

Projet : Programmer le prototype d'un RPG

On vous demande de programmer le prototype d'un RPG (Role Player Game : jeu de rôle en français). Le jeu devra être jouable en ligne de commande avec une interface permettant à l'utilisateur de créer son personnage et de lui faire effectuer un certain nombre d'actions. Toutes les interfaces seront réalisées par un menu textuel. Il n'est pas demandé de faire des interfaces graphiques.

Le joueur incarne un héros explorant un donjon divisé en salles. Dans chaque salle, il doit combattre des PNJ (Personnage non joueur). Le jeu proposera deux thèmes (médiéval, futuriste), chacun influençant les ennemis, les objets et l'ambiance du donjon.

Pour réaliser ce projet, vous devrez utiliser le langage Java et faire appel à vos connaissances des designs pattern pour la phase de conception.

Votre code suivra une architecture MVC de manière à répartir le code en trois packages.

Le package vue contiendra une seule classe Ihm permettant d'interagir avec l'utilisateur et d'afficher les informations.

Le package controleur contiendra la classe Controleur qui sera en charge du déroulement du jeu et qui interagira avec les classes du modèle et la classe Ihm.

Le package modele contiendra les classes du modèle.

Vous ajouterez une classe Main qui contiendra le main.

Ce projet est à réaliser en binôme ou en trinôme. On vous demande d'enregistrer les noms des étudiants de votre groupe dans le wiki nommé Constitution groupes projet correspondant à votre groupe avant le 28 Novembre. Pour modifier le tableau du wiki, il faut cliquer sur le wiki puis ouvrir la liste déroulante indiquant « Afficher » et sélectionner « Modifier ».

Vous pouvez constituer votre binôme ou trinôme avec des étudiants d'un autre groupe que le votre. Inscrivez votre binôme ou trinôme dans un des groupes. Nous réorganiserons les groupes pour les équilibrer si besoin.

Le travail sera à réaliser en trinôme selon le calendrier suivant :

1. Présentation et rendu de la première itération du projet le 18/12/25. Dépôt du travail sur CELENE.
2. Présentation et rendu de la seconde itération du projet le 05/01/26. Dépôt du travail sur CELENE.

Le sujet de la seconde itération sera transmis ultérieurement.

Pour la première itération, la notation prendra en compte :

1. L'analyse avec le diagramme de cas d'utilisation et rédaction des scénarios et un diagramme de classe décrivant la modélisation du domaine.
2. Un document expliquant les design patterns utilisés et leur justification.
3. Un diagramme de conception intégrant les design patterns.
4. La mise en place du jeu correspondant à la première itération.

Si vous utiliser une IA générative, vous devrez expliquer précisément pour quelle partie et de quelle manière vous l'avez utilisée.

I.Travail à réaliser pour l'itération 1

a. Initialisation

Au lancement du programme, il sera demandé au joueur de créer son personnage. Pour cela, il devra choisir une classe de héros parmi un certain nombre disponible (barbare, sorcier...). Il devra également choisir un nom. Sa classe détermine son équipement initial et la manière dont il attaque.

Le personnage est créé avec des caractéristiques déterminées par la classe à laquelle il appartient (par exemple un(e) barbare aura plus de points de force qu'un(e) sorcier(e), mais moins de points d'intelligence, ou moins de dextérité qu'un(e) assassin(e)).

Chaque classe détermine :

- Les points de vie initiaux.
- **Les attributs de base** (force, dextérité, constitution, intelligence).
 - **Force** : Influence les dégâts physiques infligés.
 - **Dextérité** : Influence la précision et la réduction des dégâts subis.
 - **Constitution** : Influence les points de vie maximaux et la réduction des dégâts subis.
 - **Intelligence** : Influence les dégâts magiques infligés.
- Le type d'attaque

Classe	Points de Vie	Force	Dextérité	Constitution	Intelligence	Type attaque
Barbare	150	20	10	15	5	Coup de hache
Sorcier	100	5	8	10	25	Sort de feu
Archer	120	12	18	12	10	Tir à l'arc
Assassin	110	15	20	10	8	Coup de dague furtif

Thème 1 : Médiéval

PNJ possibles :

- **Chevalier Errant** : Guerrier puissant, armé d'une épée et d'un bouclier.
- **Sorcière** : Lance des sorts magiques à distance.
- **Vautour** : Rapace agressif, attaque en piqué.
- **Rat Enragé** : Petit mais rapide, attaque en groupe.
- **Gobelin** : Créature malicieuse, utilise des dagues et des pièges.

Thème 2 : Futuriste

PNJ possibles :

- **Soldat Cybernétique** : Armure renforcée, arme à énergie.
- **Robot Sentinelle** : Détecte et attaque les intrus avec des lasers.
- **Alien** : Créature extraterrestre, attaque avec des griffes acides.
- **Drone de Combat** : Petit et rapide, tire des missiles.

Caractéristiques des PNJ

Chaque PNJ a :

- **Des points de vie (PV).**
- **Des attributs :**
 - **Force** : Dégâts physiques.
 - **Dextérité** : Chance d'esquiver ou de frapper en premier.
 - **Constitution** : Résistance aux dégâts.
 - **Intelligence** : Puissance des attaques magiques (si applicable).
- **Une attaque spéciale** (ex : la sorcière lance un sort de feu, le vautour fond sur le joueur).

Exemple de PNJ : Chevalier Errant

- **PV** : 120
- **Force** : 15
- **Dextérité** : 8
- **Constitution** : 12

- **Attaque** : "Coup d'épée" (dégâts = force + 1d6).

1d6 : correspond au résultat du lancé d'un dé à 6 faces.

Exemple de PNJ : Drone de Combat (Futuriste)

- **PV** : 60
- **Force** : 5
- **Dextérité** : 15
- **Constitution** : 5
- **Attaque** : "Tir de missile" (dégâts = dextérité + 2 , ignore 50 % de la défense).

b. Interface de jeu

Une fois initialisé, le programme entre dans une nouvelle interface de jeu qui permet de choisir entre les deux actions suivantes :

- **Afficher l'inventaire du personnage**, c'est-à-dire tous les objets qu'il possède.

Certains sont donnés dès le démarrage du jeu, d'autres peuvent être acquis durant le jeu. Parmi ces objets, il peut y avoir des aliments ou des potions, qui peuvent être consommés, et de l'équipement (armure, casque, chaussures, arme, . . .). On notera que les objets seront différents suivant le thème choisi que ce soit les objets consommables et les objets de son équipement. Le joueur peut utiliser ces objets de deux façons :

- Consommer un objet consommable (manger une pomme, boire une potion)
- S'équiper avec un objet de son inventaire. Attention, le joueur ne peut porter qu'un casque, qu'une paire de chaussures, qu'une arme à la fois, même si son inventaire en contient plusieurs. C'est-à-dire que parmi les chaussures dont il dispose, il y a (au plus) une paire qu'il porte.

- **Se rendre dans le donjon pour se battre.**

c. Combat

Une fois rentré dans le donjon, le personnage doit traverser un certain nombre de salles et les débarrasser de tous les PNJ qui s'y trouvent.

Structure du Donjon

- Le donjon est composé de **10 salles**.
- Chaque salle peut contenir :
 - **1 à 4 ennemis** (générés aléatoirement selon le thème).
 - **0 à 2 objets** (potions, aliments).

Types d'objets pour le thème médiéval

Potions

- Potion de soin : Restaure 30 points de vie.
- Potion de force : Augmente la force de +5 pour 3 tours.
- Potion de résistance : Réduit les dégâts subis de 20% pour 2 tours.

Aliments

- Pomme : Restaure 10 points de vie.
- Pain : Restaure 15 points de vie.
- Viande séchée : Restaure 20 points de vie.

A vous d'imaginer les types de potions et d'aliments pour le thème futuriste.

Quand il entre dans une salle, l'application affiche les ennemis et les objets présents et propose d'attaquer un ennemi ou de ramasser un objet ou de consulter l'inventaire. Après avoir consulté l'inventaire, il pourra consommer un objet ou s'équiper avec un objet de son inventaire.

Exemple d'affichage :

```
=== Salle 1 ===  
Un Chevalier Errant et 2 Rats Enragés apparaissent !  
Vous trouvez : une Potion de soin et une Pomme.
```

```
Que voulez-vous faire ?  
1. Attaquer un ennemi  
2. Ramasser un objet  
3. Afficher l'inventaire
```

Génération Aléatoire des PNJ dans les salles

Chaque salle du donjon contient un nombre aléatoire de PNJ (entre 1 et 4), générés en fonction du thème choisi. Les PNJ sont sélectionnés aléatoirement parmi ceux disponibles dans le thème, avec des probabilités différentes pour varier les rencontres.

Exemple pour le thème médiéval :

- Chevalier Errant : 30% de chance.
- Sorcière : 25% de chance.
- Vautour : 20% de chance.
- Rat Enragé : 15% de chance.
- Gobelin : 10% de chance.

Système de Combat

Déroulement d'un Combat

Le joueur entre dans une salle et affronte les ennemis présents. Les ennemis n'attaquent que si le joueur les provoque en les attaquant. Le joueur peut se défendre pour réduire les dégâts subis.

Tous les PNJ attaquent le joueur un par un si le joueur a attaqué un des PNJ.

Le combat continue jusqu'à ce que :

- Tous les ennemis soient vaincus (le joueur passe à la salle suivante).
- Le joueur soit vaincu (fin de la partie).

Pendant le combat, et à chaque coup porté ou reçu, l'interface en ligne de commande doit afficher précisément l'évolution de l'état de santé de tous les participants au fur et à mesure.

Pendant le combat, à son tour, le joueur peut :

- Attaquer un ennemi.
- Utiliser un objet (potion, aliment). Quand le joueur choisit d'utiliser un objet pendant le combat, l'impact sur son état s'affichera.

Calcul des Dégâts

Les dégâts infligés entraînent une réduction des points de vie de l'ennemi et dépendent de :

- **L'attaque du personnage** (basée sur sa classe).
- **Les attributs de l'attaquant et du défenseur.**

Formule de Base

- **Attaque physique** : Dégâts = (Force de l'attaquant + Bonus de l'arme) - (Constitution du défenseur / 2)
- **Attaque magique** : Dégâts = (Intelligence de l'attaquant + Bonus du sort) - (Constitution du défenseur / 3)

Exemple pour un Barbare

Grom (Barbare) attaque un Chevalier Errant :

- Force de Grom : 20
- Bonus de la hache : +10
- Constitution du Chevalier : 12
Dégâts = (20 + 10) - (12 / 2) = 30 - 6 = 24

Défense et Réduction des Dégâts

Quand le joueur est attaqué, il se défend (pour simplifier on pourra considérer que sa défense correspond à sa dextérité + 1d6).

Il pourra réduire les dégâts initiaux grâce à ses attributs :

- **Réduction des dégâts** = (Dextérité + Constitution) / 4

- **Dégâts subis** = $\text{Max}(1, \text{Dégâts initiaux} - \text{Réduction})$

Exemple d'Utilisation

```
Que voulez-vous faire ?
1. Attaquer un ennemi
2. Utiliser un objet
Choix : 2
Inventaire :
1. Potion de soin
2. Pomme
Choisissez un objet : 1
```

Grom utilise une Potion de soin et récupère 30 PV.

Fin du Donjon

- Après avoir traversé toutes les salles, le joueur affronte un boss final (ex : un Dragon pour le thème médiéval, à vous de choisir le boss final du thème futuriste).
- S'il vainc le boss, il termine le donjon avec succès.

d. Visualisation des salles du donjon et des personnages

Quand le joueur entre dans une pièce du donjon, on doit pouvoir visualiser le joueur, les objets présents dans la pièce ainsi que tous les PNJ. Vous pourrez représenter chaque élément avec une couleur et une lettre. Voici trois exemples de représentation que vous pouvez utiliser :



Personnage



Chevalier errant



Sorcière

IV - Quelques indications pour la couleur

Pour afficher de la couleur dans la plupart des consoles, voici quelques informations.


Vous pouvez mettre de la couleur dans la sortie standard en utilisant les valeurs ci-dessous :

```
// Strings for foreground colors
```

```
public static final String ANSI_RESET = "\u001B[0m";
public static final String ANSI_BLACK = "\u001B[30m";
public static final String ANSI_RED = "\u001B[31m";
public static final String ANSI_GREEN = "\u001B[32m";
public static final String ANSI_YELLOW = "\u001B[33m";
```

```
public static final String ANSI_BLUE = "\u001B[34m";
public static final String ANSI_PURPLE = "\u001B[35m";
public static final String ANSI_CYAN = "\u001B[36m";
public static final String ANSI_WHITE = "\u001B[37m";

// Strings for background colors
public static final String ANSI_BLACK_BACKGROUND = "\u001B[40m" ;
public static final String ANSI_RED_BACKGROUND = "\u001B[41m";
public static final String ANSI_GREEN_BACKGROUND = "\u001B[42m";
public static final String ANSI_YELLOW_BACKGROUND = "\u001B[43m";
public static final String ANSI_BLUE_BACKGROUND = "\u001B[44m";
public static final String ANSI_PURPLE_BACKGROUND = "\u001B[45m";
public static final String ANSI_CYAN_BACKGROUND = "\u001B[46m";
public static final String ANSI_WHITE_BACKGROUND = "\u001B[47m";
```

Voici un petit exemple qui affiche 

```
System.out.println(ANSI_PURPLE_BACKGROUND+ANSI_WHITE+"Hello
world !!!"+ANSI_RESET);
```

Pour plus d'informations sur les couleurs : https://en.wikipedia.org/wiki/ANSI_escape_code#Colors