

Projet Java

Porte / Monstre / Trésor

But : le but de ce projet est de créer un jeu de rôle de type [Donjons et Dragons](#) en Java.

Étape 1 : Entrons dans le labyrinthe...

La première étape doit permettre à un utilisateur de se déplacer dans un labyrinthe (ou souterrain ou donjon...). Lors du lancement, l'utilisateur devra fournir le nom de son personnage.

Le labyrinthe est composé de :

- Pièces (l'une d'elle étant l'entrée du labyrinthe) ;
- Portes (reliant les pièces entre-elles) ;

Pour simplifier, les pièces ne seront connectées qu'en damier et donc on aura, au plus, une porte au nord, au sud à l'est et à l'ouest. L'utilisateur pourra donc utiliser des commandes textuelles (N, S, E, O) pour se déplacer.

Plan de travail :

- Lire le sujet jusqu'au bout ;
- Concevoir l'application avec UML ;
- Programmer la première version du logiciel ;

Étape 2 : Et voilà les monstres et les combats...

Pour rendre le jeu plus intéressant on introduit des monstres et des combats lors de cette étape. A chaque fois que le personnage entre dans une pièce, un monstre apparaît et attaque ce personnage. Le combat se déroule en plusieurs tours, chacun durant 1 seconde. A chaque tour, le personnage ou le monstre perd un point de vie (aléatoirement). Au départ, les personnages ont 10 points de vie et les monstres 5. Le personnage pourra à chaque tour de combat choisir de continuer ou de fuir.

Un combat se termine si le personnage fuit ou si l'un des deux protagonistes se retrouve à 0 points de vie. Le participant qui atteint 0 points de vie meurt (le personnage ou le monstre disparaît).

A l'issue d'un combat, le vainqueur gagne 1 point de vie maximum supplémentaire (pour faire simple). D'autre part, dès qu'il n'y a plus de combat dans la pièce, on considère que le personnage ou le monstre encore en vie regagne tous ses points de vie.

Étape 3 : Où l'on découvre des Trésors...

Lorsqu'un personnage gagne un combat, il obtient désormais un trésor. Ce trésor peut être une armure, une arme ou un nombre aléatoire de pièces d'or.

Une armure possède un nombre de points de protection fixe qui diminue à chaque fois qu'un monstre tape le personnage. Lorsque ce nombre de points de protection tombe à 0 c'est la vie du personnage qui diminue. On pourra trouver des armures possédant de 1 à 5 points de protection.

Une arme permet de faire perdre plus d'un point de vie par tour au monstre. On pourra trouver des armes faisant de 1 à 5 points de dommage.

Étape 4 : Où l'on dépense son Or...

On va désormais ajouter des boutiques à l'entrée du labyrinthe. Celles-ci vendront des armes, des armures et des potions permettant de récupérer des points de vie. On pourra aussi y faire réparer son armure pour un prix inférieur au prix d'achat de l'armure. Exemple : une armure à 5 points de protection pourrait coûter 500 pièces d'or et pourrait être réparée pour 250 pièces d'or.

Étape 5 : Gestion de la Persistance

On veut désormais gérer la persistance : si un joueur quitte le jeu et, plus tard, redémarre son logiciel il doit apparaître au même endroit avec le même nombre de points de vie, le même nom et le même équipement (pièces d'or, armure, arme...). Il faut donc stocker ces informations dans un fichier.

Note : toute fonctionnalité supplémentaire (IHM, 3D, déplacement des monstres, intelligence artificielle, autres trésors, boucliers, portes secrètes ou verrouillées, sortilèges, classes de personnages, règles complètes de Donjons&Dragons™ ...) ne sera prise en compte dans la notation que si toutes les étapes ont été correctement traitées.

Organisation du projet et travail attendu par trinôme :

Partie I : Conception de l'application

Concevez l'application permettant de répondre aux spécifications précédemment énoncées.

Constituez un dossier qui comportera les diagrammes UML adéquats exprimant le résultat de l'analyse de cette application (au minimum : cas d'utilisation, diagrammes de séquence détaillés et de classes).

Partie II : Développement d'une maquette de l'application

Une maquette de la solution envisagée sera développée. Celle-ci devra donner une image fidèle de la solution réelle qui pourrait être diffusée.

Partie III : Évaluation

Des séances d'évaluation régulières seront organisées pour suivre la réalisation de votre projet. A chaque séance, chaque groupe devra présenter son projet et faire une démonstration à partir d'un jeu de tests. Ce même jour, vous devrez rendre le dossier de conception mis à jour sous forme de rapport **PDF** dans lequel vous rajouterez quelques éléments clés de programmation. De plus, vous devrez rendre les **sources courantes du projet évalué**. Ces différents fichiers devront être déposés sur le moodle de l'Université.