The Judgement

Documentation

Game Created By: Travis R. Dewitt
Modified By: Jake Cyr, Brian Eisenberg, Diego Holguin, and Frederick Rodriguez
3 December 2015

Table of Contents

Introduction

The game *The Judgement* was created by Travis R. Dewitt on top of the game engine that he started building. Originally, the game had no objective, was full of bugs, and had no tutorial showing you how to play. Over the course of the four SCRUM cycles dedicated to fixing the reported bugs and implementing the recommended enhancements a lot of these missing features were implemented and most of the known bugs were fixed.

The current objective of *The Judgement* is survival. The player has to collect items from the chests scattered around the map and fight off the monsters to stay alive. The vision that our team has for the game is to see multiple missions added that the player has to complete. This would go well with an added storyline.

The rest of this document is going to discuss, how to install the game, how to play the game, how the game is designed, our vision for the game, and possible enhancements that could be made.

Installation

Required downloads:

1. Java JDK
2. Game files

Optional Downloads:

1. Eclipse or other Java IDE

Installation steps:

- If the game is going to be run without the use of a Java IDE:
    a. Open terminal(Mac/Linux) or Command Prompt(Windows)
    b. Navigate to the directory housing the game files
    c. Type “java Main”
    d. The game frame will be created and the game will appear

- If an IDE(Eclipse) is being used:
    a. Create a new empty Java Project and name it “Judgement”
    b. Click File → Import → File System → Browse (Above window panes)
    c. Navigate and select the folder that holds the game files and click Open
    d. Check the box in the left pane of the window that says “Judgement”
    e. Click the Browse button under the pane with the loaded folder
    f. Choose the Judgement Java project you just created
    g. Click Finish
    h. In the Judgement project, navigate to src/Main.java
    i. Double click Main to open the class
    j. Hit the Run button to play the game

Controls

Key	Action
Arrow Keys	Move Player
A	Attack
R(hold)	Run
F	Action (ex. open chest)
ENTER	Confirm Selection
BACKSPACE	Previous Screen
ESCAPE	Save and Exit Game
F11	Toggle Fullscreen

Game Design

“First of all, a game engine is software which is used to help in the creation of video games. It covers as much as possible in the background so the next game designer doesn’t have to. The goal is to streamline the creation process so it becomes easier and less time consuming. For example, your average program you may have created before this runs once and finishes. This game engine constructs a while loop which continually runs through a series of methods which do various actions like move objects or render objects. The program continually updates until a specific button is pressed which alters the while loop to end. A window object called a JFrame is also created in the engine, then a java class called Graphics2D is used to draw on to that new surface. The engine does way more than this, but this is the first goal of an engine. Loop -> window -> draw.

The other half of the classes provided, specifically the ones inside of the project folder, are project specific files. Which means they were created with the purpose of making a video game, every video game is different. They all need different variables and have different gameplay in some way. The goal is to put all of the game specific data in separated classes or in your game class and put all of the general “any game could need this” code into the engine classes” (Dewitt, The Design).

For more information on the rest of the design features of the game engine and game, refer to the document, *The Design*, by Travis Dewitt in the Judgement files folder.

Game Vision/Future Enhancements

Our vision for the game has changed over the course of the four completed SCRUM cycles. In the beginning, the team was planning on spending more time making the game look nicer. After completing the Customer Requirements report we found that customers preferred better functionality over design. Therefore, the SCRUM cycles were dedicated mostly to functionality of the game and improving the code.

Some of the bugs/enhancements were not able to be completed due to lack of time. The list of incomplete tasks that we would like to see fixed/implemented in a future version of the game is as follows:

- Player Walking in Place Animation Bug
- Inventory Flicker Bug
- Missing Diagonal Walking Animation Bug
- Sprite Shaking Bug
- Use of Magic as a Weapon and Tool (ex. light dark cave)
- Item Collected Indicator
- Entering houses ability