

# Projet : Jeu de RPG textuel

## Introduction :

Dans ce jeu de rôle textuel, les joueurs choisiront un personnage (**guerrier**, **mage**, **archer**, etc.) et affrontent des ennemis. Chaque personnage aura une stratégie d'attaque différente et unique. Le jeu sera structuré pour utiliser les patterns **Singleton**, **Factory**, et **Strategy**.

## *Objectifs pédagogiques :*

- Comprendre et implémenter le pattern **Singleton** pour gérer l'état global du jeu.
- Utiliser le pattern **Factory** pour créer différents types de personnages.
- Appliquer le pattern **Strategy** pour définir des comportements d'attaque variables pour les personnages.

## Exigences du projet :

### *1. Singleton Pattern :*

- Créer une classe `GameManager` qui utilise le pattern Singleton pour gérer l'état global du jeu (comme les scores, les tours, ou tout autre état global).
- Assurer qu'il ne peut y avoir qu'une seule instance de `GameManager` pendant le jeu.

### *2. Factory Pattern :*

- Créer une classe `**CharacterFactory**` qui sera responsable de la création des différents personnages (**Guerrier**, **Mage**, **Archer**).
- Chaque type de personnage doit avoir des propriétés spécifiques (par exemple, points de vie, attaque, défense) et doit être créé par la `**CharacterFactory**`.

### 3. *Strategy Pattern* :

- Définir une interface **`AttackStrategy`** avec une méthode `attack`.
- Implémenter différentes stratégies d'attaque pour chaque type de personnage (par exemple, **`SwordAttack`** pour le guerrier, **`MagicAttack`** pour le mage, **`BowAttack`** pour l'archer).
- Chaque personnage doit pouvoir changer sa stratégie d'attaque à la volée (par exemple, un mage pourrait utiliser une attaque magique ou une attaque physique faible).

## Étapes du projet

### 1. *Initialisation du jeu* :

- Créer une instance unique de **`GameManager`**.
- Utiliser **`CharacterFactory`** pour créer des personnages et les ajouter à l'état global du jeu via **`GameManager`**.

### 2. *Choix du personnage* :

- Demander au joueur de choisir un personnage (***guerrier, mage, archer***).
- Créer le personnage choisi en utilisant **`CharacterFactory`** et assigner une stratégie d'attaque initiale.

### 3. *Simulation de combats* :

- Simuler des tours de combat où le personnage utilise sa stratégie d'attaque actuelle.
- Permettre au joueur de changer de stratégie d'attaque entre les tours.

### 4. *Gestion de l'état du jeu* :

- Utiliser **`GameManager`** pour suivre les statistiques du jeu, comme les points de vie restants, les ennemis vaincus, etc.

## Prerequis avancé :

- Prendre en compte la rareté de l'attaque pour l'ennemie
- faire 10 combats
- Avoir un ennemi différents à chaque étage
- Mettre un Heal en place
- Historique des combats avec ***GameManager***