Mat:

I began the week fixing the Diamond-Square algorithm. I accomplished this by changing the diamond step to only check for values within the same subarray. Next I started applying the heightmap to a PlaneBufferGeometry. This caused some strange behavior. The sections of the geometry affected by the height map were no longer visible. After troubleshooting, I discovered it was due to the way I was attempting to apply the values. I needed to convert the 2-dimensional height map array to a 1d array. After that, (because the position array of the Plane Buffer Geometry was stored as a 1d array) I applied the values of the heightmap to every third value of the position array. This set the z value for each vertex. Finally, the terrain was appearing as desired. Next, I implemented a method to round the height values to numbers ending in quarter numbers (0, 0.25, 0.5 and 0.75). This gave the map the polygonal appearance we wanted. Next, I intend to continue focusing on the map, leaving the task of refactoring for spring break.

Emily:

This week, I focused on changing the linked list to an array. We decided that this would be a simplified way to do what we need to do with the models. I created two arrays: character array and enemy array. I was able to attach the models within these arrays to an instance of the Actor class. To do so, the name was checked which determined the subclass to be used. Once the instance of the subclass was made, I created an *actor* attribute that is attached to the object, essentially linking the created Actor object to its model. Next, using this character array, I was able to cycle through characters by pressing ‘r’. I did so by using a variable named *characterCount* that acts like the index of the array, changing it as needed when ‘r’ is pressed. I also edited the colors of the enemy models using blender. Additionally, I knew that we needed to start on some kind of HUD and was thinking of a way to signify that the user wants to end their turn. To accomplish this, I added a “End Turn” button in HTML that signals the enemy characters to begin their movements and, eventually, their attacks. As of right now, when this button is clicked, an enemy model will move five spaces forward. I also added an “Attack” button that does not currently have a use, but will eventually cue the selected model to attack a selected target as long as it is in range.