

Rapport de Moto volante



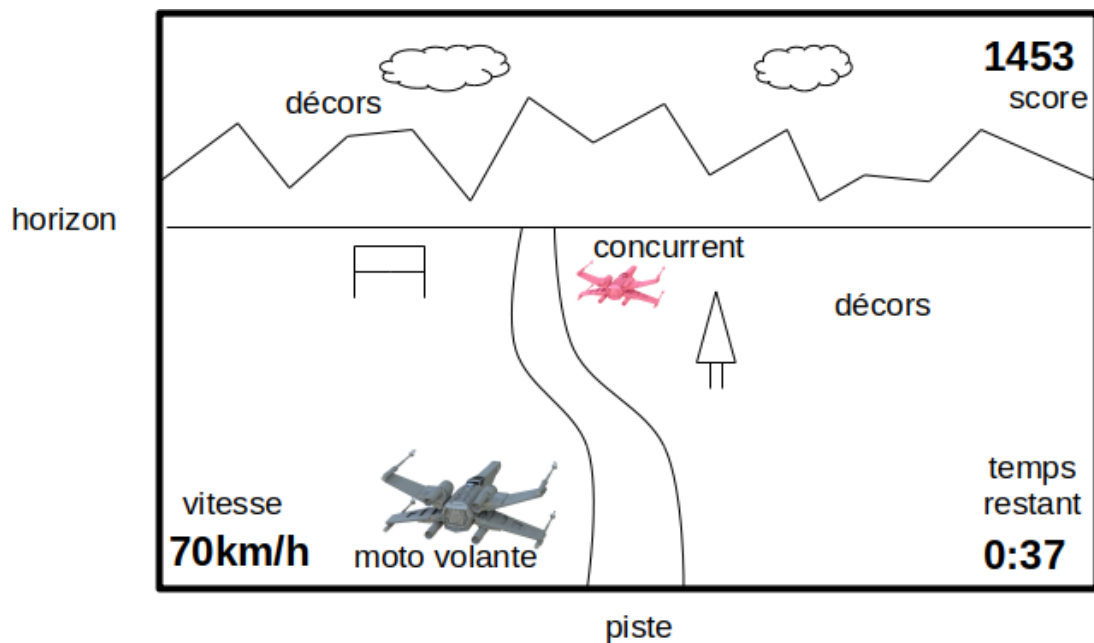
Sous la direction de
Thi Thuong Huyen Nguyen

Réalisé par Zhenhai XU
Yongjie LIU

License 3 Informatique

I. Introduction

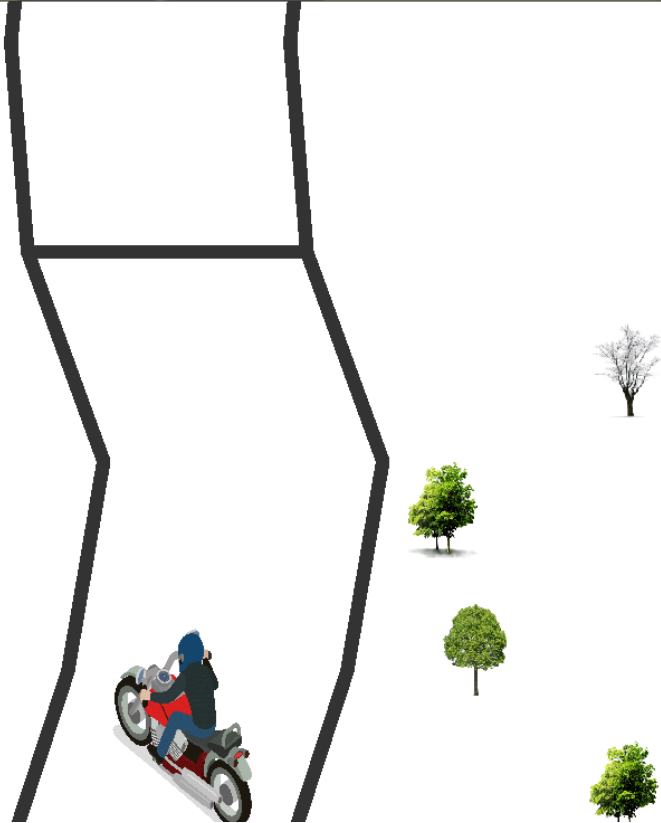
L'objectif du projet est de réaliser un jeu vidéo des années 80 de type « course de voiture » en vue à la première personne (le véhicule est vu de derrière). L'originalité de ce jeu est de permettre au joueur de piloter une sorte de moto sur coussin d'air pouvant se déplacer aussi horizontalement pour dépasser ses concurrents. La figure ci-dessous donne une vision schématique du jeu :





Rest temp : 1:53

Metre : 132 M



II. Analyse globale

Il y a 6 fonctionnalités principale à réaliser :

- L'interface graphique avec la moto, arrière-plan et la piste.
- Le défilement automatique de la piste.
- Mouvements la moto selon les touches du clavier.
- Mouvements du décor de fond selon les touches du clavier.
- Le mécanisme de calcul de vitesse de la moto.
- Affichage des données de jeu : Temps restant, kilométrage déjà parcouru, points gagnés...

III. Plan de développement

Liste des taches :

Séance 4 : Mise en place du projet

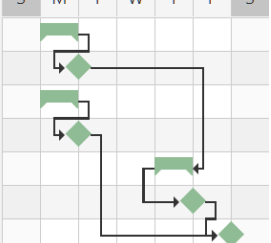
- Analyse du problème. (Binôme)
- Rédaction du plan de développement. (Binôme)
- Conception, développement et test d'une fenêtre avec la moto et la piste. (Zhenhai XU)
- Conception, développement et test du mécanisme de déplacement de la moto. (Zhenhai XU)
- Documentation du projet v0.1. (Binôme)

Title	Start Time	End Time	Sun 1/31 - Sat 2/06						
			S	M	T	W	T	F	S
Analyse du problème	02/01/2021	02/01/2021	◆						
▲ Conception, développement d'une fenêtre avec la moto et la piste.	02/02/2021	02/03/2021		■	■				
test de la fenêtre avec la moto et la piste.	02/05/2021	02/05/2021					◆		
▲ Conception, développement du mécanisme de déplacement de la moto.	02/02/2021	02/03/2021		■	■				
test du mécanisme de déplacement de la moto.	02/05/2021	02/05/2021					◆		
Documentation du projet v0.1.	02/06/2021	02/06/2021						◆	

Séance 5 et 6 : moteur de jeu et première version

- Conception, développement et test de l'animation de la piste à vitesse constante (Yongjie LIU)
- Conception, développement et test de la gestion du clavier en continu (Zhenhai XU)
- Conception, développement et test de la génération des décors de fond. (Yongjie LIU)
- Conception, développement et test de la Mouvements du décor de fond selon les touches du clavier. (Zhenhai XU)
- Documentation du projet v0.2. (Binôme)

Title	Start Time	End Time	Sun 2/07 - Sat 2/13						
			S	M	T	W	T	F	S
▲ Conception, développement et test de l'animation de la piste à vitesse constante	02/08/2021	02/09/2021							
test de l'animation de la piste à vitesse constante	02/09/2021	02/09/2021							
▲ Conception, développement et test de la gestion du clavier en continu	02/08/2021	02/09/2021							
test de la gestion du clavier en continu	02/09/2021	02/09/2021							
▲ Conception, développement de la Mouvements du décor de fond selon les touches du clavier.	02/11/2021	02/12/2021							
test de la Mouvements du décor de fond selon les touches du clavier.	02/12/2021	02/12/2021							
Documentation du projet v0.2.	02/13/2021	02/13/2021							



- Consulter les commentaires de l'enseignant et prévoyez de les modifier. (Binôme)
- Réécrire une partie du code et supprimer le code répété à l'aide de l'héritage. (Zhenhai XU)
- Conception, développement et test du décompte des kilomètres et l'affichage. (Zhenhai XU)
- Amélioration la gestion du clavier en continu. (Zhenhai XU)
- Conception, développement et test de la génération des décors décorations au bord de la piste. (Yongjie LIU)
- Conception, développement et test de la Mouvements au bord de la piste

selon les touches du clavier. (Yongjie LIU)

- Documentation du projet v0.3(Binôme)

Title	Start Time	End Time	Sun 2/14 - Sat 2/20						
			S	M	T	W	T	F	S
Consulter les commentaires de l'enseignant et prévoyez de les modifier.	02/15/2021	02/15/2021							
Réécrire une partie du code et supprimer le code répété à l'aide de l'héritage.	02/15/2021	02/16/2021							
Conception, développement et test du décompte des kilomètres et l'affichage.	02/16/2021	02/17/2021							
Amélioration la gestion du clavier en continu.	02/17/2021	02/18/2021							
Conception, développement et test de la génération des décors décorations au bord de la piste.	02/15/2021	02/16/2021							
Conception, développement et test de la Mouvements au bord de la piste selon les touches du clavier.	02/16/2021	02/18/2021							
Documentation du projet v0.3	02/20/2021	02/21/2021							

IV. Conception Générale.

Dans ce projet, nous utilisons la pensée MVC et la divisons en trois parties, à savoir la Vue du Modèle et le Contrôle.

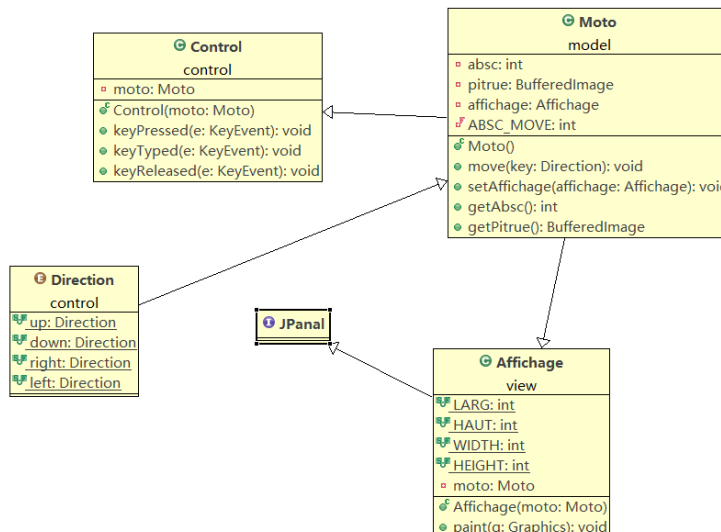
- Le package Modèle est responsable des données et de l'état de l'arrière-plan, des véhicules et des voies. La modification des données dans le modèle peut changer les détails de fonctionnement du jeu.
- Le package Vue est responsable du dessin, c'est-à-dire en acceptant les données dans d'autres packages, en dessinant la moto, l'arrière-plan et la voie, ainsi que le kilométrage et le temps restant affichés à l'écran.
- Le package de contrôle met à jour les données du package vue en modifiant l'état de l'objet et réalise une animation à travers plusieurs peintures.

V. Conception détaillée

Affichage hériter la classe JPanel, il a obtenu les paramètres nécessaires à la dessiner à partir d'autres packages. On utilise les constants LARG et HAUT pour définir la taille de la fenêtre, et les constants WIDTH et HEIGHT pour définir la taille de la moto.

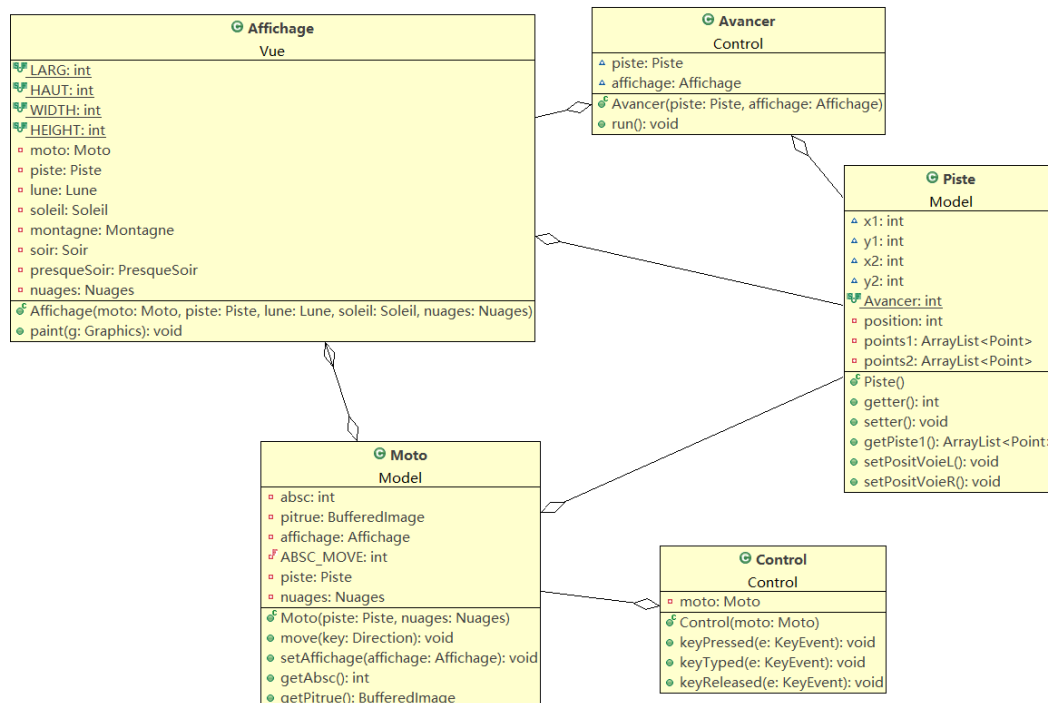
Contrôle En utilisant l'interface KeyListener, obtenez le clavier utilisé par le joueur, obtient le clavier utilisé par le joueur via l'interface du keylistener, et transmet les paramètres obtenus à moto.

La moto charge l'image et affiche une image de la moto sur la fenêtre. La fonction de move () modifie les coordonnées de la moto à travers les paramètres reçus, de sorte que la moto puisse se déplacer à gauche et à droite.



Après avoir réalisé les fonctions les plus élémentaires, nous avons commencé à réaliser la fonction de défilement de voie. Générez aléatoirement des points et ajoutez-les au tableau, puis lisez ces points dans une boucle pour dessiner une piste.

Nous utilisons des threads pour modifier en continu les coordonnées des points pour réaliser le défilement de la piste.

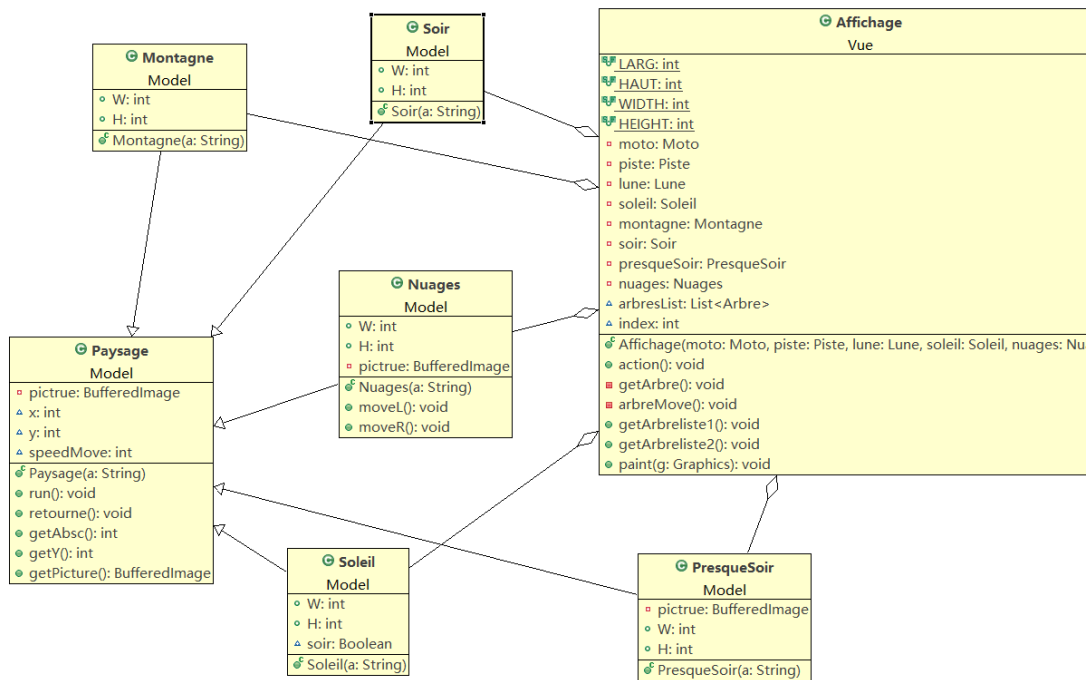


Algorithme de génération de pistes :

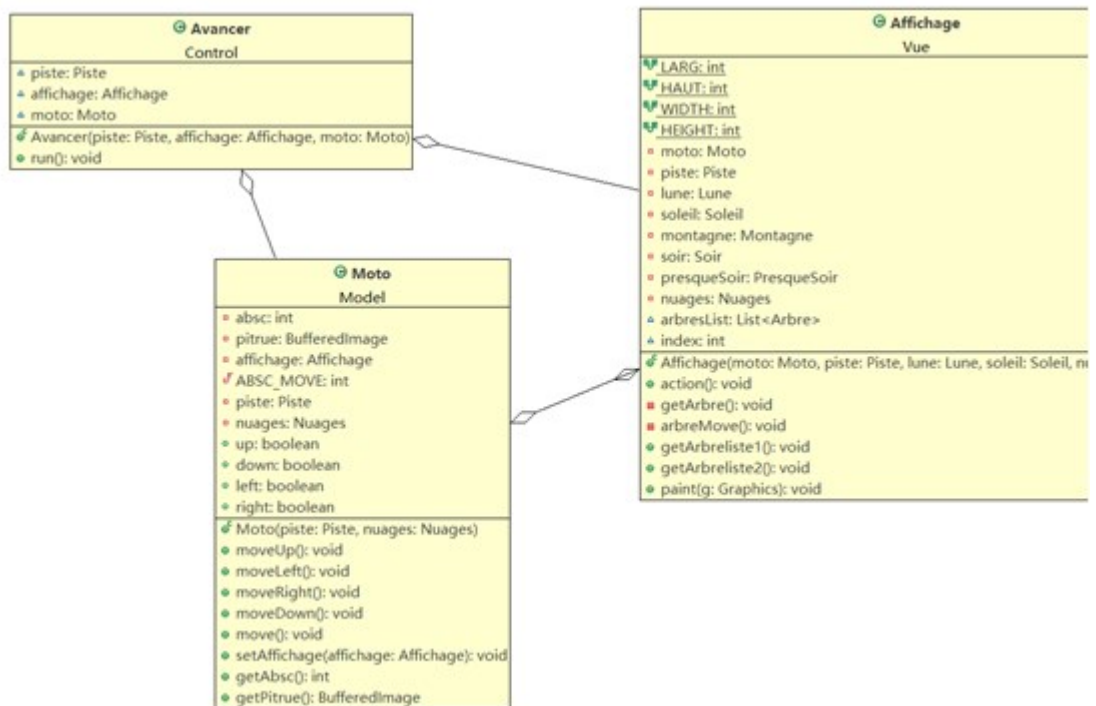
Puisque les deux segments de ligne de la voie sont relativement parallèles, il suffit de générer un point aléatoire d'un segment de ligne. En utilisant `Random().nextInt()`, nous pouvons obtenir aléatoirement un nombre à utiliser comme coordonnée de la trace.

Ensuite, nous avons ajouté des décors de fond, nous avons remplacé trois photos dans un cycle pour représenter le passage d'une journée, avec des nuages et de la piste se déplaçant selon le mouvement de la moto.

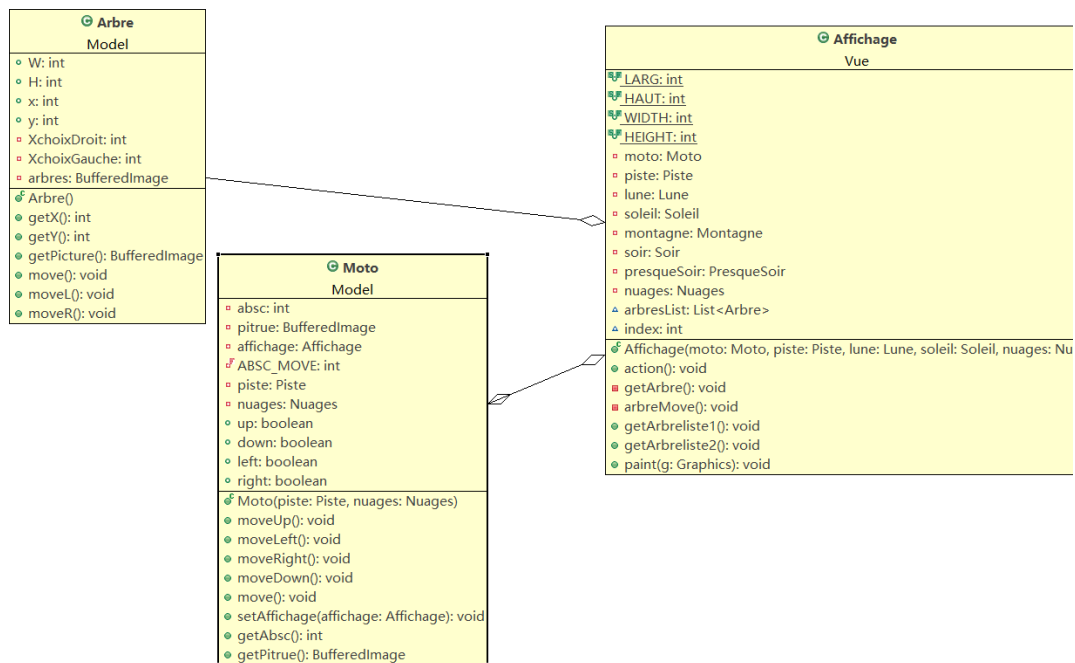
Nous créons une classe parente Paysage pour réduire la duplication de code grâce à l'héritage



Nous avons utilisé Booléen comme commutateur. Par exemple, lorsque vous appuyez sur ← sur le clavier, activez l'interrupteur et déplacez la route et la décoration vers la droite. Lorsque on relâche ←, désactivez l'interrupteur et la piste et les décors arrêtent de bouger. Cela peut être une bonne solution au conflit de touches du clavier.

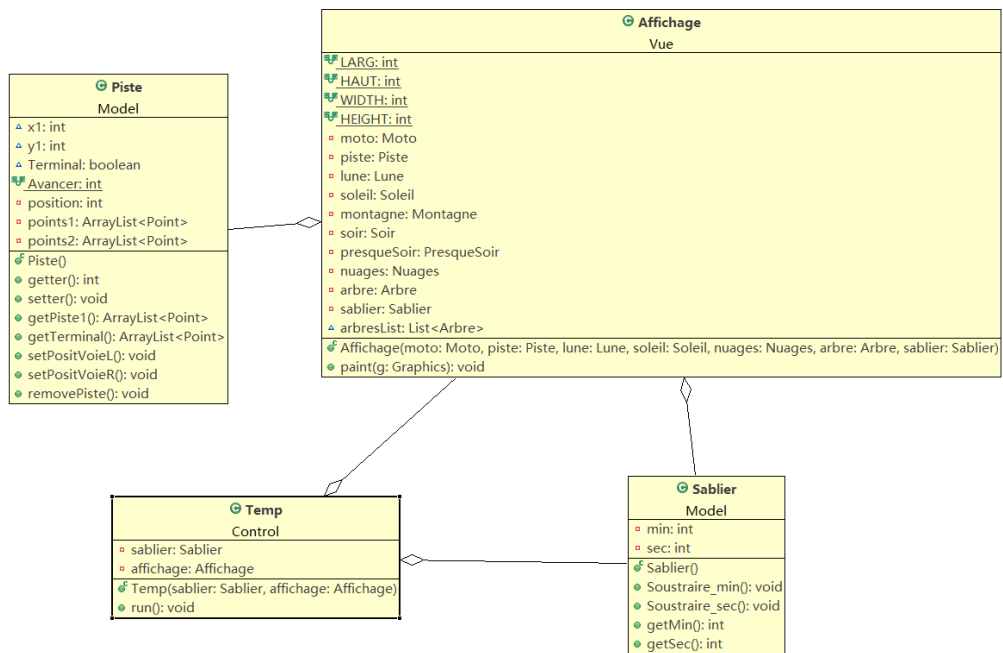


Le paysage des deux côtés de la route est plus compliqué que les décors de fond. Nous générons aléatoirement des points de coordonnées, les mettons dans un tableau, puis chargeons l'image. Au fur et à mesure que la piste défile, les décorations des deux côtés de la route se déplacera vers le bas, et à travers une boucle Ils se déplaceront également à gauche et à droite avec la touche du clavier.



Et puis on crée un thread Temps pour afficher le sablier qui compte le temps. Et on crée la class sablier qui fait le comptage décroissant.

Pour réaliser la ligne terminal(terminus), on les fait sur la class piste, on cree un nouvel objet point2 pour déposer les points qui est le terminus.



VI. Documentation utilisateur :

- Prérequis : Java avec un IDE (ou Java tout seul si vous avez fait un export en .jar exécutable)
- Mode d'emploi (cas IDE) : Importez le projet dans votre IDE, sélectionnez la classe Main à la racine du projet puis « Run as Java Application ». Appuyé le clavier sur la fenêtre pour faire déplacer de la moto.
- Mode d'emploi (cas .jar exécutable) : Appuyé le clavier sur la fenêtre pour faire déplacer de la moto. La moto peut juste déplacer A gauche ou A droite.

VII. Documentation développeur

1. On doit savoir la fonctionnalité de chaque classe. Classe Model pour définir l'état de la moto. Classe affichage, pour définir l'affichage de l'écran avec de la moto. Class Control pour déplacer de la moto.
2. On doit savoir les constants, il y a plusieurs de constants pour connaître.

VIII. Conclusion

Au cours de la première semaine de création de ce jeu, nous avons lu le sujet, analysé les besoins du jeu et connu le sens du développement. Nous avons développé un prototype de jeu, une fenêtre, une piste, un horizon et un petit bloc noir.

Dans la deuxième semaine de création de ce jeu, nous avons remplacé les petits blocs noirs par des images de motos. Nous avons des pistes courbes et de nouvelles pistes peuvent être générées. Certains arrière-plans ont été ajoutés. L'arrière-plan et les pistes changeront avec le mouvement de la moto. Produire du mouvement.

Au cours de la troisième semaine, nous avons consulté les commentaires du professeur et réalisé que la gestion du clavier était imparfaite, et nous avons apporté des améliorations. De plus, comme nous n'avons pas bien implémenté la duplication de code prévue et que nous n'avons pas utilisé l'héritage, nous réfléchissons de manière plus réfléchie à l'avenir.

Cette semaine, nous avons fait Ajout de points de contrôles (représentés par une ligne horizontale sur la piste), Ajout d'un décompte de temps et détermination de la fin de partie, Ajout d'obstacles et détection de collisions qui ralentissent le véhicule.

•

•