Interactive 3D Modeling Exercise

Today's Lecture

- **Review: Course objectives**
- **Interactive 3D Modeling of CS105**
  - VRML 2.0
  - CosmoWorlds

Objective of This Course

Investigate **automatic** methods for creating and fixing 3D models suitable for interactive graphics.

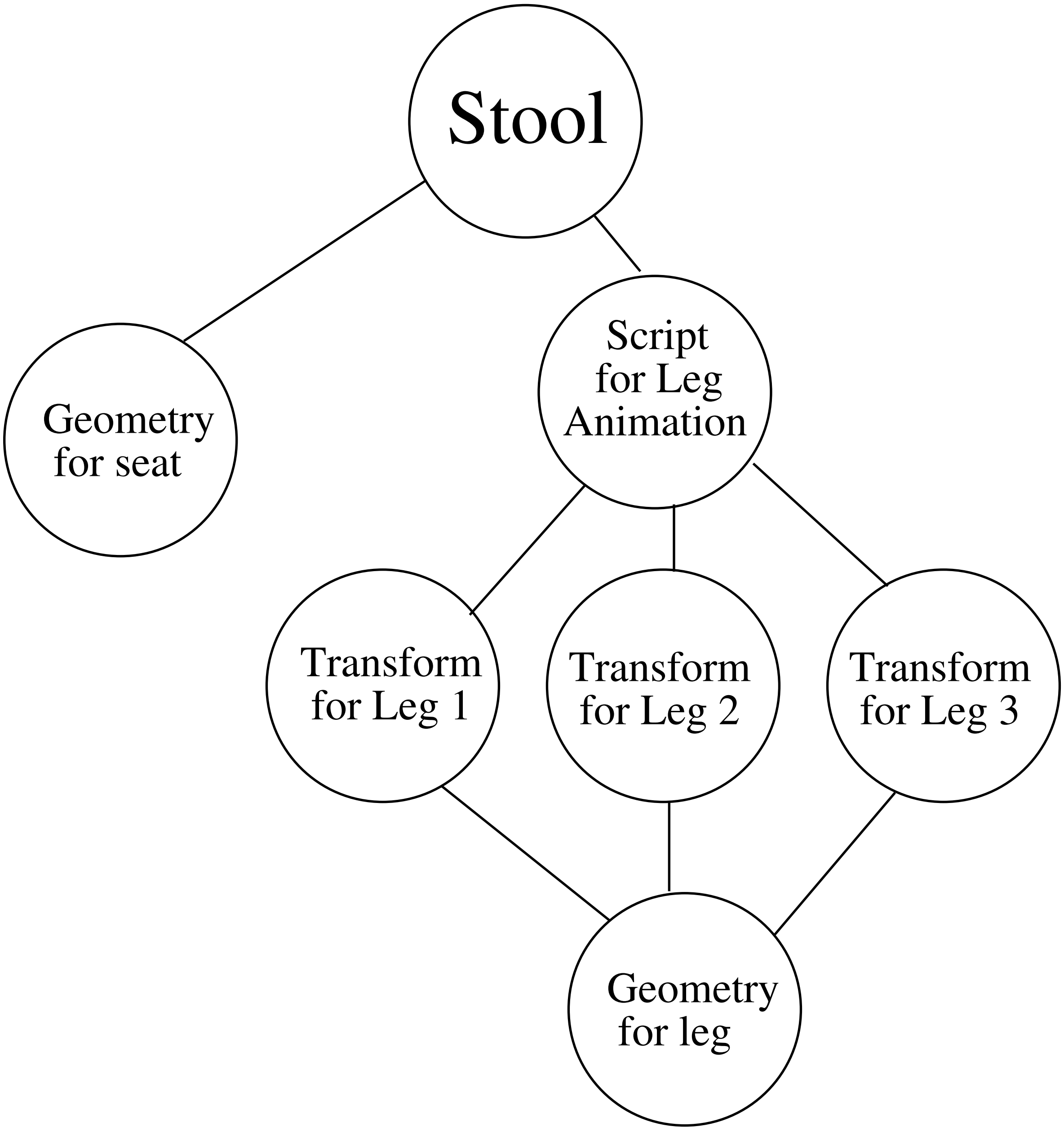
- **Model Generation**
  - Procedural methods
  - Capture from sensors
  - Derivation from other data
- **Model Manipulation**
  - Fixup
  - Simplification
  - Optimization
- **Model Simulation**
  - Lighting
  - Appearance
  - Dynamics

Interactive 3D Modeling

- **Programming assignment #1**
  - Jointly we will create a realistic 3D model of CS 105 suitable for interactive visualization with any VRML 2.0 browser.
  - We will partition the model into several ‘‘pieces’’ to be created by different people
    - *walls, floors, ceilings, desks, chairs, etc.*
  - Each piece of the model should adhere to agreed upon guidelines so that they all go together
  - The union of all the pieces should be a beautiful 3D model of CS 105.

VRML 2.0

- **Representation = Scene Graph**



VRML 2.0 Nodes

- Geometry
  - Box
  - Cone
  - Cylinder
  - ElevationGrid
  - Extrusion
  - IndexedFaceSet
  - IndexedLineSet
  - PointSet
  - Sphere
  - Text
- Geometric Properties
  - Color
  - Coordinate
  - Normal
  - TextureCoordinate
- Appearance
  - Material
  - ImageTexture
  - PixelTexture
  - TextureTransform
  - FontStyle

VRML 2.0 Nodes (cont)

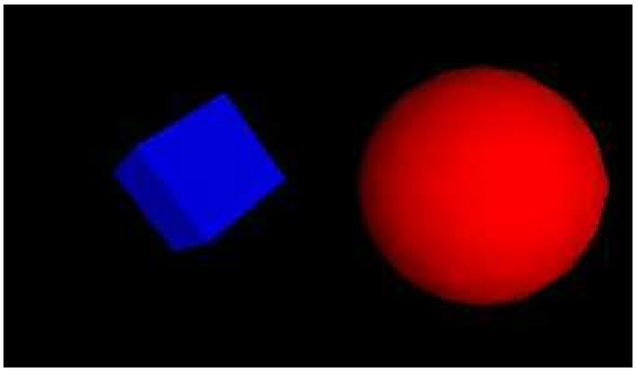
- Grouping nodes
  - Anchor
  - Billboard
  - Collision
  - Group
  - Transform
- Special Groups
  - Inline
  - LOD
  - Switch

VRML 2.0 Example

```
#VRML V2.0 utf8
Transform {
  children [
    Transform {              # a red sphere
      translation 3 0 1
      children [
        Shape {
          geometry Sphere { radius 2.3 }
          appearance Appearance {
            material Material { diffuseColor 1 0 0 }   # Red
          }
        }
      ]
    }

    Transform {              # a blue box
      translation −2.4 .2 1
      rotation     0 1 1  .9
      children [
        Shape {
          geometry Box {}
          appearance Appearance {
            material Material { diffuseColor 0 0 1 }  # Blue
          }
        }
      ]
    }

  ]
}
```



Cosmo Worlds

- Modeling
  - Primitives (cubes, spheres, etc.)
  - Polygonal modeling (extrude, split, etc.)
  - Clip art
- Appearance
  - Materials
  - Textures
  - Texture coordinates
  - Lighting
  - Normals
- Interaction
  - Switches
  - Scripts, triggers, and routes
  - Keyframe animator
- Scene Composition
  - Placement
  - Anchors, links, etc.

Cosmo Worlds (cont)

- Optimization
  - Polygon reducer
  - LODs
  - Billboards
- Navigation
  - Viewpoints
  - Collision proxies
- Digital Media
  - Sound editor
  - Image editor
  - File import

http://www.cosmo.sgi.com/products/studio/worlds

CS 105 Modeling Requirements

- **Guidelines**
  - Coordinate systems
  - Protos
  - Naming
- **Modular structure**
  - Inlines
  - Instancing (DEF/USE)
- **Efficient rendering**
  - Bounding volume hierarchy
  - LODs for all complex objects
  - Textures replace surface detail
  - Collision proxies
  - Billboards

Chapter 10 of ‘‘The VRML 2.0 Handbook’’