

Software Engineer Project - Jungle Temple

Nicolò Fioranzato, Marco Stradiotto, Pietro Trabuio, Gianmarco Zin

2024-25

1 Introduction

Jungle Temple is a text-adventure dungeon crawler. The game's objective is to defeat the final boss of the temple and leave with the treasure. The player must move through the intricate corridors of the temple, fight dangerous creatures and interact with various characters to obtain the items to finish the game.

2 Game structure

The game map is made up of many rooms, interconnected by corridors. All the rooms can be accessed from the beginning of the game except for the final boss room: to access this room, the player has to defeat all the other bosses in the dungeon. Each boss, except for the final one, has a specific item the player must equip in order to defeat it. These items are provided by the traders, which are three NPCs scattered throughout the map. Each one of these asks the player to bring them an item while giving the player an item required to beat one of the three bosses.

The game heavily revolves around combat. In order to inflict damage, the player must have a weapon equipped. Each combat is a 1v1 encounter between the player and an NPC, regardless of how many enemies are in the room. One enemy in the game moves randomly through the rooms.

In some rooms, the player can find items to heal himself. Some enemies also drop these healing items when defeated.

3 Navigation

The player can move between rooms by using the **north**, **east**, **south**, **west** commands which move the player through the corresponding corridor, or by using the **back** command which brings the player back to the last room he visited.

4 Item Interaction

The **look** command prints the list of items in a room. the player can then pick an item up by using the **take [item]** command, if the item is pickable and if it doesn't exceed the player's max inventory weight. The player can consume an item in his inventory or an unpickable item in the room with the **use [item]** command. The **drop [item]** command can be used to remove an item from the player's inventory and drop it in the current room. The **inventory** command prints the list of items in the player's inventory.

5 Character interaction

The player can use the **talk [entity]** command to speak with a trader. He can then use the **give [item] to [entity]** command to give the trader his requested item.

6 Combat

When an enemy is in the same room as the player, the game automatically locks on to it. The player can then use the **attack** command to inflict damage to it. The player can also use the **parry** command to try to parry the enemy's attack. If successful, the enemy's attack has no effect and the player's next attack deals bonus damage. The player can also flee from an enemy by using the navigation commands, but receives damage when doing so.

7 Statistics

The player can use the **stats** command to view his statistics.

8 Save/Load Configuration

There are two options available for loading the game:

- **New Game:** load a default game.
- **Load Game:** load a previously saved game.
- **Save Game:** save the current game.

This is due to AWS with the cloud storage S3, an object storage service that offers industry-leading scalability, data availability, security, and performance.

9 Technologies used

Name	Version	Description
Java	21.0.1	Environment and programming language used to develop the application.
JUnit	5.11.0-M1	Framework for automated testing of Java classes.
Jakarta	4.0.2	Software framework that allows Java EE developers to map Java classes to XML representations.

10 Download and install the software

The first step is to clone the repository at the github link open the downloaded project with an IDE (such as IntelliJ), and run the `Main.java` file located at the following path: `src/main/java/it/unipd/edids/Main.java`