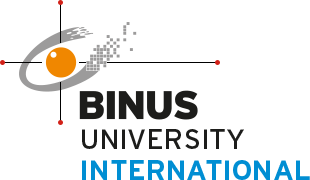
Even Semester (2023)



# BINUS UNIVERSITY

# BINUS INTERNATIONAL

|  |
| --- |
| **Final Project Cover Letter****(Individual Work)** |

**Student Information:**

**Surname:** Ruby **Given Name:** Louis **Student ID:** 2602185431

**Course Code :** COMP6699001 **Course Name :** Object Oriented Programming

**Class :** L2CC **Lecturer:** Jude Joseph Lamug Martinez, MCS

**Type of Assignment:** Final Project Report

**Submission Pattern**

**Due Date :** 16 June 2023 **Submission Date :** 16 June 2023

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

**Plagiarism/Cheating**

BiNus International seriously regards all forms of plagiarism, cheating, and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity, and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

**Declaration of Originality**

By signing this assignment, I understand, accept, and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

Louis

**Table of Contents**

[**BINUS UNIVERSITY 1**](#_19cx9o9obpvi)

[**BINUS INTERNATIONAL 1**](#_jo3eoycuxwgl)

[Final Project Cover Letter 1](#_gowitmewym32)

[(Individual Work) 1](#_yu7qx7pvtfzf)

[**Project Specifications 4**](#_qntzhc2ck0l0)

[**Solution Design 5**](#_db7nxsb37mel)

[**Implementations 5**](#_ewrs23c43w9w)

[**Classes**](#_y9ixzt1svg0a) **6**

[**Evidence of Working Program 18**](#_l1yckrbe4nmi)

# Project Specifications

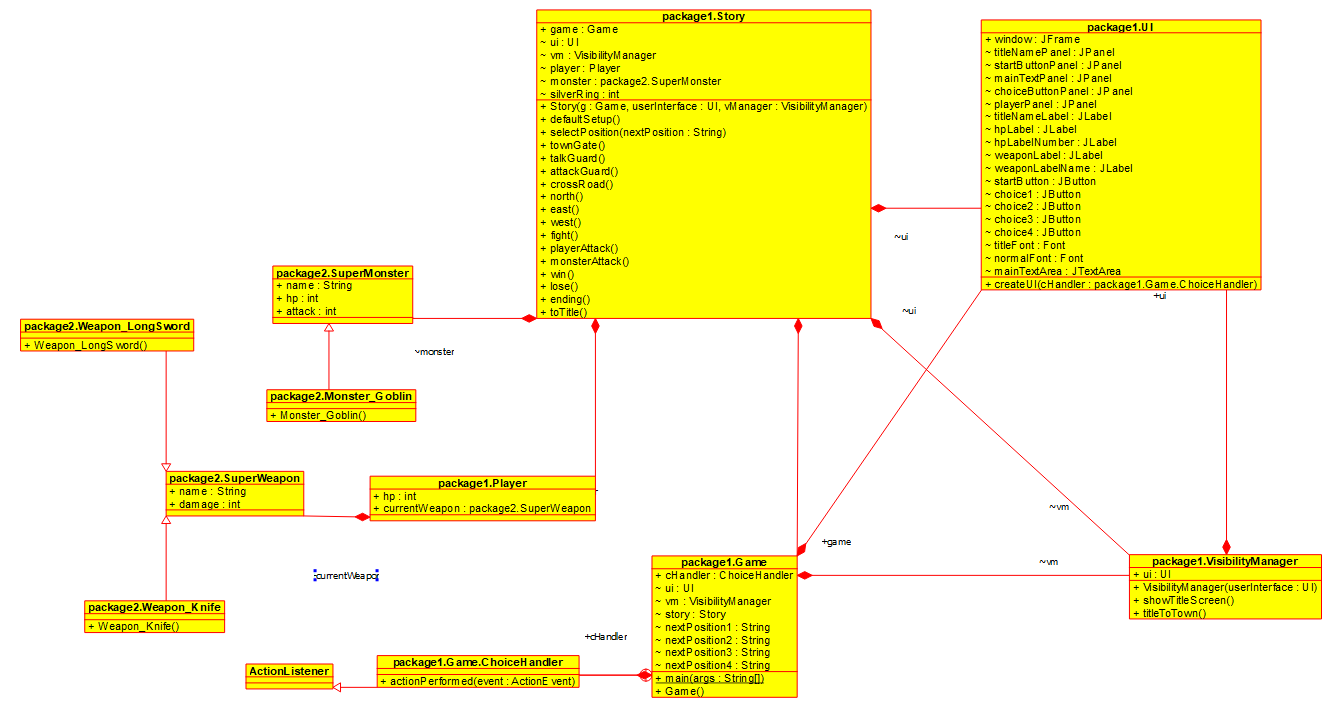
**Problem to Solve**

Boredom is a commonly found within time without doing anything. Boredom can be considered a problem for several reasons, even though it can be beneficial in stimulating creativity, it can also have negative effects on an individual’s well-being and overall quality of life.

**Solution**

By playing games, individual can alleviate boredom and be entertained. Engaging in games can stimulate the mind, by providing a sense of challenge and accomplishment. It is also a way to pass time. This text-based story game can help with boredom.

# Solution Design

.

**Library Used**

* java.awt.event.ActionEvent

Represents an action event such as buttons, menu item, or a checkbox when is activated by user. To handle this, ActionListener is needed.

* java.awt.event.ActionListener

It defines a contract for objects that want to handle action events generated by buttons, menu items, or checkboxes.

* javax.swing

Creates graphical user interfaces (GUI).

* java.awt

Handling graphic related operations.

**Classes**

**Game Class**

This is the code to start the game. It handles which position are the user heading depending on their choice.

package package1;  
  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class Game {  
 ChoiceHandler cHandler = new ChoiceHandler();  
 UI ui = new UI();  
 VisibilityManager vm = new VisibilityManager(ui);  
 Story story = new Story(this, ui, vm);  
 String nextPosition1,nextPosition2,nextPosition3,nextPosition4;  
  
 public static void main(String[] args) {  
 new Game();  
 }  
  
 public Game() {  
 ui.createUI(cHandler);  
 story.defaultSetup();  
 vm.showTitleScreen();  
 }  
  
 public class ChoiceHandler implements ActionListener {  
  
 public void actionPerformed(ActionEvent event) {  
 String yourChoice = event.getActionCommand();  
  
 switch(yourChoice){  
 case "start" : vm.titleToTown();story.townGate();break;  
 case "c1" : story.selectPosition(nextPosition1);break;  
 case "c2" : story.selectPosition(nextPosition2);break;  
 case "c3" : story.selectPosition(nextPosition3);break;  
 case "c4" : story.selectPosition(nextPosition4);break;  
 }  
 }  
 }  
}

**Player Class**

Super class for the object Player.

package package1;  
  
import package2.SuperWeapon;  
  
public class Player {  
 public int hp;  
 public SuperWeapon currentWeapon;  
}

**Story Class**

Story texts.

package package1;  
  
import package2.Monster\_Goblin;  
import package2.SuperMonster;  
import package2.Weapon\_Knife;  
import package2.Weapon\_LongSword;  
  
public class Story {  
 Game game;  
 UI ui;  
 VisibilityManager vm;  
 Player player = new Player();  
 SuperMonster monster;  
 int silverRing;  
  
 public Story(Game g, UI userInterface, VisibilityManager vManager){  
 game = g;  
 ui = userInterface;  
 vm = vManager;  
 }  
 public void defaultSetup(){  
 player.hp = 10;  
 ui.hpLabelNumber.setText("" + player.hp);  
 player.currentWeapon = new Weapon\_Knife();  
 ui.weaponLabelName.setText(player.currentWeapon.name);  
 silverRing = 0;  
 }  
 public void selectPosition(String nextPosition){  
 switch(nextPosition){  
 case "townGate": townGate();break;  
 case"talkGuard": talkGuard();break;  
 case"attackGuard": attackGuard();break;  
 case"crossRoad": crossRoad();break;  
 case"north": north();break;  
 case"east":east();break;  
 case"west": west();break;  
 case"fight":fight();break;  
 case"playerAttack":playerAttack();break;  
 case"monsterAttack":monsterAttack();break;  
 case"win":win();break;  
 case"lose":lose();break;  
 case"ending":ending();break;  
 case"toTitle":toTitle();break;  
 }  
 }  
  
 public void townGate(){  
 ui.mainTextArea.setText("You are at the gate of the town. \nEntering the town is your main objective \nA guard is standing in front of you. \n\nWhat do you do?");  
 ui.choice1.setText("Talk to the guard");  
 ui.choice2.setText("Attack the guard");  
 ui.choice3.setText("Explore");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "talkGuard";  
 game.nextPosition2 = "attackGuard";  
 game.nextPosition3 = "crossRoad";  
 game.nextPosition4 = "";  
 }  
 public void talkGuard(){  
 if(silverRing==0){  
 ui.mainTextArea.setText("Guard: Hello stranger. I have never seen your face. \nI'm sorry but we cannot let a stranger enter our town.");  
 ui.choice1.setText("Proceed");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "townGate";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 else if(silverRing ==1){  
 ending();  
 }  
 }  
 public void attackGuard(){  
 ui.mainTextArea.setText("Guard: Hey don't do that!\n\nThe guard fought back and hit you hard.\n(You receive 4 damage)");  
 player.hp = player.hp -4;  
 ui.hpLabelNumber.setText("" + player.hp);  
 ui.choice1.setText("Proceed");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "townGate";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void crossRoad(){  
 ui.mainTextArea.setText("You are at a crossroad.\nIf you go south, you will go back to the town.");  
 ui.choice1.setText("Heal on the river");  
 ui.choice2.setText("Head to the forest");  
 ui.choice3.setText("Go back to town's gate");  
 ui.choice4.setText("Fight Goblin");  
  
 game.nextPosition1 = "north";  
 game.nextPosition2 = "east";  
 game.nextPosition3 = "townGate";  
 game.nextPosition4 = "west";  
 }  
 public void north(){  
 ui.mainTextArea.setText("You reached the river of immortality. \nYou drink the water and rest at the riverside. \n\n(Your HP is recovered by 2)");  
 player.hp = player.hp +2;  
 ui.hpLabelNumber.setText("" + player.hp);  
 ui.choice1.setText("Go back");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "crossRoad";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void east(){  
 ui.mainTextArea.setText("You walked into a forest and found a Long Sword!\n\n(You obtained a Long Sword)");  
 player.currentWeapon = new Weapon\_LongSword();  
 ui.weaponLabelName.setText(player.currentWeapon.name);  
 ui.choice1.setText("Go back");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "crossRoad";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void west(){  
 monster = new Monster\_Goblin();  
 ui.mainTextArea.setText("You fight the GOBLIN head on!!");  
 ui.choice1.setText("Fight");  
 ui.choice2.setText("Run");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "fight";  
 game.nextPosition2 = "crossRoad";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void fight(){  
 ui.mainTextArea.setText("Goblin : " + monster.hp + "\nWhat do you do?");  
 ui.choice1.setText("Attack");  
 ui.choice2.setText("Run");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "playerAttack";  
 game.nextPosition2 = "crossRoad";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void playerAttack(){  
 int playerDamage = new java.util.Random().nextInt(player.currentWeapon.damage);  
  
 ui.mainTextArea.setText("You attacked the Goblin and gave " + playerDamage + "damage!");  
 monster.hp = monster.hp - playerDamage;  
  
 ui.choice1.setText("Proceed");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 if(monster.hp>0){  
 game.nextPosition1 = "monsterAttack";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 else if(monster.hp<1){  
 game.nextPosition1 = "win";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 }  
 public void monsterAttack(){  
 int monsterDamage = new java.util.Random().nextInt(monster.attack);  
 player.hp = player.hp - monsterDamage;  
 ui.hpLabelNumber.setText("" + player.hp);  
  
 ui.choice1.setText("Proceed");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 if(player.hp>0){  
 game.nextPosition1 = "fight";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 else if(player.hp<1){  
 game.nextPosition1 = "lose";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 }  
 public void win(){  
 ui.mainTextArea.setText("You defeated the monster!\nThe monster dropped a ring!\n\n(You obtained a Silver Ring)");  
 silverRing = 1;  
  
 ui.choice1.setText("Go back");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "crossRoad";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void lose(){  
 ui.mainTextArea.setText("You are dead!\n\nGAME OVER");  
  
 ui.choice1.setText("To the title screen");  
 ui.choice2.setText("");  
 ui.choice3.setText("");  
 ui.choice4.setText("");  
  
 game.nextPosition1 = "toTitle";  
 game.nextPosition2 = "";  
 game.nextPosition3 = "";  
 game.nextPosition4 = "";  
 }  
 public void ending(){  
 ui.mainTextArea.setText("Guard: Is that the Silver Ring?\nOh you killed that goblin!?\nYou are truly remarkable!\nWelcome to our town!\n\nTHE END");  
 ui.choice1.setVisible(false);  
 ui.choice2.setVisible(false);  
 ui.choice3.setVisible(false);  
 ui.choice4.setVisible(false);  
 }  
 public void toTitle(){  
 defaultSetup();  
 vm.showTitleScreen();  
 }  
}

**UI Class**

Handles the UI. Sizing, fonts and colors.

package package1;  
import javax.swing.\*;  
import java.awt.\*;  
  
public class UI {  
 JFrame window;  
 // to create window  
 JPanel titleNamePanel, startButtonPanel, mainTextPanel, choiceButtonPanel, playerPanel;  
 // display panels  
 JLabel titleNameLabel, hpLabel, hpLabelNumber, weaponLabel, weaponLabelName;  
 // display texts  
 JButton startButton, choice1, choice2, choice3, choice4;  
 // buttons  
 Font titleFont = new Font("Times New Roman", Font.*PLAIN*, 90);  
 // setting font and size for title  
 Font normalFont = new Font("Times New Roman", Font.*PLAIN*, 28);  
 // setting font and size for start  
 JTextArea mainTextArea;  
  
  
 public void createUI(Game.ChoiceHandler cHandler){  
  
 //WINDOW  
 //initialize window  
 window = new JFrame();  
 window.setSize(800, 600);  
 //close button  
 window.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 // bg color  
 window.getContentPane().setBackground(Color.*white*);  
 //disable default layout for JFrame in order for our custom layout to work  
 window.setLayout(null);  
  
 //TITLE SCREEN  
 // initialize panel  
 titleNamePanel = new JPanel();  
 // Panel size  
 titleNamePanel.setBounds(100, 100, 600, 150);  
 // Color of panel  
 titleNamePanel.setBackground(Color.*white*);  
 // Title  
 titleNameLabel = new JLabel("ADVENTURE");  
 // Color of text  
 titleNameLabel.setForeground(Color.*black*);  
 // To call titleFont and use the font and size  
 titleNameLabel.setFont(titleFont);  
 // add panels  
 titleNamePanel.add(titleNameLabel);  
  
 //initialize startButtonPanel  
 startButtonPanel = new JPanel();  
 //set size of panels  
 startButtonPanel.setBounds(300, 400, 200, 100);  
 //color of button panel  
 startButtonPanel.setBackground(Color.*white*);  
 //initialize start button  
 startButton = new JButton("START");  
 //color  
 startButton.setBackground(Color.*white*);  
 //color of text  
 startButton.setForeground(Color.*black*);  
 // setting fonts  
 startButton.setFont(normalFont);  
 //ignore lines that will form by default  
 startButton.setFocusPainted(false);  
 // so button works  
 startButton.addActionListener(cHandler);  
 startButton.setActionCommand("start");  
 startButtonPanel.add(startButton);  
  
 //to make them show  
 window.add(titleNamePanel);  
 window.add(startButtonPanel);  
  
  
 //GAME SCREEN  
 //initialize new panel  
 mainTextPanel = new JPanel();  
 //set size of new panel  
 mainTextPanel.setBounds(100, 100, 600, 250);  
 //color  
 mainTextPanel.setBackground(Color.*white*);  
 //font and sizing  
 window.add(mainTextPanel);  
  
 // main text area  
 mainTextArea = new JTextArea();  
 // setting size  
 mainTextArea.setBounds(100, 100, 600, 250);  
 // background color  
 mainTextArea.setBackground(Color.*white*);  
 // text color  
 mainTextArea.setForeground(Color.*black*);  
 // font  
 mainTextArea.setFont(normalFont);  
 // helps with making the words not overlapping the window  
 mainTextArea.setLineWrap(true);  
 mainTextArea.setWrapStyleWord(true);  
 mainTextArea.setEditable(false);  
 //to show it on the panel  
 mainTextPanel.add(mainTextArea);  
  
 //initialize button panels  
 choiceButtonPanel = new JPanel();  
 //sizing  
 choiceButtonPanel.setBounds(250, 350, 300, 150);  
 //color  
 choiceButtonPanel.setBackground(Color.*white*);  
 // custom layout instead of default  
 choiceButtonPanel.setLayout(new GridLayout(4,1));  
 // add choiceButtonPanel to the window  
 window.add(choiceButtonPanel);  
 // first choice button  
 choice1 = new JButton("Choice 1");  
 // color  
 choice1.setBackground(Color.*white*);  
 // text color  
 choice1.setForeground(Color.*black*);  
 // font  
 choice1.setFont(normalFont);  
 // without this there will be lines on the option buttons  
 choice1.setFocusPainted(false);  
 // for it works when click  
 choice1.addActionListener(cHandler);  
 // for the program to differentiate choice 1, 2, 3, and 4  
 choice1.setActionCommand("c1");  
 // adding the first button panel to the choice panels area  
 choiceButtonPanel.add(choice1);  
 choice2 = new JButton("Choice 2");  
 choice2.setBackground(Color.*white*);  
 choice2.setForeground(Color.*black*);  
 choice2.setFont(normalFont);  
 choice2.setFocusPainted(false);  
 choice2.addActionListener(cHandler);  
 choice2.setActionCommand("c2");  
 choiceButtonPanel.add(choice2);  
 choice3 = new JButton("Choice 3");  
 choice3.setBackground(Color.*white*);  
 choice3.setForeground(Color.*black*);  
 choice3.setFont(normalFont);  
 choice3.setFocusPainted(false);  
 choice3.addActionListener(cHandler);  
 choice3.setActionCommand("c3");  
 choiceButtonPanel.add(choice3);  
 choice4 = new JButton("Choice 4");  
 choice4.setBackground(Color.*white*);  
 choice4.setForeground(Color.*black*);  
 choice4.setFont(normalFont);  
 choice4.setFocusPainted(false);  
 choice4.addActionListener(cHandler);  
 choice4.setActionCommand("c4");  
 choiceButtonPanel.add(choice4);  
 // choice 1, 2, 3, and 4 is all the same function  
  
 //initialize panel on top for hp and weapon  
 playerPanel = new JPanel();  
 //setting size of panel  
 playerPanel.setBounds(100, 15, 600, 50);  
 // bg color  
 playerPanel.setBackground(Color.*white*);  
 // custom layout  
 playerPanel.setLayout(new GridLayout(1,4));  
 // adding playerPanel on the window  
 window.add(playerPanel);  
 //initialize hp label  
 hpLabel = new JLabel("HP:");  
 //font  
 hpLabel.setFont(normalFont);  
 // text color  
 hpLabel.setForeground(Color.*black*);  
 // add hp text to the player panel  
 playerPanel.add(hpLabel);  
 // initialize hp label number  
 hpLabelNumber = new JLabel();  
 // font  
 hpLabelNumber.setFont(normalFont);  
 // text color  
 hpLabelNumber.setForeground(Color.*black*);  
 // adding hp number panel to the player panel  
 playerPanel.add(hpLabelNumber);  
 // initialize weapon  
 weaponLabel = new JLabel("Weapon:");  
 // font  
 weaponLabel.setFont(normalFont);  
 // text color  
 weaponLabel.setForeground(Color.*black*);  
 // bg  
 weaponLabel.setBackground(Color.*red*);  
 // adding weapon text to the player panel  
 playerPanel.add(weaponLabel);  
 // initialize the weapon name text  
 weaponLabelName = new JLabel();  
 // font  
 weaponLabelName.setFont(normalFont);  
 // text color  
 weaponLabelName.setForeground(Color.*black*);  
 // adding weapon label name to the player panel  
 playerPanel.add(weaponLabelName);  
  
 //make everything appear on screen  
 window.setVisible(true);  
 }  
}

**VisibilityManager Class**

Controls which window is showing depending on the user choice.

package package1;  
  
public class VisibilityManager {  
 UI ui;  
 public VisibilityManager(UI userInterface){  
 ui = userInterface;  
 }  
 public void showTitleScreen(){  
 //Show title screen  
 ui.titleNamePanel.setVisible(true);  
 ui.startButtonPanel.setVisible(true);  
  
 //Hide game screen  
 ui.mainTextPanel.setVisible(false);  
 ui.choiceButtonPanel.setVisible(false);  
 ui.playerPanel.setVisible(false);  
 }  
 public void titleToTown(){  
 // Hide title screen  
 ui.titleNamePanel.setVisible(false);  
 ui.startButtonPanel.setVisible(false);  
  
 //Show game screen  
 ui.mainTextPanel.setVisible(true);  
 ui.choiceButtonPanel.setVisible(true);  
 ui.playerPanel.setVisible(true);  
 }  
}

**Monster\_Goblin Class**

Monster’s name, hp and attack.

package package2;  
  
public class Monster\_Goblin extends SuperMonster{  
 public Monster\_Goblin(){  
 name = "Goblin";  
 hp = 20;  
 attack = 6;  
 }  
}

**SuperMonster Class**

Super class for object Monster.

package package2;  
  
public class SuperMonster {  
 public String name;  
 public int hp;  
 public int attack;  
}

**SuperWeapon Class**

Super class for object Weapon

package package2;  
  
public class SuperWeapon {  
 public String name;  
 public int damage;  
}

**Weapon\_Knife Class**

Knife name and damage.

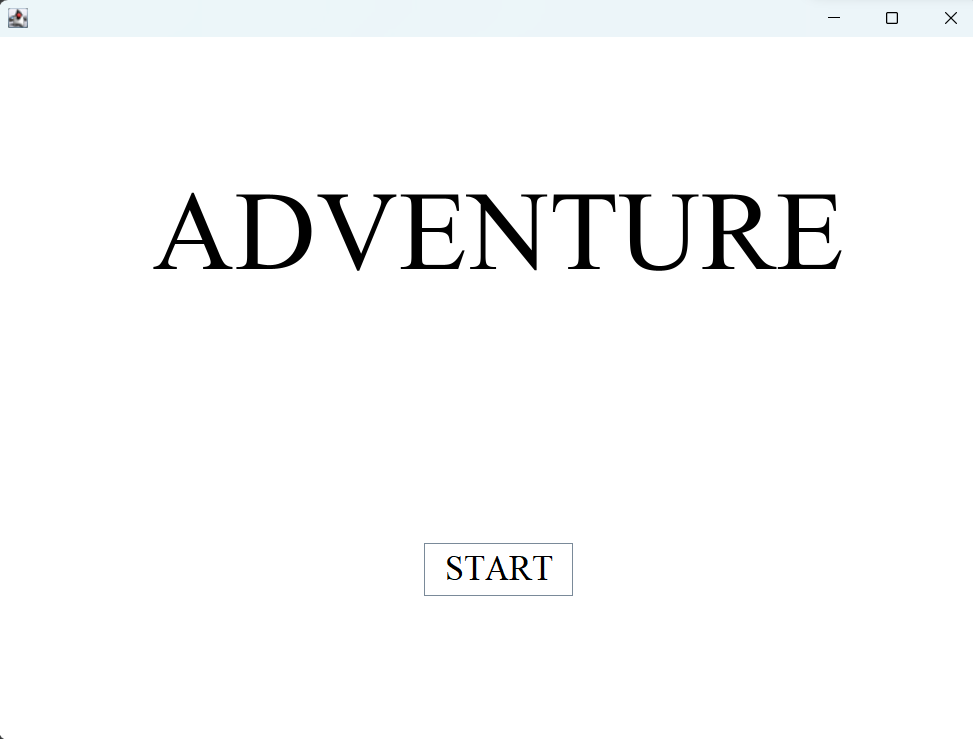
package package2;  
  
public class Weapon\_Knife extends SuperWeapon{  
 public Weapon\_Knife(){  
 name = "Knife";  
 damage = 3;  
 }  
}

**Weapon\_LongSword Class**

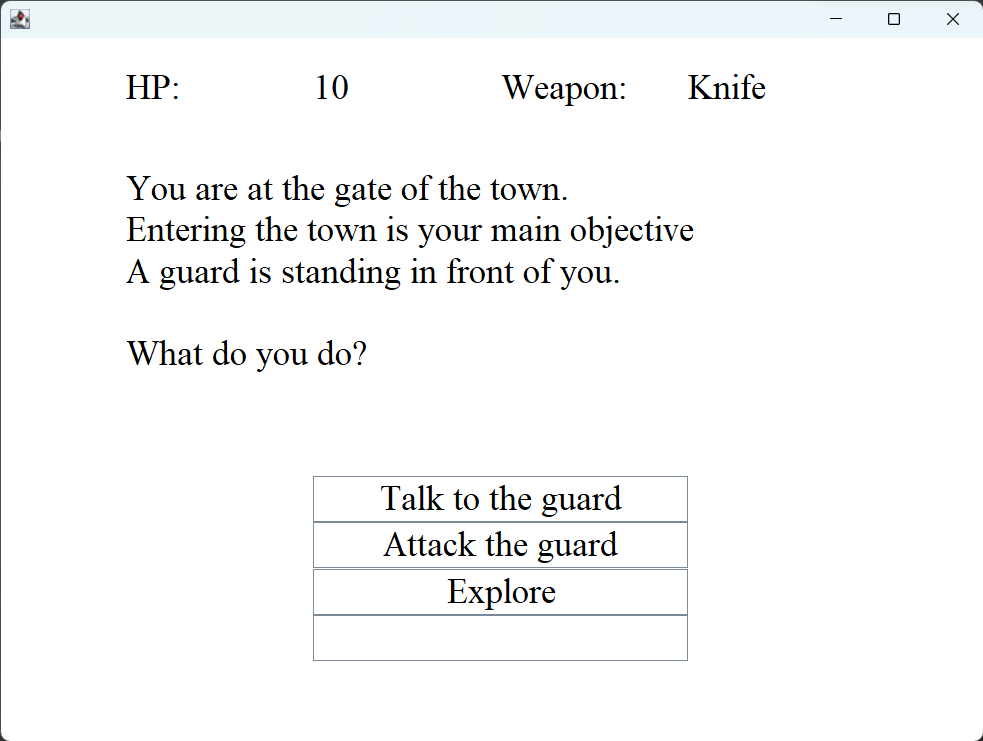
Long Sword name and damage.

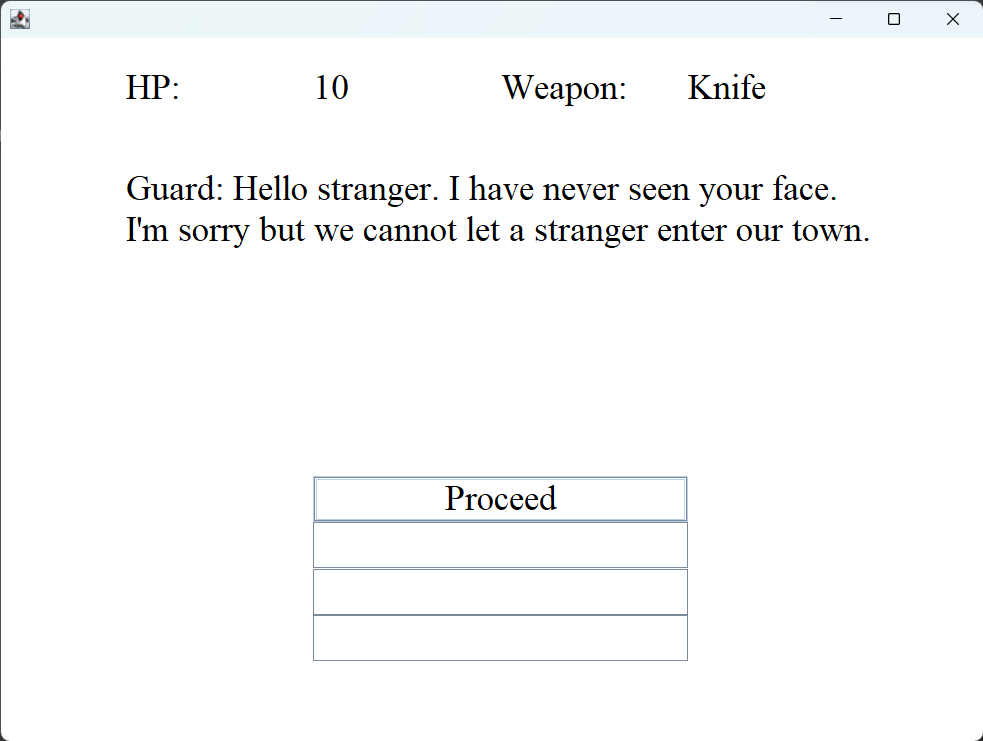
package package2;  
  
public class Weapon\_LongSword extends SuperWeapon{  
 public Weapon\_LongSword(){  
 name = "Long Sword";  
 damage = 8;  
 }  
}

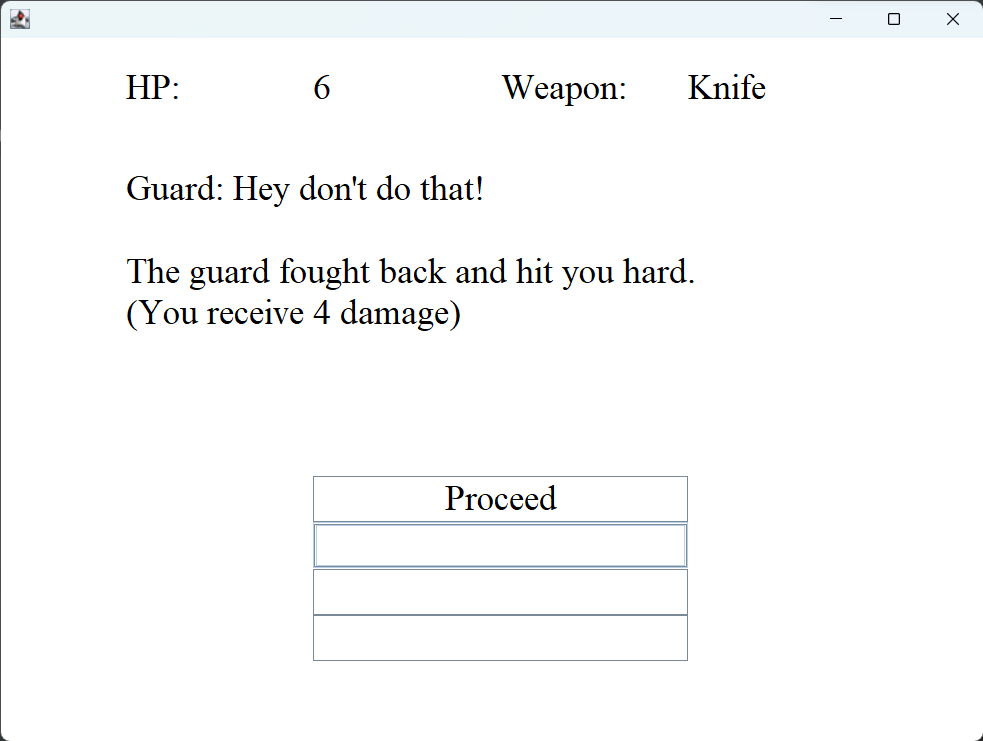
**Evidence Of Working Program**

**Start Screen**

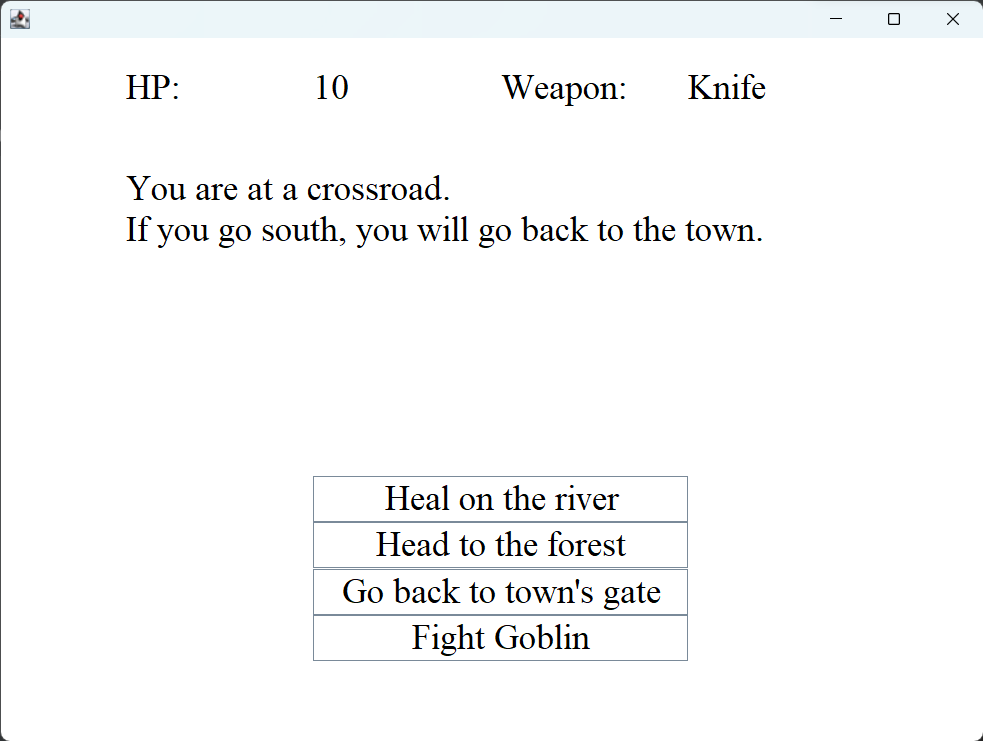
**Town’s Gate**

****

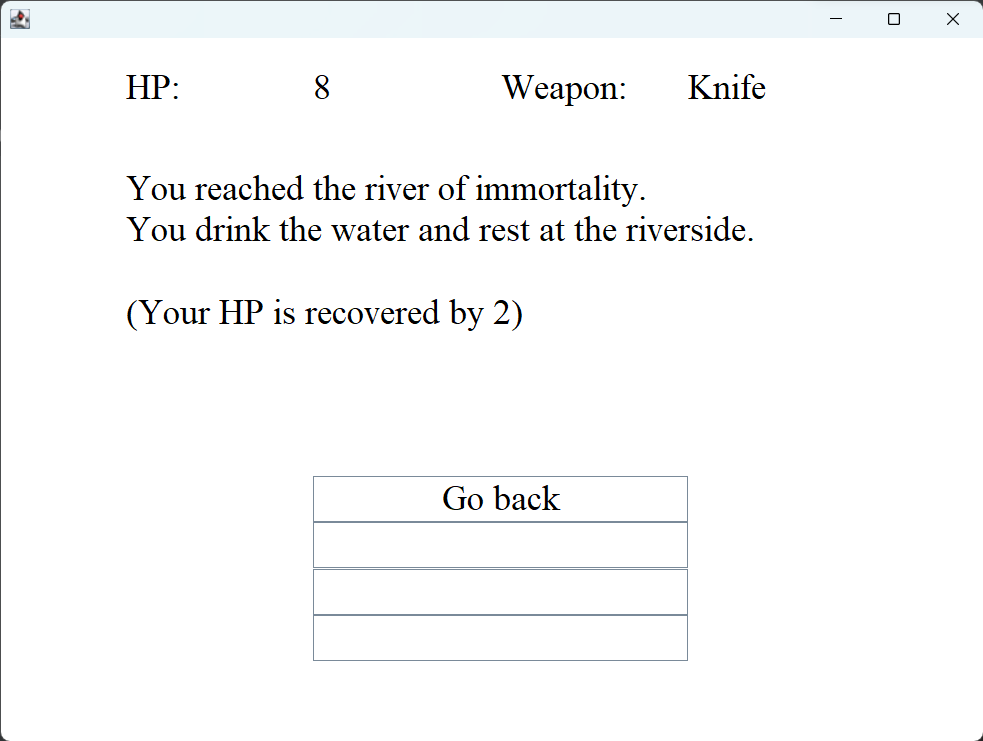
**Talk To Guard**

**Attack Guard**

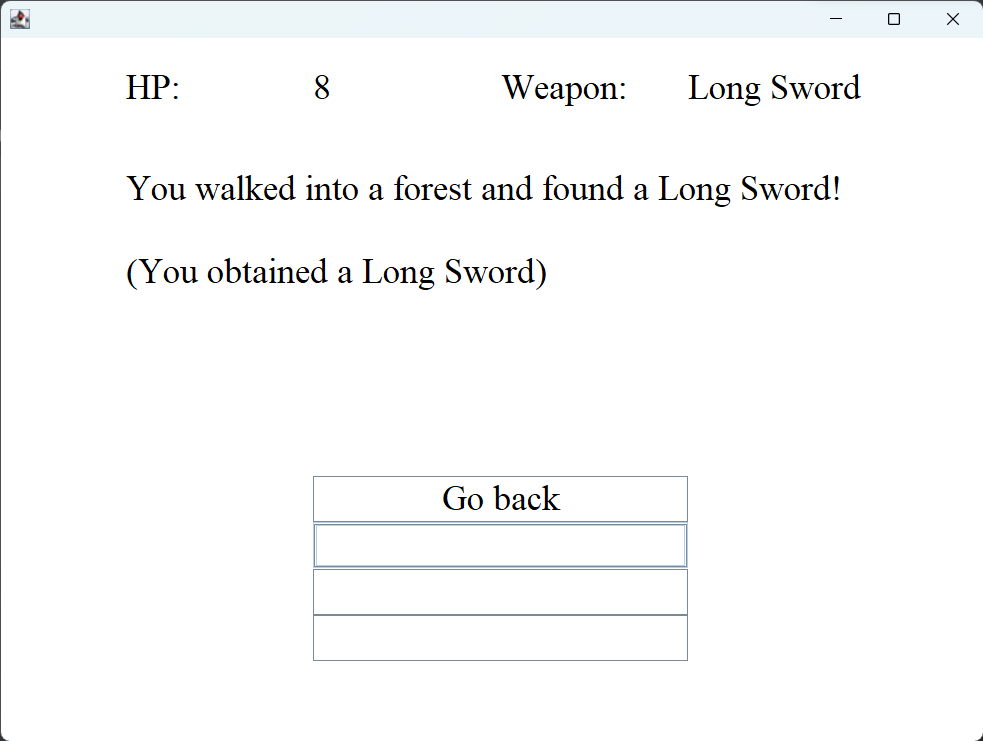
**Explore**

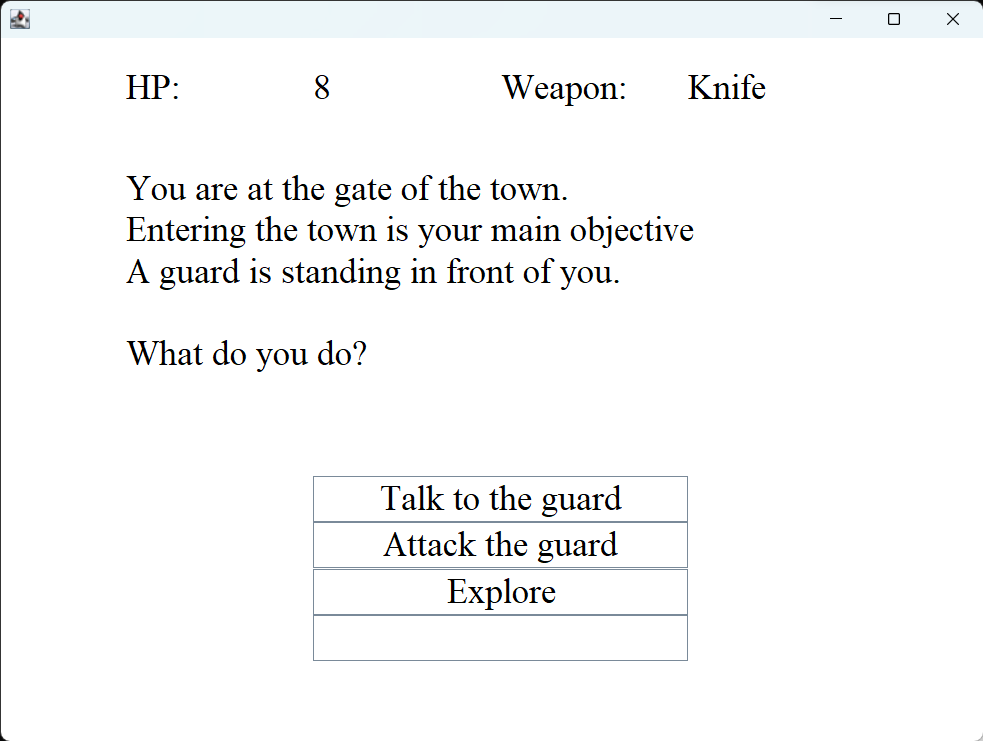
****

**Heal**

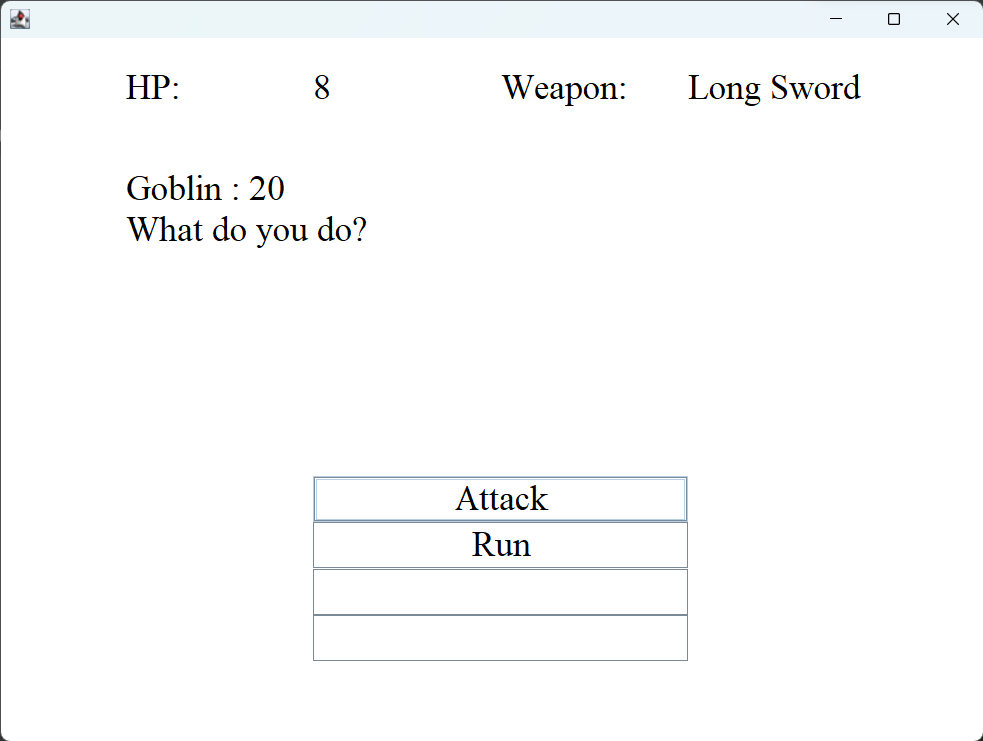
****

**Forest**

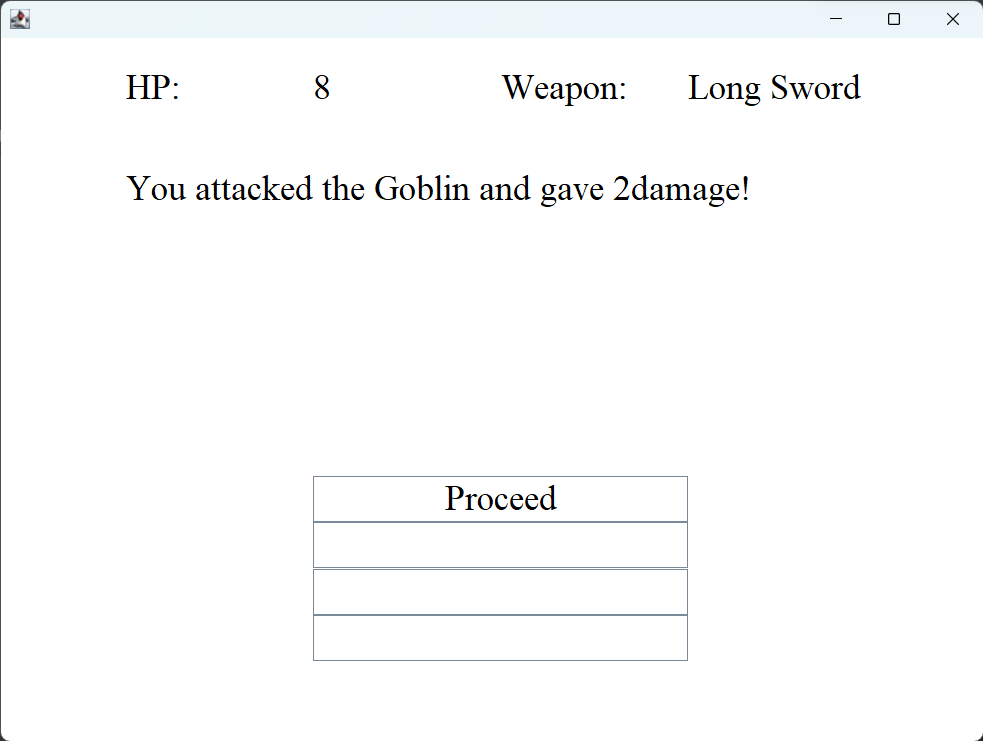
****

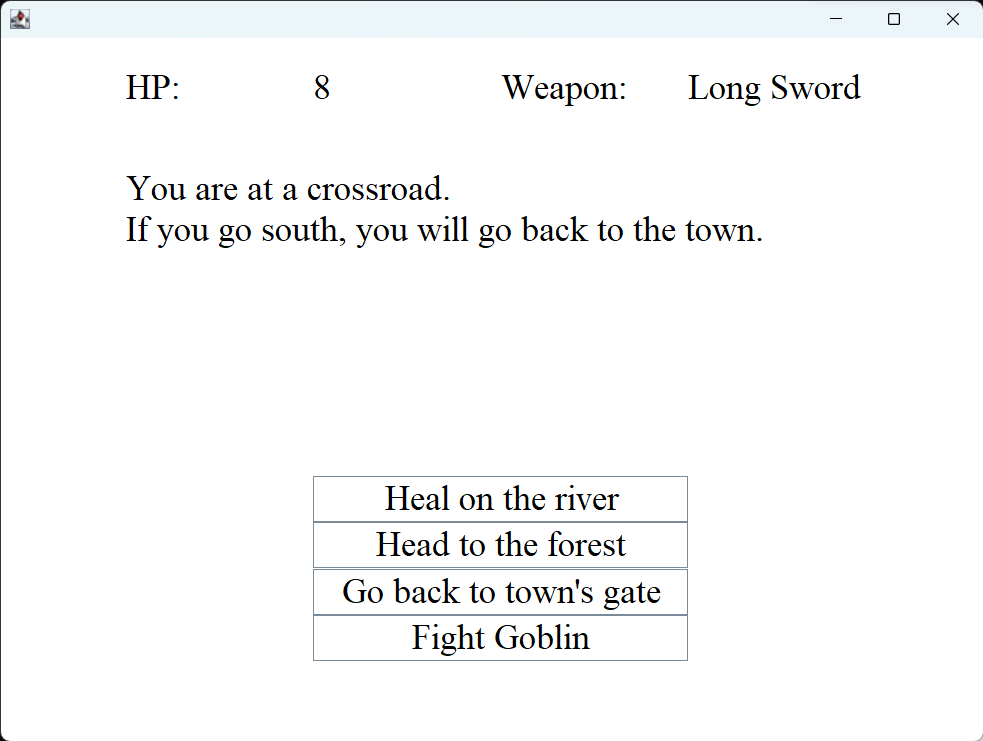
**Back to Town’s Gate**

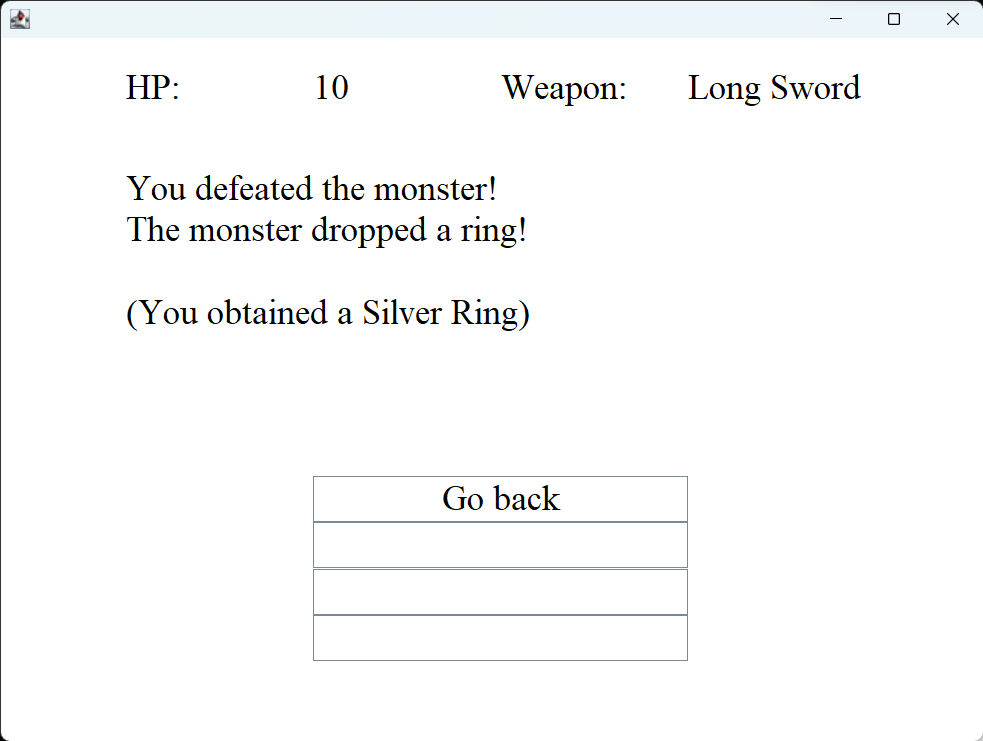
**Fight Goblin**

****

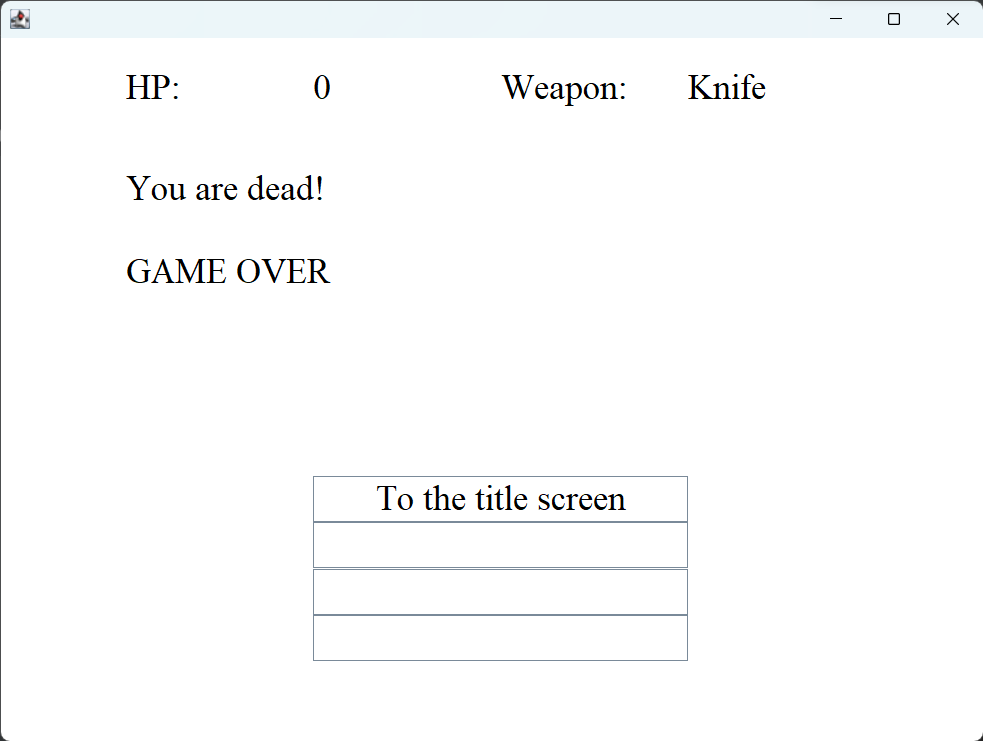
**Attack**

****

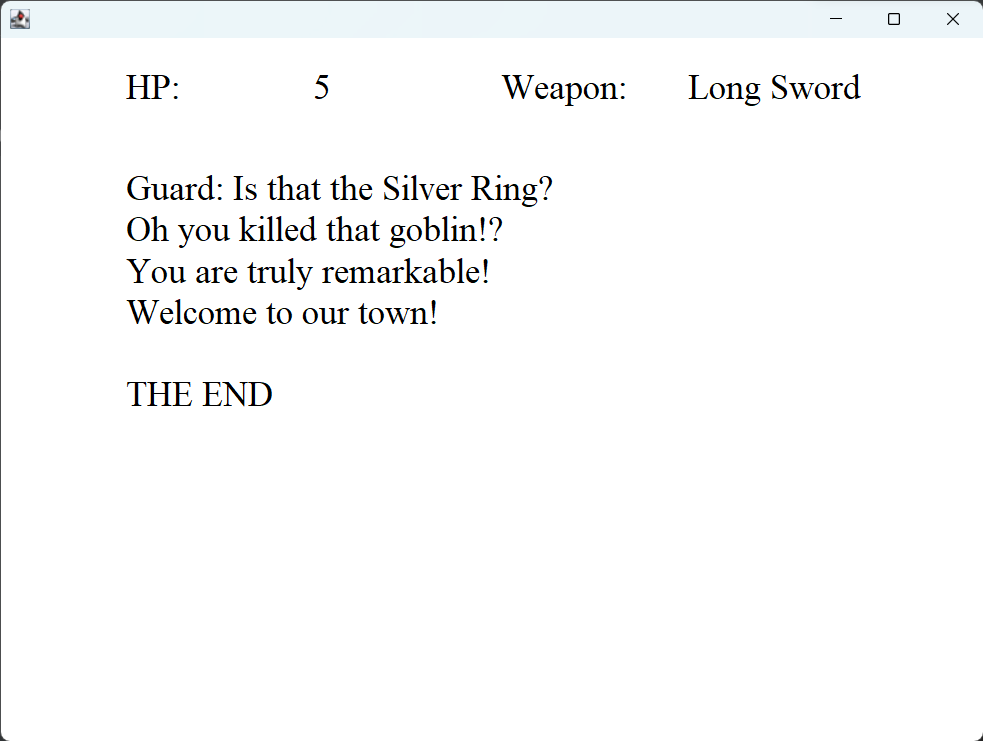
**Run**

**Win against monster**

**Dead**

****

**Win Scenario**

****