Vanessa Reyes
CSCI 2270
Data Structures

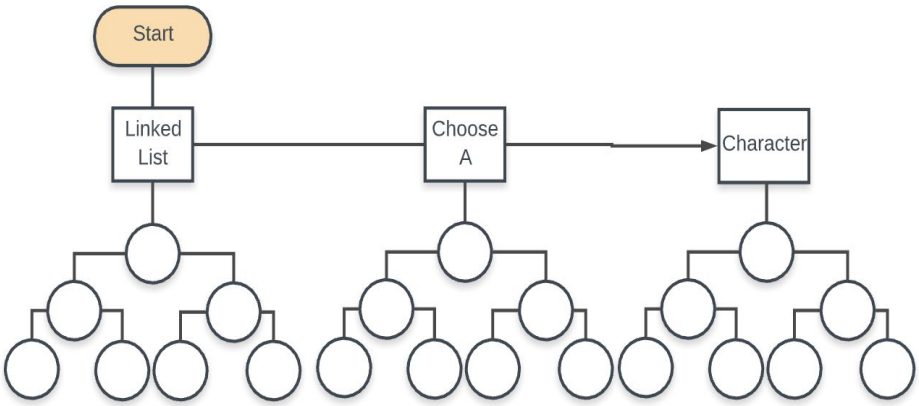
Create Your Own Adventure Zombie Game

Data Structures

The data structures used in the project are a linked list and a binary tree. The linked list is used to represent the different characters within the game. Each linked list points to the root/parent of a binary tree.

The linked list holds all the data for each character, such as their name, character number, strengths, weaknesses, profession, and weapon. The binary tree holds the character's storyline. At each node, there is a message that is a description of the level of the story and presents options to the user to traverse down the tree. Based on the user option, the tree will traverse left or right.

Methodology



When the user starts the game, they begin at the root of the linked list. The player can then traverse through the linked list where each node represents a different character. When a user chooses a certain character, the character node contains a pointer to the root of that character's binary tree. The game continues to the root of that character tree and displays the message at that node. The message contains the story and choice options. When the message is done displaying, the user then inputs their choice in the story and that choice correlates to either the left or right child of that node. When traversing down the tree, some nodes end the game or win the game.

Data

The data that has been used in this project are different text files that contain information for each character and their story. The text files are parsed through and entered into the necessary data structures.

The character information is parsed into the linked lists. Then each character's story is then parsed and entered into each individual character's binary tree.

Throughout the game, the program keeps track of the player's choices with a queue and can be used to save their game for future use as well as checking if they create a new high score. When a player saves the game, the player's choices that are saved in the queue are put into a text file to be parsed later if the user chooses to continue from a saved game. For a high score, if a player wins, the high score text file

will be read. The program will compare the new score and the current high score for that current character story to see if the player has beat the high score. If the player beats the high score, the player will enter in their name and the data will be updated and outputted back into the high scores text file.

Results

- Lose
    - When the player is traversing the binary tree, playing the game, they can reach a node that will end the game. This is presented through a symbol that is read when parsed.
    - When the game is ended, the player will be asked if they want to play again and they can choose if they want to or not
        - If the player chooses to play again, then they will have the option to choose another character and start their story from the beginning
        - If the player chooses to not play again, then the program will end
- Win
    - When the player is traversing the binary tree, playing the game, they can reach a node that will win the game. This presented through a symbol that is read when parsed.
    - When the game is ended through a win, the player will be asked if they want to play again and they can choose if they want to or not
        - If the player chooses to play again, then they will have the option to choose another character and start their story from the beginning
        - If the player chooses to not play again, then the program will end
- New High Score
    - If the player wins the game, the program will take the player's score and read the text file that holds the current high scores. If the player beats the high score, then it will update the score and output it into the text file.
- Save Game
    - If the player chooses to save the game during their story, they can press 0 when asked for their choice. This will then take the player's choices that are in a queue and output them to a saved game text file nicely to be parsed later.
    - When the user begins a new game, if they saved the game last time, there will be an option for the user to continue from their saved game. The saved game text file will be parsed and read. The parsed data will be inserted into a text file and the program will start the game from the player's saved point.
    - The saved game only lasts for one game, so if a person saves a game and begins a new game, they only have that new started game to start from their saved game. The next game they begin, their saved game info is not available.