

RÉALISATION D'APPLICATION

RAPPORT ITÉRATION 1

Jeu d'aventure

Fiche de cas d'utilisation

Auteurs
Loïc LOKO
Ghassen GHOUIBI
Abdelrahim BOUCHIHA
Groupe 5

Sommaire

I Présentation	2
II Découverte de zuul-bad	2
I. Que fait cette application ?	2
II. Quelles commandes le jeu accepte-t-il	2
III. Que fait chaque commande ?	2
IV. Combien de salles y a-t-il dans le jeu ?s	2
V. Rôle des classes	2
VI. Scénario libre	2
VII. Plan du jeu	3
VIII. zuul-v1 rooms exits (Voir code)	3
IX. Les scenarios alternatifs	3
X. getExit (Voir code)	3
XI. getExitString (Voir code)	3
XII. HashMap, setExit(Voir code)	3
XIII. keySet(Voir code)	4
XIV. getExitString CCM (Voir code)	4
XV. getLongDescription (Voir code)	4
XVI. look (Voir code)	4
XVII. eat (Voir code)	4
XVIII. help (Voir code)	4

I Présentation

Dans le cadre du cours "Réalisation d'application", nous avons eu la tâche de réaliser l'itération 1 du jeu d'aventure intitulé "Zuul". Le but de cette itération est de découvrir certains facteurs qui influencent la conception d'une classe. Nous répondons ici à la question : Qu'est ce qui rend la structure d'une classe bonne ou mauvaise ? Ainsi dans cette itération, en répondant aux différentes questions, nous avons découvert des principes importants dans la conception d'une classe. Ce rapport contient donc les différentes réponses aux questions.

II Découverte de zuul-bad

Zuul-baad est un projet qualifié de mauvais car il contient de mauvaises conceptions, nous l'avons donc exécuté et exploré afin de le comprendre, ensuite nous avons suivi des procédures afin de le corriger.

I Que fait cette application ?

Cette application est un environnement pour un jeu d'aventure. Dans cette version, Il y a quelques pièces et la possibilité pour un joueur de marcher entre ces pièces.

II Quelles commandes le jeu accepte-t-il

Le jeu accepte les commandes suivantes :

1. go
2. quit
3. help

III Que fait chaque commande ?

La commande go sert à se déplacer, ici nous nous déplaçons dans 4 directions : east, north, south west.

IV Combien de salles y a-t-il dans le jeu ?s

Ce jeu est composé de 5 pièces si nous comptons la pièce de départ.

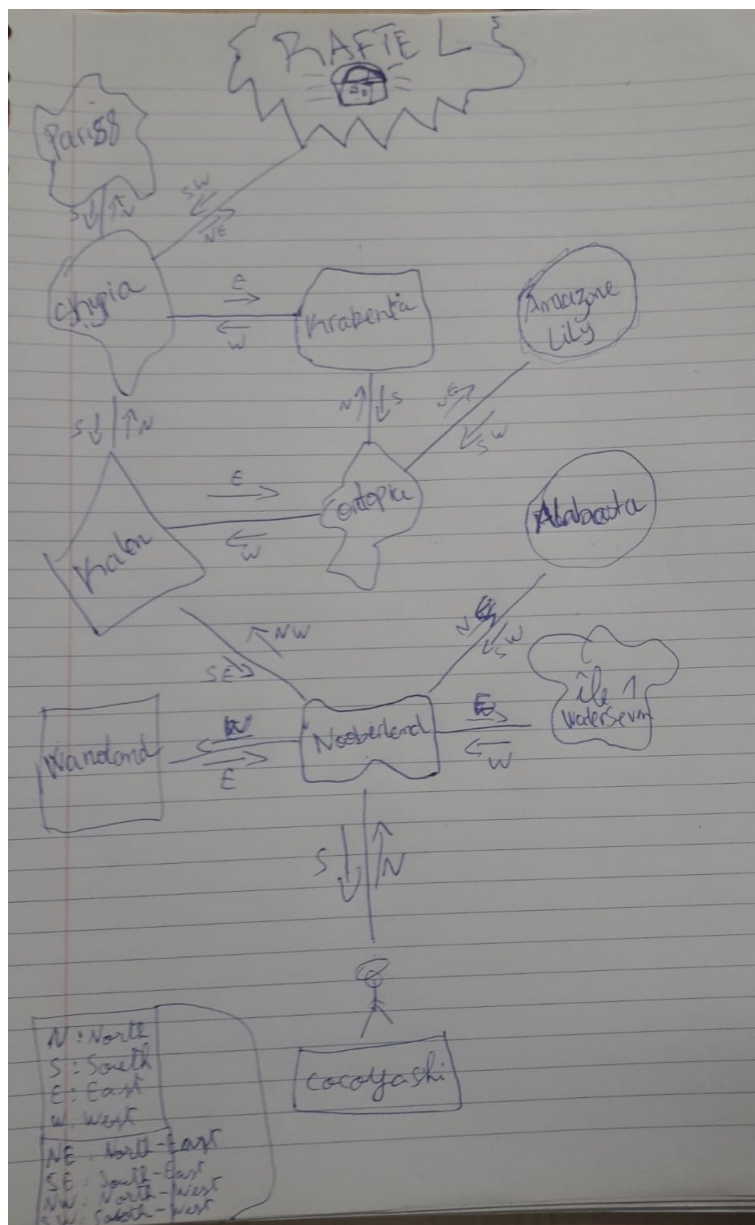
V Rôle des classes

La classe Scanner est utilisée dans le fichier parser.java afin de lire toutes les entrées saisies par l'utilisateur.

VI Scénario libre

Vous êtes le chef d'équipage d'un bateau de pirates, vous naviguez sur les mers afin de visiter différentes îles, se ravitailler en vivres ou équipements, combattre d'autres équipages et la marine afin de s'approprier et trouver les plus grands trésors dont le One Piece.

VII Plan du jeu



VIII zuul-v1 rooms exits (Voir code)

IX Les scenarios alternatifs

X getExit (Voir code)

XI getExitString (Voir code)

XII HashMap, setExit(Voir code)

XIII `keySet` (Voir code)

La méthode `keySet()` est utilisée pour obtenir une vue sur les clés contenues dans cette map. L'appel de méthode retourne une vue des touches contenues dans cette map.

XIV `getExitString CCM` (Voir code)

La méthode `getExitString` crée un string, récupère (une vue) les valeurs contenus dans la map `exits` puis parcourt l'ensemble des éléments afin de créer par la suite une chaîne de caractère. Ici elle récupérera le nom des sorties.

XV `getLongDescription` (Voir code)

XVI `look` (Voir code)

XVII `eat` (Voir code)

XVIII `showAll, showCommands` (Voir code)