

Projet Technologique « Battlefield Mario » : Fiche n°5



Dans cette fiche nous allons mettre en place quelques éléments pour la création, l'enregistrement et le chargement de niveaux pour notre jeu.

1 Editeur de carte

Faites en sorte que l'appui sur la touche *e* fasse basculer le jeu en mode d'édition. Dans ce mode, le personnage de Mario disparaît et la case en haut à gauche de l'écran se met en surbrillance par rapport au reste des éléments. On parlera alors de case tampon. L'idée de ce mode est de déplacer le curseur de case en case et de permettre de définir l'objet (et donc la texture) qui sera placé à la position du curseur.

1. Redéfinissez le comportement des touches des flèches (haut, bas, gauche et droite) afin de déplacer le curseur (case mise en surbrillance) selon la direction prise
2. Chaque appui sur la touche *Space* doit changer la texture de la case du curseur pour appliquer la texture de l'objet sélectionné. On ajoute donc l'objet en question à la position courante du curseur. Si le tampon possède la même texture que l'objet courant, ce dernier est effacé.
3. La touche *Tabulation* doit parcourir la liste des objets de la carte définies par les appels à `map_object_add`.

2 Sauvegarde/chargement

Dans cette partie, vous allez développer les fonctions de sauvegarde (`map_save`) et de chargement des cartes (`map_load`).

2.1 Sauvegarde

1. La touche `s` doit appeler la fonction `map_save`
2. Nous vous laissons libre de choisir le format d'enregistrement de votre carte dans un fichier (i.e. les différentes informations d'une carte pour permettre son chargement par la suite). A cette fin, vous utiliserez les appels systèmes de gestion de fichiers (*open, read, write, lseek, ...*).

Pour information, une carte est caractérisée par ces éléments :

- ses dimensions (largeur et hauteur);
- le nombre d'objets différents qu'elle peut contenir;
- le contenu de chaque case (une matrice d'entiers);
- les caractéristiques de chaque type d'objet (nom du fichier image, nombre de sprites, propriétés)

Pour vous aider, la définition de fonctions suivantes peut s'avérer utile :

map_width() : retourne la largeur de la carte;

map_height() : retourne la hauteur de la carte;

map_objets() : retourne le nombre d'objets différents que la carte renferme;

map_get(x,y) : retourne le type d'objet à la position (x, y).

map_get_name(obj) : retourne un pointeur sur le chemin d'accès au fichier PNG décrivant la texture de l'objet;

map_get_frames(obj) : retourne le nombre de sprites;

map_get_solidity(obj) : retourne `MAP_SOLID`, `MAP_LIQUID`, `MAP_AIR` ou `MAP_SEMI_SOLID`

map_is_destrucible(obj) : retourne 1 si l'objet est destructible, 0 sinon;

map_is_collectible(obj) : ...

2.2 Chargement

La fonction `map_load` doit effectuer à peu près les mêmes opérations que celles réalisées dans `map_new`. Les données ne proviennent pas du programme mais d'un fichier. Faites en sorte de pouvoir charger une carte en utilisant l'option `-l` suivi du chemin d'accès au fichier de la carte à charger, lors du lancement du jeu

Testez les opérations de sauvegarde et d'enregistrement de cartes.