

compte rendu projet cpp

Nils PONSARD Clément MATHIEU-DRIF Hugo TRITZ
BENNACER Chakib Raphaël LESBROS

19 janvier 2020

Nous avons décidé de faire le jeu Space Invader en utilisant minGL pour le rendre plus agréable à jouer et pour qu'il ressemble plus au jeu original

1 Ajouts au jeu

- Ajout de musique et de bruitage
- Ajout d'une page d'accueil avant de démarrer le jeu
- Ajout d'un menu de pause permettant de quitter ou de reprendre le jeu
- Ajout d'un système de niveaux qui rajoute des invaders et qui leur donne plus de vie au cours des niveaux
- Ajout de powerUps donnant différents bonus : Vie supplémentaire, Bonus de score.
- Ajout d'un système de score augmentant à chaque invader détruit ou à chaque roquette détruite.
- Ajout d'un système de vie, on perd une vie quand on se fait toucher

Ces fonctionnalités ont été testé en les utilisant dans le jeu, en adaptant les paramètres pour faciliter les tests.

2 Gestion des paramètres

Les paramètres sont stockés dans config.yaml, il utilise 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.

fonctionnalité testé 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, votre nom (en trois lettres) est demandé. Ce score est enregistré dans le fichier scores.yaml qui est créé si inexistant. Le fichier contient max 3 scores triés dans l'ordre décroissant. Si une valeur incorrecte est présente, la ligne est supprimée et le score de la prochaine partie sera enregistré. L'insertion d'un score est faite pour que la liste des scores soit 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 mettant des valeurs incorrectes.

4 Système de collisions

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

5 Modifications de minGL

Nous avons modifié le code source de minGL pour pouvoir faire un jeu interactif, ces modifications sont :

- Ajout de la fonction `displayText` permettant d'afficher du texte à l'écran, testé 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 pour savoir quelles touches sont pressées sans interrompre l'exécution du jeu.
- Modification de la gestion de l'affichage (`updateGraphic` et `callDisplay`) pour enlever un 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` (pas pressée)