Assignment 4 Overview

Saving and Loading

To save a game, I used the FileInputStream and FileOutputStream objects and I also used the ObjectInputStream and ObjectOutputStream objects. Using these IO objects, I created a Game class which stored the current Room, Player, and the Level of the game. This object was then stored to a binary file when the save button was pressed. When the load button is pressed, the object is loaded from the binary file. The binary file is called saved_game.txt.

Levels of Difficulty

In order to support different levels of difficulty. I created an interface called ILevel which has a method called createAdventure. Previously, the game was in a class called Adventure, but I refactored this name to be called LevelZero instead. LevelZero was also changed so that it implemented the ILevel interface. A new level called LevelOne was created and it also implemented ILevel. This ILevel interface will also support the creation of many other levels in the system. In order for the player to choose the level, I created a LevelFactory class, which contains a static method called chooseLevel which has a return type of ILevel, and it takes in an interger parameter. This method will create a new instance of Level 0 if 0 is passed in or a new instance of Level 1 if 1 is passed in. After New Game has been pressed, the Player will click on the number of the Level they wish to play and the creation of the proper Level is handled through this LevelFactory class.

Level Enhancements

At the beginning of Level 1, a player is given a flashlight which starts with a battery life of 100%. Every time a movement action is taken, the battery life of the flashlight will decrease 5%. If the Flashlight battery reaches 0%, the game will end and the player will be forced to begin a new game. There are two batteries that are placed in two different rooms. If a player picks up a battery, their flashlight charge will go back to 100%. Batteries are not placed in the players inventory, but the flashlight is.

The second enhancement made to the game is that in order to exit the cave and win the game, the player has to have found both the key and the oil can. If one of these items is missing, the player will be unable to beat the level. The oil and key are placed in different rooms. To add this functionality, a new RustedDoor object was created that inherits all the properties from the Door class, but it has an Oil object instead.

The last enhancement made to the game was the creation of a Portal object in one of the Rooms. If the Player chooses to enter the portal, they will be teleported to a random room. A player could win the game if the portal teleports them to the exit, but there is a small chance of this occurring.

Instructions

To start the game, you may either click on the executable AdventureGame.jar file which is included, or you may run the project in Eclipse. You will be given a flashlight at the beginning of the game. If your flashlight battery gets to 0% you will be unable to see and you will lose the game. To keep your flashlight charged, you must pick up batteries that are located in some of the rooms. To win, you must exit the cave without running out of light while picking up any necessary items required to unlock

the final door. Note that in one of the rooms is a magic door that will teleport you to a random room if you should choose to step through it.