**Rapport du jeu Zuul**

**Auteur :** DEBELLE Hugo **Groupe :** 4

**Phrase thème :** Un détective dans un musée devant résoudre des énigmes en changeant de salle pour trouver comment un objet a été volé

**Résumé du scénario :**

Un détective dans un musée devant résoudre des énigmes en changeant de salle pour trouver comment un objet a été volé. Pour gagner, il doit choisir la bonne phrase retraçant les événements du vol. S’il choisit la bonne phrase, il gagne sinon il perd.

**Plan réduit :**

Rectangle signifie salle à la même hauteur

Ovale signifie salle à une hauteur plus haut

Flèche signifie liaison disponible d’une salle vers l’autre

**Une image contenant diagramme, ligne, capture d’écran, conception

Le contenu généré par l’IA peut être incorrect.**

**Scénario réduit :**

TODO

**Détail des lieux, items, personnages :**

TODO

**Situations gagnantes et perdantes :**

Le joueur a 1 essaie dans lequel il doit choisir parmi des phrases qui retrace tous les événements du vol de l’objet. S’il choisit la bonne phrase, il gagne sinon il perd.

**Enigmes, mini-jeux, combats :**

TODO

**Commentaires :**

A remplir à la fin du jeu

**Réponses aux exercices**

Exo 7.5 :

J’ai dû modifier la methode goRoom et printWelcome pour qu’elle appelle printLocationInfo permettant de supprimer la répétition de code

Exo 7.6 :

J’ai ajouté la fonction getExit(direction) dans Room permettant de retourner la piece en fonction de la direction et d’éviter la répétition de code. Dans la procédure printLocationInfo j’ai fait un for de toutes les directions évitant la répétition des ifs. Plus tard, il faudrait surement créer une fonction retournant un tableau de toutes les directions possibles

Exo 7.7 :

J’ai créé la fonction getExitString() qui retourne la String avec toutes les sorties disponibles pour la room actuelle. J’ai ensuite remplacé le for de la procédure printLocationInfo par l’appelle de cette fonction.

Il faut mettre getExitString() dans Room car c’est dans room que on a toutes les sorties et qu’on sait si elles sont null ou non. Aussi, si on veut ajouter une sortie, on a juste a modifier cette fonction. On affiche dans Game car c’est dans cette procédure (printLocationInfo) qu’on s’attend à l’affichage. Room ne doit rien afficer

Exo 7.8 :

Il y a la méthode setExits qui devient inutile puisque remplacé par la méthode setExit (sans s) car elle permet de mettre une et une seule room dans une direction souhaitée

Exo 7.9 :

Ajouts du keySet dans GetExitString permettant d’itérer sur toutes les directions crées. Cela nous évite de mettre toutes les directions à la main

Exo 7.10 :

La méthode getExitString permet de retourner un String contenant les sorties disponibles de cette room. Pour cela on itère sur toutes les directions créent et ajoute ces directions dans le résultat final.

Exo 7.10.2 :

Class Game contient moins de méthode que la class Room dans la javadoc car la javadoc affiche que la doc des méthodes publiques hors la class Game possède beaucoup de méthode privés

Exo 7.11 :

On a créé une fonction getLonDescription dans la classe Room permettant de retourner la description à afficher au joueur. On créer ceci dans la classe Room car si on souhaite ajouter ders infos dans une Room on a juste cette classe à modifier et non cette classe + la classe Game

Exo 7.12 :

Une image contenant texte, diagramme, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Exo 7.13 :

Le diagramme ne change pas durant l’exécution de go command car toutes les rooms et les objets en général sont déjà créent

Exo 7.14 :

On a ajouté la commande look dans CommandWords et dans processCommand permettant d’exécuter la commande look. Enfin, on a regardé s’il y avait un second mots et si c’est le cas, on envoie un message d’erreur

Exo 7.15 :

En suivant le même principe que la commande look, ajout de la commande eat

Exo 7.16 :

Ajout des procédures showAll et showCommands permettant de print toutes les commandes disponibles. Utiles pour help car quand on ajoute des commandes, elles sont automatiquement print et on a plus besoin de les rajouter à la main pour le help.

Exo 7.17

Si on veut ajouter une commande, on doit toujours modifier la classe Game car on doit modifier la fonction processCommand qui permet d’executer une méthode souhaiter en fonction de la commande tapé

Exo 7.18 :

Rename de la méthode showAll en getCommandList et showCommands en getCommands.

En effet ces 2 méthodes retournent maintenant la liste des commandes au lieu de les afficher directement.

Eco 7.18.2 :

Remplacer la concatenation d’un String par StringBuilder.append() dans les fonctions Room. getExitString() et CommandWords.getCommandList()

**Déclaration obligatoire anti-plagiat**

Aucun code n’a été repris. Seul le code fournis dans les exercices a été utilisés.