

24/03/2017

Course de robots

U.V 2.1 Projet Python

Partie I

Alexandre Chojnacki
Alexandre Corazza

Sommaire

Table des matières

Sommaire.....	2
Analyse du sujet.....	3
Diagramme des classes.....	4
Description des méthodes importantes.....	5
Description des tests et des résultats.....	5
Conclusion.....	5

Analyse du sujet

Des robots s'affrontent sur un plateau. Chaque joueur choisi 5 cartes parmi 9 distribuées qui lui permettront de programmer son robot. Le robot se déplace en suivant les instructions données. Le premier à atteindre la case d'arrivée remporte la partie.

Des cases ayant des effets sur les robots sont présentes sur le plateau.

Certaines modifient les points de vie du robot:

La case trou détruit le robot.

La case réparation redonne des points de vie au robot.

Des lasers peuvent endommager le robot.

Chaque robot possède initialement 9 points de vies. Il en perd 1 à chaque fois qu'il subit des dégats.

Si le robot descend en dessous de 5 points de vies, le joueur pourra choisir uniquement autant de cartes qu'il a de points de vie.

D'autres cases modifient la position ou l'orientation du robot:

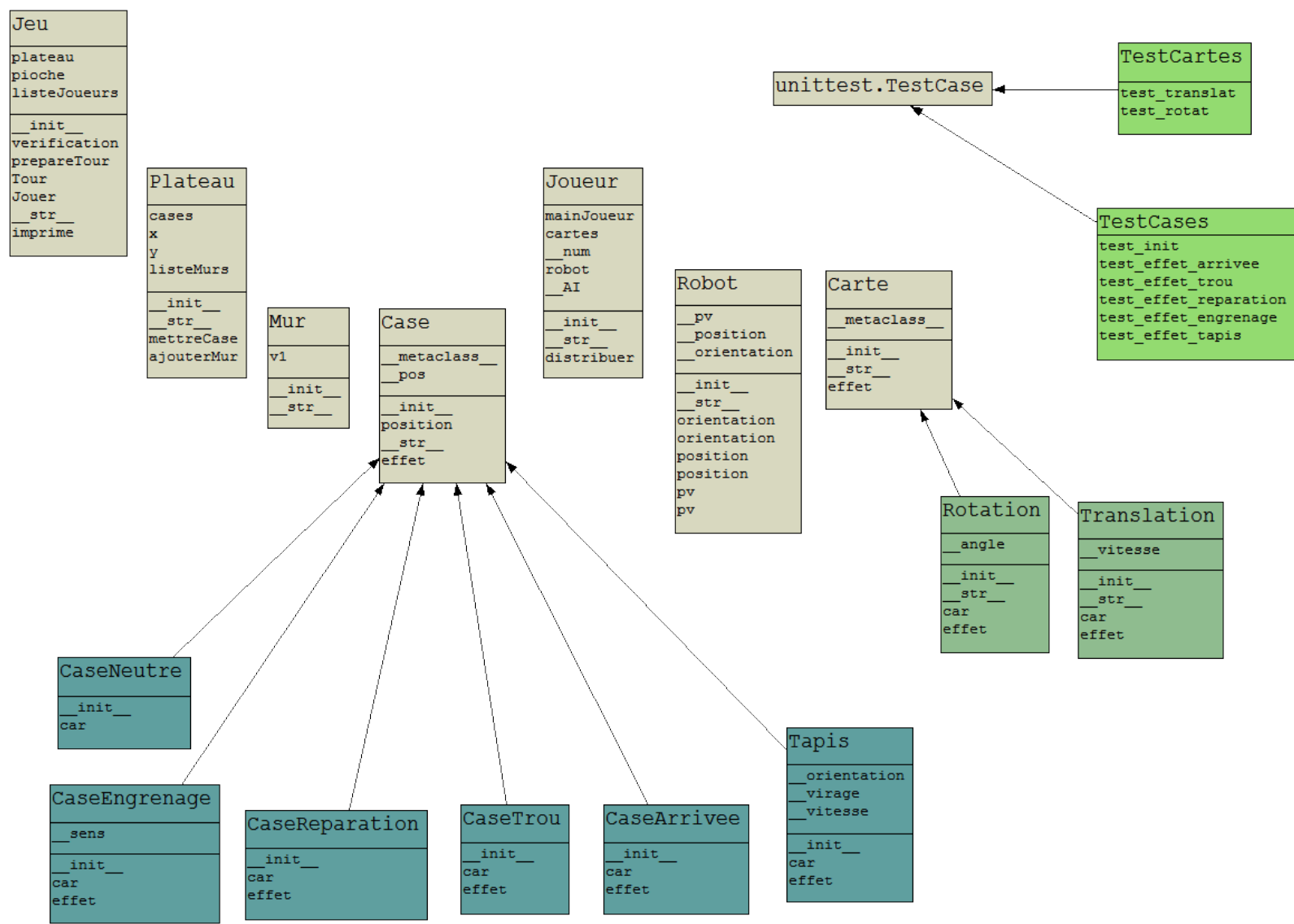
Un tapis roulant translate le robot et peut également le faire pivoter lors d'un virage

La case engrenage fait pivoter d'un quart de tour le robot vers la droite ou la gauche

Les murs ne peuvent pas être traversés, le robot reste sur place si il tente d'avancer sur un mur.

Diagramme des classes

Ces différents objets nous ont permis dans un premier temps d’organiser les différentes classes.



Description des méthodes importantes

Le constructeur de la classe Jeu récupère le plateau et les cartes avec lesquels la partie sera jouée, et lance la création de tous les joueurs par la méthode `creerJoueur`, qui instancie le robot et les variables du joueur.

La méthode “jouer” fait tourner le jeu jusqu’à obtention d’un vainqueur via les méthodes “preparetour” et “tour”.

“preparetour” permet au joueurs et aux futures IA de choisir leur cartes.

“tour” exécute le code des robots sans interruption.

Les méthodes d’effet des classes Cases et Cartes sont des méthodes clefs pour la partie réflexion du la partie.

Description des tests et des résultats

Nous avons effectué des tests sur les fonctions élémentaires simples à vérifier, très nombreuses et essentielles au bon déroulement du programme. Ainsi le débogage restant n’est à effectuer que sur quelques fonctions principales, plus compliquées à valider à l’aide de tests unitaires.

Par exemple si on teste le tableau final obtenu par la méthode informatique `Jeu.jouer` en le comparant au tableau calculé à la main, les dysfonctionnements du programme n’apparaissent pas clairement.

Conclusion

Ce projet nous a permis de nous sensibiliser sur l’importance des tests unitaires et du travail en amont sur la création des classes: on ne peut pas coder au fur et à mesure, il est important de commencer par une analyse du sujet en passant par un diagramme des classes.