

# Rapport projet : You are bad at racing

## Description du jeu

Le jeu *You are bad at racing* est un jeu 3D dans lequel le joueur incarne une voiture et la dirige tout au long d'une route tortueuse. Il n'y a qu'un seul chemin qui est le bon, mais il faut le trouver, il est donc normal de se tromper de route. À plusieurs moments, la route est suspendue dans le vide, ce qui fait que le joueur peut tomber et se retrouve à recommencer le parcours soit au début de la partie, soit au dernier checkpoint, ce qui signifie que le jeu peut être considéré comme étant un *die and retry*. Le joueur doit donc avancer petit à petit dans le parcours pour trouver le point d'arrivée tout en faisant attention au vide ainsi qu'aux différents éléments sur sa route. Ce concept de notre jeu est fortement inspiré de *You Suck At Parking*, de part le style graphique, la gestion de la physique, ou encore le positionnement de la caméra.

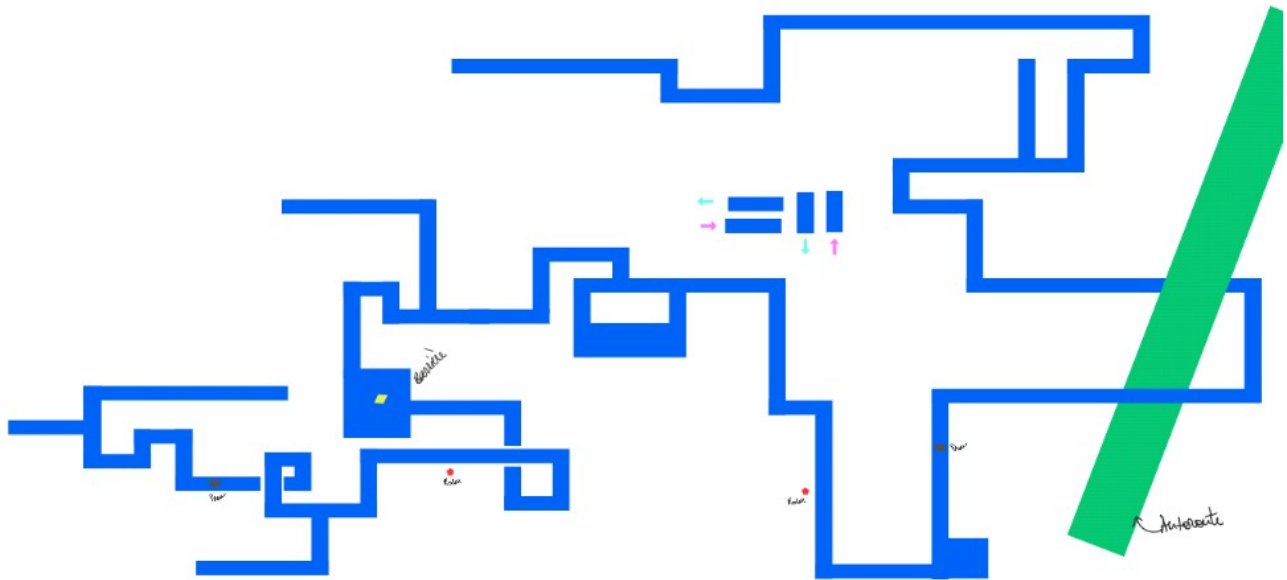
## Interface utilisateur / Ressources (assets) / Level Design

En terme de ressources nous avons demandé à des amis de produire tous les éléments 3D du jeu. De ce fait, la voiture, la map sont réalisés par Augustin, avec l'aide de Mathis pour les barrières.

Lorsque le jeu se lance, le menu principal apparaît, il est ensuite possible de choisir entre : commencer une partie ou quitter le jeu.

Notre HUD est volontairement assez simple : il y a le niveau d'essence du joueur qui est affiché, le timer de la partie. Ce timer se réinitialise lorsque l'on recommence la course mais continue de tourner lorsque l'on revient à un checkpoint.

Le level design a été réalisé par Rose avec la première image suivante :



## Répartition du travail prévue

Tâches de <b>Ianis</b>	Valeur de temps arbitraire
Créer des IA	10
Contrôles de la voiture	20
Les obstacles	8

La physique a demandé beaucoup de modifications, j'ai commencé par utiliser des mouvements fait avec la translation et la rotation de Unity, mais il n'y avait pas d'inertie, et donc pas d'effet de dérapage lorsque le joueur fait de gros virage. De ce fait, j'ai cherché une autre méthode pour le faire et c'est comme ça que j'ai commencé à utiliser les *wheel collider* d'Unity. Néanmoins, le résultat n'était vraiment pas probant : les problèmes étant que premièrement il est très compliqué de paramétrer correctement les valeurs pour avoir un résultat agréable, mais en plus nous souhaitions avoir un ressenti arcade, tandis que le *wheel collider* est plus destiné à donner un ressenti réaliste. Ne parvenant pas à avoir quelque chose de satisfaisant au niveau des déplacements, j'ai finalement opté pour une nouvelle stratégie avec l'utilisation d'une sphère sur laquelle s'applique les forces. Le rendu était beaucoup mieux, mais la voiture « glissait » trop. Pour corriger ça, j'ai fini par projeter le vecteur vitesse de la sphère sur le vecteur *forward* du mesh de la voiture, ce qui a donné un meilleur ressenti.

Tâches de <b>Rose</b>	Valeur de temps arbitraire
Tester plusieurs méthodes de création de route	18
Interfaces : Menu principal + HUD (essence, timer)	15
Imaginer et organiser un niveau	2
Effets (shader	3

Pour la réalisation de la route, j'ai réalisé plusieurs essais pour la produire, avec notamment le plugin RoadArchitect, EasyRoad 3D ou d'autres, mais les rendus n'étaient vraiment pas convaincants et particulièrement du fait des croisements des routes, nous avons donc demandé à des proches spécialisés dans la production d'éléments 3D de nous produire la carte correspondante à l'image présentée plus haut.

Tâches en <b>commun</b>	Valeur de temps arbitraire
Créer les actions des objets ( <i>props</i> )	5
Ajouter les objets à la map pour instancier le niveau et appliquer leur effets à la voiture	8

Pour finaliser le projet, il était nécessaire de fusionner notre travail et produire les éléments restants ensemble, nous avons donc peaufiné ensemble les interactions avec les objets, leur positionnement et les différents états du joueur (explosion, arrivée, mort, respawn au dernier checkpoint, ...). De ce fait, nous avons à gérer énormément de prefab tout au long de notre projet.

## Défis - au-delà du fait en cours

Les plus gros défis était la physique et notamment les valeurs qui y sont associés, nous avons passé énormément de temps sur le choix des valeurs et la manière de la gérer afin d'avoir le meilleur ressenti possible en tant que joueur.

La réalisation de *shaders* était aussi une nouvelle expérience, il était important d'utiliser le mode de rendu URP proposé par Unity pour avoir de bons rendu, mais cela à été compliqué à initialiser et nous nous sommes retrouvé plusieurs fois coincés car nous ne connaissions pas bien ce mode de rendu et ce qu'il impliquait.

L'IA a aussi été un élément complexe à mettre en place : il ne fallait pas qu'elle suive trop parfaitement les mouvements du joueur car cela ne rend pas du tout réaliste. Il était donc nécessaire de mettre un peu de latence avec les réactions de l'IA. De plus si l'IA tourne en même temps que le joueur mais qu'il y a le virage entre les deux, l'IA va se jeter directement dans le vide, ce qui n'est pas réaliste non plus. Pour donner une impression d'anticipation du virage car la voiture doit commencer à tourner un petit peu avant le virage, nous avons tiré des rayons vers le bas à intervalle fixe sur une ligne tirée entre l'IA et le joueur. Si le pourcentage des rayons dirigés vers le bas collisionnent vers le sol dépasse 60 %, l'IA peut tourner.