

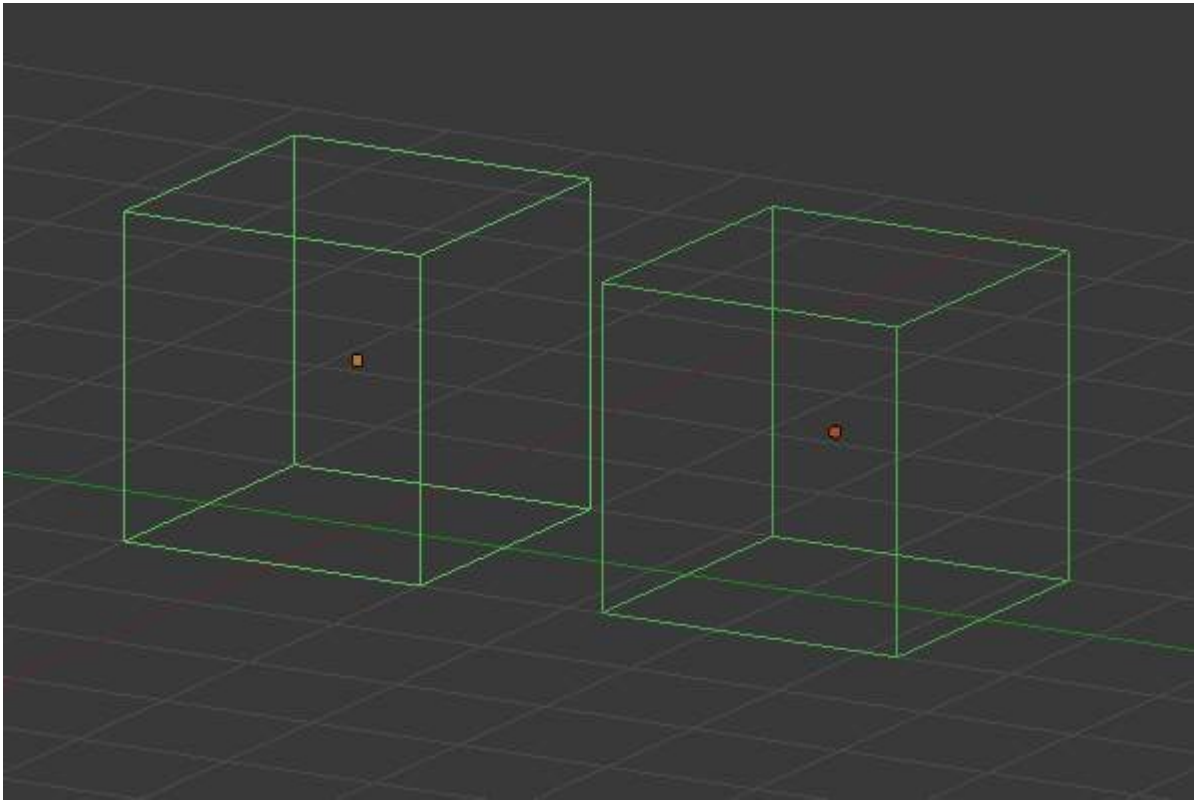
3.2.6 Editors - 3D View - Relationships

Relationships.....	1
Grouping objects.....	1
Options.....	2
Select Grouped.....	4
Parenting Objects.....	4
Object Parent.....	5
Object (Keep Transform) Parent.....	5
Armature Deform Parent.....	7
Armature Deform Parent With Empty Groups.....	10
Armature Deform With Automatic Weights.....	12
Armature Deform With Envelope Weights.....	13
Bone Parent.....	17
Relative Parenting.....	18
Vertex Parent.....	20
Vertex Parent from Edit Mode.....	20
Vertex Parent from Object Mode.....	20
Options.....	21
Move child.....	21
Remove relationship/Clear Parent.....	21
Hints.....	21

Relationships

- Grouping objects
 - Options
 - Select Grouped
- Parenting Objects
 - Object Parent
 - Object (Keep Transform) Parent
 - Armature Deform Parent
 - Bone Parent
 - Vertex Parent
 - Options
 - Hints

Grouping objects



Grouped objects

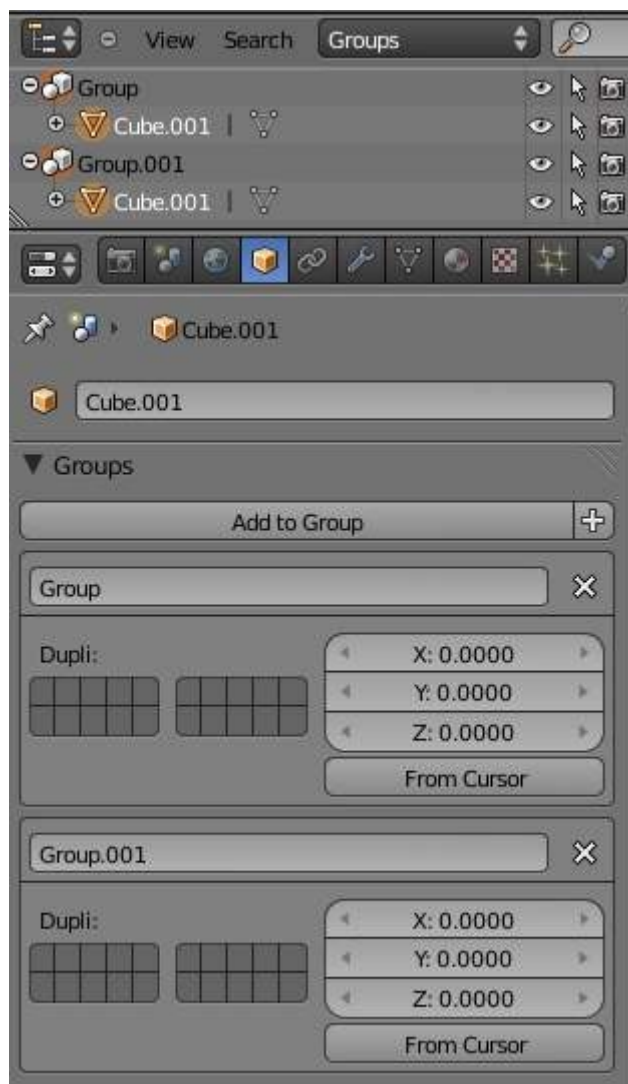
There can be many objects in a scene: A typical stage scene consists of furniture, props, lights, and backdrops. Blender helps you keep everything organized by allowing you to group like objects together.

Group objects together without any kind of transformation relationship. Use groups to just logically organize your scene, or to facilitate one-step appending or linking between files or across scenes. Objects that are part of a group always shows as light green when selected; see image (*Grouped objects*).

Options

Creating a Group

Ctrl-G creates a new group and adds the selected object(s) to it.



Naming a Group

Naming a Group

All groups that an object has been assigned to are listed in the *Object Properties Panel* 's *Relations* panel. To rename a group, simply click in the groups name field. To name groups in the *Outliner* window, select *Groups* as the outliner display from the header combo box, and **Ctrl**-**LMB** click on the group name. The name will change to an editable field; make your changes and press **Return**.

Restricting Group Contents via Layers

The cluster of layer buttons attached to each group determines from which layers the group objects will be included when duplicated. If your group contains objects on layers 10, 11 and 12, but you disable the layer 12 button in the group controls, duplicates of that group (using the *Dupligroup* feature) will only show the portions of the group that reside in layers 10 and 11.

Appending or Linking Groups

To append a group from another *.blend* file, consult *this page*. In summary, **File** ▸ **Link / Append Link** Select a *.blend* file and, and then the group.

Removing Groups

To remove a object from a group, under the object context button, open the “Groups” pane. Find the name of the group from which you wish to remove the object, and click the x to the right of the group name.

Select Grouped

Reference

Mode: *Object mode*

Menu: *Select → Grouped*

Hotkey: **Shift-G**

Select objects by parenting and grouping characteristics. See *Select Grouped* for more information.

Parenting Objects

When modeling a complex object, such as a watch, you may choose to model the different parts as separate objects. However, all of the parts may be attached to each other. In these cases, you want to designate one object as the parent of all the children. Movement, rotation or scaling of the parent also affects the children.

To parent objects, select at least two objects (select the *Child Objects* first, and select the *Parent Object* last), and press **Ctrl-P**. The *Set Parent To* dialog will pop up allowing you to select from one of several possible different parenting types. Selecting one of the entries in *Set Parent To* confirms, and the child/children to parent relationship is created.

The last object selected will be the *Active Object* (outlined in light orange), and will also be the *Parent Object*. If you selected multiple objects before selecting the parent, they will all be children of the parent and will be at the same level of the hierarchy (they are “siblings”).

The *Set Parent To* pop-up dialog is context sensitive, which means the number of entries it displays can change depending on what objects are selected when the **Ctrl-P** shortcut is used.

For non-inverse-mode, press **Shift-Ctrl-P** instead. This creates an alternative parent-child-relationship where child-objects exist entirely in the parent’s coordinate system. This is the better choice for CAD purposes, for example.

Moving, rotating or scaling the parent will also usually move/rotate/scale the child/children. However moving/rotating/scaling the child/children of the parent will not result in the parent moving/rotating/scaling. In other words, the direction of influence is from parent to child and not child to parent.

In general when using the **Ctrl-P** or [3D View Editor Header > Object Menu > Parent Menu] entires to parent objects, the *Child Objects* can only have one *Parent Object*. If a *Child Object* already has a *Parent Object* and you give it another parent then Blender will automatically remove the previous parent relationship.

Blender supports many different types of parenting, listed below:

- Object
- Armature Deform
- Bone
- Curve Deform
- Path Constraint

- Lattice Deform
- Vertex
- Vertex (Triangle)

Object Parent

Object Parent is the most general form of parenting that Blender supports. It will take selected objects and make the last selected object the *Parent Object*, while all other selected objects will be *Child Objects*.

Object (Keep Transform) Parent

Object (Keep Transform) Parent works in a very similar way to *Object Parent* the major difference is in whether the *Child Objects* will remember any previous transformations applied to them from the previous *Parent Object*.

Since explaining this in an easy to understand technical way is hard, let's instead use an example to demonstrate.

Assume that we have a scene consisting of 3 objects, those being 2 Empty Objects named “EmptyA” and “EmptyB”, and a Monkey object. See figure 1.



Figure 1 - Scene with 2 Empties and a Monkey, no parenting currently active.

Figure 1 shows the 3 objects with no parenting relationships active on them.

If you select the Monkey object by RMB click and then Shift-RMB click “EmptyA” object and press Ctrl-P and then select *Object* from the *Set Parent To* pop-up dialog box. This will result in “EmptyA” object being the *Parent Object* of the Monkey object. With only “EmptyA” selected rotating/scaling/moving it will result in the Monkey object being altered respectively.

Scale the “EmptyA” object, so that the Monkey becomes smaller and moves to the left a little. See figure 2.



Figure 2 - Scene with Monkey object being the Child Object of “EmptyA”. “EmptyA” has been scaled resulting in the Monkey also being scaled and moved to the left.

If you select only the Monkey object by **RMB** click and then **Shift - RMB** click “EmptyB” object and press **Ctrl - P** and select *Object* from the *Set Parent To* pop-up dialog box. This will result in “EmptyB” object being the *Parent Object* of the Monkey object. Notice that when you change the parent of the Monkey the scale of the Monkey changed. See figure 3.



Figure 3 - Scene with Monkey object having its a parent changed from “EmptyA” to “EmptyB” and the resulting change in scale.

This happens because the Monkey object never had its scale altered directly, the change came about because it was the child of “EmptyA” which had its scale altered. Changing the Monkey’s parent to “EmptyB” resulted in those indirect changes in scale being removed, because “EmptyB” has not had its scale altered.

This is often the required behaviour, but it is also sometimes useful that if you change your *Parent Object* that the *Child Object* keep any previous transformations it got from the old *Parent Object*; If instead when changing the *Parent Object* of the Monkey from “EmptyA” to “EmptyB” we had chosen parenting type *Object (Keep Transform)*, the Monkey would keep its scale information it obtained from the old parent “EmptyA” when it is assigned to the new parent “EmptyB”; See Figure 4.

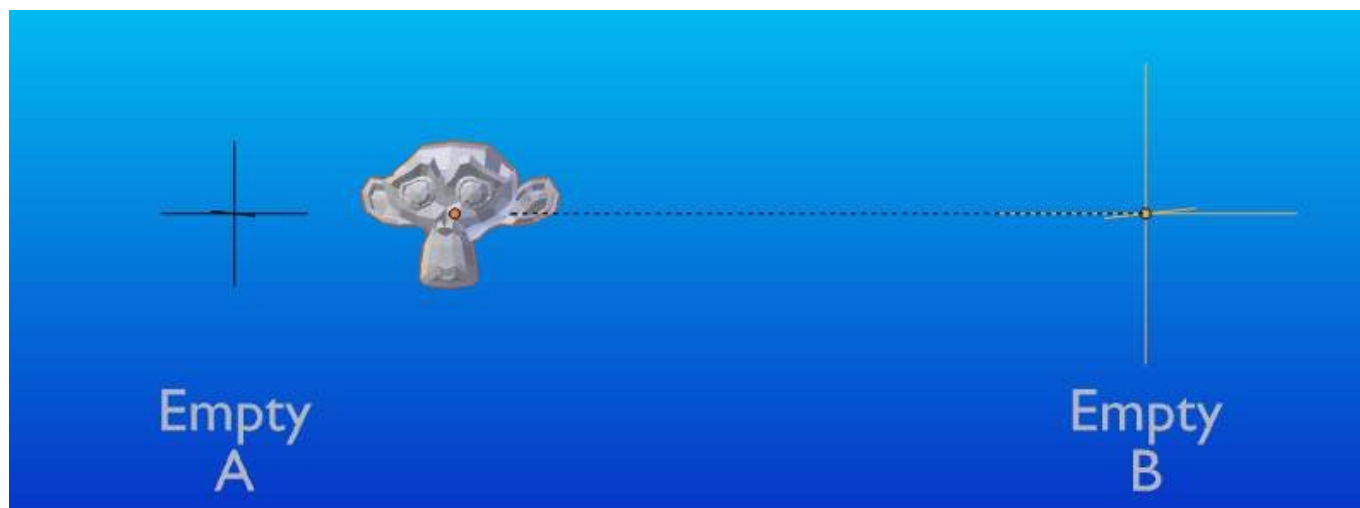


Figure 4 - Scene with Monkey object having its a parent changed from “EmptyA” to “EmptyB” using ‘Object (Keep Transform)’ parent method.

If you want to follow along with the above description here is the blend file used to describe *Object (Keep Transform)* parenting method:

File:Parent_-_Object_(Keep_Transform)_(Demo_File).blend

Armature Deform Parent

An Armature in Blender can be thought of as similar to the armature of a real skeleton, and just like a real skeleton an Armature can consist of many bones. These bones can be moved around and anything that they are attached to or associated with will move and deform in a similar way.

In Blender Armature Object Types are usually used to associate certain bones of an Armature to certain parts of a Mesh Object Types Mesh Geometry. You are then able to move the Armature Bones and the Mesh Object will deform. See figure 5.

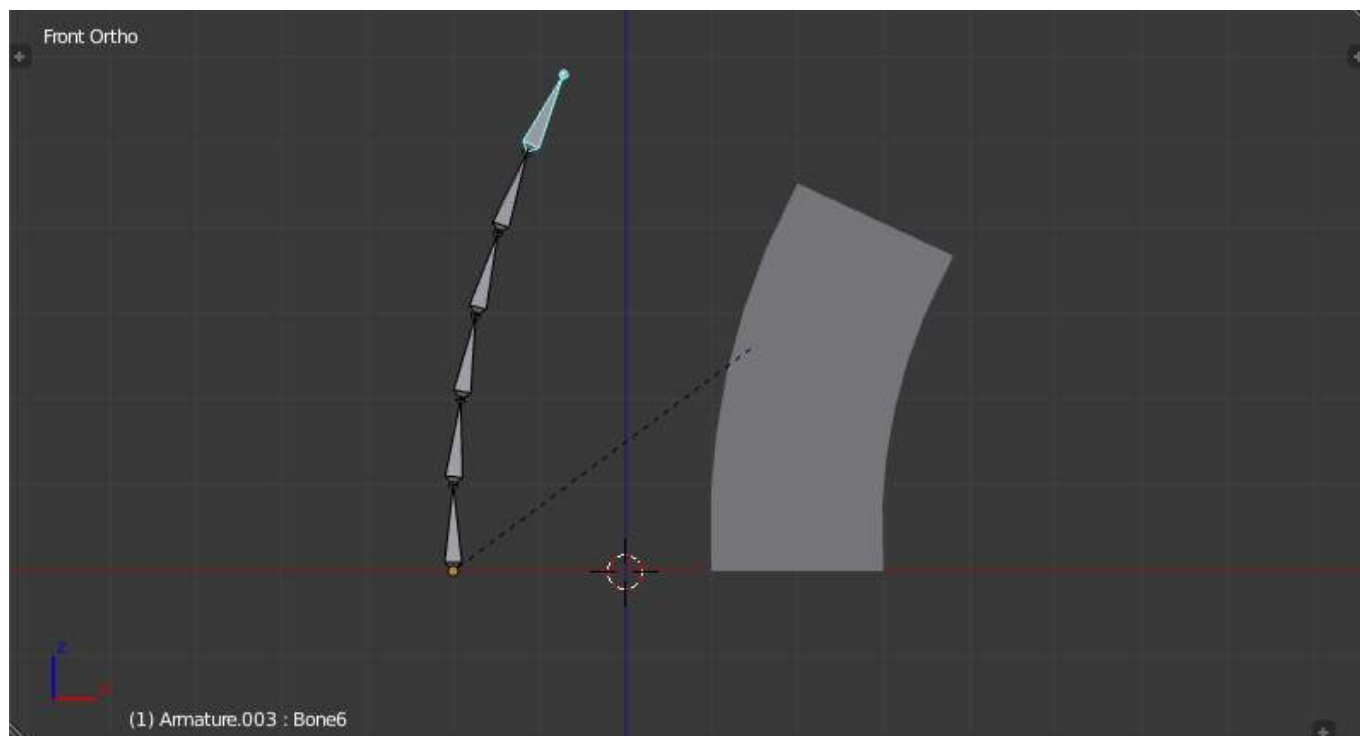


Figure 5 - Armature Object Bone associated with another Mesh Object, as the bone move the Mesh deforms similarly.

Armature Deform Parenting is one of the most flexible ways of associating Bones in an Armature to another Object, it gives a lot of freedom but that comes at the price of a little complexity, as there are multiple steps involved in setting up Armature Deform Parenting such that deformations are actually carried out.

Blender has several different ways of Parenting an Armature to an object, most of them can automate several of the steps involved, but all of them ultimately do all the steps we describe for Armature Deform Parenting.

Using the Armature Deform Parenting operator is the first step in setting up the relationship between an Armature Object and it's Child Objects.

To use Armature Deform Parenting you must first select all the Child Objects that will be influenced by the Armature and then lastly select the Armature Object itself. Once all the Child Objects and the Armature Object are selected press **Ctrl-P** and select Armature Deform in the Set Parent To pop-up dialog. See figure 6.

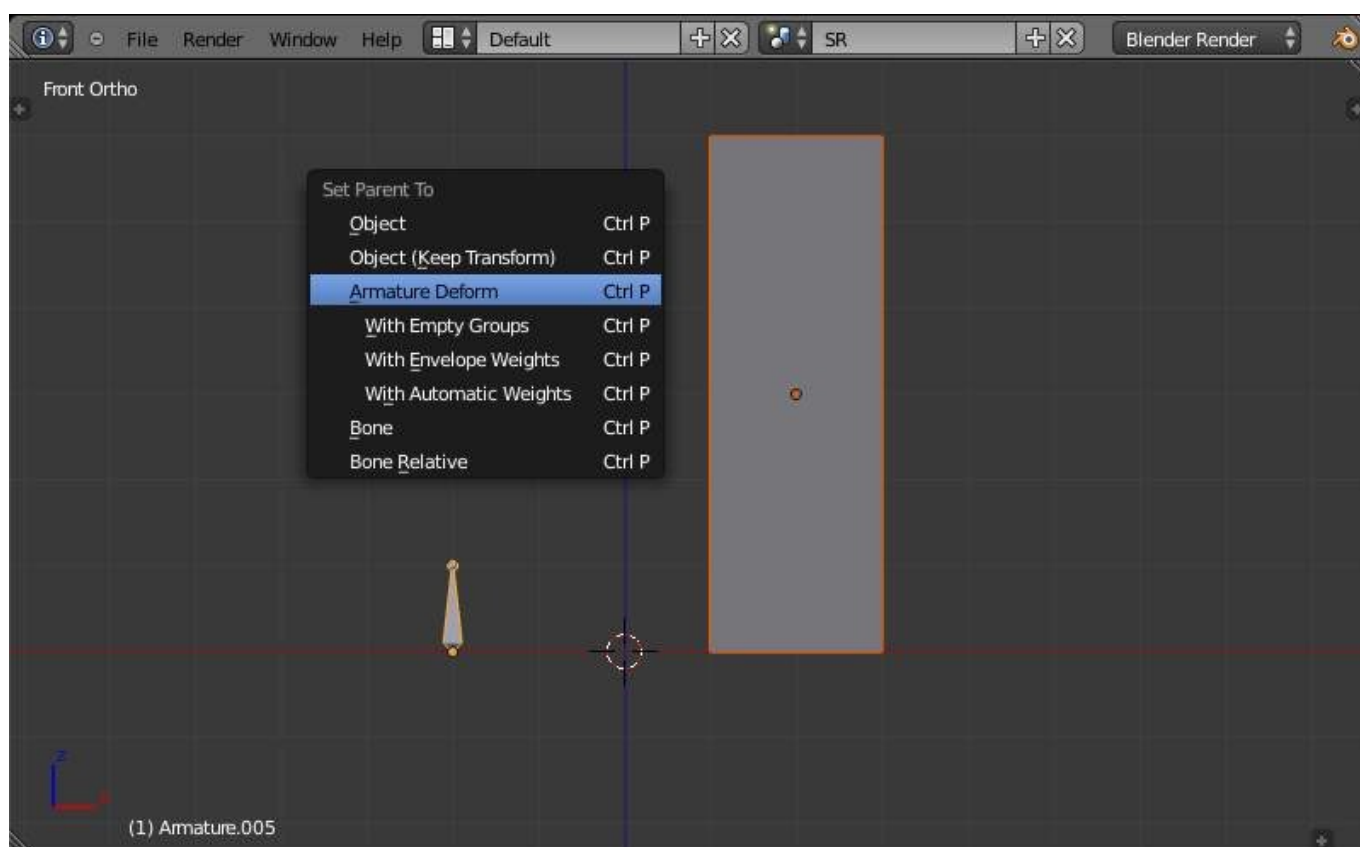


Figure 6 - Set Parent To dialog with Armature Deform Parenting option highlighted.

Once this is done the Armature Object will be the Parent Object of all the other Child Objects, also we have informed Blender that the Bones of the Armature Object can be associated with specific parts of the Child Objects so that they can be directly manipulated by the Bones.

At this point however all Blender knows is that the Bones of the Armature could be used to alter the Child Objects, we haven't yet told Blender which Bones can alter which Child Objects or by how much.

To do that we must individually select each Child Object individually and toggle into Edit Mode on that Child Object. Once in Edit Mode we can then select the vertices we want to be influenced by the Bones in the Armature. Then with the vertices still selected navigate to [Properties Editor > Object Data Context > Vertex Groups Panel] and create a new Vertex Group with the same name as the Bone that you want the selected

vertices to be influenced by.

Once the Vertex Group has been created we then assign the selected vertices to the Vertex Group by clicking the Assign Button. By default when selected vertices are assigned to a Vertex Group they will have an Influence Weight of 1.0 This means that they are fully influenced when a Bone they are associated with is moved, if the Influence Weight had been 0.5 then when the bone moves the vertices would only move half as much. See figure 7.

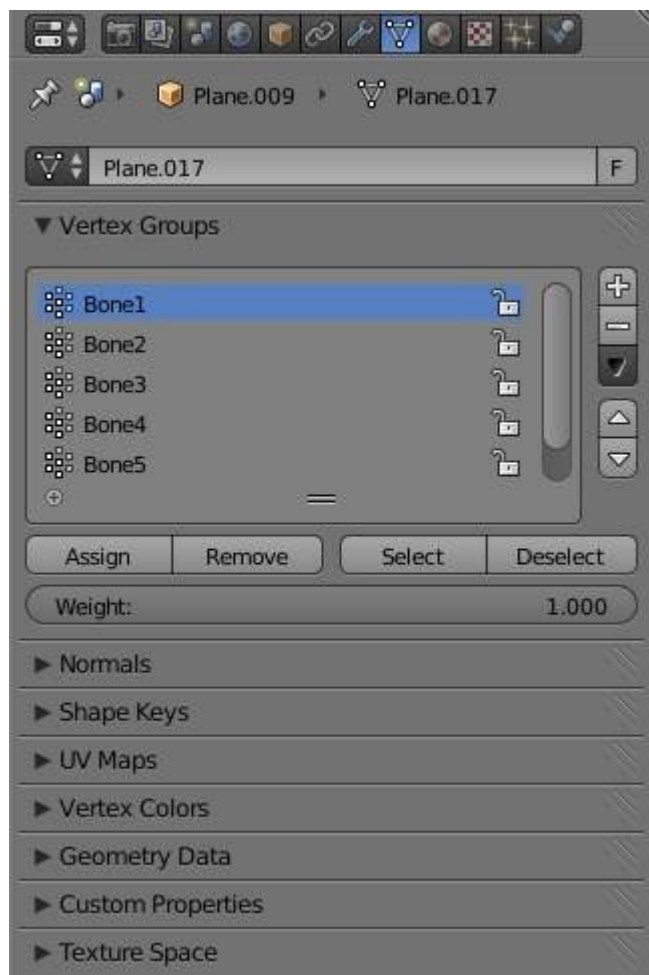


Figure 7 - Properties Editor > Object Data Context > Vertex Groups Panel with Assign Button and influence Weight Slider highlighted.

Once all these steps have been carried out, the Bones of the Armature Object should be associated with the Vertex Groups with the same names as the Bones. You can then select the Armature Object and switch to Pose Mode in the [3D View Editor Header > Mode Select Button]. See figure 8.

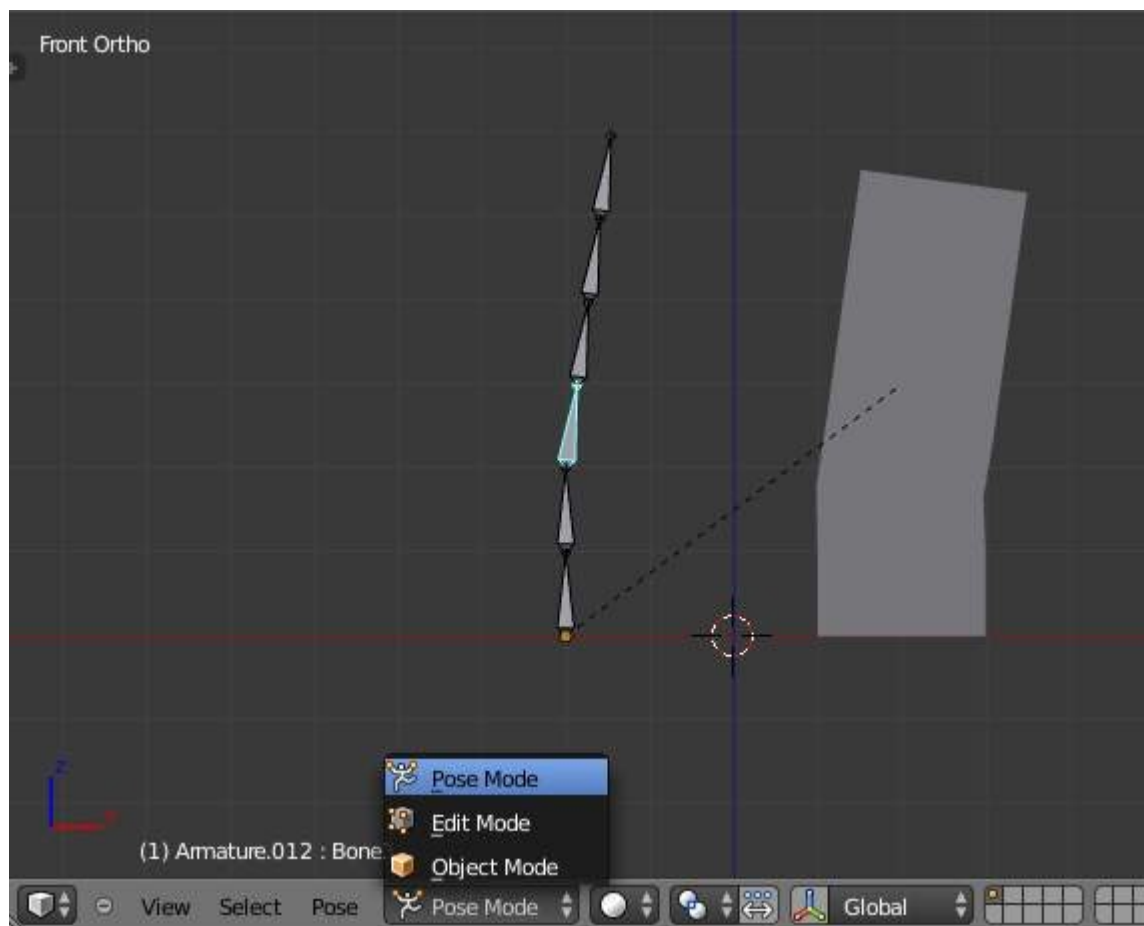


Figure 8 - 3D View Editor Header > Mode Select Button] set to Pose Mode, with Armature Bone highlighted in Cyan and effecting the Mesh Object

Once in Pose Mode transforming one of the Bones of the Armature that has been associated with vertices of an object will result in those vertices also being transformed.

Armature Deform Parent With Empty Groups

The Armature Deform With Empty Groups parenting method works in almost the same way as Armature Deform parenting with one difference. That difference is that when you parent a Child Object to an Armature Object the names of the bones in the armature are copied to the Child Objects in the form of newly created Vertex Groups, one for each different deforming armature bone name. The newly created Vertex Groups will be empty this means they will not have any vertices assigned to those Vertex Groups. You still must manually select the vertices and assign them to a particular Vertex Group of your choosing to have bones in the armature influence them.

For example if you have an Armature Object which consists of 3 bones named BoneA, BoneB and BoneC and Cube Mesh Object type called Cube. If you parent the Cube Child Object to the Armature Parent Object the Cube will get 3 new Vertex Groups created on it called BoneA, BoneB and BoneC. Notice that each Vertex Group is empty. See figure 21.

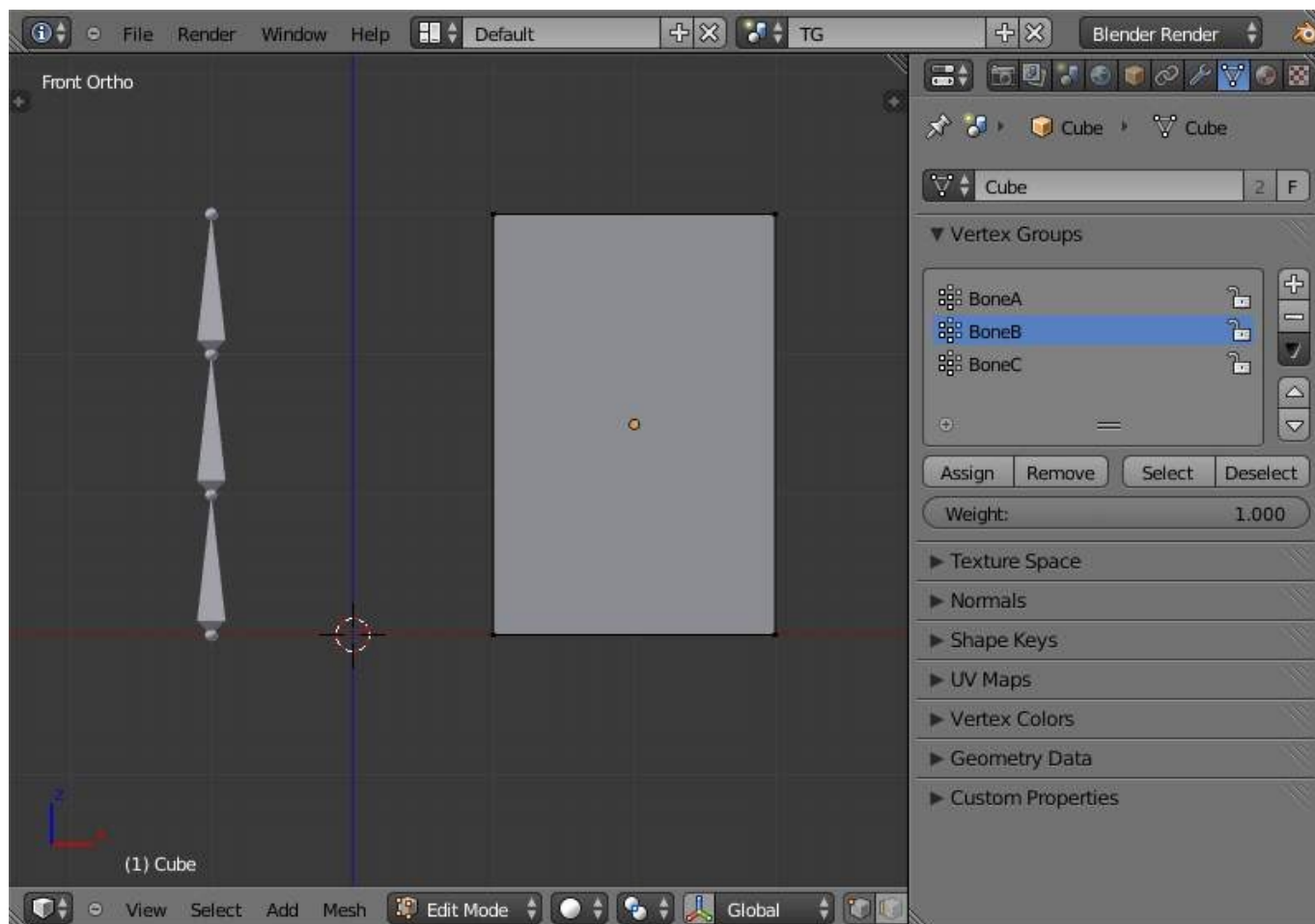


Figure 21 - Cube in Edit Mode showing the 3 created Vertex Groups after it was parented using Armature Deform With Empty Groups to an Armature with 3 Bones named BoneA, BoneB and BoneC with the Vertex Group Panel shown. All the Vertex Groups are empty.

Bones in an Armature can be generally classified into 2 different types:

- Deforming Bones
- Control Bones

Deforming Bones - Are bones which when transformed will result in vertices associated with them also transforming in a similar way. Deforming Bones are directly involved in altering the positions of vertices associated with their bones.

Control Bones - Are Bones which act in a similar way to switches, in that they control how other bones or objects react when they are transformed. A Control Bone could for example act as a sliding switch control, when the bone is in one position to the left it could indicate to other bones that they react in a particular way when transformed, when the Control Bone is positioned to the right, transforming other bones or objects could do something completely different. Control Bones are not directly used to alter the positions of vertices, in fact Control Bones often have no vertices directly associated with themselves.

When using the Armature Deform With Empty Groups parenting method Vertex Groups on the Child Object will only be created for Armature Bones which are setup as Deforming Bone types. If a Bone is a Control Bone no Vertex Group will be created on the Child Object for that bone.

To check weather a particular bone in an armature is a Deforming Bone simply switch to Pose Mode or Edit

Mode on the armature and select the bone you are interested in by RMB it. Once the bone of interest is selected navigate to [Properties Editor > Bone Context > Deform Panel] and check if the Deform tickable option is ticked or not. If it is the selected bone is a Deforming Bone, otherwise it is a Control Bone. See figure 22.

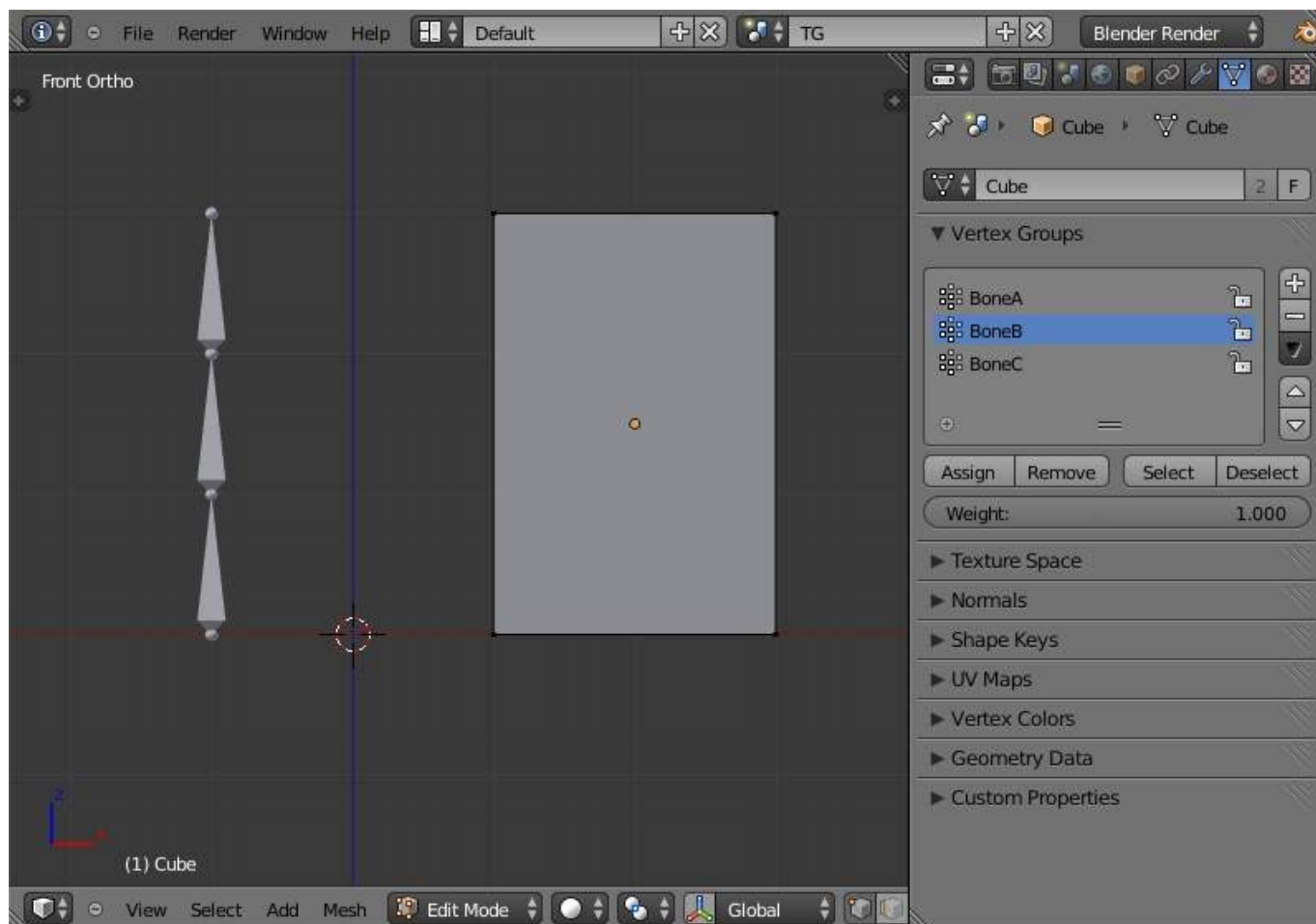


Figure 22 - 3 Bone Armature in Edit Mode with 2nd bone selected with [Properties Editor > Bone Context > Deform Panel] displayed an ticked, indicating the bone is a Deforming Bone.

Armature Deform With Automatic Weights

Armature Deform With Automatic Weights parenting feature does everything Armature Deform With Empty Groups does with one extra thing. That extra thing is that unlike Armature Deform With Empty Groups which leaves the automatically created Vertex Groups empty with no vertices assigned to them; Armature Deform With Automatic Weight will try to calculate how much Influence Weight a particular Armature Bone would have on a certain collection of vertices based on the distance from those vertices to a particular Armature Bone.

Once Blender has calculated the Influence Weight vertices should have it will assign that Influence Weight to the Vertex Groups that were previously created automatically by Blender on the Child Object when Armature Deform With Automatic Weights parenting command was carried out.

If all went well it should be possible to select the Armature Object switch it into Pose Mode and transform the bones of the Armature and the Child Object should deform in response. Unlike Armature Deform parenting you won't have to create Vertex Groups on the Child Object, neither will you have to assign Influences Weights to those Vertex Groups, Blender will try to do it for you.

To activate Armature Deform With Automatic Weights you must be in Object Mode or Pose Mode, then select

all the Child Objects (usually Mesh Object Types) and lastly select the Armature Object; Once done press **Ctrl-P** and select the Armature Deform With Automatic Weights from the Set Parent To pop-up dialog.

This method of parenting is certainly easier setup but it can often lead to Armatures which do not deform Child Objects in ways you would want, as Blender can get a little confused when it comes to determining which Bones should influence certain vertices when calculating Influence Weights for more complex armatures and Child Objects. Symptoms of this confusion are that when transforming the Armature Object in Pose Mode parts of the Child Objects don't deform as you expect; If Blender does not give you the results you require you will have to manually alter the Influence Weights of vertices in relation to the Vertex Groups they belong to and have influence in.

Armature Deform With Envelope Weights

Works in a similar way to Armature Deform With Automatic Weights in that it will create Vertex Groups on the Child Objects that have names matching those of the Parent Object Armature Bones. The created Vertex Groups will then be assigned Influence Weights. The major difference is in the way those Influence Weights are calculated.

Influence Weights that are calculated when using Armature Deform With Envelope Weights parenting are calculated entirely visually using Bone Envelopes. See figure 28.

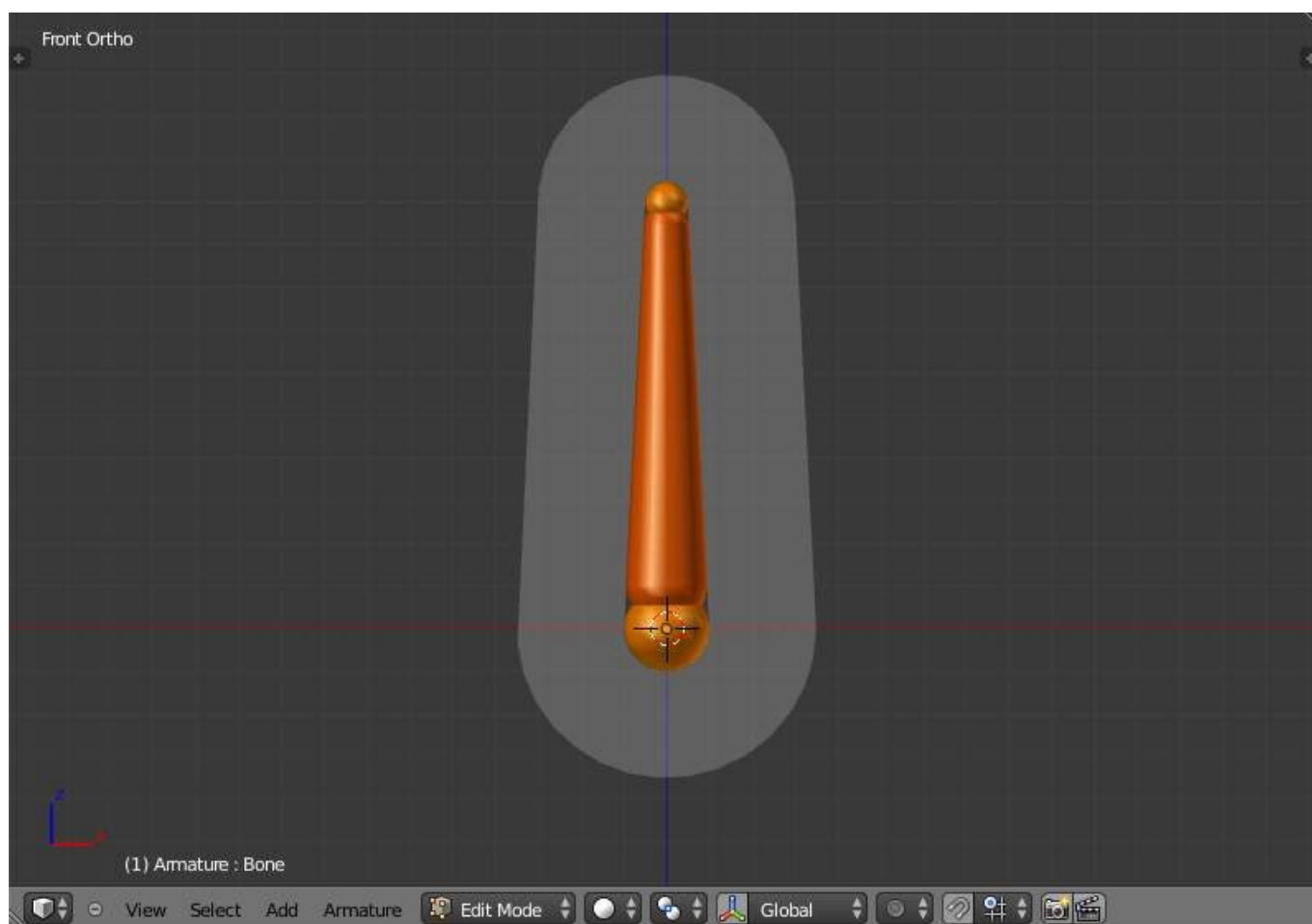


Figure 28 - Single Armature Bone in Edit Mode with Envelope Weight display enabled. The gray volume around the bone is the Bone Envelope.

Figure 28 shows a single Armature Bone in Edit Mode with Envelope Weight activated. The gray semi-

transparent volume around the bone is the Bone Envelope.

Any Child Object that has vertices inside the volume of the Bone Envelope will be influenced by the Parent Object Armature when the Armature Deform With Envelope Weights operator is used. Any vertices outside the Bone Envelope volume will not be influenced. See figure 29.

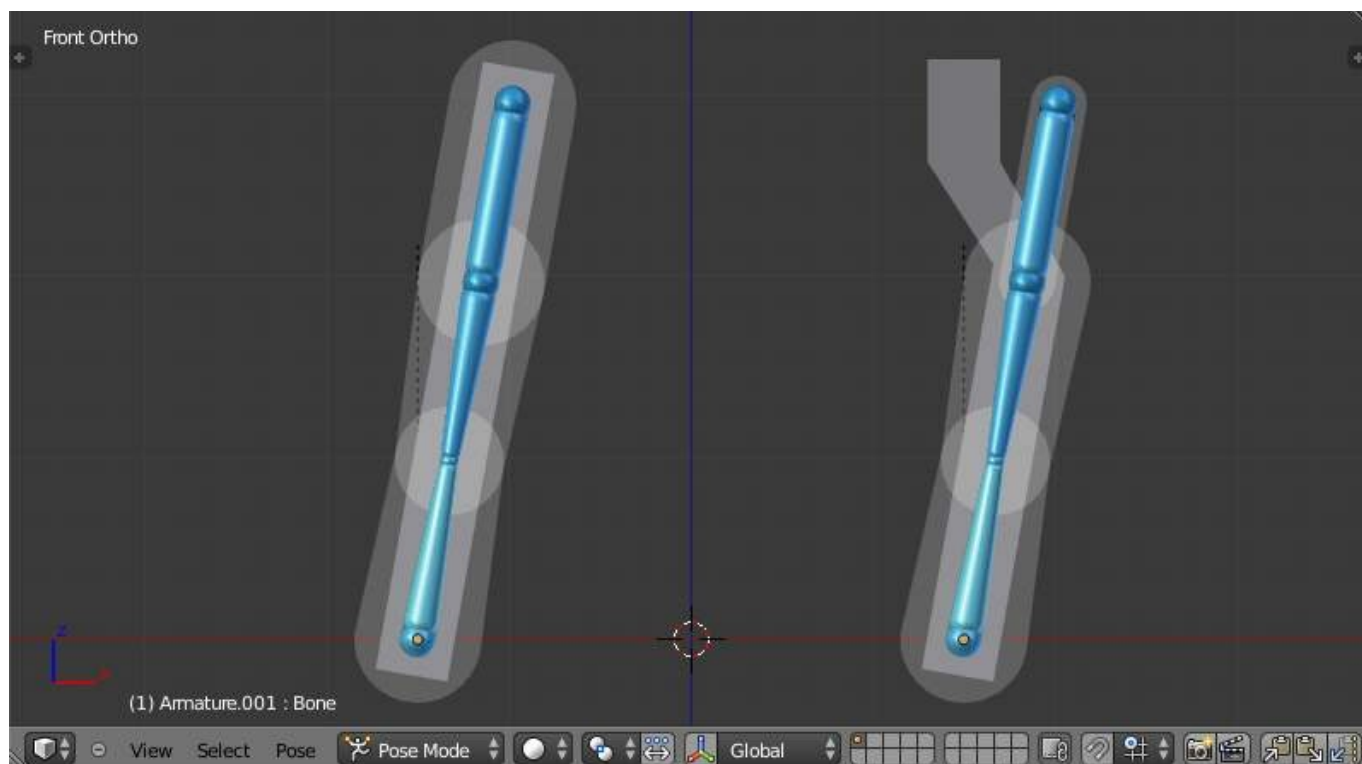


Figure 29 - 2 sets of Armatures each with 3 bones, the first set has all vertices inside the Bone Envelope, the second did not. When the bones are transformed in Pose Mode the results are very different.

The default size of the Bone Envelope volume does not extend very far from the surface of a bone; You can alter the size of the Bone Envelope volume by clicking on the body of the bone you want to alter, switch to Edit Mode or Pose Mode and then pressing **Ctrl-Alt-S** then drag your mouse left or right and the Bone Envelope volume will alter accordingly. See figure 30.

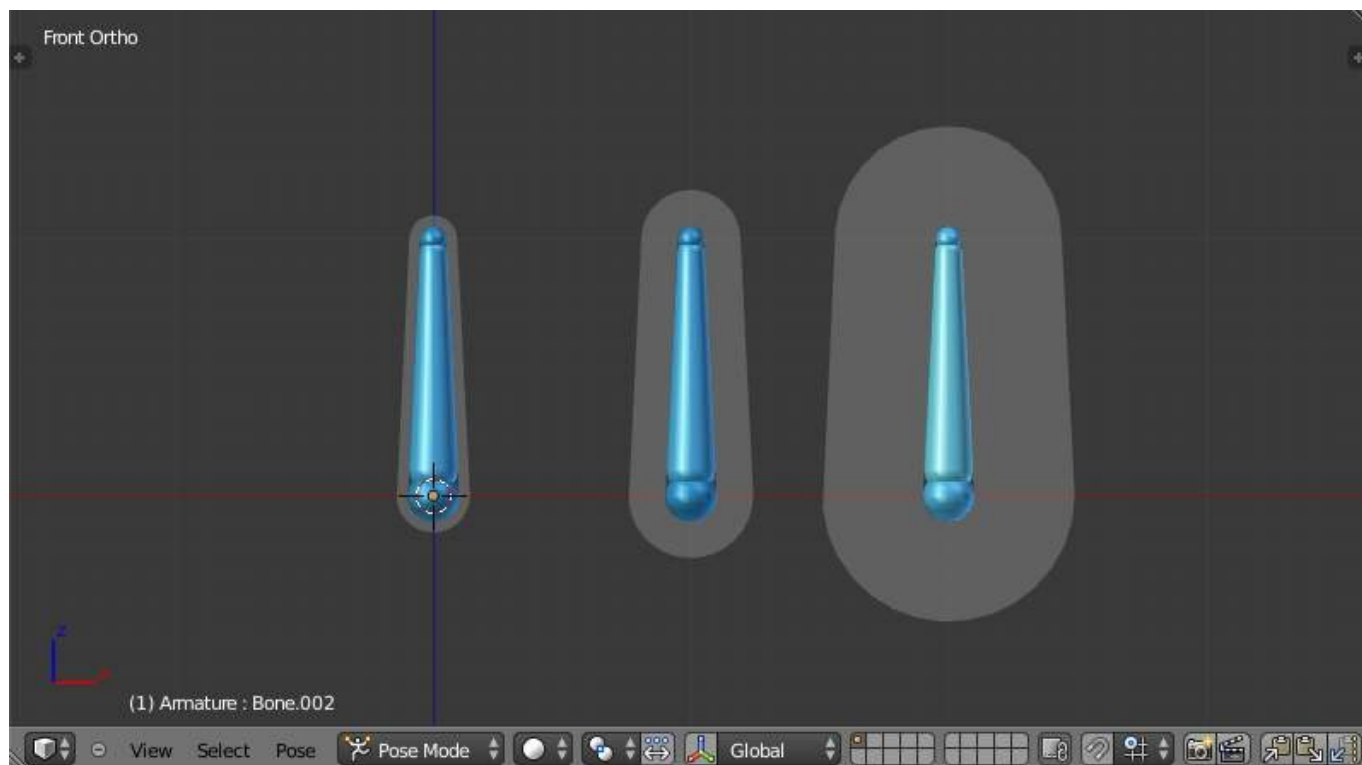


Figure 30 - Single Armature Bone with various different Bone Envelope sizes.

You can also alter the Bone Envelope volume by selecting the Bone you wish to alter and switching to Edit Mode or Pose Mode, then navigate to [Properties Editor > Bone Context > Deform Panel > Envelope Section > Distance field] then enter a new value into it. See figure 31.

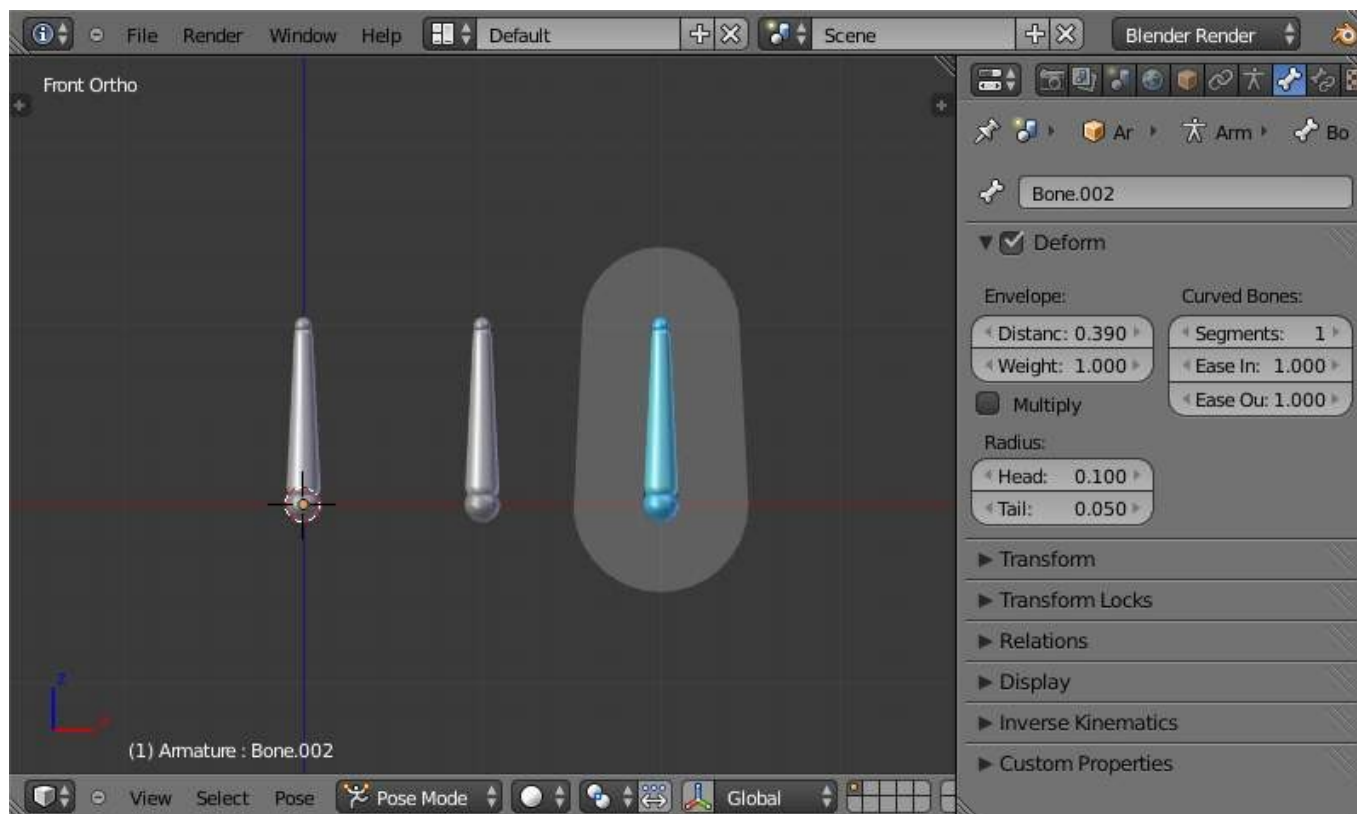


Figure 31 - [Properties Editor > Bone Context > Deform Panel > Envelope Section > Distance field] highlighted.

Altering the Bone Envelope volume does not alter the size of the Armature Bone just the range within which it

can influence vertices of Child Objects.

You can alter the radius that a bone has by selecting the head, body or tail parts of a bone while in Edit Mode, and then press **Alt - S** and move the mouse left or right. This will make the selected bone fatter or thinner without altering the thickness of the Bone Envelope volume. See figure 32.

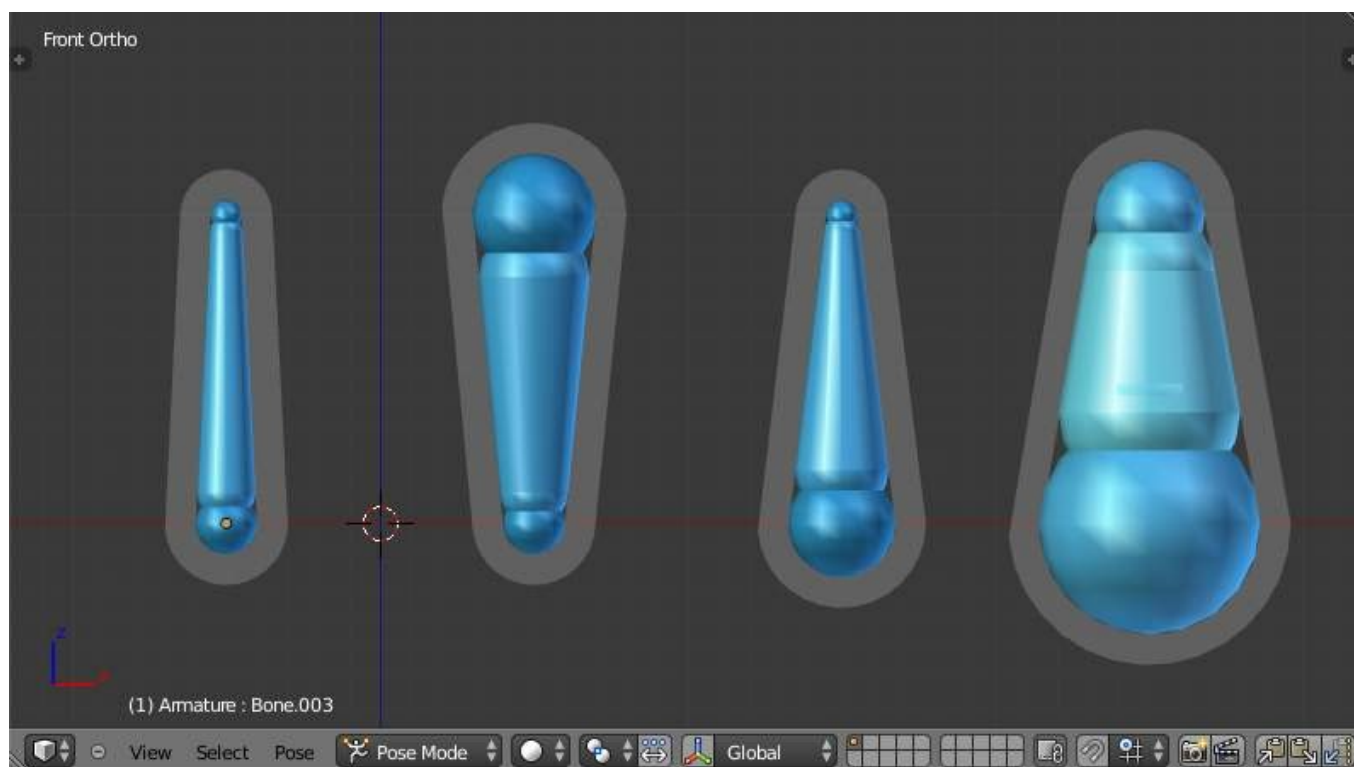


Figure 32 - 4 Armature Bones all using Envelope Weight. The 1st with a default radius value, the 3 others with differing Tail, Head and Body radius values.

You can also alter the bone radius by selecting the tail or head of the bone you wish to alter and switching to Edit Mode, then navigate to [Properties Editor > Bone Context > Deform Panel > Radius Section] and entering new values for the Tail and Head fields. See figure 33.

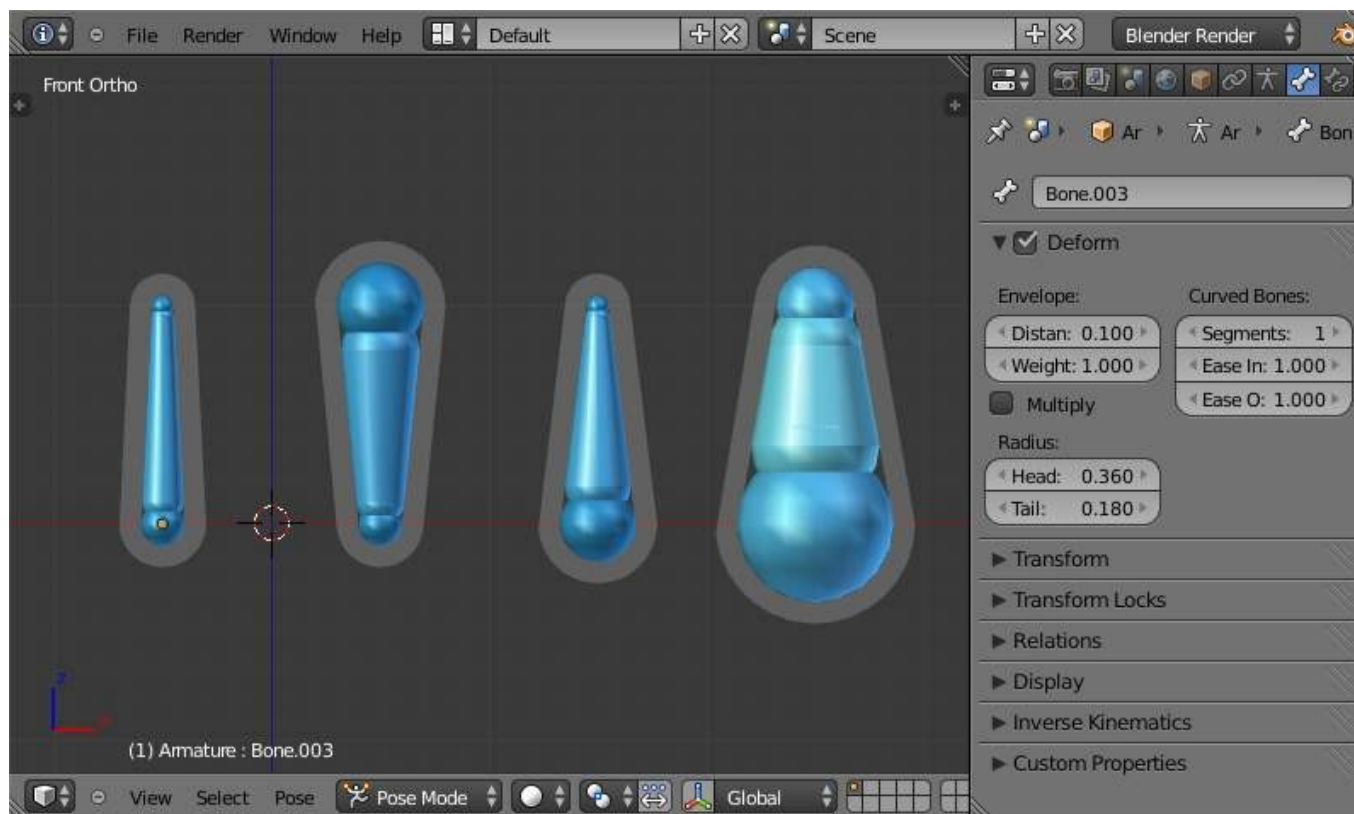


Figure 33 - [Properties Editor > Bone Context > Deform Panel > Radius Section] head and tail fields highlighted.

Note

If you alter the Bone Envelope volume of a bone so that you can have it include/exclude certain vertices after you have already used Armature Deform With Envelope Weights, by default the newly included/excluded vertices won't be effected by the change. When using Armature Deform With Envelope Weights it only calculates which vertices will be affected by the Bone Envelope volume at the time of parenting, at which point it creates the required named Vertex Groups and assigns vertices to them as required. If you want any vertices to take account of the new Bone Envelope volume size you will have carry out the Armature Deform With Envelope Weights parenting again; In fact all parenting used in the Set Parent To pop-up dialog which tries to automatically assign vertices to Vertex Groups works like this.

Bone Parent

Bone parenting allows you to make a certain bone in an armature the Parent Object of another object. This means that when transforming an armature the Child Object will only move if the specific bone it is the Child Object of moves. See figure 34.

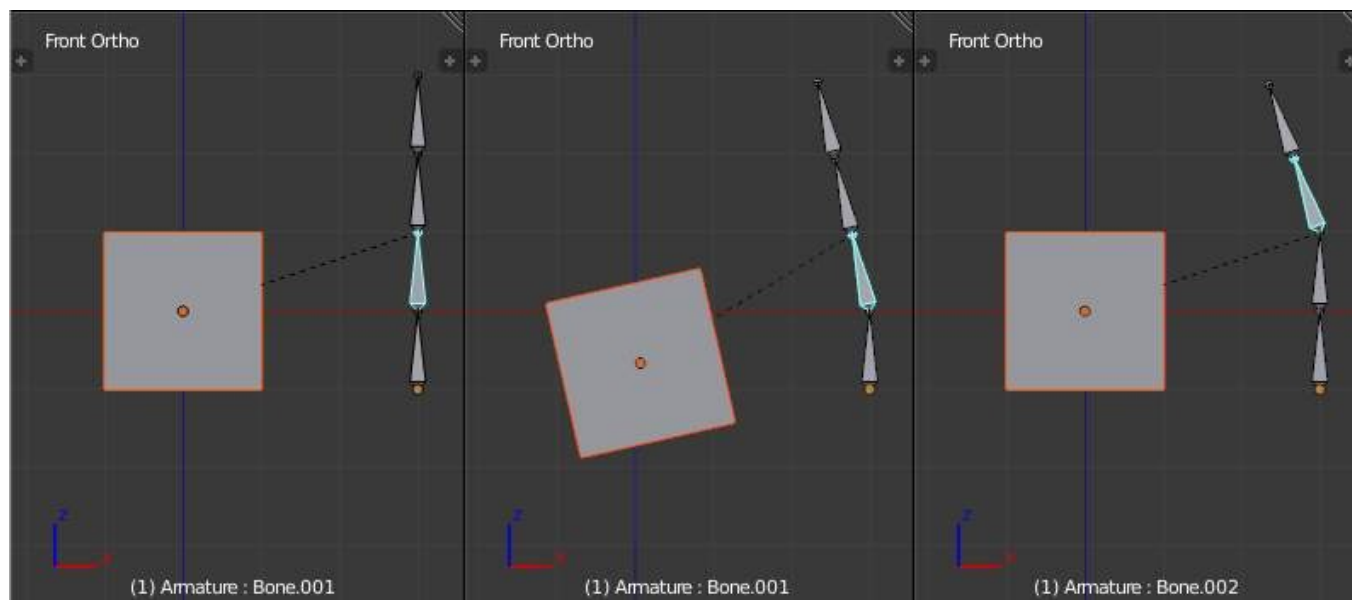


Figure 34 - 3 pictures of Armatures with 4 Bones, with the 2nd bone being the Bone Parent of the Child Object Cube. The Cube is only transformed if the 1st or 2nd bones are. Notice altering the 3rd and 4th bones has no effect on the Cone.

To use Bone Parenting, you must first select all the Child Objects you wish to parent to a specific Armature Bone, then **Shift-RMB** select the Armature Object and switch it into Pose Mode and then select the specific bone you wish to be the Parent Bone by **RMB** selecting it. Once done press **Ctrl-P** and select Bone from the Set Parent To pop-up dialog.

Now transforming that bone in Pose Mode will result in the Child Objects also transforming.

Relative Parenting

Bone Relative parenting is an option you can toggle for each bone. This works in the same way as Bone parenting with one difference.

With Bone parenting if you have parented a bone to some Child Objects and you select that bone and switch it into Edit Mode and then translate that bone; When you switch back into Pose Mode on that bone, the Child Object which is parented to that bone will snap back to the location of the bone in Pose Mode. See figure 37.

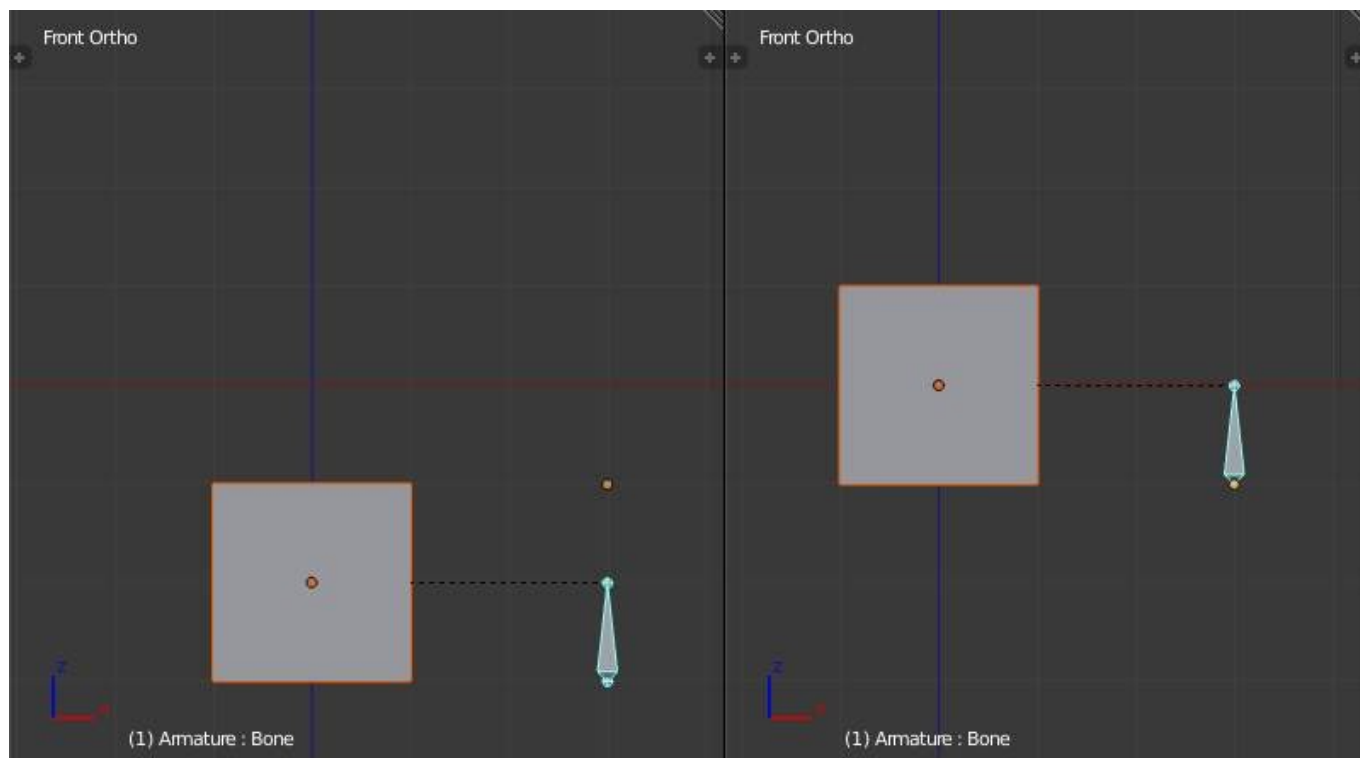


Figure 37 - [Single Armature Bone which has a Child Object cube parented to it using Bone parenting. 1st picture shows the position of the cube and armature before the bone is moved in Edit Mode. 2nd picture shows the position of the cube and armature after the bone was selected in Edit Mode, moved and switched back into Pose Mode. Notice that the Child Object moves to the new location of the Pose Bone.

Bone Relative parenting works differently; If you move a Parent Bone in Edit Mode, when you switch back to Pose Mode, the Child Objects will not move to the new location of the Pose Bone. See figure 38.

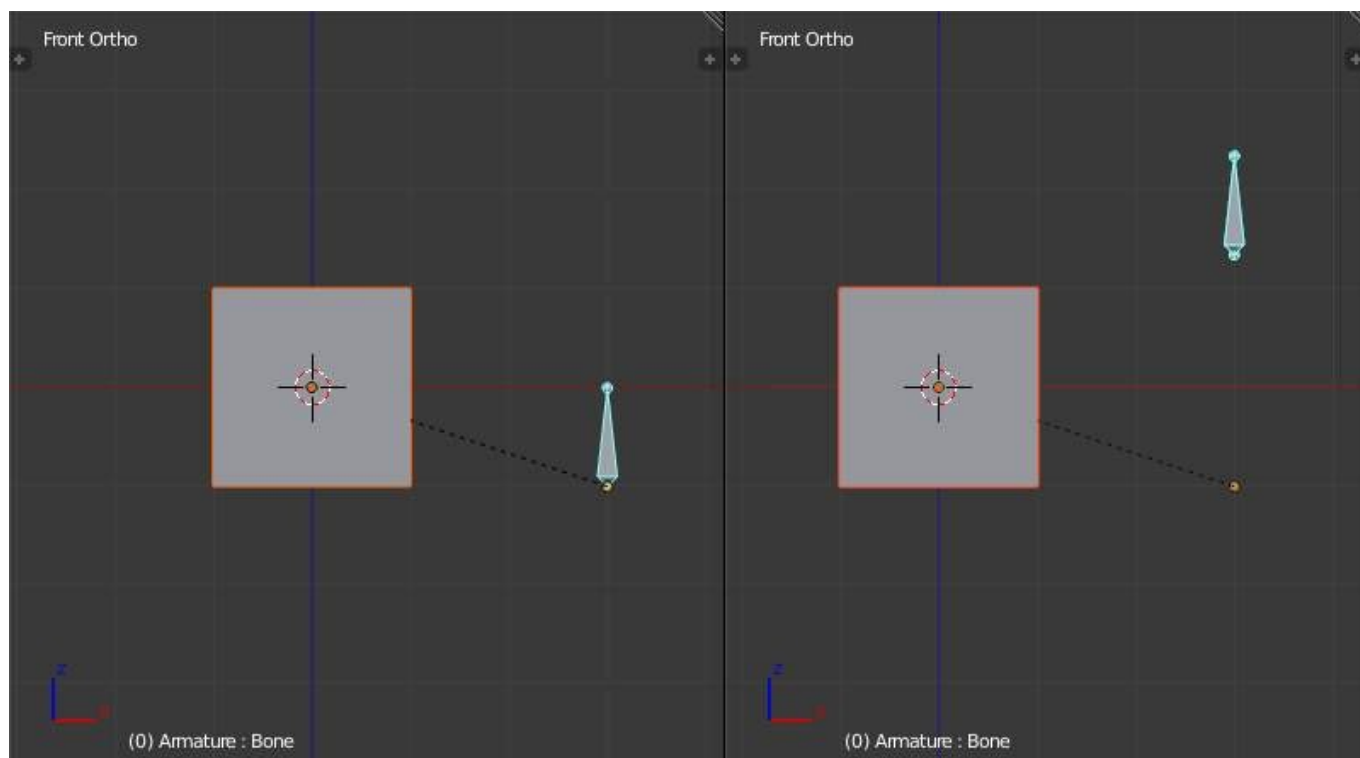


Figure 38 - [Single Armature Bone which has a Child Object cube parented to it using Bone Relative parenting. 1st picture shows the position of the cube and armature before the bone is moved in Edit Mode. 2nd picture shows the position of the

cube and armature after the bone was selected in Edit Mode, moved and switched back into Pose Mode. Notice that the Child Object does not move to the new location of the Pose Bone.

Vertex Parent

You can parent an object to a single vertex or a group of three vertices as well; that way the child/children will move when the parent mesh is deformed, like a mosquito on a pulsing artery.

Vertex Parent from Edit Mode

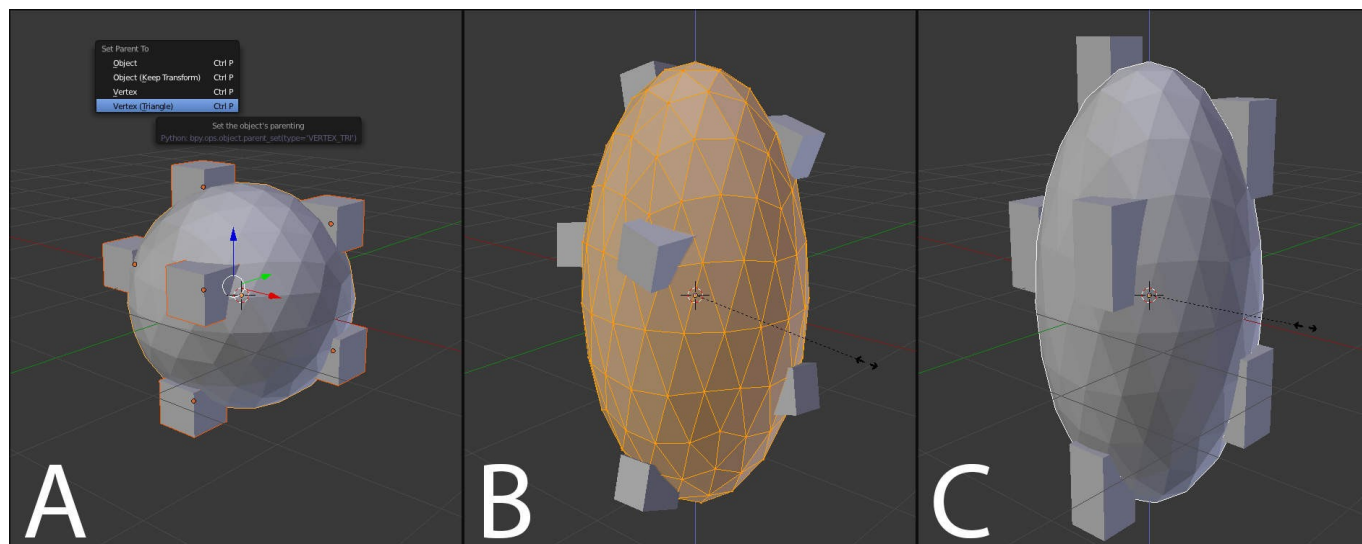
In *Object* mode, select the child/children and then the parent object. **Tab** into *Edit mode* and on the parent object select either one vertex that defines a single point, or select three vertices that define an area (the three vertices do not have to form a complete face; they can be any three vertices of the parent object), and then press **Ctrl-P** and confirm.

At this point, if a single vertex was selected, a relationship/parenting line will be drawn from the vertex to the child/children. If three vertices were selected then a relationship/parenting line is drawn from the averaged center of the three points (of the parent object) to the child/children. Now, as the parent mesh deforms and the chosen parent vertex/vertices move, the child/children will move as well.

Vertex Parent from Object Mode

Vertex parenting can be performed from object mode, This is done like regular object parenting, Press **Ctrl-P** in object mode and select *Vertex* or *Vertex (Triangle)*.

The nearest vertices will be used from each object which is typically what you would want.



See:

1. The small cubes can each be automatically parented to a triad of nearby vertices on the icosphere using the “Vertex (Triangle)” in the set parent context menu.
2. Reshaping the object in edit mode then means each of the cubes follows their vertex triad parent separately.
3. Re-scaling the parent icosphere in object mode means the child cubes are also rescaled as expected.

The parent context menu item means users can rapidly set up a large number of vertex parent relationships, and

avoid the tedious effort of establishing each parent-child vertex relationship separately.

Note

It is in fact a sort of “reversed” *hook*

Options

Move child

You can *move* a child object to its parent by clearing its origin. The relationship between the parent and child isn't affected. Select the child object and press **Alt - O**. By confirming the dialog the child object will snap to the parent's location. Use the *Outliner* view to verify that the child object is still parented.

Remove relationship/Clear Parent

You can *remove* a parent-child relationship via **Alt - P**

The menu contains:

Clear Parent

If the parent in the group is selected nothing is done. If a child or children are selected they are disassociated from the parent, or freed, and they return to their *original* location, rotation, and size.

Clear and Keep Transformation

Frees the children from the parent, and *keeps* the location, rotation, and size given to them by the parent.

Clear Parent Inverse

Places the children with respect to the parent as if they were placed in the Global reference. This effectively clears the parent's transformation from the children. For example, if the parent is moved 10 units along the X axis and *Clear Parent Inverse* is invoked, any selected children are freed and moved -10 units back along the X axis. The “Inverse” only uses the last transformation; if the parent moved twice, 10 units each time for a total of 20 units, then the “Inverse” will only move the child back 10 units, not 20.

Hints



Outliner view

There is another way to see the parent-child relationship in groups and that is to use the *Outliner* view of the *Outliner* window. Image (*Outliner* view) is an example of what the *Outliner* view looks like for the (*Parenting Example*). Cube A's object name is `Cube_Parent` and cube B is `Cube_Child`.