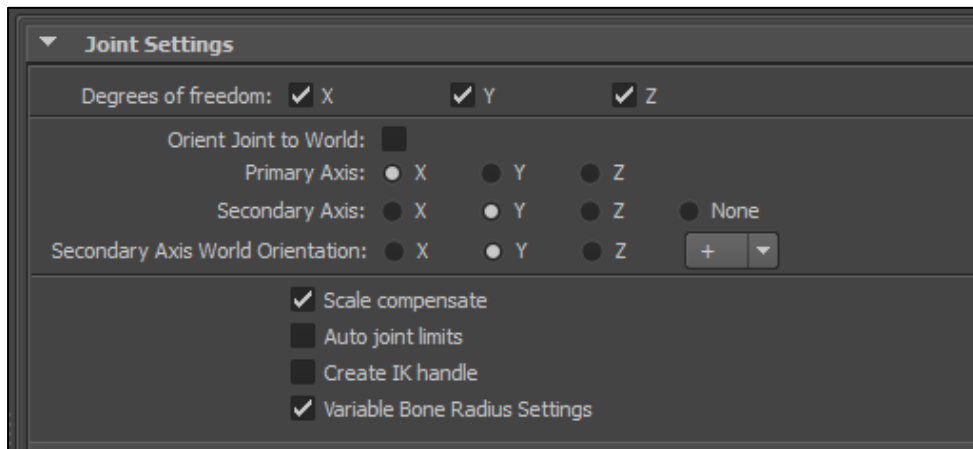


IK ARM RIG SET UP.

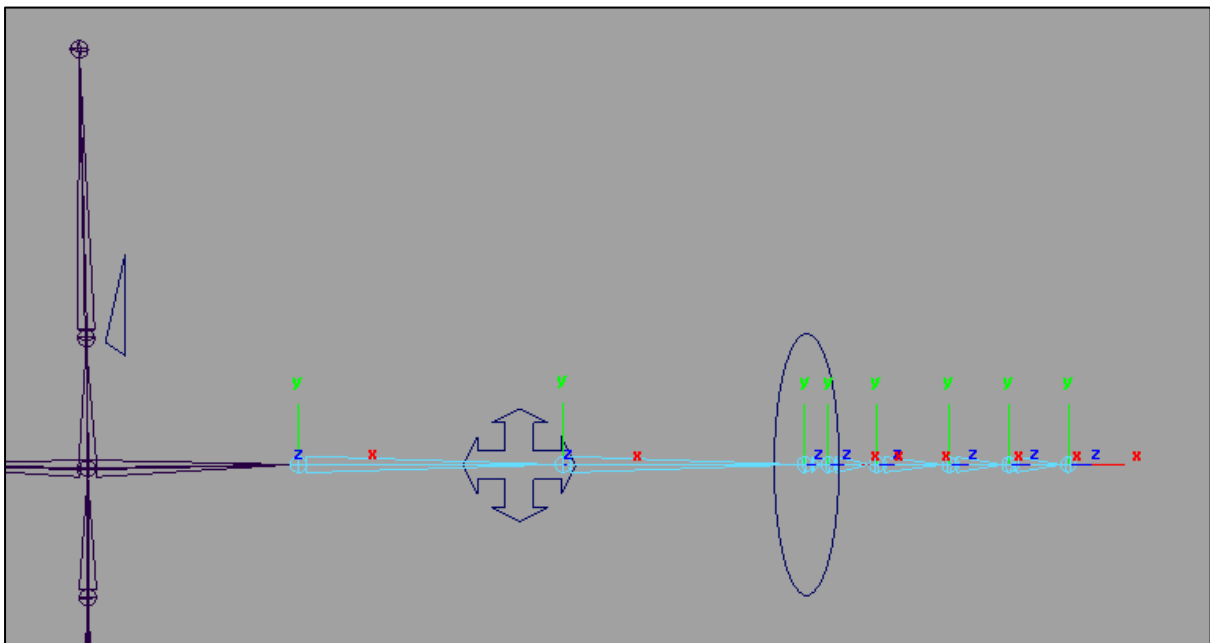
Setting up the joints for the arm joints system.

Joint tool.



Before starting, bring up the options for the skeleton joint tool.

- Reset the tool to be at default settings, this will ensure that the chain of joints created will automatically be oriented to local orientation which means the X-axis of will point down the bone and the Y-axis pointing up.



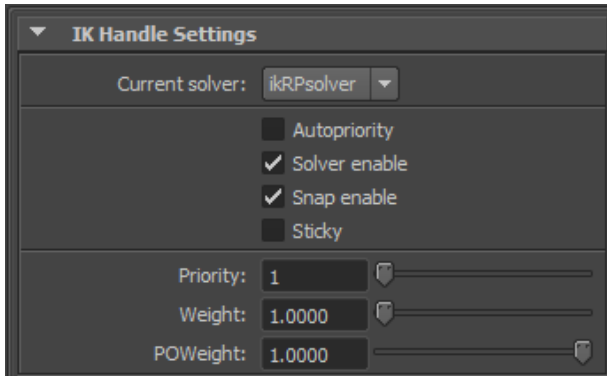
Create the joints for the arm by following the image above.

- l_shoulder_bnd_jnt
- l_elbow_bnd_jnt
- l_armIK_bnd_jnt-> This is the joint to which the ik Handle would attach.
- l_wrist_bnd_jnt
- l_hand_bnd_jnt

(Finger joints can be added for a more advanced arm and hand rig).

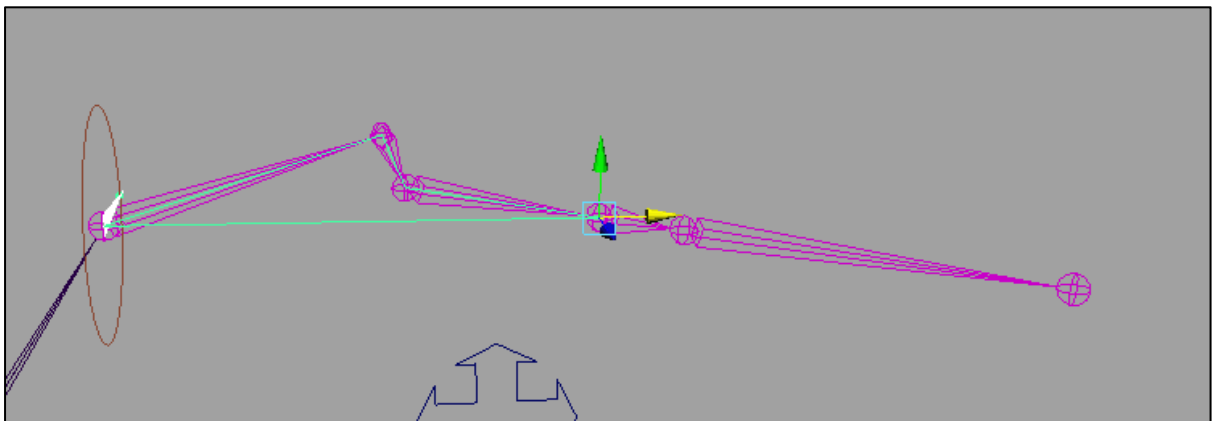
Because this is a rig that needs to rotate just like the leg, im going to use a Rotation plane ikHandle.

Rotation Plane solver:

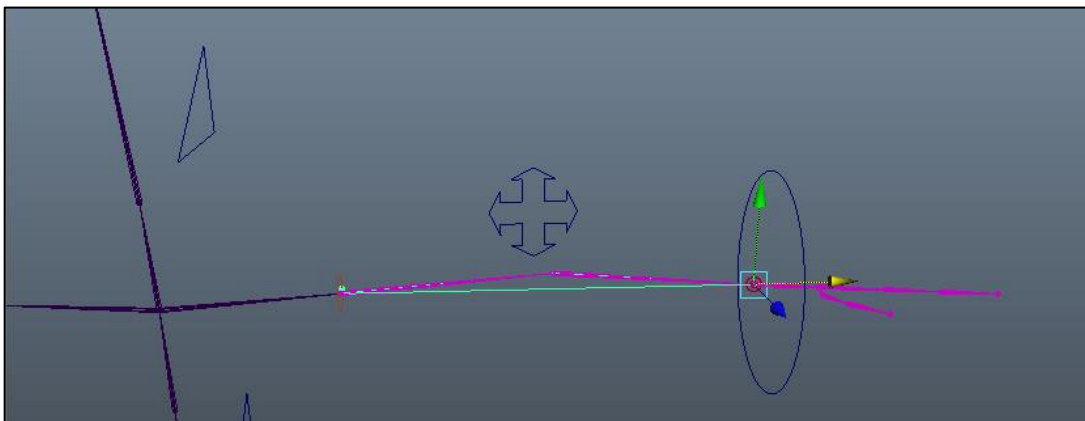


With the tool settings first reset to default for a fresh tool, change the ik solver to be set to ikRPsolver for rigging the arm.

- By skipping the elbow joint, the solver is allowed to make joints behave like a natural bending arm would.



- So use ikRPsolver from shoulder to the armIk joint I created before the wrist for this purpose. At make sure the arm bends right by setting the preferred angle.



Rename the ik handle to I_armIK_handle so I can keep my work and outliner clear.

CONTROLS.

Ik handles essentially need to be able to be manipulated from another level of control for the animator to easily get around the rig and use it with easy selectable control without potentially breaking the rig.

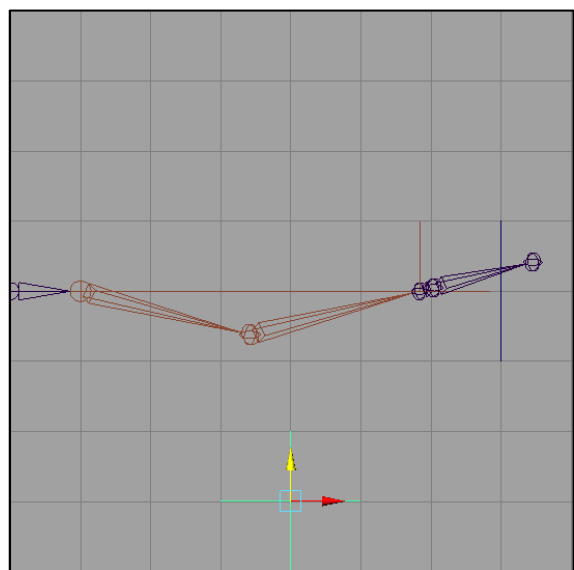
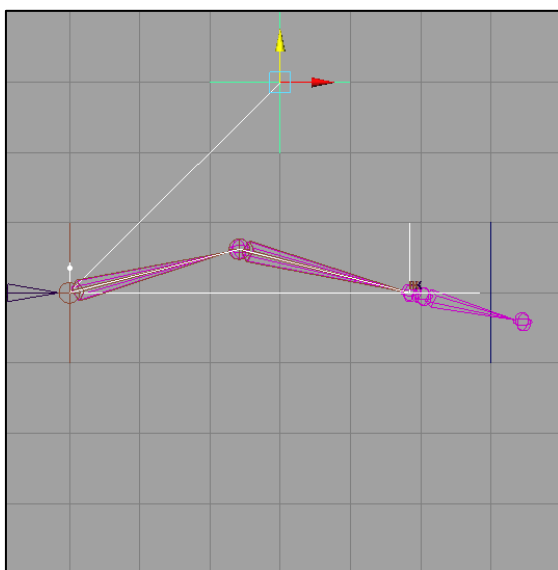
- Controls can be revolved manually from self-constructed cv curve shapes, or in some cases like the ikArm control, it is simply a pre-made nurbs circle, basic but clear.
- It is usually up to the rigger to determine the look of the rig and what shapes are to indicate specific control

For this arm rig any shapes would suffice as this set up is quite a simple rig.

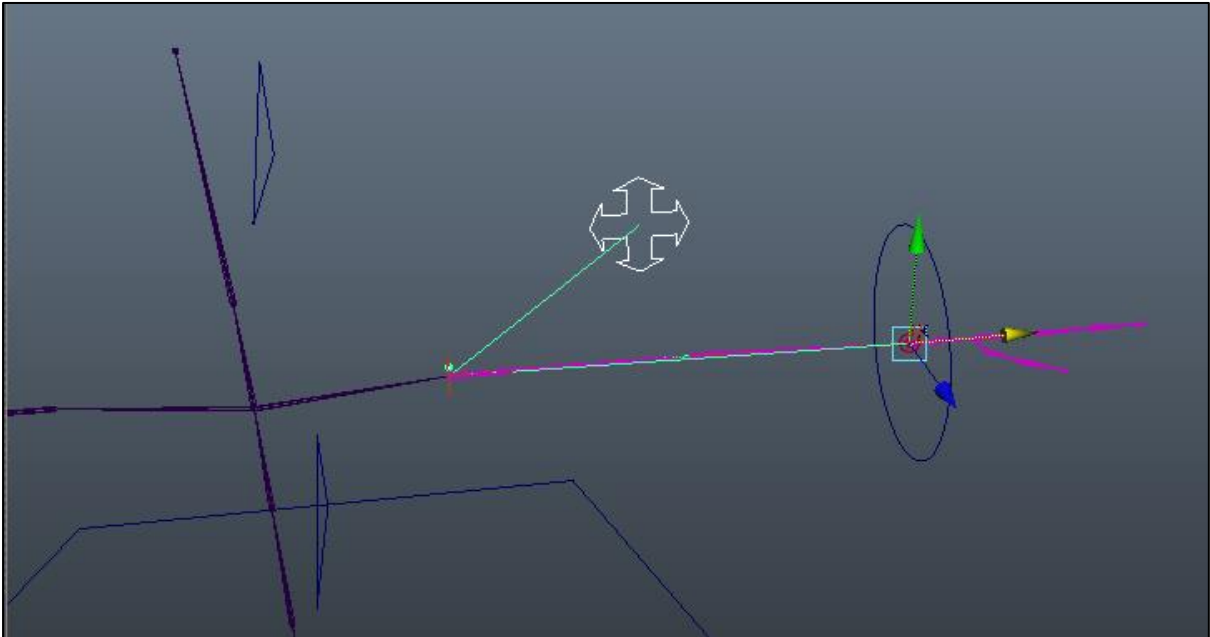
- Create the shapes.
- Snap arm control on top of the armIk joint because this is the point where the ik is created and from where I want to easily manipulate the arm and wrist control for basic character posing.
- After everything is in place, delete the construction history and freeze the transformations so there is no more data left on the controls to cause future rigging problems.
- Rename the controls according to their purpose in what they would be doing.
 - l_armPV_cntrl
 - l_arm_cntrl

Arm elbow Pole Vector Constraint:

Pole vector constraints are used as added control to translates of the elbow joint whenever the armIk control is used to bend the arm.



Select the control (master), then the object to which you want to constraint the control shape to (the slave), and apply command.



I want allow the armPV Control, to control the rotation plane of the ikHandle. Visually and technically it means the elbow would point at the control in the local x rotation.

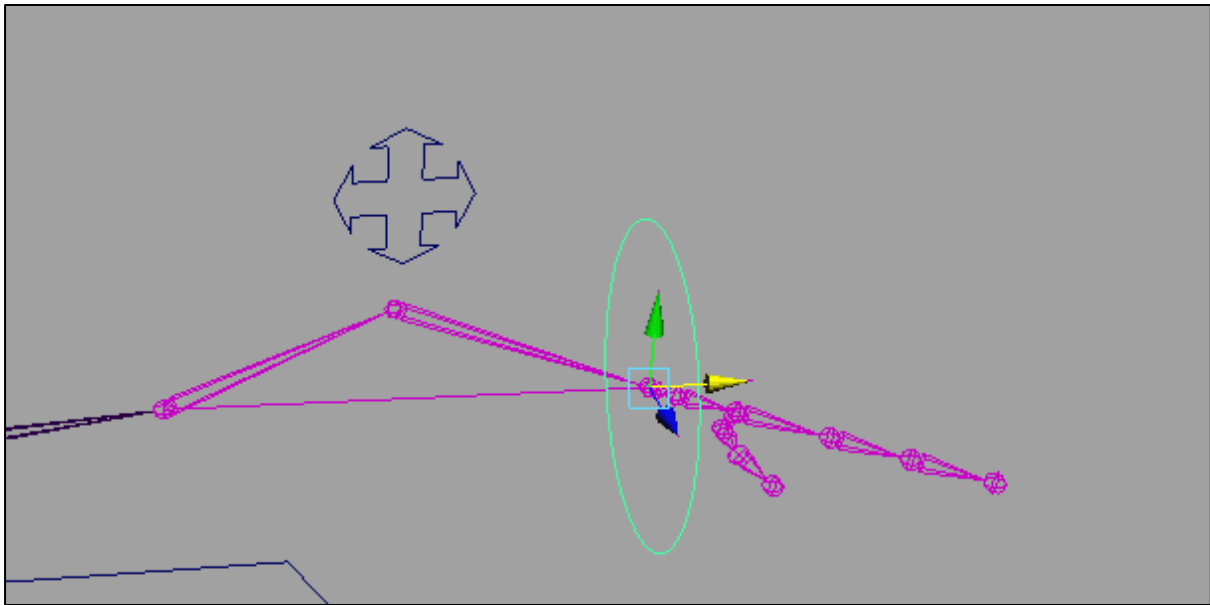
- Select Elbow Control (master of the ik rotation plane)
- Shift select I_armIK (slave of the control)
- Apply Pole Vector Constraint.

Arm Control with a Point Constraint:

A parent constraint is to be used in this case for the arm control because I am interested to in making the control to be in charge of both rotates and translates of the ikHandle.

- Select I_arm_cntrl
- Shift select I_armIK_Handle
- Apply parent constraint.

There is still no control over the hand, it is stiff and not animator friendly.



Solution:

Single Chain Solver

A solver which is used to create control ik handles from only one joint to the next.

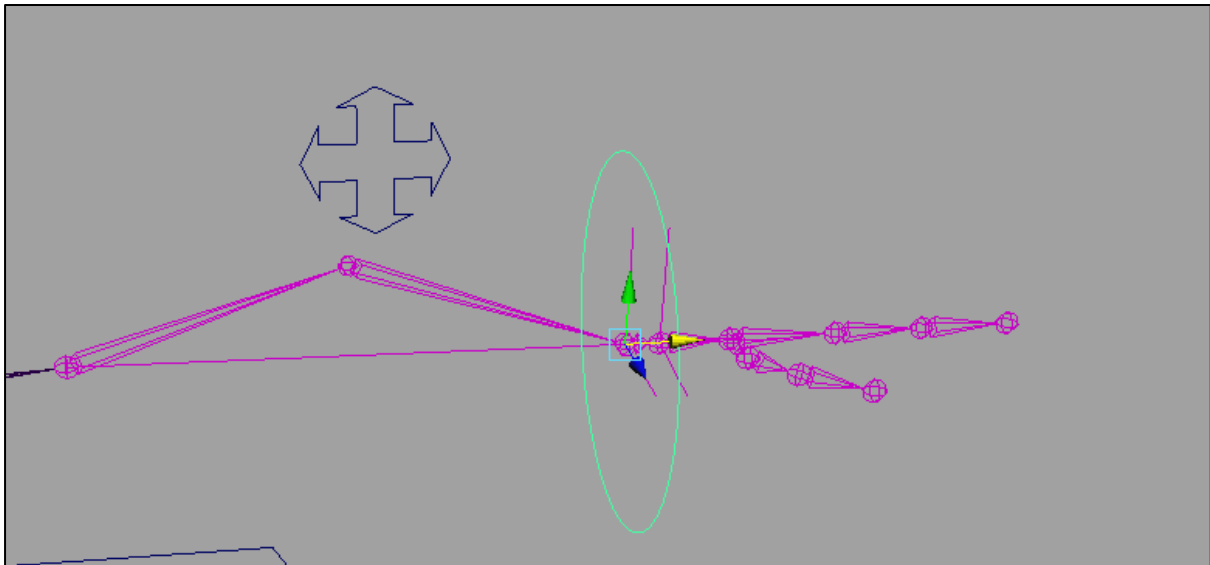
- Use the ik Handle tool with its solver settings changed to SCsolver.
- Create the single chain ik Handle from the l_armIK_bnd_joint to the l_wrist_joint.
- Name ik handle l_wristIK_Handle.

Buffer group (grouping something to itself)

Anything new created in Maya is in world space (an empty group node, a null node)

- It is used to protect the information of the object inside that group. So whatever I do to that group, connecting it to other objects, the object will remain unaffected.
 - Group the two IK Handles together which would then act as the buffer group. Name it l_armIK_grp.
 - Now we want this “pocket universe” to rotate from the correct point of rotation (l_armIK_jnt).
 - By default the centre pivot of the group is at 0 on the grid as the default location for anything new that is created.
 - Point-snap the pivot of the l_armIK_grp to l_armIK_bnd_jnt.

The control is the master of the l_armIK_grp

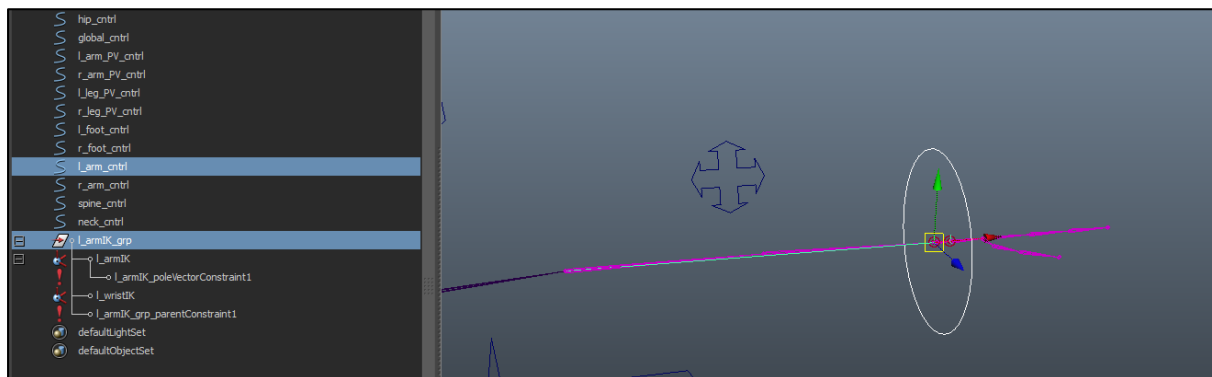


Parent Constraint

- A command which Constraints rotation and translation as the parent controls the child to the buffer group.
 - Select arm Control as the master
 - Shift select buffer group containing the ik Handles
 - Apply parent constraint.

Refer to above image to see the results of these steps.

Complete IK arm Rig Set up:



Cleaning up after is very critical. Always make sure objects are named properly, history is cleaned and freeze transforms when applicable.

To finish off, lock and hide unnecessary attributes for a clean finalized character arm rig.