

# Rapport PCII

Martin Berthier & Quentin Bertrand

2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Analyse Globale</b>	<b>3</b>
<b>3</b>	<b>Plan de développement</b>	<b>4</b>
<b>4</b>	<b>Conception détaillée</b>	<b>6</b>
<b>5</b>	<b>Résultat</b>	<b>10</b>
<b>6</b>	<b>Documentation utilisateur</b>	<b>12</b>
<b>7</b>	<b>Documentation développeur</b>	<b>13</b>
<b>8</b>	<b>Conclusion et perspectives</b>	<b>14</b>

# Chapitre 1

## Introduction

Le but de ce projet est de réaliser un jeu de course de moto en pseudo 3D. Le but du jeu étant de jouer une moto en vue 3ème personne et de devoir se mouvoir avec les touches directionnelles (gauche et droite) dans un environnement. On devra passer des portes avant un certain laps de temps si on ne veut pas perdre. Il y aura aussi d'autres motos dans cette course qui seront vos ennemies et que vous devrez dépasser. La vitesse sera un facteur à prendre en compte, en effet plus vous roulez plus vous gagnerez en vitesse mais *a contrario* si vous roulez autre part que sur la route ou si vous heurtez un ennemi et/ou obstacle vous perdrez de la vitesse. La difficulté de ce jeu étant de réussir à appréhender la vitesse tout en esquivant les autres joueurs et les différents obstacles tout en passant les portes dans le temps imparti. Le score du joueur se base sur la distance parcourue.

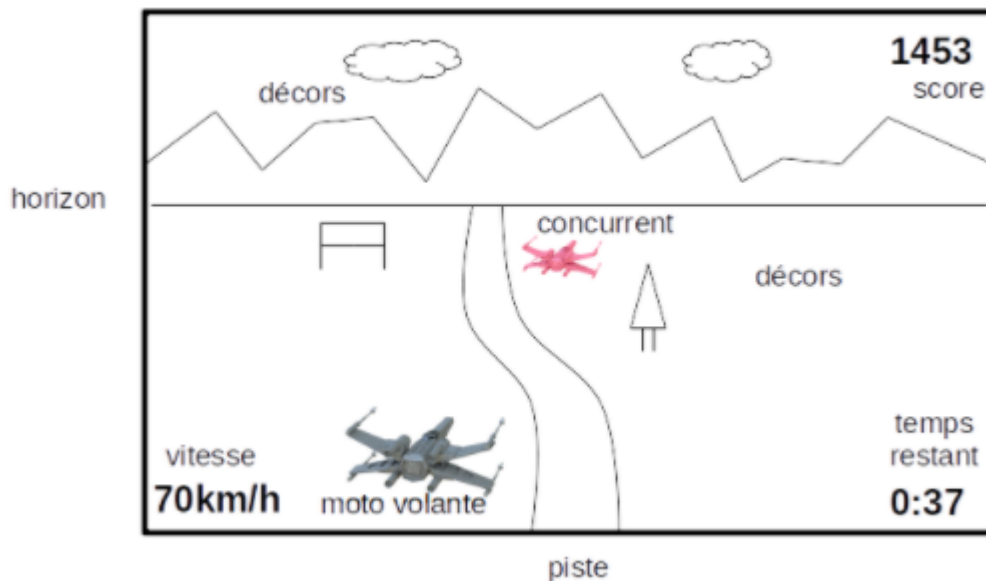


FIGURE 1.1 – Aperçu de base que le jeu devrait avoir

## Chapitre 2

# Analyse Globale

En cette première séance, les principales fonctionnalités à développer sont l’affichage de la fenêtre (facile), l’affichage d’un horizon ainsi que d’une route représentée par des lignes brisée (facile), la représentation de la moto par un rectangle et implémenter son déplacement (difficulté moyenne), implémentation de la vitesse de la moto (accélération et freinage) et faire varier la vitesse de la route en fonction de cette vitesse (difficulté élevée). Et tout cela en respectant le *model MVC*. Lors de la deuxième séance, nous avons implémenté de nombreuses choses. Comme par exemple :

- L’affichage d’un *background* qui défile de gauche à droite de façon infinie
- Un affichage de la route plus optimisé
- Un ajout de plusieurs fonctionnalités au niveau de l’affichage comme :
  - La distance parcourue
  - La vitesse
  - *Timer*
  - Un message lorsque le temps est écoulé

La difficulté de chacune de ces fonctionnalités varie beaucoup, les tâches qui ont été les plus difficiles sont l’affichage de la route qui a été *refactor* un bon nombre de fois et l’affichage infini du *background*. Il n’y a qu’une seule tâche de difficulté intermédiaire : l’affichage de la vitesse avec un sprite de compteur de vitesse, les tâches restantes ont toutes été plus ou moins facile à réaliser. Dans le projet dans son ensemble, chacune de ces fonctionnalités sont importantes car elles constituent la base de notre jeu, ce sont les mécaniques de bases du jeu.

## Chapitre 3

# Plan de développement

Pour chaque fonctionnalité de cette première séance, nous avons estimé un temps de travail moyen :

- Affichage de la fenêtre, de la route et de la moto, environ 15 minutes
- Implementer le mouvement de la moto de façon avancée, environ 30 minutes
- Implementer la route de façon plus avancée, environ 30 minutes

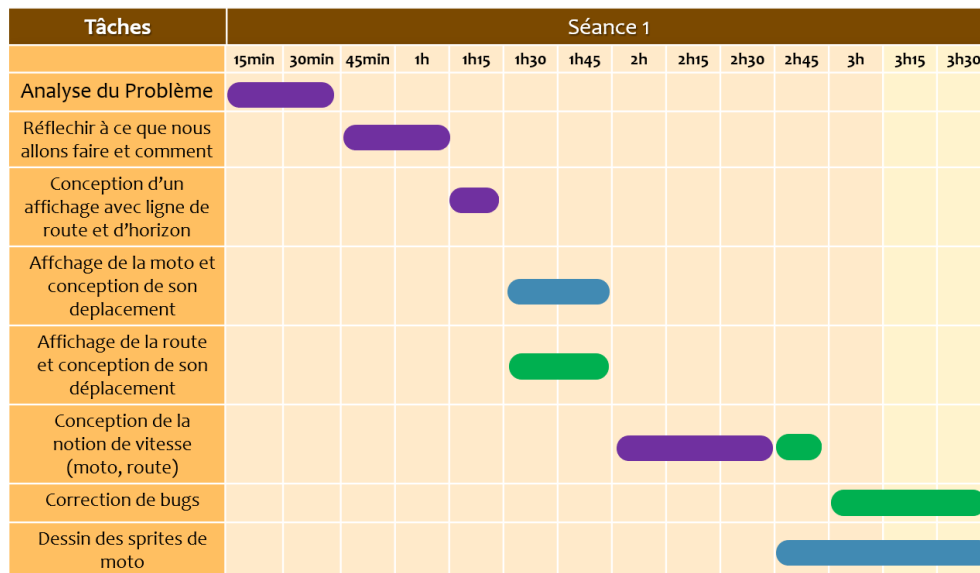


FIGURE 3.1 – Diagramme de Gantt de la Séance 1

Sur ce diagramme on peut y apercevoir 3 couleurs : le violet signifie que nous avons travaillé ensemble sur cette tâche, le vert correspond aux tâches réalisées par Martin et celle en bleu par Quentin.

Pour cette deuxième séance, nous estimons le temps moyen de travail par tâches à :

- *Background* environ, 2 heures
- Route environ, 2 heures
- Affichage :
  - Vitesse environ, 45 minutes
  - Fin de jeu environ, 30 minutes
  - Temps environ, 30 minutes
  - Distance parcourue environ, 30 minutes

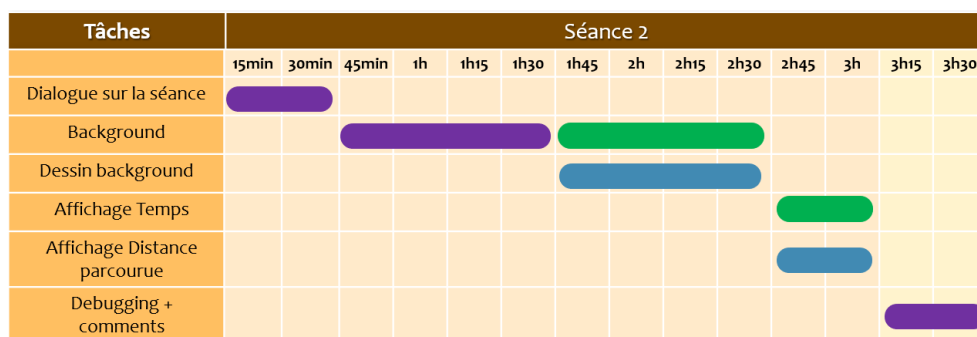


FIGURE 3.2 – Diagramme de Gantt de la Séance 2

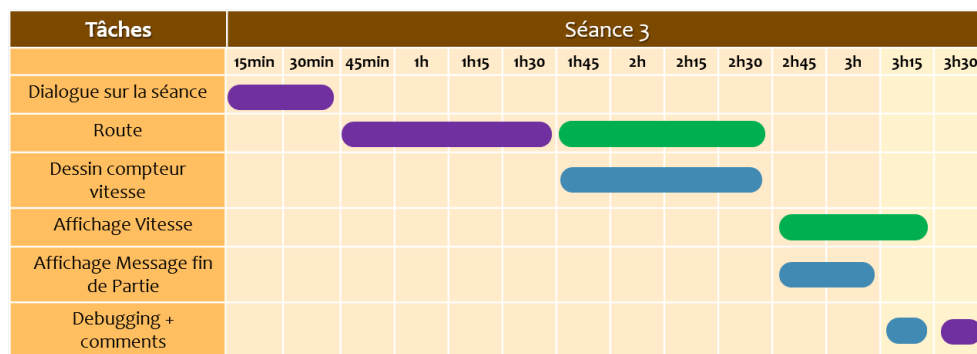


FIGURE 3.3 – Diagramme de Gantt de la Séance 3

Le code des couleurs est exactement le même que pour la séance 1.

## Chapitre 4

# Conception détaillée

Pour la conception, nous avons tout d'abord réparti notre projet en 3 *packages* différents : *Model*, *View* et *Controler* afin d'affecter les classes correspondant à chaque partie du *model* dans le bon *package*. Dans la *View*, nous avons mis tout ce qui gèrait l'affichage de notre jeu, dans le *model*, tout ce qui faisait varier notre jeu et enfin dans *controler*, tout ce qui concerne le *keyManager* et le *keyListener*. Nous avons donc réalisé chaque fonctionnalité à des endroits bien précis :

- Affichage de la fenêtre, de la route et de la moto : les 3 se font dans les classes suivantes *TH\_Game* pour la fenêtre et dans la classe *Gfx* pour la route et la moto mais la route ainsi que la moto sont 'initialisés' respectivement dans les classes *Road* et *Moto*. Il y a plusieurs constantes en ce qui concerne la moto :

```
Const : The player's horizontal speed
public final static int MOVE_SPEED = 3;
Const : the player's width
public final static int WIDTH = 50;
Const : the player's height
public final static int HEIGHT = 50;
Const : the player's initial position
public final static int Y = Gfx.HEIGHT - HEIGHT*2;
Const : the moto's maximum speed
public static final int MAX_SPEED = 350;
```

FIGURE 4.1 – Constante nécessaire pour la moto

- Le mouvement de la moto se fait en plusieurs temps, tout d'abord on récupère la touche qui a été actionnée et on interprète ensuite cette action avec le *keyManager* afin de pouvoir modifier la position end de la moto en fonction de la touche et de l'afficher dans la classe *Gfx*.
- Pour ce qui concerne la route, son affichage est fait grâce aux classe *Road* et *Gfx* mais son mouvement en fonction de la vitesse de la moto est due à la classe *TH\_Scrolling* qui est un *Thread* qui va plus ou moins vite en fonction de la contante *MAX\_SPEED* de la classe *Moto*.
- Le *keyListener* se met à jour grâce à la classe *TH\_Turn* qui est un *Thread* qui le met à jour en permanence et qui dépend également de la constante *MAX\_SPEED*.





Pour cette deuxième séance nous avons changer beaucoup de choses par rapport à la première.

- En ce qui concerne la Moto, elle ne se déplace plus, elle reste au centre de l'écran et c'est tout ce qui s'y trouve qui se déplace de  $+$  ou  $-$  l'*offset* de la moto.
- Pour la route, nous sommes passés par 2 phases :
  - Remplacer les *drawLine* par des segments (méthode que nous n'avons pas gardé)
  - Rajouter plus de perspective en utilisant des *curbs* (méthode que nous utilisons actuellement)
- L'affichage d'un *background* qui bouge en fonction de la moto de façon infinie en mettant les *backgrounds* les un apres les autres afin de donner l'illusion d'une image infinie.
- Pour ce qui concerne les affichages divers, le temps et la distance parcourue sont des *print* basiques. Pour ce qui concerne la vitesse nous récupérons la vitesse que nous utilisons pour faire varier un *theta* pour faire tourner notre *sprite* d'aiguille.

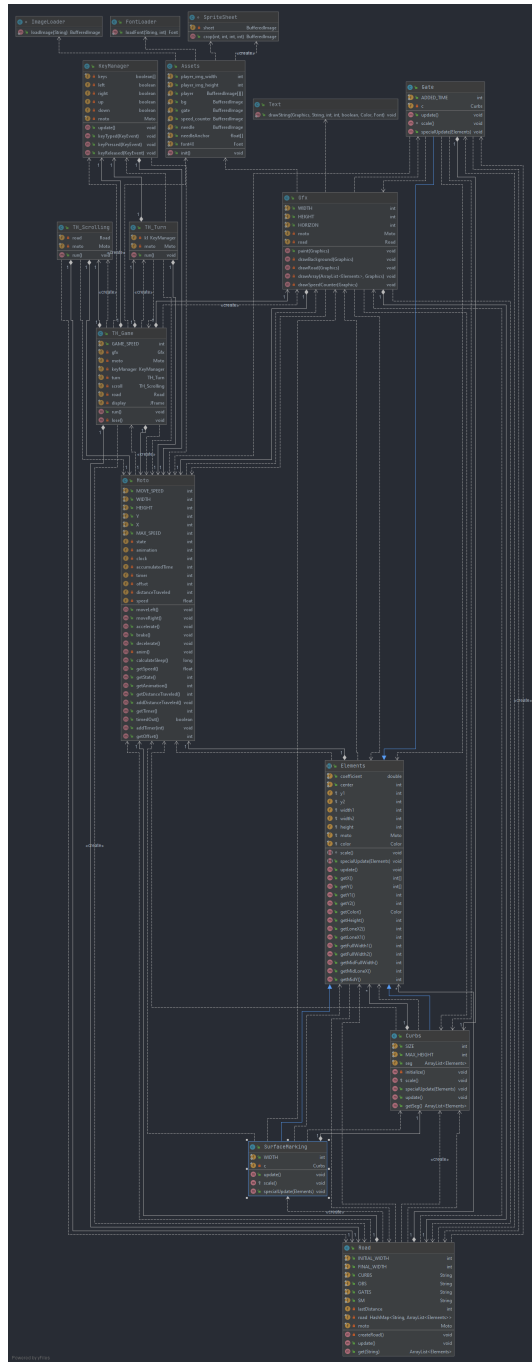


FIGURE 4.3 – Diagramme de classe de notre projet à la séance 3

# Chapitre 5

## Résultat

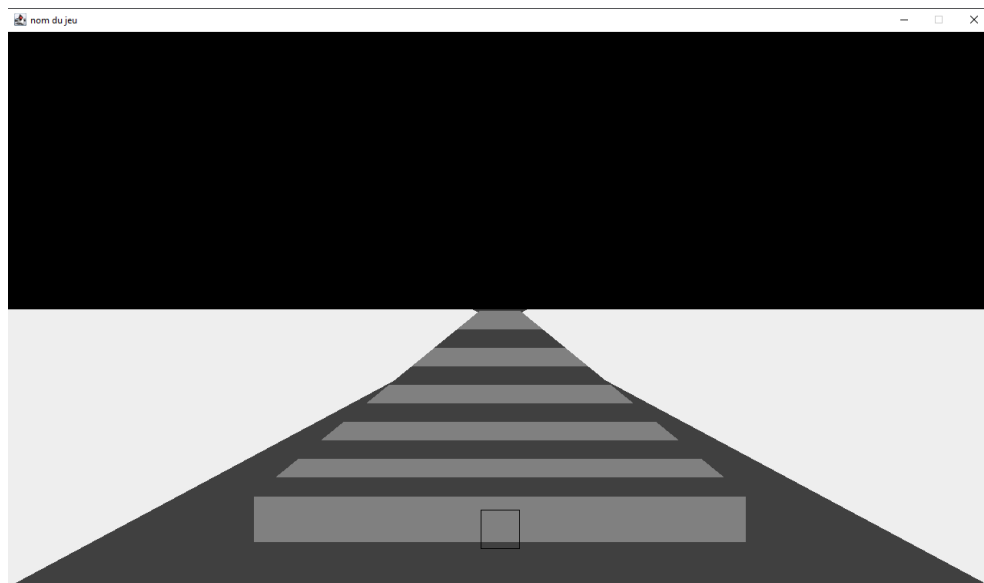


FIGURE 5.1 – Aperçu du jeu à la séance 1

Vous pouvez voir ci-dessus l’affichage de notre jeu à la fin de la première séance de notre projet.

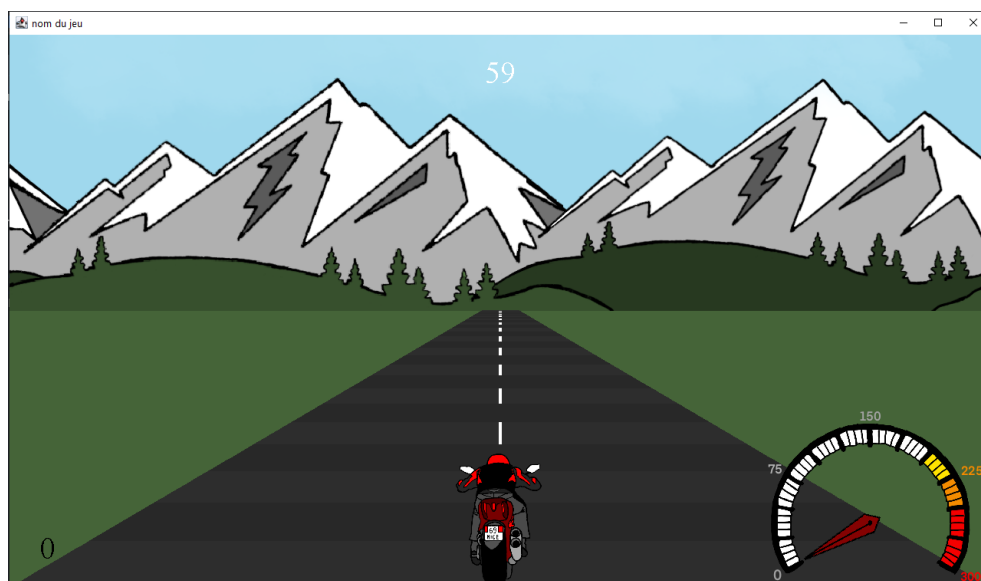


FIGURE 5.2 – Aperçu du jeu à la séance 3

Vous pouvez voir ci-dessus l’affichage de notre jeu à la fin de la troisième séance de notre projet. En haut au milieu on peut y voir le temps restant, en bas à gauche la distance parcourue et enfin en bas à droite le compteur de vitesse.

## Chapitre 6

# Documentation utilisateur

Pour pouvoir jouer au jeu il y a quelques petites choses à savoir. Tout d'abord, un *IDE Java* est nécessaire (ou *Java* seul si jamais vous avez fait un *export .jar* en exécutable). Si vous êtes dans le premier cas et que vous avez un *IDE Java*, il vous faut importer le projet dans votre *IDE*, sélectionner la classe *Main* et enfin lancer l'application en cliquant sur *Run as Java Application*. Vous pouvez utiliser les flèches directionnelles afin de faire bouger le rectangle et de le faire accélérer et freiner. Si vous êtes dans le second cas, que vous possédez un exécutable *.jar*, il vous suffira de double cliquer sur le fichier exécutable et d'utiliser les flèches directionnelles pour faire bouger le rectangle et de le faire accélérer et freiner.

## Chapitre 7

# Documentation développeur

Séance 1 :

Pour l’instant, nous en sommes qu’au début de notre projet, toutefois il y a certaines choses à savoir. Tout d’abord, la méthode *main* se situe dans la classe du même nom afin de rendre les choses plus simple. Chaque classe mérite d’être regardée, les classes *Road*, *Moto*, *Gfx* et *TH\_Game* peuvent être très utiles à la compréhension du code, avec le *KeyManager* évidemment. Vous pouvez changer la vitesse du jeu en changeant la vitesse maximale de la moto, c’est-à-dire la constante *MAX\_SPEED* dans la classe *Moto*.

Séance 3 :

Le code devient vraiment complet, chaque classe à son importance et ses spécificités. Comme par exemple les classes *Curbs*, *Elements* et *Road* et évidemment, toutes les classes de *Thread* vus précédemment.

# Chapitre 8

## Conclusion et perspectives

Séance 1 :

A ce stade de notre projet, nous avons plutôt bien avancé vis-à-vis du jalon proposé mais il nous reste encore plusieurs choses à implémenter, comme afficher une sprite de moto (qui a été dessinée durant la séance mais pas encore implémentée), afficher la vitesse ou encore implémenter et afficher un score. Et il reste évidemment des choses à améliorer comme l’affichage de la route qui n’est pas encore parfait.

Séance 3 :

Nous avons bien avancé dans le projet, que ce soit au niveau des fonctionnalités de bases ou de certaines fonctionnalités plus avancées. Toutefois, il nous reste de nombreuses choses à faire comme par exemple :

- Ajout d’obstacles
- Ajout de courbes dans la route
- Ajout d’un menu
- Ajout de plusieurs niveaux de difficulté
- de nouveau *sprites* et animations pour le joueur ainsi que les différents affichages du *HUD*
- Ajout de *SFXs* et de musique