TITLE: CREATING A SIMPLE TEXT ADVENTURE GAME

INTRODUCTION:

In this report, we will discuss the process of creating a simple text adventure game. The game we developed was designed to be a fun and engaging experience for players, while also being easy to create with limited resources. Our team consisted of twenty-four members together to develop the game.

The list is attached to a separate file in the repository.

The code is a Java program that creates a simple GUI for a text-based adventure game. It uses the Swing library for creating the GUI components such as JFrame, JLabel, and JButton.

DEVELOPMENT PROCESS:

Storyline: The first step in creating the game was to develop a storyline. We decided on a fantasy theme, where the player would take on the role of a hero trying to escape from an island.

Descriptions: We wrote detailed descriptions for each location in the game, including descriptions of the environment, objects, and characters.

Puzzles: We created several puzzles for the player to solve throughout the game, including questions and answers.

Characters: We created non-playable characters for the player to interact with, including friendly characters who provided information and quests, and hostile characters who hindered the player's progress.

User Interface: We designed the user interface to be a with a java swing GUI where the user types in the desired options.

CODE EXPLANATION

The Main class has instance variables for JFrame, JLabel, and JPanel components, and methods for preparing the GUI and showing the event demo.

The prepareGui() method sets the properties of the JFrame component such as its title, size, and layout. It also creates the JLabel and JPanel components and adds them to the JFrame. Finally, it sets a window focus listener for the JFrame that exits the program when the window is closed.

The showEventDemo() method sets the properties of the JLabel and JButton components, and adds the JButton to the JPanel. It also sets an ActionListener for the JButton that disposes of the JFrame and creates a new Game object when the button is clicked.

The main method creates an instance of the Main class and calls the showEventDemo() method to start the GUI.

Overall, the code is a simple implementation of a GUI for a text-based adventure game, but it could be improved by using a more modular design and better separation of concerns. For example, the GUI could be split into multiple components or panels, and the game logic could be separated from the GUI components. Additionally, the code could be refactored to use more descriptive variable names and better coding practices.

CONCLUSION:

The process of creating the simple text adventure game was a fun and rewarding experience. We learned a lot about game design and development, and we were able to create a game that

we were proud of. We hope that this report will serve as a helpful guide for others who are interested in creating their own text adventure games.