Joshua Sims

Pyramid Pairs

I developed a game of pairs – the user selects matching pyramids from several rows of pyramids. In this game, there are two pyramids per match. If two matching pyramids have been selected, they will be removed from the scene. The objective is to remove all pyramids from the scene.

Before interacting with the pyramids, the user must click on the large, colorful box. After clicking that box, the box will turn black and the camera will shift into the box such that it is facing the pyramids. The user can then play the game. To select a pyramid, the user must click it. As many as two pyramids can be selected at the same time. Once the pyramids are selected, they will rotate to reveal their identifying colors. If two matching pyramids are selected, they are removed from the scene after a very brief delay – that delay is intentional. With the delay, the user can better witness which pyramids are being removed from the scene before they are removed. Aside from clicking pyramids, the user can press the “c” button to rotate and reveal all pyramids at once. This will not “select” all pyramids, however – the user still has to click pairs of pyramids to remove them. Instead, this feature allows the user to quickly discover the colors on the bottoms of all the pyramids so that they may finish the game at a faster pace.

The most difficult part of this project was getting the pyramids to behave correctly according to the user’s interactions. There were a substantial number of edge cases to consider when programming the behavior of the pyramids. In addition to the gameplay, the placement of the box in the scene and the camera was also extraordinarily difficult. After many minor changes and frequent glances between code and scene, I positioned the box and camera where I wanted them.

The clipping range turned out to be far closer than I anticipated and this made the placement of the large box quite troublesome. If the box was too far away, it would be partly clipped, but if it were too close, it would block the user’s vision on startup. What surprised me most is how comfortable I became with THREE.JS while developing this project. The 3D coordinates and the behavior of the THREE.JS library became more intuitive to me. I find that THREE.JS is still less convenient than Unity, however, sine Unity has a graphical editor which allows for a more intuitive manipulation of the scene.

Regarding the visual aesthetics of my project, I was pleased with everything that I developed. I like the flipping of the pyramids, the entry into the large box, and the shading and colors on the pyramids. I would have preferred to have included more visual effects, but I at least finished the basic effects.

Given more time with this project, I would include a visual effect where the pyramids separate vertically into segments just before they are removed from the scene. If I did that, I would then program the segments to rotate slowly and then more rapidly just as the pyramid is about to disappear. Extending even further upon the visual aesthetics of the scene, I might include some larger pyramids, statues, and desert dunes in the background. Alternatively, I could create wacky, psychedelic objects and put those in the background. For better gameplay, I could implement a scoreboard – perhaps multiple for multiple players. I could also increase the number of pyramids which would make the game more challenging and suitable for multiple players. Given an entire semester with this project, I would like to make it a distributed application such that users could play the game together over the internet.