



Projet synthèse d'image

FLAPIMAC

Manon SEBAOUN - Marion SEMINEL



Rapport – FLAPIMAC



Sommaire

| | |
|---|---|
| I - Présentation du projet | 3 |
| II - Fonctionnalités réalisées..... | 3 |
| III - Procédure de compilation | 4 |
| IV - Lancement du jeu | 4 |
| V - Modules complémentaires | 4 |
| VI - Fonctionnalité non implémentée | 4 |
| Screenshots..... | 5 |

I - Présentation du projet

FLAPIMAC est un jeu solo en 2D se jouant au clavier. Le joueur contrôle un vaisseau évoluant dans un monde rempli d'obstacles, d'ennemis et de bonus. Le but : éviter de percuter les obstacles et les ennemis et récupérer les bonus éparpillés dans les différents niveaux. Actuellement, le jeu comporte trois niveaux.

Le projet a été réalisé en langage C avec OpenGL et la librairie SDL.

Les différents éléments du jeu sont représentés par une forme texturée :



Vaisseau (joueur)



Ennemi



Projectile



Obstacle

II - Fonctionnalités réalisées

1° Chargement du niveau : chargement à partir d'une image au format PPM. Le programme récupère les informations du header puis lit chaque pixel. Si le pixel lu est rouge, vert ou bleu, il est affiché à l'écran. Sur une image, un pixel **rouge** représente un obstacle, un **vert** un ennemi et un **bleu** un bonus.

2° Déplacement du vaisseau : déplacement vertical uniquement à l'aide des touches fléchées haut et bas du clavier.

3° Tir de projectiles : possibilité de tirer des projectiles en appuyant sur la barre espace. Le nombre de tirs n'est pas limité.

4° Collisions vaisseau/obstacles/ennemis/projectiles : plusieurs fonctions ont été créées selon le type de collision (vaisseau/obstacle, projectile/ennemis etc). Le programme teste tout le long du jeu s'il y a collision ou non. S'il y a collision, la fonction renvoie 0, sinon 1. Selon le type de collision, le comportement du jeu change :

- **Collision vaisseau/obstacle, vaisseau/ennemi ou projectile ennemi/vaisseau** : arrête le jeu → game over. Le joueur est envoyé sur un écran Game Over.

- **Collision projectile vaisseau/obstacle** : si un projectile rentre en collision avec un obstacle, il est tout simplement détruit. Cela n'a aucun impact sur l'obstacle en revanche. Le jeu continue.
- **Collision projectile vaisseau/ennemi** : si un projectile rentre dans un ennemi, ils sont tous les deux détruits. Le jeu continue.

III - Procédure de compilation

Le programme s'exécute à partir d'un fichier Makefile qui fait notamment appel aux différentes librairies utilisées (SDL, SDL_image...) et crée l'exécutables du jeu. Pour compiler, il faut donc entrer la commande "**make**" dans le terminal.

IV - Lancement du jeu

Pour lancer le jeu, il suffit d'entrer la commande "**./bin/flapimac**" dans le dossier du jeu. Un écran d'accueil s'ouvre alors. Pour jouer, il faut cliquer sur "Start".

V - Modules complémentaires

1° Écran d'accueil avec menu : lorsque le programme est exécuté, le joueur a d'abord accès à un écran d'accueil duquel il peut lancer le jeu, lire les règles ou bien... quitter (mais ce serait dommage de quitter sans jouer !). L'interaction avec l'écran se fait grâce à la souris. Lorsque le joueur perd la partie, un écran de Game Over s'affiche et propose de nouveau le menu.

2° Score : le score est déterminé par le temps de survie du joueur. Plus il va loin dans le jeu, plus son score est élevé. Afin de calculer le score, nous avons utilisé une horloge.

3° Musique : ajout d'une musique d'ambiance.

4° Passage au niveau supérieur : le jeu comporte actuellement 3 niveaux. Lorsque le jeu arrive au bout d'un niveau, il lui est indiqué qu'il a gagné (avec une image dédiée) et qu'il peut accéder au niveau supérieur. Il a également la possibilité de refaire le niveau s'il le souhaite ou de quitter la partie.

VI - Fonctionnalité non implémentée

Bonus : nous avions fait en sorte que le programme détecte la présence de bonus mais nous n'avons finalement pas implanté ces derniers, le code n'étant pas encore parfaitement fonctionnel (notamment l'effet procuré par le bonus). L'effet souhaité était de rendre le joueur invincible pendant un laps de temps (donc s'il y avait collision entre le vaisseau et un obstacle/ennemi, le jeu continuait). D'autre part, nous avions un soucis de "maintien" de l'invincibilité : l'effet devait se déclencher si le joueur touchait le bonus (donc s'il y avait

collision et que la fonction renvoyait 0). Or, dès que la fonction ne renvoyait plus 0 (soit une fraction de seconde plus tard), l'effet disparaissait immédiatement au lieu de se maintenir pendant le laps de temps fixé.

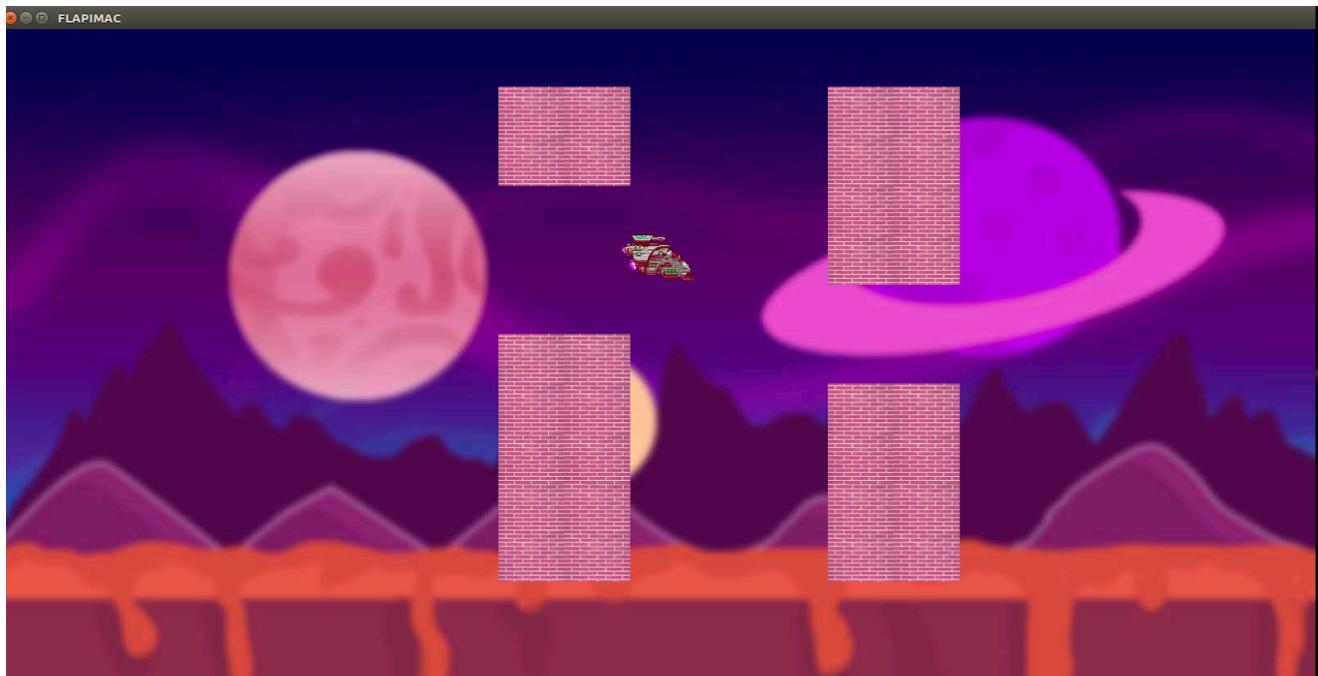
Screenshots



Ecran de démarrage du jeu



Ecran des règles



Aperçu du jeu (niveau 1)



Ecran - victoire (arrivée à la fin d'un niveau)



Ecran - Game Over