Mat:

I turned to terrain generation. After researching some ways to do so, I found the diamond square algorithm which seems to be what we're (I am) looking for in terms of Layer 0 of our terrain. After finding some examples to follow, I began coding it up. I ran into issues with populating the heightmap array. So my next chunk of research went into arrays in javascript and how best to handle them. This research resembled me periodically picking up my laptop and smashing my face into it for 30-45 minutes then putting it down and doing more googling from my phone. Eventually, I learned about the array.map() function which I used to initialize a square 2d array of 0s. Finally, I was able to begin implementing the algorithm. I decided to try it recursively, but that is presenting some problems with arrays larger than 5x5. This week, I will finish debugging the algorithm and apply the heightmap to our terrain.

Carson:

This week, I started to work on our flood fill implementation. Since we are working on a grid-based game, we figured something like this would be useful for us. Most of the week was trial and error in getting our implementation just right. In the end, we decided to fill our entire grid with invisible highlights. Then, once the character spawns in, we use flood fill to find all the highlights in the characters range and set their visibility to true. We also plan to use the same approach to set the highlights back to no visibility. As an added extra, I added a temporary skybox to our game. This is currently subject to change, but for now, we are using the same tool that group 5 used.This next week, I plan to finish implementing flood fill, refactor my code to be up to date with everyone else, and assist the others in any way I can.

Emily:

This week, my goal was to create a linked list that stores the models. At first, I was struggling because I would only be able to see the elements of the list within the block of code that it is created in. When I tried to access it outside of the block, everything was undefined. To solve this, I got some help from a peer and realized that the models were not completely loaded yet. Because of this, I was able to place everything that needed to access the list in a *manager.onLoad()* call so that it happens after everything is done loading. The linked list is now able to be used correctly, however I still need to work on some minor issues regarding switching to the next list element.