CS 4900
Week 8 Writeup
Group 1: Merritt Hancock, Kenda Blair, Ryan Trull, Alan Bettis

Alan: I built a system to load models in, replacing the placeholder cubes that had long represented our player and enemies. Currently all enemies use the Milcap model, and the cursor uses the player model, but now that Kenda has finished most of the current models, we can replace the individual enemy models and have a specific cursor model by next week. I also noticed some issues with our system, as there were a number of cyclical dependencies between our files. I refactored the codebase to fix this issue by extracting a widely shared import and putting it into its own file to be globally accessed. I also fixed a few bugs in the Pathing and finalized our Action Point (AP) system.

Merritt: I gave us the ability to switch scenes and current levels. This was done by assigning the current level and current scene to variables housed in our controller class. Additionally, I implemented independent level creation to give us better memory performance. Previously each level in the game was added to memory when the game was launched, but now each level is created upon selection from the menu through a loading manager. I also designed and built a menu system and a loading screen giving us a better experience for the user. Making these changes required a good bit of refactoring which was largely a result of the scene switching.

Kenda: I modeled, rigged, and animated the verm enemy model and created animations for prior existing models (Slime, Milcap, and cursor). Currently, the Pinpod enemy model is incomplete and the Verm enemy model requires a texture. I will complete those tasks as well as work on implementing animation in ThreeJS this week.

Ryan: I created the attack state for the Milcap Soldier AI. Now, when the player is within its attack range (default 1 tile adjacent), it will cause 0.5 mass in damage. When moving out of the enemy’s attack range, its state will change back to pursuit again. I also created two new enemy types based on the concept enemies we presented in a prior week. These two, Verm and Pinpod, have their own state machines with differing AI. The Verm has a cowardly AI. It starts out scavenging around the map, and when it sees the player, it will retreat to its nest, which is just a specified position on the map, and hide for a few turns. The Pinpod has a simpler AI. It doesn’t move since it is a cocoon type enemy. Every few turns it will extend harmful spikes from its body that will damage anything next to it. A few turns after that, it will retract those spikes and give the player an opportunity to get close to it. The last thing I did was implement enemy absorption. When the player is within range of an enemy with lower or equal mass, the player can despawn the enemy and steal the enemy’s mass to add to its own.

Next Steps: For next week, we want to...

- Implement animations for our player and enemies, as well as replace the rest of the enemy models and cursor model.

- Implement spike ability and ability absorption. Additionally, we would like to look into tooltips to display relevant information for the user.
- Implement tooltips for UI
- Finish animations for Verm and Pinpod models
- Begin researching how to build our Level Editor