**PC2R – Rapport de projet**

**Ricochet Robots**

**1. Description du jeu**

Dans notre version du jeu de plateau Ricochet Robots (originellement Rasende Robot), le jeu est composé d’une matrice 16x16 cases et de 4 pions robots et d’une case cible, initialement placés selon une énigme donnée. Les cases du plateau peuvent être vides ou contenir des murs.  
Le but du jeu est de faire arriver le robot qui a été désigné dans l’énigme, à la case cible. Pour cela le joueur doit s’aider des robots et des murs. Les robots ne peuvent se déplacer qu’en ligne droite, vers le haut, la gauche, le bas et la droite et ne s’arrêtent qu’au moment où ils rencontrent un obstacle (un mur ou un autre robot).

Le jeu se déroule en sessions. Chaque session est composée en un nombre indéfini de tours.  
Chaque tour se décompose en 3 phases :   
- Une phase de réflexion : dans laquelle les joueurs ont 5 minutes pour proposer le nombre de coup d’une solution à partir d’un plateau et d’une énigme.   
- Une phase d’enchère : qui démarre à la fin des 5 minutes de réflexion ou lorsqu’un joueur trouve une solution, dans laquelle les autres joueurs peuvent enchérir en proposant un nombre de coups différent.  
- Une phase de résolution : dans laquelle les joueurs, en fonction du nombre de coups qu’ils ont proposé, ont un à un 1 minute pour proposer leur solution, tant qu’une solution n’a pas été validée par le serveur. Si une solution est correcte, la phase de résolution se termine et le joueur ayant proposé cette dernière gagne 1 point.

Le jeu étant multi-joueurs, le but est d’arriver à un score objectif de 10 avant tous les autres. Une fois le score objectif atteint, une nouvelle session se lance.

ICI INSERER SCREEN D’UN PLATEAU AVEC ENIGME

**2. Serveur**

Le serveur a été réalisé en C. Il utilise principalement les bibliothèques Sys/Socket, Pthread et String. Nous détaillerons les choix d’implémentation qui ont été faits pour le code du serveur qui est réparti en 5 fichiers C, accompagné de 4 fichiers entête.

**2.1 Etablissement du serveur**

L’établissement du serveur se fait grâce à la bibliothèque Sys/Socket. Le serveur utilisé est un serveur de type connecté, dont l’adresse est l’adresse locale, écoutant sur le port 2016 et dont le nombre de connexion correspond à SOMAXCONN, une variable déterminant le nombre maximum de connexions possibles, définie dans la bibliothèque. La socket associée au serveur est réutilisable, même en cas d’utilisation récente (ce qui est très utile en phase de tests).

**2.X Utilisation**

Pour utiliser le serveur, il suffit de compiler grâce au fichier Makefile, puis de lancer l’exécutable **main** qui s’est créé dans le dossier **bin/**.