

Examen de Programmation Objet (Java)

2019–2020

- Durée : 2h
 - Type : examen machine à distance
 - Rendu : une archive contenant les sources Java et un fichier `readme`, à déposer sur AREL
-

L'objectif de l'examen est de réaliser un jeu où le joueur contrôle un personnage dans une grille 2D de dimensions fixées. Dans chaque case de la grille, le joueur peut tomber sur des ennemis à combattre et/ou des items à ramasser. À chaque tour de jeu, le joueur choisit une direction (haut, droite, bas, gauche) dans laquelle déplacer son personnage, en utilisant le pavé numérique du clavier. Si son personnage se retrouve dans une case où il y a des ennemis, il les combat un par un. Selon leur type, les ennemis ont des stratégies d'attaque différentes (*e.g.*, infliger des points de dégâts au joueur, voler un item dans son sac...). Le joueur ramasse ensuite tous les items présents dans la case, seulement s'il n'y pas ou plus aucun ennemi. Selon leur type, les items ramassés produisent un effet immédiat sur le joueur (*e.g.*, lui redonner des points de vie, augmenter sa force d'attaque...) ou sont stockés dans le sac du joueur (*e.g.*, un item permettant de battre des ennemis spéciaux...). Pour gagner, le joueur doit atteindre un objectif prédéfini (*e.g.*, éliminer tous les ennemis, trouver un item particulier, atteindre une case particulière...). Le joueur peut également perdre la partie selon certaines conditions (*e.g.*, il n'a plus de points de vie, il tombe dans un piège, le temps est écoulé...).

Un squelette de code de la classe `Jeu` est fourni. La méthode `jouer()` lance la boucle de jeu, permettant notamment au joueur de déplacer son personnage dans la grille, effectuer les combats avec les ennemis et ramasser les items si possible. Les méthodes `gameOver()` et `victoire()` déterminent si le joueur a perdu ou gagné la partie, respectivement. À vous de compléter ce code et d'ajouter d'autres classes pour réaliser votre propre jeu.

Votre code doit :

- Compléter la boucle de jeu ;
- Définir les conditions de victoire et de *game over* du jeu ;
- Comporter des affichages console pour visualiser l'état du jeu à chaque tour ;
- Définir au moins 3 types d'ennemi avec des stratégies d'attaque différentes ;
- Définir au moins 3 types d'item avec des effets différents ;
- Utiliser le mécanisme de gestion d'exception pour garantir sa robustesse ;
- Avoir une méthode `main` qui crée et configure une partie ;
- Éviter la duplication et utiliser le polymorphisme ;
- Comporter des commentaires structurés afin de donner une description de chaque classe et des méthodes importantes, ainsi que de mentionner votre nom ;
- Être accompagné d'un fichier `readme` expliquant votre jeu.

Pensez à bien vous assurer du bon déroulement du rendu, car toute archive incorrecte ou vide sera notée zéro.

```

public class Jeu {
    public void jouer() {
        Scanner sc = new Scanner(System.in);
        while (!gameOver() && !victoire()) {
            switch (sc.nextInt()) {
                case 8:
                    ... // le joueur se déplace vers le haut
                    break;
                case 6:
                    ... // le joueur se déplace vers la droite
                    break;
                case 2:
                    ... // le joueur se déplace vers le bas
                    break;
                case 4:
                    ... // le joueur se déplace vers la gauche
                    break;
                default:
                    return;
            }
        }
        if (gameOver()) {
            System.out.println("Game over");
        } else if (victoire()) {
            System.out.println("Bravo !");
        }
        sc.close();
    }

    private boolean gameOver() {
        ... // à compléter
    }

    private boolean victoire() {
        ... // à compléter
    }
}

```