

FK SPINE RIG SETUP:

The proper functionality of the rest of the character rig parts are solely dependent on the correctness of the spine rig.

The spine rig is literally the “Spine” of the character. Meaning that the spine is what holds everything together and therefore it is extra crucial to understand the joints, local and world orientation, types of constraints and above all, understanding the difference between Inverse Kinematic and Forward Kinematic.

IK vs FK:

FK – hierarchy (blended constraints).

Can shift keys to break joints, making one part of limb to move first then the next part to move later. One thing moving after another up the chain

IK – its name Implies inverse kinematics

- Subvert hierarchy.
- IK gives sticking point (for walking typically the only kind of set up you need for legs.)
- Can't do breaking of joints with IK.

If you are creating a character walk cycle you are going to need an FK setup for the arms so you can break the joints (after key poses), spread out the animation and offset keys.

Constraint Theory:

Straight parent (Translate and Rotate): select the child then the parent; apply - parent will drive the child will follow. In multiples selections, select all the children first then the parent as the last selection then apply the straight parent command.

Parent constraint (Translate and Rotate): Select the master first then the slave and the master will make the slave follow. In rotation, the slave will follow the world orientation in, relation it will stain in line with the rotation and pivot around the world of the master.

Point constraint (Translate): master first then the slave – in translates the slave will follow where the master goes.

Orient constraint (Rotate): master first then the slave – the slave will rotate as the master rotates but it will stay at the same pivot and rotate on its original spot.

Joints and orientations.

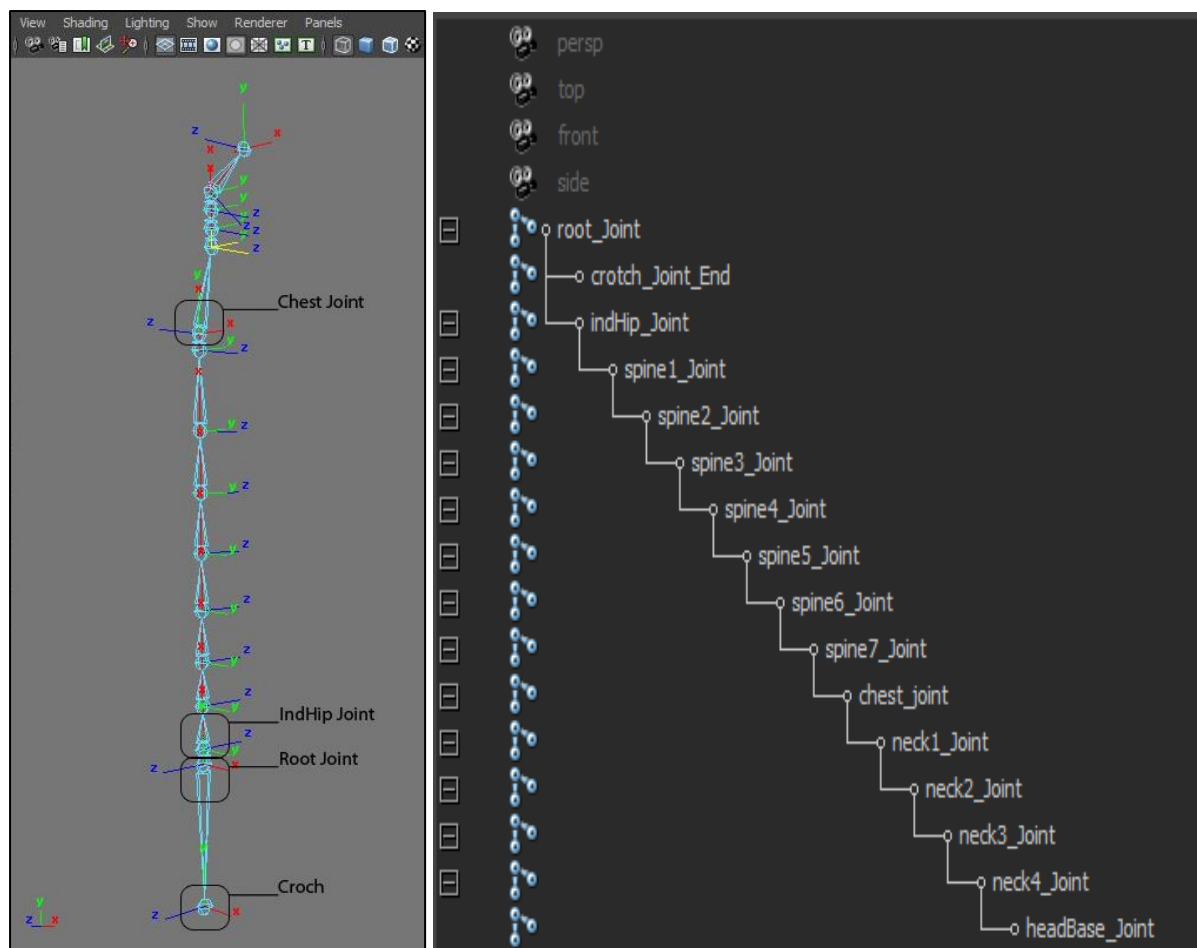
The Root joint is for deformation purposes and the hip joint is so we can wiggle the hips independently from the rest of the rig system.

- If a joint has multiple children, we call it the core joint.

When setting up a spine joints system, it is important to know, always:

- The root joint and chest joint is oriented to world because they need to communicate with the “outside world” orientations. These two joints are also the parents for the leg and the arm to attach to.
- The rest of the joints are by default built to behave in their own local orientation.

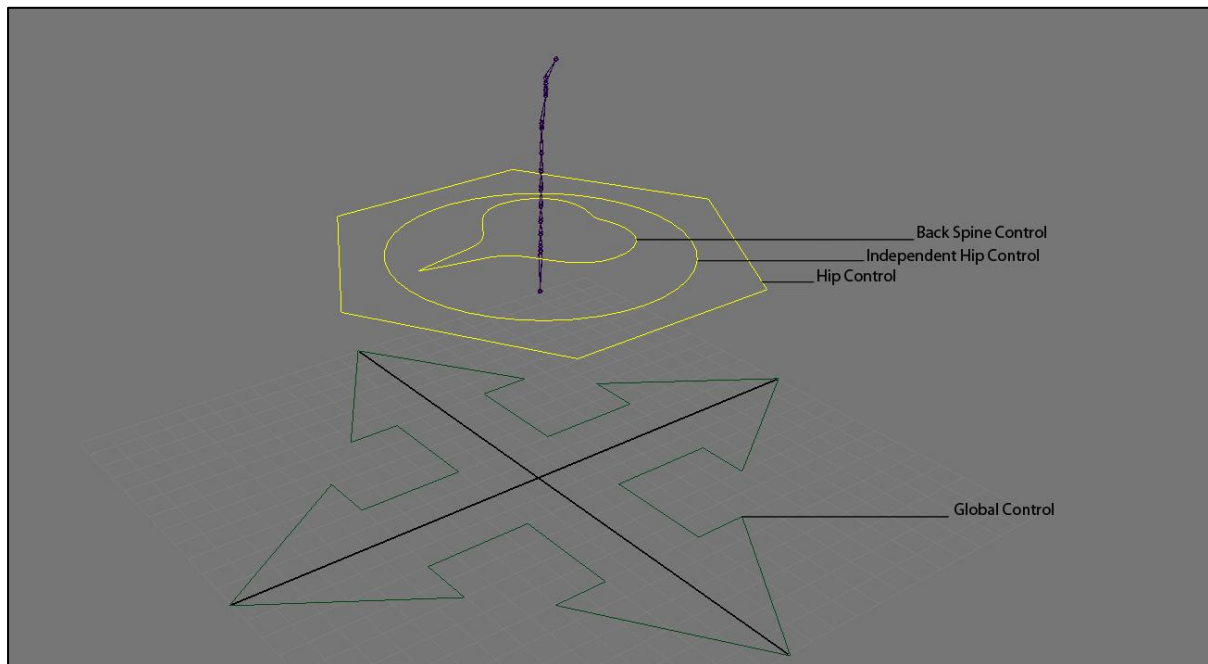
Setting up the spine joints:



Create the spine joints and name them accordance, by following the above image.

- ❖ The crotch (tail bone) is created separately and parented as a child to the root joint. The root joint is the ultimate parent joint of the whole character rig body.
- ❖ Because this spine rig is FK, hierarchy based, I am not dependent on any kind of IK handles or solvers.
- ❖ This rig will be solved by Control curves, constraints and hierarchy parenting.

Setting up Controls for the spine.



- **Create control curve shapes:**
 - ◆ Snap Back Spine Control to Spine1_joint.
 - ◆ Snap Independent Hip Control to indHip joint.
 - ◆ Snap Hip Control to Hip joint.
 - ◆ Global control remains at default construction location; 00.00.00 on grid.
 - Freeze Transformations and Delete History on controls for clean inputs.

Next I need to think about the Joints orientations compared to the controls I just created which by default anything created new would be set in world orientation.

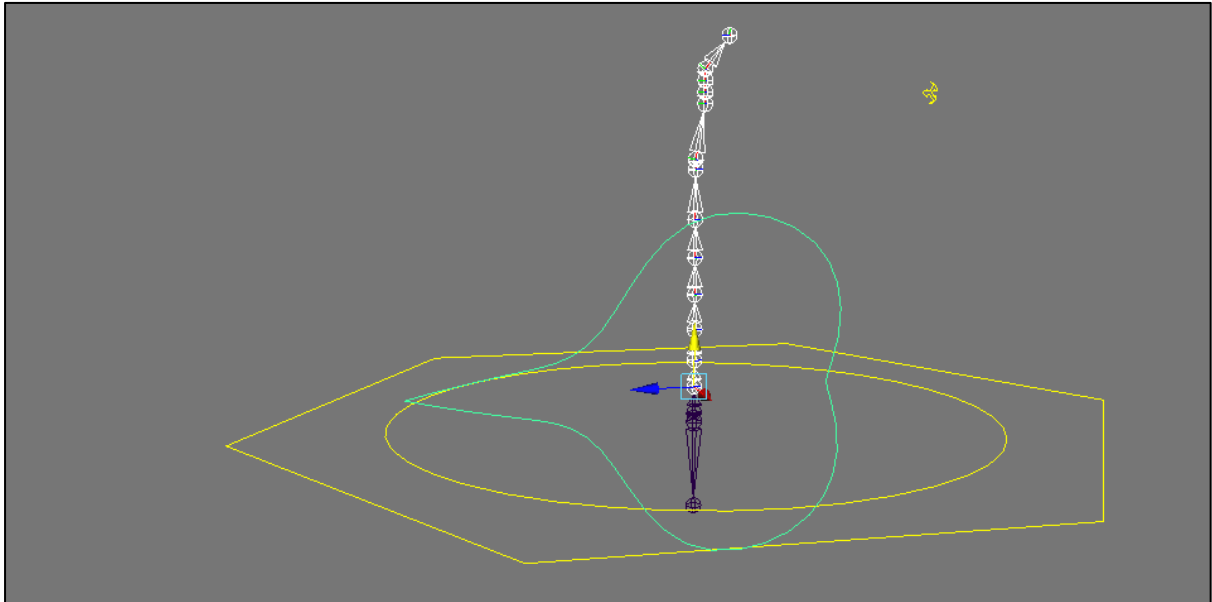
I need to find a way to make the controls constraint with the joints on local orientations without needing to use the maintain offset setting in the constraints menu option settings. The less I need to depend on such automatic “fix for me” settings, the more I can understand by my own knowledge how to trouble shoot and solve a rigging issues.

This will also provide me with a larger self comprehension of the rig im creating, allowing me to know why something worked, why something didn't work and understanding how it worked when I applied a different solution than an automatic button that did everything for me.

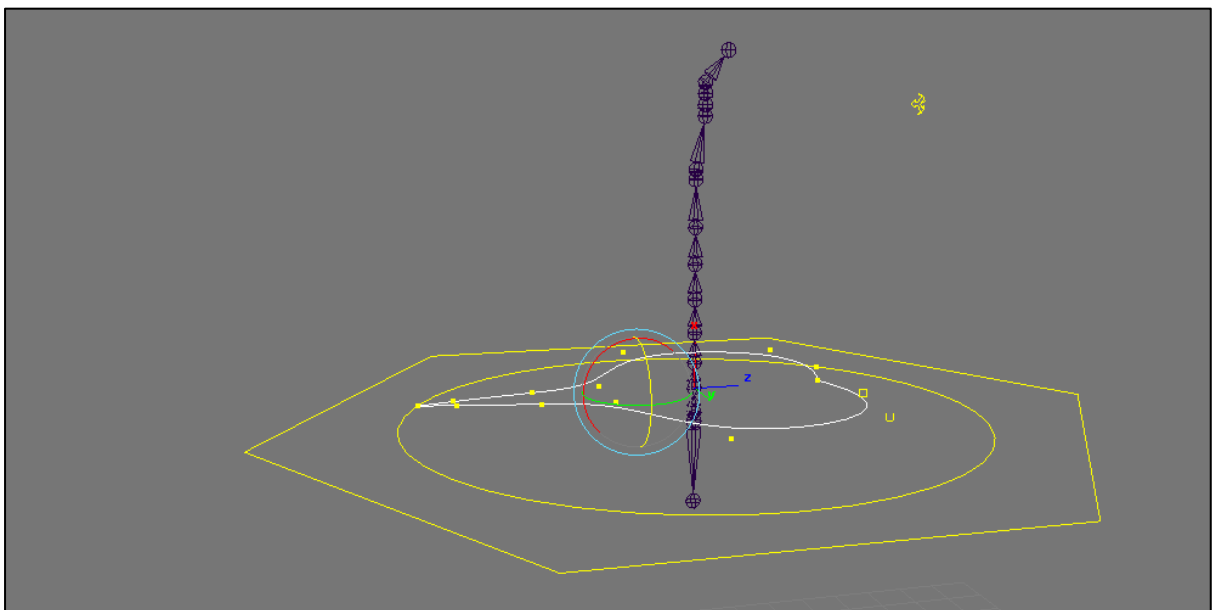
Back Spine Control Curves:

Group the Back Control curve to itself, select spineJoint1 and shift select the new buffer group in the outliner. Apply a parent constraint with maintain offset off, because we want the buffer group to snap into place, forcing the control to align with the joints local orientation.

(Despite appearance, the curve itself has no idea that its translates actually changed, so all its inputs are still at zero value).



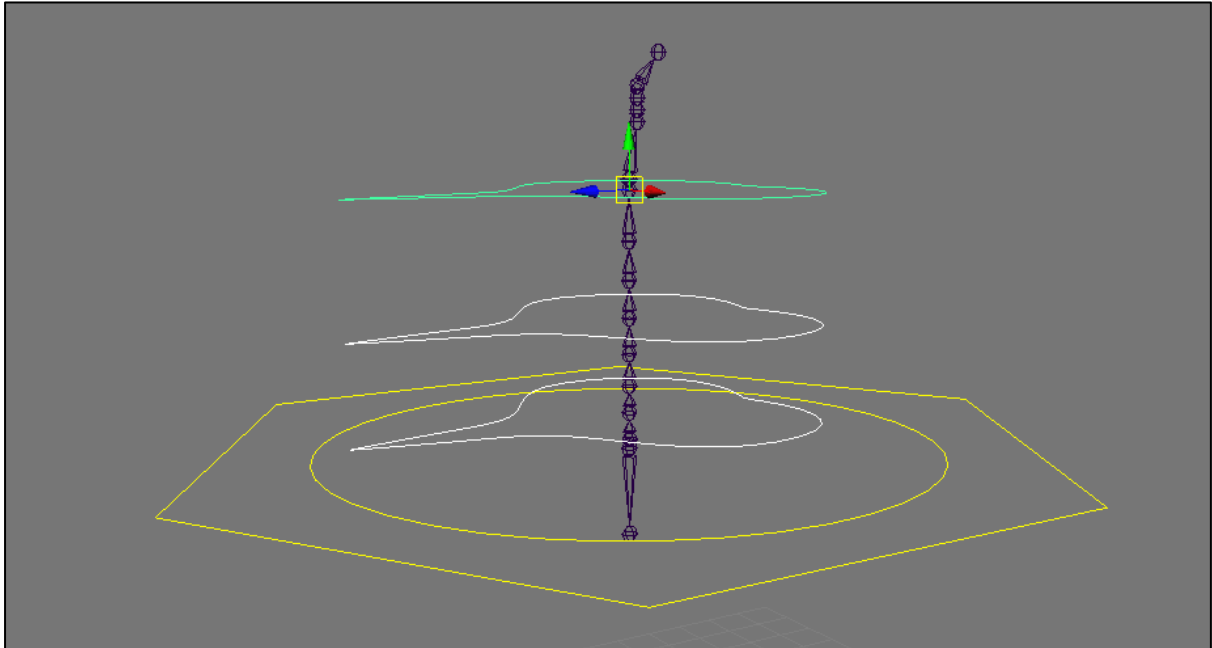
In component mode I can now select all the vertices and rotate them to fit the curve around the spine chain (all its inputs still unaffected) now the curve is the same on local orientation with the spine so no later nasty flips will occur)



Delete constraint because now it has finished with doing its job so I don't need it anymore.

Make sure the duplicate curve in the buffer is the correct name which should be backControl_1

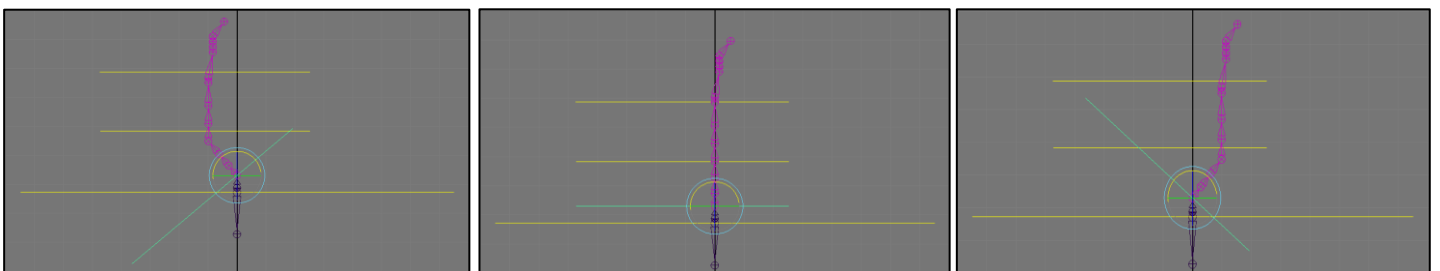
Duplicate the curves again and snap backControl_2 to joint 4 and backControl_3 to joint 7
(so we have 2 joints open in between).



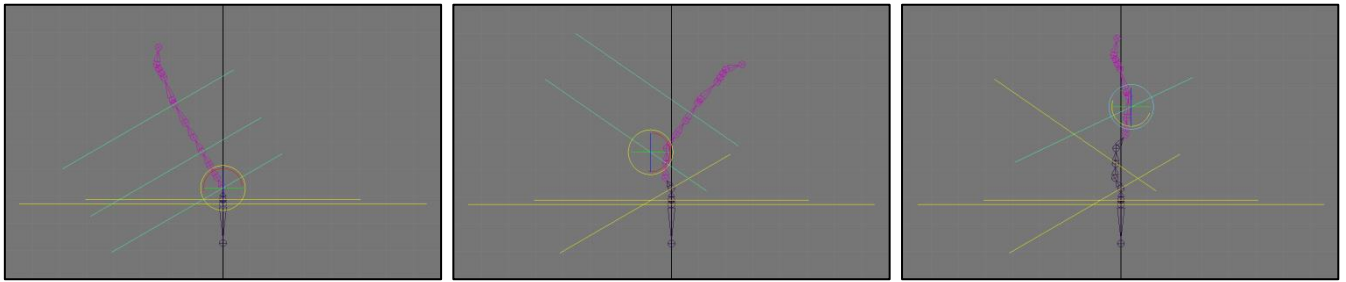
We are now going to apply an orient constraint for the backControl_1 to the spineJoint_1
 Because it's a constraint: (Master first (curve) then slave (joint)).

- **It needs to be an orient constraint** because I only want to affect the rotation values for this connection and not the translates as the joints need to know their translates location in accordance to world space.
 1. Select Back Control 1, shift select the joint I want it to control, spine2_joint.
 2. Apply Orient Constraint, maintain offset off, because I already ensured for the orientations to match up in order not to get a mathematical offset to occur.
- **Repeat Step 1 and 2 for back Control 2 and 3.**

Controls get left behind when I rotate another curve in the hierarchy so I need to parent the curves to each other in the hierarchy to make them follow in relation with each other when one or the other is used to control the joints.



All I need to do in Outliner is drop Control 3 on to Control 2 and drop 2 into Control 1. (1 will now be in charge of 2 and 3, 2 will above be in charge of 3 and 3 will rotate by itself.



I can now add the weighted blend properties of the joints in relation with the curves.

Orient constraint: (don't need maintain offset)

BC1 to joint 2 and 3

BC2 to joint 2 and 3

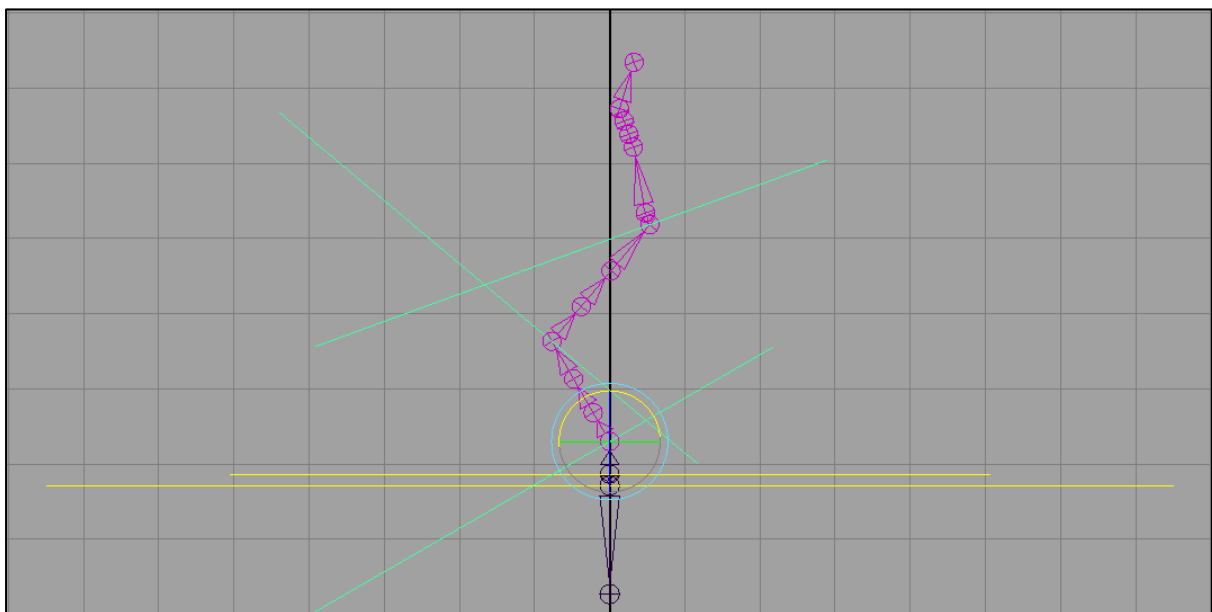
BC2 to joint 5 and 6

BC3 to joint 5 and 6

- The joint closer to a curve will have the most blend weight
- ie: Jnt2 is closer to BC1 and should have a stronger blend value influence.
Jnt3 is closer to BC2 and should have a less influence value from BC1 and a stronger influence value from BC2
- Repeat the same theory for the rest of the joints blend with the backCurve2 and 3.

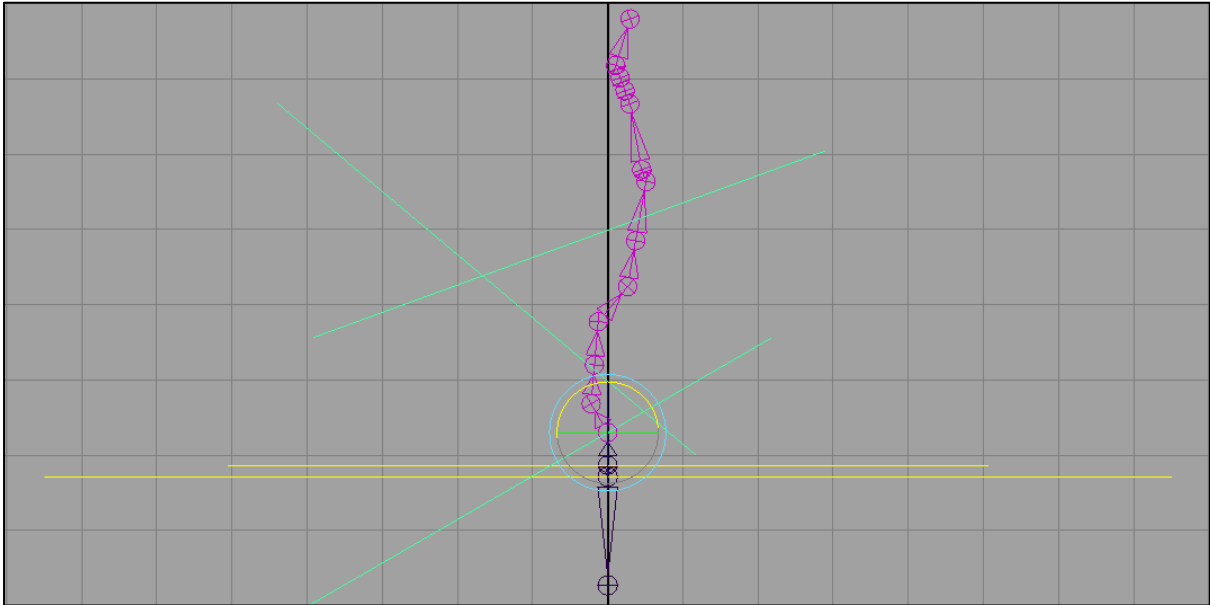
FK SPINE

Controls orient constrained to in between Joints, Weighted Blends: [0; Zero]



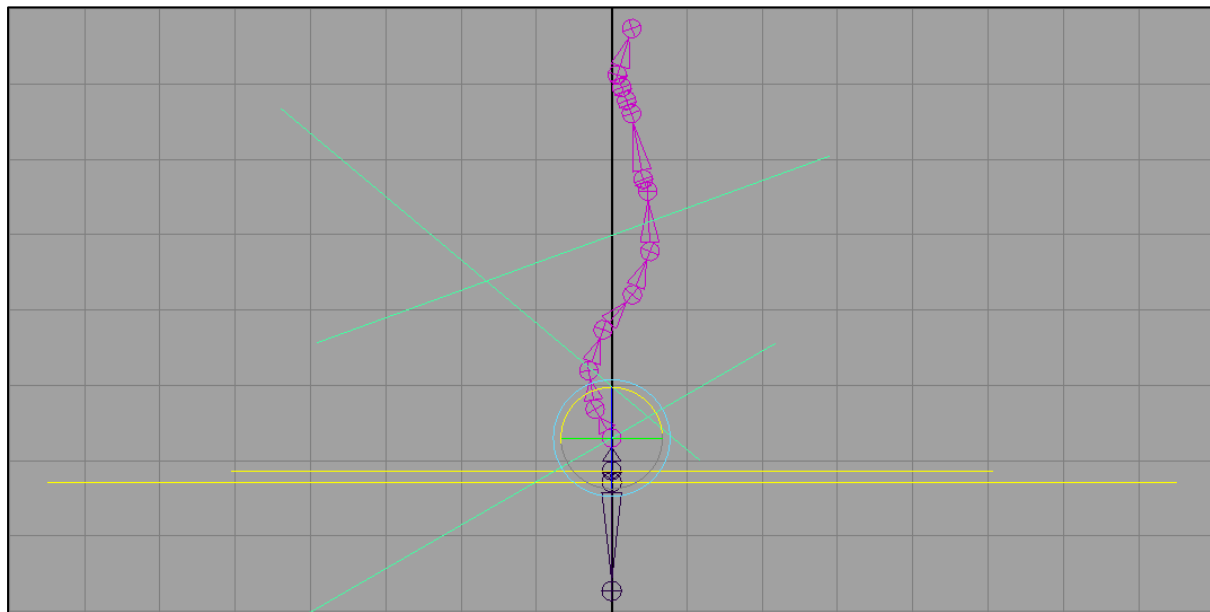
FK SPINE

Controls orient constrained to in between Joints, Weighted Blends: [**1; Default**]



FK SPINE

Controls orient constrained to in between Joints, Weighted Blends: [**Modified**]



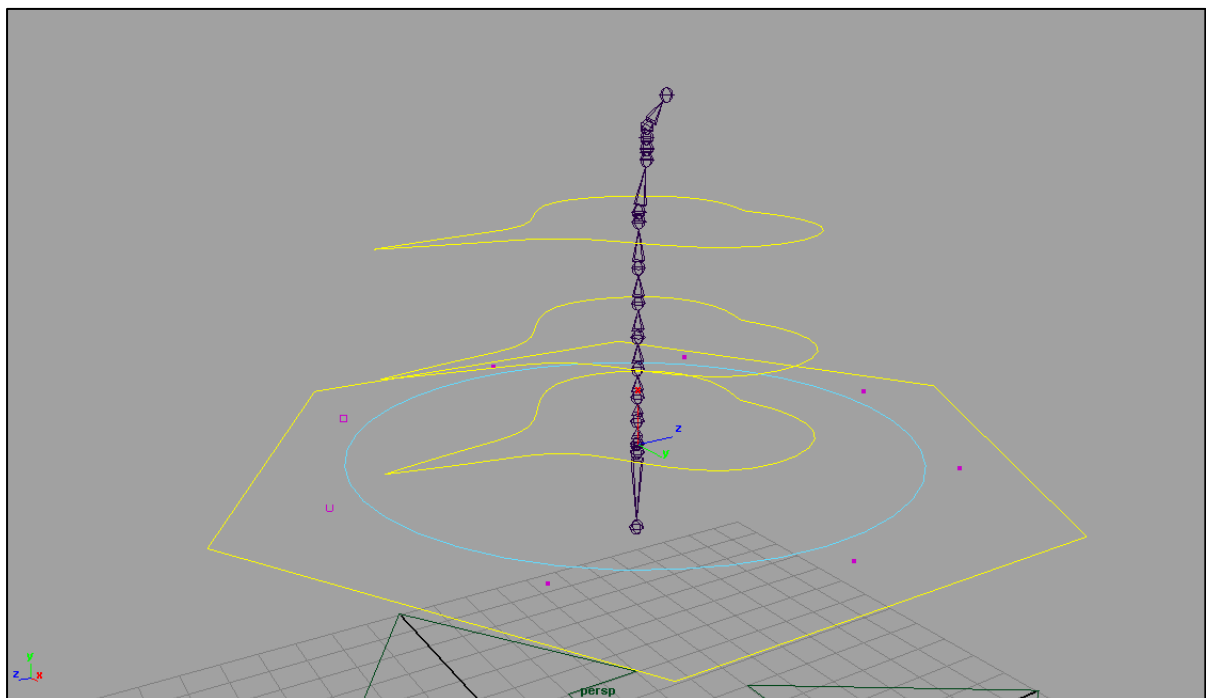
I used these numbers to blend the rotation calculation to control how smooth I wanted the bend in the spine to be when I use the controls to rotate and manipulate the spine:

| | | | |
|----------------|------------------|---|-----|
| Joint 2 | Back Control 1WO | - | 0.7 |
| | Back Control 2WO | - | 0.3 |
| Joint 3 | Back Control 1WO | - | 0.3 |
| | Back Control 2WO | - | 0.7 |
| Joint 5 | Back Control 2WO | - | 0.7 |
| | Back Control 3WO | - | 0.3 |
| Joint 6 | Back Control 2WO | - | 0.3 |
| | Back Control 3WO | - | 0.7 |

Independent Hip Control:

The independent_Hip_Control needs to control the indHip_joint.

- The control curve is still in world orientation and therefore it does not yet match up with joint which is set to local orientation.
- I can't rig the control until I change that. I can do this by simply repeating the same steps I used for the Back Control curves.



Refer to image above to see where the next steps will end up.

- Group the independent Hip Control to itself to create a buffer group.
- Select indHip_joint and then the buffer group; apply parent Constraint.
- Leave maintain offset off to make the control flip in to place on to the joint, the rotation of the control is not correct because of the curve's world orientation.
- But the data is still protected by the buffer group with zero input values, so I can now manually change the control's orientation to match the joint's local orientation and as a result the control should be in the correct rotated position.
- In component mode I can rotate the control curve 90 degrees in Z in appearance while the actual inputs still remain unaffected when I exit component mode.

After completing the steps, delete the constraint which is no longer needed. If I don't do this and forget it by leaving it in there, the constraint will actually cause calculation problems and make the joints flip when I try to constraint the control to the joint regardless of newly matched up local orientations.

- **Complete the independent Hip Control** by Orient constraining it to the indHip_joint.

At this point however, the back Curves in turn are now left behind when I rotate from the new point of the hierarchy, the hip Control.

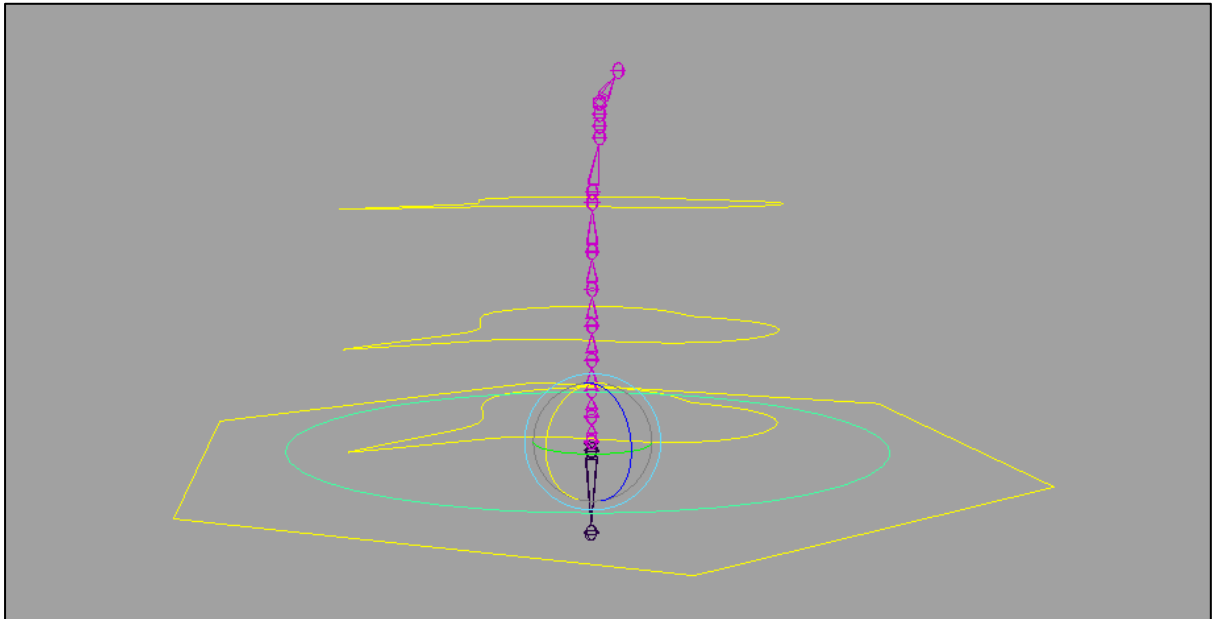
- The joints rotate, the back controls stay behind. Make the curves controlling its assigned joint remain in location with the joint when they are rotated in hierarchy by the hip Control.
 - Select spine2_joint; shift select backControl_1 -> point constraint.
 - Select spine4_joint; shift select backControl_2 -> point constraint.
 - Select spine7_joint; shift select backControl_3 -> point constraint.

FINALIZING HIERARCHY:

- ✚ The hip Control is the main parent for the spine joint system.
- ✚ The root joint always needs to have a world orientation by default because it will have the rest of the character rig set up parts integrated on this joint as a separate step.
- ✚ For this reason the root joint is already reoriented to world orientation at the start so it can match the orientation of the hip Control to manipulate the whole spine (upper body) from its root.

Hip Control:

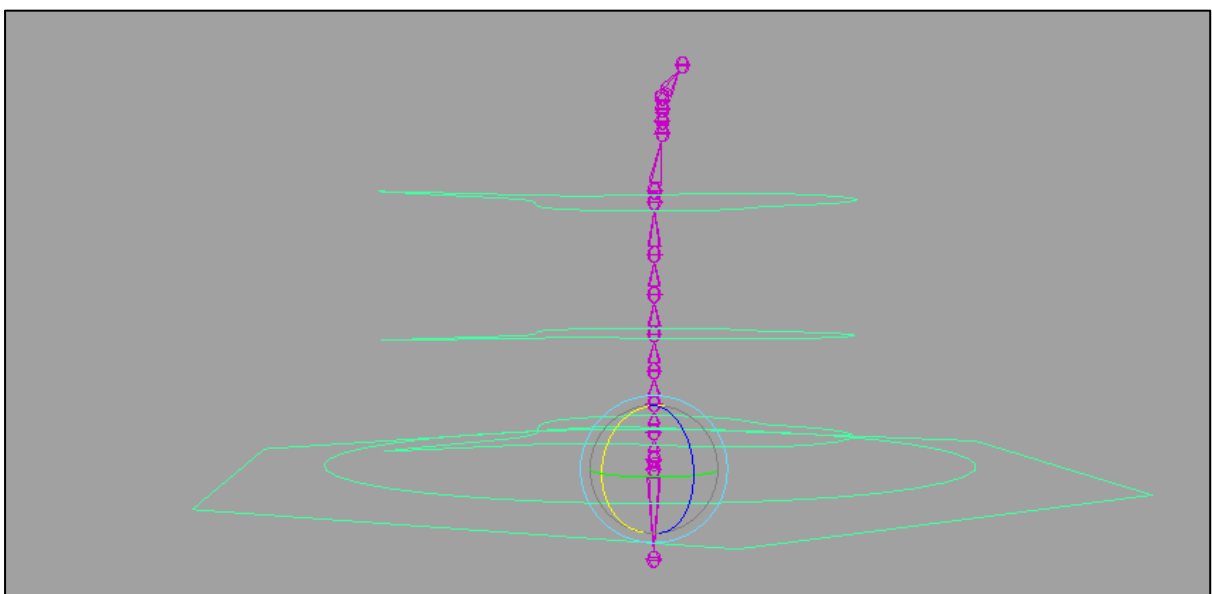
- I want to have the control influence the translate values and the rotations of the spine rig.
- Parent constraint the hip Control to the hip.



Hierarchy:

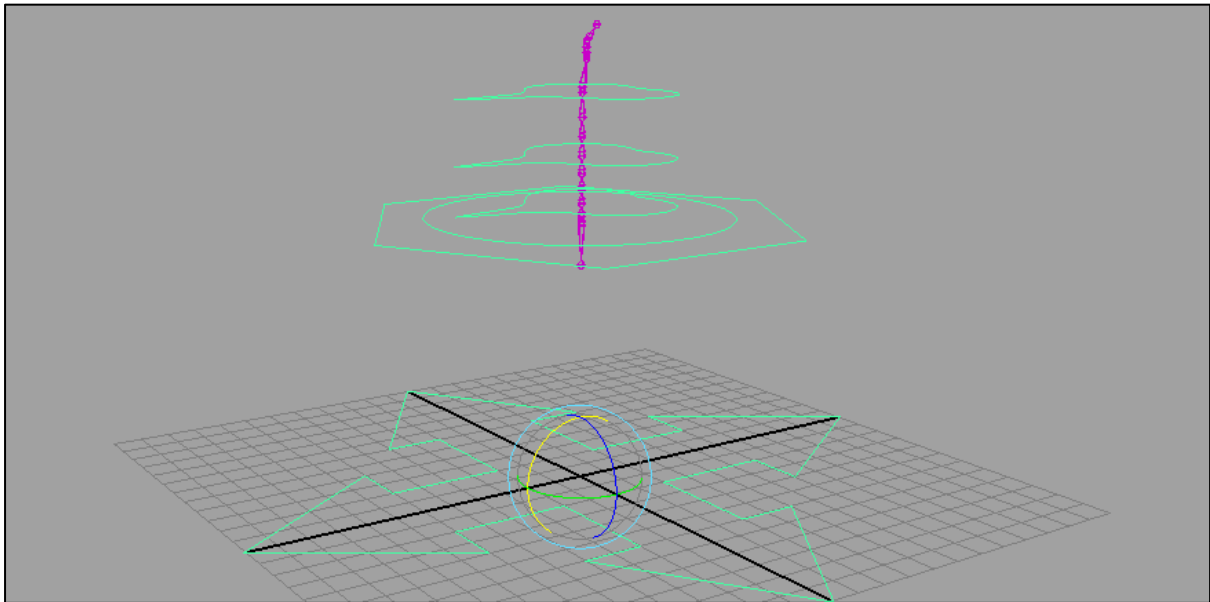
- As the parent of the Back controls, select Back Control 1 and make that curve a child, with its hierarchy, children of the Hip control.
- Not including the independent Hip Control in this selection because I would like it to rotate the joints independently from the rest of the hierarchy.
- Separately by itself, parent the independent Hip control to the Hip control so it is not left behind, allowing for the hip control to be the ultimate parent for the spine from the root due to Hip control's parent constraint to the root joint.

(While still leaving the independent hip Control indented as an individual selection in the hierarchy).



Global Control:

- Make the Hip Control child of the global Control. (With the hip Control's children in the hierarchy already following).
 - Select hip Control, straight parent to global Control.



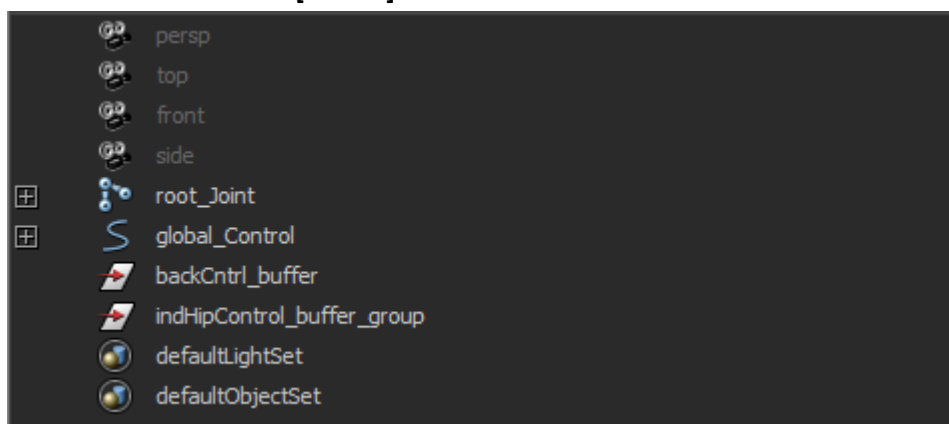
Clean up the Outliner:

As of yet, the global control scale does not work to uniformly scale the whole spine joint system with its controls in proportion.

- Simply now, include the joints in the global Control hierarchy by parenting the joints to the global Control.
- Leave out the control buffer groups because it will break the rig otherwise when scaling and is not necessary in the hierarchy for scale to function properly.

Global control is now the ultimate parent of the whole hierarchy, a complete and clean FK Spine Joint Rig Set Up.

Clean Final Outliner: [Closed]



Clean Final Outliner: [Open]

