GD2P02 – Physics Programming

Summative – 3D Physics Project: Cloth Simulation

Post-Mortem

General

The Cloth Physics summative was attempted and completed by a group consisting of two members: John Bu and myself. We decided to group up for this project due to the large scope of the task set out for us, and considering it was a completely new coding challenge as well as using a completely alien middleware: Bullet Physics.

We decided to use our existing frameworks that we had developed for our previous OpenGL courses and narrowed it down to John's framework.

We went forward with the project by dividing up responsibilities. John would handle integration of the Bullet Physics middleware, the cloth physics and the mesh handling, while I would handle half of the additional features as well as the wind forces in the level.

Bullet Physics

We decided to use Bullet Physics as it already had a large amount of tools and implementation of basic as well as advanced physics concepts was easier and faster than modifying our existing frameworks to have those physics capabilities. Since we were using John's code framework for the project, the task of integrating the Bullet Physics middleware into the framework went to John. He was more experienced with his framework and it would be easier for me to refactor and manipulate code rather than messing around with the foundation of the framework itself.

Modularity of cloth

One of the requirements of the assignment was to allow the user to modify values that would affect the shape and size of the cloth, along with changing the amount of wind force applied by the fan object in the level. The first step was to ensure that the values were modifiable in the source code. This meant that I had to refactor a few integral functions in various classes to basically accept few more parameters and pass them on to linked objects if required. Once these dependencies were set up, it was time to implement a sort of user interface for the player to discern which values were being modified as well as set up input. However, during the last stretch of the assignment, it became clear that any kind of mouse-picking or screen interaction

with these values would come in-between the already existing object pick-up mechanic. Hence we decided to read these modifiable values from a settings.ini file, which can be used by the player to change the values and see the result of said change the next time they run the application.

Wind and Fan

The first task regarding wind forces in this project was to set up and implement a 'global' wind which would react with the cloth over the entire simulation. This wind force had to be random and gentle enough as to not tear or flip the cloth itself. By applying random impulse forces over a set amount of time to each point of the cloth grid, I was able to implement a natural looking wind effect which further increased the 'life-like' aspect of the cloth itself.

This same concept was somewhat carried over to the moveable fan object. The forces were applied in the same way where the affected areas of the cloth would receive impulse forces on the respective grid points. However, since the fan object is movable, the cloth parts had to be checked every frame regarding whether they are intersecting the stream of force given out by the fan object. If any cloth part, torn or un-torn, intersects the wind force area of the fan, then it pushed in the negative-z direction (into the screen).

Resetting of simulation

To tackle this aspect of the task, we decided to allow the player to press the 'R' key on the keyboard to reset the entire demo by deleting the current level and generating a new one to avoid any memory leaks. However, it became apparent that deleting the entire level was causing major exceptions and errors that broke the build of the game. Along with the instability this reset brought, the hierarchy of the project was disrupted whenever a level was deleted, along with general OpenGL framework parts not functioning correctly. We then decided to avoid deleting the level and instead reset just the cloth parts as well as bring the objects back to their original positions. The cloth parts had to have their member variables reset along with have themselves rearranged in the grid format while setting their positions to the origin.

Collisions with other objects

Bullet Physics made it easy to implement the Sphere and Capsule shapes as they were pre-built collision shapes within the middleware. The cube was also used to test the collisions with the cloth, however the brief specified a pyramid object to be present as well. The pyramid turned out to be a challenge as Bullet Physics had no pre-built methods to build a pyramid and render

it as well as to construct it by using other collision shapes. In the end, we decided to go with the cube, sphere and capsule objects to interact with the cloth.