What I Implemented This Week

If you pull the master branch at https://github.com/hareeshganesan/Vooga, you will get the work I did for this week.

This week, my work focused on adding a better AI system that implements strategies, allowing for time loops of Actions using the ActionSeries object, and implementing some of the State machine necessary to help the game designer develop a story mode.

The details of the design for the AI strategy can be found in the design document. For the state pattern, I set up a GameState class to tie together all of the different menu screens and leveling screens. Using the GameState class, a transition state method in the GameObjects would set the next state for the main game engine to whatever the next state specified in that GameObject is. An example of the GameStates can be done by using the Q key in the CombatInstance to go back to the Title screen.

For the AI system, per my discussions with Tanner, I implemented an AI that could be used to generate new behavior based on selecting ActionSeries from different strategies. To do this, I had to create the FollowAction, AvoidAction, ActionSeries classes and everything additional in the ai package. The AI also works with the collision system now. I reimplemented the functionality using a calculateLocation method in order to make sure collision checking was done before FighterSprites were completely moved.

To test my functionality

Go into the createAIStrategy method in LevelObjectFactory and modify the weights as you see fit, taking care that at least one of the weights is 1. I recommend setting .5, .5 and then 0, 1 to see significantly different behavior.

Also, press the Q key in game to go back to the home screen.