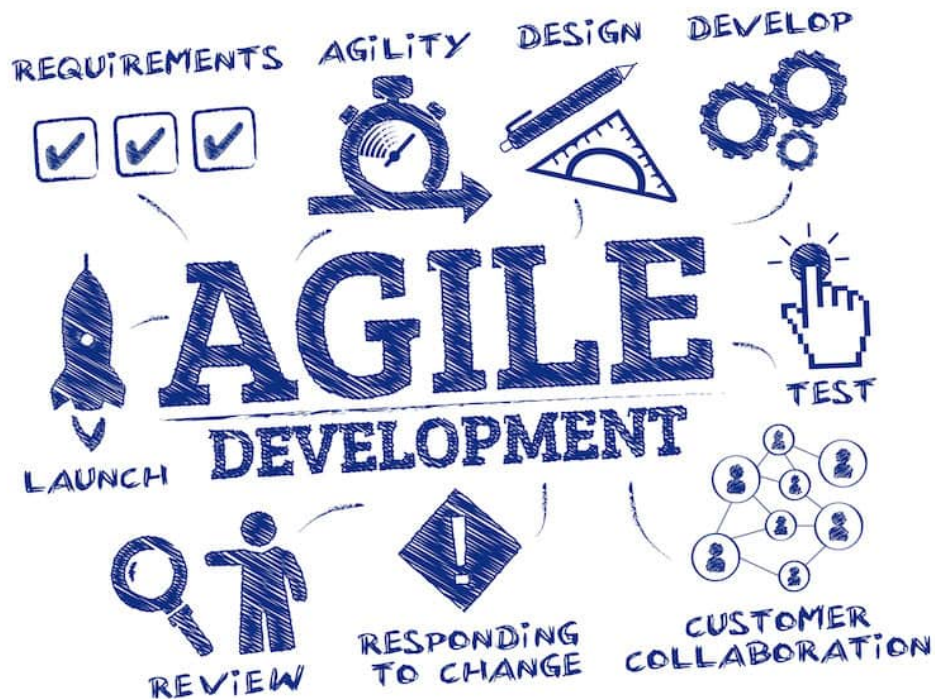


Compte rendu PROJET :
Conception Agile de Projet Informatique
et Génie Logiciel



Réalisé par : ANTONY RAJAN Arun, CAYRAT Thomas, DEKKERS Eiko
Guidé par : LACHAND-PASCAL Valentin

Introduction :

Le principe du projet est de créer un jeu type "Balle au Prisonnier" au format MVC, avec les librairies de JavaFx, sous IntelliJ, suivant la méthode Agile.

Définition des termes :

- Jeu "Balle au prisonnier" :

Ce jeu est un jeu de sport qui oppose deux équipes. Le but étant d'éliminer tous les joueurs de l'équipe adverse en les touchant à l'aide d'une balle.

Une partie est composée d'un terrain délimité en deux camps (un par équipe), une balle et des deux équipes (composées chacune d'un personnage contrôlé par un joueur et deux personnages contrôlés par ordinateurs).

- JavaFX :

JavaFX est un framework et une bibliothèque d'interface utilisateur issue du projet OpenJFX, qui permet aux développeurs Java de créer une interface graphique pour des applications de bureau, des applications internet riches et des applications smartphones et tablettes tactiles.

- Méthode Agile :

La méthodologie Agile se base sur une idée simple. Planifier la totalité de votre projet dans les moindres détails avant de le développer est contre-productif.

La méthode Agile recommande de se fixer des objectifs à court terme. Le projet est donc divisé en plusieurs sous-projets. Une fois l'objectif atteint, on passe au suivant, et ce jusqu'à l'accomplissement de l'objectif final. Cette approche est plus flexible. Puisqu'il est impossible de tout prévoir et de tout anticiper, elle laisse la place aux imprévus et aux changements. L'avantage majeur de l'approche Agile relève de sa flexibilité. L'équipe projet réagit rapidement aux changements du client et aux imprévus.

Par contre, la méthode Agile n'est en outre pas adaptée pour les entreprises avec une structure hiérarchique très forte, à cause de son fonctionnement collaboratif.

- Modèle MVC :

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- Un modèle (Model) contient les données à afficher :

Élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application. Par exemple pour une application de banque, le modèle représente des comptes, des clients, ainsi que les opérations telles que dépôt et retraits, et vérifie que les retraits ne dépassent pas la limite de crédit.

- Une vue (View) contient la présentation de l'interface graphique :

Partie visible d'une interface graphique. La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle. Dans une application de bureau classique, la vue obtient les données nécessaires à la présentation du modèle en posant des questions. Elle peut également mettre à jour le modèle en envoyant des messages appropriés. Dans une application web une vue contient des balises HTML.

- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur :

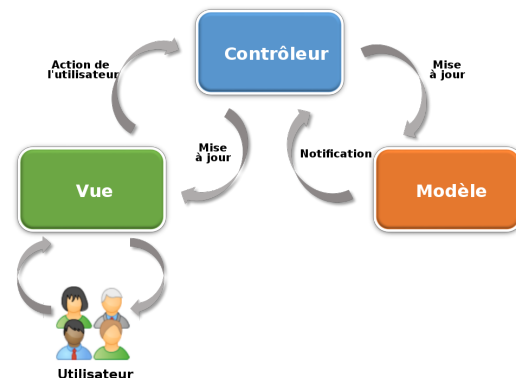
Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

Design Patterns :

Concernant le projet, nous voulions nous servir de plusieurs Design Pattern, tel que le MVC, le singleton et le factory.

Un Design Pattern (Patron de Conception) est une façon de résoudre un problème de conception logiciel récurrent. Il existe différents types de design pattern, les Design Pattern de création, structurels, comportementaux.

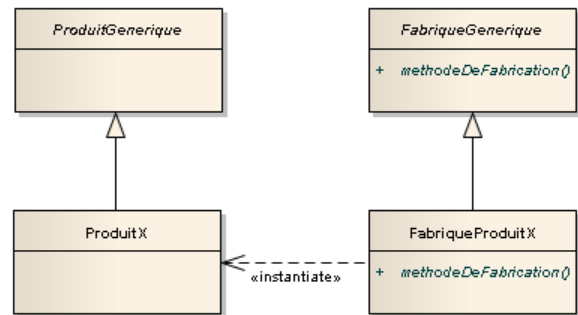
- Le MVC est séparé en 3 parties, Modèle qui traite en arrière-plan les éléments à afficher, Vue qui affiche à l'utilisateur les éléments du jeu et Contrôleur qui récupère les inputs de l'utilisateur.



- Le Singleton (patron de création), l'objectif est de restreindre l'instanciation d'une classe à un objet qui sera en 'global'. Il est utilisé lorsqu'on a besoin d'exactly un objet pour coordonner des opérations dans un système. Le modèle est parfois utilisé pour son efficacité, lorsque le système est plus rapide ou occupe moins de mémoire avec un seul objet qu'avec beaucoup d'objets similaires.

| Singleton | |
|-----------|----------------------------------|
| - | <u>singleton : Singleton</u> |
| - | Singleton() |
| + | <u>getInstance() : Singleton</u> |

- Le Factory (patron de création), l'objectif est de permettre d'instancier des objets dont le type est dérivé d'un type abstrait. La classe exacte de l'objet n'est donc pas connue par l'appelant.



Fonctionnalités :

Débogage du code existant et fonctionnalités implémentées :

- Les Joueurs peuvent se déplacer librement à l'horizontale (à une vitesse aléatoire fixée au lancement de l'application).
- Les Joueurs peuvent contrôler l'angle de tir de la balle seulement sur un arc de cercle de 180° devant le joueur.

Chaque équipe contient 1 joueur et 2 bots contrôlés par ordinateur.

Les Joueurs étant un objet de la classe Player, contrôlés par l'utilisateur.

Les ordinateurs ayant hérité de la classe Player avec seul modifications qu'ils sont contrôlés aléatoirement.

En cours d'implémentation :

- Afficher et Tirer la balle, la balle allant dans la direction du joueur, si en contact avec un joueur adverse : réceptionner ou se faire toucher par la balle.

Manque de temps pour :

- Un agencement du projet en pattern MVC avec un singleton pour créer les Joueurs et une Factory pour création des balles.
- un score des points du match

Extensions possible pour le projet :

- Modification du nombre de joueurs par équipe
- Mettre une limite de point pour avoir une équipe vainqueur
- Une fenêtre de Menu contenant "Pause", "Réinitialiser la partie", "Réglage son", "Réglage FPS", "Quitter".

Problèmes rencontrés :

Enorme perte de temps, à cause de JavaFx, Maven et IntelliJ qui ne fonctionnaient pas sur nos PC pour on ne sait quelle raison. Nous avons dû travailler à trois sur un PC jusqu'en novembre. La situation s'est débloquée le vendredi 11/11 : "vrai début du projet".

Suite à cela, nous avons commencé à implémenter les fonctionnalités de base du projet tels que les Ordis, et l'affichage des membres des équipes.

Puis vient l'implémentation et l'affichage des balles qui nous a bloqué du mardi 15/11 au mardi 29/11.

