Adventure Game

Submitted for the Partial Fulfillment of the Requirements for the Degree of

#### Bachelor of Technology

in Computer Science and Engineering

**J.C. Bose University of Science and Technology, Faridabad**

**Submitted By:-**

Sahil Malik :17/cs34 Sanchit Mishra:17/cs35 Shivam Chandila: 17/cs37

**anangpuria**

**Under the Guidance of**

Deepika Ma’am Dr. PAWAN BHADANA

**(Project guide) (Principal)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING/INFORMATION TECHNOLOGY**

**B.S Anangpuria Institute of Technology & Management Alampur, Faridabad**

# Acknowledgement

We are over helmed in all humbleness and gratefulness to acknowledge my depth to all those who have helped me to put these ideas, well above the level of simplicity and into something concrete.

I would like to express my special thanks of gratitude to my teacher as well as our principal who gave me the golden opportunity to do this wonderful project on the topic "**Adventure Game**", which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them. Any attempt at any level can't be satisfactorily completed without the support and guidance of my parents and friends. I would like to thank my parents who helped me a lot in gathering different information, collecting data and guiding me from time to time in making this project, despite of their busy schedules, they gave me different ideas in making this project unique.

Thanking you,

**Sahil Malik (17/CS34) Sanchit Mishra (17/CS35) Shivam Chandila (17/CS37)**

# Index

|  |  |
| --- | --- |
| 1. | Abstract |
| 2. | Introduction |
| 3. | Problem statement |
| 4. | Proposed methodology |
| 5. | Requirements(Hardware and Software) |
| 6. | Diagrams |
| 7. | Results(output) |
| 8. | Conclusion |
| 9. | Future scope |
| 10. | References |

Abstract

This is a project based on Java.IO(Input Output) with JavaFx which is used for the GUI(Interface) .

# Introduction

* **What is Java.IO?**

**Java I/O** (Input and Output) is a Package in java which is used to process the **input** and produce the **output.** All these streams represent an input source and an output destination. The stream in the java.io package supports many data such as primitives, object, localized characters, etc.

A stream can be defined as a **sequence of data.**

* **What is JavaFX?**

**JavaFX** is a Java library that is used to develop Desktop applications as well as Rich Internet Applications (RIA). JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms.

* **What is the concept of this project?**

Similarly as the POKEMON games & series was based on where you could travel to locations while simply going Up-Down or East-West .There are 150 locations which we can keep on going through and when the project is fully fledged we while we able to walk around and interact with things in 2-Dimensionally.

# Problem

This is to implement how java.IO processes the data and takes the data from the files and further provides it to the GUI part which will have to work in one team together to provide a seamless interaction between front end and back end to provide a 2D game simulation.

# Proposed Methodology

* A class **Location** which can contain information about the Location that we are currently on.
* **Java.IO** to import data from the files and provide it to the process when the class in loaded into memory.
* **JavaFX** to create a GUI(Graphical User Interface) where the current Location will be shown and the directions where we can move next to.

# Requirements

## Hardware:

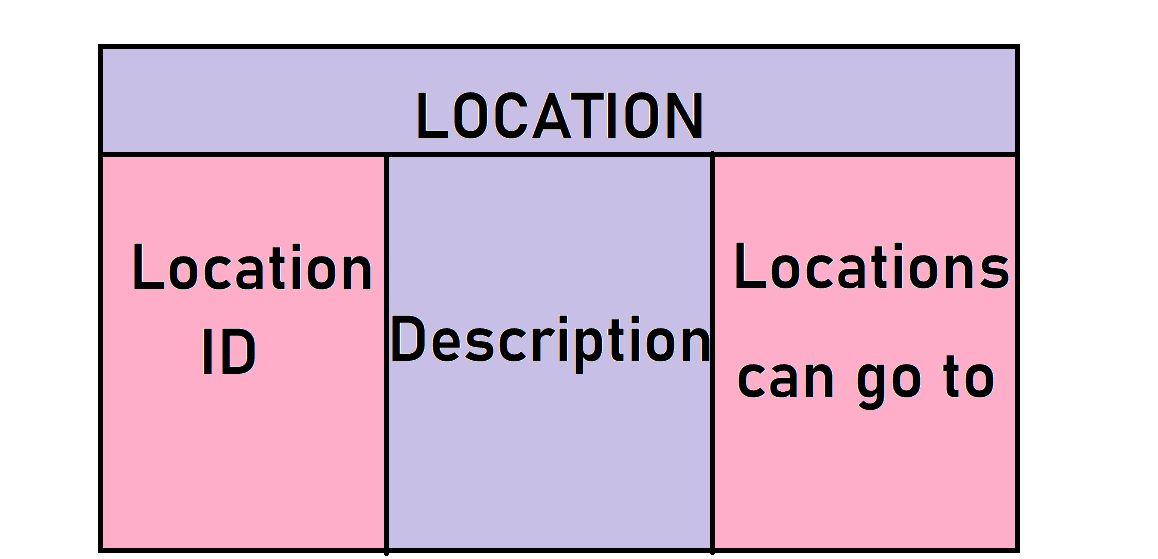
* CPU: 2 x 64-bit 2.8 GHz 8.00 GT/s CPUs
* RAM: 4 or above GB (or 16 GB of 1600 MHz DDR3 RAM)
* Storage: 300 GB. (600 GB for air-gapped deployments.) Additional space recommended if the repository will be used to store packages built by the customer. With an empty repository, a base install requires 2 GB.
* Internet access to download the files from Cloud or a USB drive containing all of the files you need with alternate instructions for air gapped installations.

## Software & system requirements:

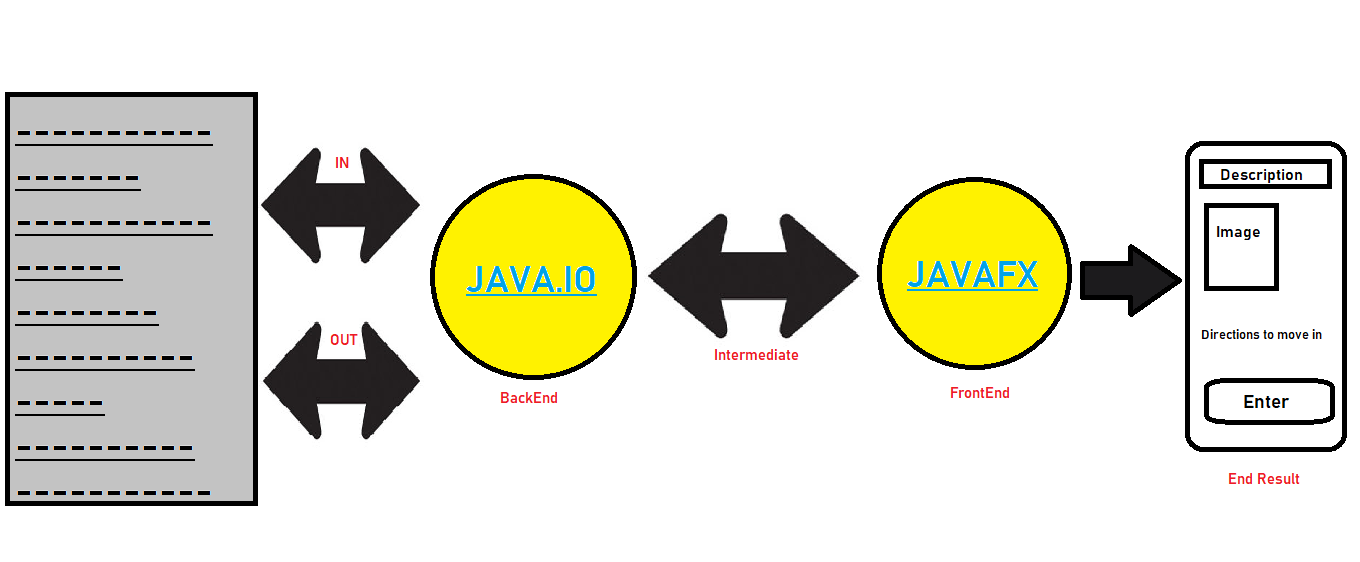
* RHEL/CentOS 6.5 to 7.4, Ubuntu 12.04+.
* JDK 8 or above.
* Intellij or eclipse.
* Ubuntu users may need to install cURL.
* JDK environment may be Windows, macOS or Linux.

# Diagrams/Flow Chart

Location Variable/Class



Intermediate Processes



Results

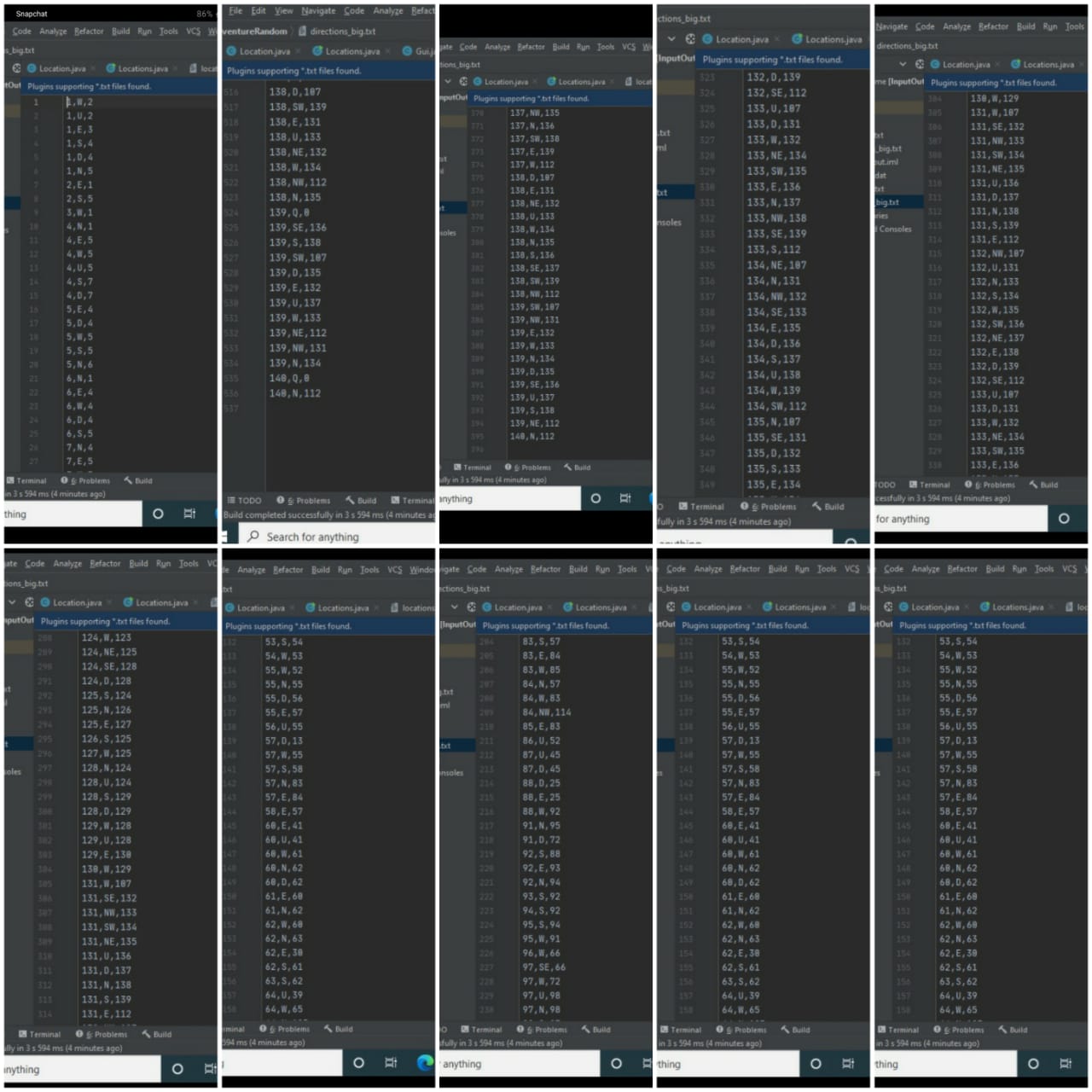
Dataset :

### Locations

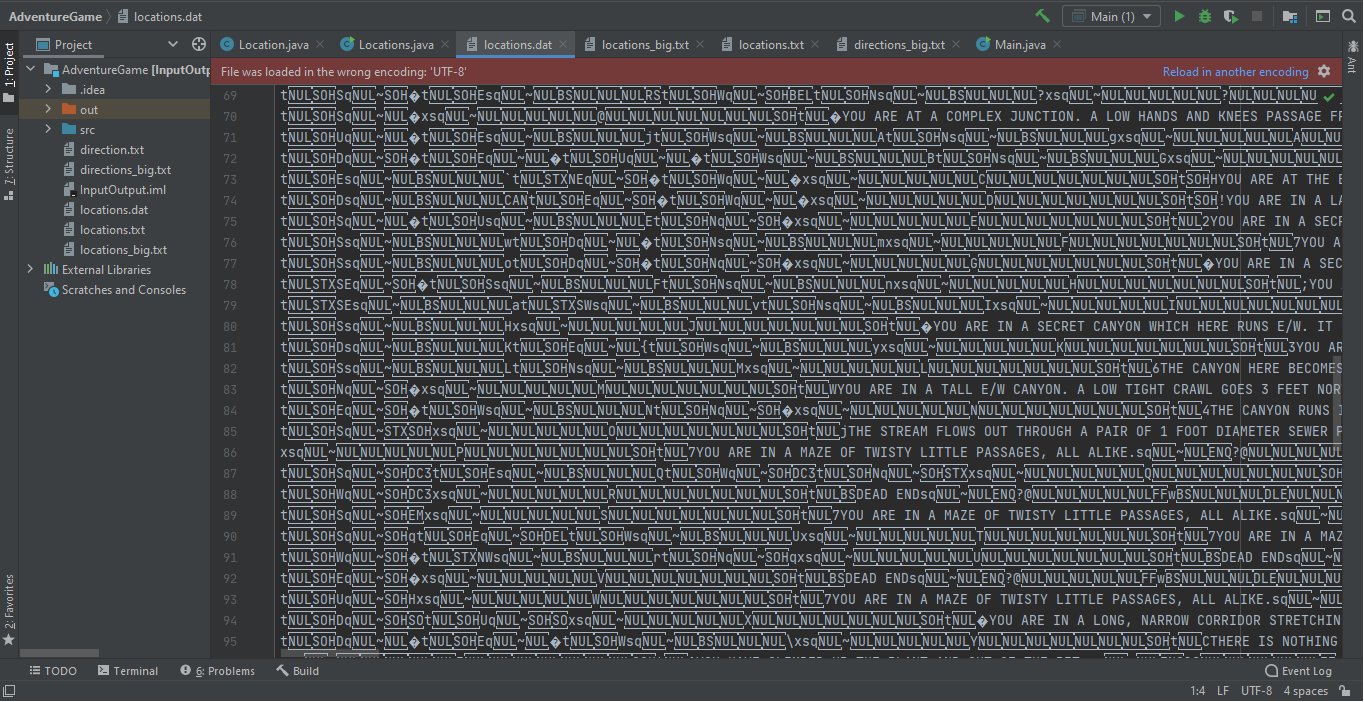
### C:\Users\user\Pictures\Screenshots\Screenshot (11).png

### C:\Users\user\Pictures\Screenshots\Screenshot (25).png

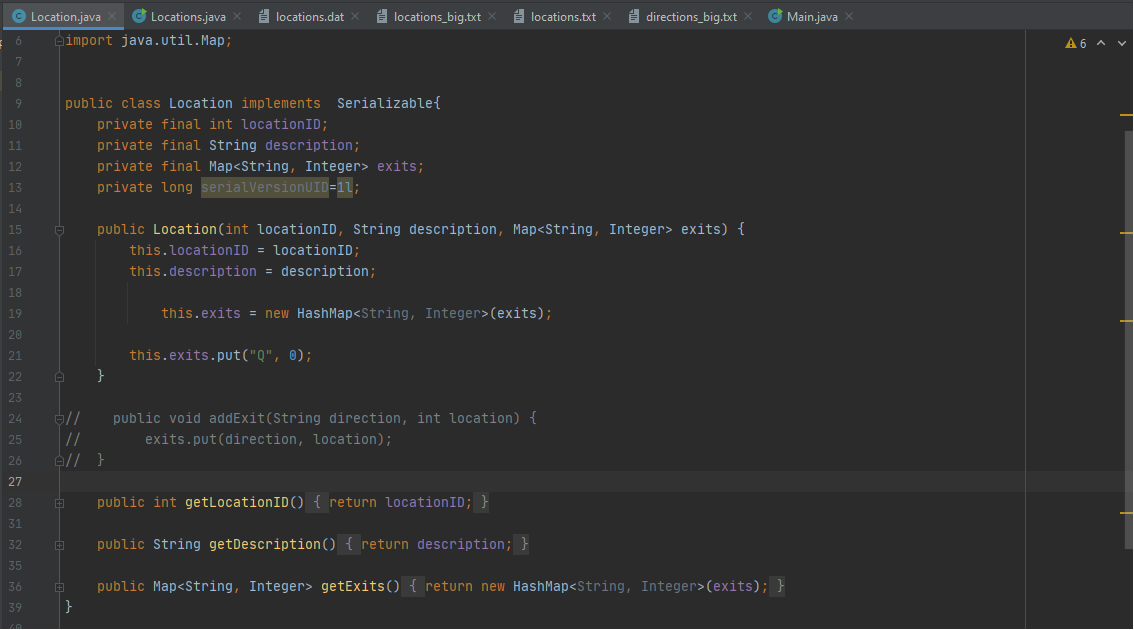
* Available Directions :



* Encrypted file in objects :



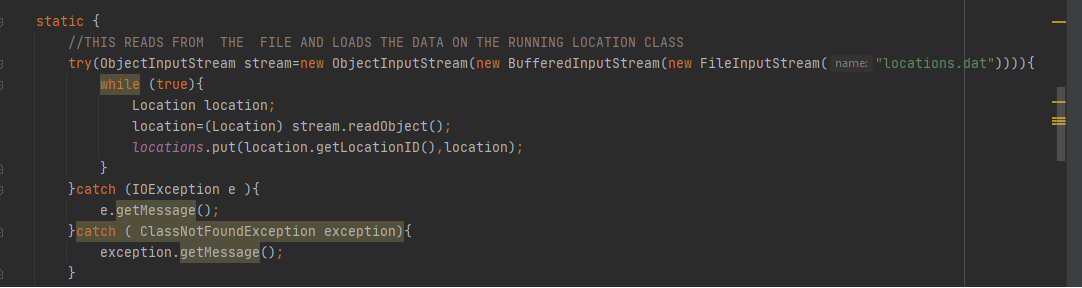
Location Class:



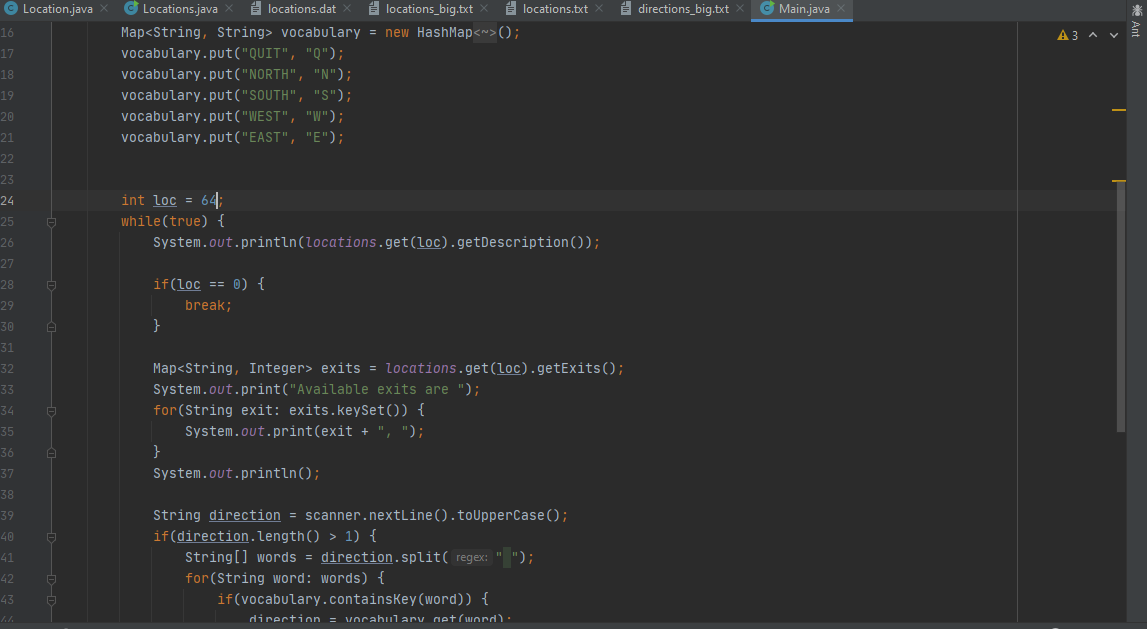
* For writing into the file :



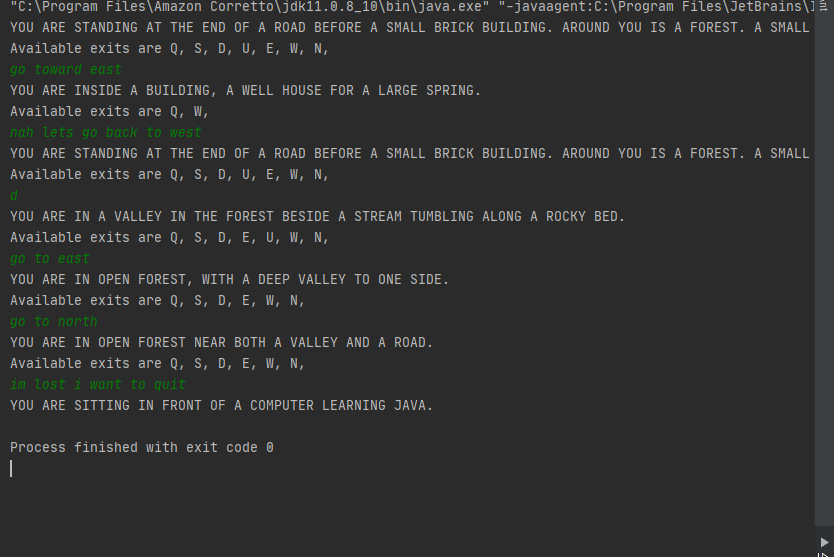
* For Reading from the file :



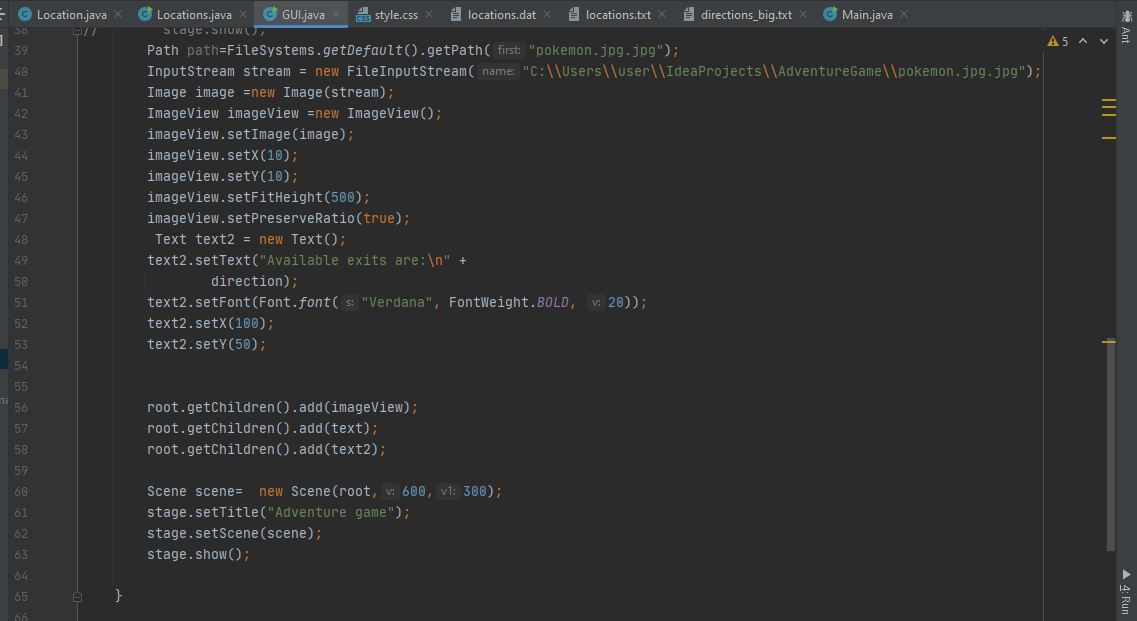
* Main Class :



* OUTPUT :



GUI:



RESULT:



# Conclusion

The conclusion of this project is when this project shows the interaction between GUI and the Files and classes which leads to a interactive display where we can travel to places.

# Future Scope

The future scope is that when this project comes at its peak at full fledged we can interact with things , move around in a area will lead to a seamless interaction.

# References

* Java tutorials
* Udemy.
* Github
* Oracle

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*