Game Engine Report

SUNNY SARKAR | JOSEPH DENG | MITCHELL PHELPS

SUMMARY:

Our project could be summarized as a role-playing game engine. Our project allows a user to create a story with up to three choices per event, which creates a choice-driven text-based adventure. The user is also able to create multiple characters and enemies, and edit their characteristics, like strength, speed, and health. After creating the story, the game incorporates a combat system that allows the players of the game to interact with friendly characters and enemies.

DATA STRUCTURES:

~Graphs:

- Graphs are used in our project as the combat system that allows characters to attack an enemy.
- Graphs were chosen because graphs allowed for each character to be connected to each other character, allowing the characters to interact with each other in a fluid manner.

~Linked Lists:

- Linked Lists are used in our project to easily store and access the characters the creator creates.
- Linked Lists were chosen because of its easy access and storage.

~Binary Trees:

- Binary Trees are used in our project to store and access the story that the user creates, offering up to three different paths with each node created.
- Binary Trees were chosen because binary trees followed the structure of a role-playing game, in which a game contains a story with multiple choices that lead to multiple endings. Binary Trees were also easy to create as well as traverse.

METHODOLOGY:

~StoryTree:

- The story tree consists of nodes that contain the event description and the description of the three choices that the player can choose (Figure 2). Each child of the root node has the same structure, and so on and so forth.
- The user creates the game by creating the parent node, then going to one of the child nodes and creating its description. The user is then given the choice of going to the next level. If the user chooses not to continue, the user gets to fill up the description of the two other nodes on the same level. This is repeated recursively until the user ends back at the parent (Figure 1).

~CharacterList:

- Character List is simply a Linked List that contains both enemy character nodes and the player's character nodes. The characters are differentiated from friend or foe by using a boolean that declares true if enemy and false if friendly. The character nodes also contain character attributes like speed, strength, and health, as well as the name of the character (Figure 3).

~CombatGraph:

- Combat Graph is a graph that has directed edges to enemies. For example, each enemy will have an edge directed toward each non-enemy, and every non-enemy has an edge directed towards each enemy. This allows for an attack action to be performed to any non-team member.
- Arrays were also used to store turn orders derived by the speed stat on the characterList.

RESULTS:

Due to the massive number of possible results, a specific output is not pre-determined. Since this is a game engine, the outputs are entirely due to the user's imagination.

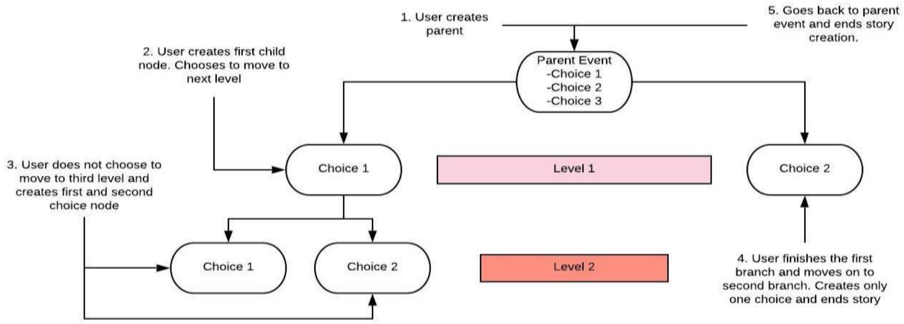
1. User creates parent
5. Goes back to parent event and ends story creation.
2. User creates first child node. Chooses to move to next level
Parent Event
-Choice 1
-Choice 2
-Choice 3
3. User does not choose to move to third level and creates first and second choice node
Choice 1
Level 1
Choice 2
Choice 1
Choice 2
Level 2
4. User finishes the first branch and moves on to second branch. Creates only one choice and ends story

Figure 1

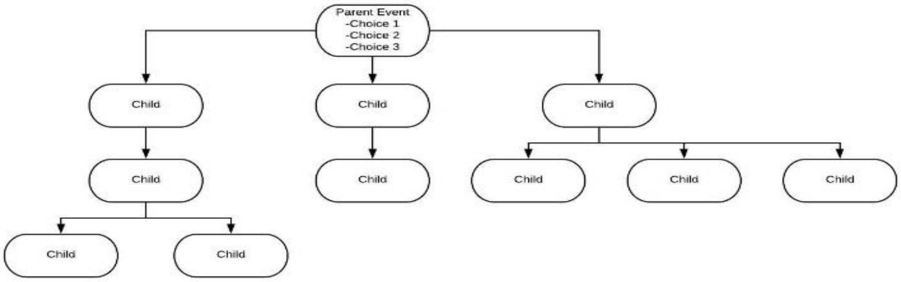

Figure 2

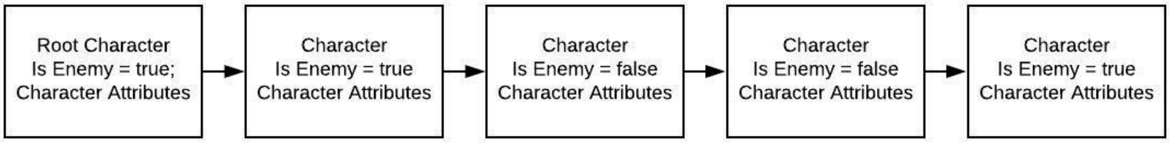

Figure 3