

Projet 1 - Chasse au trésor

Objectif :

L'objectif de ce TP est de programmer un jeu simple en utilisant la programmation orientée objet.

1 Principes du jeu

Nombre de joueurs : 1

Matériel :

- ▷ Une carte du monde correspondant à une grille rectangulaire (dimensions paramétrables).
- ▷ Un héros que l'on déplace de cases en cases sur la grille.
- ▷ Des monstres posés aléatoirement sur la carte du monde.

But du jeu : Trouver le trésor et rester en vie.

Règles :

- ▷ Le héros commence en case (0 ; 0).
- ▷ Le joueur déplace le héros à chaque tour de jeu sur une case adjacente.
- ▷ Des monstres sont présents sur chaque case du monde avec une probabilité de 30
- ▷ Un trésor est placé sur une case du jeu. Un monstre est forcément sur la case.

2 Cadrage du code

Le code sera structuré en deux classes d'objets : la carte du monde et les personnages (monstres ou héros).

Voici les éléments de base de chaque classe :

2.1 Classe Personnage

1. Attributs

- ▷ Nom
- ▷ Position
- ▷ Points de vie

2. Méthodes

- ▷ Déplacement sur une case adjacente
- ▷ Attaque d'un autre personnage

2.2 Classe Monde

1. Attributs

- ▷ Largeur
- ▷ Hauteur
- ▷ Grille

2. Méthodes

- ▷ Initialisation de la carte du monde (mise en place des monstres et du trésor, lancement d'une partie)
- ▷ Visualisation de la carte du monde
- ▷ Déroulement d'une partie
- ▷ Combat entre deux personnages

3 Précisions sur le jeu

3.1 Vie des personnages et combats

Pour simplifier au maximum au départ :

- ▷ Le joueur démarre avec 100 points de vie.
- ▷ Tout monstre démarre avec 20 points de vie.
- ▷ Un combat est une succession d'attaques alternées entre le héros et un monstre.
- ▷ C'est forcément le héros qui engage le combat.
- ▷ Une attaque enlève aléatoirement de 1 à 5 points de dégâts.

3.2 Visualisation du monde

Pour simplifier, on pourra représenter le monde comme une grille du type :

1						
2		_		_		_
3		_		_		X
4		_		_		_

où X est la position du héros.

Les monstres restent invisibles et ne sont détectés que lorsque le héros arrive sur leur case. Le combat s'engage alors forcément.

3.3 Lancement du jeu

La création de l'objet monde peut lancer le jeu. Il suffit pour cela d'ajouter l'instruction `self.jeu()` à la méthode `__init__` pour la classe `Monde`.

```
Martin inflige 3 points de dégâts à Monstre.
Il reste 2 points de vie à Monstre.
Monstre inflige 4 points de dégâts à Martin.
Il reste 26 points de vie à Martin.
Martin inflige 4 points de dégâts à Monstre.
Vous avez gagné le combat ! Il vous reste 26 points de vie.

|_|_|_|_|_|
|_|X|_|_|_|
|_|_|_|_|_|
Déplacements possibles : ['G', 'D', 'H', 'B']
Vers où voulez-vous vous déplacer ? Gauche = G, Droite = D, Haut = H, Bas = B
Déplacement impossible
Vers où voulez-vous vous déplacer ? Gauche = G, Droite = D, Haut = H, Bas = B
```

FIGURE 1 – Exemple de rendu dans la console Python d'un extrait de partie

3.4 Pour aller plus loin

- ▷ Sur 3 cases sans monstres, placer une potion qui rajoute 5 points de vie au héros.
- ▷ Sur 2 cases avec monstres obligatoires, équipements qui améliorent l'attaque ou la défense du héros de +2 dans les combats. (on pourra pour cela rajouter des attributs `arme` et `armure` à la classe `Personnage`).
- ▷ Rendre les monstres dynamiques (ils se déplacent sur une case adjacente à chaque tour)
- ▷ Créer les attributs `Points d'attaque` et `Points de défense` et améliorer le calcul des attaques avec ces paramètres.
- ▷ Créer au moins deux catégories de monstres avec différentes caractéristiques.
- ▷ Faire une version graphique du jeu avec `tkinter`.