Lecture 21
Animation

Computer Animation: Overview

1) Scripting
2) Keyframing

3) Kinematics
4) Motion capture/processing

5) Higher level animation [“Procedural”]
6) Dynamics and simulation

Example of Scripting

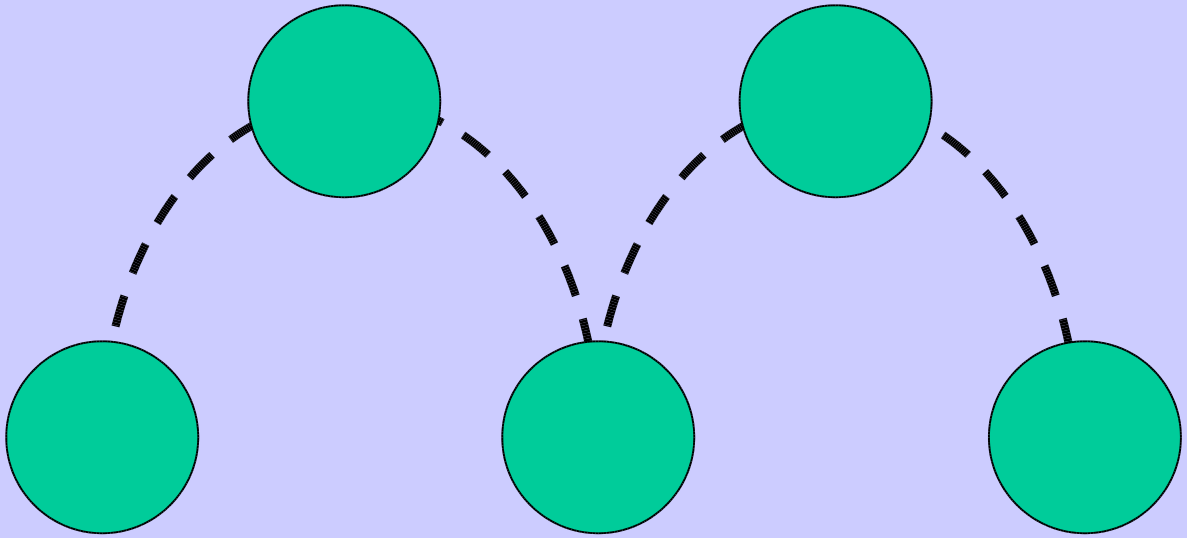
Specifying the parameters at every frame

```
define spinningCube()
    rotAngle = pi*frameNumber / 50

define carScript()
    carTranslation = 10*(frameNumber / 100)
    wheelRotation  = pi*frameNumber / 5
```

Keyframing

Specify only the important frames,
interpolate the frames in-between

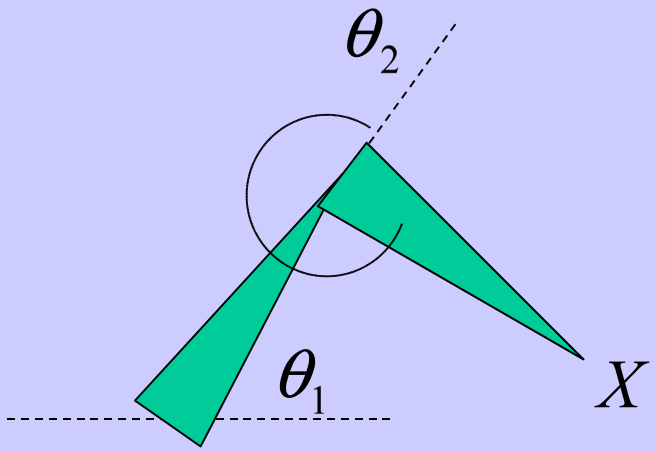


What and how to interpolate is important

Forward Kinematics

Given the character's state,
calculate its pose

$$X = f(\theta)$$

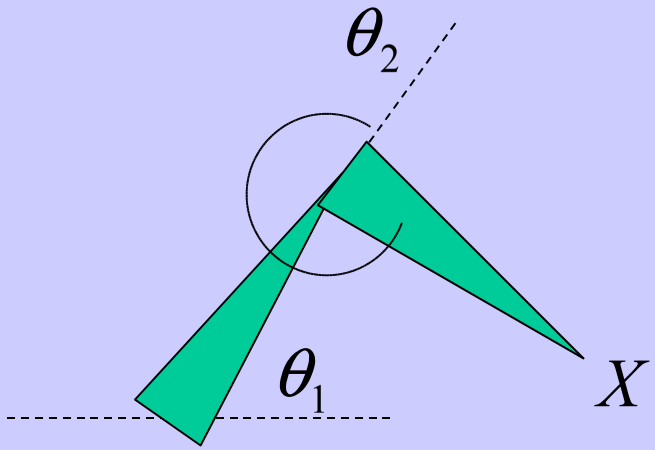


$$X = \begin{bmatrix} l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

Inverse Kinematics

Given the character's pose,
calculate its state

$$\theta = f^{-1}(X)$$



$$\theta = \begin{bmatrix} \dfrac{-(l_2 \sin\theta_2)x + (l_1 + l_2\cos\theta_2)y}{(l_2 \sin\theta_2)y + (l_1 + l_2\cos\theta_2)x} \\ \cos^{-1}\dfrac{(x^2 + y^2 - l_1^2 - l_2^2)}{2l_1 l_2} \end{bmatrix}$$

Motion processing (cf. “roto”)

Creepy, wax museum doll quality!

Routinely used in video games
Less frequently in music videos (eg. Herbie Hancock – Dis is da Drum)
Even less so in movies – Final Fantasy, Matrix, Polar Express..

Behavioral Animation

Animating by describing an actor's behavior

An actor's behavior defines how the actor interacts with other actors and the environment

```
TRex()
  if(player is close)
    eatPlayer()
  else if(can see player)
    chasePlayer()
  else
    wander()
```

Behavioral Animation

Useful for crowd animations

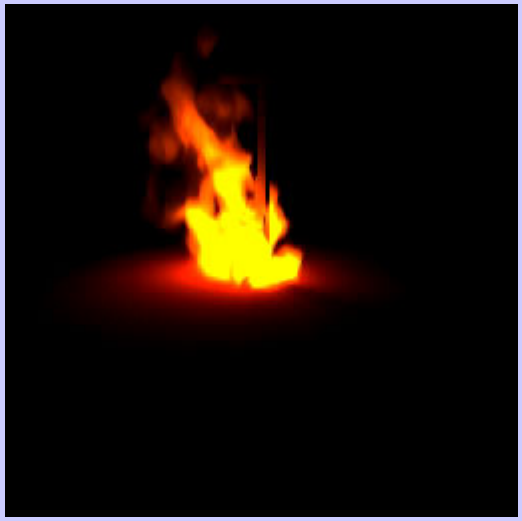
Dynamics – Particle Systems

Particle Systems [Reeves83]

Represent “fuzzy” objects (such as fire, smoke) as a collection of particles

Particles contain local state

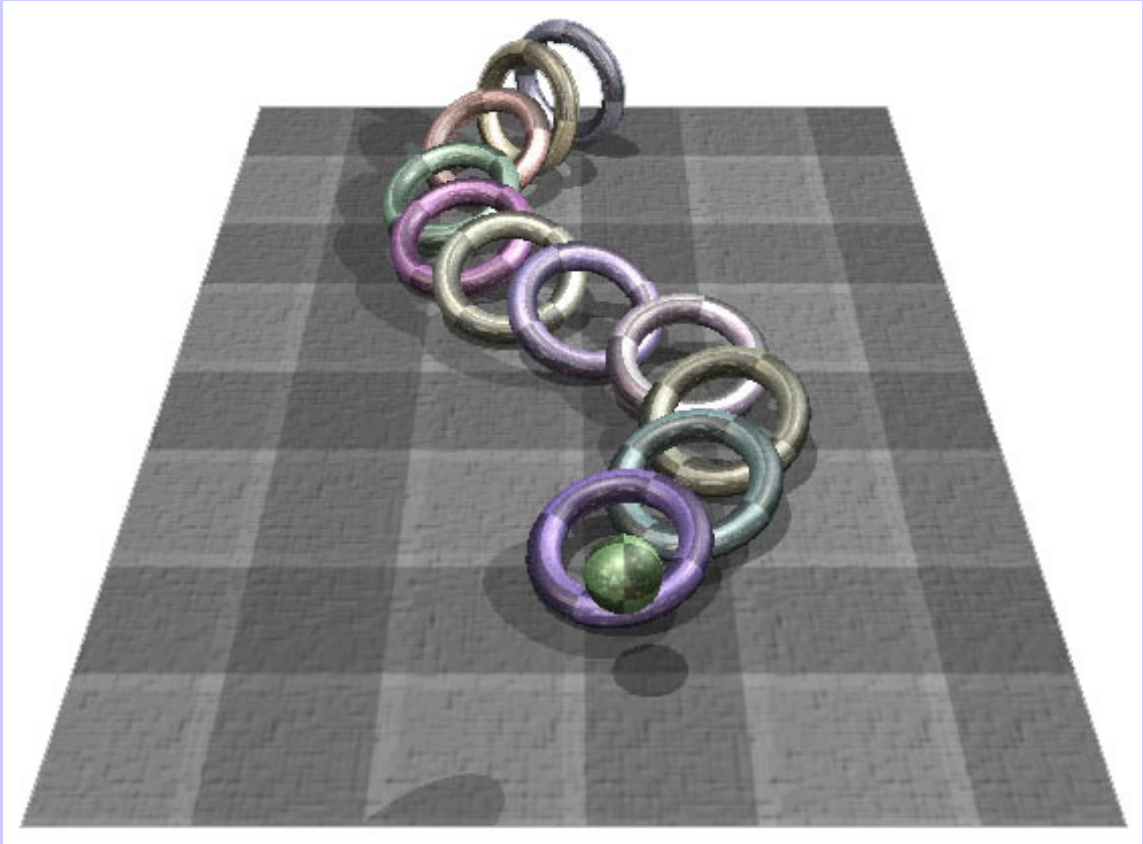
- Position
- Velocity
- Age
- Lifespan
- Rendering properties



Dynamics – Rigid Bodies

Rigid Bodies

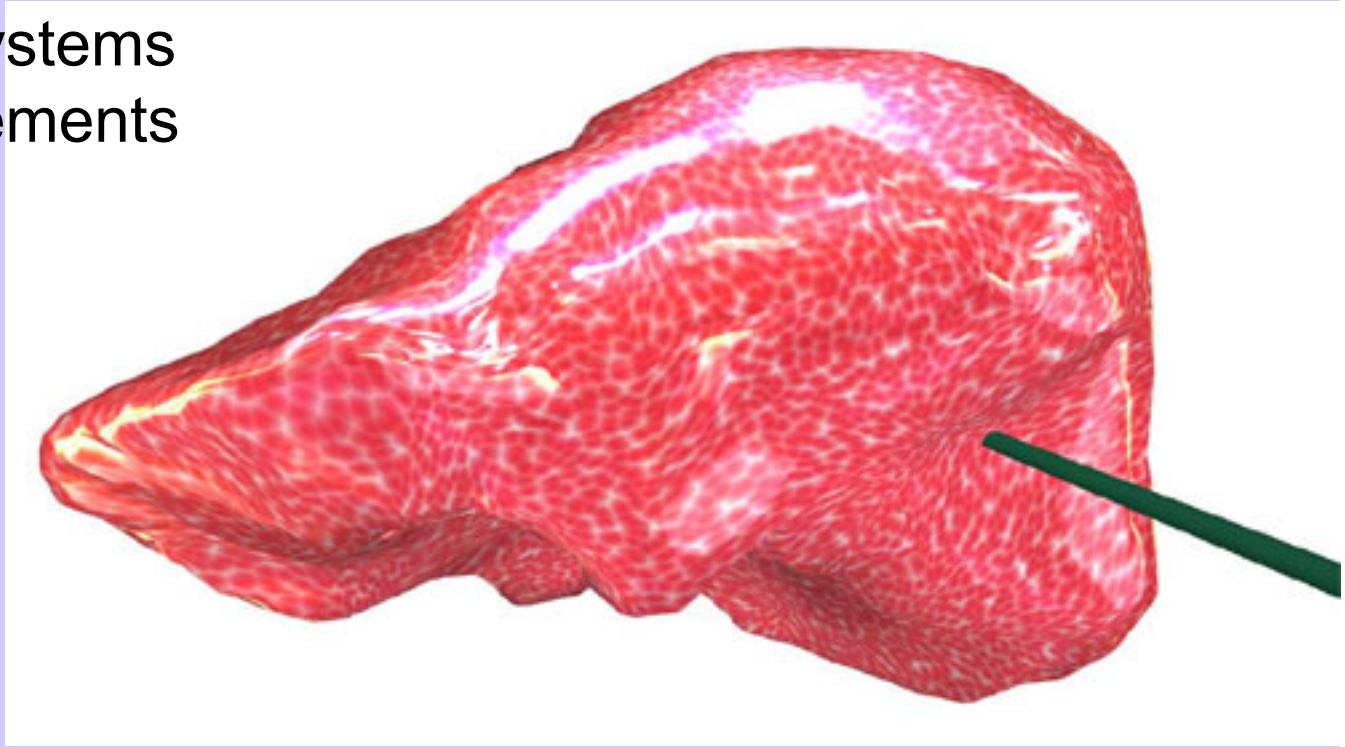
- Integration
- Collisions
- Constraints



Dynamics – Deformable Objects

Deformable Objects

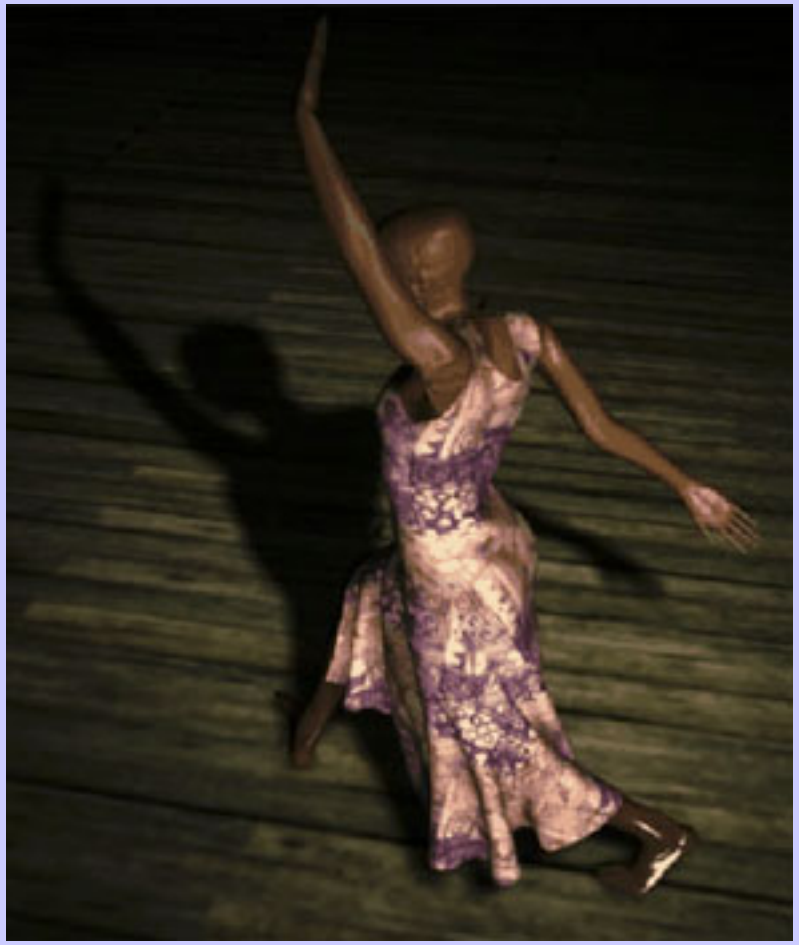
- FFD
- Spring systems
- Finite Elements



Dynamics – Cloth

Cloth Simulation

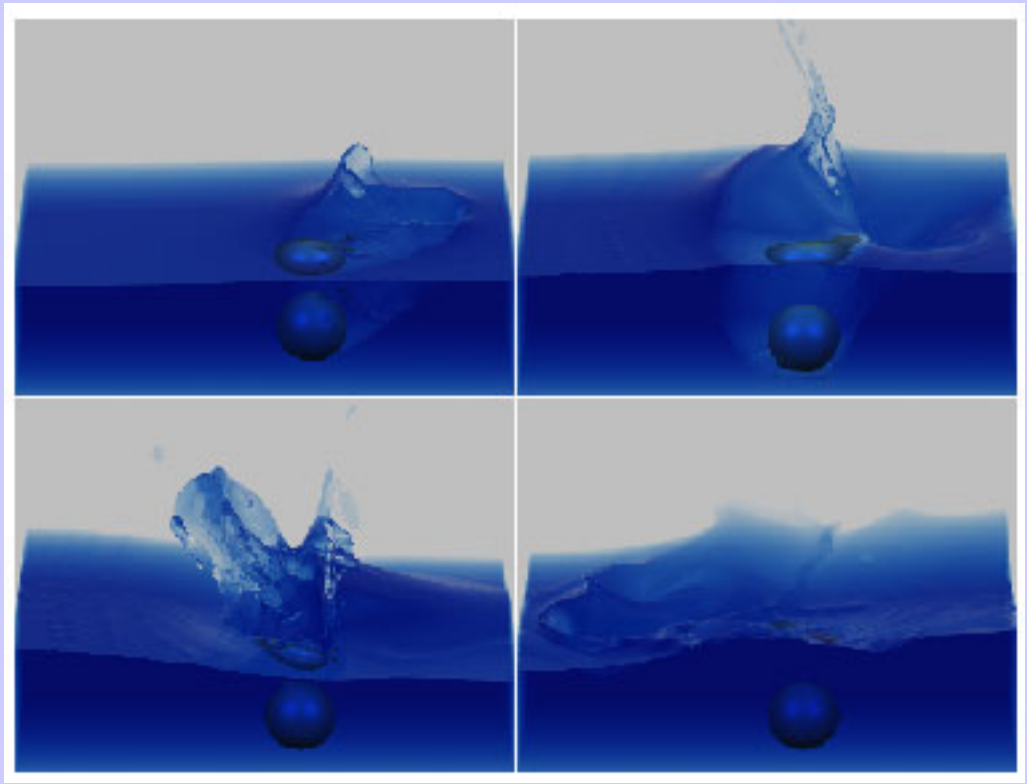
- Stable Integration
  - even with large time steps…
- Adaptivity
  - reduce amount of computations
- Material Properties



Dynamics – Fluids

Fluid Simulation

- Navier Stokes, plus *lots* of topology changes



Real Time Animation

Zelda (GameCube)

Offline Animation – Anything Goes

Final Fantasy

Pixar movies