Explore 3D

Donald Clark | clarkds@umich.edu
Saurav Lohani | slohani@umich.edu

Assignment 3 Post-Mortem

Things that went well

1. Once we implemented generic map creation with text files using a topology map and a terrain identifier map, we were able to quickly create interesting maps without needing to hardcode game objects. We also devised an algorithm to automatically check whether large objects made up of many terrain primitives can be hollowed out (e.g. pyramids, mountains). This not only allowed us to minimize terrain creation/rendering, but it also allowed us to come up with interesting objects, such as buildings.
2. Utilizing some of the code from Donald's 2D Game (Explore) was crucial because it helped us work on other aspects of the game while still maintaining cool features from the previous game.
3. Implementing a generic way to model terrain and items made adding new ones extremely easy. The goal was to make it appear as though the game had many different elements, but in essence the only change in code for some of these primitives were 1-2 lines.
4. Utilizing Git as our version control system made collaborating remotely super easy. We often branched from master for various tasks/features, and we helped code review each other's work when big changes were being merged.

Things that did not go well

1. Starting late on the project after our initial game proposals were rejected caused us to pull many all-nighters to catch up. We were bummed out that we couldn't implement every feature we had hoped to because of time. It also didn't help that both of us used up all of our late days on the 2D game.
2. Although our generic map generation enabled us to mass-produce maps, the fact that there were two maps (topology and terrain id) made it super hard to make the text files themselves. This was because there was a need to parallelize locations of certain things as coordinates. For example, to make a small pyramid, 15-20 points had to be coordinated between the two maps, which were tedious. Still, we were willing to do this since we did not want to hardcode maps.
3. Generating many terrain primitives lagged the gameplay somewhat. We decided to go down the path of making maps look interesting at the cost of little performance. However, there were times when the gameplay lag was so bad that we had to go back and fine-tune some of the algorithms to win back performance. At the end, things worked out well, but the amount of time we had to spend to go back and refactor code for performance upgrades could have been used towards another feature addition.