CLOTH SIMULATION

AN EXPLORATION IN PHYSICS IN MODERN OPENGL

DEMO TIME.

BACKGROUND ON MATH REQUIRED

- Newton's Second Law F=Ma
- Hooke’s Law.
- Integration Methods
- Normal Calculation
- Constraint Solving

NEWTON'S SECOND LAW

$$F_{\text{net}}(v) = \text{Mg} + F_{\text{wind}} + F_{\text{air resistance}} - \sum_{\text{Springs} \in v} k(x_{\text{current}} - x_{\text{rest}}) = \text{Ma}$$

M = mass of vertex

g = gravity vector = $(0, -9.8, 0)$

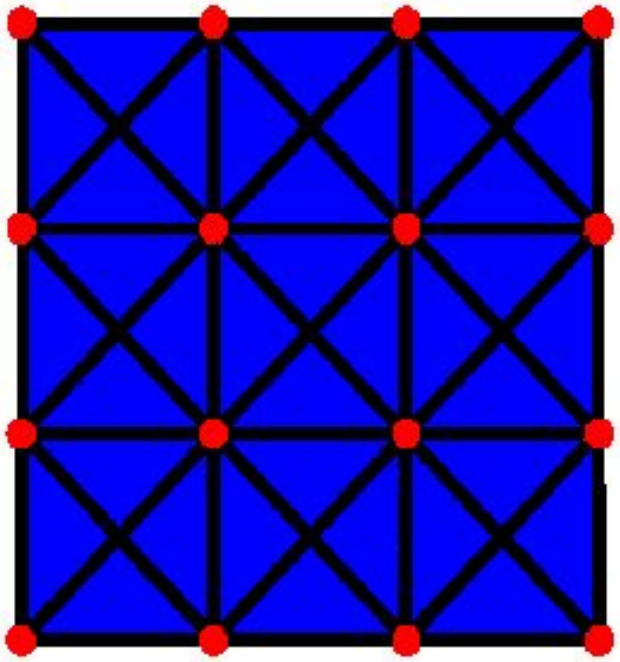
k = spring constant

x_{current} = current length of spring

x_{rest} = rest (initial) length of spring

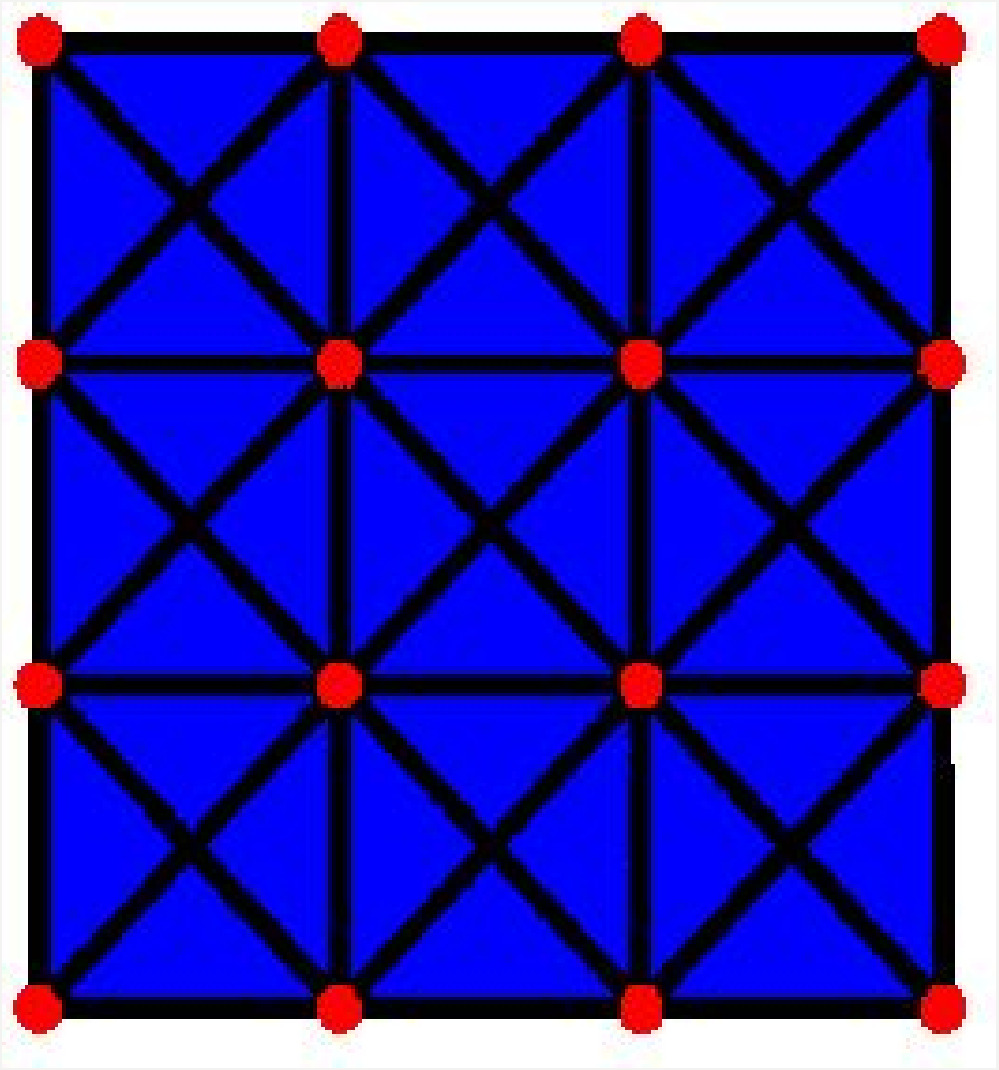
F_{wind} = wind vector

$F_{\text{air resistance}} = -a * \text{velocity}(v)^2$

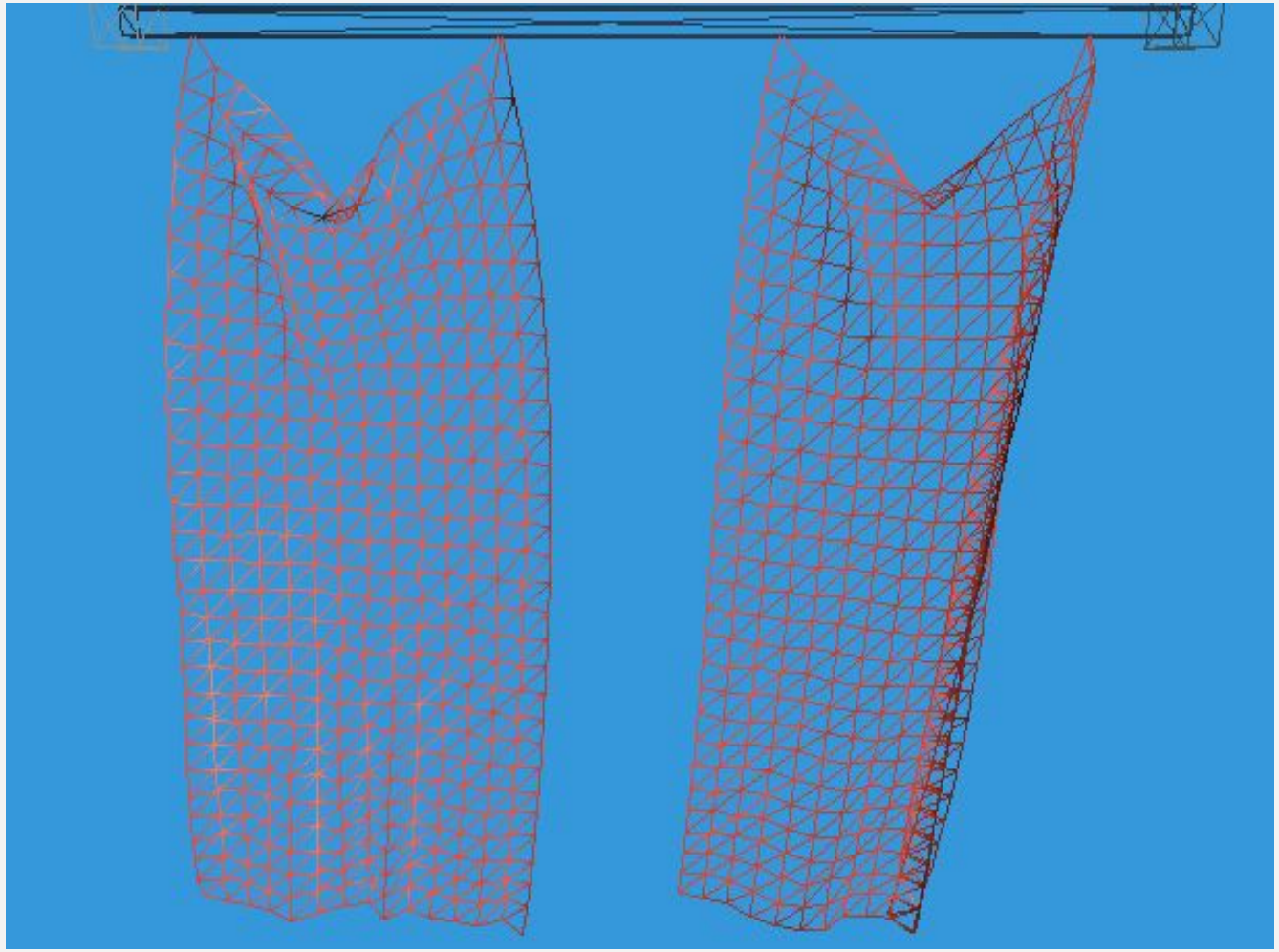


HOOKE'S LAW

$$-\sum_{\text{Springs} \in v} k(x_{\text{current}} - x_{\text{rest}})$$



OUR MESH LAYOUT



Masses and Springs

```
class Mass
{
   public:
       Mass(glm::vec3 position, bool movable);
       virtual ~Mass();
       void addSpring(Spring spring);
       void addForce(glm::vec3 force);
       void addPosition(glm::vec3 positionDelta);
       void calculateNewPosition();
       void constraintSolve();
       bool movable;
       glm::vec3 position;
       ...
```

```
class Spring
{
   public:
       Spring(Mass* fixedMass, Mass* dynamicMass);
       virtual ~Spring();
       void satisfyConstraint();

   private:
       Mass* fixedMass;
       Mass* dynamicMass;
       float restLength;
};
```

General Structure

- main.cpp
  - Cloth
    - Has many Masses
    - Has many Springs connecting those masses
    - Has many Vertexes
  - Hanger
    - Has many Racks
    - A Rack is cube structure scaled to be different objects.

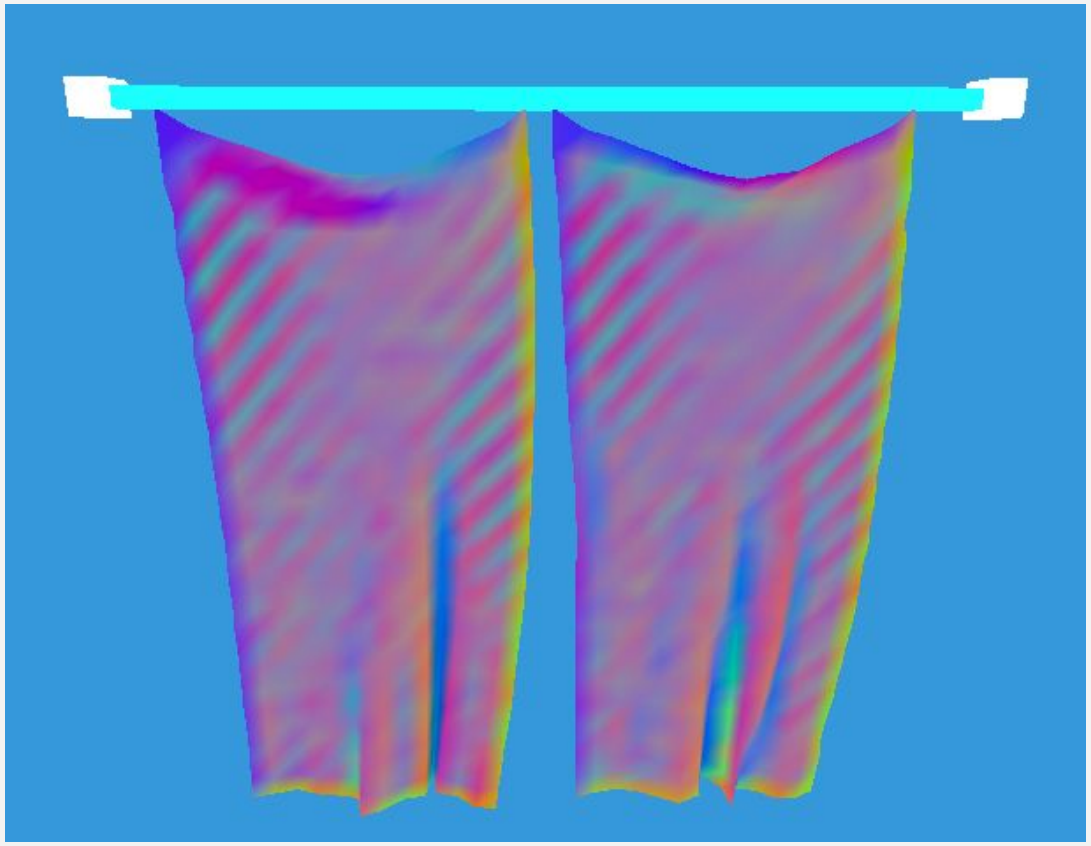
INTEGRATION METHODS

- Turing Our Force Into Motion
- Verlet Integration

$$x_{t+\Delta t} = 2\,x_t - x_{t-\Delta t} + \left(\frac{dv}{dt}\right)_t (\Delta t)^2$$

FACE NORMALS

- What would a cloth be without lighting



FACE NORMALS CONT.

- Using the cross product to find a perpendicular vector
- Add contributions to each vertex
- Normalize Results

CONSTRAINT SOLVING

- How does the cloth know how to move
- Non Determinate Algorithm
- i.e. we take passes in order to solve

SOME INTERESTING OPENGL ODDITIES

- GL_STATIC_DRAW vs. GL_DYNAMIC_DRAW
  - Create it once, set it once.
  - Create it once, change it a lot.

```
glBufferData(GL_ARRAY_BUFFER, this->positionSize *
sizeof(GLfloat), positionCords, GL_DYNAMIC_DRAW);
```

INTERLEAVING VS. MULTI VBO

- There is quite the question about how one should layout their data to the GPU

SOMETHING LEARNED

- VBA can be thought of as a dynamic struct.
- Bind Calls add fields to the VBA that can be referenced by the Draw calls
- OpenGL runs on global state.

FUTURE WORK

- Implementing primitive sphere collisions is considered quite trivial we would definitely like to pursue this further.
- Implement Energy Conservation Model.

QUESTIONS?