

# Projet Synthèse d'Image

## Beexel odyssey

### Utilisation

---

Le jeu se déplace automatiquement vers la fin du niveau. Le joueur peut se déplacer dans la fenêtre avec les touches directionnelles

"←↑↓→" ou "ZQSD", simultanément. Le joueur ne peut pas sortir de l'écran.

Le joueur débute avec un certain nombre de points de vie. Une fois à zéro, la partie est perdue au prochain dégât reçu.

Le joueur perd rapidement les points au contact du terrain. Il perd un point si il est touché par les attaques des ennemis, trois points si il contacte un ennemi directement.

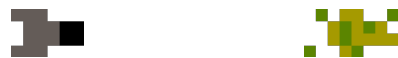
L'objectif du joueur est d'esquiver les ennemis, leurs attaques et le terrain jusqu'à la ligne d'arrivée. Il peut tirer des projectiles pour se défendre à l'aide de la touche "ESPACE". Les attaques du joueur détruisent les ennemis, mais ne pare pas leurs attaques.

Le niveau contient des bonus qui aident le joueur si il rentre dedans.

- Le bonus de santé ajoute un point de vie au joueur.
- Le bonus d'arme donne une chance sur 3 l'augmentation de projectiles par attaque, sinon il augmente la cadence de tir.
- Le bonus de vitesse multiplie la vitesse maximale du joueur.



*Joueur (gauche), Ennemi (droite)*



*Attaque du joueur (gauche),  
de l'ennemi (droite)*



*Bonus de santé, arme, vitesse*

## Compilation et Exécution

---

*Beexel odyssey* est développé pour et sur MacOS 10.13 et compatible avec Linux. La compilation requiert la librairie SDL 2 et SDL 2 image.

### **Compilation:**

Exécuter les commandes suivantes:

```
git clone https://github.com/piergabory/projet_rasterisation.git
cd ./projet_rasterisation
make clean
make
```

### **Exécution:**

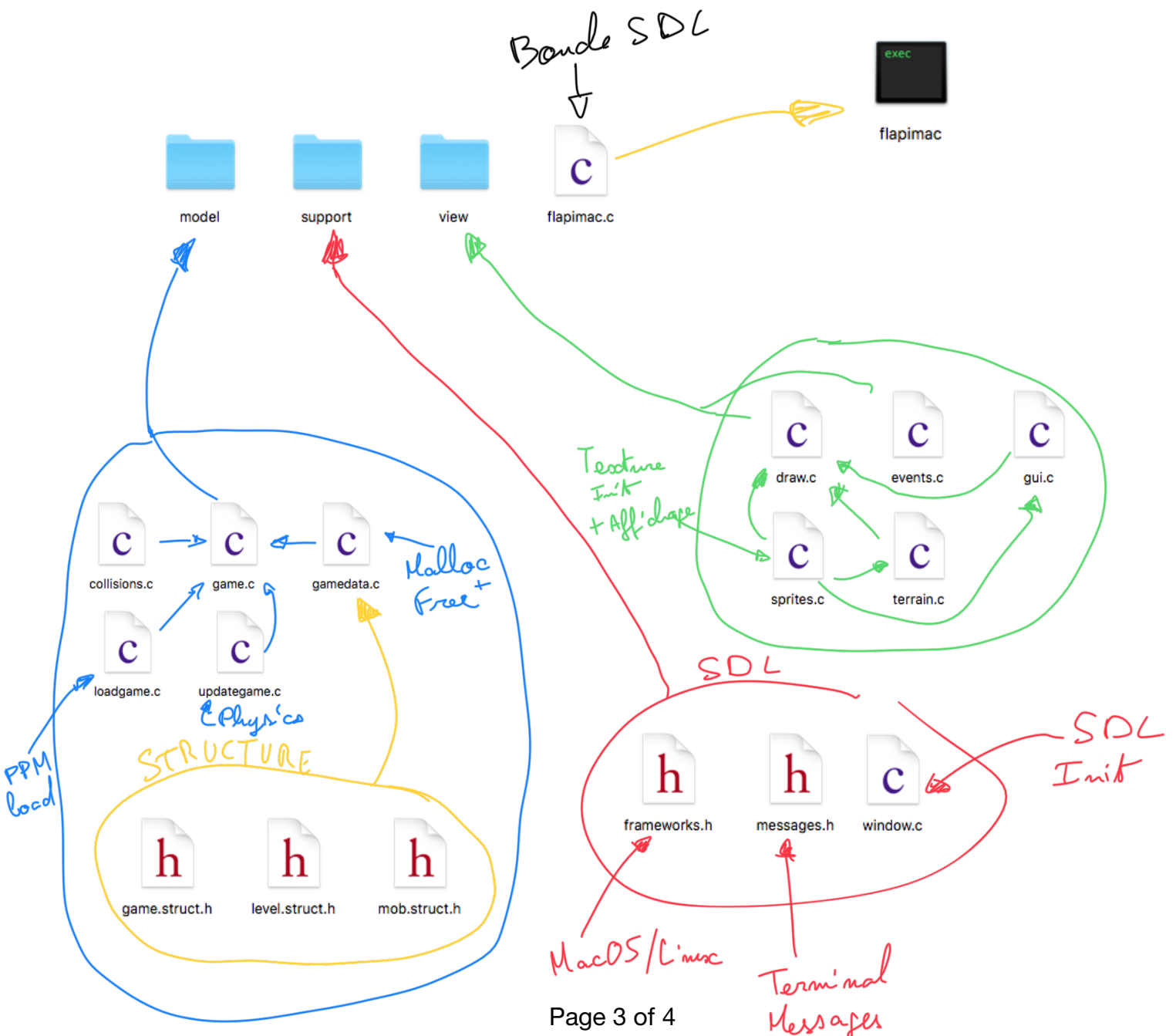
```
make run
```

## Fonctionnalisées réalisées

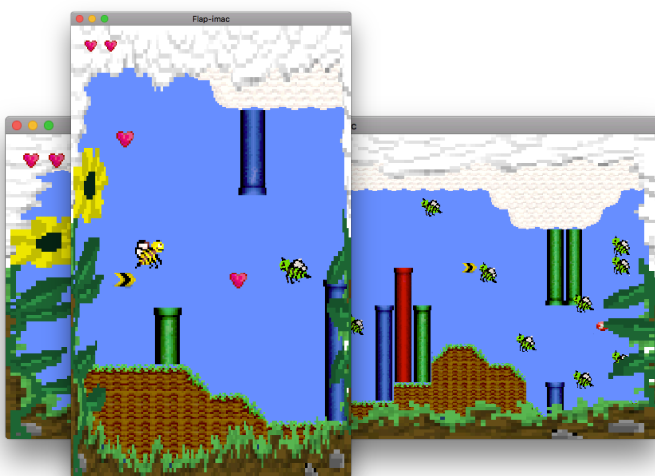
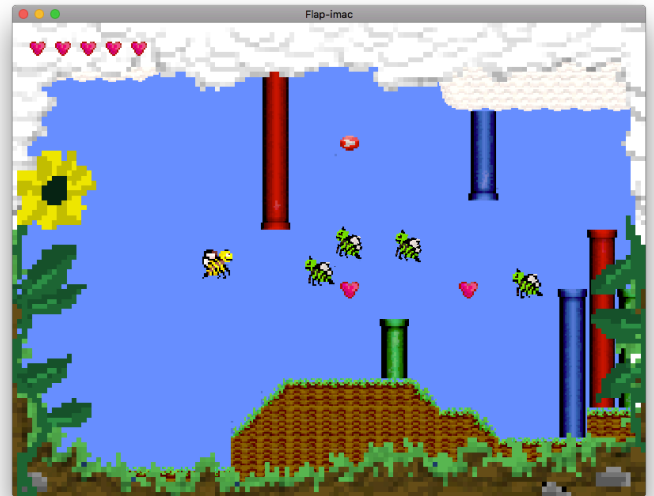
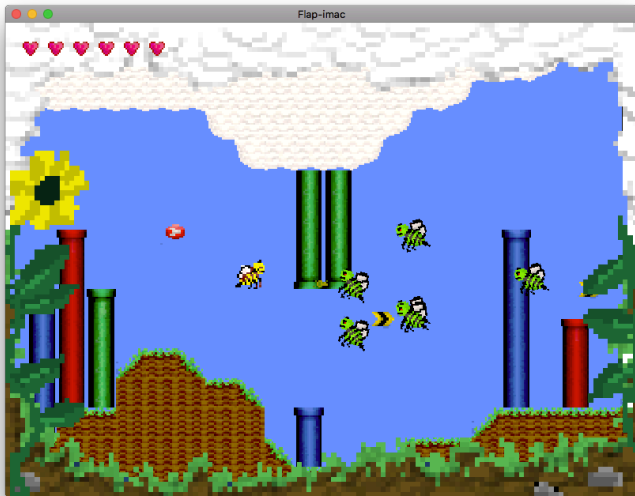
---

- Chargement d'un fichier de texture
- Chargement d'un fichier de carte de niveau en *portable pixel map*
- Déplacement du joueur dans le niveau (vertical/horizontal/diagonal)
- Déplacement des ennemis
  - Déplacement déterminé quand distant au joueur
  - Déplacement vers le joueurs à l'approche
- Projectiles lancés par le joueurs
- Projectiles lancés par les ennemies
- Suivis de la santé du joueur
- Bonus de santé, arme, vitesse
- Détection de la mort du vaisseau et de la fin du niveau (rangée de blocs)
- Écran de fin de jeu
- Fin du programme (touche "ESCAPE" à la fin d'une partie)
- Collisions avec le terrain
- Collisions avec les projectiles/ennemis/bonus
- Utilisation de la transparence
- Gestion de la re-dimension des fenêtres
- Différents mode de tirs de projectiles
- Animation des sprites

Potentiellement d'autres.



## Captures



redimensionnable