Maxime Routhier Couture – 11 064 947

Vincent Fortier – 11 084 764

Keven Fecteau – 11 089 305

Antoine Croteau – 11 081 857

Znake

Rapport détaillé

# Présentation

Znake est une adaptation multi-joueurs du vrai jeu de Snake. Pour remettre en contexte, le vrai jeu de Snake consistait à récupérer une pomme en avançant constamment avec notre serpent. Lorsque l’on récupérait une pomme, notre serpent s’agrandissait, ce qui rendait le jeu de plus en plus difficile. La partie se terminait lorsque le joueur entrait en collision avec le mur ou lui-même.

Dans notre cas, Znake enlève complètement le concept de la pomme. Le serpent s’agrandit sans arrêt. Le but du jeu est tout simplement d’être le dernier serpent en vie dans une arène contre d’autres serpents (joueurs). Un serveur s’occupe de contrôler l’avancement de la partie, alors que les clients eux s’occupe de dessiner le jeu ainsi que d’envoyer leurs directions.

# Comment jouer

Pour jouer, il faut tout d’abord lancer les programmes suivants :

* /Executables/SnakeServeur/SnakeServer.exe
	+ Ceci est le serveur, il doit être démarré en premier.
* /Executables/SnakeClient/Snake.exe
	+ Pour ce qui est du client, deux instances sont nécessaires pour pouvoir lancer une partie

Pour le serveur, on ne fait que cliquer sur « Démarrer le serveur ». Le nombre de joueurs est inchangeable, car notre application ne marche pour l’instant qu’avec deux joueurs. Pour ce qui est des deux clients, une fois le serveur lancée, on appuie sur « Entrée » et la partie va débuter lorsque les deux clients ont fait cette action.

Pour contrôler le serpent, il ne suffit que d’utiliser les touches directionnelles pour se diriger. La partie se termine lorsqu’un des deux serpents fonce dans un mur ou dans un serpent.

# Compilation

Ouvrir /Snake.sln avec Visual Studio 2010

Pré requis : XNA Game Studio doit être installé sur votre poste (gratuit) (http://www.microsoft.com/en-us/download/details.aspx?id=23714)

Faire build->Build Solution

# Architecture de l’application

Tout d’abord, il faut préciser que le concept utilisé est une logique de Client/Serveur. En gros, le serveur (un thread) s’occupe de faire avancer la grille de jeu selon les commandes des joueurs et de renvoyer les nouvelles positions aux clients. Pour les clients (un thread par client), ceux-ci s’occupe d’afficher la grille et d’envoyer leurs directions au serveur.

## Client

* Se connecte au serveur
* Attend le début de la partie
* Reçoit un numéro de joueur
* Tant que la partie est en cours ->
	+ Envoi des directions au serveur
	+ Reçoit les nouvelles positions des serpents des joueurs
	+ Dessine la grille
* Ferme la connexion

**Serveur**

* Démarrer le serveur
* Attend que 2 joueurs se connectent
* Envoyer un numéro de joueurs à chacun des joueurs connectés
* Démarre la partie
* Tant que la partie est en cours ->
	+ Reçoit les directions accompagnées du numéro des serpents
	+ Renvoi les nouvelles positions des serpents aux joueurs
* Envoie un code spécial dans le paquet contenant usuellement les nouvelles positions des joueurs à ces-dits joueurs, avec une indication du gagnant.
* Ferme les connexions et éteint le serveur

## Sockets

Les sockets sont utilisés pour la communication entre le serveur et les clients. Les données sont envoyées et reçus entre les deux par le protocole TCP.