Nevin Leh
Software Engineering A4

1. To save data I first created a class called Data. In this I saved the Adventure and Player classes. I then serialized this object. This means I had to implement java.io.Serializable on the above classes along with any classes those classes contained. I then saved this object in /tmp/adventureGame.ser with ObjectOutputStream. All of this is in the saveGame method of AdventureGameModelFacade.

To load the game I open /tmp/adventureGame.ser with on ObjectInputStream and then instantiate the classes as the ones in the AdventureGameModelFacade. I created a save button and a load button to start these steps. I relied heavily on some code provide at http://www.tutorialspoint.com/java/java_serialization.htm

2.To use the abstract factory design pattern I first created an adventureFactory. This factory a createClass method for all classes needed in the design. It did not have some classes because these were instantiated by the classes within adventureFactory and were not affected by changing of levels.

I then created two classes that extended AdventureFactory: ZeroFactory and OneFactory. These were the factories for level zero and one. Each had the methods from adventureFactory that instiated the object needed for either level one or level zero. Some things such as Key and Treasure remained unchanged between levels so they were left alone. Others such as Player had to be changed so an interface was made for these. With ZeroPlayer and OnePlayer implementing the Player Interface.

3.The changes I made consisted of giving the player three lives and making objects that, when picked up, would take these lives. To do this I first added a lives variable to the OnePlayer class. I also made a Trap class that extends Item. I changed some of the logic in the pickUp method of OnePlayer so a life would be lost If a Trap item was picked up. If it is the third trap, the program abruptly ends. The trap also disappears, ending up neither in the room or the inventory. To test this there are three traps named “a dangerous looking toad”. Pick all three up and the program should end.

4. Saving and loading- There is a save game button on the gui. Pressing it will save the current state of the game. To load a game press the loadgame button and the most recently saved game will come up.

Changing levels-Press the change difficulty button and select what difficulty. This will restart the game at the chosen difficulty.

Picking up and dropping items items-Enter the index of the item you want to pick up or drop into the third text box and press pick up or drop.

5.Just import breezy swing and press run!