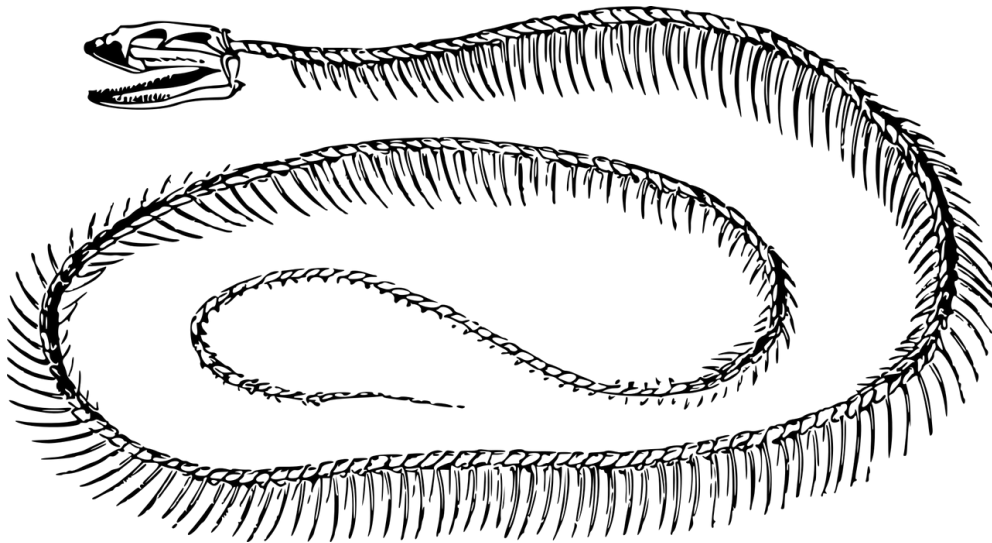


MASTER I INFORMATIQUE
DESIGN PATTERN

DÉCEMBRE 2017



Projet de Développement
Snake en réseau

Auteurs

David DEMBELE

Alassane DIOP

Rahmatou Walet MOHAMEDOUN

Professeur

Benoit DA MOTA

Table des matières

1	Introduction	2
1.1	Contexte	2
1.2	Objectifs	2
2	Protocole réseau	3
2.1	Motivation	3
2.2	Justification	3
2.3	Éventuels problèmes	3
2.4	Implémentation	3
2.4.1	Protocole UDP	3
2.4.2	Protocole TCP	4
3	Conception	5
3.1	Cas d'utilisation	5
3.2	Points de changement et d'évolution	5
3.3	Design envisagé	5
3.4	Justification	5
4	Conclusion	6

Chapitre 1

Introduction

1.1 Contexte

Dans le cadre de l'UE Design Pattern, de notre premier semestre de Master 1 Informatique à l'Université d'Angers, nous avons poursuivi la phase d'analyse d'un développement de jeu de Snake en réseau.

Le jeu peut se jouer en local ou en réseau, entre différents joueurs ce qui nécessite une implémentation des protocoles réseaux, pour gérer les transmissions entre les joueurs mais aussi entre les joueurs et le serveur. Ceci nous a amené à mettre en place une sorte de document RFC¹ pour expliquer les protocoles réseaux utilisés et comment les utiliser.

La conception de ce jeu est assez complexe, raison pour laquelle il nous a fallu réfléchir sur le choix des technologies et pour cela nous avons préféré adopter un patron de conception² pour garantir une solution éprouvée et validée par des experts, mais aussi trouver une solution avec les bonnes pratiques de conception.

1.2 Objectifs

L'objectif attendu est de permettre à différents joueurs de se connecter au jeu et de jouer entre eux. Chaque joueur doit disposer d'un serpent qu'il contrôle à l'aide du clavier, avec les touches de direction (HAUT, BAS, GAUCHE, DROITE). Le serpent doit se déplacer pour ramasser des bonus ; ces bonus lui permettent de grandir, éventuellement de devenir momentanément invisible, d'augmenter ses points de vie, de faire un retour dans le temps pendant un certain nombre de secondes.

Les serpents pourront s'entre-tuer ; un serpent tué se transforme en bonus en faveur du serpent tueur ainsi le dernier serpent à survivre sera le gagnant.

Le jeu doit permettre aux joueurs de faire des demandes de connexion et de déconnexion, d'envoyer leurs déplacements au serveur.

Le serveur doit gérer les connexions et déconnexions des joueurs et centraliser les actions des joueurs. En effet il gère le plan de jeu partagé par tous les joueurs.

À chaque connexion d'un joueur, le serveur doit récupérer les informations du joueur, lui attribuer un serpent, son niveau de compétence et son niveau de jeu.

À chaque déplacement d'un serpent, il doit faire une mise à jour du plan de jeu pour prendre en compte la nouvelle position du serpent, en informer les autres joueurs.

1. Les requests for comments (RFC), littéralement « demande de commentaires », sont une série numérotée de documents officiels décrivant les aspects techniques d'Internet, ou de différents matériels informatiques (routeurs, serveur DHCP). Wikipédia

2. Un patron de conception (souvent appelé design pattern) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels. Wikipédia

Chapitre 2

Protocole réseau

2.1 Motivation

Dans ce jeu, nous avons utilisé les protocoles TCP¹ et UDP².

Le protocole UDP nous sert dans le cas d'une transmission sans accusé de réception ; en effet le protocole UDP repose sur une communication unidirectionnelle. Lorsqu'une machine A envoie un paquet à une machine B, les paquets sont reçus par B sans accusé de réception vers la machine A.

Contrairement au protocole UDP, le protocole TCP repose sur une communication bidirectionnelle. La communication entre deux machines nécessite toujours un accusé de réception du récepteur vers l'émetteur. Dans le cas d'une communication où il faudra limiter la perte de paquet, TCP pourra nous servir.

2.2 Justification

Le protocole UDP est utilisé dans les cas suivants :

- déconnexion : lorsqu'un joueur souhaite quitter le jeu il envoie une requête de déconnexion au serveur, le serveur reçoit la requête et lui déconnecte du jeu sans lui en informer (sans accusé de réception).
- envoi du plan de jeu : lorsque le serveur centralise les actions des utilisateurs, il met à jour le plan du jeu et l'envoie aux joueurs ; cette opération peut être répétée autant de fois possible sans perturber le jeu.

Le protocole TCP est utilisé à chaque fois qu'il y ait échange entre le serveur et les joueurs sauf dans le cas d'une déconnexion ou d'un envoi du plan de jeu.

Les échanges d'informations entre le serveur et les joueurs doivent être fiables, sans erreurs notamment dans le cas d'une connexion d'un joueur au serveur, de la réception par les serveurs des actions des joueurs.

2.3 Éventuels problèmes

Assurer les communications entre les joueurs et entre les joueurs et le serveur, avec l'intégrité des paquets, s'avère complexe ; des erreurs liés à la transmission des paquets peuvent survenir.

En effet, à la fin d'une partie, il peut y avoir un paquet renseignant la position ou le déplacement d'un serpent ; si ce paquet arrive après la réception du paquet indiquant la fin de la partie, le joueur émettant le paquet pourra être bloqué.

1. Transmission Control Protocol (littéralement, « protocole de contrôle de transmissions »), abrégé TCP, est un protocole de transport fiable, en mode connecté. Wikipédia

2. Le User Datagram Protocol (UDP, en français protocole de datagramme utilisateur) est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport du modèle OSI, il appartient à la couche 4, comme TCP. Wikipédia

2.4 Implémentation

2.4.1 Protocole UDP

Protocole UDP

2.4.2 Protocole TCP

Protocole TCP

Chapitre 3

Conception

3.1 Cas d'utilisation

3.2 Points de changement et d'évolution

3.3 Design envisagé

3.4 Justification

Chapitre 4

Conclusion