

Comp220 Graphics & Simulation

Demo Concept

The concept for the demo I have created is to show what techniques I have learned to use in OpenGL as well as the Bullet Physics Engine in a basic form to meet the requirements to satisfy the brief. My scene is very basic and features two meshes with two separate textures as well as one of the meshes being rendered in five times. The tank mesh I have has bullet physics implemented on it so when the scene is rendered the tank has gravity applied and falls to the floor then it can be controlled via the arrow keys on the keyboard for interactivity. I have post processing which allows me to change the effects of how the screen is displayed. I didn't have time to implement the game idea I wanted to show off which is why it is quite empty. I have a raycast in it too which when clicked over objects with physics (only one in is the tank) it will tell you you've hit something, otherwise it will say nothing hit.

Intended Outcome

I had an idea in mind to make an OpenGL game in which the user would play as a mesh and be forced to work their way through a level with obstacles in the way, the invert gravity feature would be used when the level needed to be flipped upside down to pass certain obstacles. The raycast feature I have would be used to shoot your gun at enemies and when clicked on they would be destroyed. Now I have raycast and physics working I may continue this on to make this game possible. Due to only getting them working last minute this was not possible.

Scope

I don't think the scope of my project was too big I do however feel that I left it far too long to be able to implement my idea as I got stuck getting physics to work with my gameobject class and I stupidly missed a few lines of codes in my camera class which made it so my raycast wouldn't work. Next time my goal is to get everything working as soon as possible so I can get the best grade possible and to make a fun intractable demo/game.

Bullet Physics

Bullet physics was added in class from Brian which he used their gravity simulation to simulate gravity. I took what he had and added it to a

gameobject which was my tank. Bullet also allows the user a sense of interactivity by being able to move the meshes by applying different forces to it. The bullet engine is supposed to be used to see how objects would act in real life which is why it is such a good physics engine to use.

Raycast

The raycast uses bullet physics and tells you whether or not you hit an object which uses bullet physics. It is not visible in the actual window, it is only available and visible on the separate window. I planned on having bullet implemented on more objects than one and it was going to do different things to different objects like destroy them when clicked for a deeper sense of interactivity. The raycast works by clicking the left mouse and currently it only prints into the other window and tells you if you hit the tank or not. If I had gotten this working earlier I would have been able to do more with it.

Post Processing

The post processing effects I implemented are very simple. They change the way things are displayed on the screen, I currently have four effects to choose from. I currently have black and white, greyscale, red and blue or you can chose to have none of them and have it normal. They don't do much for interactivity however you can switch them on a key press to choose how you want it to look. These effects are created by manipulating the pixels like we did last year for tinkering graphics.

Requirements

My demo satisfies the requirements of the contract by featuring three skills mentioned on the brief. It has bullet physics which simulates real life physics in demo and allows the user to partially interact with the demo by being able to move the tank with the arrow keys which gives the user something to interact with. The other effect I chose to add was a raycast, this is another thing mentioned on the brief, and it is in demo but only currently prints to output window. The last effect I have is post processing, this changes the way the output looks by making it black and white, greyscale, red and blue or you can chose to have none of them and have it normal.