

Documentation

Fonctionnement : À l'aide du menu, le joueur peut choisir de jouer à plusieurs mods, dans chaque mod, il y a divers niveaux, à savoir des cartes de jeu.

Lorsqu'aucune connexion n'est établie, le joueur a la possibilité de jouer avec quelqu'un d'autre sur le même clavier, le joueur 1 utilise les flèches directionnelles pour se déplacer et la touche Entrée pour poser une bombe tandis que le joueur 2 utilise les touches ZSQD pour se déplacer et la touche Espace pour poser une bombe. Nous avons choisi de limiter le nombre de bombes à 1 avec un rayon de 3 au début de chaque partie.

Il peut également choisir de se connecter avec un autre joueur via le menu, ainsi, chaque joueur utilisera les flèches directionnelles pour se déplacer et la touche Entrée pour poser une bombe. Quand l'un des 2 joueurs quitte l'application, la connexion est donc interrompue et l'autre joueur se verra éjecter du jeu.

Sinon il choisit tout simplement de quitter l'application.

Limites : Le jeu est jouable en réseau privé mais pas en réseau public, le reste des consignes semble avoir été accompli.

Les problèmes que nous avons rencontrés et les solutions que nous avons trouvées : Avant de commencer à programmer le jeu bomberman, nous avons compris qu'il fallait d'abord savoir comment utiliser la bibliothèque termios.h, cependant quand nous avons essayé de lire la page man termios, nous avons vite été découragés par la quantité d'informations fournies sur la page man, en effet, il y avait tout simplement beaucoup trop de choses à lire, nous avons donc abandonné l'idée de lire tout cela. Ainsi, nous avons cherché sur Internet pour comprendre le termios avec des tutoriels, nous avons ensuite pu saisir que le termios mettant le terminal en mode brute en envoyant directement caractère par caractère à la fonction read au lieu d'attendre que l'utilisateur appuie sur Entrée avant d'envoyer quelque chose. Plus tard, nous nous sommes rendu compte que dans le sujet du projet, tout était expliqué dans la partie (B) Manipuler le terminal et qu'il fallait juste appliquer 3 fonctions du termios.h. À ce moment-là, nous avons pris un peu de retard par rapport aux autres.

Vers la fin, nous étions confrontés à un « segmentation fault (core dumped) » quand nous faisions plusieurs malloc et free dans des fonctions, il nous a fallu 2 ou 3 jours pour comprendre que dans une de nos fonctions, celle qui servait à concaténer des chaînes de caractères avec des realloc avant de renvoyer le nouvel espace alloué était mal utilisée. En effet, cette fonction s'appelle :

char * optimisation (char * ancienne chaîne, const char* chaîne à ajouter);

Ainsi, au lieu de faire :

```
nouvelle chaîne = optimisation (ancienne chaîne, chaîne à ajouter);  
write (1, nouvelle chaîne, strlen(nouvelle chaîne));  
free (nouvelle chaîne);
```

Nous avons fait :

```
optimisation (ancienne chaîne, chaîne à ajouter);  
write (1, ancienne chaîne, strlen(ancienne chaîne));  
free (ancienne chaîne);
```

Cette grave erreur nous a véritablement gênés. Nous avons découvert cette faute par pur hasard en nous baladant dans notre code, nous estimons que nous avons eu de la chance... Ce fut le problème le plus coriace à notre avis.

Avant d'entamer la partie sur le serveur, nous ne savions pas par quoi commencer, il y avait très peu de données dans le sujet du projet, nous étions par conséquent obligés d'aller chercher un tutoriel sur Internet pour apprendre à programmer un serveur pour pouvoir communiquer 2 ordinateurs à distance. Nous tenons à remercier les personnes qui ont contribué au tutoriel concernant les sockets en C sur le site <http://sdz.tdct.org/sdz/les-sockets.html>. Sans quoi, nous n'aurions pas réussi à programmer un serveur. Le tutoriel était un peu long à lire, cela nous a pris un peu de temps malgré tout.

Les contributions des divers membres du groupe : **Piaonan Lin** a principalement réalisé l'affichage de la carte sur le terminal, le déplacement des personnages avec des collisions et les boosts ainsi que la mise en place du serveur.

Lucas Michelin a principalement réalisé la gestion des bombes, le menu au début d'une partie et les commentaires des fonctions que nous avons écrites.

Bien évidemment, nous tenons à préciser que nous nous sommes aidés lorsque l'autre était bloqué dans la réalisation de sa propre tâche. Donc aucun d'entre nous n'a réalisé tout seul sa tâche spécifiée dans son propre coin.