Emily:

The first thing I did this week was look up what Dr. Mihail mentioned regarding how we may need to calculate a character's range. I was able to find the .normalize() function within the Vector2 class that might help with this. Next, I wanted to make my highlights more neat and easier to access individually, so I added them all to an array. As I was working on this, I realized that if our map size gets altered in any way, we would need to manually change numerous values in order for our movement to work correctly. I took the first steps to prevent this by creating constants that represent the map edges so our characters do not go outside of those boundaries. Lastly, I used the *visible* attribute on the highlights mesh to make the appropriate highlight(s) disappear when the banana is on the edge of the map.

Mat:

I began the week by examining the graphics code we've used thus far and began working on ways to generalize model loading. The first iteration consisted of passing strings to a function that would apply the string as the model type in the source string. Then, I found some new models to use that turned out to probably be way bigger than we should use (because they are intended for 3D printing).

Next, I found that the models loading asynchronously gave me issues when hard coding some test lines. So I started researching the LoadingManager. Lastly, I implemented the manager in my model factory. The loader loads all the models properly, but calls the onLoad function too early for some reason, so my next task is to troubleshoot the loading manager.

Carson:

For this week, I did not do much programming. Instead, I decided to focus more on researching a way to implement procedural generation, which was more difficult than I first thought. In the end, I discovered a technique called Plot and Parcel which is used in one of my personal favorite video games. Since I discovered this, I started to work on a static tutorial level, which is still being worked on, and I have some ideas for a static map for our procedural generation. As a side note, I realized I was focusing too hard on finding what other people had done using Three.js and not what games of the same genre have done. I then reevaluated what I could do better and that is: look more into the games that inspired us to make this type of game and see what we could potentially use from them.