Charles Henderson
Daniel Chen
Jason Hsu

Interactive Flythrough of the Universe

The original assignment as we got it was supposed to make a 3d flythrough of the universe utilizing a 3d TV to fully immerse the user into an environment, using game systems or a computer hooked up to the TV along with game controllers to control flight through a myriad of galaxies in the sky. Some of the expected design challenges that Prabhat included figuring out a control scheme that would allow movement to be intuitive and easy for users, figuring out a way to potentially implement scene graph in order to let planets and objects orbit each other, figuring out how to deal with what limited screen space using different, wider field of views, and attempting to turn galaxies into generated planets and stars when the camera gets close enough to them.

Starting off, we made some variations to the project description in order to take advantage of our group members individual skills and backgrounds in graphics and mobile app development. For example, instead of exclusively supporting game controllers, one idea that popped up during our plan was to make the program able to grab mobile phones with an app installed and make it so that they were able to just walk up, install, and then use their phones as controllers on the spot. Halfway through the quarter, however, one of our group members dropped the class which made us rethink some of the more complicated ideas that we had thrown into our design that would be potentially too much work with a less member. As such, we scaled back our ideas before the second quarter came along to something that was a lot more doable, leaving out some of our more wackier ideas like adding in death stars or alien ships and random explosions from stars going supernova or other additions to the project that would make it a more interesting experience for people of all ages.

Starting the actual coding mostly involved reading in the data that we received from Prabhat, a whopping 1.7million points, and then mapping the 255 pictures that were sent along with it to the points, matching each picture such that it matches up with the galaxy type that is specified for each point. To start off with, a few thousand points were loaded in and small spheres using nVidia's glow effect tutorial were rendered in space before we moved on to actually figuring out billboarding for the images such that they would be facing the camera from any location. This was done using the SDL libraries on OpenGL and most if not all of our code ended up being in C.

Loading in all of the points, however, proved to be problematic as memory would not be large enough in most cases. As such, it was decided that a data structure called an octree would be required in order to partition the individual points such that it would be easier on the memory requirements in terms of loading in points and would make the program run more efficiently. Essentially, what an octree does is divide a cubic space into 8 parts repeatedly, subdividing each result cube multiple times until a desired level is found. Then, when the camera is in a certain quadrant, all the points from that quadrant can be loaded into memory in order for the program to dynamically allocate it for only the galaxies that are close enough to the user for them to need to be rendered. This took care of the memory issues that would be cause by trying to load in all 1.7 million points as well as mapping the images to each point thousands of times.

Eventually, Jason received a Leap Motion from a kickstarter. Seeing the capabilities on this new device made it look rather enticing to use. The device itself is rather small and uses a USB connector to hook up to the computer. How it works is it has infrared sensors inside of the device that are aimed upwards and essentially detect and map where the users fingers are and where they're pointed into a 3d space using a vector. Using the SDK to turn this new technology into a controller still turned out to be a sort of a challenge. In the end, the control scheme that was used was to have two hands involved in the control. The first hand was used to rotate the

camera while the other hand was used to move back and forth as a sort of throttle controller. While not entirely intuitive, once users tried it out, we found them understanding it pretty easily. Unfortunately, however, at the actual senior design showcase, we found that bright lights actually messed with the motion detection and given the bright lights at the pavilion, it was quite difficult at times to use it as a control scheme, forcing us to switch to a keyboard and mouse setup at one of the stations.

There were a few ideas that came up, but were unfortunately unable to have been finished by the time we got around to it. One of the ideas we wanted to do was to take information from some database and attribute it to a specific galaxy such that, when a specific galaxy is pointed to, any known information on that galaxy would be displayed. Or, if we had gotten the galaxies to turn into points, we could do the same for individual planets as well. Another idea was to get the velocity of the user and perhaps some sort of minimap to display on the screen to show the user where they are in the galaxy. Unfortunately, all of these problems were present due to the fact that the True Type font libraries that were built using SDL were not cooperative or easy to use and would've been much more trouble to implement. There was also a lack of data on the individual galaxies and, as such, it was difficult if not impossible to try to display information that we didn't have.

The final iteration of the project that was shown at the showcase was somewhat similar to the final version that we showed during the last senior design class meetup. However, due to technical difficulties at the actual showcase with the bright lights messing with the Leap Motions, it didn't go quite as smoothly as we had hoped.

However, in the end, this class was a good learning experience and a good experience in trying to grow some self discipline in working on projects without any sort of teacher or professor to guide us. It taught us how to manage our time and schedule meetups to work on coding or research or contacting related parties outside of actual class.