

PG 110 : Bomberman

Fonctionnalités implantées

Gestion des collisions

Modification de la fonction *player_move_aux* qui autorise ou non un mouvement pour gérer les collisions avec les éléments de la carte. Modification de la fonction *player_move* pour détecter si le joueur va sortir de la carte et empêcher le mouvement si besoin est.

Bombes

Bombes stockées dans un tableau de 20 bombes. Au fur et à mesure que les bombes sont posées, elles tourneront sur un morceau du tableau contenant autant de cases que le nombre de bombes que le joueur est autorisé à poser simultanément. La dernière bombe prenant alors l'espace occupé par la dernière à avoir explosé. A chaque bombe est affecté un timer, incrémenté à chaque tour de la boucle principale. Un module *bomb.c/.h* a été créé.

Gestion des vies

Le joueur perd une vie dès qu'il se trouve dans le champ d'explosion d'une bombe ou qu'il croise la route d'un monstre. Au moment de l'explosion, on fait correspondre la position du joueur et celles des explosions et à chaque tour de boucle principale, la position du joueur est comparée à celle de chaque monstre.

Bonus/Malus

Selon les probabilités d'apparition de bonus données dans le fichier *constant.h*, des bonus prédéfinis peuvent apparaître. On utilisera les fonctions *random_bonus*, *display_bonus* et la fonction *map_display*, qui a été modifiée pour prendre en compte le type *CELL_BONUS*.

Panneau d'informations

Le squelette du panneau d'informations était déjà très abouti, il a simplement fallu remplacer la variable déterminant le chiffre à afficher par le pointeur sur l'attribut du joueur, qui lui même sera modifié tout au long du jeu.

Monstres

De manière analogue aux bombes, les monstres sont stockés dans un tableau de monstres. Les fonctions relatives aux monstres sont très proches de celles relatives au joueur. La grande différence réside dans l'absence de vie et les mouvements quasi-aléatoires. Un module *monster.c/h* a été créé. Les monstres initialement chargés sur la carte diffèrent de ceux que l'on peut trouver sous les caisses.

Fichiers cartes et chargement

Chaque carte est stockée dans un fichier *.txt* qui reprend la taille de cette dernière et les éléments la composant, dont la position du joueur et des monstres. Elles sont stockées dans le dossier *map* et dans le dossier portant le numéro du niveau auquel elle appartient.

Passer de carte en carte, de niveau en niveau

Au sein d'un même niveau, le joueur doit ouvrir des portes fermées à l'aide de clés qu'il devra ramasser. Chaque porte donne accès à la carte suivante. Une image s'affiche à chaque franchissement de carte/niveau. Le programme cherchera à savoir si le joueur est sur la dernière carte du niveau ou non. Si oui, il chargera la fonction *level_inc*, sinon il chargera la fonction *level_map_inc*. Nous sommes partis sur une base de deux cartes par niveau, mais on peut aisément y remédier.

Affichage du niveau en cours

La structure *game* contient l'attribut *curr_level*, qui pointe vers le niveau actuel du joueur. Chaque niveau porte un numéro qu'il suffit de reporter dans le panneau d'information.

Sauvegarde / Chargement partie

En pressant la touche *S* au cours du jeu, l'utilisateur verra sa partie sauvegardée dans un fichier *save.txt* placé à la racine du dossier du jeu. Cette sauvegarde contiendra sur la première ligne, toutes les informations propres au joueur, sur la seconde, le niveau et la carte en cours, sur la troisième, la taille de la carte. Ensuite viendra la carte, implémentée de la même manière que dans les fichiers carte, puis les positions de tous les monstres. Il n'est pas possible d'effectuer plusieurs sauvegardes, chaque sauvegarde écrasant la précédente.

Pause

A tout instant, le joueur peut presser la touche *P* qui mettra le jeu en pause, il devra alors presser à nouveau cette touche pour reprendre le jeu. La fonction qui s'en occupe, colle juste une image sur l'ensemble de la fenêtre et attend que l'utilisateur presse à nouveau *P*.

Editeur de carte

Le système proposé limite la flexibilité du jeu en rendant compliquée l'édition d'une carte. Un outil a été implémenté, permettant de simplement coller à l'aide des flèches directionnelles et de la touche *Entrée* les éléments de la carte souhaitée sur une carte vierge. Une fois satisfait de sa carte, il faut

alors presser la touche *D* pour que la carte soit sauvegardée dans le dossier *map* sous le nom *map_default.txt*, il suffit alors de la déplacer dans le dossier du niveau souhaité.

Thème

Nous avons remplacé l'essentiel des sprites par ceux de la version GameBoy Advance de *The Legend of Zelda*.

Bugs

En mode debug, à la fermeture du jeu, soit lors de la fin du jeu ou lorsque le joueur perd toutes ses vies, un *Segmentation Fault* nous est retourné. Même après plusieurs heures de debugging vain, nous n'avons pas réussi à identifier la source du problème. D'autant que le problème ne survient qu'en mode Debug, ajoutant à notre incompréhension.