

## **Introduction**

I used Netbeans to write a 2D turn based game in Java using the Swing GUI framework.

## **Techniques Used**

- parameter passing
- random number generation
- for loop
- while loop
- nested loops
- method returning a value
- 2D arrays
- programmer defined template classes
- encapsulation
- opening a file
- saving to a file
- sorting

## **Structure of the program**

The program is built of GUI, communication, data storage and manager classes. The manager classes take in a few ints describing a collection of data and methods and then either return the data or execute the methods. The communication class take care of interactions between GUIs like pausing GUIs.

## **Data Structures Used:**

I used an ArrayList which is a class in Java that allows you to make arrays that can be extended. I used this for listing all the attacks that are to happen in a turn because how many are being tracked can be anything between 0-72(or more with modding) and each is a small array of ints. The Graphical system for displaying player movement along with displaying of creature/weapon artwork is based on 2D char arrays. The storage of player-controlled creatures is taken care of by an array of objects of the DCS(Dynamic Creature State) class. The DCS class used has ~30 attributes which are mostly ints. Six of the attributes are IDs for the creature and it's attacks, the rest are modifiers and I also used arrays to store creature and weapon stats. For saving player progress a file writer is used where the player chooses the saving and loading location.

## **Main Unique Algorithm**

The fight GUI takes care of all the ingame combat. It does this by checking whether every fighting capable entity on the player's team has used an attack. When this condition is met the algorithm takes the attack arraylist of int arrays and sorts it into a 2D array that then gets run through by a for loop that uses the data in the array to run the attacks through the actionmanager class. This is done by the actionmanager finding the correct attack class and telling it to run it's dothing method. This method takes care of any stat changes of the creatures participating in the battle. At the end of all the attacks being completed the algorithm checks whether or not there are creatures that have less HPmod than they have health on either team. If the enemy team has no creatures where  $HP_{mod} < HP * (HPXP / 50,000)$  then the player wins if the player's team doesn't have any creatures that meet this condition the player loses. If neither team is completely wiped out the player picks the next attacks to use and the process starts again.\

## **GUI**

My game uses 2 main GUIs. The first is the "World" GUI which is the GUI the player sees when they start the game. The other is the fight GUI which pops up when the player enters combat. Both GUI were designed using Java Swing GUI components. The most common Swing components are Buttons and TextAreas(with disabled input) since they are used for controls and displaying all of the games visuals. Both menus use TabbedPanes for separating categories of the GUI. The Options menu also makes use of sliders and CheckBoxes. The GUI does not taken in any values given by the player which avoids all parsing errors that would otherwise have to be dealt with.

## **Software Tools Used**

To write this I used the IDE Netbeans by Oracle. It is focussed on Java development and has many useful code templates. It has 2 view modes a code view and a GUI design view this makes designing GUIs much simpler because you have a GUI for designing your GUI. The result is that you do not have to test how the GUI looks every time you add a new element to it. Netbeans also has support for adding shortcuts for simple code and your own code templates which makes repetitive tasks much faster to than writing them again every time. Netbeans was the best choice main due the GUI designer because it allowed me to test how the GUI looks and works way fewer times than if I was just writing the GUI myself where I would have to compile the whole project everytime I change the layout of the GUI.