

Système et réseaux : Compte Rendu de Projet

Valentin VERSTRACTE & Evan PETIT

L3 — November 17, 2021

1 Structure générale de l'application

La structure générale de l'application est découpé en 3 programmes shell. GestionJeu, JoueurHumain, JoueurRobot. JoueurHumain lancera aux besoins des subshell pour résoudre des problèmes de pipes, entrées utilisateur bloquantes

2 La communication

La communication entre les différents shell s'effectue à travers de pipe pour transmettre des messages données. On utilisera des fichiers avec l'extension tmp pour synchroniser des données aux besoins.

2.1 Les pipes

Les processus utilisent les pipes pour constamment lire ou envoyer des messages. La mise en place "d'une api" facilitera cette communication. Un message est constitué de la forme suivante :

- La première lettre est un chiffre qui décrit une action
- La deuxième lettre est un point virgule
- Le reste est un message pouvant être afficher dans la console

Exemple : gestionJeu envoie à tout les joueurs "3;Félicitations, le tour n°\$ROUND' est terminé, on passe au tour suivant". Ainsi 3 décrit que le tour est terminé et envoie un message avec le numéro du tour que l'on peut afficher.

2.2 Les fichiers tmp

On utilise les fichiers tmp pour stocker les valeurs de certaines variables et ainsi résoudre un problème de synchronisation. En effet, lire une pipe et lire une entrée utilisateur sont deux actions bloquantes (qui bloquent l'exécution du code). On se retrouve donc bloquer pour JoueurHumain qui doit lire des entrées utilisateur et recevoir des données / messages via le pipe. Pour résoudre ce détail JoueurHumain va lancer l'écouteur des pipes en fond et l'entrée utilisateur "en avant". Le problème est le suivant : les pipes et la lecture utilisateur utilisent des variables en commun. Le problème est que les pipes sont lancés en fond (dans un subshell) et que les variables ne seront pas synchroniser malgré le même nom que celle des entrées utilisateur. Ainsi on enregistre / modifie au besoin la valeur des mêmes variables dans un fichier tmp pour chaque joueur et variable.

3 La gestion du temps

La gestion du temps n'est pas réellement existante, chaque processus réagit aux pipes et (pour le joueurHumain) à l'entrée clavier. On effectue des actions en conséquences.