

Party Blowout!

– Rapport de projet –



Titouan **GUIONNEAU**
Clovis **LECHIEN**
Jacques **REMY**
Lilas **REYNAUD**

21 Juin 2021

Table des matières

1	Introduction	4
2	Reprise du cahier des charges	6
2.1	Le groupe	6
2.1.1	Titouan GUIONNEAU - Chef de Projet	6
2.1.2	Clovis LECHIEN	6
2.1.3	Jacques REMY	7
2.1.4	Lilas REYNAUD	7
2.2	Notre projet	8
3	Avancée du projet	10
3.1	Multijoueur - Titouan	10
3.1.0.1	Comment fonctionne Photon ?	10
3.1.0.2	Les messages RPC et leurs applications	11
3.2	Mécaniques	13
3.2.1	Intelligence Artificielle - Titouan	13
3.2.2	Déplacements - Jacques & Clovis	16
3.2.3	Animations - Jacques	17
3.2.4	Objets utilisables - Clovis & Titouan	19
3.2.5	Minijeux	21
3.2.5.1	Course Fruitée	22
3.2.5.2	Shooter	22
3.2.5.3	Guess Who	23
3.2.5.4	Le système de rounds	23
3.3	Visuels	24
3.3.1	Interface - Clovis & Titouan	24
3.3.1.1	FPS / Guess Who	24
3.3.1.2	Course Fruitée	26
3.3.1.3	Écran de fin	26

3.3.1.4	Mise à jour des informations	27
3.3.1.5	Reconnaître les joueurs	27
3.3.2	Modélisation - Clovis	29
3.3.2.1	Les voitures	29
3.3.2.2	Les armes	30
3.3.3	Maps - Clovis & Jacques	32
3.3.4	Menu - Lilas & Clovis	34
3.4	Audio - Jacques & Clovis	37
3.5	Site Web - Lilas	37
3.6	Installation - Titouan	39
3.7	Créations auxiliaires - Lilas	40
4	Chronologie de l'avancée du projet	42
5	Récits individuels	43
5.1	Titouan Guionneau	43
5.2	Clovis Lechien	44
5.3	Jacques Remy	46
5.4	Lilas Reynaud	48
6	Conclusion	50
7	Annexes	52
7.1	Bibliographie	52

1 Introduction

Ce rapport de projet a pour but de vous présenter le projet de notre groupe **BigHappyDove**. Ces six derniers mois, nous avons développé *Party Blowout!*. Notre jeu vidéo, composé de ses trois minijeux est arrivé à son stade final. Nous sommes fiers de vous présenter ce qu'il en est.

Nous avons à présent trois minijeux qui, ensemble, amènent le joueur dans une ambiance enfantine et amusante. Cela se fera, soit en lui rappelant un pique-nique en famille à travers *la course fruitée*, en le ramenant dans l'ambiance d'un cache-cache revisité avec notre *Guess Who*, ou bien en le faisant voyager vers ses vacances en montagne avec *la bataille de boule de neige*.

Ces derniers mois ont marqué la dernière ligne droite avant ce moment : le rendu final. Nous avons donc dû tout mettre en place afin d'atteindre l'objectif que nous nous étions donné dès le cahier des charges. En effet, nous savions que l'idée d'avoir trois minijeux serait un challenge, ce qui, maintenant terminé, nous rend fiers d'avoir eu cette ambition, qui a su porter ses fruits.

Dans ce rapport, nous allons tout d'abord reprendre notre cahier des charges, afin de vous rappeler nos objectifs initiaux. Ensuite, nous ferons un récit de l'évolution de chaque partie de notre projet. Celui-ci reviendra sur notre progression depuis le début du projet, mais nous développerons dans de plus grandes profondeurs nos derniers ajouts. Nous suivrons ce récit par nos ressentis personnels. En effet, nous avons tous appris de nouvelles choses, et tiré des leçons de ce projet, nous allons donc vous les détailler. Finalement, nous ferons notre dernière

conclusion en attendant la soutenance finale pour la présentation de notre jeu complet.

Nous vous souhaitons, une dernière fois, bonne lecture !

2 Reprise du cahier des charges

2.1 Le groupe

Notre groupe, que vous commencez maintenant à bien connaître, est **BigHappyDove**. Il est composé de Titouan Guionneau, notre chef de projet, Clovis Lechien, Jacques Remy, et Lilas Reynaud. La formation de ce groupe s'est faite plutôt naturellement l'an dernier. Le choix de notre chef de groupe été raisonné. En effet, Titouan ayant plus d'expérience en programmation, et surtout en projet de groupe dans ce cadre, il a bien su nous encadrer. Les autres membres ont chacun apporté de leurs qualités personnelles à la création de notre jeu *Party Blowout* !.

2.1.1 Titouan GUIONNEAU - Chef de Projet

J'ai commencé la programmation il y a très longtemps par de petits programmes en Python. J'ai fait beaucoup de LUA sur Garry's Mod pour faire des modes de jeu / addons. Je suis plutôt familier avec le travail en équipe dans le cadre de la programmation, j'espère de tout mon cœur qu'on sera apte de mener à bien ce projet !
On m'a donné la *lourde* responsabilité du rôle de 'Chef de projet'. Ce sera pour moi une manière d'apprendre à mener un projet comptant plusieurs membres.

2.1.2 Clovis LECHIEN

Depuis la primaire je suis baigné dans le monde merveilleux des jeux-vidéos. Cependant, c'est en terminale que j'ai ressenti le besoin d'aller explorer plus loin ce monde et notamment la programmation. La spécialité ISN était donc un bon compromis pour m'initier à la programmation et aux travaux en groupe. Comme vous avez pu le

remarquer en lisant mon résumé, je suis un novice en programmation ; cependant, je compte faire de cette faiblesse un atout en mettant à profit ma motivation grandissante envers ce milieu.

2.1.3 Jacques REMY

Depuis mon jeune âge, j'ai l'ambition de consacrer ma carrière professionnelle aux métiers du numérique et j'ai rapidement compris que cela passerait par la validation d'un diplôme d'ingénieur. Très tôt sensibilisé à l'informatique, j'ai commencé à apprendre le python et le C++. Par simple curiosité au début, puis naissant un réel engouement et surtout une envie d'en apprendre davantage. D'où le choix de la spé ISN l'année dernière, avec laquelle j'ai pu réaliser mes premiers projets de groupe en JAVA. Enfin, en dehors de mon appétit pour le numérique, je suis une personne ouverte d'esprit, plutôt bon camarade. Mon équilibre personnel passe par la pratique du théâtre, du sport et du scoutisme. Le travail en équipe ne devrait donc pas être un problème !

2.1.4 Lilas REYNAUD

Je n'avais jamais réellement fait d'informatique poussée avant mon arrivée à l'EPITA, mais c'est domaine qui m'a toujours intéressé. Étant une personne très curieuse, et ayant exploré de nombreuses choses autant dans les arts que dans les sciences, l'ingénierie informatique est pour moi une opportunité de relier la création et l'apprentissage scientifique. Ainsi, malgré mon manque d'expérience en programmation, j'espère que ma curiosité et ma motivation me permettront d'être un membre efficace de ce projet.

2.2 Notre projet

Nous voulions, dès le début, faire un jeu coloré, accueillant, dans un esprit joyeux, d'où son nom si festif. Nous nous sommes alors orientés vers l'esprit de jeux tels *Mario Party* ou *Fall Guys*.



Mario Party



Fall Guys

Nous avons donc, à présent, un jeu vidéo multijoueur dans lequel le joueur pourra soit héberger, soit rejoindre une partie. Celle-ci sera composée d'une suite de minijeux : un jeu de voitures, *la course fruitée*, un FPS avec *la bataille de boules de neige*, et enfin un cache-cache entre les joueurs et des intelligences artificielles, notre *Guess Who*.

Un de nos buts majeurs était d'apprendre à apprendre, et de découvrir de nouveaux domaines. Nous pouvons dire que cela a été le cas pour chacun des membres de notre groupe. Que cela soit passé par des nouvelles connaissances en programmation C#, ou la découverte d'outils tels Blender et Unity, ou même la création d'un site internet ou de contenu audio, nous avons tous été enrichis par ce projet.

Un autre intérêt important du projet est le travail de groupe. En effet, nous avons du répartir les tâches équitablement, en prenant en

compte les qualités de chacun des membres. De plus, nous devions réussir à gérer les difficultés humaines auxquelles nous avons fait face, ce qui est difficilement évitable pour un projet ambitieux et de longue durée. Nous avons du rester attentifs à l'avancée de chacun, et adapter si besoin nos objectifs tout au long du projet. Ces acquis seront tout aussi utiles pour nos projets futurs, en tant que cadre, si nous venions à gérer une équipe.

Pour discuter des jeux vidéos qui ont inspiré *Party Blowout!*, il est compliqué de retracer l'histoire des jeux se basant uniquement sur des minijeux. Cependant, certains jeux sont constitués d'une série de minijeux réunis. C'est le cas pour *Wario Ware* et *Mario Party*, disponibles sur les différentes consoles.

Les joueurs PC n'ont cependant pas été dépourvus de ce type de jeux : sur *Counter Strike : Source* et *Counter Strike : Global Offensive*, des minijeux ont été créés par leur communauté.

Garry's Mod, tournant sur le même moteur de jeu que les deux derniers, compte aussi beaucoup de minijeux différents créés par les joueurs (*DeathRun*, *Team Deathmatch*, *Murder*, *Melonbomber*, *Sandbox*, *Trouble In Terrorist Town...*).

3 Avancée du projet

3.1 Multijoueur - Titouan

Le multijoueur étant au coeur de notre projet, il a été mis en place dès le début. Tous nos scripts doivent pouvoir fonctionner de manière synchronisée entre tous les joueurs. Cette fonctionnalité est implémentée grâce à Photon, qui fonctionne en *Peer To Peer*, comme expliqué dans le cahier des charges. Dans Photon est inclus un système de lobby qui est directement géré par un serveur tiers. De notre côté, nous devons principalement nous occuper des joueurs et du fonctionnement du lobby.

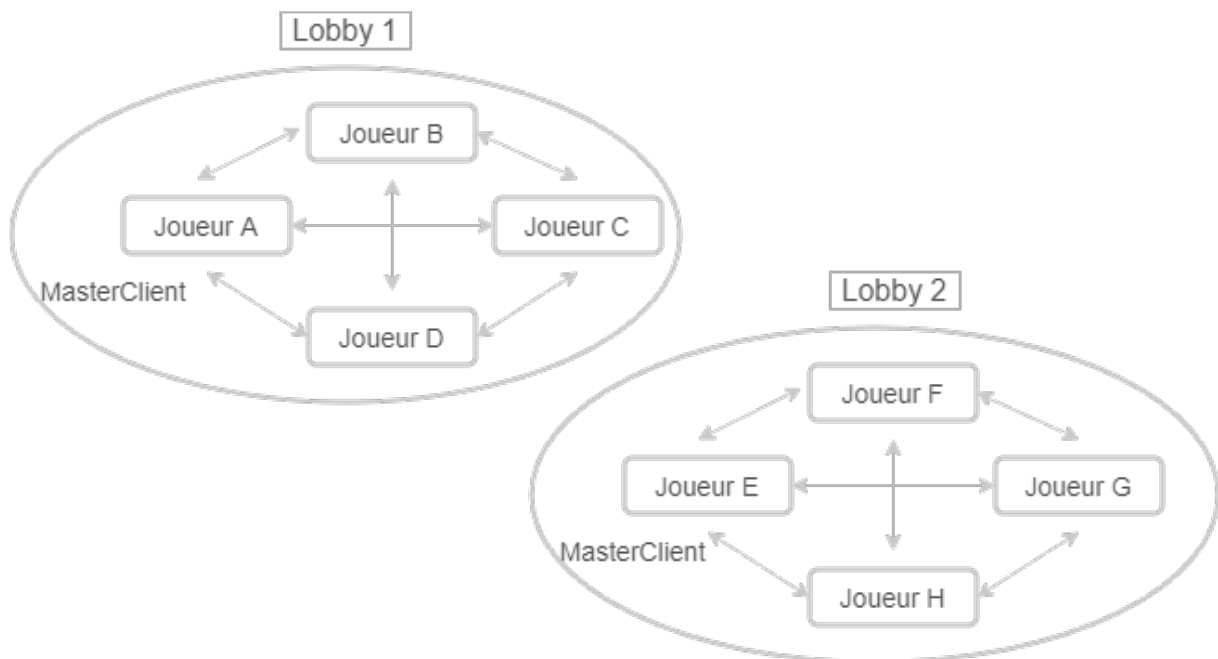
Nous avons très rapidement implémenté le multijoueur dans le menu, pour donner l'accès au joueur à un système de matchmaking. De ce fait, l'utilisateur peut héberger une partie ou rejoindre une partie (de manière aléatoire ou avec l'aide d'un identifiant).

3.1.0.1 Comment fonctionne Photon ?

La partie n'est pas hébergée par un seul joueur, en effet, tout le monde héberge certains éléments : Par exemple, un joueur A va envoyer aux autres joueurs sa position, sa rotation, ses mouvements et plus, il en est de même pour le joueur B, C etc... On dit que les joueurs gagnent la propriété de certains objets et donc ont la responsabilité d'envoyer aux autres joueurs les informations concernant ces objets pour que tout le monde puisse être synchronisé.

Cependant, dans chaque lobby est désigné un *Master Client*. Ce dernier a la grande responsabilité d'héberger les éléments principaux pour

le jeu, comme les IAs, les objets au sol, certaines informations concernant la map... autrement dit, tous les objets et les informations qui sont générés lors du chargement de la map. Ce *Master Client* doit aussi prendre le relais dans le cas où un joueur ne peut pas assurer l'hébergement d'un objet. Par exemple lorsque le joueur se déconnecte, on voudrait éviter que certains objets soient désynchronisés ou bien disparaissent. Le schéma suivant résume ce qui a été dit à l'instant :



les flèches correspondent aux données échangées, nous pouvons voir qu'il n'y a aucun serveur gérant la partie.

3.1.0.2 Les messages RPC et leurs applications

Photon supporte aussi les messages RPC, c'est-à-dire des messages pour déclencher une ou des fonctions chez les joueurs. Ces messages peuvent être envoyés par n'importe qui et à n'importe qui en fonction

de la situation. Ils peuvent aussi transporter des valeurs qui seront injectées en argument dans les fonctions.

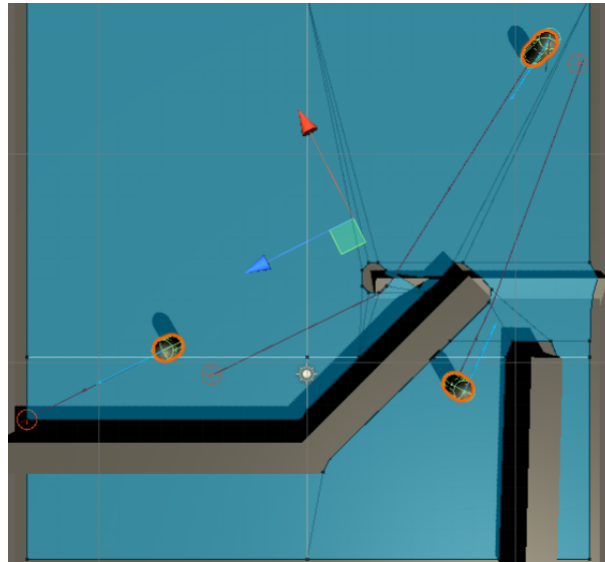
Entre la première et la deuxième soutenance, la majeure partie des nouveautés liées au multijoueur étaient la correction de quelques bugs. Pour ceux-ci, le système de messages RPC nous a beaucoup aidé, notamment concernant les problèmes pour partager les attributs des joueurs sur le réseau. Le reste du multijoueur était simplement son implémentation dans les différentes parties du projet au fur et à mesure de son développement.

Pour la soutenance finale, nous avons préparé le multijoueur pour les modes de jeux en synchronisant certaines variables essentielles (par exemple le temps restant, ou bien le score etc...). De plus, le multijoueur est prêt pour gérer les passages d'un mode de jeu à un autre dans les meilleures conditions (grâce aux synchronisations avec les messages RPC). La plus grande difficulté était de conserver certaines informations concernant les joueurs, tout en supprimant / mettant à zéro d'autres informations qui devaient être mises à jour. Cependant, nous avons un système stable qui permet de changer de mode de jeu pour tout le monde et de manière synchronisée.

3.2 Mécaniques

3.2.1 Intelligence Artificielle - Titouan

L'intelligence artificielle est utilisée dans le minijeu du *Guess Who*. Dans celui-ci, elles doivent se déplacer de manière aléatoire, mais naturelle. Unity nous a énormément facilité la tâche, en ce qui concerne le déplacement des IAs. En effet, il est généré grâce à un *NavMesh*¹, qui est une zone signifiant à l'IA où est-ce qu'elle peut marcher. Elle peut alors aller d'un point A à un point B de manière autonome et en trouvant le chemin le plus court. Pour la première soutenance, nous avions des intelligences artificielles qui reproduisaient ce type de mouvements :



Une NavMesh avec trois IA se déplaçant. Nous pouvons voir la NavMesh en bleu et la trajectoire des agents.

Cependant, à ce stade-là de l'avancement du projet, leurs déplacements n'étaient pas du tout naturels. En effet, elles essayaient d'opti-

1. En français : surface de navigation

miser leur parcours entre les points aléatoirement choisis comme points d'arrivée successifs. Ainsi, elles avaient tendance à longer les murs, ce qui les rendait très facilement différenciables par rapport aux joueurs.

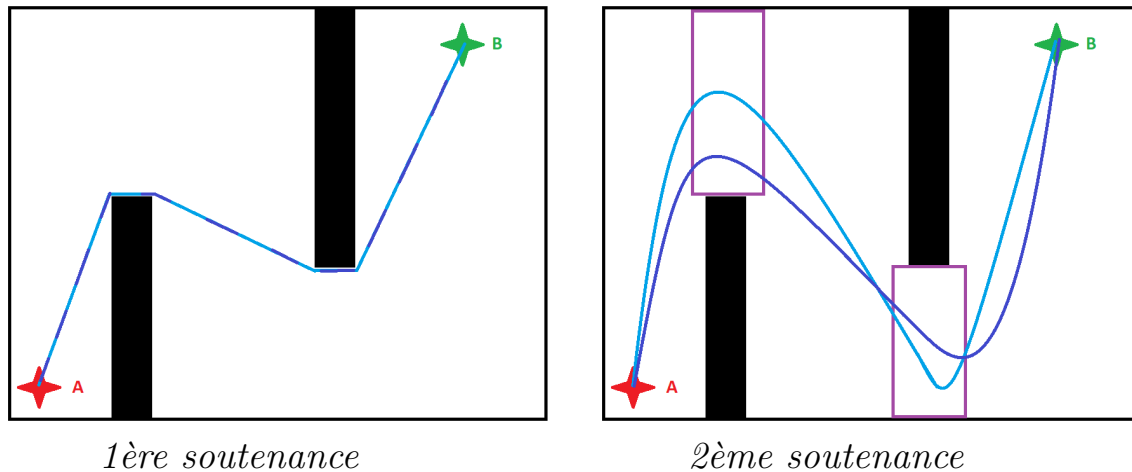
Pour résumer, voici les problèmes concrets que nous avons entre la première et deuxième soutenance :

- Leur vitesse était constante, contrairement à un joueur qui peut marcher, courir ou bien s'arrêter.
- La rotation des IA ne corrélait pas avec leurs déplacements, on se retrouvait donc par exemple avec des IA qui marchaient toutes à droite par rapport à leur rotation.
- Leurs trajectoires étaient "trop" optimisées, de ce fait les IA n'hésitaient pas à longer tous les murs si cela pouvait faire gagner quelques mètres.
- n IA partant du même point A pour aller au même point B , avaient toutes la même trajectoire, dû au problème énoncé ci-dessus. De ce fait, les IA se suivaient toutes les unes derrière les autres, même si cela implique à être collées.

Pour la deuxième soutenance, leurs déplacements ont donc été revus. En effet, nous leur avons donné la capacité de changer de vitesse, entre la course, la marche, et l'arrêt, tout cela aléatoirement. De plus, nous avons créé un système de checkpoints chaînés. Dans la map sont placés plusieurs checkpoints.

Chacun d'eux est lié aux checkpoints voisins. Lorsque l'IA traverse un checkpoint, elle est redirigée vers un checkpoint voisin et ainsi de suite. Cela permet à l'IA de garder un objectif proche, ce qui l'empêche de trop se coller au mur en voulant aller vers un objectif se trouvant derrière plusieurs murs. De plus, chaque checkpoint est en réalité une zone vaste et non un simple point. Lorsque l'IA doit se diriger vers un checkpoint, elle se dirige en réalité vers un point aléatoire contenu

dans cette zone. Les schémas suivants mettent en valeur les différences entre la première et la seconde soutenance (le violet représentant les checkpoints) :

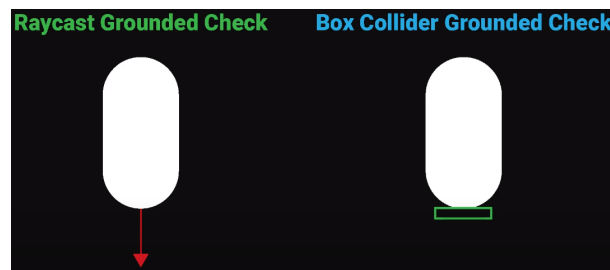


Nous avons donc réussi à obtenir un déplacement beaucoup plus naturel, afin de proposer un jeu intéressant aux détectives du *Guess Who*. Bien évidemment, nous avons aussi animé les mouvements de l'IA, pour qu'elle puisse marcher comme un vrai joueur.

Pour la dernière soutenance, les mouvements de l'IA ont de nouveau été revisités. En effet, il arrivait que l'IA fasse des moonwalks dignes de ceux de Michael Jackson. Nous avons donc rapidement corrigé ces petits problèmes. De plus, nous nous sommes aperçus que les IA ne faisaient pas de bruit lorsqu'elles marchaient, contrairement aux joueurs qui, eux, en font. Nous avons alors ajouté le bruit de marche aux IAs pour que les joueurs ne puissent pas être démasqués à cause de leurs bruits de pas. Dernièrement, nous avons appliqué les mêmes textures aux joueurs qu'aux IAs, *sinon ce serait trop simple*.

3.2.2 Déplacements - Jacques & Clovis

Nous avons créé un script de déplacement complet pour le joueur à pied dès la première soutenance. Ce script comprend une vision libre en première personne, des mouvements bidirectionnels, ainsi que des ajouts comme une touche de sprint ou bien des sauts et doubles sauts, qui doivent par avance vérifier si le joueur est au sol. Pour ce faire, nous n'avons pas utilisé de *Raycast* mais plutôt une méthode de *Box Collider* qui consiste à placer une fine boîte rectangulaire invisible sous l'objet du joueur.



De plus, afin de rendre le mouvement plus "lisse", nous avons utilisé la méthode *SmoothDamp* qui est plutôt compliquée à comprendre mais qui nous donne un très beau rendu. Enfin, en ce qui concerne les forces appliquées sur le *GameObject* telles que la gravité, nous avons simplement utilisé le composant *Rigid Body*.

Nous avons aussi commencé les déplacements des véhicules pendant cette première période de travail. Cependant, entre les deux soutenances intermédiaires, nous nous sommes rendu compte que les *Wheel-Colliders*, que nous avons commencé à mettre en place à la première soutenance, sont bien trop poussés pour ce que nous voulions faire. En effet, ils nécessitent une structure et une implémentation extrêmement réalistes pour obtenir un résultat qui fonctionne correctement. Notre objectif est plutôt d'avoir un jeu avec des déplacements rapides qui

laissent la place au joueur de s'amuser et de commettre des erreurs s'il avait l'imprudence de vouloir aller trop vite. De ce fait, nous avons choisi d'utiliser l'alternative du *Rigidbody*. Il a alors fallu utiliser une sphère qui fait office de "moteur" pour la voiture. En effet, la voiture n'est en réalité qu'un modèle 3d sans réelle hitbox. À l'opposé, la sphère n'est pas visible mais possède une hitbox et c'est sur celle-ci que sont appliquées les forces nécessaires pour faire avancer le véhicule.

3.2.3 Animations - Jacques

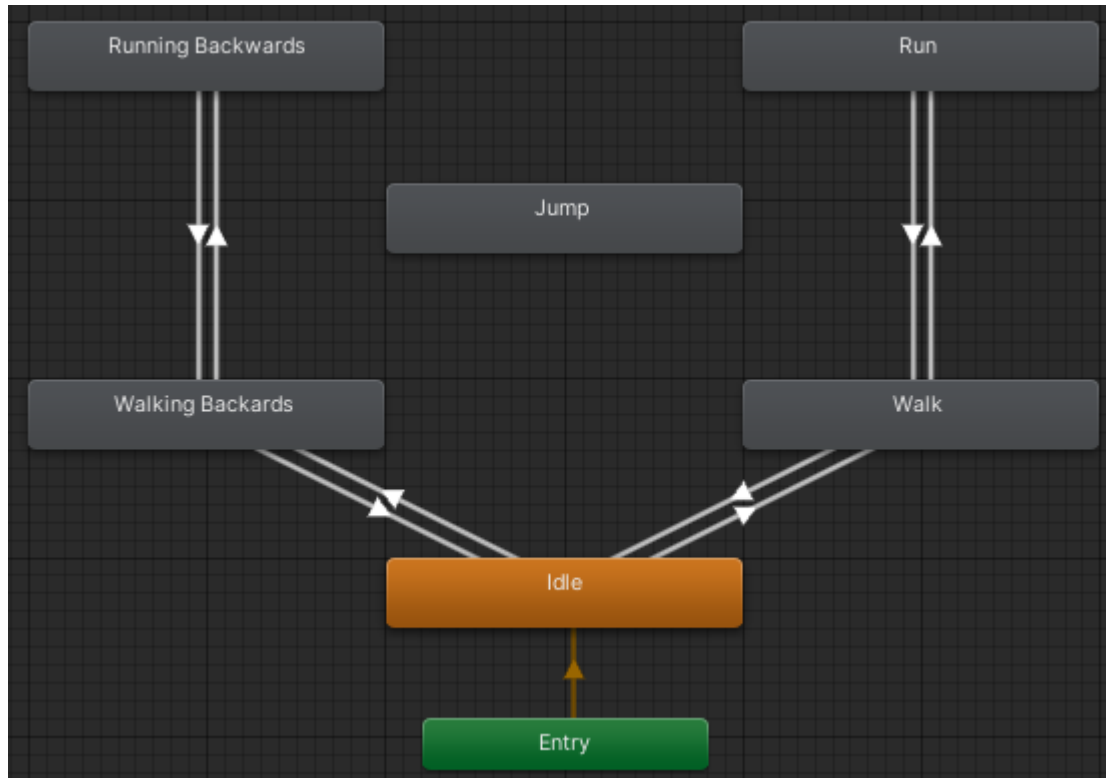
Concernant les animations, nous nous sommes concentrés sur cet aspect du projet entre les deux soutenances intermédiaires.

L'animation du personnage doit être divisée en 2 parties :

- Le modèle 3D qui est composé d'un squelette avec différentes articulations.
- L'animation qui, de son côté, définit l'ensemble des mouvements de chaque articulation de notre modèle.

Les animations peuvent donc être implémentées sur n'importe quel modèle ayant les mêmes articulations. Nous avons d'utiliser un modèle humanoïde. Nous nous concentrerons ici sur la deuxième partie : l'animation du modèle.

L'animation passe par un **animator**. C'est un composant qui fait le lien entre notre modèle (ou plus précisément son *Rig*) et l'ensemble des animations dont nous avons besoin. Pour se faire nous avons créé un script *animationStateController* qui nous a permis de relier l'**animator** au personnage.



Animator component

Nous pouvons observer ci-dessus plusieurs rectangles désignant chaque "état d'animation" (en anglais : *animation state*). Ils sont reliés entre eux par des flèches caractérisant des transitions entre chaque état. Sur ces transitions, nous avons appliqué un booléen (par exemple : *isWalking*) que nous utiliserons dans notre script.

Notre script récupère chaque composant dans l'**animator** et se compose principalement de conditions. Par exemple : si le joueur n'avance pas, c'est-à-dire que *isWalking* est à **False** ; et s'il appuie sur la touche avancer : on passe l'attribut *isWalking* à **True**. Bien entendu, ceci est

un cas assez simple, mais plus on y ajoute d'animations, plus les conditions se complexifient, afin de prendre en compte chaque possibilité. De plus, afin d'optimiser au maximum notre script, nous avons transformé chaque paramètre en "Hash".

Ainsi, si le joueur ne fait rien, le personnage sera dans la partie "Idle" (l'état par défaut, caractérisée par la couleur orange), c'est-à-dire qu'il n'y aura qu'une simple animation de respiration. Si le joueur appuie sur la touche avancer par exemple, l'état passera de "Idle" à "Walk" et le booléen "isWalking" prendra la valeur **True**, puis, s'il appuie sur la touche accélérer : le personnage se mettra à courir en donnant la valeur **True** à "isRunning". L'effet inverse se produira, si le joueur lâche la touche, chaque paramètre prendra de nouveau la valeur **False**.

Une fois l'animation des personnages terminée, nous avons synchronisé chaque animation en multijoueur, et ce, grâce au composant *Photon Animator View*. Celui-ci permet de synchroniser les paramètres. Nous avons choisi le mode "*Discrete*" qui synchronise seulement lorsque l'attribut sera modifié.

3.2.4 Objets utilisables - Clovis & Titouan

Avant la première soutenance, nous avons déjà implémenté une classe d'armes complète avec des attributs directement modifiables dans Unity (par exemple une cadence de tir modulable, des dégâts personnalisables, des chargeurs à tailles variables, etc.). Cette classe a été faite de manière à pouvoir être modifiée au cours du projet pour y ajouter de nouvelles fonctionnalités. Nous avons également introduit un système d'inventaire d'armes qui se parcourt avec le scroll de la souris pour se balader entre les différentes armes que le joueur possède à un moment donné. Le tir se faisait à l'aide d'un RayCast, qui par-

courait une certaine distance en regardant si elle rencontrait un joueur sur un passage.

Pour la seconde soutenance, la plus grande partie de notre progrès concernait l'adaptation pour le fonctionnement en multijoueur, qu'il s'agisse des armes elles-mêmes ou du système d'inventaire. Nous avons utilisé le système d'événements proposé par Unity, ainsi que les messages RPC de Photon. Le mélange des deux a offert une réelle synergie pour gérer les objets utilisables en multijoueur.

Pour la soutenance finale, nous avons modifié quelques aspects au niveau des armes, ce qui les rendent plus amusantes à utiliser :

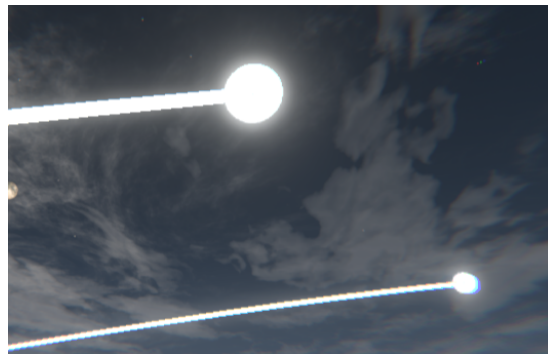
- **Des modèles uniques d'armes ont été créés**, par Jacques et Clovis. Il existe trois modèles d'armes (Fusil d'assault, Pistolet, Sniper) comme précisé dans la partie "Modélisation".
- **Le muzzleflash** a été revisité (en plus d'être maintenant synchronisé pour les autres joueurs). Nous avions auparavant des images qui apparaissaient lorsqu'on tirait. Ce n'est maintenant plus le cas. Nous utilisons maintenant le *Particle System* inclus dans Unity. Cela permet d'avoir un muzzleflash en 3D émettant de la lumière. De ce fait, il est davantage réaliste.



Le muzzleflash actuel. Nous pouvons voir qu'il émet de la lumière.

- **Les projectiles** sont complètement différents. Auparavant, comme décrit ci-dessus, les projectiles étaient seulement des Raycast invisibles pour les joueurs. Si ce Raycast rencontrait un autre joueur, alors ce joueur perdait de la vie.

Le Raycast était peu satisfaisant pour nous comme pour les joueurs. Nous les avons alors remplacés par des boules de neiges qui sont projetées d'une certaine force dans la même direction que le canon. Cela permet un aspect plus satisfaisant et plus modulable (par exemple, nous pouvons à présent changer la force soumise lors du tir : pour le sniper on souhaitera une balle plus rapide que celle du pistolet). Les balles sont suivies par une traînée blanche en plus d'émettre légèrement de la lumière comme montré ci-dessous :



Deux boules de neiges tirées à partir d'une arme.

3.2.5 Minijeux

Au début du projet, nous avons les bases pour créer les mécaniques de jeu de chacun des minijeux, cependant, ayant tous travaillé sur des choses différentes, cela ne donnait aucun jeu visiblement jouable. Les minijeux ont réellement commencé à prendre forme surtout entre les deux soutenances intermédiaires, et plus particulièrement la course

fruitée, qui était déjà jouable lors de cette seconde soutenance. Pour plus de détails sur l'avancée de chacun des minijeu :

3.2.5.1 Course Fruitée

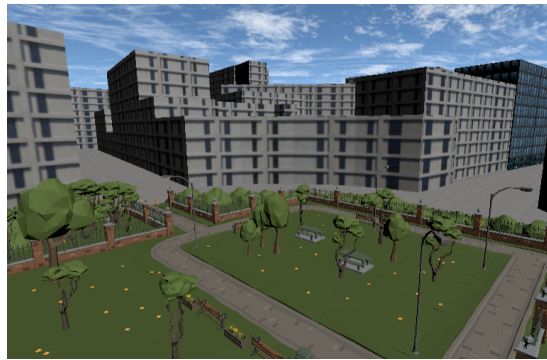
Dès la seconde soutenance, les voitures étaient prêtes, ainsi que le système de checkpoints pour les circuits. Il restait tout de même quelques détails à régler, mais le mode de jeu était déjà largement jouable. Désormais, ces problèmes ne sont qu'un lointain souvenir, puisqu'à l'heure actuelle le mode de jeu est terminé ! On peut se cogner, se retourner, s'envoler ou même implorer dans d'atroces souffrances. Le bouton respawn est là pour nous sauver la mise et nous faire réapparaître au dernier checkpoint passé.

3.2.5.2 Shooter

Lors de la deuxième soutenance, ce mode de jeu était déjà jouable, même s'il manquait encore quelques éléments dont un système de points, pour finaliser le minijeu. Le côté esthétique, avec la map et les différentes modélisations, ont été laissés pour la dernière période de travail. Maintenant la carte est là et elle est prête à vous mettre des flocons plein la vue !

3.2.5.3 Guess Who

Lors de la deuxième soutenance, le système d'IA était déjà bien mis en place. Ce qu'il restait à faire concernait surtout l'interface, afin de permettre à la chasse de se passer au mieux, et le côté esthétique avec la carte. Désormais aucun de ces problèmes n'est d'actualité comme vous pourrez le voir avec cette grande map à l'effigie du quartier d'Epita.



3.2.5.4 Le système de rounds

Lors de la deuxième soutenance, nous n'avions rien permettant de lier les minijeu entre eux. Pour remédier à cela, nous avons créé un système de rounds, où à chaque fois qu'un round est passé, les joueurs sont transférés vers un autre minijeu de manière aléatoire. De ce fait, il n'y a pas de réelle fin, les joueurs continuent à jouer indéfiniment sur les différents modes de jeux.

Comme précisé dans la partie *Multijoueur*, le réel problème a été de synchroniser les informations importantes entre les joueurs.

3.3 Visuels

3.3.1 Interface - Clovis & Titouan

L'implémentation de l'interface a suivi la progression des minijeux. En effet, nous nous sommes rendu compte qu'il serait plus judicieux de la mettre en place une fois les minijeux concrètement mis en place. Ainsi, elle a commencé à prendre forme après la première soutenance, lorsque nous avons vu se former trois jeux indépendants, et non plus de simples bases utilisables comme lors de la première soutenance.

3.3.1.1 FPS / Guess Who

Pour les minijeux en FPS, autrement dit le *Guess Who* et la *bataille de boules de neige*, nous avons implémenté une interface simple pour la vie, où la couleur de la croix à gauche varie en fonction des points de vie du joueur. Nous avons aussi un décompte pour les munitions.

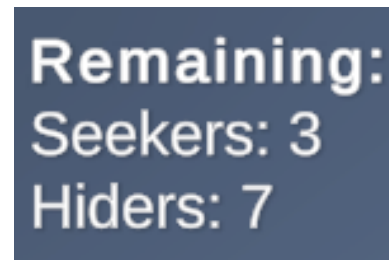
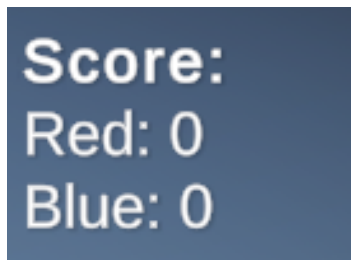


Une interface simple pour la vie, où la couleur de la croix à gauche varie en fonction des points de vie du joueur

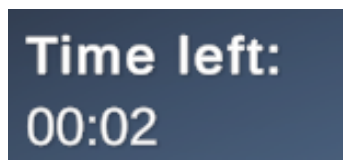


L'interface pour les munitions

L'interface proposée pour la seconde soutenance n'était pas suffisante pour les modes de jeux *Guess Who* et pour le *Shooter*. En effet, pour le *Shooter*, nous étions incapables de connaître le score de chaque équipe. Il en est de même pour le *Guess Who*, comment savoir le nombre de personnes restant à trouver ? Et surtout comment l'utilisateur pouvait savoir quand le round allait se terminer ? C'est pourquoi nous avons implémenter ces interfaces pour afficher les informations manquantes :



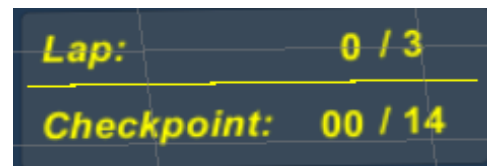
Le score pour le Shooter et les personnes restantes pour le Guess Who



Interface pour montrer le temps restant

3.3.1.2 Course Fruitée

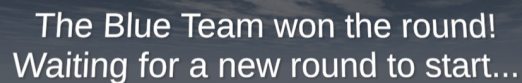
En ce qui concerne la *course fruitée*, l'UI comprend le meilleur temps de course, le nombre de tours courant, ainsi que le chronomètre de tour actuel, propre à chaque joueur. L'interface est assez claire et simple, elle permet au joueur d'avoir une idée sur son avancée dans le circuit tout en gardant un oeil sur son temps de course. Ceci permet un jeu plus clair et agréable pour les joueurs.



L'interface est assez claire et simple, elle permet au joueur d'avoir une idée sur son avancée dans le circuit tout en gardant un oeil sur son temps de course.

3.3.1.3 Écran de fin

De plus, nous avons ajouté un écran de fin lorsqu'une personne (ou une équipe), gagne une partie comme montré ci-dessous :

A screenshot of a game's end of round screen. It features a dark blue background with white text. The text reads "The Blue Team won the round!" followed by "Waiting for a new round to start..." on the next line.

The Blue Team won the round!
Waiting for a new round to start...

Écran de fin de round

À noter que nous sommes en train de préparer une transition pour passer d'un mode de jeu à un autre, mais nous ne l'avons pas encore implémentée. Cela permettra d'avoir quelque chose de moins rudimentaire.

3.3.1.4 Mise à jour des informations

Pour éviter que l'interface soit mise à jour en permanence, ce qui peut causer des pertes de performance sur le long terme, nous avons utilisé le système d'événement par Unity. Par exemple, lorsqu'un joueur se fait toucher par le tir de quelqu'un, cela va appeler un événement *OnTakeDamage* qui va par la suite appeler différentes fonctions² dont celle qui va permettre de mettre à jour la vie du joueur sur son interface. C'est exactement le même principe avec le jeu de course. Quand le joueur passe dans un checkpoint ou effectue un nouveau tour cette information est transmise aux autres joueurs du salon.

3.3.1.5 Reconnaître les joueurs

Nous avons rencontré un autre problème : comment pouvons-nous reconnaître nos amis dans le jeu ? Jusqu'à la deuxième soutenance, nous n'avions aucun moyen d'identifier les joueurs. De ce fait, nous avons ajouté un élément d'interface se trouvant au dessus de la tête des joueurs pour pouvoir les identifier, voir leurs points de vie ainsi que de savoir dans quelle équipe ils sont :

2. Cela va aussi envoyer un message RPC à tout le monde pour que l'attribut du joueur soit mis-à-jour



Un joueur de l'équipe bleue avec 50PV et un autre de l'équipe rouge avec 80PV

Nous pouvons aussi reconnaître les joueurs dans la *Course Fruitée* :



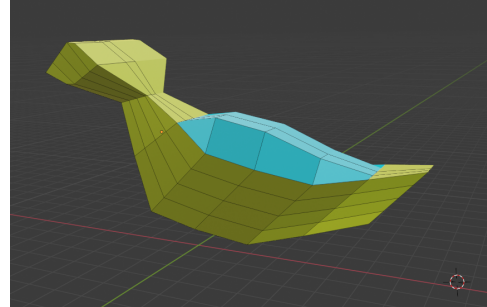
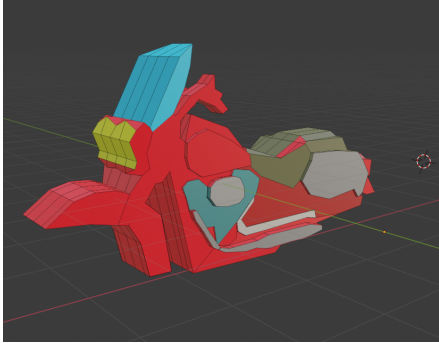
Bien évidemment, ces informations n'apparaissent pas dans le mode de jeu *Guess Who* pour ceux qui doivent se cacher parmi les joueurs.

3.3.2 Modélisation - Clovis

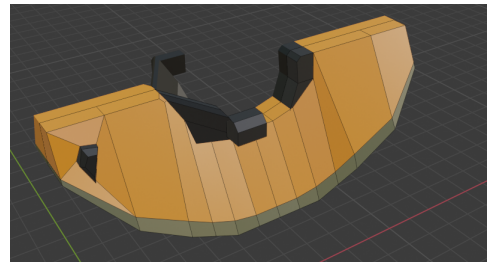
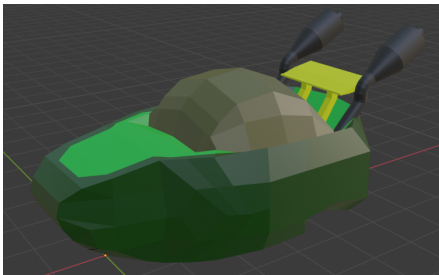
Pendant la première période de travail, nous avons testé des modèles pour quelques personnages différents, ainsi qu'un véhicule et des armes. À l'époque, ce n'était pas une priorité, le côté fonctionnel était à favoriser. De ce fait, nous nous occupions de la modélisation lorsque nous en avions le temps : entre deux grosses avancées par exemple pour se changer les idées et prendre du recul pour ne pas faire que du code. C'est au tournant suivant dans l'avancée du jeu que la modélisation a réellement pris une part importante du projet. Il ne s'agissait plus désormais de se concentrer seulement sur un jeu fonctionnel, il fallait aussi que ce-dernier soit agréable à jouer visuellement. Il est difficile d'imaginer que *Breath of The Wild* ait eu un tel succès s'il avait ressemblé à pong.

3.3.2.1 Les voitures

Pour *la course fruitée*, nous avons, avant la dernière soutenance, décoré le circuit avec de nombreuses modélisations faites maison : des bouteilles, des couverts, ou autre. Toutes ces modélisations pour amener l'aspect visuel du pique-nique de ce minijeu. Nous avons aussi modélisé quatre véhicules différents, dans le thème du jeu. Nous avons donc la Moto Cerise, la Formula Banane, l'Avocat Tuning, et le Scooter Melon.



La Moto Cerise en haut à gauche et la Formula Banane en haut à droite.



L'Avocat Tuning en haut à gauche et le Scooter Melon en haut à droite.

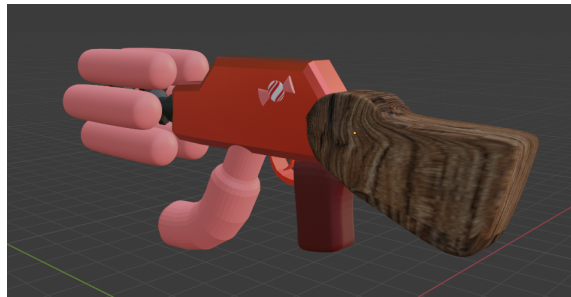
3.3.2.2 Les armes

Quel est le point commun entre le Guess Who et le FPS ?
Ah ? Une réponse peut-être ?..
BINGO ! Les armes en effet ! :)

Eh oui, pour ces deux minijeux il nous fallait des armes qui ont leur propre charme et qui ressemblent au reste de l'univers de notre jeu. Nous avons donc décidé de faire trois armes bien distinctes : un pistolet, un fusil d'assaut gatling et un sniper.

Le style grotesque de ces armes contribue à renforcer le côté comique et à atténuer la violence des deux minijeux concernés, en effet c'est toujours plus amusant de lancer une boule de neige sur son ami à l'aide de son lanceur gatling modèle Ak-47 compression à air avec recharge 600g throwback préféré plutôt que de lui tirer dessus à main nue.

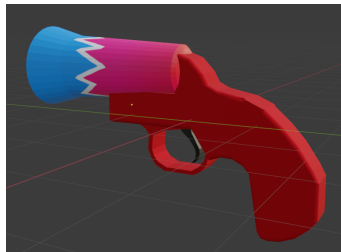
Dans le jeu ces trois armes possèdent des attributs différents et équilibrés afin d'attirer les joueurs à les utiliser dans des situations différentes, empêchant ainsi la sur-utilisation d'une seule arme considérée comme étant trop forte **tousse touse ak47 gatling touse touse**
Sans plus attendre, les photos des armes que vous pourrez utiliser dans *Party Blowout!* :



Le fameux fusil d'assaut à boules de neige modèle Ak-47 Gatling



Un sniper digne des plus grands snipers à boule de neige au monde ! Avec son canon élargie, sa puissance de tir inégalable, et sa lunette de visée ultra performante, vous ne pourrez plus en manquer un seul



Le très attendu Pistolet Neige avec son canon légèrement agrandi pour un étalement de la neige sur la tronche de vos ennemis plus efficace

3.3.3 Maps - Clovis & Jacques

Avant la première soutenance, nous avons schématisé les différentes maps sur papier. Cependant, leur réalisation concrète s'est faite plus tard. En effet, nous avons une map de test lors de la première soutenance qui comportait tous les éléments de base : des *spawn points*, des barrières et des éléments de décor.

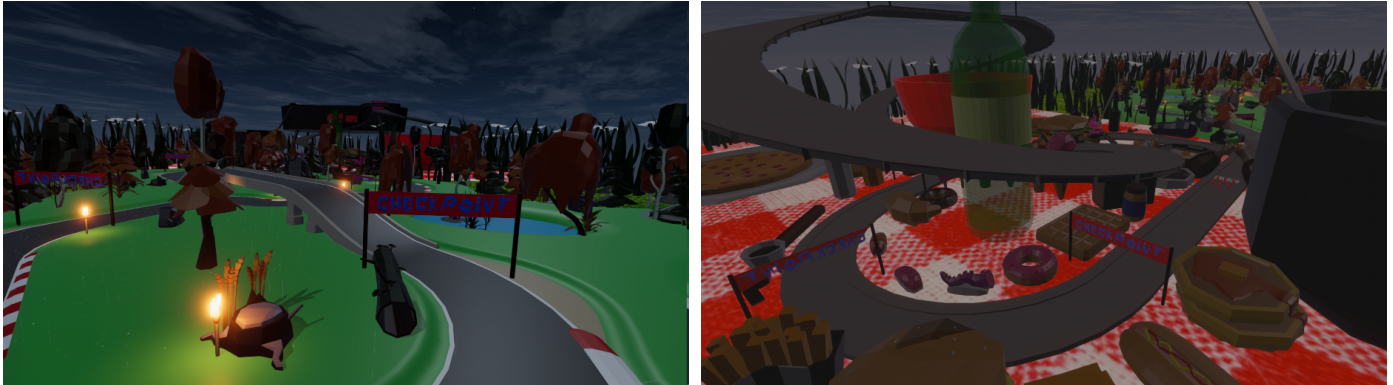
En ce qui concerne les points d'apparition : pour ne pas avoir à renseigner chaque coordonnée à la main (dans notre programme qui fait apparaître les joueurs sur la map), nous avons implémenté une

méthode qui consiste à placer des capsules, visibles dans l'éditeur de Unity mais qui disparaissent une fois la partie lancée. Afin de faciliter leur placement, il nous fallait trouver un moyen de facilement connaître leur rotation. Pour cela, nous avons attaché un cube orienté vers l'avant du *spawn point*, nous indiquant dans quel sens il se trouve. Maintenant, nous n'avons plus qu'à récupérer les composants à l'intérieur l'objet réunissant tous les points d'apparition.

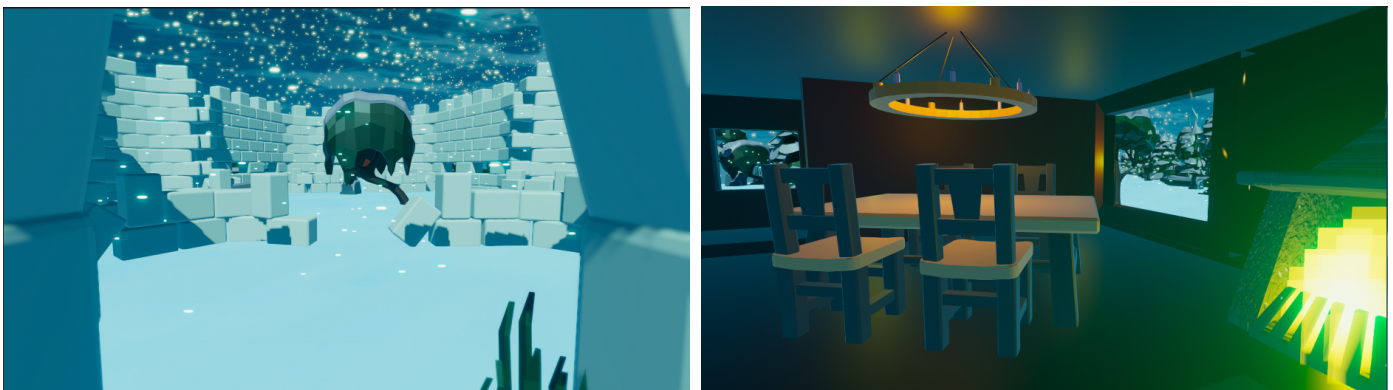
Le gros du travail, qui nous a approché des maps que nous avons sur notre projet final, s'est fait plus progressivement. Pour la deuxième soutenance, nous avons créé un système de Checkpoints pour le jeu de course. Ces derniers ne sont pas visibles pour les joueurs sur la map pour éviter de polluer le paysage du circuit mais ils sont fonctionnels et synchronisés. Pour les repérer sans faire apparaître la hitbox du collider, nous avons décidé d'opter pour une arche qui se fixe de chaque côté de la route. Le côté esthétique de la map de la *course fruitée* a aussi été créé durant cette période-ci.

Pendant cette même période de travail, les spawnpoints ont été légèrement retravaillés. Dès à présent ils peuvent faire apparaître n'importe quelle entité (Joueur ou IA). De plus, les entités réapparaissent sur les mêmes checkpoints lorsqu'elles meurent.

Toutes les maps de notre jeu sont à présent terminées, elles sont toutes passées sous la loupe de notre expert en post-processing Clovis (*hehe*), désormais elles sont belles et fonctionnelles.



Partie Forêt et Partie Nourriture de la carte Course Fruitée



Vue sur l'intérieur de la forteresse enneigée et d'une maison à proximité du feu :)

3.3.4 Menu - Lilas & Clovis

Le menu a évolué tout au long du projet. Lors de la première soutenance, nous avons implémenté les fonctionnalités principales du menu d'accueil et du menu de paramètres. Nous avons fait le choix de faire une page d'accueil offrant trois options à l'utilisateur. La première est de quitter le jeu. Le bouton correspondant est simplement lié à un

script pour fermer l'application. Nous avons aussi fait un lien vers un menu de paramètres, avec un curseur de volume.

La partie la plus importante du menu, est celle permettant au joueur de créer et/ou de rejoindre un salon. Dans cette sous-partie, le joueur a également la possibilité d'enregistrer un surnom personnalisé avec un champ de texte à remplir. S'il choisi de ne pas le faire, il gardera son pseudo aléatoire. Nous avons donc dû implémenter le système de lobby pour le multijoueur.

Le côté esthétique de celui-ci a été mis en place pour la dernière soutenance. Nous avons beaucoup de changements à réaliser dans ce domaine. En effet, la première version avait un but purement fonctionnel. Elle nous avait permis de nous familiariser avec les outils Unity des menus, mais nous savions que c'était temporaire. Nous avons donc changé les codes esthétiques. Désormais le menu est une scène à part entière, on peut y retrouver un morceau de chaque carte pour donner à l'utilisateur une idée des minijeux auxquels il va devoir jouer. Cette scène est équipée d'une caméra rotative qui se déplace dans chacune de ces mini-scènes.

En cette dernière période de travail, nous avons mis en place un menu de pause. Celui-ci permet notamment au joueur d'accéder aux paramètres, pour, par exemple, baisser le son du jeu au cours d'une partie. Il suit les mêmes codes esthétiques que le menu de paramètres principal. En effet, les boutons pour naviguer dans ce menu sont sur la gauche, sur un fond rose. Le côté droit est d'un bleu translucide, qui laisse transparaître la scène du minijeu sur lequel le joueur était lorsqu'il à appuyé sur la touche échap. Lorsque le menu est ouvert, le joueur qui l'a activé ne peut plus se déplacer sur la map, mais il reste visible en multijoueur.

L'aspect esthétique du menu d'accueil a de nouveau été mis à jour. Nous avons mis en scène les personnages à l'arrière du menu, qui avait déjà été rendu relativement transparent auparavant. En effet, la caméra tourne autour de représentations des ambiances de chaque jeu, pour mettre le joueur dans l'ambiance de *Party Blowout!* dès son arrivée dans le menu principal. Nous avons donc la représentation d'un panneau *Party Blowout!*. Sur un autre plan du menu, il y a un checkpoint de la *course fruitée* autour duquel sont placés des véhicules et des éléments de décors du pique-nique estival. Ensuite, nous sommes plongés dans l'hiver, avec la *bataille de boules de neige* et son ambiance givrée. Nous avons finalement une mise en scène pour le *Guess Who*, comprenant plusieurs personnages dans un village, que cela soit des IAs ou bien des potentiels joueurs.



Aperçu du menu principal en rotation

3.4 Audio - Jacques & Clovis

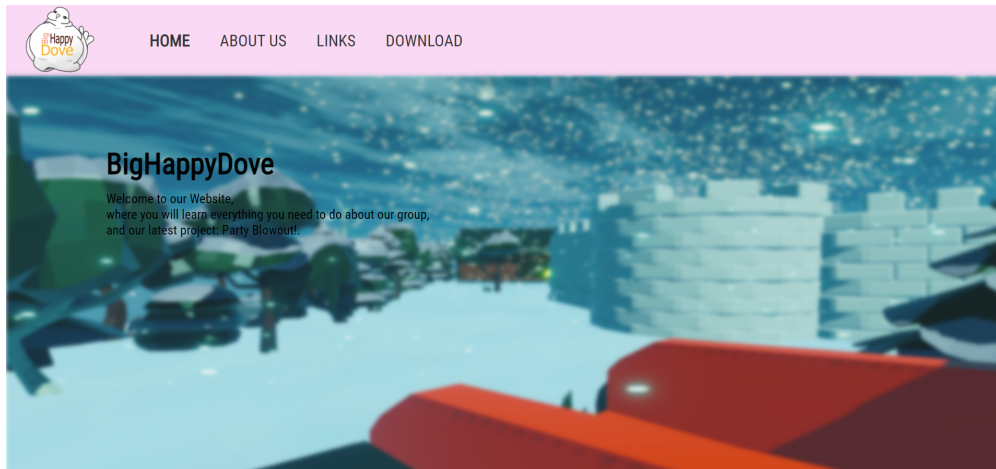
Dès la première soutenance, nous avons déjà créé notre musique d'accueil, la musique officielle de notre jeu. Nous avons ensuite implémenté des bruitages dans les minijeux, comme par exemple les bruits de tirs, ou ceux correspondants aux différents déplacements du joueur. Les cartes aussi possèdent chacune leur propre bande-son, cependant ces-dernières sont copyright © Nintendo alors *chuuuuuuttt.... (pas un mot ou c'est le procès)*.

3.5 Site Web - Lilas

Dès le commencement du projet, nous avons pris la décision de créer notre site internet entièrement, avec du code HTML, CSS, et même un peu de Javascript. Pendant le premier mois de projet, nous avons surtout mis en place le squelette en HTML. Nous avons créé les différents scripts représentant les pages du site et implémenter une bar de navigation entre celles-ci, ainsi que des sous-sections dans les différentes pages. Concernant le design, nous avons implémenté dans la page d'accueil un diaporama en Javascript en arrière plan. Nous avons aussi fait une base de CSS pour mettre la barre de navigation à l'horizontale, avec notre logo de groupe en haut à gauche. Cependant, à ce stade-là, le côté esthétique était encore en second plan.

Lors de la deuxième soutenance, il était alors temps de s'atteler au design du site. Ainsi, nous avons adapté les images du slideshow de la page d'accueil, pour qu'elles reflètent l'avancée du jeu. De plus, nous avons flouté ce diaporama afin qu'il ne pose pas de problème pour comprendre le texte en avant-plan. Nous avons aussi donné plus de profondeur aux boutons, en permettant qu'ils soient surlignés lorsque l'on déplace la souris au dessus de l'un d'entre eux. De plus, la page ouverte a été mise en gras dans la barre de navigation. Le contenu

du site, lui aussi, a été mis à jour. Nous avons adapté la timeline, et quelques difficultés rencontrées, ainsi que les liens vers nos rapports ou la bibliographie.

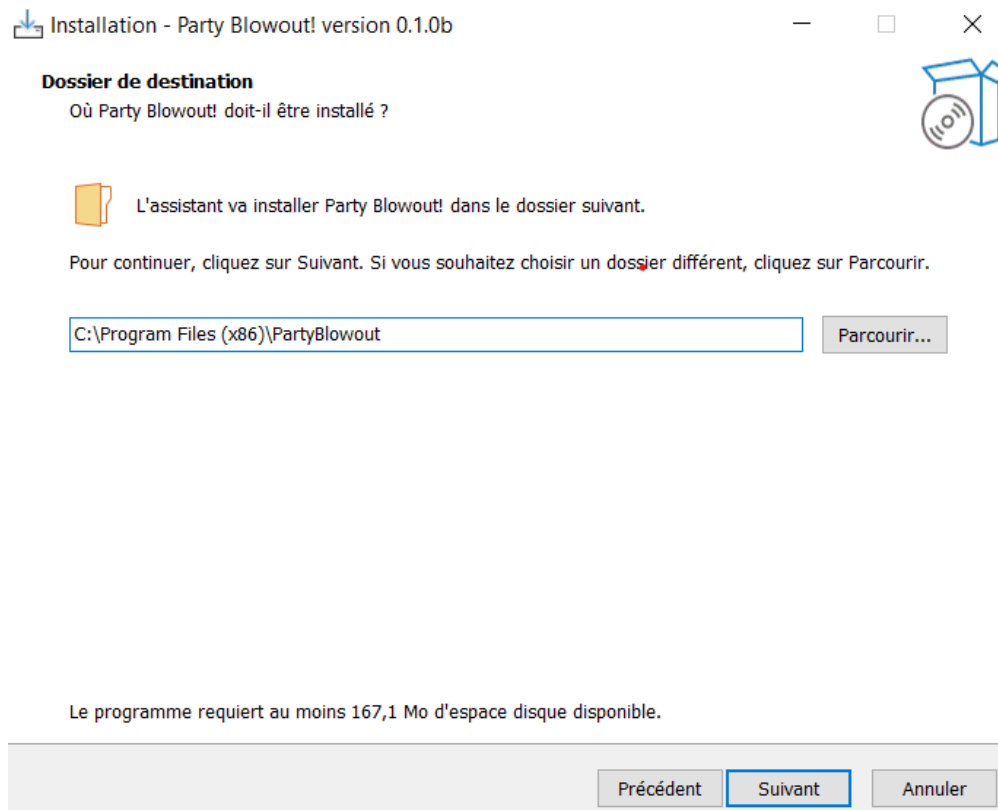


Page d'accueil du site internet

Pour le rendu final, nous avons adapté le contenu du site afin qu'il représente entièrement notre parcours de ces six derniers mois. Nous avons aussi certaines implémentations techniques à mettre en place, notamment pour télécharger le jeu. Des détails esthétiques ont aussi été développés, pour se rapprocher au plus possible d'un rendu professionnel, car malgré notre noviceté nous voulions créer le site de A à Z.

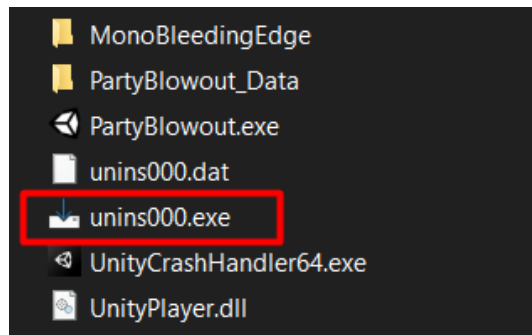
3.6 Installation - Titouan

Comme demandé pour le projet, nous avons créé un programme pour installer le jeu. Cela a pu être possible grâce à *Inno Setup* qui nous a grandement facilité la tâche.



Installation du jeu

Cependant, (du moins au moment où j'écris cela, (peut-être que j'aurais trouvé une solution d'ici la soutenance), je n'ai pas trouvé comment rendre la désinstallation *user friendly*). En effet, pour désinstaller le programme, il faut soit se rendre dans le panneau de configuration, soit exécuter le .exe dans le dossier du jeu.



Dossier du jeu avec le .exe pour le désinstaller

3.7 Créations auxiliaires - Lilas

Lors de cette dernière période de travail, nous devons nous intéresser à certains éléments qui touchent plus au côté commercial. Nous avons dû créer une boîte dans laquelle présenter notre jeu. Nous avons choisi d'utiliser une pochette de jeu DS. Nous avons créé la couverture de celle-ci, pour permettre aux joueurs potentiels de s'imprégner de l'ambiance de *Party Blowout!* dès le premier regard.



Voici la pochette qui habillera le rendu physique de notre jeu

4 Chronologie de l'avancée du projet

Soutenances	1ère	2ème	3ème
Réseau			
Multijoueur*	80%	100%	100%
Mécaniques			
I.A	60%	90%	100%
Déplacements	50%	80%	100%
Objets utilisables	30%	75%	100%
Minijeux	5%	50%	100%
Visuel			
Interface	0%	60%	100%
Modélisation	33%	70%	100%
Maps	33%	80%	100%
Menu	40%	40%	100%
Audio	30%	75%	100%
Site-Web	50%	80 %	100%

Vert : Objectif dépassé
Jaune : Objectif presque atteint
Rouge : Objectif non atteint
Sans couleur : Objectif atteint

* Correspond au système de lobby, les synchronisations seront gérées au fil du temps

5 Récits individuels

5.1 Titouan Guionneau

Ce fut un projet très intéressant et très riche sur plusieurs aspects, que cela soit sur le plan humain comme sur le plan technologique. Tous les membres du groupe ont pu apprendre à utiliser divers outils pour tenter de mener à bien ce projet.

En prenant un peu de recul, je pense qu'on aurait pu fournir davantage de travail sur le projet de manière générale, car le rendu est satisfaisant mais sans plus. On aurait pu peaufiner pour donner à l'utilisateur une expérience plus enrichissante.

Cependant je suis très content du travail qui a été effectué sur l'aspect graphique ainsi que sur la modélisation, c'est sûrement les deux parties qui ont été les plus poussées, ce qui compense énormément l'aspect technique du projet.

Même s'il y a eu plusieurs moments sans motivation, le groupe a bien travaillé dans sa globalité, et je pense qu'on peut être content d'avoir un jeu fonctionnel avec quelques features intéressantes.

5.2 Clovis Lechien

Ce projet a été pour moi mon premier projet de groupe. En effet étant novice dans le milieu de la programmation, ce-dernier a su confirmer mon choix d’avenir. Malgré la déception que j’ai pu ressentir au vu de mes notes sur les dernières soutenances je n’ai pas démordu et j’ai accentué ma charge de travail sur ce projet tout au long de son avancée.

Certes le résultat final n’est pas celui d’un jeu AAA mais il n’en reste pas moins bon pour autant. J’ai le sentiment d’avoir contribué au succès de ce projet et j’en suis très fier. Maintenant il ne me reste plus qu’à le partager à mes copains et les enfants de ma famille d’accueil où je résidais à Lyon qui débordent d’impatience à l’idée de le tester !

J’ai aussi l’ambition de continuer d’améliorer le jeu les après-midi des vacances où je ne travaillerai pas, j’ai très envie de rajouter des cartes à nos modes de jeu. Pourquoi pas même revenir à la rentrée prochaine avec un jeu complet qui possède au moins 2 cartes par gamemode, mais bon ne rêvons pas trop, il s’agirait aussi de profiter de ces vacances pour se reposer.

Si je devais regretter quelque chose dans ce projet ça serait sûrement le fait que le travail n’a pas été équitable pour tout le monde et c’est vraiment dommage car la première victime de ce problème est le jeu en lui-même. En effet certains aspects du projet reste assez vide et n’apporte pas grand chose à ce-dernier.

Mes tâches au niveau de ce projet ont été extrêmement diversifiées. J’ai eu l’occasion en début de projet d’apporter mon aide sur différents aspects comme les armes par exemple mais rien qui ne dépasse l’aide qu’a pu apporter Titouan au reste du groupe. J’ai eu la lourde

responsabilité de créer de A à Z le jeu de la course fruitée et je suis très fier du résultat : le jeu est dynamique, les voitures sympathiques et équilibrées et la carte belle. J'ai aussi eu le plaisir de pouvoir gérer l'aspect esthétique du jeu avec la création de deux cartes sur les trois et du menu, j'ai aussi géré le post-processing pour toutes les maps ainsi que la lumière. Enfin j'ai continué tout au long de ce projet la création de modèles 3d afin de rendre le jeu plus vivant et d'apporter aussi une dose de travail personnelle à cet aspect qui aurait pu être intégralement fait à l'aide de modèles sur Internet. A ce jour je cumule plus de 42h de travail sur Blender depuis le début de ce projet.

Pour conclure je dirai que ce projet fût une expérience agréable pour ma part. Certes il a eu ses doses de stress et ses périodes mortes mais je n'ai jamais cessé de fournir des efforts afin de rendre le jeu agréable pour les joueurs et pour moi-même. Parti d'une envie de créer un mix entre Fall Guys avec ses graphismes et Mario party et son gameplay, je suis fier aujourd'hui de vous présenter *Party Blowout!*



Merci de votre lecture :)

5.3 Jacques Remy

Déjà enfant, l'idée de créer un jeu vidéo était pour moi un rêve. Et aujourd'hui, ce rêve est devenu réalité.

Durant toute la période de création du jeu, j'ai pu apprendre énormément sur différents logiciels. Tout d'abord, avant même avoir débuté le projet, je me suis mis en tête de créer une musique. Alors curieux et enthousiaste à l'idée de créer quelque chose de nouveau pour moi, je me suis lancé dans la musique à l'aide du logiciel FL Studio, et, après de nombreux (très nombreux) essais, je suis parvenu à obtenir un résultat qui me plaisait. Puis, j'ai appris à utiliser Unity, principalement sur le développement des personnages, c'est-à-dire : leurs déplacements, leur animations ... En parallèle, j'ai aussi beaucoup utilisé le logiciel Blender, que ce soit pour la création d'un personnage (qui n'a jamais vu le jour finalement.. :c), la création d'assets ou bien la création de map. En particulier la création de la map du *Guess Who*, qui a sûrement été l'une des tâches les plus longue et compliquée que j'ai eu à faire. Entre la création des bâtiments, ayant les mêmes proportions que ceux présents dans le quartier d'Epita, la création de certains assets, ou même la création des textures qui, au passage, ne passaient pas de Blender à Unity (ou en tout cas très difficilement). Et enfin, ayant déjà quelques bases en after effect, je me suis amusé à créer une animation de machine à sous qui tire au hasard les minijeux, nous servant de transition. Sinon, je me suis aussi pas mal intéressé à la création de matériaux sur Blender et de shader sur Unity (comme la texture de la glace par exemple), un aspect simpliste à première vue, mais qui s'est avéré être une tâche plutôt ardue.

Ce jeu a été pour moi une véritable expérience, une épreuve durant laquelle j'ai appris énormément ; mais aussi un accomplissement, la réalisation de ce rêve enfantin. Je peux désormais affirmer avoir réel-

lement apprécié ce projet qui, sans l'aide de mes camarades, aurait peut-être pris des années à faire. En effet, c'est grâce à l'organisation de notre chef de projet et à l'aide de chacun des membres que nous en sommes arrivés là, et je ne regretterai jamais d'avoir participé à cette aventure, car je suis fier de pouvoir à présent vous faire découvrir notre jeu : *Party Blowout!*



Un grand merci à vous !

5.4 Lilas Reynaud

Dans sa généralité, je pense que le projet du second semestre a été une expérience enrichissante. Nous avons tous, à différentes échelles, appris de nouvelles choses.

Pour ma part, je trouve que la création de *Party Blowout!* et de son site internet m'a beaucoup offert. Dès la première période de travail, j'ai appris à faire des recherches de documentation de façon plus autonome. En effet, avant même de pouvoir commencer à coder le jeu, j'ai dû me renseigner sur le fonctionnement de Unity. Le menu, la partie dont je me suis majoritairement occupée, et que j'ai commencé dès le départ, m'a obligé à me familiariser avec la gestion de l'interface directement implémentée dans le logiciel. J'ai, pendant les quelques premiers mois, découvert cet aspect, en créant les bases de notre menu principal. Nous savions, en tant que groupe, que son état n'était pas final, mais cela m'a permis de comprendre le fonctionnement d'un nouvel outil, auquel je me suis habituée au fil des mois, pour mettre à jour ce menu principal tout au long du projet. J'ai aussi pu utiliser ces nouvelles compétences pour la création d'un menu de pause, mis en place plus tard dans le projet, une fois les minijeux bien avancés.

La seconde grande partie de mon travail sur ce projet a été la création complète du site internet. En effet, c'est un aspect du projet qui m'a beaucoup motivé, même avant le lancement. Ainsi, j'ai créé le début du site web à partir d'une multitude de tutoriels et de documentation en ligne. Travaillant seule sur cette partie du projet, j'ai dû faire l'effort de chercher par moi-même les solutions à tous les problèmes que j'ai pu rencontrer, notamment le fait que mon slideshow d'arrière plan n'était pas en arrière plan. Mais, après quelques jours de recherche et de tests plus ou moins fructueux, j'ai finalement trouvé une solution satisfaisante. Après avoir fait les fondations du site pour la première

soutenance, j'ai mis à jour le site petit à petit pendant les mois restants. C'est passé par des avancées esthétiques, mais aussi par l'adaptation des informations contenues dans le site lui-même, pour qu'il reflète notre progression générale.

Concernant le plan humain d'un projet de groupe, je trouve que **Bi-gHappyDove** m'a beaucoup apporté, et beaucoup fait grandir. Dès le commencement du projet, j'ai trouvé que le groupe était bien organisé par rapport à d'autres projets auxquels j'ai pu participer dans le passé. Nous avons choisi un chef de projet capable de nous donner une structure de travail claire dès le premier jour. Cependant, personnellement, j'ai vécu, entre les deux soutenances intermédiaires, une période compliquée. Je me rends compte avec le recul que cela a apporté une tension en plus à notre groupe. Malgré tout, je suis soulagée de voir que les quelques conflits qui en ont résulté ont été encadrés, et nous n'avons pas laissé cela affecter le bien général de notre projet.

Dans la globalité, je pense que nous pouvons être satisfaits de notre jeu. Nous avons pu créer un jeu fonctionnel, et esthétique, en l'espace de six mois. Pour moi, c'est un des projets de plus grande ampleur auxquels j'ai participé. Malgré les petits soucis affrontés, je suis heureuse d'y avoir pris part.

6 Conclusion

En l'espace de six mois, nous avons réussi à créer notre jeu vidéo *Party Blowout* de A à Z. Nous avons décidé en janvier de faire noter projet autour d'une série de trois minijeux. Nous avons, en effet, réalisé *La Course Fruitée*, un jeu de voiture dans l'ambiance d'un pique-nique estival. Le modélisation des véhicules représente une salade de fruits, et les autres éléments d'un repas sur une nappe posée dans l'herbe sont présents tout au long du circuit.

Notre deuxième minijeu est un FPS. Pour celui-ci nous avons décidé d'emporter le joueur en hiver, avec la *Bataille de boules de neige*. En effet, les munitions sont des boules de neige et la map garde cette ambiance de vacances hivernales.

Le dernier jeu est le *Guess Who*. Nous avons ici adapté un classique des journées festives enfantines : le cache-cache. En effet, certains joueurs seront cachés parmi des intelligences artificielles. Bien qu'ils soient visibles aux yeux du chasseur, ils pourront se fondre dans la masse, et tromper le joueur détectif pour rester inaperçus.

Notre ressenti général est satisfaisant, considérant tous les challenges que nous avons surmontés ensemble. Étant donné la durée relativement restreinte de ce projet, par rapport à son ampleur, nous aurions probablement voulu perfectionner certains aspects du jeu. Cependant, nous sommes tout de même arrivés à un beau résultat, ce qui est déjà une victoire en elle-même.

Nous espérons que ce rapport final du projet vous aura permis de bien comprendre les différents aspects de *Party Blowout !*, et les différentes tâches auxquelles nous avons fait face ces derniers temps. Nous

espérons également sincèrement que vous avez apprécié suivre le parcours de **BigHappyDove** durant tout ce semestre, et que notre soutenance finale saura vous convaincre de la valeur de notre jeu.

7 Annexes

7.1 Bibliographie

- Photon PUN 2 - InfoGamer
- Photon PUN 2 - RugBud Redfern
- Documentation Unity
- Brackeys
- StackOverflow *avec modération*
- Easy Tutorials - Tutoriels de design de site web
- Brackeys - Menus in Unity
- w3schools - HTML tutorials
- w3schools - CSS framework tutorials
- Polarith - Documentation
- Guide pour InnoSetup