

Projet PCII

Introduction

Nous allons à travers ce projet réaliser un jeu vidéo de type « course de voiture » à la première personne. Le but de ce jeu est de contrôler un véhicule aérien qui doit esquiver les obstacles et dépasser les autres véhicules appelés les concurrents. Nous allons, tout d'abord, utiliser la bibliothèque Swing afin de programmer l'interface graphique et après avoir affiché la moto, implémenter la gestion des déplacements par le biais des flèches directionnelles du clavier.

Analyse globale

Les principales fonctionnalités à développer sont l'interface graphique avec la création de la moto et la gestion du clavier qui va permettre de la déplacer sur la fenêtre de jeu.

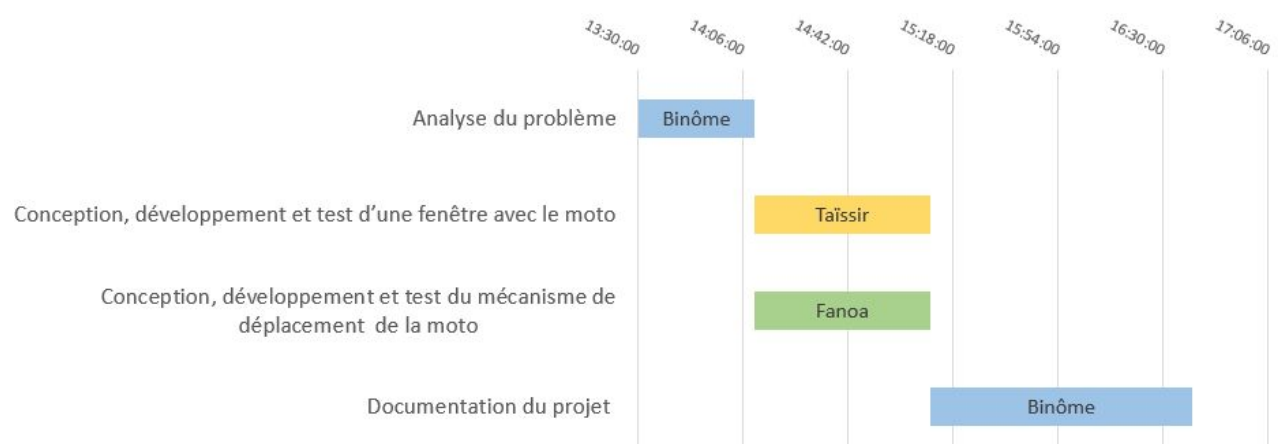
Ces fonctionnalités sont prioritaires car elles permettent de mettre en place l'environnement de jeu et sont plutôt simples à implémenter.

Plan de développement

Liste des tâches : (théorique)

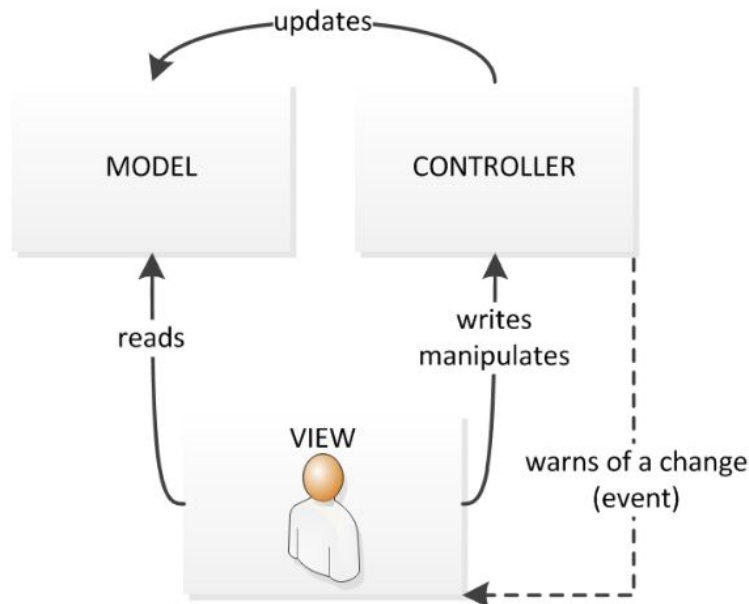
- Analyse du problème (40 mn)
- Conception, développement et test d'une fenêtre avec le moto (60 mn)
- Conception, développement et test du mécanisme de déplacement de la moto (60 mn)
- Documentation du projet (90 mn)

Diagramme de Gantt des différentes premières tâches du projet:



Conception générale

Nous avons adopté le motif MVC, Model View Controller pour le développement de notre interface graphique.



Le motif MVC (source : Wikimedia)

Ce motif est composé de quatre classes principales :

- Affichage pour la vue
- Etat pour le modèle
- Control pour le contrôleur
- Main pour lancer le programme

Conception détaillée

Le développement d'une fenêtre de test consiste principalement en la création et l'implémentation d'une classe Affichage. Cette classe hérite de la classe JPanel, ce qui nous permet de l'ajouter à une fenêtre (on réalise cette opération dans le main avec l'instruction "fenetre.add(affichage)". Tout ce qui est "peint" par l'affichage y figurera donc. La classe affichage est construite autour d'une méthode principale, héritée de JPanel, la méthode paint(Graphics g). C'est au sein de cette méthode qu'on dessine sur la fenêtre les différents éléments du programme. Ici, nous avons choisis pour commencer de d'utiliser des images temporaires pour

représenter facilement chacun des éléments du jeu. Beaucoup des éléments de cet affichage sont provisoires car l'objectif est ici de faire le minimum pour pouvoir interagir avec le modèle. Une fois que les fonctionnalités de base du modèle seront implémentées, un affichage plus intéressant pourra être produit.

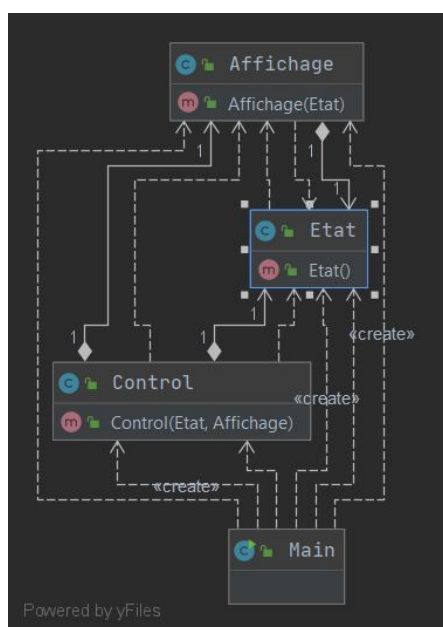
Pour le déplacement de la moto dans la fenêtre de jeu, nous utilisons la programmation événementielle avec la classe KeyListener. La position de la moto est définie dans la classe Etat avec la constante "pos", qui va subir des modifications de coordonnées selon la touche du clavier appuyée. En effet, la classe Etat contient quatre méthodes qui vont gérer le déplacement : moveUp(), moveDown(), et moveLeft() et moveRight(). Ces méthodes vont être utilisées dans la méthode KeyPressed de la classe Control qui va appeler l'une des quatre méthodes selon la touche pressée.

La classe Control implémente l'interface KeyListener pour la gestion du clavier comme on peut le voir dans le diagramme suivant :

Diagramme de classe présentant l'implémentation de l'interface KeyListener dans la classe Control :

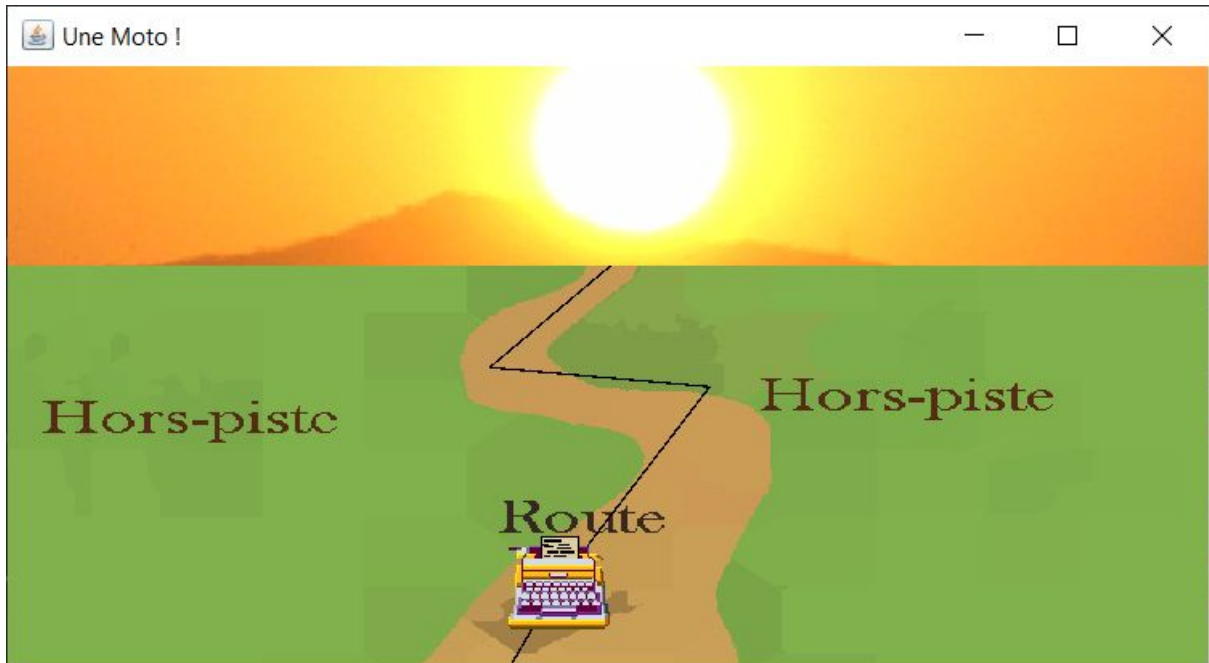


On peut également observer l'ensemble des interactions entre les classes sur un diagramme plus général :



Résultat

L'image ci-dessous montre l'affichage de la fenêtre lorsque le programme est lancé, on y retrouve la moto au centre avec son environnement que nous allons implémenter au fil des séances.



Documentation utilisateur

- Prérequis : Java avec un IDE
- Mode d'emploi : importer le projet dans l'IDE, sélectionner la classe Main puis "Run as Java Application"

Une fenêtre s'ouvre avec la moto au centre, utiliser les flèches directionnelles pour la déplacer dans son environnement.

Documentation développeur

Les classes principales à regarder sont celles qui implémentent le schéma MVC : Affichage, Etat, Control. La classe Main permet de lancer le programme.

Les principales constantes que l'on peut modifier pour changer le fonctionnement du code sont, tout d'abord dans la classe Affichage, la taille de la fenêtre de jeu défini par les constantes "WIDTH" et "HEIGHT". Les constantes liées à la moto peuvent aussi être modifiées, notamment sa largeur et hauteur définis respectivement par les constantes "largeurMoto" et "hauteurMoto" et de combien de pixels elle peut se déplacer définie par la constante "move".

Dans la classe Etat, la position initiale de la moto peut être modifiée en modifiant les coordonnées de la constante "pos".

Les prochaines fonctionnalités seront la création des obstacles, des concurrents que la moto doit dépasser et l'animation de la piste infinie.

Conclusion et perspectives

Nous avons donc commencé à créer l'environnement de jeu avec, tout d'abord, la moto et son déplacement dans la fenêtre d'affichage, et le décor que nous allons compléter au fil des séances.

On pourra voir s'ajouter à ce décor les concurrents représentés par d'autres motos et les obstacles que le joueur devra éviter.