

EPITA

Rapport de Soutenance n°2 - projet S2

THE OTHER SIDE

Membres : Hugo Schlegel, Antoine Riquet, Angelina Kuntz, Mathéo Romé

Groupe : Hanabi



*Pour des raisons professionnelles, un des membres de notre groupe ne sera plus disponible après 16h,
excusez nous du dérangement.*

Table des matières

Introduction	4
Origine	4
Objectif	4
Scénario	4
Tableau de répartition des tâches : Deuxième soutenance	4
 Hugo	6
Début du deuxième rendu	6
Difficultés	6
Création d'assets pour le jeu	6
Gestion de l'évolution du stress	7
Menu pause	9
Implémentation des derniers Items	10
Site web	10
Plan pour la prochaine soutenance	11
 Mathéo Romé	12
Début du deuxième rendu	12
Difficultés	12
Correction de bugs	12
Item passif : feu de camp portatif	12
Système du respawn	12
Système de stockage des données	13
Système des salles torches	13
Passage entre les scènes	14
Multijoueur	15
Animation du Dash	18
Prochaine Soutenance	18
 Angelina	19
Début du deuxième rendu	19
Correction de bugs	19
Système capteur	19
Génération aléatoire	19
Musique	21
Création de toutes les salles des zones 1 et 2	21
Montage vidéo	24
Prochaine Soutenance	25
 Antoine	26
Début du deuxième rendu	26
Effets visuels du stress	26
Récapitulatifs de l'ensemble des objets	27
Intelligence artificielle	28

Conclusion pour cette soutenance	29
Conclusion	30

Introduction

Origine

Ce projet est né d'une passion commune à l'ensemble du groupe pour le jeu Celeste et des jeux de type Roguelike. Celeste est un jeu vidéo de plate-formes mélangeant action et puzzle sorti en 2018. Dans celui-ci, chaque ensemble d'actions doit être réfléchi et organisé pour achever un tableau. L'histoire du jeu est basée sur le modèle Kübler-Ross, expliquant les cinq phases du deuil. Notre but, au travers de ce projet, est de réaliser un jeu mélangeant jeu de plate-formes et Roguelike.

Objectif

Après avoir implémenté les bases du projet pour la première soutenance, l'objectif de cette deuxième soutenance est de consolider cet ensemble et de l'enrichir. Contrairement à la première soutenance, tous les membres du groupe ont toutes leurs tâches définies et maîtrisent déjà les bases de Unity, qui permettra, dans un vif espoir, d'être efficace et de réussir tous les objectifs qu'ils se sont attribués.

Scénario

Le nom du projet The other side est fondé sur l'intrigue du jeu, plus précisément sur l'aspect du dédoublement de la personnalité du personnage principal.

L'histoire se déroule à une époque similaire à la nôtre, peu avant le nouvel an, et ainsi, des feux d'artifices. Le personnage principal est un homme qui manque de confiance en soi et qui cherche sa place dans le monde, un but à son existence. Le personnage souffrant du dédoublement de sa personnalité, cherche à gravir une montagne afin de se réunir avec son double et d'accomplir un exploit pour se redonner de la valeur et obtenir la reconnaissance des autres.

Le personnage de base sera visible de tous contrairement à son double. Cependant, cela permettra au double de percevoir ce qui est invisible pour le commun des mortels. Le personnage détient de sa personne et se veut visible et invisible à la fois, créant ainsi un double qui existe et n'existe pas en même temps. L'entité du double se base sur le théorème de superposition quantique émis par Schrödinger.

Tableau de répartition des tâches : Deuxième soutenance

Petite pique de rappel de ce que chaque membre devait accomplir pour la deuxième soutenance d'après le cahier des charges.

Soutenance 2	Hugo	Matheo	Angelina	Antoine
Dessiner l'animation du dash	x			
Implémenter l'animation du dash		x		
Implémenter les variations du stress du personnage, augmentation continue du stress	x			
Implémenter les variations visuelles du stress, blur et musique saturée				x
Implémenter la génération aléatoire			x	
Implémenter une salle capteur			x	
Implémenter une salle torche		x		
Implémenter les salles feu de camp	x			
Implémenter la bibliothèque de la zone 1			x	
Implémenter la bibliothèque de la zone 2			x	
Implémenter une IA capable de reproduire les mouvements du joueur				x
Permettre au joueur 1 de stocker localement toutes les données de la partie (Améliorations achetées, Strawberries et Starfruits en stock)		x		
Implémenter la stabilisation du stress dans les salles boutiques et feu de camp	x			
Implémenter un menu pause	x			
Implémenter tous les items achetables de la boutique				x
Implémenter l'item passif : Long-fall boots	x			
Implémenter l'item passif : Feu de camp portatif		x		
Implémenter l'item actif : The World	x			
Avoir rempli le site web dans son intégralité à l'exception du lien de téléchargement	x			

Hugo

Début du deuxième rendu

Pour cette seconde soutenance, j'aurais encore eu la chance de travailler sur de nombreux aspects de notre jeu. J'aurais premièrement continué à travailler sur les assets de notre jeu avant de m'occuper de gérer l'évolution du stress de notre personnage ainsi que l'implémentation des salles "feu de camp". La suite aura pour moi été de travailler sur le menu pause puis de terminer de créer tous les items restants et enfin remplir et peaufiner le site web. Travailler sur cette soutenance, difficultés mises à part, aura été plus fluide que pour la première, ayant déjà acquis les bases de Unity et n'ayant pas à faire des allers-retours entre mon code et divers tutoriels.

Difficultés

Pour être tout à fait honnête, cette soutenance n'aura pas été des plus agréables pour moi, le décès d'un membre proche de ma famille il y a peu, en plus de me démorraliser aura posé de gros problèmes d'organisation, et une grande majorité de mon temps fut passée loin de chez moi et dans des conditions inappropriées au travail, et on peut rajouter à cela les journées complètes que je me devais de réserver à ma famille. En bref je n'ai pas eu autant de temps que je ne l'aurai voulu pour travailler sur cette soutenance et j'aurais dû me motiver bien plus que la normale. J'ai cependant réussi à faire l'intégralité de ce qui était prévu pour cette soutenance et ferai de mon mieux pour la prochaine. Il est aussi important de noter le gros bug que nous avions eu avec Unity au moment de recommencer à travailler sur notre projet, que Mathéo expliquera en détail dans sa partie.

Création d'assets pour le jeu

J'ai été chargé pour cette soutenance de finir de dessiner les assets de notre personnage principal, soit les animations de dash ainsi que ceux de notre IA, à l'allure fantomatique et ressemblant à notre personnage. Tout comme pour la première soutenance, j'ai utilisé Piskel pour dessiner les quelques assets restants, assets dont je suis plutôt satisfait, maîtrisant maintenant bien mieux le logiciel.

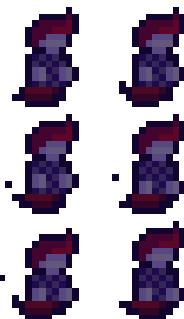


FIGURE 1 – Tile set de notre fantôme

Ci-dessus la tile set de notre IA. Elle aura été faite pour ressembler à notre personnage tout en ayant un air plus menaçant. Celle-ci n'est visible que quand notre stress passe un certain seuil, après lequel elle retrace tous nos mouvements, la moindre erreur ou doute pouvant nous être fatal (cf. le rapport d'Antoine sur l'IA).

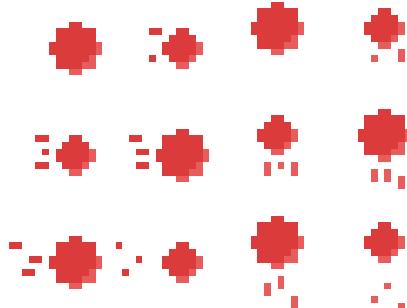


FIGURE 2 – Deux tile sets du dash classic

Ci-dessus deux des tile sets du dash classic, respectivement le dash vers la droite et le dash vers le haut. Ces tile sets au premier abord plutôt simples n'apparaîtront à l'écran qu'au cours du dash, il en existe une pour chacune des 8 directions possibles pour le dash classic, mais étant toutes quasi-identiques (ne différent chacune que de part une légère rotation) je ne vous en présente que deux.

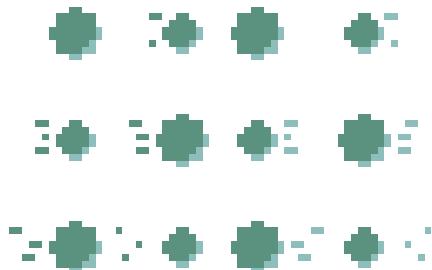


FIGURE 3 – Les deux tile sets du dash bouncy

Très similaires à celles du dash classic, voici les deux seules tile sets du dash bouncy (elles ne sont qu'au nombre de deux comme le dash de la classe bouncy ne peut aller que vers la gauche ou la droite).

Gestion de l'évolution du stress

Ma seconde tache était d'implémenter l'évolution continue du stress, les joueurs gagnent maintenant du stress à raison d'un point toutes les deux secondes. Celui-ci augmente en continu avec un maximum de 200 points de stress, palier qui une fois atteint résulte en l'apparition de l'IA. Il existe cependant quelques exceptions à cette règle, j'ai fait en sorte que le stress n'augmente pas dans les salles histoire et les salles feu de camp, qui sont faites pour que le joueur puisse prendre

son temps et se reposer, et ce, en vérifiant au moment d'y entrer l'index de la salle afin d'en déduire son type, n'augmentant le stress que si l'index ne correspond pas à l'unde des salles histoire ou feu de camp.

Je me suis justement aussi occupé d'implémenter ces salles feu de camp. Comme mentionné juste avant, le stress ne monte pas dans ces salles, une seule apparaîtra par étage de la montagne et il est impossible de chuter ou mourir dans celles-ci. Au milieu de ces salles se trouvera toujours un feu de camp et si le joueur s'approche du feu, soit qu'il touche son collider, son stress sera diminué de soixante pour cent. J'ai délibérément fait en sorte que le stress ne soit retiré au joueur que s'il s'approche du feu de camp de telle façon à ce que, s'il le souhaite, il puisse rendre sa run plus difficile en décidant de ne pas faire usage de ce bonus.

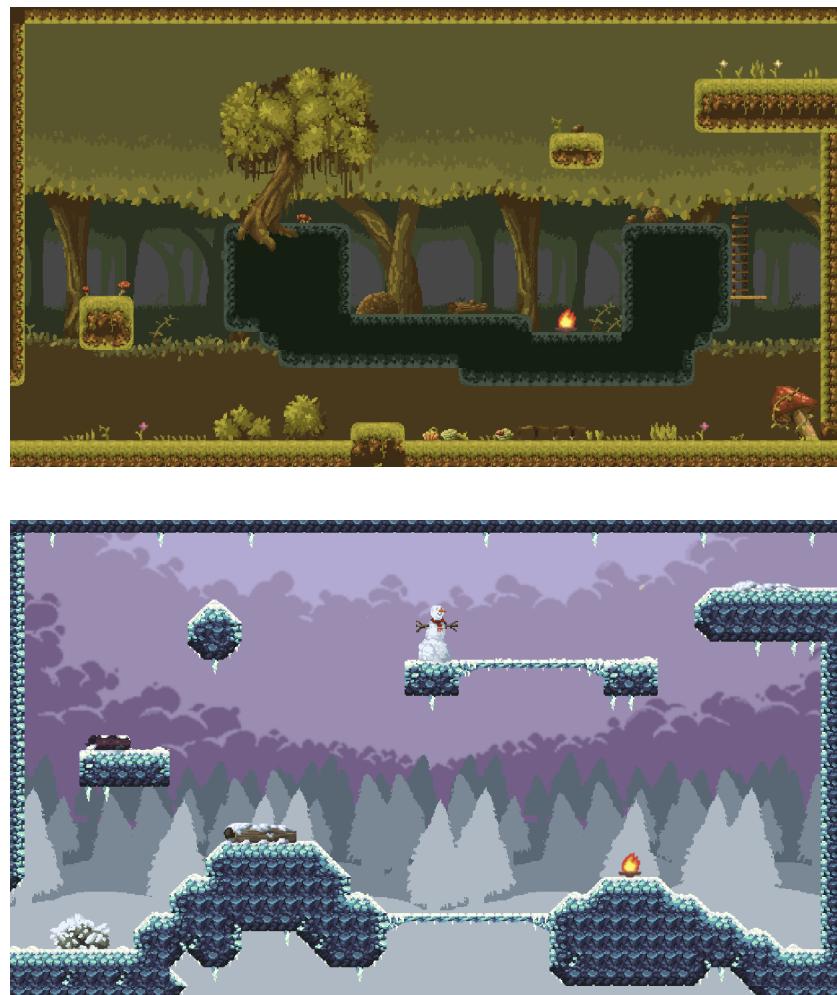


FIGURE 4 – Les salles feu de camps des zones 1 et 2

Menu pause

Tout de suite après avoir terminé de travailler sur les salles feu de camp, je me suis occupé de faire le menu pause de notre jeu. Il peut être ouvert à n'importe quel moment en appuyant sur la touche echap, freezant le jeu, le temps arrêtant de s'écouler et assombrissant le fond d'écran. Sur ce menu sont visibles trois boutons, *RESUME*, *MENU*, *OPTIONS* et *BACK*, qui permettent respectivement de reprendre le cours du jeu, de retourner au main menu, d'accéder au sous menu options du menu pause et de quitter le jeu. Pour nous ramener au main menu, le bouton *MENU* nous ramène à la salle d'index 0, qui correspond toujours à l'index du main menu. À noter qu'il est aussi possible de sortir du menu pause à tout moment en pressant la touche echap.



FIGURE 5 – Menu pause

Le sous menu options est presque identique à celui présent dans le main menu, on y retrouve les même options de *Fullscreen*, *graphics* et *Volume*. Le bouton *BACK* lui nous ramène au premier menu pause. Il est également possible de reprendre le cours du jeu à partir de ce menu en pressant la touche echap.

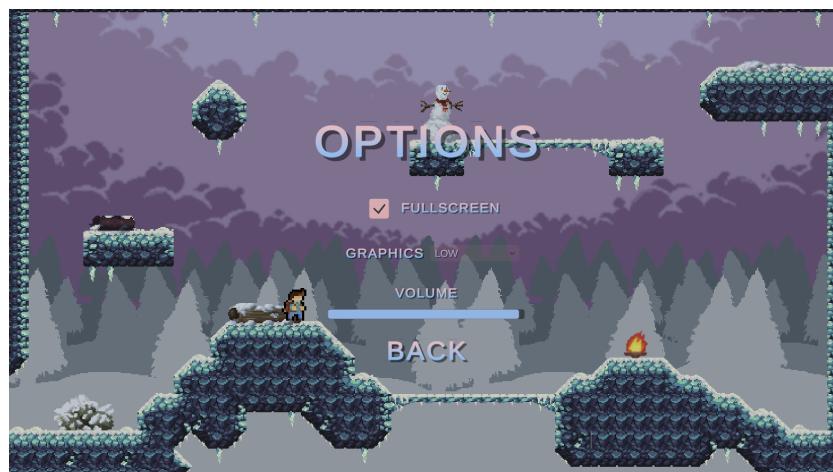


FIGURE 6 – Menu pause

Implémentation des derniers Items

Pour ce qui est des items, je me suis occupé de l'item passif *long-fall boots* et de l'item actif *The world*.

Comme tous les items actifs, les *Long-fall boots* peuvent être trouvées dans des boutiques dans trois variétés différentes, *broken*, *used* et *new* qui, une fois achetées, resteront de façon permanente dans l'inventaire passif. Cet item à pour effet de réduire le nombre de points de stress gagnés au moment de tomber dans un trou. S'il est dans l'état *broken*, l'item nous sauve 1 point de stress en tombant, s'il est dans l'état *used* il nous en sauve 2 et s'il est dans l'état *new* il nous en sauve 3. Ces valeurs peuvent sembler petites, mais il faut rappeler qu'il est possible de posséder plusieurs fois le même item passif afin d'additionner ses effets, il est donc même possible si l'on a la chance de trouver plusieurs fois cet item de ne plus du tout gagner de stress en tombant.

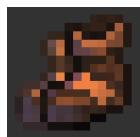


FIGURE 7 – Long-fall boots

Pour ce qui est de *The world*, étant un item actif, un fois possédé, il est stocké dans le slot des items actifs et il doit être activé afin d'utiliser son effet. Une fois utilisé, cet item permet d'effectuer un dash identique à celui de la classe light, soit une téléportation vers la gauche, le haut ou la droite. Le cooldown avant la prochaine utilisation de l'item est de 15 secondes, comme tous les autres items actifs de mouvement.

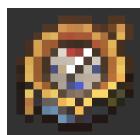


FIGURE 8 – The world

Site web

Pour ce qui est du site web, je me suis occupé de le remplir dans sa quasi-intégralité, laissant juste de côté les sections *Téléchargement* et *Bibliographie* que je ne serai en mesure de compléter qu'une fois le jeu terminé, comme nous ne cessons de chercher de nouvelles sources d'inspiration et que notre jeu n'est pour l'instant pas complet. Je me suis aussi occupé de changer les noms et de revoir certaines sections afin d'au mieux pouvoir présenter notre jeu. La section *Accueil* est une simple présentation de notre groupe ainsi que de ce que notre jeu se veut être. La section *Pourquoi The other side* elle met en avant les origines de notre projet ainsi qu'un bref résumé du scénario. La section *Avancement et objectifs* contient les fichiers.pdf de nos rapports de soutenance et de notre cahier des charges, pour les plus curieux qui voudront en apprendre plus sur comment nous avons fait pour venir à bout de notre projet. La section *Comment jouer* est la plus longue d'entre

toutes, elle contient tout ce que vous devez connaître de jouer à *The other side*, toutes les mécaniques des salles, des différentes classes, les contrôles et autre y sont expliqués. Enfin, la section *Contact* contient les mails de tous les membres du groupe ainsi que l'adresse et le nom de notre établissement.

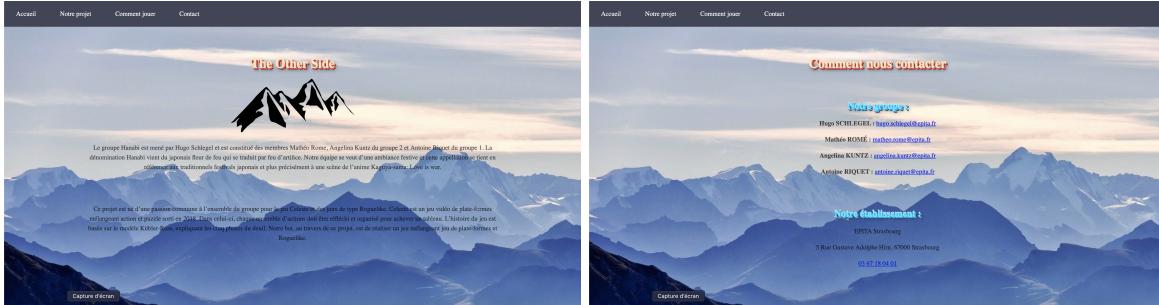


FIGURE 9 – La page d'accueil et contacts de notre site

Plan pour la prochaine soutenance

Pour la prochaine soutenance, je commencerai par travailler sur la synchronisation de l'inventaire passif des deux joueurs, en effet seul le joueur 1 est capable d'acheter des items, il faut donc que leurs effets puissent être communs aux deux. Par la suite, je m'occuperai avec les autres membres du groupe d'implémenter la génération aléatoire des pièces et des items actifs dans les salles, chose que nous avions omis de mentionner dans le cahier des charges et que nous ferons donc tous ensemble. Et pour finir, moi et tous mes camarades participerons à la grande chasse au bug finale, qui viendra clôturer le projet.

Mathéo Romé

Début du deuxième rendu

Pour cette soutenance, je devais m'occuper d'implémenter les animations des classes *Classique* et *Bouncy*, la classe *Light* étant une téléportation il n'y a pas besoin d'animation. J'ai également implémenté un item passif, le feu de camp portatif. J'ai également créé un type de salle épreuve supplémentaire : la salle torche. Enfin, j'ai créé un système de stockage de données afin de conservé ses données entres les parties. De plus, j'ai grandement amélioré le système de multijoueur, ajoutant une interface de création de salon ainsi que la synchronisation de données entres les joueurs.

Difficultés

Lors de cette semaine, notre groupe a été confronté à un problème majeur qui était lié à la mise à jour du launcher Unity Hub. Quand j'ai lancé le projet, celui-ci s'est lancé comme projet 3D. Nous avions eu du mal au début à trouver l'origine du problème, après plusieurs heures de recherches et de tentatives à convertir notre projet pour le rendre compatible avec l'ancienne version d'Unity Hub, j'ai réussi à remplacer le projet 3D par une ancienne version de notre projet qui était encore présente sur l'ordinateur d'un de mes coéquipiers qui n'avait pas encore pull la version bug du projet.

Correction de bugs

Afin d'améliorer la qualité de jeu, j'ai dû apporter des modifications à la classe *Light* qui avait de nombreux problèmes lors de sa téléportation : en effet, il lui arrivait assez régulièrement de pouvoir traverser les murs extérieurs de la scène et ainsi tomber dans le vide, mais aussi d'atterrir à l'intérieur d'un obstacle dans la scène. Ces bugs ne se manifestaient que dans certains cas si bien que nous nous n'en sommes pas immédiatement rendues compte : il fallait être contre directement contre le mur pour provoquer ce bug et par manque de temps lors de la dernière soutenance nous n'avions pas vraiment testé ce cas-là. J'ai donc modifié le comportement de ce dash en ajoutant un nouveau collider, plus petit, qui ne s'active que durant le dash et en désactivant le collider principale. Cela a ainsi permis de régler le de téléportation lorsque le joueur se tenait contre un mur.

Item passif : feu de camp portatif

Pour cette soutenance, mon premier objectif a été d'implémenter l'item *feu de camp portatif* qui permet d'augmenter l'intervalle de temps auquel le stress augmente. L'intervalle de stress augmente de 0.25/0.5/0.75 selon la rareté de l'objet. Grâce au modèle des autres items, cette partie a été assez rapide à faire.

Système du respawn

Tout jeu de plateforme 2D à un système de respawn digne de nom lorsque son personnage meurt, par exemple lorsque celui-ci tombe dans le vide. Si rien n'est fait, le malheureux chutera vers l'infini et au-delà. Pour empêcher cette tragédie, si le joueur a la malchance de tomber, le



FIGURE 10 – Item feu de camp portatif

gameObject que j'ai intitulé "void" sera là pour le rattraper. Notre bon ami void est une plateforme transparente qui recouvre toute la salle en longueur en dessous de celle-ci. Si le joueur entre en contact avec lui, il sera téléporté à son point de spawn positionné au début de la salle et le stress des deux joueurs augmentera de 10.



FIGURE 11 – Void en dessous des salles

Système de stockage des données

Je me suis ensuite occupé de créer un système de sauvegarde qui permet aux joueurs de sauvegarder leurs données telles que les pièces, les raspberries ainsi que les améliorations achetables à la fin d'une partie (ces dernières n'ont pas encore été implémentées.) puis de charger ces données au début de la prochaine partie. Il est à noter que c'est la sauvegarde du joueur un qui sera prise en compte lors de la sauvegarde et de la récupération de données. De plus, le système ne sauvegardera les données qu'en fin de partie, à la mort du personnage ou à son arrivée à la fin du jeu : quitter la partie en cours de route laissera la sauvegarde tel qu'elle était au début de partie.

Système des salles torches

Par la suite, je me suis occupé d'implémenter le système des salles torches, pour ce faire, je me suis basé sur le système des capteurs. Pour que le joueur déverrouille sa porte, il doit activer

5 torches dans un ordre bien précis. Si une torche est allumée trop tôt, toutes les torches déjà allumées vont s'éteindre et les joueurs devront recommencer depuis le début. Pour résoudre ce puzzle, les deux joueurs devront donc coopérer, en effet, le premier joueur peut activer les torches tandis que le second à connaissance de l'ordre dans lequel il faut les allumer.



FIGURE 12 – Salle torche du joueur 1 et du joueur 2

Le premier joueur ne peut pas activer une torche si ce n'est pas son tour et s'il se trompe toutes les torches vont s'éteindre et le joueur doit recommencer depuis le début. Pour allumer une torche, le joueur doit se tenir à côté d'une d'entre elle c'est à dire que le collider du joueur est en contact avec le collider de la torche qui reconnaît que c'est un joueur et il doit appuyer sur E. Lorsqu'une torche est activée le sprite de la torche éteinte change pour l'animation de la torche allumée.

Pour l'animation de la torche j'ai utilisé le pack gratuit de [Torch Sprite Pack by Asymmetric](#)

Passage entre les scènes

De nombreux objets de notre jeu doivent être présents à tout moment quelle que soit la scène. C'est bien sur le cas du joueur et de tout ce qui l'entoure (sa caméra, son canvas) mais aussi pour l'audioManager gérant la musique, le canvas servant au menu pause ainsi qu'un gameManager. Pour cela, j'ai utilisé une fonction de Unity, *DontDestroyOnLoad* qui permet d'indiquer à Unity de ne pas détruire les objets affectés par cette fonction lors d'un changement de scène.



FIGURE 13 – Torches allumées et éteintes



FIGURE 14 – Torche animation

Multijoueur

La plus grosse partie de travail sur cette soutenance a été l'amélioration du multijoueur. Certes le multijoueur étais déjà fonctionnel mais il étais améliorable et c'étais dans ce but que je ne m'étais pas fixé énormément de travail pour cette soutenance.

Je me suis tout d'abord occupé d'améliorer le menu précédemment créé par Hugo afin de pouvoir choisir le mode de jeu entre solo et multijoueur (le mode solo n'est pas encore fonctionnel.).

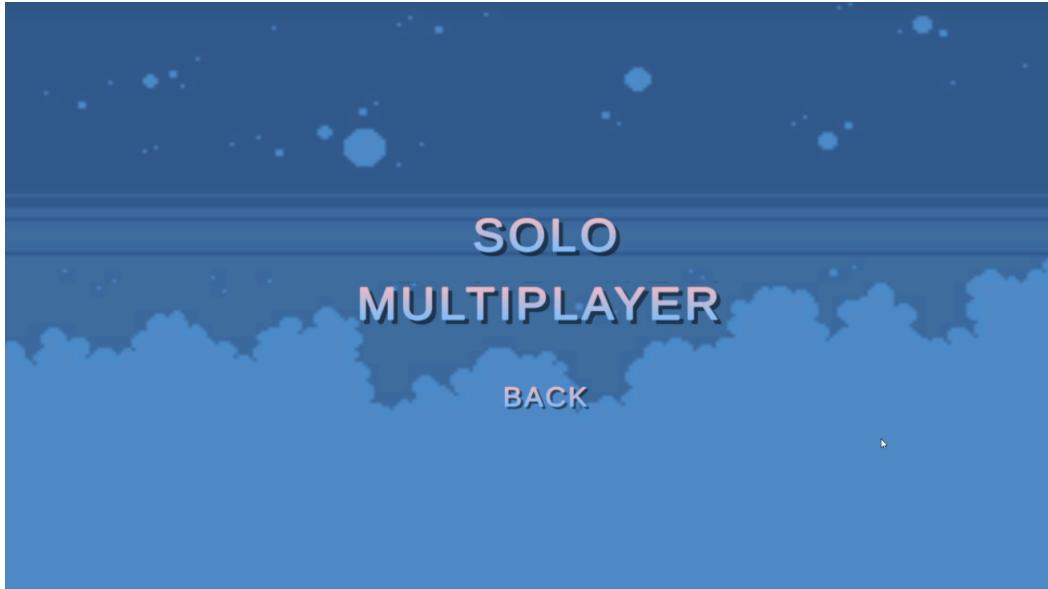


FIGURE 15 – Menu pour choisir le mode de jeu

Cliquer sur le bouton multijoueur dirige alors le joueur vers une interface où il pourra choisir son pseudonyme, créer un salon avec un nom spécifique et enfin voir la liste des salons déjà créé et les rejoindre.

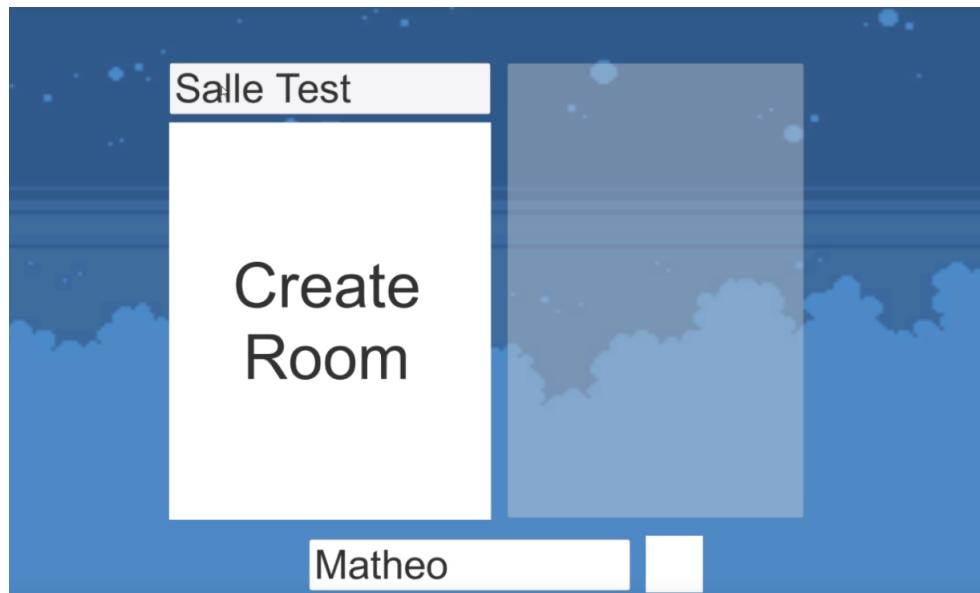


FIGURE 16 – Crédit d'une room et sélection du pseudo du joueur1

Une fois que le joueur a rejoint une salle, il aura accès sur la gauche à une interface permettant de quitter le salon et ainsi nous ramener à l'écran de sélection de salons. Il y a ensuite la liste des noms des joueurs présent dans la salle située au centre de l'écran. Sur la droite de l'écran, se situent trois boutons permettant de choisir la classe voulue pendant la partie. Cliquer sur un des boutons désactivera automatiquement les deux autres ainsi que le bouton correspondant à cette classe chez l'autre joueur. De plus, le joueur 2 devra cliquer sur le bouton "N" pour signifier qu'il est prêt à lancer la partie. Une fois que chacun a choisi une classe différente et que le joueur 2 à

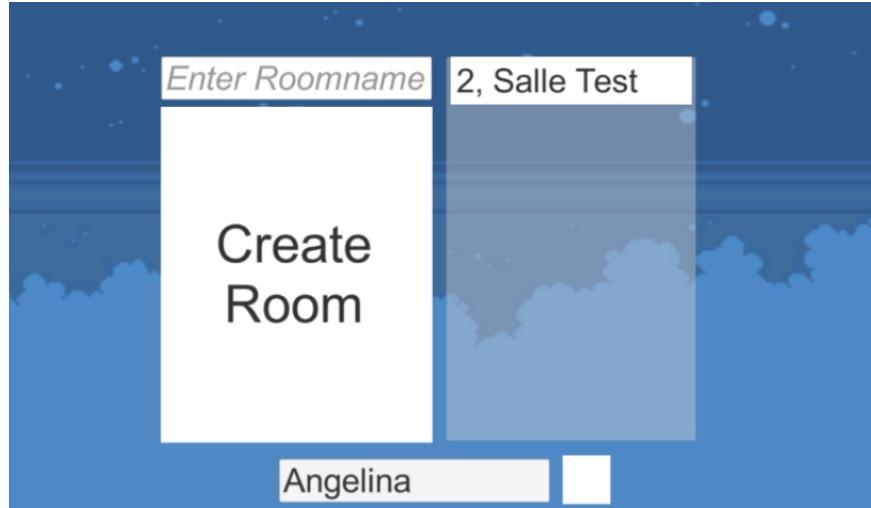


FIGURE 17 – Affichage de la room du joueur1 sur l'écran du joueur2

cliqué sur le bouton prêt, le joueur 1 peut lancer la partie en cliquant sur "start game", instantiant un joueur pour chaque personne.



FIGURE 18 – Menu à l'intérieur d'un salon avec le statut yes/no et la sélection de la classe

Une fois la partie lancée, il fallait encore gérer la synchronisation de certaines informations entre les deux joueurs, notamment le stress qui est partagé par les deux joueurs. Pour cela, j'ai utilisé la procédure RPC de photon permettant d'envoyer une série d'instruction aux autres joueurs présent sur la salle à exécuter. Cela permet notamment à ce que tous les joueurs du salon voient leur stress augmenter en cas de chute de l'un des joueurs par exemple.

Cette partie a été, pour moi, une des plus compliquées à faire depuis le début du projet, car elle demandait des connaissances dans un sujet que je ne maîtrisait pas du tout et qui nécessitait des connaissances que je n'avais pas encore. J'ai donc du implémenter mes fonctions à l'aide

de différents tutos sur Internet et en lisant la documentation présente sur le site de photon pour avancer pas à pas dans la compréhension de photon.

Animation du Dash

Enfin je me suis occupé d'ajouter aux classes *Classique* et *Bouncy* les animations manquantes c'est-à-dire celles des dash. Encore une fois, c'est Hugo qui s'est occupé de créer les tileset. La classe *Classique* possède ainsi huit animations de plus : une pour le dash avant/arrière, une pour les dash diagonaux vers le haut, une pour les dash diagonaux vers le bas, une pour le dash vers le haut et une pour celui vers le bas. La classe *Bouncy* quant à elle possède une animation de plus : celle animant le dash vers l'avant/arrière.

Prochaine Soutenance

Pour la prochaine soutenance qui demandera le moins de travail parmi toutes, je vais devoir d'implémenter le mode de jeu solo, c'est-à-dire que le joueur ayant créé une partie pourra choisir deux personnages avec deux classes différentes et devra jouer les deux personnages. Il pourra changer entre les deux personnages à tout moment de la game. Il me faudra également résoudre les différents bugs que nous allons sûrement découvrir assez couramment.

Angelina

Début du deuxième rendu

Pour cette soutenance, je devais m'occuper d'implémenter la génération aléatoire de salles, le système des salles puzzle capteurs. Enfin finir entièrement les bibliothèques des deux premières zones. Je souhaitais débuter par la création du système de capteur pour par la suite le donner suffisamment tôt à Mathéo pour que cela lui serve de base pour la création du système de torche.

Correction de bugs

Avant d'attaquer ce travail, j'ai commencé par améliorer certaines mécaniques déjà réalisées pour la première soutenance. À présent, lorsque le joueur aura la clé en sa possession, celle-ci verra son box collider détruit pour éviter d'avoir des bugs tel que le joueur pouvant marcher sur la clé. Pour ce qui est des salles boutons, le bouton aura dorénavant une seconde supplémentaire passant de 7 secondes à 8 car certaines salles étaient vraiment impossible à faire et il aurait fallu rendre la salle très facile à finir ce qui enlèverait du challenge au joueur. Pour ce qui est du dialogue, il est désormais possible de parler qu'une seule et unique fois au marchand. Auparavant, si les items vendus ne convenaient pas, le joueur pouvant ouvrir et fermer la boutique sans limite jusqu'à ce que la génération aléatoire d'item lui donne l'item qu'il souhaitait. Correction également d'un bug avec le dialogue des tableaux des salles histoires, avant, une fois le dialogue lancé, il restait sur l'écran du joueur tant qu'il ne finissait pas la lecture de tout le dialogue. Maintenant, si le joueur s'éloigne du tableau, le dialogue se fermera automatiquement et le joueur devra reprendre la lecture du début s'il reparle au tableau.

Système capteur

Pour le système capteur, pour que le joueur déverrouille la porte, il doit activer 6 capteurs. Les capteurs fonctionnent ainsi : de base, ils apparaissent bleu clair, lorsque le joueur touche un capteur, c'est-à-dire lorsque le collider du joueur entre en collision avec le collider du capteur, le capteur devient violet et le dash du joueur est réinitialisé. Lorsque tous les capteurs sont activés, la porte est déverrouillée. Je prévois pour la prochaine soutenance pour rendre ce système jouable en coopération, les capteurs du premier joueur ouvriront la porte du deuxième joueur et vice-versa. (Illustration des capteurs activés et désactivés plus bas.)

Génération aléatoire

Pour la génération aléatoire, j'ai réfléchi à un système utilisant des listes comportant tous les index de scènes s'actualisant tout au long du jeu. Toutes les scènes auront un index bien défini dans les builds settings qui ne changeront pas.

Actuellement, il y aurait 65 scènes de prévu. Lorsque le joueur lancera le jeu, il sera sur la scène 0, c'est-à-dire la main menu et chaque fois que le joueur ira dans une nouvelle zone il sera téléporté sur la scène "jump x.1". Donc, lorsque le joueur lancera une partie tout au début, il commencera son aventure sur la scène "jump1.1" soit l'index 1 dans les builds settings. Le système de changement de salle est comme précédemment une fleur avec laquelle il faut rentrer en contact. Chaque fleur aura le script "Don't destroy on load", cela implique que toutes les fleurs seront initialement

Menu démarrage	ZONE 1	ZONE 2	ZONE 3	ZONE 4
0	total 12	total 12	total 12	total 12
	1 jump1.1	15 jump2.1	30 jump3.1	45 jump4.1
	2 jump1.2	16 jump2.2	31 jump3.2	46 jump4.2
	3 key1.1	17 key2.1	32 key3.1	47 key4.1
	4 key1.2	18 key2.2	33 key3.2	48 key4.2
	5 captor1.1	19 captor2.1	34 captor3.1	49 captor4.1
	6 captor1.2	20 captor2.2	35 captor3.2	50 captor4.2
	7 torch1.1	21 torch2.1	36 torch3.1	51 torch4.1
	8 torch1.2	22 torch2.2	37 torch3.2	52 torch4.2
	9 button1.1	23 button2.1	38 button3.1	53 button4.1
	10 button1.2	24 button2.2	39 button3.2	54 button4.2
	11 shop1	25 shop2	40 shop3	55 shop4
	12 campfire1	26 campfire2	41 campfire3	56 campfire4
		27 history2	42 history5	57 history8
	13 history1	28 history3	43 history6	58 history9
	14 room_choice	29 history4	44 history7	59 history10
				60 last_scene
				61 defi1
				62 defi2
				63 defi3
				64 defi4
				65 crédit

FIGURE 19 – Tableau récapitulatif des index des scènes du jeu

présentes sur les scènes "jump x.1" et seront téléporté sur chaque scène suivante générée aléatoirement. Pour ce qui est du multijoueur, la fleur du premier joueur comporte un booléen qui est sur false de base et devient true lorsque le deuxième joueur a atteint sa fleur. Lorsque ce booléen est vrai et que le premier joueur atteint sa propre fleur, le changement de salle aura lieu et évidemment le booléen repassera à false. Chaque zone aura sa propre fleur qui correspondra au paysage de la zone (fleur de forêt, fleur de neige, fleur de caverne) et chaque fleur aura un script avec deux listes, une liste avec les index des salles puzzles et une autre avec les index des salles histoires.



FIGURE 20 – Fleurs des zones 1 2 et 3.

Pour chaque zone, le joueur pourra parcourir entre 10 et le total des salles, c'est un changement par rapport à ce que j'avais dit dans le rendu de la première soutenance, car 8 salles faisaient trop peu à mon goût. Chaque fois qu'il est possible, les salles histoires, feu de camp et le shop auront 1 chance sur 5 de sortir. Pour les trois premières salles, uniquement des salles histoires ou puzzles peuvent sortir. Au-delà de ces trois-là, il sera possible d'avoir feu de camp ou shop. Si le joueur n'a pas eu de feu de camp au bout de 7 salles, celle-ci sortira automatiquement et si le joueur n'a pas eu de shop au bout de 8 salles elle sortira obligatoirement. Il n'y a pas d'obligation sur les sorties des salles histoires, chaque aventure sera unique et si le joueur souhaiterait découvrir l'histoire dans son intégralité, il devra recommencer le jeu. Il n'y a pas d'obligation sur les sorties des salles histoires, chaque aventure sera unique et si le joueur souhaiterait découvrir l'histoire dans son intégralité, il devra recommencer le jeu. Pour la zone suivante, la scène chargée en première sera toujours la "jump x.1". Chaque fois qu'une salle est chargée, son index est retiré de la liste et donc le joueur ne pourra jamais avoir deux fois la même salle.

Petite particularité de la zone 1, celle-ci ne comporte qu'une seule salle histoire, dans cette salle

le tableau annonce au joueur que pour la suite de son aventure, il devra choisir entre la zone 2 et la zone 3, c'est-à-dire qu'au total le joueur ne va parcourir que 3 zones et s'il voudrait toutes les faire, il devra recommencer le jeu. La salle qui sera après la première salle histoire sera la "room choice" et celle-ci n'aura pas une sortie mais bien deux, une allant dans la zone 2 et l'autre allant dans la zone 3. (visuel de la salle plus bas.)

Musique

Modification de l'audio manager, auparavant, il y avait un audio manager sur chaque scène avec une seule musique. Maintenant il n'y a qu'un seul audio manager qui sera généré sur la toute première scène du jeu c'est-à-dire notre bon main menu avec le script "Don't Destroy On Load" qui comportera une playlist de toutes les musiques du jeu. Avant, chaque fois que le joueur changeait de salle la musique recommençait. A présent, la musique sera continue et le script de l'audio manager testera juste l'index de la salle. Si l'index correspond à une salle histoire la musique joué sera celle des salles histoires, si c'est une salle défi ce sera la musique des salles défis et sinon il testera si l'index est compris dans les intervalles [1,12] ou [15,26] ou [30,41] ou [45,56]. Selon l'intervalle dans lequel l'index est compris, la musique de la zone correspondante sera jouée.

Création de toutes les salles des zones 1 et 2

Mon objectif majeur de cette semaine était de créer toutes les salles inférieures à l'index 30 si on se réfère au tableau des index des salles. Les salles qui étaient déjà créées étaient : "Main menu", "jump 1.1", "jump 1.2", "key 1.1", "button 1.1", "shop1" et "history 1".

Ici la salle "captor 1.1" avec des capteurs désactivés en bleu et des capteurs activés en violet.



FIGURE 21 – Salle Captor1.1

Ici la "room choice" qui est après la salle "history1" qui mène soit à la zone 2 "sentiers enneigés" la sortie du haut, soit la zone 3 "cœur de la montagne" la sortie du bas.

Viennent ensuite toutes les autres salles que j'ai du faire. Pour la deuxième zone, j'ai utilisé un asset gratuit supplémentaire [Kauzz Winter Forest by Kauzz](#) qui vient du même créateur que l'asset de la zone 1. J'ai utilisé certaine décos telle que la maison ou bien encore le bonhomme de neige pour compléter avec l'asset de la zone 2. J'ai également réalisé un mélange entre les deux backgrounds de l'asset qui était prévu de base et de celui-ci. J'ai utilisé trois layers de sapin du nouvel asset et je l'ai rajouté sur le background originel du ciel violet.

Pour créer des salles jouables en multijoueur, une fois que j'avais fini de faire la salle du joueur 1 positionnée en coordonnées x = 0 y = 0, je créais une deuxième salle identique à la première sur



FIGURE 22 – Room Choice

la même scène, mais de coordonnées $x = 0$ et $y = 40$.



FIGURE 23 – Captor 1.2 et Button 1.2

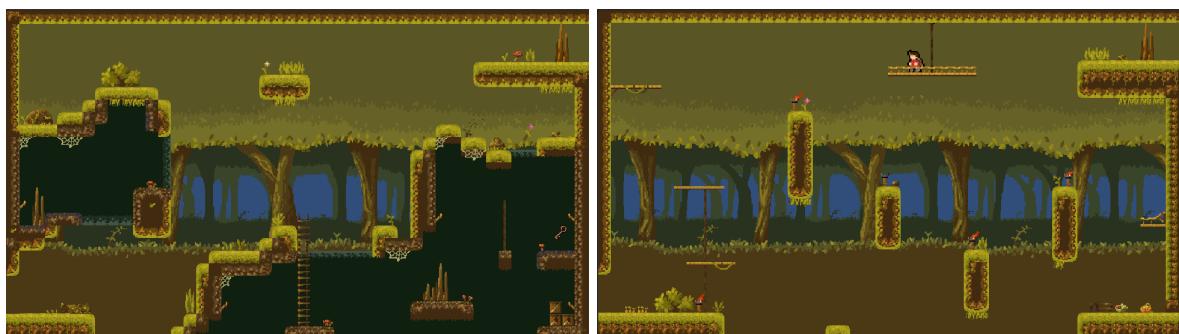


FIGURE 24 – Key 1.2 et Torch 1.1

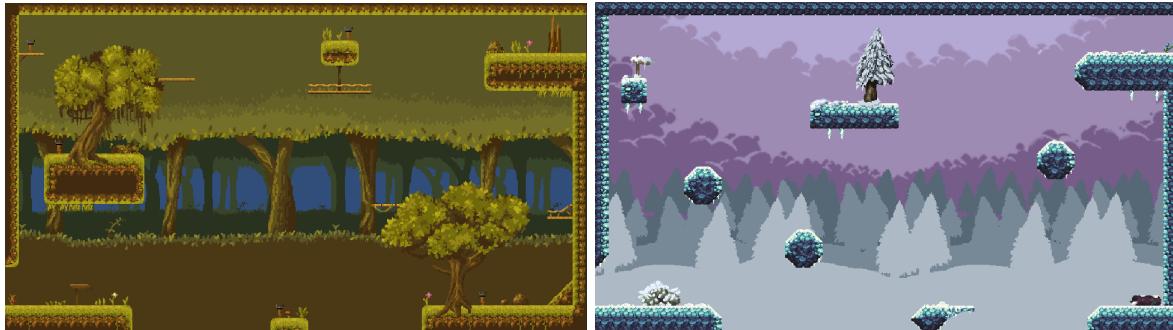


FIGURE 25 – Torch 1.2 et Jump 2.1



FIGURE 26 – Key 2.1 et Key 2.2



FIGURE 27 – Button 2.1 et Button 2.2



FIGURE 28 – Captor 2.1 et Captor 2.2



FIGURE 29 – Torch 2.1 et Torch 2.2

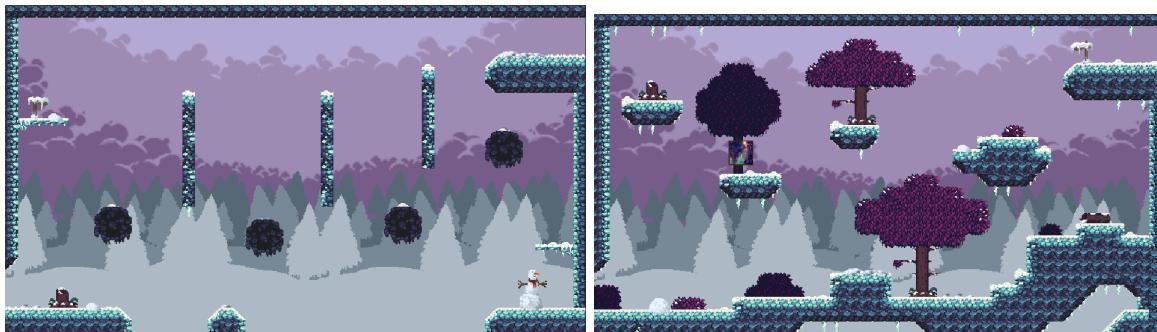


FIGURE 30 – Jump 2.2 et History 2

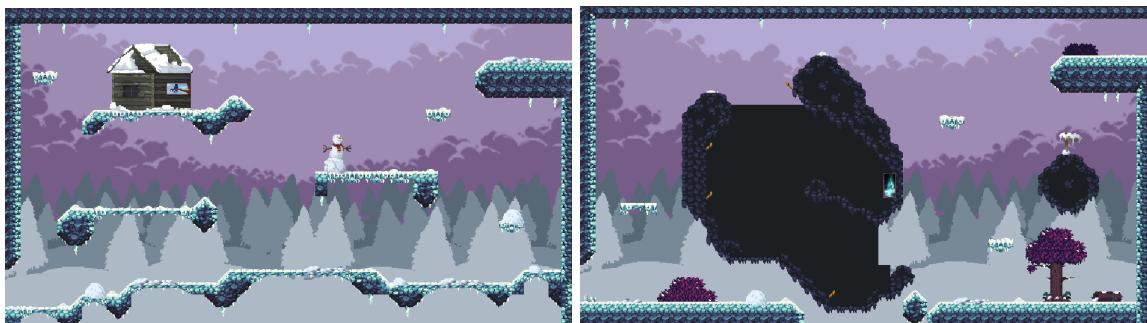


FIGURE 31 – History 3 et History 4

Montage vidéo

Pour réaliser la vidéo de la soutenance, nous avons dû recourir à de nouveaux logiciels. Pour enregistrer l'écran et le gameplay, j'ai utilisé le logiciel gratuit OBS Studio. Son utilisation était assez instinctive évitant des soucis et des difficultés incongrues.

Pour le tournage, nous étions en équipe en vocal. Je filmais tout en partageant mon écran en jouant avec un coéquipier tandis le reste du groupe étaient devenu des directeurs de production. Suite à cela, j'ai pu m'attaquer aux montages, armée de toutes les runs qu'on a pu filmer et d'un script que nous avons rédigé tous ensemble. Pour faciliter le montage, nous avons délimité chaque partie présentée en un nombre précis de seconde. Pour le montage, j'ai pu découvrir le logiciel de ShotCut qui est également un logiciel disponible gratuitement. J'ai avancé dans cette aventure

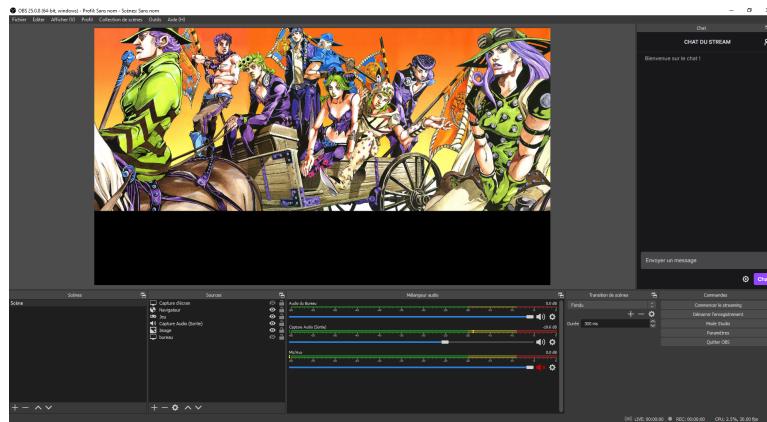


FIGURE 32 – Screenshot OBS Studio

soutenue par des vidéos tutos sur youtube. Enfin, j'ai pu transmettre la vidéo montée à mon groupe pour qu'il puisse s'enregistrer avec les bonnes images sous les yeux pour avoir le rendu le plus propre possible. Pour les introductions personnelles, nous nous sommes mis d'accord pour que chacun filme la sienne de son côté et qu'elle soit ajouté à la fin au montage au début de la vidéo.

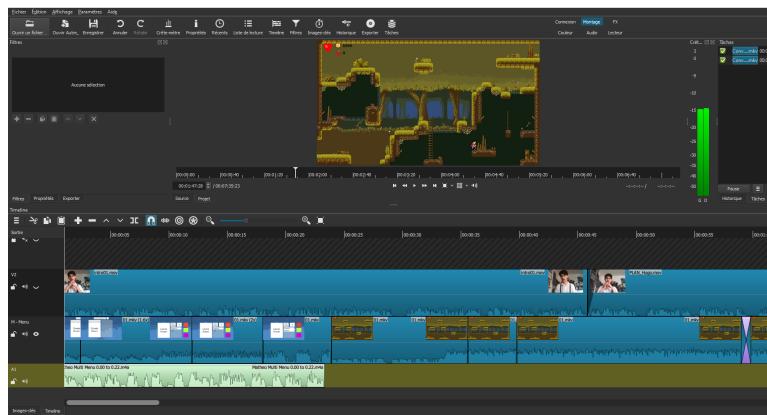


FIGURE 33 – Screenshot Shotcut

Prochaine Soutenance

Pour la prochaine soutenance, je vais de voir m'occuper d'implémenter les salles défis et créer entièrement les bibliothèques des zones 3 et 4. Évidemment la chasse aux bugs sera inévitable et j'ai déjà plusieurs améliorations possibles de ce que j'ai déjà implémenté en tête.

Antoine

Début du deuxième rendu

Mon rôle dans cette seconde soutenance a été d'implémenter les effets visuels du stress, finir les objets du jeu (en collaboration avec Hugo) et pour finir, faire l'intelligence artificielle. Pour l'ensemble de mes tâches, je me suis appuyé sur des tutoriels et certaines documentations internet pour avoir des notions sur le travail que je devais faire. Ce milieu de projet fut assez compliqué pour moi, car je n'avais jamais abordé la notion d'intelligence artificielle, et il a fallu me documenter afin de trouver des pistes pour débuter sa conception.

Effets visuels du stress

Dans le but de créer des effets visuels pour le stress, nous avons décidé au début avec l'ensemble du groupe, de faire du flou et de saturer la musique au fur et à mesure que le stress augmentait. Cependant, par souci d'esthétisme, nous avons préféré abandonner l'idée du blur, car, lorsque nous appliquions celui-ci sur l'écran du joueur, cela dénaturait beaucoup le jeu. Pour remplacer celui-ci, nous avons donc choisi d'utiliser des filtres qui obscurcissent l'écran du joueur au fur et à mesure. Nous avons choisi de mettre 3 stades pour ces filtres, le premier est transparent, nous voyons le jeu parfaitement bien (c'est dans le cas où notre stress est inférieur ou égal à 100). Le second filtre obscurcit un peu plus l'écran et concerne la plage de stress comprise entre 100 et 150. Le dernier filtre obscurcit l'écran et ajoute un effet rouge, il se met lorsque le stress est supérieur à 150. Parallèlement aux filtres, le son de la musique du jeu sera de plus en plus distordu. Sa distorsion comment à 100, tout comme le second filtre, est augmentée à 150 avec l'arrivée du troisième filtre.



FIGURE 34 – Filtre numéro 1



FIGURE 35 – Filtre numéro 2

Récapitulatifs de l'ensemble des objets

Concernant les objets pour cette soutenance, je me suis occupé, en collaboration avec Hugo, de la finition de tous les objets. Me concernant, je vais vous lister l'ensemble des objets que j'ai faits dans ce projet (pour connaître leurs effets, je vous invite à relire le rapport de projet de la première soutenance).

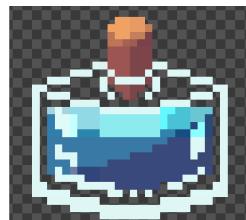


FIGURE 36 – Potion de Vitesse



FIGURE 37 – Guide de l'épicier

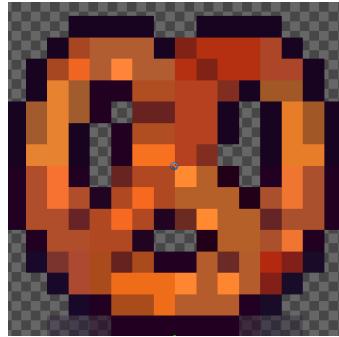


FIGURE 38 – Pretzel d'ur chance



FIGURE 39 – La Tarte

Intelligence artificielle

Concernant l'intelligence artificielle que je devais faire pour cette soutenance, elle m'a paru infaisable au premier abord. En effet, j'ai passé plus de deux journées complètes pour trouver un système fonctionnel qui puisse refaire les mouvements de notre personnage. Tout d'abord, ma première idée fut de stocker dans une liste les positions successives de mon personnage afin que l'intelligence artificielle les suive avec un certain décalage et cela permettrait de faire en sorte que l'intelligence effectue les mêmes mouvements que mon personnage avec un certain décalage. Cependant, cette première idée fut abandonnée, car elle ne marchait pas du tout. J'ai donc pensé que c'était la structure de liste qui était embêtante, car cela me faisait une liste infinie de positions du joueur et donc que la structure de donnée n'était pas adaptée à mes besoins. Je me suis alors tourné vers les files, car cette structure était parfaitement adaptée à mes besoins, mais encore une fois, mon intelligence artificielle ne faisait pas du tout ce que je demandais et allais vers mon joueur par le chemin le plus rapide au lieu de reproduire les mouvements de celui-ci. De ce fait, j'en ai donc conclu que ce n'était pas la structure de données qui était erronée, mais plutôt ma façon de faire l'intelligence artificielle. Par la suite, je suis retourné faire des recherches sur l'internet afin de trouver des idées qui permettraient d'aboutir à mon intelligence artificielle. Ainsi, j'ai trouvé une méthode qui permet d'effectuer une update d'une fonction (autre que la fonction update prédefinis) sur un intervalle de temps et ainsi n'effectue l'update que toute seconde ou demi-secondes (intervalle de temps que je devais définir). J'ai donc créé une fonction qui permet de façonner un (path = chemin) jusqu'à mon joueur, et parallèlement à ça, je crée une liste de chemins (jusqu'à mon joueur) et j'update le path actuel toutes les 0.14 secondes avec les paths que je mets dans ma liste. Cette méthode m'a permis d'aboutir à une intelligence artificielle complète qui reproduit à l'identique les mouvements de notre joueur.

Je vais maintenant vous montrer comment fonctionne l'intelligence artificielle en pratique, grâce à des exemples visuels. Premièrement, j'ai créé un layer appelé "obstacle" qui définit dans chaque salle les éléments qui doivent être considérés par l'intelligence artificielle comme des obstacles qu'il doit prendre en compte dans son path et donc contourner. Voici un exemple de l'une des salles :



FIGURE 40 – Fonctionnement de l'intelligence artificielle

Sur l'image ci-dessous, nous pouvons remarquer de nombreux éléments graphiques inhabituels sur la scène. Tout d'abord, la zone bleue représentée sur la scène est la zone où l'intelligence artificielle a le droit de circuler et au contraire, les éléments qui n'ont pas de filtre bleu sur eux sont les éléments considérés comme des obstacles par l'intelligence artificielle. Sur l'image ci-dessous, nous pouvons remarquer de nombreux éléments graphiques inhabituels sur la scène. Tout d'abord, la zone bleue représentée sur la scène est la zone où l'intelligence artificielle a le droit de circuler et, au contraire, les éléments qui n'ont pas de filtre bleu sur eux sont les éléments considérés comme des obstacles par l'intelligence artificielle. Cependant, si l'intelligence artificielle touche notre joueur, celle-ci va disparaître et notre joueur va mourir (la vraie mort n'étant pas actuellement gérée par notre jeu, celui-ci va être téléporté au spawn point). La seconde manière de faire disparaître l'intelligence artificielle sans mourir, est de faire diminuer son stress en dessous des 190 points de stress, c'est-à-dire que pour que le fantôme disparaîtse, le joueur doit avoir un montant de stress inférieur ou égal à 189.

Conclusion pour cette soutenance

Pour conclure, afin d'effectuer le travail pour cette soutenance, j'ai dû me documenter un maximum et je pense avoir passé une grande majorité du temps à le faire. De plus, j'ai pu apprendre de nombreuses choses grâce à celle-ci et j'ai hâte de voir la fin de notre jeu.

Conclusion

Pour conclure cette seconde partie de projet, nous pensons avoir bien réussi à avancer sur notre jeu et avons accompli tous les objectifs que nous nous étions fixés dans le cahier des charges. Bien sûr nous avons rencontré des difficultés, mais ce projet reste toujours aussi intéressant et permet une approche de travail différente des cours "normaux", plus libre où chacun se fixe ses objectifs. Nous pensons avoir presque fini notre jeu, et n'avons plus que quelques fonctions à remplir pour que notre jeu soit fonctionnel.