

COMPTE RENDU DE PROJET CPP

Chakib BENNACER - Raphaël LESBROS

Clément MATHIEU DRIF - Nils PONSARD - Hugo TRITZ

19 janvier 2020

Nous avons décidé de réaliser le jeu "Space Invader" en utilisant minGL en tentant le rendre plus agréable à jouer et de le faire ressembler autant que possible au jeu original.

1 Ajouts au jeu

- Ajout de musique et de bruitages
- Ajout d'une page d'accueil avant le démarrage du jeu
- Ajout d'un menu de pause permettant de quitter momentanément le jeu.
- Ajout d'un système de niveaux qui génère de nouveaux "invaders" et leur octroie des vies supplémentaires au fur et à mesure du déroulement des niveaux.
- Ajout de "powerUps" donnant différents bonus : vie supplémentaire, bonus de score....
- Ajout d'un système de score augmentant à chaque envahisseur ou à chaque roquette détruit.
- Ajout d'un système de vie : le joueur perd une vie lorsqu'il est touché.

Ces fonctionnalités ont été intégrées au jeu, en adaptant les paramètres afin de faciliter les tests.

2 Gestion des paramètres

Les paramètres sont stockés dans config.yaml, avec le format de paramètres yaml. Si le fichier n'existe pas il est créé avec les valeurs par défaut. Si une valeur n'est pas présente ou incorrecte dans le fichier de configuration, la valeur par défaut est chargée.

La fonctionnalité a été testée en supprimant le fichier de configuration, en le modifiant avec des valeurs incorrectes ou manquantes.

3 Enregistrement des scores

Quand un nouveau meilleur score est enregistré, le nom du joueur (en trois lettres) est demandé. Son score est enregistré dans le fichier scores.yaml qui est créé si inexistant. Le jeu conserve les 3 meilleurs scores. Si une valeur incorrecte est présente, la ligne est supprimée et le score de la prochaine partie sera enregistré. Lors de l'insertion d'un nouveau score, la liste des scores reste toujours triée dans l'ordre décroissant.

Cette fonction a été testée en jouant et en battant les meilleurs scores, en supprimant le fichier des scores et en utilisant des valeurs incorrectes.

4 Système de collisions

L'écran de minGL étant composé de pixels et non de caractères, nous avons réalisé une fonction pour tester la collision entre deux rectangles avec des positions et des tailles en pixels. Cette fonction a été validée avec la fonction `testCollision()` dans le dossier `testsFonctions`.

5 Modifications de minGL

Nous avons modifié le code source de minGL pour pouvoir rendre le jeu interactif; ces modifications sont les suivantes :

- Ajout de la fonction `displayText` permettant d'afficher du texte à l'écran: fonction testée en l'intégrant au jeu et en le lançant.
- Ajout de la fonction `setBgColor` permettant de changer la couleur d'arrière-plan.
- Modification de la gestion des touches afin de savoir quelles touches sont pressées sans interrompre l'exécution du jeu.
- Modification de la gestion de l'affichage (`updateGraphic` et `callDisplay`) pour supprimer le clignotement du texte.
- Ajout de la ligne `glutSetKeyRepeat(GLUT_KEY_REPEAT_OFF)` qui désactive la répétition de touches pour une meilleure navigation dans les menus.
- Ajout de la fonction `ResetKey()` permettant de réinitialiser l'état de la touche à « false » (non pressée)
