

COMPTE RENDU PROJET

PROGRAMMATION PG110 : BOMBEIRB

EL FAIZ MOHAMED

ABID MOOMEN

Encadrant : Guillaume Mercier

Introduction :

La réactualisation du jeu se fait en quatre fonctions :

- game_update():cette fonction parcourt toutes les cartes et gère les collisions entre le joueur, les bombes et les monstres. Cette fonction appelle aussi la fonction game_monster_get_player() qui gère l'intelligence des monstres.
- game_display() :cette fonction affiche : la carte actuelle, le panneau d'informations, les monstres ,les bombes et le Player.

-game_speed_one() :cette fonction est appelée chaque seconde. Elle est responsable de la gestion des bombes.

-game_speed_lvl() : cette fonction gère le mouvement des monstres en fonction de leur vitesse.

Chargement des cartes :

Au début, on charge les toutes les cartes dans le jeu à partir du dossier /data.

Si le joueur passe par une porte ouverte, on change la carte actuelle à l'aide de la fonction game_get_current_map() .

Gestion des portes :

Lorsque le joueur arrive sur une porte ouverte, deux cas de figures se présentent :

-Si la porte mène au niveau supérieure, on appelle la fonction player_inc_lvl() qui incrémente le niveau du jeu.

-Si la porte mène au niveau inférieur, on appelle la fonction player_dec_lvl() qui décrémente le niveau du jeu.

Si la porte est fermée, le joueur reste bloqué devant la porte.

Gestion des Bombes :

La fonction map_bomb_dec() balaye toute les cartes, lorsque le joueur pose des bombes, cette fonction diminue leurs TTL et appelle la fonction map_explosion() qui explose la

bombe et appelle la fonction `map_bomb_effect()` pour transformer les cellules `CELL_BOX` en `CELL_EXPLOSION` puis en `CELL_BONUS` correspondants.

On appelle aussi la fonction `map_clear_explosion()` qui balaye toute les cartes et change les cellules d'explosions en `CELL_EMPTY` si il ne contiennent pas de bonus.

Gestion des bonus et des malus :

Lorsque le joueur se déplace sur un `CELL_BONUS`, on récupère le type de bonus avec la fonction `map_get_sub_type()` et on agit en fonction du bonus pour augmenter/diminuer les attributs du Player suivant le bonus rencontré. Cela est fait dans la fonction `player_move_aux()`.

Gestion des vies :

Dans la fonction `game_update`, -on fait appel à trois fonctions :

- `map_bomb_player()` : elle decremente le nombre de vie de player s'il se trouve dans une cellule `CELL_EXPLOSION` en appelant la fonction `player_down()`.
- `map_bomb_monsters()`:elle tue les monstres qui se trouvent dans une cellule `CELL_EXPLOSION` en utilisant la fonction `map_monsters_kill()`.
- `monster_collision()` : elle decremente le nombre de vie de player s'il a les mêmes coordonnées x et y d'une structure monstre.

La fonction `player_down()` ne diminue la vie du player que lorsqu'il n'a pas été touché récemment.(on repère ca à l'aide du champ `player->timehurt` dans la structure player).

Gestion des monstres :

Pour chaque carte, on crée un tableau de structures monstres ,et on ajoute les monstres au tableau lors de leurs créations.

La création des monstres se fait lorsque la cellule est de type `CELL_MONSTER` dans la fonction `map_display()` incluse dans le `game_display()`. Pour ce faire, on utilise la fonction `map_add_monster()`,qui rajoute un monstre au tableau `map->monsters` de la map concernée. Donc la création de monstres s'effectue si on charge une cellule `CELL_MONSTER` de la carte ou dans le cas de l'apparition du cellule `CELL_MONSTER` lors de l'explosion d'une bombe.

Les monstres bougent grâce à la fonction `monster_move()`.

On a ajouté un module d'intelligence artificielle grâce à la fonction `monster_get_player()` qui abscisses du monstre et du joueur et change la direction des monstres.

On n'a pas arrivé à :

- Conserver la portée initiale des bombes pour lesquelles la mèche est déjà allumée.
- Le module d'intelligence artificielle n'est pas parfait : si le monstre se déplace verticalement vers le joueur et il y'a un obstacle dans son chemin, il peut pas s'en passer.
- Implementer trois mondes different (easy,medium,hard).
- Changer la taille de la fenêtre.