

Project Specification - Major Practical

Introduction

Our project implements a simple version of a Dungeons and Dragons 5th edition character sheet. It allows the users to roll for their ability scores, assign them where they want, choose a playable race, choose their proficiencies and equipment, and at the end when they finish making the Fighter character, they are ran through a mini scenario were they can use their character abilities.

Design Description

Assessment concepts

Memory allocation from stack and the heap

- **Arrays:** The Rolled stats are made from a dynamic array
- **Strings:** The Proficiencies are store in a dynamically created array of strings
- **Objects:** Fighter class, Weapons, Armor, and other items

User Input and Output

- **I/O of different data types:** Command-line. Users enters several integers and strings for their character options, these affect the final outcome of the character.

Object-oriented programming and design

- **Inheritance:** The child Classes from the Player class, and the child classes from the Item class inherit from their parents
- **Polymorphism:** The Pure Virtual Class_Creation method from the Player Class is inherited to its children and is implemented in different ways
- **Abstract Classes:** The Player and Item classes are abstract classes, you can't create any instances of them.

Testing

- **Test Inputs / Expected Outputs:** The Stats roll function is run 10000 times to see if the values lie within its limits, they do.
- **Automated Testing:** The fighter class has copies of its methods that takes in parameters instead of user inputs, therefore it is automated.
- **Regression Testing:** The program was tested after every iteration, and fixed if it needed to be fixed.

Testing Plan

All the test files are run by a single makefile.

-Tested all abstract classes, and they produced an error as expected.

Start by testing the basic functions, the dice rolling, and stat rolling function.

-Dice rolling is checking if the random numbers are generated between the minimum and maximum. By running the makefile several times we found that the numbers lie between their expected values.

-Next is the stat roll. We checked this by doing 10,000 iterations and seeing if the maximum and minimum numbers are between the expected minimum and maximum numbers (i.e between [3-18]).

-Tested the weapon child class by creating instances of it. The objects are created fine and their respective methods work accurately.

- Tested Fighter Class and its methods by creating copys of methods that have parameters instead of user input

Schedule Plan

Week 8

- Write the basics of the classes, functions, and other things, and think of ways to properly implement everything,

Week 9-10

-Finish the child classes of the player class. (Once we finish one class, we can copy that framework onto the other classes.)

- Fighter
- Finish the basic variables
- finish the constructor
- finish the methods
- player creation method (ie. where the player chooses their options)

-Finish the Armor child class for the item class.

-Finish the main file.

Week 10-11

-Think about making a short text based adventure game based on the characer statistics that we worked on the previous weeks.

- If we have more time, add more player child classes.