

Module 1106 : PROJET

ALGORITHMIQUE-PROGRAMMATION

RICOCHETS ROBOTS

Serge Stinckwich, IUT CAEN – Département Informatique (Version 1.2)

14 novembre 2013

1 Description du projet

On se propose dans un premier temps de réaliser une variante du jeu **Ricochets Robots** : https://fr.wikipedia.org/wiki/Ricochet_Robots. La variante que l'on veut réaliser se joue seul.

Le jeu se joue sur une grille fermée de 16 par 16 cellules avec 4 robots de couleurs différentes (rouge, vert, bleu, gris), un ensemble d'obstacles (murs) et un objectif qui se trouve dans une cellule de la grille.

Pour déplacer un robot, le joueur choisit une direction (Nord, Sud, Est, Ouest) et le robot se déplace en ligne droite jusqu'à rencontrer un obstacle ou un robot. Le but du jeu est de déplacer chacun à leur tour les 4 robots afin d'atteindre l'objectif en s'aidant éventuellement des autres robots. Une fois la mission des 4 robots réalisée, on déterminera le score du joueur par le nombre de déplacements nécessaires.

Les grilles seront construites de manière aléatoire, éditées par un utilisateur ou bien prédéfinies. Trois grilles prédéfinies seront proposées par votre programme. On trouvera des exemples de grilles de jeu ici : <http://jeux.prise2tete.fr/p2t-ricochet/p2t-ricochet.php5>.

Les positions initiales des robots seront tirées aléatoirement pour une grille aléatoire, ou bien choisies si il s'agit d'une grille éditée par l'utilisateur ou encore déterminées à priori lors de grilles prédéfinies.

Une fois que la version solitaire fonctionne, on réalisera une version multi-joueurs à partir de la description du jeu ici : https://fr.wikipedia.org/wiki/Ricochet_Robots. Le programme final devra permettre de jouer à ces deux versions. Dans la version multi-joueurs, les joueurs utilisent 4 robots et un objectif. Les joueurs jouent simultanément en disant quel est le nombre de déplacements qu'ils envisagent. A la fin du jeu, on compte le nombre d'objectifs de chacun des joueurs pour déterminer le gagnant.

Votre programme devra permettre de visualiser une séquence de jeu en affichant à chaque tour : le plateau de jeu, qui doit jouer (dans le cas de la version à plusieurs joueurs), le score courant du joueur (ou des joueurs).

Pour ceux qui veulent aller plus loin, vous pouvez réaliser au choix : des graphiques avec animation, une version en 3D, une version pour jouer à plusieurs en réseau, une version pour jouer à plusieurs sur Internet.

2 Notes de mise en œuvre

Dans un premier temps, définissez les principales structures de données dont vous allez avoir besoin pour représenter les éléments du jeu : plateau de jeu, informations sur les joueurs et gestionnaire de jeu (contenant les informations globales concernant l'état du jeu courant). Pour chaque structure de données, définir les données correspondantes et les fonctions associées.

Dans un deuxième temps, définir la boucle de contrôle globale du jeu qui assurent les différentes phases du jeu.

Pour les besoins du jeu, il est demandé de réaliser un joueur aléatoire qui joue aléatoirement.

Il est conseillé d'avoir en permanence un programme qui compile et qui définit une version partielle du jeu.

Seul devra être réalisée une version textuelle de ce jeu. Le programme devra permettre de visualiser la partie et de gérer un système de high-score.

Si vous souhaitez développer une version graphique de votre programme, il est préférable dans un premier temps de définir une version non graphique afin de bien séparer les fonctions associées au jeu et l'affichage graphique du jeu.

3 Travail à rendre

Le projet est à réaliser en binômes (ou, avec l'accord de votre chargé de TP, en monôme). Le travail à rendre est un projet sous forme d'une archive zip à **déposer sur la plateforme Filex à l'adresse suivante : <http://ent.unr-runn.fr>**.

La procédure pour déposer votre projet est la suivante :

- sélectionner Université de Caen Basse Normandie puis renseigner les identifiant et mot de passe de votre messagerie universitaire.
- cliquer sur le lien Dépôt de fichiers. Télécharger depuis votre ordinateur le fichier à transmettre dans le champ Parcourir puis Envoyer.
- l'envoi du document génère une fenêtre avec un lien direct vers celui-ci. Copier ce lien pour l'insérer dans un mail adressé à votre enseignant chargé de TP.

Le nom de l'archive doit avoir la forme suivante : `Nom1Nom2.grTP.zip` ou `Nom1.grTP.zip` où `Nom1` et `Nom2` sont les noms de famille des membres des polynômes et `grTP` est le nom du groupe de TP auquel ils appartiennent (1.1, 1.2, etc.). **Note** : Lors des séances de TP, les enseignants chargés de TP suivront l'avancement de votre projet et pourront vous aider sur certains points difficiles.

Ce qu'il faut rendre :

- Le code source complet de votre application en C largement commenté.
 - Un exécutable testé **sous Linux** et opérationnel avec sa documentation d'installation et d'utilisation. **ATTENTION** : si votre programme fait appel à d'autres bibliothèques externes (comme par exemple la bibliothèque SDL), il est impératif de les inclure pour les besoins de test.
 - Un court rapport d'une longueur comprise entre 3 et 10 pages présentant :
 - les fonctionnalités implémentées (très brièvement).
 - organisation du programme : découpage en fonctions, rôle de ces fonctions, explications du programme.
 - l'organisation et la répartition des tâches au sein du binôme durant la durée du projet (brièvement).
 - bilan qualitatif du travail, difficultés rencontrées, points qui vous ont paru intéressants.
 - un mode d'emploi avec quelques illustrations (p. ex. capture d'écrans, scénario d'exécution...), destinées à montrer l'opérationnalité de votre application.
 - Une conclusion sur l'apport (ou non) du projet en termes technique, scientifique, humain.
- Le code source ne doit pas faire partie du rapport d'une dizaine de pages (sinon en annexe).**

4 Évaluation du projet et calendrier

Le projet est à rendre le **10 janvier 2014 avant minuit** (tout retard conduira à des pénalités). L'évaluation sera réalisée en fonction des critères ci-dessous :

- *qualité technique du code* : découpage en fonctions, modularité et réutilisabilité, instructions, algorithmes, efficacité, gestion des erreurs lors de saisies, ou lorsque les données fournies en paramètres des fonctions sont incorrectes.
- *lisibilité du code* : présentation du programme (indentation), usage de variables et de fonctions ayant des noms explicites, commentaires pour préciser les points difficiles dans les algorithmes, paramètres des fonctions . . .
- *documentation fournie* : organisation du programme et son mode d'emploi, bilan.
- *présentation orale* : démonstration du programme et questions sur le travail réalisée.