

P O D R A C E R

Architecture du projet

Le projet est divisé en plusieurs sous-projets et dossiers :

```
podracer
├── ai                # intelligence artificielle
├── assets            # ressources utilisées par la fenêtre du jeu
│   ├── fonts
│   └── textures
├── engine            # moteur de jeu
├── lib               # bibliothèques utilisées par le projet
├── shared            # contient les classes utilisées par ai, engine et window
└── window            # fenêtre de jeu
```

Répartition des tâches

Yohann JERRAIN : Implémentation de l'IA, bonus de gestion de la vie des pods

Axel COCAT : Classes messaging, implémentation de l'engine, fenêtre SFML, sauvegarde et import de fichiers de niveaux

Julien DUPONT : Génération du niveau

Fonctionnalités bonus

Import / export de niveaux

En plus du nombre de pods par joueur, on peut passer un argument optionnel d'import ou d'export au lancement du moteur de jeu.

Pour importer un fichier, on passe l'argument "<fichier", pour exporter on passe l'argument ">fichier".

Par exemple, avec `podracer-engine.exe 2 >mylevel.json`, le moteur effectuera une génération de carte et sauvegardera le résultat dans `mylevel.json`.

Si on lance ensuite `podracer-engine.exe 2 <mylevel.json`, le moteur chargera directement le niveau précédemment exporté.

Gestion de la vie des pods

Chaque pod a ses propres points de vie, ils commencent avec 100pv et perdent 5pv à chaque collision (avec un autre vaisseau ou avec un obstacle).

Lorsque les points de vie d'un pod arrivent à 0, il est considéré comme détruit et ne pourra plus bouger.

Les collisions sur les pods détruits sont désactivées.

Bibliothèques utilisées

- [SFML](#): gestion de la fenêtre et de ses événements. ([licence](#))
- [Json for Modern C++](#): sérialisation et désérialisation en JSON. ([licence MIT](#))