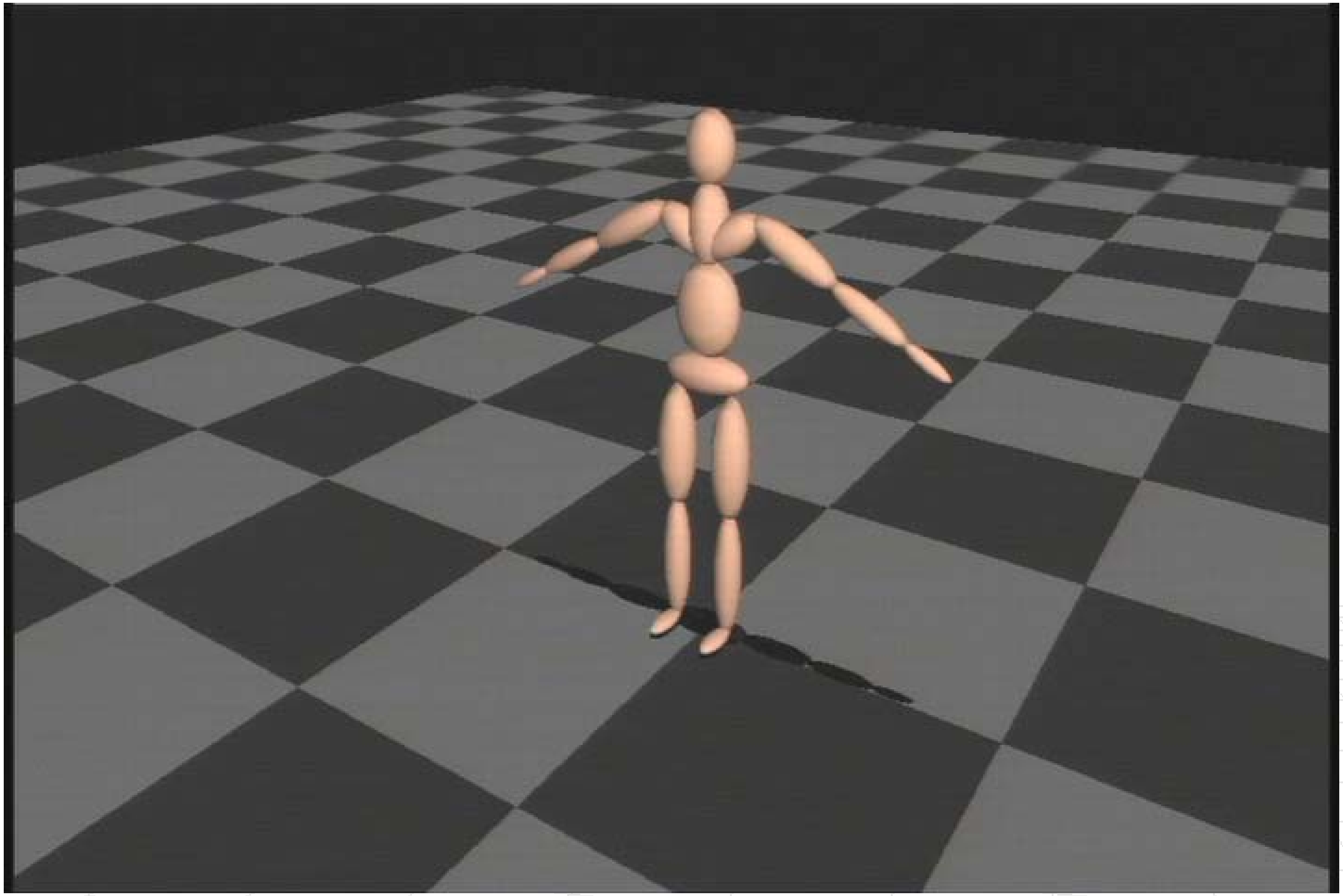
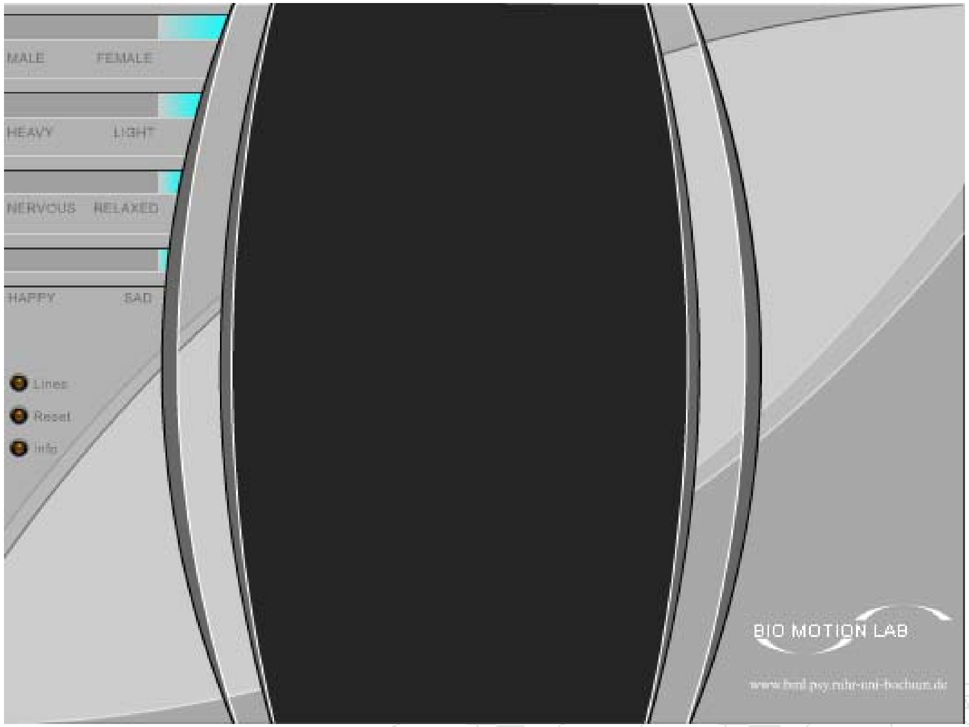
Announcements

- Alex will have **office hours** on Thursday 11am-12pm in SW625A

Inverse Kinematics Example



MoCap Demo



NikoTroje

- **Approach**
  - Measure motion of many actors using MoCap system
  - Derive “basis” that span the different styles of motion

Physics-Based Animation

Computer Graphics, CSCD18

Fall 2008

Instructor: Leonid Sigal

Animation vs. Simulation

- **Animation:** Make objects change over time according to a scripted actions within story-line
  - It's generally hard to achieve physical realism

- **Simulation:** Predict how objects change over time according to physical laws
  - Math becomes **very complicated very quickly**

Note that nature can compute everything (all of you in the class, floor response forces, etc.) on the fly, computers can't

  - Issues to think about in simulation
    - Simulation accuracy
    - Simulation stability

Forward vs. Inverse Dynamics

- **Forward Dynamics**
  - Given a sequence of forces find the motion that result **x(t)**
  - Reasonably easy to solve simple cases, but very sensitive to initial conditions, and difficult to control and predict paths of objects

- **Inverse dynamics**
  - Given path **x(t)** and masses, find forces forces that generated it
  - Often under constrained and very difficult to solve computationally

Physics-based Simulation

- **Solve**

$$\vec{\mathbf{F}} = \mathbf{m}\vec{\mathbf{a}}$$

- Sounds easy, and it is for very simple rigid objects that do not articulate, deform, melt, etc.

- **Approach**
  - Forces and masses determine accelerations
  - Velocity and positions are estimated numerically

Physics-based Simulation

- **Solve**

$$\vec{\mathbf{F}} = \mathbf{m}\vec{\mathbf{a}}$$

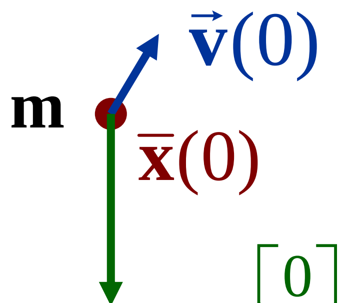
- Sounds easy, and it is for very simple rigid objects that do not articulate, deform, melt, etc.

- Doing if for realistic objects is hard
  - Imagine modeling water with droplets
  - How about, hair with folic particles

Final Fantasy: The Spirit Within, 2001
Dr. Aki Ross' hair is only half as dense as the average human crop, however, that still left 60,000 strands to put realistically into motion. When designing the computer graphics, a fifth of the time spent was devoted to those 60,000 hairs.

Simple Particle System

- Let's say we have a small particle of mass (**m**)
- We can then (given initial conditions) compute the motion of this particle



$$\mathbf{F} = \mathbf{m}\mathbf{g} = \mathbf{m}\begin{bmatrix} 0 \\ 0 \\ \mathbf{g} \end{bmatrix}$$

Euler Simulation

$\bar{\mathbf{x}}(\mathbf{t})$ - position

$\vec{\mathbf{v}}(\mathbf{t}) = \dfrac{\mathbf{d}\bar{\mathbf{x}}(\mathbf{t})}{\mathbf{dt}} = \bar{\mathbf{x}}(\mathbf{t}+1) - \bar{\mathbf{x}}(\mathbf{t})$ - velocity

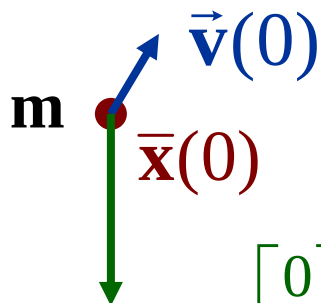
$\vec{\mathbf{a}}(\mathbf{t}) = \dfrac{\mathbf{d}\vec{\mathbf{v}}(\mathbf{t})}{\mathbf{dt}} = \vec{\mathbf{v}}(\mathbf{t}+1) - \vec{\mathbf{v}}(\mathbf{t})$ - acceleration

$$\vec{\mathbf{F}} = \mathbf{m}\vec{\mathbf{a}}$$

$$\vec{\mathbf{a}}(0) = \frac{\mathbf{F}}{\mathbf{m}} = \frac{\mathbf{m}\vec{\mathbf{g}}}{\mathbf{m}} = \vec{\mathbf{g}}$$

Simple Particle System

- Let's say we have a small particle of mass (**m**)
- We can then (given initial conditions) compute the motion of this particle



$$\mathbf{F} = \mathbf{m}\mathbf{g} = \mathbf{m}\begin{bmatrix} 0 \\ 0 \\ \mathbf{g} \end{bmatrix}$$

Euler Simulation

$$\overline{\mathbf{x}}(1) = \overline{\mathbf{x}}(0) + \vec{\mathbf{v}}(0)$$

$$\vec{\mathbf{v}}(1) = \vec{\mathbf{v}}(0) + \vec{\mathbf{g}}$$

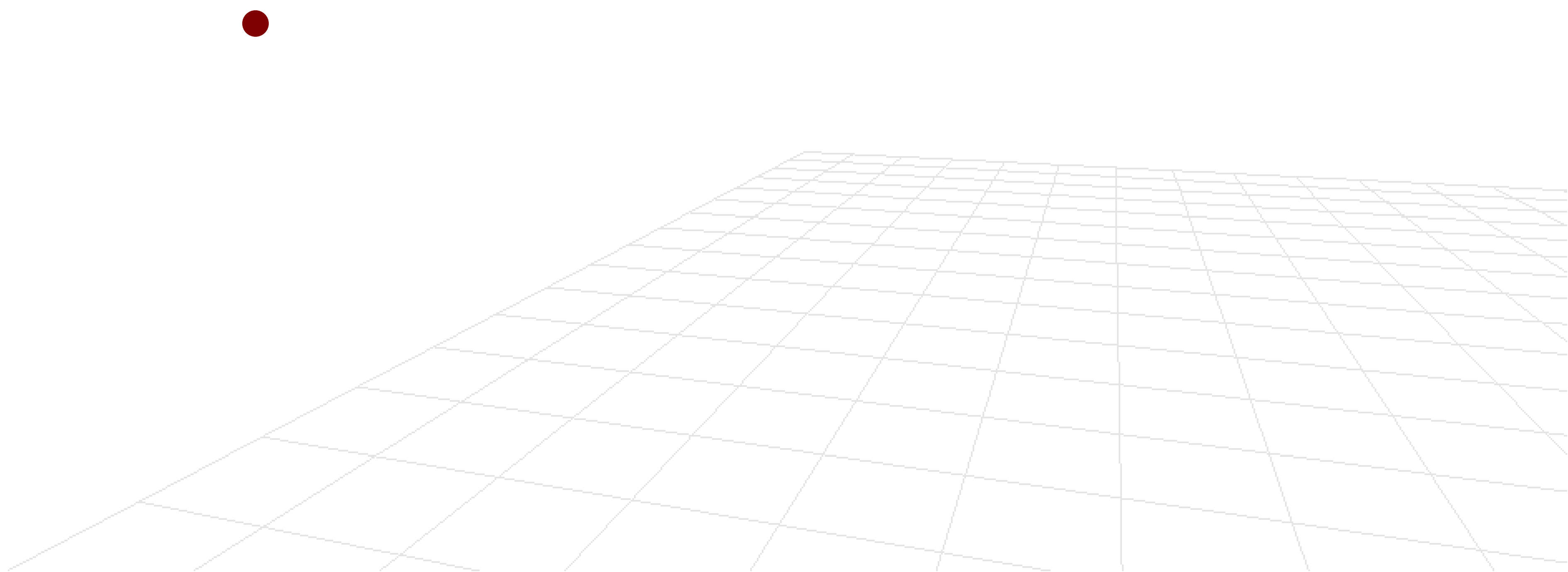
$$\vec{\mathbf{a}}(0) = \vec{\mathbf{g}}$$

$$\vec{\mathbf{F}} = \mathbf{m}\vec{\mathbf{a}}$$

$$\vec{\mathbf{a}}(0) = \frac{\mathbf{F}}{\mathbf{m}} = \frac{\mathbf{m}\vec{\mathbf{g}}}{\mathbf{m}} = \vec{\mathbf{g}}$$

Simple Particle System

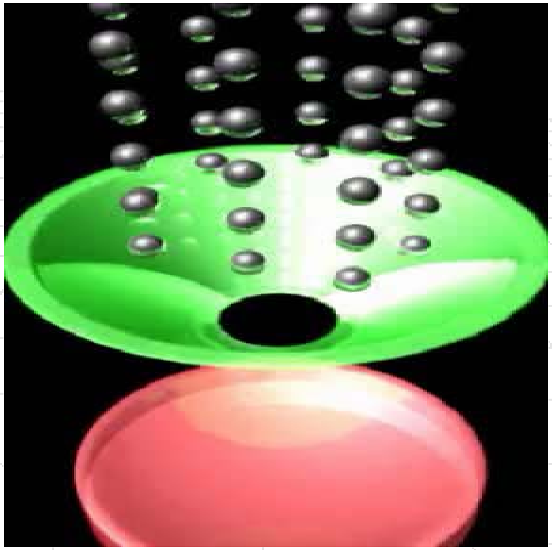
- Let's say we have a small particle of mass (**m**)
- We can then (given initial conditions) compute the motion of this particle



Collisions

- **Modeling collisions is very hard**

- To deal with collisions typically you need to
  - Detect when collisions happen
  - Stop simulation at the time of collision
  - Adjust the conditions (e.g. depending on the elasticity of collision)
  - Re-start simulation



What can we model with this?

- With many many particles we can model
  - Fireworks
  - Waterfalls
  - Water fountains

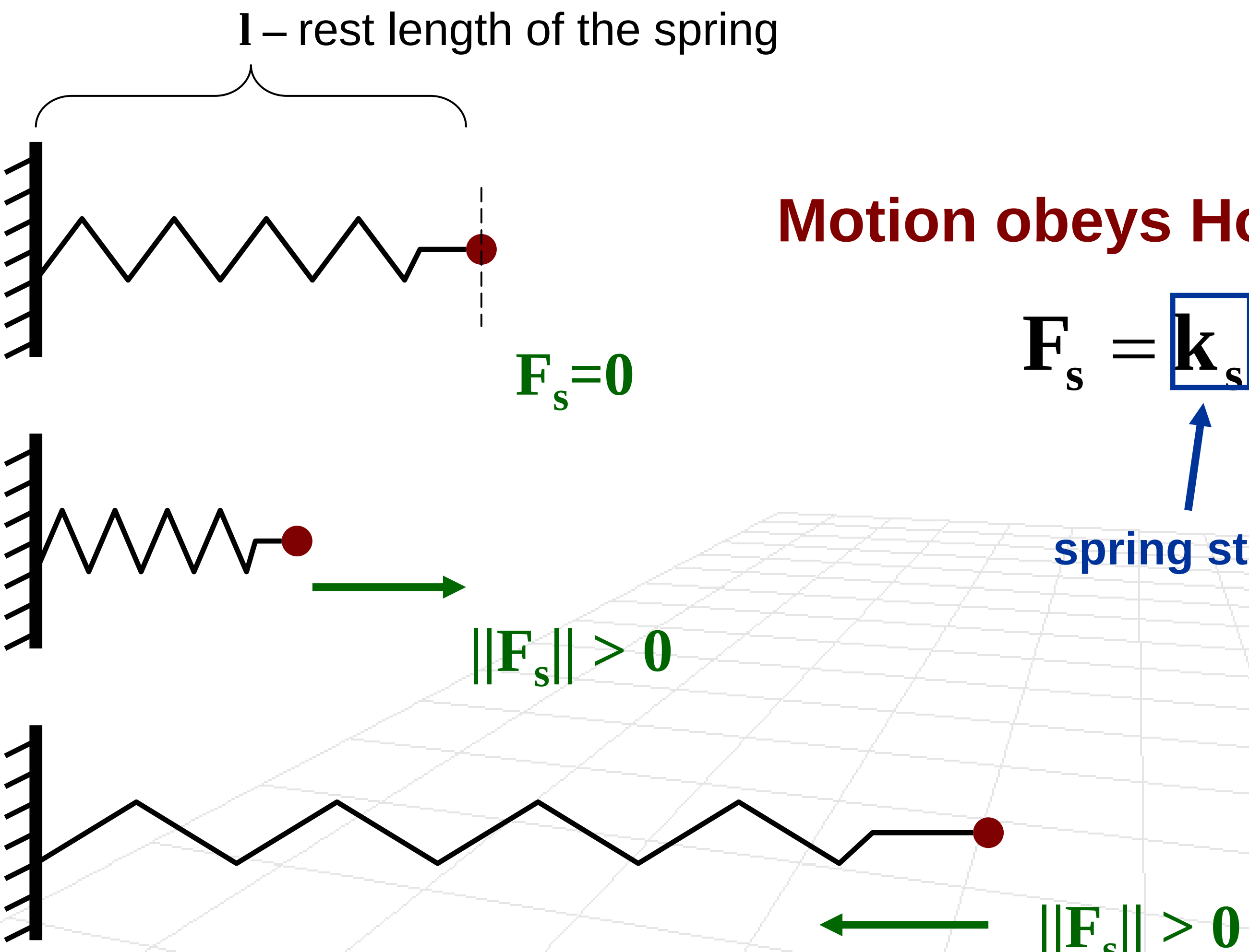
- What if we are willing to add additional forces to our particles (e.g. thermodynamic forces)
  - Fire
  - Wind
  - Clouds

- Particles may also have
  - Shape (birds), color, age

http://www.youtube.com/watch?v=nUrodylyJlg

Mass-Spring System

- Let's say we have the same mass (**m**) but on a spring



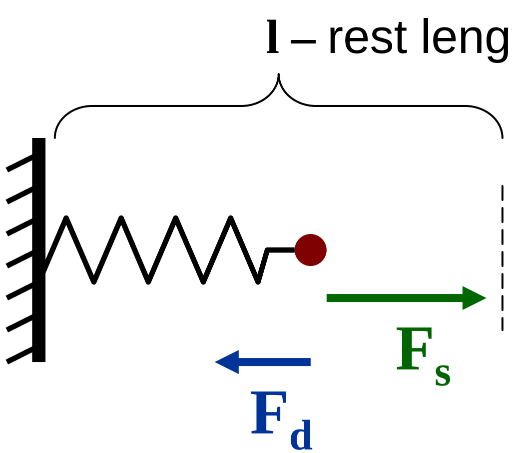
Motion obeys Hooks Law:

$$\mathbf{F_s = k_s(x - l)}$$

spring stiffness

Mass-Spring System Damping

- **Problem:** springs store energy, so the system will oscillate forever (unrealistic) … typically in all real systems energy is dissipated



Motion obeys Hooks Law:

$$\mathbf{F_s} = \mathbf{k_s}(\mathbf{x} - \mathbf{l})$$

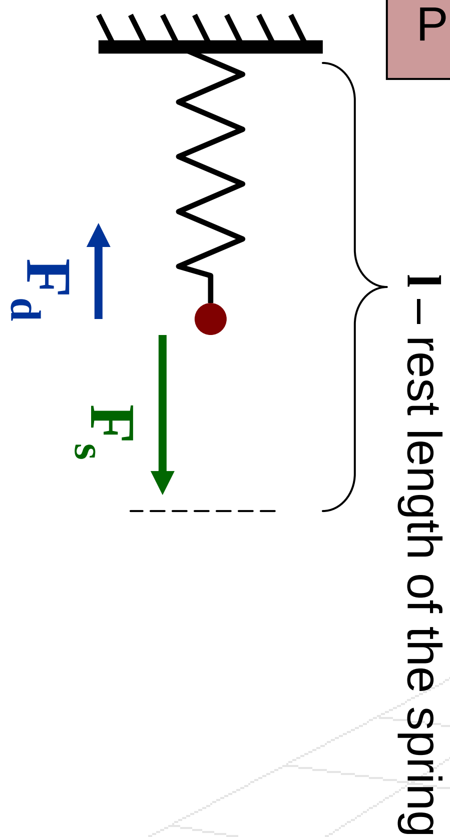
Damping:

$$\mathbf{F_d} = -\boxed{\mathbf{k_d}}(\mathbf{v})$$

damping constant

Mass-Spring System with Gravity

Total force on the particle: $$\mathbf{F} = \mathbf{F}_s + \mathbf{F}_d + \mathbf{F}_g$$



P.S. From here we can do simulation using Euler as before

Motion obeys Hooks Law:

$$\mathbf{F}_s = \mathbf{k}_s(\mathbf{x} - \mathbf{l})$$

Damping:

$$\mathbf{F}_d = -\mathbf{k}_d(\mathbf{v})$$

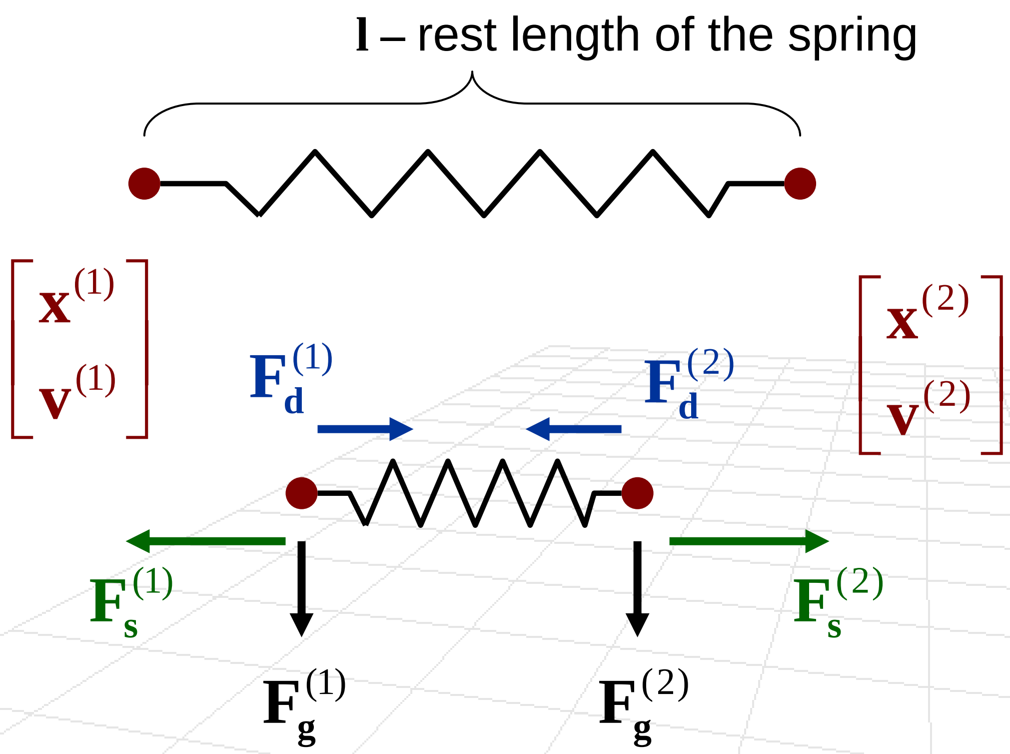
Gravity:

$$\mathbf{F}_g = \mathbf{mg}$$

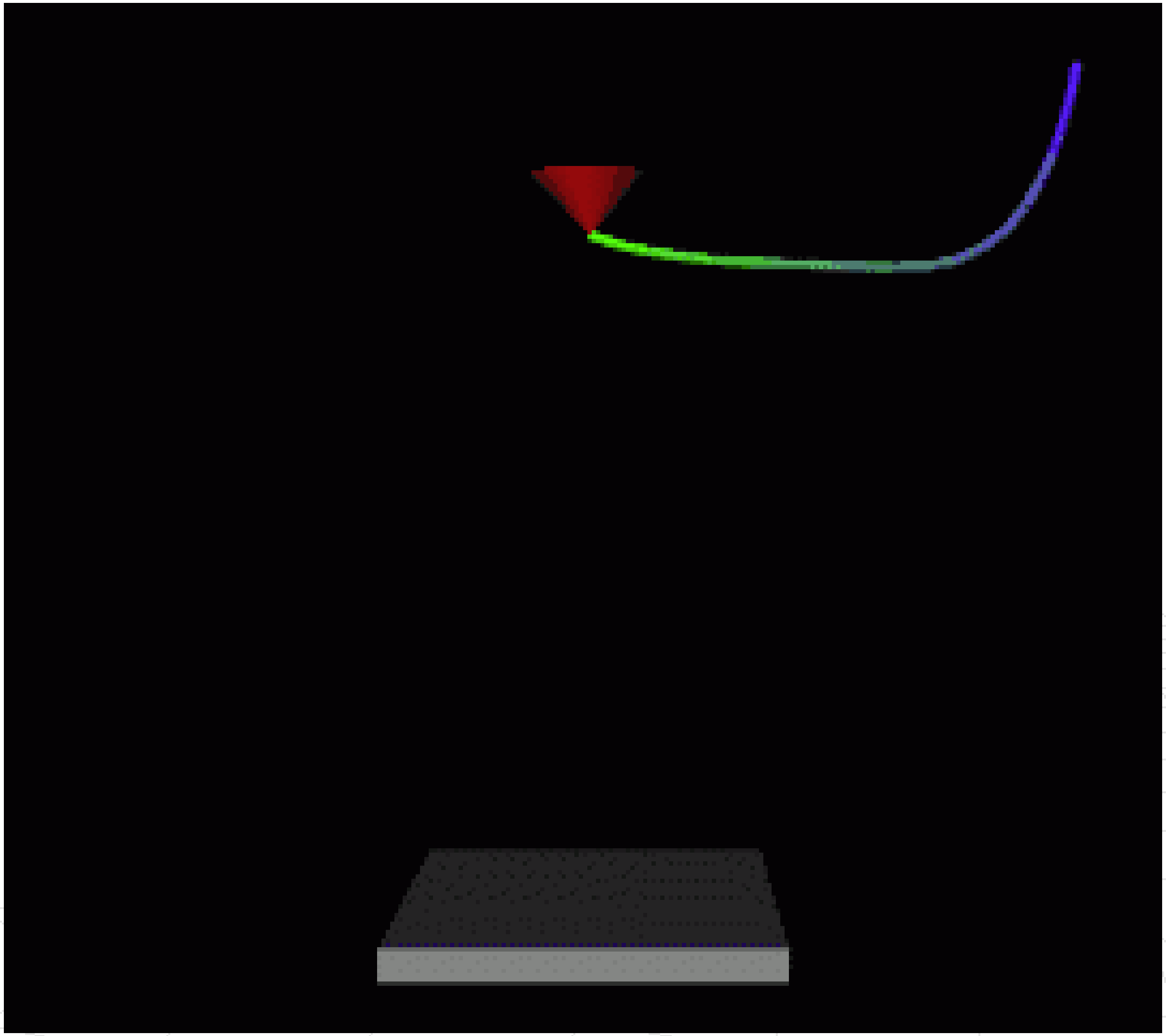
Mass-Spring System with Two Particles

- For every particle we need to keep track of **position**, **velocity** and **acceleration** (solved for by accounting for the forces on that particle)

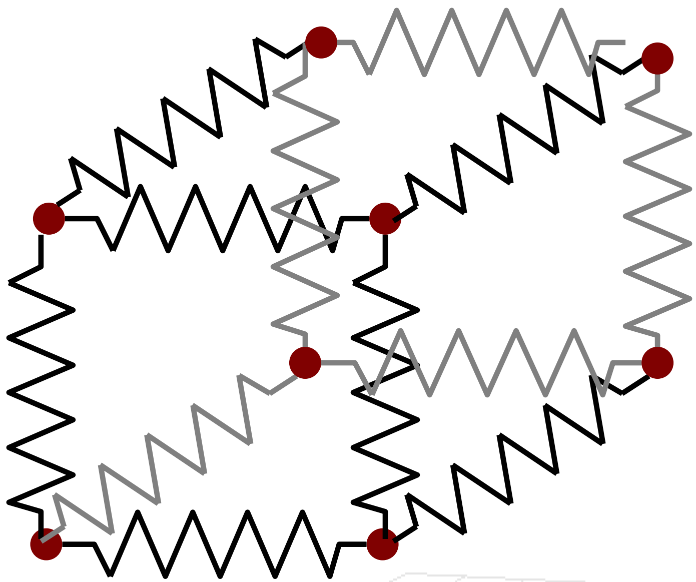
- Easy to extend to 3D (but we're not going to do it in interest of time)



Example



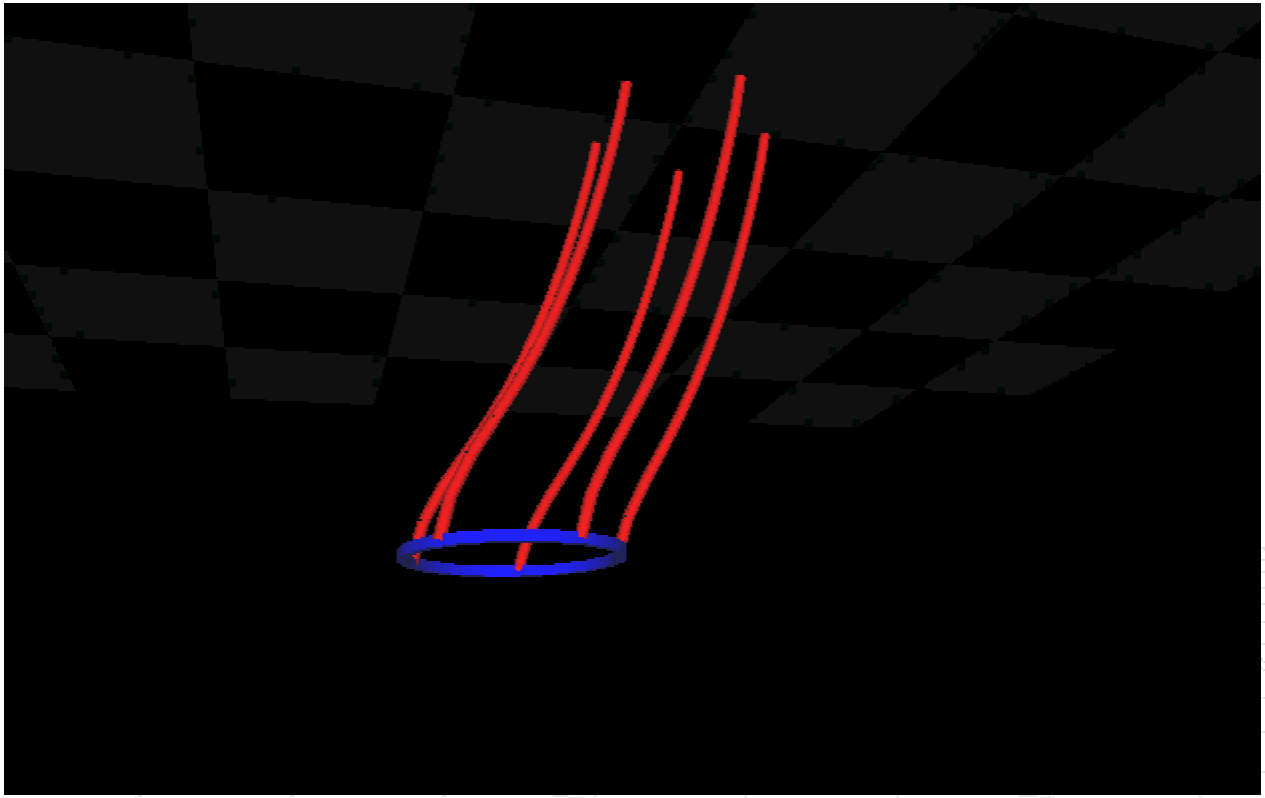
More complex spring-mass systems



Jelly Cube

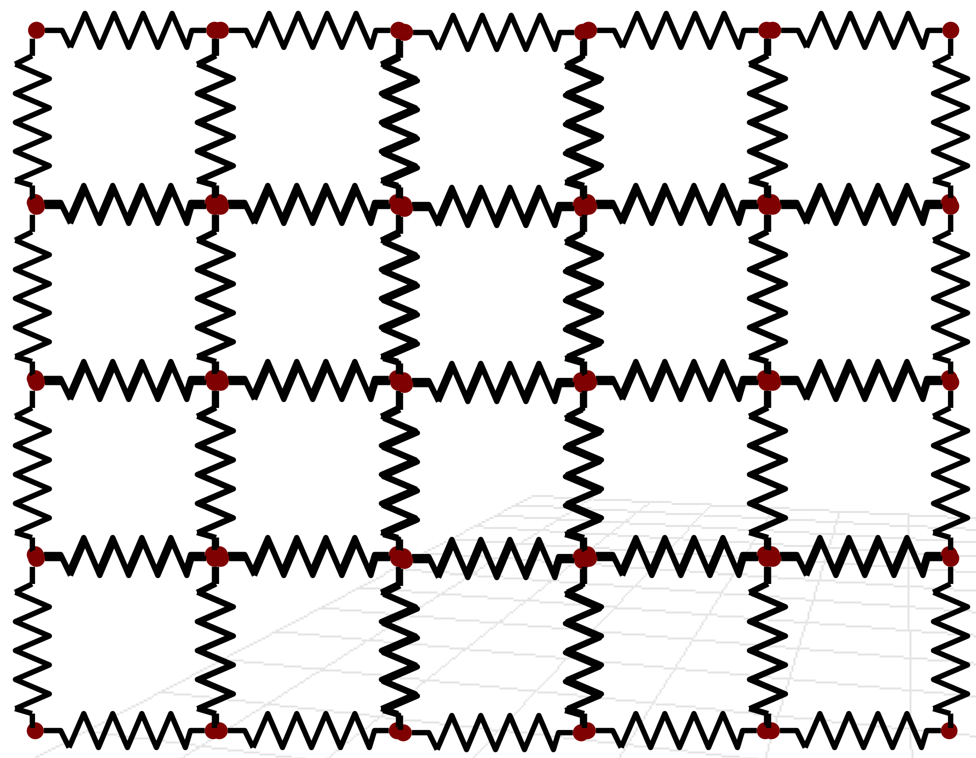
By applying forces we can make the cube “wiggle”, but we can also make it as “strong” as we want by setting the spring constants very high

Example

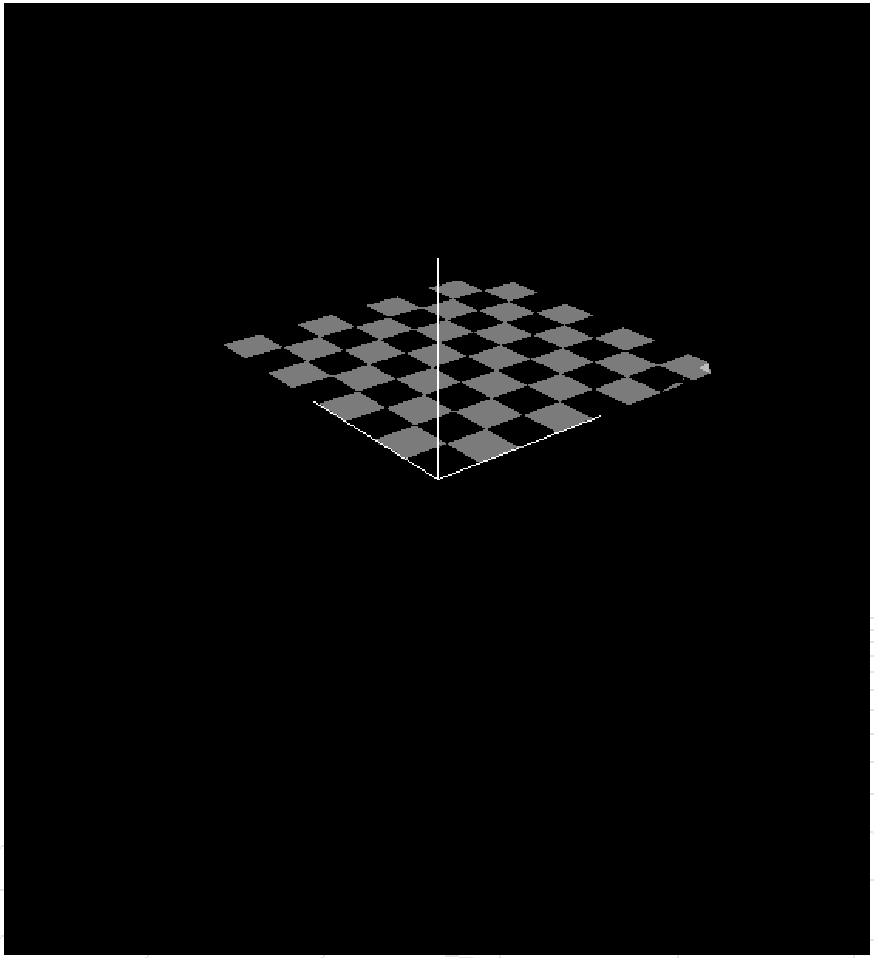


What if we have a really big system?

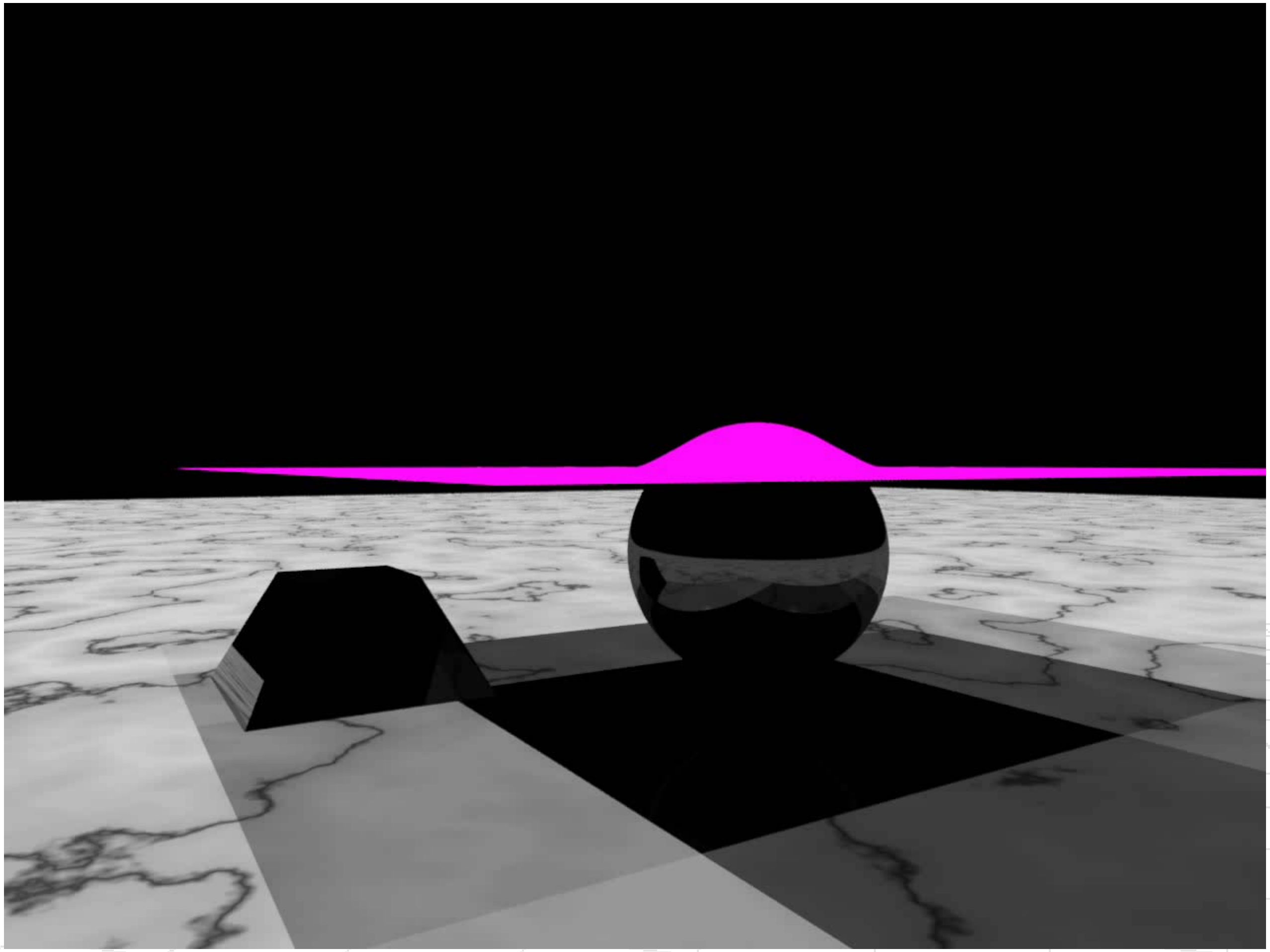
We can model cloth !!!



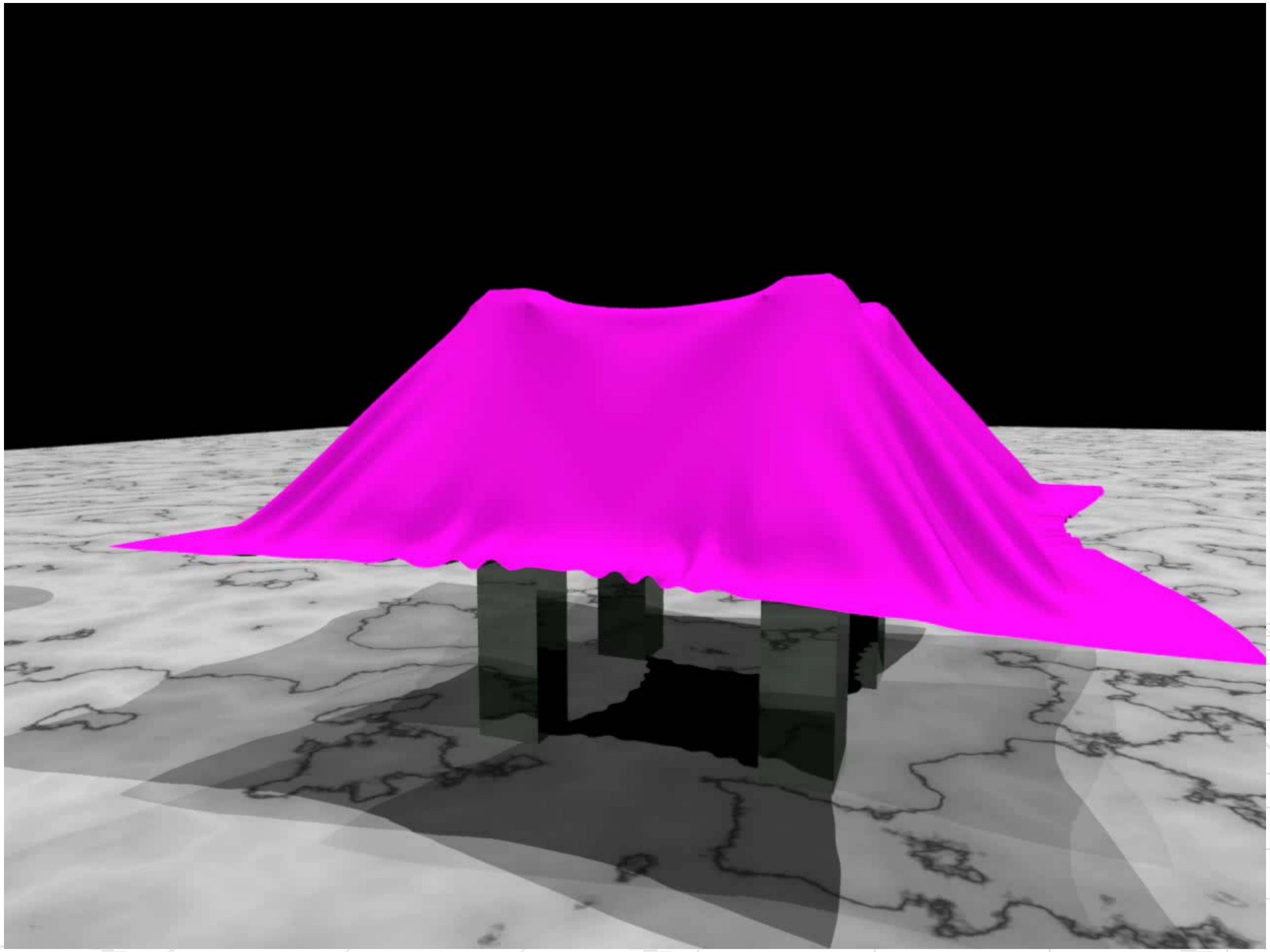
Cloth Example



Cloth with Collisions



Cloth with Collisions



Rubbery “Cloth”

Physics-based Animation

- **Pros:**
  - Very realistic
- **Cons:**
  - Very expensive and slow
  - Very difficult to control
- **Use:**
  - Various physical phenomenon