COLLET Arnaud / RICORDEL Maxime

LP S2IMa
IUT Vannes
session 2018/2019

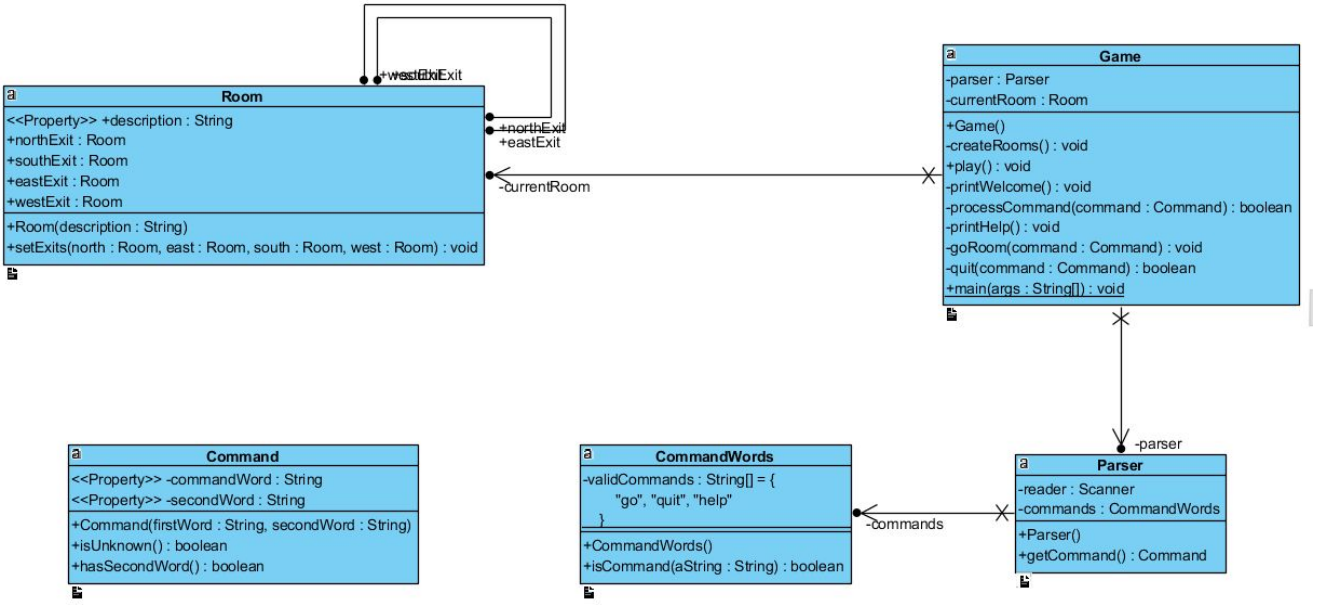
Compte-rendu TP Game

Compréhension de l'application

1. Telecharger l'application game, compiler et lancer le jeu. La classe Game possède une méthode main

Étape effectuée avec Eclipse.

2. Faire la rétro-conception du diagramme de classes UML



Reverse avec l'outil Visual Paradigm

3. Explorer le code de l'application, les commentaires fournissent quelques informations et répondre aux questions suivantes :

- Que fait cette application ?

Il s'agit d'un mini jeu d'aventure en mode texte. Le joueur peut se déplacer entre différentes pièces. Il n'y a pas de but pour le moment.

- Quelles sont les commandes acceptées par le jeu ?

Il y a 3 commandes acceptées dans le jeu : go, quit, help.

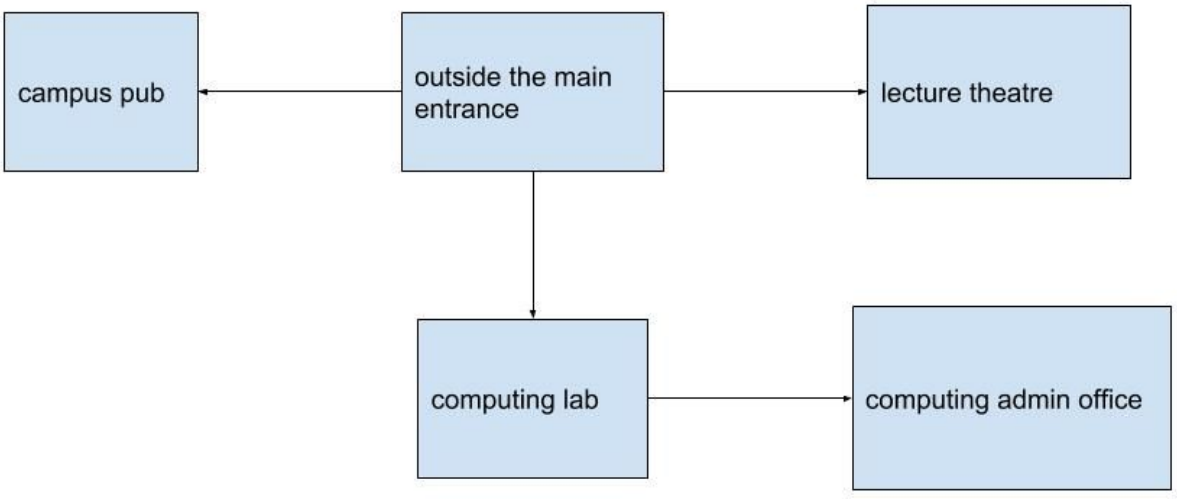
- Que fait chaque commande ?

- go : avec 2e mot north/east/south/west. Le joueur se déplace entre les pièces suivant 4 directions ("no exit here" si impossible)
- quit : pour quitter l'application (booleen passe à vrai)
- help : affiche une aide au joueur (un message et les commandes)

- Combien y-a-t-il de pièces dans ce scénario ?

Il y a 5 pièces dans ce scénario (outside/theatre/pub/lab/office)

- Dessiner une carte des pièces existantes :



Map des pièces

4. Identifier le travail réalisé par chaque classe : noter son objectif

- classe Game :

Il s'agit de la classe principale de l'application. Elle crée et initialise les autres classes. Elle crée les *Rooms*, le parseur et lance le jeu (comporte le *main*).

- classe Room :

Classe correspondant aux différentes pièces du jeu. Ces pièces comportent des sorties (north/south/east/west) et pour chaque direction on sait quelle est la pièce suivante (ou *null*).

- classe Parser :

Cette classe permet de lire les entrées de l'utilisateur et vérifie que celles-ci correspondent à des commandes prévues par l'application. Elle retourne la commande sous la forme d'un objet de la classe Command.

- classe Command :

Une commandes est constituée de 2 mots. Une commande est déjà vérifiée par la classe CommandWords pour voir s'il s'agit de mots valides. Si l'utilisateur rentre une commande invalide (mot inconnu) le command word est *null* et si la commande n'a qu'un seul mot, le second est passé à *null*.

- classe CommandWords :

Cette classe correspond à une énumération des mots valides pour les commandes du jeu. Elle reconnaît les commandes saisies par le joueur.

Personnalisation du jeu

➔ Implémentation d'une interface graphique Swing
➔ Nouvelle classe Jeu comportant le main : lancement soit par le biais de l'interface graphique soit par la console

Resolution des problemes de conception

➔ Voir l'archive .zip

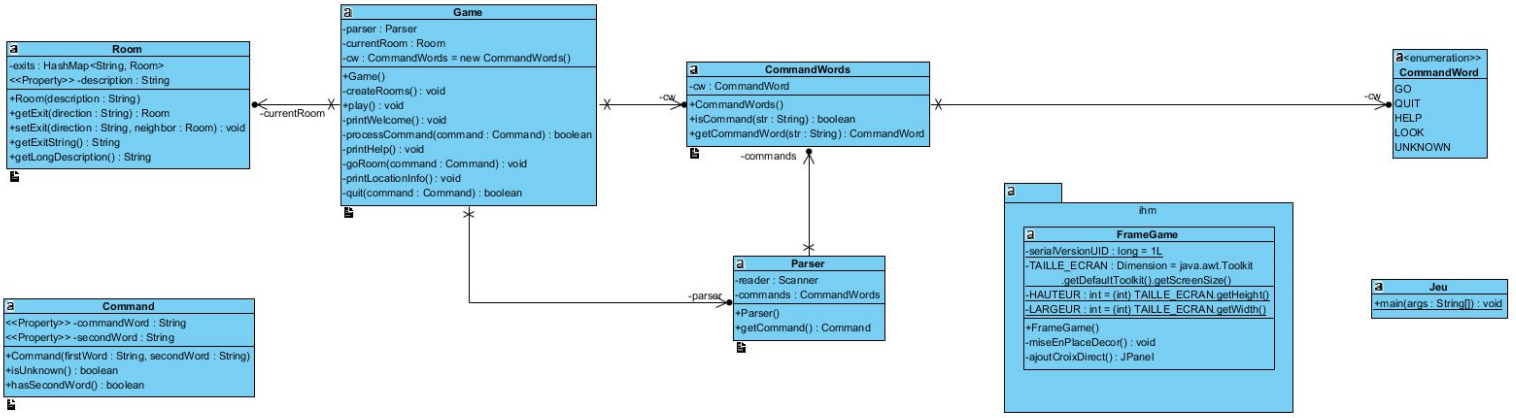


Diagramme de classes final