CS 4900
Week 9 Writeup
Group 1: Merritt Hancock, Kenda Blair, Ryan Trull, Alan Bettis
Details

Changelog:

1) Moved game inside of html canvas.
    - Merritt 04/03/20
2) Gave player a state machine which contains its abilities
    - Ryan 04/03/20
3) Implemented the Spike ability for the player
    - Ryan 04/03/20
4) Created div over game body.
    - Merritt 04/04/20
5) Added win level method in controller.
    - Merritt 04/04/20
6) Created a new level which will contain the pinbeast boss
    - Ryan 04/05/20
7) Added the level to the Level 3 button on game menu
    - Ryan 04/05/20
8) Fixed Verm absorption exploits
    - Ryan 04/05/20
9) Fixed Pinpod absorption exploits
    - Ryan 04/05/20
10) Continued working on the win screen.
    - Merritt 04/05/20
11) Created a menu button and wrote it's event handler.
    - Merritt 04/05/20
12) Created ability to win a level and change level after.
    - Merritt 04/05/20
13) Fine tuned animations for Milcap and Verm started Verm UV map
    - Kenda 04/04/20 4PM - 6PM
14) Finished Verm UV, started pinpod remodel, converted completed models to gltf
    - Kenda 04/05/20 12 PM - 3:30 PM, 5:30 PM - 10:15 PM
15) Remodeled pinpod, uv mapped pinpod, started rigging pinpod
    - Kenda 04/06/20 3:15 PM - 6:45 PM
16) Finished rigging pinpod, animated pinpod, converted pinpod
    - Kenda 04/07/20 12:30 PM - 4 PM
17) Implemented rotations for enemies so that they face the direction they move
    - Alan 04/05/20
18) Created Pinbeast enemy type (Boss enemy)
    - Ryan 04/07/20

19) Created an outline of AI states for Pinbeast
    - Ryan 04/07/20
20) Implemented removal of objects from scene when returning to level select
    - Alan 04/07/20
21) Added resource tracking from https://threejsfundamentals.org/threejs/lessons/threejs-cleanup.html that can properly free up memory.
    - Alan 04/07/20
22) Moved texture loading from tile construction to board construction, resulting in massive performance boost.
    - Alan 04/07/20
23) Researched necessary functionality for level editor
    - Alan 04/05/20
24) Fixed a bug that allowed buttons to be clickable even when invisible in div
    - Merritt 04/07/20

Decisions

- Use upload and download of .json files for level editor
- Decouple Controller and LevelManager classes, so that LevelManager initializes the level objects and Controller handles rendering and scene management.

What we learned

- A good bit of css styling commands and how to use javascript to change css properties.
- The .dispose() function must be called to free up memory from assets in Three.js, removing them from the scene graph is not enough.
- Our texture memory leak was the result of loading the textures for each individual tile on our board, resulting in tremendous memory consumption.

Summary

Alan: My tasks for this week primarily involved assisting with resource management and animation implementation. After some difficulty implementing parts of this system myself, I found a tutorial (linked above and in the relevant code) on tracking resources, and followed it mostly to the letter for removing assets from the scene and freeing the associated memory. I also discovered the source of that “camera” memory leak, and it was that the textures were being loaded for every individual tile, rather than loaded once and applied to each tile. In the process of fixing this, I managed to break the textures for the level, so pushing past that is a major priority for next week.

Merritt: My tasks this week involved continuing the building out the menu system and adding different screens within the gameplay. I began by refactoring our html file and moving the game

within a canvas. I then researched different ways to implement the win and loss screen and settled on using an html div. Next, I created a div that, using css, managed to get over the game canvas. I then added text, a button, and css to style the screen. To get the screen to function I wrote an event handler for the button that made the menu visible, utilized the loading screen, and deleted all of the contents of the scene from the previous level. For the deletion of the previous level, I had a good bit of help from Alan in debugging the process. I also recently had to do some tweaking to the divs and such to fix different bugs such as turning off pointer controls.

Kenda: My tasks this week included fine tuning pre-existing models and creating the pinpod. The pinpod enemy is completely modeled, textured, rigged, weight painted, and animated.

Ryan: My main goal for the week was to implement player abilities now that our player can absorb enemies. Our pinpod enemy contains a spike ability which, when absorbed will pass on to the player. The player can then press ‘space’ and use the spike ability which will damage enemies directly adjacent. This, paired with AP, adds an element of strategy to the game where the player can take steps to avoid damage while dealing out their own. It also allows a player to lower an enemy’s mass in order to absorb it or remove it from the level. With that implemented, I moved on to the final enemy of our first zone which is the Pinbeast. The Pinbeast is a grown up Pinpod that we are using as our first boss. First, I started out by creating a new level which is a boss room mapped to Level 3 in our game menu. Then I created the Pinbeast enemy type and made an outline of its states. Pinbeast is currently not completed and not added to the new level.

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

- Implement animations in the enemies
- Develop the Pinbeast boss AI
- Begin work on the level editor
- Fix the “technicolor bug” that happens as a result of textures not loading before the board is generated.
- Begin implementing tooltips
- Have a complete Zone 1 for our game that consists of three levels