**map\_editor.py**

**Provenance :**

Création entièrement originale.

**Utilités :**

* Créer les cartes du jeu

**Fonctionnement :**

Ce programme permet d’utiliser toute une interface graphique pour créer efficacement les cartes du jeu.

Le début du programme définie une classe « Bloc » qui regroupe toutes les caractéristiques d’un bloc : nom / collisions / s’il appartient à une image composée de plusieurs blocs / s’il doit être affiché devant le joueur / s’il mène à une carte. Cette classe permet également d’afficher les différentes caractéristiques du bloc.

Les fonctions « addBackup » et « loadBackup » permettent d’enregistrer les 100 dernières modifications et de pouvoir annuler ces dernières : ce sont les fonctions utiles au fonctionnement du « Contrôle Z » implémenter dans l’éditeur.

La fonction « WriteMessage » permet d’afficher un message ou une erreur sur l’écran : lors de son appelle, on affiche le texte donné en paramètre.

La foncion « readFile » est très importante : cette fonction interprète les fichiers texte des cartes déjà créées. En effet, le texte de ces fichiers est séparé par des caractères spécifiques : « $ », « / », « | » ; cette segmentation simplifie l’organisation du fichier. Ainsi, pour chaque couche de la carte, pour chaque ligne de la couche et pour chaque bloc on la ligne, on récupère les caractéristiques du bloc et on l’ajoute à la liste de la carte.

La foncion « writeMapFile » est aussi importante et procède à l’action inverse de la fonction précédente. En effet, cette fonction va encoder tous les blocs de la carte dans un fichier texte qui sera exploitable par le jeu, ou l’éditeur de carte lui-même.

La fonction « save » permet d’enregistrer une carte avec un nom spécifique ou non. Elle se base sur l’utilisation de la fonction précédente.

Ensuite, le code correspond à une gigantesque boucle dans laquelle chaque état du programme active des fonctions différentes et précises. Pour plusieurs états de l’éditeur, on affiche simplement des boutons utiles.

Une image contenant texte, capture d’écran, logiciel, Système d’exploitation

Description générée automatiquement

Si l’utilisateur est en train de manipuler des blocs, de modifier la carte, on effectue cette suite d’instruction :

* Détecter si des touches sont appuyées pour déplacer la caméra
* Remplir l’écran d’un fond blanc pour effacer l’affichage de la frame précédente.
* Dessiner le contour de la carte
* Afficher tous les blocs, avec potentiellement une caractéristique (collisions / appartenant à un groupe de blocs…)
* Afficher le numéro de la couche sur laquelle on se trouve
* Récupérer les coordonnées de la souris, et leur appliquant le décalage de l’affichage de la caméra pour obtenir ses coordonnées réelles dans la carte
* Récupérer le bloc ainsi cliqué
* Modifier la caractéristique du bloc actuellement affichée (collisions / groupe de bloc/ etc) ou placer un bloc à l’endroit cliqué (si le bloc appartient à un groupe de blocs qui forment une grande image, on affiche tous les blocs de l’image)

Une image contenant capture d’écran, vert, conception

Description générée automatiquement

Si le joueur est en train de choisir un bloc de la carte bibliothèque, on effectue exactement la même suite d’instruction, sauf qu’à la lieu de placer un bloc, on enregistre le type de celui que l’on clique, pour pouvoir le reporter sur notre carte.

Une image contenant texte, capture d’écran

Description générée automatiquement

La suite du programme permet de détecter quelle touche est pressée et quelles sont les instructions à exécuter. Les instructions implémentées dans le programme permettent les choses suivantes :

* Entrer une chaine de charactères pour choisir un fichier / sauvegarder sous un certain nom / enter une taille de carte / etc
* Afficher le menu d’aide
* Afficher la carte bibliothèque
* Afficher les collisions / l’appartenance à un groupe de blocs / si le bloc s’afficher derrière le joueur / si le bloc mène à une autre carte
* Afficher toutes les couches en même temps ou non
* Rajouter/retirer une ligne
* Rajouter/retirer une colonne
* Zoomer / Dézoomer
* Annuler la dernière modification

Si la souris est cliquée, on vérifie si elle est en collisions avec un bouton. Si c’est le cas, on actionne la fonction du bouton concerné.

Si la molette est tournée, on change de zoom de la caméra.