

# Projet RPG Indiana Jones

---

Auteurs : Nathan LEPAGE & Antonin TERRASSON

## Table des matières

- [Projet RPG Indiana Jones](#)
  - [Table des matières](#)
  - [Présentation du projet](#)
  - [Contexte historique](#)
  - [Structure du code](#)
    - [main.rs](#)
    - [utils.rs](#)
    - [ui.rs](#)
    - [grid.rs](#)
    - [entity.rs](#)
    - [player.rs](#)
    - [monster.rs](#)
    - [combat.rs](#)
    - [item.rs](#)
    - [equipment.rs](#)
  - [Conclusion](#)
    - [Total des lignes de code : 1500 lignes](#)

## Présentation du projet

Bienvenue dans notre projet de jeu RPG inspiré par Indiana Jones !

Ce jeu vous plonge dans une aventure palpitante où vous incarnez un explorateur intrépide à la recherche d'un artefact caché dans un labyrinthe rempli de dangers.

Votre mission, si vous l'accepté est de naviguer à travers ce labyrinthe, collecter des objets et des équipements, combattre des monstres, et finalement atteindre l'artefact pour remporter la victoire.

## Contexte historique

Vous êtes Indiana Jones, l'explorateur légendaire, et vous avez entendu parler d'un artefact ancien caché dans un labyrinthe mystérieux. Ce labyrinthe est rempli de monstres redoutables. Armé de votre fouet et de votre chapeau emblématique, vous devez braver ces dangers pour récupérer l'artefact et prouver une fois de plus que vous êtes le plus grand aventurier de tous les temps.

Mais attention, chaque pas peut être votre dernier, et les monstres rôdent dans l'ombre, prêts à vous attaquer à tout moment.

## Structure du code

Le projet est structuré en plusieurs modules, chacun responsable d'une partie spécifique du jeu.

Voici une description détaillée de chaque fichier et de son rôle dans le projet :

## main.rs

- **Auteur : Antonin TERRASSON**
- **Longueur : 150 lignes**
- **Description :** Le fichier principal du jeu. Il initialise la grille de jeu et l'interface utilisateur, et lance les différents threads pour gérer les actions du joueur, les mouvements des monstres, et la régénération de la santé du joueur. Des canaux sont utilisés pour communiquer entre les différents threads et mettre à jour l'état du jeu en temps réel.

## utils.rs

- **Auteur : Antonin TERRASSON**
- **Longueur : 50 lignes**
- **Description :** Ce module contient des fonctions utilitaires pour le jeu. Il inclut des fonctions pour lire des entrées utilisateur depuis la console, telles que `read_number` pour lire un nombre et `read_key` pour lire un caractère. Ces fonctions sont utilisées pour interagir avec l'utilisateur et obtenir des entrées nécessaires au fonctionnement du jeu.

## ui.rs

- **Auteur : Nathan LEPAGE & Antonin TERRASSON**
- **Longueur : 200 lignes**
- **Description :** Ce module gère l'interface utilisateur du jeu. Il contient la structure `UI` qui maintient l'état de la carte, des équipements et des objets à afficher. Il fournit également des fonctions pour afficher des messages spécifiques comme le message de bienvenue, la demande de taille de la carte, et les messages de victoire ou de défaite.

## grid.rs

- **Auteur : Antonin TERRASSON & Nathan LEPAGE**
- **Longueur : 300 lignes**
- **Description :** Ce module gère la grille de jeu. Il contient la structure `Grid` qui maintient l'état de la grille, des murs, des monstres, des objets, et des équipements. Il fournit des fonctions pour initialiser la grille, placer les éléments aléatoirement, et gérer les interactions entre le joueur et les autres éléments du jeu. C'est dans ce fichier qu'est généré le aléatoirement labyrinthe.

## entity.rs

- **Auteur : Nathan LEPAGE**
- **Longueur : 100 lignes**
- **Description :** Ce module définit la structure de base `Entity` et le trait `EntityTrait` pour les entités du jeu. Il fournit des méthodes génériques pour gérer les entités comme obtenir leur nom, icône, description, points de vie, points d'attaque, position, et visibilité.

## player.rs

- **Auteur : Nathan LEPAGE & Antonin TERRASSON**

- **Longueur : 150 lignes**
- **Description :** Ce module gère le joueur. Il définit la structure **Player** et ses méthodes associées. Il permet de créer un joueur, gérer ses équipements et objets, et définir ses actions comme attaquer, utiliser des objets, et se déplacer.

## monster.rs

- **Auteur : Nathan LEPAGE & Antonin TERRASSON**
- **Longueur : 150 lignes**
- **Description :** Ce module gère les monstres du jeu. Il définit la structure **Monster** et ses méthodes associées, ainsi que le gestionnaire de monstres **MonsterManager**. Il permet de créer, gérer et manipuler les monstres dans le jeu.

## combat.rs

- **Auteur : Nathan LEPAGE & Antonin TERRASSON**
- **Longueur : 100 lignes**
- **Description :** Ce module gère les combats du jeu. Il définit la fonction **start\_combat** qui permet de gérer un combat entre le joueur et un monstre. Il prend en compte les actions du joueur comme attaquer, fuir, et utiliser des objets.

## item.rs

- **Auteur : Nathan LEPAGE**
- **Longueur : 100 lignes**
- **Description :** Ce module gère les objets du jeu. Il définit la structure **Item** et ses méthodes associées, ainsi que le gestionnaire d'objets **ItemManager**. Il permet de créer, gérer et manipuler les objets dans le jeu.

## equipment.rs

- **Auteur : Nathan LEPAGE**
- **Longueur : 100 lignes**
- **Description :** Ce module gère les équipements du jeu. Il définit la structure **Equipment** et ses méthodes associées, ainsi que le gestionnaire d'équipements **EquipmentManager**. Il permet de créer, gérer et manipuler les équipements dans le jeu.

## Conclusion

Total des lignes de code : 1500 lignes

Nous espérons que vous apprécierez jouer à ce jeu autant que nous avons apprécié le développer.

Bonne chance, et que l'aventure commence !

Nathan & Antonin