

MANUAL

Project Description.....	1
Brief Game Introduction.....	1.1
Game Mechanics.....	1.2
Hostile room:.....	1.2.1
Trap room:.....	1.2.2
Loot room:.....	1.2.3
Boss room:.....	1.2.4
Healing room:.....	1.2.5
How To Install.....	2
Windows.....	2.1
Maven:.....	2.1.1
Jar:.....	2.1.2
Exe:.....	2.1.3
MacOs.....	2.2
Execution environment and Libraries.....	3
Execution environment and tools.....	3.1
System.....	3.1.2
Development and Testing.....	3.1.3
Maven.....	3.1.4
External libraries.....	3.2
Firebase.....	3.2.1
Java Sound Api.....	3.2.2
Jansi.....	3.2.3
Jackson.....	3.2.4

Project Description

Brief Game Introduction

This Project is named “*The Legend Of Jordan*”.

It’s an RPG, Roguelike text adventure, that consists of exploring a dungeon made of 3 floors, each filled with many different rooms. Luck plays an important role during the game, as one of the most important game mechanics is the die roll, which will be used to establish how much damage you will deal to your enemy, how to increase your statistics when you level up, and more.

The game offers you 5 different classes that have unique perks and starting weapons, and each weapon has a particular effect and dice type. Depending on the class you choose, you will have to change your game style in order to beat the game.

Before that you will have to navigate through different types of rooms, where you will encounter a lot of different enemies, you will be hit by different kinds of traps, you will find items, and more.

After each floor you will be able to access the Healing Room, which acts as a bridge between floors. In this room you will be fully restored and you will have the possibility to interact with the Smith and the Merchant.

These are special friendly NPCs that will respectively upgrade your weapon and sell you items.

The goal of the game is to beat the final boss, located on the third floor, in order to escape the dungeon and become a Legend.

Game Mechanics

As said before, the game is a *Text Adventure*, so everything is handled by typing some specific command on the console, although there are a lot of graphic (ASCII) elements to improve the game clarity and overall experience.

When you first launch the program, you will see the main menu of the game, where you can choose whether you want to continue the last game (only if a last game is available), start a new game or load a saved game.

The game will automatically save your current game progress in a .json file each time you move to a new room, and these local saves will be uploaded to Firebase Storage each time you quit the game via the “quit” command.

To reach the end and win the game, you will have to defeat all the different bosses you will encounter at the end of each floor.

Along the way, defeating mobs and bosses, you will level up after gaining enough XP points; this will increment your base attack and defense by a random number established by a roll of the dice.

The game will constantly prompt you to decide which action to perform from a pool of available commands depending on the current context and the type of room you reach.

Here follows a description of these different scenarios.

Hostile room:

In these rooms you will fight a Mob, whose stats (attack and defense) will vary based on its predefined level (the enemies will get stronger as you reach higher floors).

Mobs have abilities that might trigger during the fight, which is turn based, and has the player always attacking first.

On his turn the player can choose one of the following actions, of which only “attack” will actually end his turn:

- ❖ Attack → to roll his weapon dice and attack the enemy. This will end his turn.
- ❖ Skill → to use the character skill (only if enough mana is available). The skill will be used instantly.
- ❖ Use → to use an Item from his inventory (only one Item per turn is allowed).

When the player ends his turn the mob will simply attack the player back.

[The attack mechanic is the same for every mob → $\text{damage} = (\text{mob's base attack} + \text{weapon dice roll}) - \text{player's defense}$]

Once the fight is over the mob will give the player some money and XP, based on its level, and might drop a random Item.

Trap room:

There are three types of trap in this game, two of which might not always be triggered:

- ❖ Damage Trap → deal 10 damage to the player → (33% of avoiding it)
- ❖ Seal Trap → steal 10 money from the player → (33% of avoiding it)
- ❖ Teletransportation Trap → moves the player to a random room of the same floor (excluding the Boss room) and made him forget about the map → not avoidable

All Trap rooms can be deactivated if the player has and chooses to use the Item “EMP” upon entering a Trap room.

Loot room:

In this room the player will always find a random item of the game.

If the player has no free space in his inventory the system will allow the player to either leave the room and the item, (the item will not be lost as it will still be available to pick if the player returns in the room), or drop an Item from his inventory to free some space and pick the room's Item.

There is one particular Loot room in the game, located on the second floor.

We consider that a *Special Loot Room*, because unlike other Loot rooms it is normally locked and can be unlocked if the player has in his inventory the “*Special Key*” item when entering the Special Loot Room.

The Special Key is a special item that can be found only by killing “*The Glitched*”, a special mob that will teleport to another random Hostile room (visited or not) on the same floor after it attacks. The player only has 5 turns to kill it and obtain the special key, otherwise he will completely disappear and the Special Loot Room will no longer be unlockable.

This particular Loot room will also always give the “*Nanite Leech Blade*” weapon, the strongest of the game.

Boss room:

Each floor in the game has its final Boss. Defeating the final (third) boss will lead to the game ending. Each boss has a different passive ability.

- ❖ 1° Floor → Cyfartal, Even Lord of the Dice → if the player's weapon roll is Even, set it to zero.
- ❖ 2° Floor → Leth, Halver Lord of the Dice → halves the player's weapon dice roll.
- ❖ 3° Floor → Flygande, Flying Lord of the Dice → it has 0,333% chance to take no damage and heal itself for an amount of hp equal to the player's weapon dice roll.

The first and the second boss will always drop XP, money and a "*Weapon Shard*" item.

Upon entering a Boss room the player can no longer exit it, so once he defeats the boss moving in any direction will take him to the *Healing room*.

Healing room:

The Healing room is located in between floors, and it houses the *Smith* and the *Merchant*.

When the player enters this room, he will always have your mana and health fully restored.

The Smith will upgrade the player's equipped weapon if the player gives him a Weapon Shard, while the Merchant will sell a selection of items.

Moving to any direction will move the player to the next floor.

How To Install

There are various ways to install The Legend Of Jordan. You can choose the one you prefer depending also on which machine you are gonna play the game on.

Important note:

We strongly recommend you to set the console on fullscreen to have the best experience, otherwise the graphics component won't be shown properly.

Windows

You have three options to install the game on a Windows machine. For the first two methods you will need to download the [Java Developer Kit](#) (v - 19.0.2).

Maven:

1. Go to the [Maven official web page](#) and download it.
2. Follow [these](#) steps to set it up properly
3. Go to the [official repository on GitHub](#). You can choose to download the zip file or use git clone in a folder to download the code.
4. Now open the Command Prompt and go to the directory of the folder you have just downloaded
5. Use “mvn clean package” to compile the code
6. Use “mvn exec:java” to execute the code and play the game.
7. Have fun!

Jar:

1. TODO where to find .zip file
2. Go to the folder you just downloaded and open the Command Prompt
3. Use “java -jar “*Name of the .jar file*” to execute the game
4. Have fun!

Exe:

1. Go here to download the .zip folder where you can find the .exe executable
2. Extract the folder you have just downloaded
3. Open the TLOJ.exe file
4. Have fun!

MacOs

You have two options to install the game on a MacOS machine. Both require you to download the [Java Developer Kit](#) (v - 19.0.2).

Important note:

We strongly advise to set the system on dark theme to have the best experience, otherwise the ASCII arts will not be shown as intended.

Maven:

1. Go to the [Maven official web page](#) and download it.
2. Follow [these](#) steps to set it up properly
3. Go to the [official repository on GitHub](#). You can choose to download the zip file or use git clone in a folder to download the code.
4. Now open the Command Prompt and go to the directory of the folder you have just downloaded
5. Use “mvn clean package” to compile the code
6. Use “mvn exec:java” to execute the code and play the game.
7. Have fun!

Jar:

5. TODO where to find .zip file
6. Go to the folder you just downloaded and open the Command Prompt
7. Use “java -jar “*Name of the .jar file*” to execute the game
8. Have fun!

Execution environment and Libraries

Execution environment and tools

System

The system has been compiled against versions of the Java Runtime 63.0 and JDK 19.0.2.

Development and Testing

The game has been developed and tested on Windows (10 and 11) and MacOS 11.7.10

Maven

Due to the complexity of the project we opted to use Maven software to handle its structure and dependencies

The Maven version used is 3.9.6

External libraries

Firebase

To fulfill the cloud save/load requirement we chose to use Firebase's storage solution. Its library allows access to the remote cloud storage where the system automatically saves the game data after the user quits.

The Firebase library version used is 9.2.0

Java Sound Api

To make the experience more immersive we decided to integrate some soundtracks.

To achieve that we used the Java Sound Api

Jansi

To improve the looks and the graphics of the game we used Jansi library to enable support for special ASCII characters and colors on Windows machines.

The Jansi version used is 2.3.2

Jackson

To save and load games both locally and on the cloud we store all the related data in JSON files.

The Jackson library handles the serialization/deserialization to/from JSON of such data.

The Jackson version used is 2.12.3