

# Rapport intermédiaire de Projet long

Mathieu Crocombette–Pasquet, Philippe Hinault

March 2024

## 1 Introduction

Le but du projet est de faire un jeu de tactique militaire en temps continu, les règles du jeu sont détaillées dans le fichier `rules.md` (à la racine du projet). La réussite du projet se mesure en évaluant à quel point le jeu est jouable, c'est à dire :

- L'interface graphique est-elle fonctionnelle ?
- Les unités peuvent-elles se déplacer / se battre / être posées sur le plateau ?
- Le terrain peut-il être généré aléatoirement ?
- Le jeu est-il fluide ? (moins de deux secondes entre chaque unité de temps)
- Le jeu peut-il terminer ?
- Un joueur peut-il affronter une IA ?

Nous réalisons le projet en Ocaml dans le but de monter notre niveau dans ce langage. Nous organisons notre travail en nous répartissant les tâches. Notre expérience pour ce projet se base sur des logiciels que nous avons déjà codés, soit : Un solveur de jeu solitaire (en Ocaml, avec lequel nous avons découvert le langage), un jeu Catane (en Java, où nous avons eu un plateau à gérer)

## 2 Implémentation

Notre code s'articule autour de modules dont le diagramme de dépendance correspond à la figure 1 2

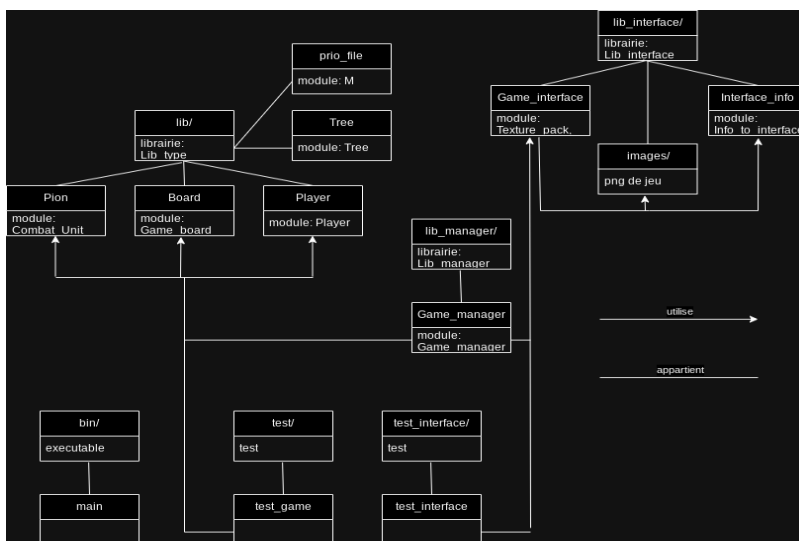


FIGURE 1 : Diagramme module.

Nos données sont représentées sous forme de type record. Des tables de Hachage (Module Map) sont utilisées pour implémenter des files de priorité et des graphes.

Nous utilisons certaines librairies externes :

- LWT pour les threads.
- YoJson pour la sérialisation
- Raylib pour l'interface graphique

Notons que nous avons des difficultés pour utiliser les librairies LWT et Raylib en raison de la pauvreté de leur documentation en Ocaml. Aussi, la gestion d'erreur de Raylib est inexistante.

### 3 Jalons

Nous avons pris du retard sur le planning initialement prévu. Le nouveau planning est donc celui de la figure 2 3

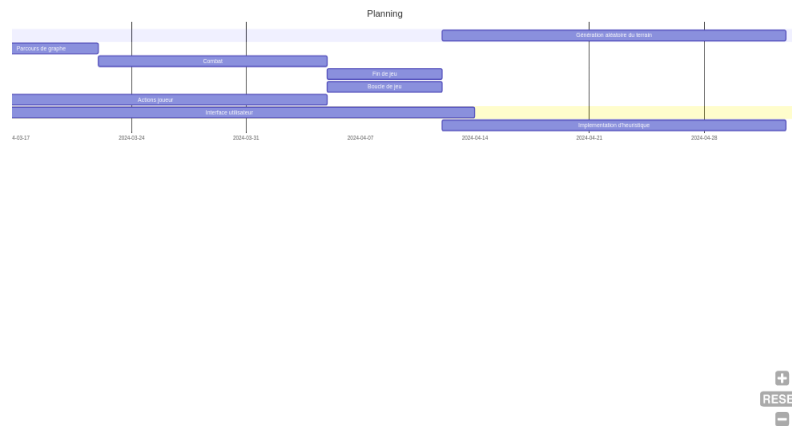


FIGURE 2 : Planning.

**Ce qui a été fait :**

- Les structures de données pour les pièces, les joueurs et le terrain
- L'affichage simple du jeu en interface graphique et terminal
- Les structures de communication entre l'interface et les contrôleurs du jeu

**Ce qui est en train d'être fait :**

- Les fonctions de parcours de graphe
- L'interface graphique

**Ce qu'il reste à faire :**

- La génération aléatoire du terrain (bruit de perlin)
- Les actions que peuvent faire les joueurs
- La boucle fluide du jeu
- L'implémentation de l'IA