



ENSTA Bretagne

Projet informatique

Rapport intermédiaire

Romain DUSSOT et Thomas LE MASSON

24 mars 2017

Introduction

Le sujet reprend le principe du jeu RoboRally, par le créateur du jeu de cartes Magic sorti en 1994 chez Wizard of the Coast. Ce jeu se joue originellement de 2 à 8 joueurs. Dans notre cas le nombre de joueurs sera limité pour le moment. Les règles se trouvent aussi légèrement modifiées. Nous allons commencer par parler rapidement du jeu de société pour expliquer les règles de bases, ensuite nous détaillerons nos choix de conception.

Chapitre 1

Le jeu RoboRally, les règles

1.1 Le but du jeu

Le but du jeu est de rempoter la course définie sur le plateau. Pour ce faire, plusieurs moyens s'offrent à vous, comme :

- faire le trajet prédéterminé en passant par tous les checkpoints puis en finissant sur l'arrivée ;
- détruire vos adversaires en utilisant le plateau de jeu et ses fonctionnalités tels que les lasers.

1.2 Les cartes, les points de vie

Un joueur dispose de 3 vies, donc son robot peut mourir 3 fois. Ce robot a 9 points de vies par vie. Ce qu'il lui permet d'encaisser 9 points de dégâts par vie. Ces points de vie représentent également le nombre de cartes qu'il peut récupérer par tour. Le joueur récupère donc 9 cartes par tour de programmation.

1.3 Le déroulement

1.3.1 La phase d'initialisation

On choisit un parcours de jeu. Chaque joueur choisit un robot qui sera placé sur le plateau. Chaque joueur dispose de 3 vies et de 9 points par vie.

1.3.2 Les tours

Un tour se décompose en 5 séquences immuables :

- On distribue les cartes programmes
- On choisit les 5 cartes de programme de ce tour
- On déclare ou si l'on se place hors tension pour ce tour (possibilité expliquée juste après).
- On joue dans l'ordre les cinq phases de programmation : exécution Programme, activation éléments de plateau, résolution les interactions, enregistrement drapeau, réparation.
- On prépare le tour suivant

1.3.3 La fin

Le jeu se finit par plusieurs cas de figures :

- l'arrêt d'un des joueurs sur la case d'arrivée en étant passé par tous les checkpoints ;
- la destruction et la perte de tous les points de vie de tous les joueurs sauf un.

1.4 Le plateau

Le plateau dispose de plusieurs cases avec des fonctions bien précises.



Figure 1.1 – Plateau de jeu

1.4.1 Les cases simples

Les cases simples n'ont pas d'action sur les robots présents sur le plateau. Celles-ci sont construites à partir du sol et des murs qui y sont liés. On trouve donc 4 murs possibles sur la case, qui empêchent les robots de traverser et qui peuvent permettre de stopper sa course. La direction ou l'emplacement des robots ne sont modifiés par ces cases.

1.4.2 Les cases munies de laser

Sur certaines cases, on va pouvoir trouver des lasers qui vont infliger 1 point de dégâts au robot sur sa route. Les lasers ne peuvent être arrêté que par un mur. Ceux-ci se manifestent à chaque étapes de tour.

1.4.3 Les cases agissant sur les robots

Les giratoires Ces cases font tourner les robots à chaque étape. Elles modifient la situation actuelle. Le sens de rotation est défini selon le type de giratoire (à droite ou à gauche). Le graphisme de ses cases permet de voir directement le sens de rotation assigné à la case.

Les tapis roulants Les tapis roulants peuvent ou non agir sur la direction principale du robot, mais leur principal action est de faire translater le robot de 1 ou 2 cases selon le tapis. Comme les giratoires, ces cases agissent à toutes les étapes.

Jouer avec ses éléments Ces éléments étant assez complexe à utiliser car il modifie la situation. Nous n'en prendrons pas compte au début de notre projet, nous nous intéresserons au fait de les ajouter plus tard.

Chapitre 2

Notre travail

2.1 Définition des classes

Dans un premier temps, nous avons choisi de faire la distinction entre les cartes, les robots et les joueurs, en les définissant dans des classes différentes plus simple à créer. L'objectif étant d'utiliser des fonctions pour les faire interagir entre elles lors du déroulement du jeu. Nous avons donc défini trois classes principales : les cartes, les robots et les joueurs.

2.1.1 la classe "Carte"

Il existe dans ce jeu quatre types de cartes : « avancer », « reculer », « tourner » et « demi-tour ». On peut avancer de une, deux ou trois cases. On peut reculer d'une case. On peut tourner d'un quart de tour vers la gauche ou vers la droite. On peut faire demi-tour.

Nous avons donc défini la classe « carte » avec quatre classes héritières.

Le constructeur de la classe « carte » est défini par quatre items :

- La priorité ;
- Le nombre de rotation ;
- Le nombre de case ;
- Le numéro du joueur à laquelle elle appartient.

Chacun des types de cartes est donc caractérisé par ces quatre items.

Les cartes de types « avancer » et « reculer » ont un nombre de rotation égale à zéro, les cartes « avancer » ont un nombre de case variant de 1 à 3 et la carte « reculer » a un nombre de case égale à -1. Les cartes « tourner » ont un nombre de case égale à zéro, un nombre de rotation égale à 1 indique une rotation vers la droite et un nombre de rotation égale à -1 une rotation vers la gauche. La carte « demi-tour » a un nombre de case nul et un nombre de rotation de 2.

2.1.2 la classe "Robot"

Chaque joueur dispose d'un robot qu'il déplace à l'aide de ses cartes vers la ligne d'arrivée. Nous avons choisi de faire la distinction entre un joueur et son robot pour simplifier le programme et éviter de créer une classe trop complexe.

Le constructeur de la classe « robot » est défini par quatre items :

- La vie ;
- La direction de départ ;
- Les coordonnées ;
- Le plateau : nous avons choisi de lier le robot au plateau afin de faciliter les mouvements de celui-ci au cours de la partie.

Le robot peut avancer, reculer et tourner. Nous avons décidé de créer des fonctions, au sein de la classe « robot », qui ont pour objectif de modifier les caractéristiques du robot en fonction de la carte jouée.

On a alors une fonction « Turn », qui modifie la direction du robot en prenant en entrée le nombre de rotations de la carte jouée. On a également une fonction « Forward », qui en prend en compte la direction du robot et fait évoluer les coordonnées du robot en fonction du nombre de case de la carte jouée.

2.1.3 la classe "Joueur"

Chaque joueur a autant de carte dans sa main que de point de vie et est lié à son robot. Au début de la partie le joueur choisit un robot qui est placé aléatoirement sur le plateau de jeu.

Pour s'assurer que le vainqueur ait bien respecté les conditions de victoire en passant dans l'ordre par les différents check-points, nous avons créé dans le constructeur de la classe « joueur » un item « fil rouge » qui prend en compte cet aspect du jeu.

Le constructeur de la classe « joueur » est donc défini par quatre items :

- Le nombre dans la main du joueur ;
- Le nombre de cartes jouées ;
- Le robot ;
- Le fil rouge, qui est une liste de 0/1 si les conditions de victoire sont vérifiées et déclenche la fin de la partie.

La classe « joueur » est également composée de deux sous classes : Utilisateur et IA, que nous n'avons pas encore définies.

```

*#####'
$!!!!!!!!!!&
$!!!!!!!!!!&
$!!!!!!!!!!&
$!!!!!!!!!!&
$!!!!!!!!!!&
$!!!!!!!!!!&
$!!!!!!!!!!&
$!!!!!!!!!!&
)%%%%%%%%%(

```

Figure 2.1 – Comment écrire un plateau en format texte

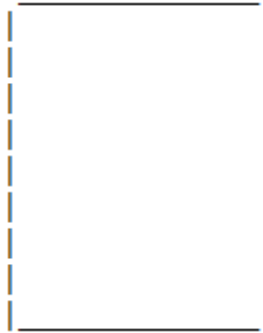


Figure 2.2 – Aperçu du plateau généré

2.1.4 Le plateau

Pour définir le plateau de manière la plus simple pour nous. Nous avons décidé d'utiliser un dictionnaire personnalisé qui nous permet de répertorier toutes les cases existantes et de leur attribuer un caractère. Ensuite, grâce à la lecture de fichier texte incluse dans python, nous allons seulement lire notre fichier texte. Un exemple sera plus parlant :

2.2 Diagramme des classes récapitulatif

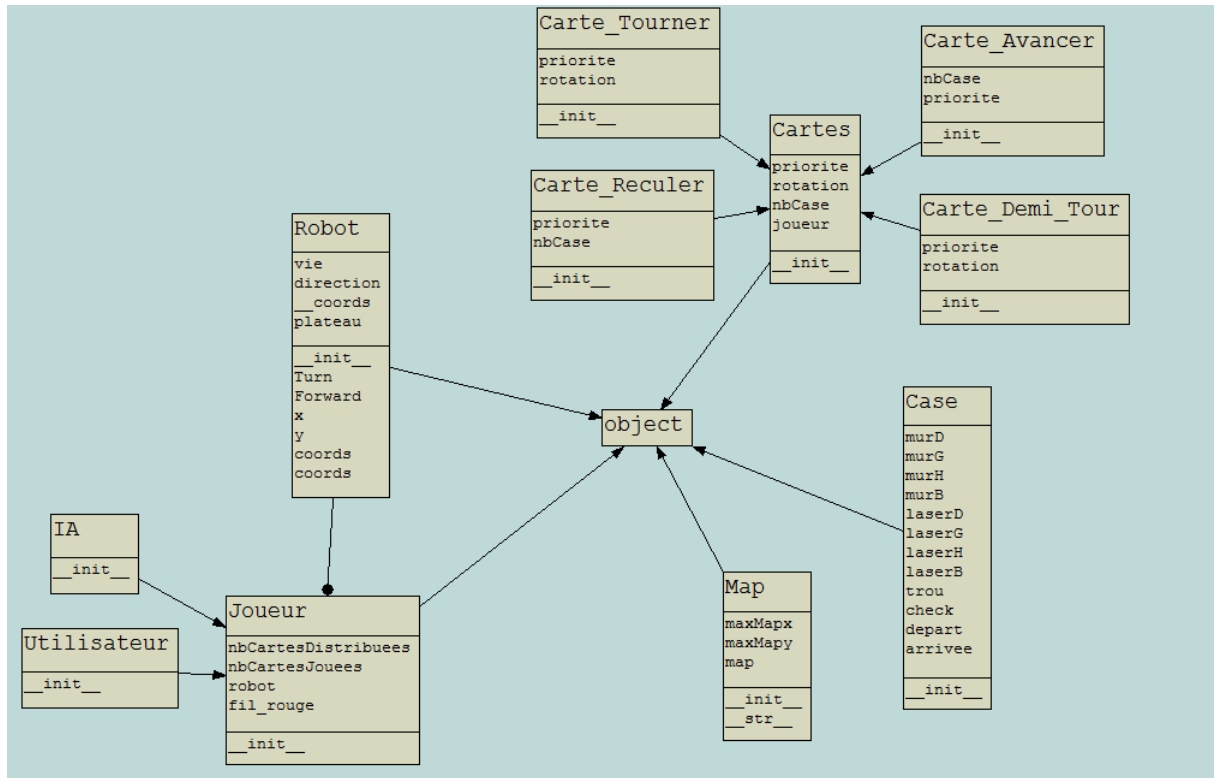


Figure 2.3 – Diagramme des classes

2.3 Déroulement du jeu en vue de la programmation

2.3.1 Initialisation

La fonction initialisation remplit ce rôle en prenant en entrée le nombre d'IA et le nombre de joueur. Elle attribue à chaque joueur et à chaque IA son numéro de joueur et crée deux listes « Bot » et « Joueur » où elle les range. Elle charge également le plateau.

2.3.2 Les tours

La phase de distribution est décrite par une fonction « Distribution ». Cette fonction attribue aléatoirement à chaque joueur autant de carte que son nombre de point de vie et crée la relation d'appartenance d'une carte à

son joueur. Les cartes distribuées sont rangées dans des listes qui représentent les mains de chacun des joueurs.

La phase de choix est une phase d'interaction où l'utilisateur va entrer une liste ordonnée des 5 cartes qu'il veut jouer lors de ce tour. Nous n'avons pas encore défini la fonction liée à l'IA qui sera responsable du choix des cartes.

La phase d'application est composée de deux fonctions : une fonction de tri, qui trie une liste par ordre de priorité de chaque carte. Ensuite la fonction « Application des cartes prend la première carte de chaque joueur pour les mettre dans une liste, qui est ensuite triée par la fonction de tri. La liste triée est parcourue et toutes les cartes sont appliquées une à une. Cette séquence est réalisée dans une boucle qui est réalisée 5 fois.