

# Party Blowout!

– Rapport de la deuxième soutenance –



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

*26 Avril 2021*

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Plan de soutenance</b>	<b>4</b>
<b>3</b>	<b>Reprise du cahier des charges</b>	<b>5</b>
3.1	Multijoueur - Titouan . . . . .	5
3.2	Mécaniques . . . . .	5
3.2.1	Intelligence Artificielle - Titouan . . . . .	5
3.2.2	Déplacements - Clovis / Jacques . . . . .	7
3.2.3	Animations - Jacques . . . . .	8
3.2.4	Objets utilisables . . . . .	11
3.2.5	Mini-jeux . . . . .	11
3.3	Visuels . . . . .	11
3.3.1	Interface - Titouan / Clovis / Lilas . . . . .	11
3.3.2	Modélisation - Clovis . . . . .	13
3.3.3	Maps - Clovis / Jacques . . . . .	14
3.3.4	Menu - Lilas / Titouan . . . . .	15
3.4	Audio - Jacques . . . . .	16
3.5	Site-Web - Lilas . . . . .	16
3.6	Point sur l'organisation . . . . .	17
<b>4</b>	<b>Chronologie de l'avancée du projet</b>	<b>18</b>
4.1	Objectifs pour la dernière soutenance . . . . .	18
<b>5</b>	<b>Récits individuels</b>	<b>19</b>
5.1	Titouan GUIONNEAU - Chef de projet . . . . .	19
5.2	Clovis LECHIEN . . . . .	19
5.3	Jacques REMY . . . . .	20
5.4	Lilas REYNAUD . . . . .	20
<b>6</b>	<b>Conclusion</b>	<b>21</b>
<b>7</b>	<b>Bibliographie</b>	<b>22</b>

## 1 Introduction

Dans ce rapport de soutenance, nous allons vous exposer l'avancement de notre groupe **BigHappyDove** sur notre projet de jeu *Party Blowout!* depuis la dernière soutenance.

Les trois mini-jeux de notre projet, à savoir, un shooter, un jeu de course et un *Guess Who*, se rapprochent petit à petit de leur aspect final, suivant les codes esthétiques de *Fall Guys*. Il est vrai que ceux-ci ne sont pas encore finalisés, cependant, nous commençons à avoir une structure fonctionnelle.

Depuis février, nous avons beaucoup avancé. En effet, nous nous sommes grandement rapprochés de vrais jeux. Notre progression est à la fois fonctionnelle, et cette fois-ci, esthétique également. Nous avons attaqué des problèmes plus complexes que pour la première soutenance. Dans beaucoup de domaines, nous avons réparé, ou optimisé des fonctions existantes. Nous avons aussi commencé des parties du projet que nous avions jusque là mises de côté.

Dans notre rapport, nous vous exposerons donc les différents défis auxquels nous avons dû faire face, ainsi que notre avancée actuelle et le reste des tâches que nous devons effectuer avant le rendu final. Tout d'abord, nous allons reprendre notre progrès dans chaque partie du projet, à la fois sous forme de paragraphes explicatifs, mais aussi de tableaux d'avancées. Ensuite, nous verrons où nous nous situons par rapport aux objectifs que nous avons fixés en février. Nous ferons aussi des récits individuels de nos contributions personnelles, et des problèmes que nous avons pu rencontrer ces derniers mois. Finalement, nous concluons ce rapport en rappelant les nouveautés du jeu, puis nous ouvrirons sur notre optimisme concernant les derniers objectifs afin d'atteindre notre jeu dans son aspect terminal.

## 2 Plan de soutenance

### **Introduction**

- Précision du projet
- **Avancée des tâches** : Situation actuelle

### **Tâches individuelles et objectifs restants**

- **Titouan GUIONNEAU** - Multijoueur / Intelligence Artificielle
- **Clovis LECHIEN** - Véhicules / Jeu de Course
- **Jacques REMY** - Déplacements / Points d'apparition
- **Lilas REYNAUD** - Menu / Site Internet

### **Démo**

- Démonstration des avancées du jeu et du site.

### **Conclusion**

- Le ressenti général
- Récapitulatif

## 3 Reprise du cahier des charges

### 3.1 Multijoueur - Titouan

Quelques bugs ont été corrigés. Pour essayer de rendre le gameplay plus agréable pour les joueurs, nous avons dû faire quelques optimisations pour garder un équilibre parfait entre performance et fiabilité.

Le système de message RPC a été du vrai pain béni pour nous, cette fonctionnalité sert à envoyer des "messages". Lorsqu'un joueur reçoit un message, il doit exécuter une fonction liée à ce message (avec ces différents arguments). Cela nous a permis de résoudre beaucoup de problèmes dont notamment pour partager les attributs des joueurs sur le réseau.

### 3.2 Mécaniques

#### 3.2.1 Intelligence Artificielle - Titouan

Utilisée dans le mode de jeu *Guess Who*, l'intelligence artificielle avait déjà été implémentée lors de la première soutenance grâce aux merveilleux outils directement intégrés dans Unity. Cependant nous avons relevés plusieurs problèmes concernant les IA :

- 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.

Comment pallier tous ces problèmes ? Dans un premier temps nous avons pensé à utiliser *Polarith*, un plugin pour créer des IA plus "réalistes" qui peuvent par exemple se balader sur la map sans chercher à aller à un point en particulier. Cependant l'implémentation de ce plugin était un peu trop *overkill* pour ce qu'on voulait faire, car il contenait une cinquantaine de fonctionnalités... inutiles pour notre cas.

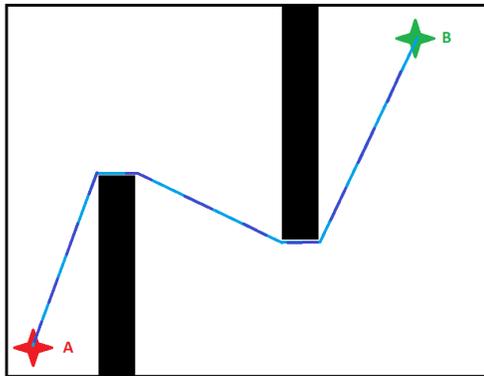
Nous avons donc décidé de repartir sur les outils intégrés de Unity et de les modeler pour arriver à un résultat satisfaisant. Voici les différents changements effectués :

- De manière aléatoire, les IA peuvent maintenant courir, marcher ou ne pas bouger, en prenant soin de faire appel aux bonnes animations et de synchroniser le tout pour tous les joueurs.
- Nous avons créé un système de checkpoints chaînés : Dans la map sont placés plusieurs checkpoints. Chaque checkpoint 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.
- 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. De ce fait, deux IA partant du même point pour aller au même checkpoint, n'auront jamais la même trajectoire.

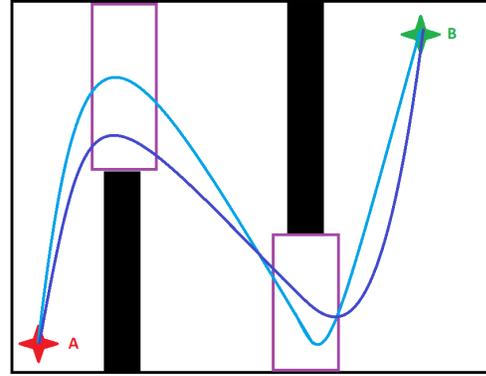
Dans le cas où deux IA doivent aller d'un point A à un point B, nous pouvons très facilement voir l'amélioration à travers ces deux schémas (en bleu clair et bleu foncé les trajectoires des deux IA)<sup>1</sup> :

---

1. A préciser que dans le jeu, il n'y a pas de point de départ ou d'arrivée fixes, les IA se déplacent de checkpoint en checkpoint.



1ère soutenance



2ème soutenance  
(où les checkpoints sont en violet)

### 3.2.2 Déplacements - Clovis / Jacques

Les déplacements dans le jeu de la course fruitée ont été implémentés. En effet, nous avons synchronisé les véhicules, pour que leurs déplacements soient visibles en multijoueur. De plus, le contrôle du véhicule pour le joueur a été revu entièrement. Maintenant, ce contrôle est dynamique. Nous avons retiré les WheelColliders que nous avions à la première soutenance, et remplacé ceux-ci par un système se basant sur le RigidBody. Pourquoi nous demanderez vous ?

Eh bien il s'est avéré que cet outil était bien trop poussé pour ce que nous voulions faire : les WheelColliders sont des objets qui nécessitent une structure et une implémentation extrêmement réaliste pour obtenir un résultat qui fonctionne correctement.

Par exemple, avec ces derniers, si nous ne voulions pas faire de tonneau sur le côté lors d'un virage serré, il fallait rajouter un objet servant de centre de gravité du véhicule, des ajustements millimétrés pour les suspensions et amortisseurs (quelques exemples : rigidité, flexibilité, "zone morte", c'est-à-dire à partir de quelle quantité de force appliquée la suspension commence à réagir, etc.), un bras de suspension qui vient empêcher les suspensions de s'écarter trop l'une d'elles ; bref nous pensons que vous comprenez toute la complexité de cette structure.

Notre objectif pour la maniabilité n'est pas d'avoir un logiciel réaliste comme "BeamNG.drive" mais plutôt un jeu avec des déplacements rapides qui laissent la place au joueur de s'amuser et de commettre des erreurs s'il aurait l'imprudence de vouloir aller trop vite.

C'est pourquoi l'alternative avec un RigidBody s'est avérée être très intéressante : il donne aux déplacements un aspect *Arcade Game* qui n'est pas sans rappeler ceux que l'ont pouvait trouver dans les vieux jeux vidéos comme *Pole Position* ou même des titres plus récents comme l'excellent *Reckless Racing* sur mobile. Pour arriver à un tel résultat, il a 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

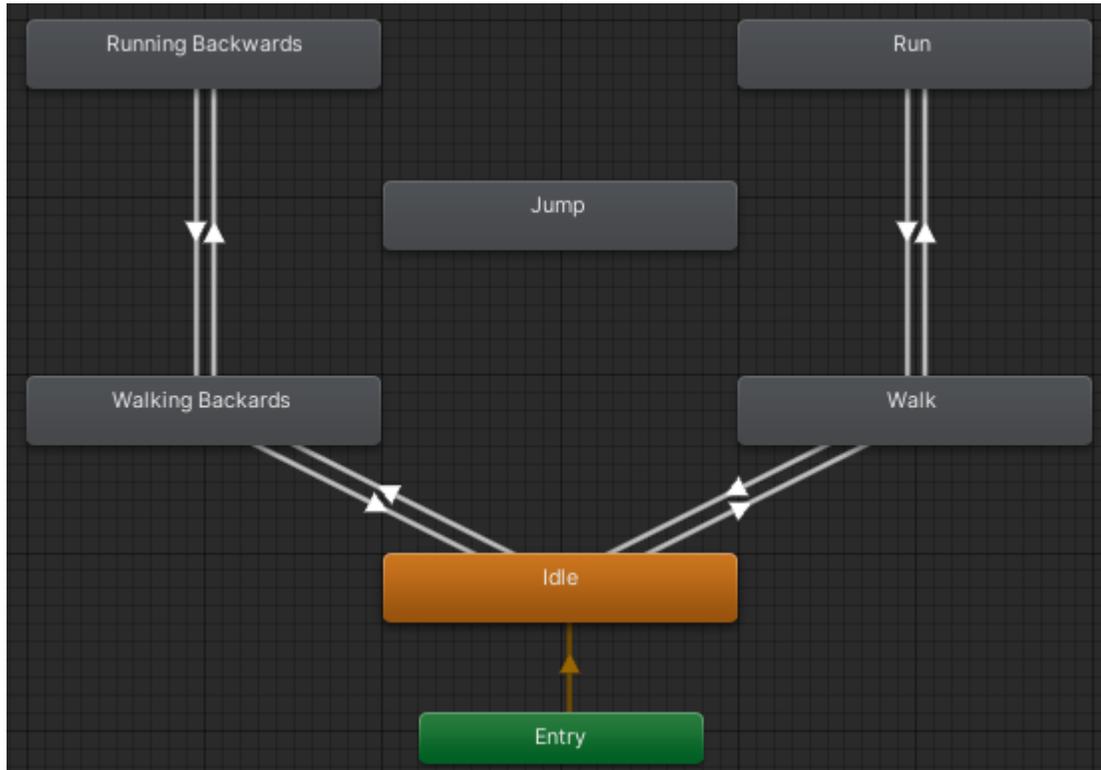
L'animation du personnage doit être divisée en 2 parties :

- 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.

On peut donc utiliser une animation sur n'importe quel modèle ayant les mêmes articulations. C'est pourquoi nous avons utilisé un modèle humanoïde. Nous nous concentrerons cependant ici à 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.

Observons plus précisément comment marche notre composant **animator** :



*Animator component*

Nous pouvons observer ci-dessus plusieurs rectangles désignant chaque "état d'animation" (en anglais : *animation state*). Chacun est relié 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.

Celui-ci 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 rajoute 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, caractérisé 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. Pour cela, plusieurs choix s'offrent à nous :

- le mode "Disabled" qui ne synchronise pas le paramètre (pour des animations qui n'ont pas besoin d'être synchronisées, dans notre cas nous n'en avons pas besoin)
- le mode "Discrete" qui synchronise seulement lorsque l'attribut sera modifié (c'est-à-dire, si le paramètre isWalking passe à True par exemple)
- et enfin, le mode "Continuous" qui synchronise en permanence les paramètres, ce qui n'est pas non plus nécessaire pour les mouvements des joueurs.

Vous l'aurez donc compris, notre choix s'est porté uniquement sur le mode "Discrete" qui est le plus approprié pour le moment.

### 3.2.4 Objets utilisables

Les armes ont été retravaillées pour pouvoir fonctionner en multijoueur, il en est de même pour le système d'inventaire, qui avait du mal à marcher en réseau. 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.

### 3.2.5 Mini-jeux

Regardons les mini-jeux un par un :

- **La Course Fruitée** : Les voitures sont prêtes, ainsi que le système de checkpoints pour les circuits. Il reste quelques détails à régler, mais le mode de jeu est largement jouable.
- **Le Shooter** : Ce mode de jeu est complètement jouable, même s'il manque quelques éléments dont un système de points, pour finaliser le mini-jeu.
- **Le Guess Who** : Ayant un système d'IA au point, nous pouvons dire sans problème que ce mode de jeu est quasiment prêt. Sa finalisation n'est qu'une question de petits détails à régler, surtout au niveau de l'interface.

Les mini-jeux, dans leur ensemble, sont presque prêts. La plus grosse difficulté sera de créer un système pour que les joueurs puissent passer d'un mode de jeu à un autre sans qu'il n'y ait de conflit.

## 3.3 Visuels

### 3.3.1 Interface - Titouan / Clovis / Lilas

Pour le *Guess Who* et le *Shooter*, nous avons commencé à créer une interface, qui reste pour l'instant très simple. Cependant elle permet au joueur d'avoir en main les informations essentielles :

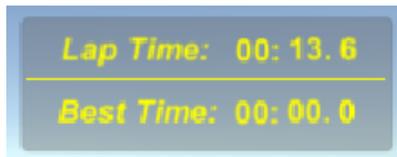
L'interface du joueur est à présent fonctionnelle dans le jeu de course. En effet, l'UI comprend le meilleur temps de course, le nombre de tours courant, ainsi que le chronomètre, propre à chaque joueur. Ceci permet un jeu plus clair et agréable pour les joueurs.



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



L'interface pour les munitions



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.

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 fonctions<sup>2</sup> 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.

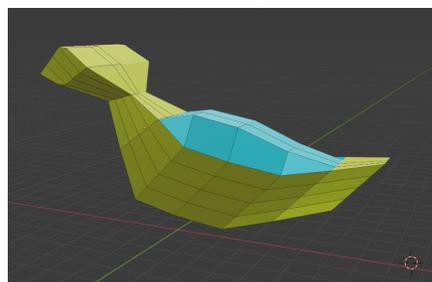
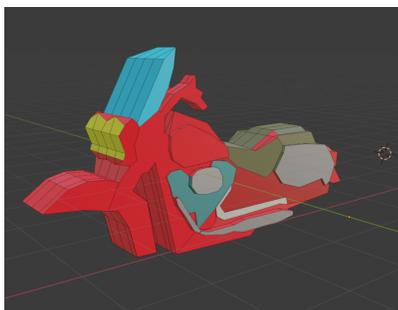
---

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

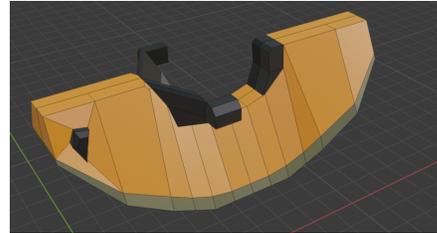
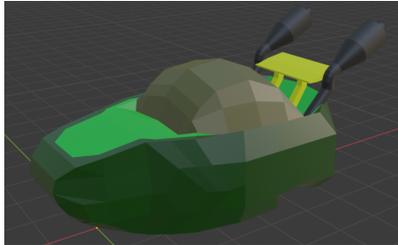
### 3.3.2 Modélisation - Clovis

Concernant la partie modélisation du projet, nous nous sommes plus concentrés sur cet aspect du projet durant ces dernières semaines. En effet, ce n'était pas un de nos objectifs principaux de la première soutenance, mais à présent, vu les progrès nécessaires en vue de la seconde soutenance, nous avons renforcé nos efforts.

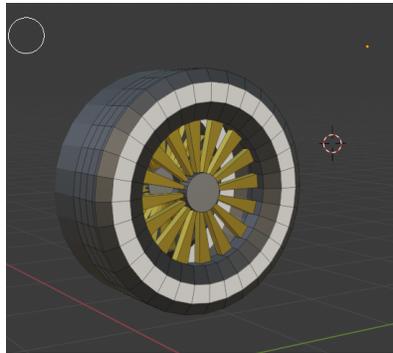
Nous avons grandement avancé le circuit et la map pour le jeu de Course Fruitée. Nous avons également décoré celui-ci avec de nombreuses modélisations faites maison : des bouteilles, des couverts, ou autre. Il ne faut pas oublier que notre jeu de course a pour thème la nourriture. Nous avons décidé de commencer à implémenter le côté esthétique du projet suite aux remarques de M.Blet. Cet aspect avait été jusque là mis en arrière plan. Pour ce même mini-jeu, nous avons 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. Ces modèles sont susceptibles de changer, cependant l'aspect général et l'idée de conception de chaque véhicule restera intacte. Nous sommes conscients qu'il nous reste des progrès à faire dans la modélisation pour notre projet mais tout cela arrivera à temps lorsque les parties purement techniques et le gros de la programmation sera finie. C'est à ce moment que pourra s'exprimer notre "fibre artistique".



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.



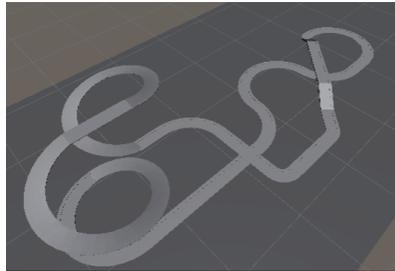
La roue utilisée pour nos véhicules

### 3.3.3 Maps - Clovis / Jacques

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 place de chaque côté de la route.

Quant au décor présent autour du circuit, il nous reste encore beaucoup de travail de modélisation pour l'améliorer, mais c'était notre intention car, encore une fois, l'esthétique passe après le fonctionnel. A la dernière soutenance tout ceci sera fini et le résultat n'en sera que plus beau.

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.



Voici le tracé de notre circuit pur sans décoration, il présente deux boucles s'élevant dans les airs ainsi qu'un pont passant au dessus d'une portion de la course.

### 3.3.4 Menu - Lilas / Titouan

Pour le menu d'accueil du jeu, nous avons beaucoup de changements esthétiques à réaliser. 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.

Pour cette seconde soutenance, nous avons donc changé les codes esthétiques. Nous avons placé les boutons sur la gauche, et laissé le côté droit plutôt transparent afin de voir notre joueur dans une scène en arrière-plan. Nous avons donc créé une image de fond adaptée à ce but, dans les couleurs que nous souhaitons garder pour l'ensemble de notre jeu, à savoir les roses et les bleus.

En petit détail fonctionnel, nous avons lié le bouton du volume à la musique d'accueil que nous avons déjà créée à la première soutenance, mais pas encore implémentée à l'époque.

### 3.4 Audio - Jacques

Durant cette période séparant la première et la seconde soutenance, nous nous sommes attelés à créer et implémenter les musiques que nous avons créés, vous pourrez par exemple entendre la musique *officielle* de *Party Blowout!* dans le menu principal.

Il reste toutefois une marge de progression conséquente dans ce domaine de notre projet. En effet, il nous reste à mettre en place la plupart du sound design de nos mini-jeux. Nos objectifs pour la dernière soutenance sont donc d'avoir fini entièrement la partie son de notre jeu, étant donné que nous disposons déjà d'une base solide de divers sons que nous avons préparés cela ne nous posera donc aucun problème.

### 3.5 Site-Web - Lilas

Par rapport à la première soutenance, qui avait permis de créer les bases HTML du site internet, il était temps de s'occuper du côté esthétique de celui-ci. 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 implémenté des images en arrière-plan, pour les autres pages du site, afin qu'il colle à l'aspect esthétique visé pour notre projet. Ceci était plutôt un travail de design, grâce à une tablette graphique. Ainsi, nous avons un site web au visuel plus professionnel, plus proche de ce que nous aurions pu faire avec des templates.

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.

### 3.6 Point sur l'organisation

Si l'on compare notre organisation depuis la première soutenance, elle n'a pas réellement changé. Nous utilisons toujours les branches *Git* pour s'organiser un maximum tout en évitant les conflits, cela permet d'avoir un projet extrêmement stable. Chaque branche (hormis *Master* et *Pre-Master*), correspond à une fonctionnalité différente. Cela permet à tous de savoir qui travaille sur quoi et quand. De plus, grâce aux Webhooks de *Git* que nous avons intégrés sur *Discord*, tous les membres du groupe reçoivent une notification lorsque quelqu'un effectue un push sur le projet, peu importe la branche.

Les réunions hebdomadaires du jeudi soir sont toujours d'actualité, cela nous permet de débattre sur certains points concernant le projet, mais aussi à vérifier l'avancée de tous. C'est aussi un moment pour parler des différents problèmes rencontrés et trouver ensemble des solutions.

Plus nous avançons dans le projet, plus nous sommes contraints à travailler sur les codes publiés par les autres membres du groupe. C'est pourquoi nous essayons d'organiser au maximum la structure du projet, mais aussi d'écrire des commentaires ou de la documentation dans les fonctions. Le fait que tout le monde touche un peu au code des autres ne semble pas avoir freiné le projet, mais donne au contraire une certaine dynamique car tout le monde peut réparer ou bien améliorer un quelconque code.

## 4 Chronologie de l'avancée du projet

### 4.1 Objectifs pour la dernière soutenance

Soutenances	Prévision	Réalité	Dernière
<b>Réseau</b>			
Multijoueur*	100%	100%	100%
<b>Mécaniques</b>			
I.A	80%	90%	100%
Déplacements	70%	80%	100%
Objets utilisables	60%	75%	100%
Mini-jeux	66%	50%	100%
<b>Visuel</b>			
Interface	60%	60%	100%
Modélisation	66%	70%	100%
Maps	66%	80%	100%
Menu	75%	40%	100%
<b>Audio</b>	75%	75%	100%
<b>Site-Web</b>	90%	80 %	100%

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

## 5 Récits individuels

### 5.1 Titouan GUIONNEAU - Chef de projet

Le bilan pour cette deuxième soutenance est plutôt bon, même si nous avons rencontré différents problèmes (informatiques comme humains), nous avons réussi à y faire face d'une manière ou d'une autre. Dans ce groupe il y a une réelle cohésion d'équipe, on n'hésite pas à regarder le code des autres si besoin pour donner notre point de vue ou bien même des solutions.

Je suis très optimiste pour la fin de ce projet, même si le jeu est objectivement moche, nous avons réussi à mettre en place toutes les mécaniques pour les trois mini-jeux qui sont quasiment prêts. Dès à présent il est seulement question de polir le jeu pour qu'il soit entièrement jouable et pour qu'il ait un aspect visuel plus convaincant.

On est toujours aussi déterminé pour terminer ce jeu !

### 5.2 Clovis LECHIEN

Cette deuxième période de travail sur le projet a été parsemée de différents événements qui auraient pu mettre à mal l'ambiance du groupe et donc compromettre l'avancée de ce projet, cependant chaque membre du groupe a su faire la part des choses et mettre à profit ses forces et ses faiblesses pour ne pas interrompre le travail régulier sur le jeu. Le résultat est au rendez-vous comme vous pouvez vous en rendre compte avec les progrès titanesques ayant été accomplis : à la première soutenance le jeu présentait des choses intéressantes mais qui n'étaient pas vraiment exploitées et à cette soutenance nous vous présentons un jeu composé de trois mini-jeux dont la plupart sont pratiquement terminés.

La conclusion de ce projet S2 se profile à l'horizon et je suis heureux d'avoir pu rejoindre un groupe où chacun a su apporter sa part de travail. Il ne nous reste maintenant pratiquement plus que la partie esthétique du projet, ça sera donc du gâteau pour moi comme vous avez pu le remarquer avec les modèles de véhicules (je suis particulièrement fier de la voiture avocat).

### 5.3 Jacques REMY

Au cours de cette deuxième partie, et malgré un départ difficile, nous avons réussi à surmonter maintes épreuves. Devoir gérer les cours, la vie privée et le projet, ne fut pas une simple tâche, mais heureusement nous nous sommes accroché et je suis très fier de notre groupe !

Pour ma part, je me suis principalement focalisé sur le travail d'animation des personnages. Cela m'a pris plus de temps que prévu à cause de plusieurs problèmes qui n'étaient pas forcément sur l'animation, mais qui m'ont quand même fait modifier, recommencer, re-recommencer, re-re-recommencer, re-re-re.. enfin bref, réécrire le programme de nombreuses fois. Mais, bien que la résolution de certains problèmes ne m'ait rien apporté à part du temps perdu, le reste a en partie contribué à une certaine optimisation de l'animation, des déplacements ou bien des hitbox, dont je suis amplement satisfait.

Je dirais, pour conclure, qu'une chose est sûre : je sais que nous sommes largement capables de finaliser et d'embellir notre jeu d'ici la dernière soutenance, j'ai hâte de voir le rendu, et même de pouvoir l'essayer avec plein de personnes !

### 5.4 Lilas REYNAUD

Depuis la première soutenance, le groupe a dû traverser des épreuves plus complexes que lors du début du projet. En effet, avec l'accélération du rythme de travail du second semestre, et d'autres problèmes d'ordre humain, le projet aurait pu en pâtir. Cependant, notre groupe a su rester soudé, et se soutenir à travers l'adversité que chaque membre a pu rencontrer.

Personnellement, je n'ai pas pu travailler sur le projet pendant quelques semaines, qui me paraissaient difficilement surmontables. Malgré cela, le reste du groupe, de mes amis, m'a soutenu. Nous avons, avec le chef de projet, trouvé des compromis pour que je ne sois pas totalement mise à l'écart, sans mettre en danger la progression du jeu. Finalement, cette organisation nous a réussi, et, malgré les difficultés, nous avons obtenu un jeu à un stade satisfaisant pour une deuxième soutenance.

Cette ambiance de travail et d'entraide me donne espoir pour la continuation du projet. Je nous sens capable d'avoir un jeu de qualité pour le rendu final.

## 6 Conclusion

Si l'on observe notre avancée depuis la dernière soutenance, nous pouvons soutenir les efforts réalisés dans le but de produire ce jeu. Même si nous avons pris du retard sur un nombre restreint de nos objectifs, nous restons tous très confiants pour la fin du projet. En effet, nous savons exactement ce qu'il faut terminer en vue de la troisième, et dernière soutenance.

Le ressenti général par rapport à l'état de ce projet est très bon, il y a une très bonne ambiance au sein du groupe et nous sommes toujours aussi motivés pour la fin du projet. En effet, considérant les challenges que nous avons surmonté ensemble, nous avons confiance en nos collègues, et notre capacité de mener à bien le projet.

Nous espérons que ce rapport de soutenance aura su vous éclairer quant à nos différentes avancées et aux rôles de chacun dans ce projet. Il ne nous reste désormais plus qu'à vous remercier !

## 7 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