# Rapport – Itération 1 : Jeu d’aventures

## 7.1 / Découverte zuul-bad

1) 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.

2) Quelles commandes le jeu accepte-t-il ?

Le jeu accepte les commandes suivantes :

go quit help

3) Que fait chaque commande ?

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

4) Combien de salles y a-t-il dans le jeu ?

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

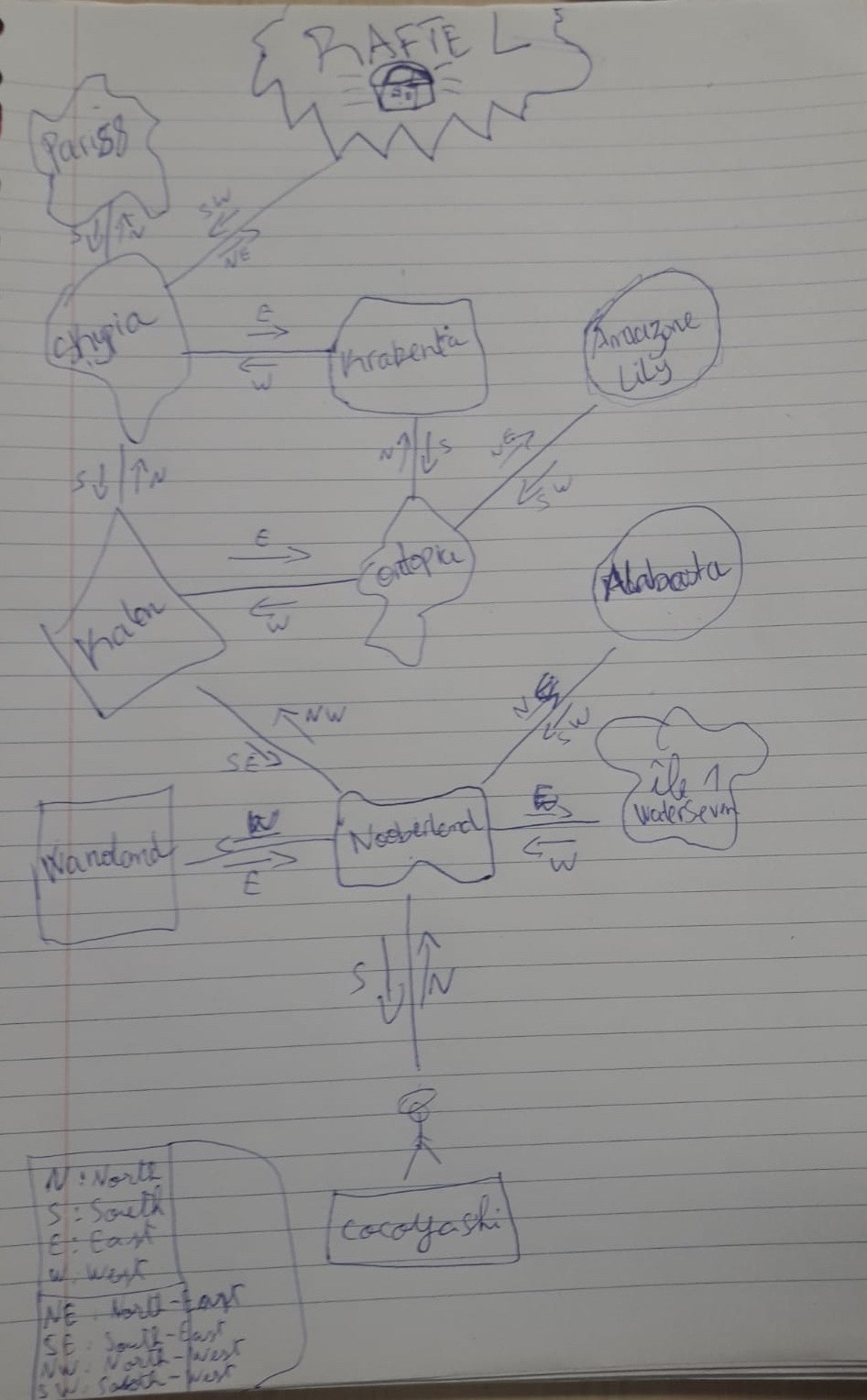
## 7.2/ Rôle des classes

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

## 7.3/ 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.

### **Plan du jeu**



## 7.4/ zuul-v1, rooms, exits ( Voir code )

## 7.5/ printLocationInfo ( Voir code )

## 7.6/ getExit

## 7.7/ getExitString

## 7.8/ HashMap, setExit

## 7.9/ keySet

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 carte.

## 7.10/ getExitString CCM

La méthode getExitString crée un string, récupère ( une vue ) les valeurs contenus dans la map exits puis parcours l’ensemble des éléments afin de créer par la suite une chaîne de caractère.

Ici elle récupérera le nom des sorties.

## 7.11/ getLongDescription ( Voir code )

## 7.14/ look ( Voir code )

## 7.15/ eat ( Voir code )

## 7.16/ ( Voir code )