Caleb Landis

Finite State Game Documentation

CSCI 43700

How to Play the Game

<https://cs.iupui.edu/~callandi/csci437/finiteStateGame/>

Clicking the link above will take the user to the introduction page of my game. The game is called *Intro to AI Training,* and the users progress through the story and gameplay exclusively by clicking links to different passages.

The Technology Behind the Game

This game was made using Twine’s passage creation tool, which also had the help of the Harlowe story format. Twine was great for creating the initial structure and story text of the game, and Harlowe allowed me to add functionality straight into the passages that I had created with Twine. The Twine/Harlowe combination is a great toolset for creating a finite state game because it allows the creators to clearly define the current state the user is in while simultaneously making it easy to display and change to the next part of the story that the player wants to go to.

Playing *Intro to AI Training*

The story starts off on a relatively linear path as the player is introduced to the story of the game. The player learns that they are taking a new class offered by IUPUI called *Intro to AI Training,* which allows students to directly work with enhanced animals that have been given special abilities through new biochip technology. The player is given a choice of one of three new creatures developed by IUPUI, each with a specific trait that defines how the user will train the creature throughout the game. Once the player has chosen a partner, they are to train, study, and learn how their partners evolve through the practice of using special moves. The player can also battle their rival, who is someone that is an even match for the player, or they can take on a greater challenge by battling the top student in the class. When the player has decided that they are ready to take their final examination of the course, they learn that they will have to battle a very strong leader in order to receive their final grade.

Development Reflection

Creating the story of my finite state game turned out to be just as fun as playing the final product. I was definitely proud of myself playing through the final build as I realized how many mechanics that I was able to introduce into the game. The game offers a surprising amount of different decisions the user can make throughout the story, and it even has decent replayablity since it allows the users to restart the game and choose a different starter creature. Depending on which partner the player chooses, the player can increase the stats of their partner deeming how they see fit. The game even has opponent level scaling which allows for your opposing students to have their stats increased as you train your own. During combat, the player must strategically choose when to temporarily boost their stats for that battle as they pay close attention to which moves their opponent are making. The player must be mindful of when to use their potions to heal their partner since they are only given potions upon leveling up. The “final boss” battle also adds plenty excitement to the game. The boss’s stats are set to be insanely strong to ensure a challenge even for players who have spent time leveling up and training their partner. The initial shock of how much damage the boss can do is very entertaining, especially to players who just went straight into the final boss battle. I also believe that the ending is a very satisfying conclusion to a fun playthrough of the game.

While I enjoyed creating the story of the game and coming up with all of these ideas that I wanted to implement, actually coding in these features proved to be a much greater challenge than being creative with the story and gameplay. Writing the passage text and building the structure of the story using Twine was very easy, but adding the functionality into each passage was much more frustrating. The relative semantics and syntax of Harlowe was not too difficult to learn. The real challenge was staying on top of formatting the code in a way that made it clear where each logic block started and ended. The Twine passage editor made it extremely difficult to edit the Harlowe code because it does a terrible job of formatting and spacing code. It didn’t help that the editor was very sensitive to missing parentheses even though keeping track of all of the parentheses became exponentially more difficult as more logic was added to a passage. Most of my time throughout development was spent reformatting and respacing how the code was displayed in the editor. It was also extremely annoying having to go back through each passage at the end of development to see where each of the unnecessary new lines were added to the actual page that the player sees.

Towards the end of creating this game, I was starting to understand the nuances of editing each passage, which allowed me to speedily clean up any bugs and visual issues. Creating the overall logic and states of the game was not super difficult as I expected, but the code editing definitely took up an ungodly amount of time. I definitely bit off more than I could chew with this project, but most of that was due to me not knowing how long I’d spend debugging format issues. I had two other very large projects due the same day as this one, which made it even more frustrating how long I had to spend formatting the structure of each passage. Although I am turning in this game late, I am still very proud of the finished product. It was fun to be creative, and it was just as fun to play through the product of my creativity. I believe that I even properly demonstrated the use of state through a choose-your-own-adventure internet game.