Réalisation du «net code» d'un jeu vidéo simple
Partie 1 : un premier serveur

Projet de programmation système (R3.05), année 2022-2023

loig.jezequel@univ-nantes.fr

1 Étude du code fourni

Avant de pouvoir commencer à travailler sur le projet, il va vous falloir comprendre, au moins dans les grandes lignes, le code qui vous est fourni. Ce code repose en grande partie sur la bibliothèque Ebiten, que vous avez déjà eu l'occasion d'utiliser l'an dernier.

Dans l'archive on trouve les fichiers suivants :

main.go qui contient simplement la fonction main, chargée de démarrer le jeu.

game.go qui contient la structure de données représentant le jeu ainsi qu'une fonction `initgame` permettant d'initialiser cette structure de données.

game-layout.go qui permet de définir la taille de la fenêtre.

game-update.go qui contient les fonctions appelées à chaque itération de la boucle principale du jeu (pour rappel, Ebiten appelle automatiquement la fonction `Update` 60 fois par seconde).

game-draw.go qui contient les fonctions permettant d'afficher le jeu à l'écran (pour rappel, Ebiten appelle automatiquement la fonction `Draw` jusqu'à 60 fois par seconde).

field.go qui contient des types et fonctions en rapport avec le terrain sur lequel les personnages courent.

runner.go qui contient des types et fonctions en rapport avec les personnages qui courent.

util.go qui contient une fonction utilitaire.

> En étudiant le code, vous devez expliquer (par exemple au moyen de schémas) comment se déroule un tour de la boucle principale du jeu (i.e. un appel à Update) : quelles sont les fonctions qui sont appelées ? à quoi servent-elles ? comment sait-on à quelle étape du jeu on est ? À quoi sert l'argument -tps lorsqu'on lance le jeu ?

2 Formalisation du fonctionnement du net code de base

Pour le moment, on souhaite simplement faire en sorte que quatre personne, chacune exécutant le jeu de son côté, puissent se connecter à un même serveur et, une fois qu'elles sont toutes les quatre connectées (et pas avant), puissent choisir leur personnage.

> Par un schéma, représenter la façon dont les 4 participants doivent interagir avec le serveur : quels sont les messages qui doivent être échangés ? comment un participant peut-il savoir quand il peut passer à la sélection de son personnage ? comment le serveur peut-il savoir que tous les participants sont connectés ?

3 Mise en place d'un premier serveur

On va maintenant passer à la réalisation du serveur décrit à la partie précédente.

Écrivez le code d'un serveur capable de recevoir des demande de connexion, de détecter quand quatre clients sont connectés, puis de prévenir ces clients qu'ils sont tous là.

Testez votre serveur en écrivant en Go un client rudimentaire qui se connecte au serveur (et en le lançant quatre fois). L'adresse du serveur devra être passée en argument aux clients.

4 Modification du code fourni pour interagir avec le serveur

Finalement, on va modifier le code du jeu pour mettre en place la connexion au serveur. On souhaite que, à l'écran d'accueil, la personne qui lance le jeu ne puisse appuyer sur espace pour passer à la sélection du personnage que si 4 joueurs sont connectés.

À quel endroit dans le code faut-il démarrer la connexion ?

À quel endroit faut-il attendre le message du serveur indiquant que quatre clients sont connectés ?

Modifiez le code fourni. Testez.

Que se passe-t-il au niveau du jeu pendant qu'un client attend les autres ? Expliquez. Proposez une solution.

Modifiez votre code pour mettre en œuvre la solution précédente afin de ne plus avoir de blocage de votre jeu quand un client attend les autres.