

SPINE IK spline / FK

Si inizia la nostra operazione dividendo la nostra mesh del personaggio in diversi punti chiave che vogliamo riggiare. Duplichiamo la mesh e con una selezione delle facce usiamo l'extract per separare le braccia, le gambe e la testa dal torso. Quest'ultimo verrà poi ulteriormente suddiviso in "proxy" tramite il multicut tool.

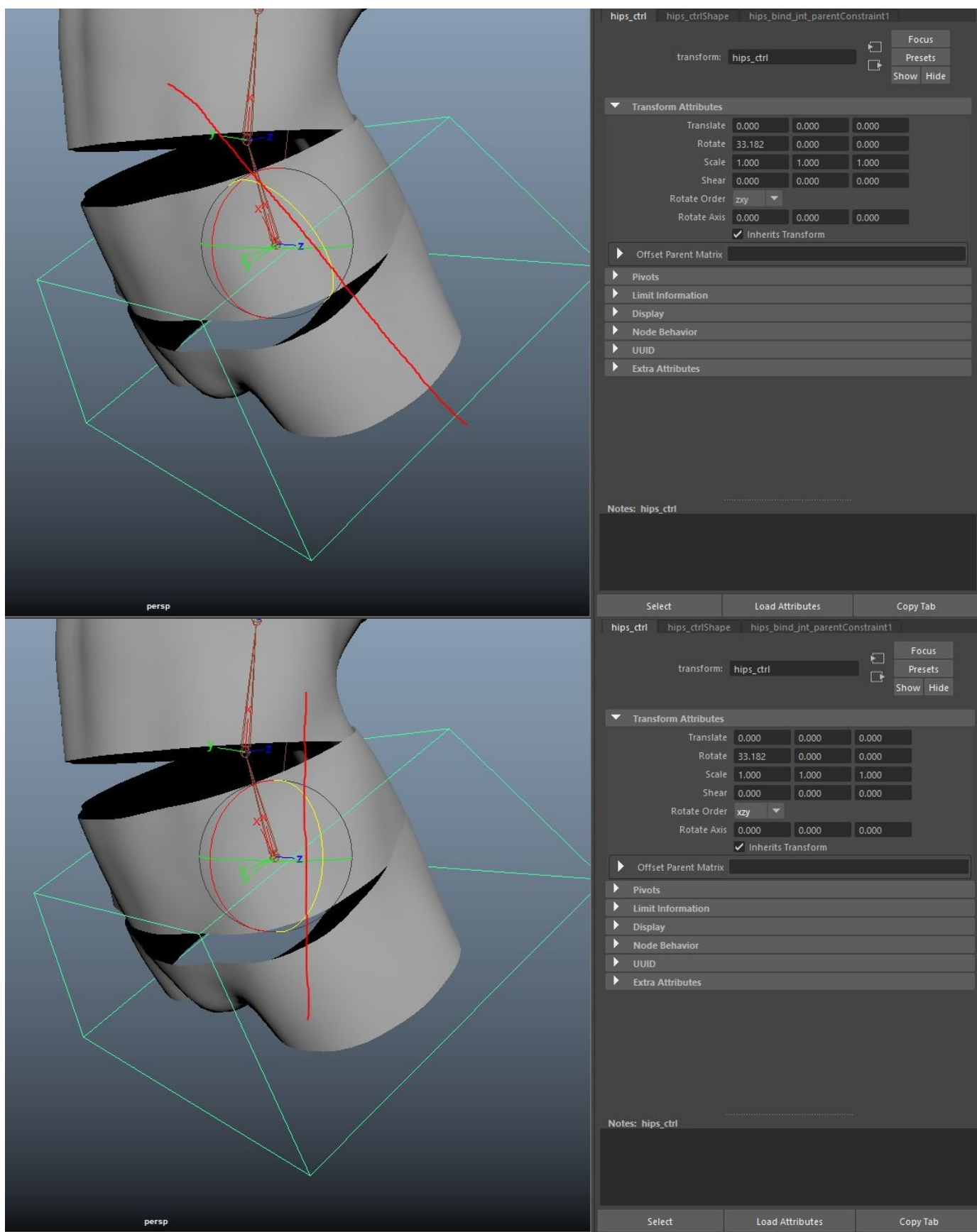
- **Joint Setup:**

- Creo una catena di Joint in side view (6)
- Li posiziono facendoli partire dall'anca fino alle spalle e li rinomino **spine1_jnt** → **spine6_jnt**.
- Se dovessimo ruotare i joint insieme otterremo risultati diversi per osso, o comunque confusionari, quindi gli orientiamo con "Orient Joint Tool". Dalle impostazioni settiamo l'asse primario su X, l'asse secondario su Z e l'asse secondario World Orientation sulla X positiva, in modo che la X punti verso il prossimo joint e che la Y punti in avanti.
- Col multicut tool taglio la mesh duplicata seguendo la posizione dei joint creati. Nelle impostazioni del tool metto la spunta su "extract faces" e setto gli offset a 0 sui tre assi. Separo con l'extract ogni nuova sezione e li rinomino **hip_proxy** / **spine1_proxy** → **spine5_proxy** / **shoulder_proxy**.
- Creo un **IK Spline** trovando lo strumento sotto "Skeleton" → "Create IK Spline Handle" cliccando poi prima su **spine1_jnt** e dopo su **spine6_jnt**.
- Rinomino la curva risultante in **spine_crv**.
- Per collegare la curva ai joint duplico la catena e cancello tutte le ossa tranne la prima e l'ultima e li rinomino in **hipBind_jnt** e **shoulderBind_jnt**.
- Seleziono questi due nuovi joint, poi **spine_crv**, vado sotto "Skin" → "Bind Skin" e dalle impostazioni setto "Bind To" su "Selected Joints" e "Max Influence" su 2 in quanto devo solo skinnare queste due ossa. Se muovo le ossa a giro ora la curva dovrebbe muoversi.
- Trascino col tasto centrale del mouse le mesh proxy dentro i corrispettivi joint, facendo attenzione a mettere **hip_proxy** e **shoulder_proxy** dentro **hipBind_jnt** e **shoulderBind_jnt**.

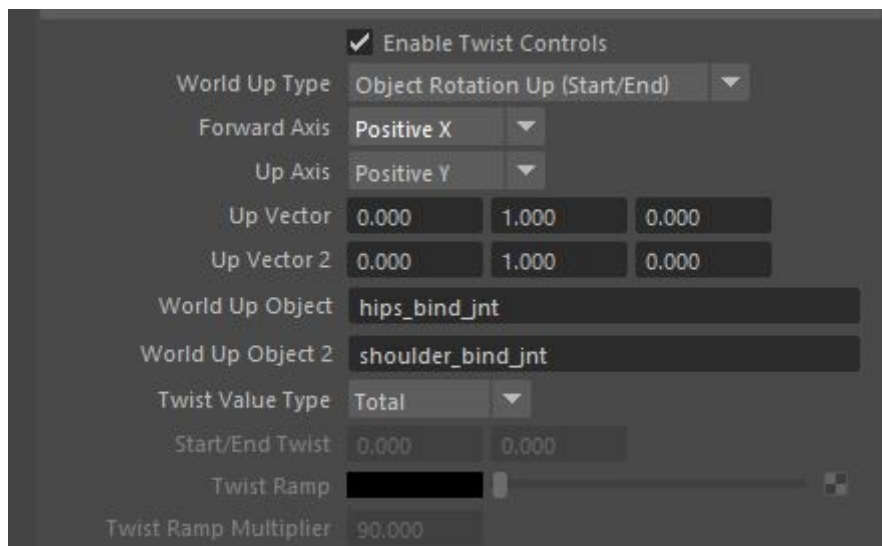
- **IK Controls Setup:**

- Creo un cubo e lo posiziono alla base dei fianchi. Gli do una shape che faccia in modo che i suoi edge con compenetrino con la mesh.
- Con "CV Curve Tools" impostato su "1 Linear" traccio la forma del cubo con lo snap al vertice, completo la curva, e cancello la mesh del cubo.
- Sposto il pivot della curva con D e lo snappo al primo joint della spine. Dopodiché freezo e cancello le storie.
- Duplico la curva per creare il controllo delle spalle, freezo e cancello storie, e rinomino entrambi in **hip_ctrl** e **shoulder_ctrl**.
- Per avere un ordine di rotazione ottimale vado sull'attribute editor di entrambi i controlli e cambio il "Rotate Order" in "ZXY". Questo mi serve per dare a Maya un ordine preciso di

priorità in cui ruotare gli assi e non sempre “XYZ” di default è ottimale. In questo caso la differenza è bassa ma cambiandolo così facciamo in modo che una rotazione su X modifichi l’asse di rotazione su Z per farla coincidere meglio con la mesh.



- Setto lo stesso ordine di rotazione anche sui due bind joint.
- Do un parent constraint per connettere i controlli ai bind joint.
- **IK Handle Twist:**
 - Finora i movimenti e le rotazioni della spina dorsale funzionano bene ma manca la torsione sui controlli. Per attivarla seleziono lo handle della spline sull'outliner, vado sull'attribute editor → "IK Solver Attribute" → "Advanced Twist Controls" e lo attivo.
 - "World Up Type" lo stesso su "Object Rotation Up (Start/End)" e setto come oggetti nei riquadri sotto che si aprono il nome dei controlli o dei joint estremi alla catena (esempio **hipBind_jnt** e **shoulderBind_jnt** come Object1 e Object2).
 - Sulla base degli orientamenti che ho seguito fin'ora queste altre impostazioni dovrebbero essere corrette per impostare bene la torsione:



- **FK (ibrido) setup:**
 - Creo una nuova catena di joint (4) che segua la prima catena e che snappi con V solo sul primo e ultimo joint.
 - Rinomino i joint **hipFK_jnt**, **spine1FK_jnt**, **spine2FK_jnt** e **shoulderFK_jnt**
 - In questo caso voglio orientare i joint nello stesso modo in cui sono orientati i controlli, quindi imposto l'asse primario su Y, il secondario su X e il secondario World Orientation sulle X positive. In questo modo ho la Y in verticale e la Z che punta in avanti.
 - Setto l'orient order di ogni joint su YZX per fare in modo che un piegamento laterale del busto (asse Z) ruoti anche l'asse di rotazione Y per una torsione corretta.
 - Gruppo i controlli IK su se stessi e rinomino in **shoulderFK_grp** e **hipFK_grp**.
 - Do un parent constraint dal joint FK al gruppo corrispondente per fare in modo che la catena FK controlli il resto del busto.
 - Creo due circle NURBS e rinomino in **spine1FK_ctrl** e **spine2FK_ctrl**. Userò questi controlli per muovere la catena FK.

- Sull'outliner clicco su "Display" → "Shapes" → seleziono il nodo shape della curva → ctrl+LMB sul joint **spine1FK_jnt** → scrivo un codice MEL "parent -r -s". Ho imparentato la shape della curva alla funzionalità e posizione del joint.

- Freezo le trasformazioni della curva e, dato che ormai rappresenta la shape del joint, rinomino il suffisso sull'outliner da JNT a CTRL. Infine cancello i gruppi che le due curve si sono lasciati dietro.

- **Squash/Stretch controlli IK:**

Per fare in modo che la mesh "segua" il controllo IK oltre i limiti massimi verticali allora dovrò implementare una forma di Squash e Stretch usando il Node Editor, sotto Windows.

In via teorica ho bisogno di scalare i joint della catena principale tramite un valore che coincide con la percentuale di stretch che la curva della spline sta subendo. Questa percentuale è dettata dalla divisione tra la sua attuale lunghezza e la sua lunghezza standard: la lunghezza di una curva può essere letta e sfruttata tramite il nodo "curveInfo" e la info si chiama "ArcLenght".

- Seleziono la curva della spline e apro il node editor.
- Con TAB digito "curveInfo", creo il nodo e lo rinomino in **spineInfo**.
- Seleziono l'output di **spine_crvShape**, seleziono "World Space [0]", lo trascino sull'input di spineInfo e seleziono "inputCurve". Ora il nodo sull'attribute editor mi farà vedere in tempo reale la lunghezza attuale della curva al nome di "ArcLenght".
- Memorizzo il valore di ArcLenght della curva a riposo.

- Sempre con TAB creo un nodo "MultiplyDivide"
- Collego al suo input 1X l'output ArcLenght di spineInfo e rinomino il nodo **spineStretch_div**.
- Sull'attribute editor setto l'operazione come una divisione, anziché una moltiplicazione e sul secondo valore scrivo manualmente l'ArcLenght della curva a riposo.

- Ora che ho ottenuto il valore di percentuale di stretch della curva devo collegarlo allo scale Y dei joint. Li richiamo nel node editor trascinandoli col tasto centrale del mouse dentro l'editor.

- Collego l'output X dello spineStretch_div all'input ScaleX di ogni joint tranne l'ultimo, poiché scalarlo non ci servirà a nulla (l'asse X è perché è l'asse primario dei joint).

- La mesh si dovrebbe stretchare in verticale ma manca una forma di preservazione del volume che avviene attraverso uno scale negativo sugli altri due assi (Y e Z) e la formula matematica universalmente riconosciuta per lo scopo è $ScaleZ = 1/\sqrt{ScaleX}$ e $ScaleY = 1/\sqrt{ScaleX}$

- Per avere la radice quadra dello ScaleX dovremo collegare l'Output1 di spineStretch_div ad un nuovo nodo MultiplyDivide che rinomineremo **spineVolumePreservation_pow**, nell'input 1X.

- Lo settiamo sull'attribute editor come Power e all'input 2 scriviamo 0,5.

- Colleghiamo l'Output1 ad un nuovo nodo MultiplyDivide che rinomineremo **spineInvert_div**, che uso per invertire dato valore in quanto lo scale da eseguire è negativo.
- Stavolta mi collego su input 2X e manualmente setto input 1 sull'attribute editor a 1 e l'operazione come divisione.

- L'output X infine lo collego sullo scale Y e Z di ogni joint. Il risultato finale vedrà 3 nodi multiplyDivide e ogni input di scale dei joint occupato dai risultati.

- **Controlli Globali:**

- Creo un quadrato tramite curve NURBS che rinomino **root_ctrl**, posiziono sui fianchi, freezo, cancello storie e posiziono il pivot correttamente. Setto l'ordine di rotazioni in ZXY.
- Gruppo tutto tranne quest'ultima curva, rinomino in **torso_grp** e li connetto insieme tramite parent constraint dove la curva controlla il gruppo.
- La curva spine subirà doppie trasformazioni e per evitare ciò vado sull'attribute editor e deseleziono "Inherits Transforms"

- Gruppo insieme la prima catena di joint (**spine1_jnt**), lo handle della spine, la curva dello spine, **shoulderBind_jnt** e **hipsBind_jnt**, e rinomino il gruppo **mech_grp**. Tutto ciò per far ordine nell'outliner.

- Proseguo nel nascondere e bloccare ("Lock and Hide") gli attributi che non dovrebbero essere toccati.

- Gruppo root_ctrl e torso_grp e rinomino in **cog_grp**. Se dovessi scalare questo gruppo la curva spine si rompe perché sul node editor crede che io la stia deformando. Per evitare ciò devo normalizzare lo scale del gruppo con l'ArcLenght della curva.

- Riapro il node editor e inserisco cog_grp col tasto centrale del mouse.

- Creo un nuovo MultiplyDivide e rinomino in **globalScaleNormalize_div**.

- Su questo nodo collego sull'input 1X l'ArcLenght dello spineInfo, e sull'input 2X lo scale Y di cog_grp. Il risultante output X va nell'input 1X di spineStretch_div, sostituendo così la vecchia connessione.

- Il node editor dovrebbe avere un aspetto finale simile a questo:

