



EPITA

PROJET JEU VIDÉO SUP S2

Rapport de soutenance finale



Projet : Best Of Ten

Groupe : Less An Year

Jules RAITIERE-DELSUPEXHE

Mathieu EVEN

Mathéo CRESPEL

Lucas BESNARD

Janvier 2022 - Juin 2022

Table des matières

1	Introduction	3
1.1	Les membres du groupe	4
1.2	Préambule du projet	4
2	Origine, nature du projet et objet de l'étude	5
3	État de l'art	5
3.1	Historique	5
3.2	Concurrents	6
3.3	Nos particularités	7
4	Aspects concrets du projet	7
4.1	Règles du jeu	7
4.2	Gameplay	8
4.3	Déroulement du développement	9
4.4	Répartition des tâches	11
4.5	Planning	12
4.6	Aspects économiques	13
5	Avancement du projet	13
5.1	Multijoueur	15
5.1.1	Synchronisation	16
5.2	Mode contre-la-montre	18
5.3	Intelligence artificielle	19
5.3.1	Fonctionnement	20
5.3.2	Problèmes rencontrés	20
5.3.3	Implémentation	21
5.3.4	Zones de zones et liste de zones	22
5.4	Interface utilisateur	22
5.5	Synchronisation des chronomètres et du classement	26
5.6	Choix de l'apparence du joueur	27
5.7	Création des parcours	28
5.8	Modélisation 3D	30
5.9	Décors	32
5.10	Communication	35

5.11	Système des 3 manches	39
5.12	Interaction avec les objets	40
5.13	Bonus	42
5.14	Sons	43
6	Récit de la réalisation	44
6.1	Jules	44
6.2	Mathieu	45
6.3	Mathéo	46
6.4	Lucas	46
6.5	Nos peines	47
6.6	Nos joies	48
7	Points d'améliorations	48
7.1	Notre jeu	48
7.2	Notre groupe	49
8	Conclusion	50

1 Introduction

Le groupe « Less An Year » est fier de vous présenter ce rapport de projet qui détaille l'évolution de notre jeu ainsi que son état final. Le développement de notre jeu intitulé « Best Of Ten » aura duré près de 5 mois. Durant ces mois de travail, nous avons beaucoup appris que ce soit dans le domaine de la réalisation de jeu vidéo, mais également sur le plan humain pour avancer en groupe. Grâce à notre jeu, remplacez des moments d'ennuis ou de morosité par des moments de détente et de joie en quelques clics !

Notre jeu est principalement inspiré de Fall Guys. C'est un enchaînement de parcours d'obstacles que vous devez franchir le plus rapidement possible. Chaque partie est composée de 3 manches, pour pouvoir gagner la partie entière, nous faisons la somme de vos temps sur les 3 manches précédentes. Si vous n'êtes arrivé que 8^{ème} à une manche, pas de panique ! De très bons résultats aux 2 autres manches vous permettront d'être dans le haut du classement. Durant ces 3 manches, vous aurez l'occasion de découvrir nos 3 décors : la forêt, la banquise et le désert. Dans chacun d'eux, découvrez un parcours inédit et plein de rebondissements. Notre jeu est multijoueur, vous avez la possibilité de rejoindre vos amis ou des inconnus dans une « salle » virtuelle afin de vous affronter simultanément. Vous pouvez aussi vous entraîner seul(e) sur chacune de nos cartes et battre votre fantôme ! Dernière chose, imaginons que vous êtes 6 amis à jouer ensemble, vous allez vite remarquer que vous serez 10 joueurs en lançant la partie. En effet, notre jeu se joue toujours à 10 joueurs, s'il n'y a pas assez de joueurs humains, des intelligences artificielles viendront avec vous. En plus de tout ça, des bonus viendront pimenter vos parties pour un maximum de plaisir.

Ce document contient une description générale du jeu, de ses origines, de ses règles, une description détaillée de l'avancement du projet, notre ressenti global sur le déroulement du projet ainsi que ce qui aurait pu être amélioré dans nos méthodes de travail et dans le jeu.

1.1 Les membres du groupe

Notre groupe est composé de 4 personnes : Jules RAITIERE-DELSUPEXHE, Mathieu EVEN, Mathéo CRESPEL et Lucas BESNARD.

- Jules : Organisé et assidu, il sera le chef de groupe du projet. Il apportera rigueur et clarté à la répartition des tâches. En tant que responsable, nous comptons sur sa logique et sa bonne perception des problèmes pour faire avancer le projet dans la technique comme dans l'organisation.
- Mathieu : Ayant réalisé quelques projets informatiques, il avait déjà une idée du déroulement des projets en général. Il va donc pouvoir apporter son expérience au groupe. Nous pourrons compter sur sa persévérance pour atteindre les objectifs fixés.
- Mathéo : Adorant les vieilles consoles et les vieux jeux, il a déjà fait de nombreux projets sur ces dernières, lui donnant une expérience sur la réalisation de jeux vidéo. Même si ce projet est peu semblable à ce qu'il a pu faire auparavant, il saura apporter son calme, ses idées et ses conseils pour le mener à bien.
- Lucas : Novice en informatique, il n'avait pas touché à une seule ligne de code avant d'intégrer EPITA. Cependant, sa détermination et sa motivation lui ont permis de faire une progression exponentielle. Nous comptons sur sa rigueur et sa capacité d'apprentissage pour tirer le projet vers le haut.

1.2 Préambule du projet

Nous avons réalisé un jeu de plateforme disponible pour Windows. Aujourd'hui, de nombreux jeux nécessitent des heures de pratique pour prendre en main tous les éléments et espérer pouvoir battre des adversaires. Ils nécessitent du temps, des dizaines de Gigaoctets de mémoire et un ordinateur puissant.

De notre côté, nous avons réussi à créer un jeu accessible à tous et rapide à comprendre. Cela est possible grâce à des par-

ties de détente entre amis ou avec des inconnus pendant une durée de 10 minutes environ. Le joueur peut se divertir grâce à 2 modes de jeu différents : contre-la-montre et multijoueur. Il évoluera dans 3 univers différents (forêt, désert et banquise), son objectif est de franchir un enchaînement d'obstacles le plus rapidement possible. Chaque partie est divisé en 3 manches et contient toujours 10 joueurs (IA comprises). C'est un jeu compétitif avec un seul gagnant : celui-ci doit obtenir le meilleur temps face à ses adversaires lors des différentes épreuves.

2 Origine, nature du projet et objet de l'étude

L'idée de Best Of Ten vient des couloirs de EPITA, de nombreux étudiants jouent entre eux, aux intercours, à un jeu de plateforme de même type : Stumble Guys qui est un clone mobile de Fall Guys. Les élèves s'amusent entre eux et rigolent, et c'est ce que l'on recherche dans la conception de notre jeu : du fun et de l'amusement. Beaucoup de jeux auxquels nous avons joué nous ont inspirés pour les graphismes en ombrage de celluloïd (*Cell Shading*) : le *Cell Shading* est une technique graphique qui consiste à représenter des décors de la réalité avec des graphismes cartoons. Après plusieurs idées ajoutées ou écartées, notre jeu a pris forme avec des originalités telles que des objets intégrés au décor avec lesquels on peut interagir ou encore des bonus récupérables sur le parcours.

3 État de l'art

3.1 Historique

Les premiers jeux de plate-forme sont nés dans les années 80, on peut notamment citer Donkey Kong proposé en 1981 par Nintendo qui est le premier jeu permettant de sauter au-dessus d'obstacles et de trous. Les jeux multijoueurs de cette même catégorie

n'ont pas tardé à faire leur apparition : Mario Bros publié par Nintendo en 1982 ou Bubble Bobble par Taito en 1986 permettait de jouer à deux joueurs sur le même écran. Roblox est le premier jeu de plate-forme en ligne sorti en 2005 dans sa version définitive. L'objectif est de construire un jeu pour qu'il soit visité par les autres joueurs. Les joueurs sont libres de construire ce qu'ils souhaitent et de partager leur création avec le reste de la communauté du jeu. Les jeux populaires sont les courses d'obstacles. Il y a donc une multitude de mini-jeux au sein de Roblox qui ont la particularité d'avoir des graphiques très simples et un environnement réalisé avec des blocs. Ce jeu nous a inspiré notamment pour créer notre environnement graphique, mais aussi dans le principe du jeu lui-même qui est le parcours d'obstacles en multijoueur.

3.2 Concurrents

- ⇒ Un des principaux jeux de ce type existant est Fall Guys qui est un jeu vidéo multijoueur fortement inspiré des jeux télévisés de course d'obstacles comme Total Wipeout et Ninja Warrior où 60 joueurs doivent surmonter des défis. Son principal point fort est de proposer plusieurs modes de jeu, plusieurs épreuves et donc de se diversifier dans le gameplay proposé. On y trouve des parcours dans lesquels il faut être le premier à l'arrivée, mais aussi des niveaux où il faut rester le plus longtemps sur une plateforme sans tomber. Lors de chaque épreuve, un classement est effectué et les derniers sont disqualifiés. Au bout d'un certain nombre d'épreuves, le seul survivant devient le gagnant.



FIGURE 1 – Fall Guys



FIGURE 2 – Mario Kart

► Mario Kart est l'un des jeux qui nous a inspiré pour le système de bonus et de classements. L'objectif principal de Mario Kart est simple : le joueur doit parcourir le plus rapidement possible des circuits dans l'optique de franchir la ligne d'arrivée en premier. Pour se faciliter la tâche, le joueur a la possibilité d'utiliser de nombreux bonus. Ils peuvent être offensifs, défensifs ou encore stratégiques, permettant au joueur de bénéficier d'une accélération supplémentaire, ou encore d'en-jammer un obstacle pour gagner du temps au tour.

3.3 Nos particularités

Pour se distinguer des jeux cités ci-dessus, notre jeu se joue systématiquement à 10 joueurs, par exemple, s'il n'y a que 4 joueurs humains, alors 6 bots viennent compléter la partie. Une de nos particularités est que nos personnages pourront interagir avec le décor et ainsi déplacer des objets (ou autres) pour ralentir la progression des adversaires. De plus, des bonus récupérables lors de la progression du joueur sur les cartes lui offre des avantages décrits plus tard dans ce rapport.

4 Aspects concrets du projet

4.1 Règles du jeu

Chaque partie se déroule sur 3 manches. 10 joueurs sont présents pendant les 3 manches. Ces manches sont des courses d'obstacles.

Le but est d'arriver le plus rapidement possible à la ligne d'arrivée sur les 3 manches : À la fin d'une partie, les temps obtenus par le joueur sur les 3 manches sont additionnés pour donner son temps final du joueur. Ce dernier est alors placé dans un classement en fonction des temps des autres joueurs, ce qui donnera le vainqueur de la partie.

Toutes les règles sont permises pour arriver le plus rapidement possible jusqu'à la ligne d'arrivée.

4.2 Gameplay

Déplacements :

- Le joueur peut se mouvoir dans 4 directions : en avant (avec la touche « z »), en arrière (avec la touche « s »), sur la gauche (avec la touche « q ») et sur la droite (avec la touche « d ») ou les flèches directionnelles.
- Il peut sauter (avec la barre espace) et courir (avec la touche « Maj » gauche, en la maintenant enfoncee).
- Tous les joueurs possèdent les mêmes capacités de déplacements et la même vitesse au départ.

Interactions :

- Le joueur peut interagir avec le décor. Il peut soulever des objets, les déplacer et les lancer (avec la touche « e »). Le type d'objet dépendra du type de terrain et de l'épreuve.
- Il peut aussi interagir avec les bonus en les collectant lors de sa partie, ainsi ce bonus ne sera plus disponible pour les autres joueurs de la partie.

Bonus :

Comme dit plus tôt, le joueur peut ramasser des bonus. Ces bonus ont uniquement des effets positifs pour le joueur et ils lui sont profitables. Types de bonus :

- Boost de vitesse : il permet d'améliorer temporairement la vitesse de déplacement du joueur.
- Temps : il améliore le temps final du joueur de 2 secondes par bonus récolté.
- Boost de saut : il permet d'améliorer temporairement la hauteur de saut du joueur.

Utilisation du décor :

Comme dit plus tôt, le joueur peut utiliser certains objets du décor à son avantage : il peut prendre certains objets pour les balancer sur ces adversaires et ralentir leur progression.

Victoires et défaites :

Avec tous les aspects vu plus haut, un joueur se trouvant dernier

pendant une course peut revenir au premier plan s'il excelle dans les deux autres manches. Ainsi, c'est bien la somme des scores des 3 manches qui permettent de désigner le vainqueur, gagner une manche ne signifie en rien gagner la partie. Cet équilibre est volontaire.

Environnement :

- Le joueur peut évoluer dans trois cartes différentes : désert, banquise et forêt.
- Les obstacles sont très variés : le décor en lui-même peut représenter un obstacle, il y a aussi des faucheuses, des murs mouvants, des objets à pousser, des canons, des bascules, etc.

4.3 Déroulement du développement

Nous avons développé ce jeu à l'aide du logiciel Unity, du langage C# et du logiciel de modélisation 3D Blender.

Un serveur Discord, un dépôt Git, un mail professionnel (less.an.year@gmail.com), un projet Unity commun (Unity Teams puis Plastic SCM) ainsi que des documents L^AT_EX collaboratifs (Overleaf) nous ont permis de travailler efficacement en groupe. Ainsi, au-delà du partage des tâches plutôt « techniques », chacun des membres du projet a participé aux tâches plutôt « administratives ». Ainsi, pour l'ensemble des soutenances, rapport de soutenance et autres documents, chacun a apporté sa pièce pour construire une finalité.

Il nous est naturellement venu à l'esprit de diviser notre projet global en étapes clés qui ont été définis dans notre cahier des charges, celles-ci ont été respectées dans les grandes lignes et nous ont permis de fixer nos priorités. Les voici comme énoncées dans notre cahier des charges :

- ⇒ 1^{er} étape : Avoir deux personnages qui puissent se déplacer sur une surface plane (position, saut et course) avec une caméra 3^{ème} personne. Avoir un multijoueur fonctionnel, c'est-à-dire avoir deux personnages qui se voient depuis une connexion

Internet. Avoir 8 IA qui se déplacent sur cette même surface qui soit visibles par les autres joueurs.

- 2^{ème} étape : Avoir un menu de départ pour lancer le jeu et apparaître sur la surface. Avoir un chronomètre synchronisé entre les joueurs et affiché sur l'écran. Avoir une zone arrêtant le chronomètre du joueur rentrant dans celle-ci sans impacter l'autre joueur. Avoir le système de manche qui se suivent.
- 3^{ème} étape : Avoir 3 apparences de personnages (*skins*) différentes. Avoir 3 cartes différentes permettant de réaliser une partie complète. Ces cartes devront contenir des obstacles et les joueurs (y compris les IA) réagiront en conséquence. Avoir des animations lors de la victoire ou défaite du joueur ainsi qu'un affichage du classement global de la partie.
- 4^{ème} étape : Avoir des bonus. Avoir 6 cartes différentes. Avoir 3 modes de jeu : mode contre-la-montre où l'on se bat contre son fantôme, mode multijoueur avec des amis et mode multijoueur avec des inconnus.
- 5^{ème} étape : Pouvoir utiliser certains éléments du décor pour ralentir ses adversaires et s'offrir des avantages.

N.B. : Un menu de débogage sera présent et mis à jour pendant l'ensemble du développement pour nous permettre de tester un maximum de cas d'utilisations de notre jeu rapidement.

Cependant, lors du développement du jeu, nous avons chamboulé ces étapes : nous avons par exemple ajouté les obstacles dès le début du jeu ou encore développé seulement 3 cartes, car la réalisation de celles-ci demandent beaucoup de temps. Voici donc comment notre projet s'est réellement déroulé :

- 1^{er} étape : Avoir un personnage qui peut se déplacer sur une surface plane (position, saut et course) avec une caméra 3^{ème} personne. Avoir des animations de déplacements et lors de l'arrivée du joueur. Avoir un multijoueur fonctionnel, c'est-à-dire avoir deux personnages qui se voient depuis une connexion Internet.
- 2^{ème} étape : Avoir un menu de départ pour lancer le jeu et accéder aux fonctions multijoueurs. Avoir une carte avec des

obstacles (en occurrence la carte forêt) et synchroniser ces obstacles en ligne. Avoir des parties avec toujours 10 joueurs.

- 3^{ème} étape : Avoir la possibilité de choisir entre 2 apparences de personnages (*skins*) différentes. Avoir 2 cartes différentes. Avoir le système de 3 manches par partie. Améliorer l'interface utilisateur. Avoir un menu pause, l'affichage d'un chronomètre et du classement. Développer l'IA.
- 4^{ème} étape : Avoir des bonus et utiliser les éléments du décor. Avoir 3 cartes différentes. Avoir 3 apparences de joueurs. Avoir 2 modes de jeu : mode contre-la-montre où l'on se bat contre son fantôme et mode multijoueur. Avoir des IA simulant des joueurs humains sur les 3 cartes. Avoir des sons.

4.4 Répartition des tâches

Comme nous l'avons évoqué précédemment, chaque membre du groupe a eu l'occasion de travailler sur chacun des aspects du projet. Ci-dessous, un tableau contenant une liste non exhaustive des tâches réalisées présente le rôle de chacun. Nous avons fait en sorte de commenter nos travaux, nos codes et de nous retrouver régulièrement afin que la globalité du jeu soit comprise par tous les membres. Ce tableau est une version actualisée de celui fournit lors du cahier des charges.

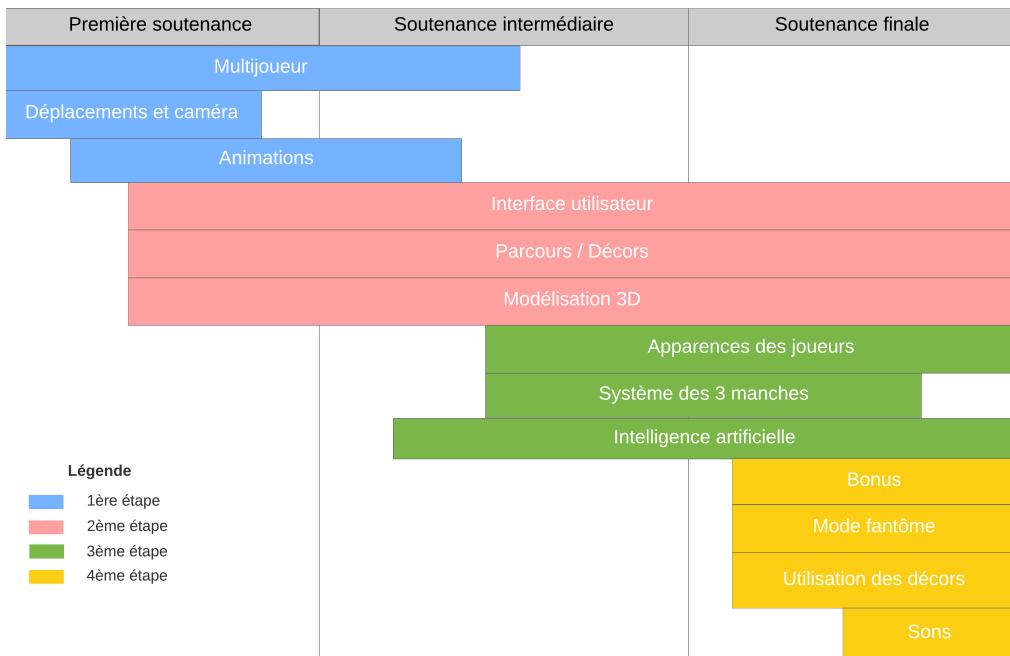
	Mathieu	Jules	Mathéo	Lucas
Multijoueur	R	S		
Déplacements et caméra	S			R
Intelligence Artificielle			R	S
Interface utilisateur	R	S		
Animations (Victoire, défaite et classement)	S			R
Skins		R	S	
Modélisation 3D	S	R		
Cartes		R		S
Système des 3 manches	R			S
Sons		S		R
Mode solo avec fantôme			R	S
Bonus et utilisation du décor	R	S		
Communication (Instagram et site web)		S	R	

R : Responsable

S : Suppléant

4.5 Planning

En complément à la répartition des tâches, un diagramme de Gantt permettant de visualiser le projet au fil des soutenances est fourni ci-dessous. Il reprend les différentes étapes du projet listées dans la deuxième partie de la section 4.3 elles-mêmes divisées en fonction des tâches vu dans le tableau précédent. Ce diagramme correspond au déroulement réel de notre projet.



4.6 Aspects économiques

Ce projet a été développé à moindre frais puisque nous avons réalisé l'ensemble des tâches. Une licence Unity nous est offerte en tant qu'étudiant, des serveurs gratuits sont disponibles pour le multijoueur et le site web.

5 Avancement du projet

Aujourd’hui, le jeu se déroule de la manière suivante : on se connecte à Internet, on lance le jeu puis on arrive sur l’écran d’accueil. Celui-ci a pour arrière-plan un aperçu de nos environnements dans lesquels les parties pourront se dérouler. Sur cet écran d’accueil sont disposés cinq boutons ainsi qu’un champ de saisie. Les boutons permettent de choisir l’apparence de son joueur, de rejoindre une salle, d’en créer une, de quitter le jeu ou lancer une partie hors-ligne :

- Le champ de saisie permet d’afficher un nom à notre joueur, après la première ouverture du jeu, le nom choisi sera sauvegardé et sera réassigné à chaque nouveau lancement du jeu.

- Lorsque l'on veut changer d'apparence, on clique sur le bouton à cet effet qui nous redirige vers une scène où sont disposés deux personnages, on peut donc choisir celui avec lequel on souhaite jouer.
- Lorsque l'on clique sur le bouton pour créer une salle, un champ de saisie est affiché afin de saisir le nom de la salle puis on entre dans la salle et la liste des joueurs est affichée. Il y a un bouton pour revenir au menu principal ainsi qu'un bouton pour lancer la partie.
- Lorsque que l'on souhaite rejoindre une salle, la liste des salles actives est affichée et rejoignable.
- On a également la possibilité de jouer seul sans connexion internet en cliquant sur le bouton offline. Ensuite, vous pouvez choisir la carte sur laquelle vous souhaitez vous entraîner.

Lorsque l'il est prêt, le créateur de la salle peut lancer la partie, notre personnage apparaît donc dans une scène à côté des autres joueurs humains et des intelligences artificielles. Notre personnage peut marcher : en avançant, reculant, en allant à gauche ou à droite grâce aux touches ZQSD. Il peut aussi sauter grâce à la barre espace ou courir en maintenant la touche majuscule gauche appuyée. Nous pouvons aussi bouger la caméra autour du personnage avec la souris. Tous ces mouvements sont accompagnés des animations correspondantes.

Le monde dans lequel nous sommes peut-être différent selon la partie, l'environnement peut changer : forêt, désert ou banquise. Le parcours, peu importe l'environnement, comporte plusieurs obstacles : certains réagissent lorsqu'ils entrent en contact avec le joueur (tourniquets, boules roulantes, etc) tandis que d'autres bougent automatiquement et repoussent le joueur lorsque les deux rentrent en collision (faucheuses, murs mouvants). En haut à gauche de l'écran se situe notre temps sur le parcours et notre place dans la course se situe en haut à droite. Les intelligences artificielles, comme les joueurs, cherchent à finir le parcours au bout duquel se trouve l'arche d'arrivée. Pendant toute la partie, on peut cliquer sur « Échap » pour afficher le menu pause qui nous permet de quitter la partie pour revenir à l'écran d'accueil. Lorsque la

ligne d'arrivée est franchie, le personnage danse pour célébrer sa victoire et son chronomètre s'arrête pour lui indiquer son temps final.

Dans le mode solitaire, l'ensemble des déplacements du joueur sont enregistrés pour que vous puissiez vous battre contre votre propre fantôme. Ainsi, vous poursuivrez votre meilleur temps pour devenir encore meilleur !

Le joueur a également accès à un tableau des scores actualisé en permanence lorsqu'il appuie sur la touche Échap, c'est-à-dire le menu pause. De plus, entre chaque manche, ce tableau des scores est affichée avec un podium des trois premiers joueurs.

5.1 Multijoueur

Pour le bon fonctionnement de notre jeu, le multijoueur est important.

Lors de l'implémentation initiale de celui-ci, Mathieu et Jules se sont formés sur YouTube à l'aide de nombreuses vidéos sur le sujet. Après une première expérience décevante avec Mirror, nous avons finalement choisi d'utiliser Photon, un pack Unity disponible sur l'Asset Store. Bien qu'il soit plus difficile à prendre en main que Mirror, il est beaucoup plus complet et permet plus de choses. Il prend en charge les serveurs dédiés qui nous permettent de jouer ensemble dès l'instant où l'on est connecté à Internet.

Ce système s'articule autour de plusieurs scripts, notamment pour la connexion au serveur et pour la création des salles. Toutes les fonctions sont déjà prédéfinies grâce à Photon, il a donc été facile d'implémenter le multijoueur en mettant bout à bout ces fonctions. Le fonctionnement est le suivant : lorsque le jeu est lancé, il n'y a encore aucune connexion à Internet, c'est seulement lorsqu'on veut créer ou rejoindre une salle que la fonction *PhotonNetwork.ConnectUsingSettings* est appelée pour se connecter au réseau, puis la fonction *PhotonNetwork.JoinLobby* est appelée en arrière-plan. En ce qui concerne la création de salles, la fonction *PhotonNetwork.CreateRoom* permet de le faire simplement. On a maintenant chaque joueur connecté au même serveur et donc un multijoueur fonctionnel.

```

void Start()
{
    Debug.Log("Connecting to Master");
    PhotonNetwork.ConnectUsingSettings();
}

public override void OnConnectedToMaster()
{
    Debug.Log("Connected to Master");
    PhotonNetwork.JoinLobby();
    PhotonNetwork.AutomaticallySyncScene = true;
}

public void CreateRoom()
{
    if (string.IsNullOrEmpty(roomNameInputField.text))
    {
        return;
    }
    PhotonNetwork.CreateRoom(roomNameInputField.text);
    MenuManager.Instance.OpenMenu("loading");
}

```

FIGURE 3 – Différentes fonctions de Photon

5.1.1 Synchronisation

Notre projet étant de plus en plus avancé, il nous est parfois compliqué de résoudre certains problèmes. L'un d'entre eux concernant le multijoueur est la synchronisation de l'ensemble des obstacles et des joueurs. Cette synchronisation est capitale pour notre jeu : elle permet à l'ensemble des joueurs de voir, d'entendre et d'interagir avec les mondes qu'ils traversent de la même manière. Il est donc important de la mettre en place le mieux possible.

Photon, l'utilitaire nous permettant d'implémenter le multijoueur de notre jeu, permet de mettre en place une synchronisation de manière très simple : il suffit, selon la documentation, d'utiliser un « component Photon View » (figure 5). Ce composant permet d'envoyer tous les détails d'un objet le possédant à un serveur et ce dernier renverra tous ces détails aux autres joueurs étant dans la même partie que l'objet. Ces détails sont sa position, sa rotation, sa taille et sa vitesse. Nous avons donc ajouté ce composant à chaque objet que nous souhaitions synchroniser.

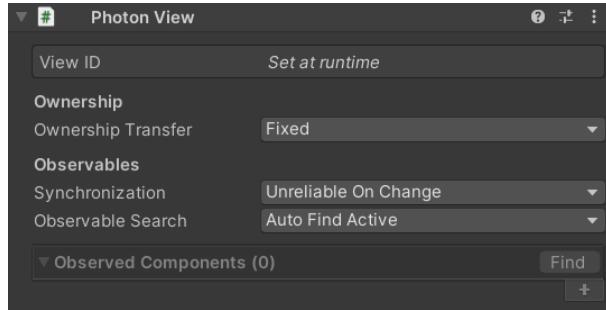


FIGURE 4 – Composant PhotonView

Cet ajout a été fait avant notre première soutenance et, sauf cas particuliers, aucun objet était synchronisé entre les joueurs (les cas particuliers étaient les objets réagissant au contact des joueurs comme les murs).

Nous avons donc recherché sur la documentation Photon et sur de nombreux sites et forums sans avoir trouvé de solutions. Nous avons testé un bon nombre de fois de changer certains paramètres, résultant uniquement à de faux espoirs. Puis, finalement, nous nous sommes rendus compte qu'il s'agissait d'un problème de « possession ».

Comme dit plus tôt, un objet possédant le composant « Photon View » envoie à un serveur des propriétés le concernant, mêmes propriétés qui seront transmises aux autres joueurs. Mais certaines données envoyées sont plus importantes que les autres. Il s'agit des données envoyées par le **possesseur** de l'objet. Si un objet est déplacé par un joueur ne « possédant » pas l'objet, il enverra quand même les données au serveur. Mais si le joueur possédant l'objet déplace ce dernier, la position de l'objet sera la dernière envoyée au serveur par le possesseur et non par l'avant-dernier joueur l'ayant touché. Il nous a donc fallu transmettre la propriété d'un objet au joueur l'ayant touché pour la dernière fois. Nous avons donc changé la propriété « ownership transfer » du composant « Photon View » de « Fixed » à « Takeover ». Ainsi, tous les objets réagissant au contact des joueurs sont correctement synchronisés.

Il reste toutes fois les objets ne réagissant pas au joueur (comme les pendules). Comme chaque joueur charge une manche locale-

ment et que les obstacles font partie d'une manche, ils ne peuvent pas être synchronisés, même avec un composant « Photon View ». Il faut donc que l'on charge l'ensemble des obstacles uniquement lorsque le premier joueur entre dans la partie. Pour ce faire, nous avons utilisé la fonction *Instantiate()* de Photon, permettant de créer les obstacles en une seule fois par partie, les synchronisant.

Nous avons donc terminé d'implémenter la synchronisation dans notre jeu.

5.2 Mode contre-la-montre

Dès la rédaction de notre cahier des charges, nous avions pour souhait que notre jeu possède un mode solitaire, c'est chose faite ! Ainsi, sur le menu d'accueil de notre jeu, un bouton intitulé « offline » a fait son apparition. Celui-ci permet d'accéder au mode solitaire qui est également un mode hors-ligne, pas besoin de connexion internet pour y accéder contrairement au multijoueur. Nous n'avions pas forcément pensé à cet aspect, mais il peut être très pratique de pouvoir jouer sans réseau : c'est un plus pour notre jeu qui est donc utilisable dans toutes les conditions.

Lucas a pu mettre en place cette partie hors-ligne du jeu grâce à des propriétés *Photon* qui permettent de se détacher du réseau uniquement en quelques lignes de code. Il s'est également chargé d'autres ajustements pour rendre l'expérience utilisateur meilleur : il n'y a pas d'intelligence artificielle dans ce mode et le joueur doit poursuivre son propre fantôme, ce qui le motive encore plus à performer. Dans ce menu hors-ligne, le joueur a plusieurs choix : s'entraîner sur une carte en particulier ou bien jouer une partie complète avec l'enchaînement des 3 manches.



FIGURE 5 – Choix de la carte en mode hors-ligne

Parlons un peu du fantôme du joueur développé par Mathéo. Si vous venez d’installer le jeu et que vous n’avez pas encore joué en mode solo, alors il n’y en aura pas, car vous n’aurez aucun meilleur temps. En revanche, dès lors que vous commencez à jouer (uniquement dans ce mode solo), votre position est enregistrée en continu grâce aux coordonnées de l’espace x, y et z. En attendant la fin de la partie, ces informations sont stockées dans une liste, à l’instant où vous passez la ligne d’arrivée, cette liste est écrite dans un fichier au format CSV avec une syntaxe particulière. Ainsi, ce fichier permet à votre ordinateur et donc au jeu de sauvegarder votre parcours pour vos prochaines parties. À l’inverse, si une ou plusieurs sauvegardes existent, votre fantôme apparaîtra dans la partie, il choisira le fichier correspondant à votre meilleur temps, supprimera les autres, puis il lira les données et progressera exactement de la même manière que lors de la partie enregistrée.

5.3 Intelligence artificielle

Il nous faut créer une intelligence artificielle (IA) « copiant » les actions des joueurs. Certaines auront des déplacements hasardeux, d’autres voudront du mal aux autres joueurs, d’autres encore iront finir le parcours le plus vite possible. C’est Mathéo qui

s'est chargé de son développement.

5.3.1 Fonctionnement

Pour commencer, l'IA ne doit pas emprunter le chemin le plus court vers la ligne d'arrivée, sinon, les joueurs n'auront aucune chance de gagner une partie. Il faut donc qu'elle suive un chemin moins précis. De plus, il ne faut pas que l'IA reste bloquée sur des obstacles du parcours. Il faut donc qu'elle puisse sauter ou bien éviter certains obstacles. Pour finir, comme les joueurs, les IA auront la possibilité de récupérer des bonus. Il faut donc que ces IA puissent les utiliser au meilleur moment.

Après avoir posé toutes ces caractéristiques, nous avons décidé d'implémenter un système de zones visibles seulement par les IA. Toutes les cartes posséderont des zones. Ces zones seront réparties sur l'entièreté des cartes, du début jusqu'à l'arrivée. Elles permettent aux IA de se diriger vers la ligne d'arrivée en passant à la fois les obstacles sans embûches et les joueurs leur voulant du mal.

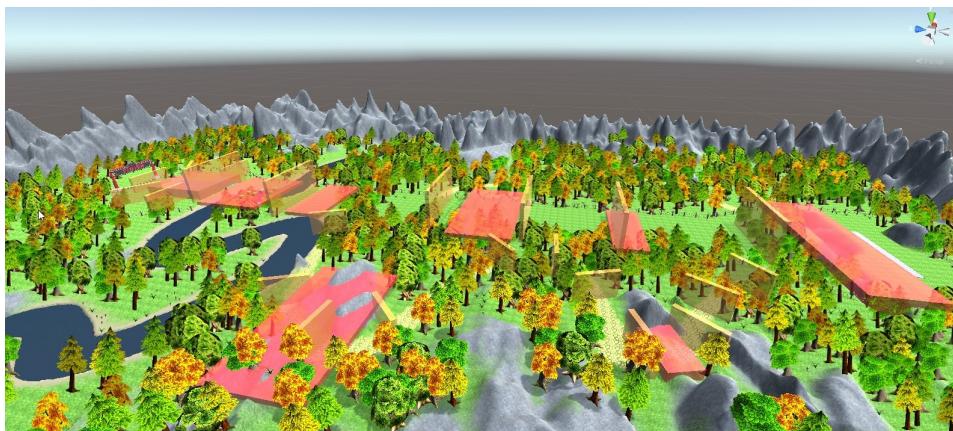


FIGURE 6 – Zones transparentes visibles uniquement par l'IA

5.3.2 Problèmes rencontrés

Avant la première soutenance, Mathéo a implémenté une intelligence artificielle se dirigeant vers un bloc en particulier. Cette

IA n'était vouée qu'à certains tests que nous avons enlevés à présent. Lors du développement, nous avons rencontré quelques problèmes et même certains nous empêchant de faire bouger notre IA.

Un problème majeur que nous avons rencontré était les déplacements incontrôlés de notre IA vers une des zones. Nous utilisons l'utilitaire « Nav mesh » de Unity, permettant de déplacer un objet dans une scène en lui indiquant uniquement son point de destination et le tout en une seule ligne de code. Le revers de la médaille était le fait que les IA pouvaient se déplacer partout dans la scène, et donc à des endroits soit trop avantageux, soit pas assez. De plus, concernant les sauts des IA, ces derniers nous ont posés de nombreux problèmes. Il nous était impossible de bouger notre IA à la fois horizontalement et verticalement à cause du composant « Nav mesh ». Il fallait donc arrêter le programme permettant de faire bouger l'IA pour la faire sauter, impliquant l'utilisation de plus de variables et donc plus de mémoire pour un simple saut.

Nous sommes donc repartis de zéro et nous nous sommes servis du composant « Rigidbody » pour déplacer notre IA.

5.3.3 Implémentation

Comme dit plus haut, nous avons utilisé le composant « Rigidbody » pour déplacer notre IA. Ce composant est utilisé par Unity pour simuler des forces sur les objets le possédant. Cependant, lors de notre dernière implémentation, nous avons aussi utilisé les coordonnées des IA elles-même pour les déplacer horizontalement. Pour ce faire, nous avons utilisé le composant « Transform » de chacune des IA pour ensuite leur appliquer la fonction « MoveToward » permettant de les déplacer sans utiliser leur « Rigidbody » et évitant ainsi des erreurs et des déplacements non naturels.

5.3.4 Zones de zones et liste de zones

Nous avons implémenté des Zones de zones. Ces zones permettent de répondre à un de nos problèmes : la possibilité d'emprunter plusieurs chemins sur une carte. Si nous avions implanté notre IA sans ces zones, lorsque ces dernières empruntaient des chemins annexes, elles ne pouvaient pas retrouver le chemin central ou bien elles se perdaient dans le décor. Lorsqu'une IA entre dans une de ces zones de zones, elle choisit au hasard un des point de contrôle dans ces zones pour s'y déplacer, et ce ainsi de suite jusqu'à la fin du parcours. Chacune de ces zones contiennent une liste de zones étant les prochaines à emprunter pour les IA.

5.4 Interface utilisateur

Comme évoqué précédemment, Mathieu est le responsable de l'interface utilisateur, il s'en est donc occupé et a terminé tous les objectifs fixés pour le jeu final. Tout d'abord, définissons ce qu'est l'interface utilisateur (UI). Pour faire simple, c'est tout ce qui est conçu dans un dispositif d'information avec lequel une personne peut interagir. Il peut s'agir d'affichage, de claviers, d'une souris et de l'apparence d'un bureau d'ordinateur avec différents boutons. Elle est très importante pour mettre en forme le projet et rendre compréhensible le jeu.

Mathieu a d'abord dû s'occuper de mettre en forme la partie multi-joueur. Pour ce faire, il a créé différents menus. Les scripts *Menu* et *Menu Manager* ont alors été créés servant respectivement à initialiser le menu, à l'ouvrir et à le fermer. Les menus sont les suivants : *Title*, *Loading*, *Error*, *CreateRoom*, *FindRoom*, *ChooseMap* et *Room*. Un menu s'active lorsqu'un bouton est pressé, à ce bouton est associée une fonction qui permet d'effectuer la tâche désirée. Le menu *Title* est actif dès le lancement du jeu et représente donc le menu principal qui permet de rentrer son pseudo et ouvrir les différents menus. Le menu *Loading* est affiché lorsque le jeu charge un autre menu ou se connecte à Internet. A chaque erreur de chargement ou de connexion, c'est le menu *Error* qui va s'afficher pour indiquer l'erreur adéquate. Le menu *CreateRoom*

permet de créer une salle en entrant le nom de celle-ci dans l'emplacement prévu à cet effet. Le menu *FindRoom* quant à lui permet de rejoindre une salle déjà existante. Ensuite le menu *ChooseMap* est le menu actif lorsqu'on choisit de jouer hors ligne en mode fantôme . Il nous permet de choisir le parcours sur lequel on veut s'améliorer ou jouer une partie entière. Pour finir, lorsqu'on a créé une pièce, on arrive dans le menu *Room* où sont affichés les différents joueurs présents dans la salle et le nom de celle-ci. C'est d'ici qu'on peut lancer la partie en cliquant sur le bouton *Play*.

En plus des différents menus évoqués, Mathieu a mis en place le choix du pseudo. Pour ce faire, il a fallu utiliser un *Input Field*, c'est-à-dire un champ de texte où l'on peut entrer le nom de notre choix. Ce nom a dû être stocké pour être placé au-dessus de la tête du joueur lors de la partie. Un script a alors été créé pour gérer tout ça et il est appliqué à chaque joueur.

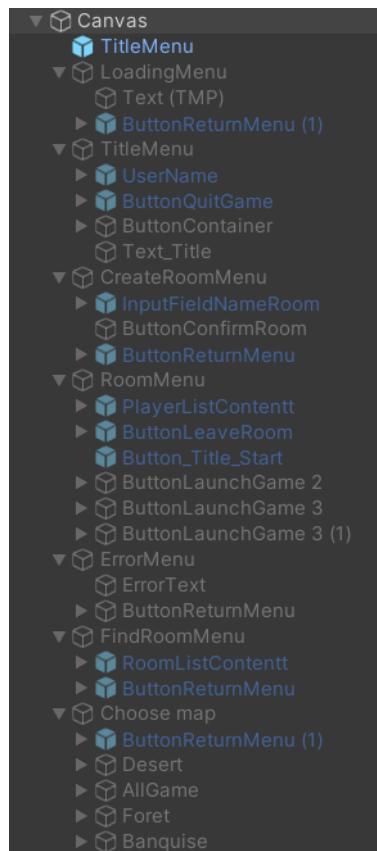


FIGURE 7 – Liste des canvas affichés au fur et à mesure

Depuis la première soutenance, une fois que l'interface utilisateur était fonctionnelle et nous permettait d'accéder à une partie, il a fallu l'étoffer et trouver l'identité graphique de notre jeu. Celui-ci nous emmène dans différents univers comme évoqué auparavant, c'est pour cela qu'en arrière-plan Mathieu a animé une caméra pour montrer un aperçu de ces décors depuis le menu principal. Pour ce faire, il a dû apprendre à créer des animations afin de bouger la caméra au sein des décors miniatures qu'il a créés spécialement pour ça. De plus, pour mettre en accord l'environnement du jeu et les différents boutons, Mathieu a dû importer un asset avec différents choix de boutons et de design de menus. Ainsi, il a pu finir de mettre au goût du jeu l'UI tout en gardant les mêmes fonctionnalités que les précédents boutons à quelques exceptions près.



FIGURE 8 – Menu Principal

Une fois la partie lancée, on ne pouvait pas quitter la partie, Mathieu a alors mis en place un menu Pause qui permet de mettre en pause notre partie et de quitter la partie en cours. Le classement de la partie y est aussi affiché en temps réel. L'onglet *Settings* où l'on peut choisir la résolution d'écran et régler le volume sonore du jeu a aussi été rajouté. Pour régler le volume, il a fallu associer le volume des hauts-parleurs à un *slider* pour pouvoir le régler au pourcentage près. Deux résolutions d'écrans sont disponibles : *640x360* et *1920x1080*.



FIGURE 9 – Settings

Lorsqu'on se trouve en jeu, on peut voir la place qu'occupe chaque joueur dans le jeu et son temps effectué sur le parcours. Un script a été créé à cet effet. Le script *UIinGame* qui sert à calculer le temps de chaque joueur et à calculer la place de chaque joueur sur la partie. Pour cette dernière, il nous a fallu trouver un moyen efficace de le faire et nous avons opté pour une solution qui consiste à calculer à intervalle régulier la distance entre chaque joueur et la ligne d'arrivée. Une fois ces distances ajoutées à une liste, il suffit de trier cette liste pour avoir le classement. Par la suite, Mathieu a donc pu afficher le tableau des scores au sein du menu pause où on peut voir la place, le temps et le nom du joueur.



FIGURE 10 – Menu Pause

À la fin de chaque parcours, il nous fallait une transition pour passer au suivant. Mathieu a donc créé une nouvelle scène où est affiché le tableau des scores cumulant les temps des parcours déjà effectués. Les secondes correspondantes au bonus de chaque joueur sont retranchées. De plus un podium est représenté avec les personnages des 3 premiers aux places adéquates. Cette scène est la même pour tout le monde puisqu'elle est lancée par le créateur de la salle. Lui seul peut lancer la prochaine manche avec le bouton *Play*. Tous les joueurs peuvent quitter la partie à ce moment du jeu avec le bouton *Home*.



FIGURE 11 – Scène finale

5.5 Synchronisation des chronomètres et du classement

Comme évoqué dans la section 5.1, la synchronisation des informations entre les joueurs est essentielle. Dans notre cas, nous avons eu à gérer la synchronisation des obstacles ainsi que celle du chronomètre et du classement qui doivent être identiques pour tous les joueurs de la partie. Au début de notre implémentation, chaque joueur calculait son chronomètre à partir du moment où il rejoignait la salle, il calculait également sa place par rapport aux autres joueurs en regardant leurs distances à l'arrivée.

Désormais, ces calculs sont centralisés : il n'y a que le créateur d'une salle multijoueur qui définit le chronomètre de chacun

des joueurs et qui calcule leur place. Individuellement, les clients, c'est-à-dire les joueurs qui ont rejoint une salle, n'ont plus qu'à calculer leur distance par rapport à la ligne d'arrivée pour permettre au créateur d'établir le classement. Pour ce faire, Lucas a défini différentes *Custom Properties*, ce sont des propriétés générées par Photon accessible à tous les membres d'une partie. L'élément principal commun à tous les joueurs est un tableau de chaîne de caractères, celui-ci contient pour chaque joueur : son nom, sa place et son chronomètre. Le chronomètre est calculé à partir d'un même temps de départ défini à chaque fois qu'une nouvelle carte est chargée. Ce tableau est en réalité le tableau des scores, comme chaque joueur connaît son nom, il n'a plus qu'à venir récupérer sa place et son temps pour les afficher sur l'écran. Ainsi, tous les joueurs ont exactement le même chronomètre et lorsque vous affichez le menu pause, tous les joueurs ont le même tableau des scores au millième près.

5.6 Choix de l'apparence du joueur

Pour une diversité des joueurs sur le terrain de jeu, nous avons choisi de proposer plusieurs apparences pour le joueur. Ces apparences sont des modèles 3D importés d'internet.



FIGURE 12 – Les différents personnages

Mathieu a mis en place un menu qui nous offre la possibilité de choisir entre 3 apparences de joueur différentes. Pour ce faire, il a placé les trois personnages sur une scène puis les a animés à l'aide de tutoriels. C'est le script *SelectPlayer* qui gère tout ce système. Chaque fois que l'on passe la souris sur l'un d'entre eux, alors le *Collider* du joueur est détecté et le personnage est illuminé pour indiquer que l'on peut le sélectionner. Ainsi, en cliquant dessus,

vous choisissez le personnage et vous êtes directement renvoyé au menu principal. Cet ajout a aussi été l'occasion pour Mathieu d'ajouter des sons à notre jeu chaque fois que l'on place la souris sur un personnage par exemple.

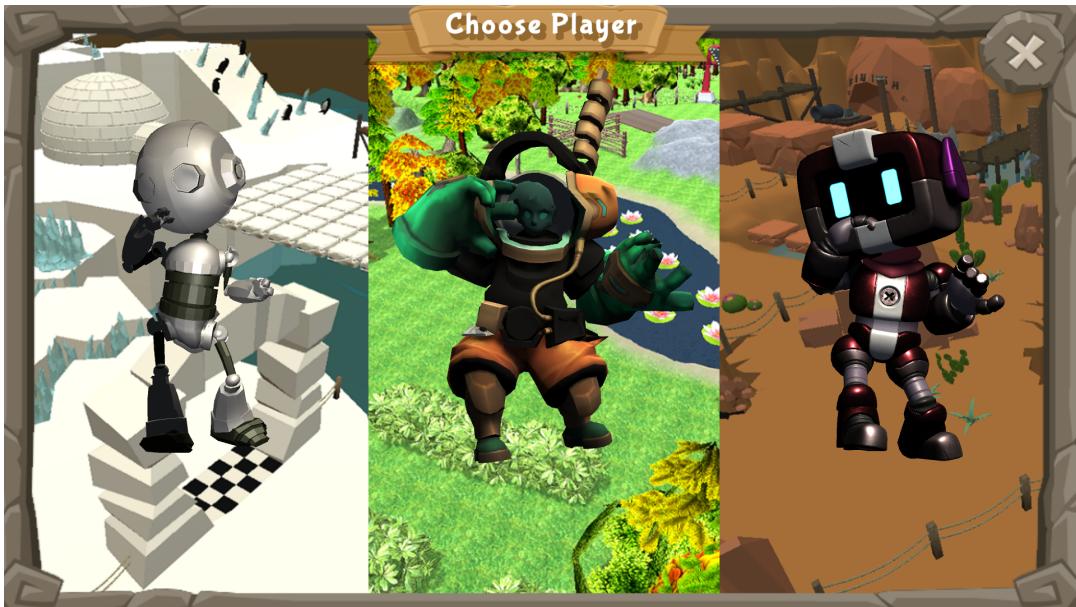


FIGURE 13 – Choix de l'apparence du joueur

5.7 Cration des parcours

Nous avons commence la cration des parcours avant la premire soutenance et cette tache importante continue depuis et nous a emmen jusqu la dernire soutenance. C'est Lucas qui a commenc en premier la cration de notre parcours fort, il a trouv un asset qui nous a beaucoup aid, ce qui nous a pouss ´a dvelopper cette partie un peu plus tt qu'initialement prvu sur le planning. Cependant, cet aspect de notre projet s'est dvelopp jusqu la fin au travers de la cration d'une multitude de parcours qui apporte une grande diversit ´a notre jeu.

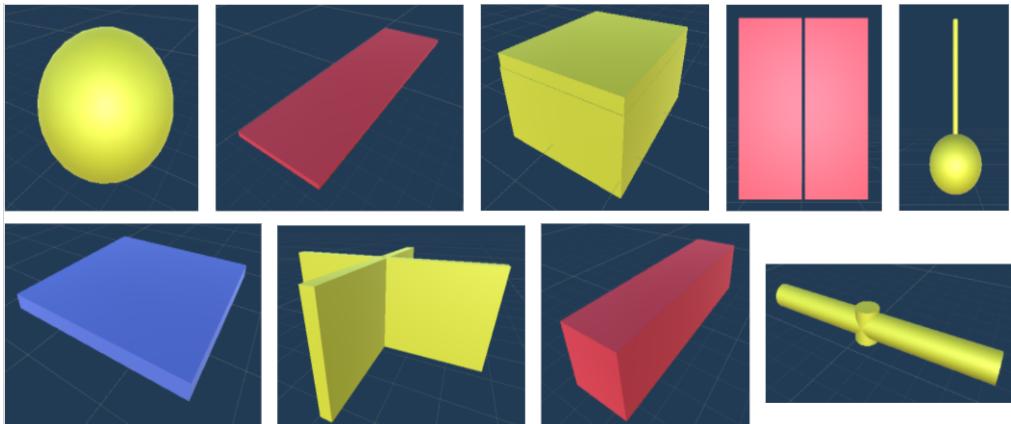


FIGURE 14 – Prefabs d’obstacles

Comme vous pouvez le voir ci-dessus, nous avons différents obstacles utilisables très facilement. La plupart d’entre eux proviennent de l’asset « Obstacle Course Pack » publié par AisuKaze Studio. Certains ont été adaptés pour notre utilisation tandis que d’autres, comme le tourniquet, ont été fait par Lucas. Ces obstacles peuvent être divisés en deux catégories : ceux utilisant la physique du moteur Unity n’ayant besoin d’aucun script et ceux étant en mouvement nécessitant un script pour les faire bouger.

Pour commencer, les obstacles utilisant la physique du moteur Unity sont les plus simples : il y a la sphère en haut à droite de la figure 14 qui se met en mouvement lorsqu’un joueur la pousse, le tourniquet qui se met à tourner en contact avec un joueur, les murs qui tombent dès qu’un joueur les traverse, etc. Ces obstacles ont tous trois composants : un *Mesh Renderer*, un *Collider* et un *Rigidbody*. Ce sont ces deux derniers qui permettent d’interagir avec le joueur et d’utiliser les phénomènes physiques que l’on connaît dans le monde réel.

Ensuite, il y a les obstacles étant en permanence en mouvement, ceux-ci nécessitent un script C# pour qu’ils puissent bouger, contrairement à la précédente catégorie, ils n’ont pas d’interaction avec le joueur en dehors du fait de le repousser. On peut par exemple citer la faucheuse en bas à gauche de la figure 14 qui tourne sur elle-même ou bien les murs vivants se déplaçant de gauche à droite.

Cet asset est très important pour notre jeu et nous a fourni

des éléments essentiels que l'on a pu comprendre grâce à la documentation Unity. Les modifications que l'on a faites et l'ajout d'obstacles ne nous a pas posé de problèmes particuliers.

Cependant, ces obstacles ne sont visuellement pas adaptés à nos environnements et nos décors. C'est donc Jules qui a pu adapter l'apparence de ces obstacles, grâce à des modèles 3D comme nous le verrons dans la partie suivante.

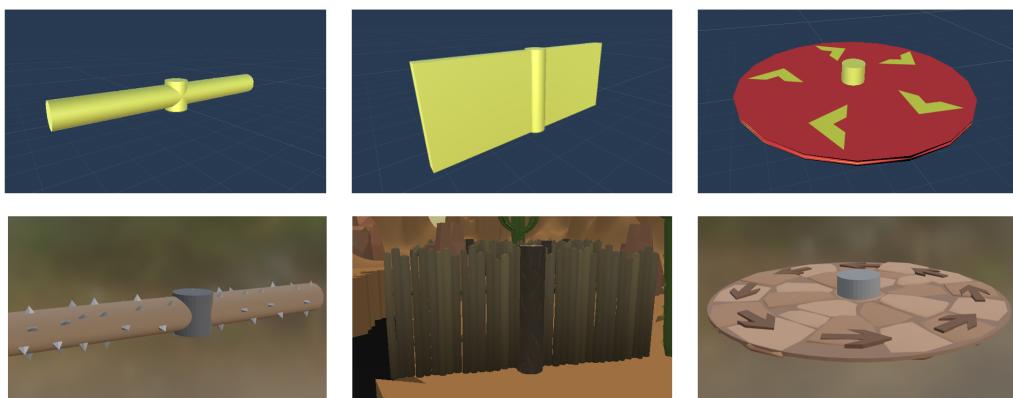


FIGURE 15 – Obstacles adaptés aux décors

Nous avons aussi ajouté des chemins alternatifs afin de rendre chaque partie unique. Grâce à cette collection d'obstacles, le travail de création de parcours consiste à les mettre bout à bout dans un ordre qui permettent au joueur d'avoir une expérience à la fois amusante et ni facile ni trop difficile. Cependant, cette tâche est difficile et nous prend beaucoup de temps. Nous avons tout de même choisi de réaliser nos parcours « À la main » plutôt que procéduralement, car ceux-ci sont tous très différents et afin de correspondre aux différents décors, ce ne sont jamais exactement les mêmes obstacles qui sont utilisés.

5.8 Modélisation 3D

Au début du projet, nous n'avions aucune expérience dans le domaine de la modélisation 3D. C'est donc Jules qui s'est chargé de découvrir et d'apprendre cette compétence. Jules utilise le logiciel Blender, un logiciel gratuit et complet. En parallèle, il a suivi une

formation sur YouTube de plus de 70 épisodes qui concerne la modélisation 3D, l'utilisation de Blender ainsi que les textures et le rendu final. Durant le développement du projet, Lucas a demandé à Jules plusieurs créations pour embellir le parcours forêt et il a aussi eu besoin d'objets spécifique pour faire les parcours désert et banquise. Certaines de ces créations sont présentes sur le jeu : les barrières, la banquise, les bonus, l'arche de départ du désert, les arches d'arrivées de la forêt et du désert, etc.

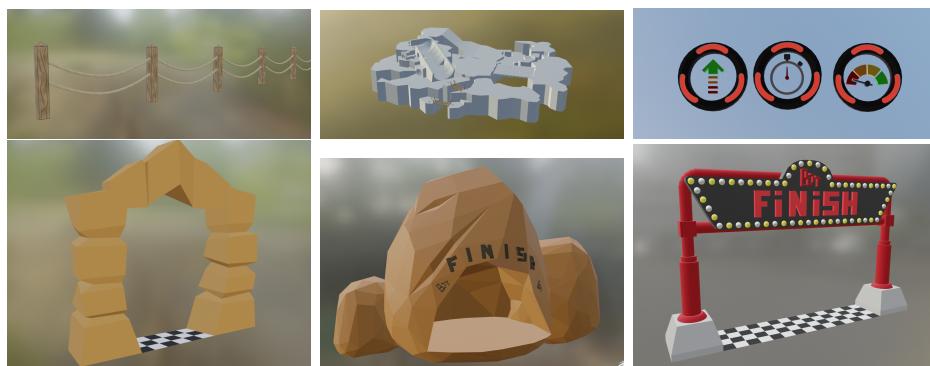


FIGURE 16 – Barrières, bûches, failles et arches de départ et d'arrivée

Toutes ces créations permettent de rendre le jeu plus esthétique et d'intégrer les obstacles à l'univers du jeu. En effet, les obstacles présents dans l'asset présenté par Lucas ne s'accordaient pas à l'esprit de notre jeu. Jules a donc dû créer des apparences 3D pour ces obstacles afin de bien les intégrer aux différents environnements du jeu. Par exemple les buche et les palissade dans la foret, la plaque tournante et les haches dans le désert ou encore les stalagmites et la piste de ski sur la banquise.

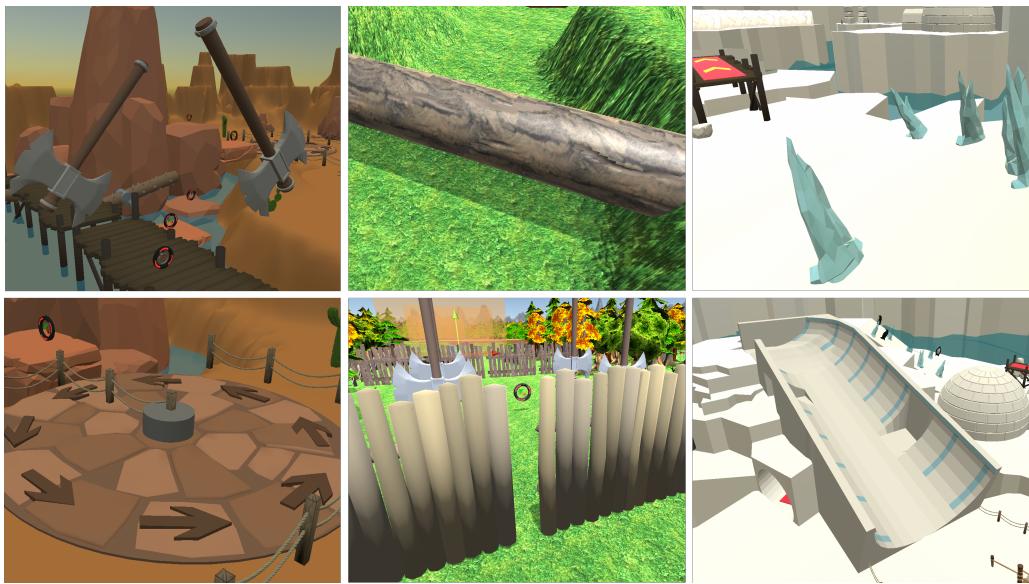


FIGURE 17 – Obstacles créés par Jules

5.9 Décors

Nous avons réalisé trois décors : forêt, désert, glacier. La forêt et le désert ont été réalisé grâce à l'outil *Terrain* fourni par Unity. Il permet de créer un terrain et de complètement le personnaliser grâce à différents onglets permettant de gérer le relief du parcours, les textures que nous appliquons au sol et ainsi que l'implantation des arbres qui a été utile pour la forêt. La banquise quant à elle est entièrement faite par Jules en modélisation 3D.



FIGURE 18 – Décor désert

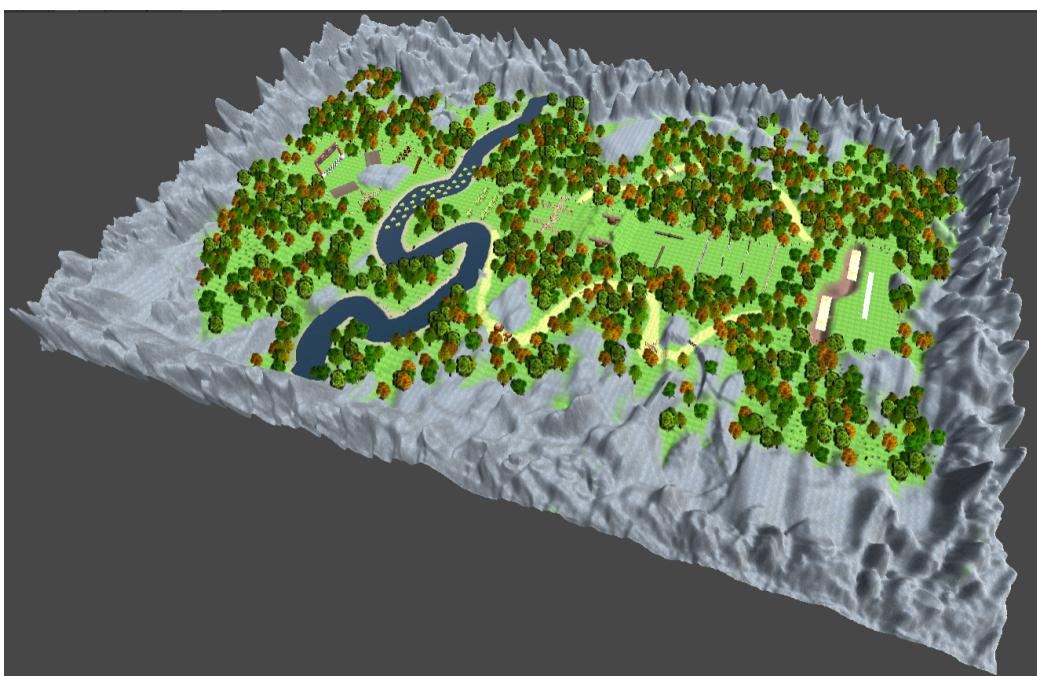


FIGURE 19 – Décor forêt



FIGURE 20 – Décor banquise

Pour toutes ces cartes, nous avons utilisé des assets afin d'avoir des éléments pré-faits pour le décor : les cactus, arbres, pierres et même pingouin. Certains décors proviennent aussi de modèles 3D trouvées sur internet comme l'igloo ou les pontons. Pour finir, d'autres décors ont été créés par Jules, comme dit précédemment.

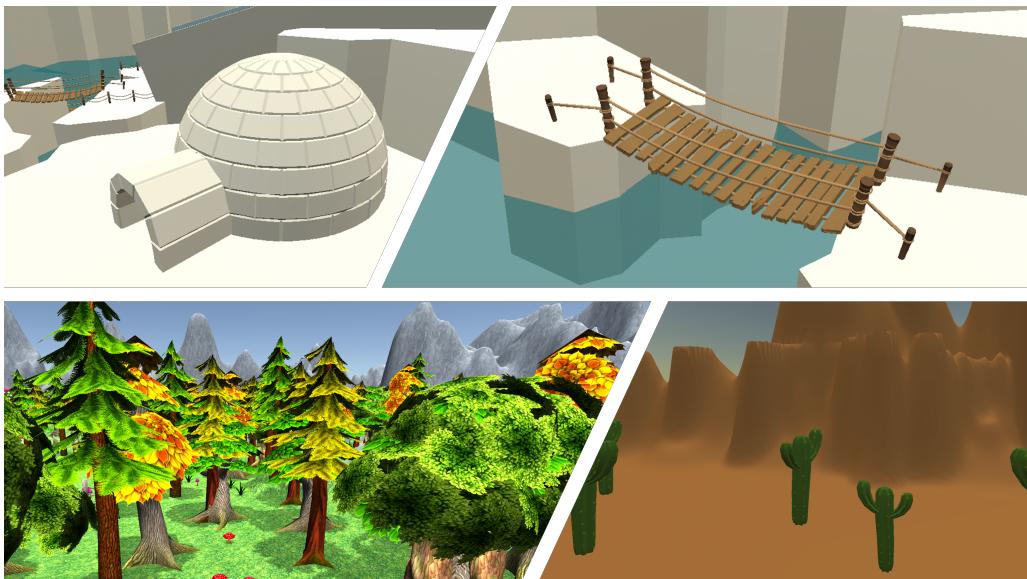


FIGURE 21 – Décors provenant d'assets et d'internet

5.10 Communication

Dès le début du projet, Jules a créé un compte Instagram sur lequel les avancées du projet sont régulièrement publiées. Afin de créer une communauté autour de notre projet, nous avons tous partagé ce compte sur nos réseaux personnels. Ce compte regroupe actuellement une centaine d'abonnés ce qui en fait l'un des projets de l'EPITA Rennes les plus suivis sur Instagram ! Nous y partageons des photos de notre jeu ainsi que des moments de travail en équipe à travers des publications et des stories.

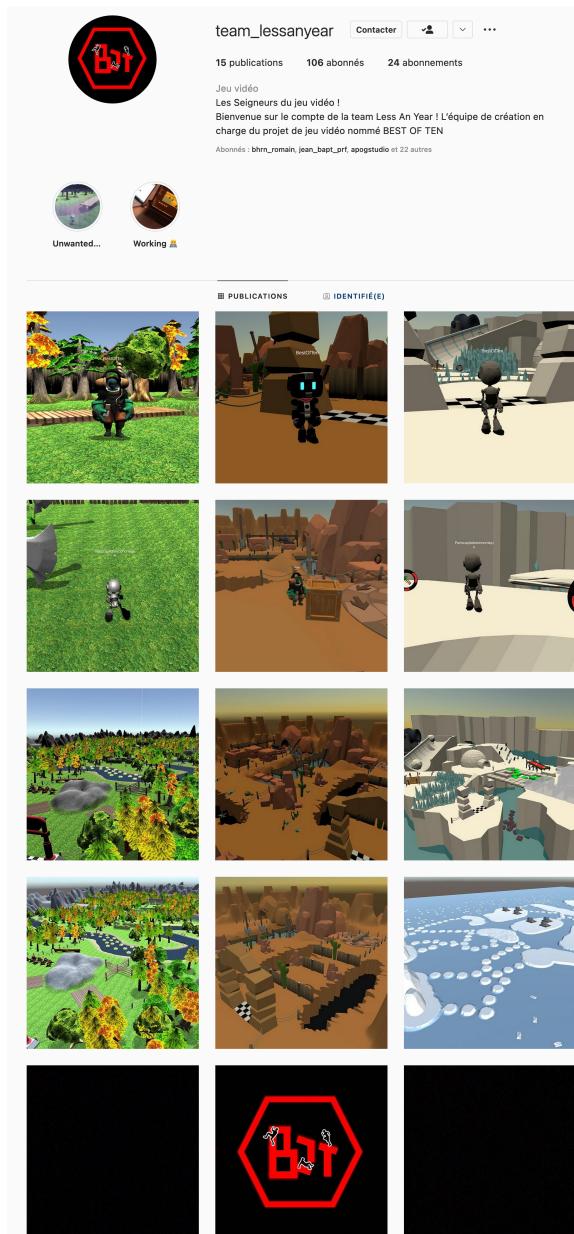


FIGURE 22 – Compte Instagram du projet (@team_lessanyear)

De plus, au début de la réalisation du jeu, Mathéo a commencé à créer le site internet du projet contenant une présentation de celui-ci, une bibliographie de tous les documents utilisés, un lien de téléchargement du jeu vidéo ainsi que la présentation des membres du groupe :

- Page d'accueil :

La page principale de notre site, la page d'accueil, est composée d'une présentation de notre projet ainsi qu'une galerie photo, permettant de montrer à quoi ressemble notre jeu. (figure 23)



FIGURE 23 – Page d'accueil

— Page présentation :

La page présentation contient une description plus détaillée des éléments de notre jeu : les règles, les cartes, les fonctionnalités phares, etc... (figure 24). Cette page contient aussi l'historique de notre projet ainsi que l'ensemble des problèmes rencontrés ainsi que les solutions envisagées.

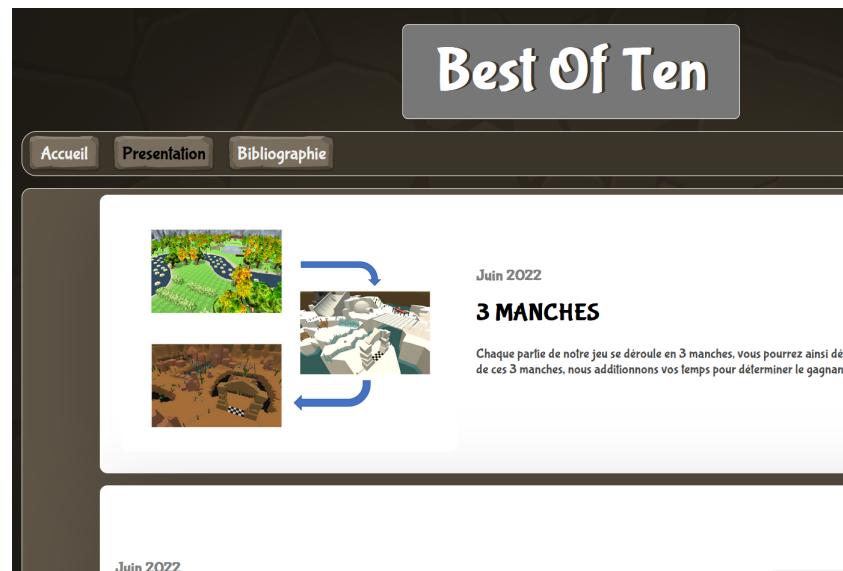


FIGURE 24 – Page présentation

– Page bibliographie :

Cette page contient l'ensemble des liens vers les sites utilisés pour notre projet. Elle compte aussi une présentation des membres. (figure 25).

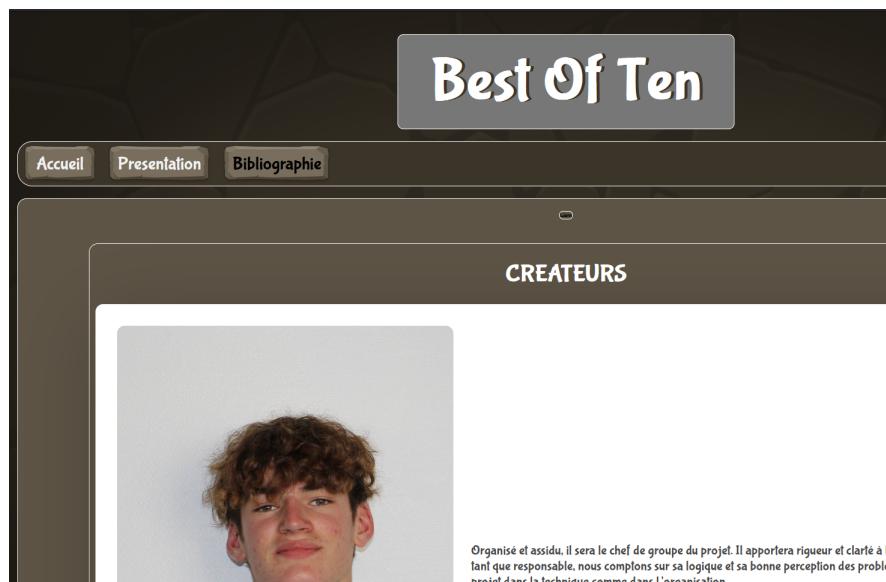


FIGURE 25 – Page bibliographie

– Page download :

La dernière page de notre site, la page download, permet de télécharger notre projet. Plusieurs options de téléchargement sont possibles : téléchargement complet du projet (jeu, rapport de projet, manuel d'installation/désinstallation, manuel d'utilisation), téléchargement du rapport uniquement, téléchargement du projet uniquement. (figure 26)

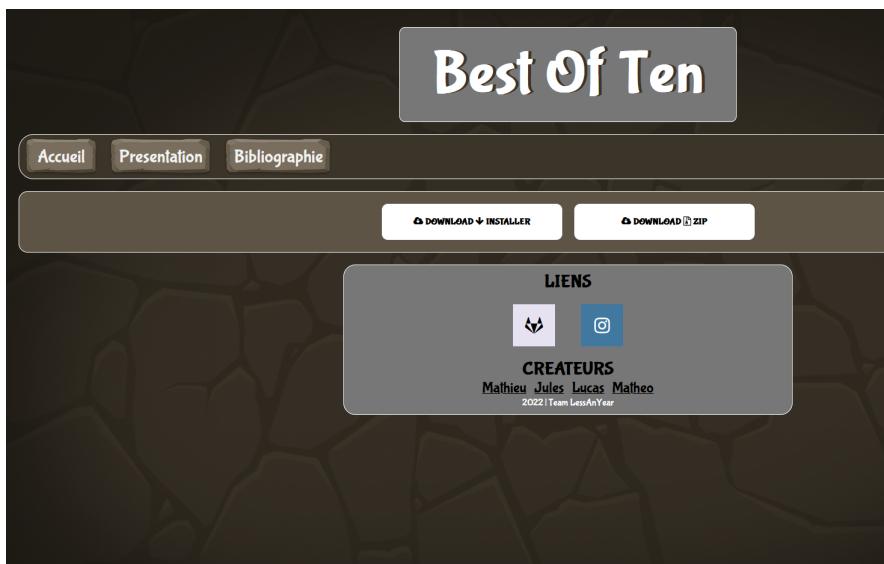


FIGURE 26 – Page download

5.11 Système des 3 manches

Le système des 3 manches par partie est une étape importante de notre jeu réalisé puisque c'est une des caractéristiques nous différenciant d'autres jeux. Lucas et Mathieu ont tout d'abord réussi à l'implémenter lorsqu'il n'y a qu'un seul joueur dans la partie. Cependant, lorsque Lucas et Mathieu ont tenté de synchroniser ceci avec le multijoueur, de nombreux problèmes sont apparus. Par exemple, attendre que l'ensemble des joueurs (IA comprises) aient terminé la manche pour les déplacer dans une nouvelle carte. Cette carte se doit d'être la même pour tous les joueurs de la partie et que les mêmes joueurs se retrouvent dans cette nouvelle carte. Les 3 environnements sont choisis aléatoirement et il ne peut y avoir un même environnement joué 2 fois pendant la même partie.

Après s'être confronté à cette liste non exhaustive de problèmes, ce système est finalement fonctionnel ! Maintenant, il y a un enchaînement de 3 manches dans 3 cartes distinctes par partie.

Entre chacune de ces manches, les joueurs sont amenés dans un autre décor où s'affiche le classement de la partie ainsi qu'un podium des 3 premiers joueurs et de leurs apparences correspondantes. Ce classement est bien le classement de la partie et non de la manche, c'est-à-dire que les temps correspondent à la sommes des manches précédentes et les positions sont recalculées en conséquence. Une fois ce classement affiché, il y a uniquement le créateur de la salle multijoueur qui peut lancer la manche suivante. Si jamais celui-ci venait à quitter le jeu, l'ensemble des autres joueurs ont accès à un bouton « Home » leur permettant de se déconnecter de la partie et de revenir à l'accueil du jeu.

5.12 Interaction avec les objets

Comme dit plus tôt, notre jeu consiste à faire un parcours d'obstacles le plus vite possible, mais ce serait ennuyant si on ne pouvait pas interagir entre joueur pour pouvoir ralentir la progression des autres sur le parcours. Il a fallu alors créer des objets que l'on peut porter et lancer sur nos adversaires, mais aussi déplacer et utiliser pour accéder à des endroits inaccessibles du parcours dans le but de gagner du temps. C'est Mathieu qui s'est chargé d'implémenter le script *PickableObject* qui nous permet de porter les objets et les lancer. Ce script doit par conséquent être rattaché à chaque objet que l'on veut pouvoir porter. Ainsi, quand on se trouve proche de l'objet avec le script, c'est-à-dire dans son *Trigger*, la touche E nous permet de le porter, le clic droit de le reposer et le clic gauche de le lancer. Le but est de bien viser ses adversaires pour les toucher. Dès l'instant où l'objet est en l'air et qu'il rentre en collision avec un autre joueur, alors celui-ci est étourdi et ne peut plus bouger pendant 5 secondes. Certains objets ont pour seule fonctionnalité d'être lancé et d'autres ont un *Box Collider* pour qu'on puisse monter dessus. Ces objets sont disposés un peu partout sur les parcours et restent en cohérence avec l'univers dans lequel on se trouve comme vous pouvez le voir ci-

dessous.



FIGURE 27 – Tronc que l'on peut prendre sur le parcours Forêt



FIGURE 28 – Caisse que l'on peut prendre, lancer et utiliser pour accéder aux endroits inaccessibles du parcours

5.13 Bonus

Les objets à ramasser ne sont pas les seuls à être présent sur notre jeu, il y a aussi les *Botcoins* sous forme de pièces avec différents dessins à l'intérieur qui représentent différentes choses. En effet, comme dit plus tôt, le joueur pourra ramasser des bonus. Ces bonus auront uniquement des effets positifs pour le joueur et ils lui seront profitables. Il lui faudra les collecter lors de sa partie, ainsi chaque bonus récolté ne sera plus disponible pour les autres joueurs. Voyons de plus près à quoi ils peuvent servir. Il y a différents types de bonus :

- Bonus de vitesse : il permet d'améliorer temporairement la vitesse de déplacement du joueur.
- Bonus de temps : il améliore le temps final du joueur de 2 secondes par bonus récolté.
- Bonus de saut : il permet d'améliorer temporairement la hauteur de saut du joueur.

Mathieu, c'est occupé d'implémenter cet aspect du jeu en créant un script qui rassemble tous les bonus. Le principe est le suivant : lorsqu'on se trouve dans le *Trigger* du bonus, celui-ci est identifié par le *tag* qui lui est rattaché puis il est détruit. Pour les bonus de saut et de vitesse, les caractéristiques initiales du joueur sont stockés dans une variable temporaire puis multipliées par 1,5. Au bout de 5 secondes, la vitesse ou le saut sont remplacés par la valeur enregistrée dans la variable temporaire. En ce qui concerne le bonus de temps, chaque fois que l'on récolte une pièce, une variable comptant notre nombre de pièces est incrémenté puis réutilisé à la fin du parcours pour enlever les secondes correspondantes au temps final du joueur. Ce script est rattaché au joueur pour faciliter l'accès à ses caractéristiques de déplacements. Ces bonus offrent des avantages temporaires sur une courte durée, mais ils permettent à chaque joueur de tirer son épingle du jeu.



FIGURE 29 – Bonus de vitesse, de temps et de saut



FIGURE 30 – Bonus sur la Banquise

5.14 Sons

L'ajout des sons sur notre jeu a été réalisé à la fin du développement de celui-ci. Dans *Unity*, le système de son est assez simple : un composant *Audio Source* permet de diffuser des sons avec différents paramètres et un composant *Audio Listener* permet d'écouter les sons émis par une ou plusieurs *Audio Source*.

Dans notre jeu, chaque scène contient un *Audio Source* contenant la musique adaptée lorsque le joueur rejoint cette scène. De plus, le joueur en lui-même possède aussi un *Audio Source* permettant aux différents bruitages d'être joué : le saut, l'étourdissement

et la collecte de pièces. En addition à cet *Audio Source*, un *Audio Listener* permet au joueur d'entendre ses bruits et la musique d'ambiance pour avoir une expérience de jeu plus agréable.

Nos sons proviennent majoritairement du jeu Mario Kart, ce sont des sons connus de tous et universels qui plaisent à tous. Les bruitages proviennent quant à eux du site de partage de bande sonore FreeSound.

6 Récit de la réalisation

Depuis janvier, notre projet évolue constamment entraînant la mise en place d'une organisation particulière afin de faire avancer le projet. Pour répondre aux attentes et développer notre jeu, nous nous retrouvons en moyenne une fois par semaine afin de redistribuer certaines tâches. Ces rendez-vous sont aussi l'occasion de faire le point sur ce que chacun a fait de son côté, de s'expliquer les différentes parties pour être certain que chacun comprenne globalement ce qu'on fait les autres et de les aider sur les problèmes qu'ils peuvent rencontrer.

6.1 Jules

La création de ce jeu a été pour moi un grand apprentissage, et pas seulement dans le domaine du jeu vidéo. Effectivement, je suis tout d'abord fier d'avoir mené à bien ce projet en tant que chef d'équipe.

Organiser un groupe m'a beaucoup plus, car l'ambiance de notre groupe était déterminante et propice au travail. En effet, l'entraide était efficace et permettait de régler plus facilement nos problèmes. L'organisation de notre groupe était bonne et la répartition des tâches équilibrail bien le temps de travail de chacun, malgré un manque de communication sur la fin du projet, résultant par une implémentation des intelligences artificielles décevante.

Durant ce projet, j'ai aussi développé des connaissances personnelles. Outre la programmation et l'implémentation de notre

jeu vidéo, j'ai acquis une nouvelle compétence : la modélisation 3D. Je suis vraiment content d'avoir découvert et beaucoup appris dans ce domaine qui ajoute une corde à mon arc du savoir-faire.

Ce travail m'a aussi appris à confectionner un cahier des charges et rédiger des rapports. Je me suis aussi amélioré à l'oral grâce aux multiples soutenances que nous avons effectuées.

6.2 Mathieu

Ce projet m'a beaucoup apporté que ça soit en terme d'organisation ou d'esprit d'équipe. Il m'a obligé à avoir une certaine régularité dans mon travail avec des objectifs à remplir en un temps donné.

Au début du projet, je me demandais comment allons-nous réussir à concrétiser nos idées jusqu'au bout. Puis au fur et à mesure, à force de s'informer et d'effectuer de nombreuses recherches qui ont demandées beaucoup d'énergie, je me suis rendu compte qu'avec du travail et de la passion on pouvait accomplir beaucoup de choses qui nous paraissent impossibles. Ce projet m'a permis d'acquérir beaucoup de connaissance et un autre point de vue sur les problèmes à résoudre.

Cependant, au cours de cette réalisation, chaque avancée entraînait de nouveaux problèmes qui nous ralentissait dans notre progression et nous prenait beaucoup de temps. À ces moments-là, la motivation n'était plus la même, mais leur résolution était des moments de joie particuliers.

Avant la deuxième soutenance, je trouvais qu'on avait toujours quelque chose d'intéressant à ajouter et que chacun arrivait à apporter sa pierre à l'édifice. Depuis la seconde soutenance, travailler demande de plus en plus de motivation et les avancées se remarquent de moins en moins.

Ces dernières semaines ont été dures en termes de communication avec certains. De plus, de mon côté, arriver à finir le jeu comme on l'espérait me semblait compliqué, mais finalement je suis fier de ce qu'on a produit malgré quelques points à améliorer.

6.3 Mathéo

La développement de ce projet a été très instructif pour moi. J'ai appris énormément de choses durant ces mois de développement.

D'une part, j'ai pris connaissance du monde complexe de la création de jeu vidéo. Même si l'éditeur Unity est censé faciliter les choses, il reste difficile à comprendre et à prendre en main au départ.

D'autre part, j'ai appris l'importance de la communication dans un groupe et le fait qu'elle soit à double sens.

En terme de compétences, j'ai beaucoup appris sur chacun des domaines concernés par la réalisation de jeux vidéo. Que ce soit dans le développement d'intelligence artificielle, les jeux en réseau, les graphismes ou bien la création de site internet, ce projet m'a beaucoup apporté !

Ma seule peine est le fait que la communication n'ai pas été l'une des meilleures dans notre groupe.

6.4 Lucas

La réalisation d'un jeu vidéo n'est pas une tâche facile, cela m'a demandé et nous a demandé un investissement conséquent. Cependant, j'ai trouvé cela très enrichissant et je suis aujourd'hui bluffé par le résultat obtenu.

Au début du projet, je trouvais cela très ambitieux de réaliser un jeu vidéo alors qu'aucun membre du groupe n'avait ouvert Unity, aucun membre n'avait réalisé de modélisation 3D et nous étions tous débutant dans le langage de programmation C#. Pendant le développement, il y a eu des hauts et des bas. La période avant la deuxième soutenance a été particulièrement difficile pour ma part, car nous avons ajouté de nombreuses fonctionnalités, mais chaque amélioration entraînait un nouveau problème. Nous avons donc été contraint de revoir nos exigences à la baisse, car la résolution de ces problèmes n'étaient pas simples...

Néanmoins, nous avons aujourd'hui réussi à avoir un jeu fonctionnel, agréable et nous avons globalement réussi à atteindre nos

objectifs. Malheureusement, j'ai un regret par rapport au comportement de notre IA qui n'est pas ce que nous espérions avoir. En effet, cette partie n'a pas été assez développé et nous n'avons pas consacré assez de temps à son développement.

En conclusion, je suis impressionné par ce que nous avons réussi à faire et par les progrès que nous avons fait en 5 mois. L'une de mes plus grande satisfaction est le fait que nous avons réalisé cela en complète autonomie sans avoir de cours comme on a l'habitude de recevoir. J'ai également bien aimé passer les soutenances qui nous ont permis de faire le point sur l'état du jeu et prendre un peu de recul sur ce que nous réalisions. C'était également agréable de recevoir des conseils de nos professeurs. Enfin, sur le plan humain, je me suis rendu compte de l'importance de communiquer et d'expliquer ce que nous faisions. Sinon, le jeu perd en cohérence et certains éléments essentiels se retrouvent lésinés.

6.5 Nos peines

Ce projet n'est pas de tout repos, en effet il nécessite une grande part de notre temps de travail. Les problèmes rencontrés ainsi que la fusion des différentes tâches individuelles nous prennent une grande partie de notre concentration et nécessitent parfois beaucoup de recherches. Par exemple, pour le menu pause, le retour au menu comportait des problèmes de connexion qui ont coûté à Mathieu plus d'une semaine de travail consacré au jeu. La synchronisation des obstacles pour que les joueurs voient tous les mêmes mouvements des obstacles a aussi été la source de nombreuses recherches du côté de Mathéo. De plus, le logiciel Unity a récemment changer la gestion de la collaboration entre développeur en passant de Unity Collaborate à Plastic. Ce changement a considérablement ralenti notre travail pendant une semaine de mise en place et d'adaptation à ce nouveau système. En outre, la réalisation du tableau des scores n'a pas non plus été une tâche facile. Les temps affichés n'étaient pas les mêmes sur les écrans des joueurs connectés à une même partie, ce qui n'est pas vraiment cohérent... Un nouveau travail de synchronisation a donc dû être

fait pour unifier ces informations. En général, dû à l'avancée du projet, les problèmes que nous avons rencontrés sont de plus en plus durs à résoudre.

6.6 Nos joies

D'un autre côté, nous avons acquis de nombreuses compétences. Ce projet nous a permis de développer nos capacités d'adaptation et nos connaissances dans plusieurs domaines insoupçonnés tels que la modélisation ou encore l'organisation et la pédagogie. Le travail en groupe nous permet de résoudre les problèmes plus rapidement en apportant de l'aide aux autres tout en leur expliquant ce qu'il ne va pas. Ensemble, nous avons fait évoluer ce projet efficacement. Nous sommes maintenant fiers de notre projet qui a été pour nous une expérience bénéfique et enrichissante.

7 Points d'améliorations

7.1 Notre jeu

Le développement de notre jeu s'arrête aujourd'hui mais de nombreuses améliorations sont encore faisables. En effet, nous n'avions pas des possibilités de création digne d'un studio de jeu vidéo !

Néanmoins, nous regorgeons toujours d'idées pour le compléter :

- De nouvelles cartes telles qu'un environnement en ville, sur une île et bien d'autres encore...
- Dans le mode solitaire, ajouter des animations au fantôme ainsi que la sauvegarde du skin avec lequel le meilleur temps a été réalisé.
- De nouvelles méthodes pour améliorer l'IA afin qu'elle soit plus aléatoire, qu'elle reproduise plus fidèlement un comportement humain, qu'elle ait des skins différents, qu'elle ait des animations, etc.

- De nouveaux personnages plus cohérents avec nos graphismes
- Une meilleure gestion des connexions, déconnexions entre les parties et du bouton *Home*.
- Une meilleure synchronisation des obstacles afin d'éviter que les joueurs traversent parfois certains colliders, ...
- Une meilleure compréhension du multijoueur pour apporter une meilleure synchronisation entre les données
- Une amélioration des performances globales.
- La possibilité de personnaliser les touches de déplacements, de saut ainsi de prise d'objets.
- Un système de niveaux de progression au fur et à mesure de la progression du joueur, ce qui pourrait lui permettre de débloquer des cartes et des personnages.

7.2 Notre groupe

Ce projet était le premier véritable projet en groupe de notre cursus à EPITA. Il nous a permis de voir les joies et les peines du travail de groupe, de la communication ainsi que celles de la répartition des tâches. Malgré nos rendez-vous assez régulier, nous pensons que nous pouvons améliorer notre ponctualité et notre régularité dans de futurs projets. De plus, tous les membres du groupe ne proposaient pas autant d'avancées sur le projet que d'autres, ce qui a pu créer du stress ou des tensions. Notre efficacité peut aussi être améliorée, choisir de meilleurs horaires pour travailler peut être une bonne solution même si c'est dur de concilier cours et projet. La communication étant un des points les plus importants lors d'un projet, elle peut toujours être améliorée. Chaque tâche était attribuées à un membre de l'équipe et celui-ci l'effectuait seul dans son coin la plupart du temps sans forcément expliquer aux autres ce qu'il avait réalisé. Plus d'explications et de mise en commun nous auront permis d'être plus efficace et mieux organisé.

8 Conclusion

Cela va maintenant faire 4 mois que la création du jeu *Best Of Ten* est lancé, nous avons acquis de nombreuses compétences et connaissances techniques dans la création de jeux vidéos à travers le moteur Unity. Au-delà de cet aspect, ce projet est également un grand pas en avant en terme d'autonomie, de travail en groupe et nous avons su mettre en place une démarche de projet en partant de rien.

Les conseils qui nous ont été donnés lors de la première soutenance nous ont aiguillé sur différents points auxquelles nous n'avions parfois pas pensé pour améliorer notre jeu, rendre celui-ci plus fluide et rendre son utilisation plus agréable. Nous remercions notre jury pour ces précieux éléments.

Ce projet se déroule à merveille dans l'ambiance de groupe. Effectivement, la cohésion de groupe est très efficace dans notre projet et nous permet de tirer vers le haut toute l'équipe en s'entraînant. Nous avons globalement respecté notre planning comme nous avions déjà réussi à le faire lors de la première soutenance et nous sommes prêts à implémenter toutes les nouvelles fonctionnalités pour la dernière soutenance.