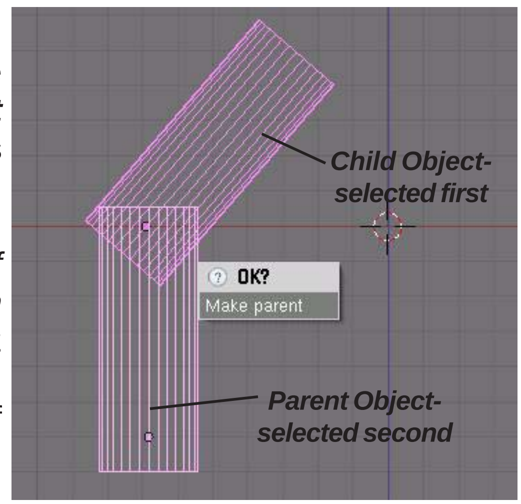
Chapter 14- Child-Parent Relationships

So far, we've talked about making and editing objects, making them look good and how to render and animate them, but how do we make things like humans or robots or anything else move about that have *several parts connected together*? This is where **child-parent relationships** become useful. It allows us to link things together without actually joining them. This allows the individual parts to move about, but still follow a "master" object.

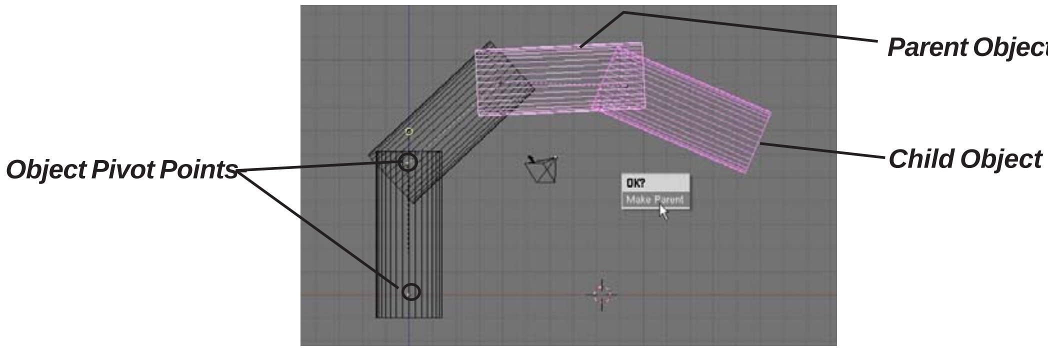
The concept of child-parent relationships is used in all animation programs and it involves an object assigned the role of a **child** and an object assigned the role of a **parent**. If the ***parent*** moves, rotates or scales, the ***child*** must do so too. **On the other end, a child can move, rotate or scale without affecting the parent.** An example would be: a *hand* is the child of the forearm while the *forearm* is the child of the upper arm and the upper arm is the child of the torso. Therefore, if the forearm moves or rotates, the hand must follow and if the upper arm rotates, the forearm *and* hand both must follow. If the torso moves, the entire arm must go with it. This is how you keep a body or machine from going to pieces!

In order to make child-parent relationships in Blender, you need to hold down the "***Shift***" key to select multiple objects. ***Select the child object FIRST, then select the PARENT object.*** The child object is always selected first. If you have a string of objects that need to be child-parented together (like the arm example), *you can only do 2 parts at a time so start at the end of the chain and do the hand and forearm first, then forearm to upper arm and so on*. After selecting the 2 objects, press *"**Ctrl**" and "**P**"* to make parent. You will see a dashed line drawn between the pivot points of the 2 objects. This shows a child-parent relationship.



<u>Quick Tip</u>: In order to delete a child-parent relationship, select both objects and press "Alt" and "P" to clear parent. This is good when you make a mistake!

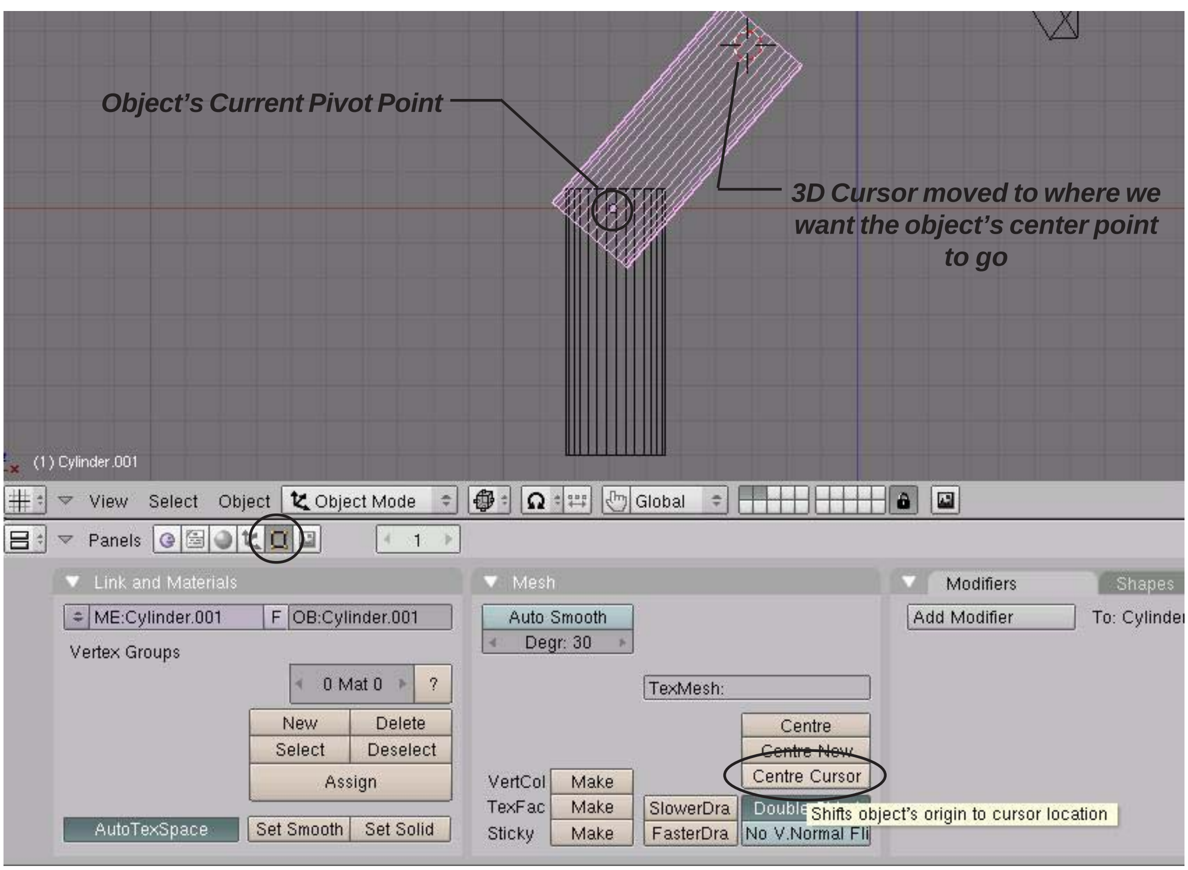
Look at the example on the next page. If we want to child-parent a few cylinders together to make a robot arm, create a cylinder and stretch it out in edit mode by moving one end of verticies. ***Remember to pay close attention to the object's pivot point***. If the object needs to pivot like an arm, you will need to keep the point at one end of the cylinder. *Always pay close attention to the object's pivot point in any case. It's easy to forget about it when moving verticies around in edit mode.* You can use the "***Center Cursor***" option in the edit button to locate the pivot to the 3D cursor's location. After you shape one cylinder, press ***"Shift" and "D"*** to duplicate it several times. Locate the cylinders and double check their pivot points. ***Moving the pivots after child-parenting them together will cause the objects to move.*** Start at the end and select the first 2 objects. Press ***"Ctrl" and "P"*** to make the relationship. Check it out to see if it's correct and go to the next set. In the next set, the previous ***Parent*** object now becomes the ***Child*** object. Make a simple animation to check the function.



<u>Quick Tip</u>: You can child-parent almost any object including cameras, empties and lamps.

Review of Moving Object Center Points

In the basic editing chapter, we discussed how to move center points of objects. You were also cautioned about moving an object while in edit mode because the verticies will move, *but not the object's center point*. To move the center point of an object, ***select the object***, place the ***3D cursor*** in the location you want the center point to go, and press the ***Centre Cursor*** button in the ***Edit Buttons***.



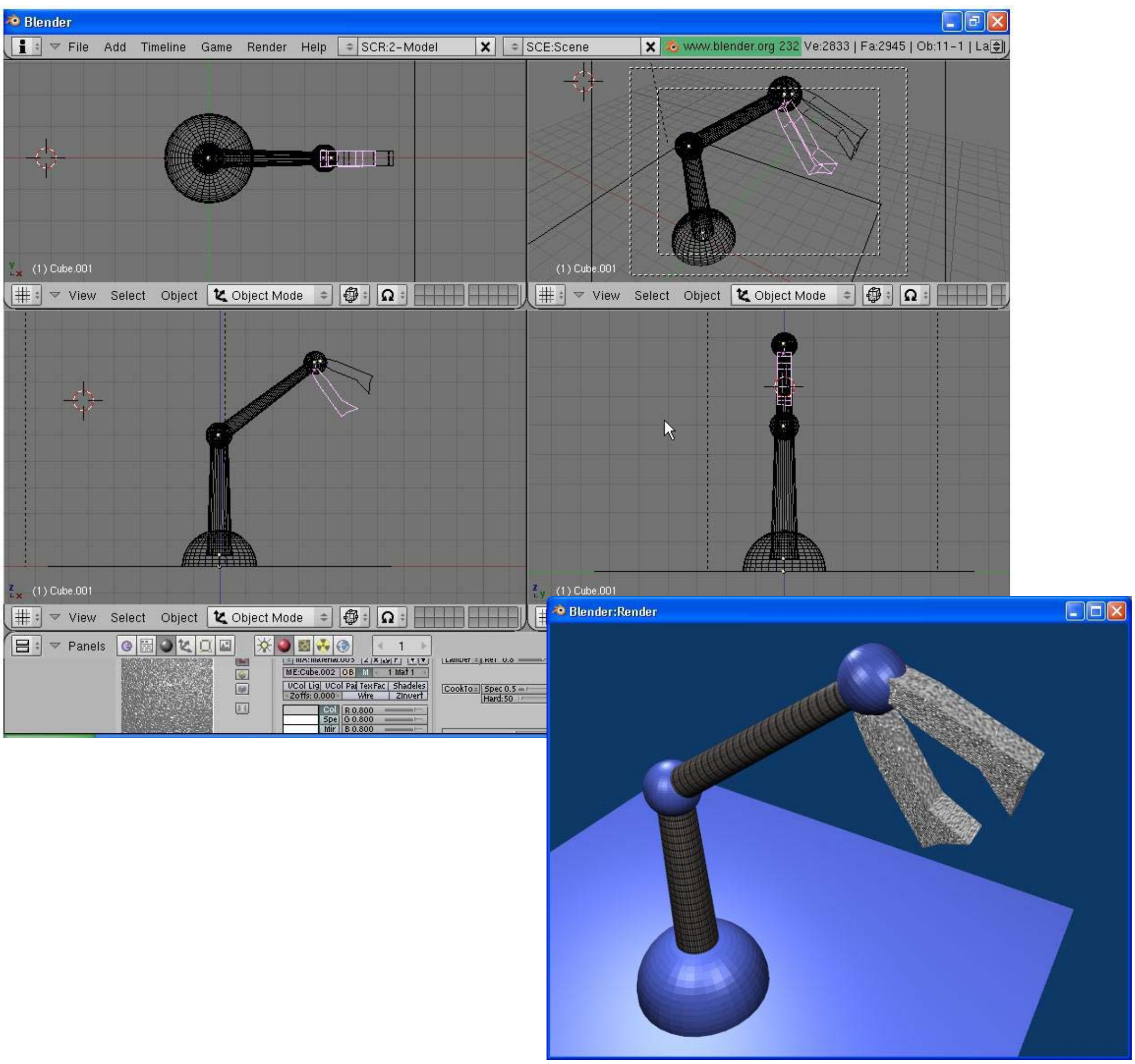
Creating a Robot Arm Practice Exercise

PRACTICE EXERCISE

Create a new Blender scene and set up the views any way you wish. ***Your job is to design a robotic arm that is child-parented together and animated***. Create all components using planes, cubes, spheres and cylinders. Place materials on all objects and develop a good scene with plenty of lighting.

After you create your scene, develop a 150 frame animation of your robotic arm moving in all directions.

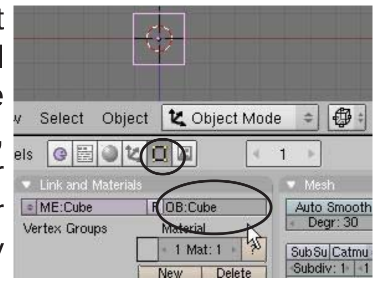
Challenge exercise:
Try to make your robot arm pick something up off the plane!



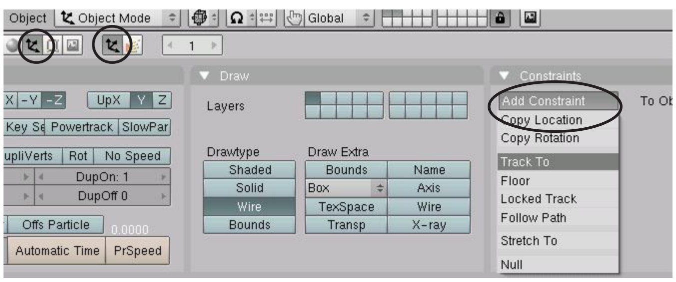
***** Call the instructor when finished*****

Tracking To An Object

There are times you want to “*constrain*” or “*follow*” a certain object in your scene. New constraints are being developed in Blender, but for now, we will just be talking about the most common one used to keep the camera focused on an object- *the “**Track To**” constraint*. The tracking constrain is useful in animating by saving you a lot of time and frustration trying to place location and rotation keys on the camera in an effort to try to keep your target centered. When used in conjunction with ***Paths*** (discussed in this chapter), you can create very smooth animation paths. ***Objects besides cameras can also be used with tracking.***



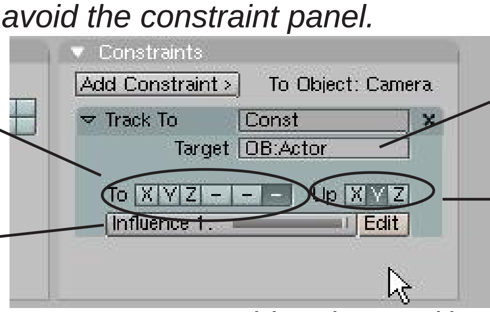
To get started with setting up a tracking constraint, Select the object you wish to use as your target and go to the *Edit* buttons. You will need to know the object name ***(OB:)***.If you haven’t gotten into the habit of naming your object, you may want to start doing so. Here, we’ll change the name of the ***Cube*** to ***Actor***. It doesn’t matter what you name it, but this is better than Cube or Cube.001 or Cube.003, etc which is what Blender will automatically name every cube you make.



Now, select the ***Camera***, go to the ***Object*** buttons and under ***Constraints***, hold down the mouse button on ***Add Constraint*** and select ***“Track To”***. In the ***Track To*** options panel, you will see a place for the ***Target OB:***. Type in ***Actor*** here. You will see a dashed line form between the camera and the plane showing the constraint between the two. If you are in camera view you will see there’s a problem- ***the camera doesn’t point to the plane!*** This used to never happen in Blender and may be fixed in newer releases but the problem deals with which axis and upward direction Blender wants to use. **To solve the problem, select the *“-Z”* in the *“To”* boxes and *“Y”* in the *“Up”* boxes**. Blender gives you more options so you can track in a variety of angles to the target object. That’s it- you now have a camera constrained to the cube. *A way to avoid the x,y,z problem when constraining a camera is to select the camera, then target and hit “Ctrl-T” to create a “TrackTo” relationship, much like child-parents and avoid the constraint panel.*

Point “To” Direction- X,Y or Z and -X,-Y or -Z. Usually a -Z direction

Influence Amount- All the way up for a solid tracking and lower numbers for a sloppy camera follow.



Target Object Name

Upward Pointing Direction- usually the “Y” direction

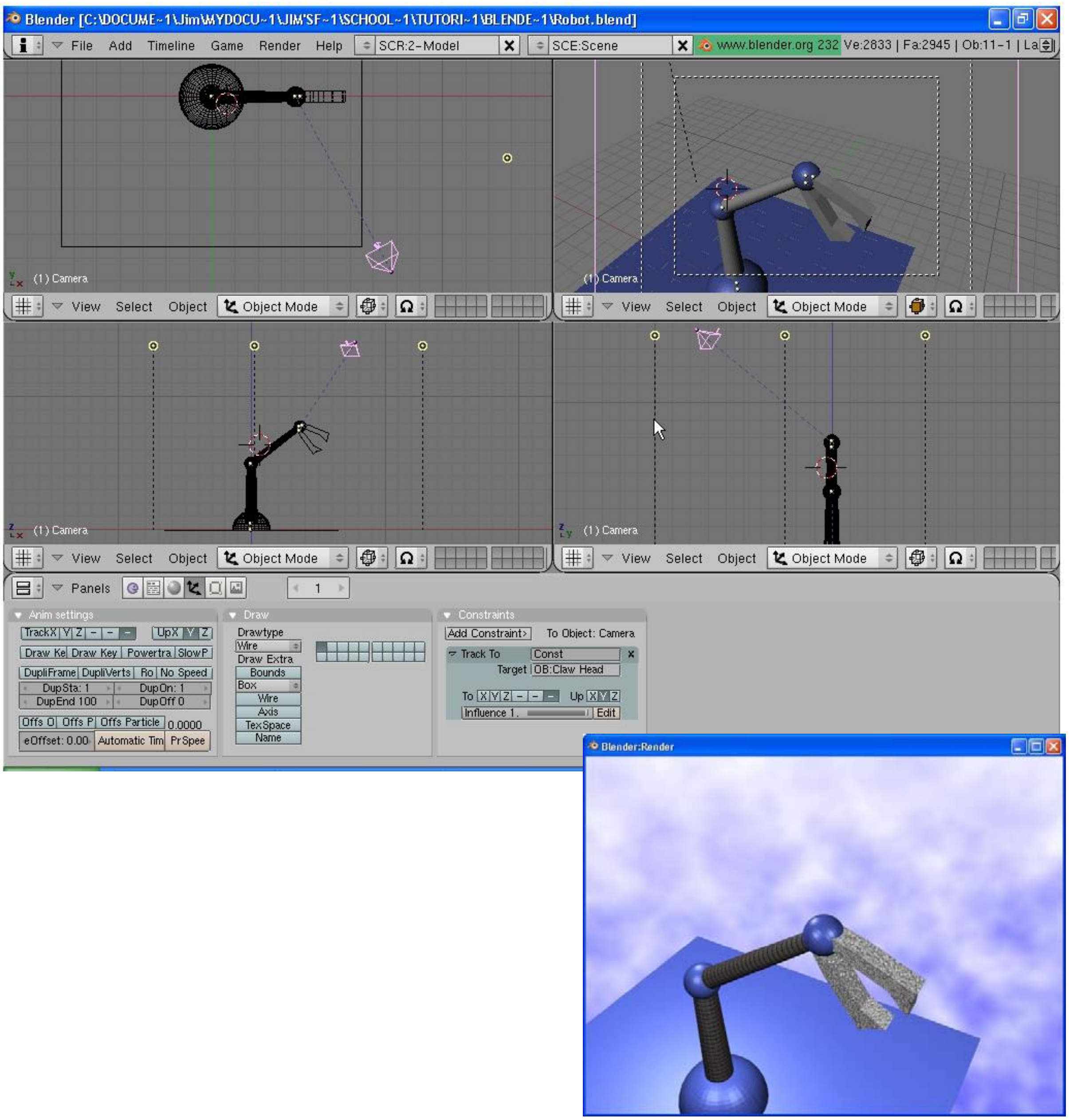
Sometimes it’s convenient to target an ***Empty*** object (created in the *Add* menu). This allows you to move your target around in your scene so the camera can focus on one object for a while, then move to something else by moving the target in that direction. You also have an ***influence*** option where the camera will track solidly to the object or allow some flowing of the camera.

Camera Constraints Practice Exercise

PRACTICE EXERCISE

Open the Robot Arm scene you made in the last exercise and add a camera constraint. You may target any part of the robot arm you like or create an Empty and target the camera to that. In the scene below, the camera was targeted to the gripper head.

After you create your scene, develop a 150 frame animation of your robotic arm moving in all directions with the camera also doing some movement.

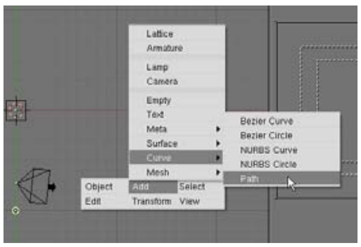


***** Call the instructor when finished*****

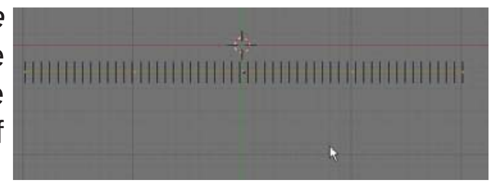
Following Paths and Curves

Sometimes you need to have an object flow along a smooth path in an animation. For example, it would be easier to have a spaceship flow along a line and angle and bank along that line then it would be for you to insert location and rotation keys throughout the animation. ***Paths and Curves*** are found in the same ***Add*** menu and can not only be used to create animation paths as discussed above, but can also be used to create extrusions. To create 3D extruded objects, you need to create a 2D sketch of a profile and a path for that shape to follow along. In this chapter, we will be working with both.

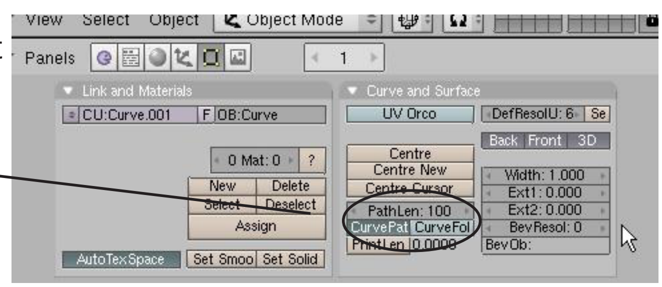
Following Paths



Your first step is to create a ***path***. Any type of **Curve** in the **Add** menu can be used as a path, but let's use the ***Path*** option. Hit the ***Space Bar***, select ***Add, Curve***, then ***Path***. You will then get a path on the screen in *Edit mode* with several points. Shape the path as desired, add more verticies through ***Subdivide*** if necessary and exit *Edit* mode.

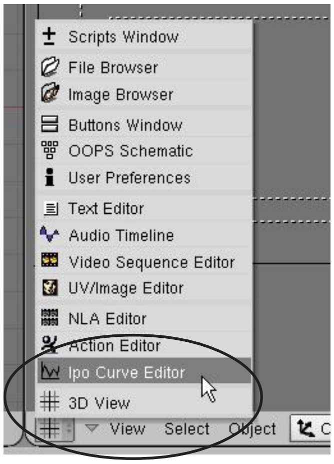


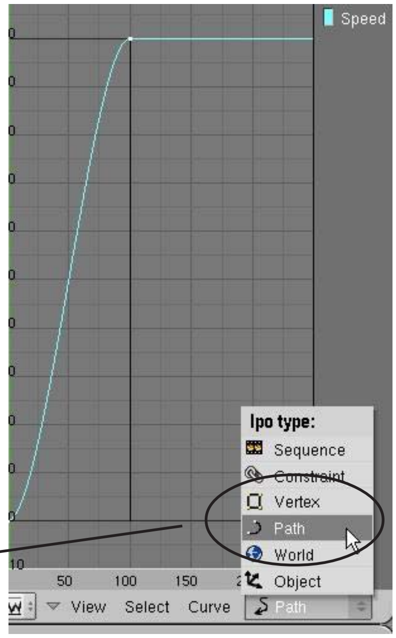
There are several ways to get the camera, object or lamp to follow the path. For now, we'll stick to the traditional way by creating a ***child-parent relationship***. Select the object first, then the path (the parent). With both objects selected, press ***"Ctrl and "P"*** to make a parent. You'll have 2 options: *"**Normal Parent**"* and *"**Follow Path**"*- select the *"**Normal Parent**"* option (even though *follow path* sounds more logical). You will see a dashed line between the 2 objects. Press *"**Alt**" and "**A**"* to see the animation along the path. In order to get the object exactly placed on the line, move the object and place it. Right now the object's animation is exactly 100 frames long and doesn't turn to follow the path. To correct this, make sure the path is selected and go to the ***Edit*** buttons. here's what you see:

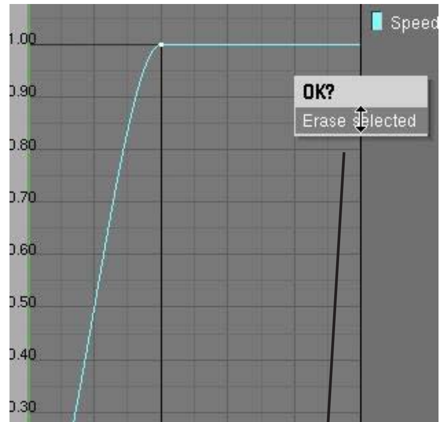


You basically have 3 options for the object following along this path. They are:
Path Length- frames it take to travel
Curve Path- will the object follow the path (already selected)
Curve Follow- makes the object curve along the path.
After you press the ***Path Follow*** button, the camera needs to be rotated and adjusted to the correct direction. After that, it will follow the path.

If you adjust the path length and hit *"Alt-"A"* again, you would expect the animation to change it's length, but it doesn't. There's a hidden "***Speed***" path that is hard to find the first time you try this. With the path selected, change the window type to the ***IPO window***. You will then need to change the ***IPO type*** to ***Path***. Delete the ***Speed*** track. See the next page:



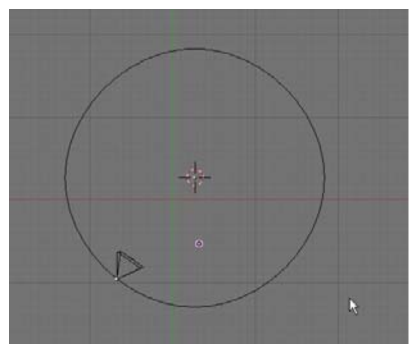




You will see the "***Speed***" track after you switch the ***IPO type to Path***.

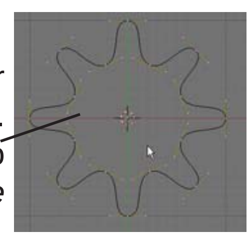
After you select and delete the speed path, the path length option will work in the Edit buttons.

Sometimes you don't want the camera to follow along the path, but look at an object as it flows along the path. **This is where you would want to use the *Curve Path*, but not *Curve Follow.*** Instead, you would put a ***Track To*** constraint on the camera so it looks toward an object as it moves along the path.



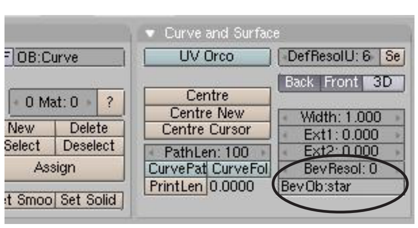
Other **Curve** objects can be used as paths also. For example, if you want a circular path, select the ***Bezier Circle*** option from the *Curve* menu. **The *Curve Path* button is not automatically pressed when you child-parent the object to the circle though, you must go into the edit button and do it manually.**

Using Curves for Extrusions



You can create a shape and extrude it along a path in Blender. For our example, we will shape a ***Bezier Circle*** and extrude it along a ***Bezier Curve***. First, create a ***Bezier Circle*** from the ***Add-Curves*** menu and shape it into an interesting object. Feel free to add more points with the *Subdivide* command. Second, create a ***Bezier Curve*** and shape it into some shape. Bezier shapes form differently and use spline points. Experiment with them to get the feel of working with them. Go to the *Edit* buttons and name both objects in the ***OB:*** block. Finally, select the ***Bezier Curve*** and go to the *Edit* buttons. You will see a **BevOb:** box. Type the name of the circle there. You will see the shape extruded along the curve!

*You can convert the new shape into a mesh to make it easier to work with by pressing the **"Alt and "C"** keys*

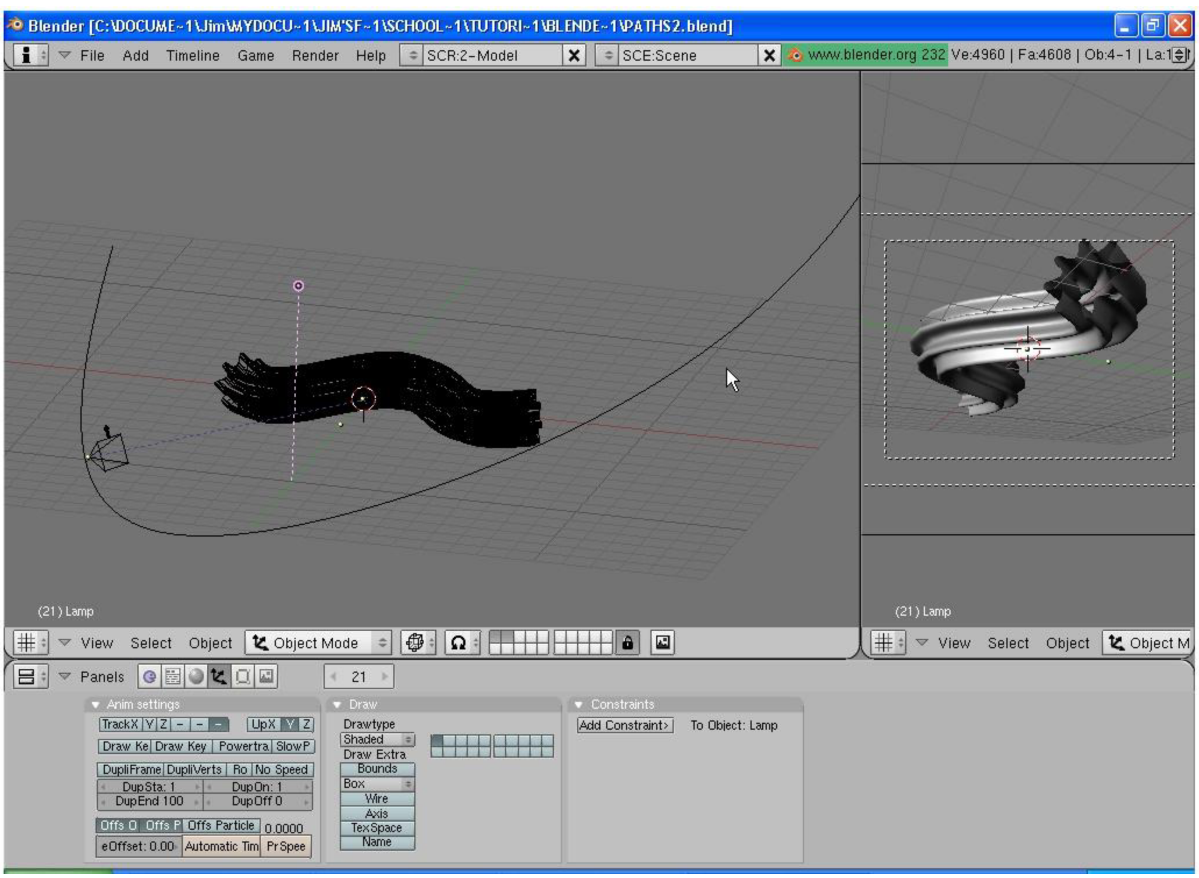


Create a new Blender file and name it ***Paths***. Develop a path for your camera that goes around a shape that you extruded along a curve. Use the extruded shape as the target for the camera so that as the camera flows along its path, it is always focused on the object. *You may need to adjust the object's center point in order for the camera to properly focus on the object. (refer to the basic editing chapter)*. Add materials to all objects. If you would like to close your extruded shape (not open on the ends) try this: Convert the extrusion into a mesh ("Alt-C"), go into Edit mode and select the end verticies. Type "**E**" to extrude, then immediately type "**S**" to scale. Scale the new verticies to close off the end. If you would like it to look like a pipe with some wall thickness, enter Edit mode and select all verticies. Press "**E**" to extrude and "**S**" to scale slightly.

Save a 100 frame animation when finished.

Challenge exercise:

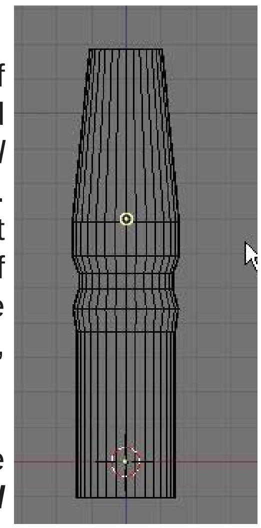
After you do the required exercise, make a new one. Before you extrude your shape along it's curve, duplicated the curve and use it for the camera path. Place the path directly in the middle of the extruded shape to make the camera flow through the "tube".



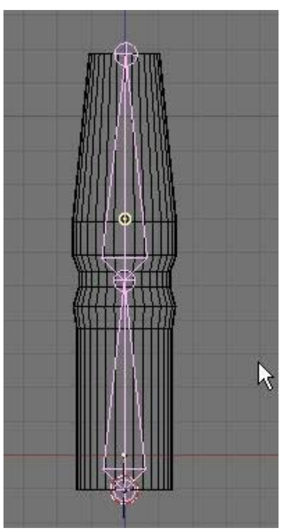
***** Call the instructor when finished*****

Using Armatures to Deform Meshes

Blender's animation capabilities are great for most object animation except when you want to animate something bending like a person in motion or a tree bending in the breeze. This calls for a mesh to deform which can't be done with traditional modifiers. We can deform a mesh in 2 ways in Blender. One way is to create a skeleton and have it deform a mesh ***(armatures)*** and the other method is to move the mesh verticies in edit mode and create sliders that deform the mesh ***(relative vertex keys)***. This chapter deals with creating armatures. The armature feature in Blender is constantly under development. For this discussion, I will stick with the fundamentals. More information can be found at www.blender.org or at www.elysiun.com.



The first thing you need to do is create a mesh that has a few groups of verticies where you would like the object to bend. Any mesh will work and to get additional verticies you can either **extrude or subdivide**. *Be careful not to create too many verticies*. It may slow your model down considerable. Let's use a cylinder to create an arm. I will use a cylinder set at the default divisions of 32. Next, I will change views and box select the top set of verticies and *Extrude* them up. I prefer to use extrude rather than subdivide to keep the vertex count down as low as possible. As I extrude the verticies, I am also using *Scale* to shape them.



Next, place the 3D cursor directly at the bottom of the shape you just made. Hit the ***Space Bar***, then ***Add*** followed by ***Armature***. You will immediately see a bone begin to form at the cursor location. Type "**G**" to *grab* the top of the bone and lengthen it to a desired size. Move your cursor up to lengthen the bone and click where you would like the joint to be.To create another bone at the top of the first one, press "**E**" to *extrude* another bone from the first one. If you run out of room to drag the mouse up, just click wherever and hit *"**Esc**"* to stop making bones. As like all other objects you create, you are in edit mode. To adjust the top bone to get it in the correct position, RMB click on the top of the bone. The small circle highlights. Press *"**G**"* for grab to move it. When finished, press *"**Tab**"* to exit edit mode. Double check the armature to make sure that the ends and joint are well aligned. To add more bones, hit **Tab** to enter edit mode again and **Add Bones** with the **Space Bar**.

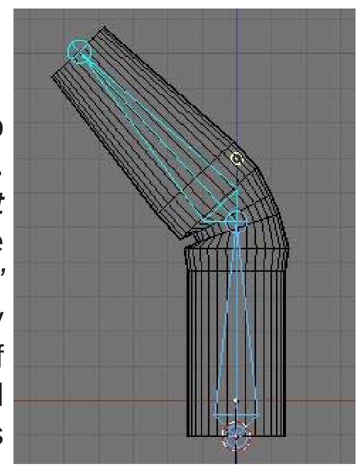
Your next step is to create a ***child-parent*** relationship between the mesh and the armature with the mesh being the ***child*** and the armature being the ***parent***. Hold down the *"**shift**"* key and select the mesh first, then the armature. Press ***"Ctrl" and "P"*** to make parent. Select the option to *"**Use Armature**"* since the armature is both of the bones together. You will then get some options as to how to create the vertex groups that will move with each bone. Use the *"**Create**

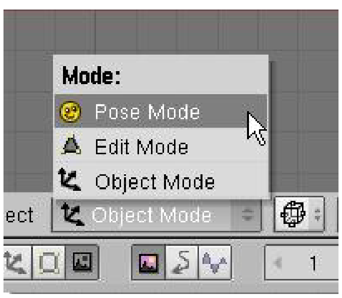
From Closest Bone" so the computer will figure it out. Sometimes this will not work if verticies are close together (like several fingers on a hand). Verticies from one finger may get grouped with bones from the finger beside it- not a good effect! We will discuss creating your own vertex groups later. That's it! Time to test your model!

To create entire skeletons or other complex armature structures, you can do the following:

> *Join meshes together to form one mesh for an entire body*. This can be done using the boolean "***W***" key or by just selecting them all and pressing "***J***" to join. Make sure they are all set up with materials and textures before you do this and some of the textures may need readjusting. *This must also be done before you child-parent any of the meshes to an armature*.
>
> Create all of your individual armature sets and join them together as you do meshes or work with child-parent relationship with the bones.

<u>*To animate you armature:*</u>

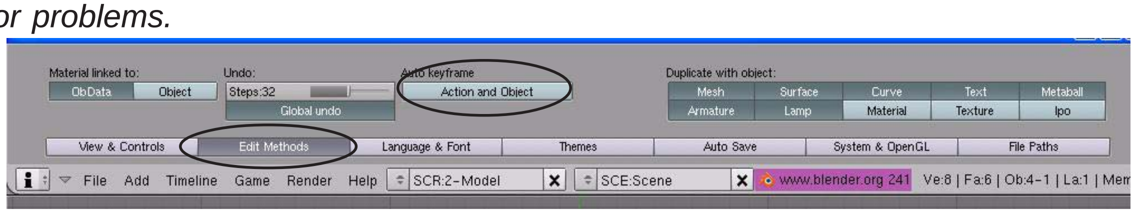




It's time to animate our "arm" model. To do this we must get into "***Pose Mode***". Change the Mode option from ***Object Mode* to *Pose Mode***. This can also be done by pressing *"**Ctrl**" and "**Tab**"* together. Select a bone to work with by RMB clicking on it. Type "**R**" to rotate it. If everything went well, the mesh should move with the bone. Place animation keys in the various frames as before to create an animation.

<u>*Automatic Keyframing with Armatures:*</u>

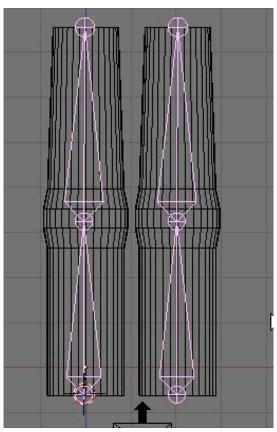
Placing animation keys on a complex armature system can be time consuming and very easy to miss a bone in a frame when you need to place a rotation key on 20 bones. That's why there's an automatic keyframe option in the top ***User Preferences*** window. Pull down the top menubar to expose the setting. Select *"**Edit Methods**"* and turn on the **Auto Keyframing button**, *"**Action and Object**".* This will automatically place keys on every bone that has been moved in a particular frame. *Remember to turn it off when finished or it can cause some major problems.*



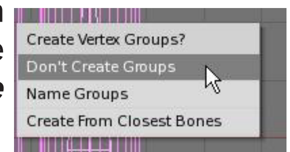
<u>Quick Tip</u>: *Even if a bone isn't moving on a particular frame, move it slightly so the automatic keyframing places a key on it. Otherwise, it may move when you don't expect it to because it was missing a key. Experiment with the features to become familiar with them.*

Creating Vertex Groups:

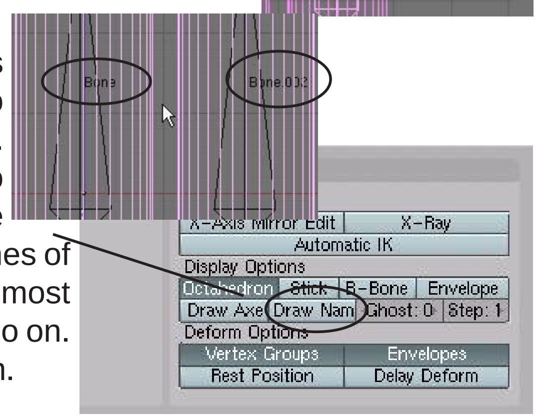
As mentioned before, sometimes when verticies are close together, Blender may have a dificult time creating clean vertex groups automatically. You will need to define the vertex groups manually. First, create your mesh and armature, then ***child-parent*** them to *"**Use Armature**"* as before, but then select *"**Don't Create Groups**"*. This will cause them to be parented, but not defined. The option "***Name Groups***" will create the vertex groups needed, but not assign verticies. This option may be better than "*Don't Create Groups"* since the computer will name the groups for you. For now, we will discuss naming and making your own groups.

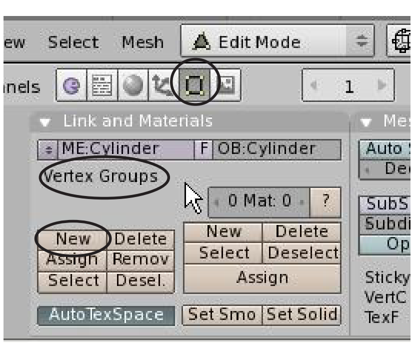


For this example, I've created a mesh we'll call *Finger*. I then created an armature and duplicated both to create 2 fingers. The next step is to ***Join*** (*"Ctrl" and "J"*) the 2 armatures together, then do the same for both meshes. ***Child-parent*** the mesh to the armature, select the *"**Don't Create Groups**"* option.



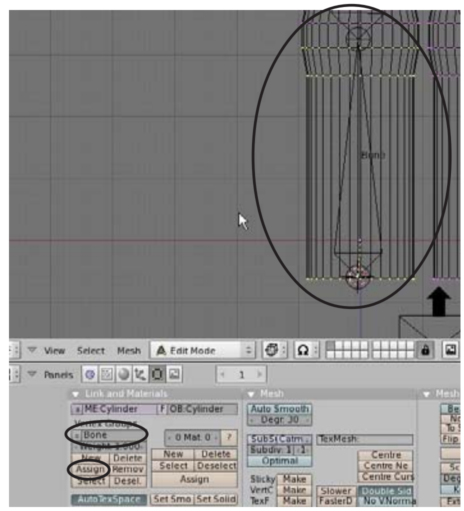
Now we need to see the names of each bone in the armature so we can assign verticies to them. Select the armature, then go to the **edit buttons** and find the "***Draw Name***" button. Pressing it will cause the names of each bone to be displayed on the screen. The will most likely show up as Bone, Bone.001, Bone.002, and so on. You will need to place these names exactly as shown.





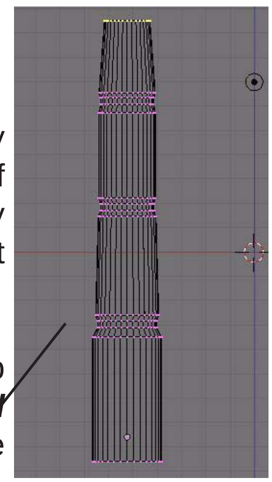
Now select the ***mesh*** and go into ***Edit Mode*** (*Tab* key). In the **edit buttons**, you will notice a group of buttons for creating ***Vertex Groups*** (this block of buttons will only be displayed if you're in edit mode). Select the *"**New**"* button and you are ready to create a new group of verticies for a bone. We will create a group for ***Bone*** (the lower left one in our model). Type this in where the word *"**Group**"* is written. Remember that each bone will need a group and the group name ***must*** match the bone (i.e. Bone.001, Bone.002).

Our next step is to select the verticies that need to be assigned to that bone. *If a group of verticies is right at the joint, they need to be selected for both bones*. After selecting all the verticies that belong to that bone, press the "***Assign***" button. You've now made a vertex group for that bone. Do this for all bones. When finished, exit edit mode and select the armature. Press "**Crtl**"-"**Tab**" to enter ***Pose Mode*** and test out the armature. If you need to modify any groups, you can go back into edit mode on the mesh to make corrections.

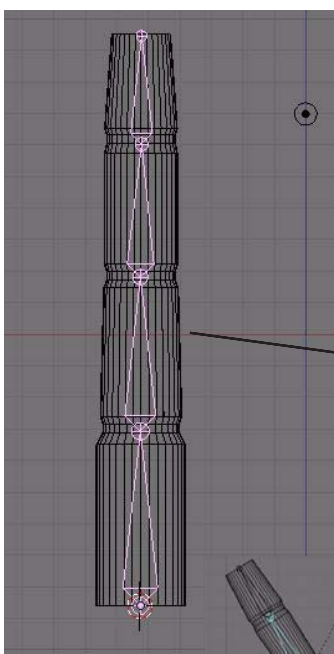


Remember we discussed the benefit of selecting the "***Name Groups***" option when parenting the mesh to the armature? It will save us a step in naming all of the vertex groups.

Using Inverse Kinomatics (IK)



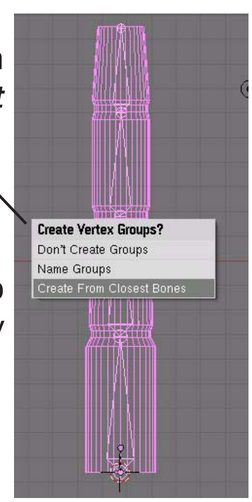
Inverse Kinomatics is used when you wish to manipulate a skeleton by simply grabbing (**G** key) the end bone of a chain and moving it with all of the connected bones following along. Inverse Kinomatics is constantly under development and has a lot of options available depending on what you want them to do. *Visit www.blender.org for more details.*

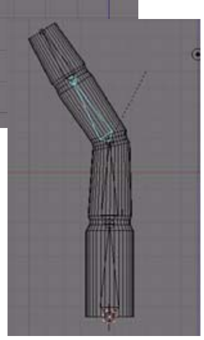


For our example, make a ***Cylinder*** in the top view, switch to the front view and ***Extrude and Scale*** the object several times to look like the example.

Next, move the cursor to the bottom of the mesh and ***Add*** an ***Armature***. Use "**G**" to grab the top of the bone and move it to the first joint. Press "**E**" to extrude another bone to the second joint. *Repeat two more times to create a 4 bone armature*.

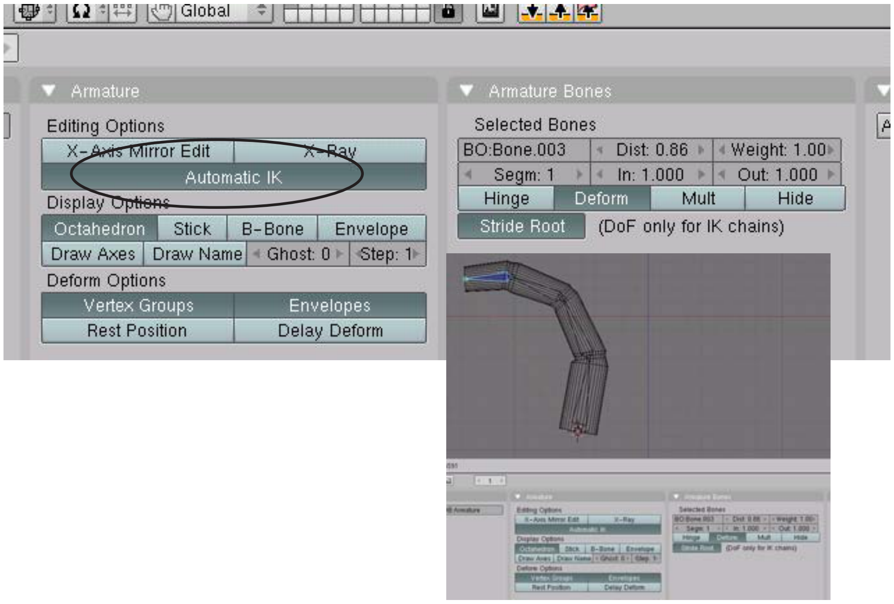
Child-Parent the mesh to the armature with ***"Crtl"-"P"*** and select the "***Create From Closest Bones***" vertex group option.





Go into ***Pose Mode*** and rotate the bones to check the vertex groups to make sure they were assigned properly.

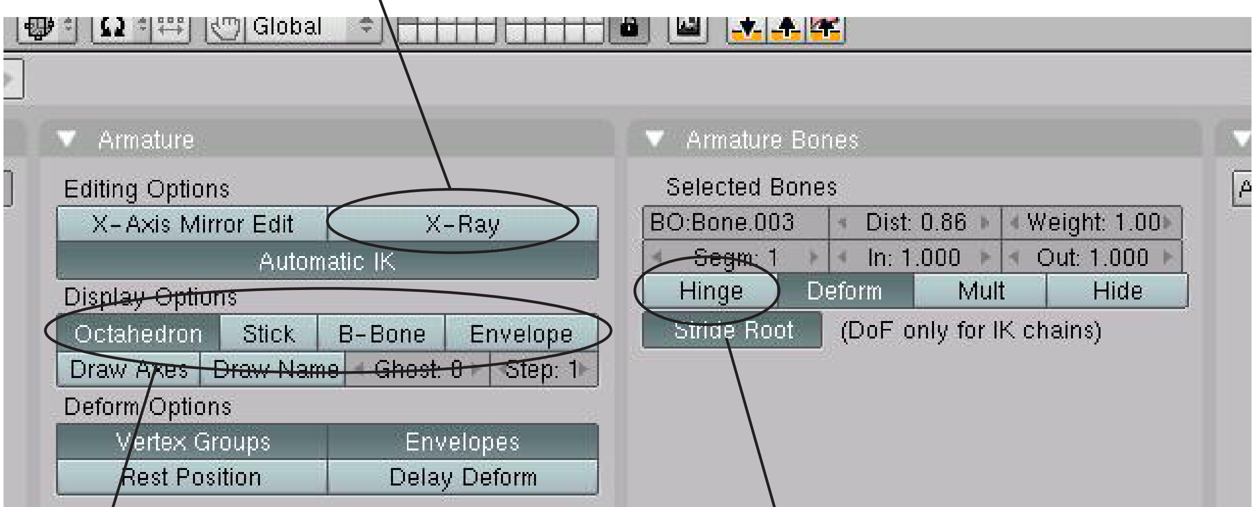
If everything checks out with pose mode, exit pose mode and go back to object mode. In the ***Edit buttons***, with the armature selected, find the ***Automatic IK*** button and select it. To see the IK in action, go back into pose mode, select the top bone and use the "***G***" key to move the bone. All of the bones should deform equally as you move the top bone around. This could be a good feature to use to animate fingers curling or grabbing.



Additional Armature Settings:

Here are a few more options that have been added to make armatures a little easier to work with:

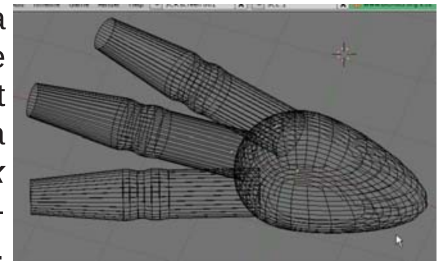
X-Ray- to see the armature when the meshes are being displayed in Solid



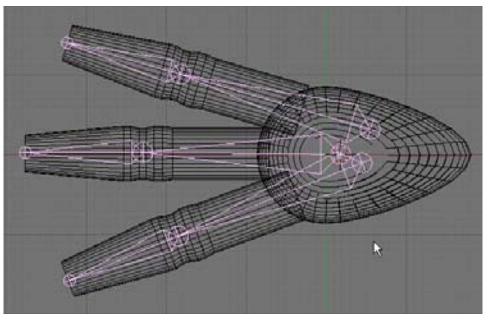
Different ways to display the armatures

Hinge- used to keep a bone from rotating when a parent bone in the chain is rotated. Good for engine parts

Creating a Skeleton Practice Exercise

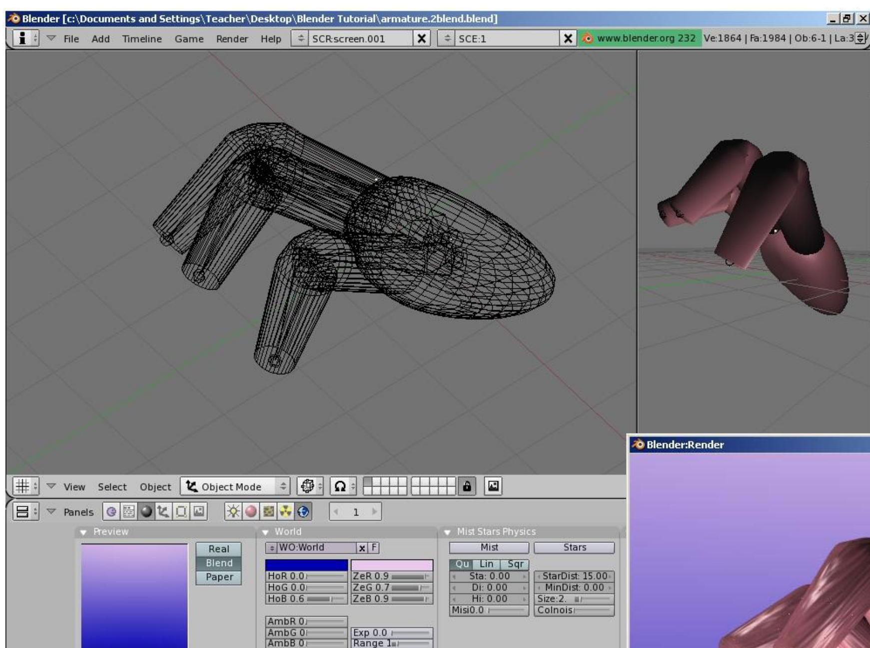


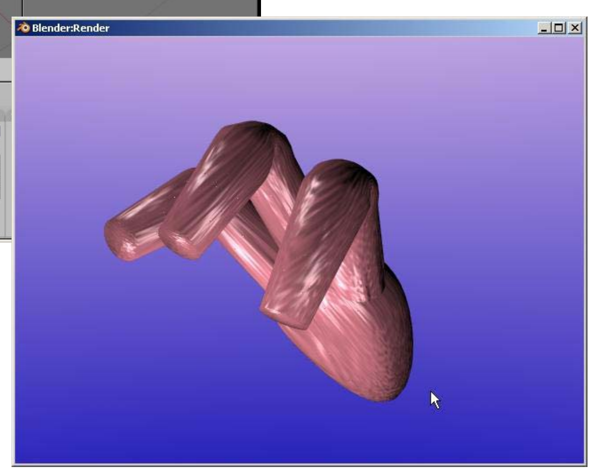
Create a new Blender file and name it _Hand_. Create a UV Sphere for the hand and a cylinder for the finger. Use ***Extrude*** to lengthen the finger and provide verticies at the joint (*one joint only*). ***Duplicate*** the finger to make a total of 3. Shape the sphere with ***Proportional Vertex Editing*** *("O" key)*. Join all meshes together. Create a 2-bone armature for each finger and ***Join*** them together. Child-parent the mesh to the armature set and create vertex groups (you may or may not be able to use the automatic setting). Place a material on the mesh. ***Add lighting and create a 100 frame animation of your scene.***



Challenge exercise:

Try to use armatures to animate some other object. Try a simple body that walks.



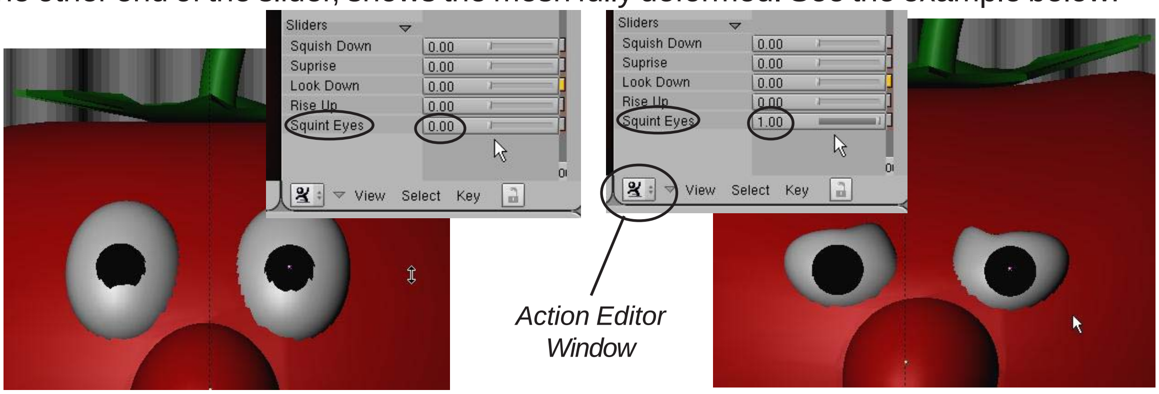


***** Call the instructor when finished*****

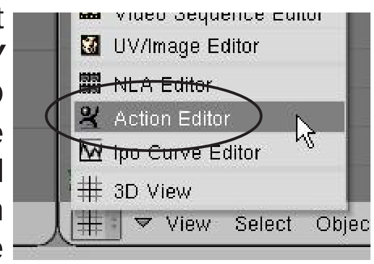
Chapter 17- Relative Vertex Keys

Creating Mesh Keys

We've discussed deforming a mesh with an armature, but what if you want to deform a mesh in other ways like have it flatten, move a mouth, blink an eye, etc. and have a way of repeating that motion whenever needed? Some of these things can be done with armatures, but sometimes it's easier to set up a **slider** that at one end, represents the mesh in one form, and at the other end of the slider, shows the mesh fully deformed. See the example below:



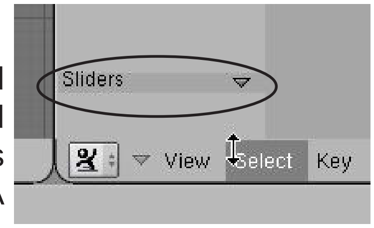
Mesh deformation using Sliders in the ***Action Editor Window*** can be a difficult process because it requires you to shape your mesh in edit mode moving verticies. With practice, this can become a worthwhile tool that will enable you to make quick and high-quality animation like the professionals do. If you notice in the above example, there are several sliders that cause different motions. By using combinations of them, a wide variety of motions can be produced *(for example, surprise and squint will combine the motions)*. These are great tools for making a character speak, blink and show expression. I'm waiting for someone to develop sliders for armatures to create easy motions.



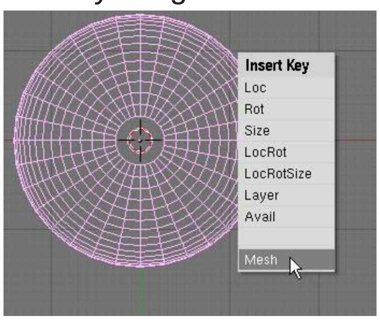
The first step in creating **Relative Vertex Keys (RTVs)** is to start with a mesh you wish to deform. For our case, we'll create a ***UV Sphere*** set at the default segments and rings of 32. Split the 3D window into 2 viewports and set the right-hand viewport to the ***Action Editor Window.*** This is another type of animation control window where animation keys you create are shown as marks on the timeline. Keys can be duplicated and moved here. When we create our RVKs, they will be shown here as slider bars.

Let's go back to the left-hand viewport (still set to the 3D view window) and begin creating RVKs. ***Unlike normal animation that requires you to move to different frames along the timeline, we will be creating all our different sliders and mesh deforms on frame one***. After the sliders are all created for our mesh, then we will use them in the Action Editor window along the timeline. With the sphere selected, make sure you're **NOT** in edit mode, but in object select mode. (*Tab* key). Hit the ***"I"*** key to insert a ***Mesh*** key. The first time you hit the ***Mesh*** key for that object, you will get an option for ***Relative Keys*** **or** ***Absolute Keys***. Select the ***Relative Keys*** option. ***You won't get this option in 2.41. You will automatially be making Relative Vertex Keys***.

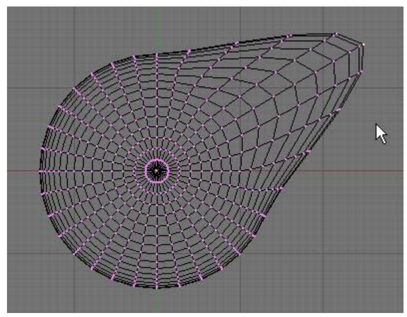
Using Action Editor Sliders



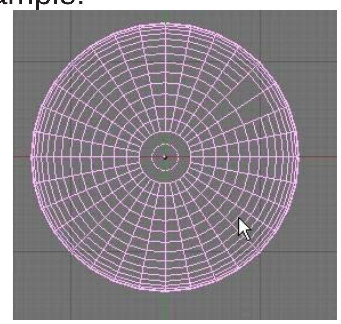
Once you hit the Relative Keys button, the word ***Sliders*** is added in the ***Action Editor*** window. However, no sliders have been added to the list as of yet. Now, with your cursor in the 3D window, press ***"I"*** to insert again a **second time** and select *"**Mesh**"* again. A "***Key 1***" slider show up in the list. It's now time to define that slider. Now, go into ***Edit Mode*** ("*Tab*" key) and modify the verticies however you want. When you exit edit mode, the slider will now deform the mesh. ***To create another slider, Insert another mesh key while NOT in edit mode***, then hit "**Tab**" to enter edit mode and modify the mesh. When you again exit edit mode, the slider will be set. Here's an example:



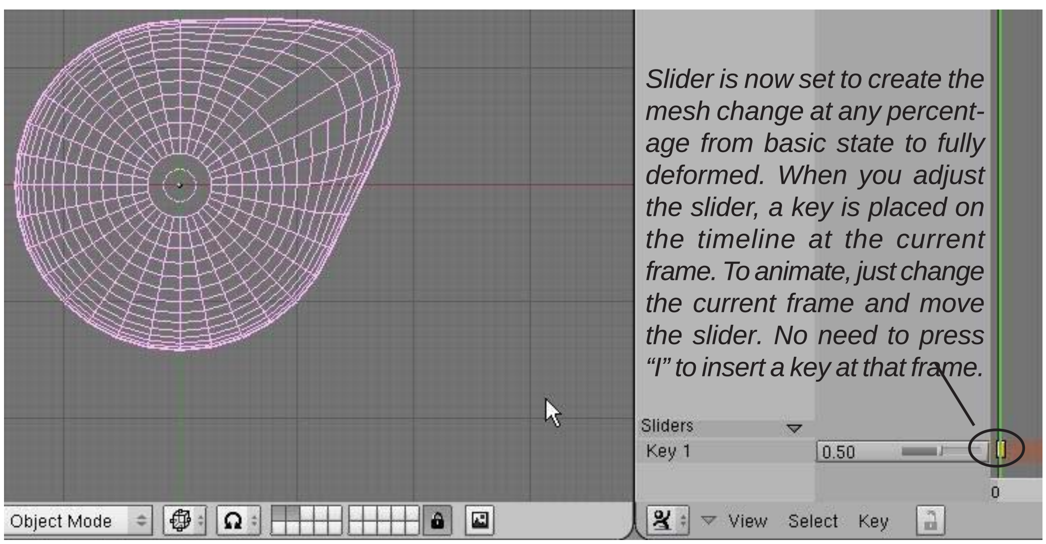
In object mode, press "I" to insert a Mesh key



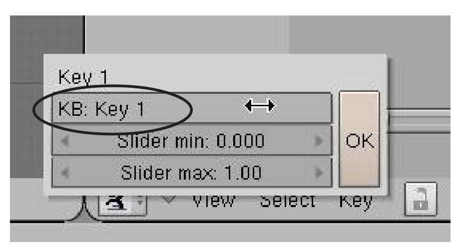
Enter edit mode (Tab) and modify mesh



Exit edit mode (Tab), mesh goes back to basic state, slider now functional.



The basic thing to remember about RVKs is that in order to create the slider, ***You must insert the key in object selection mode, then enter edit mode (Tab) to modify the mesh.*** When you exit edit mode, the slider is set. Repeat the process to create all your sliders.



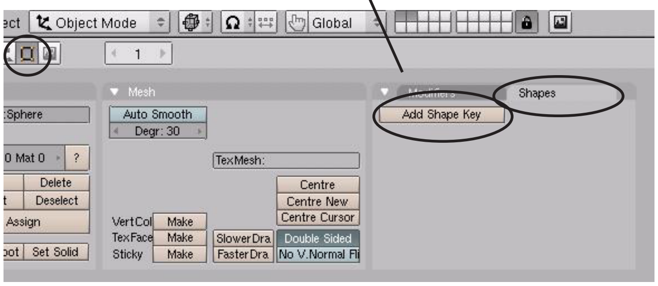
In order to name your RVK sliders, place your cursor over the name of the key you wish to change and press "**N**" for name (*or click on the slider*). This window comes up where you can name your slider. You can also adjust the min. and max. of your slider.

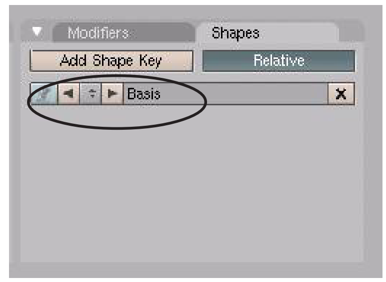
Chapter 17- Relative Vertex Keys

New RVK Creation Panel for 2.40

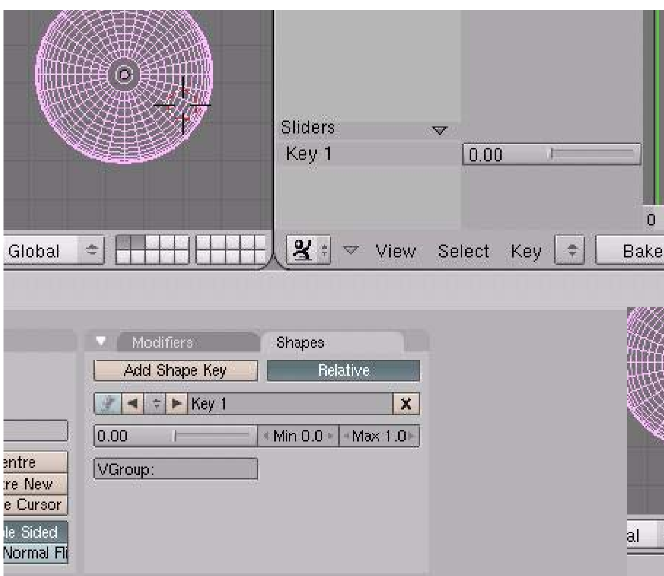
With the release of version 2.40, you have a new way to create *Relative Vertex Keys* in the *Edit* buttons. Instead of pressing the **"I"** key to create a mesh key, follow these steps:

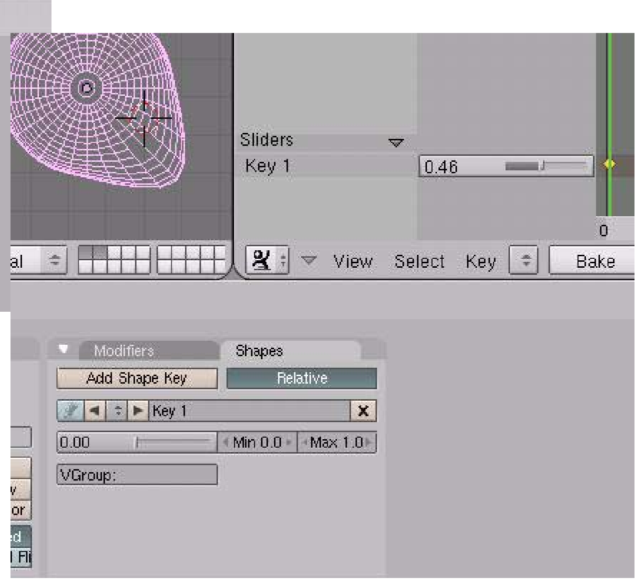
Select the mesh you wish to apply RVKs to, then, in the ***Edit*** buttons, find the ***Shapes*** panel. Click the "**Add Shape Key**". *This is the same as you hitting the "I", then mesh key the first time, but not creating any sliders in the previously discussed process.*





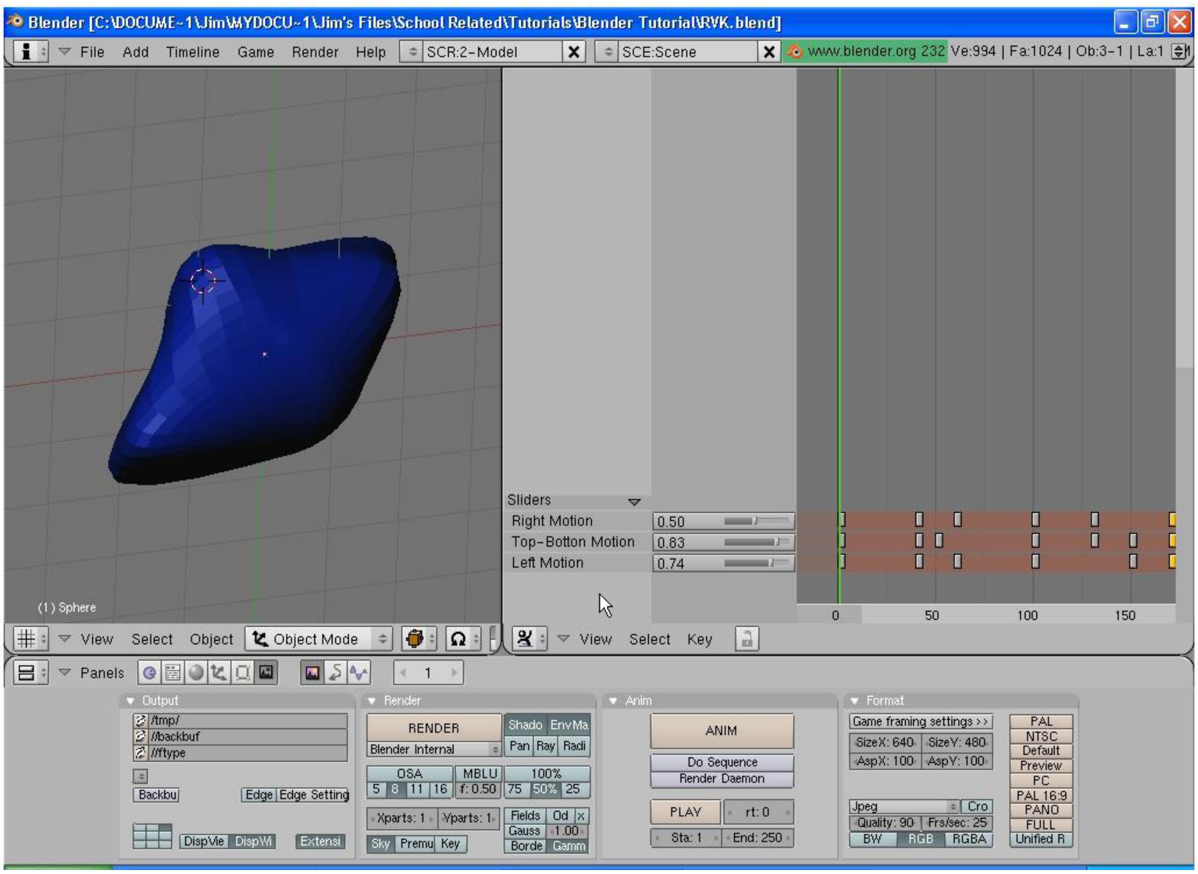
Now press the "**Add Shape Key**" again. The first RVK Slider will be displayed. At this point, enter **edit** mode and modify the mesh. When finished moving verticies, **exit edit mode** (*the verticies will return to their original positions*) and the slider will now be operational. They will also be displayed in the **Action Editor** window.





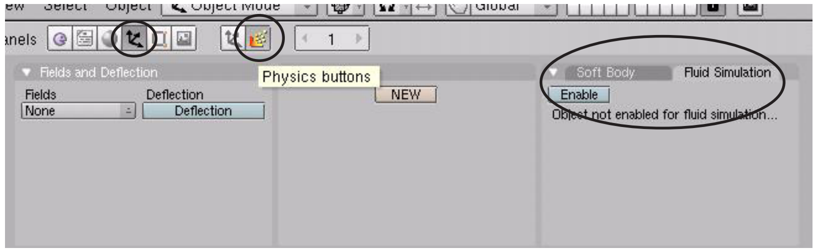
You will see that this panel only replaces what we did using the "I" to insert mesh keys as discussed on the previous pages. It just gives us another option to use.

Create a new Blender file and name it *RVKs*. This exercise can be as simple as deforming a sphere with RVKs or as complex as trying to create a face and make it talk or show expression. Create a scene with adequate lighting, world settings and materials. Create at least 3 RVK mesh sliders and use them to create a 150 frame animation. ***Create an AVI file when finished.***

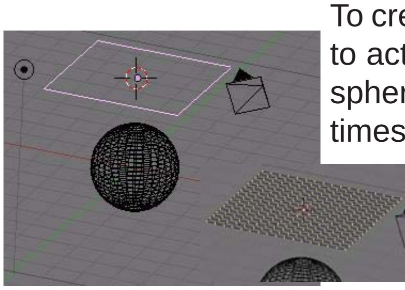


***** Call the instructor when finished*****

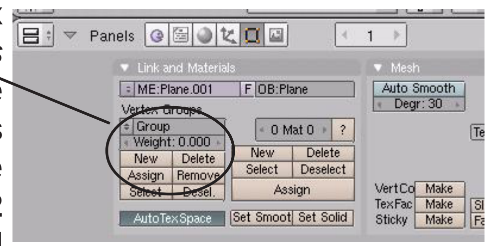
Creating Fabric and Wave Effects

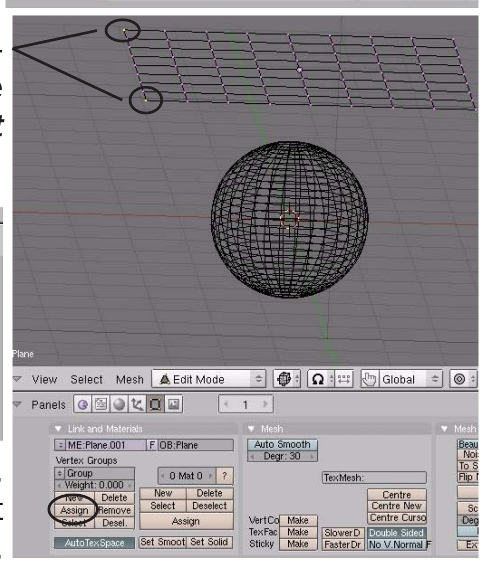


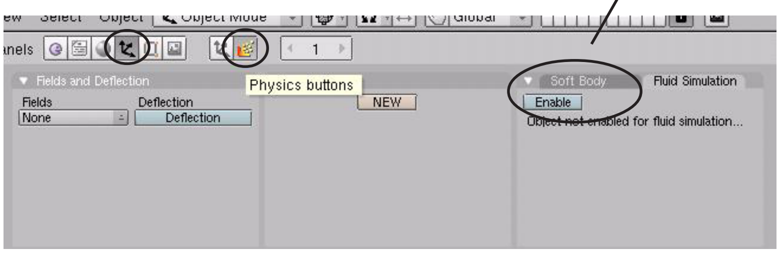
One of the newest and best features added to Blender in recent releases is Soft Bodies. Soft Bodies are used to simulate fabric, “jello”, and water. A mesh can deform a soft body mesh (clothing on a model or a boat in the waves). The **Soft Body** and **Fluid Simulation** features are found in the ***Object*** and ***Physics*** buttons.



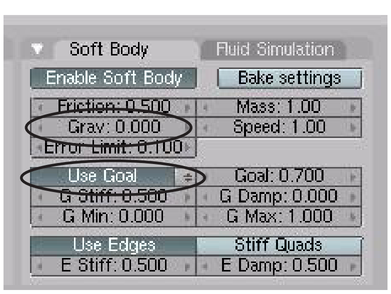
To create a simple animation of a plane with *Soft Body* applied to it to act like a sheet deforming to a sphere, create a plane above a sphere as shown to the left. In ***edit mode***, subdivide the plane **4-5** times. Soft Body shapes need a lot of verticies to look good. While still in *edit mode*, select all of the verticies and, in the **Edit** buttons, create a new vertex group (*keep the name as just* ***Group***). Set the weight to **0** and hit **Assign**. This will give all of the verticies a weight of 0 meaning they will not be anchored and be able to fall. Now deselect all of the verticies except the **2 left-hand outer verticies**. Take the Weight up to **1** and hit **assign**. This will anchor these verticies (like the corners of a flag). *Exit edit mode*. It’s now time to go to the **Object** and **Physics** buttons and hit the ***Enable Soft Body*** button as displayed below.



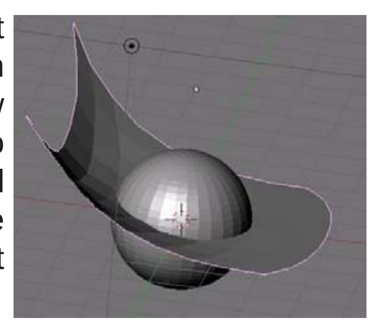


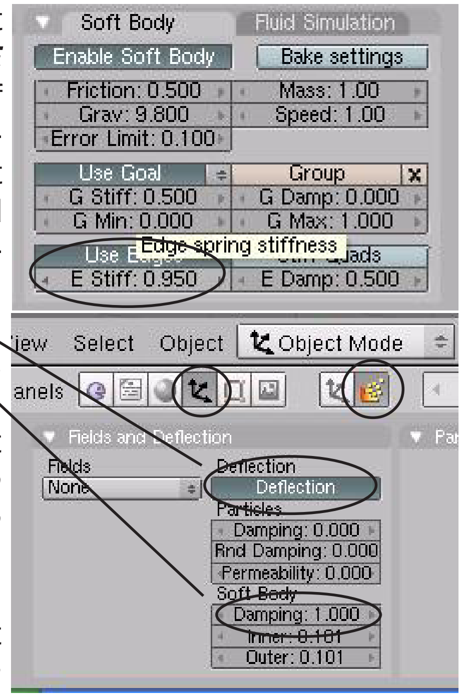


You now need to set a few things to see some animation. First, set ***Gravity*** to about **9.8** (earth settings) and select the vertex group: **Group** under the “***Use Goal***” setting.

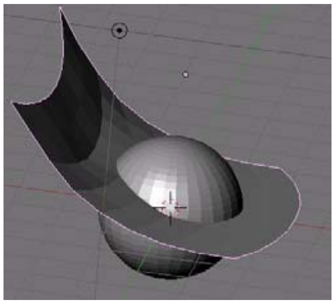


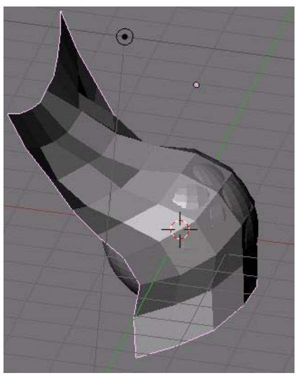
You are now ready to see the Soft Body in action! Place your cursor in the 3D window and hit “**Alt-A**” to view the animation. (you may want to shade the view) The plane should fall through the sphere and stretch quite a bit from the 2 anchor points. Not quite what what we want, but a start.

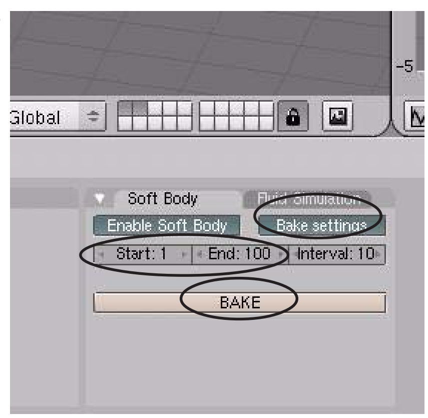




We now need to "*stiffen*" the fabric so it doesn't stretch out of shape. Under "***Use Edge***" you will see a box called "***E Stiff***" which sets the "spring" or elasticity of the verticies. If this number is taken to **0**, the shape will *stretch* out of control (try it). We want to set this at about **.95** so it does not stretch much. Try to animate the window again. You should see a difference. Now it is time to cause the fabric to interact with the sphere. Select the Sphere and turn on ***Deflection*** in the **Fields and Deflections** panel. Take ***Dampening*** up to **1**. This will cause the plane to react with the sphere. *This will slow the computer down considerable when hitting "**Alt-A**"*, but look at the example to the left. It looks good, but still passes through the object slightly. ***To improve this, you can Subdivide the Plane more and/or use a Subsurf modifier on the plane***. Experiment with the settings to get a good result. You can also speed up the process by using a sphere with fewer verticies. Once you have an effect you like and want to save the process without rendering (meaning you can only see the effect when you hit Alt-A and not by advancing frame-by-frame), you will need to "***Bake***" the animation settings. To do this, select the **Plane**, then hit the "***Bake Settings***" button. Set the **Start** and **End** frames you wish to bake and hit the "**Bake**" button. When it finishes running through all the frames, hitting the "Alt-A" button will be faster plus you will be able to cycle through all frames and see the animation. If you need to make changes, you will need to "***Free Bake***" the animation, make the changes, and "***Bake***" again.

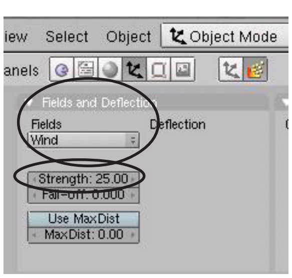


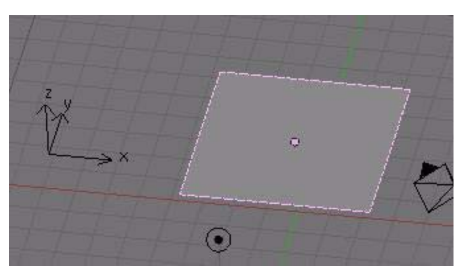


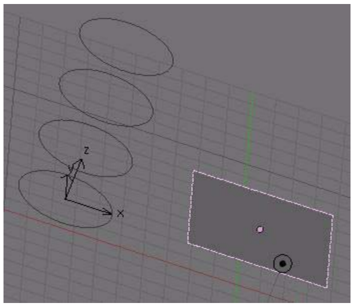


Using Forces to Manipulate Soft Bodies

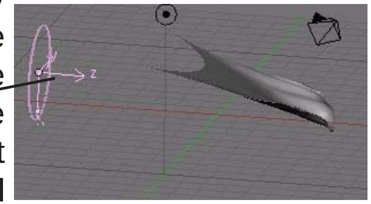
Besides collision detection with other objects, Soft Bodies can be manipulated by forces like wind. To demonstrate this process, delete the ***sphere*** from the scene (to speed things up) and add an ***Empty*** to the left of the plane. We will be using the Empty to act as the wind. Remember that an empty will not render as an object and works great for this type of application. With the ***Empty*** selected, go to the **Fields and Deflection** panel and change the Fields to "***Wind***". Set the **Strength** to about **25**.





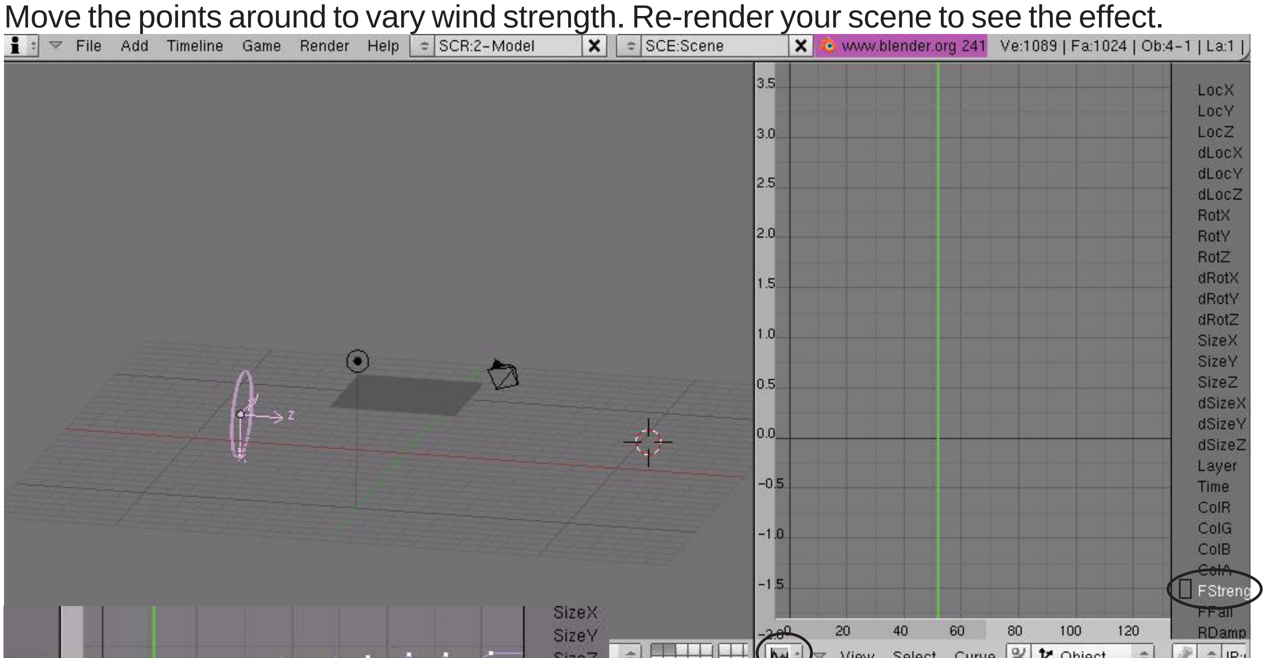


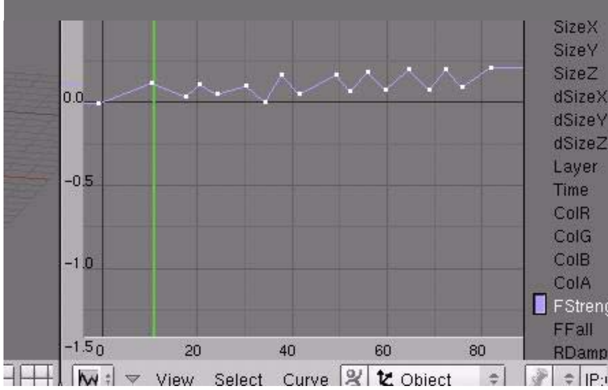
The wind may not be pointing in the right direction and may need rotating to point to the plane. Do that at this time. Hit “Alt-A” to animate the view. The wind is probably so strong that it almost blows the plane away. Try changing the ***Strength*** of the wind to about **.2** and re-animate. The effect should look much better. If you let the animation run long enough, you will start to see the plane slow down in movement and just stay blown out to the right. To fix this, you will need to animate the ***Strength*** of the wind or add animation keys to the ***Empty*** to change it’s direction. To add animation keys to the strength, *you will need to create a curve in the IPO window.* At this time, Blender will not allow you to use the “I” key in the button windows to insert keys.



To Create A Strength Curve In The IPO Window

Split your screen and turn one into the **IPO Window**. With the Empty selected, find the ***FStrength*** IPO track and select it. To put a track on it, Hold down the **Ctrl** key while **LMB** clicking at the desired point on the graph *(frame 1 at zero strength, frame 100 at about .2 strength)*. Change the curve type to ***Linear*** (**T** key) and add more points with the **Ctrl-LMB**. Move the points around to vary wind strength. Re-render your scene to see the effect.

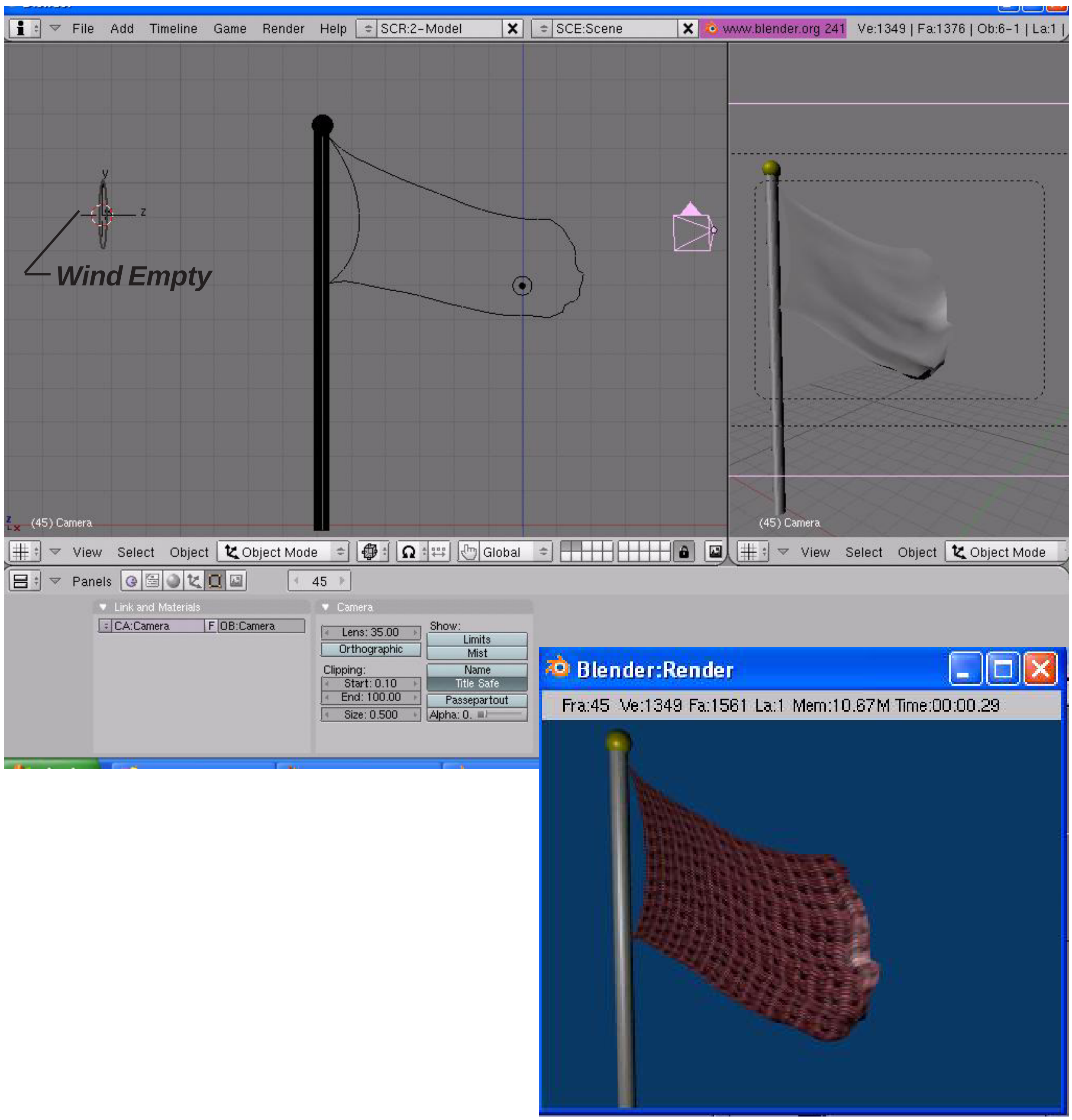




Soft Bodies can be used for a number of events. You will need to review other documents and experiment with the setting to discover what can be done. Blender also has some new work in **Fluid Simulation**. I am not including information here due to it’s experimental nature. As work progresses, review the information at www.blender.org.

Create a new file and call it *Flag*. Create a cylinder and sphere for the flagpole and a plane for the flag. Create a vertex group for the flag with all verticies (except the 2 that contact the pole) to a zero weight. The two corner verticies at the pole should be set to a weight of one. Setup a Soft Body effect for the plane as discussed and add a wind effect. Vary the wind with an IPO curve on the FStrength track. Place materials and textures on all objects.

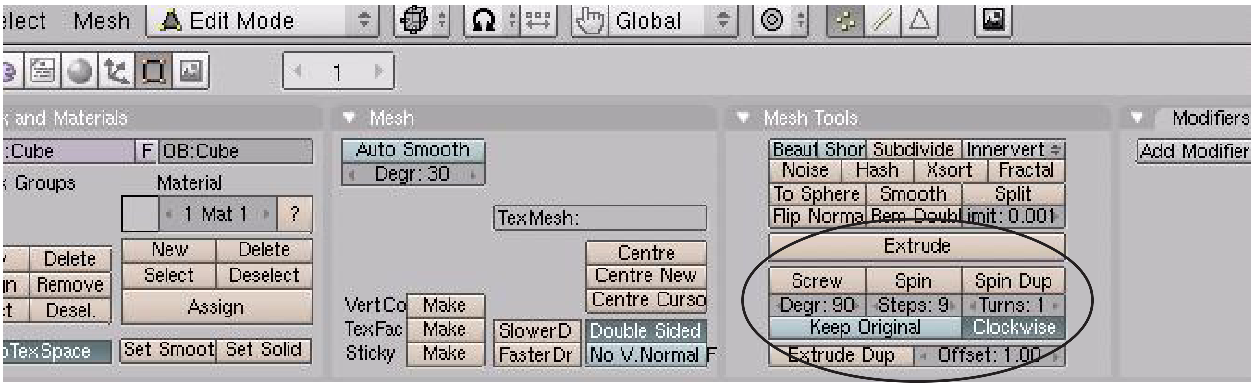
When finished, create a 200 frame animation of your scene.



***** Call the instructor when finished*****

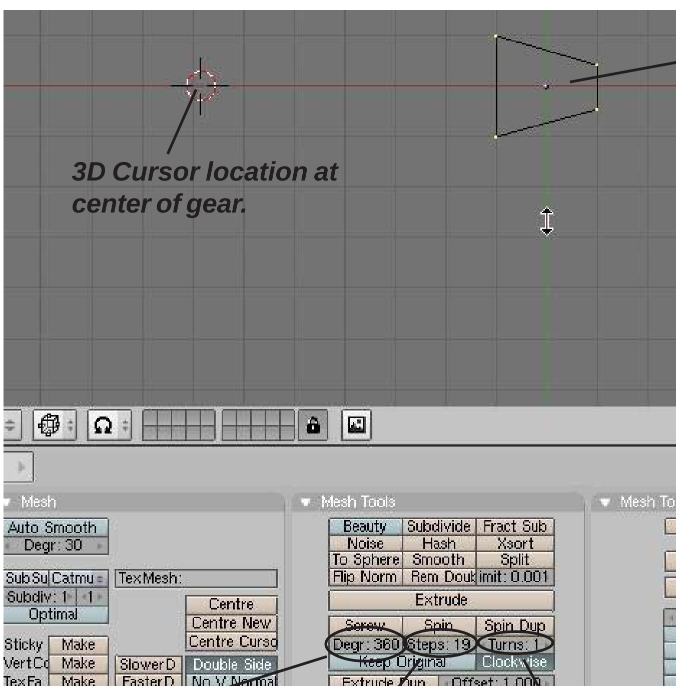
Duplicating Meshes to Create Screws and Gears

So far, we see that Blender has many features that are found in almost all 3D computer programs like the ability to extrude along paths, subtract and add meshes through boolean expressions and now we will examine ***revolving-type*** commands. The commands used for these effects are found in the *Edit Buttons* and are visible when in *Edit Mode*. The process to get them to work can be confusing to beginners. Here's what you see:



Spin Duplicate

Spin Duplicate will take a group of verticies and copy them around the *3D Cursor* location. For our example, I will use a modified cube to make a gear. Step 1 is to shape a cube into a simple gear tooth in edit mode. While in edit mode, select all verticies and review the following settings:



Cube shaped into simplified gear tooth

Degrees to go around*- set to 360 for a full circle*

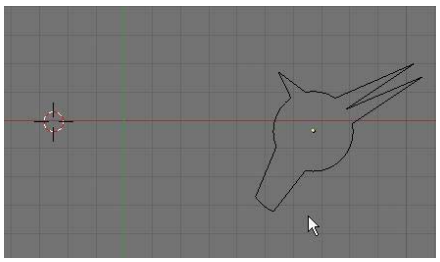
Steps*- number of instances. In our case number of gear teeth.*

Turns*- how many times to go around. We only want 1 turn for a gear.*

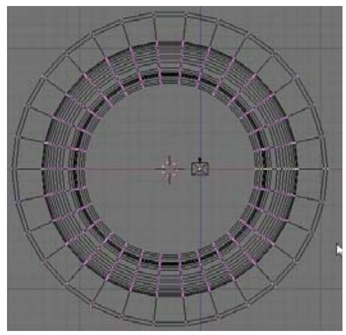
With all verticies selected and everything set to these numbers, hit the "***Spin Dup***" button and see the results. You will need to experiment with sizes and number of steps to get a good gear without any overlaps. Remember that edit mode has an **Undo command**. If it isn't right, just hit "***U***" and it will go back to your one gear tooth. Make sure all verticies are still selected and try again with some different settings. To fill in the gear and make it look realistic, add some cylinders to fill it in. You can also add a nice touch by using ***Boolean Difference*** to cut holes in the body of the gear.

Spin

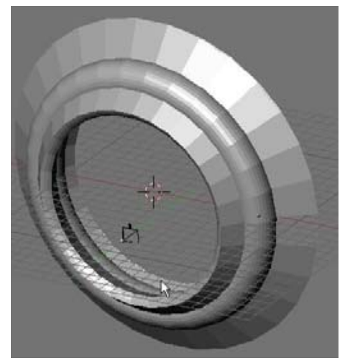
The *Spin* command operates similar to the *Spin Duplicate* command, except that it works more like a **revolve-extrude** command. You can take a plane or a circle, shape it, then revolve it around the *3D Cursor location*. For our example, we started with a ***Mesh Circle*** in the ***Top View***, then we placed the 3D Cursor at a desired location. Enter edit mode and select all verticies. Switch to the ***Front View*** and select the ***Spin*** command in the edit buttons. Notice how far it extrudes and the number of steps. ***Undo*** ("U" key) the spin so you're back to your basic shape again and change ***degrees to 360 and steps to 30***. Make sure all verticies are selected and try again. Here's our results:



Circle that has been shaped in edit mode. 3D Cursor to left of shape for center point.



All verticies selected, switched to frontal view and spin command used.

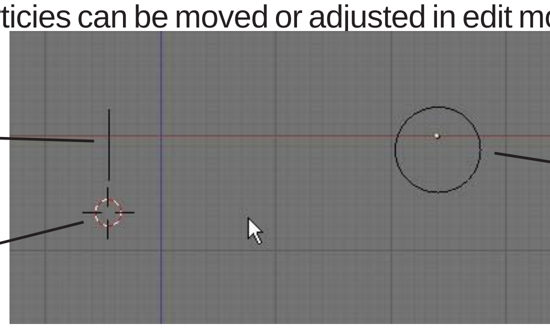


Final results of Spin command

Using Mesh Editing to Create Helix Shapes

This is the most complex of the 3 tools. This tool can be used to represent any type of ***helix object***. Springs, threads on a bolt, worm gears, etc. can be done with a little work once you know how to use it. The Screw command only works in the ***Front View*** (number pad 1) so switch to the front view to develop a spring.

In order to create a spring, you need to start with a ***Mesh Circle*** in the front view. Like the other spin tools, you need to place the *3D Cursor* in the location where you want the center to be located. ***Now you need to create a line that will represent the spacing in the turns*** (for threads on a bolt, the line is short so the threads are close together, for a spring with a large space between loops, draw a longer line). To create the line, add a ***Mesh Plane*** to your scene and delete 2 of the verticies. Size the line that is left to what you need for your model. ***Join the line and the circle together so that it is one object.*** For the best results, *place the line you created in the center of the revolved mesh* (beside your 3D Cursor). The verticies can be moved or adjusted in edit mode. Here's an example of your model:

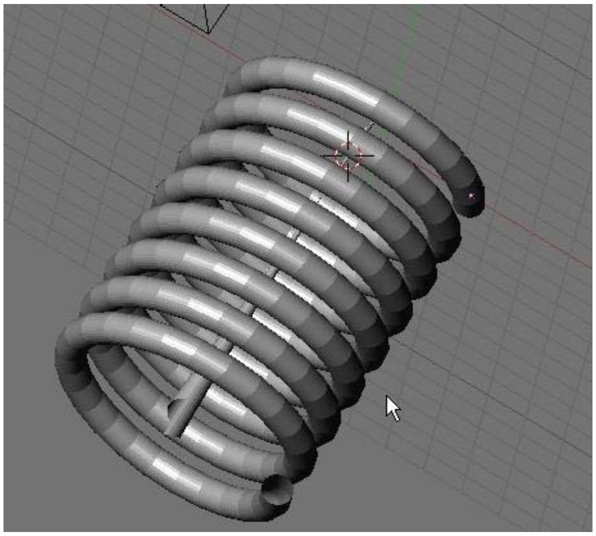


Here are the 2 verticies that form the line to controls spacing

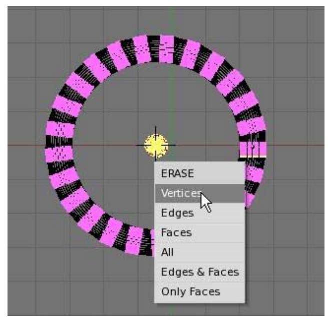
3D Cursor location

Circle that forms profile of the spring. Any shape can be used.

Set the ***Degrees*** to 360, ***Steps* to 16 or higher** (depending on the smoothness you want) and ***Turns*** to how many loops you want (I'll set it to 8). Select ***All*** verticies (including the 2 that form the line) and select the ***Screw*** button. You should see a spring on your screen. If something needs adjusting, press "***U***" to undo in edit mode to go back to your basic shape and try again. Here's our results:

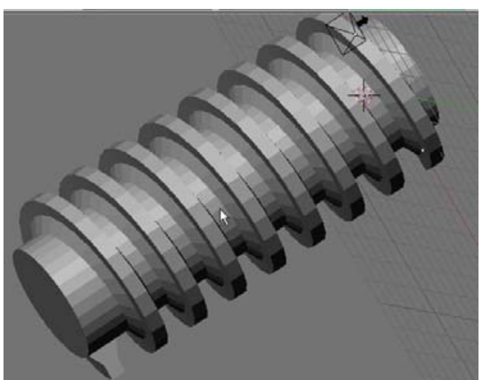


You will notice a shape in the middle of your spring. That shape is created from the verticies you used to designate the length of each loop. You placed this shape in the middle so it is easier for you to select these verticies in edit mode and erase them. Switch to a top view and enter edit mode. Select the verticies and hit "*Delete*". Now you'll just have your spring on the screen.

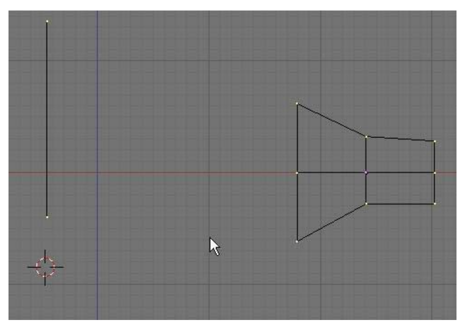


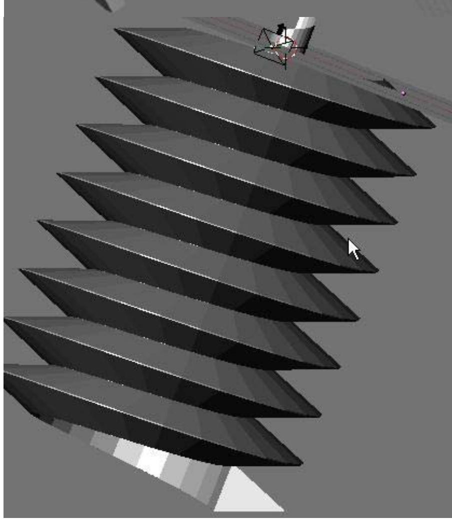
Other Shapes:

Here's how you can create some other things:

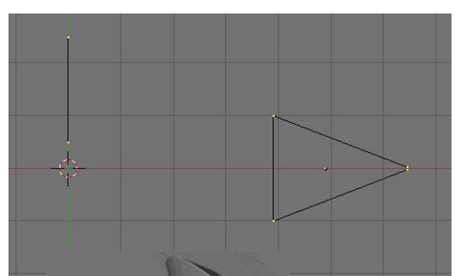


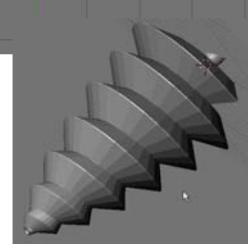
A Worm Gear created using a subdivided plane as the profile and the line set to equal twice the height of the gear profile. A cylinder has been added through the center to give it a solid appearence.



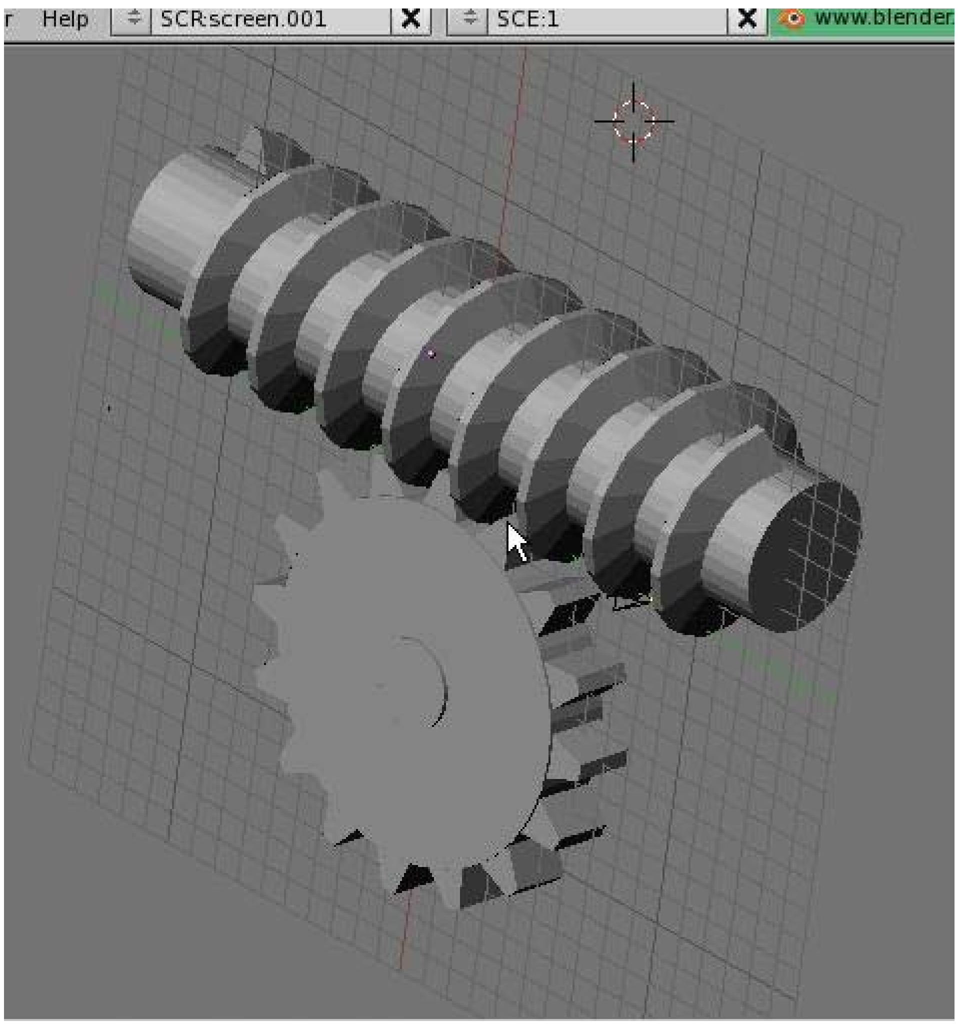


Bolt and screw threads can be created with a plane shaped into a triangle and the line for spacing set to exactly the same length as the base of the triangle. To make a pointed screw, grab the end thread verticies and scale it with proportional vertex editing ("O") for a nice look.





Create a worm gear and a spur gear to mesh with each other using the information discussed in this chapter. Add materials, textures and appropriate lighting. ***Make a 200 frame animation of the grears turning***. *Try to make them mesh perfectly!* Remember the Extend Mode options available in the ***IPO Window***. All you need to do is create a small section of the animation and let the computer do the rest!



***** Call the instructor when finished*****

Chapter 20- Game Engine Basics

We've seen that Blender is a powerful 3D rendering and animation program up to this point, but so far, all of the commands that we've looked at are in most high-power animation programs. The big difference is in the cost of the program and some features. One thing (besides price) that makes Blender stand out from the others is its integrated ***Real-Time* animation features** (aka. the Game Engine). The program integrates real-time motion with physics and logic blocks. For example, you can set your gravity in the world buttons, add friction and force settings to your materials, turn objects into actors and move them around, then have them react to other objects in the scene.

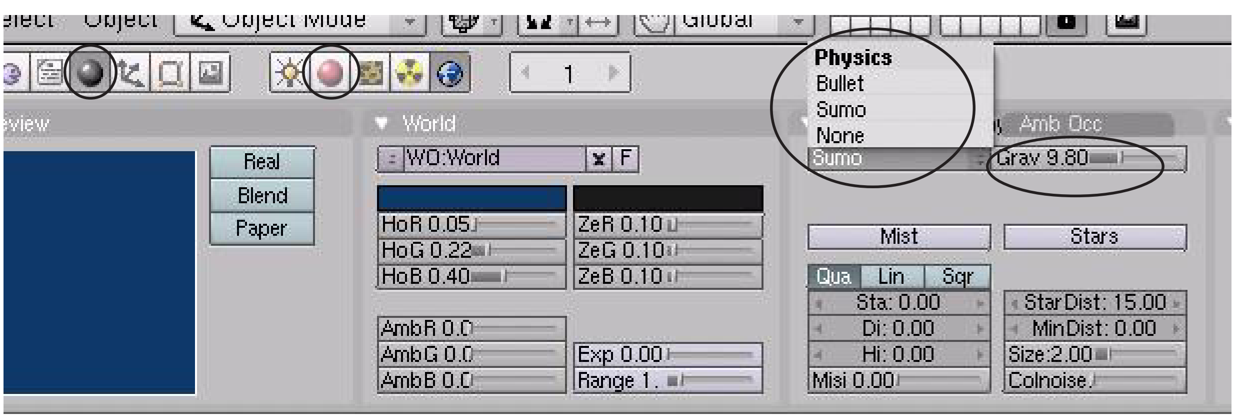
You can create games that look as good as professionally produced 3D games and real-time architectural walk throughs where doors can open and close as you approach them. The best part of this is that it can all be done *without computer programming skills.* There are other freeware game creation programs out there, but most require some programming skills. Programming skills in Python scripting are helpful in Blender, but not necessary.

This chapter cannot hope to cover everything you need to know about the game engine. We will only look at how to texture your models and describe the interface and logic. *For a more detailed description, review the Blender downloaded tutorial on the game engine.* It is well-written and describes all of the basic command options.

At the printing of this tutorial, the game engine has just recently been integrated back into the program. *It still needs a lot of work to be fully functional.* For this reason, many people still use old version 2.25 for game development. The original game engine physics module is called ***Sumo***, but a great deal of work has recently gone into a new module called ***Bullet.*** Bullet is showing a great deal of promise with accuracy and the ability to be used to create ***animation IPO tracks*** from the action.

Setting Up the Physics Engine

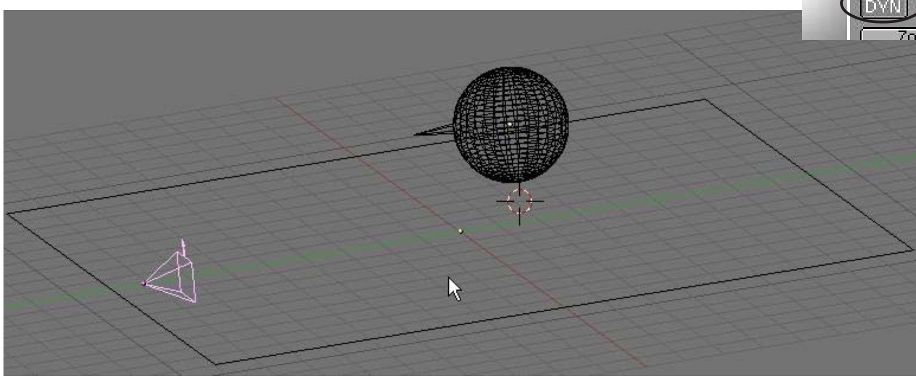
The first step to using the game engine is to set it up. To do this, go to the ***Shading*** and ***World*** buttons. Under the *Mist/Stars/Physics* tab you will set the engine. ***Sumo*** for games and ***Bullet*** for use with animation IPO curves. You can also set the ***Gravity*** at this time.



Using Logic Blocks

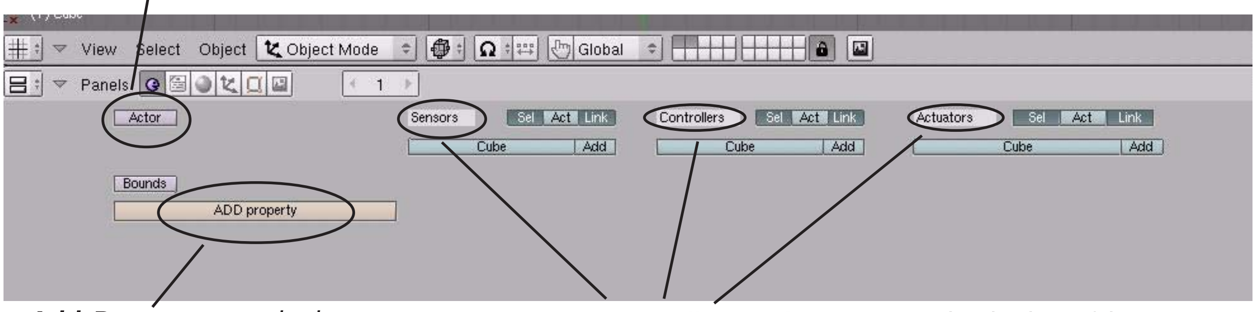
Let's make a simple scene consisting of a **plane** and a **sphere** and set the sphere above the plane. Modify the sphere by pulling one vertex out to form a *nose*. This will let us know which way is forward when we move it around. Add a material to each one, but don't bother changing any material settings. *The rendering materials and textures do not work in the game engine because the calculations would be too complex*. We will use a different process for this. We are adding materials for **physical properties (friction, elasticity)**. Here's what we have so far:

__DYN__ (Dynamic) settings in the material buttons



Select the sphere and go to the Game Buttons  (little purple pacman button). Here's what

Actor Button*- to turn the object into "live" actor.*



Add Property*- used when you want something to happen in your scene. Soomething will happen when it senses this property.*

Sensors, Controllers and Actuators*- The "brains" of the game engine. Think of it as Input-Process-Output where data is fed in, the computer processes it and something happens. There are a lot of options in these commands.*

Let's turn our sphere into an ***actor***. Click on the ***Actor*** button and choose ***Dynamic***. Look at the important options now available:

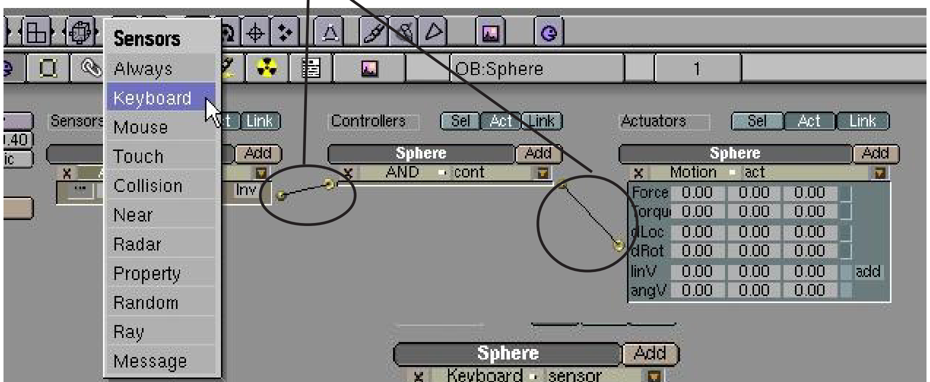
Damp*: motion dampening- keeps the object from continuing forever when you stop appliying force. I like to set this to around 0.4*

Actor Size*- You will notice a dashed line circle around the shere when you change this. This is the actor size.*

RotDamp*: rotational dampening- keeps the actor from spinning forever. I like to set this to about 0.8*

Mass*- how heavy you actor is.*

Move your cursor into the 3D window and press “***P***” for play. If the sphere is above the plane, ***it will fall to the plane showing you that it is now an actor***. Click the “**Ad**d” buttons under ***Sensors, Controllers and Actuators***. By holding the ***LMB*** down on each block, you can change it’s type. Change the sensor block from ***Always to Keyboard***. Next, connect the blocks together. Once you change the sensor to keyboard, you will see



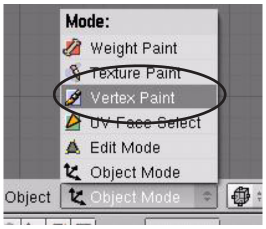
a block for ***Key***. click in that box and type the key you want to use. For our case, we’ll use the “***Up Arrow***”. We will tie a force to the up arrow so that when we press it, the sphere will move forward.

Now we’ll apply a ***force*** to the actor. You will see three columns in the ***Motio***n block. They represent ***X, Y, and Z***. The best way to change numbers in these blocks is to hold down “*Shift” and click in the box*. In the ***Force*** block, Let’s change the **Y** number to **10**. This is where you need to experiment with numbers. *If a block doesn’t move it in the direction you desire, change it back to zero and try a different one*. If it moves on the right axis, but the wrong direction, try a *negative number*. Once you get this motion right, add another row of block under the *Sensors, Controllers and Actuators,* connect them and adjust your setting to go backwards. To make the object turn left and right, work with the ***Torque*** settings and use the left/right arrow keys. There are a lot of options in these buttons. To get a more detailed description of them, refer to the *Blender Game Engine* documentation available to download from the Blender website.

Applying Materials

Wireframe is good for testing out motions, but poor for actually playing the game. Pressing the “***Alt” and “Z*”** keys will place you in game textured mode. However, when you hit “***P***” to play now, everything looks horrible. Time to add *some game shading and UV Textures*.

For straight color, you have several options to make it show up in game shading. Here’s the easiest way to go plus an option to add lighting effects. Select the Sphere and type ***“F” for Face Select***. This will highlight all the faces on the object. You have the option to select individual faces and texture them, but we won’t discuss that here. There are several good tutorials available to assist you with that. With all the faces selected, go to the ***Vertex Paint***

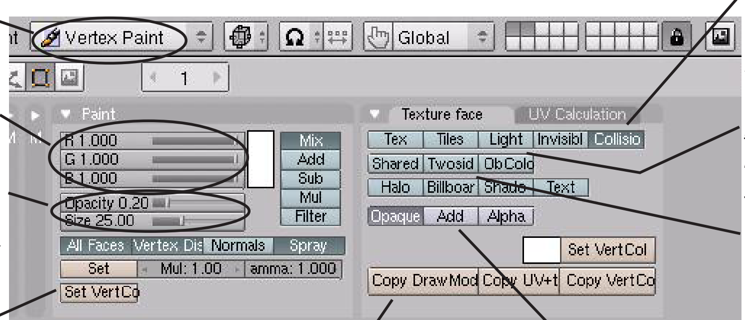


buttons. Here, you will see several options. For now, look for the color sliders and set a color that you would like to use for the sphere. Once you get a color you like, hit the "***Set Vert Col***" button to place the color on the sphere and press "**F**" to exit face select mode. Now if you hit "**P**" to play, the sphere matches that color, but it's flat with no reflections or shading. Let's fix that. First, make sure you have several lamps in your scene. Go back into face select mode ("**F**") and find the "***Light***" button. By pressing the "***Light***" button, then the "***Copy Draw Mode***" button, your sphere should take on a better 3D look. Exit face select mode and hit "**P**" to play again. Should look a lot better. *Note: the light feature works good on objects with several faces. For cubes and planes, subdivide the object a few times.* Here are a few more options in face select mode:

Vertex Paint- Acts like an airbrush to shade object.

Color Settings- change the color here.

Vertex Paint Options- Set opacity and brush size here to use vertex paint



Collision Button- if this is off, actors will pass through the object.

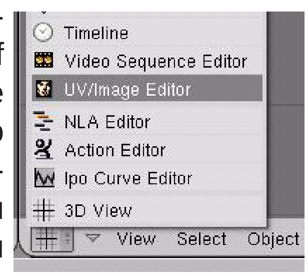
Light Button- This makes the object sensitive to light

2-Sided- places material on both sides of a face.

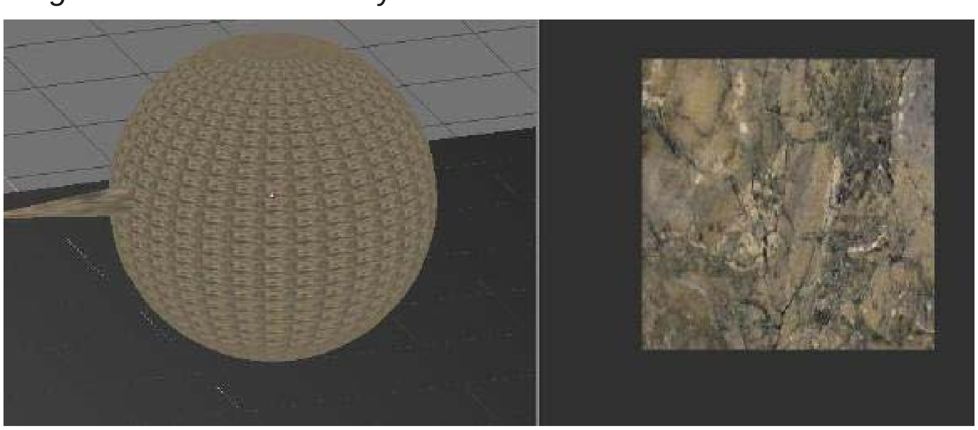
Set Vert Color- use this button after setting color.

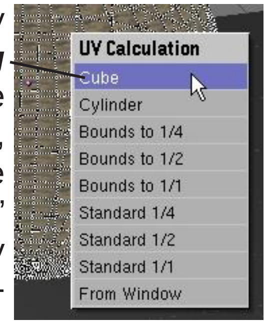
Copy Draw Mode- press this after making changes above.

Add Button- This makes the object transparent related to colors used.

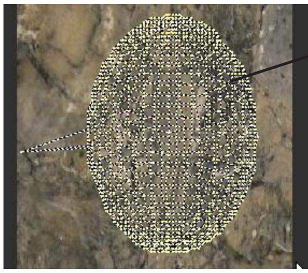


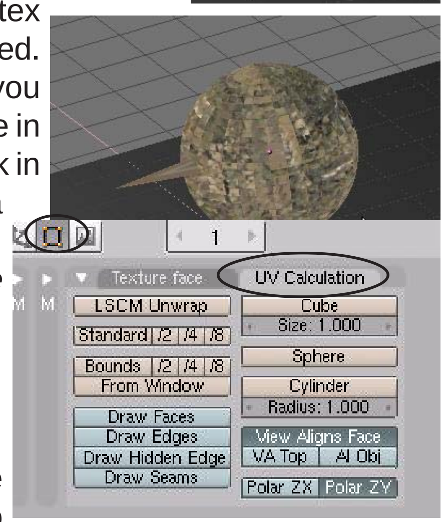
Now that we can add straight color, lighting effects and painting highlights, lets discuss adding textures throught the ***UV Texture*** options. If you haven't split your 3D window yet, do so now. We will need to use the right-hand viewport to set the textures. Change that window type to the ***UV Image Editor*** window (the person's face button). In this window, find the "***Load***" button, then browse around to find the textures you would like to use. It would be best at this time to load ***ALL*** textures you want for your scene. Now, select the sphere, type "**F**" for face select, then go over to the UV Image window and hold down the mouse button on the small white bar to browse through your loaded files. Select the texture you want. *Note: if you have placed a color and light on the sphere already, you may need to change the color back to white and turn off the Light button.* This is what you see.





Looks good, but you may not want the image mapped small on every face. Place your cursor in the 3D window and type *"U" for **UV Mapping*** options. Select ***Cube*** and a size of **1.00**. Now you see a sample of the verticies in the texture window. Select ***All*** verticies, move them and scale them to fill the sample. Place your cursor back in the 3D window and Press *"**F**"* to exit face select mode. Press "**P**" to play and try out your model. If you would like to add lighting effect and some shading through vertex paint go ahead. Features can be mixed. Be careful with how much lighting you add in your scene. It looks good, but can slow down a game in a hurry! Instead of hitting "**U**" for mapping, you can also look in the *edit* button for the ***UV Calculations*** panel. There are a lot of other things that can be done beyond this discussion. Look to blender.org and elysiun.com for more help. *Don't be afraid to experiment!*

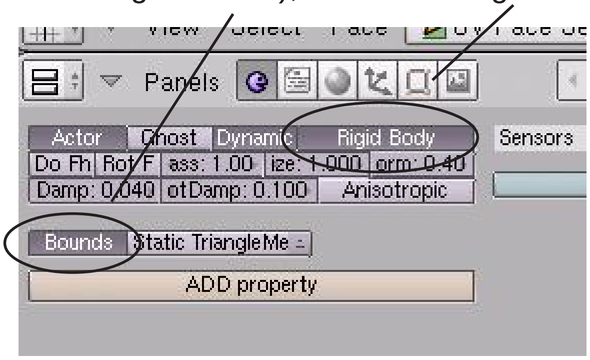


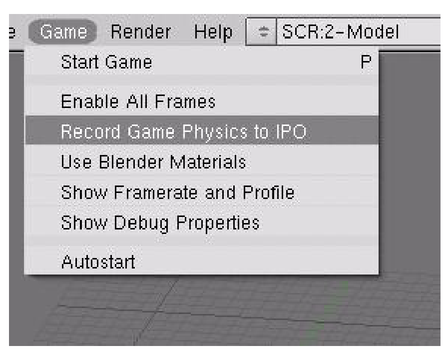


Using Game Physics in Animation

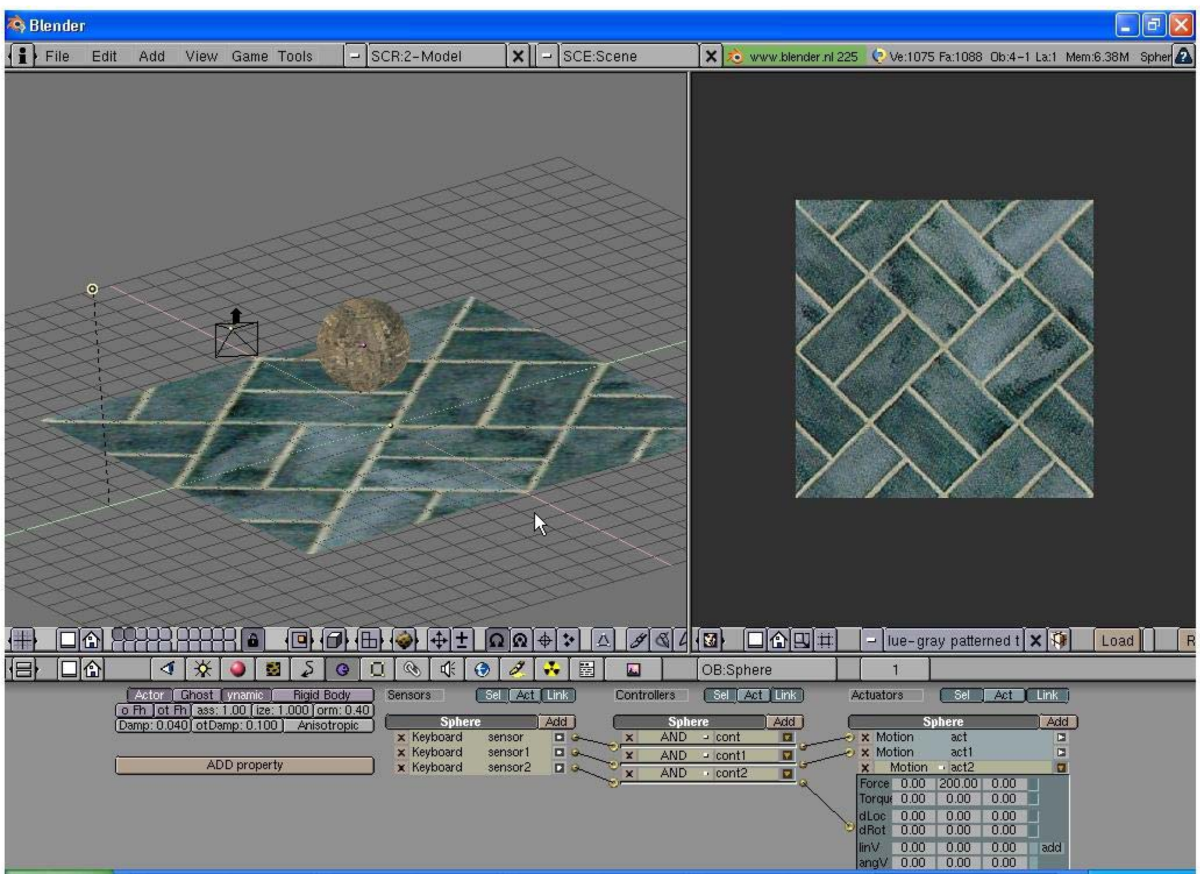
New for 2.40 is the use of the Bullet physics engine to create IPO curves for animation purposes. Now, instead of trying to insert location and rotation keys for a falling (moving) object in the IPO and 3D window, you can set up a ***Bullet*** model and tell Blender to record the action to an IPO curve for animation. No more frustration trying to make something look like it's moving correctly in a fall/fly/bounce. Blender will do it for you! To start out, you will need to go to the ***world*** buttons and set the physics engine to ***Bullet***.

The next step is to create your actors. To make something respond realistically (rolling and bouncing as it falls), select the "Rigid Body" option in the Actor panel and select a Bounds option that works for your model. Press "P" to play and fine tune your animation to get a realistic action. When you have the action correct, go to the "Game" pull-down menu and select the "Record Game Physics to IOP" option. Hit the "P" button to run the action. When finished, hit the "Esc" key. The animation will be written to an IPO curve for the object and will now work with "Alt-A" and animate!





Create a scene similar to the one discussed in this chapter. It should include one actor and a plane, both textured for game mode. Apply physics to the actor so that he can move forward and back, turn left and right. When finished, create an IPO curve of your action so a movie can be rendered from the scene.



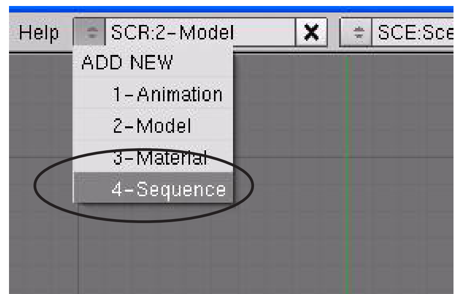
***** Call the instructor when finished*****

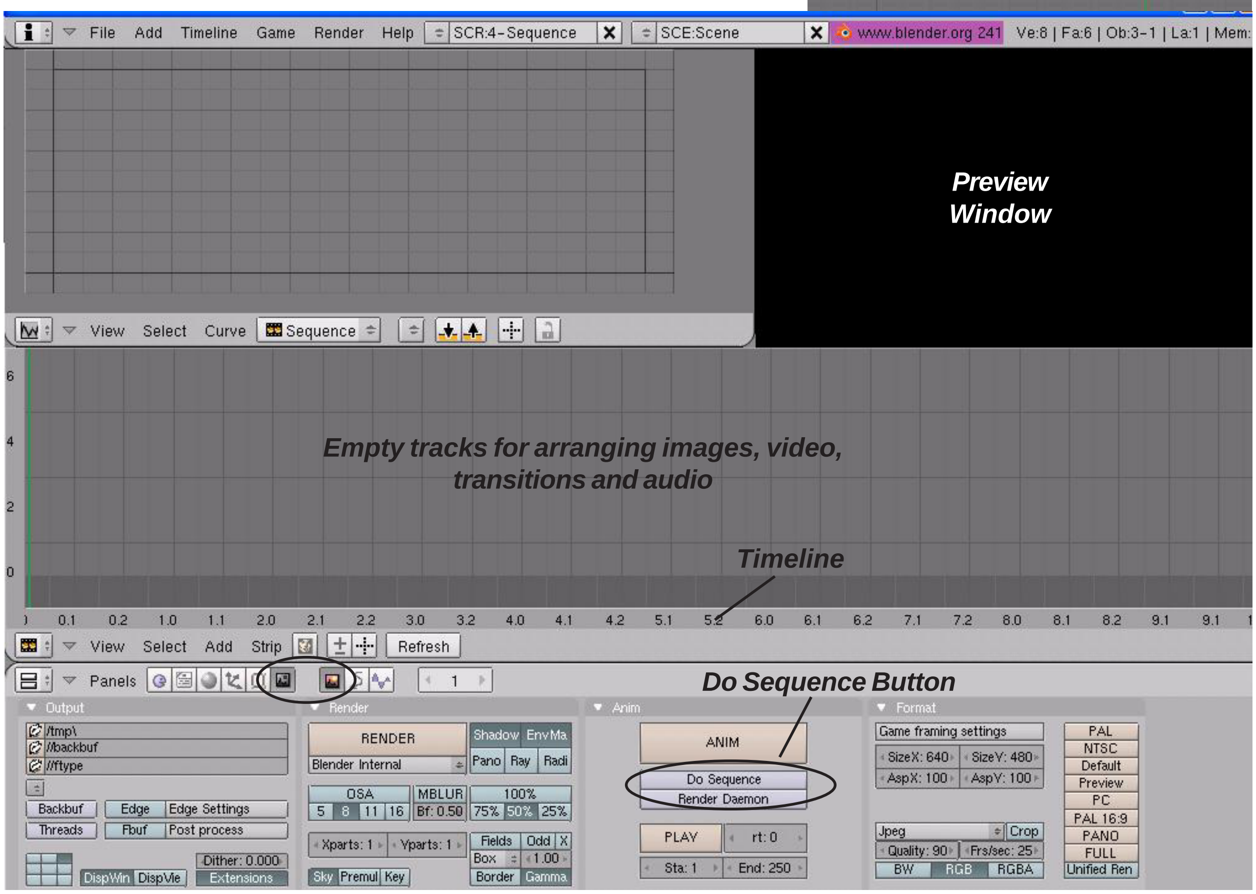
Producing a Movie From Individual Clips

Movies are never produced from one continuous camer shot. A movie is made up of short clips and images combined together with sound and effects to give us an end production. Blender has the capability of doing much of this production work. Blender can combine video together with nice basic effects and can produce an audio track, *but cannot combine the audio and video tracks together at this time.* This is something I have personally been waiting on for years. Blender is so close to being able to handle all of our production needs, but because it lacks basic audio editing features and audio/video compiling, we still do most of our final assembly in Adobe Premiere. My hope is that some developers pick this project up in the near future.

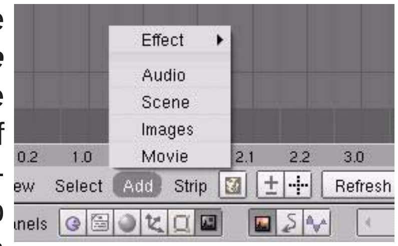
Setting Up the Sequence Editor Screen

Blender has a preset screen for *Sequence Editing*. To access it, go to the top toolbar and select the bottom screen option “***4-Sequence***”. When you select this, your screen will look like this:

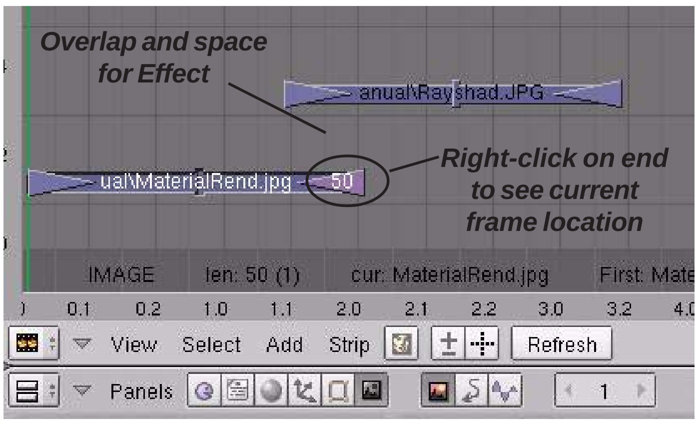




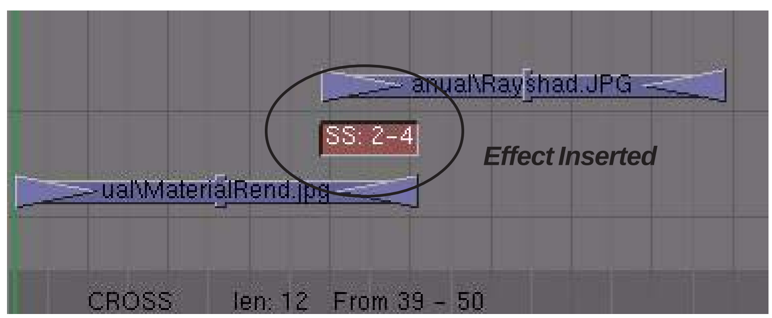
If you are familiar with programs like Premiere, this will seem familiar, if not, the process is simple. First, set up your output options in the render buttons for what you want your final movie to be. Press the “***Do Sequence***” button so you render the sequence and not the 3D window (*what the camera sees*).



Next, select the "Add" button above the buttons window. We are interested in the ***Effect*** (transitions), ***Audio***, ***Images*** and ***Movie*** options. Select a movie or image from your files to add to the timeline. Pay close attention to the frame markers at the ends of the file you are inserting. Images can be lengthened or shortened by ***RMB clicking*** on the end and using the "**G**" key to grab it. *Movies can only be shortened*. If you would like to add an ***Effect*** between the two files (*like a cross-fade*), overlap the files a second or two as shown below. *I like to leave a space between the files as shown to insert the Effect later*. Continue adding images and movies as needed.

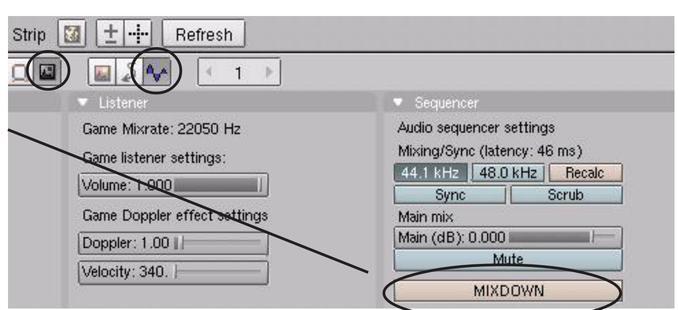


When you are ready to insert the **Effects**, you need to hold down the "***Shift***" key, then select the **first** file in the sequence, then the **second** one of the transition. *Selecting them backwards will cause the **Effect** to not function properly!* Go to the "***Add***" menu and select "***Effect***" and the effect you want to use. A simple ***"Cross***" dissolve works well for most things. Place the effect between the files. To view a sample of the move, Place your cursor up in the **Preview** window and press "***Atl-A***". Continue down the timeline and insert all transitions. If you do not wish to use transitions, but go straight from one image to another, simply "butt" the tracks to one another being careful not to overlap them. To create the final movie, **RMB click on the last image in the movie to find out the final frame.** Set this in your **End** frame movie output. Press the "***Animate***" button in the button window. Rendering time will be faster since you are only compiling already created images.



Adding an Audio Track

Do the same thing with your audio track. Be aware that some audio formats will not be supported and that you will need to experiment with them. Wave (.wav) files work best. To export the Audio Track, go to the ***Sound Block*** buttons and hit the "***MIXDOWN***" button. An audio track of the same name as your movie will be made. You will need to compile the movie and audio in another program.

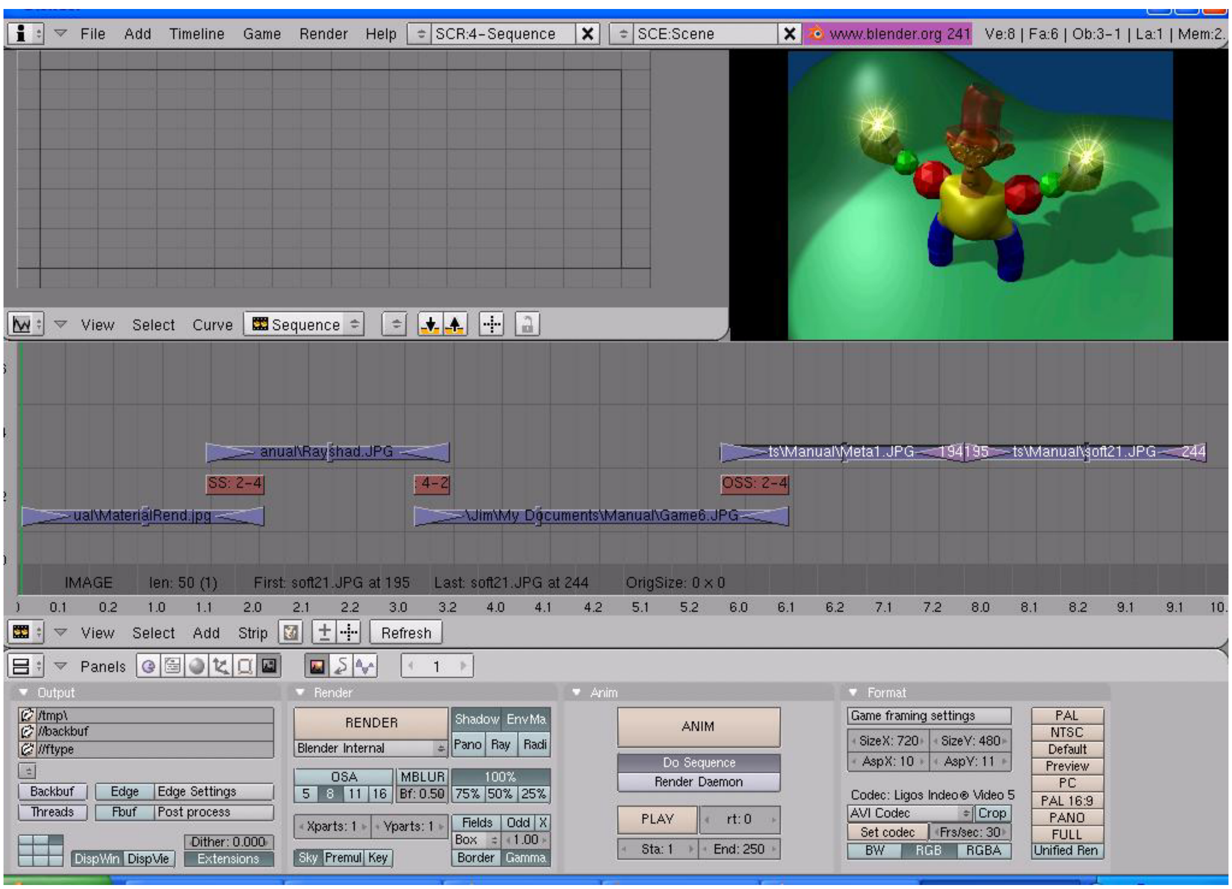


Movie Producer Practice Exercise

This is a great exercise to end the book with. Your job is to take all of the images and movies you have created through these exercises and produce a compiled movie. I recommend making a ***Black JPEG image*** to use for your **first** and **last** image (*start and end in black*). Images should be diplayed for approx. 3-5 seconds with 1 second transitions.

> ***Quick Tip**: It is best to work with movie files that have all been rendered at the same size and frames-per-second. Problems may occur when Blender tries to convert files that are different from the output you have selected.*

I hope that this manual has been helpful in learning this difficult, yet powerful program. The projects that you can create in it are limited to only what you can imagine and are willing to learn.



***** Call the instructor when finished*****