

Rapport du projet de Java : Battle Jet

Groupe 20 : Clément Brancart, Damien Delander, Faysal Bearmany

Cahier de charges :

Notre projet consiste à programmer un jeu de bataille aérienne sous le même principe qu'un jeu de bataille navale, sous un mode de jeu aléatoire et multi-joueurs.

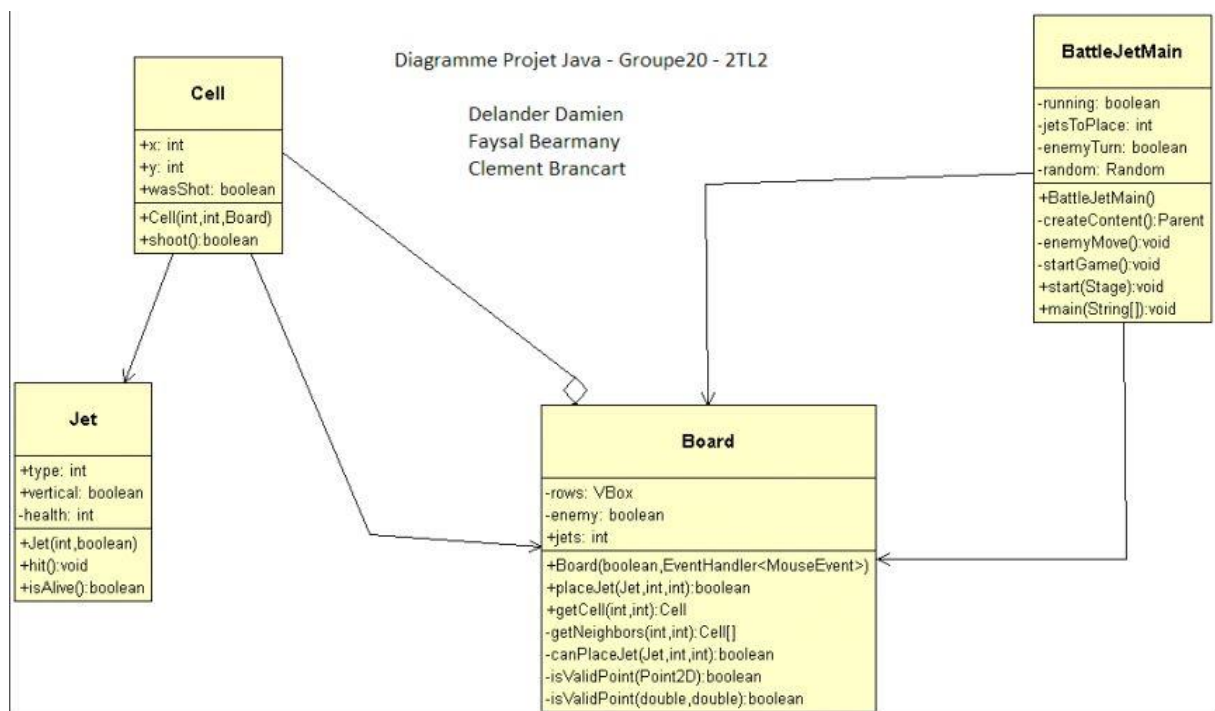
Nous nous sommes organisés de tel :

-Faysal : tout ce qui est code de fonctionnement du jeu.

-Damien : tout ce qui est interface.

-Clément : tout ce qui est « administratif » c'est à dire mettre les codes sur GitHub et rendre ce qu'on devait au professeur.

Diagramme UML :



BattleJetMain : le fonctionnement du jeu.

Board : l'interface du jeu.

Cell : toutes les cellules (cases), si elle est bleu ce n'est pas touché, si rouge touché, grise pas encore visé.

Jet : la position de l'avion.

Choix d'implémentation :

Nous avons donc fait 3 classes pour notre jeu une pour les avions(Jet), une pour le fonctionnement du jeu(BattleJetMain) et une pour l'interface du jeu(Board).

Pour la classe BattleJetMain :

- La méthode « `private Parent createContent()` » sert à la création de la fenêtre ainsi qu'aux dimensions des plateformes du jeu. Elle sert également au fonctionnement du jeu c'est-à-dire que c'est elle qui contient toutes les règles.
- La méthode « `private void enemyMove()` » gère le jeu de l'ennemi. C'est-à-dire que c'est elle qui gère le fait que l'ennemi sait quand il peut tirer ou pas.
- La méthode « `public void startGame` » place les jets de l'ennemi.
- La méthode « `public void start(Stage primaryStage) throws Exception` » sert lors du lancement du jeu comme par exemple c'est elle qui fait que la fenetre à le nom de BattleJet.
- La méthode « `public static void main(String[] args)` » sert à lancer l'application.

Pour la classe Jet :

- La méthode « `public Jet(int type, boolean vertical)` » gère la vie des bateaux.
- La méthode « `public void hit` », lorsqu'un avion est touché il perd une vie.
- La méthode « `public boolean is alive` » sait quand un jet est abattu complètement ou si il lui reste de la vie.

Pour la classe Board :

- La méthode « `public Board(boolean enemy, EventHandler<? super MouseEvent> handler)` » crée les plateformes (10 carrés sur 10 carrés), ainsi que les cliques de la souris pour le positionnement des Jets et va placer les Jets dans les plateformes.
- La méthode « `public boolean placeJet(Jet jet, int x, int y)` » s'occupe du placement des Jets ainsi que de colorer les cases déjà utilisées ou pas.
- La méthode « `private Cell[] getNeighbors(int x, int y)` » va placer les points (cases) utilisés dans une liste.
- La méthode « `private boolean canPlaceJet(Jet jet, int x, int y)` » s'occupe de savoir si on peut placer les jets à cet endroit ou pas.
- Les méthodes « `private boolean isValidPoint()` » gèrent la validation des points(cases).
- La méthode « `public class Cell extends rectangle` » va s'occuper de savoir si une cases est touchée ou pas.
- La méthode « `public boolean shoot()` » sert à la colorisation des cases selon leurs états(bleu = pas touché, rouge = touché).

Les difficultés rencontrées :

Nous avons rencontré de nombreuses difficultés mais celle qu'on tient à mettre en évidence c'est l'organisation et la gestion du temps car comme vous le voyez notre jeu ne ressemble pas à grand-chose et il ne comporte tout ce qu'on attendait et ce que vous avez demandé de faire. Et cela en partie à cause du temps que nous avons très mal géré.

Une autre difficulté a été de créer le mode de jeu aléatoire car nous partions de 0 niveau compétences en Java.

L'interface console a aussi été un très grand défi pour nous.

Les améliorations :

-Nous aimerions pouvoir créer une interface graphique d'accueil permettant de choisir le mode de jeu ainsi que de quitter le jeu et bien sûr le lancer.

-Créer un mode de jeu contre un autre pc comme dans nos attentes.

-Ajouter des images aux cases et non plus des simples couleurs.

- Pourquoi ne pas en faire une application sur Android et/ou Iphone.

Les conclusions :

-Faysal «Pour un premier projet en Java, j'ai apprécié le fait de devoir créer un jeu. Grâce à cela, j'ai pu apprendre des nouvelles techniques pour coder.»

-Clément « Je ne suis pas très content de notre projet car le rendu final n'a rien à voir avec nos attentes de départ. Par contre je suis très content d'avoir fait ça en groupe car j'ai adoré travailler en équipe et j'espère avoir à refaire des projets avec cette équipe. Et aussi ce travail nous a permis d'améliorer nos compétences et peut-être d'apprendre des techniques que l'on ne verra jamais en cours. »

-Damien « Ce projet est très compliqué, car on part de 0 comme compétence, les codes ne tombent pas encore du ciel malheureusement, donc il faut travailler sans arrêt bosser pour écrire ces lignes de code. Mais cependant je suis fier de mon groupe car nous n'avons jamais abandonné et on s'est toujours aidé. »