

IL RIG

Cos'è il rig? Con questa parola ci si riferisce ad un insieme di ossa, chiamate **Joint**, connessi tra di loro, che simulano appunto uno scheletro che guiderà nelle fasi successive la mesh, dentro la quale viene costruito, per farci delle animazioni.

Quando si ha un personaggio e lo si vuole animare infatti, è impossibile farlo deformando manualmente la mesh tra un frame e un altro. Anche un'animazione semplice richiederebbe troppo tempo per essere completata.

Ecco perché abbiamo bisogno di elementi più definiti che muoveranno la mesh per noi. I joint infatti ci daranno una gerarchia nell'outline molto ben definita e intuibile, in modo che se vogliamo muovere un arto, o una parte specifica del corpo, sappiamo a quale joint fare riferimento.

I joint però non funzionano nella stessa maniera di un altro oggetto normale utilizzabile in scena. Un altro elemento utile per fare un paragone è il **Locator**. Il locator è semplicemente un punto nello spazio e le sue trasformazioni seguono le stesse regole di qualsiasi altra mesh: può essere spostato e ruotato normalmente nello spazio.

Anche il joint è definibile come un punto nello spazio, ma la sua particolarità sta nel fatto che può essere creato in serie, creando una catena di joint che saranno figli l'un dell'altro. Inoltre il loro orientamento sarà inizialmente rivolto verso il loro figlio, cosa molto utile per avere un controllo naturale durante le loro rotazioni.

Altra particolarità molto importante da ricordare sta nel fatto che un joint non sarà mai freezabile sul translate. Questi attributi per loro hanno un significato diverso: definiscono il loro offset relativo alla loro origine. Il primo osso di una catena avrà come offset la distanza da 0,0,0 del world (il centro della scena in sostanza), mentre dal secondo joint in poi ci si riferirà alla distanza del padre.

Quest'ultima cosa rende necessario uno step in più per creare un rig completo e funzionale. Durante il processo d'animazione infatti un animatore vuole poter settare tutti gli attributi di un osso a 0 per tornare alla posa iniziale quando serve, ma, come si può intuire, ciò non è possibile in quanto le trasformazioni dei joint non possono essere modificate in quanto cambierebbe il loro offset l'un dall'altro.

Per ovviare a questo problema, dopo la costruzione dello scheletro intero, andremo a collegare ogni joint utile ad un **Controllo**, ovvero una curva NURBS il cui scopo sarà semplicemente quello di dare informazioni di trasformazioni dirette ai joint, oltre ad un buon modo di visualizzare cosa andranno a muovere in quanto sono editabili esteticamente per rappresentare quel che si desidera.

CONSTRAINTS

Prima di entrare nel dettaglio di tutte queste cose però dobbiamo capire bene la base di come si collega il tutto. I controlli infatti dovranno essere collegati ai joint in qualche maniera, in modo che le trasformazioni dei primi vengano rispecchiate da quelle dei secondi.

La forma più basilare di collegamento di due oggetti è l'**imparentamento**, ovvero l'atto di definire un "**padre**" ed uno o più "**figli**". Questo avviene selezionando prima quello che si prevede essere il figlio, poi il padre, e poi premendo **P**.

Sull'Outline noteremo che il figlio si sarà spostato "sotto" a livello di gerarchia al padre, e, grazie a questa nuova disposizione, potremo notare che il figlio sarà libero di muoversi da solo, mentre

muovendo il padre muoveremo anche il figlio nella stessa maniera.

Per annullare l'imparentamento basterà cliccare **Maiusc+P** avendo i due oggetti selezionati.

Quando muoveremo un padre ed un figlio noteremo, soprattutto nel caso delle rotazioni, che il figlio si muove seguendo il padre, ma lo farà usando il pivot del padre.

La cosa potrebbe essere utile in certe occasioni ma se vogliamo fare un esempio con dei controlli e dei joint, potremo essere in una situazione in cui un joint controlla un polso e il suo rispettivo controllo si trova posizionato sopra la mano, quindi decentrato dal polso stesso. In questo caso vogliamo fare in modo che il pivot del controllo sia equivalente al pivot del joint e per modificarne la posizione useremo il tasto **Insert** che ci farà entrare in una modalità di edit del pivot. Quando avremo posizionato il pivot al joint potremo ripremere Insert per uscire da tale modalità.

Ora potremo ruotare il controllo (padre) intorno al joint (figlio) che ruoterà a sua volta nello stesso modo

Modificare il pivot dei controlli e dei loro gruppi sarà essenziale in fase di rigging in quanto serviranno esattamente a questo scopo: loro detteranno sempre le trasformazioni dei joint, ma per fare riferimento precisamente a loro sarà necessario averne i pivot ben posizionati sui joint.

L'imparentamento appena descritto è un metodo diretto molto veloce e semplice, ma presenta un problema non ignorabile.

Un buon rig può definirsi tale solo se la gerarchia delle ossa è separata da quella dei controlli in modo che un animatore non si perda nell'outline a cercare un elemento specifico in un mare di controlli e joint mischiati tra loro. Per evitare questo abbiamo bisogno di imparentare in maniera "indiretta" controlli e joint, e per fare questo entrano in gioco diverse funzionalità tra cui i **Constraints**.

I constraints, come si può intuire dal nome, servono a legare alcuni comportamenti di un oggetto ad un altro. Questi si possono trovare solo aprendo la tab di Rigging in alto a sinistra, e appariranno nel menù a tendina Constrain.

All'interno troveremo diverse opzioni ma ci concentreremo sulle prime cinque in quanto la sesta, il pole vector, è strettamente legato ad un argomento futuro.

-Parent: il Parent Constraint imparenta due oggetti sulle loro trasformazioni e rotazioni. E' importante fare attenzione sull'utilizzo dell'opzione "Maintain Offset" in quanto determina se l'oggetto figlio verrà spostato sul pivot del padre o no. Con maintain offset infatti si intende se si vuol mantenere la distanza tra i due o no e, se si parla di un rig complesso, di solito si tiene sempre le ossa in posizione. Se vogliamo però creare un controllo distante dall'osso è buona pratica muovere il pivot del controllo e snapparlo sull'osso, togliendo dopo il maintain offset durante l'imparentamento.

-Point & Orient: due impostazioni simili tra loro che semplicemente funzionano analogamente al parent, solo che si limitano sul movimento o sulle rotazioni. Il maintain offset su queste impostazioni si riferirà quindi solo agli attributi ai quali faranno riferimento.

-Scale: lo Scale Constraint fa l'unica cosa che il parent non può fare, ovvero controllare lo scale dell'oggetto a cui è collegato. E' usato principalmente quando abbiamo bisogno di funzioni molto specifiche per il rig, come ad esempio uno squash and stretch.

-Aim: l'Aim è utile per fare quello che dice, ovvero far guardare un oggetto ad un altro ed è particolarmente indicato per i controlli degli occhi. Tra le impostazioni si deve definire un asse

che indica in “avanti” (ovvero verso l’oggetto che guarda) e un asse che indica l’alto rispetto al world.

DIMOSTRAZIONE

Per una dimostrazione creeremo una macchinina con elementi utili per spiegare i constraint e i controlli senza andare ancora a toccare i joint.

Prima modella la macchina usando le booleane per creare l’inserzione delle ruote e le ruote stesse. Dal momento che creeremo controlli con le NURBS facciamo bene a non usare constraints tra pezzo di mesh, anzi, li imparenteremo in maniera precisa e pulita sull’outline con P.

Fatta la mesh, freezata e messa per bene in ordine si procede a fare i controlli: per la precisione uno main appena sotto la macchina e uno per ruota, piazzati davanti alle ruote stesse. I controlli saranno a loro volta freezati e messi in ordine nell’outline.

Dopodiché useremo il orient constraint tra i controlli e le ruote limitando gli assi su X o Z, dipende da com’è girata la macchina, e non su tutti e tre gli assi. Possiamo anche bloccare e nascondere gli attributi che non ci servono dei controlli con “Lock and Hide selected” nel channel box.

Il main control dev’essere quindi collegato con la macchina con un parent e, siccome non ci sono particolari indicazioni da notare, si può lasciare le impostazioni di default. Ovviamente dando per scontato che i pivot siano stati gestiti bene, sennò può essere eliminato il maintain offset

Infine per aggiungere un elemento si può creare una torretta con una canna sopra la macchina. I due elementi saranno separati ma imparentati l’un l’altro (la canna figlia della base con P) e il pivot della canna sarà alla sua base. Inoltre entrambi saranno a loro volta figli della mesh principale.

Creeremo poi un locator chiamato “aimCTRL_LOC” e, dopo averlo freezato davanti alla canna, lo imparenteremo con un aim constraint alla canna. Anche qui le impostazioni vanno bene di default, però si noti del fatto che la canna flipperà quando il locator la scavalcherà in alto per via delle impostazioni dell’”up vector”, ma in questo caso possiamo non preoccuparcene.

Il locator lo terremo fuori dal gruppo dei controlli e lo possiamo usare per far mirare alla torretta qualcosa nel frattempo che la macchina può muoversi liberamente