Green Dinosaurs and Ham

My project is a choose your own adventure game with mechanics based on the Dungeons and Dragons Standard Reference Document (SRD) published under the Open Gaming License (OGL). During the module, players encounter an anthropomorphic dinosaur who communicates by producing scents based on its emotions. The players can defeat the pirate antagonists without the help of the dinosaur, but their chances are slim. The key to success is to learn how to communicate with the dinosaur through its unorthodox emotional language.

I used Twine to implement the project and hosted it on Tesla. Within Twine, I built over 50 states and use more than 60 variables. During the first few story states, we initialize all the pirate enemy variables and character variables based on which weapon the player chooses. Following the SRD, most of the randomness is achieved by rolling a twenty-sided die and adding a numeric modifier. Some rolls are measured a static target such as an armor class, and others are contested rolls meaning the player’s roll is compared to the enemy’s roll.

The win and lose state of the game are based on a final fight against a group of pirates. The player will always get help from the incredibly strong dinosaur teammate who is capable of defeating pirates with a single blow. On the dinosaur’s turn, it emits a scent. If players react by telling the dinosaur something appropriate to its emotions, the dinosaur does massive damage. If players react with a mediocre response, the dinosaur does modest damage, and an inappropriate response causes the dinosaur to miss. For example, if the dinosaur is confused, you could guide it for good results. However, if the dinosaur is happy and you try to console it, the dinosaur would miss.

The enemy randomization used to simulate artificial intelligence uses a custom system not borrowed from the SRD. A six-sided die is rolled with different actions assigned to different results. If an enemy is more included to attack with a longsword, that action is assigned to more results of the die; for example, results one through four could all result in that favorable action.

One major challenge I encountered was that links change color if you have already visited their destination. Thus, if you choose the correct emotional response to the dinosaur and that it results in a critical hit, you can then see that link as visible next round. I randomized the emotional scents and responses each round, but the link visited color ruins that. To solve this issue, I secretly have the players visit the dinosaur mood result states at the start of the game. I hide the real text until the fight starts to the player sees those states as the author, about, and disclaimer screens. This may not be the most elegant solution, but it works perfectly. Another challenge I faced was that Harlowe code, written in a near and readable way, added too many spaces to the web view of the states. I had to write all my code in a readable format, then remove all line breaks to ensure that my transitions did not appear halfway down the screen or with large gaps between them. The resulting code is not something I would ever want to troubleshoot without first expanding it back out again. An example of this issue can be found in the included image “Harlowe\_Code\_Organization\_Problem.”

There are several areas I could extend this project in the future. There are variables for the player and enemies to use special abilities that are not implemented. I could add abilities for combat maneuvers, spells, and sneak attacking. There are also some non-randomized elements that could be extended with SRD mechanics. For example, the enemies are not actually harder to hit when they are dodging, though the player is led to believe they are which is effective enough. Likewise, the dinosaur has infinite health and set damage amounts that could be controlled with variables and virtual dice. I enjoyed and learned a lot from this project and would be interested in improving and extending it in the future.