



*Note from the author:*

In opening, I would like to first introduce myself. I am a modder who is very active on the [www.crymod.com](http://www.crymod.com) forums. I have spent countless hours learning to mod Farcry. I have developed my own style and method of successfully importing weapons into Farcry. If you're new to modding or are a veteran modder who is looking for some useful information, then this tutorial is right for you. In this tutorial I will cover everything concerning proper weapons modding for Farcry. I wrote this tutorial for many reasons, the most important of which is so serious modders get the information they need to create great weapons mods for this fantastic game. When I say serious modders, I don't mean experienced modders, I mean anyone, including so called "noobs", who understand that modding takes patience, time, and effort also willing to accept that fact and go ahead and mod anyway. So sit back, relax, and get ready to learn how to get your favorite weapons into Farcry.

Before we begin, here is a list of what you need to successfully follow the tutorial:

3ds Max 7  
Mod SDK for Farcry  
Farcry (the game)  
Sandbox (included w/Farcry)

Optional programs that will help you in modding Farcry:

UltraEdit	-Used to edit scripts
Adobe Photoshop	-Used for creating textures (skinning)

Note: You don't necessarily need 3ds Max version 7, however you will need it to be able to open the weapon file included with this tutorial (Max 6 should work fine as well)

Also, I have included Max Source files with this tutorial that will aid you during the walkthrough; it is important to note that this tutorial is also a walkthrough.

To get a weapon working for Farcry you need to complete a few basic steps, these include:

Modeling

Skinning

Rigging

Animating

Scripting

Most people like to focus on one of those fields, however it isn't all that difficult to learn to do it all. In this tutorial you will get a chance to learn all of these except modeling and skinning. The reason I chose it this way, is because there are literally tons of tutorials on texturing and modeling, in particular weapons. Do a google search for weapons modeling and skinning, I'm sure you will find many useful sources as I did.

Here's a few links to get you started on your search:

<http://www.3dcafe.com/asp/default.asp>

<http://www.planetrainbowsix.com/atwar/Pain/html/texturing.html>

<http://www.3dbuzz.com/>

<http://www.3dtotal.com/>

<http://models.firearmsmod.com/?p=tut&tut=list&cat=3>

<http://www.planetunreal.com/fordy/gradientssmeshtutorials/14.html>

Let's begin by using an AK-47 I have included with this tutorial. Thus, you will not have to worry about skinning or modeling it. However, you can use another model during this tutorial if you wish since the techniques we will be using are pretty much universal for all weapons, including melee weapons like bats, swords, hockey sticks, etc.



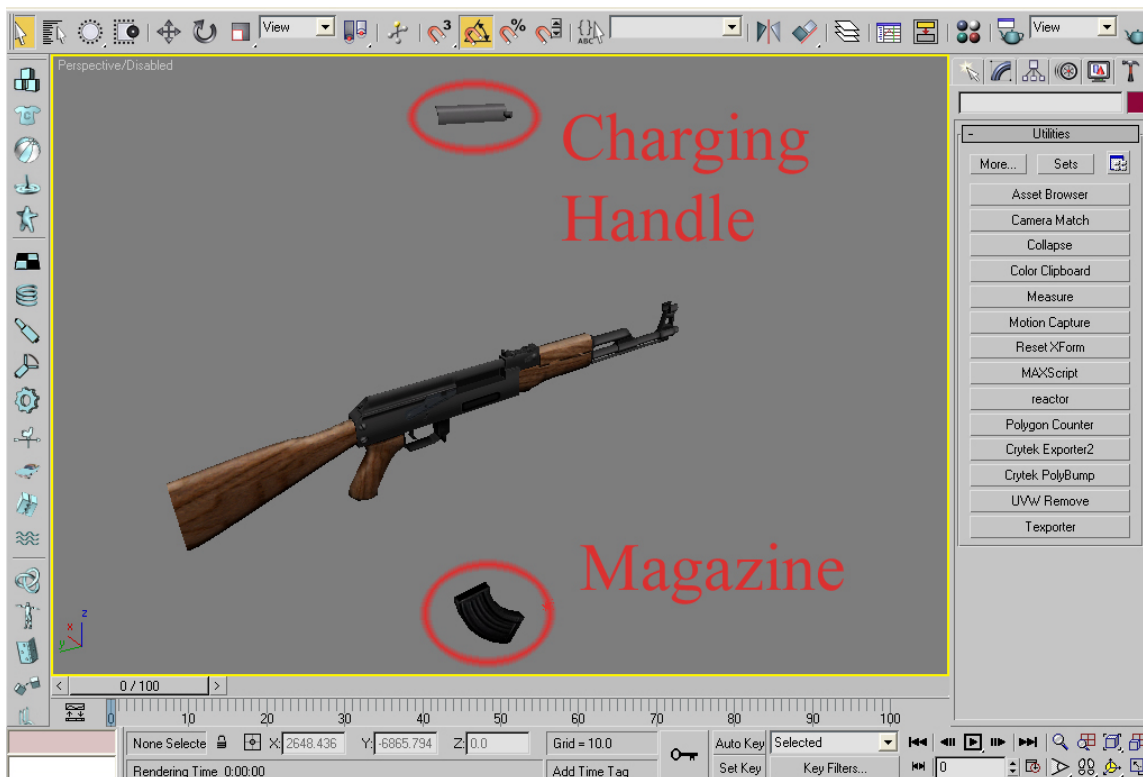
## Getting Started

Go ahead and open 3ds Max. Once open, go to File and open the AK47.max file. Go ahead and rotate the weapon to get a good look at it. If you notice, some parts have more detail than others. This is a design decision. I have designed the weapon this way because I felt it gave the model a good balance of looks and efficiency. What I mean by that is, you can make a highly detailed model. It will look pretty, but too detailed means too many excess polys you don't need to waste on little things like knobs, screws, holes, etc. Most of these details can be textured when you skin the weapon, this way the weapon still retains a nice detailed look and is more performance friendly. Go to your utilities tab and open your Polygon Counter. The new window will show you how many polygons the weapon is made up of. The AK47 is made up of 2,871 polys. This is a relatively mediocre number. It's not too low, and it isn't too high. Try to keep your first person model (the more detailed model you will see when you play from first person view) to no more than 4,000 polys. There is a lot of debate on how many polys a weapon should or "has" to be, however from experience I have experimented with very high poly counts. In one instance,

I successfully imported a weapon of over 18,000+ polys. I don't recommend that you make weapons that high though, but it is an interesting thing to push the limits of CryEngine.

## Exporting

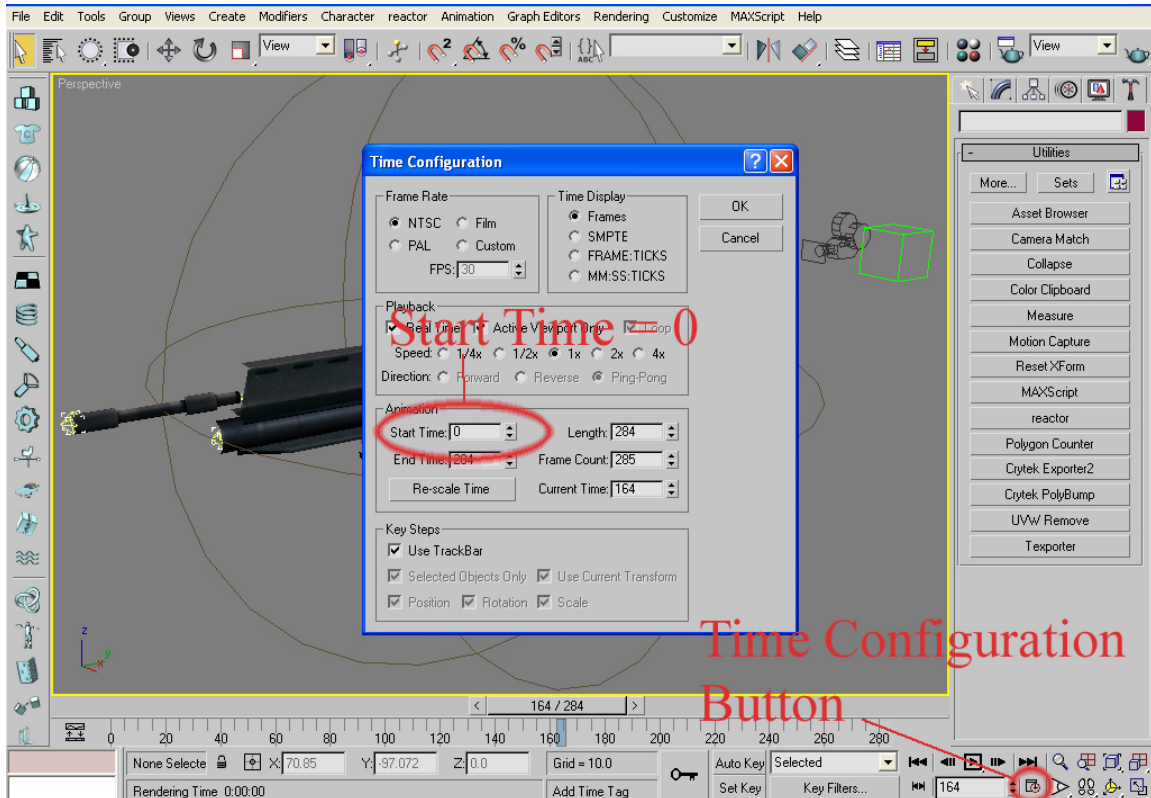
The next thing we need to do is prepare our weapon for exportation. Don't get too ahead of yourself, we're not going to export to CryEngine yet. We're going to export it to a new file so we can animate it. Notice that the animated parts are separated from the main weapon frame.



The charging handle and magazine are separated because this will make it possible for us to rig it later. If you are using your own model, go ahead and separate your animated parts like the AK47. Once the parts are separated, select your model (Make sure the entire mesh is one model and not 3 different models). Press Q to switch to the select tool, select the weapon. Go to File then select Export Selected. In the save window, type in AK47 and save it as .3ds.

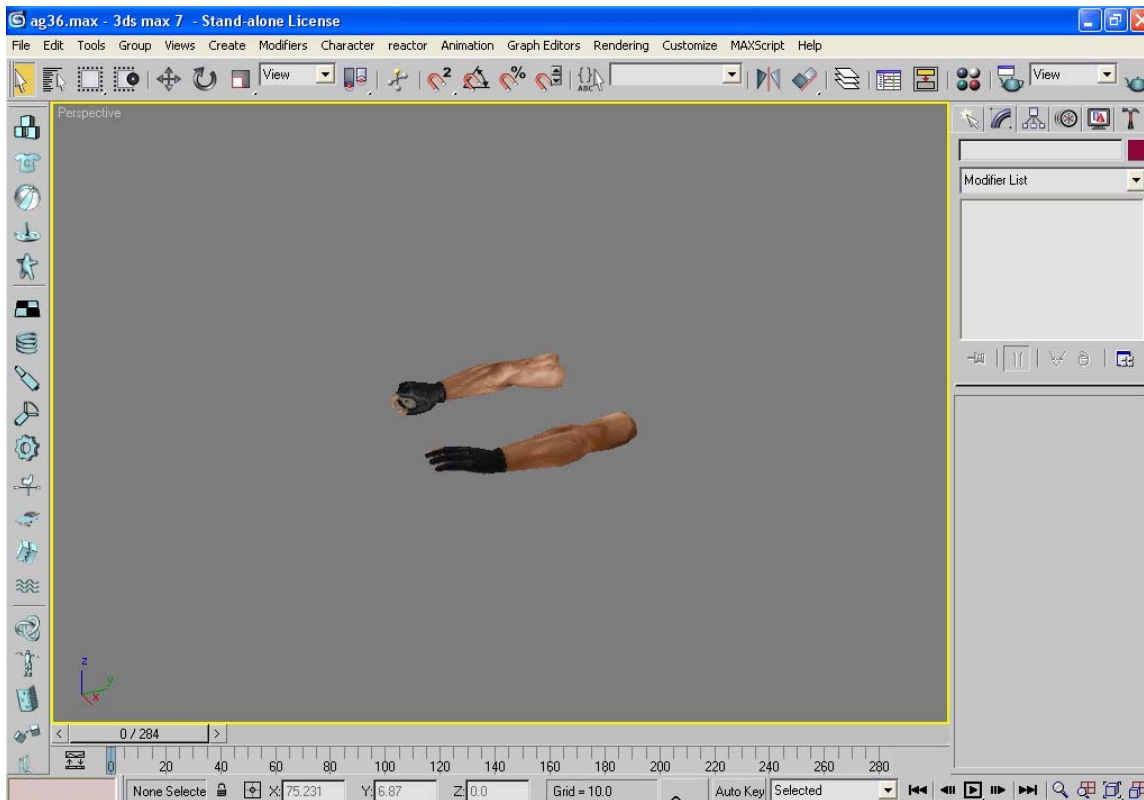
# Rigging

Here I will give you a couple of options. If you want to learn rigging in a more complete way, open the AG36.max file that comes with the Mod SDK. If you already know some rigging or you want to skip ahead a bit, go ahead and open a file I included with this tutorial called SOURCE.max. The difference is that the AG36.max file will require you to rig everything from the arms to the weapon, while the SOURCE.max file will have the arms set up with bones already. This is a file I made that allows me to rig multiple weapons much faster. I will continue as if you chose to open AG36.max. When you open the file, you may get a message basically saying that Max can't find the texture files for the scene. To fix this, locate the ag36 and hero-arms textures. These can both be found within the textures.pak located in the FCdata folder inside your Farcry directory. You may need to convert them to JPEG, so simply open them in Photoshop and save them as JPEG's and direct Max to find them in the path you saved the JPEG's at. Everything in the scene will now be textured properly. Go to the time configuration button and change the Start Time from 164 to 0, then click OK.

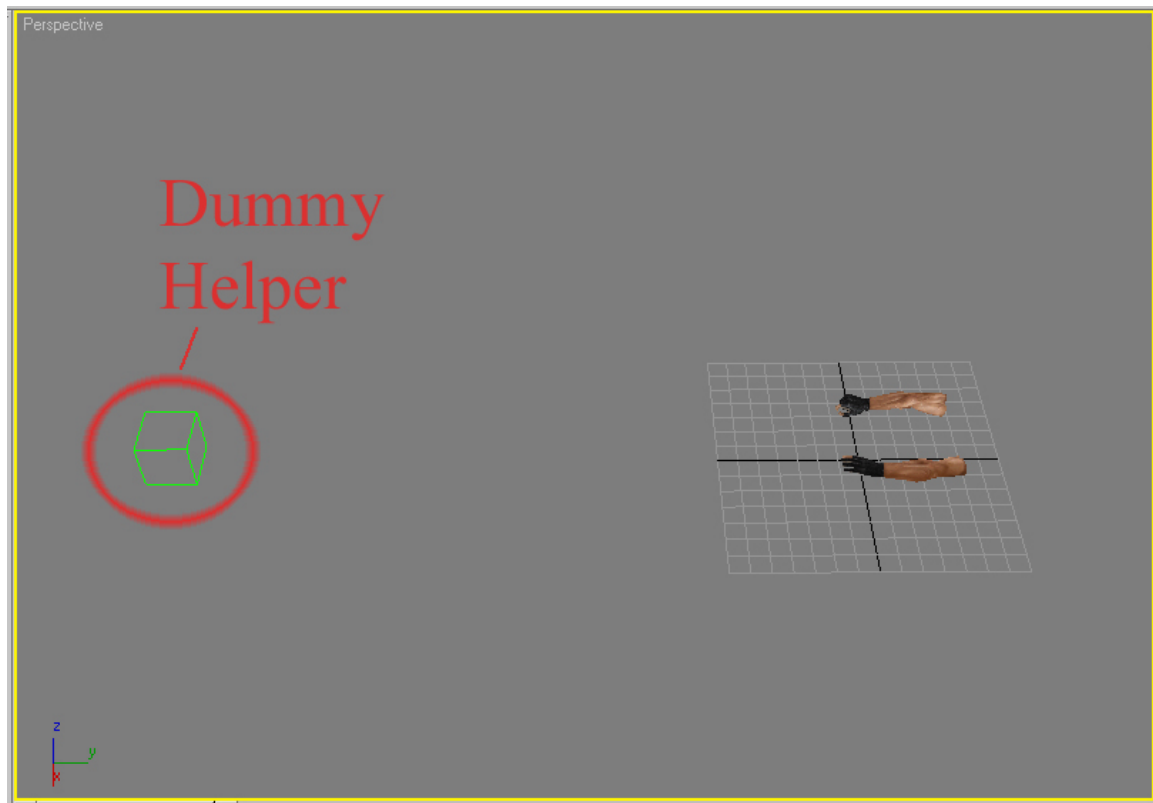


Now zoom out and select everything in the scene. Select every single key frame in the animation timeline and delete them all. Select the arms and go to the Modify tab. Select the Physique modifier and delete it. Select everything in the scene and delete it all except

the arms. Select the arms and go back to the modify tab. Expand Editable Mesh by clicking on the small “+” icon. Select Polygon and select the ag36 gun by drawing a box around it with the select tool (Q is the shortcut). Now hit Delete and you should be left with just the arms and nothing else. Make sure you didn’t accidentally delete any polys on the arms.



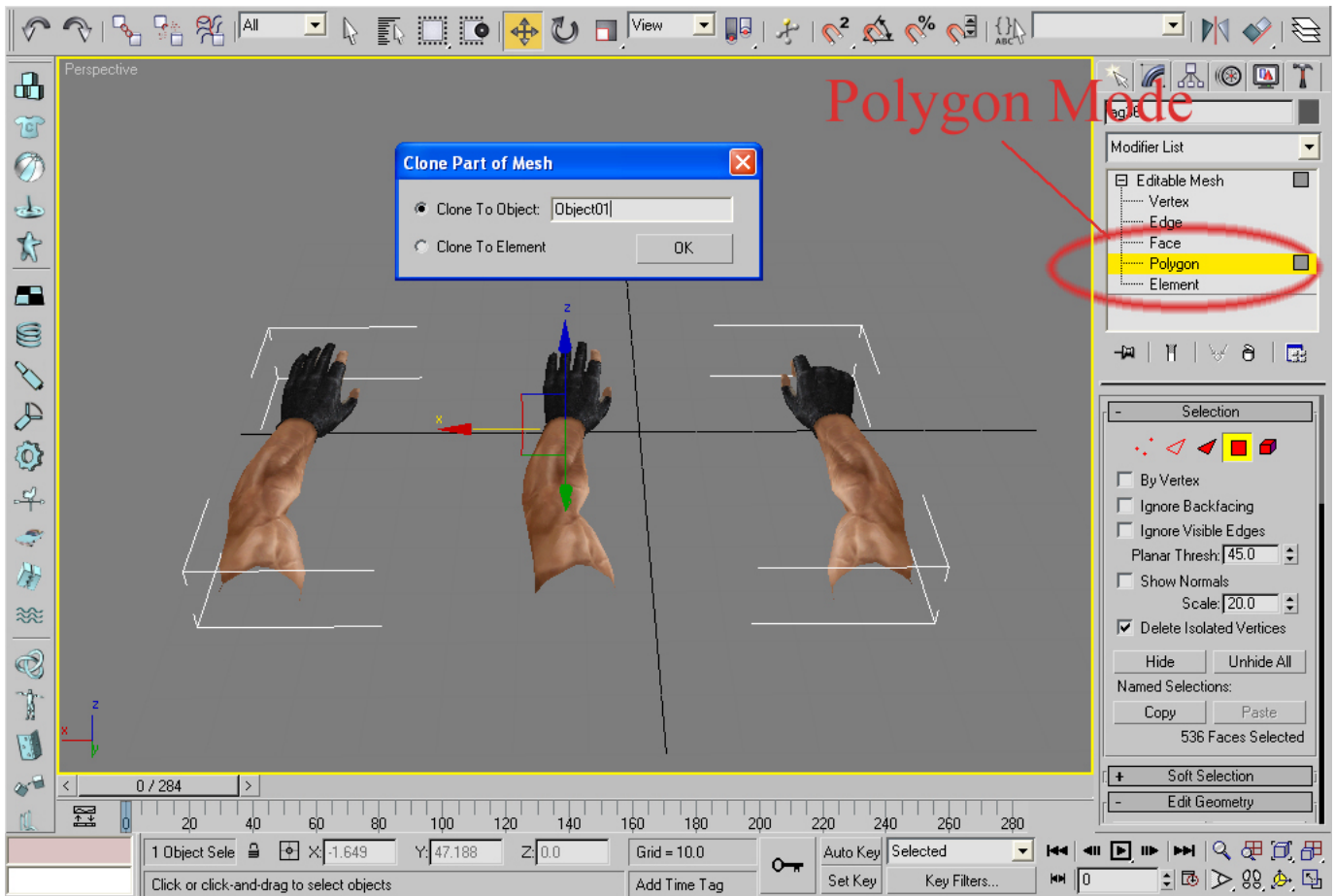
Now we can begin setting up bones for animation. Go to the Create tab, click on Helpers. Click on Dummy; it turns yellow. Now create a Dummy helper in front of the arms at a somewhat far distance, use the following image as a reference:



This placement of the Dummy Helper will make life easier later on when we begin animating. Rename the Dummy Helper to Root.

## Preparing the arms

Let's focus on the arms for now. What we're going to do next is prepare the arms in a way that it will make it much easier for us later during the rigging phase. Select the arms and in the modify tab, select Polygon mode. Draw a selection box around the left arm; the one with the fingers extended straight forward. (Note: Make sure Ignore Backfacing is checked **off**; you want to be able to select all polygons of the left arm). After you have selected the entire left arm in polygon mode, press "**W**" to quickly switch to the Move Tool. Hold "**SHIFT**" and move the left arm in the X Axis a little to the right; a copy will be made automatically. In the window that popped up, check on **Clone to Object** and click OK; by default the new arm will be called "Object01".

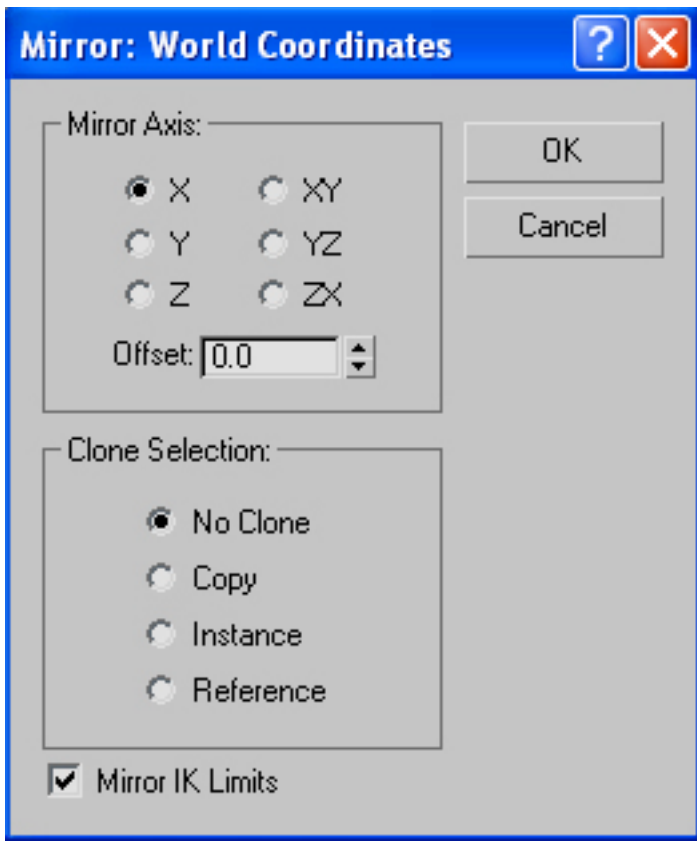


Get out of Polygon mode by clicking on “Polygon” again in the “Modify Tab” so it is no longer yellow. Select the new arm and quickly rotate around it to make sure the arm has all

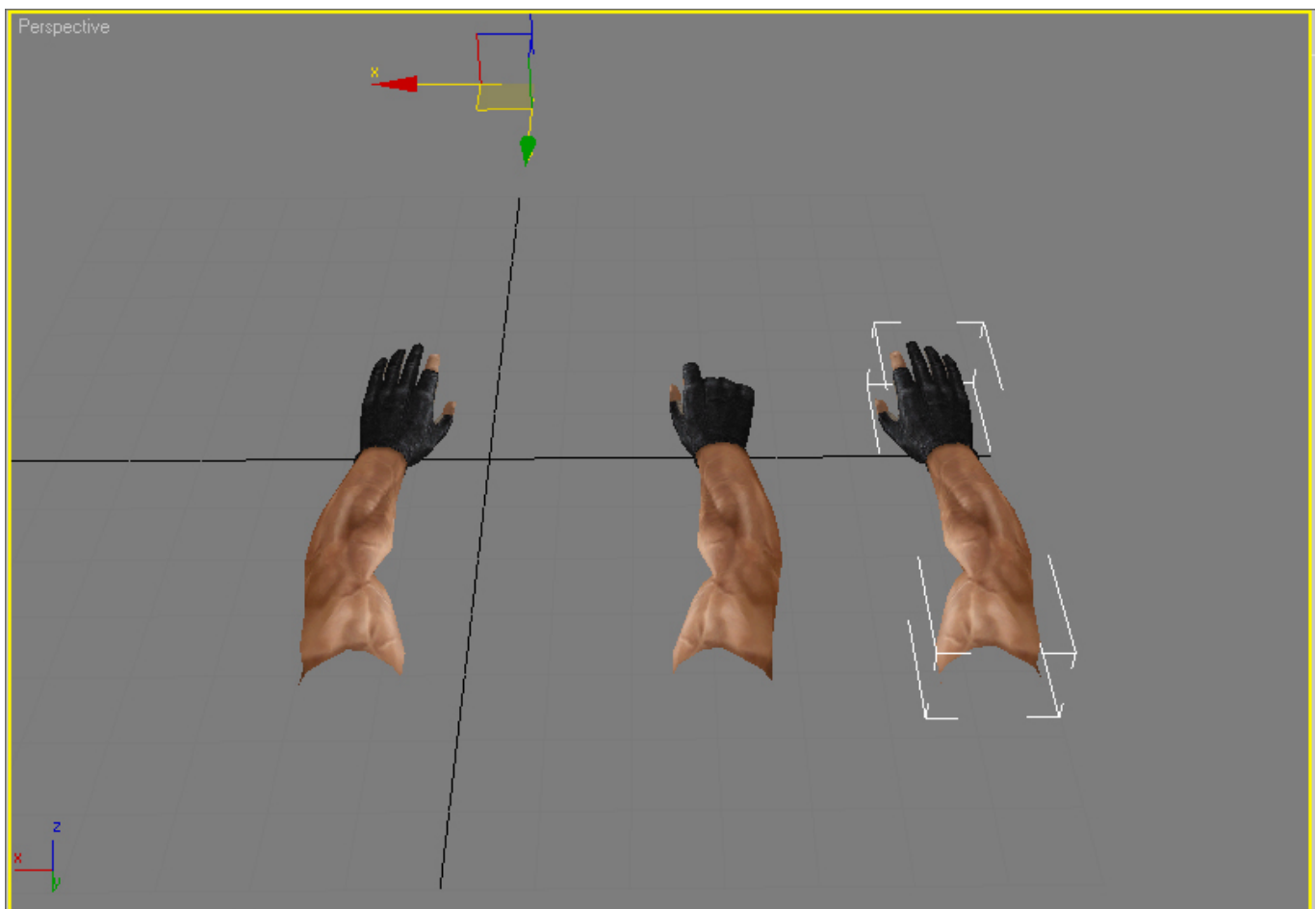


it’s polygons. Now click on the “Mirror Button” and a new window will appear. Make sure the options for “Mirror: World Coordinates” looks like this:

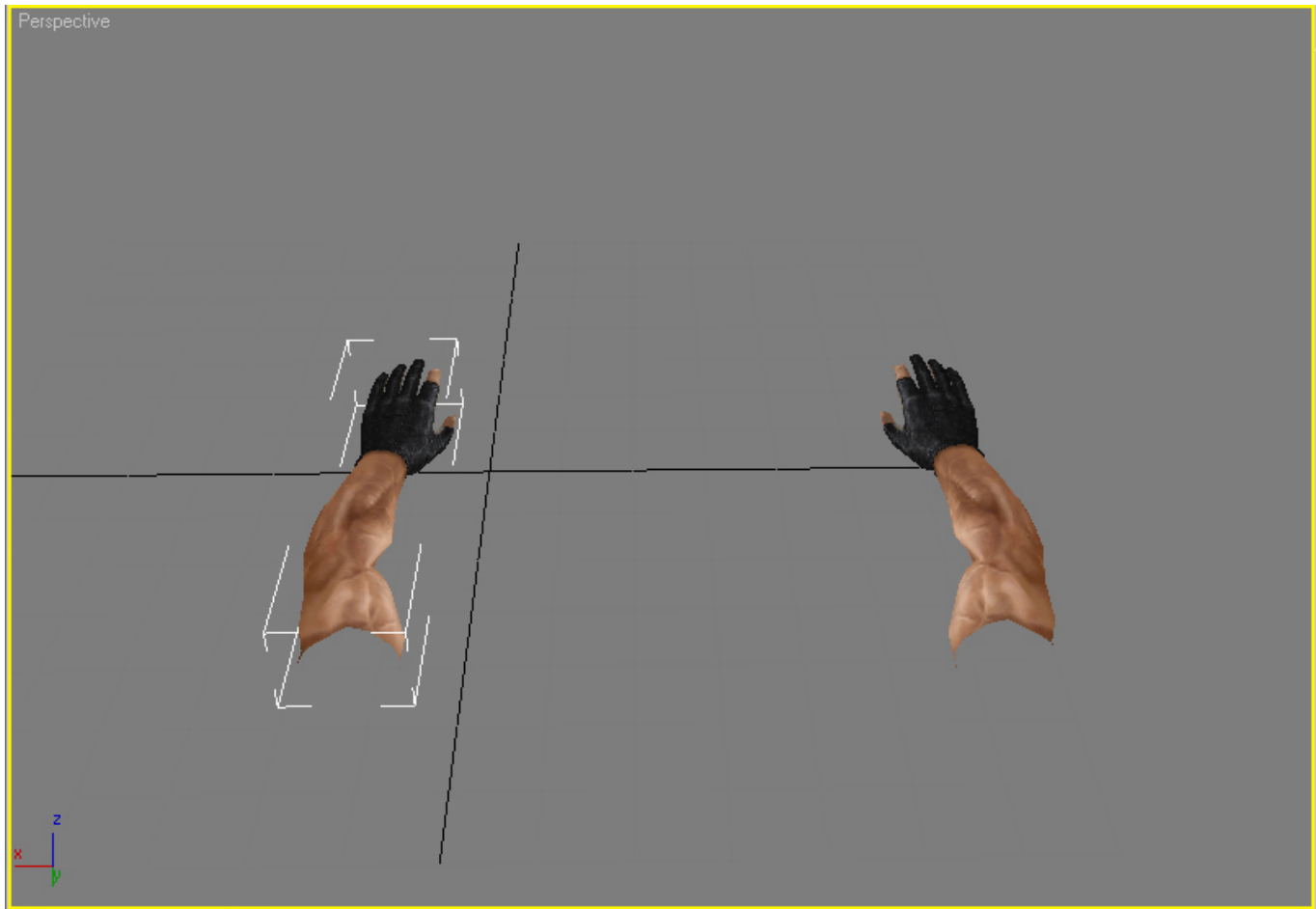




Your new arm should appear on the other side like so:



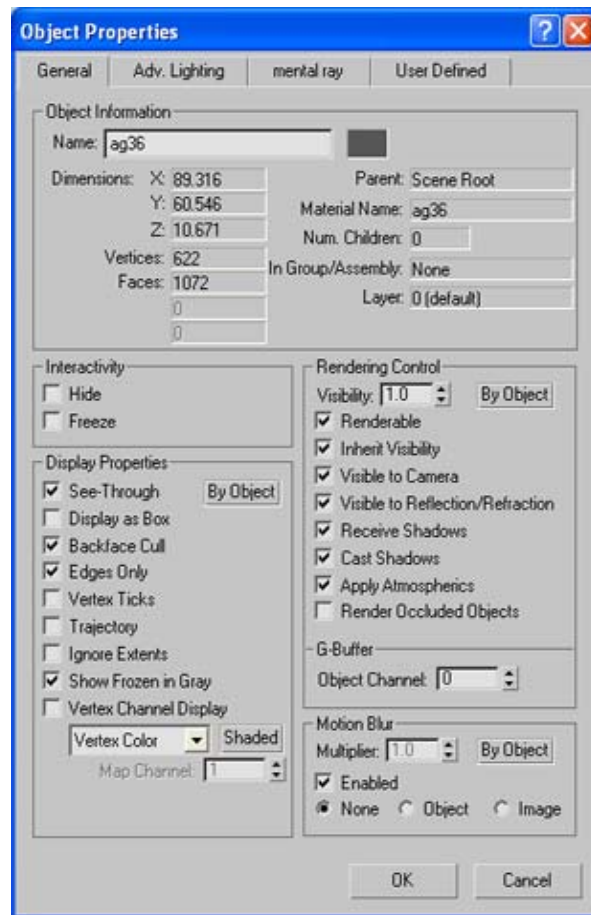
Now go back and select the original arms. Go into “Polygon” mode again and draw a selection box around the right arm with the curled up fingers. Delete the right arm and you are left with our arms ready for placing bones:



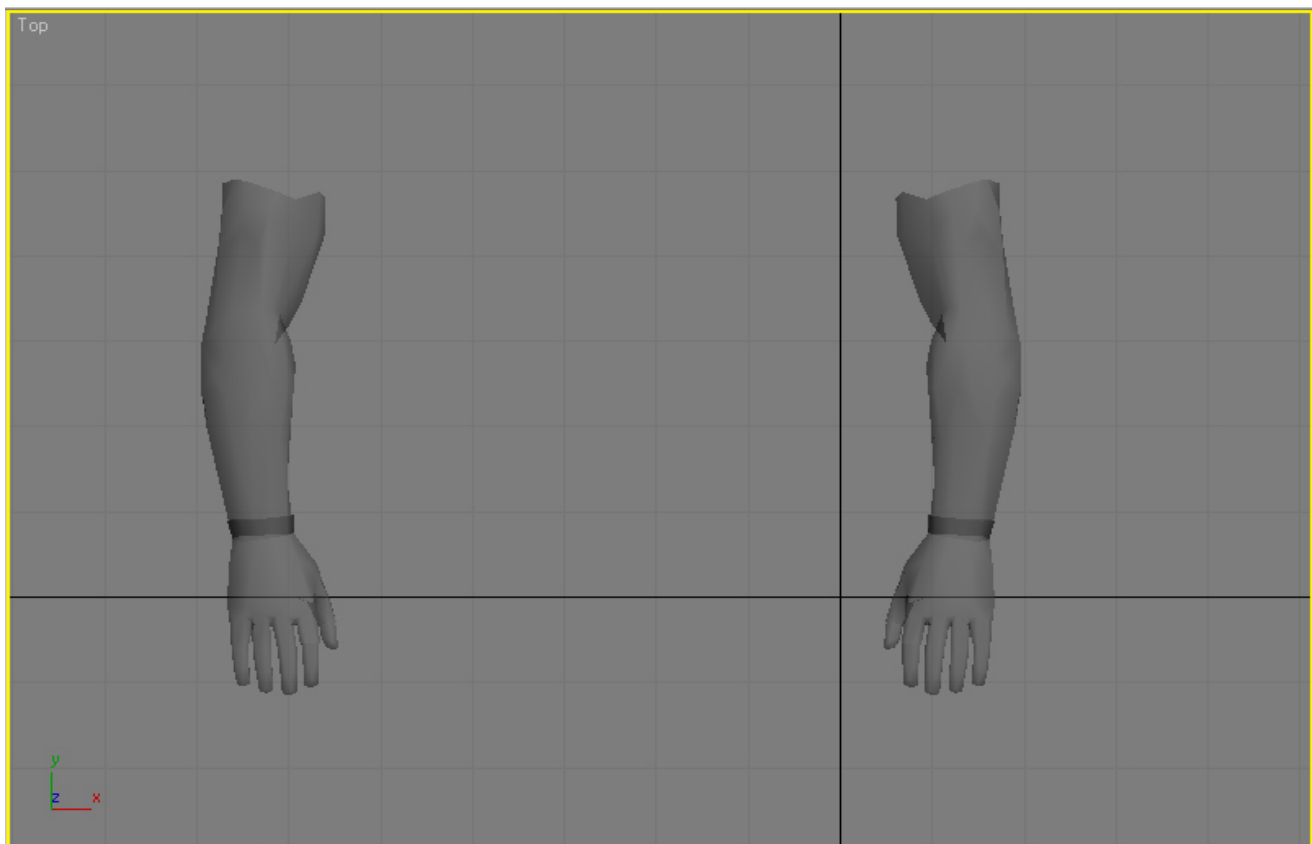
First, with the left arm still selected, get out of “Polygon” mode. Now in the “Edit Geometry” rollout, click the “Attach” button and attach the new right arm to the left arm. Both arms are now one entire mesh.



## Placing Bones

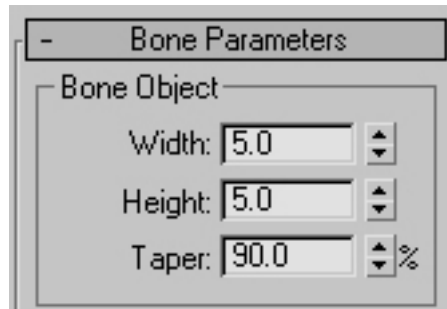
Let’s begin placing the bones for the arms. Press “**T**” to switch to the Top Viewport. Select the arms and right click the mouse, in the quad menu click on “Properties”. In the Object Properties window, check “See-Through” on.



Then click OK. You will now have arms that look like this:

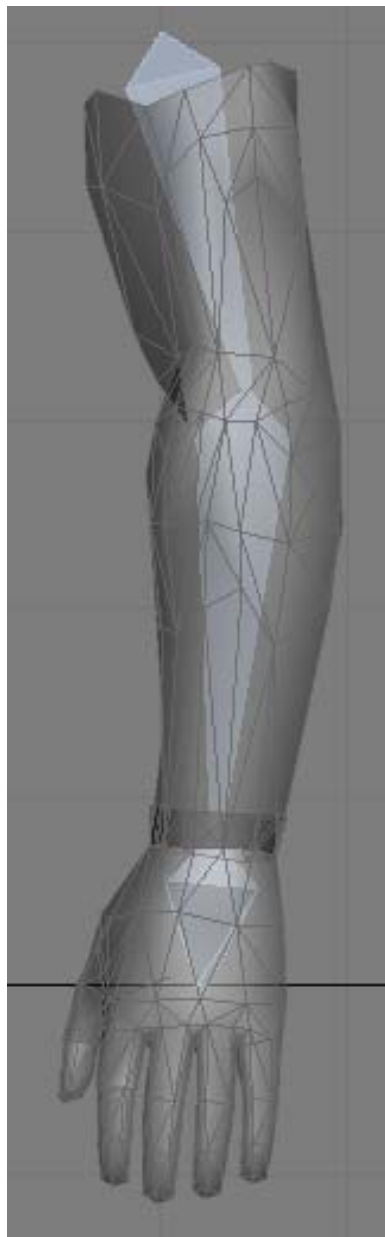


Let's place our bones now by clicking on the "Create Tab"  and then click on the "Systems" button . Click on the "Bones" button to create our bones. Change the bone width and height to 5.



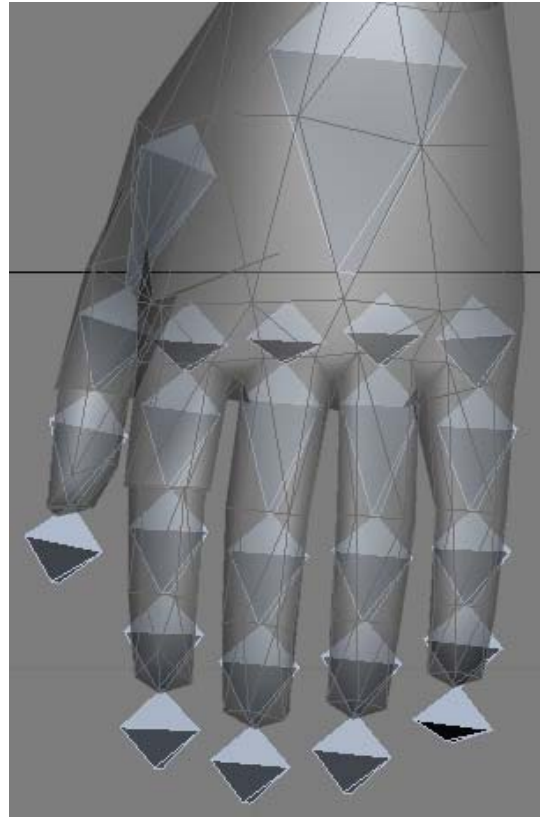
Now begin creating bones in the left arm by clicking just in front of where the arm begins, then click again at the beginning of the forearm and then at the hand again to create one chain composed of 3 bones for the left arm.

**Note:** Turn "Edged Faces" and "Smooth+Highlights" on to work easier in the Top View. Right click on the word "Top" in the upper left corner of the viewport and in the small menu turn on these options by left clicking on them (a check mark signifies the desired option is on).



Create bones the same way as we did for the arm only this time make them for the fingers. Since the fingers are smaller than the arms, change the bone width and height to 2 instead of 5. Using segments of five bones per finger is fine, but you don't need much more than about 4 for the thumb, unless if you want to for whatever reason.

**Note:** Try placing your bones as accurately as possible to make the rigging process as easy and quickly as possible, alternatively, poor bone placement can hurt you later and cause you to have to redo unnecessary hours of work-be meticulous!



After you place the bones as described above for both arms and hands, we need to work on the actual weapon. Press “Q” to get out of “Bone create mode” if you haven't already.

## Importing the .3ds File

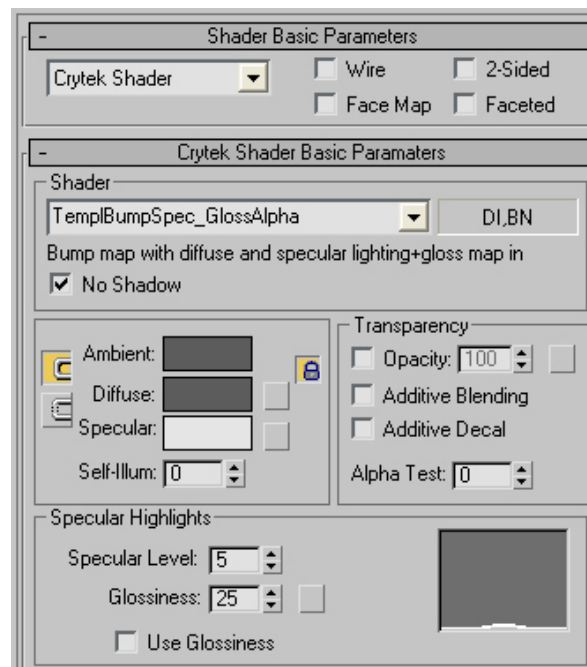
Go to “File” then “Import” and load up the AK47.3ds file you exported earlier. When you receive the warning windows, simply click OK to them. When you do, the AK47 model will be gray and appear in the scene. However, there's some problems. The first thing you notice is that it is the wrong size and it has no texture. Don't panic, we're going to fix that easily.

# Applying the Texture

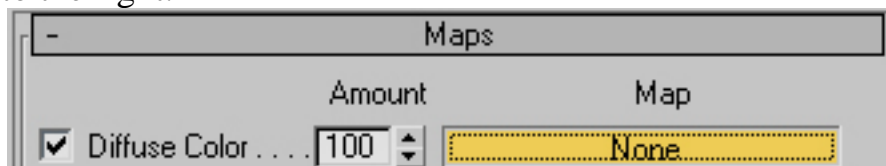


First, let's apply the correct texture to it. Open the material editor and click on any of the unused spheres. Rename it to AK47 and change the shader to **Crytek Shader** under the "Shader Basic Parameters" rollout. Under the "Crytek Shader Basic Parameters" select the shader called "**TemplBumpSpec\_GlossAlpha**". Place a check mark on the "No Shadow" box. Lock the ambient and diffuse colors and make them a neutral gray.

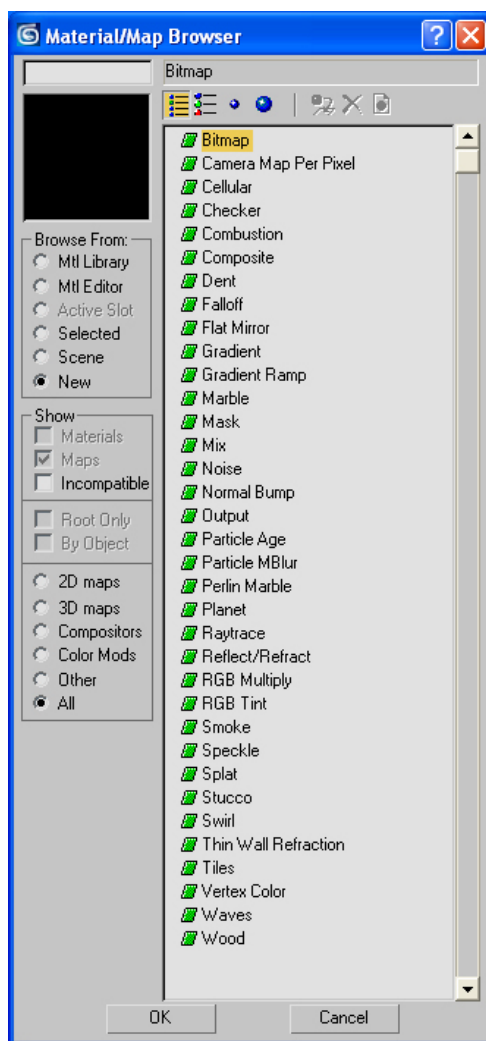
**Note:** You can adjust the Specular Level and Glossiness to your liking. Also, there are a multitude of shaders you can use; find the ones that suit your weapons to your liking.





Scroll down until you see the “Maps” rollout and left click it to open it. The first option is “Diffuse Color”. Go ahead and place a check mark on it to activate it. Then click on the “None” button to the right.




A new window opens up. Select “Bitmap” by double clicking it. Another window appears; look for the AK47.dds texture file included with this tutorial and load it up.



Now that the texture is loaded into the material editor, make sure the AK47 is selected in the Perspective Viewport. Click the “Assign Material to Selection” button  then click the “Show Map in Viewport” button . Now you can see the AK47 is correctly textured. Now let’s tackle the scaling problem.


# Scaling Uniformly

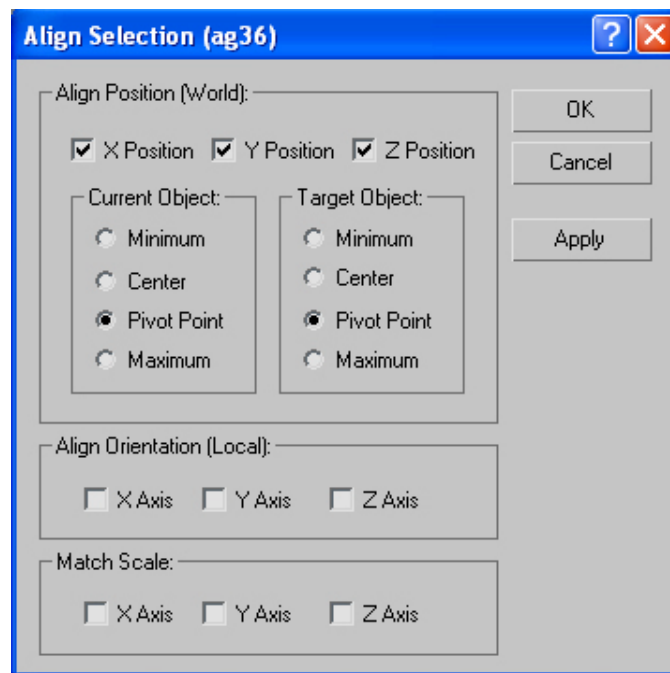
Close the material editor by hitting the shortcut key, “**M**”. Now we have to scale down the model, but we have to do it uniformly so we don’t get any unwanted mesh stretching.

Look for the “Scaling” button and make sure it is set to “Select and Uniform Scale” . If it’s not, simply left click and hold on top of the “Scale” button and you will see a few hidden buttons appear underneath; simply click on the correct scaling button. Select the AK47 and turn on scaling mode with the “**R**” button shortcut. In the bottom of the screen you can see a field where you can type numbers for the X Axis. If you type in a number and hit Enter, the model will be scaled down according to the number you entered. The smaller the number, the smaller the weapon becomes. Let’s input the number 8 and hit Enter; the weapon is now an adequate size for the arms.



The weapon is out of place so let’s quickly align it to the arms. With the AK47 still

selected, hit the “Align” button , then click on the arms. When the align window opens, select the following options:



Now click OK. Now the weapon is textured, and aligned better. Let’s adjust it though, by moving it a little farther away from the arms on the Y Axis.

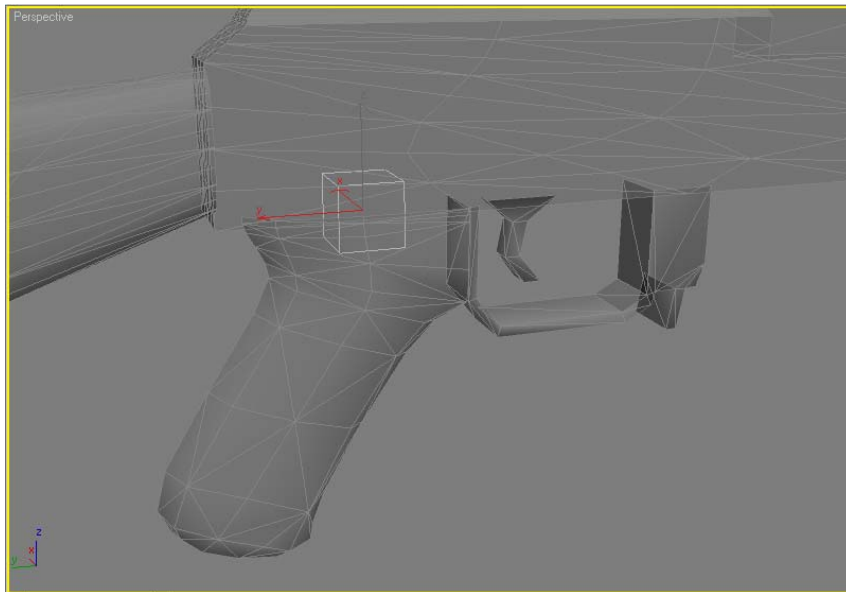




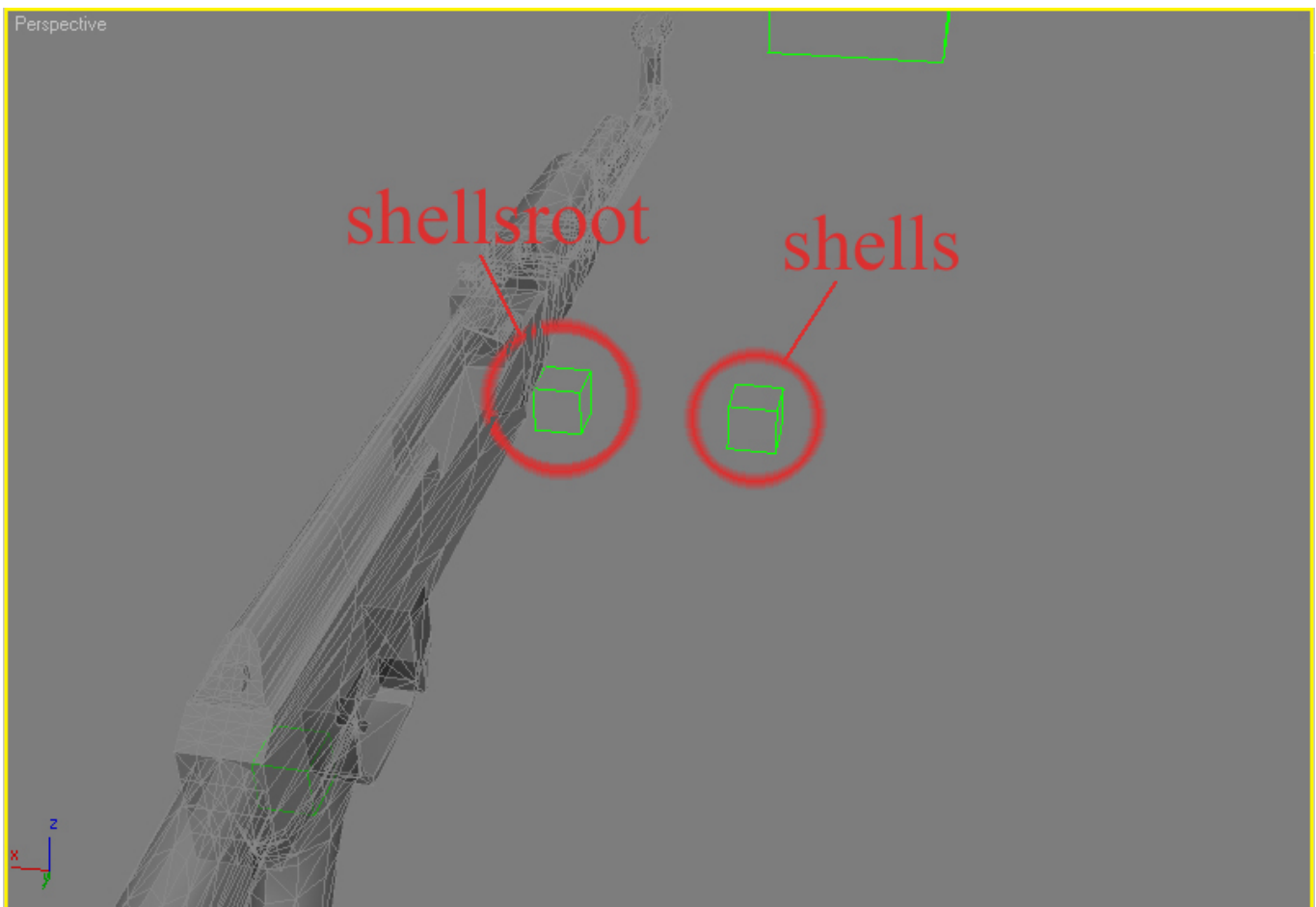
Select the arms and then attach the AK47 to them. Now make it see-through like we did before and freeze it via the quad menu (Right click on the mesh to open the quad menu).

## Setting Up Bones for the Weapon

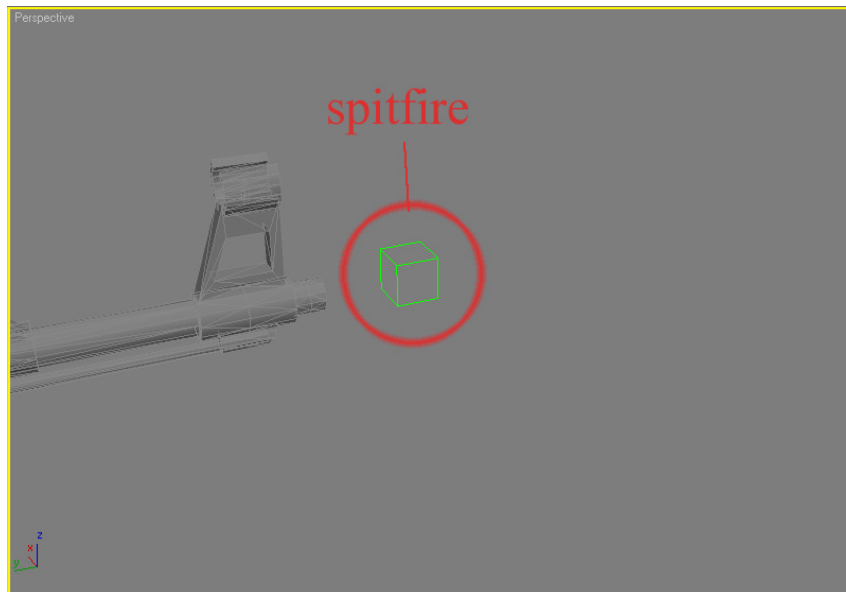
Everyone I know of in the FarCry mod community uses Box objects as the bones for their weapons. Although this sometimes works fine, I have found that this is a huge mistake and can cause you to lose hours of work redoing your rigging. The reason for this is that the CryEngine does not work well when Boxes are used as bones. This is clearly stated in one of the documentations provided by Crytek in the Mod SDK. This can cause horrendous mesh deformation and massively painful headaches when working on a weapon's animations. I myself use Dummy Helpers instead as they have proven very useful and never cause mesh deformations: making them perfect for use as your weapon's bones. Using whatever views you're most comfortable with, create and place a Dummy Helper inside the weapon near the grip and call it "**weapon**".



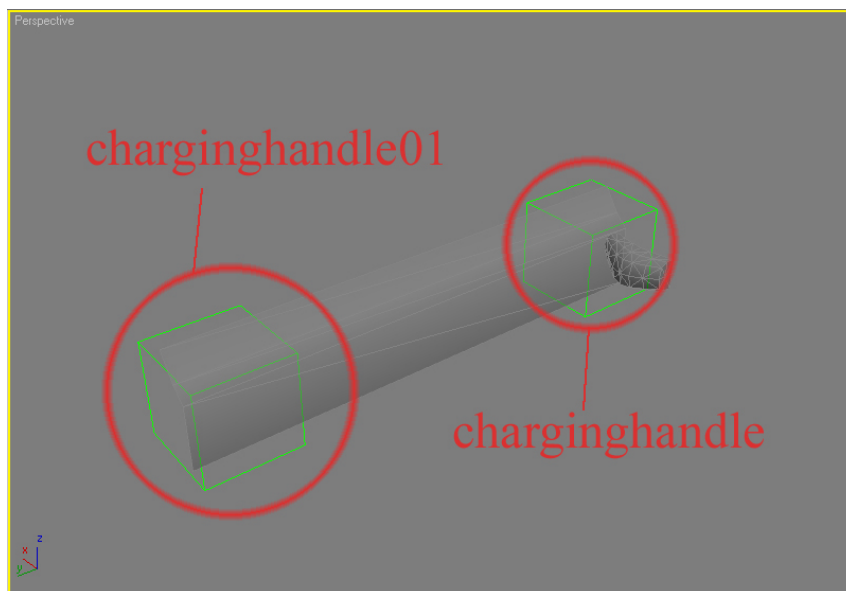
Place a copy of the Dummy helper at the end of the stock and name it “Stock”. Make another copy and place it by the ejection port and call it “**shellsroot**” and make another copy next to “**shellsroot**” and call it “**shells**”.



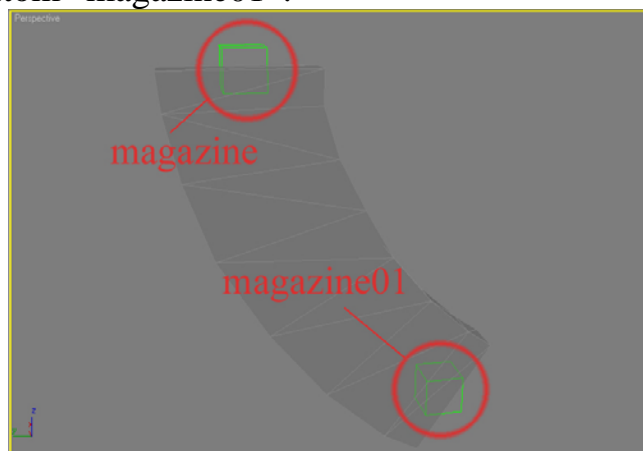
With that done, make another copy and place it in front of the barrel, call it “**spitfire**”. Make sure it is just in front of the barrel tip; if it is too close you will have clipping issues with the muzzle flash in the game.



Now make another copy and place it at the charging handle, call it “charginghandle”. Make another copy and place it at the other end of the charging handle, leave it at the default name which will be “charginghandle01”.



Last but not least, make 2 more dummy helpers for the magazine, make the top one “magazine” and the bottom “magazine01”.



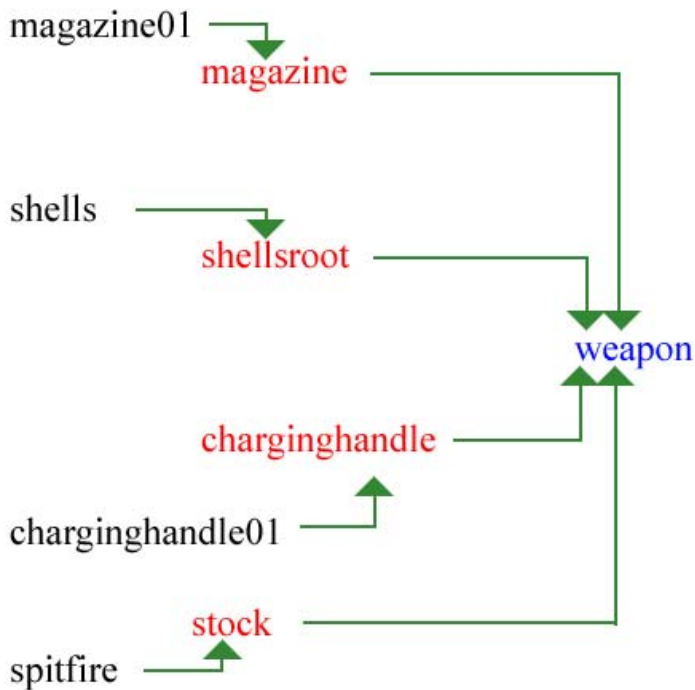
# Linking the Bones

Now let's link the gun's bones. Ultimately, the bone called "weapon" will be the parent of all of the gun bones. That is to say, if you move or rotate the "weapon" bone, all the bones

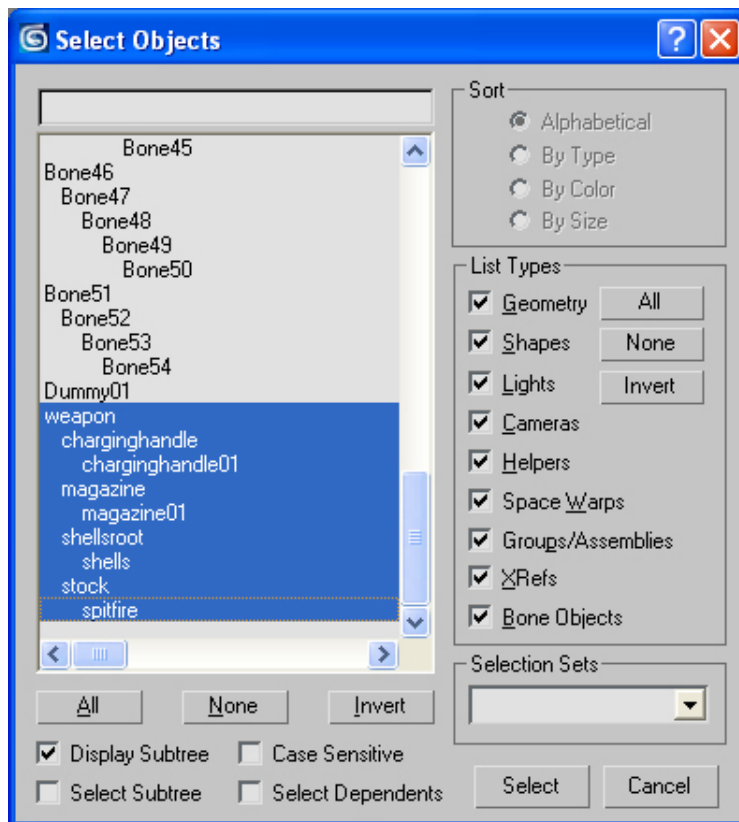


will move too. So let's begin by first clicking on the "Link" tool to activate it. Now click on "magazine01" once, then click and drag the mouse until you get on top of "magazine" and let go of the button. "Magazine" will turn white for about a second and a half, this means you linked it correctly. This also made "magazine" the parent of "magazine01". Remember, the first one you click on with the Link tool is the child, the second one you drag on top of is the parent; that's how it works. Now link "magazine" to "weapon"; in other words, click on "magazine" and drag up to "weapon" and let go of the button like we did before. If at this point you still don't understand the linking concept, go back and practice some before continuing simply because linking correctly is very important and if done wrong you will have serious problems later. To finish this up, link the gun bones in this order:

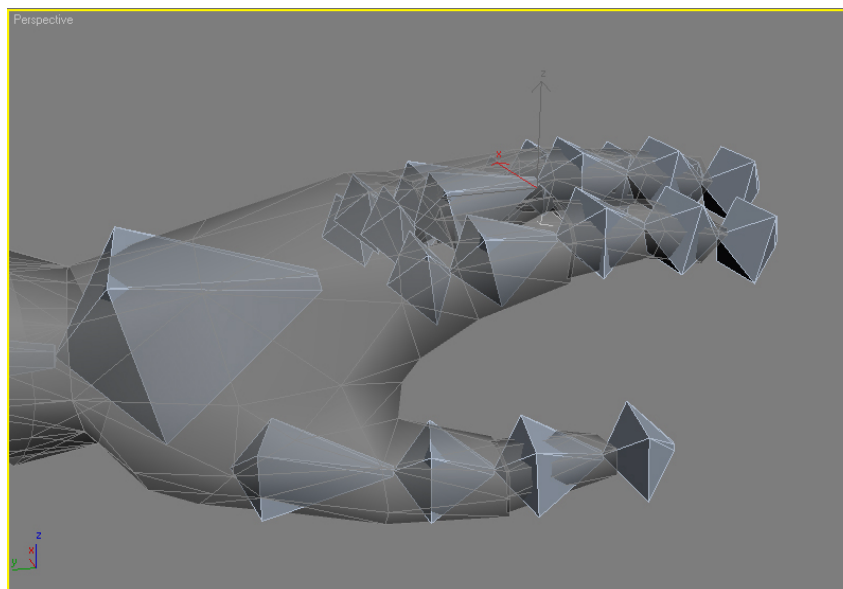
Note: "weapon" is the parent of all the bones. The red bones are the parents of the black ones. This order, or hierarchy, will be essential for animation.



To make sure you have the correct parent-child order, press "H" to open the "Select Objects" window. You will see a list of everything in the scene. Check the "Display Tree" option on by placing a check mark on it, this will display the objects in your scene by parent-child order. Yours should look like the objects highlighted in blue.

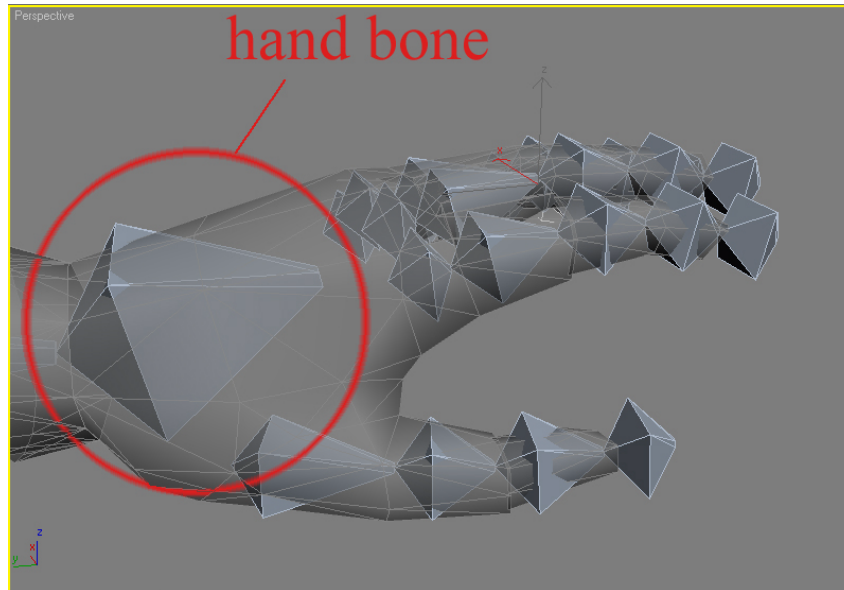


Now we have to go back and position the bones in the arms correctly. Your mesh should still be frozen; which will come in handy now. Select the arm bones and begin positioning them correctly into the arms. Do the same with the finger bones. Here's a picture to use for reference:



Once you finish positioning the bones into the arms and hands you need to make the final links. Select the 2 large arm bones in the beginning of the upper arms one by one and link them to the Root dummy helper we made earlier. Make sure the Root is the **parent**. After doing that you need to link the “**weapon**” bone to one of the hand bones. If you want to shoot left handed, link the “**weapon**” bone to the left hand; for right handed link to the

right hand bone. Assuming it will be right-handed, link the “**weapon**” bone to the right hand bone so the hand bone will be the parent.




Also, link the root finger bones (those closest to the hand bone) to the hand bone in both arms. Now if you rotate the hand bone in either hand, the fingers will rotate in the same direction as well. Ok, now everything should be linked and ultimately the Root helper is the parent of all bones. Using the “H” shortcut again, check the hierarchy again and make sure one final time that the order consists of all bones in the scene, and they truly are the children of the Root. If you did everything right and followed everything correctly, hopefully, now we can get on with applying the physique modifier. The physique modifier will allow us to make the weapon capable of animation. Note however, that for Max to have this modifier available, you need to have Character Studio. It is the key component in Max that lets you animate characters and such. I have run into people on the internet, while helping them, who had Max but not Character Studio. This effectively stopped them dead in their tracks from continuing to mod weapons. So please be careful and alert to the tools you need to mod successfully.

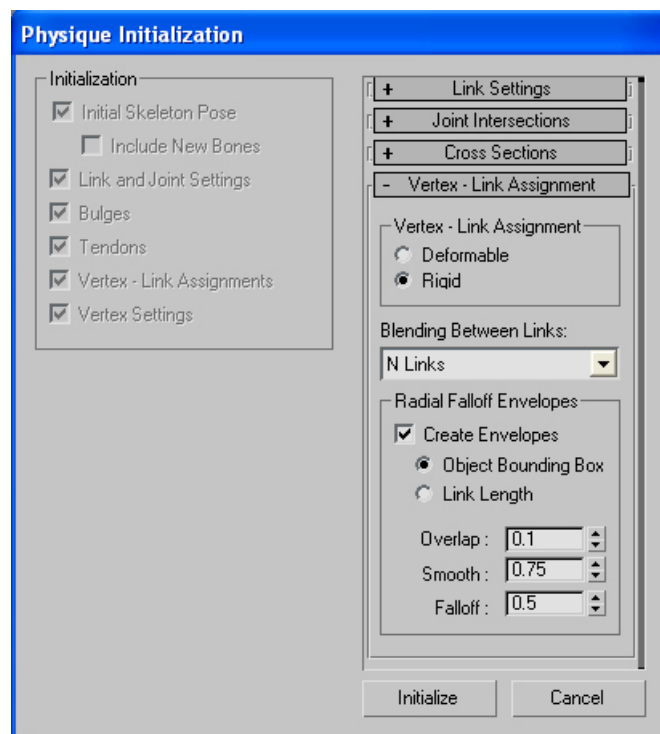
## Envelopes and Their Importance

Envelopes are something that will be troublesome to understand for someone who has never animated in Max before. If this is your first taste of animation in Max, please be patient and open minded because you are about to have something thrown at you which is pretty confusing for some people, on the other hand, others seem to catch on to it pretty quick. So don't get frustrated, just try to follow along as best you can and remember that the most important thing for you to learn on how to implement envelopes is first understanding what they are and what they do. If you are experienced with animation and envelopes, go ahead and skip onto the next section “Applying the Physique Modifier”. However, if you are new to this aspect of Max I strongly recommend you read this section first before trying to follow the tutorial any longer.

Envelopes are a component used in Max when you apply a physique modifier to a mesh. You probably don't know what a physique modifier is anyway since you're reading this section. In Max, the physique modifier allows you to animate your meshes. The principle behind it is simple really. First you make a mesh, for example the arms and gun in this tutorial. You then apply bones to that mesh. The physique modifier allows you to animate your mesh because what it does is, in layman terms, makes the mesh "stick" to the bones. This way, if you say, move the elbow, the mesh or "skin" of the model follows the movement of the bone. Just like in real life your muscles and skin stay stuck to your bones (hopefully), the physique modifier makes your mesh stick to your bones in the same way. Your mesh will act just like a real arm. This is where envelopes come in. Max doesn't know how you want your mesh to stick to the bones. So you tell it by way of the envelopes. The envelopes will determine which vertices move with which bones. The envelopes are sub objects of the selected bone. For example, the last bone on the tip of the thumb has its own envelope just as all other bones do. What you would do is adjust the size of the envelope of that thumb bone so it encompasses the correct vertices of the mesh. Since it's the bone on the tip of the thumb, it should have an envelope that engulfs the vertices only of the tip of the thumb of the mesh because in real life the bone on the tip of the thumb, if moved, only moves with it the skin and flesh of, well, the tip of the thumb. Adjusting the envelopes is a pretty long task since you need to adjust the envelopes of every single bone, because each bone has its own envelope. There is some good news though. Some bones don't need their envelopes adjusted because when you apply the physique modifier, Max makes its own initial calculations of how it "thinks" the envelopes should be based on the bones' shape, size, location within the mesh, etc. These default settings need to be readjusted by the animator for the most part. But there are a few bones that will be ready to go. In this case, from experience, I have found that the hand and forearm bones almost never need to be adjusted. That's only a minority of the bones, so yes, there will be plenty of work to do. To adjust envelopes you have a few tools. You first need to access the envelopes though. After the physique modifier is applied to the mesh, you will be able to choose the "Envelope" sub object level, much like when you select the "Polygon" sub object level when you are modeling or working on a mesh. You select the arms mesh, go to the "Modify" tab, expand "Physique" by clicking the little "+" icon, then click on "Envelope" and it will be highlighted in yellow; telling you that it is active. Now that you understand what envelopes are, or not, let's move on to the next part of the tutorial where we will give the mesh a physique modifier.

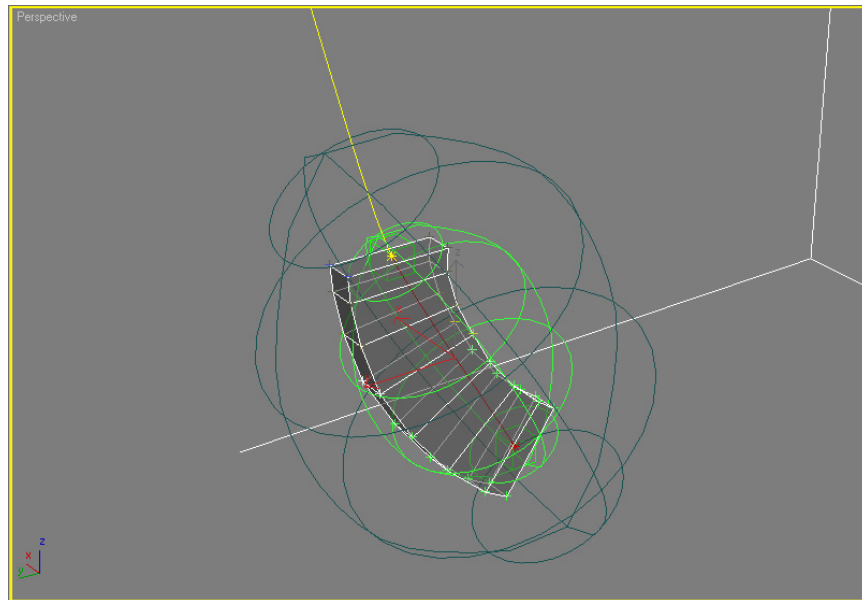
## Applying the Physique Modifier

Let's apply the physique modifier now. Select the mesh and then click on the "Modify" tab to open it. Click on the box that says "Modifier List". A drop down list opens. Scroll down to "Physique" and click on it. Next, click on the "Attach to Node" button  and then click on the Root dummy helper. When the "Physique Initialization" window opens, select the "Rigid" option and click on "Initialize".

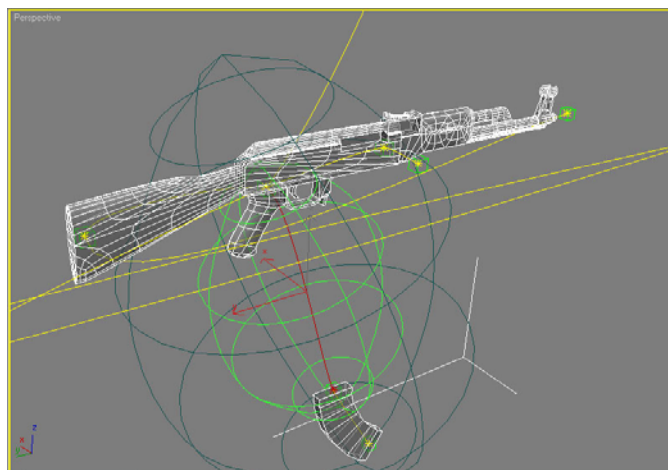


Now the model is capable of being animated. First, however, we need to adjust the envelopes I spoke about earlier. In the "Modify" tab-under "Physique"-click on "Envelope" so it is highlighted in yellow. The first thing you will notice is that strange yellow lines have appeared. These lines, if clicked on with the select tool, will reveal the envelopes. Go ahead and click on the yellow line inside the magazine. When you do, two green spherical objects appear around the line from the magazine's bottom bone to the top one. The dark green sphere is the outer radius and the light green sphere is the inner radius of the envelope.



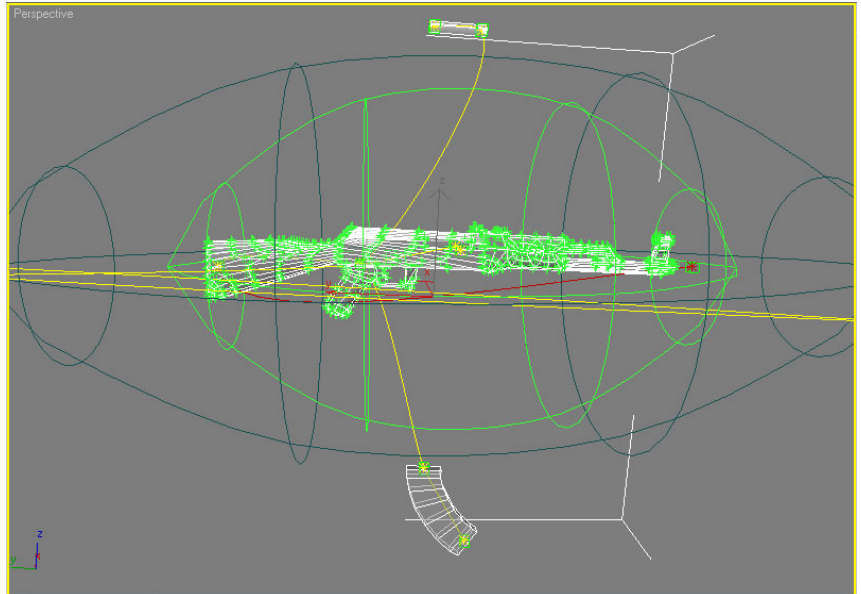


Upon close inspection you can tell that some vertices of the magazine are a light green color while others may be yellow as well as some of them being varying degrees of the two colors; other vertices aren't even highlighted. This is how it works: the very light green vertices are being influenced heavily by the envelope of that bone, meaning that if you move or rotate the corresponding bone those vertices will move with it. However, the vertices that are less influenced, the yellow ones, will not move as harshly. If you use your imagination, you can see this is very useful for animating, but for a magazine which is supposed to be a solid object, vertices moving this way will lead to stretching and deforming when you animate it; since this isn't some kind of mutant or anything weird, we need all the vertices to be light green and be completely influenced by the envelope here so that when you animate it, the magazine moves realistically. To change the influence of the envelope look at the parameters for the envelope and change the "Strength" setting from 1.0 to 100; hit enter. If you got lucky, all the vertices of the magazine will become light green, meaning that they are under the envelope's influence 100%. If some of the vertices are not light green, simply change the "Radial Scale" option from 1.0 to whatever size works best, probably something bigger than 1.0. With the magazine's envelope properly adjusted, select the next envelope above the magazine's top and change its strength to 0 so it has no influence. I know it seems wrong to do it, but it will work out in the end and later on you will see why.

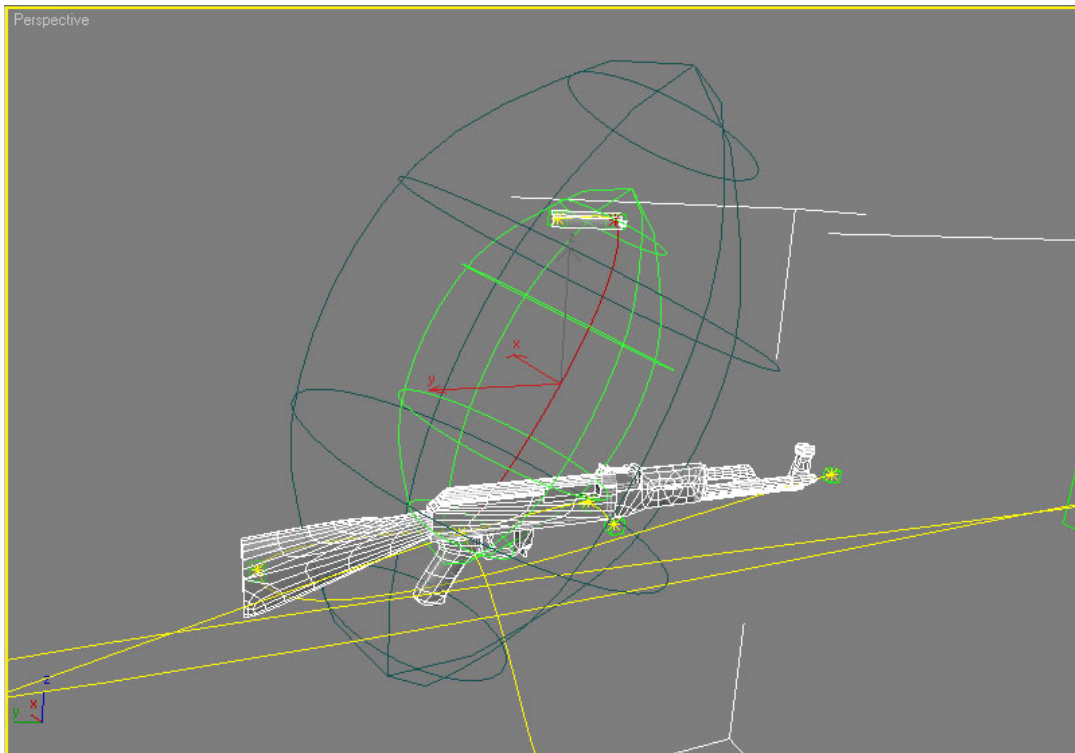


Next, select the envelope for the bone at the tip of the barrel. You can see this is the envelope for the main body of the weapon from end to end. Make sure to change the size of the envelope's radius and aim at making it engulf the entire body except for separate pieces like the charging handle, magazine, etc. Then change the strength to 100 in order to have complete influence over the required mesh vertices.

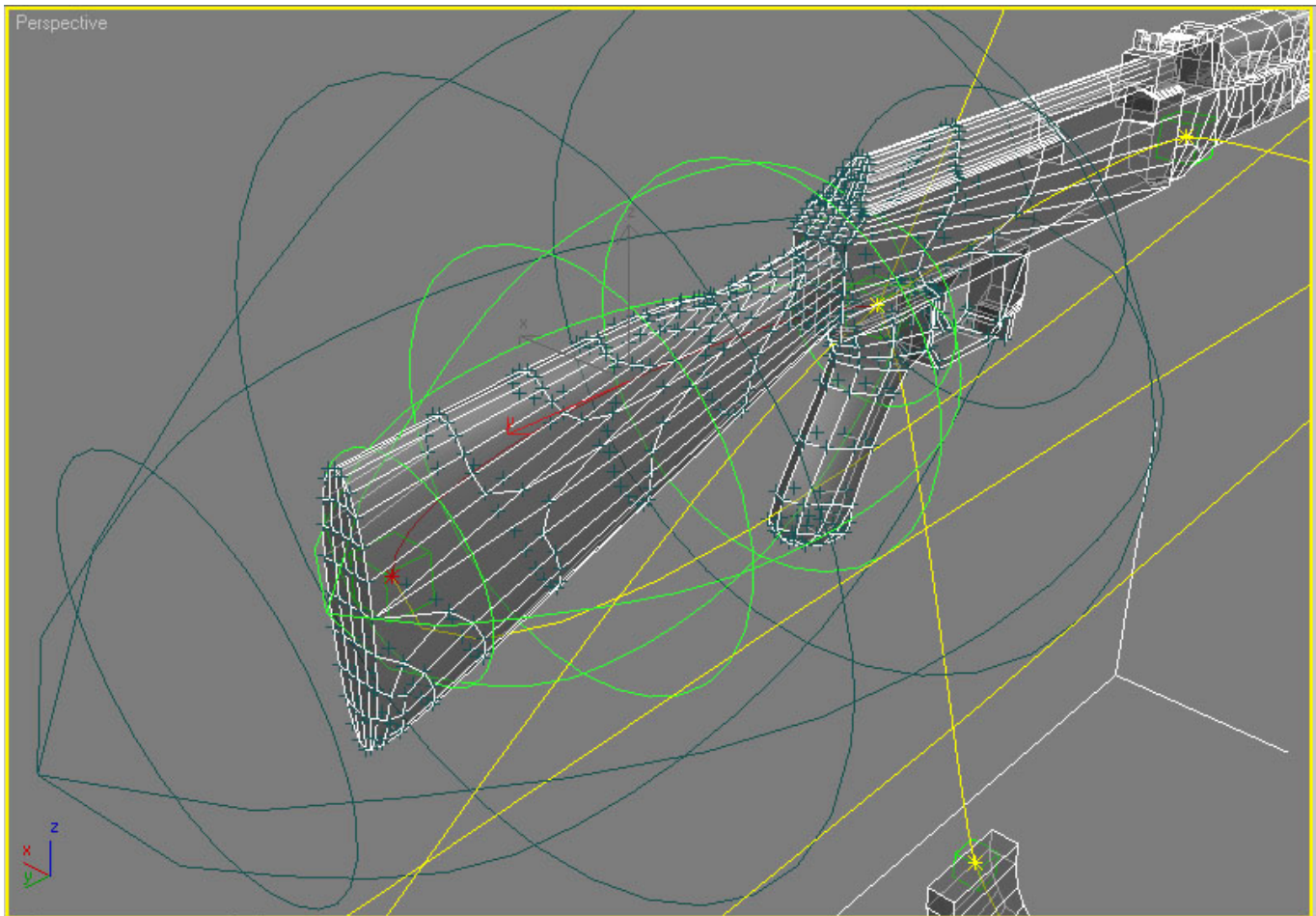
**Note:** You can adjust the outer radius (Dark green) and the inner radius (Light green) independently under the Blending Envelopes rollout. By default it is set to "Both".



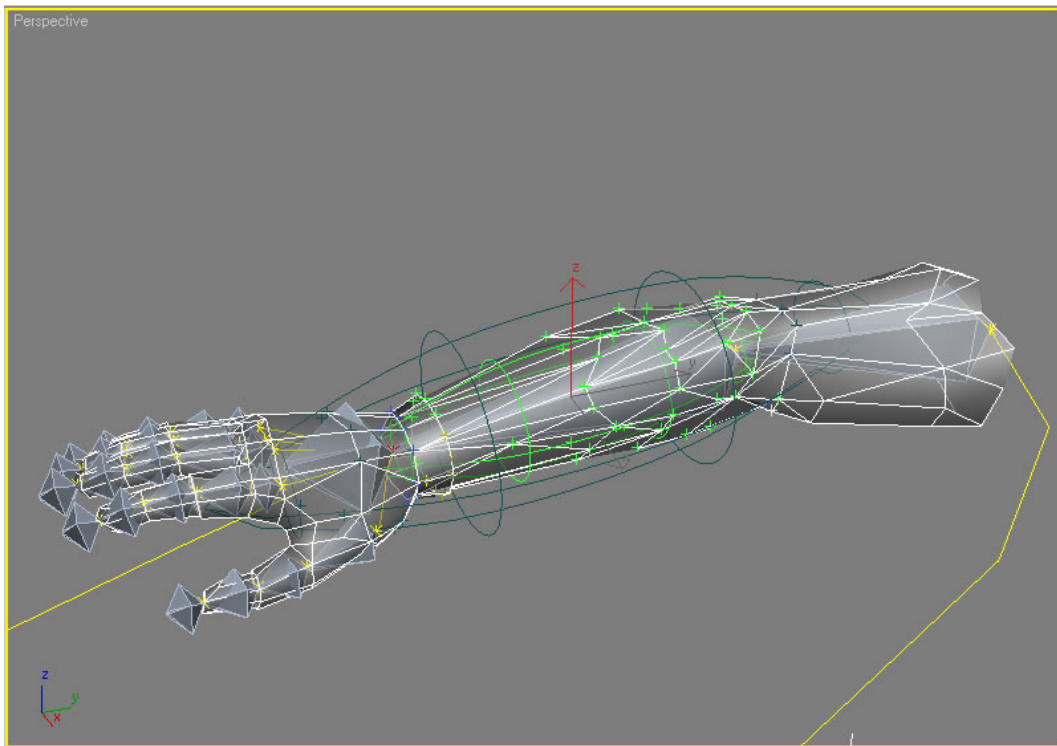
Now change the envelope that goes from the charging handle to the weapon; make it have a strength of 0.



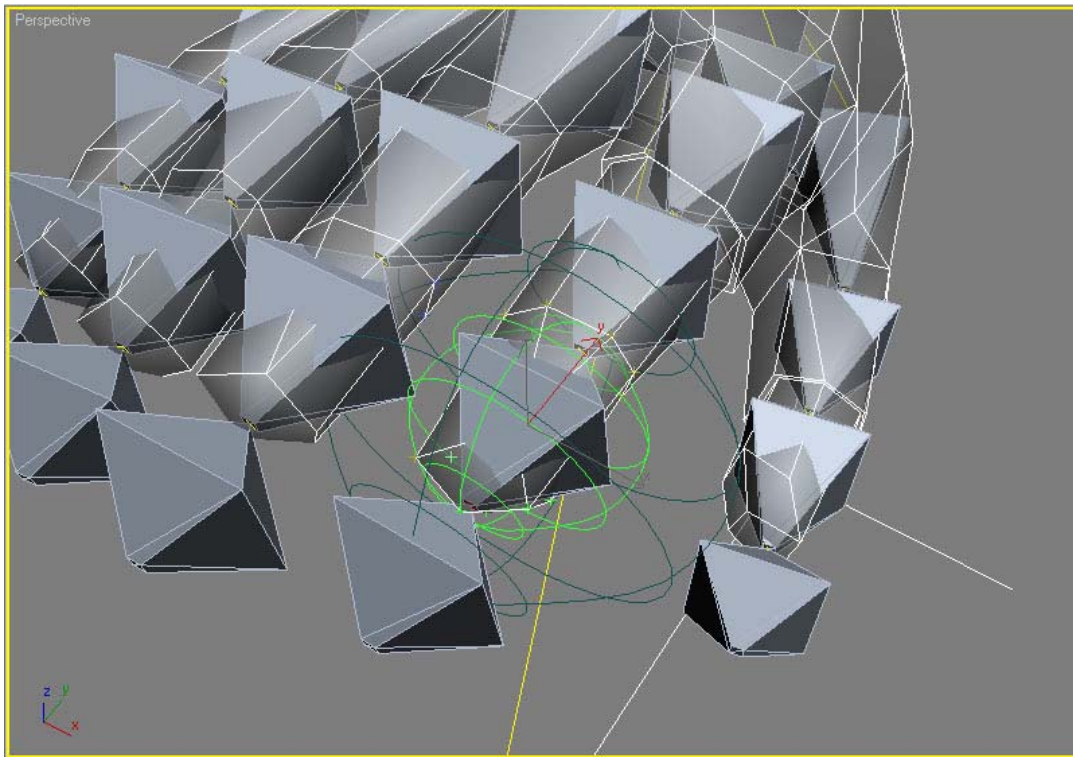
Now make the envelope for the charging handle (the one that only engulfs the charging handle) have a strength of 100, similar to what we did with the magazine earlier. Hopefully by now you are starting to understand and see how envelopes work and how they can be adjusted, if not then simply go back and try this section again until you grasp the concept better or else you will get lost very easily from here on out. There are 2 envelopes that stretch from the Root helper to the ends of the arms; make them both have 0 strength. Another envelope runs from the shooting hand to the weapon itself; make it also 0 strength. Now make the envelope of the stock 0 also.



The weapon envelopes are now set, so let's get to work on the arms and hands. These will be very important to get right because if not, then you will have awkward and ugly deformations when you animate your arms and of course we don't want that. Let's begin working on the right arm and when it's done we will move onto the left arm. Select the envelope for the forearm and increase the radius slightly, 1.15 should work fine, but if not then tweak it to what works best. It is different for everyone because remember that Max automatically "guesses" the computation of the envelopes based on the bones' placement and the mesh, so little tweaks like this are necessary.



The next envelope to tweak is the one for the arm bone where the bicep is located. You will notice not all of the vertices towards the back of it are under the envelope's influence so simply increase the Radial Scale to 1.3 and the Parent Overlap to 0.5. Now if you were to animate this bone, the mesh would act realistically by following the bone's movement correctly. Here comes the most tedious part of adjusting envelopes, at least in my opinion, the fingers. Fortunately I find that the wrist/hand bones never need adjusting, or at least that has been the case for me. However the fingers are a whole other story. We will start with the thumb since it's the easiest. Change the Child Overlap of the envelope for the tip of the thumb bone to 1.15 and that should make sure it works. That's it for the thumb. Move on to the index finger and start with the tip bone's envelope. Do the same thing we did to the tip of the thumb by changing the Child Overlap to 1.15. Do this to every single bone that is a tip of a finger, this will need to be done for all five of each hand. You will also notice that the outer radius of the envelope for most finger bones overlap vertices from neighboring fingers, this is wrong and we need to fix it or else when one finger moves, it will stretch and pull vertices from the finger(s) next to it. To save time however, let's only adjust the first two bones starting with the bones at the tip of the fingers, except of course the thumbs, which we are already done with.



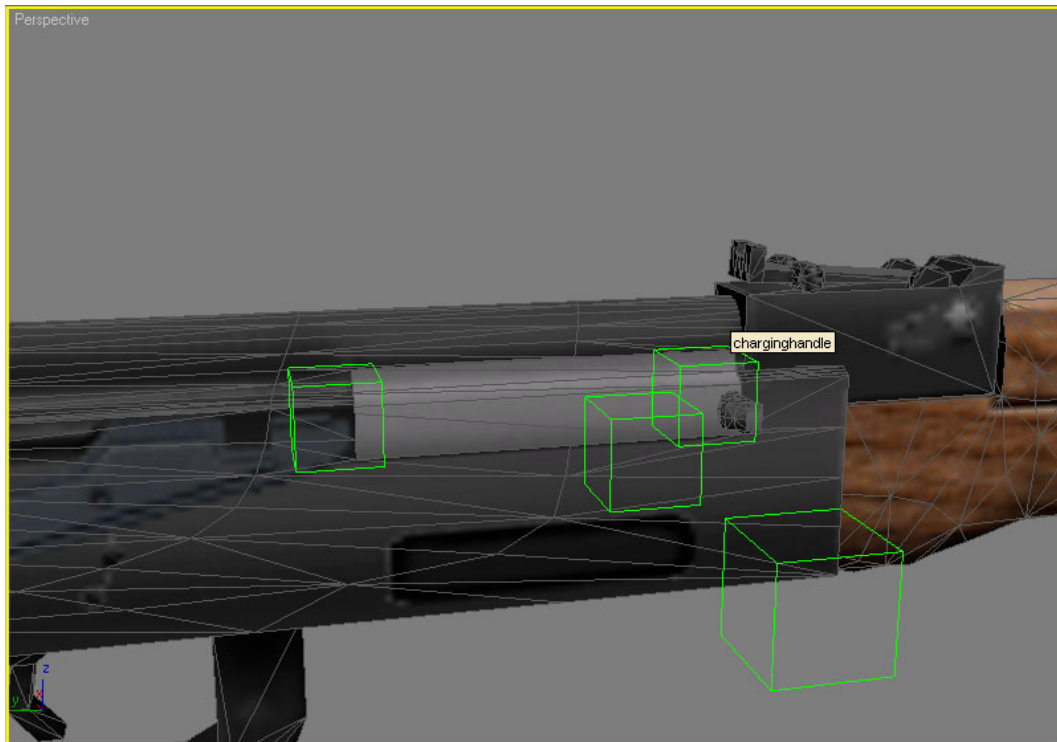
Just change the outer radius and NOT the inner radius. Decrease the outer radius slowly until you can actually see no neighboring vertices being influenced by it. It is important that you don't decrease it too much or you will have problems later when animating. Carefully adjust these outer radii for all the first two bones in each finger(except the thumbs)until the outer radius just releases the last influenced vertex of the neighboring finger. Once you have done this for all fingers, get out of the Envelope Sub object mode by clicking on Envelope in Parameters so it is no longer highlighted in yellow. Now select the last upper arm bone, not the mesh just the bone. Rotate it around a bit to test and see if the arm mesh acts and moves realistically. If there is any strange stretching, you did something wrong. Just hit Control+Z to undo. Make sure after moving it around you undo the movement so it is back to its original location. Assuming you followed correctly up to this point and the arm is working fine, go ahead and do what we did to this arm and apply all of the techniques to the left arm and test that one too.

## Preparing to Animate

Ok, so you have spent hours modeling, skinning, and rigging the model. The good news is that you are about half way through completing your weapon mod! Even greater news is that now we will animate it for use in FarCry. Before you animate it though, the arms need to be holding the gun in the ready position, as if our character is prepared to vaporize anyone or anything that gets in his way. The arms have to look like they do when you play the game from first person view. This is rather easier than you may think. We no longer need to adjust envelopes because by now you should have had them adjusted. Instead we will be directly moving the bones inside the mesh in order to get the arms and weapon in their correct position. It's as easy as simply moving the fingers and arms to the right position. Keep in mind however that certain bones should be rotated and only rotated, while others can be moved and rotated. Fingers should only be rotated and never moved. Just trust me on that one. Since you have read all the way to page 29 of this tutorial, I assume you do have some degree of trust in my advice. The hands/wrists should also only be rotated. Same goes for forearms. However, the last upper arm bones can be moved or rotated. The weapon parent bone can also be moved or rotated. Same goes for the parent bones of the magazine and charging handle. Now let's position our model. Take the charging handle and put it in place by moving its parent bone only. As long as you move the parent bone, the charging handle will move exactly as if it was a regular mesh, so go ahead and put it in its correct place in respect to the weapon, use this screen for help:

### Note:

Remember you can change the See-Through property by selecting the mesh, right clicking it, selecting "Properties from the Quad Menu and turning "See-Through" on or off.



Now put the magazine into position by moving its parent bone, like so:



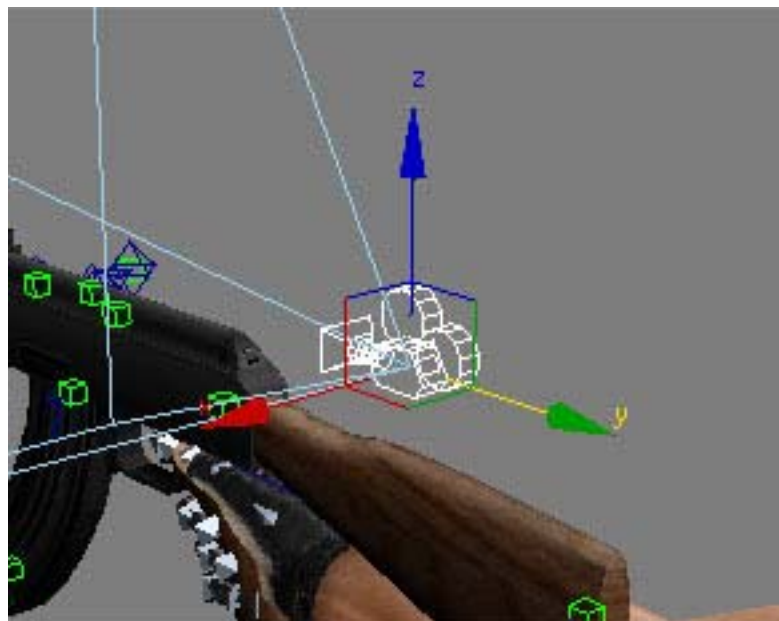
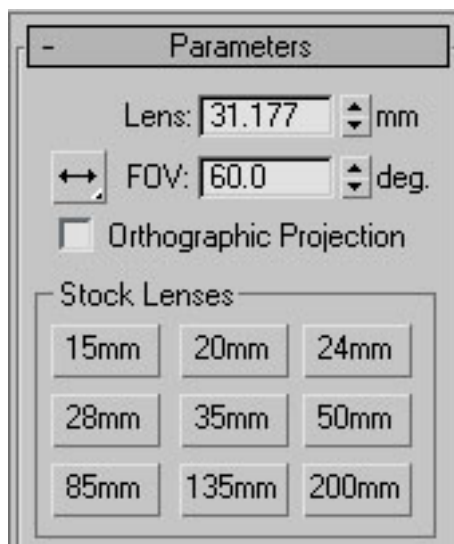
Ok, now the weapon is ready to go into the shooting hand, in my case, the right arm. Select the bone called weapon, since it is the parent of the whole gun, we will move it by using this bone mostly. Rotate the weapon 90 degrees so it is lying sideways on its left side (the charging handle should be facing up while the barrel is still pointing ahead). Move the weapon so the handle, or grip, is in the shooting hand. Just eyeball it and it should be fine. All you have to do now is wrap the fingers around the grip so the hand looks like it is holding the gun. Remember to only rotate the finger bones and not move them. By now you should have a good understanding of what's going on. Just go ahead and position the arms and hands with the weapon as a real person would hold it. Mine's looks like this:

**Note:** Don't hesitate to spend a great deal of time adjusting the arms. If not, your arms may look rather deformed because of unrealistic postures and positioning. Make the arms flow correctly and the fingers shouldn't bend backwards or anything like that. Look at your own hands for reference while holding something similar.



# Animation

We can now animate the weapon and arms. Animating weapons is where a lot of people in the Farcry modding community get stuck helplessly. Even veteran animators have trouble getting their animations to work in Cryengine. First and foremost, I really hope you have at least some understanding of how to animate in Max. Normally what you do is place a key in your timeline, move an object, set another key, move it again, set another key, and then you have a very basic animation. For Farcry it works very differently. I will now show you how to make a simple firing animation with recoil. It should be noted that it is the ONLY animation I will show you how to do. The reason I won't walk you through every animation like reloading, swimming, fidgeting, etc. is simply because it would take too much time and mainly because those animations are far too advanced for a beginner to successfully complete so I don't want you to try something like that because you may get discouraged if it seems to hard. Not to say you can't do it, but instead you should learn how to do a basic animation and then you can experiment on your own and eventually learn to make more complex animations as you practice more, as I myself did. Moving on, let's set up a camera. This step is VERY important to get down right, or else when you test your weapon in-game you may not see it at all! Go to Create and place a Free Camera in your scene. Now position it at point 0,0,0. It is extremely important the camera be placed at this location. Make sure you change its Field of Vision(FOV) to 60 degrees.



**Note:** Make sure to face the camera opposite the Y axis as in the picture above. Also, as soon as it is positioned correctly, double check to make sure it is indeed correctly positioned. As a precaution, immediately freeze the camera in place after it is positioned correctly. If not, you may accidentally move it and have problems later.

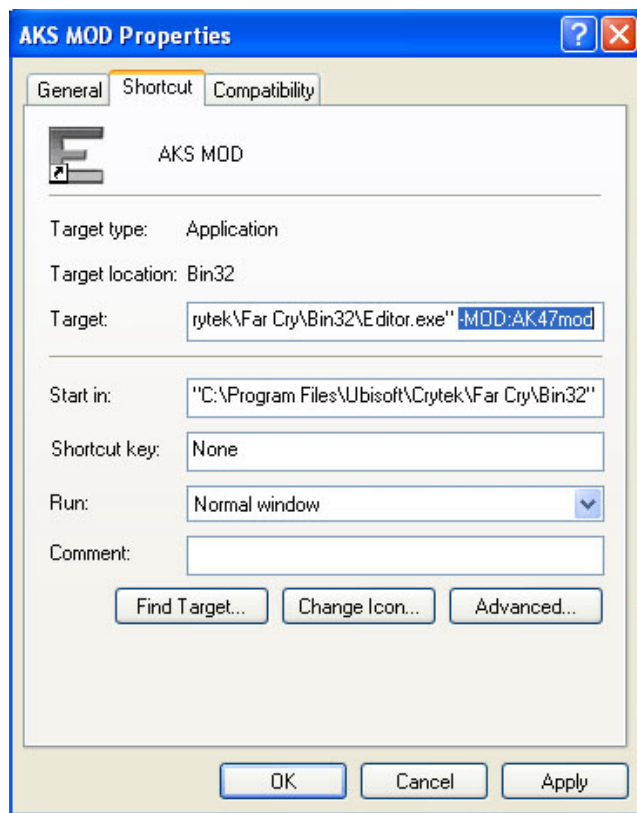


With the camera set up, we can now begin animating. Open the “Motion” tab. Select the Root helper. A host of options and rollouts are now active under the Motion Tab. Expand the Assign Controller rollout and highlight the Position controller by clicking on it once. The small question mark icon just above is the Assign Controller button. Click on it now that it is available to click. In the Assign Position Controller window that popped up, go down to TCB Position. Click on it and click on Make Default. It will ask you if you are sure you want to do this, select yes. Now click ok to exit this window. You have just assigned the correct position controller for your Root helper in order to animate it. There are many controllers in Max for animating, but Cryengine works with TCB controllers, so we will use these for our position, rotation, and scale controllers. So select Rotation this time and assign it a TCB controller, and do the same for scale. Now all three should have TCB controllers assigned. Now on the time line, make sure the time slider is at frame 0. With the Root helper still selected, go to the PRS Parameters rollout under the Motion Tab. On the left is the column for Create Key. Click on all three, Position, Rotation, and Scale to create these keys at frame 0, the first frame of the firing animation. In the timeline you will see a key made up of three different colors at frame 0. Each of the three colors is for each of the 3 keys; i.e. position, rotation, scale. Move the time slider to frame 4. Now create a position and rotation key (we don’t need a scale key in this circumstance). You should now have a position and rotation key set at frame 4. With the time slider still at frame 4, activate the “Auto Key” button (shortcut is N). Now move the Root helper backwards a bit and you will notice that everything moves with it. This is because the Root helper is the parent of everything (except the camera). Use this technique to simulate the recoil. Once you have it as far back as you like, hit the Auto Key button again to deactivate it. Move the time slider to frame 8. Now this will be the last frame in the firing animation. Select the first key from frame 0 and hold shift and move the copy that is automatically made to frame 8, the final frame of the animation. This very simple animation is only 8 frames long and has only 3 key frames. Play it and see if you are satisfied. Even though this basic technique is so simple, it can be applied in more complex animations, like a reload animation. We are now going to go into a whole other process that will momentarily have us working out of Max, but to facilitate the overall process, keep Max open but minimized.

# File Placement

The next step to take is to prepare the file directories in preparation of the new weapon mod. Let's begin by going to the Farcry mods folder. Once there, create a new folder and name it anything you want. Usually you use the name of your mod. For example, you could call it AK47mod or the like. Now go into the new folder and create a folder called "Objects"; in there create a new folder called "Weapons"; in there create a new folder called "AK47". This AK47 folder is where the AK47's texture, 1<sup>st</sup> person model, 3<sup>rd</sup> person model, animations, and .cal files will go. One thing you should be aware of is that file placement is very important. Just as important is the naming conventions of EVERYTHING. Cryengine and its scripts rely a lot on file names, so if one word somewhere is misspelled, your mod won't work. This can be very frustrating because it can be hard to find out why your weapon's muzzle flash won't work or the model doesn't appear in-game or Sandbox crashes with a strange error message. This has happened to me and every modder I have spoken with or worked with. To try to avoid this, ALWAYS triple and even quadruple check any new scripts, folders, or files you create, paying close attention to their names. With that said, let's continue creating the needed folders. Go back to the main AK47mod folder located in the Mods folder. Create a new folder here called "Scripts". In this folder create a new one called "Default"; in there create a new one called "Entities"; in there create two new folders. One called "Weapons" the other called "PICKUPS". That does it as far as folders needed are concerned. One last thing; we need to make a short cut to Sandbox so we can load the new weapon mod and test it out. Make a copy of the Sandbox short cut and place it on your desktop. Rename it to AK47Mod. Right click on it and open its properties. In the Target field, add the following that is highlighted in blue:

**Note:** Remember to add the space right before the "-" in front of the word "MOD" in the Target field. As always, naming conventions must be followed exactly or this shortcut won't load your mod in Sandbox. Click on Apply then click on OK. This new short cut will now load everything from the AK47mod folder. You can continue to add new weapons to that folder and expand the mod by adding more weapons, vehicles, characters, etc.



# Scripting

Now it's time to script the weapon for use in FarCry. Let's begin with the script called ClassRegistry.lua. Go in the FCData folder and open the "Scripts.pak" file and search for ClassRegistry.lua; extract it into the Scripts folder of your mod. Now we are going to edit this script file so go ahead and open it. I myself use UltraEdit, but you can open it with windows notepad. Find the following area of the script and add the yellow highlighted line EXCATLY as shown:

```
-- weaponpickups
{"Pickup", "PickupFalcon",27,"Pickups/PickupFalcon.lua"},
{"Pickup", "PickupAG36",28,"Pickups/PickupAG36.lua"},
{"Pickup", "PickupMP5",29,"Pickups/PickupMP5.lua"},
{"Pickup", "PickupMachete",30,"Pickups/PickupMachete.lua"},
{"Pickup", "PickupShotgun",31,"Pickups/PickupShotgun.lua"},
{"Pickup", "PickupSniperRifle",32,"Pickups/PickupSniperRifle.lua"}, -- 32
{"Pickup", "PickupOICW",33,"Pickups/PickupOICW.lua"},
{"Pickup", "PickupRL",34,"Pickups/PickupRL.lua"},
{"Pickup", "PickupP90",35,"Pickups/PickupP90.lua"},
{"Pickup", "PickupM4",36,"Pickups/PickupM4.lua"},
{"Pickup", "PickupShocker",37,"Pickups/PickupShocker.lua"},
{"Pickup", "PickupWrench",38,"Pickups/PickupWrench.lua"},
{"Pickup", "PickupM249",39,"Pickups/PickupM249.lua"},
{"Pickup", "PickupAKS",300,"Pickups/PickupAKS.lua"}, --300
{"Pickup", "PickupG36c",301,"Pickups/PickupG36c.lua"}, --301
{"Pickup", "PickupBat",302,"Pickups/PickupBat.lua"}, --302
{"Pickup", "PickupHK",305,"Pickups/PickupHK.lua"}, --305
{"Pickup", "PickupMK23",306,"Pickups/PickupMK23.lua"}, --306
{"Pickup", "PickupUMP45",307,"Pickups/PickupUMP45.lua"}, --307
{"Pickup", "PickupM92F",308,"Pickups/PickupM92F.lua"}, --308
{"Pickup", "PickupMossberg500",309,"Pickups/PickupMossberg500.lua"}, --309
{"Pickup", "PickupMP5Navy",310,"Pickups/PickupMP5Navy.lua"}, --310
{"Pickup", "PickupSword",311,"Pickups/PickupSword.lua"}, --311
{"Pickup", "PickupNAsub",312,"Pickups/PickupNAsub.lua"}, --312
{"Pickup", "PickupNaeagle",313,"Pickups/PickupNaeagle.lua"}, --313
{"Pickup", "PickupNApistol",314,"Pickups/PickupNApistol.lua"}, --314
{"Pickup", "PickupColt",315,"Pickups/PickupColt.lua"}, --315
{"Pickup", "PickupAK47",316,"Pickups/PickupAK47.lua"}, --316
```

Save the script and close it. Now go into your mod's 'Scripts\Default\Entities\PICKUPS' folder and grab another script from FarCry like we did before. This time grab the "PickupM4.lua" and place it in this folder. Open the script and edit it like this:

```
Script:LoadScript ("scripts/default/entities/pickups/basepickup.lua");

local params={
    weapon="AK47",
    ammotype="Assault",
    model="Objects/Weapons/AK47/AK47_bind.cgf",
    default_amount=30,
    sound="Sounds/Weapons/M4/m4weapact.wav",
}

PickupAK47=CreateWeaponPickup(params);
```

Save it as "PickupAK47.lua" and close it. Now go to this folder location because we have plenty of scripts to edit: "Scripts\Default\Entities\Weapons". Locate the original "M4.lua" script(in the Scripts.pak file within FCData) and get it into this location. Open it and edit it as the following guide will show:

```
AK47 = {  
    -- DESCRIPTION:  
    -- Single shot is powerful, with a more recoil  
    -- Auto is not as powerful, less recoil and does not travel as far  
    -- Its a louder than the MP-5 but also more powerful  
    -- good all round gun  
  
    name                = "AK47",  
    object              = "Objects/Weapons/AK47/AK47_bind.cgf",  
    character           = "Objects/Weapons/AK47/AK47.cgf",
```

also edit this part of the script:

```
CreateBasicWeapon(AK47);  
  
-----  
--ANIMTABLE  
-----  
--AUTOMATIC FIRE  
AK47.anim_table={  
AK47.anim_table[1]={  
    idle={  
        "Idle11",  
        "Idle12",  
    },  
    reload={  
        "Reload1",  
        "Reload2",  
    },  
    --fidget={  
    --    "fidget11",  
    --},  
    fire={  
        "Fire11",  
        "Fire21",  
    },  
    swim={  
        "swim"  
    },  
    activate={  
        "Activate1"  
    },  
}  
-----  
--SINGLE SHOT  
AK47.anim_table[2]=AK47.anim_table[1];
```

Save it as “AK47.lua” and close it. Working in the same folder, bring in the script called “WeaponsParams.lua”. Scroll down to the M4’s parameters and copy it; paste it and change the name of the new copy of the M4 code to AK47 like this:

```
-----  
--AK47 (Assault rifle)  
-----  
  
AK47={  
    --standard weapon params  
    Std=  
    (  
        --AUTOMATIC FIRE-----  
        (  
            fire_mode_type = FireMode_Instant,  
            damage_type = "normal",  
            aim_improvement=0.04,  
            aim_recoil_modifier = 0.6,  
            accuracy_decay_on_run=0.5,  
            min_accuracy=0.65,  
            max_accuracy=0.93,  
            reload_time=2.3,  
            fire_rate=0.08,  
            distance=200,  
            damage=40,  
            damage_drop_per_meter=.02,  
            bullet_per_shot=1,  
            min_recoil=0.3,  
            max_recoil=0.6,  
            iImpactForceMul = 25,  
            iImpactForceMulFinal = 40,  
            iImpactForceMulFinalTorso = 130,  
            hud_icon="auto",  
            accuracy_modifier_prone = 0.5,  
            accuracy_modifier_crouch = 0.7,  
            recoil_modifier_standing = 1.0,  
        )  
    )  
}
```

Save it in this location(Scripts\Default\Entities\Weapons) and close it. Next, go get a script called “WeaponSystem.lua” from the “Scripts.pak” and bring it into this same folder we are working in right now. Open it and edit it like this:

```
AK47 = {  
    id = 316,  
    script = "Weapons/AK47.lua",  
},
```

Once done, save it and close it. That will do it for the initial scripting. Initial because we can still fine tune many settings for the weapon via these scripts. Things like fire rates, fire modes, ammo capacity, and many others can be altered. However, I’m only showing you the basics in order to get the weapon working, so you will have to experiment with other settings and parameters on your own since that’s outside the scope of this tutorial.

## Exporting Animations

Now we need to get all our .cgf, .dds, .cal, and .caf files in this folder directory: “Objects\Weapons\AK47”. Go ahead and place the AK47.dds texture that came with this tutorial in this location. Do the same for the hand-hero.dds texture as well. Next, go back into Max so we can export the AK47.cgf and AK47\_bind.cgf files. In Max, go to the Utilities Tab and activate the Cryexporter2 plugin. Select the AK47 mesh in Max and hit the Add Selected button; the AK47 now appears in the list of objects to be exported to Cryengine. Now click on the “Export Nodes” button.



When the save window opens, name the file to be exported as **AK47** and save it as **.cgf**; save it in this location: “Objects\Weapons\AK47”. Within the same plugin, go down to the Bone Export rollout and in the perspective view select the Root helper. Now click on the Add Selected button. Make sure the option to ignore dummies is off and select the Manual Range check box to on. Change the Time Range from Start 0 to End 8. Now click on Export Bones. Your Options should look like this:



When Max asks you how you wish to save the animation, save it as **.caf** and name it **AK47\_fire11**. Make sure it is saved in “Objects\Weapons\AK47”. Go into the “Objects.pak” and grab the “M4.cal” file. Place it in the same directory we have been working with so far. Open the M4.cal file and edit it like this:

**Note: Save the .cal file as “AK47.cal” and close it.**

```

$AnimDir = .
$Autounload = no
$DelayLoad = no

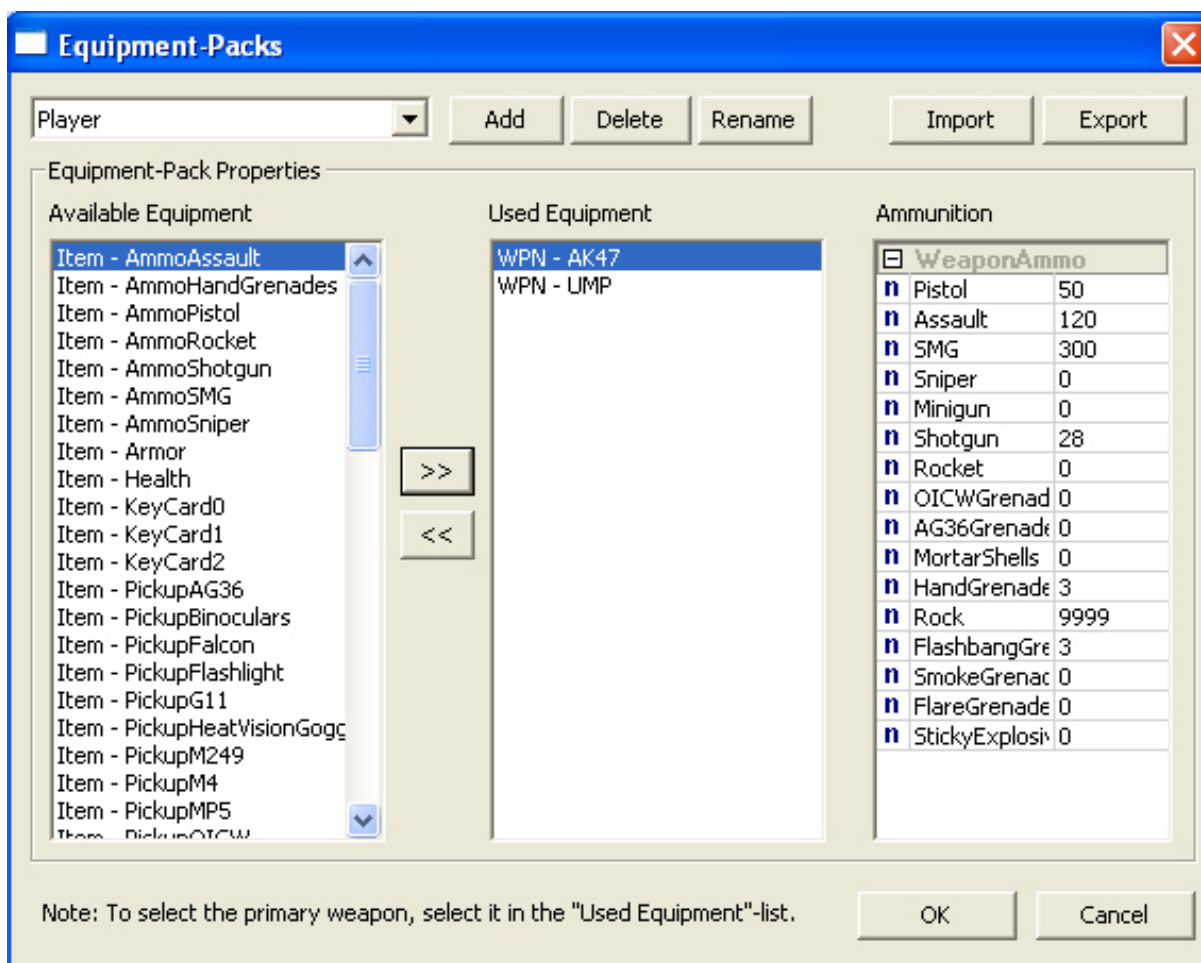
activate1 = AK47_activate1.caf
fidget11 = AK47_fidget11.caf
fire11 = AK47_fire11.caf
fire21 = AK47_fire21.caf
fire23 = AK47_fire23.caf
grenade11 = AK47_grenade11.caf
grenade21 = AK47_grenade21.caf
idle11 = AK47_idle11.caf
idle12 = AK47_idle21.caf
reload1 = AK47_reload1.caf
reload2 = AK47_reload2.caf
swim = AK47_swim.caf
deactivate1 = AK47_deactivate1.caf
$Autounload = yes
$DelayLoad = yes

```

Now we need to create the bind version, which is the 3<sup>rd</sup> person model of the weapon. Open the Max source file included with this tutorial called “AK-47\_Bind.max”. Open the Cryexporter2 utility under the Utility Tab. Under the Object Export rollout click on the Pick button and in the perspective view select the AK47 model. Click the Export Nodes button and save the file as **.cgf** again and name it **AK47\_bind**. Make sure to save it in the same location as we have been saving the animation files, the first person model, and cal file. Once that’s done you can close Max as we won’t be using it anymore.

## Testing the Mod

Now let’s test the new weapon for Farcry. Using the shortcut we made earlier for Sandbox called AK47mod, open Sandbox. Create a very simple new level to test weapons in. Go up to the Mission button that runs at the top where File is. In the drop down menu, click on Edit Equipment-Packs. Create a new equipment pack for the player. In the Available Equipment list scroll down until you see “WPN-AK47” and add it to the Used Equipment list.



Click OK. Go back to Mission and this time select Properties. In the new window, under the Main tab, click on the button underneath where it says “Player-Equipment-Pack”. Now select the equipment pack you made earlier with the AK47 and click OK. Then click OK again in the Mission Properties window. Now just press Control+G and have fun with your new weapon! Congratulations on your new mod and enjoy.



# General FAQ

## Modeling

**Can I use Meshsmooth to make my model look better in Farcry?**

**I myself do use Meshsmooth. However, you need to know how to apply it correctly because if not the model will have too many polys, making it unusable for any game. For example, I used Meshsmooth many times on the AK47 yet I was able to keep the poly count below 3,000 polys.**

**How do I make the charging handle and magazine part of the AK47 mesh like you say to do early in the tutorial?**

**Simply select the AK47 mesh and then click the attach button and select anything you wish to attach to the main mesh.**

**I'm trying to use Meshsmooth but it deforms my mesh in weird ways, how are you supposed to control it?**

**It works by smoothing groups. Polygons that have the same smoothing group ID will be smoothed, alternatively, polygons with different smoothing ID's will not be smoothed. Also remember that polygons that share vertices will be smoothed if they have the same smoothing ID.**

## Skinning

**Can I apply multiple textures to one model?**

**Yes.**

**How big can textures on weapons be?**

**I used a texture that was 1024x1024 pixels for the AK47. You can experiment with different sizes but try not to go too high.**

## Rigging

**Some of the finger bones whose envelopes I adjusted aren't working correctly, why?**

**Try changing the strength settings of the envelope to 100.**

## Animating

**How do I animate a reload?**

**Try practicing the technique I showed in this tutorial for animating the firing animation. Use the technique to animate more complex animations. You may also want to look at animation tutorials on the internet.**

**My animation will not export correctly, why?**

**Remember to turn the option to “Ignore Dummies” to off in the Cryexporter2.**

**The muzzle flash appears “clipped” in Sandbox, why?**

**You have placed the “spitfire” bone too close to the tip of the barrel, try repositioning it.**

## Scripting

**Sandbox crashes giving me a strange error about not being able to load a script or something, why?**

**You have made a mistake in one of the scripts you edited. Go back and make sure you find and correct the line of code which is wrong. Also, you should make a backup of each script you edit so you can revert back to the correct scripts and narrow down the faulty script when you troubleshoot.**

**My weapon doesn't appear in the list of available weapons in Sandbox, why?**

**You made a mistake in one of the scripts. The culprit could most probably be the “ClassRegistry.lua”, the “WeaponsParams.lua”, or the “WeaponSystem.lua”. Go back and carefully review every line of code you added or altered.**

## Miscellaneous

**When I try to create bones in Max, I simply don't have that option available, why?**

**Check to make sure you have Character Studio with your Max installation. If not, then you will not be able to animate I'm afraid.**

Thank you for reading my tutorial. I will expand the FAQ section of this tutorial as I get suggestions and more questions to add to it. If you wish to have a question addressed in the FAQ section, please visit the [www.crymod.com](http://www.crymod.com) forums and post any suggestions there. Please limit your concerns to anything concerning weapons modding only. Lastly, I would like to thank Chavez and L337\_Haxs for helping me when I was learning to mod, thanks guys. Goodluck and happy modding!