

Rapport de projet PCII



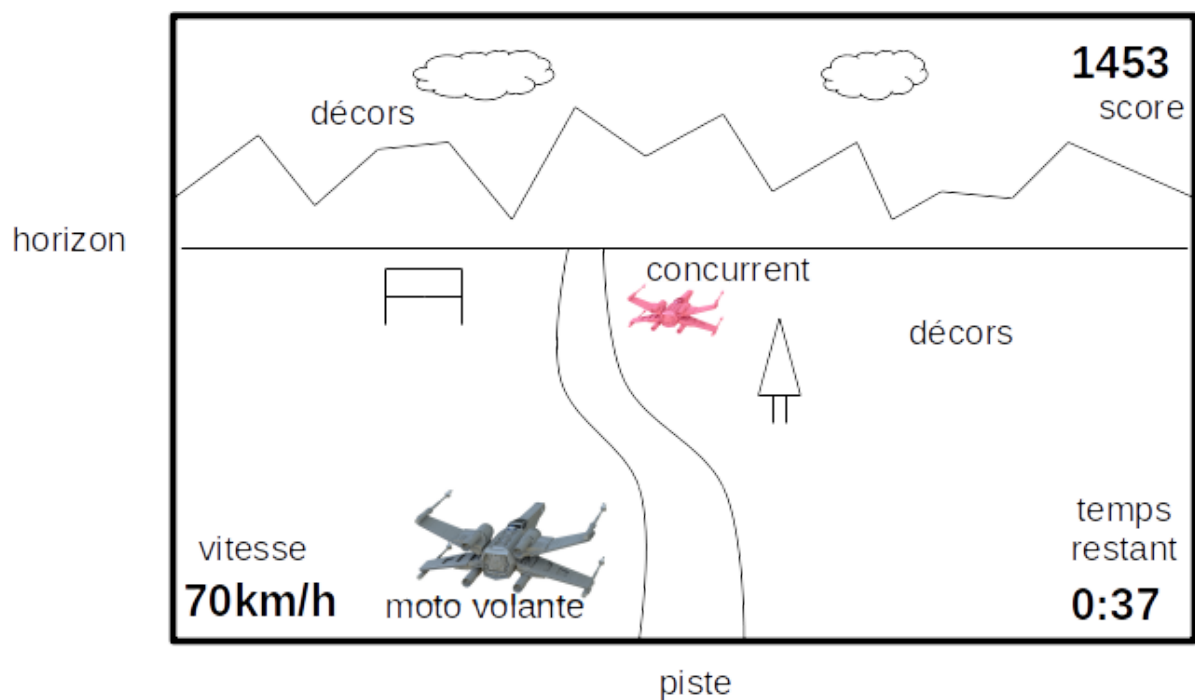
Réalisé par
Hongyu YAN

Sous la direction de
Thi Thuong Huyen Nguyen

Année universitaire 2020-2021
Licence Informatique

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 :



Analyse globale

Il y a 5 fonctionnalités principales:

1. l'interface graphique avec le véhicule, l' horizon et la piste,
2. le défilement automatique de la piste,
3. l'apparition des points de contrôles à intervalles réguliers
4. la réaction de le véhicule aux clavier de l'utilisateur,
5. le mécanisme de calcul de l'accélération du véhicule en fonction de la position par rapport à la piste ,
6. Des données de jeu : temps restant et kilométrage

Nous nous intéressons d'abord uniquement à un sous-ensemble de fonctionnalités qui sont prioritaires et simples à réaliser:

- Création d'une fenêtre dans laquelle est dessiné le véhicule, l' horizon et la piste ;
- Déplacement du véhicule à droite et à gauche lorsqu'on tape les flèches du clavier

Plan de développement

Liste des tâches:

- Séance 4 : Mise en place du projet
 - Analyse globale
 - Rédaction du plan de développement
 - Conception, développement et test d'un fenêtre avec une simple ligne brisée fixe (la piste) et limitée par un horizon, le véhicule
 - Conception, développement et test du mécanisme de déplacement de le véhicule
 - Documentation du projet v0.1

ID	Title	Start Time	End Time	Sun 1/31 - Sat 2/06						
				M	T	W	T	F	S	S
1	Analyse globale	02/01/2021	02/01/2021							
2	Rédaction du plan de développement	02/01/2021	02/01/2021							
3	Conception, développement et test d'un fenêtre avec la piste et limitée par un horizon, le véhicule	02/06/2021	02/06/2021							
4	Conception, développement et test du mécanisme de déplacement de le véhicule	02/06/2021	02/06/2021							
5	Documentation du projet	02/07/2021	02/07/2021							
6	Introduction + Analyse globale + Plan de développement	02/07/2021	02/07/2021							
7	Conception générale + Conception détaillée + Résultat + Documentation utilisateur + Documentation développeur	02/07/2021	02/07/2021							
8	Conclusion et perspectives	02/07/2021	02/07/2021							

- Séance 5 et 6 : moteur de jeu et première version
 - Conception, développement et test de l'animation de la piste à vitesse constante
 - Conception, développement et test de la gestion du clavier en continu
 - Conception, développement et test de l'affichage de décompte des kilomètres
 - Conception, développement et test des mouvements du décors de fond selon les touches du clavier
 - Conception, développement et test du calcul de l'accélération et la vitesse
 - Documentation du projet v0.2
- Séances 7 et 8 : mécanique du jeu
 - Conception, développement et test de la détection de collisions
 - Conception, développement et test du mécanisme de points de contrôles
 - Conception, développement et test du mécanisme d'un décompte de temps
 - Documentation du projet v0.9

- Séance 9 et 10 : Finalisation du projet
 - Conception, développement et test des décors
 - Conception, développement et test de l'animation de la piste
 - Conception, développement et test d'un écran d'accueil
 - Conception, développement et test du mécanisme d'adversaires
 - Conception, développement et test de la création d'une sensation de profondeur
 - Documentation du projet v1
 - Préparation de la soutenance

Conception générale

L'interface graphique s'est construite autour du modèle MVC. Le package *vue* s'occupe de dessiner l'interface, c'est à dire, une fenêtre avec des voitures, une piste et des décors dedans.

Le package *model* définit l'ensemble des données qui caractérisent l'état de l'interface. La modification de ces données correspond à un changement de l'affichage dans l'interface graphique.

Le package *control* effectue les changements dans l'état et informe la *vue* des changements. Il modifie les états des dessins afin de faire déplacer la voiture, avancer la piste et les décors, puis ils informent la vue des changements pour redessiner.

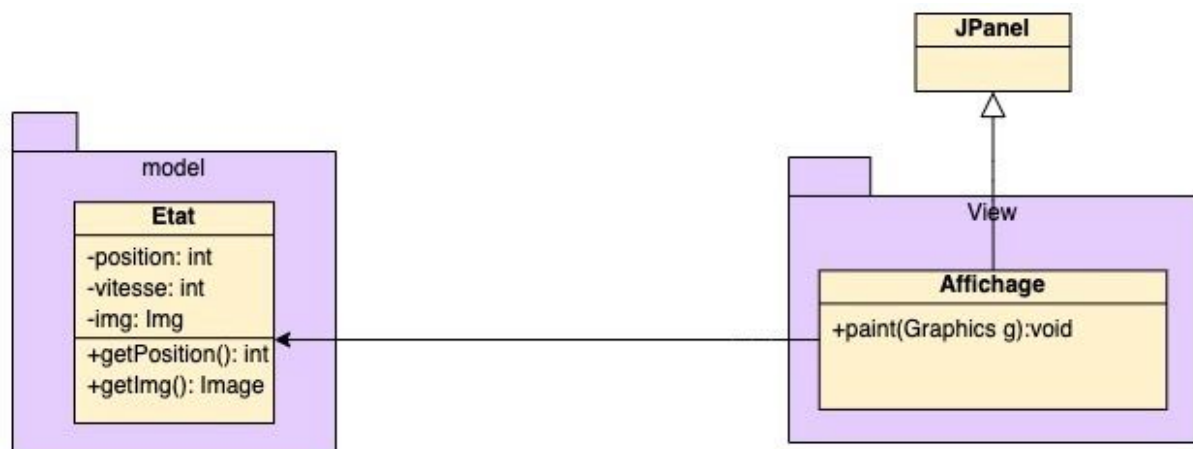
Conception détaillée

Pour la fenêtre avec une voiture, j'utilise l'API Swing et la classe `JPanel`. Je définis les dimensions de la voiture et de la fenêtre dans des constantes.

Constante(s) de la classe `Affichage`:

- `LARG` : Largeur de l'interface
- `HAUT` : Hauteur de l'interface
- `WIDTH` : Largeur de la voiture
- `HEIGHT` : Hauteur de la voiture

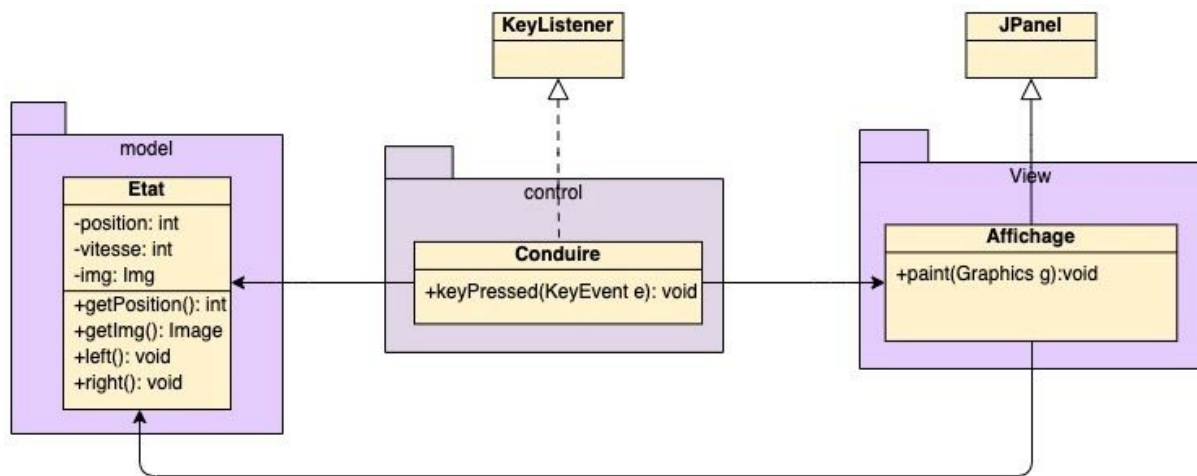
Voici un diagramme de classe pour afficher une fenêtre avec un véhicule dedans.



Pour le déplacement du véhicule, j'utilise la programmation événementielle avec la classe `KeyListener` et la distance est définie dans une constante.

Constante(s) de la classe `Etat` :

- `DEPLACE` : La valeur du déplacement de quelques pixels du véhicule



Pour le défilement automatique de la piste, je dois d'abord générer aléatoirement les coordonnées et j'ai proposé un algorithme.

piste() : void

x <- **new** Random().nextInt(50) + Affichage.**LARG**/2 - 50

y <- Affichage.**HAUT**/3

Ajouter Point(x,y) dans la liste des points

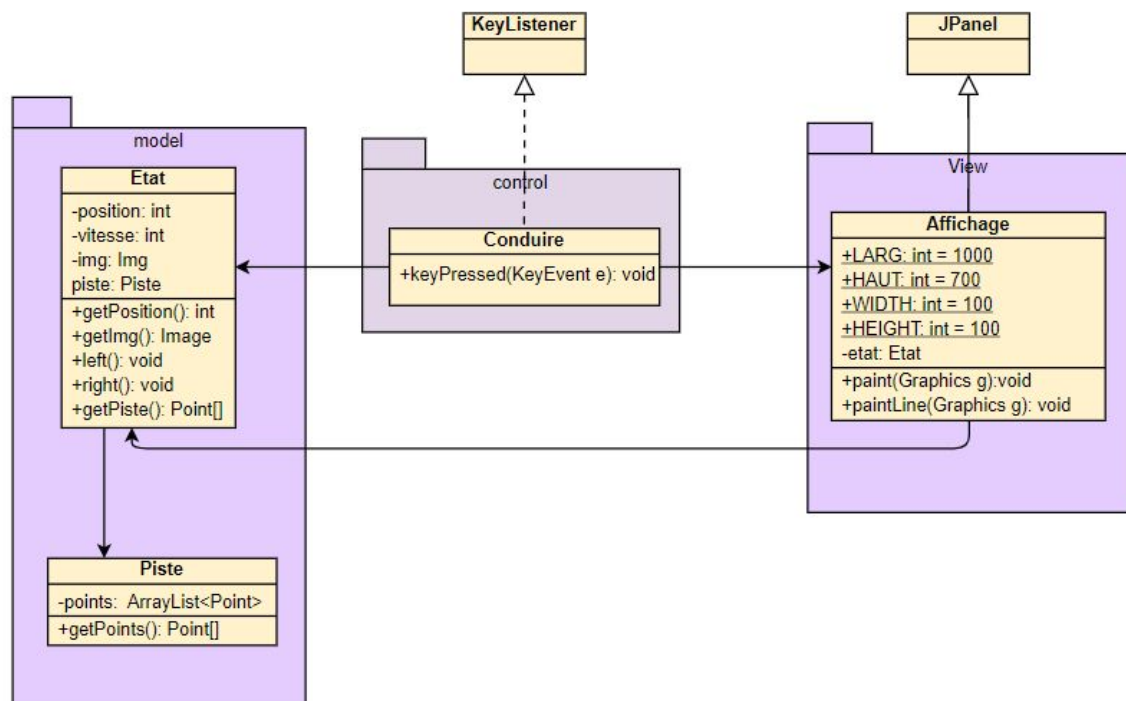
tant que y <= Affichage.**HAUT** **faire**

x <- **new** Random().nextInt(50) + Affichage.**LARG**/2 - 50

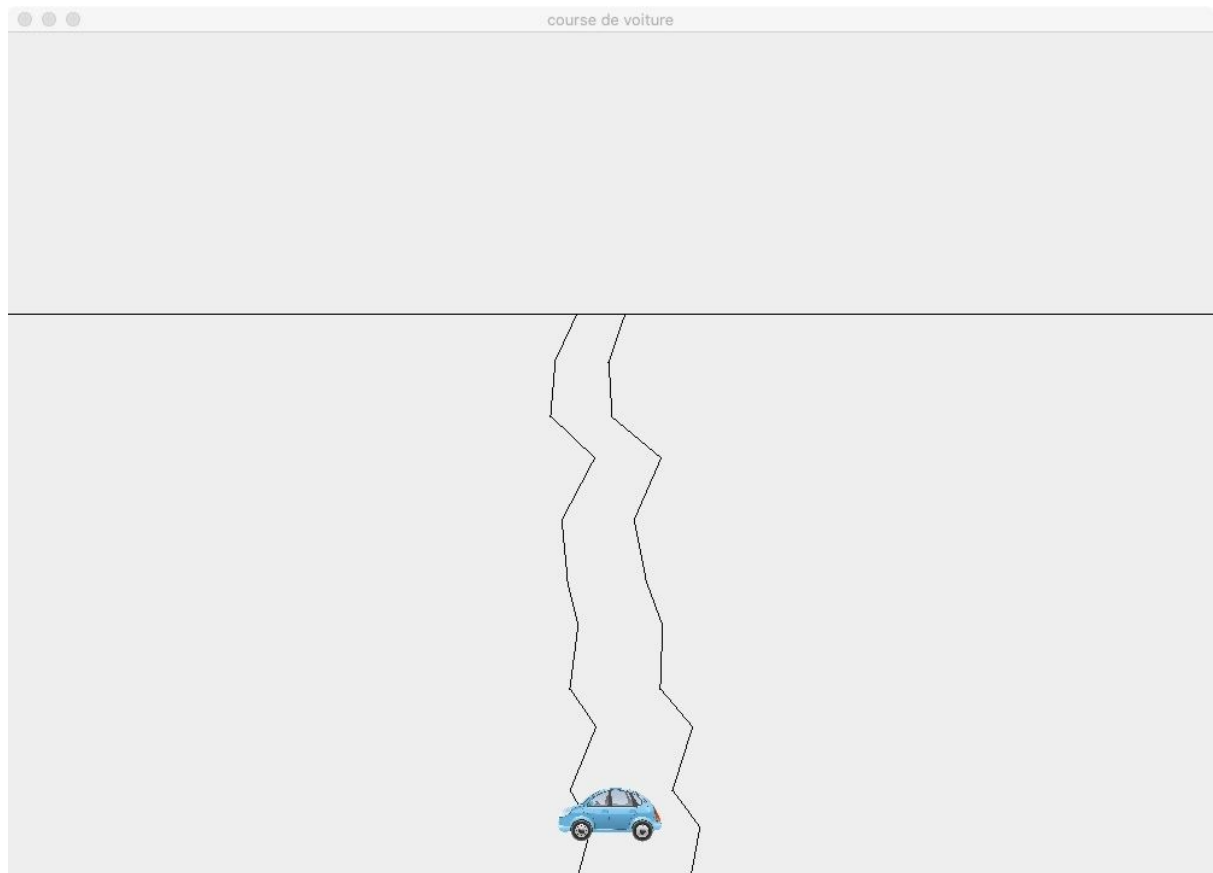
y <- y + **new** Random().nextInt(30) + 30;

Ajouter Point(x,y) dans la liste des points

fin tant que



Résultat



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 ». Appuyez sur les touches gauche et droite du clavier pour contrôler le mouvement du véhicule
- Mode d'emploi (cas `.jar` exécutable) : double-cliquez sur l'icône du fichier `.jar`. Appuyez sur les touches gauche et droite du clavier pour contrôler le mouvement du véhicule

Documentation développeur

C'est la classe Main qui contient la méthode `main`. J'utilise le model MVC pour organiser mes codes.

Allez dans le package `view` pour voir comment afficher l'interface. Vous pouvez modifier les valeurs des constants définis dans la classe `Affichage` afin de changer la taille de la fenêtre et celle du véhicule.

Les prochaines fonctionnalités seront l'animation de la piste à vitesse constante pour donner l'impression que le véhicule avance, la création des décors et ses déplacements automatiques.

Conclusion et perspectives

Pour la séance 4:

Nous avons réalisé une analyse globale de ce projet et formulé un plan de développement.

Ensuite, Nous avons réalisé les fonctionnalités que nous avions prévu de mettre en œuvre dans cette séance:

- Création d'une fenêtre dans laquelle est dessiné le véhicule, l' horizon et la piste ;
- Déplacement du véhicule à droite et à gauche lorsqu'on tape les flèches du clavier

Dans la prochaine étape, nous mettrons en œuvre la version moteur de jeu et la première.