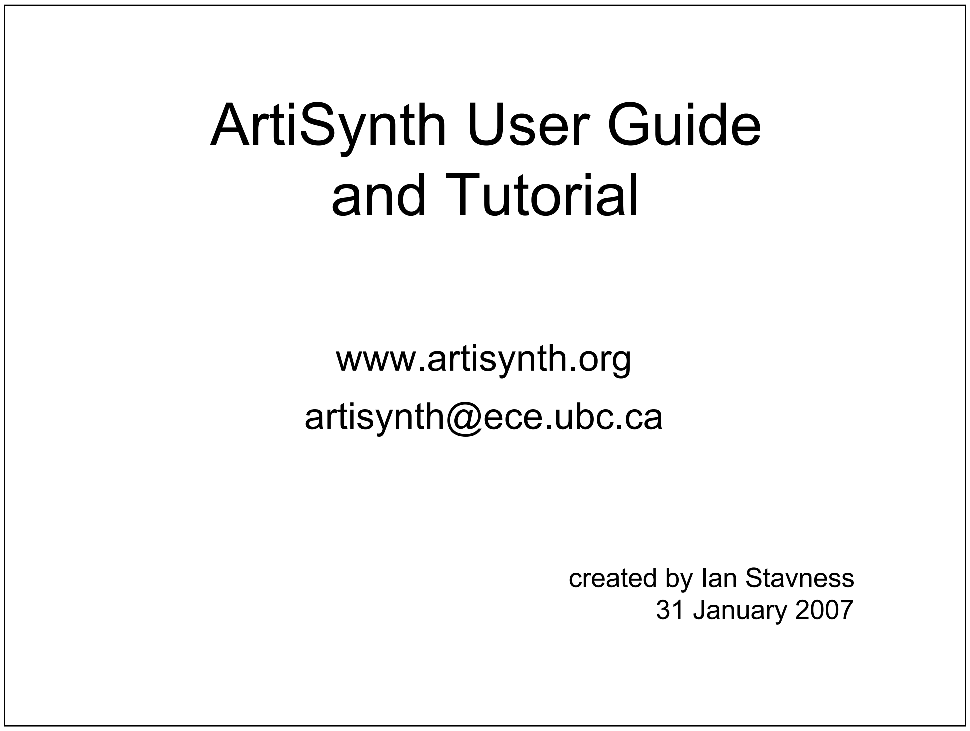


ArtiSynth User Guide and Tutorial

www.artisynth.org

artisynth@ece.ubc.ca

created by Ian Stavness
31 January 2007

Welcome to the ArtiSynth Tutorial. This “User Guide” is intended to provide a basic introduction to programming and modeling within ArtiSynth.

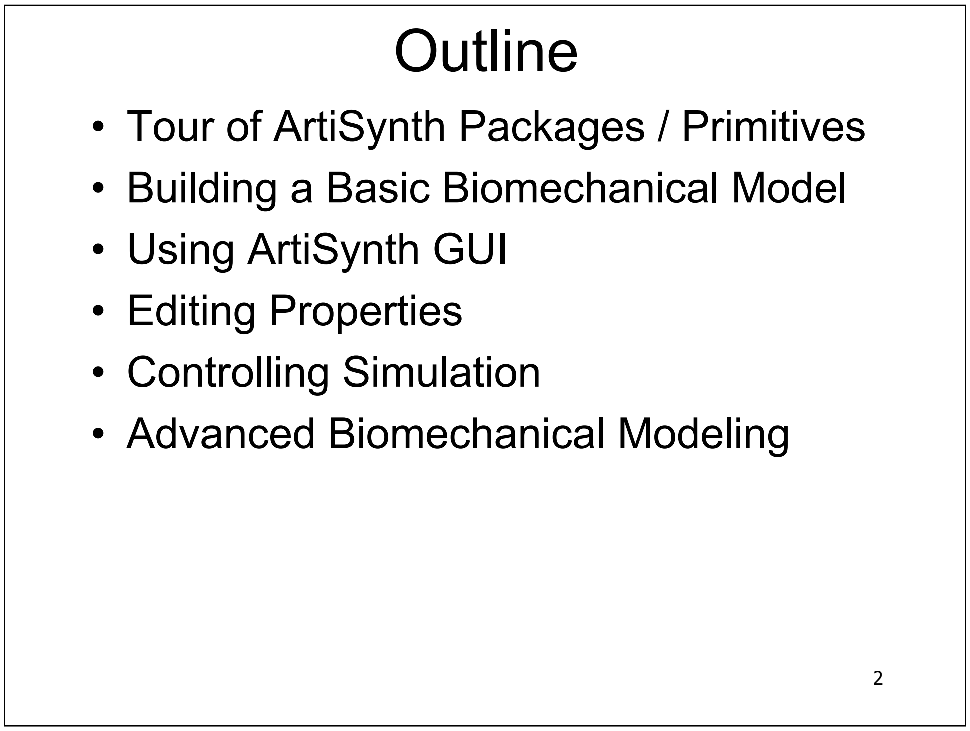
ArtiSynth source code is available on our website at:

www.artisynth.org/download

Please refer to Java classes for the demos presented in this tutorial, which can be found in the source code at:

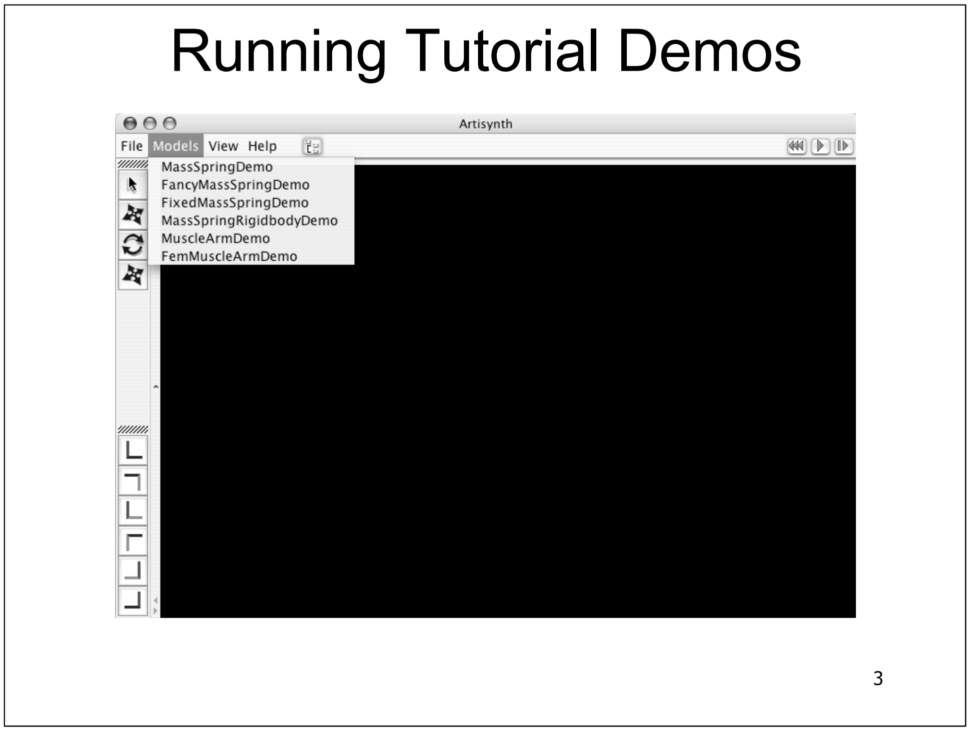
artisynth_2.1/src/artisynth/models/tutorial

The guide is based on a Tutorial Session given at the ArtiSynth Workshop at the International Seminar of Speech Production at Ubatuba, Brazil in December 2006.



Outline

- Tour of ArtiSynth Packages / Primitives
- Building a Basic Biomechanical Model
- Using ArtiSynth GUI
- Editing Properties
- Controlling Simulation
- Advanced Biomechanical Modeling



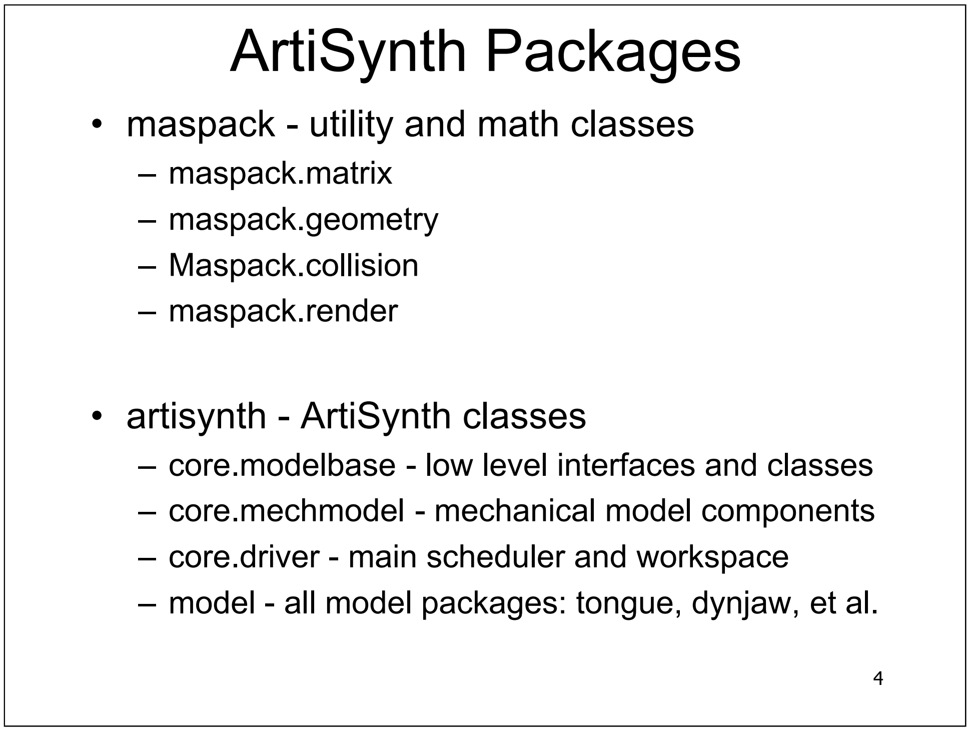
Running Tutorial Demos

In order to run the AritSynth Tutorial Demos you must do the following:

- Rename artisynth_2.1/.demoModels to artisynth_2.1/originalDemoModels
- Copy artisynth_2.1/src/models/tutorial/.demoModels to artisynth_2.1
- Copy artisynth_2.1/src/models/tutorial/activation.txt to artisynth_2.1

To run the regular ArtiSynth Demos once you have completed the tutorial:

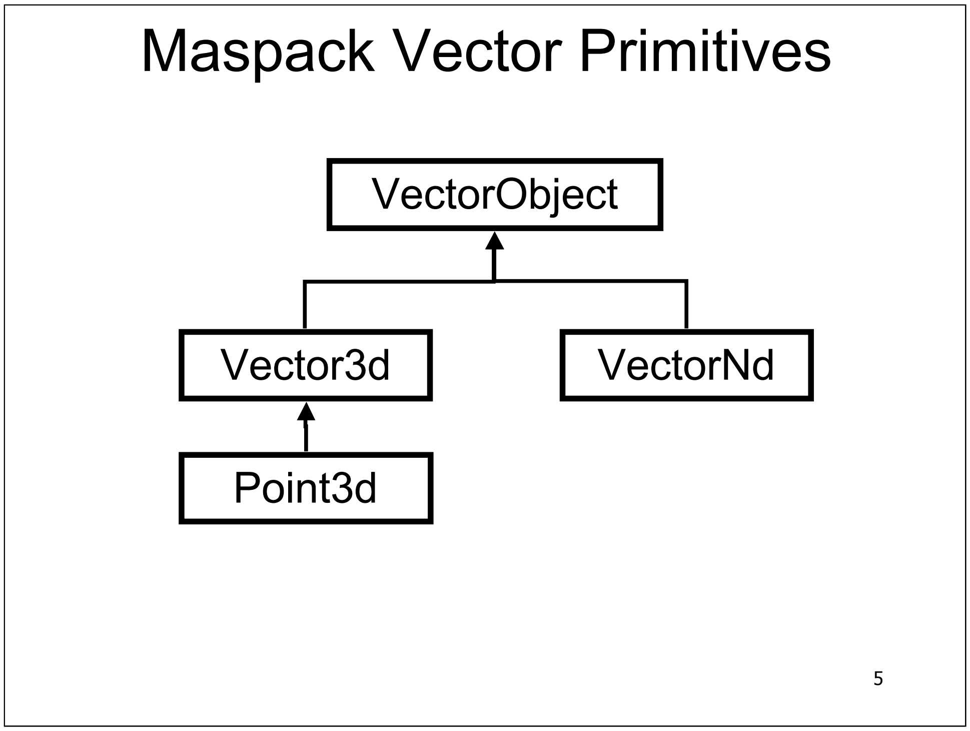
Rename artisynth_2.1/originalDemoModels to artisynth_2.1/.demoModels



ArtiSynth Packages

- maspack - utility and math classes
  - – maspack.matrix
  - – maspack.geometry
  - – Maspack.collision
  - – maspack.render

- artisynth - ArtiSynth classes
  - – core.modelbase - low level interfaces and classes
  - – core.mechmodel - mechanical model components
  - – core.driver - main scheduler and workspace
  - – model - all model packages: tongue, dynjaw, et al.



Maspack Vector Primitives

VectorObject

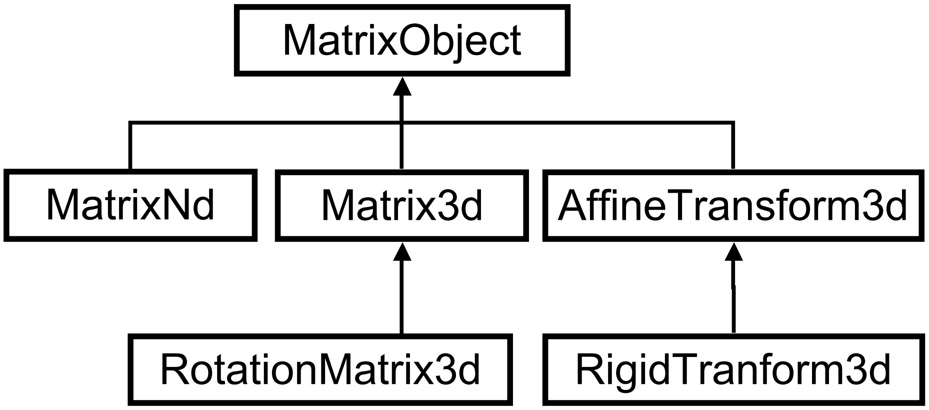
Vector3d

VectorNd

Point3d

5

Maspack Matrix Primitives



MatrixObject

MatrixNd

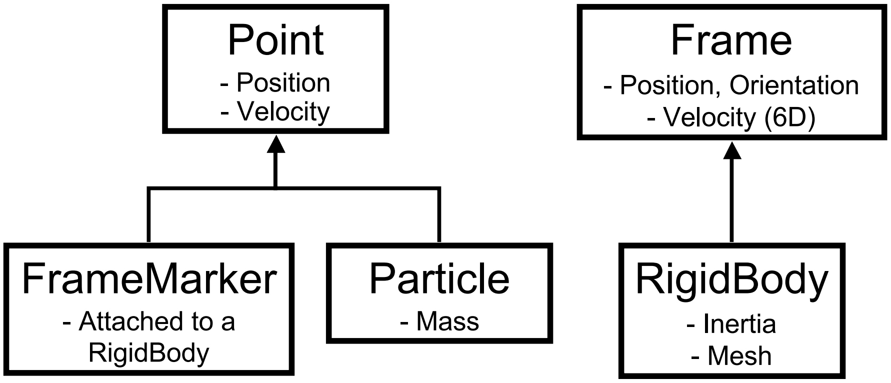
Matrix3d

AffineTransform3d

RotationMatrix3d

RigidTranform3d

Basic Dynamic Components



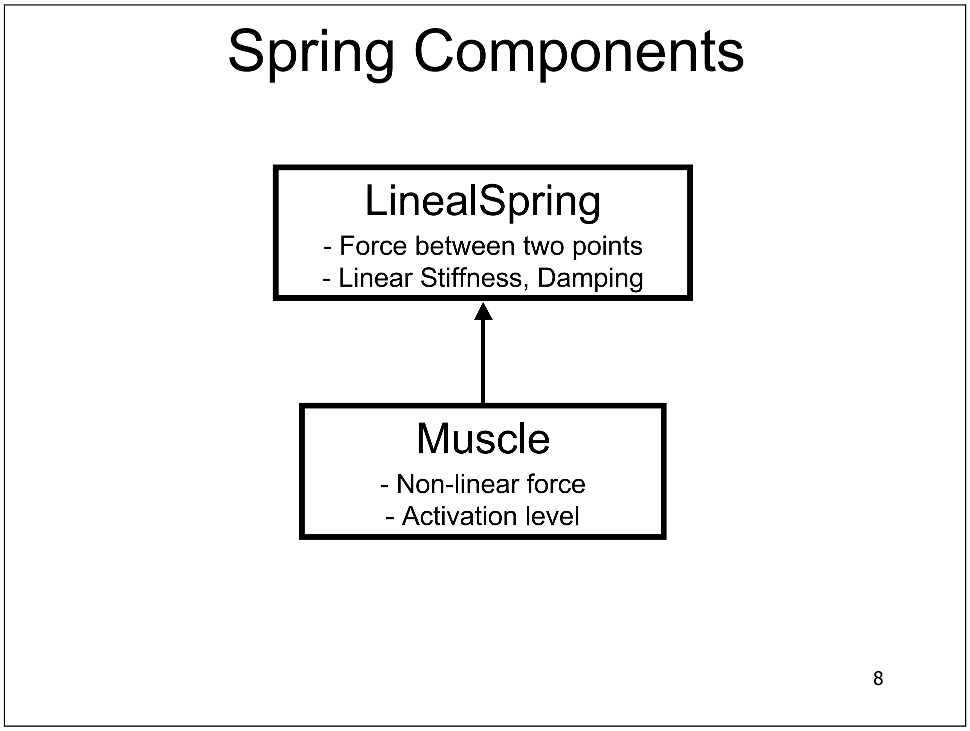
Point
- Position
- Velocity

Frame
- Position, Orientation
- Velocity (6D)

FrameMarker
- Attached to a RigidBody

Particle
- Mass

RigidBody
- Inertia
- Mesh



Spring Components

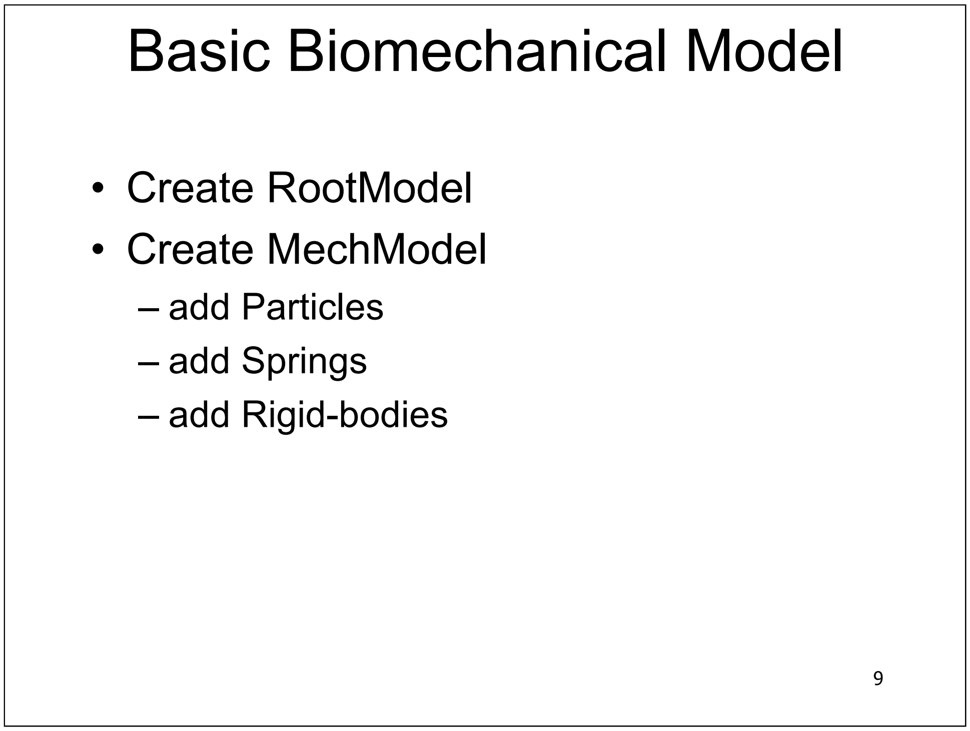
LinealSpring

- Force between two points
- Linear Stiffness, Damping

Muscle

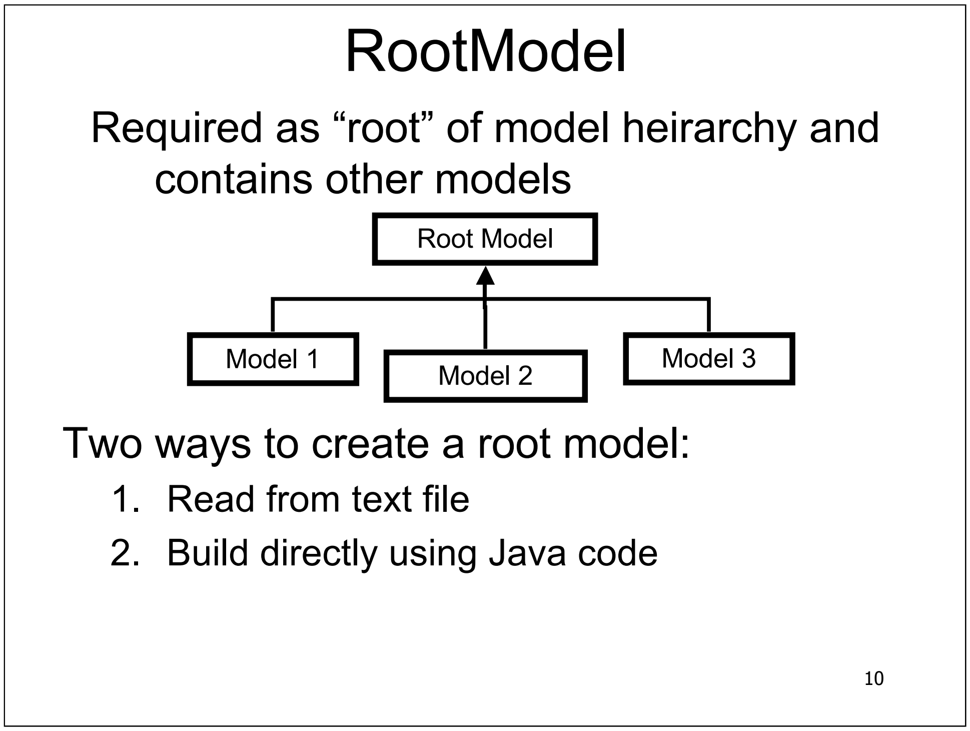
- Non-linear force
- Activation level

8



Basic Biomechanical Model

- Create RootModel
- Create MechModel
  - – add Particles
  - – add Springs
  - – add Rigid-bodies



RootModel

Required as “root” of model heirarchy and contains other models

Root Model

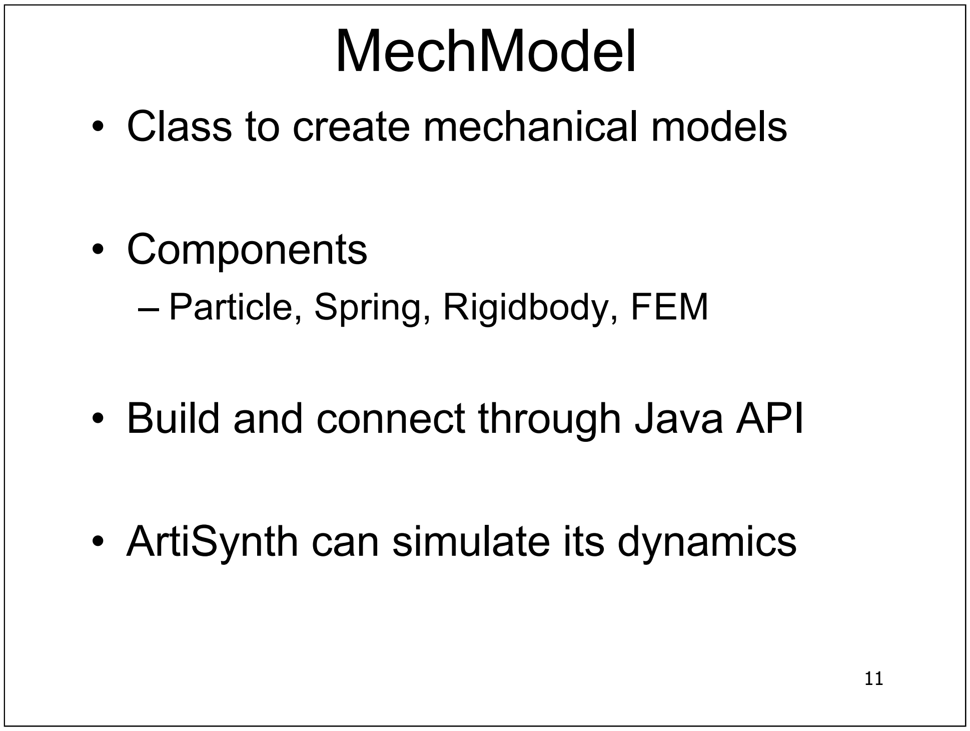
Model 1

Model 2

Model 3

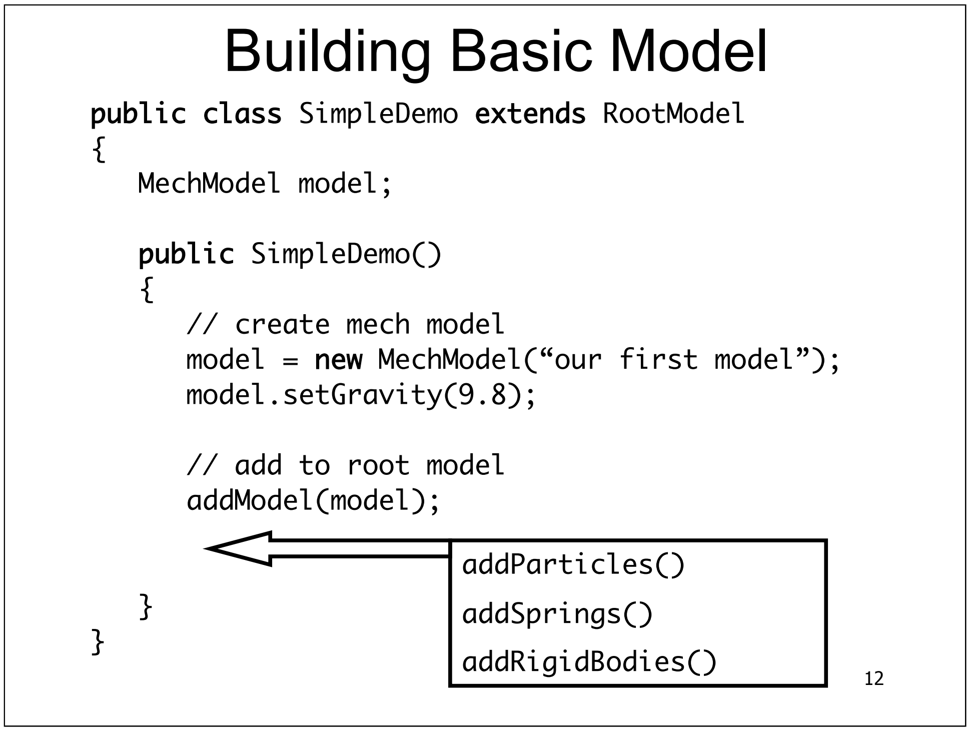
Two ways to create a root model:

1. Read from text file
2. Build directly using Java code



MechModel

- Class to create mechanical models
- Components
  - Particle, Spring, Rigidbody, FEM
- Build and connect through Java API
- ArtiSynth can simulate its dynamics



Building Basic Model

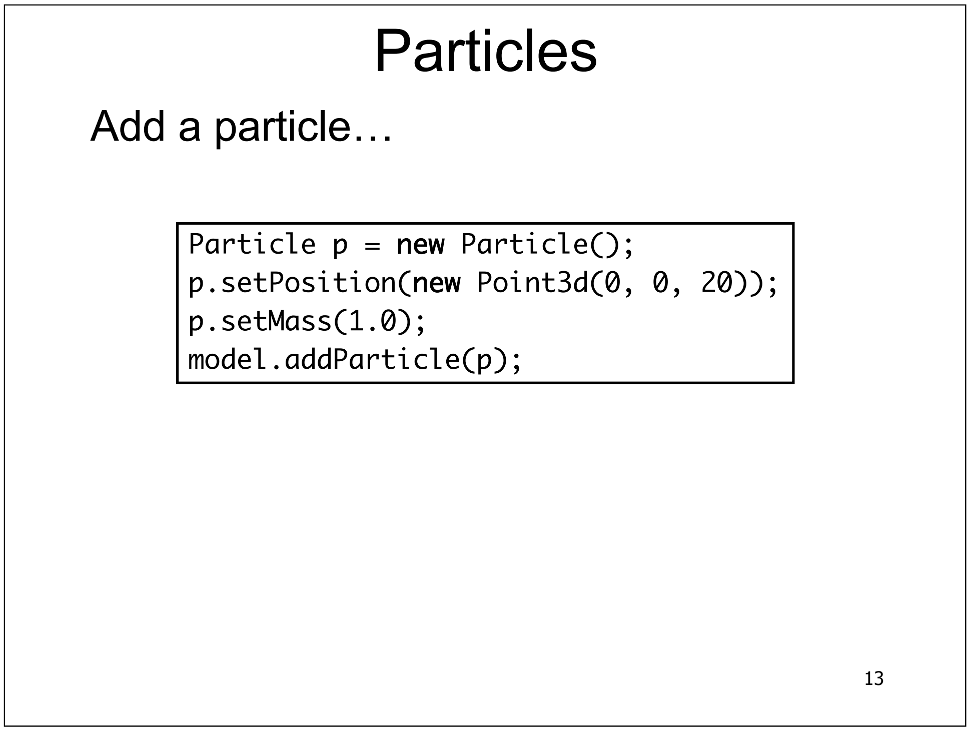
```
public class SimpleDemo extends RootModel
{
   MechModel model;

   public SimpleDemo()
   {
      // create mech model
      model = new MechModel(“our first model”);
      model.setGravity(9.8);

      // add to root model
      addModel(model);


   }
}
```

addParticles()

addSprings()

addRigidBodies()

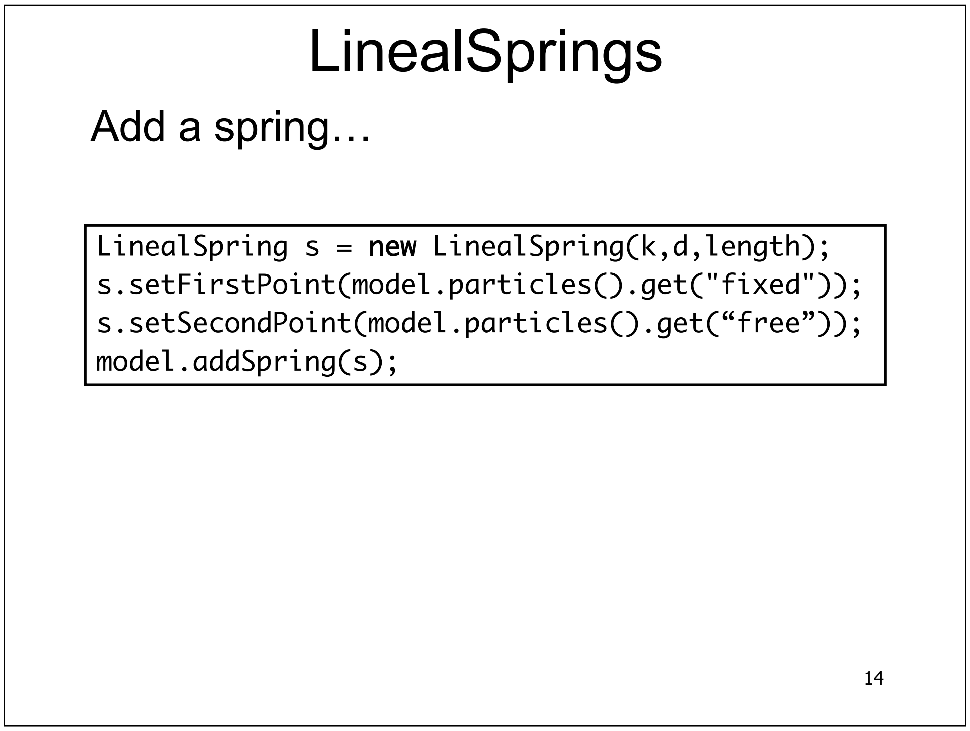
12



Particles

Add a particle…

```
Particle p = new Particle();
p.setPosition(new Point3d(0, 0, 20));
p.setMass(1.0);
model.addParticle(p);
```



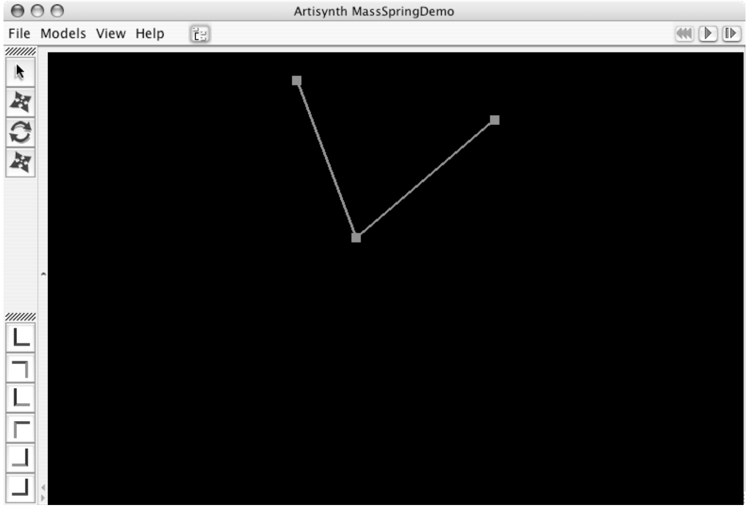
LinealSprings

Add a spring…

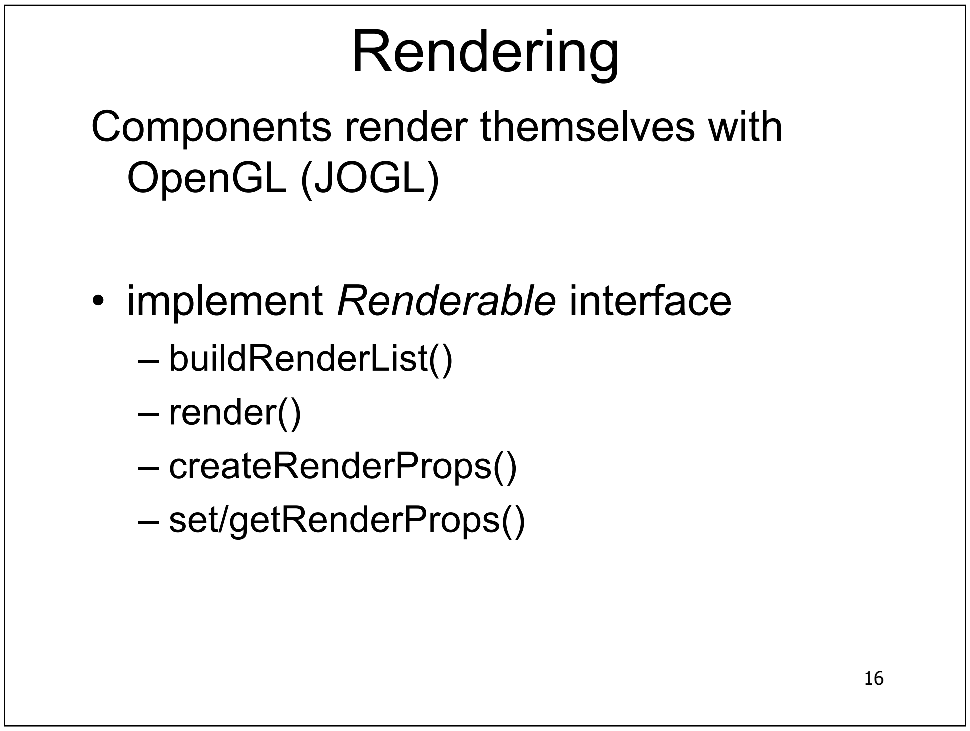
```
LinealSpring s = new LinealSpring(k,d,length);
s.setFirstPoint(model.particles().get("fixed"));
s.setSecondPoint(model.particles().get(“free”));
model.addSpring(s);
```

14

Particle Spring Demo



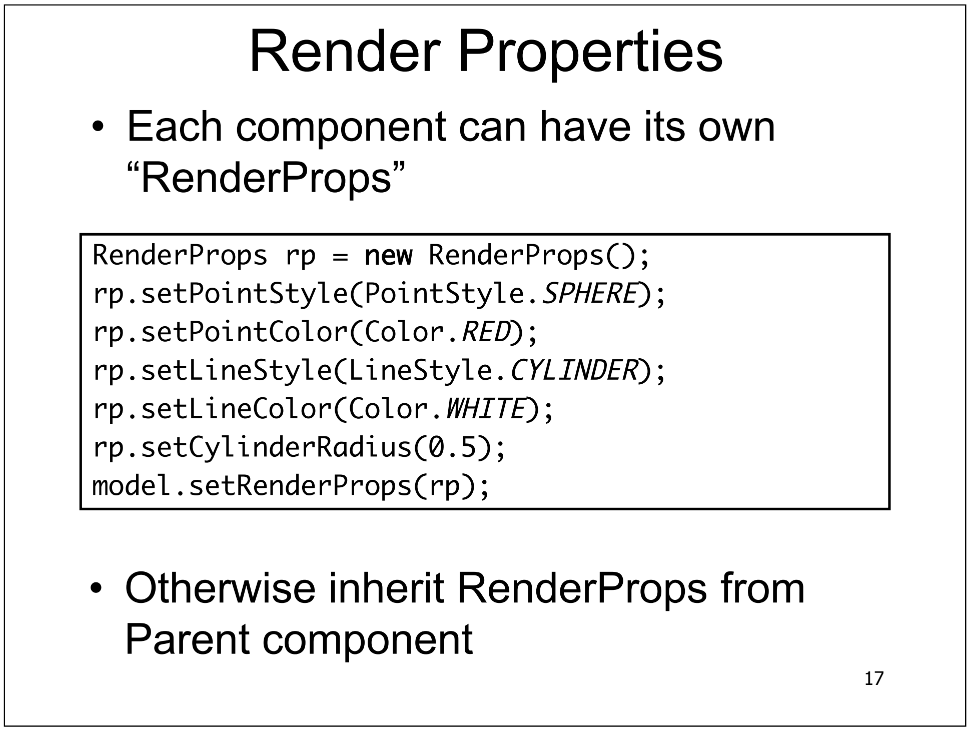
Start ArtiSynth and select Models > MassSpringDemo
Refer to artisynth_2.1/src/artisynth/models/MassSpring.java



Rendering

Components render themselves with OpenGL (JOGL)

- implement *Renderable* interface
  - – buildRenderList()
  - – render()
  - – createRenderProps()
  - – set/getRenderProps()



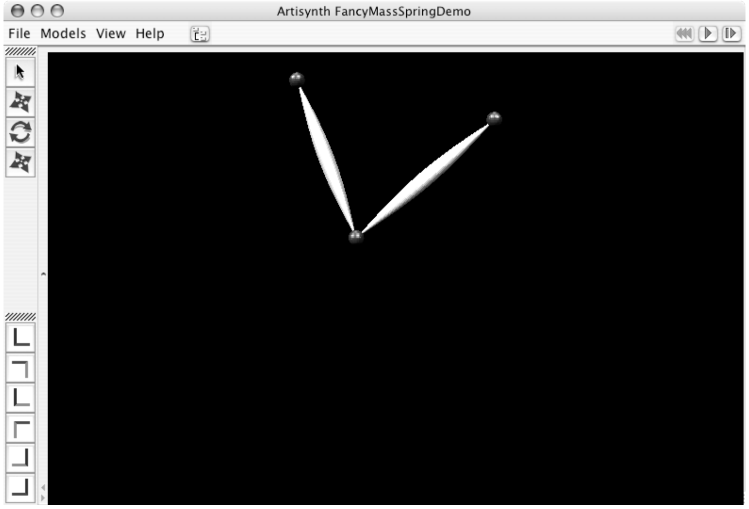
Render Properties

- Each component can have its own “RenderProps”

```
RenderProps rp = new RenderProps();
rp.setPointStyle(PointStyle.SPHERE);
rp.setPointColor(Color.RED);
rp.setLineStyle(LineStyle.CYLINDER);
rp.setLineColor(Color.WHITE);
rp.setCylinderRadius(0.5);
model.setRenderProps(rp);
```

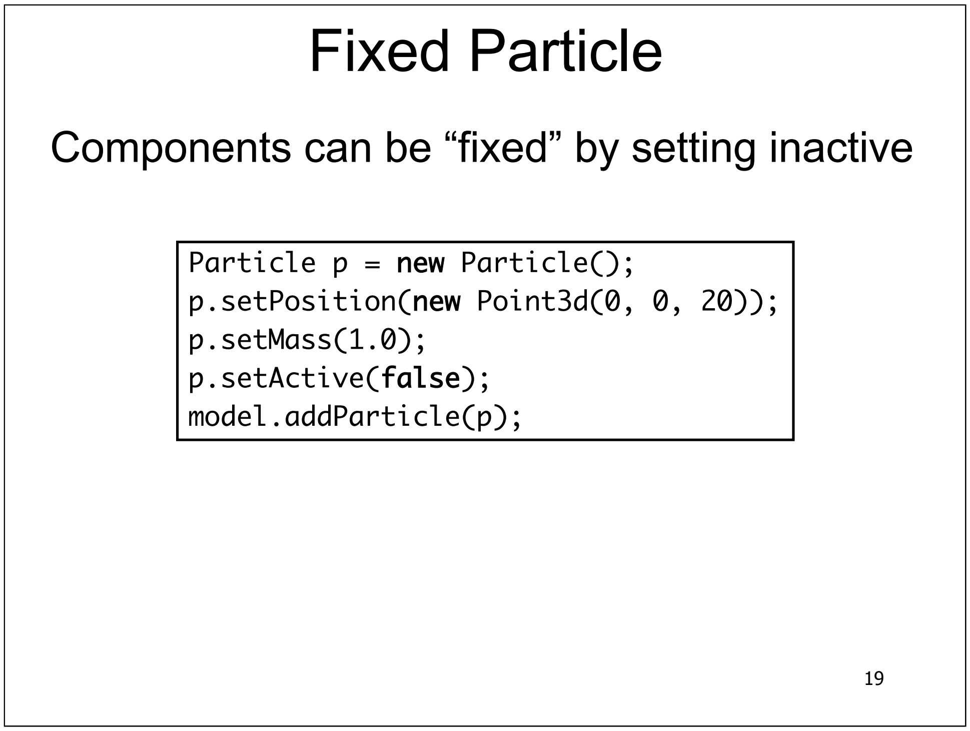
- Otherwise inherit RenderProps from Parent component

Particle Spring Demo



Start ArtiSynth and select Models > FancyMassSpringDemo

Refer to artisynth_2.1/src/artisynth/models/FancyMassSpring.java

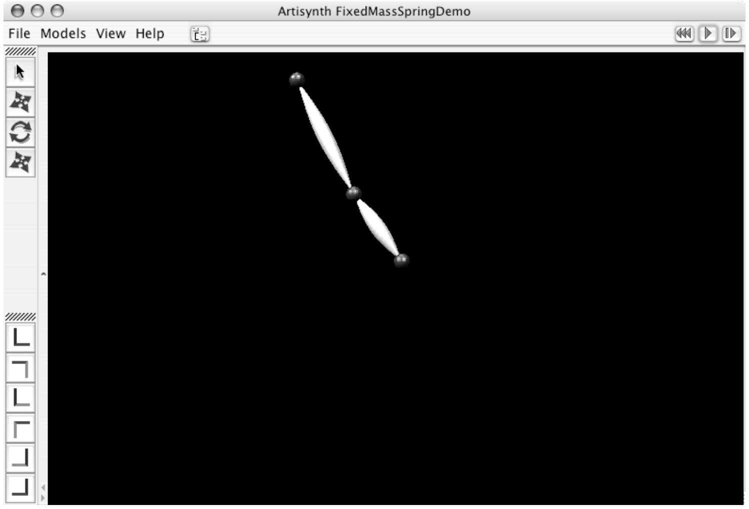


Fixed Particle

Components can be “fixed” by setting inactive

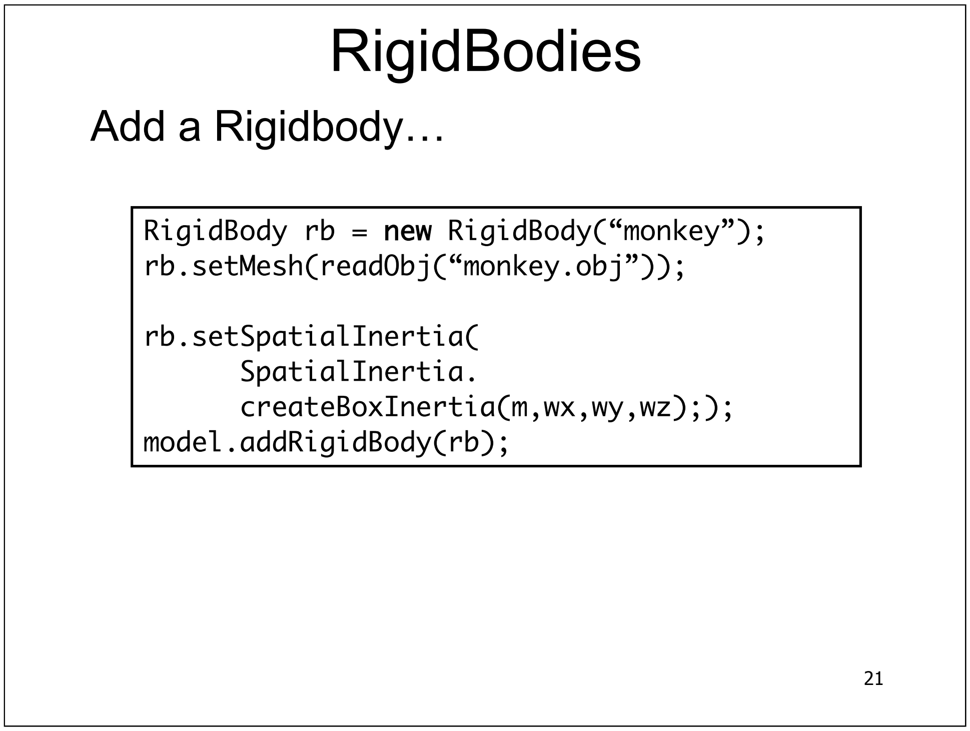
```
Particle p = new Particle();
p.setPosition(new Point3d(0, 0, 20));
p.setMass(1.0);
p.setActive(false);
model.addParticle(p);
```

Fixed Particle Demo



Start ArtiSynth and select Models > FixedMassSpringDemo

Refer to
artisynth_2.1/src/artisynth/models/FixedMassSpring.java



RigidBodies

Add a Rigidbody…

```
RigidBody rb = new RigidBody(“monkey”);
rb.setMesh(readObj(“monkey.obj”));

rb.setSpatialInertia(
      SpatialInertia.
      createBoxInertia(m,wx,wy,wz););
model.addRigidBody(rb);
```

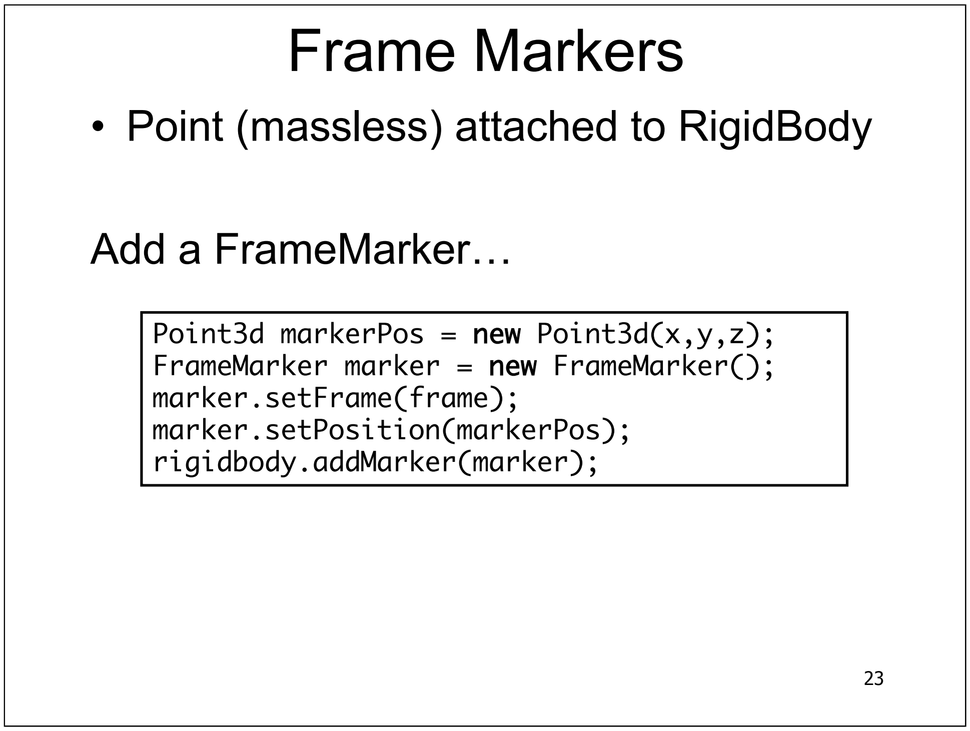
Point Attachment

- Dynamic connection between:

```
Point  -> Particle
Point  -> RigidBody
```

- To attach in code

```
model.attach(particle, rigidbody);
```



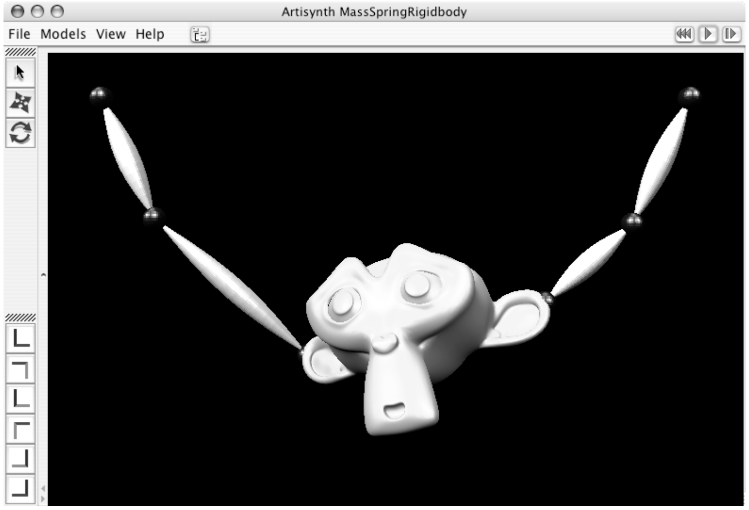
Frame Markers

- Point (massless) attached to RigidBody

Add a FrameMarker…

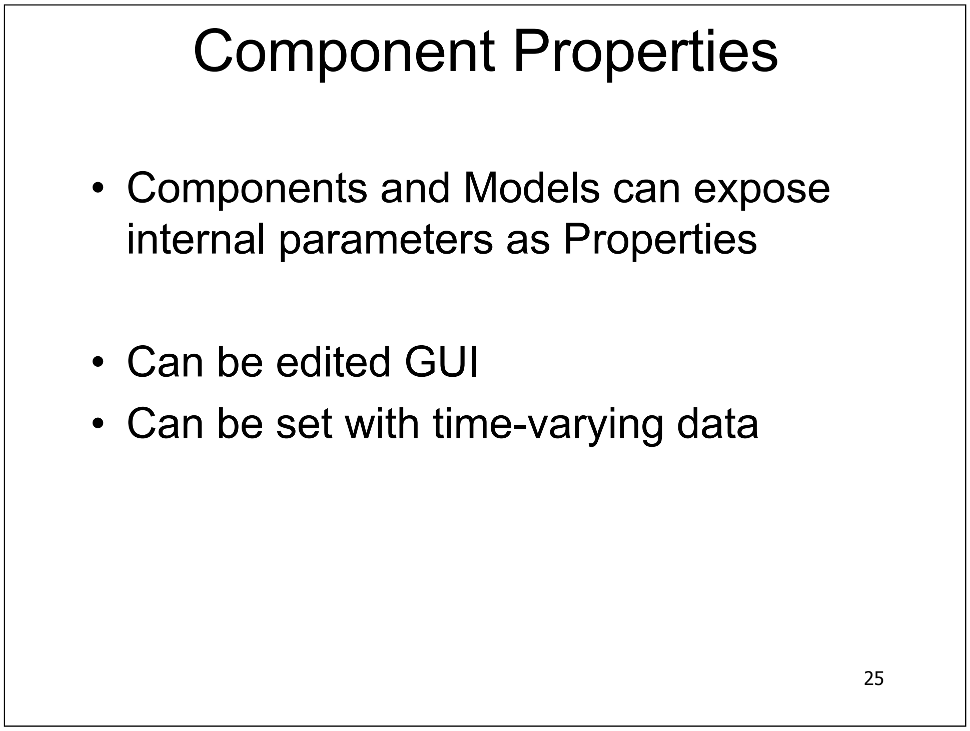
```
Point3d markerPos = new Point3d(x,y,z);
FrameMarker marker = new FrameMarker();
marker.setFrame(frame);
marker.setPosition(markerPos);
rigidbody.addMarker(marker);
```

RigidBody Demo



Start ArtiSynth and select Models > MassSpringRigidbodyDemo

Refer to
artisynth_2.1/src/artisynth/models/MassSpringRigidbody.java

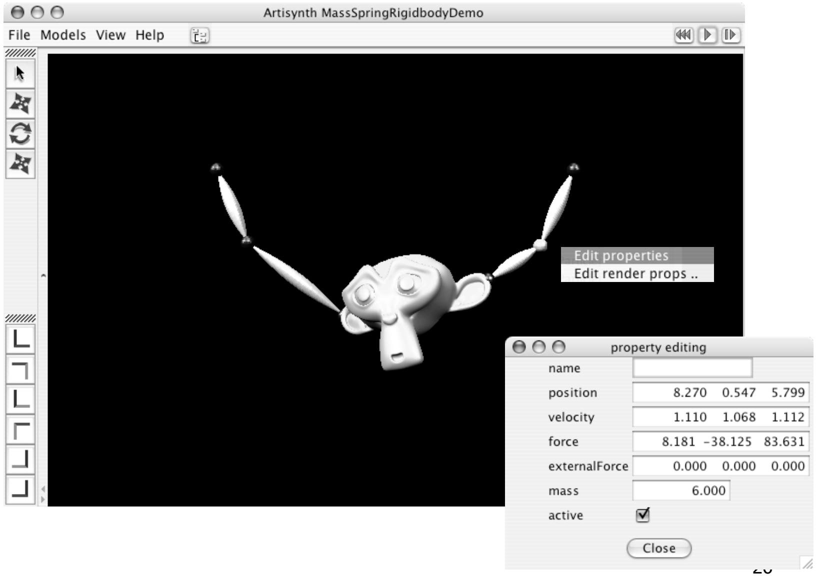


Component Properties

- Components and Models can expose internal parameters as Properties

- Can be edited GUI
- Can be set with time-varying data

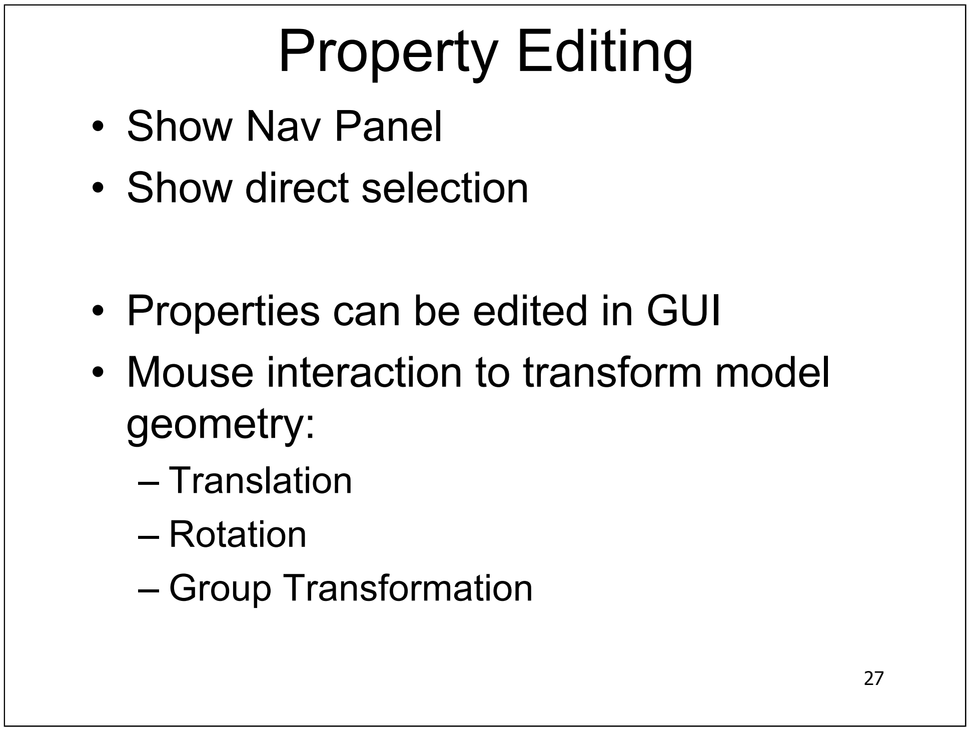
Property Editing Demo



name			
position	8.270	0.547	5.799
velocity	1.110	1.068	1.112
force	8.181	-38.125	83.631
externalForce	0.000	0.000	0.000
mass	6.000		
active	☑		

Right-Click on a component (e.g. the second particle from the right) and Select “Edit Properties”

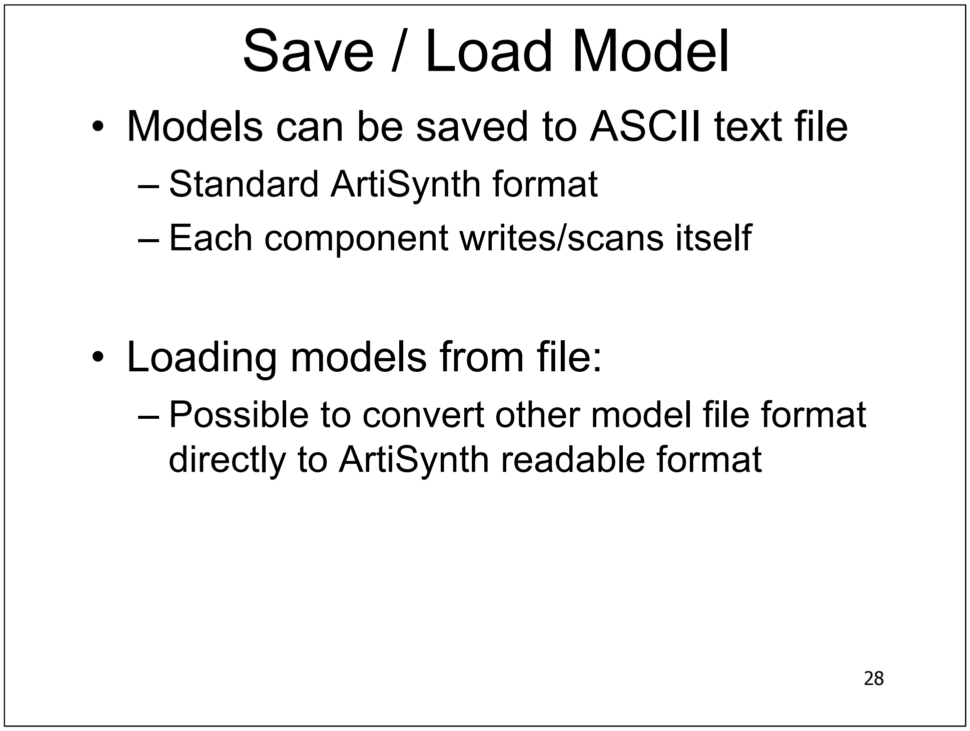
A window opens that allows you to edit properties for that component.



Property Editing

- Show Nav Panel
- Show direct selection

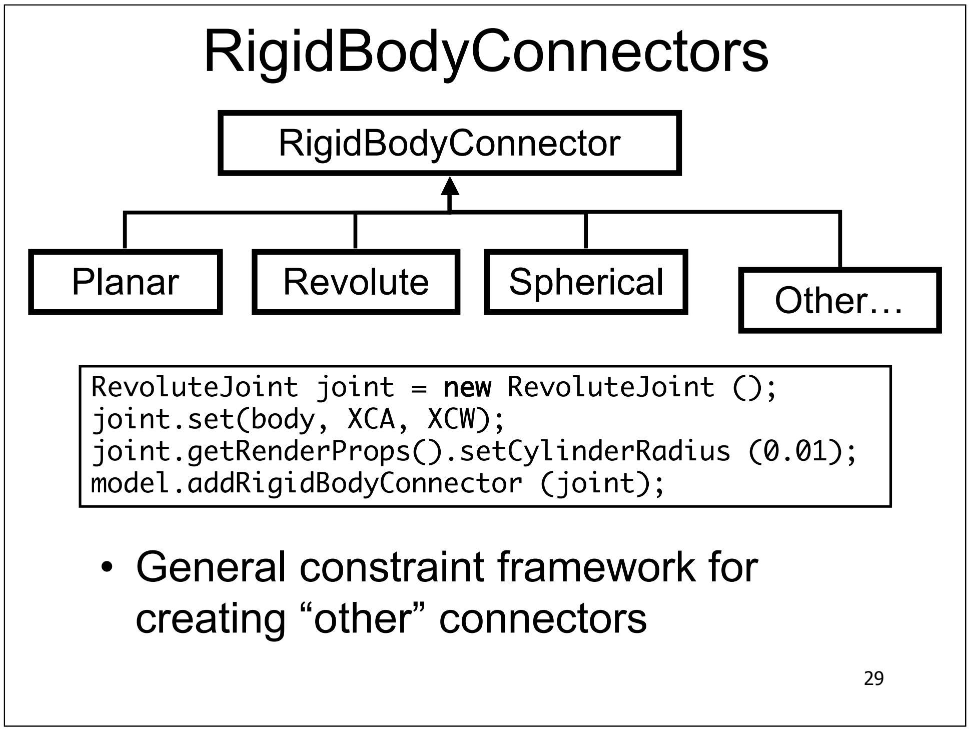
- Properties can be edited in GUI
- Mouse interaction to transform model geometry:
  - Translation
  - Rotation
  - Group Transformation



Save / Load Model

- Models can be saved to ASCII text file
  - Standard ArtiSynth format
  - Each component writes/scans itself

- Loading models from file:
  - Possible to convert other model file format directly to ArtiSynth readable format



RigidBodyConnectors

RigidBodyConnector

Planar

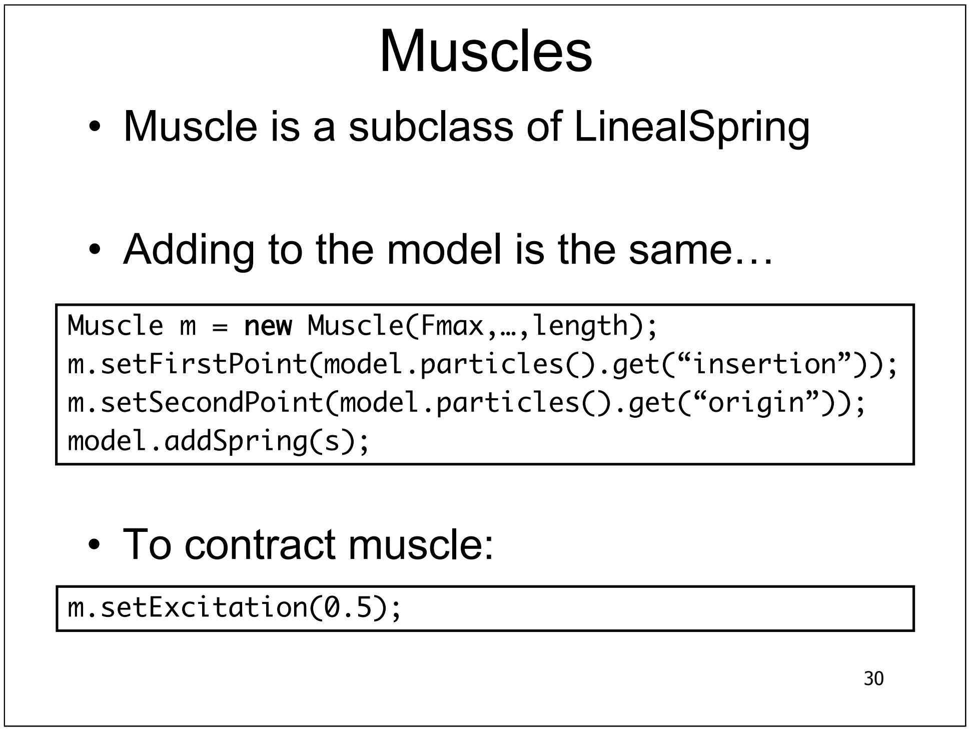
Revolute

Spherical

Other…

```
RevoluteJoint joint = new RevoluteJoint ();
joint.set(body, XCA, XCW);
joint.getRenderProps().setCylinderRadius (0.01);
model.addRigidBodyConnector (joint);
```

- General constraint framework for creating “other” connectors



Muscles

- Muscle is a subclass of LinealSpring

- Adding to the model is the same…

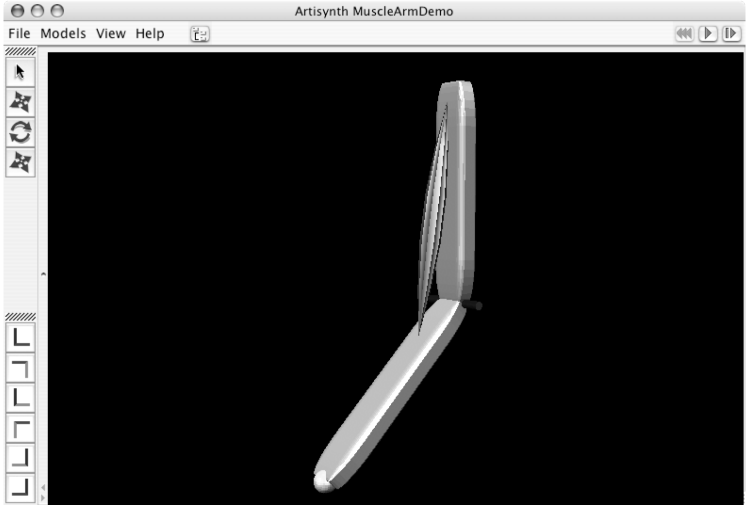
```
Muscle m = new Muscle(Fmax,…,length);
m.setFirstPoint(model.particles().get(“insertion”));
m.setSecondPoint(model.particles().get(“origin”));
model.addSpring(s);
```

- To contract muscle:

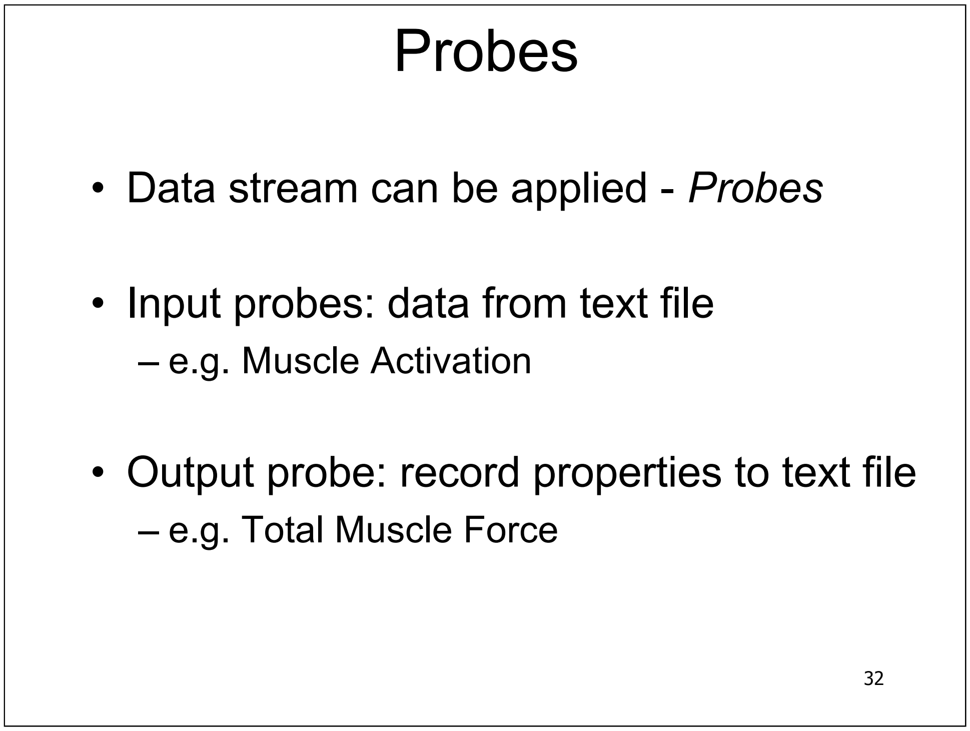
```
m.setExcitation(0.5);
```

30

Simple Arm Demo

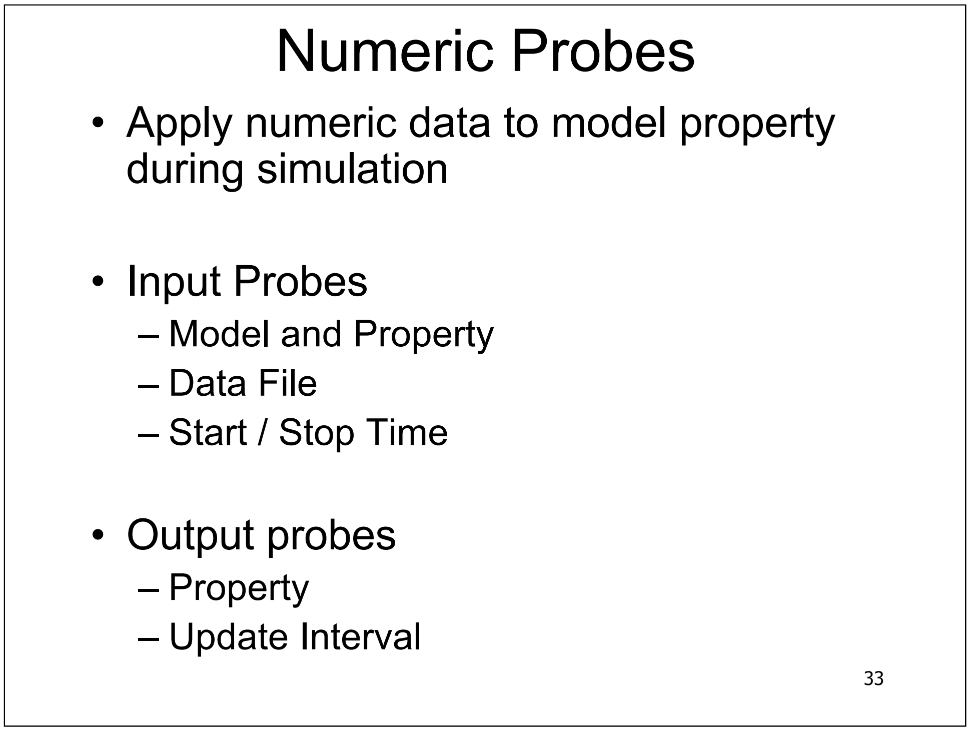


```
Start ArtiSynth and select Models > MuscleArmDemo
Refer to artisynth_2.1/src/artisynth/models/MuscleArm.java
```



Probes

- Data stream can be applied - *Probes*
- Input probes: data from text file
  - e.g. Muscle Activation
- Output probe: record properties to text file
  - e.g. Total Muscle Force



Numeric Probes

- Apply numeric data to model property during simulation

- Input Probes
  - – Model and Property
  - – Data File
  - – Start / Stop Time

- Output probes
  - – Property
  - – Update Interval

Add Input Probe

- Create in code

```
NumericInputProbe inprobe = new NumericInputProbe();
inprobe.setElement(model);
inprobe.setProperty(
    model.getProperty(“springs/0/stiffness”));
inprobe.setAttachedFileName(inFilename);

Main.getWorkspace().addInputProbe(inprobe);
```

- Create through Jython console

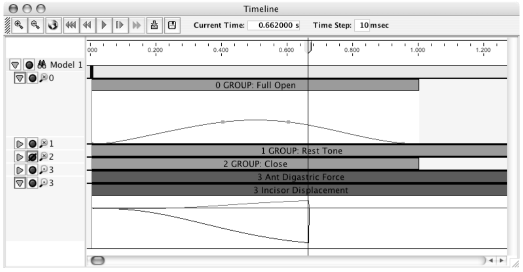
- Soon to come: adding through GUI

Simulation Control: Timeline

Graphical simulation control

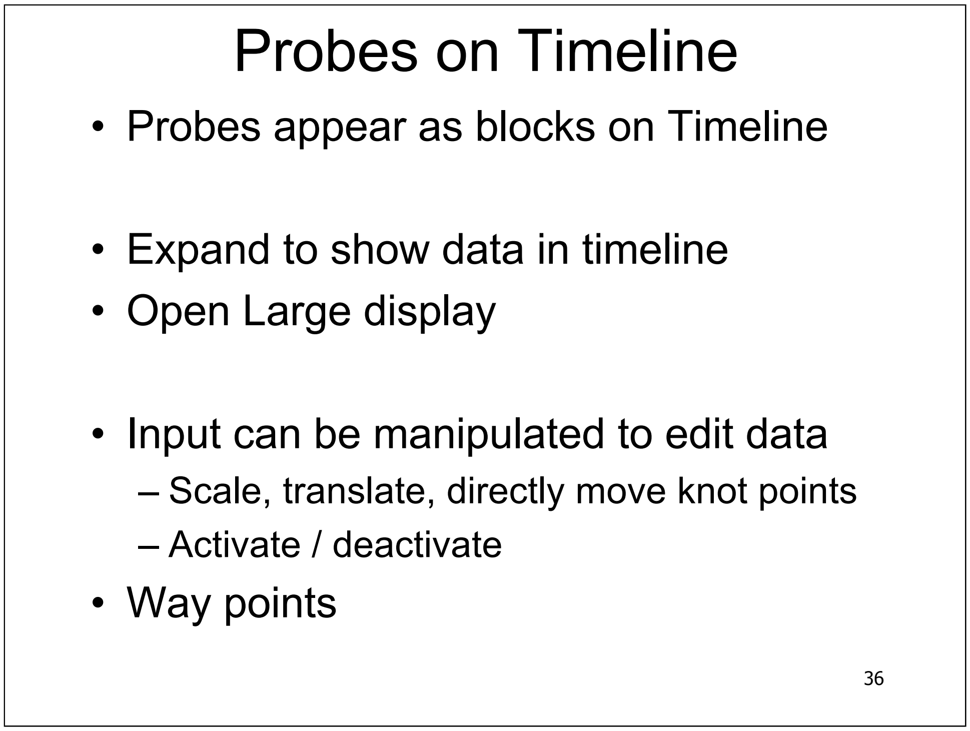
1. Manipulate input data
2. Display output data
3. Control simulation progression

Uses input / output Probes



Current Time: 0.662000 s

Time Step: 10 msec



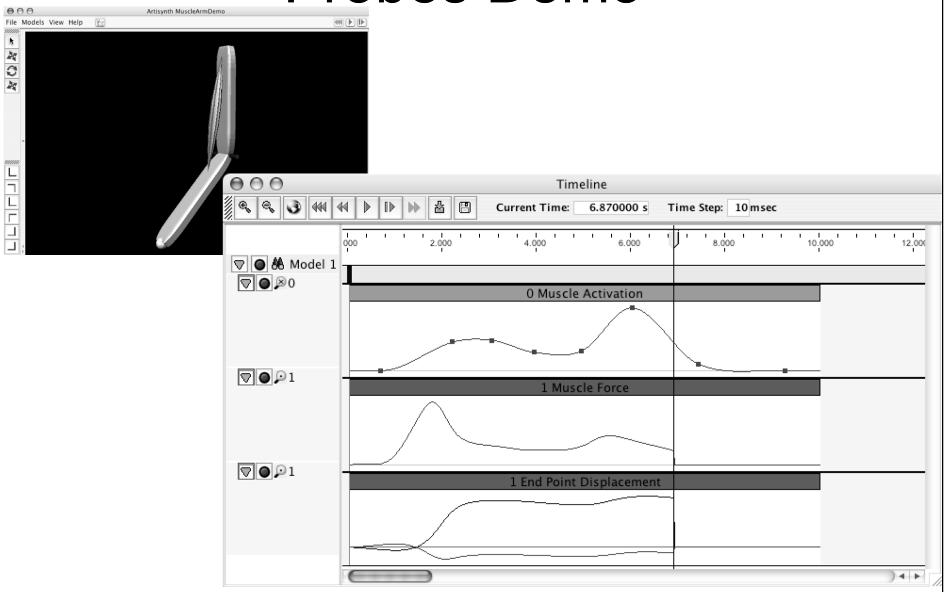
Probes on Timeline

- Probes appear as blocks on Timeline

- Expand to show data in timeline
- Open Large display

- Input can be manipulated to edit data
  - – Scale, translate, directly move knot points
  - – Activate / deactivate
- Way points

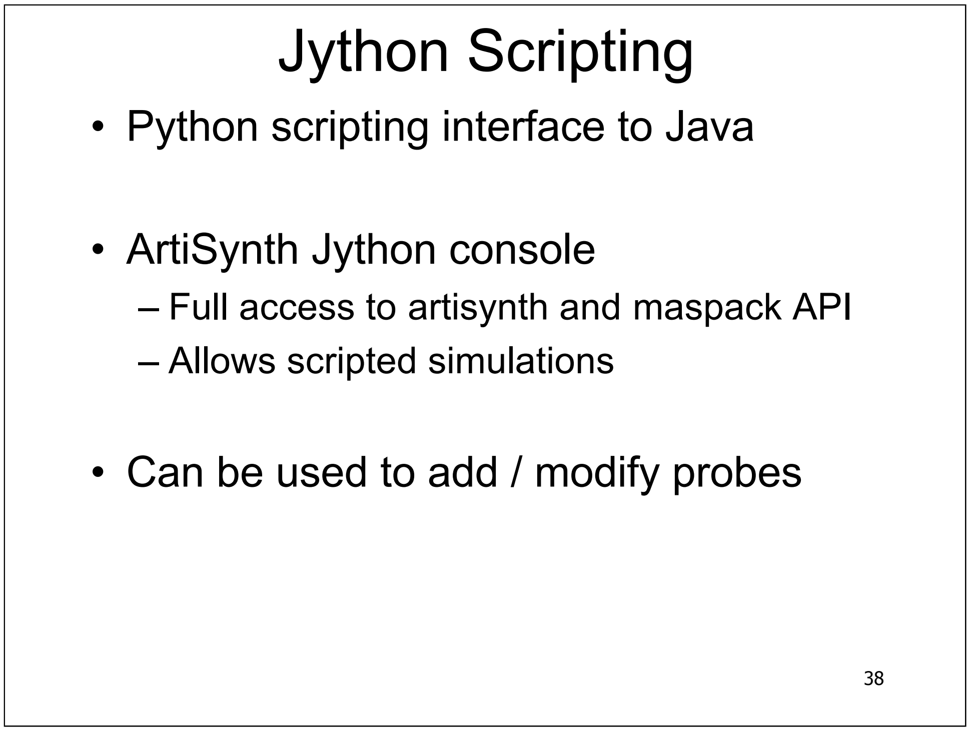
Probes Demo



Current Time: 6.870000 s

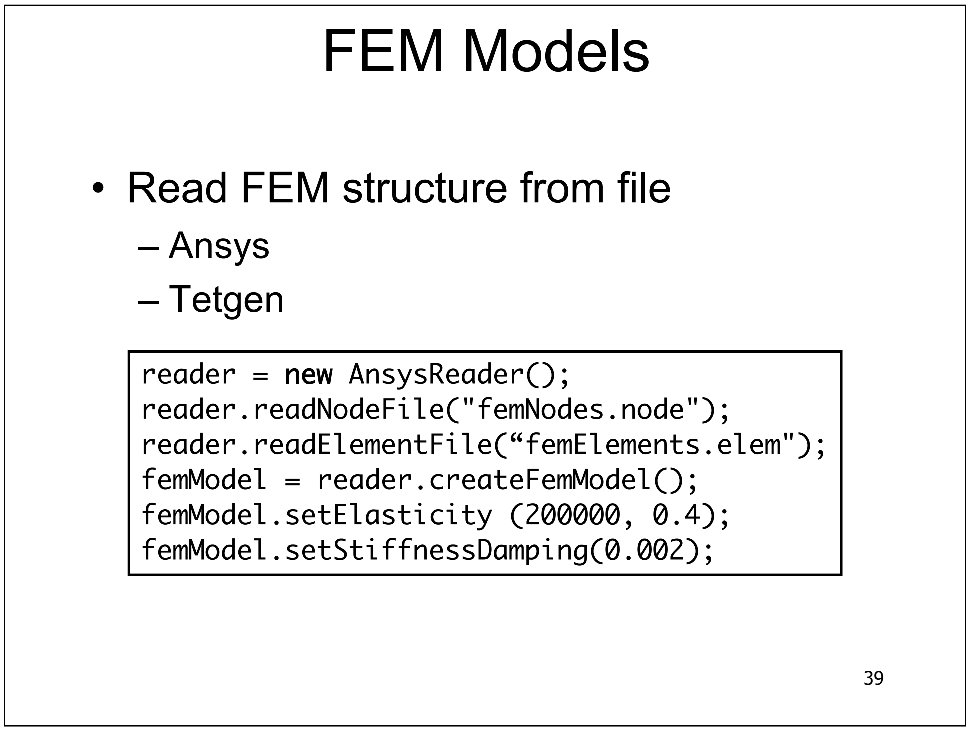
Time Step: 10 msec

37



Jython Scripting

- Python scripting interface to Java
- ArtiSynth Jython console
  - – Full access to artisynth and maspack API
  - – Allows scripted simulations
- Can be used to add / modify probes

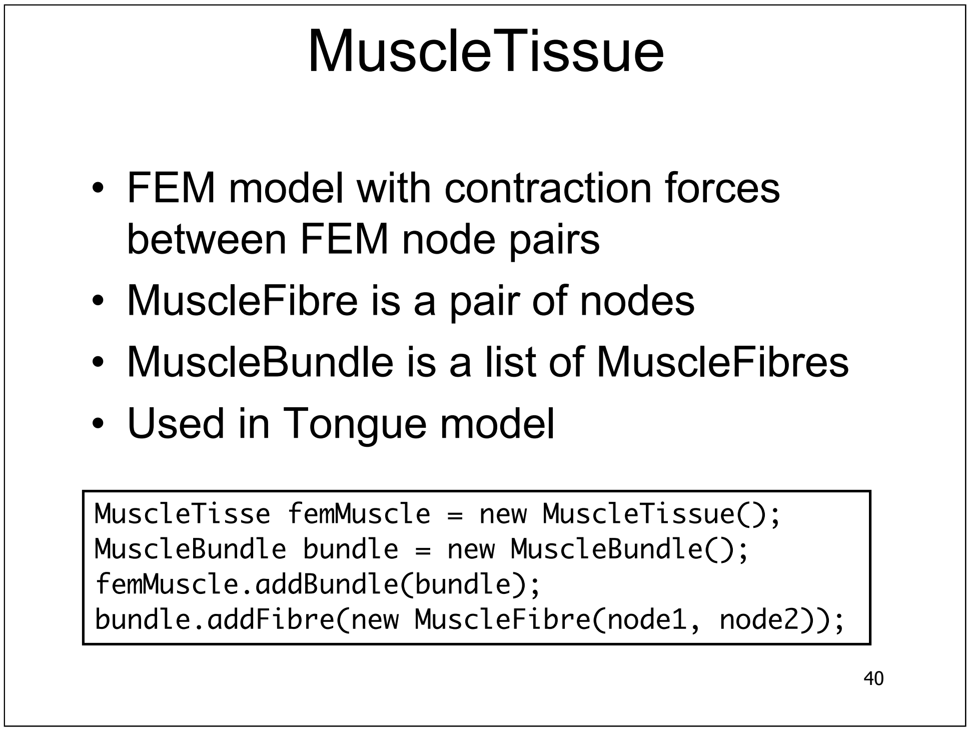


FEM Models

- Read FEM structure from file
  - Ansys
  - Tetgen

```
reader = new AnsysReader();
reader.readNodeFile("femNodes.node");
reader.readElementFile(“femElements.elem");
femModel = reader.createFemModel();
femModel.setElasticity (200000, 0.4);
femModel.setStiffnessDamping(0.002);
```

39

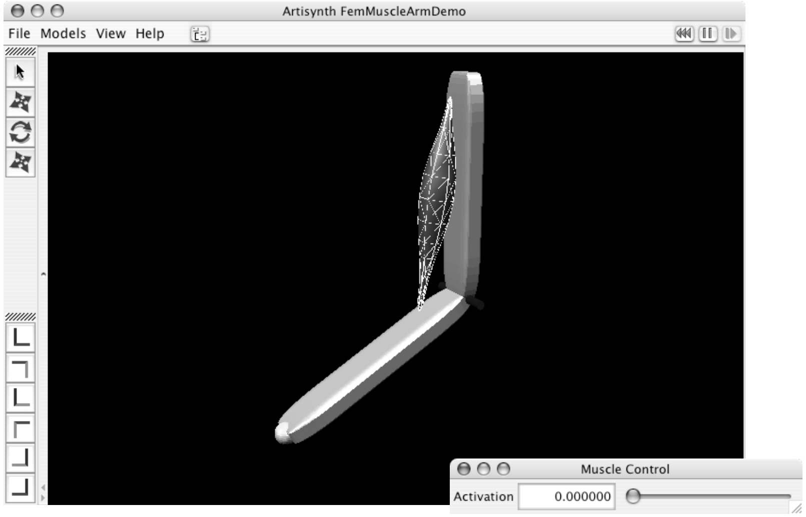


MuscleTissue

- FEM model with contraction forces between FEM node pairs
- MuscleFibre is a pair of nodes
- MuscleBundle is a list of MuscleFibres
- Used in Tongue model

```
MuscleTisse femMuscle = new MuscleTissue();
MuscleBundle bundle = new MuscleBundle();
femMuscle.addBundle(bundle);
bundle.addFibre(new MuscleFibre(node1, node2));
```

Muscle Tissue

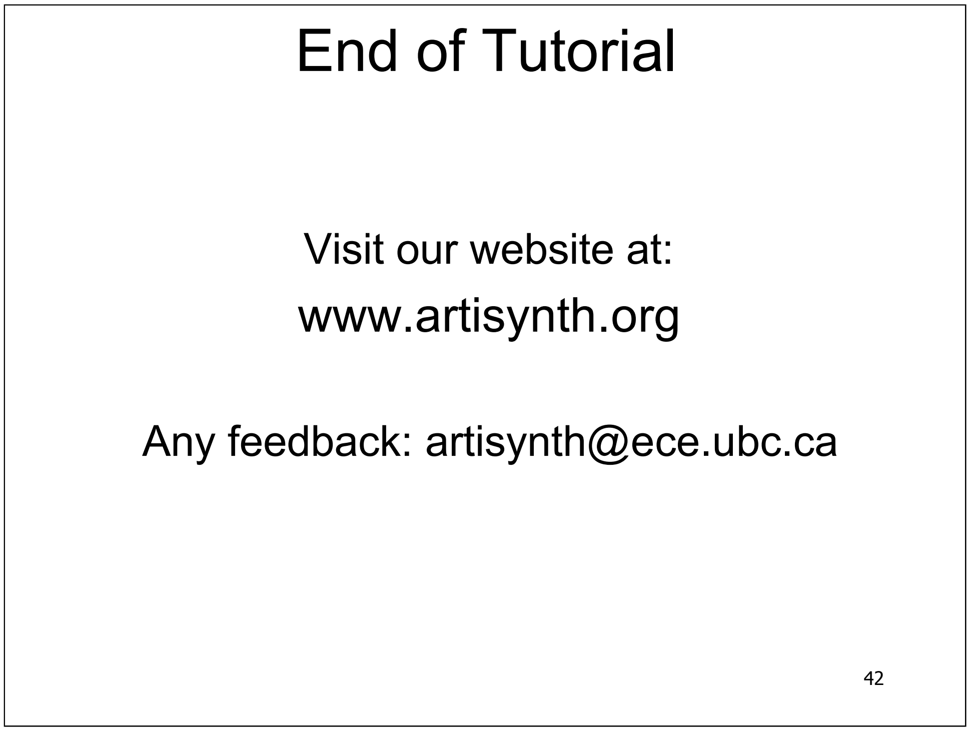


Activation 0.000000

41

Start ArtiSynth and select Models > FemMuscleArm

Refer to artisynth_2.1/src/artisynth/models/FemMuscleArm.java



End of Tutorial

Visit our website at:

www.artisynth.org

Any feedback: artisynth@ece.ubc.ca

To run the regular ArtiSynth Demos once you have completed the tutorial:

Rename artisynth_2.1/originalDemoModels to artisynth_2.1/.demoModels