16/05/2018

RAPPORT PROJET PROGRAMMATION :

REALISE PAR : ELLOUZE MALEK

1. GESTION DES DEPLACEMENTS :

- Modification dans la fonction map_is_inside pour que le joueur ne dépasse pas le cadre de la carte.
- Modification dans player_move_aux pour que le joueur ne puisse pas aller sur les éléments du décor.
- Modification dans player_move pour que le joueur puisse déplacer les caisses suivant les conditions citées.

2. CHARGEMENT DES MONDES :

- Implémentation de la fonction map_load_file qui permet de lire le contenu d'un fichier structuré comme dans l'énoncé et attribuer à chaque cellule de la carte le code qui lui correspond.
- Implémentation de la fonction maps_init qui permet, pour un nombre maximal de cartes choisis, de charger les cartes.
- Les cartes sont stockées dans un dossier map sous le dossier sources.
- Le codage des différentes cellules est effectué dans map.h.

3. CHARGEMENT DES CARTES :

- Le chargement des cartes se fait à travers les fichiers easy.txt ou hard.txt stockées dans le dossier game.
- Lecture des fichiers sous game dans la fonction game_menu pour déterminer les paramètres nécessaires au chargement de la carte (niveau maximal, niveau actuel …).

4. PANNEAU D'INFORMATION :

- Modification effectuée dans la fonction game_banner_display.
- Implementation des fonctions player_get_life, player_get_nb_key, player_get_portée … pour modifier les valeurs à chaque fois dans le panneau d'information.

5. GESTION DES BOMBES :

- Les bombes sont configurées comme étant une liste chaînée où chaque bombe pointe sur la bombe suivante.
- On ajoute des bombes dans la liste de bombes via la fonction bomb_list_add (ajout au début de la liste) et on les supprime via la fonction bomb_list_remove(suppression à la fin de la liste).
- Les fonctions incrémentation_du_compteur et bomb_update_state permettent d'incrémenter le compteur de la bombe et donc de diminuer la mèche jusqu'à explosion.
    1. Fonction bomb_display :
        - Les fonctions scenery_stop_explosion et box_stop_explosion permettent respectivement de limiter l'étendue de la bombe et de ne faire exploser qu'une seule caisse dans une direction donnée.
        - La fonction conditions_of explosion permet de définir les différentes conséquences de l'explosion vis à vis des éléments de la carte

(apparition des bonus dans les caisses, apparition de monstres, tuer les monstres…).

- La fonction player_dec_life permet de diminuer le nombre de vies du joueur s'il se trouve dans la zone d'explosion.
- Si le joueur n'a qu'une seule bombe, une bombe est automatiquement ajoutée dans son inventaire.

6. BONUS ET MALUS :

- L'ajout ou le retrait des bonus se fait via les fonctions du type player_set_BONUSTYPE.

7. GESTION DES VIES :

- Si le joueur n'a plus de vie, un écran game over s'affiche. Il faut appuyer sur n'importe quelle touche pour quitter.

8. GESTION DES MONSTRES :

- Le monstre est une combinaison du player vis à vis sa manière de bouger et de bombes vis à vis la modification automatique de ses mouvements.
- Si le joueur va sur un monstre, il perd une vie et il devient invulnérable pendant un certain temps grâce à la fonction player_set_invulnerability_on.
- La vitesse des monstres augmente avec les niveaux. L'incrémentation est faite dans la fonction next_lvl.

9. FIN DE PARTIE :

- Quand le joueur arrive sur la case de la princesse, un écran vous avez gagné s'affiche. Cet affichage est géré dans la fonction game_menu.

10. PAUSE :

- Une variable pause est ajoutée dans la structure game et initialisée à 0. L'appui sur la touche p change la valeur de pause à 1 et désactive donc les fonctionnalités du jeu jusqu'à ré appui sur p.

11. SAUVEGARDE :

- Une fonction save_map a été implémentée mais non utilisée dans le jeu.

12. BONUS :

- Un menu interactif pour accéder aux différentes cartes grâce à la fonction game_menu.