

Brag Sheet:

The biggest challenge I faced while coding this was getting the textures to work. I messed up the xcode set up part and was stuck with gray screens for days. After fixing my setup, textures worked perfectly. I was also glad to find the SPHERE_MAP capabilities so I did not have to do this by hand :).

The second biggest challenge I faced was the particle system. I found it difficult to give my particles proper physics without using the exact code from the example, as I could not change the laws of physics, and saw that code as the best way to solve the problem of particles. In the end, I took out the parts of the particle2.cpp code that I decided not to use, and decided changing the size of the collision box would be a good route to take to get some points for my work.

I also decided that I could spawn some bugs in different spots based off of the i value in the for loop while rendering my particles, bringing my code away from the example code. I also used bees as my particles, using a translation to place them instead of placing a GL_POINT at a certain coordinate.

An accomplishment I achieved was getting the light to follow around the flying bugs at the right level. Additionally, the light did not look great on a large square ground. To fix this, I added a function to draw a platform of many cubes below where the light spawns for the bee, and the butterfly. For the fly, the cube platforms remain statically in the code.

Another accomplishment was my use of clipping planes in this project. In previous projects, I could not get this to work at all. This time, I was able to clip off a part of the log in the water.

Another accomplishment was getting selection with color picking and pixel location to work in the left and right window respectively, while also running animations in both windows. I came up with a solution to draw a bug with or without a hidden buffer based on which window it was in. Then, when I was selecting in one window, I was able to keep the normal colors of the bugs in the other windows without the hidden buffer showing through.

Another accomplishment and challenge was drawing a bezier curve flower. The example code helped me find the desired points, and I was able to include the curve in my code and translate/rotate it as needed.

Another accomplishment was the transparent butterfly wings used in the left screen. This was achieved by examining multiple example files and slides, and I was able to achieve translucent wings on my first try.

My final accomplishment was designing a bezier surface to look like a rock. This brought my project together well in the end, so the dragonfly had something to fly over. I looked at bezierSurface.cpp and figured out how to manipulate the control points to my liking.