CS 4900
Week 11 Writeup
Group 1: Merritt Hancock, Kenda Blair, Ryan Trull, Alan Bettis
Details

Changelog:

1) Worked on the lose screen.
    - Merritt 04/17/20
2) Populated Level 1 with enemies
    - Ryan 04/17/20
3) Created terrain for Level 2
    - Ryan 04/17/20
4) Populated Level 2 with enemies
    - Ryan 04/17/20
5) Added Level 2 to main menu
    - Ryan 04/17/20
6) Continued tooltip implementation and added game directions on the menu.
    - Merritt 04/19/20
7) Worked on css for the game.
    - Merritt 04/19/20
8) Worked on tooltips.
    - Merritt 04/19/20
9) Fixed getUIData() bug.
    - Merritt 04/19/20
10) Worked on lose screen design.
    - Merritt 04/20/20
11) Finished lose screen and win screen
    - Merritt 04/20/20
12) Worked on right tool tip
    - Merritt 04/21/20
13) Animated Pinbeast and adjusted weight painting
    - Kenda 04/18/20 4:30 PM- 6:30 PM, 04/19/20 3:30 PM- 7:15 PM, 04/21/20 2 PM- 5:45 PM
14) Fine tuning animations on enemy models, and made absorb animation for slime
    - Kenda 6 PM - 7:15 PM
15) Started making alternate slime model for when pinpod ability is absorbed.
    - Kenda 7:30 PM - 7:45 PM
16) Fixed double movement bug for enemies
    - Alan 04/19/20
17) Fixed bug where player could be moved without selection
    - Alan 04/19/20
18) Added utility function to check if enemy is adjacent to player

    - Alan 04/20/20
19) Fixed overlay such that it will display if nearby enemies are absorbable or not
    - Alan 04/20/20
20) Refactored Movement/selection
    - Alan 04/20/20
21) Shrunk Level 1 and rebalanced for “winnability”
    - Ryan 04/21/20
22) Shrunk Level 2 and reworked enemy pathing
    - Ryan 04/21/20

Decisions

- Cursor now moves a bit differently, lots of things were moved a bit differently

What we learned

- Tween.js is a good library for position interpolation

Summary

Alan: I spent this week working on mostly bug fixes and refactoring of the cursor. At the start of the week, a lot of things were broken, and now they are (mostly) fixed. Enemies usually move correctly now, and the cursor/selection/movement system has a much more sane layout under the hood, using a finite state machine to switch functionality every time it selects/deselects/moves things. Unfortunately in testing during the end-of-cycle merge, we noticed more issues, which will be of prime importance to fix going into this week. Regarding animation implementation, it has proven more difficult than expected, as I must synchronize movement of the model with playing of the animation. Tween.js seems to be a good library to interpolate positions between two points, which I intend to use to move further.

Merritt: This week I focused mainly on the tooltip implementation and the win and loss screens. For the loss screen, I added two separate buttons. The replay button restarts the level for the user, and the menu button redirects the player to the menu screen. The win screen is similar to the loss screen except that it only has a menu button. Additionally, I added gameplay instructions on the menu screen. For the tool tips, I continued the two separate div design I mentioned last week and wrote a method to update the text as the cursor is being moved. The left tip focuses on the player/slime information and the right tip changes based on what tile the cursor is currently over. If it is an enemy it displays relevant information about the enemy for the user, and if the space has no occupant then it just displays information about the empty tile.

Kenda: I animated the pinbeast model and fine tuned animation for Verm’s alert animation. I also created a new animation for slime.

Ryan: This week, for me, was all about getting our first two levels built and integrated into the game menu. At the end of last week, I built the terrain for level 1, but had no enemies to engage with. Added three enemies (one guarding the exit) which must be absorbed to get to the exit. Then, I created the terrain for level 2 and populated it with enemies. This level requires a certain strategy to beat it, wherein the player has to absorb every enemy (other than the one guarding the exit) and a pinpod (to gain spike ability) in order to lower the exit guard's health enough to absorb it. After adding both levels to the game menu, we decided both were too large. To fix this, I removed/reshaped both levels' terrains to make them more compact while preserving the means to win the level.

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

- Finish implementing animations and rotations of entities
- Fix movement breaking bugs, make enemies selectable again.
- Add some kind of error handling for out of bounds cursor/pathing errors.
- Design a death screen
- Begin looking into and implementing sound.
- Finish polishing up the win screen and the tooltips.
- Finish the Boss Level (Level 3) by:
    - Implementing randomized pinpod spawns
    - Creating a Charge attack state for pinbeast
    - Implementing a three-hits kill system for the pinbeast where it can only take damage when in its spawning state