SOFT351
Jordan Paynter
10471173

Wall Collision Information

Users point of view:

- Using the 2010 visual studio solution.
- Motion of blocks through a wall, to interact with each other.
- Controls:
    - WASD to move forwards, left, backwards, and right.
    - Page up to speed up and Page down to slow down the camera movements.
    - Q and E to flip which way the camera faces.
    - Z to start, and X to stop the cubes from firing and reset them.

Programmers point of view:

- Keyboard interface to check keys,
- Appropriate loops to reduce code and re use when creating cubes, and processing calculations for them.
- Methods broken down where appropriate to increase re use.
- Memory released where appropriate.
- Code for rendering text but commented out due to incompatibility with DX11.
- Code for movement is there but needs tweaking for full effect.
- Movements are determined each frame calculated from collisions with other cubes, based on there angles of collisions and speed.

Engineering issues:

- Product is focused more on good practice than performance, ensure products behave correctly.
- Had large problems getting DirectX to work mostly due to Windows 10, and linker errors.

What I started with:

- Tutorial05 - Matrices fx shader with with no technique from the DirectX11 adapted tutorials from Nigel.

My own work includes:

- Efficiency improvements to code, breaking down, and compacting.
- Ability to move cubes during rendering.
- Keyboard action from direct input.
- Calculations between the blocks to determine next movements.

Brief Pitch

The software demonstrates objects bumping into others, interacting in a normal like fashion. Forces are calculated through trigonometry and the force of the original cube. All collisions are treated as elastic collisions, in where no energy is lost. Using the keyboard controls provided you can view the wall from different angles and also reset the simulation when you need to.

SOFT351
Jordan Paynter
10471173

Evaluation

Although I feel that my overall goal was not met I think that much of the code is there, it simply needs tweaking to get perfect results. I have focused on the code it self, including the more complicated physics behind the collisions and when they should happen. I have also tried to allow as much reuse when possible and reduce bulk chunks of code. Also whenever possible the code has been made scalable, for example, to add more cubes you would only need to adjust the cube count and add some coordinates.

Difficulties I found while making this project made it hard to proceed. Due the the recent release of Windows 10, my current operating system, support on getting DirectX to run is limited and as such put me far behind. I found that because of the new OS the SDK, software development kit, changed and as such, effected the whole project. I also en counted a few errors that I still haven't figured out the reasoning to, such as linker and manifest errors.

If I was to under take this project again I would try to add a algorithm to allow the cube count to be adjusted on the fly, which would calculate new coordinates and place the cubes. I would also like to add a HUD (heads up display) for the controls. Finally I would also play with the collisions more so they work as intended.