

Rapport de projet : Bubble-Bobble

IHDC B132

Cédric Evrard

1 Introduction

L'object de ce projet est de réaliser notre propre version du jeu video Bubble-Bobble [1]. Ce jeu video sera réalisé à l'aide du langage de programmation C ainsi qu'avec la bibliothèque graphique OpenGL et de la bibliothèque GLUT.

La réalisation du jeu est divisé en 2 étapes, une premiere étape sera de réaliser le coeur du jeu, c'est à dire, un personne fonctionnel, la présence d'ennemis le tout dans un seul niveau. La deuxième partie sera libre et l'objectif de celle-ci sera d'améliorer de la meilleur des faons possible le jeu via, par exemple, l'ajout d'une intelligence artificielle pour les ennemis, la création de plusieurs ennemis, de plusieurs niveaux, ...

2 Projet

2.1 Présentation des écrans

2.1.1 Accueil

L'écran sera affiché à l'ouverture du programme. Sur cet écran (Figure 1), il y aura le titre du jeu ainsi qu'un menu de sélection permettant d'accéder aux autres écrans de l'application.

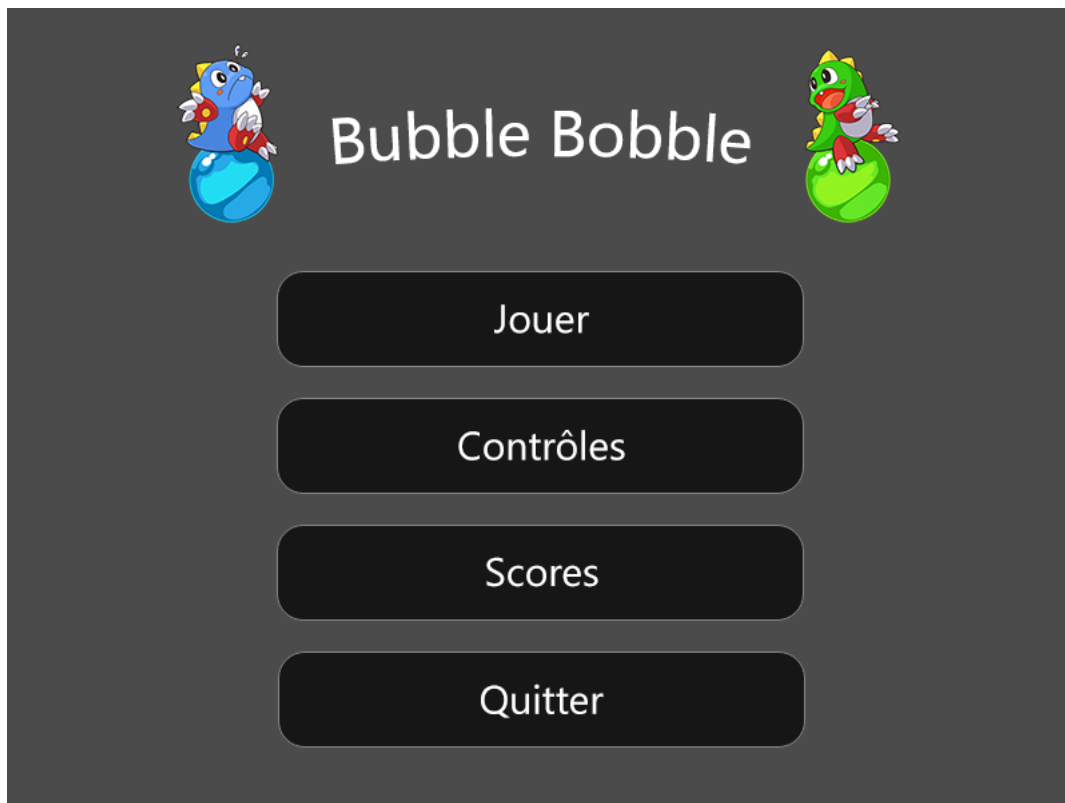


Figure 1: Écran d'accueil

2.1.2 Contrôles

Cet écran (Figure 2) sera affiché avant chaque partie afin de présenter au joueur les contrôles du jeu afin qu'il puisse facilement savoir comment jouer au jeu.

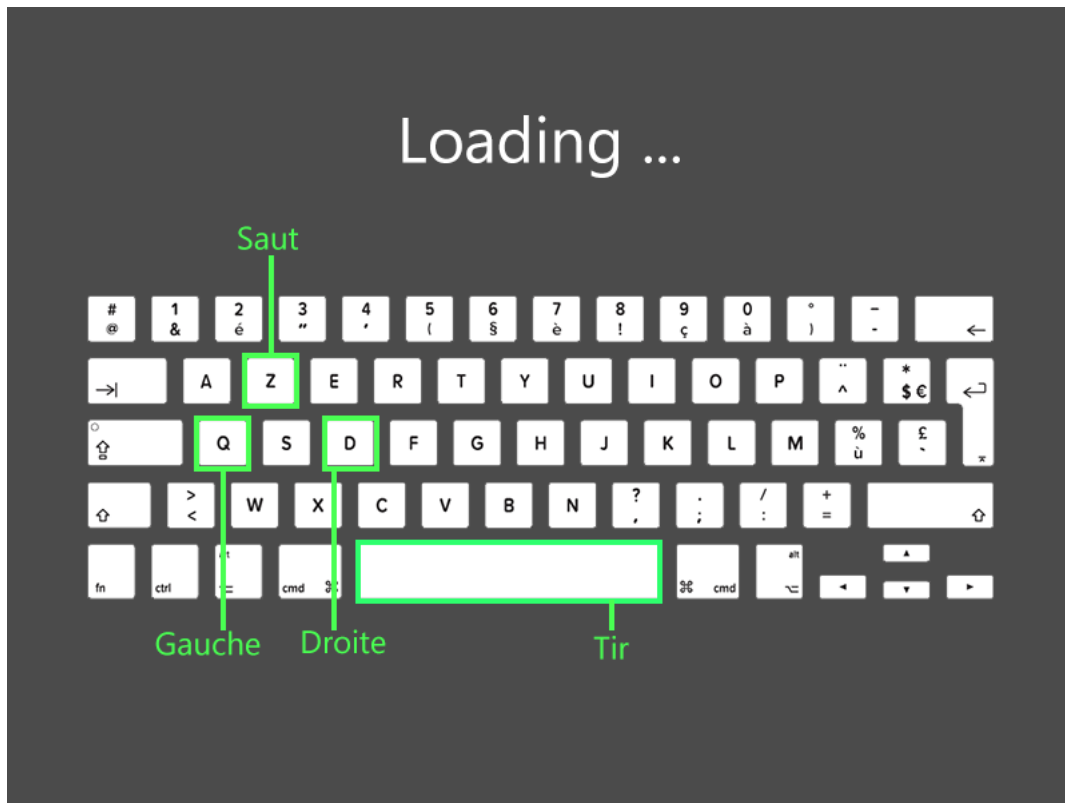


Figure 2: Écran des contrôles affichés au chargement

2.1.3 Jeu

Écran principal de l'application (Figure 3). Il présentera le niveau ainsi que certaines informations au joueur. Le score du joueur sera affiché en haut à gauche, le niveau en haut à droite et le nombre de vie, en bas à gauche.

2.2 Fonctionnement du coeur du jeu

Cette section d'écrira les différents éléments pour le fonctionnement du jeu. Comment le monde est représenté ou comment la détection entre deux éléments du jeu sera détecté.

2.2.1 Représentation de la carte

La carte sera divisé en une grille de 32 cellules de large sur 25 cellules de haut. Chaque cellule de cette grille sera un mur ou une zone vide. Ensuite, chaque bloc aura une taille de 25 pixels de large sur 22 pixels de haut.

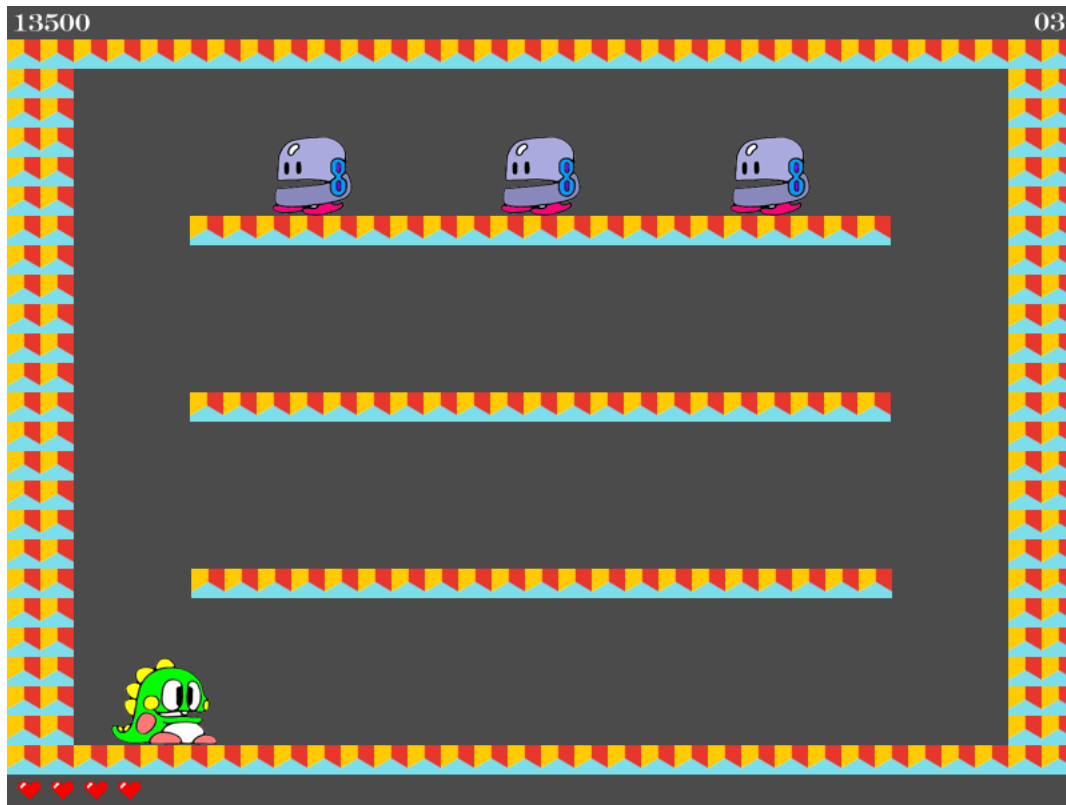


Figure 3: Écran de jeu

2.2.2 Représentation des personnages

Les personnages du jeu, que ça soit pour le joueur ou pour les ennemis, seront représentés à l'aide d'une image qui sera positionnée sur un axe x-y. Le personnage sera entouré d'une "hitbox" [2]. La hitbox sera un rectangle qui entour le personnage (voir exemple figure 4). Ce rectangle permettra de savoir quand deux éléments entre en collision. En effet, lorsque la position d'une "hitbox" rencontre une autre "hitbox", cela signifie que deux personnes se sont rencontrés.

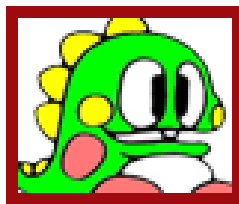


Figure 4: La "hitbox" est représentée en rouge

Le même système sera utilisé pour les bulles lancées par le joueur. Même si les bulles sont représentés via une image ronde, la "hitbox" de celles-ci seront aussi des rectangles afin de faciliter les calculs liés à la détection de "hit".

2.3 Structures utilisées

2.4 Découpage du code

3 Conclusion

La principale difficulté de ce projet va résider dans le fait d'optimiser la gestion des hitbox afin que l'application ne soit pas ralentie par la recherche de collision. En effet, à chaque tour de la boucle principale de GLUT, on va devoir vérifier si deux éléments sont entrés en collision, il faut donc optimiser au maximum cette gestion.

La deuxième difficulté sera l'utilisation de la bibliothèque GLUT. Celle-ci étant une bibliothèque ayant déjà un certain nombre d'années, il peut être difficile de trouver certaines documentations ou alors les exemples présentés pour OpenGL utilisent de nouvelles bibliothèques.

4 References

- [1] "Bubble Bobble - StrategyWiki," https://strategywiki.org/wiki/Bubble_Bobble.
- [2] "Hitbox - VALVe Developer Community," <https://developer.valvesoftware.com/wiki/Hitbox>.