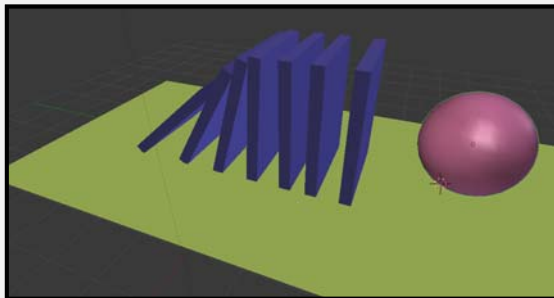
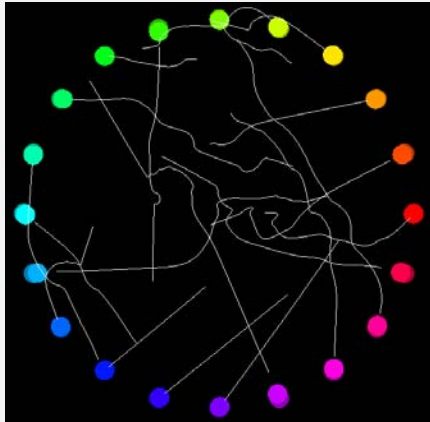


# Animation



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)

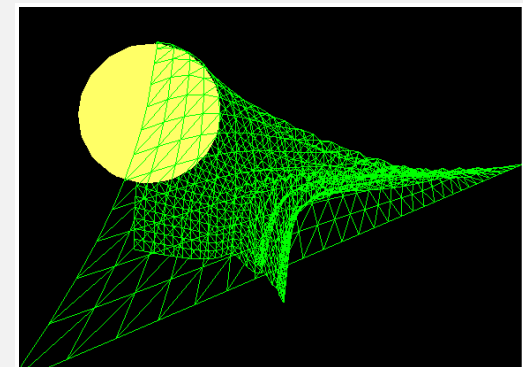
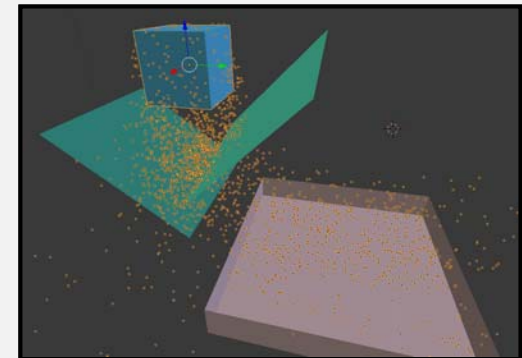


**Oregon State**  
University  
Computer Graphics



**Oregon State**  
University  
Mike Bailey

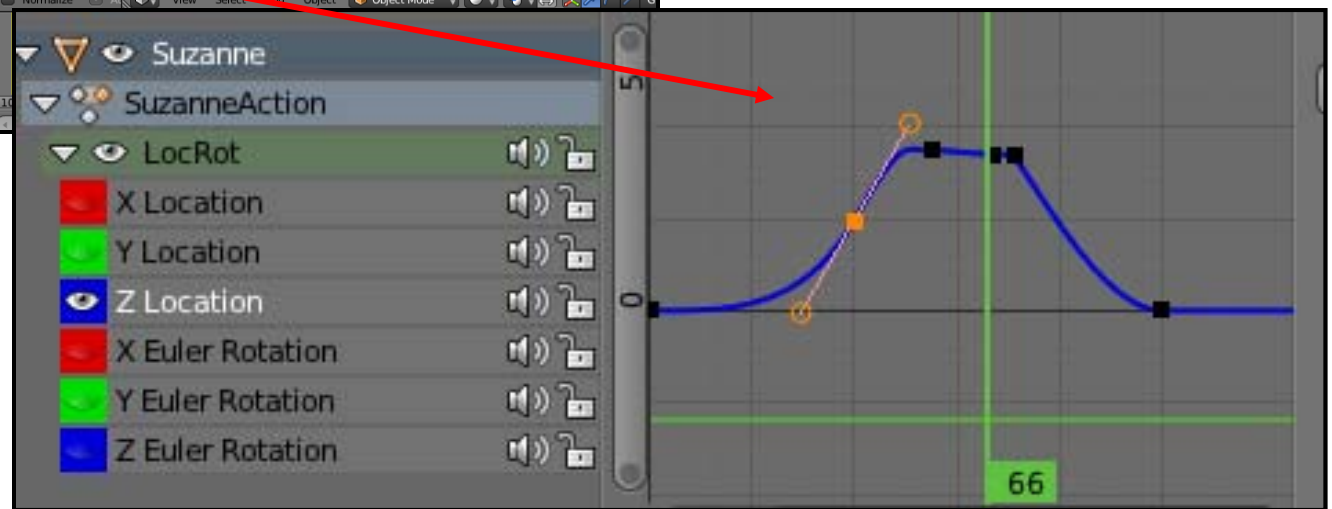
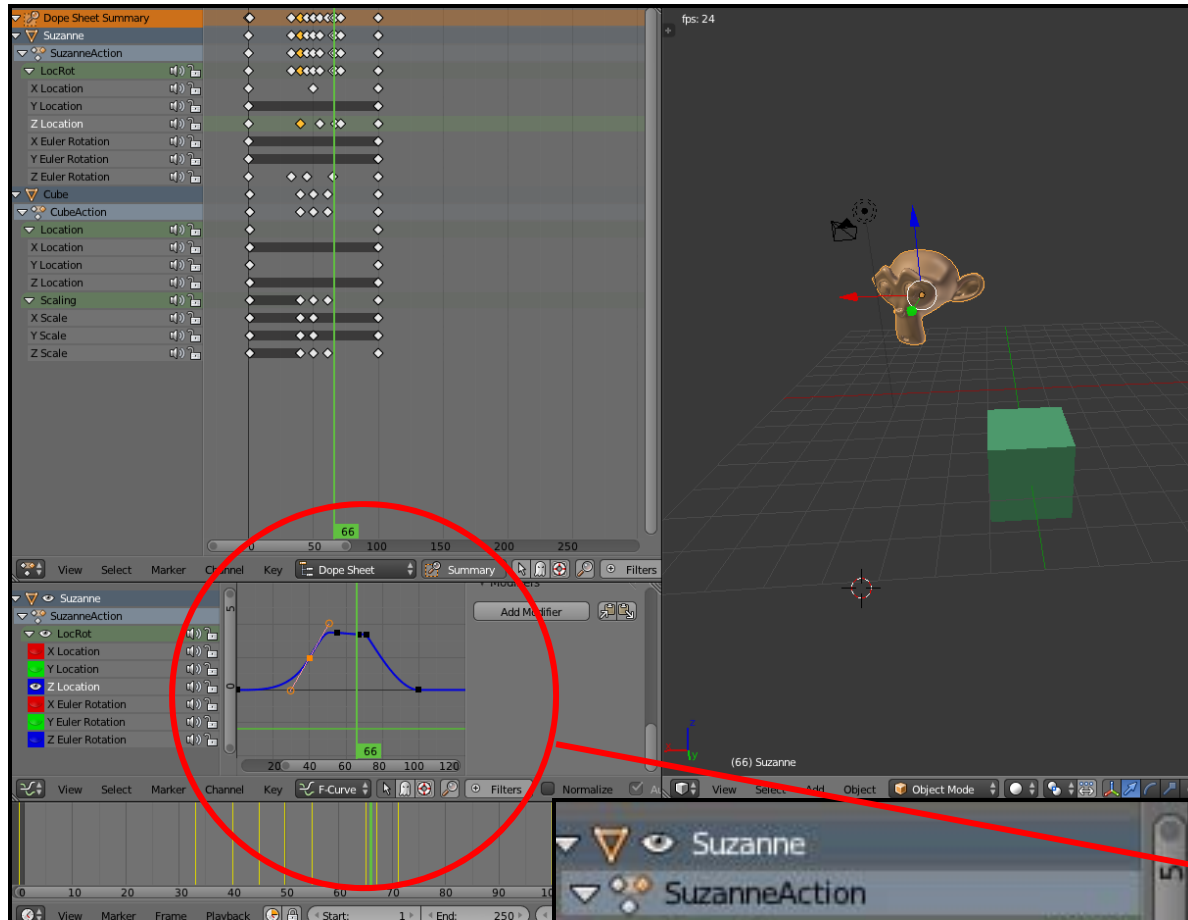
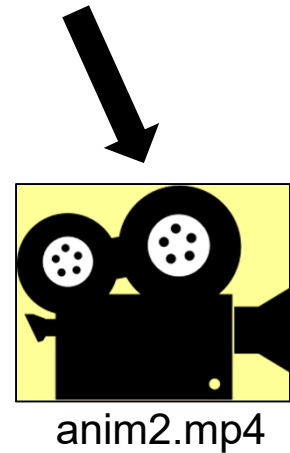
mjb@cs.oregonstate.edu



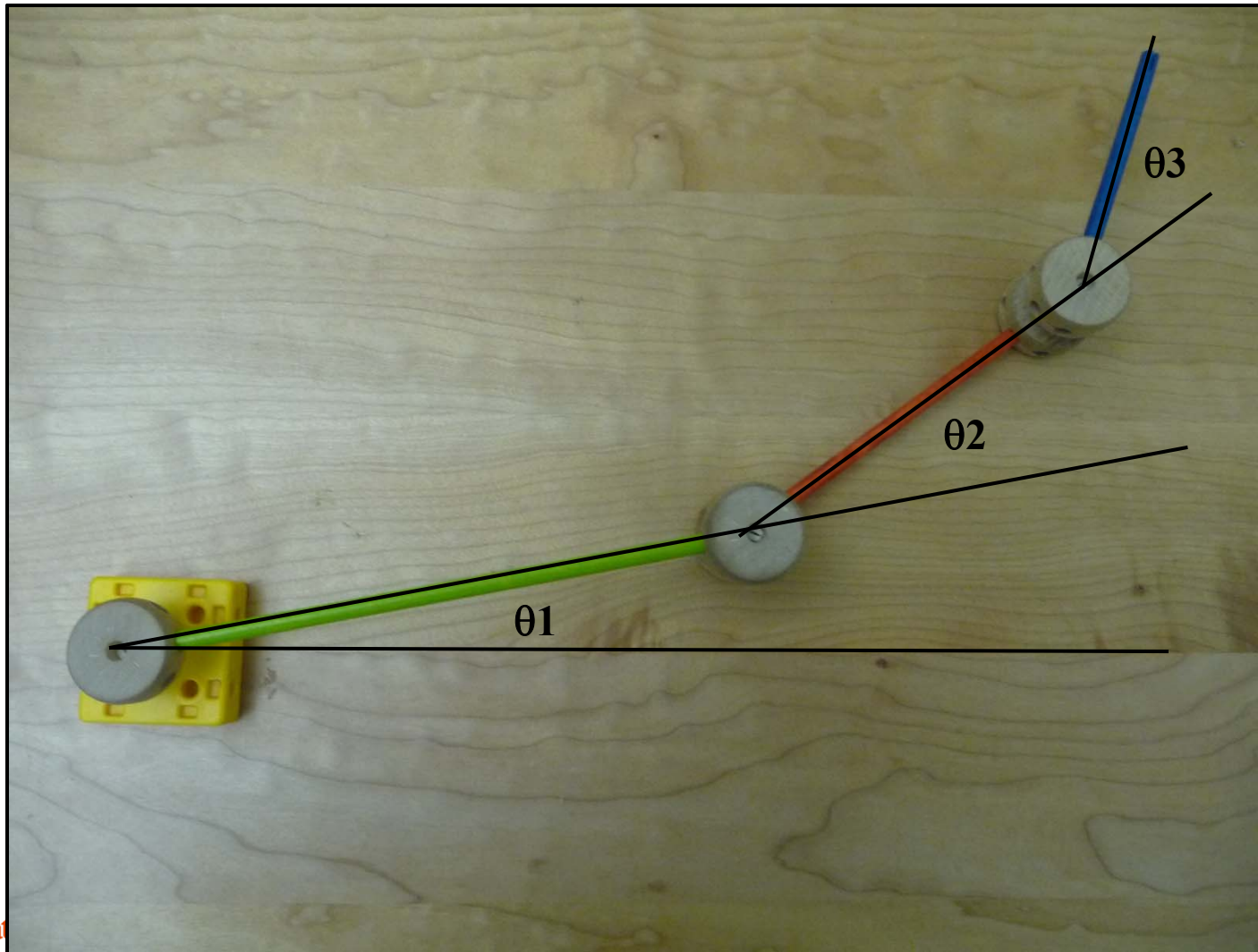
# Keyframe Animation

2

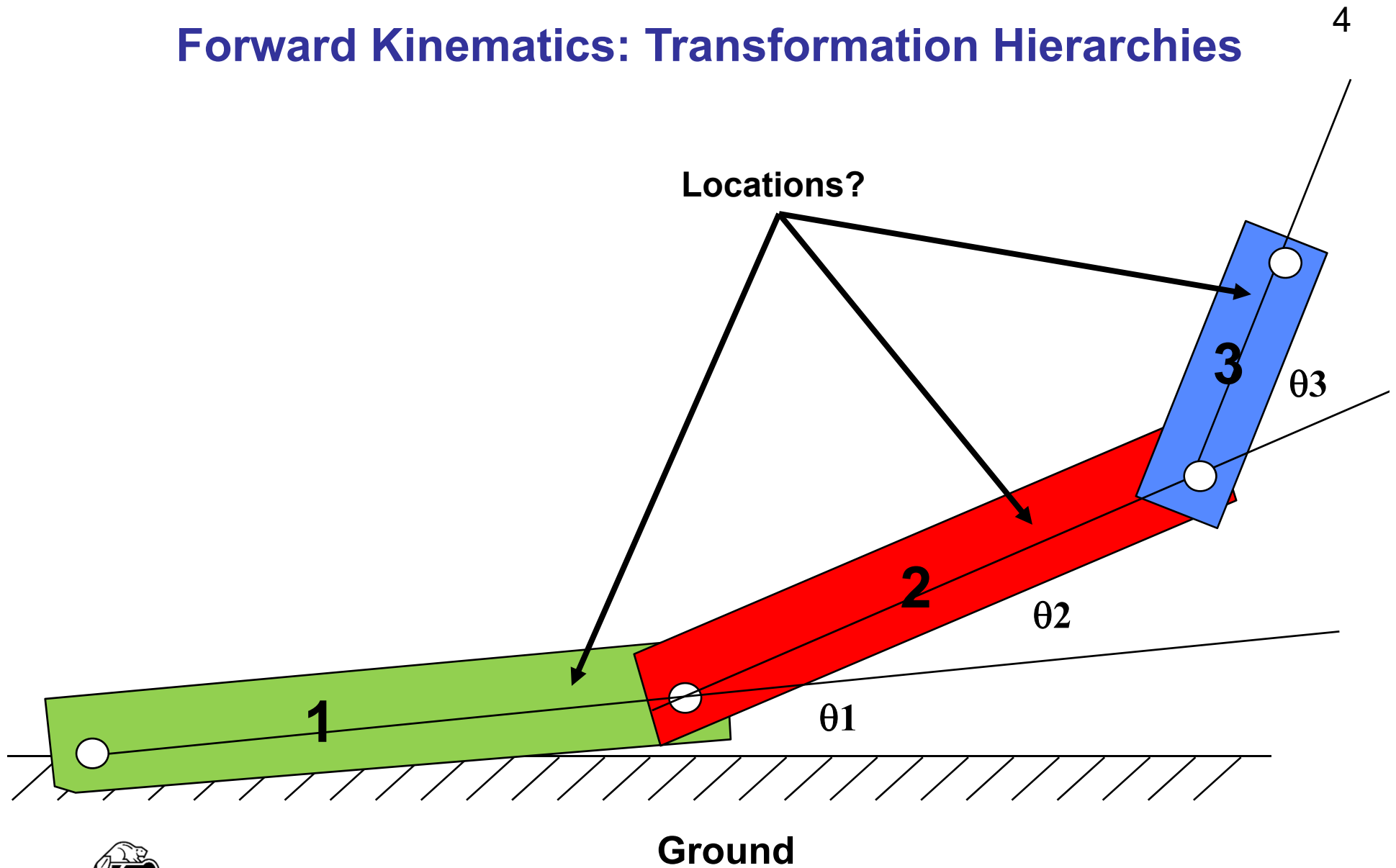
These icons refer to explanatory videos on the class web site



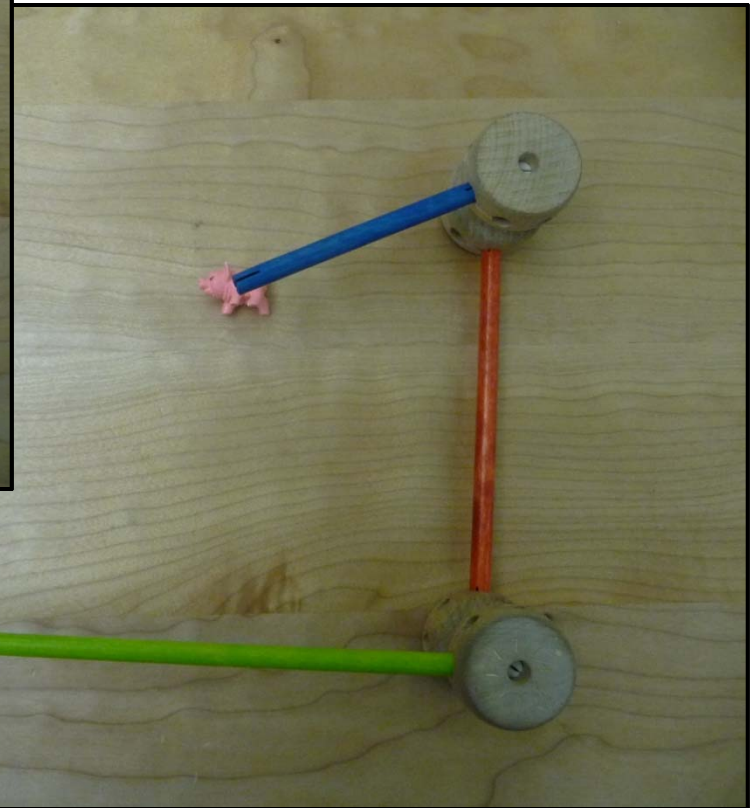
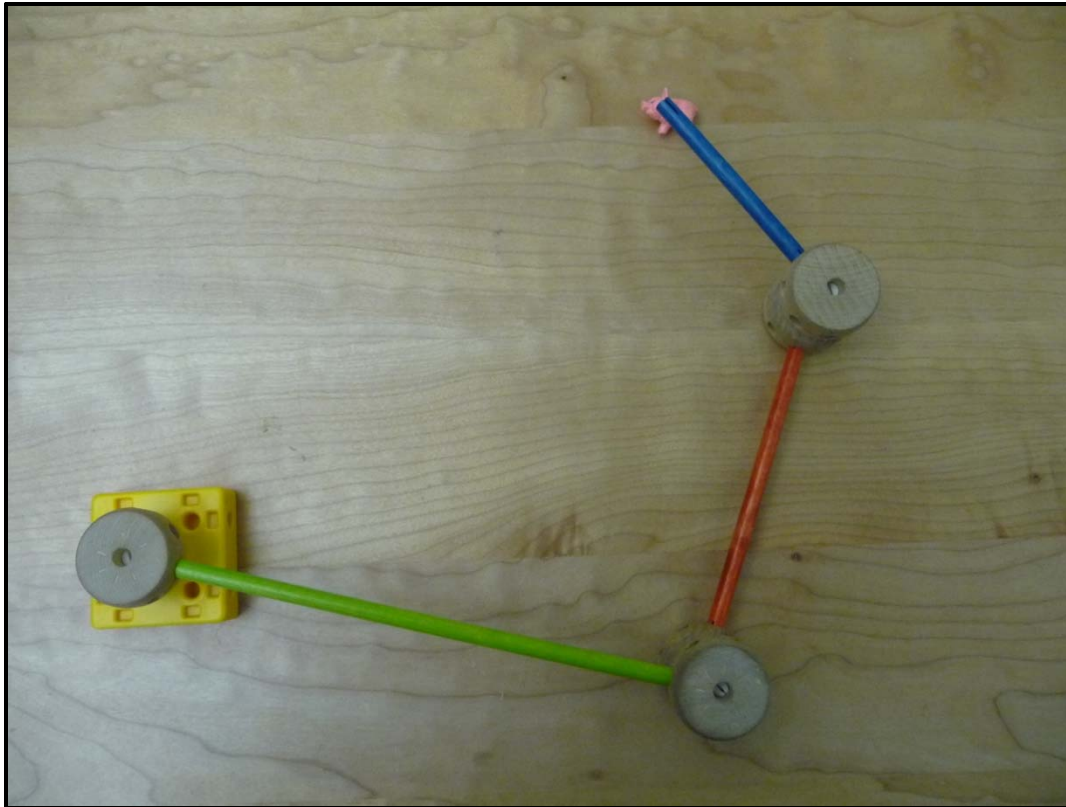
# Forward Kinematics: Change Parameters – Things Move (All Children Understand This)



# Forward Kinematics: Transformation Hierarchies



## Inverse Kinematics (IK): Things Need to Move to a Particular Location – What Parameters Will Make Them Do That?



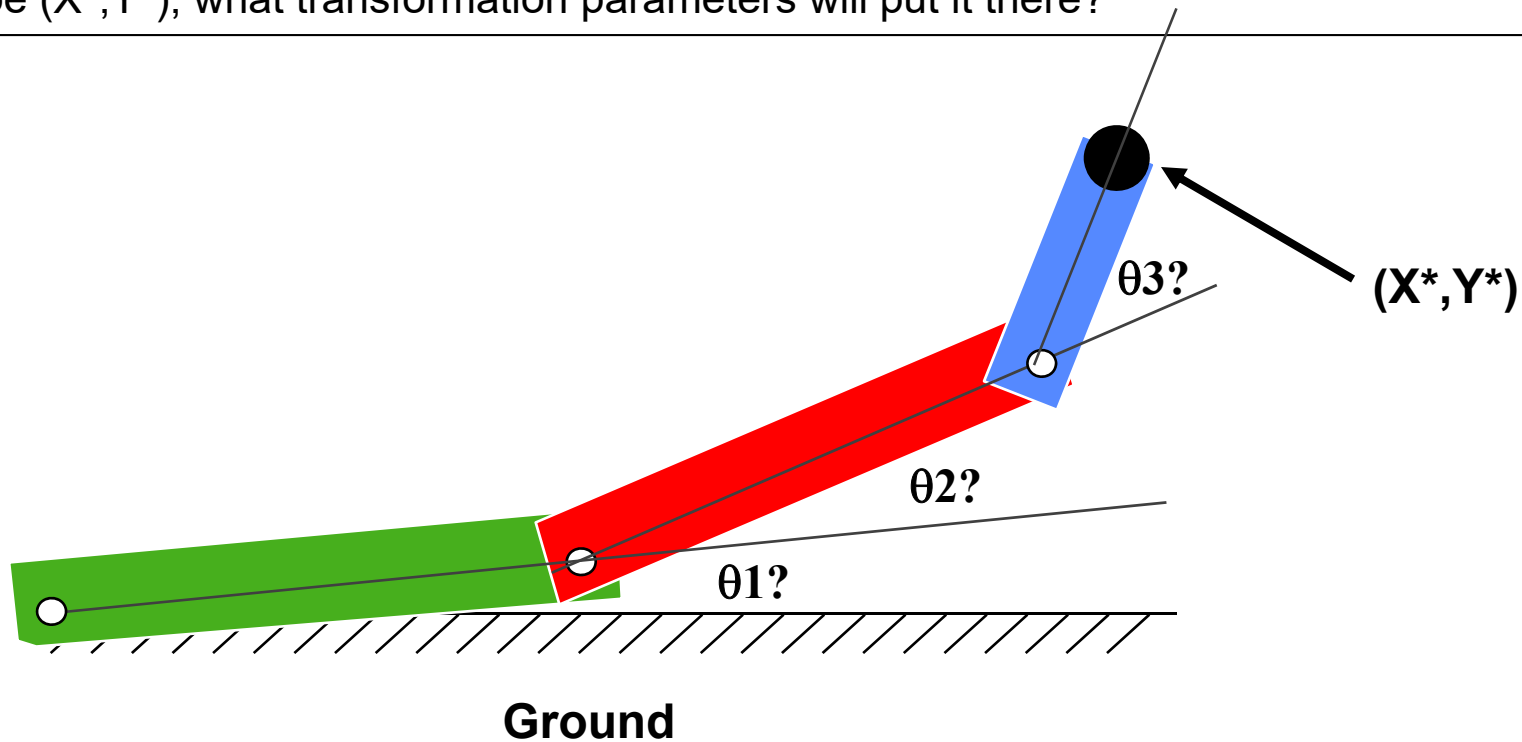
Of course, there will always be target locations that can *never* be reached. Think about that spot in the middle of your back that you can never scratch! 😊



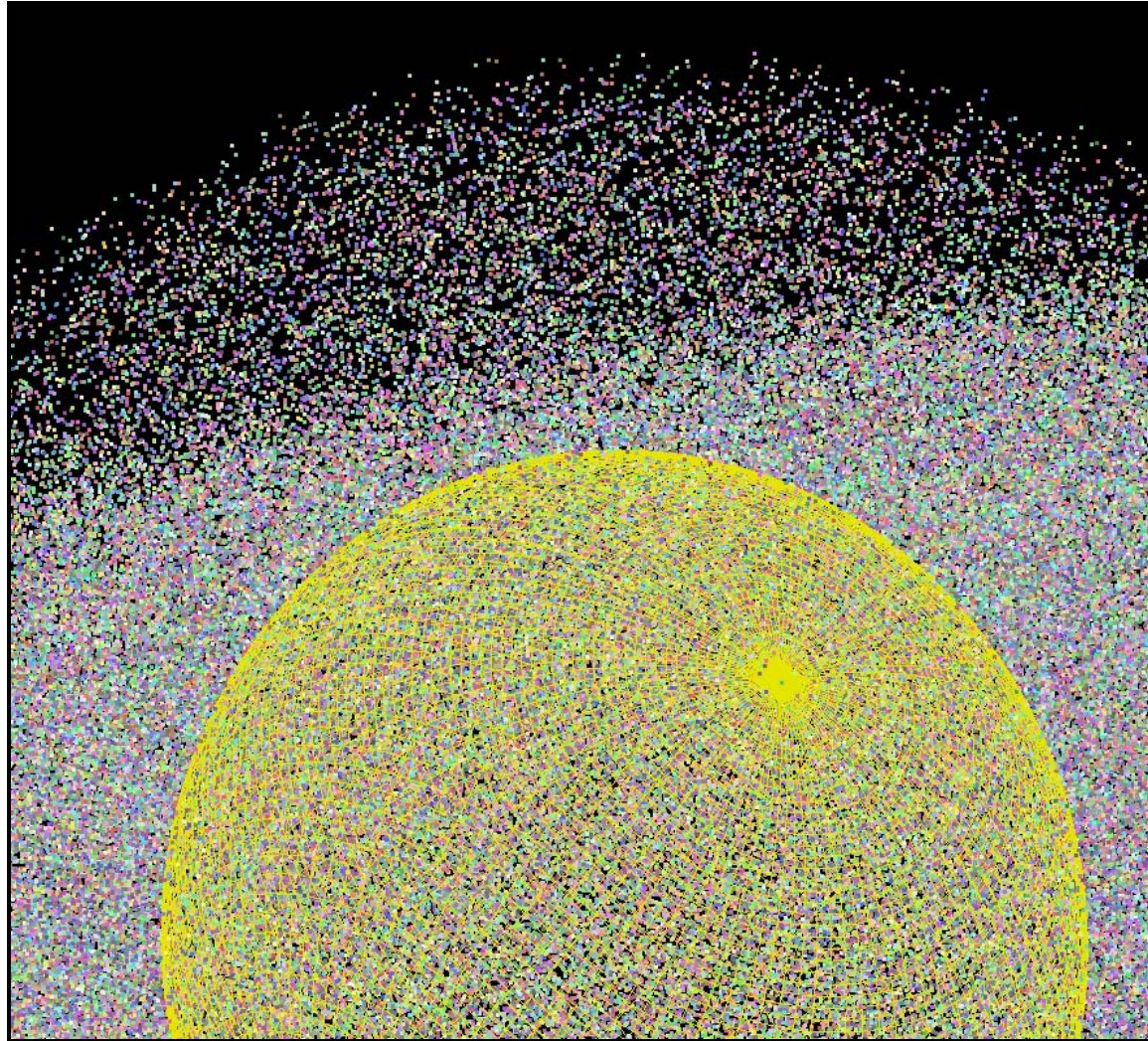
# Inverse Kinematics (IK)

**Forward Kinematics** solves the problem “if I know the link transformation parameters, where are the links?”.

**Inverse Kinematics (IK)** solves the problem “If I know where I want the end of the chain to be  $(X^*, Y^*)$ , what transformation parameters will put it there?”

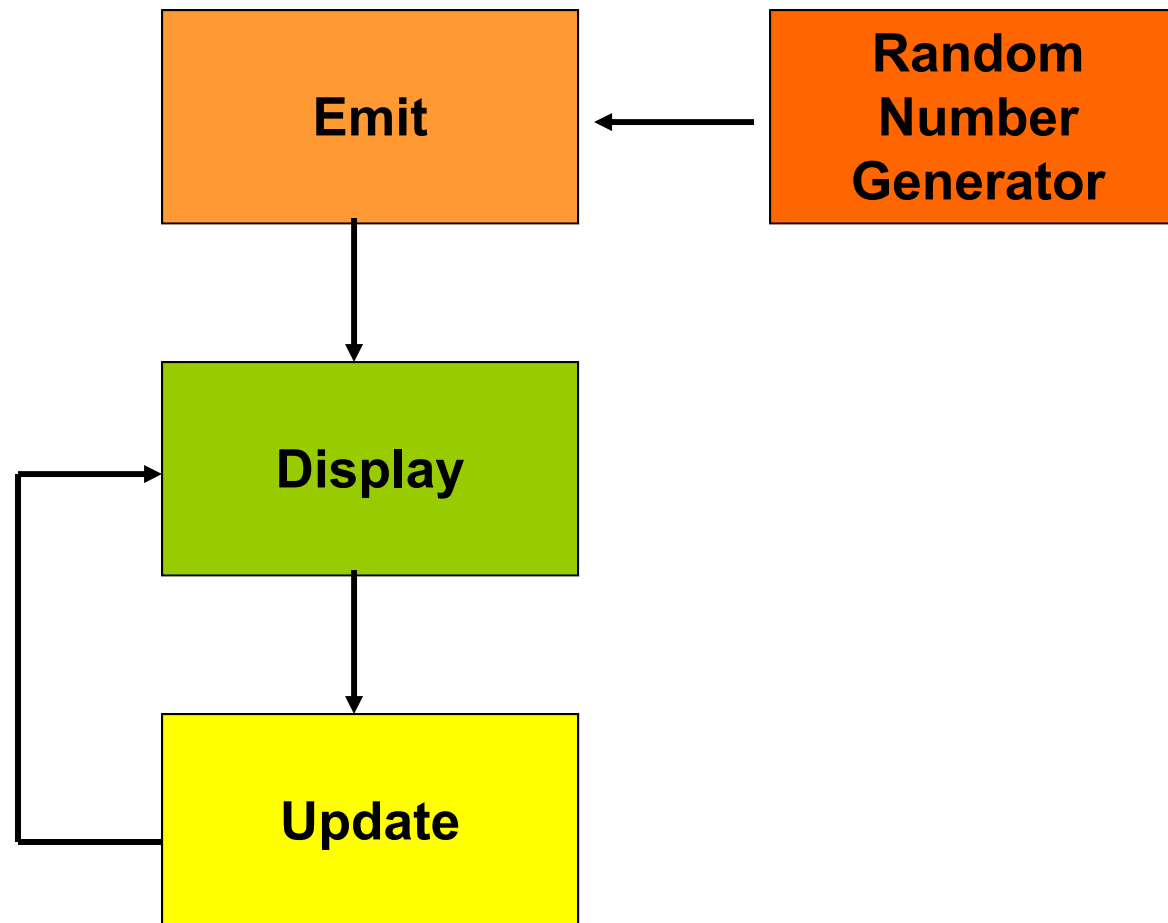


# Particle Systems: A Cross Between Modeling and Animation?



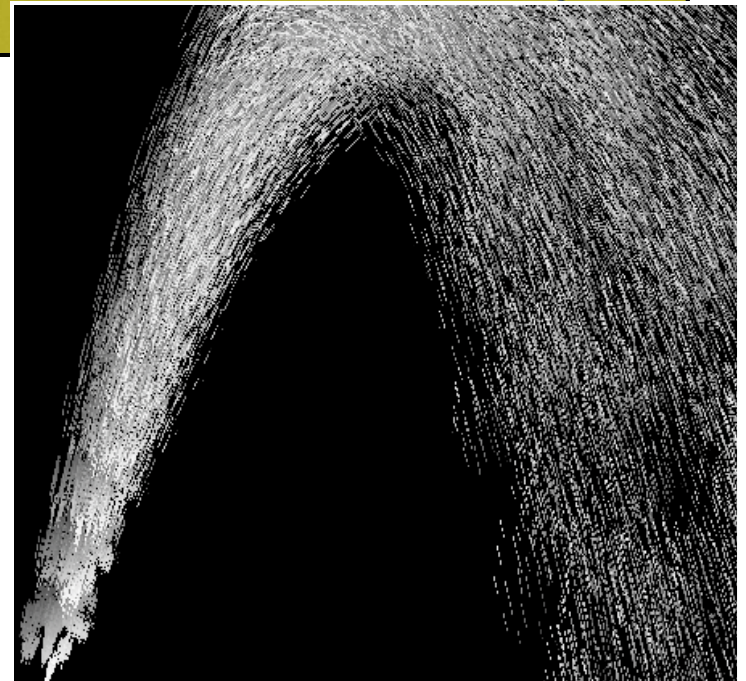
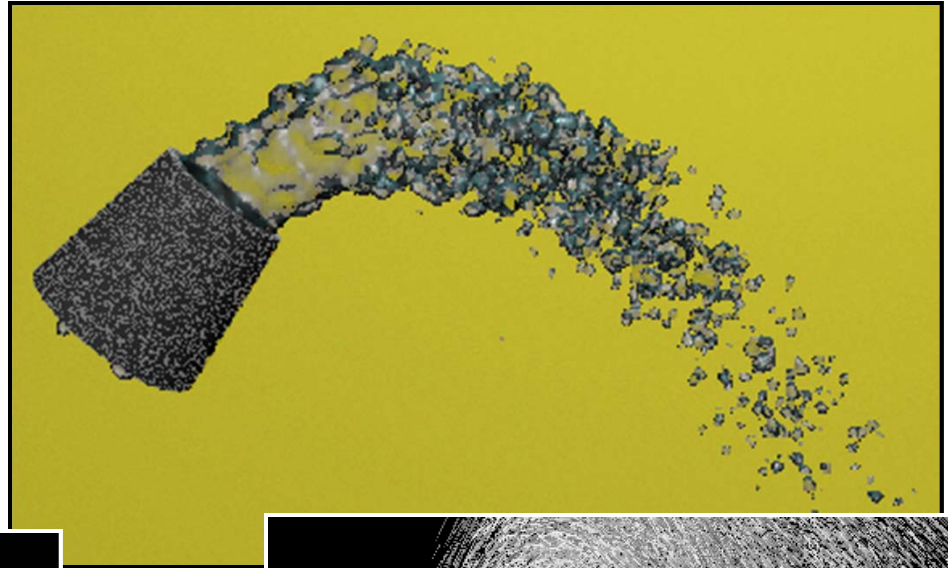
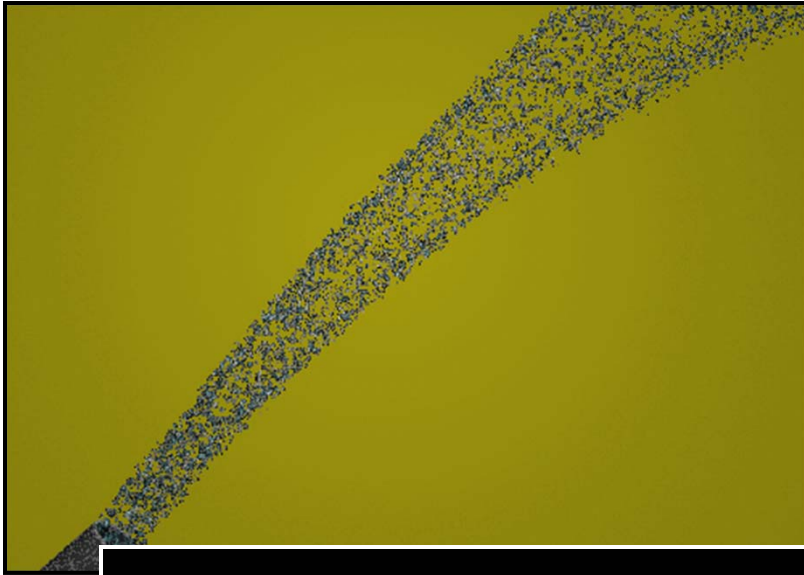
# Particle Systems: A Cross Between Modeling and Animation?

The basic process is:

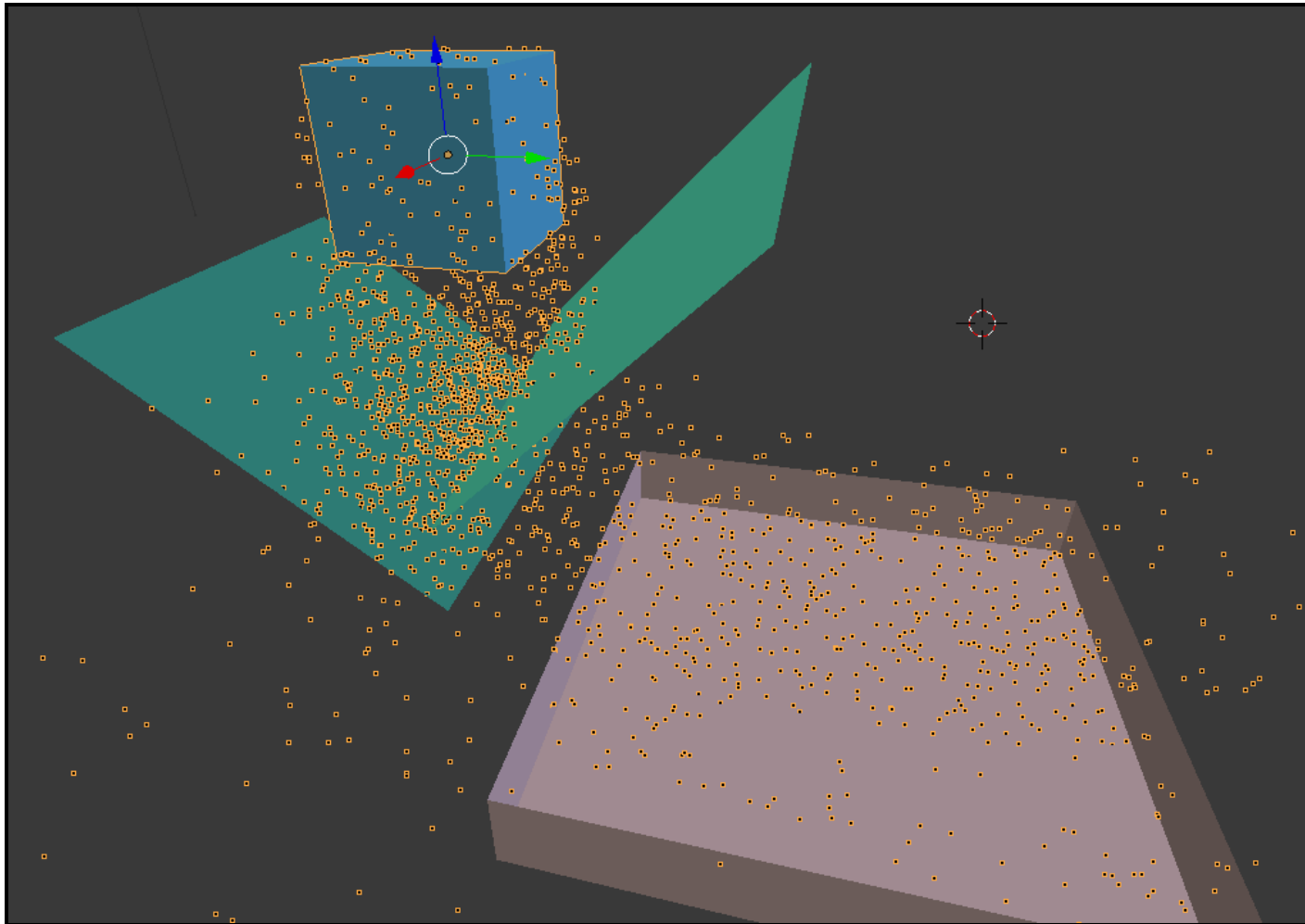




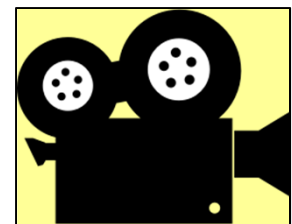
# Particle Systems Examples



# Particle Systems Examples

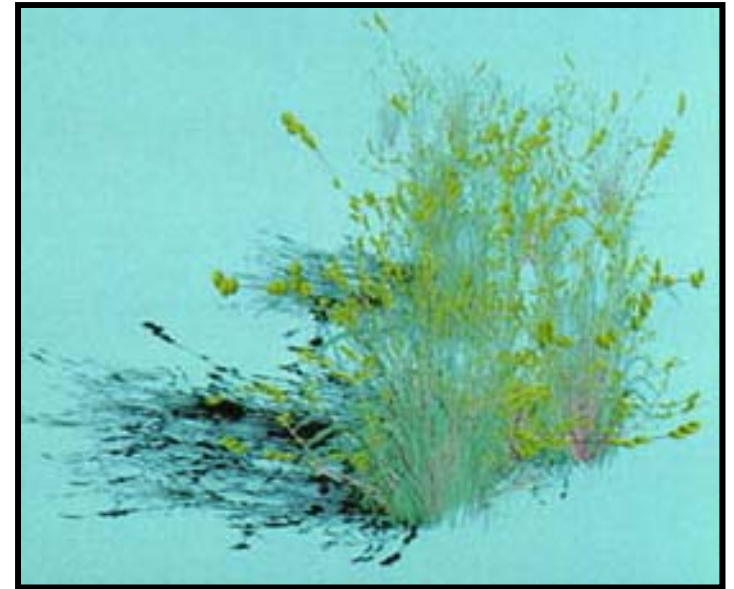


Oregon State  
University  
Computer Graphics



particles.mp4

# Particle Systems Examples



# A Particle System to Simulate Colliding Galaxies in *Cosmic Voyage*



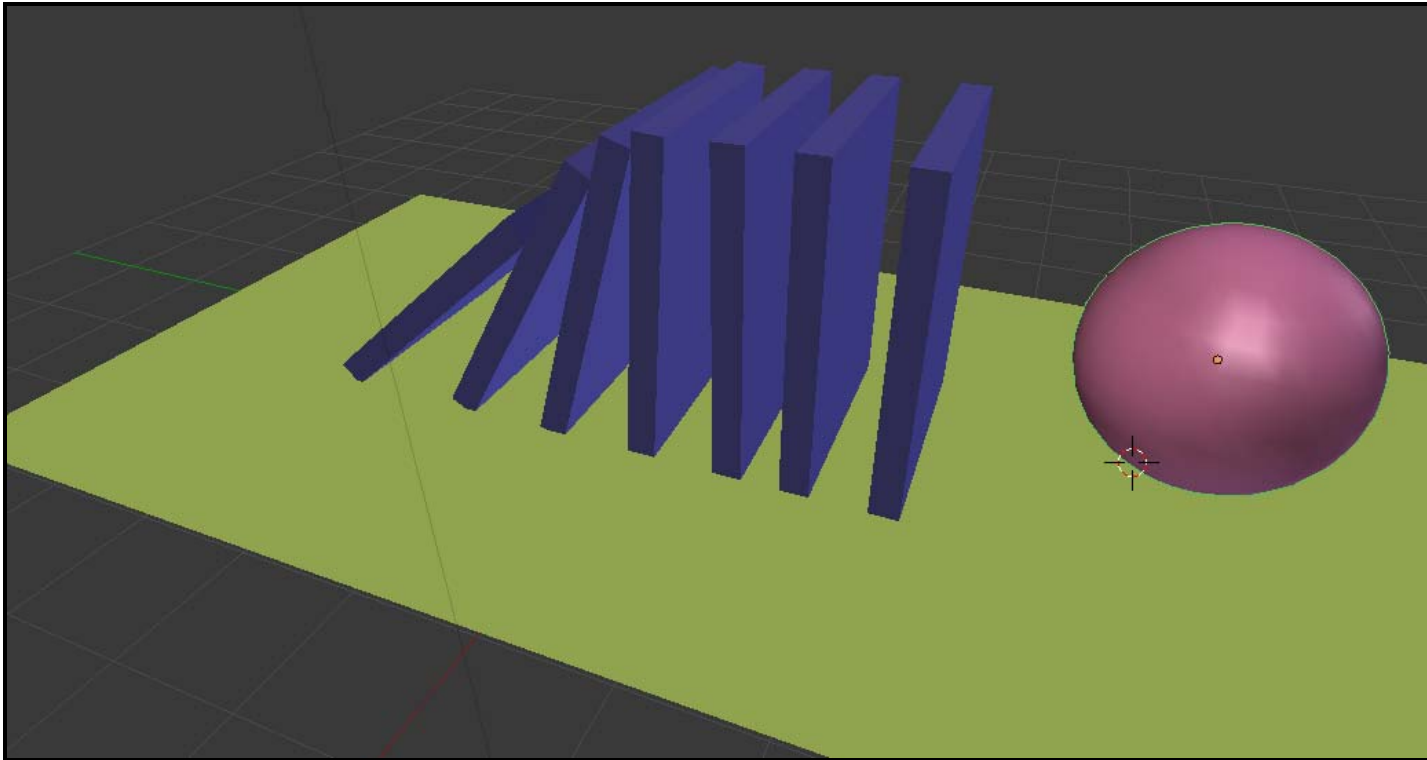


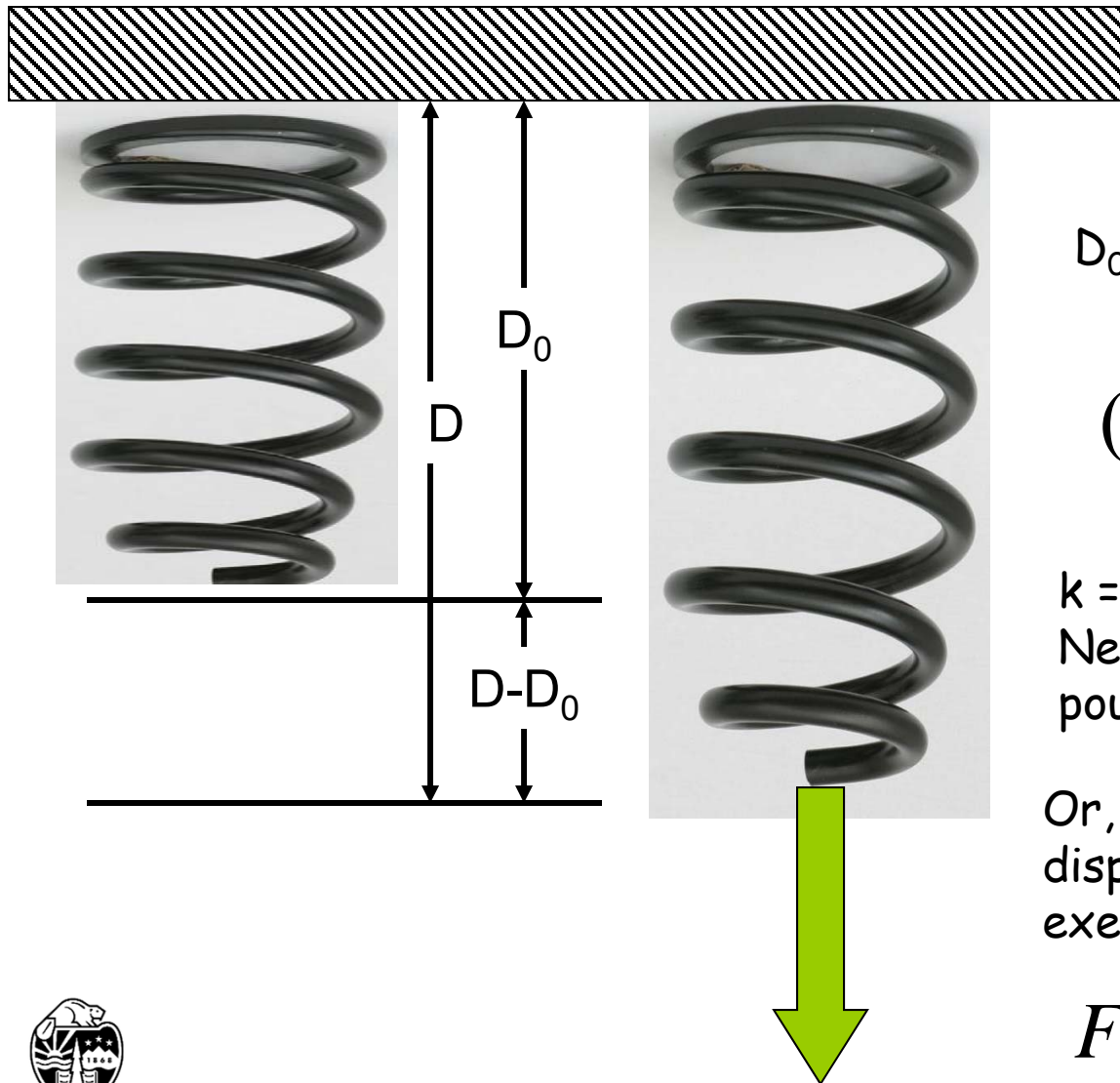
# Particles Don't Actually Have to Be “Particles”



# Animating using Physics

14





$D_0$  = unloaded spring length

$$(D - D_0) = \frac{F}{k}$$

$k$  = **spring stiffness** in  
Newtons/meter or  
pounds/inch

Or, if you know the  
displacement, the force  
exerted by the spring is:

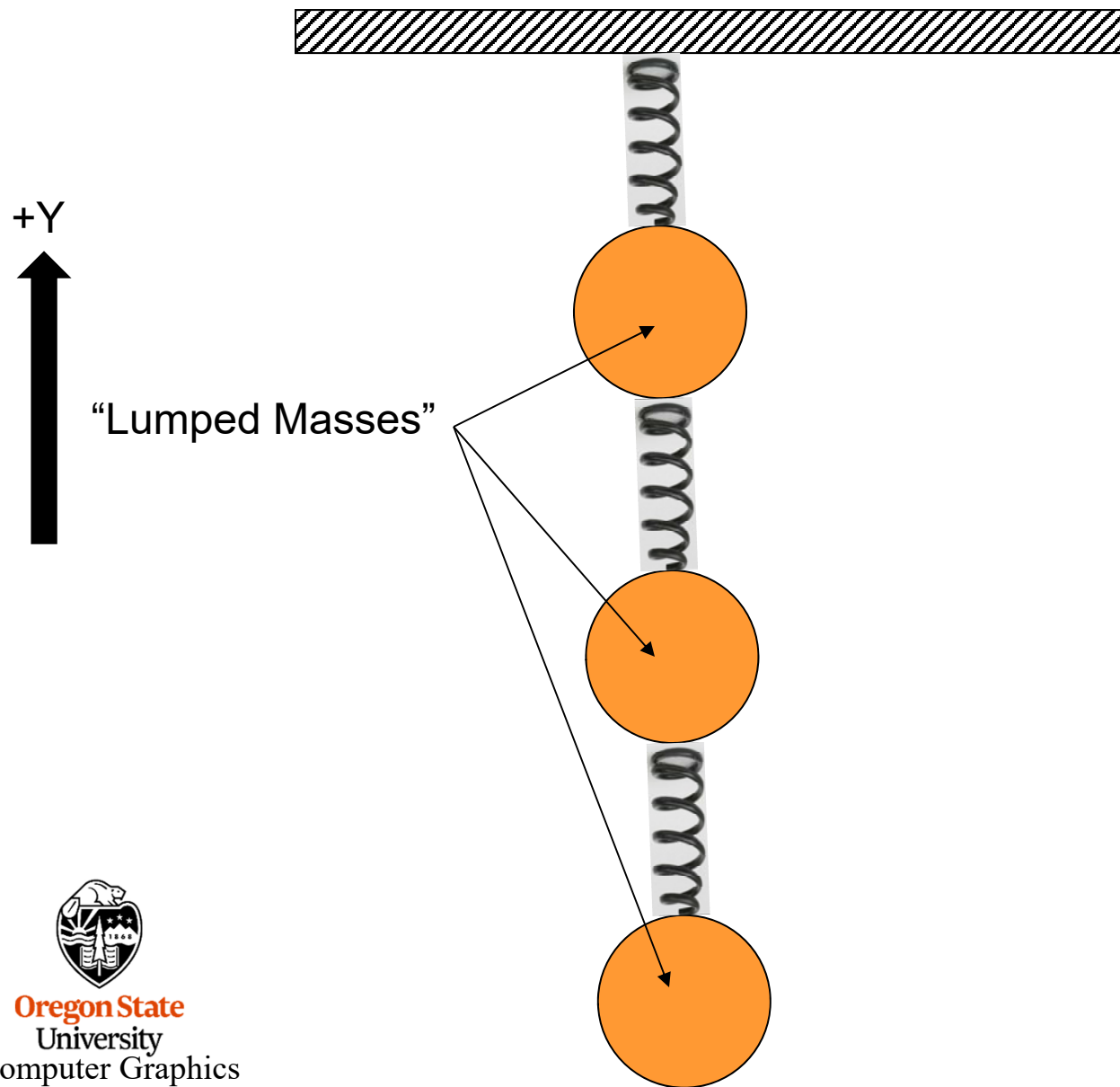
$$F = k(D - D_0)$$

Force =  $F$

This is known as Hooke's law

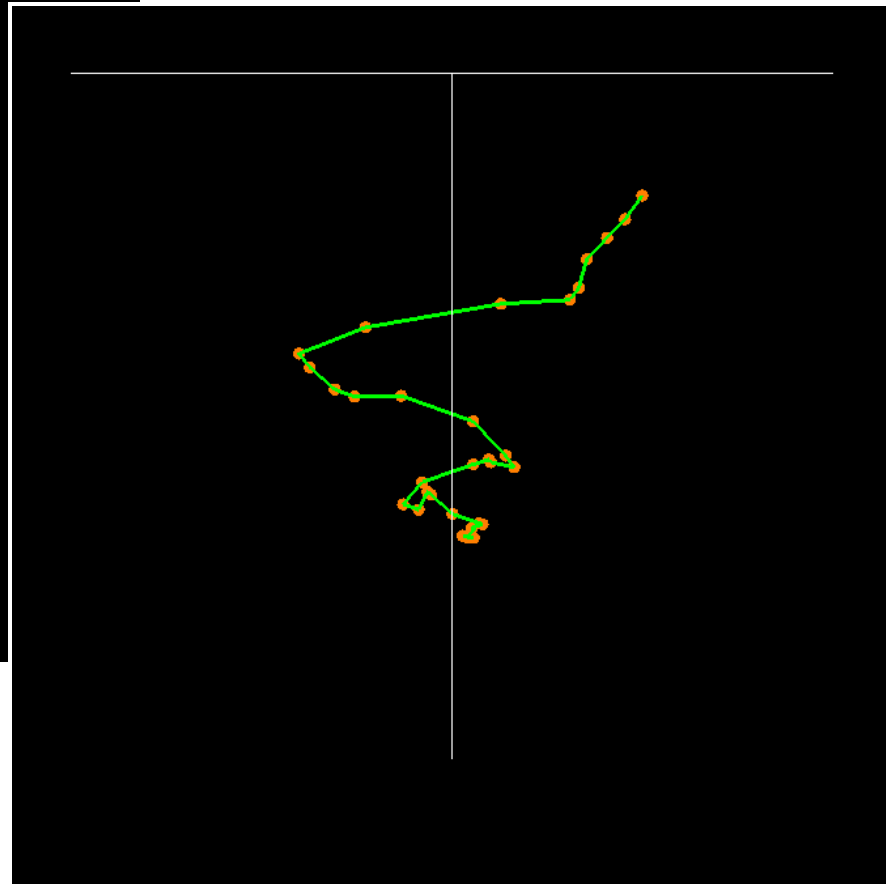
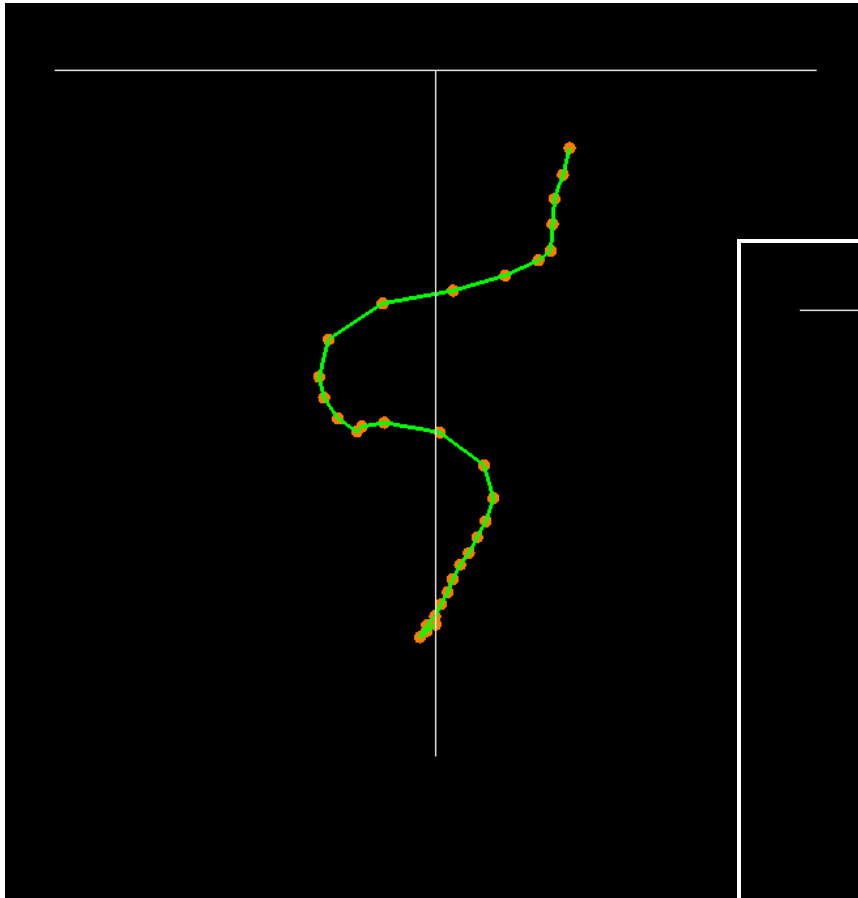
# Animating using the Physics of a Mesh of Springs

16

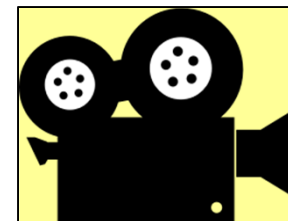
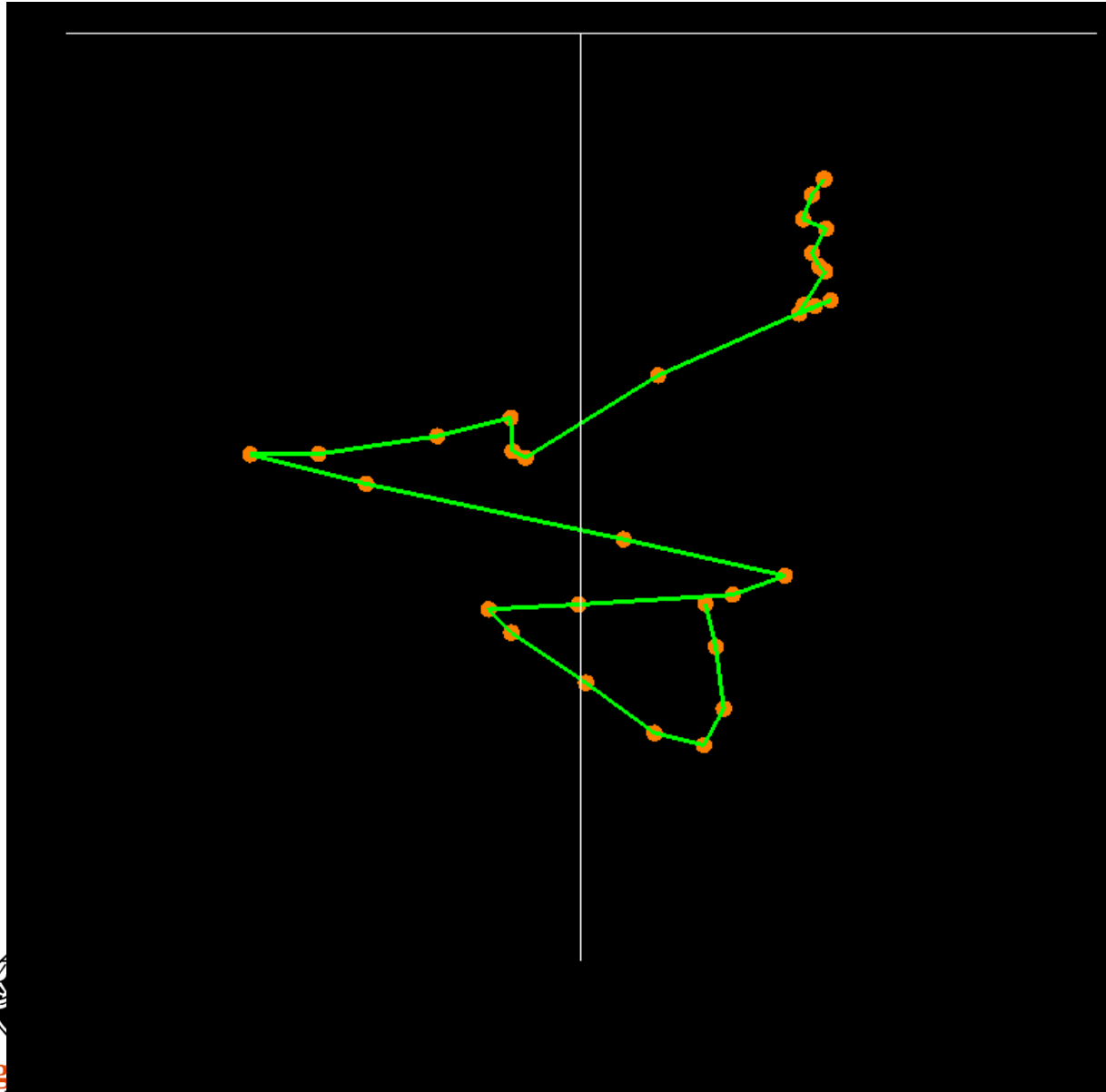




# Simulating a Bouncy String

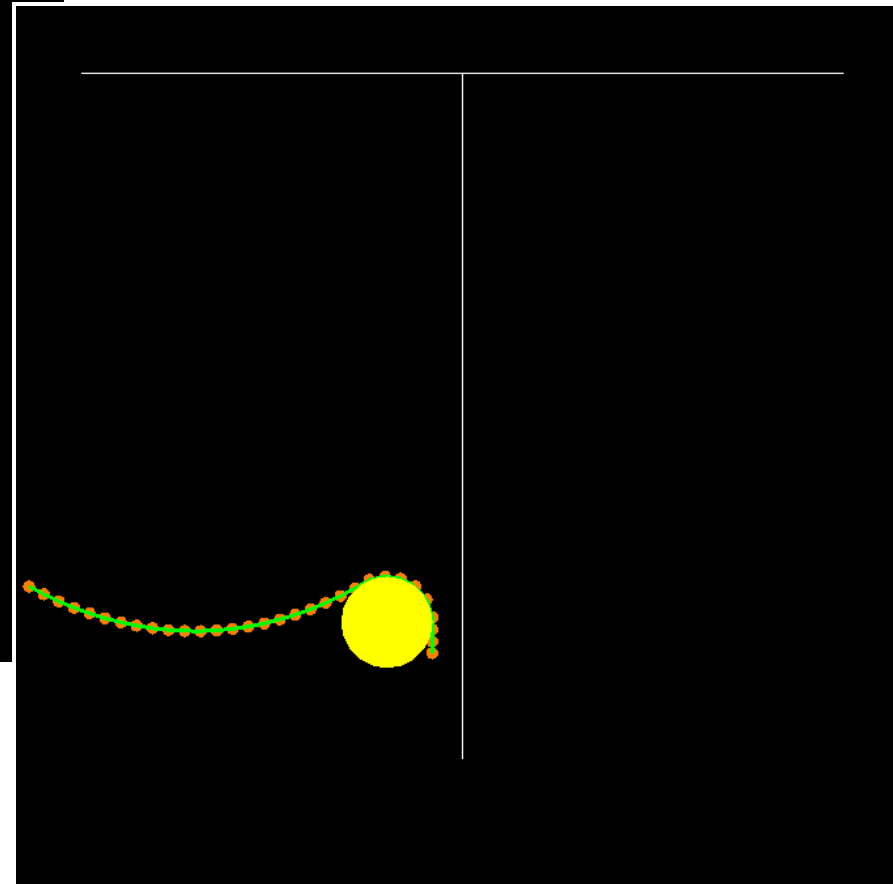
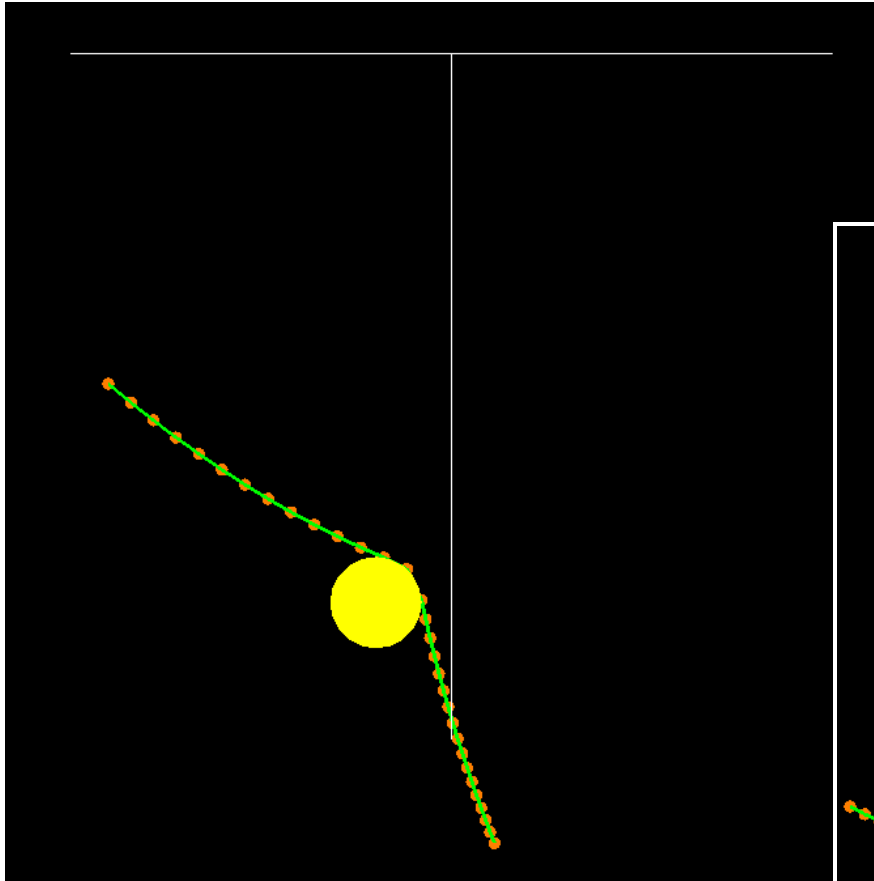


# Simulating a Bouncy String

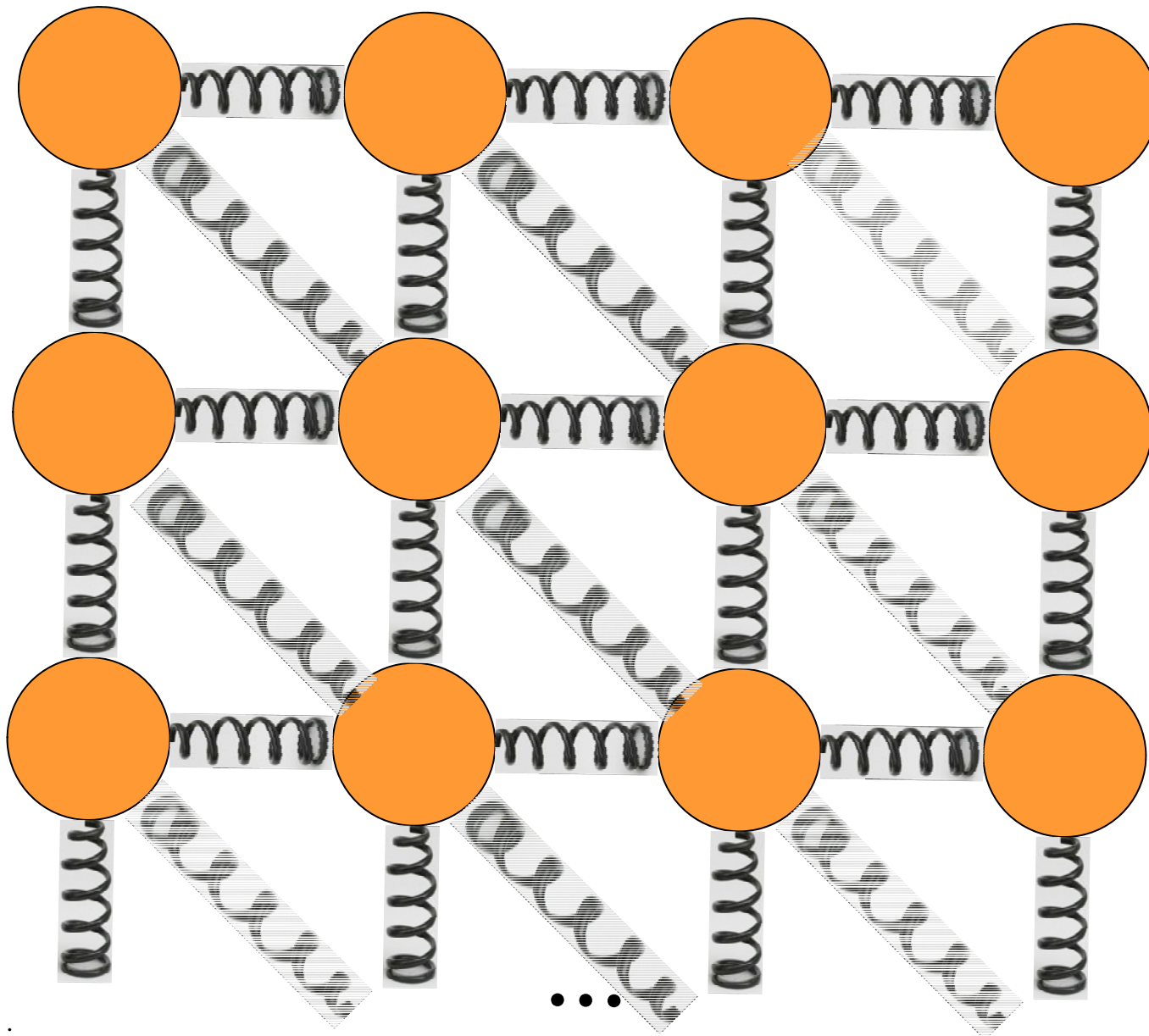


string.mp4

# Placing a Physical Barrier in the Scene

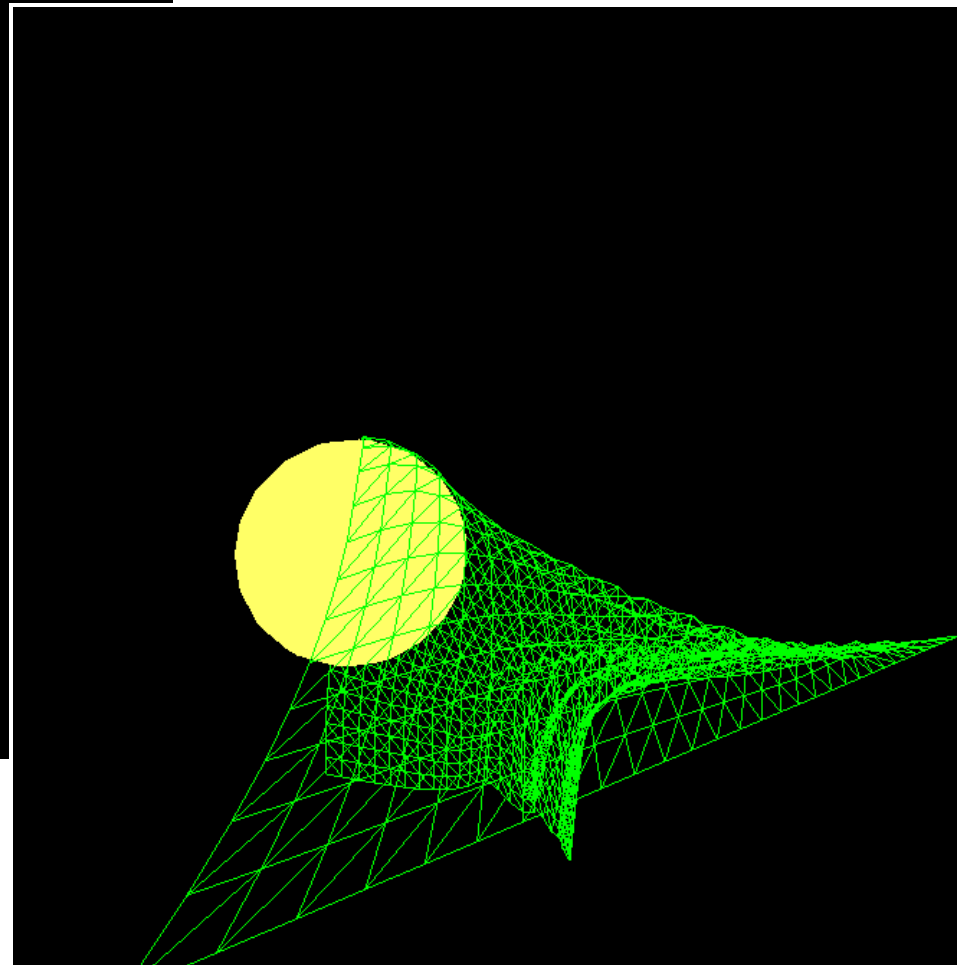
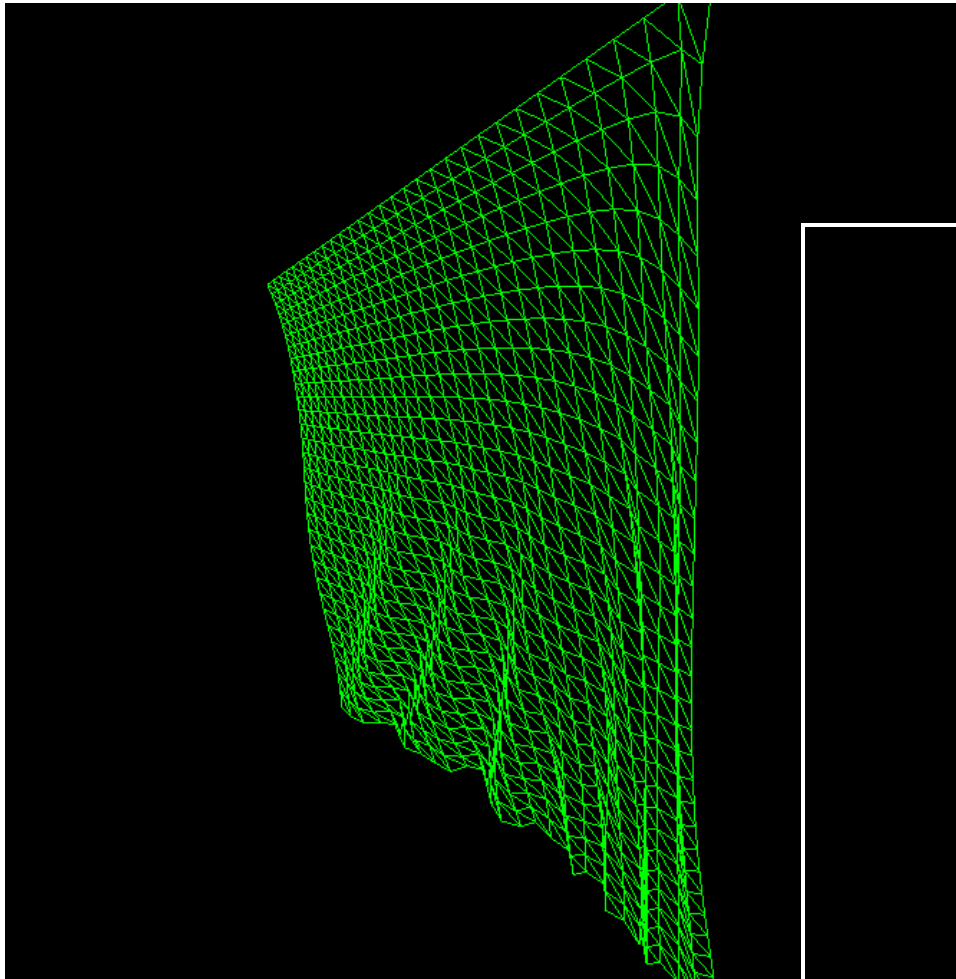


# Animating Cloth

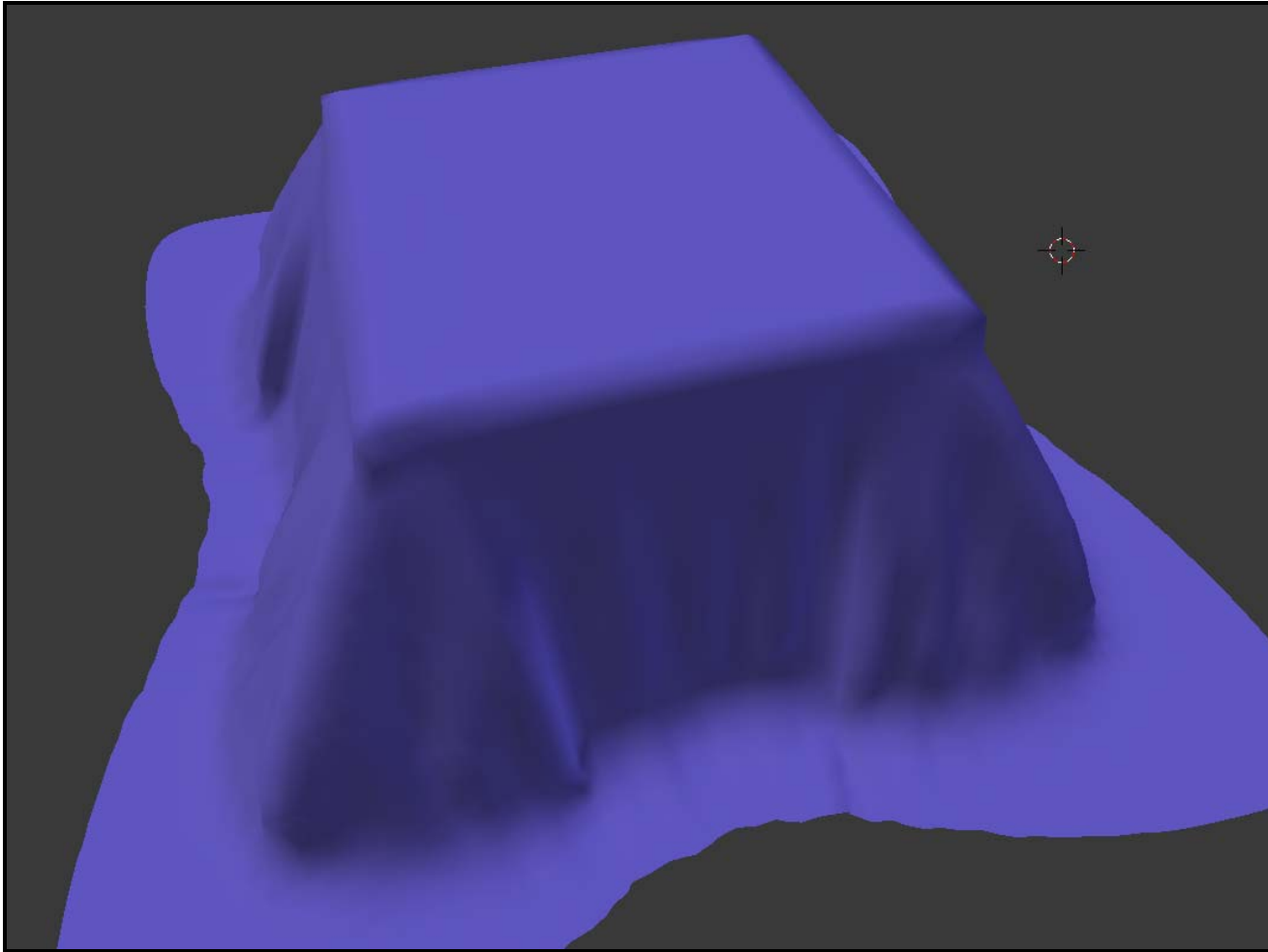




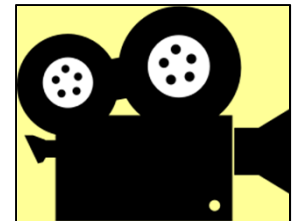
# Cloth Examples



# Cloth Example



**Oregon State**  
University  
Computer Graphics



cloth.mp4

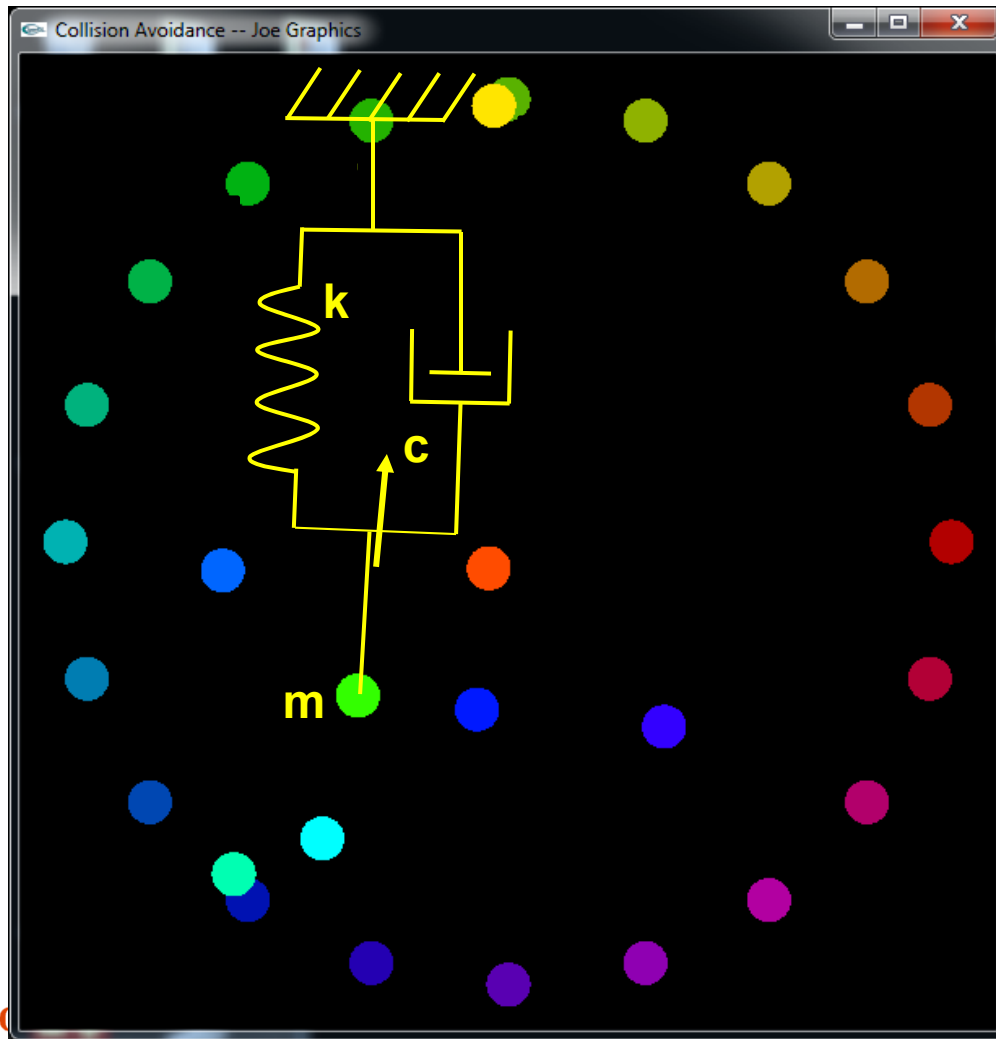
## Cloth Example



Pixar

# Functional Animation: Make the Object Want to Move Towards a Goal Position

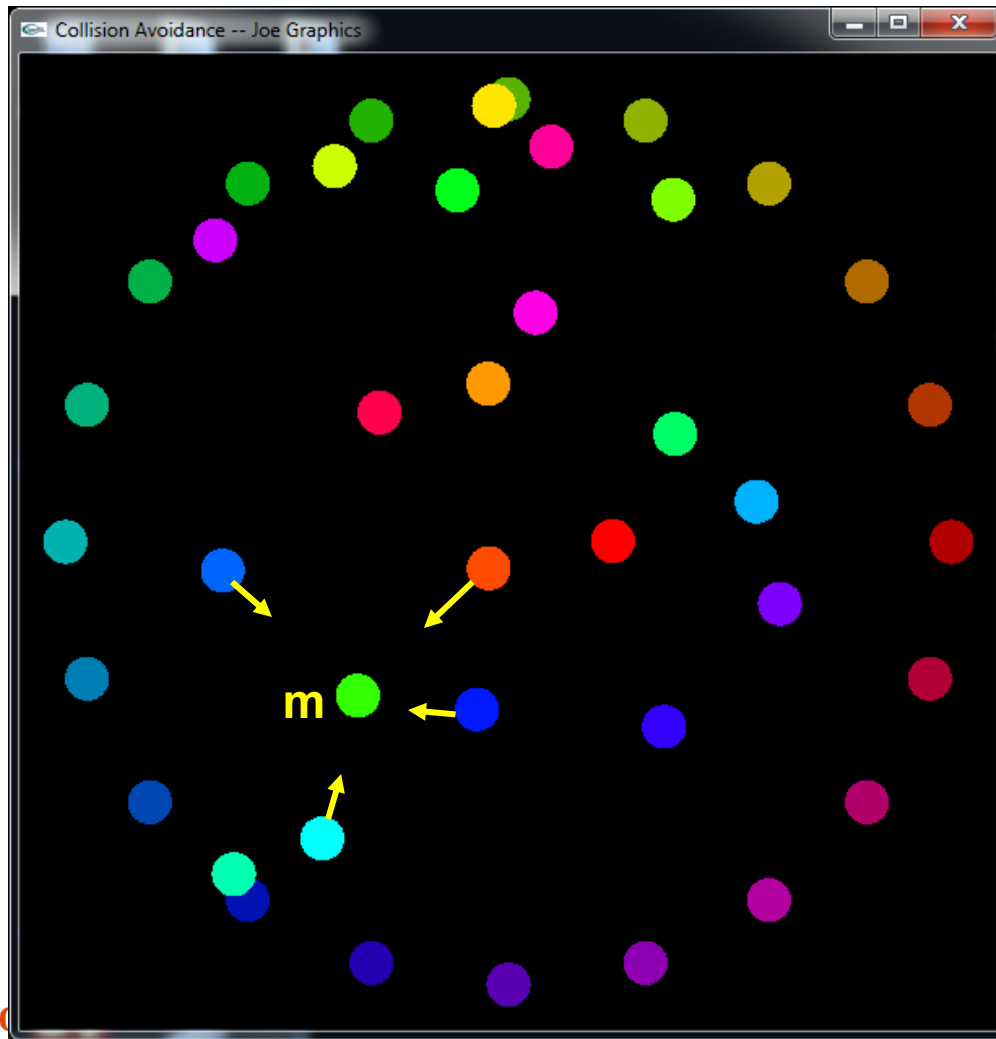
24



$$m\ddot{x} + c\dot{x} + kx = 0$$



# Functional Animation: While Making it Want to Move Away from all other Objects



$$m\ddot{x} = \sum F_{repulsive}$$

$$F_{repulsive} = \frac{C_{repulse}}{d^{Power}}$$

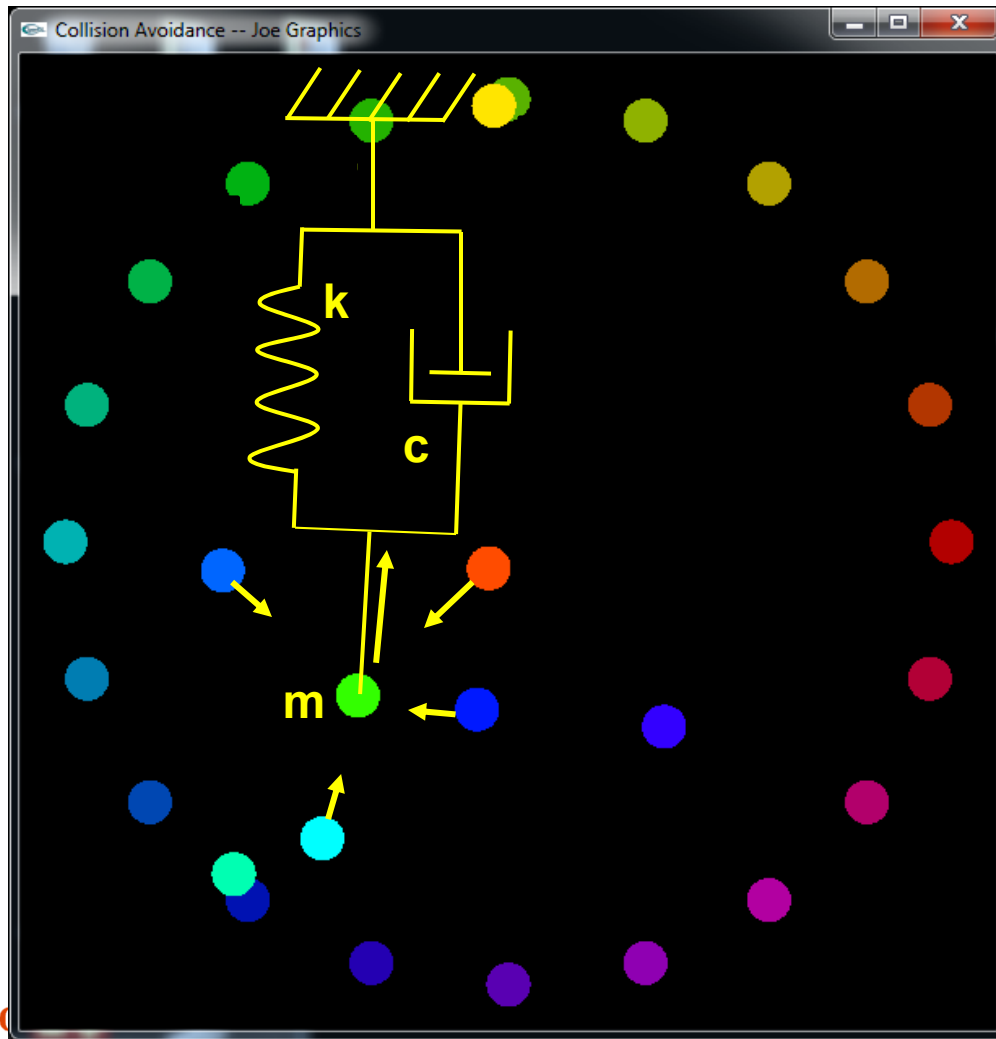
Repulsion Coefficient

Distance between the boundaries of the 2 bodies

Repulsion Exponent

# Total Goal – Make the Free Body Move Towards its Final Position While Being Repelled by the Other Bodies

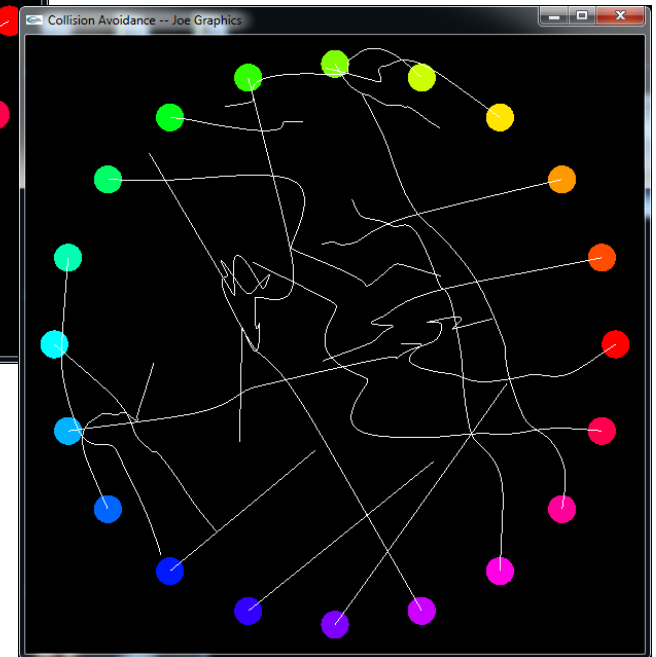
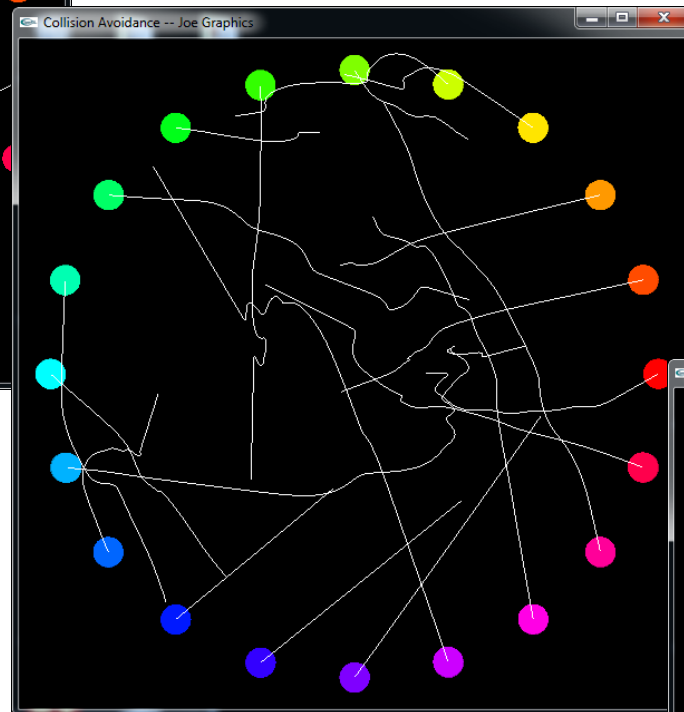
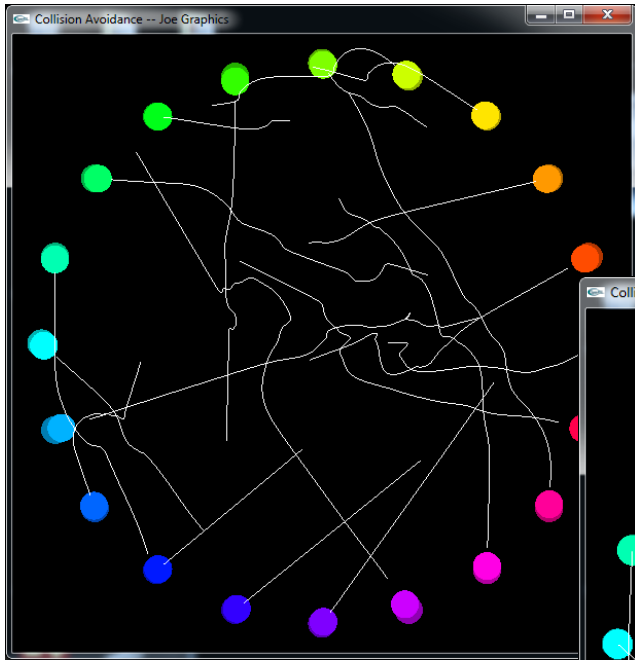
26



$$m\ddot{x} + c\dot{x} + kx = \sum F$$

# Increasing the Stiffness

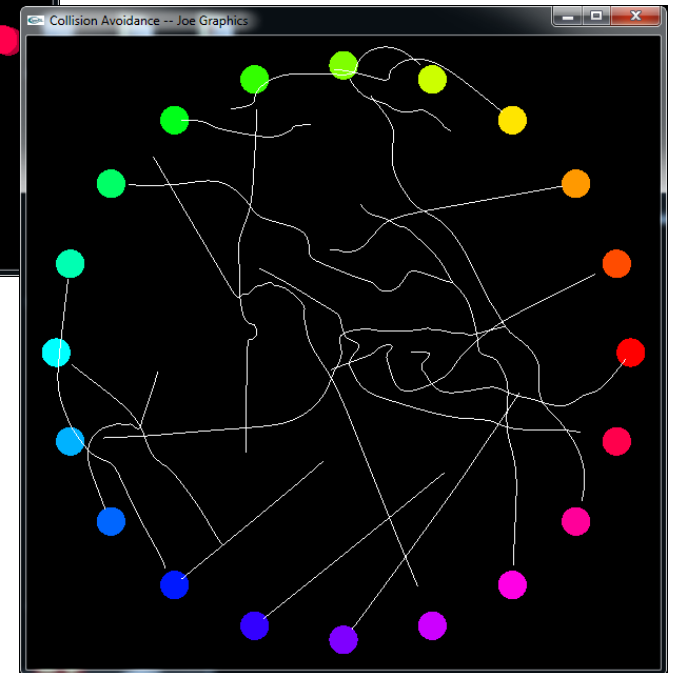
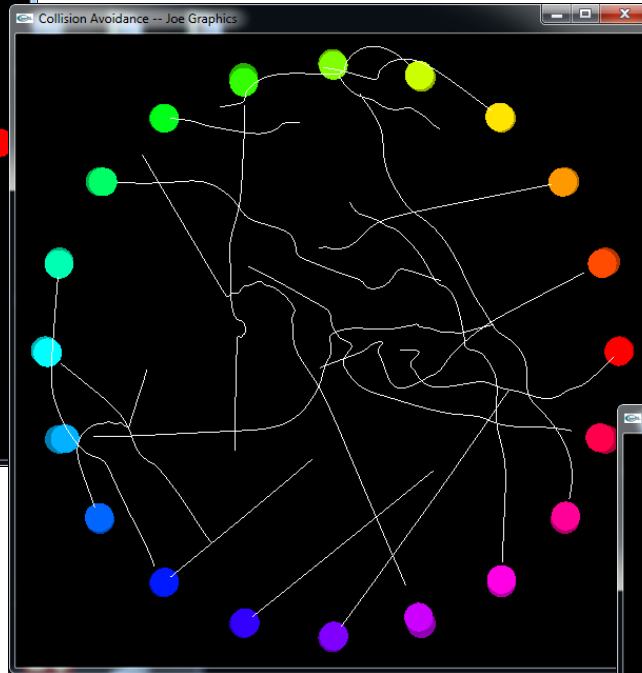
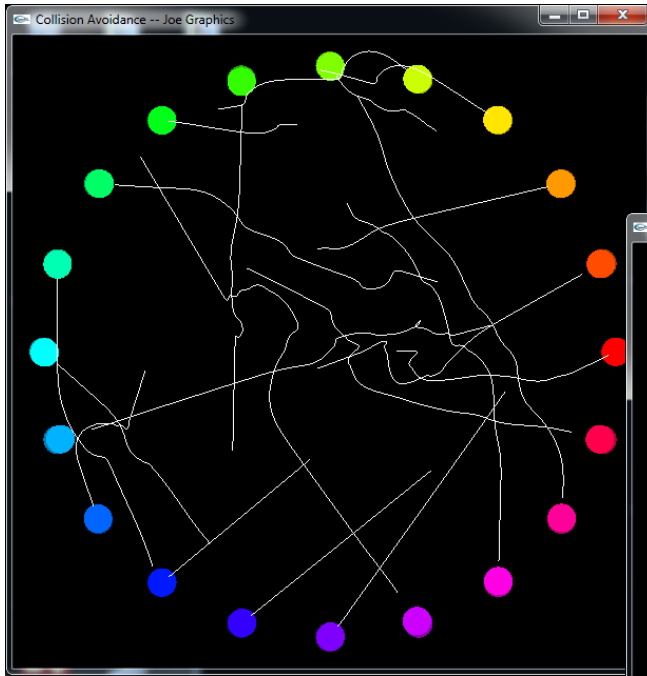
27



Stiffness = 3, 6, 9

# Increasing the Repulsion Coefficient

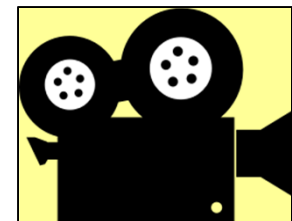
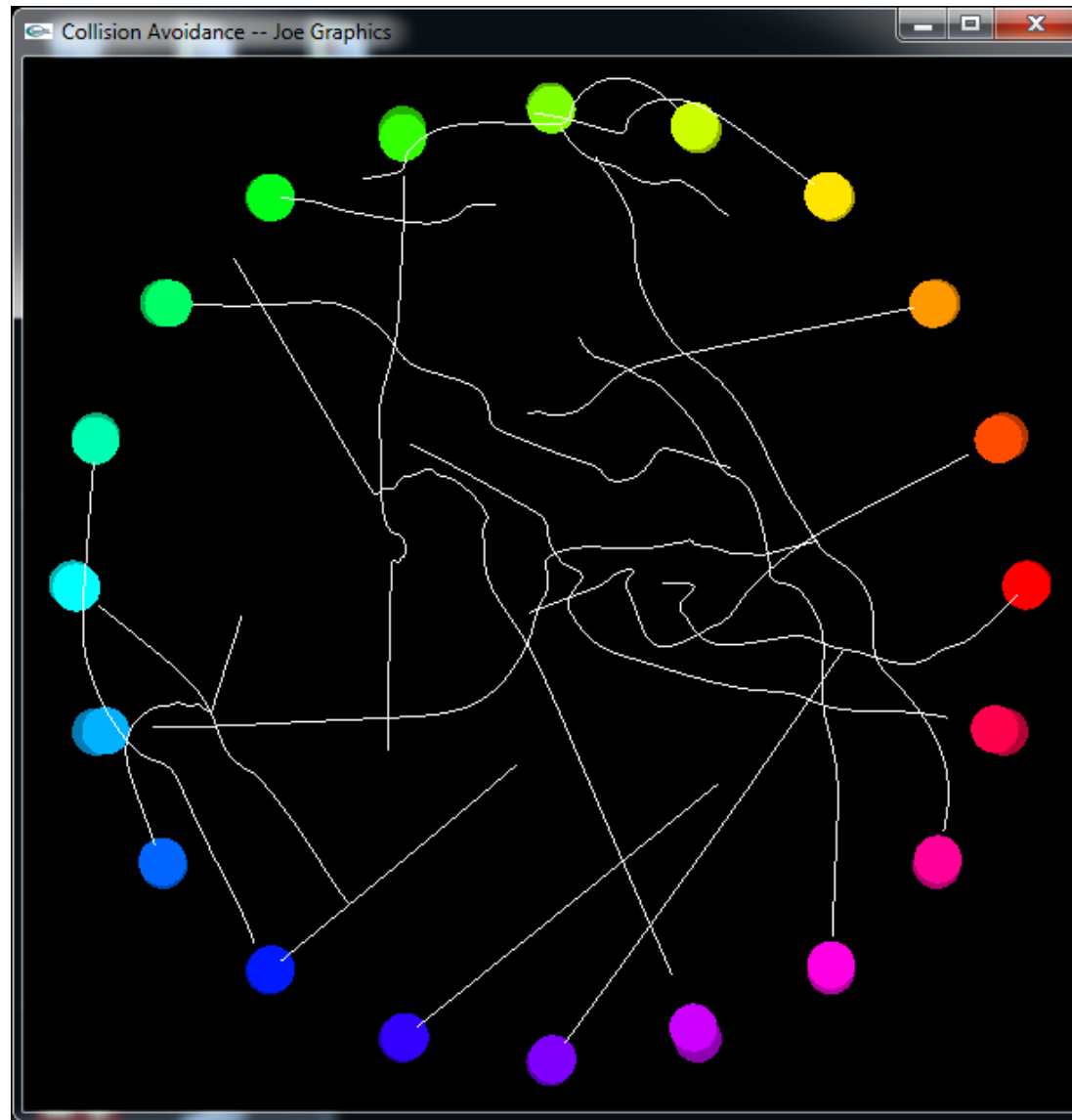
28



Repulse = 10, 30, 50

# Functional Animation

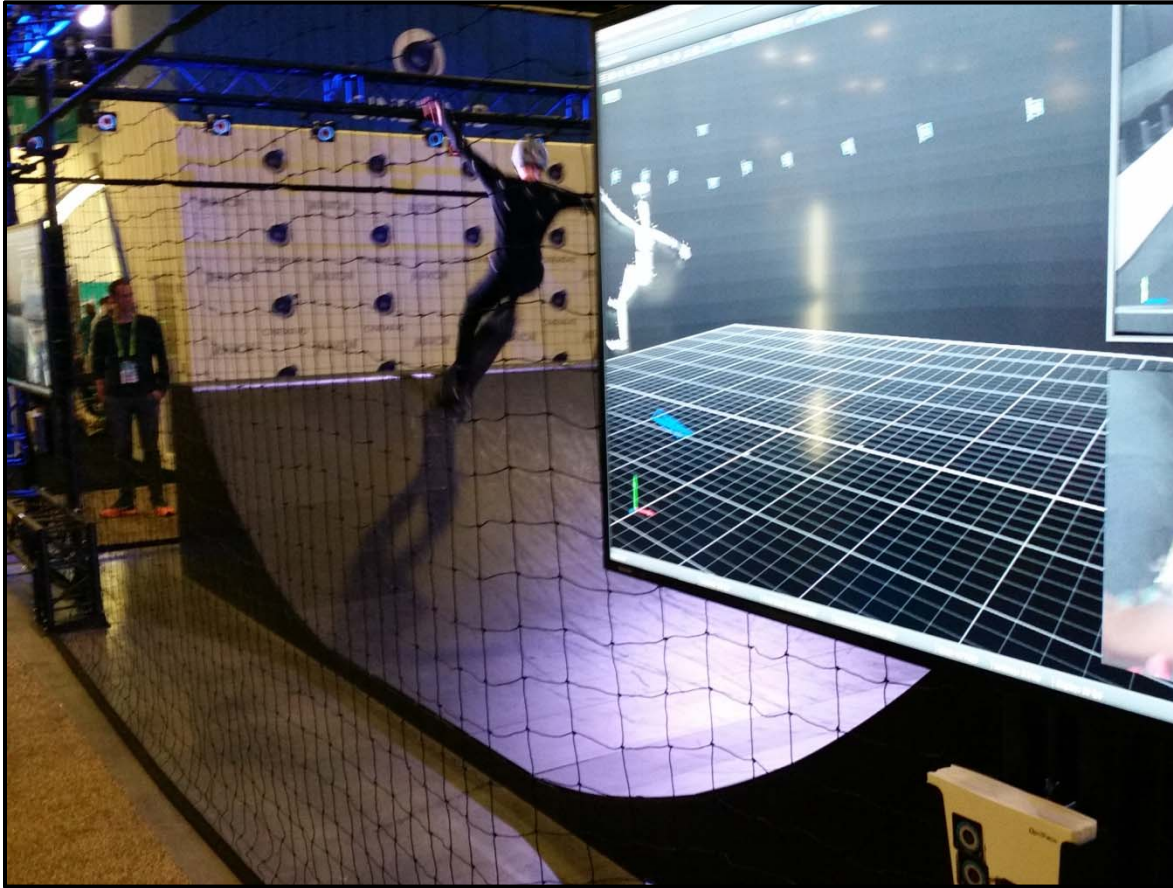
29



avoid.mp4



# Motion Capture as an Input for Animation



## Motion Capture is for Faces Too



# Tron I – Probably should have used physics, but didn't



## Card Trick



## Pixar Animated Shorts

