayé de mettre les différents éléments en commun, pour éviter d'avoir uniquement des éléments isolés. Dans ce rapport de soutenance, nous allons donc vous détailler chacune des étapes de la création de notre jeu GAMMA, à travers le point de vue de chacun des membres du groupe. Cela vous permettra de prendre connaissance de ce que chacun a effectué durant ce premier mois, mais également les joies et les peines de chacun. Nous allons ainsi vous détailler comment chacun d'entre nous a réussi (ou pas...) à obtenir ce qu'il souhaitait.

1 Level Design & build

1.1 Premier niveau

Gaël

Le Level Design a été la première partie du projet sur laquelle j'ai travaillé. En effet, avoir au moins un niveau sur lequel travailler était essentiel. Ainsi, j'ai fait au mieux pour que l'on puisse avoir le plus tôt possible le premier niveau construit. Tout d'abord, nous avons décidé sur quelle mécanique

de jeu nous voulions faire fonctionner le niveau. Nous avons donc posé les bases de plusieurs éléments qui devaient prendre en compte le fait que le jeu est avant tout un jeu de coopération, nécessitant donc d'être à deux pour pouvoir le faire. Après avoir posé les bases de ce que nous voulions faire, et les outils que nous avons imaginés pour le faire, nous avons commencé à designer le premier niveau, d'abord sous forme d'un croquis papier. Le fait que ce soit le premier niveau nous obligeait à ne pas faire un niveau trop difficile puisqu'il faut que le joueur s'habitue aux mécaniques de jeu, tout en ne faisant pas du niveau un niveau trop facile. Dans cette optique, nous avons mis en place le principe de plates-formes ne s'activant que lorsque les deux joueurs sont dessus, nécessitant le fait que ceux-ci soient coordonnés et avancent au même rythme. Nous avons également pensé à des portails qui permettent de se téléporter d'un endroit à l'autre de la carte. Nous avons ajouté ensuite quelques autres éléments plus classiques comme des bonus. Puis afin de compliquer tout de même le niveau et entraîner le joueur à une certaine dextérité dans les contrôles, nous avons placé un "labyrinthe" en fin de niveau, avec certains passages permettant d'accéder à un bonus.

Pour modéliser ce niveau, j'ai donc utilisé Unity directement, en créant une scène. Au départ je pensais créer un terrain où commencerait le niveau, puis petit à petit à l'aide des plates-formes on se trouverait au-dessus du terrain. Cela permettrait à celui qui tombe de reprendre du début sans perdre de vie. Le souci que j'ai rencontré a d'abord été dans la compréhension de l'outil terrain de Unity, il m'a fallu suivre un cours vidéo en anglais pour comprendre comment ce dernier fonctionnait. Après avoir suivi ce cours j'ai essayé de créer un terrain dans l'esprit du jeu que nous avons imaginé, c'est-à-dire en respectant les couleurs et le design que nous voulions faire. Au bout d'un certain nombre d'essais avec différentes configurations de terrain, j'ai finalement changé d'idée : aucune des configurations possibles ne correspondait au design auquel nous avions pensé pour le niveau, ce qui m'a donc poussé à repenser la manière dont j'allais créer le niveau. J'ai suivi un cours sur le site OpenClassrooms pour apprendre à maîtriser complètement Unity. Ce cours m'a permis de revoir mon idée du niveau et d'avoir enfin de vraie base sur le logiciel. Comme nous n'avons jamais été formés à Unity (hormis un TD facultatif et une conférence à laquelle il m'était impossible de participer car je travaillais), je m'étais donc retrouvé démuni devant. D'autant plus que la première fois que j'ai codé un jeu vidéo je n'ai pas eu le droit de me servir d'éditeur, ce qui m'a poussé à penser complètement chaque paramètre d'un jeu et pas seulement ce que doit faire mon jeu. J'ai donc dû comprendre comment m'y prendre avec une vision globale et non plus m'occuper d'un objectif précis à chaque fois.

J'ai donc recommencé la conception du niveau en utilisant pour les plates-formes des cubes. Cela m'a donc pris beaucoup de temps, au départ j'ai conçu le niveau à l'aide de blocs de plusieurs cubes, mais cela a posé problème : je me suis rendu compte que je pouvais tout simplement faire des cubes de la taille de mes blocs de cubes pour mes plates-formes. Cela aurait pu n'être qu'un détail mais en réalité c'était quelque chose de nécessaire, vu que certaines plates-formes devaient avoir des propriétés particulières. Il était plus favorable de les regrouper en une, car cela permettait de réduire le nombre de scripts et donc la taille du programme. Pour continuer dans la conception du jeu, j'ai décidé de placer les emplacements des mécaniques de portails, puisque ces dernières étaient incluses dans le design du niveau 1, bien que les mécaniques n'étaient pas encore été codées. J'ai utilisé des cylindres pour symboliser les portails.

Le souci que j'ai rencontré a ensuite été dans le design du niveau afin de jouer au mieux avec les mécaniques qui n'étaient pas encore codées, et surtout la 3D. Le problème avec une esquisse sur papier est que cela ne nous permet pas de visualiser les volumes et espaces nécessaires pour éviter que le joueur puisse sauter à un endroit où il ne devrait pas.

Enfin, le dernier problème que l'on a rencontré pour le Level Design est le design des personnages. En effet, ceux-ci ont été réalisés sur Blender par notre chef de projet. Or, sur Blender, l'échelle du modèle 3D du personnage était bien plus grand que le niveau créé, posant donc problème lors de l'implémentation du personnage. Nous l'avons donc redimensionné afin qu'il soit proportionnel aux plates-formes et aux champs de vision de la caméra. Mais après avoir redimensionné les personnages, nous avons dû redimensionner les plates-formes car ces dernières se trouvaient finalement être trop large par rapport au personnage.

Finalement cette première découverte du Level Design a permis de poser les bases de la gestion de Unity, et nous a appris à imaginer des solutions créatives et efficaces pour pallier aux problèmes que nous rencontrons lors de la conception d'un niveau. Cela nous a aussi permis de voir que le Level Design doit être pensé intelligemment afin d'être en cohérence avec le Game Design.

1.2 Deuxième niveau

Laure

C'est moi qui me suis occupée de la conception du deuxième niveau de Gamma. Après avoir discuté avec le reste du groupe et avoir dessiné un croquis du niveau, il a fallu créer une nouvelle scène sur Unity. J'ai repris très exactement les éléments que Gaël avait déjà implantés dans le premier niveau, à savoir les plateformes et les portails pour éviter de nouveaux problèmes d'échelle. Il a fallu concevoir pour ce deuxième niveau de nouvelles mécaniques de gameplay, et introduire la réelle nécessité de la coopération ainsi que les capacités individuelles de chaque personnage. Au début du niveau, on retrouve un portail que seul le personnage féminin peut prendre, tandis que le personnage masculin doit pousser un bloc pour le faire tomber d'une plateforme et pouvoir débloquer son chemin. Ensuite, le personnage féminin doit activer un bouton pour faire s'évaporer une paroi qui bloque le chemin. Plus tard, des interrupteurs contrôlent le déplacement d'une plateforme, sur laquelle les personnages doivent passer un par un. Celui qui reste sur les interrupteurs a le choix de la direction de la plateforme (droite-gauche) pour amener son coéquipier de l'autre côté. Le travail effectué par Gaël concernant les plateformes du premier niveau m'a permis de gagner un temps considérable car l'échelle des plateformes était déjà défini ; le plus fastidieux a été de placer chaque bloc à des coordonnées entières ou avec des puissances négatives de 2 pour que cela soit plus simple à manipuler et à éditer si jamais c'était nécessaire.

Malheureusement, il aurait été trop beau d'avoir un niveau aux coordonnées parfaites, dans lequel nous n'aurions eu qu'à poser le modèle du personnage et ses scripts. Nous travaillons avec GitHub, qui nous permet de mettre en commun notre avancée sans écraser la sauvegarde de chacun. Cela s'est révélé aussi pratique que farceur : nous avons eu une grosse frayeur d'abord, lorsque l'un de nos push a tout bonnement effacé le premier niveau de sa scène. Nous avons tant bien que mal réussi à le récupérer. Cependant, il nous est arrivé la même chose avec le deuxième niveau, et une autre mauvaise surprise : l'échelle du niveau a complètement changé, ce qui a chamboulé les coordonnées de toutes les plateformes que je m'étais appliquée à positionner simplement. Nous avons donc à présent un niveau dont les coordonnées sont au millième de précision, ce qui rend évidemment la tâche plus difficile pour éditer le niveau. Nous espérons tout de même pouvoir arranger cela.



FIGURE 1 – Premier niveau

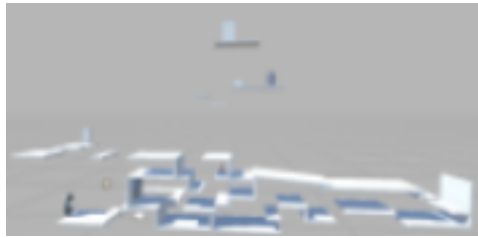


FIGURE 2 – Deuxième niveau

2 Game Design & gameplay

2.1 Téléporteur

Louis

Pour ajouter un nouvel élément de gameplay et continuer de travail sur les positions des personnages ainsi que leur déplacement dans la map, j'ai décidé de travailler sur la création d'un téléporteur. Ce téléporteur a pour rôle de déplacer l'un des joueurs d'un endroit défini à un autre en passant à travers une surface. Je me suis inspiré d'un tutoriel trouvé sur YouTube pour se faire, car je ne savais pas vraiment comment m'y prendre. J'ai créé des objets 3D sous la forme d'un téléporteur inspiré du jeu Portal. J'ai ensuite décidé d'insérer une caméra à l'intérieur de ces téléporteurs, afin d'avoir la vision sur l'endroit où le joueur arrivera après avoir passé le téléporteur. J'ai utilisé en plus de cette caméra un render texture qui est associé à la vision de la caméra, et qui lui permet d'être actualisé avec les éléments en mouvement de l'autre côté du téléporteur. Ainsi, le render texture de la caméra du premier téléporteur est associé au deuxième téléporteur, et inversement. Il ne restait ainsi plus qu'à appliquer un script à ces deux téléporteurs afin de faire changer le personnage d'endroit. Ceci étant fait, j'ai rencontré un problème. En effet, mon personnage était bien téléporté, mais celui-ci était téléporté à l'intérieur du prochain téléporteur, ce qui avait pour effet de téléporter le joueur indéfiniment, comme une boucle infinie. Afin de régler ce problème, j'ai ajouté un léger déplacement du personnage vers l'avant lors de

la téléportation. Après avoir ajouté notre propre modèle 3D, nous avons encore rencontré des problèmes sur cette téléportation, qu'il a fallu de nouveau résoudre à travers des modifications dans le script.

Romain Pour compléter le travail de Louis et remédier à des collisions très ennuyeuses lors de la téléportation, j'ai décidé de modifier un peu le code et rajouter une position à définir (celle d'un GameObject) qui représente la position d'arrivée.

2.2 Timer

Laure

GAMMA comportera des niveaux qui seront munis d'un timer. Cela impliquera plusieurs choses concernant chaque niveau : tout d'abord, ce timer indiquera aux joueurs combien de temps ils ont, au maximum, pour terminer le niveau. Nous ne voulons pas que cela conduise à du speedrun obligatoire, mais nous pensons que cela peut constituer un petit bonus satisfaction aux joueurs qui auront vite résolu les niveaux. Si jamais le timer tombe à zéro, la sentence est irrévocable : c'est un Game Over pour les joueurs, qui perdent immédiatement tous leurs points de vie et qui doivent tout recommencer du début. De plus, le timer aura une fonction de bonus score à la fin de chaque niveau, ce qui rejoint la satisfaction du joueur d'avoir fini le niveau rapidement, car il sera récompensé pour ses efforts.

L'implémentation du timer n'a pas été si compliquée que ça. J'ai eu de l'appréhension concernant le code en lui-même, car je mets en général du temps à comprendre ce que fait une fonction, et j'avais un peu de mal à comprendre les principes de classes, attributs etc. lors des TPs de C# en cours. Cependant, avec l'aide d'un tutoriel disponible sur YouTube qui expliquait pas à pas ce que devait faire le timer, et avec une explication brève de ce que faisait une fonction "OnGUI", j'ai réussi à créer un timer qui diminue d'une unité à chaque seconde, qui s'affiche en haut de l'écran à gauche dans un rectangle.

2.3 Score

Laure

Je parlais de bonus score concernant le timer, mais encore faut-il avoir implémenté un score en tant que tel. Celui-ci dépendra des bonus récupérés par les joueurs pendant le niveau, comme les notes de musique, ainsi que du timer comme dit plus haut. L'implémentation du score n'a pas été beaucoup plus difficile que celle du timer, à ceci près qu'elle requerrait l'appel à un autre

script écrit par Louis pour faire disparaître la note de musique une fois le joueur passé dessus et comptabiliser la valeur du bonus au score total. Grâce à ce script, j'ai pu comme pour le timer afficher le score dans un rectangle sur la partie haute de l'écran, qui a la capacité de se mettre à jour à chaque passage sur un bonus. Le rectangle obtenu dans ma fonction "OnGUI" rendait mieux visuellement que le canvas initialement prévu par Louis, c'est pourquoi nous avons préféré supprimer le canvas.

2.4 Effets de plateformes

Gaël

Au vu de mon implication sur le Level Design du niveau 1 et le site internet, je n'ai pas eu le temps de travailler beaucoup sur le Game Design. Néanmoins j'ai grâce aux cours d'OpenClassrooms pu apprendre comment gérer les collisions, comment récupérer les coordonnées d'un objet, comment les modifier, etc. J'ai donc commencé à me pencher sur le script des plateformes qui tombent et du principe de plateformes aux mouvements activés par un bouton. Pour cela j'ai donc évidemment regardé en détail comment fonctionnait la fonction `onCollisionEnter` de Unity. La compréhension de cette fonction m'a pris beaucoup de temps et ce principalement car la majeure partie de la documentation à ce sujet est en anglais, or mon niveau d'anglais n'étant pas encore à son paroxysme, ma compréhension s'est vue plus longue que d'ordinaire.

Pour les plateformes descendantes quand le personnage monte trop longtemps dessus, il me fallait donc récupérer le vecteur de position du personnage joué afin de pouvoir prendre en compte la collision entre la plateformes et le personnage joué. Une fois la collision prise en compte il fallait lancer un compteur qui au bout de 5 secondes supprime la plateforme. Le souci que j'ai rencontré est avec la fonction `Find`, je n'arrivais pas à récupérer les coordonnées du vecteur du personnage. La raison est que lorsque le jeu se lançait la fonction `spawn` renommait le personnage en y ajoutant un « (Clone) » à la fin du nom du personnage. Une fois ce problème réglé, je me suis retrouvé avec un autre problème auquel je n'ai pas eu le temps de pallier : la collision n'enclenche toujours pas le compteur.

2.5 Respawn, End Game, Déplacements

Romain

J'ai, pour éviter d'avoir une tombée infinie lors de la soutenance, décidé

d'implémenter un système de respawn quand le personnage tombe. J'ai rajouté des variables représentant les points de vie ainsi que les dégâts infligés aux joueurs lors de la chute. Quand le joueur n'a plus de vie, un message "GAME OVER" s'affiche au milieu de l'écran. Dans la même idée, et en modifiant le script du Timer de Laure, j'ai aussi implémenté le même message quand le temps est écoulé. Je n'ai pas réussi à trouver comment désactiver les commandes des joueurs, sachant qu'on n'utilise pas le "Player Controller" de Unity. Un petit souci à résoudre serait également de remettre à zéro l'orientation du joueur quand il respawn, car pour l'instant elle reste inchangée.

Pour les déplacements du joueur, nous avons décidé de ne pas reprendre le script de l'asset que nous avons jugé trop compliqué pour le moment, étant donné que nous n'avons aucune animation et pas encore besoin de commandes telles que l'accroupissement, le changement course/marche. Le saut pose encore quelques soucis, nous n'avons pas réussi à trouver comment simplement faire pour que le personnage ne puisse pas sauter indéfiniment, ou à des hauteurs différentes selon le temps de pression de la barre espace.

3 Multijoueur

Louis

Nous avons décidé de commencer par implémenter le multijoueur de notre jeu. En effet, nous préférons avoir le multijoueur fonctionnel pour ensuite implémenter les différents éléments du jeu sur ce multijoueur. Nous voulions éviter d'avoir un jeu fonctionnel en seulement en solo, mais avec des problèmes en multijoueur qui seraient très difficiles à reprendre.

N'ayant aucune base sur Unity, j'ai passé beaucoup de temps à regarder différents tutoriels pour apprendre à manier Unity. Ces tutoriels m'ont ainsi montré comment faire du multijoueur. J'ai donc commencé à faire des petits personnages mis en réseau ensemble, en utilisant uniquement les éléments d'origine de Unity. La prise en main de Unity et du réseau ne m'a pas semblé trop difficile. Cela m'a vite plu et rassuré pour la suite du projet. J'ai essayé de tester un maximum des options afin de comprendre le maximum des éléments de Unity pour ne plus me poser de questions pour la suite. Après ces nombreux essais qui m'ont vraiment aidé à me familiariser avec le logiciel, j'ai essayé d'implémenter quelques éléments qui seront présents dans le jeu final, tels que de petits objets collectables par les joueurs en réseau. Cette partie fut légèrement prise de tête. En effet, j'ai eu un peu de mal à comprendre la façon dont les objets étaient pris en compte par le multijoueur et les dif-

férents joueurs connectés sur le réseau. J'ai ainsi appris à créer de petites animations pour rendre les objets plus attractifs par le joueur, à ajouter des assets pour changer l'apparence de l'objet et finalement utiliser les canvas pour créer un système de score. Ce fut une partie très intéressante qui m'a permis d'appréhender une grande partie de Unity. J'ai eu à créer des prefabs, dupliquer des éléments, apprendre à utiliser la partie animation de Unity et finalement tout implémenter sur le réseau. Après plusieurs recherches et des discussions avec des élèves de l'année précédente nous avons décidé d'utiliser l'asset Photon pour la gestion du multijoueur. Il a fallu que je m'entraîne à manier cet asset qui facilite plusieurs choses pour le multijoueur. En suivant les tutos nous avons donc réussi à obtenir des personnages en multijoueur sur deux ordinateurs différents. Il restait à changer le personnage créé pour les tests pour le personnage créé par Laure sur Blender. Ce personnage devait pouvoir évoluer sur la map. Pour cela il a fallu décider de son "spawnPoint" qui représente le début de la map, là où le joueur va commencer le niveau. Il fallait aussi qu'un autre joueur se connectant sur un autre ordinateur puisse le rejoindre. C'est donc Photon qui nous a permis de mettre les personnages et de les synchroniser sur un réseau.

Romain

Je me suis occupé des scripts Photon qui permettent de se connecter à un serveur Photon depuis n'importe quel ordinateur, et de faire apparaître un personnage contrôlable par le joueur et visible par les autres personnes connectées au serveur.

Avec Photon j'ai donc réussi à faire spawn et bouger un personnage issu d'un asset (ses commandes et animations aussi), qui nous est encore utile car le modèle de Laure et ses commandes ne sont pas encore finis. Je me suis occupé de faire toutes les modifications nécessaires sur les prefabs des joueurs que l'on voulait faire apparaître pour que le système Photon démarre bien et fasse ce que l'on voulait. Il y a encore des lags sur le serveur entre les joueurs mais rien de bien méchant ou difficilement réglable. Le plus compliqué sera de faire pour que tout se synchronise correctement par la suite.

4 Gestion des collisions & physique

Laure

La gestion des collisions et de la physique a été une partie du projet qui m'a semblée très floue. Étant donné que je m'occupe principalement de la partie graphique, j'ai tout d'abord pensé à tenter de gérer les colliders de

mes modélisations directement sur Blender. J'avais lu qu'il était possible de le faire, certains disaient qu'il était même plus facile de les faire sur Blender pour ensuite les exporter sur Unity. Cependant, à cause de tous les problèmes que j'ai rencontrés lors de la modélisation, gérer tout cela sur Blender me paraissait assez risqué et très, très long. Finalement, pour la note de musique, j'ai préféré ne pas gérer le fait que ce devait être un objet rigide sur Blender. Je me suis contentée de la modéliser, puis c'est lors de l'export vers Unity que Louis et moi avons ajouté les propriétés "Rigid Body" à la note. Nous lui avons ajouté une capsule en guise de collider, ce qui nous a permis ensuite de la faire interagir avec le personnage lorsque celui-ci passe dessus via un script de comptage de score.

Pour ce qui est du personnage, lui appliquer des colliders a été un gros dilemme. Nous possédons actuellement le squelette des personnages que nous planterons dans le jeu, qui est le même et pour le garçon, et pour la fille. Cependant, il faut encore lui mettre des vêtements, des cheveux, et un visage pour différencier les deux personnages. De plus, les colliders doivent être bien plus précis sur les personnages que sur la note de musique. En effet, il faut gérer le fait qu'il y aura des animations sur ces personnages, donc des découpages de colliders pour que ceux-ci s'appliquent aux bras, aux jambes, à la tête et au tronc du personnage, car il aura des interactions spécifiques avec son environnement (pousser des blocs, encaisser des projectiles lancés par des mobs, etc). On ne pouvait pas se contenter ici d'une seule capsule peu précise. Sauf que l'animation du personnage n'était pas prévue pour cette première soutenance, car finalement secondaire au bon déroulement du jeu. Du coup, je me suis retrouvée un peu coincée quant à la décision à prendre. Finalement, après concertation avec le reste du groupe, nous avons décidé d'appliquer un collider au squelette du personnage quand même, de manière à pouvoir travailler avec quitte à devoir refaire les manipulations par la suite. Cela s'est également fait sur Unity, car j'étais trop peu à l'aise avec Blender.

5 Sound Design

Louis

Étant responsable du Sound Design du jeu, j'ai dû apprendre à utiliser les logiciels dont je vais me servir pour créer les musiques. Comme nous nous sommes principalement concentrés sur le Multijoueur et sur le Game Design, j'ai manqué de temps pour réellement commencer à composer. J'ai préféré apprendre à utiliser Ableton Live, qui est difficile à prendre en main car il est très direct et ne propose pas une interface très accessible. Cependant j'ai

persisté et je me suis entraîné, j'ai suivi des tutoriels et appris à utiliser des modules complémentaires au logiciel d'origine. Je me suis donc re-familiarisé avec mon clavier MIDI. Étant musicien depuis plusieurs années, la principale difficulté n'a pas été dans le fait de reprendre l'instrument en mains, mais principalement l'utilisation et la pratique du logiciel. Il faut donc que je m'habitue au logiciel pour n'avoir que la partie musicale sur laquelle réfléchir et ainsi obtenir de meilleurs résultats. De plus, je me suis renseigné sur les styles de musiques des platformers afin de me rapprocher des musiques déjà existantes. J'ai écouté les musiques de Mario et des différents grands platformers. J'ai essayé d'analyser ces musiques de relever les principaux éléments communs à ces musiques. J'ai ainsi pu relever un rythme très répétitif et adapté à la vitesse du jeu, qui peut varier selon le contexte du niveau. Il nous faudrait, après analyse, une mélodie facilement reconnaissable et peu complexe qui permet au joueur de la retenir. Finalement une quantité d'instrument réduite pourrait suffire à créer un bon thème. Les sons utilisés sont principalement réalisés par un synthétiseur ou un ordinateur.

6 Modélisation 3D

6.1 Personnages

Laure

Nous avons choisi de nous concentrer d'abord principalement sur les personnages pour cette première soutenance, car nous avons pensé que cela serait le plus technique à modéliser, donc le plus long et fastidieux.

N'ayant jamais touché à Blender avant, les premiers essais ont été assez difficiles. La prise en main du logiciel est très compliquée car beaucoup de raccourcis claviers à maîtriser pour gagner un temps considérable plutôt que d'aller dans les menus déroulants à chaque manipulation. De plus, nous avons eu un gros souci car, alors que j'avais presque terminé le corps du personnage, un malheureux fail clavier involontaire a effacé tout mon travail et je n'ai pas réussi à revenir en arrière. Du coup, la modélisation a pris presque deux fois plus de temps qu'il n'aurait fallu. J'ai appris à mes dépens qu'il vaut mieux sauvegarder le plus souvent possible...

J'ai commencé par modéliser un corps de base, sans vêtements ni autres détails, car nous avons choisi de garder la même morphologie pour les deux



FIGURE 3 – Croquis au propre

personnages, ce qui nous permettra de gagner du temps et pour la modélisation, et pour l'animation par la suite. A partir d'un cube et avec la fonction Mirror très pratique, j'ai avons pu réaliser un « corps » d'origine en entier, dans le style « cartoonish » que nous voulions donner à nos protagonistes. Le Chara Design des deux personnages que les joueurs pourront contrôler a été réalisé en amont sur le logiciel Paint Tool SAI à l'aide d'une tablette graphique. Nous avons pris la décision de proposer aux joueurs d'incarner un homme et une femme, pour des raisons d'égalité. Chaque personnage a une couleur attribuée pour rendre plus facile la compréhension du joueur une fois dans un niveau : en effet, nous prévoyons de symboliser les actions possibles pour chaque personnage à l'aide d'un code couleur : rouge pour le garçon, et jaune pour la fille.

Avec quelques inspirations telles que le style de personnage proposé par les *The Legend of Zelda* de consoles portables et du style chibi cher aux monde des mangas, j'ai croqué nos personnages de face et de profil. Cela m'a guidée pour les proportions et l'aspect général du personnage une fois sur Blender, car j'ai pu y glisser les croquis en arrière plan et modéliser par dessus. Le squelette est à présent modélisé, ce qui représente la partie la plus compliquée de la modélisation du personnage. Les vêtements et les textures seront ajoutées pour la deuxième soutenance.

6.2 Collectables & co

Laure

Ensuite, je me suis occupée de modéliser un des collectables, à savoir la note de musique qui représentera un bonus pour le score de fin de niveau. J'ai eu quelques soucis en modélisant, car je me suis retrouvée avec plusieurs



FIGURE 4 – Visualisation dans l'environnement prévu

"couches" de vertices et de faces, ce qui m'empêchait d'éditer la forme de la note. J'ai donc passé un certain temps à trouver ce qui n'allait pas, et encore plus à supprimer tout le surplus et faire le tour de toute ma modélisation pour être sûre que tout était bon.

Finalement, le dernier et plus gros souci que nous avons rencontré en voulant exporter les modèles Blender vers Unity (sous format .fbx, car le format .blend n'est pas optimal), a été la différence entre la convention d'axes sur Blender et sur Unity. En effet, sur Unity c'est l'axe Y qui pointe vers le haut, contre l'axe Z sur Blender. Cela pose un problème pour la mise en place des colliders, et pour les déplacements du personnage également. Nous avons mis beaucoup de temps à trouver un moyen de contrer cela, mais finalement à l'aide d'un tutoriel proposé sur YouTube nous avons pu nous débrouiller pour que le personnage apparaisse dans Unity avec les bonnes coordonnées. Il faudra encore lui apporter des modifications, donc potentiellement effectuer de nouveau la manipulation, mais cela prendra bien moins de temps les prochaines fois.

7 Site Internet

7.1 Création

Gaël

Pour la création du site j'ai commencé par d'abord réfléchir à l'aspect que je voulais lui donner. Ainsi j'ai donc regardé les différents types de site internet que l'on peut trouver sur le world wide web, et parmi ceux que l'on peut trouver il a fallu que j'en choisisse que je puisse réaliser. Une fois l'idée fixé sur le Design j'ai commencé à d'abord utiliser un éditeur de site en ligne,

Wix afin d'avoir un premier rendu visuel du site, or comme nous n'avons pas le droit d'utiliser un éditeur en ligne, vu que ce dernier ne permet d'avoir accès au script du site permettant de l'ouvrir en local j'ai dû changer d'éditeur. Ainsi j'ai donc utilisé Open Element afin de pouvoir éditer le site en local et avoir accès à l'ensemble des dossiers et fichier CSS le composant. L'utilisation d'un éditeur représente un gain de temps non négligeable pour rédiger le site. En effet grâce à ce dernier nous pouvons nous occuper directement du rendu du site et codé le PHP pour le blog plus tard. Pour réaliser le site je me suis donc inspiré d'un Template que j'ai trouvé correspondant à peu près à l'idée que j'avais du site.

8 Menus

Louis / Romain

Romain et moi avons décidé de travailler légèrement sur un menu. Cela nous a permis à tout les deux de comprendre le fonctionnement des canvas de Unity et de voir toutes leurs possibilités. Nous avons donc tous ensemble discuté de la façon dont on voulait que le menu soit implémenter. L'idée qui en est ressortie est que nous allons avoir un terrain sur lequel le personnage choisit par le joueur sera en mesure de se déplacer. Ce sera un terrain très restreint pour ne pas limiter les performances dans le menu et pour ne rien dévoiler au joueur mise a part les touches et la maniabilité du personnage dans l'environnement. Nous nous sommes donc renseigné sur la façon de créer un menu sur Unity à travers de nombreux tutoriels pour ne pas se limiter à une façon de faire. Nous avons appris à manier les canvas ainsi qu'un asset qui nous à semblé indispensable "TextMesh". Celui si nous permet d'avoir de nombreuses polices d'écriture et de modifier énormément d'options sur les différents textes des canvas. La gestion des différents éléments et leur incrustation sur le canvas pour que ceci ne bouge pas lorsque la taille de l'écran varie est encore aujourd'hui un problème. Nous souhaitons le résoudre pour la deuxième soutenance. Nous avons alors ajouté les éléments sur le canvas. Un bouton quit et un bouton play fonctionnels, un menu déroulent pour les niveaux pour le moment quasi-fonctionnel ainsi que des boutons pour choisir son personnage. Lors de la pression du bouton play, la bonne scene du bon niveau se charge mais le joueur ne spawn pas, nous n'avons pas encore réussi à implémenter ça.

9 Intelligence Artificielle

Romain

Pour l'IA, j'ai décidé de ne pas faire quelque chose de trop compliqué au départ. J'ai repris des prefabs que j'avais modélisées lors d'un autre tutoriel, en attendant d'avoir d'autres models et du coup faire une animation basique avec ce que j'avais. J'ai réussi à coder un script qui calcule la distance entre l'ennemi et le joueur : à une certaine distance l'ennemi réagit, sa tête devient rouge et il se tourne en direction du joueur. Cependant, je n'ai pas encore réussi à faire actualiser la distance entre l'ennemi et le joueur au fil du temps, ce qui fait que la seule distance prise en compte est la distance de départ entre le point de spawn du joueur et la position de l'ennemi. C'est sûrement un problème avec Photon car le tutoriel que j'ai suivi était set en local. Ensuite, j'ai commencé à coder des déplacements automatiques pour l'ennemi, une sorte de chemin de patrouille, qui marche quasiment : l'IA ne retourne pas au premier point de départ et ses rotations sont quelques fois hasardeuses. Mon objectif est donc de réussir à faire synchroniser les distances avec Photon, peaufiner le code des chemins suivis par les IA, ce qui me permettra de rajouter à ma guise et celles de mes coéquipiers des éléments de gameplay, d'interaction entre le joueur et les PNJ. J'ai déjà vu comment faire tirer un personnage, l'implémentation ne devrait pas poser de problème.

10 Avancement prévu pour la 2^{ième} soutenance

D'ici la deuxième soutenance de ce projet, nous espérons réaliser ces différents éléments :

- **Multijoueur** : Nous souhaitons avant tout essayer d'optimiser au maximum le multijoueur. Continuer à travailler sur Photon pour continuer à nous habituer et construire une base solide pour implémenter les autres éléments du Gameplay facilement sur le multijoueur. De plus il nous faut améliorer la gestion des deux personnages, de part la gestion des collisions entre eux mais aussi leurs déplacements dans les différents niveaux. Il faudra également prévoir deux sets de contrôles différents, un pour chaque personnage, lorsque le multijoueur est utilisé sur un même ordinateur (au pad numérique, via les flèches directionnelles, via z-q-s-d, ou autre ?)

- **Menu** : Le menu est déjà en grande partie fonctionnel. Le design est fait en très grande partie. Il nous reste à rendre le bouton du volume fonctionnel et ajouter quelques fonctionnalités comme le mode plein écran par exemple. Il faudra également adapter le menu à différentes résolutions, car selon la résolution les boutons ont tendance à se chevaucher, ce qui rend le menu illisible.
- **Sound Design** : La partie Sound Design n'est pas encore réalisée. Cependant les principales recherches, inspirations et l'apprentissage du logiciel sont faites. Cette partie devrait être réalisée en grande partie d'ici la deuxième soutenance. En effet, c'est une partie qui sera réalisée avec grand plaisir par les personnes concernées car il s'agit pour eux d'une passion.
- **Site Internet** :
Le site Web devrait voir dans son évolution venir les pages permettant de télécharger le rapport et le jeu ouverte d'ici la prochaine soutenance. Quand à la page "L'espace des Gammarois" , elle devrait être commencée d'ici la prochaine soutenance mais finit seulement pour la troisième soutenance. Quant au design du site il ne devrait pas évoluer grandement.
- **Game Design & Gameplay** : Les premiers éléments de gameplay ont commencé à être implémentés. Les portails, les objets collectables et des plateformes mouvantes ont déjà été ajoutés. Il nous faut maintenant implémenter des mobs, les interrupteurs, ainsi que les plateformes mobiles haut/bas et gauche/droite. Tous ces éléments sont bien sûr à implémenter au multijoueur, afin de toujours avoir un multijoueur fonctionnel quel que soit l'ajout des éléments. De plus, il va nous falloir placer ces éléments dans les différents niveaux. Les deux premiers niveaux étant déjà modélisés et dessinés sur papier, leur implémentation devrait être rapide.
- **Level Design & Build** : Les deux premiers niveaux sont entièrement dessinés. Ils vont être repris plusieurs fois avec l'ajout des éléments de gameplay mais la base est déjà là. Il faudra impérativement réussir à solutionner ce problème de coordonnées sur le niveau 2. Nous prévoyons de dessiner et modéliser les deux prochains niveaux, ainsi que commencer la scène relative à la partie musicale de fin de niveau.

Conclusion

À l'issue de cette première partie du projet, le bilan est plutôt positif : malgré une mise en place et prise en main du GitHub compliquée, qui est venue un peu plus tard que le début de la confection du jeu à cause de la découverte de certaines restrictions faisant barrage à notre évolution collective sur le projet imposées par Unity Collab qui était notre premier choix de partage, de nombreuses péripéties mettant en jeu tout notre projet comme les différents conflits entre nos fichiers, les mises à jour Unity faisant complètement planter le GitHub et notre projet, des erreurs venues de nulle part qui nous suppriment nos scènes entières, nous avons réussi à atteindre les objectifs que nous nous étions fixés et avoir une avancée globale satisfaisante. Plus notre projet se complète, plus ce qui nous reste à faire se voit perceptible et, l'expérience faisant, nous savons de mieux en mieux où chercher, comment chercher, et surtout adapter ce que nous trouvons, faisons, pour que ce soit efficace et que cela corresponde à nos attentes.

Les différents niveaux mettront maintenant surtout notre imagination à l'épreuve, le multijoueur a déjà une bonne base, quelques implémentations sont bien sûr encore nécessaires surtout au niveau de la gestion des deux personnages, le sound design est une partie qui n'a pas été beaucoup avancée mais qui le sera pour la prochaine soutenance, car c'est avant tout un hobby. Le site internet est quant à lui en bonne voie, les modifications devant être faites se feront au fur et à mesure de notre avancée. La modélisation des personnages, PNJ, et autres objets de notre monde est sur une bonne lancée. Les intelligences artificielles seront, quoique prise de tête, intéressantes à coder, et enfin l'esthétique et la mise en place du menu/lobby est à finir. La partie réseau sera une tâche importante.

Nous nous connaissons maintenant tous un peu plus, et savons à quoi nous attendre par la suite. Mener à bien notre projet ne se fera pas facilement : plus nous avancerons, plus les chemins que nous emprunterons seront semés d'embûches, mais nous continuerons de relever le défi et nous acharner à la tâche pour tout surmonter.