Shawn M. Joseph 5-10-17

CMPT 220

Dr. Pablo Rivas

Death Calculator: Final Write-Up

Abstract

This is the final write-up of my java project which is a text-based adventure game that is moved along the inputs that a player/user makes. The input by the user can also determine if the program would end, because most inputs will increment or decrement the number of a variable which the user should do their best to not increment past ten. I reflect on what allows the Death Calculator to run.

Introduction

My motivation stemmed from my limited knowledge when it comes to programming and the underlying concepts that are a part of programming. I wanted to take a stab at how programming could be used for the creation of games through functions such as having a player, having items, locations, and etc., that a user could interact with. The original concept was to have an application, but it was changed to a game to add more depth to what Java concept I could work with in regards to my code.

Detailed System Description

The two Java objects I worked with were DeathCalculatorAdv.java and Player.java The game runs in the main method of DeathCalculatorAdv, and the method uses the attributes and methods in the Player class to output a series of printed lines and input lines for the user to interact with.

The DeathCalculatorAdv calls five methods: chooseName(), chooseWeap(), choosePath(), getLocation(), and winGame(). These methods are placed, so that when one method is done the next method is activated; all of these methods are from the Player class. DeathCalculatorAdv issa Player. I put the relationship like this, so that DeathCalculator shares the attributes and relevant content of it's super class Player. The majority of method are about user input, and throwing and catching exceptions so that the user input matches what the system is asking for.

The user input is for the specific attribute choice determines the increments/decrements of the DeathCalc attribute which can end the program before the user is able to complete the text-adventure.

Requirements

I believe you can get a computer as great as Windows 98 and you will still be able to run this program. There are no intensive graphics, no strong engines, and etc., so the most you will need is a computer with a command line that can run simple programs.

Literature Survey

There are a lot of system that exist, and the bulk of them in modern day isn't even based on purely text and user's input alone. The majority of games, whether they are AAA or not, are honestly much more complicated than the program I created. My program's uniqueness comes from the concept of using a variable that would end the game whenever the user would push it's value to the predetermined value.

The other text adventure games generally rely on factors such as health points, items, and enemies, while my game is played based on how the user operates. It is suppose to push the user to make decisions that are positive, so that they could complete the game without the game automatically exiting them out. This is purely concept based, and could incorporate a lot of the other features within a program.

User Manual

The user is greeted with a series of messages for that will guide them along the text-based adventures. All the user has to do is type in the inputs that the system is going to ask for.

Proper Use & Error Prevention

- Inputs should be limited to letters
- Users should be able to read English language
- Don't replicated code in another language, for example Javascript

Conclusion

In the end, this final project is a means to the end of me improving my knowledge of programming concept in general; not only Java. The attempt I made to try and develop my own Game gave me a better understanding on how to handle my code, comment on specific ways that I implemented my code, and how I should stylized my code so that it is effective to read. I gained a lot of experience in Object-Oriented Programming (OOP) in both what I could accomplish and where I reach my limit. From this project, I hope to build better and more efficient programs in the future.

References/Bibliography

Liang, Daniel. Introduction to Java Programming. 10th Edition. Pearson. 2015, Armstrong Atlantic State University