Math Analysis

Jesse Ancira

**01\_Pattern**, when first starting out we started in the grid, really easy and simple. Made a pattern only took a few minutes to complete, but it was a nice introduction to the class I would say. **02\_Pixel\_Art**, next up we had to make a more complex figure in the grid. Still fairly easy maybe had to put a little more thought into it. The difficulty of this project really just depended on what you chose to make, but still nothing drastic. I made a Creeper from Minecraft and since the creeper itself is made out of pixels it was a walk in the park. The Grid was easy and simple but was time consuming if you had a large picture/pattern. It would also be a headache if you didn’t think out the size of the object/pattern you were making before hand.

**03\_Savinginbricklayerlite**, now this is where things take a turn and now were working in Bricklayer Lite to make a picture/pattern with a set code. Took more time than the grid but it was something new**. 04\_Objectsinbricklayerlite**, Still working with the same set of code and still easy just took more time but what made it easy was the short pattern**. 06\_Name\_Project,** Now beginning to have a more larger picture which required more code. We had to code our name into Bricklayer Lite. You had a choice to either make this in Level 1, Level 2 or Level 3. I started off in Level 1 because I had no knowledge of the other two levels. The time it took to actually do this in Level 1 was not worth it. I could have been working for days on this and wouldn’t get anywhere. So I was being told by friends to try Level 3. Level 3 was something I am so thankful for when it comes to Bricklayer Lite. Level 3 saved lives I’m sure, with the ability to have just one put2D and a function could mean your whole 64x64 background. Id say level 3 reduced time/effort be 99.9 percent! (not really) From than working on level 1. I put in the whole code because I thought it was all necessary, above each of the codes I put in what that piece of code was.

**07\_Flagproject,** on this specific code it was really the same thing I did in the last project with the name. I made the code for each flag then just offset it to the location I wanted it to go. This just code was just really repetitive in my opinion. Didn’t take any critical thinking and the flags are not accurate but you can tell which ones which. **11\_AllFunctions,** now this project was some fun. This was actually the first time I had ever used the line function. For the life of me I couldn’t not figure out how it worked. I eventually figured it out and this was also the first time I had to use any negatives in my code. I made this code as simple as I could because it needed to be continued by someone else. All my other codes were unorganized and all over the place because they were for my understanding only. You can see a number pattern in the x and y axis in the offsets. By understanding that part of the code you can continue it with no problem.

**12\_Real\_Image,** In this project I really didn’t do much. I used the ring function for this one and just offset it to where I needed the rings to be. The code is extremely short and very simple. I was running out of time on this one and I needed something quick. I picked the Audi logo because it was something I knew wouldn’t take a lot of code and I knew everyone around me would recognize it and know what it was. **13\_SpaceFillingCurves,** Thiscode actually tookme a while to complete. I used the some methods I did in the previous projects. Coded the object and just offset it to where I needed it to be. The thing I learned while doing this is that the more code I have in one function the less I have to offset it. Then when the whole object was finished instead of having a bunch of separate functions I took the code out of each function and put it into one huge one. Then I would offset that to the desired location which required less offsets. I would say I have a okay understanding of bricklayer lite and how it works but I know there is still a lot for me to learn about it. I