

# Art 484: Advanced 3D Animation

Neal McDonald  
[mcdo@umbc.edu](mailto:mcdo@umbc.edu)

Lecture 2, Feb 7

# Rigging

- Models can be deformed; you could animate that way
- But it's better to rig the model
- The rig lets you move the model around reversibly
  - a character will need to hit the same pose over and over

# Judging a rig

- a rig is good enough— that is, it's done when:
  - it allows all required poses
  - protects the model that it carries

# Goals for pro rigs

- protects the model that it carries
- prevents disallowed poses
- does not interfere with animation
  - ease of use
  - intuitiveness
  - conformant to standards– it is what the shop expects
- correct, competent use of all appropriate tools

# The difference between student and pro rigs/productions

- Student rigs
  - made and used by the student
  - used for one project, maybe 4 scenes
  - storyboard known imprecisely
  - rigger is free to make changes to everything
- Pro rig
  - made by a team of specialists
  - used in maybe 100's of scenes
  - storyboard is an actual legal document
  - the results are given to the animating team

# put more bluntly

- Your rig has to work only for you
  - if other people find it confusing, so what?
- You must control the time you spend on it
  - you will graduate!
  - not all of Maya's tools are appropriate for you-- **yet**
- Your rig is part of your cinematography
  - if your rig can't do X, change the shot!
  - pros can't do that.

# a note about rigging books

- they are written either by industry types
  - (for members of the rigging team)
- or they are written by the Maya team
  - (to brag about their awesome functionality)
- both groups will scorn you (and me) for not achieving OCD-level mastery of allll of the tools
- Do you want to be a pro rigger? No.
  - Then don't allow them to set your standards.

# the role of invention

- there is not one correct way to rig
- there might be several rigs for the same character
  - with no rig fulfilling all character requirements
- there are many techniques



# The role of planning

- Do not make more rig than you need
  - if you never see a character below waist, don't make shoes– or legs!
    - related: do you *need* to see its legs?
  - do not worry about getting your rig to do everything
  - do not even worry about whether your rig is free of bugs

# two kinds of student project

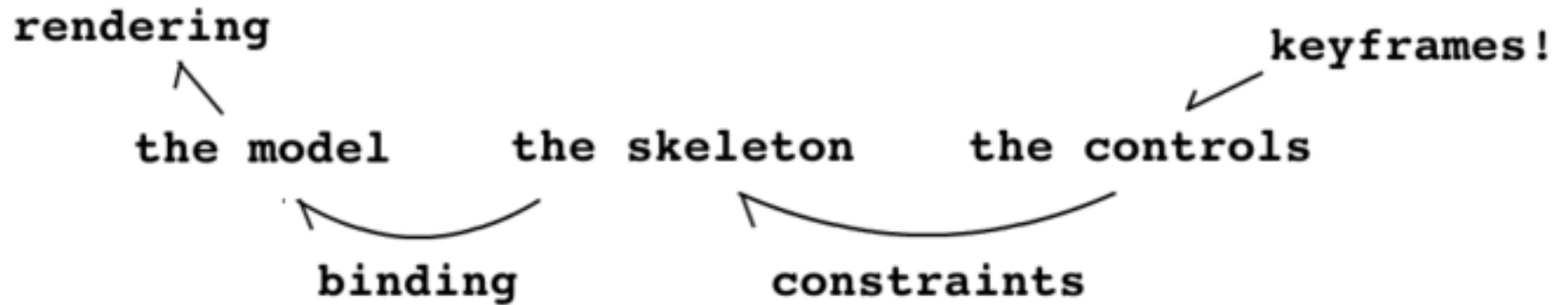
- formalist
  - make the model, make the scene, then try to treat the scene as a real one and play director in it
  - terribly wasteful! but you can get brilliant results
  - in 2-5 years
- control-freakist
  - play in your storyboards
  - finalize your storyboards
  - go to Maya with a list of tasks, and do them

# The final project

- One scene, with three characters interacting
- First task: imagine a moment
- Second task: who are the characters?
  - You will write out their histories for a grade.
  - Their histories are part of assignment D
  - Once you have their histories, you can give them a walk.

Let's get into the tech

# with a picture!



# rigged character components

- the model
  - is attached to the skeleton by “binding”
  - deformed by the skeleton
- the skeleton
  - deforms the model
  - is attached to the controls by constraints
- controls
  - the interface to the skeleton
  - animator manipulates and sets keys

# classifying Maya features

- **the model:** polys, NURBS, subdivs, particle systems, anything visible
- **Binding:** rigid and soft
- **the skeleton:** groups, joints, IK handles, deformers
- **Constraints:** point, parent, orient, etc.,  
– also: connections and expressions
- **the controls:** just geometry, but attached

# Controls?

- Maya does not have a class of object called “controls”
- They’re just regular objects used in a special way.
- This is a major change in how you animate!
  - You don’t even select the model!
  - “What, we’ve been doing it wrong?” Sorta!

# flow of control

- the animator makes the controls move
- the movement of the controls causes the constraints to move the skeleton
- the binding deforms the model to follow the skeleton
- vague, disrupted or conflicting instructions cause bugs!

# flow of questions

- another way to think about it– backwards
- you have to render your model: what does it look like?
  - depends on where the skeleton is– where is it?
    - depends on where the controls are– where are they?
      - depends on what the keyframes say!

# context

- there are two kinds of interface in Maya!
  - the kind you've been learning
  - your rig

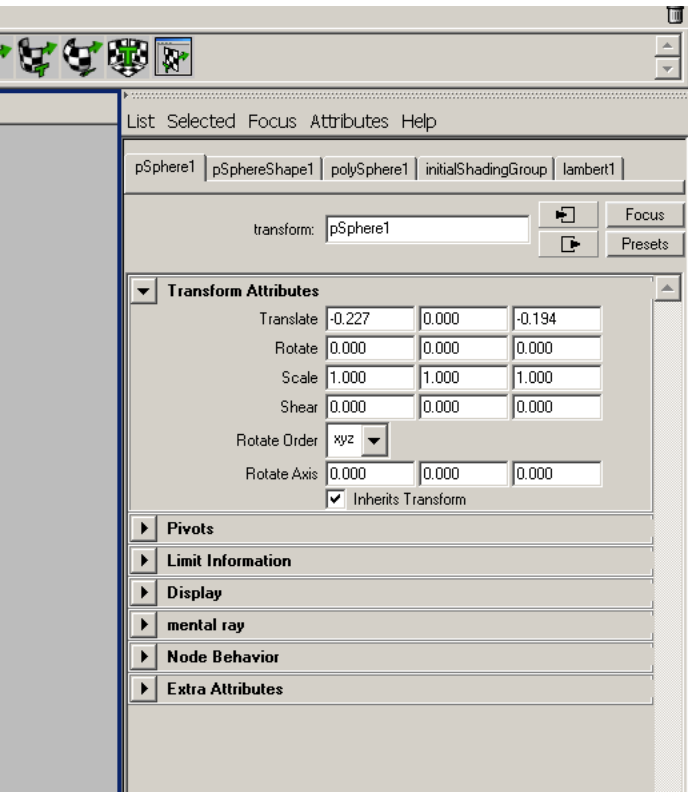
# technological roots

- The two underlying technologies are groups and expressions.
  - All rigging uses groups or something like a group.
  - Animation, constraints, connections- all are kinds of expression.
- If you understand groups and expressions, then you can rig what you want.
  - Otherwise? Not.

# Starting with expressions

- Looking at attributes.

# The Attribute Editor



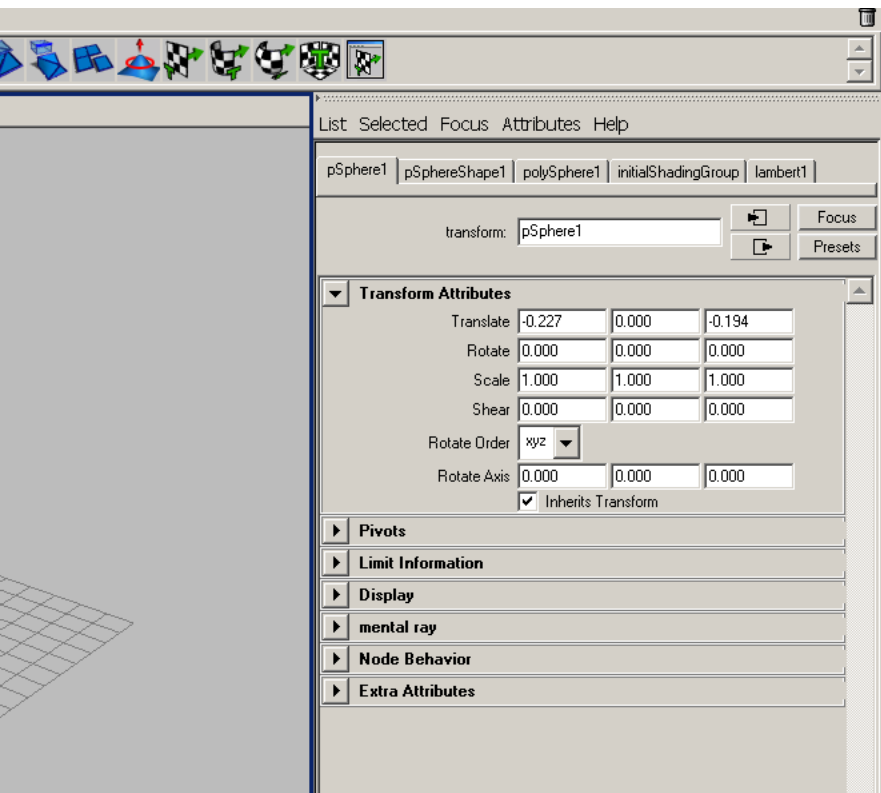
- Bring up with Ctrl-A
- Note Tabs: one for each component of the selected object
- There can be 100 tabs
- Several are special

# Note communications

- Make a sphere, bring up the attribute editor, and move the sphere
- Numbers change
- Change the numbers: sphere moves
- 3 columns: X, Y, and Z
- Translate, rotate, scale
  
- INFORMATION IS MOVING FROM THE BOX TO THE SPHERE

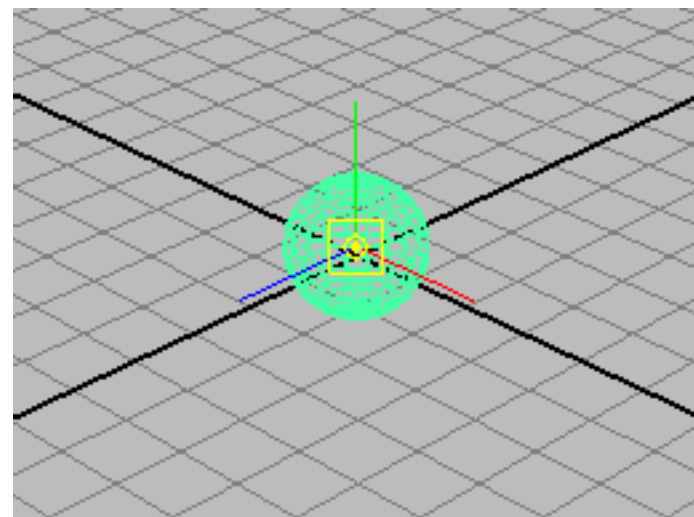
# The group tab

- Groups are the foundation of rigging
- Groups are objects
  - them move things
  - maybe many things
- allow one translate, rotate, and scale
- Also have origins
  - "pivots", centers



# Pivots?

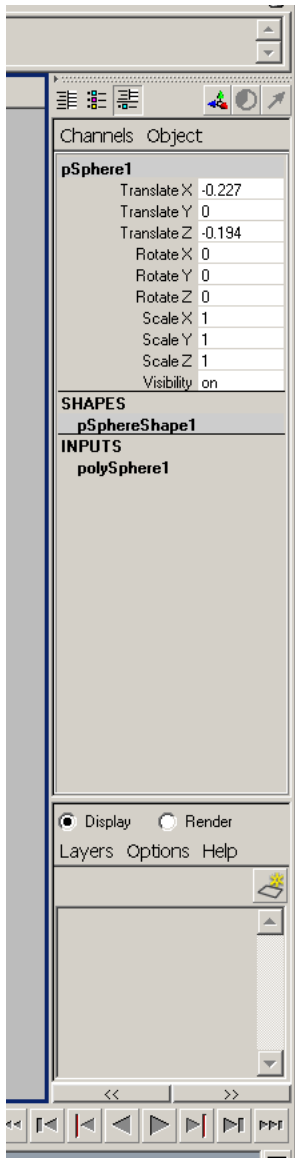
- Rotations happen around a center. That's the group's pivot
- Scales also happen from a center
- Get the pivot move manipulator by pressing "insert" (on Macs, "help")
- XForm w/ no arrows
- Don't animate pivots



# Uses of the Attribute Ed for groups

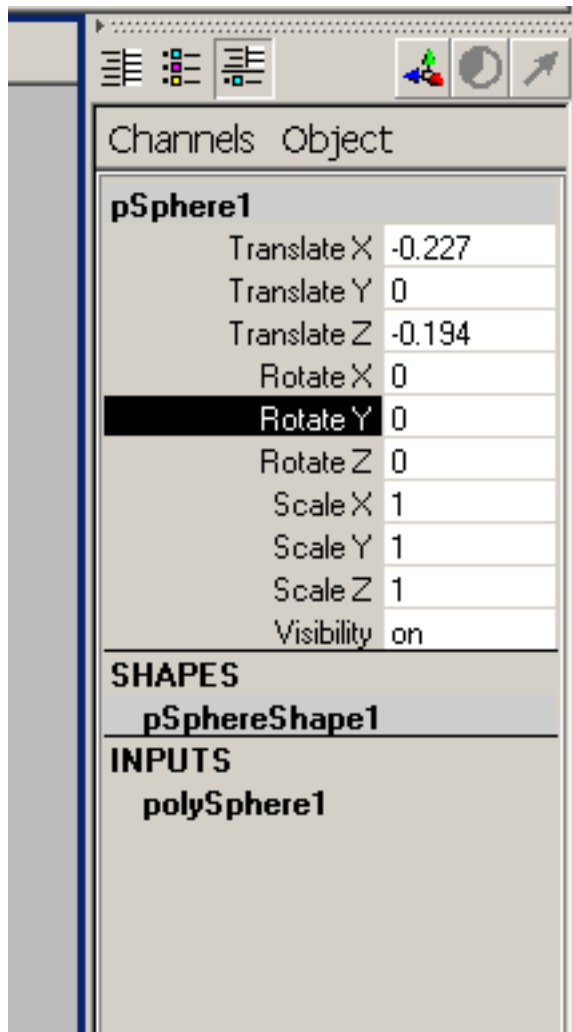
- Sanity checking
  - rotations of  $>1000$  degrees? Bad
  - scales of  $<0.00001$ ?  $>1000000$ ? Bad.
- Alignment
  - make all rotations match, exactly-- this is CG, right?
  - line things up– copy/paste translation values
- Setting animation keys on single attributes
  - Right-click on an entry box-- set key

# Channel box



- Hit control-a many times: attr ed, chanbox, attr ed, chanbox, AE, CB, AE, CB... heh.
- Vertical layout, more like outliner
- Not all attributes are visible-- only the "keyable" (animatable) ones
- Text entry, just like attribute editor

# Selecting Attributes



- Here, rotate-y is selected
- Middle-mouse drag in main screen now adjusts
- Right-click gives many, many options!
  - locking, unlocking, set key, delete connections, add attribute, expressions
- Work fast: MM-Drag, right-click, set key, repeat

# Locking Attributes

- Your first line of defense against accidentally ruining your model while animating at 4AM
- If you do not intend to move it, lock it.
  - right-click on the attribute name
  - select lock
  - unlocking is in the same menu
- Once an attribute is locked, it ignores attempts to set it

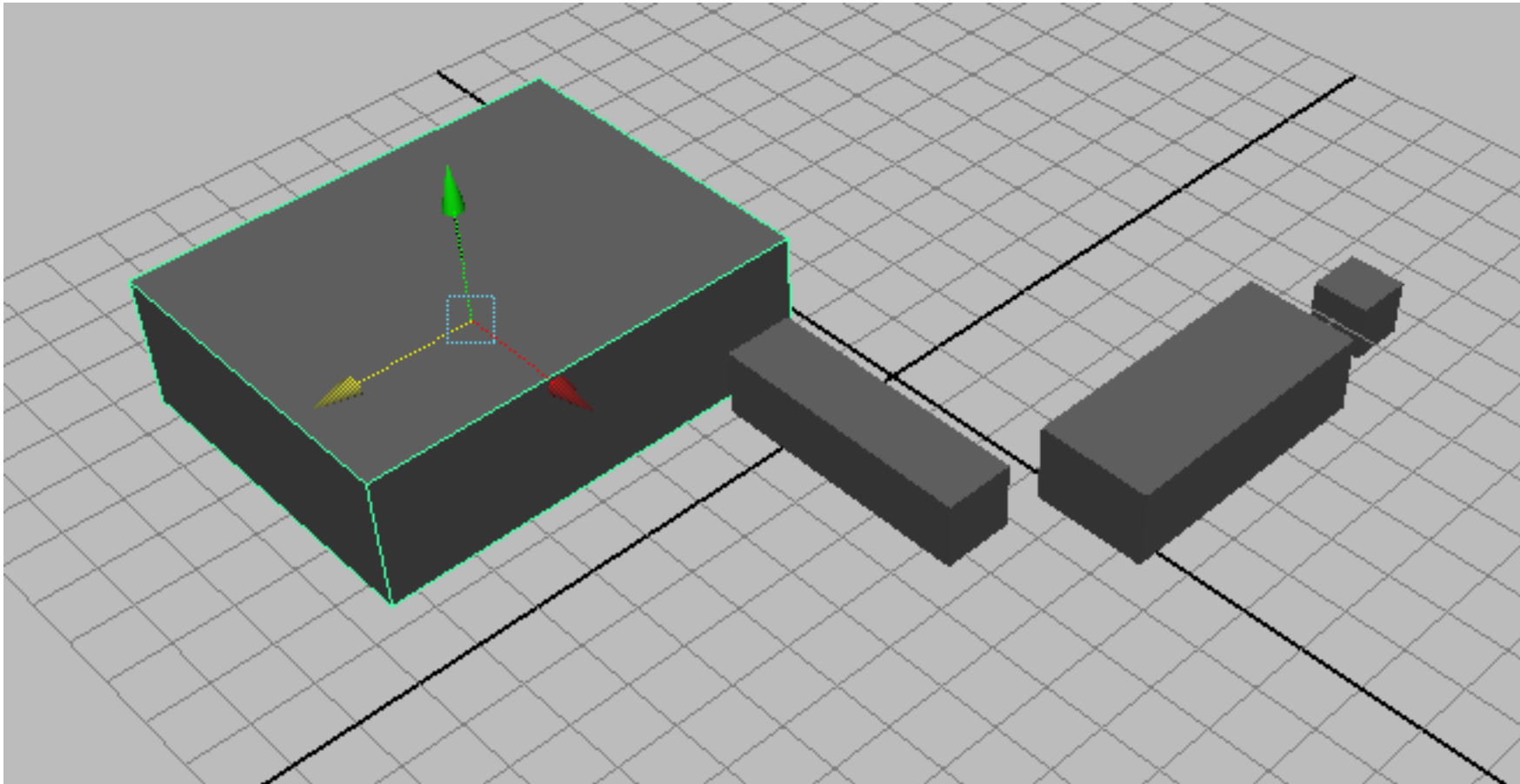
# Switching to Groups Basics

# Hierarchies

- Stacks of groups! Groups owning groups that own groups!
- The outliner is very useful for keeping track of this
- before you understand skeletons, you must be able to work with group hierarchies
- skeleton joints ARE groups

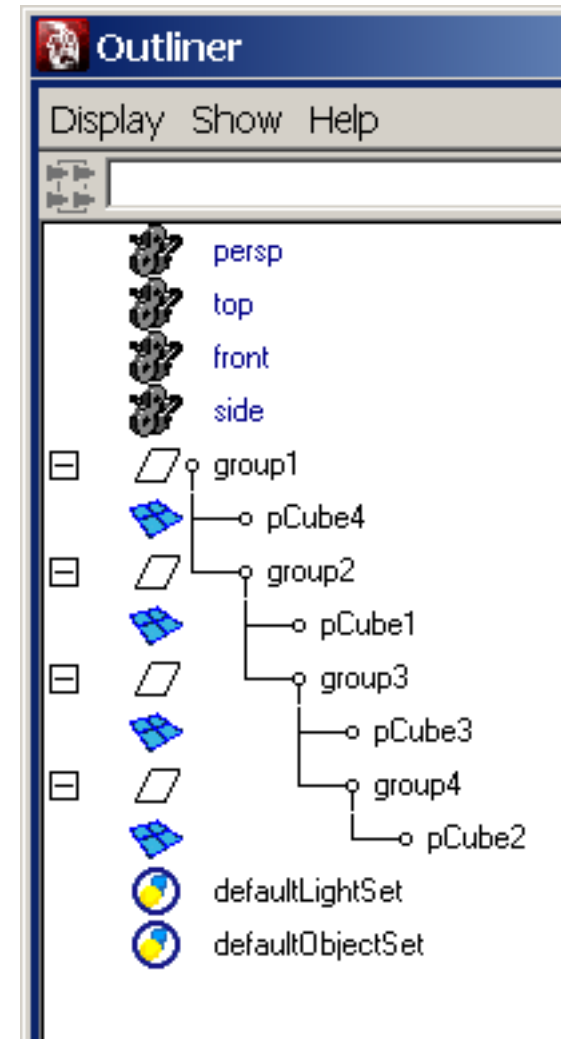
# Let's make an arm that bends

- Make some evocative boxes (hey, Popeye!)



# Making an arm

- Select one, Ctrl-g, select the next
- Open the outliner
  - (main menu->>window->outliner)
- Use middle-mouse drag to put the groups inside each other



# Try bending it

- The pivots are wrong-- the rotations all go around the origin
- However, the hand does follow the forearm, the forearm follows the upper arm
  - that's good!
- move the pivots to be on the joints
  - upper arm pivot goes on the shoulder
  - lower arm, elbow, hand, wrist

controls flows from the root to its children, but not the other way

- move the root, move all its children
- move the children, root just sits there
  
- this is called “forward kinematics”
- this is the default behavior

# Skeletons and joints

- skeletons are sets of joints
- joints are groups
  - each connected joint owns the next
  - only the “root” joint appears in the outliner
  - shift-click to open all
- the skeleton tool enables the rapid creation of many nested groups
- animation->skeleton->joint tool
  - since it’s a tool, you have to press enter to get out

# Note about the connections

- The balls are the joints– the pivots!
- the pyramids connecting them are just for illustration
- but if you click on the pyramids connecting joints, it selects the joint at the base: great.
- there are no “bones”-- only joints

# manipulating joints

- middle-mouse dragging on joints in the outliner works on joints exactly the same way it works on groups
  - unless you tell the outliner to alphabetize!
- Maya's terminology is backwards: to "parent X to Y" makes Y the parent of X
- Maya's tools for joint manipulation are weird and confusing– use the outliner.

# Make a spider skeleton

- Make 8 legs, a body and a head, out of joints
- use middle-mouse drag to join them up
- EZ

# Inverse Kinematics

- sometimes, you want children to move their parents
- touching a button with your fingertip
  - the tip is at the bottom of the hierarchy
  - the elbow, wrist, finger joints should follow

# IK solvers

- imagine your arm
  - you know where your shoulder is
  - you know where your hand needs to be
  - how should you spin those joints to get the hand to there?
    - you can solve that problem
    - so can an IK solver

# Two flavors

- Maya's IK solvers are not general
  - you can make anything with polygons– those tools are general
  - Maya's IK solvers are for arms and tails.
    - it knows what you want to do

# Planar IK

- is for arms
- your shoulder, elbow and wrist make a triangle
- you move the body to place the shoulder
- you move a marker to place the wrist
- Maya figures out how to bend stuff
- you'll also need to control the “pole vector”, which spins the elbow around

# Spline IK

- is for spines and tails
- you make a long chain, attach a Spline IK, and Maya makes a NURBS curve
- you control the tail/spine by moving that curve's CV's

# Make planar and spline IK

- put them on your spider

# planar gotchas!

- The pole vector is hard to get to, but you absolutely must.
- remember the flow of control: the IKS's are just tools for arranging joints
  - they can fight, if more than one is controlling the same joint
  - they can malfunction
  - you can misapply them– they really want to be arms.

# spline gotchas!

- they really want to be tails
- they don't do well for arms
- there should be more joints than cv's

# Think you're done? Hah!

- To be discussed
  - MEL expressions
  - constraints
  - binding
  - smooth binding
  - set driven keys
  - deformers
  - blend shapes
  - controls