ITECH7201 - Software Engineering: Analysis and Design

Assignment 2

Extend the functionality of the Maze game

Student Name: Melvin M. Flores

Student ID: 30352985

Lecturer: Mustafa Hashmi

Date Submitted: 02 October 2019

Table of Contents

[1. Group Details 2](#_Toc20952575)

[1.1 Group Name 2](#_Toc20952576)

[1.2 Group Members: 2](#_Toc20952578)

[2. Introduction 2](#_Toc20952581)

[3. User Stories 3](#_Toc20952582)

[3.1 Laboratory 7 Functionality 3](#_Toc20952583)

[3.2 Laboratory 8 Functionality 3](#_Toc20952584)

[3.3 Item Management Commands 4](#_Toc20952585)

[3.3.1 listItems 4](#_Toc20952586)

[3.3.2 getItem 4](#_Toc20952587)

[3.3.3 dropItem 4](#_Toc20952588)

[3.3.4 equipItem 4](#_Toc20952589)

[3.3.5 unequipItem 5](#_Toc20952590)

[3.3.6 purchase Item 5](#_Toc20952591)

[3.3.7 sellItem 5](#_Toc20952592)

[3.3.8 getStatus 5](#_Toc20952593)

[3.3.9 flee 6](#_Toc20952594)

[3.3.10 usePotion 6](#_Toc20952595)

[3.4 Basic Combat Functions 7](#_Toc20952596)

[3.5 Collectable Items 7](#_Toc20952597)

[4 Class Diagrams of Lab 7 and Lab 8 8](#_Toc20952598)

[4.1 Lab 7 Class Diagram 8](#_Toc20952599)

[4.2 Lab 8 Class Diagram 9](#_Toc20952600)

[5 Sequence Diagrams 9](#_Toc20952601)

[5.1 getItem 9](#_Toc20952602)

[5.2 getmazestatus 10](#_Toc20952603)

[6. Reflection on Learning. 11](#_Toc20952604)

# Group Details

## 1.1 Group Name

## FS group (Federation Superstars / Flores Sakshi)

## 1.2 Group Members:

## Melvin Flores – 30352985

## Sakshi Panchal – 30370118

## Introduction

The Player awakens in Melvin’s library located at the towering Geats Olympus, and he is wondering where he was because the last memory he remembered was studying for the October 15 finals of Java in ITECH7201 at the Federation University Library with the goal of making Mustafa proud, his wise and smart lecturer. Books are scattered everywhere. The Player’s eye catches a golden flying book and was surprised that it was directly headed at him. He ducked under the table, but the book lifted the table. He removed his shirt and courageously catches the flying book. In his amazement, the book talked and told him that an adventure awaits him, and the only way he can get back to his own world is to reach the Frisian Palace. As a reward, he was given a starting life point of 20, and a map to aid his quest. But he was given a warning, if his life point drops to less than 1, he will not exist and will be erased in the memory of all the persons who knew him in his world.

## User Stories

## 3.1 Laboratory 7 Functionality

In Laboratory 7, a text-based adventure game was introduced to us. We were also given a document containing its background information. The script, game rules and reference tables were provided in that document. In the actual laboratory, we were given with a ready-made code and our tasks was to copy-paste codes to several classes as follows:

1. ParsedInput
2. Parser
3. IMazeClient
4. SimpleConsoleClient
5. DungeonMaster
6. Player
7. Location

Understanding the codes in each class were very difficult on my side, as my skill in Java language is at beginner-level. Instructions in the laboratory document were followed, and syntax errors occurring were fixed. Eventually, the program run, and the Move command was successfully implemented in the MazeGame program.

## 3.2 Laboratory 8 Functionality

Command Pattern was introduced and implemented in Laboratory 8. The modification of java classes from Laboratory 7 were continued, and we followed the steps in the laboratory document. The codes were harder to understand compared from the previous week, and the syntax errors which occurred were fixed a few minutes before the laboratory ended. Ultimately, the program run, and implementing codes like Quit command, Look command and Move command were a lot easier and code looks a lot cleaner. Also, it is easier to extend other commands in the MazeGame program.

The Command pattern is a behavioural pattern which means it is mainly concerned with the functionality that the object’s display. A clunky, complex and hard to maintain code is improved by the application of a Command pattern into a flexible, modifiable and extendable program. It also saves time and entails faster program development.

## Item Management Commands

## listItems

Player wants to look at his arsenal of weapons, armor and shields. These items are stored in the players inventory of items. An example of a weapon is a dagger, but a player can only be armed with a single weapon at a time. A weapon inflicts a damage to a hostile Non-Playing Character (NPC). Padded and buckler are examples of an armor and a shield, it provides protection from an NPC attack. A player may wear an armor and carry a shield.

## getItem

Player will get an item in a location where a weapon, armor or shields are lying around. These items are stored in the location inventory of items. These items will either inflict damage to an NPC, protect the player, or both.

## dropItem

Player drops an item he is holding if his carrying capacity is exceeded, thus leaving the item on the location. He may also drop an item because he wants to change a different and more powerful item. These items will be left in the location in which he dropped the item.

## equipItem

Player will either hold or wear an item. For example, he might be holding a dagger weapon on his hand. He might also be wearing an item, like a padded armor on his body, and a buckler shield on his arm. This will make the player more powerful against the NPCs.

## unequipItem

Player removes the item he is holding or wearing. He may want to remove it because he wants to replace it with a more powerful weapon, armor or shield.

## purchase Item

Player can accumulate money in the form of gold pieces (gp) during his quest in the mazegame. From his money, he can buy different items which will advance him to the next level of the game. He will also encounter powerful NPCs as the game progress, so he will be needing an effective weapon, armor or shield to defeat the NPCs. Item can only be purchase on a shop which are Blacksmith forges. In our mazegame, we have 2 shops, the Danes Stonewood Shop and the Swedes Amphitheatre Shop.

## sellItem

In the case that the Player wants to dispose his item, he can sell it back to the shop but with a 20% discount. There may be different reasons for selling, some of which is to accumulate enough money to buy a more powerful item, and to buy a specific item that will defeat the NPC. Selling can only be done in the shops, in either the Danes Stonewood Shop, or at the Swedes Amphitheatre Shop.

## getStatus

The player might be lost and wandering around the mazegame. To keep the player back in track, he can get the status of the mazegame, and it will show him in which location or shop he is currently on.

Five locations were implemented. The central and starting location is at Melvin’s Library, which is located at the Geats Olympus. Two locations situated East and West from Melvins Library are Wulfings Plains and Jutes Riverbed. Another location, Angles Island, is located at south of Danes Stonewood Shop. Lastly, the Frisian Palace is located at the west of Jutes Riverbed.

Two shops were also implemented, the Danes Stonewood Shop, and the Swedes Amphitheatre Shop.

## flee

If the Player does not want to engage in a confrontation with an NPC, he can run away. Sometimes, knowing your current situation and capabilities, it is best to run away rather than fighting a battle which you know you will surely lose. This is best used when the Players life points is at critical level, or he is being continuously battered and receiving damages by a hostile NPC risking his life in danger.

## usePotion

A potion can regain the life points of a badly damaged and endangered Player. If the Player drinks the potion, his life points will improve, either making it possible to defeat the NPC or to just run away. The potions are sold in other shops like the shop which is under-construction named Melvin’s coffee shop.

## Basic Combat Functions

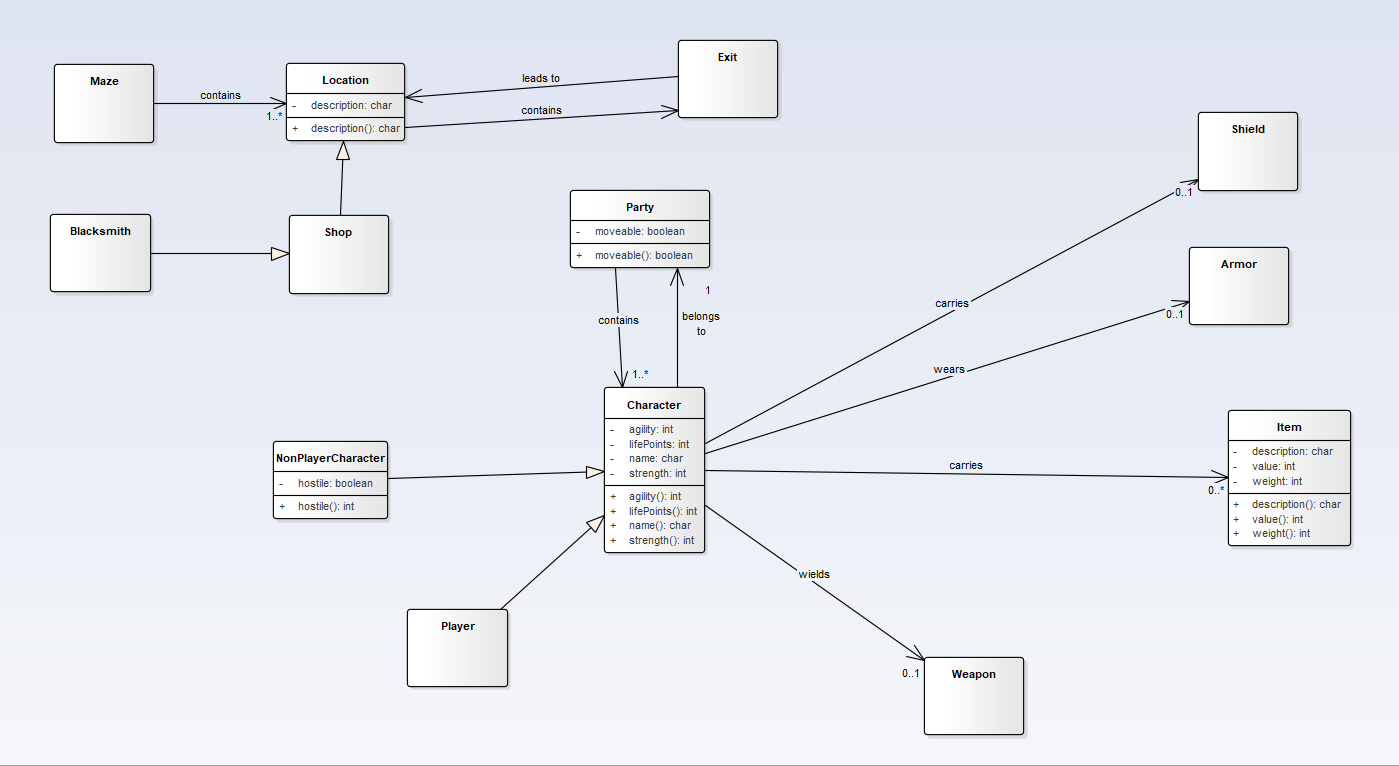
While the Player is making his journey to the maze game, he will move from one location to another, or from one shop to another, or combination. As he moves across each location, he will encounter a Non-Playing Character (NPC). This NPC may be friendly or hostile. If the NPC is hostile, he may choose to attack the NPC first. In response to his attack, the NPC will give a counter-attack. In the Players point of view, items like weapons, armor and shields can be held or wore, to maximize the attack to the NPC, as well as minimize the NPCs counter-attack. But similarly, the NPC can also hold and wear items making the combat more challenging. Drinking a potion can also be used to restore the life points of a Character, either the Player or the NPC. And the Character, has also the option to run away. The Character will die if his life point is less than 1. As the player moves from one location to another, the difficulty of the combat increases as the NPC become smarter and greedier in using the items and drinking the potions. The journey ends once the Player reaches the location Frisian Palace, a castle in a hill made with glistening glow of golds and sparkling diamonds, wherein the road is paved with golden bricks.

## Collectable Items

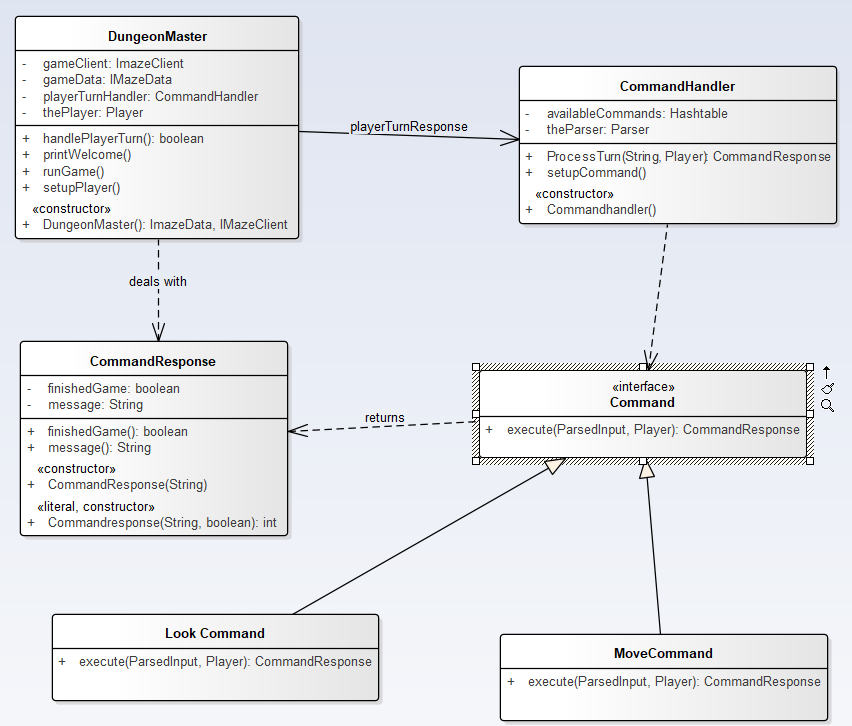
Legend tells a story of a Blacksmith who forges 100 pieces of a limited armor, which is sold only in the Swedes Amphitheatre Shop, but the price is too expensive. These are considered collectible items because it inflicts the greatest damage, as well as gives the best protection under NPCs attack. The item is a full-plate armor, selling at a whopping 1,500 gold price.

# Class Diagrams of Lab 7 and Lab 8

## 4.1 Lab 7 Class Diagram

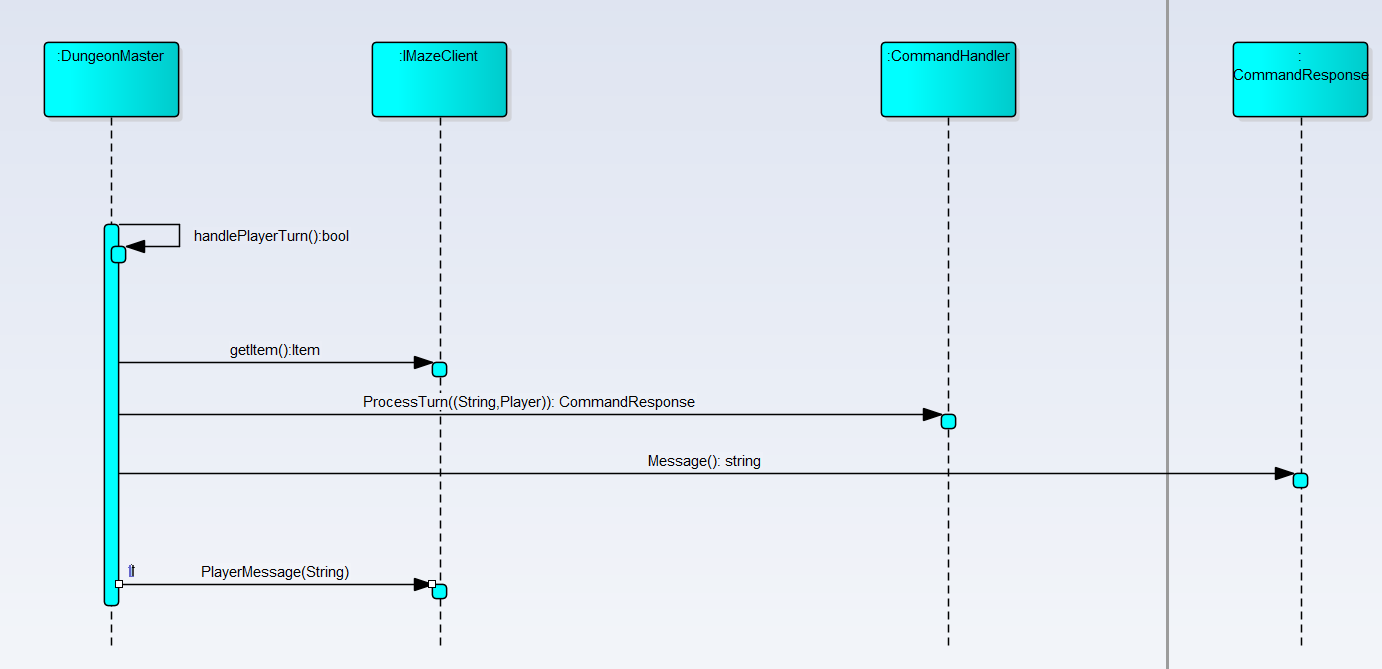


## 4.2 Lab 8 Class Diagram

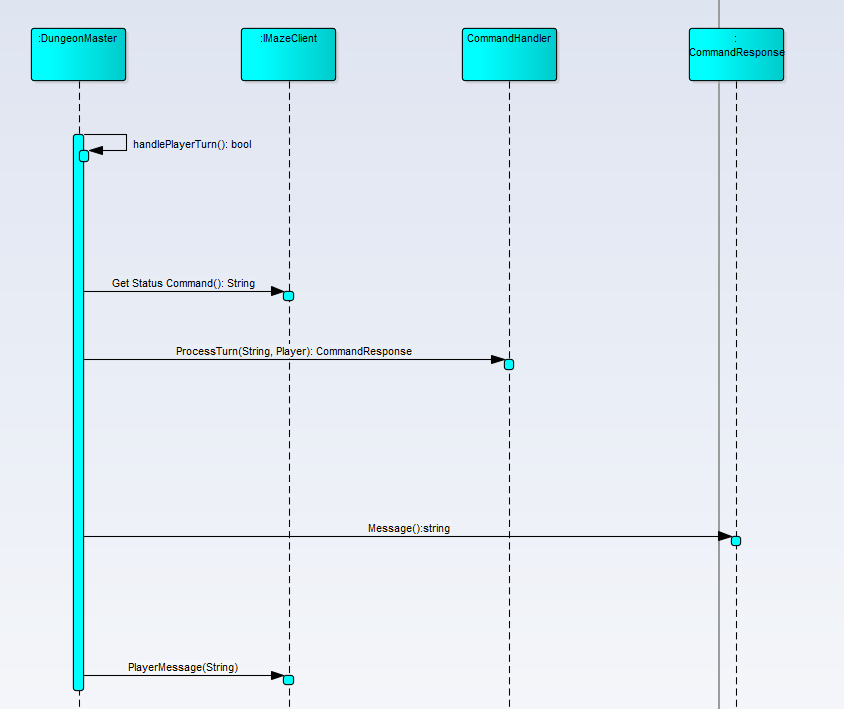


# Sequence Diagrams

## 5.1 getItem



## 5.2 getmazestatus



# Reflection on Learning.

Wrapping up the Java class with a console based role-playing game, was both a rewarding and disappointing activity for me. Rewarding, because as a Java beginner, the equipItem, unequipItem and getStatus functionality was made to work by myself. It was frustrating because usage of Arrays, List and Object was not allowed by the program because errors are showing left and right and cannot be fixed. Big challenges were encountered during the coding part, and some functionalities were not coded like the basic combat functions because the code classes were already given and giving me a hard time to understand a code which I did not started. Better to have coded from the scratch, as total control will be solely based on me. But my best efforts for learning and understanding were given in this task. It was a great experience and learning the Java programming language this semester was significantly attained.