

# Compte-rendu du Tron

Quentin SAUVAGE

15 Février 2017

# Chapitre 1

## Fonctionnement du programme

Le jeu demande tout d'abord au joueur de choisir la taille du plateau, entre 5 et 30. Après avoir validé la taille choisie, le jeu génère le plateau puis place les deux joueurs dessus. Tant que le joueur n'a pas appuyé sur une touche, aucun joueur ne peut se déplacer. Lorsque que le joueur appuie sur une touche, il se déplace dans la direction choisie si la touche est une flèche directionnelle, ou vers l'ouest si ce n'est pas le cas.

À chaque tour, le jeu vérifie que les joueurs peuvent bien se déplacer dans la direction voulue. Si ce n'est pas le cas, ils sont forcés de choisir une autre direction, et si aucune des directions n'est valable, alors le joueur en question meurt. Un joueur ne peut pas se déplacer si il est sur le point d'entrer en contact avec une case de sa couleur ou de la couleur adverse, ou si la case est en dehors du plateau (dans ce dernier cas le jeu est mis en pause et attend que le joueur change de direction).

Une fois qu'un joueur meurt, le joueur adverse marque un point. Ainsi, si les deux joueurs meurent en même temps, ils gagnent tous les deux la manche. Le jeu se termine dès qu'un joueur remporte trois manches, et il demande alors au joueur s'il veut recommencer ou s'il veut quitter. Si le joueur choisit de recommencer, le jeu lui demande alors de choisir à nouveau la taille du plateau.

## Chapitre 2

# Structure du programme

Voici le rôle de chacune des classes du programme :

- Case permet de gérer la position des joueurs et leurs déplacements.
- Constantes était à la base une interface. Je l'ai transformée en classe abstraite afin de pouvoir manipuler ses attributs (nbColonnes, nbLignes et nbPixels).
- DemoJeu permet simplement de créer le jeu. -Direction est une énumération représentant les quatre directions pouvant être choisies par les joueurs.
- Fenetre : une JFrame qui contient tous les éléments graphiques du jeu.
- Jeu contient toutes les données relatives au jeu(joueurs, fenêtre, etc).
- Panneau : un JPanel qui dessine le plateau de jeu.
- PanneauChoix : un JPanel qui demande au joueur la taille du plateau et notifie le jeu de ce choix.
- PanneauFin : un JPanel qui demande au joueur s'il veut refaire une partie.
- PanneauManches : un JPanel qui permet d'afficher à côté du plateau le nombres de manches remportées par chaque joueur.
- Personnage définit les méthodes et attributs communs à l'ordinateur et au joueur humain.
- PersonnageHumain : un Personnage qui contient la méthode permettant de vérifier que le joueur peut changer de direction.
- PersonnageIA : un Personnage qui contient l'algorithme calculant la trajectoire de l'ordinateur.
- SlidePlateau est un JSlider graphiquement modifié.

## Chapitre 3

# Fonctionnement de l'IA

### 3.1 Premier fonctionnement de l'IA

Afin de pouvoir rapidement simuler une partie, j'ai d'abord donné à l'ordinateur un comportement entièrement aléatoire. Celui-ci avait une certaine probabilité à chaque tour de changer de direction, peu importe les cases environnantes.

### 3.2 L'IA actuelle

L'IA actuelle a une chance sur deux de changer de direction à chaque tour. Elle vérifie alors si la nouvelle direction est adoptable. Si elle ne l'est pas, elle prend la suivante (dans le sens des aiguilles), jusqu'à ce qu'elle puisse avancer. Avant de choisir une direction, elle met à jour son état à "mort" si aucune direction n'est viable. Elle avance ainsi dans tous les cas.

### 3.3 Essais

J'ai essayé de faire en sorte que l'IA se déplace à chaque tour dans la direction qui lui offrait le plus de cases libres, sans tenir compte du joueur. Ce choix était néanmoins bugué, puisque l'IA avait tendance à partir dans une direction jusqu'à toucher un mur, passer sur la colonne ou ligne adjacente, et retraverser le plateau. J'ai donc abandonné cette idée.