TO RUN: Please run the executable in the build folder

Controls: W,A,S,D to move camera
X, Spacebar to move up and down with camera
Mouse: to drag camera around

Simulation: The cloth is affected by gravity and slight wind and collides with the sphere by moving towards and away from it.

Forces with particles:

As a first step, we can use forces to model the movement of particles to make it look realistic. Forces can be translated into acceleration using Newton's second law and a numerical integration technique.

Simple F = ma

```
void addForce(Vec3 f)
{
        acceleration += f/mass;
}
```

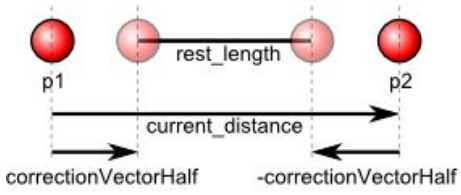
In our scenario, specifically for cloth, we would use verlet integration to convert from acceleration to position for each particle. The motivation for using verlet integration is because it is the most stable, easy to implement, and fast in comparison to other integration methods. Euler is fast and easy but not stable and Runga Kutta is stable and accurate but is not fast. To simulate air resistance we can add damping by decreasing the velocity over time. An easy way to do this would be multiplying the velocity vector with an amount between 0 and 1. Closer to 0 makes the particle move slower and slower each time step.

```
Pos = pos + (pos - oldPos) * (1.0 - DAMPING) + acceleration * deltaTime^2
Source code:
        Void verlet(){
                Vec3 temp = pos;
                Pos = pos + (pos - oldPos) * (1.0 - DAMPING) * acceleration *
deltaTime^2;
                Old_pos = temp;
                Acceleration = vec3(0,0,0) //reset acceleration vector
          }
        }
```

Constraints based system for cloth simulation:

I would like to introduce the topic of constraints to create the connection and movement between each particle in the cloth simulation so that they stay in a grid. The more "effective" we are at returning the particles to the same relative position, the more rigid the cloth will be. All connections in the cloth are based on a rest length. What I mean by that is we will base our connections on constraints of distances between pairs of particles. Therefore, each constraint

between two particles has a distance that it would like to return to for the cloth to be at rest -- we call that distance rest_distance.



https://viscomp.alexandra.dk/?p=147

Source code:

```
Void StatisfyConstraint(){
	Vec3 p1_to_p2 = p2-&gt;getPos()-p1-&gt;getPos(); // vector from p1 to p2
	float current_distance = p1_to_p2.length();
	Vec3 correctionVector = p1_to_p2*(1 - rest_distance/current_distance);
	Vec3 correctionVectorHalf = correctionVector*0.5;
	p1-&gt;offsetPos(correctionVectorHalf);
	p2-&gt;offsetPos(-correctionVectorHalf);
}
```

We can also add multiple constraints to our system. For example, we can add edge constraint to clamp the movement of a specific edge to make it look like the cloth is hanging on something.

```
for(int i = 0; i < intial_row; i++){
	Pos[i] = intial_Pos[i];
}
```

Collision response with Cloth and Sphere:

Collision response with a sphere is fairly easy and can be added as another constraint on the particles. We compare the distance of each particle, and if it is less then radius of the sphere we can then apply an offset to the particle based on the contact normal and position of the sphere.

Post mortem:

There are alot of thing I could have done better for this project, but in end the I felt like it was really gratifying to pull off something this cool. Even though, the amount of code was a wasn't as big as I thought it would be it was all about learning the process of making it. My simulation is extremely inefficient and unoptimized which can cause unnecessary lag. For the vertex buffer I am doing a double memory copy which is extremely bad, I could use SIMD to auto vectorize my operations for the vector math, have less render state changes, and clean up

my overall architecture. As for physics part of the simulation, I would like to implement various other models of simulating cloth and compare the results. In example would be an energy based system or maybe a mass spring system.

Sources:
https://viscomp.alexandra.dk/?p=147
http://www.darwin3d.com/gamedev/articles/col0599.pdf
https://graphics.stanford.edu/~mdfisher/cloth.html