

## Rapport Projet ZuulBad

### I.A) Auteur

Aloïs Fournier

### I.B) Thème (phrase-thème validée)

Défier les 9 cercles de l'enfer afin d'accéder au paradis

### I.C) Résumé du scénario (complet)

à faire

### I.D) Plan (complet)

A faire

### I.F) Détail des lieux, items, personnages

### I.G) Situations gagnantes et perdantes

### I.H) Eventuellement énigmes, mini-jeux, combats, etc.

### I.I) Commentaire (ce qui manque, reste à faire, ...)

## II. Réponses aux exercices (à partir de l'exercice 7.5 inclus)

7.5) La duplication de code étant l'ennemi numéro 1 en IGL, on a créé une méthode `printLocationInfo()` qui permet de connaître les informations de la pièce courante (description et sorties)

7.6) dans `setExits` on effectue une série de checks pour déterminer si les sorties sont null ou pas, ce qui est redondant et a pu être fixé par l'ajout d'une méthode `getExit()` qui retourne la room correspondant au paramètre `pDirection` qui est égal à l'un des 4 points cardinaux.

7.7)

Afin d'éviter la duplication de code, il est possible d'améliorer l'efficacité de la méthode `"printLocationInfo()"`. Pour ce faire, on crée une nouvelle méthode appelée `"getExitString()"`, qui permet d'obtenir une chaîne de caractères déjà formatée indiquant les sorties de la pièce. Cela permet de donner cette responsabilité à la classe `"Room"`, qui en est directement concernée, et qui est donc plus adaptée, car la classe `"Game"` est le moteur de jeu.

On modifie ensuite `printLocationInfo()` pour inclure les changements.

7.8)

On remplace les 4 attributs par une `HashMap` qui facilitera l'ajout d'autres directions.

Suite à ce changement, il faut adapter le constructeur naturel

ainsi que la procédure `getExit()` car les attributs ont changé

On doit également changer la méthode setExits()

On doit donc également réécrire la création de la sortie des Rooms.

Afin de pallier au problème que le programme n'affiche plus que la direction est inconnue mais uniquement qu'il n'y a pas de porte, on crée un nouvel attribut et une nouvelle méthode statique dans la classe CommandWords.

On l'intègre maintenant dans goRoom() afin d'avoir les bons messages.

7.8.1)

On commence par adapter nos changements sur CommandWords en rajoutant « up » and « down ».

On modifie donc l'altitude du départ et l'arrivée.

7.9)

On modifie la méthode getExitString() afin d'utiliser un Set qui est peuplé des keys (identifiants) de la HashMap créée dans les exercices précédents

7.10.1)

On annote toute les méthodes et classes créées pour la javadoc

7.10.2)

On génère donc la javadoc avec BlueJ

III. Mode d'emploi (si nécessaire, instructions d'installation ou pour démarrer le jeu)

IV. Déclaration obligatoire anti-plagiat (\*)