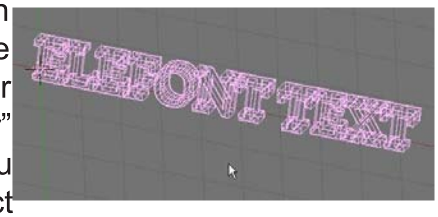
Effects (build, wave, particles)

Blender comes with some nice animation effects that can be applied to meshes. These effects can be used to create wave effects, things appearing, fire, smoke, sparks, clouds, shooting stars, fireworks, etc. The list is only limited to your imagination. For any of these effects to work well, you usually need a high number of verticies. You can get this by using the subdivide command in edit mode.

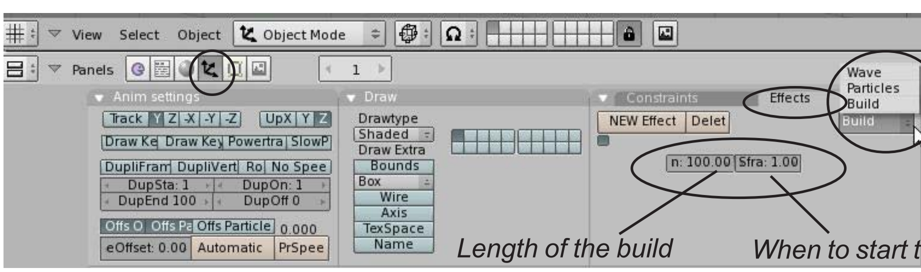
Build Effect

The build effect is nice for simulating something building linearly along time. Text is a good example of something you may want to use with a build effect. Blender generated text cannot be built, but EleFont text can used.



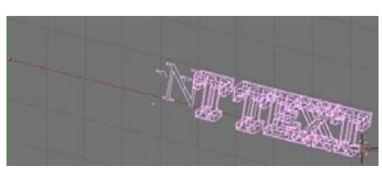
The first step in creating a build effect is to have a mesh that you would like to build. For this example, we will use this EleFont text. A high vertex count is usually needed for a nice effect. Notice the location of the object's "center" point. You may need to move this to get the effect you desire depending on what you plan to do with the object (discussed in the basic editing chapter). When bringing text into Blender from EleFont, the letters will all be separate objects. Be careful when joining them together. You need to join them together in a straight sequence (hold down the shift key and start at one end and select the letters in sequence-don't jump around). After all letters are selected, press *"Ctrl" and "J"* to join them. *Jumping around when selecting them to join will cause it to build oddly.*

With the object selected, go to the *Object* buttons, select the *Effects* tab, click the "*New Effects*" button, then choose *"Build"* from the list of effects. You will have 2 things to set- the *Length* and the *Start Frame* (how long do you want it to take to build and when in the animation do you want it to start).

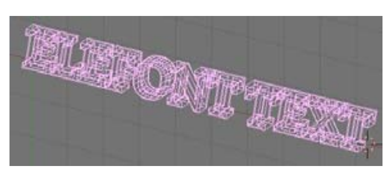


Don't be alarmed if the object disappears when you select build. You will only see the amount of it that should be there on your current frame. Press *"Alt" and "A"* to see it animated on the screen. This also now works in shaded view (*"Z"* key).

Build an frame 1

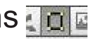


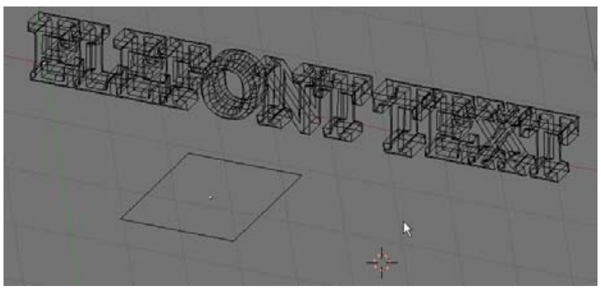
Build an frame 50



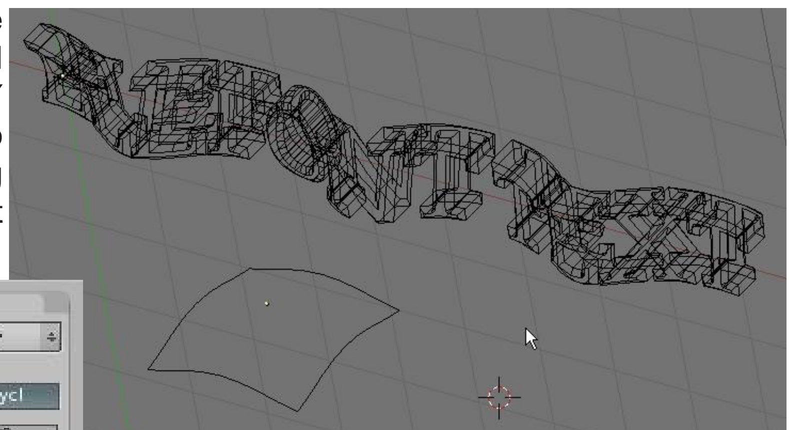
Build an frame 100

Wave Effect

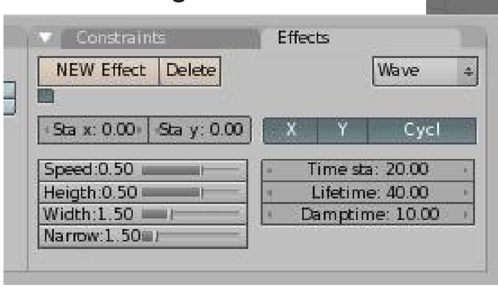
The wave effect can be used to make something look like it's swimming like a fish or blowing in the breeze among other things. Again, you need to start with something that has a good vertex count. For this discussion, I will use the EleFont text and a plane. The text already has a high number of verticies, but the plane does not. With the plane selected, enter edit mode (*Tab* key) and select all verticies (*"A"* key). In the Edit buttons  find *"Subdivide"* and press it about 3 times. You can also subdivide while in edit mode by pressing the *"W"* key and selecting *Subdivide* from the menu. Exit edit mode (*Tab* key). Here's what we have so far:



Once again, with the object selected, go to the *Object* buttons, select the *Effects* tab, click the "*New Effects*" button, then choose *"Wave"* from the list of effects. Place your cursor in the 3D window and press *"Alt and "A"* to see your animated wave effect. Hit escape to exit. Here's the basic wave effect on the text and plane:



The basic wave settings have the object in *Cyclic* mode and waving on the X axis and Y axis.These can be set to achieve different effects along with the height and width, start time and length of effect.



Particle Effects

This is probably the nicest and most flexible of the effects. When you turn an object into particles, it can be used to simulate snow, fire, smoke, clouds, sparks and much, much more. When an object is turned into particles, it no longer exists as a solid shape and releases particles as per the settings you used on it. With particles, you can set the size (using halo),texture, color and transparency through the material buttons. You can set the particles to come off the object in a sequence or radomly by using *Hash* in the *Edit* buttons. Particles can be set to be pulled using X,Y, and Z forces. You can control the number of particles, how long the particles live, when to start and end, if they have a starting speed and much more than we will describe in this chapter. Like all of the other features we've discussed, experimentation beyond this chapter is the best way to learn.

For our example, let's create a *UVSphere* and keep the segments and rings at 32. Once again, with the object selected, go to the *Object* buttons, select the *Effects* tab, click the "*New Effects*" button, then choose *"Particles"* from the list of effects. You get a few more options with particles than you do with the other 2 effects. Here are the ones we're interested in at this point:

Delete- *Start over if needed*

Recalc- *Sometimes after making changes to the setting, you need to hit this to update the animation*

Static- *Use this to make the particles "stick" and not be animated- Good for clouds*

Start and End- *When do you want the effect to start and end.*

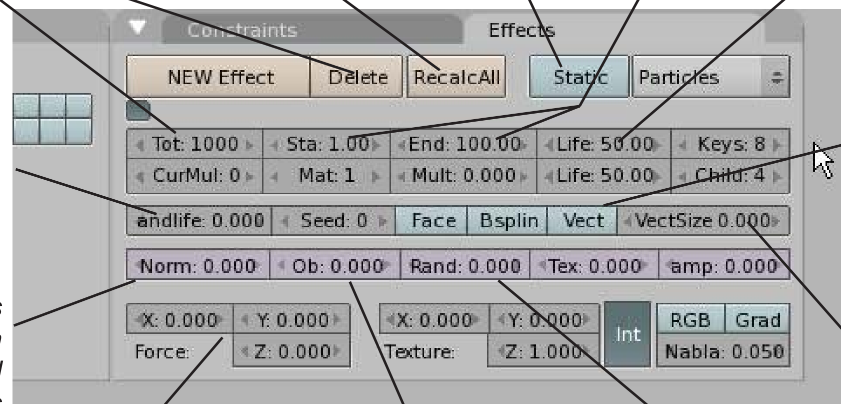
Life- *Sets how long each particle should live after it is created.*

Tot- *Total number of particles in the entire animation*

Random Life *Use this to slightly vary when particles die.*

Norm- *Gives the particles a starting speed (blasts outward)*

Vect- *Gives the particles a vector (draws as a line (like a spark trail)*



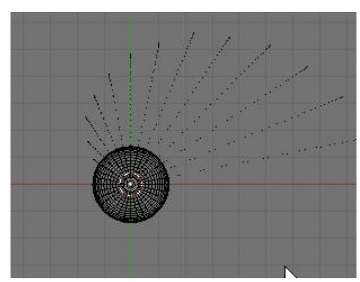
Force X, Y, Z- *Push or pull the particles in a given direction*

Ob- *If the object is animated, the object can effect the starting speed*

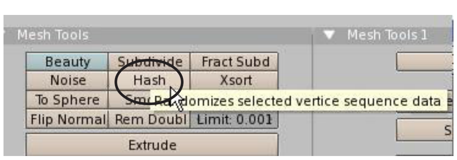
Rand- *This can give the starting speed a random variation*

Vect Size- Controls the size of the vector trail.

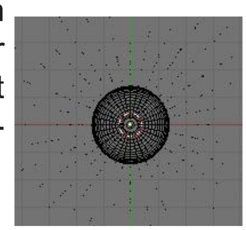
As mentioned before, there are a lot of settings here not discussed. For most basic particles effects, these are the ones we need to adjust. As you become more comfortable with Blender and effects, try out some of the others. To change any of the above settings, you can click to the left or right of the box, drag the mouse in the box using the LMB, or for more accuracy, hold down the *"Shift"* key and mouse click in the box. You can then keyboard in any number.



Let's work with our example. Make sure the sphere we created is not in edit mode (*Tab* key) and is selected. Select the particles effect and change the *Norm* setting to about 0.2 or 0.3. This will give the particles a starting speed causing them to "blast out". Move your cursor to the 3D window and hit *"Alt-A"* to see the animation. It should look something like this:



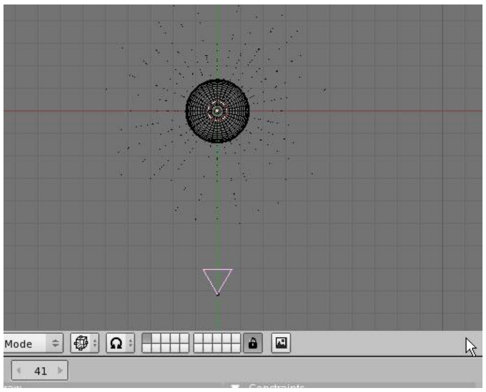
The particles come off in a pattern. This may be what you desire, but if you're making a flame or explosion effect, you need a more random release of particles. To do this, enter edit mode (*Tab* key) and select all verticies (*"A"* key). Go to the edit button and select "*Hash*". This will randomize the vertex sequence. Re-animate the window with *"Alt-A"* again and see the effect.

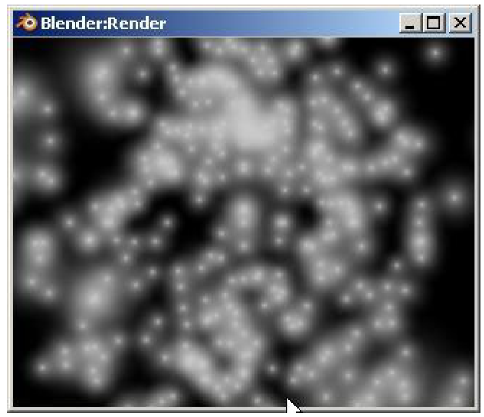


Materials on Particles:

Particles emit their own light so it isn't necessary to have lamps in the scene to see them when rendered, however, you will need to add a material to them to give them color and to control their size and transparency.

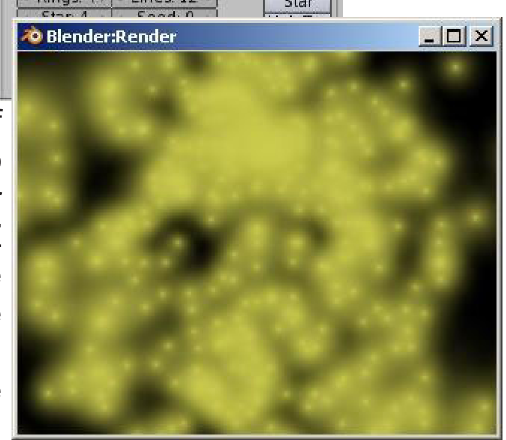
If you look at the scene we created so far, we have a camera and a sphere with a particle effect. The verticies have been randomized with the *Hash* command. If we change our current frame to a higher number like frame 50 and press the *"F12"* key to render, this is what we see:





We see a randomized particle system without a material added to it. The particle size may be adequate for your needs, but in order to control size, we need to add a material and use the halo effect. You may also need to select *"Z Transparent"* and adjust the *"Alpha"* to set a transparency effect for the particles. The transparency effect is ideal for flame effects where you use multiple objects with particles on them and add different colors to get a realistic looking flame.

Here, I've taken the sphere and placed a material on it. With the "Halo" button pressed, change the halo size to affect the size of the particles. Color can also be adjusted.



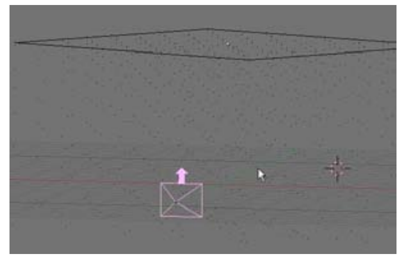
The halo size needs to be adjusted for the size of your scene and what you are using the particles to simulate. For example, smoke needs a smaller particle count and a larger halo setting and a Z Transparent and Alpha setting very low so the smoke looks transparent. Fire may need a higher particle count and a smaller halo setting to look good. Depending on what you want, you can turn on the rings, stars and line in the halo settings.

Basic Settings:

Here are some basic settings that can be used for a variety of effects. The numbers given in these examples are based on a 100 frame animation. If you lengthen the animation or change the size of the objects, you will need to adjust things like the total number of particles, forces and starting speeds. These settings can be “tweeked” to your own personal preferences. they are just intended to get you to a starting point.

Snow

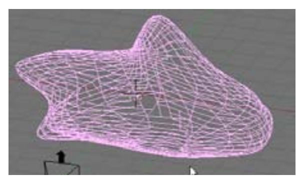
Subdivide a plane 3-4 times (or more) and “*Hash*” the verticies. Change the following settings:



Tot Frames: 1000	Random Life: 0
Sta frame: 1	Norm: 0
End Frame: 100	Force Z: -0.05
Life: 75	Vector: Off

Clouds

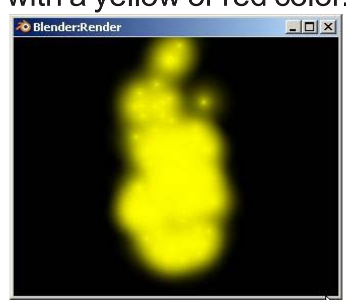
Create a *UVsphere* and shape it with *Proportional Vertex Editing*. Pull the shape around to try to look like a cloud. *Hash* the verticies and put a material on it. Hit ZTransparent and take *Alpha* down very low (.1-.2). Set the *Halo Size* to about 4.00.



Tot Frames: 100	Random Life: 0
Sta frame: ---	Norm: 0.03
End Frame: ---	Force : 0 for all
Life: ---	Vector: Off

Fire

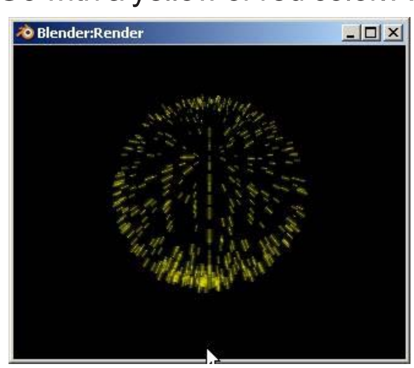
Start with a *UVSphere*, *Hash* and put a material on it. Go with a yellow or red color. Add a *Halo* effect and set the *Halo Size* to 1.2.



Tot Frames: 500	Random Life: 0.5
Sta frame: 1	Norm: 0
End Frame: 100	Force Z: 0.3
Life: 25	Vector: Off

Fireworks

Start with a *UVSphere*, *Hash* and put a material on it. Go with a yellow or red color. Add a *Halo* effect and set the *Halo Size* to 0.5.

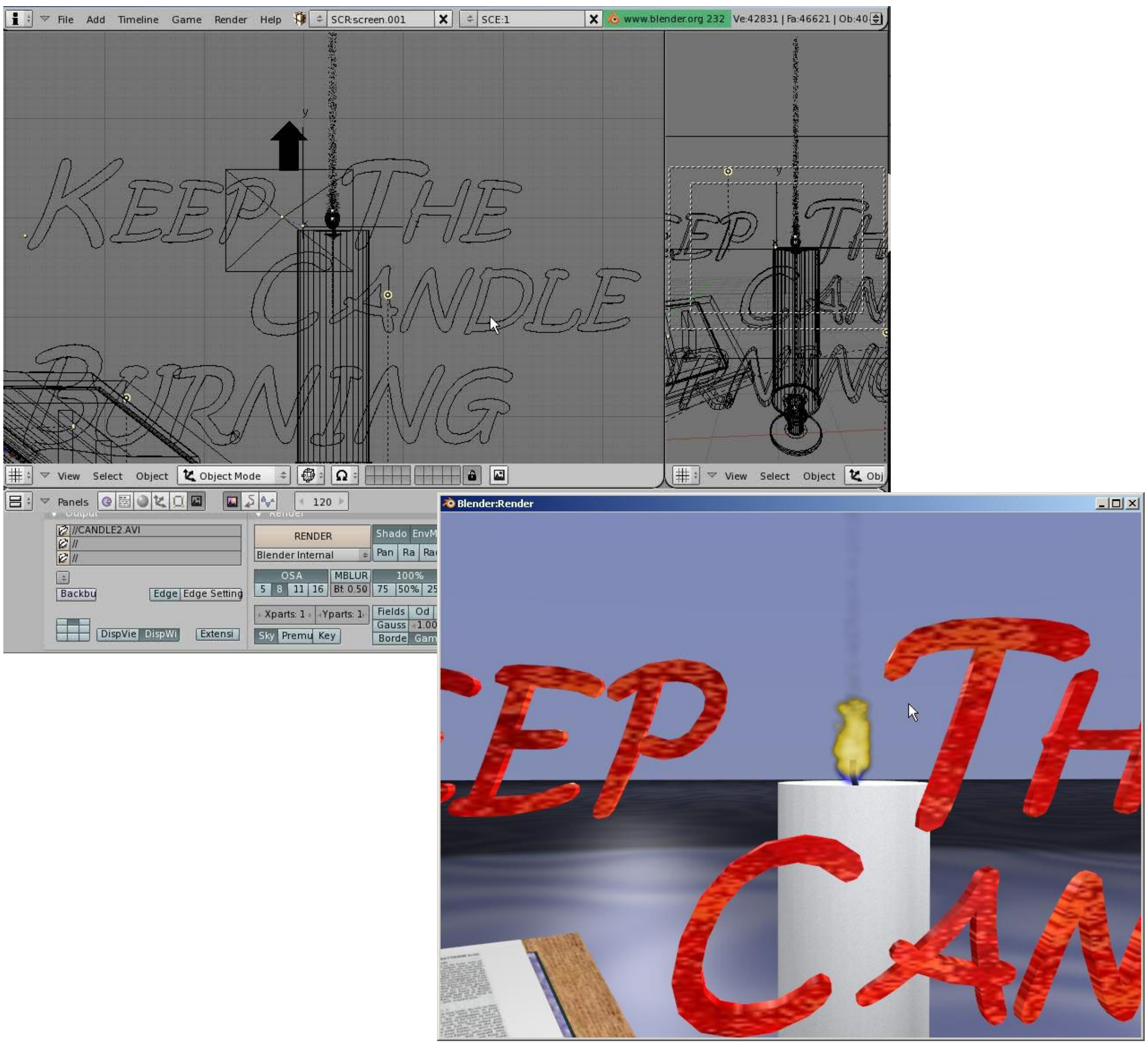


Tot Frames: 500	Random Life: 2
Sta frame: 1	Norm: 0.1
End Frame: 1	Force Z: -0.1
Life: 30	Vector: On- Vect.Size-1

Effects Practice Exercise

Create a new Blender scene and set up the views any way you wish. Create a candle using cylinders (candle and wick) and add 3 UVSpheres at the wick to use as particle systems. Create a red flame, yellow flame and smoke trail. Use *ZTransparent* and *Alpha* to cause a mixing of the red and yellow components of the flame and a rising smoke trail. Also, create some 3D text in EleFont and place a wave effect on it within the scene. Set all animation lengths to 100 frames and create an animation of your scene using AVI Codec and Indeo Video 5.1 settings.

Challenge exercise:
If time permits, add a build effect and a fireworks effect in the background. You can also add more effects to your flame for more realism.



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

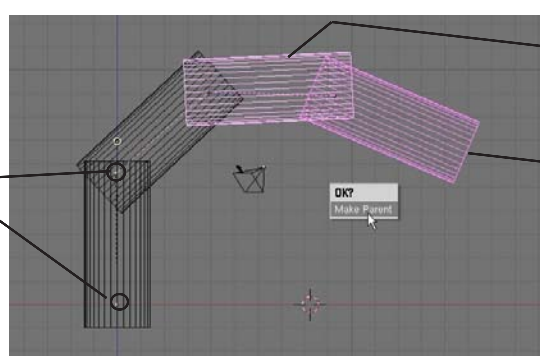
Child-Parent Relationships

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. *In order to delete a child-parent relationship, select both objects and press "Alt" and "P" to clear parent.*

Look at the example below. 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 objects 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. Make a simple animation to check the function.

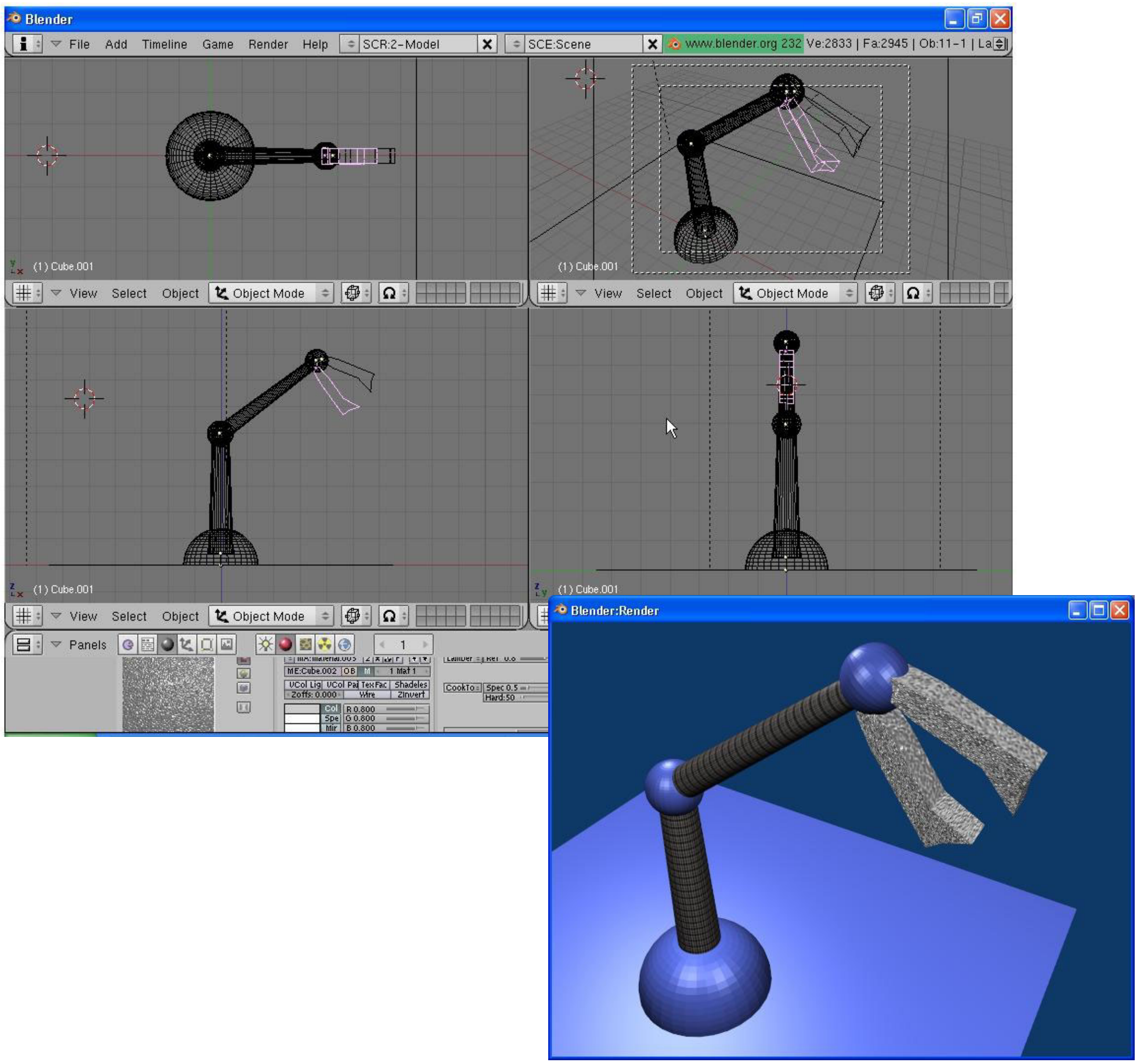


Child-Parent Relationships Practice Exercise

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.

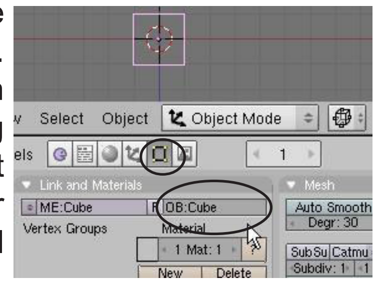
Challenge exercise:
Try to make your robot arm pick something up of the plane.



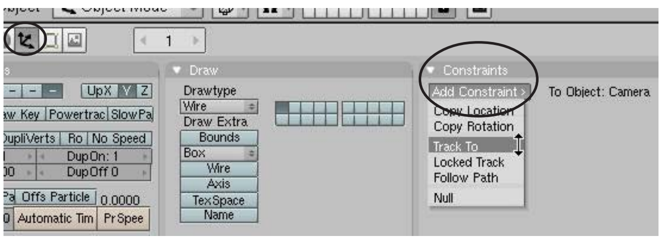
***** Call the instructor when finished*****

Working with Constraints

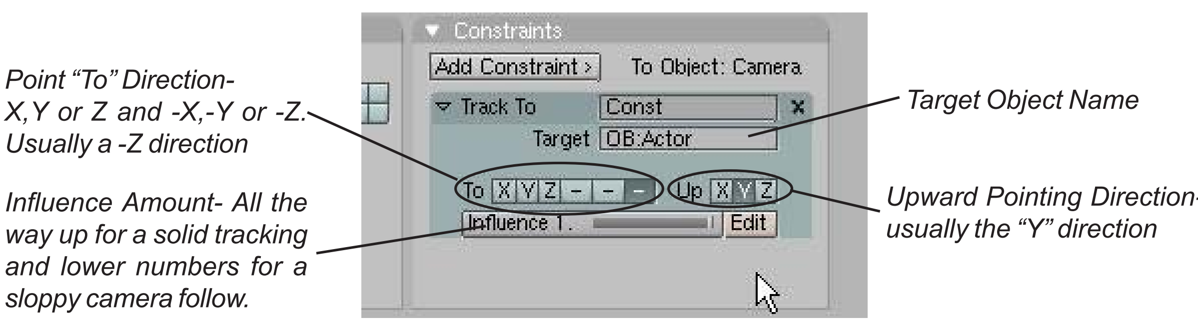
There are times you want to “constrain” or “follow” a certain object in your scene. New constraints are being developed in Blender, but for now, we will just be talking about the most common one used to keep the camera focused on an object- *the “Track To” constraint*. The tracking constrain is useful 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 the next 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.

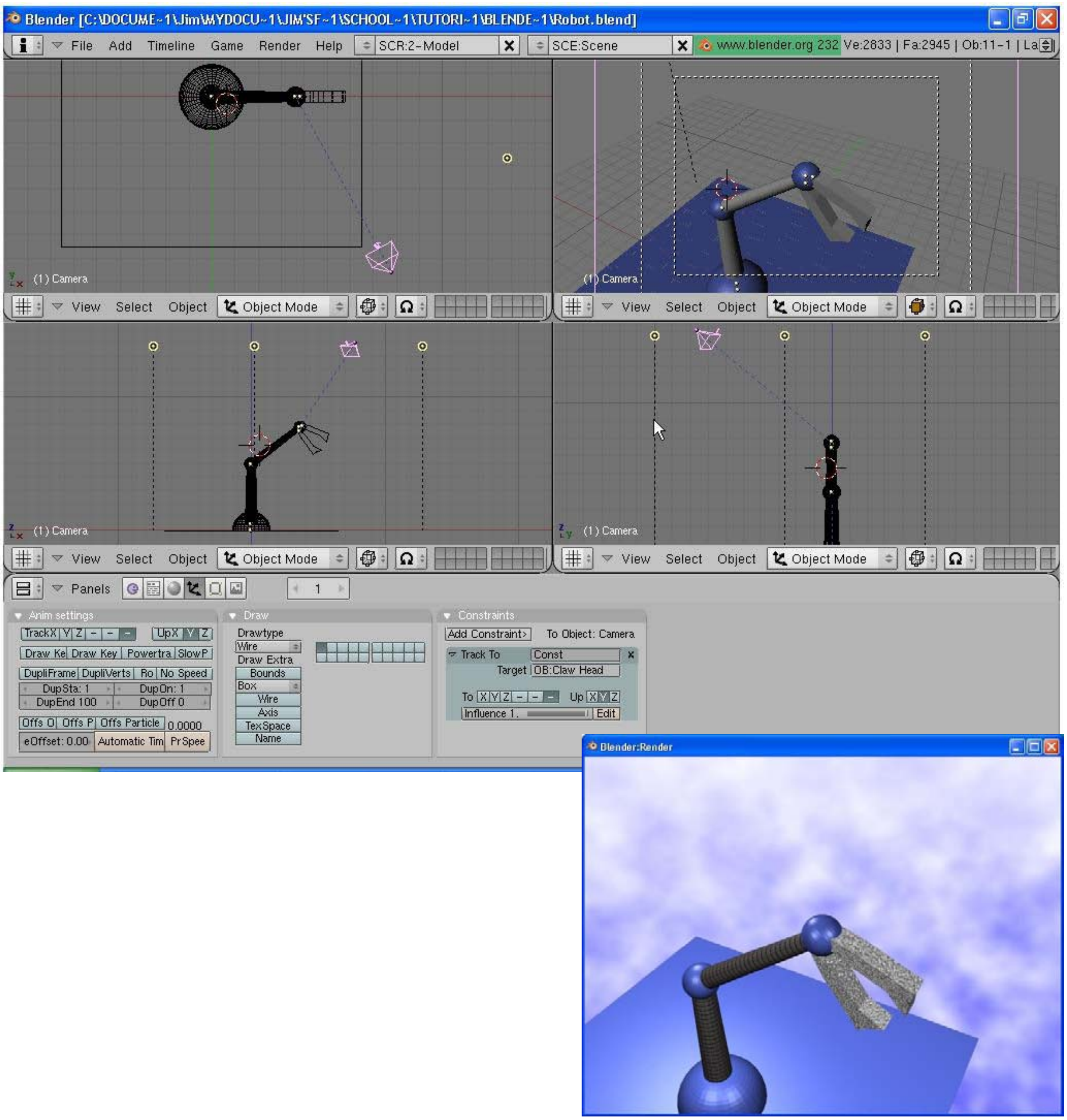


Sometimes it’s convenient to target an *Empty* object (created in the *Add* menu). This allows you to move your target around in your scene so the camera can focus on one object for a while, then move to something else by moving the target in that direction. You also have an *influence* option where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

Camera Constraint 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.

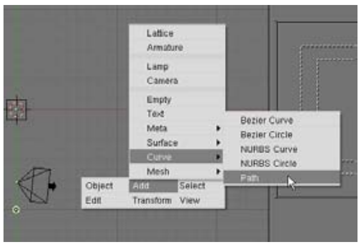


***** Call the instructor when finished*****

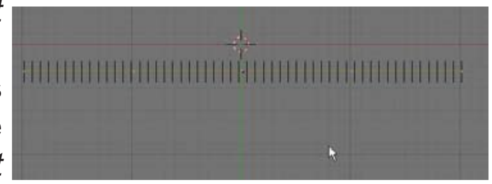
Working with Paths and Curves

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.

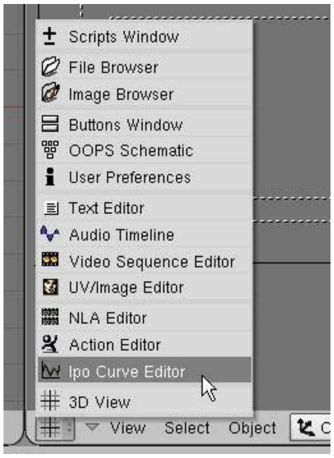


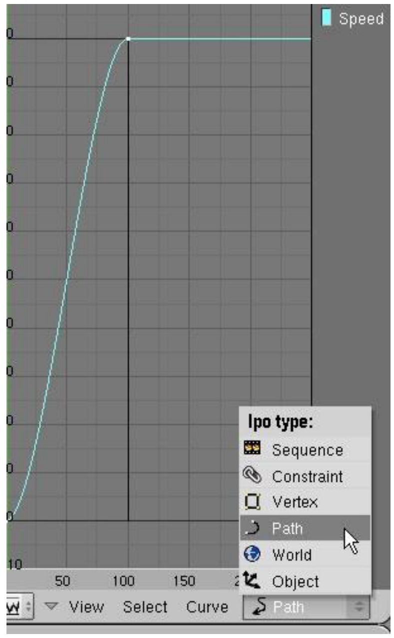
There are several ways to get the camera, object or lamp to follow the path. For now, we'll stick to the traditional way by creating a *child-parent relationship*. Select the object first, then the path (the parent). With both objects selected, press *"Ctrl and "P"* to make a parent. You'll have 2 options: *"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.

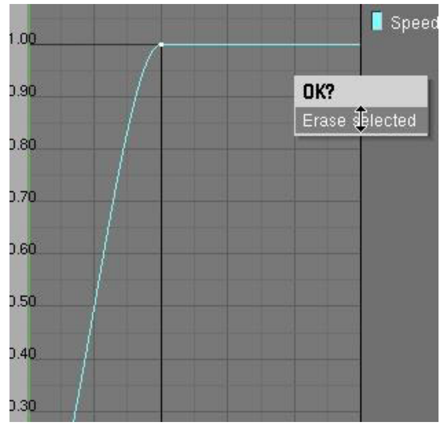
After you press the *Path Follow* button, the camera needs to be rotated and adjusted to the correct direction. After that, it will follow the path.

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



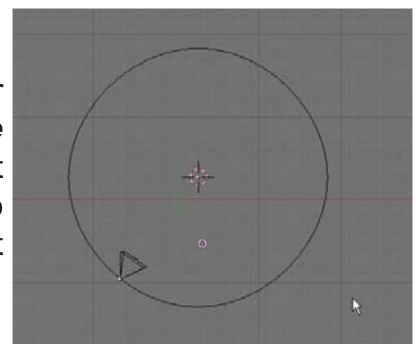


You will see the "*Speed*" track after you switch the IPO type to Path.



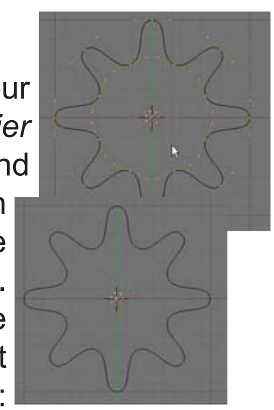
After you select and delete the speed path, the path length option will work in the Edit buttons.

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 constraint on the camera so it looks toward an object as it moves along the path.



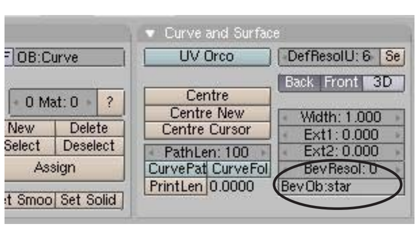
Other Curve objects can be used as paths also. For example, if you want a circular path, select the *Bezier Circle* option from the *Curve* menu. The *Curve Path* button is not automatically pressed when you child-parent the object to the circle though, you must go into the edit button and do it manually.

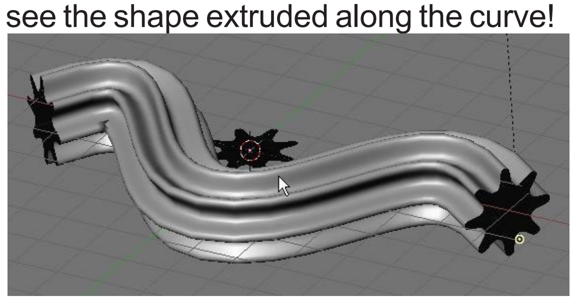
Using Curves for Extrusions



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*



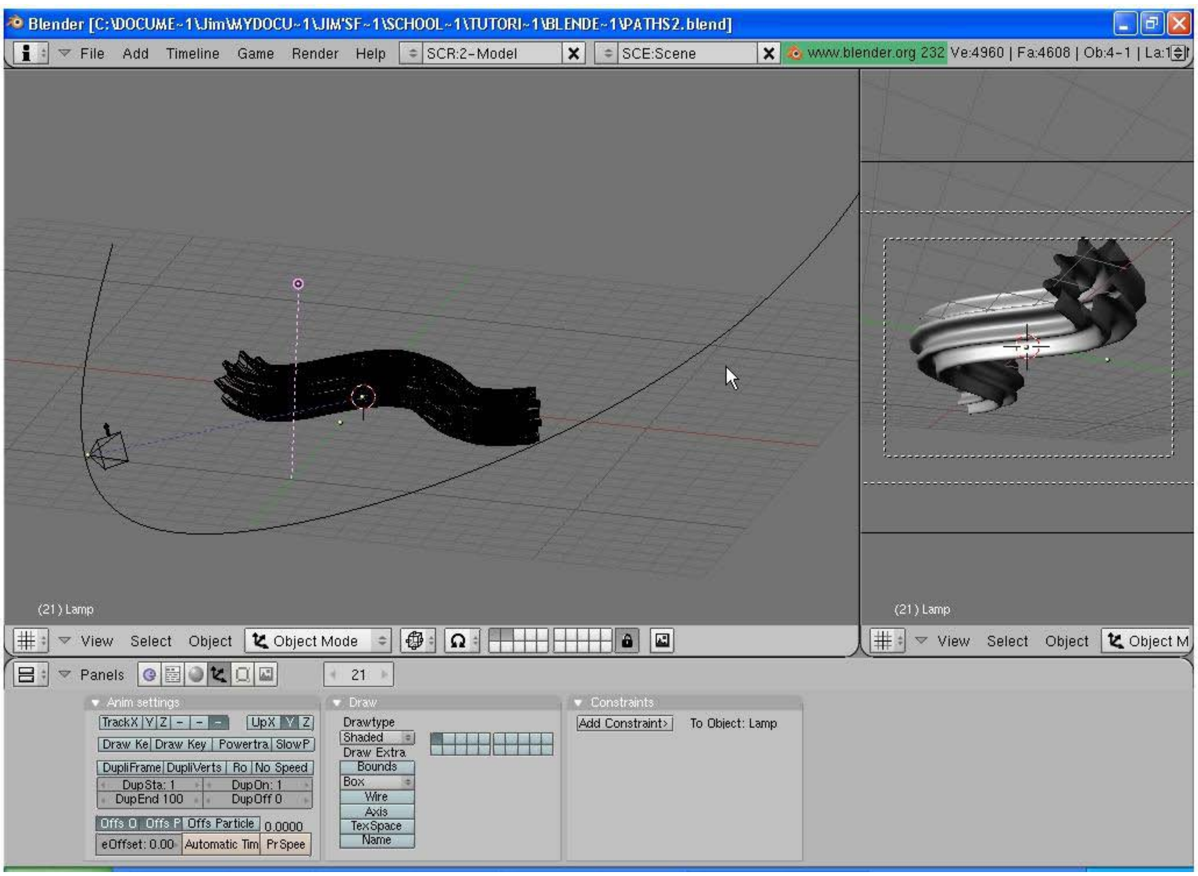


Paths and Curves Practice Exercise

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.

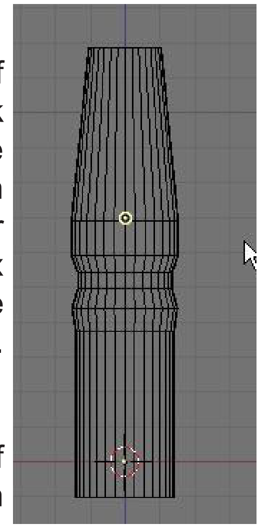
Challenge exercise:
After you do the required exercise, make a new one. Before you extrude your shape along it's curve, duplicated the curve and use it for the camera path. Place the path directly in the middle of the extruded shape to make the camera flow through the "tube".



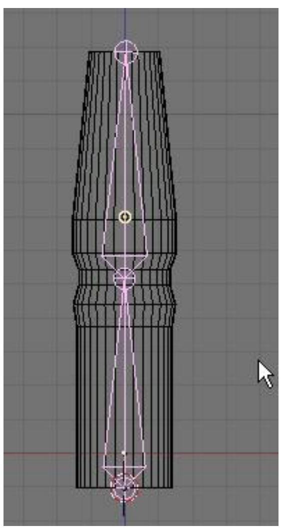
***** Call the instructor when finished*****

Armatures (bones)

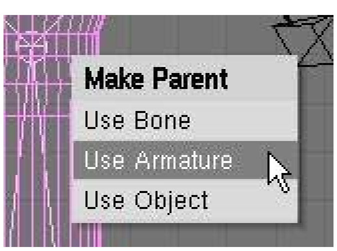
Blender's animation capabilities are great for most object animation except when you want to animate something bending like a person in motion or a tree bending in the breeze. This calls for a mesh to deform which can't be done with traditional modifiers. We can deform a mesh in 2 ways in Blender. One 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 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. Move your cursor up to lengthen the bone and click where you would like the joint to be. Another bone begins to form. Continue up to the top with that bone. 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.

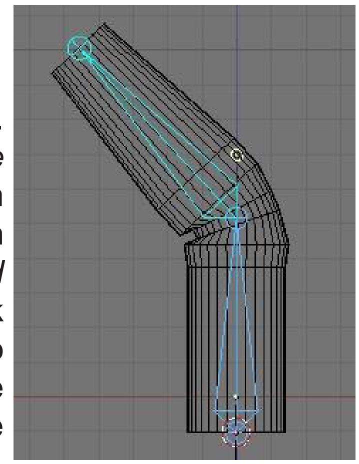


Your next step is to create a *child-parent* relationship between the mesh and the armature with the mesh being the *child* and the armature being the *parent*. Hold down the *"shift"* key and select the mesh first, then the armature. Press *"Ctrl"* and *"P"* to make parent. Select the option to *"Use 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!

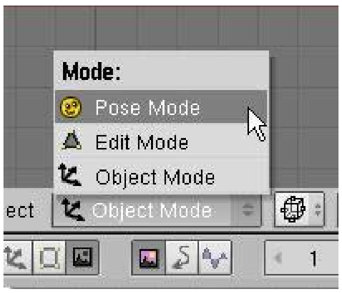
To create entire skeletons or other complex armature structures, you can do the following:

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 texture 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.



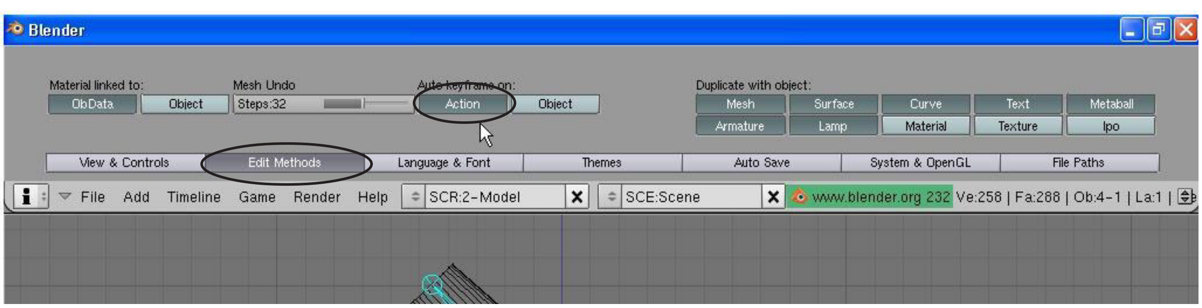
To animate you armature:



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.

Automatic Keyframing with Armatures:

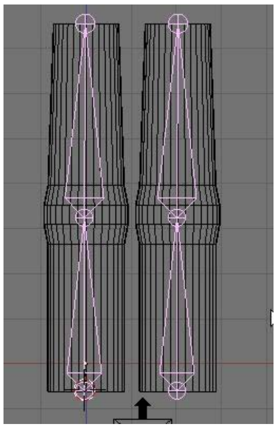
Placing animation keys on a complex armature system can be time consuming and very easy to miss a bone in a frame when you need to place a rotation key on 20 bones. That's why there's an automatic keyframe option in the top *User Preferences* window. Pull down the top menubar to expose the setting. Select *"Edit Methods"* and turn on the Auto Keyframing button called *"Action".* 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.*



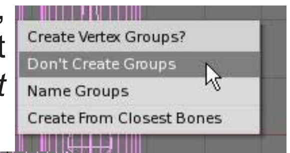
Even if a bone isn't moving on a particular plane, move it slightly so the automatic keyframing places a key on it. Otherwise, it may move when you don't expect it to because it was missing a key. Experiment with the features to become familiar with them.

Creating your own vertex groups:

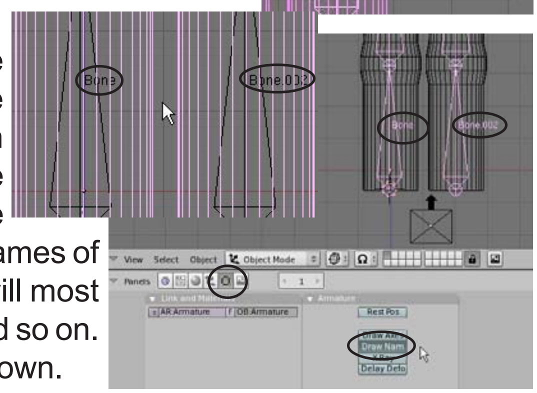
As mentioned before, sometimes when verticies are close together, Blender may have a dificult time creating clean vertex groups automatically. You will need to define the vertex groups manually. First, create your mesh and armature, then *child-parent* them to *"Use Armature"* as before, but then select *"Don't Create Groups"*. This will cause them to be parented, but not defined.

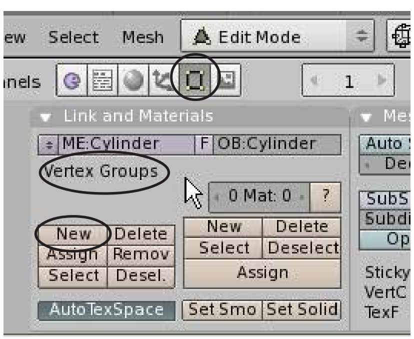


For this example, I've created a mesh we'll call *Finger*. I then created an armature and duplicated both to create 2 fingers. The next step is to *Join* (*"Ctrl" and "J"*) the 2 armatures together, then do the same for both meshes. Child-parent the mesh to the armature, select the *"Don't Create Groups"*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



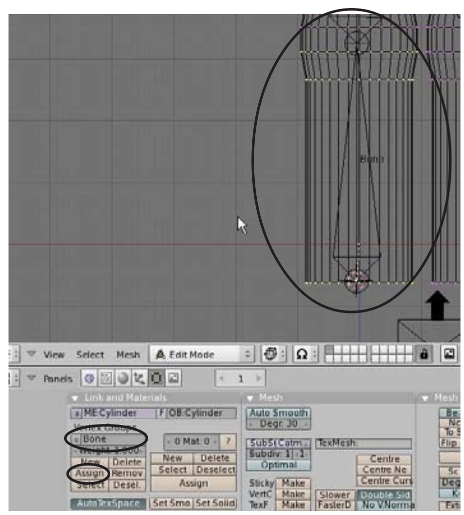
Now we need to see the names of each bone in the armature so we can assign verticies to them. Select the armature, then go to the edit buttons and find the "Draw Name" button. Pressing it will cause the names of each bone to be displayed on the screen. The will most likely show up as Bone, Bone.001,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



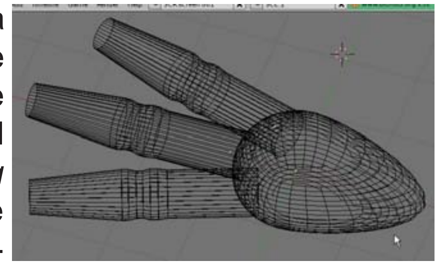


Now select the *mesh* and go into *Edit Mode* (*Tab* key). In the edit buttons, you will notice a group of buttons for creating *Vertex Groups* (this block of buttons will only be displayed if you're in edit mode). Select the *"New"* button and you are ready to create a new group of verticies for a bone. 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).

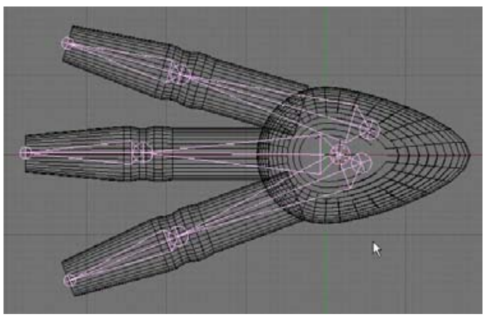
Our next step is to select the verticies that need to be assigned to that bone. If a group of verticies is right at the joint, they need to be selected for both bones. After selecting all the verticies that belong to that bone, press the *"Assign"* button. You've now made a vertex group for that bone. 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.



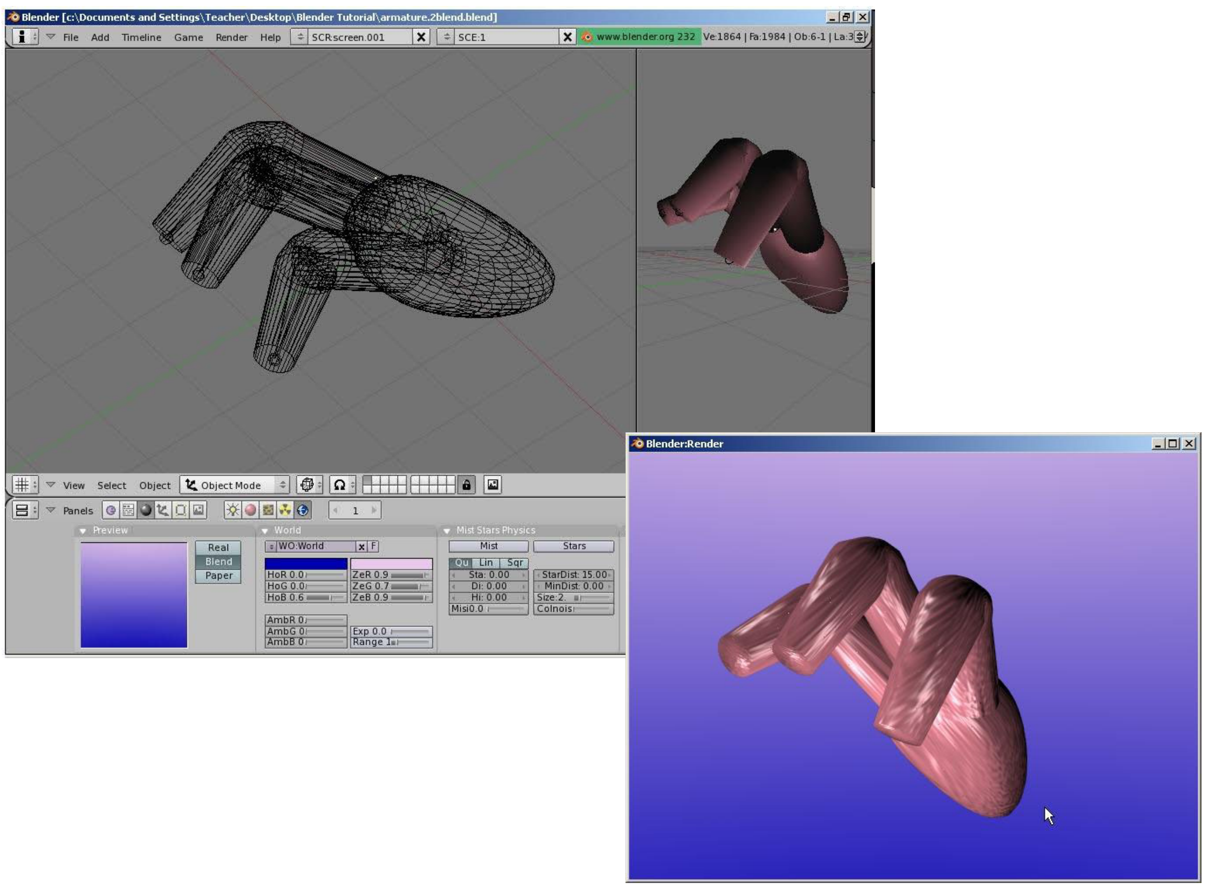
Working With Armatures Practice Exercise



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.



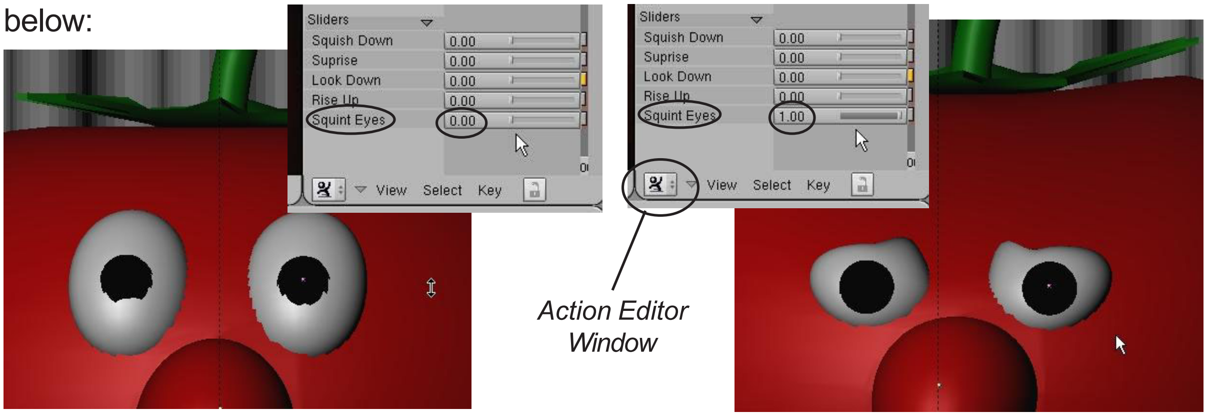
Challenge exercise:
Try to use armatures to animate some other object. Try a simple body that walks.



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

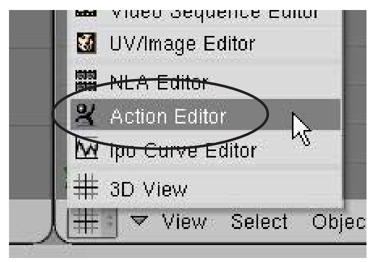
Relative Vertex Keys (mesh deformation)

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, represent 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.

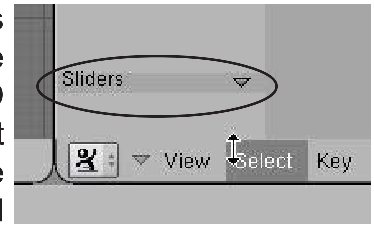
The first step in creating Relative Vertex Keys (RTVs) is to start with a mesh you wish to deform. For our case, we'll create a UV Sphere set at the default segments and rings of 32. Split the 3D window into 2 viewports and set the right-hand viewport to the *Action Editor Window.* This is another type 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.

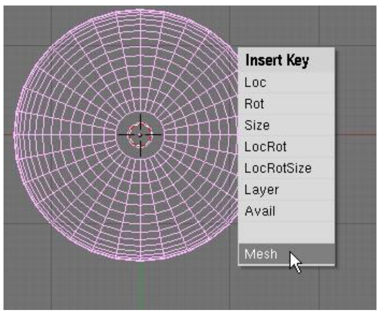


Let's go back to the left-hand viewport (still set to the 3D view window) and begin creating RVKs. Unlike normal animation that requires you to move to different frames along the timeline, we will be creating all our different sliders and mesh deforms on frame one. After the sliders are all created for our mesh, then we will use them in the Action Editor window along the timeline.

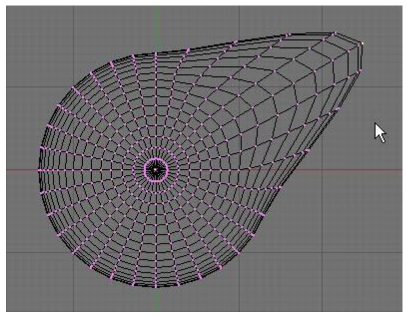
With the sphere selected, make sure you're **NOT** in edit mode, but in object select mode. (*Tab* key). Hit the *"I"* key to insert a *Mesh* key. The first time you hit the *Mesh* key for that object, you will get an option for *Relative Keys* or *Absolute Keys*. Select the *Relative Keys* option.

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:

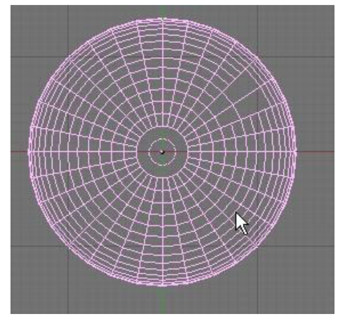




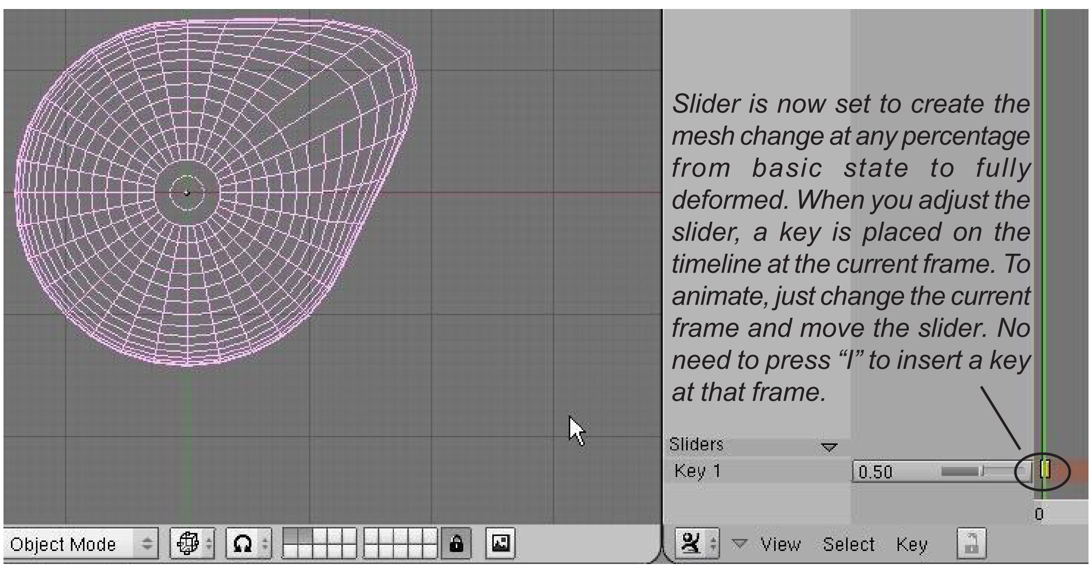
In object mode, press "I" to insert a Mesh key



Enter edit mode (Tab) and modify mesh

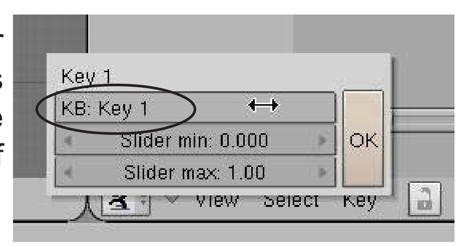


Exit edit mode (Tab), mesh goes back to basic state, slider now functional.



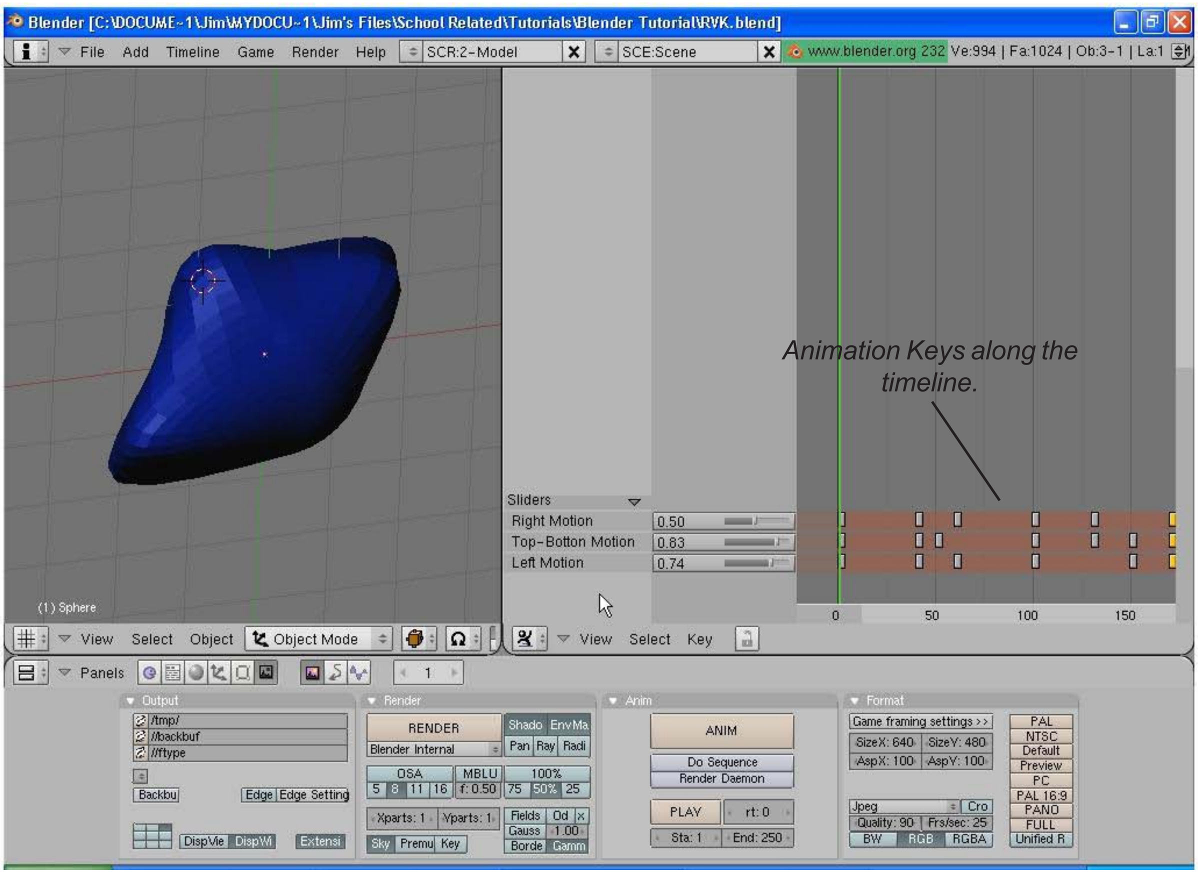
The basic thing to remember about RVKs is that in order to create the slider, You must insert the key in object selection mode, then enter edit mode (Tab) to modify the mesh. When you exit edit mode, the slider is set. Repeat the process to create all your sliders.

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. This window comes up where you can name your slided. You can also adjust the min. and max. of your slider.



Relative Vertex Keys Practice Exercise

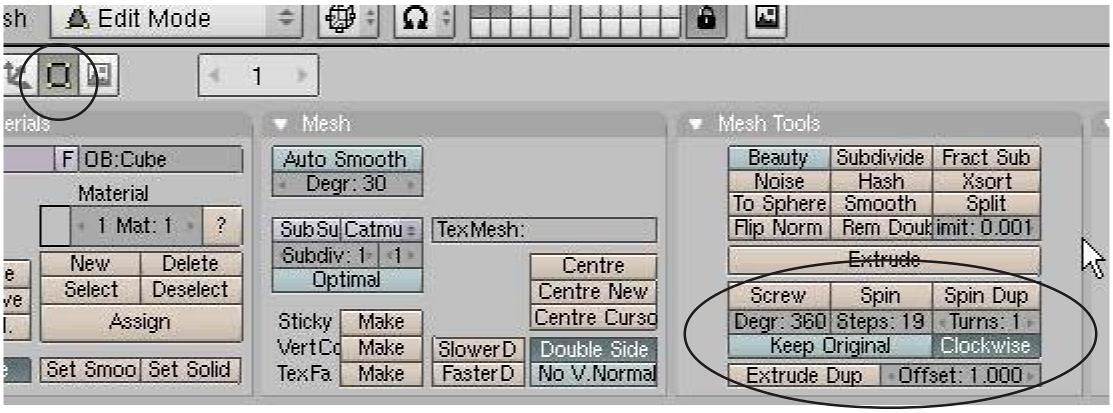
Create a new Blender file and name it *RVKs.* This exercise can be as simple as deforming a sphere with RVKs or as complex as trying to create a face and make it talk or show expression. Create a scene with adequate lighting, world settings and materials. Create at least 3 RVK mesh sliders and use them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



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

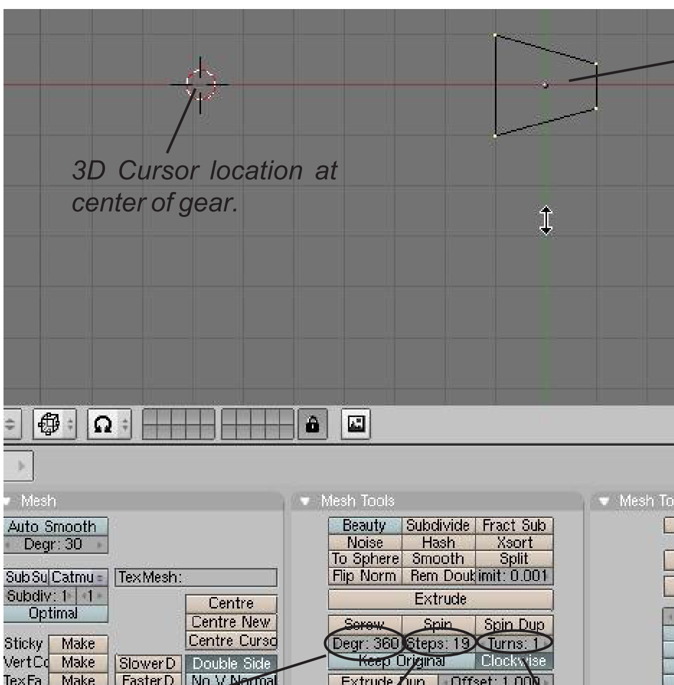
Creating Springs, Screws and Gears

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

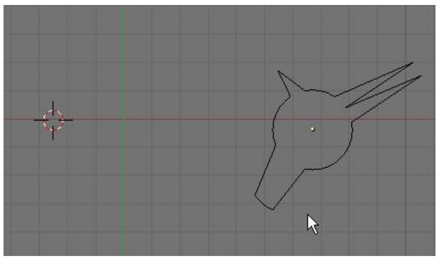
Steps- number of instances. In our case number of gear teeth.

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

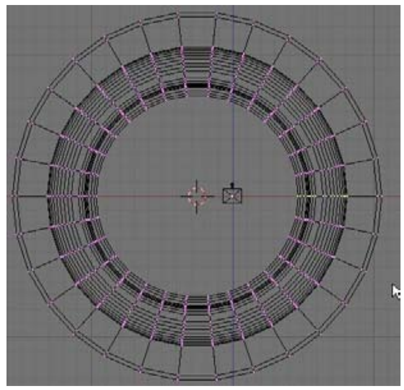
With all verticies selected and everything set to these numbers, hit the "*Spin Dup*" button and see the results. You will need to experiment with sizes and number of steps to get a good gear without any overlaps. Remember that edit mode has an Undo command. If it isn't right, just hit "*U*" and it will 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.

Spin

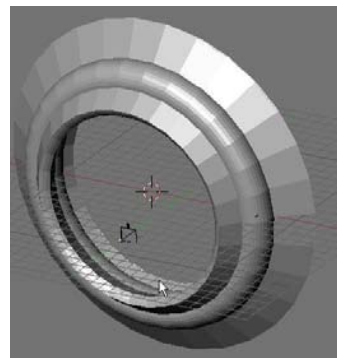
The *Spin* command operates similar to the *Spin Duplicate* command, except that it works more like a revolve-extrude command. You can take a plane or a circle, shape it, then revolve it around the 3D Cursor location. For our example, we started with a *Mesh Circle* in the *Top View*, then we placed 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

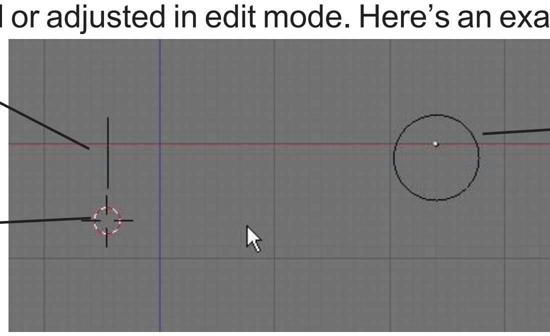
Screw Tool

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:

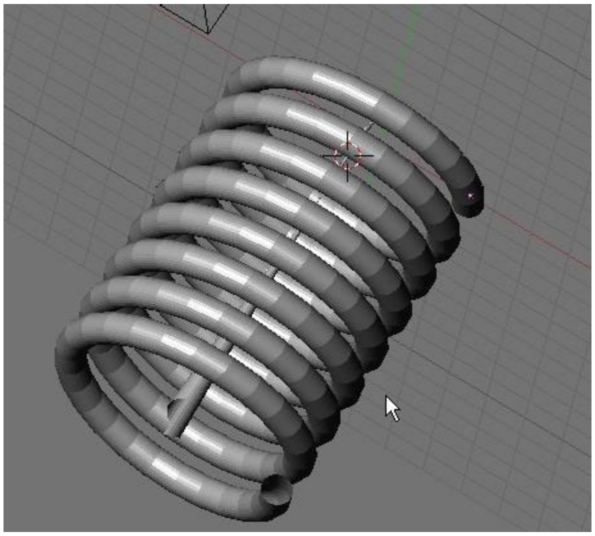
Here are the 2 verticies that form the line to controls spacing

3D Cursor location

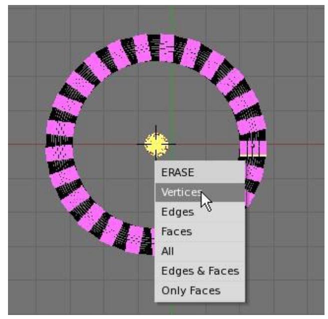


Circle that forms profile of the spring. Any shape can be used.

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:

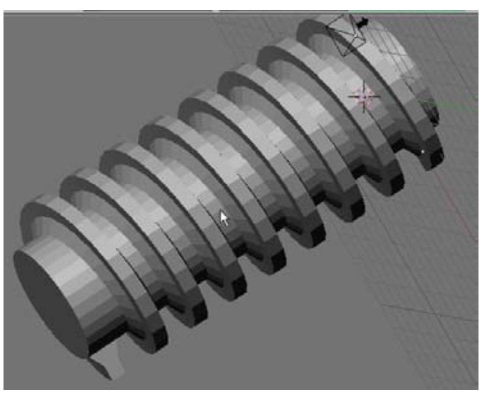


You will notice a shape in the middle of your spring. That shape is created from the verticies you used to designate the length of each loop. You placed this shape in the middle so it is easier for you to select these verticies in edit mode and erase them. Switch to a top view and enter edit mode. Select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

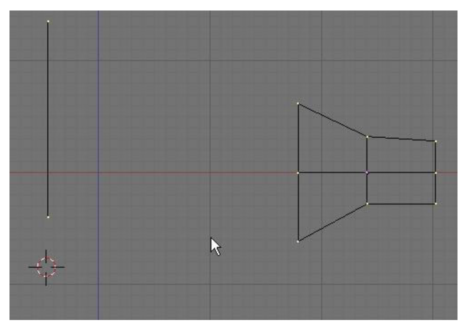


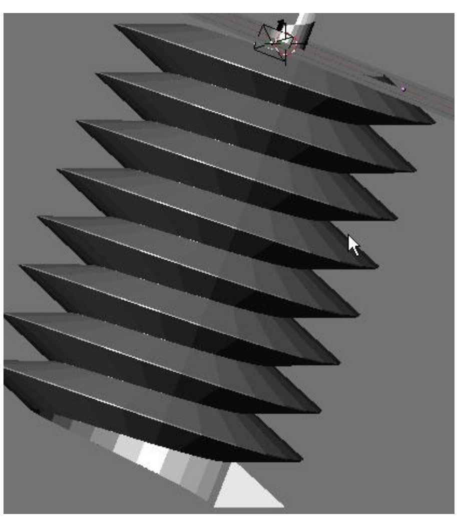
<u>*Other Shapes:*</u>

Here's how you can create some other things:

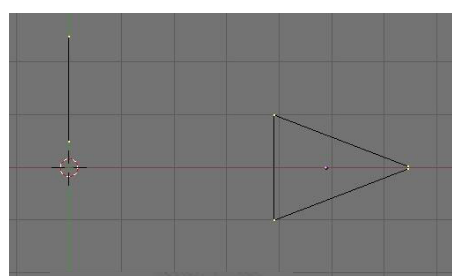


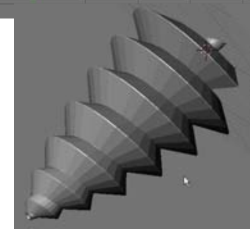
A Worm Gear created using a subdivided plane as the profile and the line set to equal twice the height of the gear profile. A cylinder has been added through the center to give it a solid appearence.





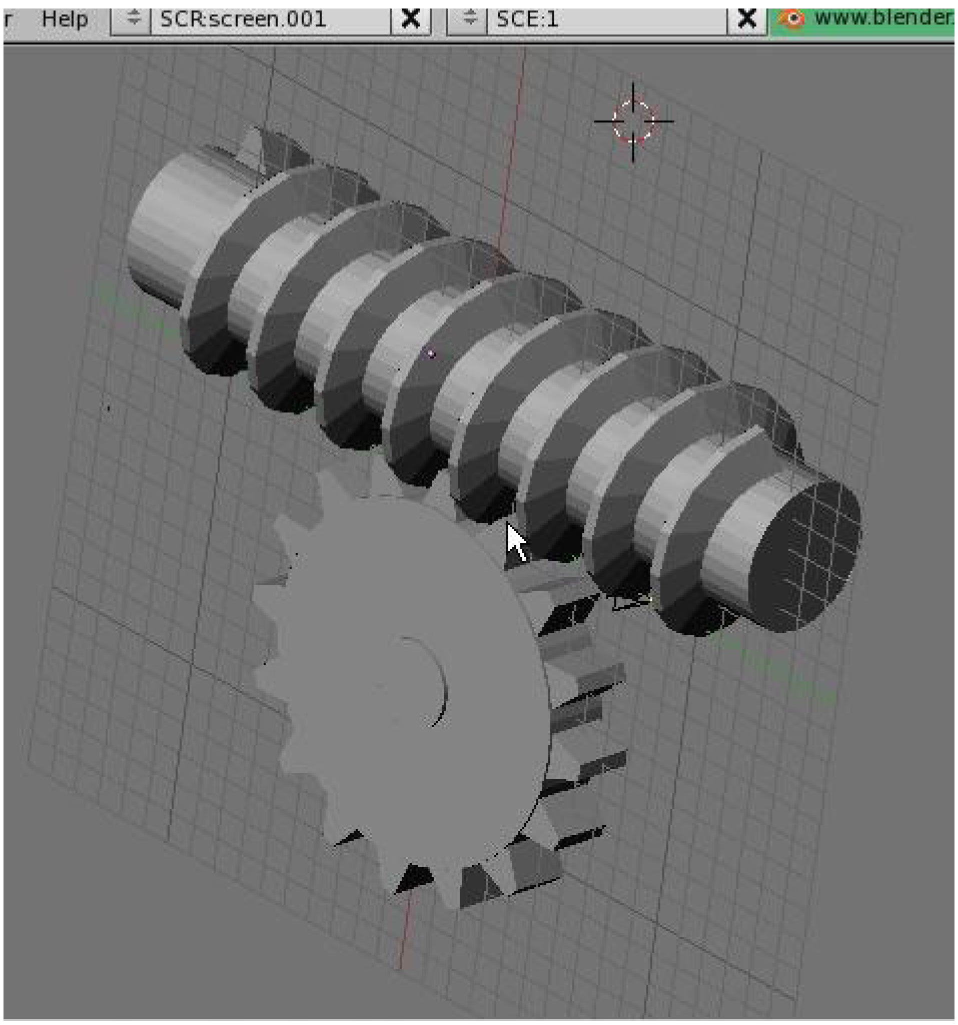
Bolt and screw threads can be created with a plane shaped into a triangle and the line for spacing set to exactly the same length as the base of the triangle. To make a pointed screw, grab the end thread verticies and scale it with proportional vertex editing ("O") for a nice look.





Gear Design Practice Exercise

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*****

Basic Game Engine Description and Set-up

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.

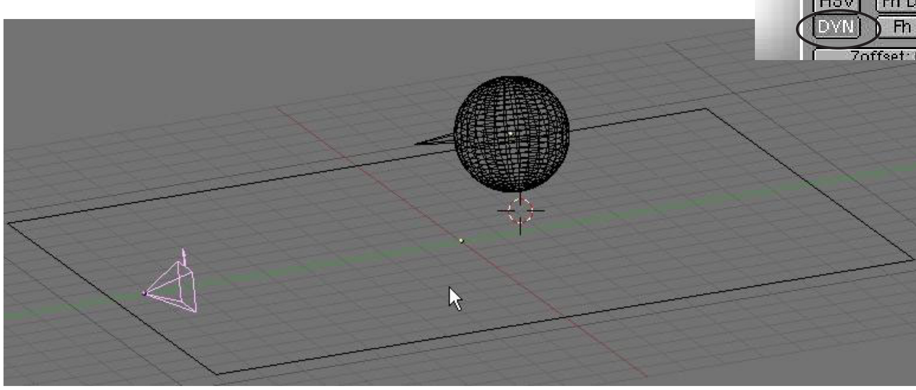
You can create games that look as good as professionally produced 3D games and real-time architectural walk throughs where doors can open and close as you approach them. The best part of this is that it can all be done *without computer programming skills.* There are other freeware game creation programs 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 not been officially placed back into the program so we need to use *Blender Publisher (2.25)* to create the games. The foundation is working on several physics engines and should have them back in very soon.

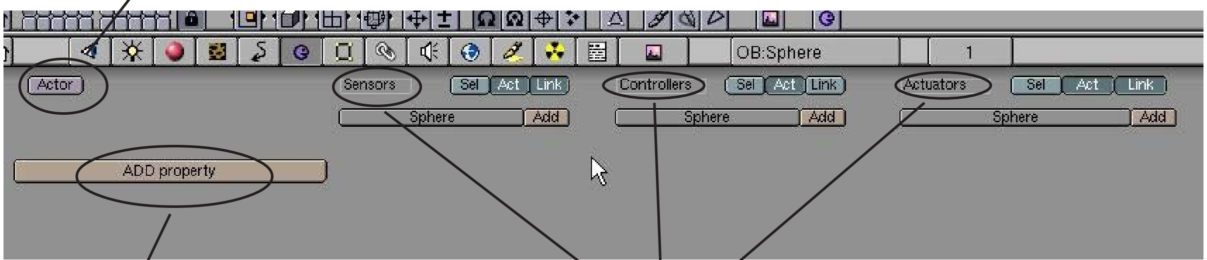
Let's make a simple scene consisting of a plane and a sphere and set the sphere above the plane Modify the sphere by pulling one vertex out to form a nose. This will let us know which way is forward when we move it around. Add a material to each one, but don't bother changing any material settings. The 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:

<u>DYN</u> (Dynamic) settings in the material buttons



Select the sphere and go to the Game Buttons (little purple pacman button). Here's what you see in the buttons window:

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:

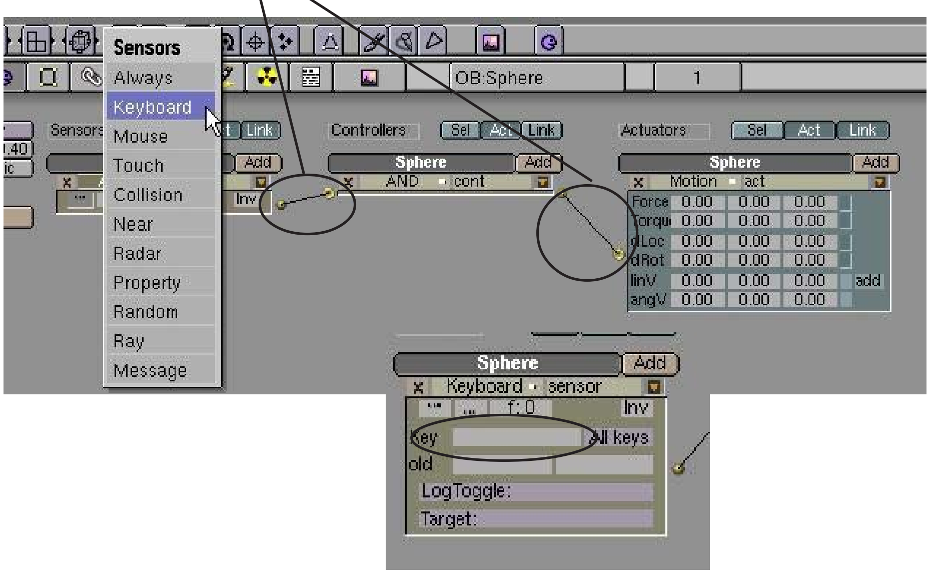
Damp: motion dampening- keeps the object from continuing forever when you stop applyiing force. I like to set this to around 0.4

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 "Add" 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.

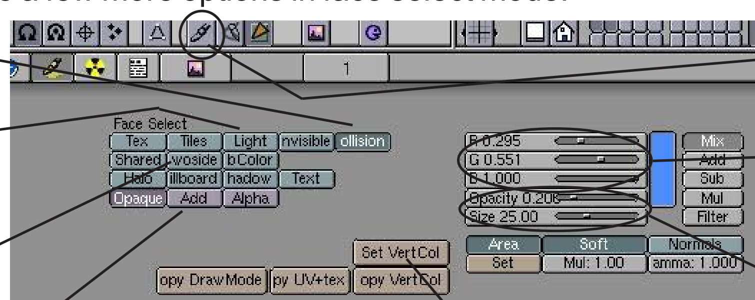


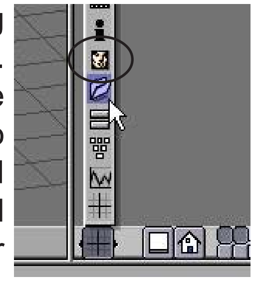
Now we'll apply a force to the actor. You will see three columns in the Motion block. They represent X, Y, and Z. The best way to change numbers in these blocks is to hold down "*Shift*" and click in the box. In the *Force* block, Let's change the Y number to 10. This is where you need to experiment 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.

Materials in the Game Engine

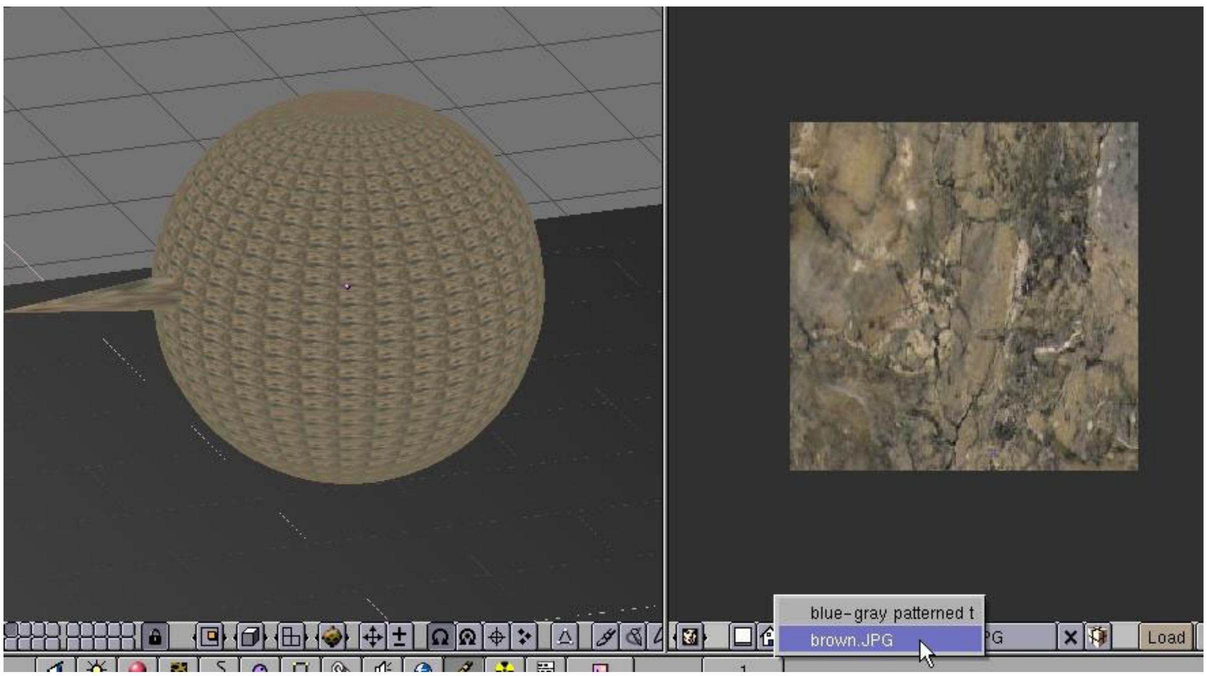
Wireframe is good for testing out motions, but poor for actually playing the game. Pressing the "*Alt" and "Z*" keys will place you in game textured mode. However, when you hit "*P*" to play now, everything looks horrible. Time to add *some game shading and UV Textures*.

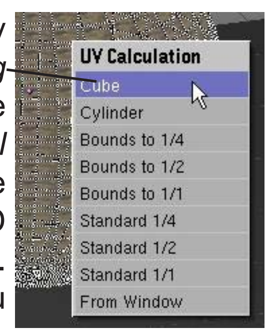
For straight color, you have several options to make it show up in game shading. Here's the easiest way to go plus an option to add lighting effects. Select the Sphere and type *"F"* for *Face Select*. This will highlight all the faces on the object. You have the option to select individual faces and 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 *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:



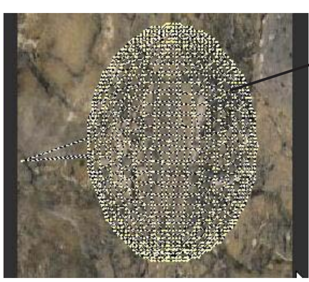


Now that we can add straight color, lighting effects and painting highlights, lets discuss adding textures throught the *UV Texture* options. If you haven't split your 3D window yet, do so now. We will need to use the right-hand viewport to set the textures. Change that window type to the *UV Texture* 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 Texture 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.

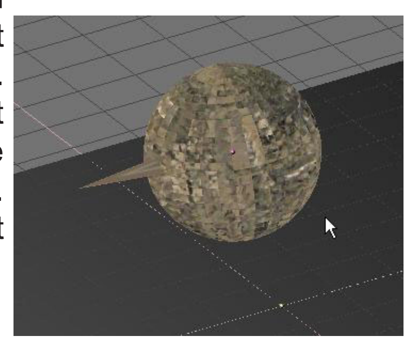




Looks good, but you may not want the image mapped small on every face. Place your cursor in the 3D window and type *"U" for UV Mapping* options. Select *Cube* and a size of 1.00. Now you see a sample of the

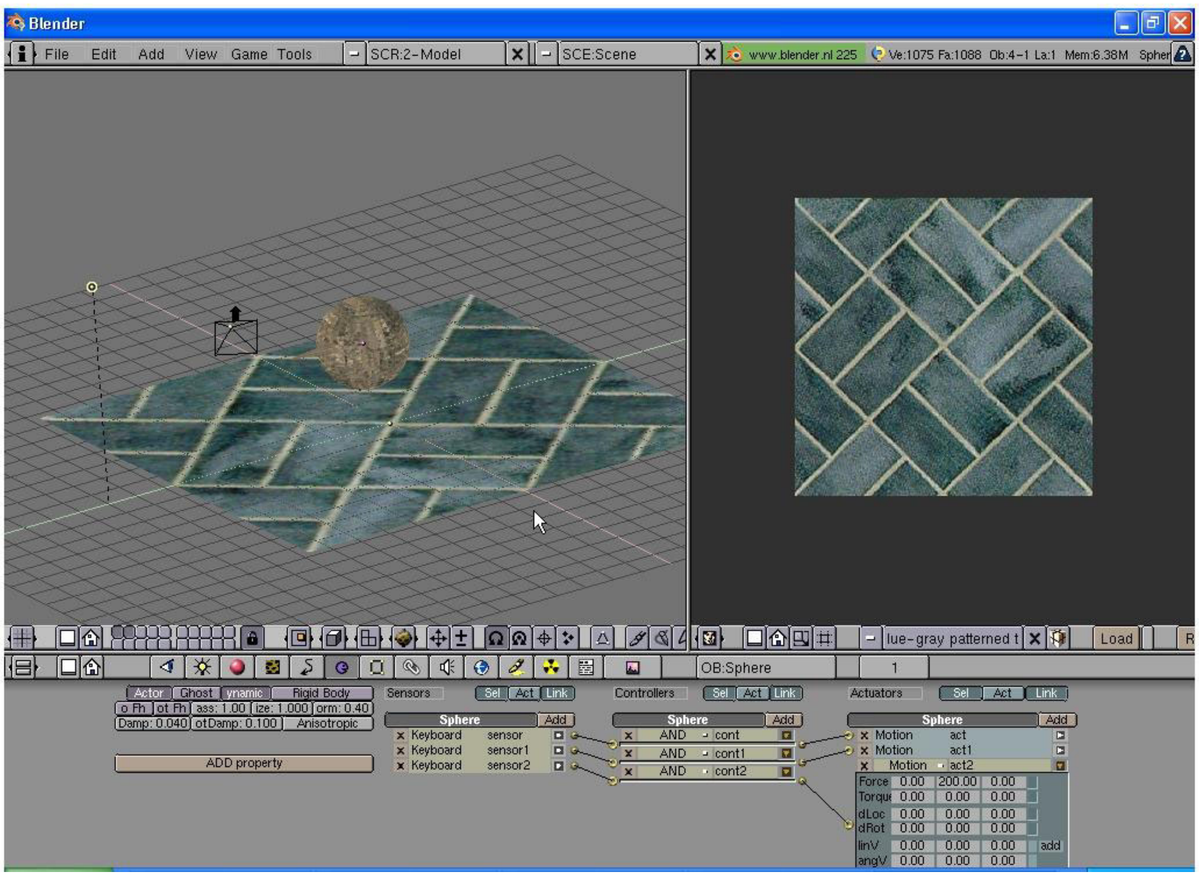


verticies in the texture window. Select *All* verticies, move them and scale them to fill the sample. Place your cursor back in the 3D window and Press *"F"* to exit face select mode. Press "P" to play and try out your model. If you would like to add lighting effect and some shading through vertex paint 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! 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!



Game Engine Practice Exercise

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.



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