

BONNETAIN Baptiste (SDIA)
KAOUKAB Anas (RIO)

Rapport du projet de programmation avancée

Le projet était un remake du jeu connu « Asteroids » .
Il a été programmé en C++ en utilisant la bibliothèque SDL2.

Les bibliothèques requises sont

- SDL.
- SDL_image.
- SDL_ttf.

Implémentation du jeu :

❖ Vaisseau:

- Le vaisseau qui traverse un côté de l'écran revient par le côté opposé
- Le vaisseau a une barre de vie limitée, il meurt s'il percute trop d'astéroïdes.
- Mouvement du vaisseau : Accélération qui émet une petite flamme orange, Décélération si aucune accélération, Gravité et inertie prises en compte, Téléportation possible
- Un astéroïde qui entre en collision avec le vaisseau n'est pas détruit, le vaisseau bénéficie en revanche d'une courte période où il ne peut prendre des dégâts. Celle-ci se remarque puisque le vaisseau clignote alors.

❖ **Astéroïdes:**

- Nombre minimum d'astéroïdes
 - Vitesse et rotation aléatoire pour chaque astéroïde
 - Quand l'astéroïde quitte l'écran, il revient par le côté opposé de l'écran
 - Un astéroïde qui touche le vaisseau le blesse
 - Des astéroïdes de différentes tailles sont présents
 - Les tirs ne détruisent les astéroïdes que s'ils sont de taille minimale, sinon ils les scindent en deux astéroïdes de taille inférieure
-
- ❑ Un score est implémenté pour calculer la performance du joueur. A chaque fois qu'il détruit un astéroïde, son score augmente (l'augmentation du score dépend de la taille d'astéroïde).
 - ❑ Un medikit de la forme d'une croix rouge qui grandie/diminue est généré aléatoirement pendant le jeu, ce qui permet au joueur de récupérer de la santé.
 - ❑ Un fond d'écran a été ajouté ainsi qu'une musique pendant le jeu.
 - ❑ Deux menus sont présents, l'un avant de commencer et l'autre après la destruction du vaisseau, qui annonce la fin du jeu et le score réalisé.

Niveau 1

- un vaisseau capable de se déplacer sur l'écran et de tirer ✓
- à chaque tir, un projectile est émis depuis le vaisseau et se déplace à une vitesse supérieure à ce dernier ✓
- quand le projectile quitte l'écran, il est détruit
- la topologie de la zone de jeu est implémentée. Vous êtes libre de choisir la topologie que vous voulez. Un vaisseau qui s'approche du bord peut tout aussi bien revenir par le bord opposé ou se voir interdire le franchissement (et vous pouvez même imaginer d'autres scénarios). ✓

Niveau 2

- des astéroïdes traversent la zone de jeu ✓
- un astéroïde qui touche le vaisseau le blesse avant de s'auto-détruire
- le vaisseau a une barre de vie limitée, il meurt s'il percute trop d'astéroïdes ✓

Niveau 3

- les tirs détruisent les astéroïdes ✓
- des astéroïdes de différentes tailles sont présents ✓
- un niveau minimal d'astéroïdes est garanti ✓
- un astéroïde qui entre en collision avec le vaisseau n'est plus détruit, le vaisseau bénéficie en revanche d'une courte période où il ne peut prendre des dégâts ✓

Niveau 4

- les tirs ne détruisent plus les astéroïdes mais les scindent en deux astéroïdes de taille inférieure de moitié ✓
- lorsqu'un astéroïde est trop petit, il est automatiquement détruit. ✓

Au-dessus du niveau 4

Une fois à ce stade, vous pouvez ajouter des extensions (multi-joueur, plusieurs types de munitions, cohortes organisées d'astéroïdes, medikit, etc.)

Medikit ✓

Comment jouer ?

[HAUT] : Accélération dans le sens d'orientation du vaisseau

[GAUCHE] : Tourner à gauche

[DROITE]: Tourner à droite

[BAS] : Téléporter le vaisseau

[ESPACE] : Tirer

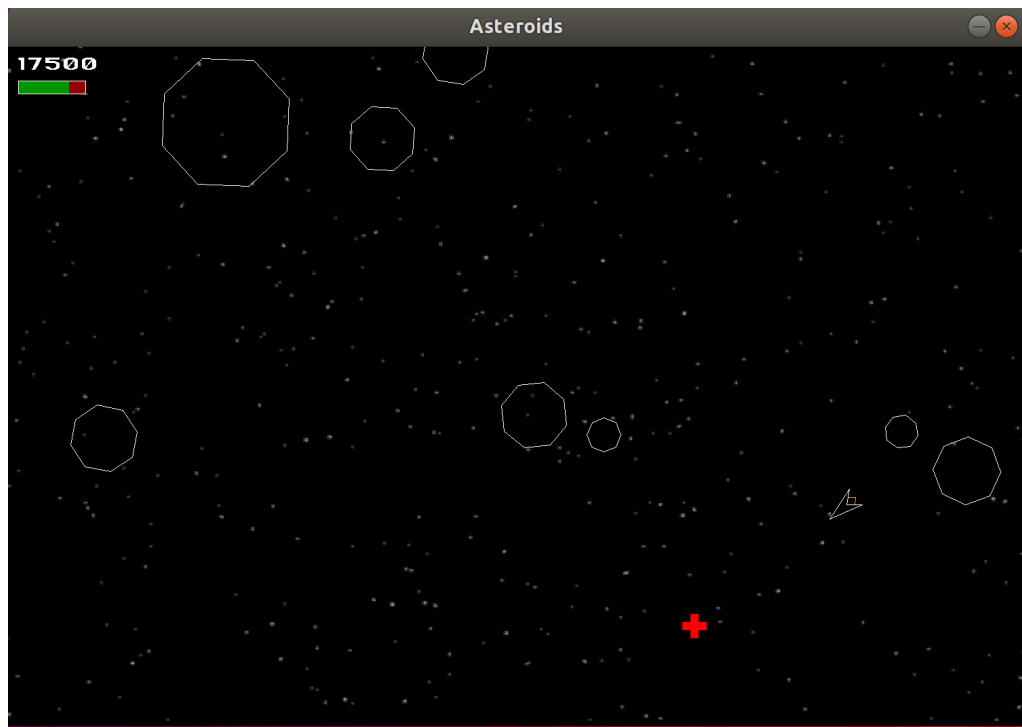
[ESC] : Quitter le jeu

Screenshot du jeu:

Interface d'entrée



Interface lors du jeu



Interface du fin de jeu:



Le diagramme de classe/UML:

