

ProjetOS

BOUTON Nicolas

April 2019

1 Introduction

Le but du projet est de créer une bataille navale **équipe vs équipe**. Chaque équipe à un nombre donné de joueur qui ont de la coque et du kerosen. Si un joueur n'a plus de coque ou de kerosen il meurt. Si tous les joueurs d'une équipe sont mort, alors l' équipe n'est plus dans la partie.

En plus de créer ce jeu, il faut utilisé les différents **appels systèmes** qu' on a vu en cours :

- la gestion des fichiers d'entrée/sortie ;
- la gestion des processus ;
- la gestion des tubes.

Une bibliothèque pour gérer la carte nous a été fournis.

2 Structure

2.1 Structure fichier

Permet d'ouvrir et lire le fichier en argument

2.2 Structure info

Contient toutes les information que le serveur a besoin pour gérer les bateau.

2.3 Structure message

Permet au serveur et aux bateau de communiquer des information. Dans les tubes j'écris les variables de la structure car j'ai pas réussi à écrire l'adresse.

2.4 Structure equipe

Permet à l'équipe de communiquer la position des bateaux trouvés.

2.5 Structure liste

Cette structure sert uniquement pour la catégorie déplacement et contient l'id des bateaux qu'ils veulent se déplacer et la position où ils veulent aller.

2.6 Structure serveur

Envoie les information aux threads qui s'occupent des action.

3 Fichier

Je stock les 256 premier caractère dans une chaine de caractère. Rien de spécial à dir sur ce fichier.

4 Main

Le fichier main.c sert à :

- initialiser le jeu (carte, tube, etc ...),
- créer les équipes en créant des processus,
- gérer serveur.

5 Serveur

Le serveur une fois après avoir créer les équipes avec les tubes envoie les information aux bateau, attends de recevoir les informations de chaque bateaux pour le prochain tour. Ensuite il procède a l' exécution des actions demandés par les bateaux.

Si un bateau n'a plus de **kerosen** ou n'a plus de **coque** le serveur met à jour la variable **enVie** et la variable **monAction** de la structure **message**.

A chaque traitement d'une catégorie et d'un tour, il vérifie les bateaux qui sont en vie.

Un fois que le nombre max de tours est atteinds ou qu' il ne reste plus qu'une équipe en vie, il envoie un message aux bateau pour qu'ils arrêtent leurs thread et donc les équipes termines dans leurs processus et le programme fini.

6 Equipe

Créer les threads qui seront les bateaux. Ils peuvent communiquer la position de l'ennemie qu'ils ont trouvés via une variable critique. Si un bateaux est mort

ils continue a envoyer son message aux serveur met son action n'est pas traité par le serveur. A chaque tour les bateaux si un bateau n'a pas de cible il regarde si ses alliés en ont une et la prend.

Algo du bateau : si il est mort il fait rien, sinon si sa coque est j 30 et qu' il a assez de kerosen alors **répare**, sinon si il n'a pas de position d'ennemie ou radar trop vieux **radar**, sinon si il peut faire **BST** alors il le fait, sinon si il peut **ATK** il le fait, sinon **MOV**.

7 Action

Ma fonction BST et SCN ont un problème mais je ne sais pas pourquoi.

8 Info supplémentaire

- le fichier script permet de compiler le programme (avec ./script)
- ./SoD fichier pour excécuter le programme
- J'ai un peu commenté les .h