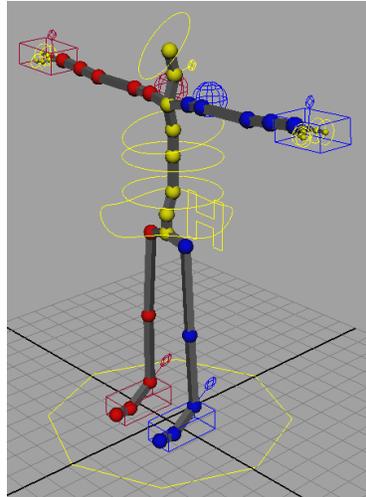


eRig Automated Rigging Tools

Version 1.1

12/17/10



Overview

eRig is a Maya tool for the automated generation of character rigs. To build a rig, you place various parts (see *Parts tab*) and connect them together to make a skeleton. You then position the joints (see *Positioning Joints*), press the build button, and out pops a finished control rig. All that's left is to adjust the controllers (see *Changing Controller Sizes*), bind your character (see *Attaching Your Character and Exporting Animations*), and animate. eRig even makes it easy to bake your animation down to a basic skeleton and remove the control rig when you're ready to export, making it simple to integrate with game engines.

If you have any questions or feedback, please contact me at Elliot@ElliotB.com.

Installation

eRig is known to work with Maya 2010 and 2011. If you are using a version prior to 2011, you must install the PyMel python library (<http://code.google.com/p/pymel/>). This should already be installed on all IMGD lab machines.

To install eRig, put the eRig.py file in your *Documents/maya/*version*/scripts* folder. Then start up Maya. Open the script editor by going to the Window menu and choosing *Window->General Editors->Script Editor*. Select the Python tab and enter the following two lines:

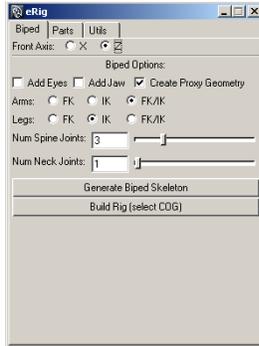
```
import eRig
eRig.openWindow()
```

Finally, hit CTRL+Enter and the eRig window should pop up. If you like, you can select the two lines of code and middle-mouse-drag them to your shelf to make a button for opening eRig.

The Window

The eRig window is divided into 3 tabs. Each of these will be explained below.

Biped tab



The biped tab allows for fast creation of generic biped rigs. These rigs will include at minimum: a spine with a variable number of joints, left and right arms, a neck and head, hips, and left and right legs. The options are described below:

Front Axis – This option tells the rig which way the character is facing (either down the x-axis or down the z-axis). Make sure to set this to the proper direction or the rig may not be created correctly.

Add Eyes/Jaw – These options, when checked, will add a pair of eyes and a jaw as children of the head joint.

Create Proxy Geometry – This option toggles the creation of low-poly proxy geometry when the rig is made. If you have a high-poly character, or if you have no character and just want to animate something, the proxy geometry gives you a simple stand in.

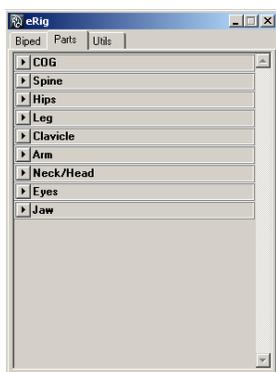
Arms/Legs – These options select the type of controllers that will be created for the arms and legs. They are explained in more detail in the next section (**Parts tab**).

Num Spine/Neck Joints – These allow you to select the number of joints created for the spine and neck.

Generate Biped Skeleton – Generates the base skeleton using the selected options.

Build Rig – Builds a rig based on the selected skeleton. Once you have positioned all of your joints, select the root COG joint and press this button. Note that the original skeleton will be deleted and re-created with the rig when you do this, so don't parent anything important to the joints.

Parts tab



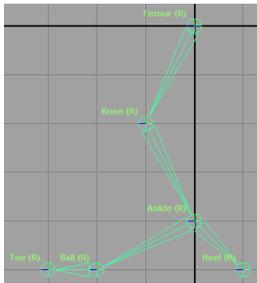
The Parts tab allows you to create individual rig parts to add to your skeleton. Some parts have options that allow you to customize them, largely as described in the **Biped tab** section. To add a given part to the skeleton, select the first joint in the part, SHIFT-select the desired parent joint in the hierarchy, and press “p”. To un-parent a part, select it and press “SHIFT-p”.

COG – The COG (‘center of gravity’) is the root joint of the rig. Every rig made by eRig must have a COG. When adding a COG joint, you can select the front-facing axis along with whether or not to create proxy geometry.

Spine – The spine part creates a user-selected number of joints that all get FK controllers. The last joint (name “Torso_Root”) will not have a controller and is used solely to dictate where the last spine points. If nothing else is attached to the last joint, it will be removed in the final rig.

In addition to being used for normal spine joints, this part can be used when you just need a series of joints (eg. Fingers, tails, etc). Use the side options if you plan to place this part on the right or left of the character. If the Name field is left blank, it will name the part as “Spine”. If you enter something in this field, all joints will use that name instead, and the last joint will be “*name*_tip” instead of “Torso_Root”.

Hips – The hips part is a single joint which is normally placed as the parent of a character’s legs. The controller for the hips is in the shape of an ‘H’.



Leg – The leg part will create a left or right leg, as shown in the picture to the left. The “Ball” joint should go at the start of the toes (the ball of the foot). The “Toe” joint should go at the tip of the toes. The “Heel” joint is only used if IK is enabled for the leg, and should go at the bottom/back of the foot. It defines one of the alternate pivot points for the IK foot.

The Type selection dictates what type of controls will be made for the leg. The choices are FK only, IK only, and both FK and IK along with a control to blend between the two. FK creates simple circle controllers for the Femur, Knee, Ankle, and Ball joints. IK creates an IK chain from the Femur to the Ankle, with a control attribute for directing where the knee should point. IK also creates a reverse-foot-rig, which allows you to pivot the foot about various points and is helpful for walk animations. The IK also allows has an “Pin” control; when activated, the IK controller will stay in place when other parts of the body (hips, COG) are moved. When pin is off, the IK controller will move with the rest of the body.

When FK/IK is used, an extra ‘lollipop’ controller will be created at the ankle to control the blending between the two modes.

Clavicle – The clavicle part creates a single joint with a sphere-shaped controller. It is generally used to provide extra control over its child parts; for instance, it can be placed between the spine and the shoulder to allow the shoulder to move (used for actions like shrugging shoulders). In quadrupeds, a clavicle will often be placed between the hips and the legs.



Arm – The arm part will create a left or right arm, as shown in the picture to the left. The “Hand” joint is simply used for aiming the wrist joint and will be not get a controller in the final rig. A forearm joint will automatically be created halfway between the “Elbow” and “Wrist” joints when the rig is built. This forearm joint is designed to create more realistic skin deformation when the wrist is twisted. The wrist controller in both IK and FK mode will automatically control the forearm joint.

As with the leg, the arm allows you to choose between FK only, IK only, and blended FK/IK switching. The IK for the arm is much simpler than that for the leg; it controls the shoulder and elbow, allowing you to control the rotation of the wrist. As with the leg, you can control the twist of the elbow and the pinning

of the arm from the IK controller. If FK/IK switching is selected, the “lollipop” control will be attached to the hand joint.

Neck/Head – This part creates a neck series of neck joints ending in a head joint. The number of neck joints is user selectable, and works like the spine part. The head joint at the end has a child called “Head_Tip” which is used for aiming and will be removed when the rig is built if nothing else is attached to it. The neck joints get “lollipop” controllers while the head gets a circle.

Eyes – The eye part creates a single joint for either the left or right side. All eyes that are direct children of the same joint (say, the head) are considered to be in a group. When the rig is created, one circle controller is made for each eye-group. The rotation of the controller will control the rotation of the eyes.

Jaw – This part creates a “Jaw” and a “Jaw_Tip”, the latter of which is use only for aiming and should be placed at the tip of the chin (or end of the mouth if there is no chin). The jaw gets a circle controller.

Utils tab



The Utils tab houses utilities to make rigging easier. The main utility is for mirroring joints. Select the root of the chain you wish to mirror and select the axis you want it mirrored across. The newly mirrored joint chain will have the same parent as the original. Multiple joints may be selected at once to mirror them all. If the joints have a non-center side (right or left) and the *change side* option is selected, the side will be switched on all joints in the mirrored chain.

There are two other utilities that are geared toward making skinning easier. They will either ‘hold’ the weights on a selected joint hierarchy, or ‘un-hold’ the weights. See the skinning tutorial video for more details.



Attaching Your Character and Exporting Animations

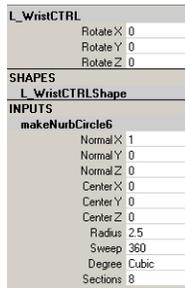
Rigs created with eRig are designed to make animation exporting easy. Open the outliner (**Window->Outliner**) and you should see something like the picture on the left. Your character should be attached (bound or parented) to the bind_COG skeleton, and you should never have to open the RigGroup group.

Since there are many instances where all you want is an animated skeleton without all of the controllers and IK handles, eRig is designed with a dual skeleton system. The controllers control the first skeleton, hidden in the RigGroup group. This skeleton, in turn, controls the bind_COG skeleton, which means you can bake the animation onto the bind_COG and get rid of the control rig entirely. To do this, first select the bind_COG and go to **Edit->Keys->Bake Simulation->Option-Box**. Change the *Hierarchy* option from *Selected* to *Below* and hit the “Bake” button. Once it finishes, your animation has been baked to the bind skeleton. Just select the RigGroup group and delete it and you can export a simple skeletal animation. Just remember to save before doing this so you have a backup of the rigged animation!

Tips and Tricks

Changing Controller Sizes

After the rig is built, you will need to adjust the sizing and positions of the controllers. However, because they are now controlling the joints you cannot just move/rotate/scale them normally. There are two options for adjusting the controllers:

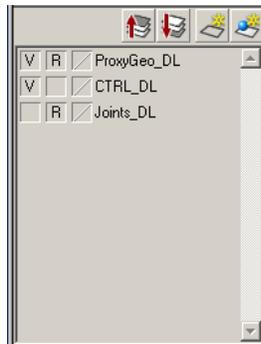


Circles – All of the circle controllers (including the COG and MasterCtrl) have a node in their history called “makeNurbCircle#” which should be listed in the channel box under *Inputs*, as shown on the left. You can adjust these attributes to change the circle controllers. In particular, the radius and sweep options are quite useful.

Other Controllers – Since all of the controllers are just NURBS curves, you can adjust their CVs to change their shape. To select the CVs, right-click on a controller and select “Control Vertex”. With these selected, you can move/rotate/scale as normal.

Changing Joint Sizes

Changing the size of the joints can help you get a better idea of where they are placed. There are two ways to change the joint sizes. First, you can change the base size for all joints by going to **Display->Animation->Joint Size**. The other option is to change the size for individual joints. If you select a joint and look in the channel box, under the normal Translate/Rotate/Scale/Visibility boxes you will see a Radius box. Adjusting that will change the size for just the selected joint.



Adjusting the Proxy-Geometry

Just like described for the circle controllers before, the proxy-geometry has history that you can use to change its size. Also, since the geometry is made of polygon shapes, you can just model them like you would any polygon object. First though, you have to unreference the proxy layer in the layers list in the lower-right (as seen in the picture on the left). To un-reference the layer, just click on the second box until it is empty (the other two states are “T” and “R”). Just remember to set it back to “R” when you have finished so that you don’t accidentally select the geometry.

Using the MasterCTRL

The MasterCTRL is placed at the origin by default and controls everything in the rig. If you move/rotate the MasterCTRL, it will move the COG and will even move pinned IK controllers. This is useful for placing your character in the scene at a starting point.

It also has uses for making certain animations like walk-cycles. In a walk-cycle, you normally want the character to be walking in place at the origin, but it is hard to make the foot-planting look realistic. Instead, you can key the MasterCTRL to move forward, animate the character as though it is walking forward, and finally remove the keys on the MasterCTRL. With the keys gone, the character will once again be walking in place.

Positioning Joints

You can use the move, rotate, and scale tool to position your joints in the scene. However, these tools will also move all of the child joints too. There are two ways to get around this. First, you could un-parent the children, move the joint, and then re-parent the children. The other option is to select the joint, switch the move tool, and hold 'd' while moving the joint. This will adjust the position of that individual joint without moving its children (tip: this can also be used to adjust the pivot of normal geometry objects when modeling).