UNIVERSITÉ DE NANTES

Master 1 ALMA

Projet de Réseaux et Protocoles Internet

Bomber Unicorn

Étudiants :
Coraline MARIE et Vincent RAVENEAU

Intervenant :
Pierrick PASSARD

21 mars 2014

Table des matières

Introduction

Dans un monde où les connexions sont de plus en plus présentes, il est indispensable, en tant que futurs ingénieurs en informatique, de connaître et de comprendre les différentes structures de réseaux existentes sur lesquelles notre monde évolue.

Dans le cadre du cours *Réseaux et Protocoles Internet*, nous avions pour objectif la création d'une application réseau de type *client/serveur*. Pour ce faire, nous avons recréé le jeu multijoueur de BomberMan avec de nouveaux graphismes : `Bomber Unicorn`.

Ce rapport présente donc la création de `Bomber Unicorn` au travers de nombreuses étapes de travail, ainsi que les différentes fonctionnalités de ce jeu. Pour cela nous verrons d'une part la conception de l'application, et le développement d'autre part.

Conception

Dépendances externes

Avant même de concevoir un logiciel, il faut au préalable le penser, ainsi que définir ses besoins et ses diférentes fonctionnalités. Dans cette optique, notre cahier des charges nous imposait de concevoir l'application en `langage C`. Ce langage ne fût pas choisi pour ses performances dans la conception d'applications réseau, mais plutôt dans une optique pédagogique. En effet, il est moins évident de programmer des échanges entre clients et serveur en `langage C` plutôt qu'en `Java` ou en `C++`, où tout est mis à disposition pour aider le programmeur.

En ce qui concerne la partie graphique de l'application, nous avons choisi d'utiliser le `CSFML` qui est le binding officiel de la SFML pour le langage C. La `SFML` étant une interface de programmation destinée à construire des jeux vidéo ou des programmes interactifs en `C++`.

Fonctionnalités du logiciel

Pour ce projet, aucune fonctionnalité ne nous a été imposée, excepté qu'il fallait créer une application de type *client/serveur*. Pour ce faire, nous avons décidé d'utiliser le serveur en tant que moteur du jeu et les clients en tant que joueurs. Le serveur est capable de gérer jusqu'à XXX parties en simultané (si la machine qui le lance est assez puissante), et chaque partie comprend de 2 à 4 joueurs. Pour des raisons de simplicité, de performance et de fluidité, une partie se déroule en tour par tour (un client doit attendre son tour pour jouer).

Les règles du jeux sont simple, pour gagner, un joueur doit éliminer tous les autres en réduidant leur point de vie à 0. Pour ce faire, chaque joueur dispose au départ de 3 points de vie, et peut poser jusqu'à 3 bombes en même temps sur la carte. Lorsqu'une bombe est posée, elle change d'état progressivement jusqu'à exploser. Un joueur peut être blessé par ces propres bombes ? ? ? ? ? Lorsqu'une bombe explose, les joueurs présent autour de la bombe sont touchés et perdent un point de vie ? ? ? ? ?

Interface graphique

En ce qui concerne l'interface graphique, nous l'avons entièrement construite à l'aide de la `CSFML`. Il s'agit d'un ensemble d'éléments graphiques, appelés *sprite*

que l'on place dans une fenêtre. Nous avons créé l'intégralité de nos sprite nous même à l'aide d'une base de donnée libre d'images et d'un logiciel infographique.

Voici quelques exemples de sprites :

Voici la carte créée avec les sprites :

Développement

Les clients et le serveur communiquent en s'envoyant divers messages. Ces messages concerne l'état du jeu ou l'état des joueurs (position, point de vie, nombre de bombes, ...). Ces messages sont sous la forme de chaînes de caractères et agissent sur les variables du jeu.

Serveur

Lorsque le serveur est lancé, il reçoit en paramêtre le nombre de parties à gérer, et le nombre de joueurs nécéssaire par partie.

Client

Chaque client reçoit en paramêtre un nom (celui du joueur) et une adresse (le port du serveur).

Conclusion