

Status Summary

Title of Project: Choose-Your-Own-Adventure Zombie World

Names: Tri Bui, Gianni Valentine

Work Done:

Stat Class: The Stat Class has been completed with getters and setters for each character Stat. Using this class we can update the character's in-game statistics and call upon them to see if certain activities are able to be executed.

characterGUI / Application Class: The application class containing the user interface has been created and initialized. Using Python's Tkinter we created a basic GUI where the user will be able to load a new game, save a game, load a saved game, make decisions and view stats.

class storyInterface: This class has been set up. This class is where both the game's story and decisions are held, along with the Decision parser and Decision tree saver to be exported to the Save Class. When a Decision has been selected from one of the three options a,b,c, this class is called and passed the value, it will then return the follow-up Story and attached options for the user as a string. The code structure is essentially complete here, but the story text is currently just a placeholder.

Changes or Issues Encountered

A few things have changed. 1.) We had planned on allowing users to select from 4 options given at each step of the story we have now shortened this to 3 options. We wanted there to be about 18 steps to the ending of the story depending on the user's actions. This would mean we would have to write 4^{18} different story options and decisions. So by cutting the options down to only 3 will make it more manageable on the story writing side. 2.) We are going to have an early ending to the story depending on the user's stats and actions. This will also decrease the number of stories that need to be written. 3.) We have removed the stat: Weapon and ammo to simplify the number of stats a user must keep track of. 4.) The unique identifier system will now be using a,b,c,aa,ab,ac etc instead of 1a,1b,2a,2b,2c etc. This way we will be able to parse this string in order to know exactly where any user is in the story. Currently, this is being stored in a list in the storyInterface Class but will be moved to our Save Class in the next interaction.

Patterns:

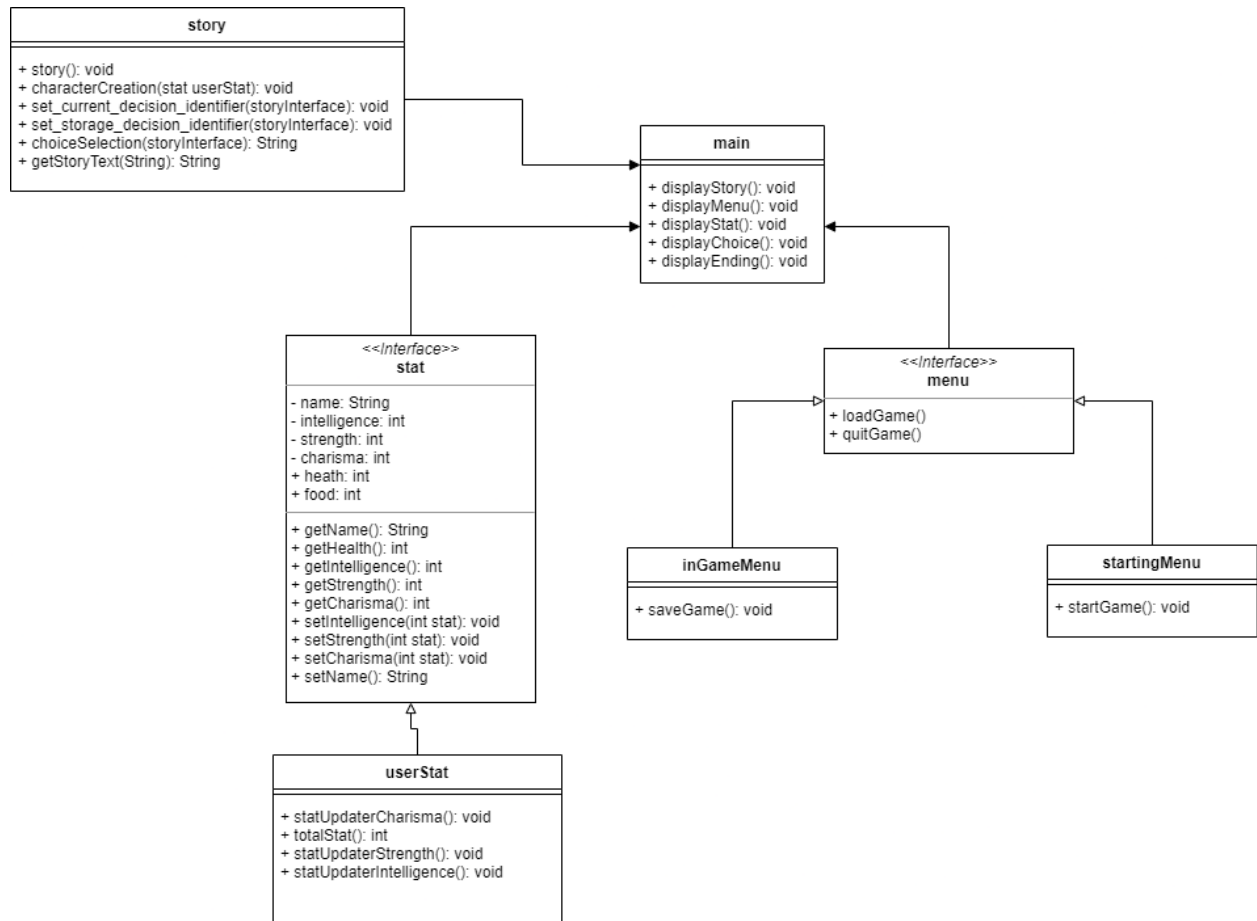
We plan for our game to have three possible endings. One of which is dying, with sub-ending of each possible way to die. Each of these endings will be used at different levels and story states of the game, in all scenarios the game is over but there are different ways to get there. Thus we thought that using the strategy pattern would be the best option. We plan to create an ending() interface with Ending1, Ending2, and Ending3 implementing it. Using a context class to determine which method should be overridden.

Plan for Next Iteration

The current plans for the next iteration are to start and complete the Save/Export Class along with the Character Manager. We will also be wrapping up the GUI for it to work with the save class. Also, the story will have been written.

- Save/Export Class
- Character Manager
- Main
- Finish GUI
- Write the full story with all endings
- Create the 3 ending class and interface

Class Diagram



Link To Code: <https://github.com/gianni321/CSCI4448>