

README: Jumpy Man - Final (Version 1.0.0)

GITHUB LINK: <https://github.com/johnhutchinson99/classicplatformer>

This is a classic platformer game that is being created for a computer science course project.

In this game, your character is placed in a world map and must avoid enemies/obstacles in order to reach a final destination/win the level. There are several different types of enemies: “maces” which move up and down, “pound cakes” which move left and right, and “dirt cubes” which randomly pop up from underneath a platform. You only have one life, however, you can attack enemies and kill them before they kill you! There are coins which when collect, add to your score. You can also increase your score by winning/advancing through levels. There is also a timer and background music/sounds. Have fun!

N.B. Screen shots of the game at the end of this README.

Getting Started: Accessing the Demos

These are the instructions for accessing Demos 1, 2, 3 of our project, the text-based and GUI-based version of our game respectively.

Pre-requisites

To compile and run this project, you must have [Java SE Development Kit 8](#) installed.

Accessing Demo 1 - Compiling and Running: Using Eclipse

1. Download the repository ([Version 0.0.1](#)) as a ZIP file (on Github.com, choose “Clone or download” >> “Download as ZIP”) and unzip on your computer.
2. Open an Eclipse workspace.
3. Within Eclipse, go to “File” >> “Open Projects from File System”.
4. In resulting the pop-up, click “Directory” to choose the “Import Source” as the project folder you unzipped in Step 1. The name of the folder should be “classicplatformer”
5. Now that the project is imported into Eclipse, run the project by right-clicking “Gameplay.java” in the Eclipse Package Explorer and choosing “Run As” >> “Java Application”.

Accessing Demo 1 - Compiling and Running: Using Command Line

1. Download the repository ([Version 0.0.1](#)) as a ZIP file (on Github.com, choose “Clone or download” >> “Download as ZIP”) and unzip on your computer.
2. Open command line and set your present working directory to the src directory, which is found within the folder you unzipped in Step 1.
3. Run “javac *.java”
4. Run “java Gameplay”

Accessing Demo 2 - Compiling and Running: Using Eclipse

1. Follow Steps 1-4 of from “Accessing Demo 1 - Compiling and Running: Using Eclipse”, except download [Version 0.0.2](#) in Step 1.
2. Now that the project is imported into Eclipse, run the project by right-clicking “GUI.java” in the Eclipse Package Explorer and choosing “Run As” >> “Java Application”.

Accessing Demo 2 - Compiling and Running: Using Command Line

1. Follow Steps 1-3 of from “Accessing Demo 1 - Compiling and Running: Using Command Line”, except download [Version 0.0.2](#) in Step 1.
2. Run “java GUI”

Accessing Demo 3 - Compiling and Running: Using Eclipse

1. Follow Steps 1-4 of from “Accessing Demo 1 - Compiling and Running: Using Eclipse”, except download [Version 0.0.3](#) in Step 1.
2. Now that the project is imported into Eclipse, run the project by right-clicking “GameplayGUI.java” in the Eclipse Package Explorer and choosing “Run As” >> “Java Application”.

Accessing Demo 3 - Compiling and Running: Using Command Line

1. Follow Steps 1-3 of from “Accessing Demo 1 - Compiling and Running: Using Command Line”, except download [Version 0.0.3](#) in Step 1.
2. Run “java GameplayGUI”

Accessing Final Game - Compiling and Running: Using Eclipse

1. Follow Steps 1-4 of from “Accessing Demo 1 - Compiling and Running: Using Eclipse”, except download [Version 1.0.0](#) in Step 1.

2. Text-based: Now that the project is imported into Eclipse, run the project by navigating to the “textbased” package and right-clicking “Gameplay.java” in the Eclipse Package Explorer and choosing “Run As” >> “Java Application”.
3. GUI: Now that the project is imported into Eclipse, run the project by navigating to the “gui” package and right-clicking “GameplayGUI.java” in the Eclipse Package Explorer and choosing “Run As” >> “Java Application”.

Accessing Final Game - Compiling and Running: Using Command Line

1. Follow Steps 1-3 of from “Accessing Demo 1 - Compiling and Running: Using Command Line”, except download [Version 1.0.0](#) in Step 1.
2. Text-based: From the src directory, navigate to the textbased directory. Run “java Gameplay”
3. GUI: From the src directory, navigate to the gui directory. Run “java GameplayGUI”

Running Tests

Running JUnit Test in Eclipse

1. Follow Step 1 of “Accessing Final Game - Compiling and Running: Using Eclipse”
2. Ensure that the JUnit 4 library is included in the Java Build Path of the project.
3. In the Eclipse Package Explorer, go to the “backend” package and then right-click “PhysicsTest.java”. Hoover over “Run As”, then select “JUnit Test”.

Running JUnit Test in Command Line

1. Follow Step 1 of “Accessing Final Game - Compiling and Running: Using Command Line”
2. Download [junit-4.12.jar](#) and [hamcrest-core-1.3.jar](#) and place them in the src folder.
3. In the command line, change the working directory to the src folder, and then the “backend” folder, and then compile using “javac -cp .:junit-4.12.jar:hamcrest-core-1.3.jar *.java”
4. Run “java -cp .:junit-4.12.jar:hamcrest-core-1.3.jar org.junit.runner.JUnitCore PhysicsTest”

References

The following sources were used for graphics.

Background and platforms:

<https://opengameart.org/content/bevouliin-free-mountain-game-background-for-game-developers> (Accessed Aug 06/2019)

<https://opengameart.org/content/bevouliin-free-nature-game-background-for-gamedevs> (Accessed Aug 06/2019)

<https://bayat.itch.io/platform-game-assets?download> (Accessed Aug 10/2019)

<https://opengameart.org/content/wooden-sign-game-decoration> (Accessed Aug 06/2019)

Enemies:

<http://www.aigei.com/s?q=%E5%86%92%E9%99%A9%E5%B2%9B&type=game&term=gif> (Accessed Aug 06/2019)

<https://opengameart.org/content/chocolate-monster-sprite-sheets> (Accessed Aug 10/2019)

<https://bayat.itch.io/platform-game-assets?download> (Accessed Aug 10/2019)

Player:

<https://giphy.com/stickers/creepy-walking-s6hx6Pcg8o8j6> (Accessed Aug 06/2019)

Sounds:

Gun sound: <https://www.zapsplat.com/music/science-fiction-weapon-gun-blaster-gunshot-2/> (Accessed Aug 11/2019)

Background music: Boss Fight: <https://patrickdearteaga.com/chiptune-8-bit-retro/> (Accessed Aug 11/2019)

Physics equations were taken from Anna Harlick's Physics 233 Winter Course

Screenshots



Figure 1: A screen shot of the start menu of our game.



Figure 2: A screen shot of a portion of the first level of our game.



Figure 3: Another screen shot of our gameplay.

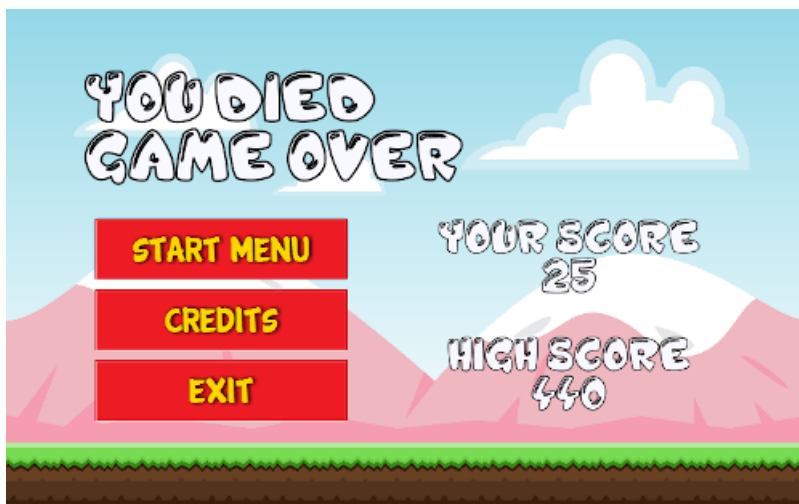


Figure 4: A screen shot of the game over menu of our game (i.e. the screen if you die instead of win the game).