

## Details

We began by laying out the initial framework for the game, starting from Dr. Mihail's example file. We wanted to get something quick to show, so we left parts of the code base a bit messy. We set up a grid system to define a game board, keyboard-based input, the beginnings of a camera system, and player movement.

### Changelog:

1. Began working on importing 3D models. Found two loaders online and added them into a newly created asset folder. One was from a graphics class code example. Other was from master file of three.js. My thinking behind them not working is an error on my part of not setting them up correctly. Also found a 3D model for slime and added it to a folder for models. Not sure if this model has the correct file extension.
  - a. ->Merritt, 1/23/20
2. Created a flexible grid making system which populates the terrain. Can be used to make different shaped boards. Created three basic terrain types as well. One is a normal space represented by grass. Another is an untraversable space represented by brown color.
  - a. ->Ryan, 1/24/20
3. Created new level and textured assets for better visual representation.
  - a. ->Ryan, 1/26/20
4. Created initial framework for player movement. Created class to house the methods
  - a. ->Merritt, 1/24/20
5. Implemented methods for player movement and moved all player functionality to separate file.
  - a. ->Merritt, 1/25/20
6. Moved keyboard button press checks to separate file titled "keyboard\_input.js" and implemented the boolean values as a dictionary.
  - a. ->Alan, 1/25/20
7. Fixed a lag spike issue by removing a redundant render function call from main.js
  - a. ->Alan, 1/25/20
8. Began implementing movement though many bugs are still present.
  - a. ->Merritt, 1/26/20
9. Started camera movement system started with horizontal and vertical rotation  
->Kenda 1/24/20
10. Fiddled with code to try and reduce lag. Tried to implement zoom using camera.fov. Did not work. Implemented camera reset.  
->Kenda 1/25/20

11. Rekeyed camera controls to match design specs, removed vertical rotation and moved horizontal rotation to q and e, made arrow keys free camera move, added zoom on o and p  
    ->Kenda 1/26/20
12. Added locking mechanism to keydown/keyup events related to movement, ensuring that there will only be one move per key press.
  - a. -> Alan 1/26/20
13. Fixed bugs related to movement/camera system. Right movement was based off of start position instead of current position, and backward movement was not mapped to any keys.
  - a. -> Alan 1/26/20

## **Decisions**

We decided to make our game a turn-based puzzle game where the player is a slime whose goal is to absorb enemies, grow, and complete the level. Each player action is a turn which will then prompt a turn from the enemies. Absorbing certain enemies gives the player abilities which can be used to continue solving the puzzle.

The game board, we decided, was going to be a grid with the goal of having multiple potential terrain heights. So far, our game board is created using a grid script which contains createGrid methods to generate different levels. This is performed using 2D Arrays which contain ID values of the different terrain types. By reading the array, it is possible to generate 1x1 tiles in succession on the 3D playing space.

We used a javascript dictionary object to store key code values, which are then referenced and used by the keydown and keyup event listeners.

We decided to implement a cursor to look at specific tiles/enemies/the player, which can also be used to confirm and select actions. When fully implemented, this will replace the current movement system of direct WASD movement.

## **What We Learned**

Event Listeners drive most of the action in Javascript, but they need to be managed by semaphores or some other locking mechanism to ensure order amongst the function calls.

Duplicate render calls make the render loop lag due to rendering multiple times.