

# Alla ricerca di Blenemo

Preso da manie di grandezza, e non trattenuto da alcuna modestia, ho deciso di mostrarvi come, con non troppi passaggi, ma con molta, moltissima, pazienza, è possibile dare vita ad un piccolo personaggio virtuale animato, sulla falsariga del nostro pesce pagliaccio preferito, quel Nemo che i maestri della Pixar hanno saputo davvero dotare di un'anima.

Quello che sto per presentarvi non è un tutorial come ci si potrebbe aspettare di solito, non mi metterò a descrivere ogni singola azione, tasto premuto o tecnica adoperata, tutte queste cose sono presupposti che il lettore deve già avere, avendo letto l'ottimo manuale di Blender.

**P.S.** Alla fine ho avuto un po' pietà di voi ed ho variato leggermente l'impostazione, adesso troverete non pochi riferimenti al programma, ai tasti, alle schermate, alle azioni, ma si può ugualmente affermare che non ci troviamo di fronte ad un tutorial classico.

Prendete questo lavoro come una serie di suggerimenti applicati ad un semplice progetto, la creazione di una breve animazione di un personaggio *modellato*, *texturizzato* e *skinnato* (ossia dotato di un'armatura di ossa per l'animazione scheletale) da noi.

Il processo non è semplice, mi ha preso molto tempo, perché essendo stata la prima volta che mi impegnavo su un progetto più serio del solito toroide con le riflessioni ho dovuto affrontare molti problemi di cui non mi ero mai preoccupato prima.

Questa breve guida vuole essere soprattutto questo, una visione globale sugli aspetti da tenere in conto, e sul pensiero da seguire, quando si ha in mente un risultato. Spero che anche l'appassionato non praticante di grafica fatta al computer possa trovare in questa una serie di risposte sulle tecniche base adoperate ma soprattutto sulla dedizione, la pazienza e la concentrazione che un grafico che ama quello che fa investe quotidianamente nelle sue creature.

## Modelli di riferimento

Dopo questo scoccante preambolo, che però mi sentivo di farvi, arriviamo alla prima meta pratica.

Ok, abbiamo deciso che, ispirati dalla tenerezza del piccolo Nemo, vogliamo modellare un piccolo pesce pagliaccio.

Mmm, voi sapete com'è fatto un pesce pagliaccio?

No, non sto dicendo com'è fatto più o meno, qual è più o meno la forma delle branchie o vagamente di che colore è... intendo dire: ma sapete DAVVERO come è fatto un pesce pagliaccio tanto da poterlo riprodurre fedelmente ad occhi chiusi?

No direte voi, beh, neanche io a dire la verità! (e ci mancherebbe, mica studio ittiologia! :-D )

Non c'è alcun problema, Internet, e soprattutto [Google](#), ci vengono in aiuto. In questa prima fase, infatti, dobbiamo essere sicuri di avere bene presente come sia fatto il soggetto da rappresentare, e per fare ciò dobbiamo raccogliere quanto più materiale grafico è possibile, ben vengano enciclopedie, il National Geographic o la stessa Internet, come dicevo poco più su.

L'importante è definire correttamente un profilo di forma e colore del nostro piccolo amico.

## Modellazione

Il nostro Nemo è quello che si definirebbe un modello *organico*, ovvero non ha spigoli di alcun tipo, da dovunque ci dovessimo mettere ad osservarlo, ci apparirebbe sempre "bombato", levigato nelle forme rotonde.

Non è agevole per noi creare un oggetto di questo tipo servendoci di un'approssimazione, per quanto buona, basata su vertici, spigoli e facce, i mattoni principali della grafica 3d generata al computer.

Il rischio al quale andremmo in contro è rappresentato o dalla scarsa qualità grafica, o dall'alto numero di poligoni utilizzati (come sinonimo di faccia, ovvero una superficie che ha per perimetro spigoli, uniti nei vertici).

A dire la verità, nei videogiochi si usa proprio questo approccio, qualsiasi cosa noi volessimo modellare, la costruiremmo sempre a partire da questi elementi fondamentali, con tutti i rischi del caso quando dobbiamo lavorare con oggetti tondeggianti.

Infatti si parla di modelli *low poly* (a pochi poligoni) nei VG, ma in quel caso l'uso sapiente delle texture sopperisce alla mancanza di dettagli tridimensionali.

Non è nelle nostre intenzioni creare un modello per un videogioco, siamo allora più liberi di usare qualsiasi tipo di strumento che il nostro programma ci mette a disposizione.

Uno di questi sono le *Subdivided Surfaces* di Catmull-Clark, cioè un modo per rappresentare matematicamente l'oggetto, a partire da un'armatura semplificata, formata di vertici, spigoli e facce (una *mesh*), che è facilmente gestibile.

In parole povere, nel modellare, noi utilizzeremo questa mesh di base, che ci servirà solamente per inscrivere al suo interno l'oggetto vero e proprio, definito da formule matematiche. Questo vuol dire che saremo noi a stabilire quanti vertici usare, e quindi quanto "pesante" risulterà il modello, per approssimare quello matematico.

Non lasciatevi spaventare dal nome, sono davvero semplici ed anche divertenti da usare, e lo vedremo subito!

La prima cosa da fare, per iniziare a modellare, è sceglierci, dal materiale raccolto precedentemente, una foto di pesce pagliaccio in posizione quanto più è possibile ortogonale al punto di vista della foto stessa.

Mi spiego meglio, è fondamentale che voi disponiate di una *vista laterale* del piccolo animale, in modo da apprezzare al meglio, la variazione di almeno due dimensioni delle tre di cui è formato naturalmente il pesce.

Perché ho detto "almeno due dimensioni", ma noi dobbiamo modellarlo in 3d, come facciamo ad avere un riferimento per quanto riguarda la terza dimensione, lo "spessore", del pesce?

Beh, idealmente, dovremmo disporre di almeno due fotografie, una dall'alto ed una laterale, ma vi sfido a trovare una foto di pesce pagliaccio dall'alto che sia adoperabile per i nostri scopi!

Io perlomeno non ci sono proprio riuscito, ed ho deciso di ricostruire la terza dimensione un pò ad occhio, ed un pò facendomi un'idea dalle tante altre foto, le quali, qualcuna meglio e qualcuna peggio, ci danno un'indicazione sullo spessore del nostro soggetto.

Apriamo Blender e carichiamo l'immagine scelta come sfondo della vista laterale.



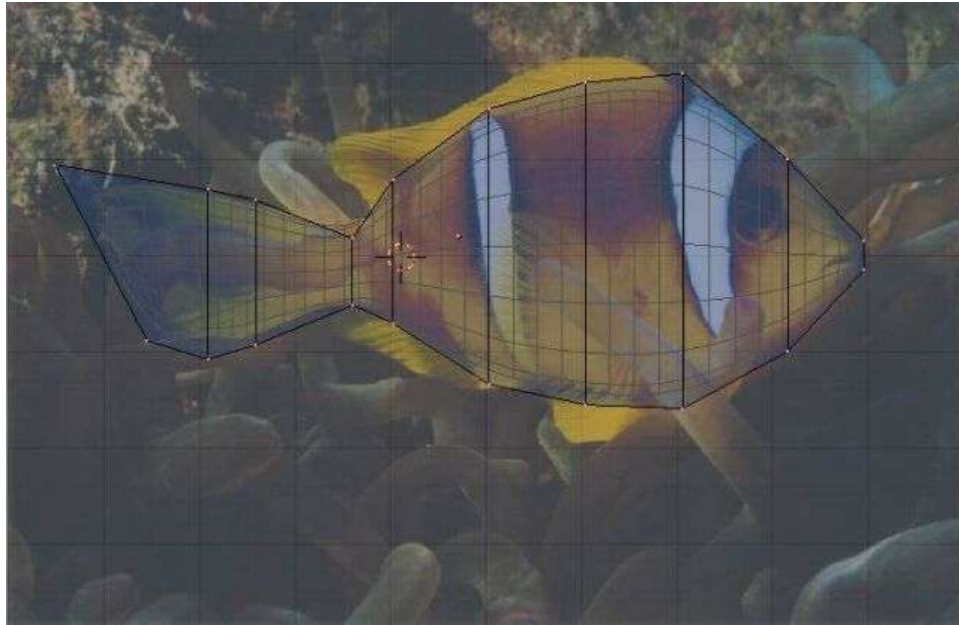
Sembra impossibile, ma a partire da un semplice cubo definiremo a poco a poco la sagoma del nostro amico pesce, basterà semplicemente estrarre una delle sue facce lungo l'asse orizzontale della vista, e poi ripetere il procedimento con quella appena creata, e così via, aggiustando di volta in volta l'altezza rispetto alla foto che abbiamo sullo sfondo, come si vede qui sotto.



Adesso è ora di attivare le SubSurf con l'apposito bottone presente nell'Editing Tab (F9), per

renderci conto di quello che effettivamente stiamo creando.

Non dimentichiamoci di portare a 2 lo slider *Subdiv* e quello ad esso adiacente, in modo da impostare a 2 il livello di suddivisione, sia nella fase di editing che in quella di rendering. Dopodiché, selezionando tutti i vertici, premete il tasto *Set Smooth*, in questo modo il tipo di suddivisione adoperato sarà di quello che crea un modello ben levigato e non composto di facce piatte.



Ora che vediamo, all'interno della mesh circoscritta, la forma che il nostro modello avrà in definitiva, possiamo aggiustare al meglio i contorni, in modo da rispecchiare perfettamente la foto. Una volta che il risultato ci soddisferà, possiamo considerare terminata questa prima fase.

**Nota:** La foto di riferimento che ho utilizzato mostra quanto questo modello di base sia poco dettagliato, mancano le pinne laterali e quelle al di sopra ed al di sotto del corpo. Per rendere quanto più semplice possibile questo piccolo tutorial ho considerato conclusa qui la fase di modellazione.

Ma nulla vi vieta di ricreare anche gli altri particolari, sempre ricorrendo alla tecnica usata per il corpo, come io stesso ho fatto per il mio modello originale.

## Texturizzazione

Per texturizzare il pesce vi dico da subito che utilizzeremo la tecnica dell'*UV Mapping*.

Sapete già di certo che le texture sono immagini applicate ai modelli tridimensionali per riprodurle fedelmente le caratteristiche visive, per dargli colore, lucentezza, riflessioni, rilievi o trasparenza particolari, per esempio.

L'*UV Mapping* è una particolare tecnica di "mappaggio" (ovvero di corrispondenza tra il modello e la texture), più raffinata di un mappaggio classico che, considerando il nostro oggetto come una primitiva, lo avvolge di conseguenza.

Mi spiego meglio, dovendo dotare il nostro pesce di particolari cromatici, ad esempio la tipica striscia argentea che gli gira attorno completamente, dobbiamo sistemare il colore proprio nel

punto che ci serve, non possiamo considerare il pesce come una palla e avvolgerlo con una texture tutta rossa.

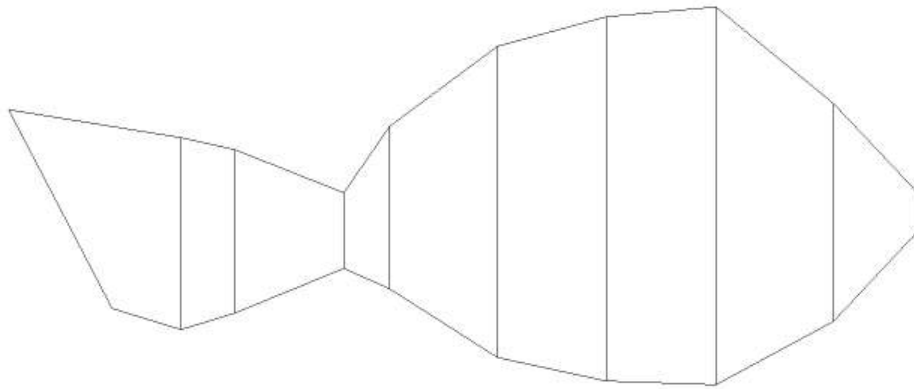
Le due lettere U e V corrispondono rispettivamente all'asse x ed y della texture, che poi verranno mappate su quelle delle facce dell'oggetto, avvalendoci dell'*UV/Image Editor* di Blender.

Le texture non scendono dal cielo solamente perché servono a noi, ma dovremo crearcele ad hoc, usando come base un'immagine che rappresenti la sagoma del nostro modello.

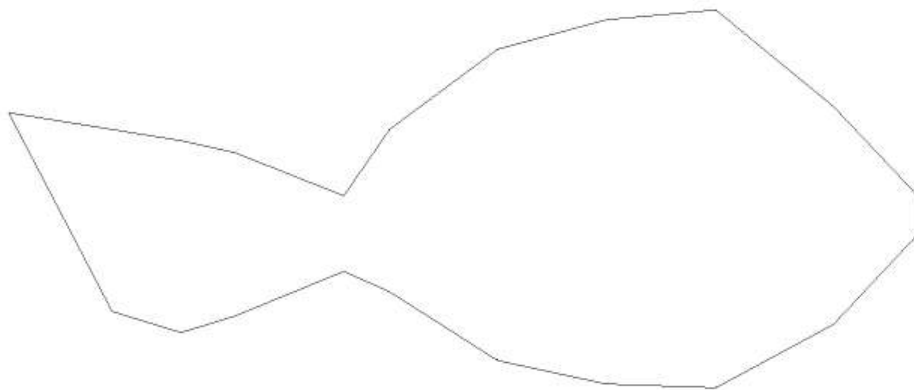
Armiamoci quindi di pazienza e di qualche script, e vediamo come fare.

L'ideale sarebbe stato utilizzare [l'UV Export.py](#) di theeth, ma a me non ha funzionato. Ho deciso allora di adoperare l'ottimo [UVMapper](#) (disponibile solo per win, ma funzionante con Wine), coadiuvato dall'[wavefrontOBJ.py](#) per convertire le mesh in OBJ, l'unico formato accettato dall'UVMapper.

Giocando un po' con i parametri del programma, ricaveremo la sagoma del nostro amato pesciolino, assicuratevi di esportarla ad una risoluzione conveniente, vi consiglio almeno il doppio della risoluzione del filmato che vorrete creare. Questo ci assicura che anche una visuale ravvicinata del modello texturizzato non sia di cattiva qualità.



Una volta salvata l'immagine, caricatela subito in GIMP ed asportate a mano le linee divisorie delle singole facce.

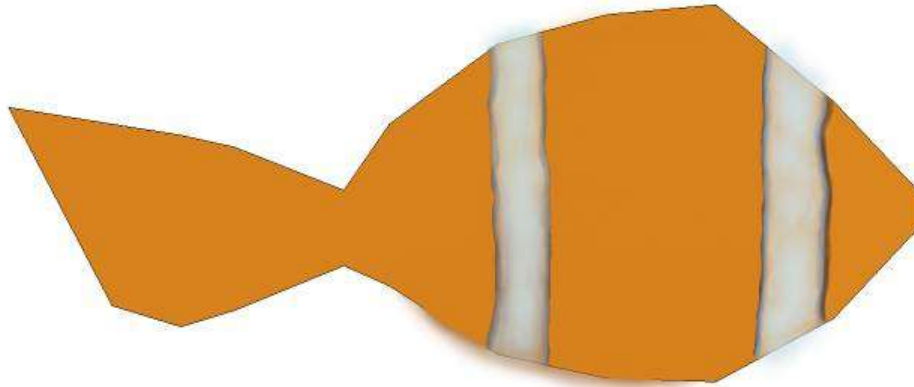


Ora siamo finalmente pronti per mettere mano alla tavolozza e sporcarci di colore!

Carichiamo un'immagine di riferimento, tra le tante che abbiamo messo insieme all'inizio, e cerchiamo di riprodurre su un layer a parte, la texture cromatica del pesce.

Usando molto il *color picker* e l'*airbrush* dovremmo in poco tempo arrivare a qualcosa di

simile:

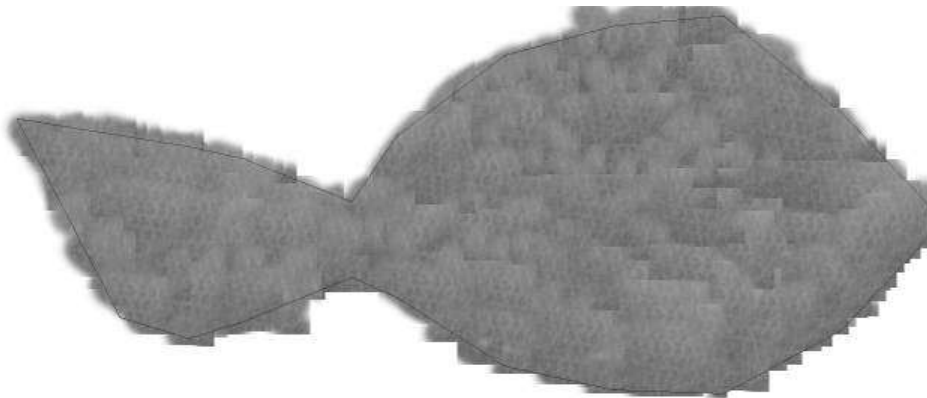


Considerato la semplicità del risultato che vogliamo ottenere e le nostre (mie pure :-)) scarse capacità, una texture così va già bene, e vi stupirete di quanto migliori la resa dell'oggetto quando gliela applicheremo.

Passiamo ora a disegnare la texture per il *bump*, cioè per i rilievi del modello, nel nostro caso, per ricreare le squame del pesce.

Qui il processo è un po' più complicato, ho ritagliato da un'altra immagine di riferimento una porzione dove si vedessero bene le squame, poi l'ho desaturata e ritoccata nei suoi parametri cromatici, in modo da avere un'immagine in b/n a luminosità uniforme, requisito fondamentale per questo tipo di texture, che usa i toni di grigio per determinare l'altezza dei rilievi.

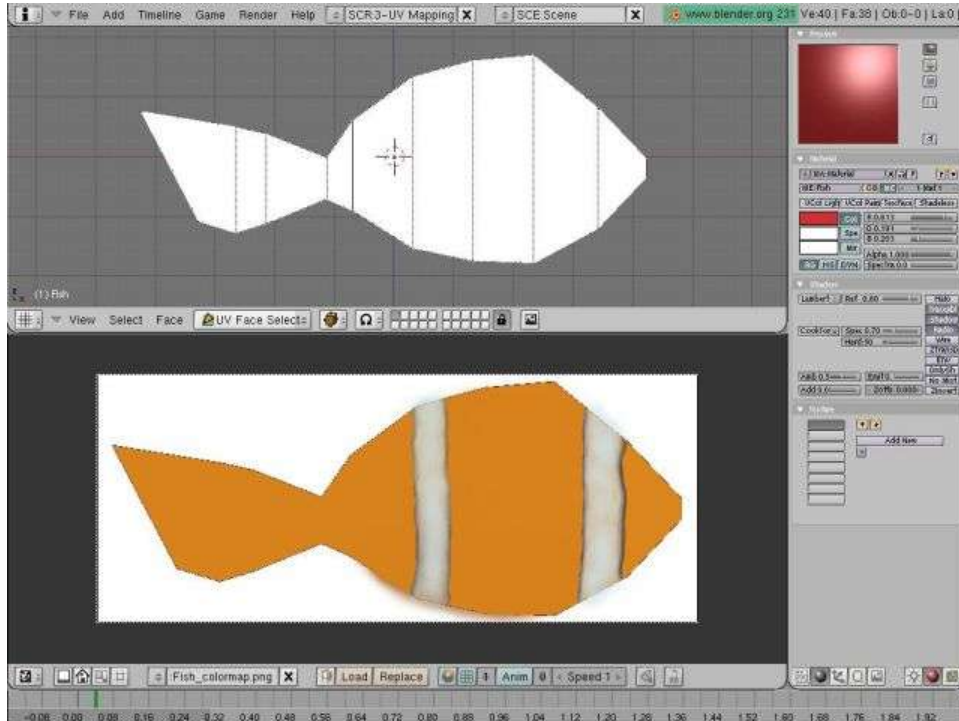
Ora che abbiamo una piccola immagine cromaticamente neutra come base, usando lo strumento *clone* la riproduciamo su un terzo layer, avendo l'accortezza di rendere invisibile il secondo, contenente la *colormap*.



Ci fermiamo qui, uh, che fatica... ci facciamo bastare queste due texture personalizzate, potevamo continuare ancora, potevamo crearne una per ogni differente caratteristica visiva mappabile contemplata in Blender, ma non aspiriamo a questa maniacale completezza. Ci basta sapere che qualsiasi risultato ne verrà fuori, sarà solo nostro!

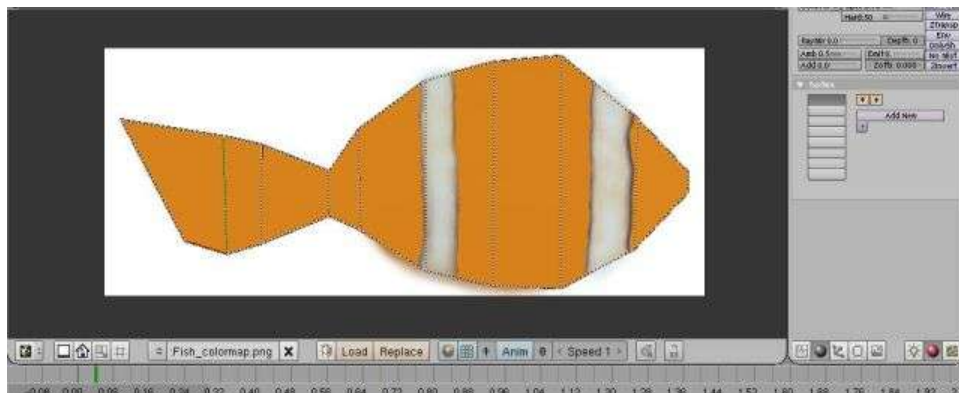
Terminata questa fase, che normalmente sarebbe demandata ad un team di grafici 2d professionisti e specializzati, ma che noi, da veri factotum hobbysti della grafica abbiamo assunto sulle nostre spalle, è tempo di passare alla mappatura vera e propria.

Tornate a Blender, passate allo schermo *Material* e clonatelo facendo *ADD NEW*, volendo, potete chiamare questo nuovo schermo come ho fatto io, con molta originalità, *UV Mapping*. Centrate una delle aree sulla vista laterale del modello, mentre per l'altra, passate all'*UV/Image Editor*.



A questo punto siete pronti per mappare sul serio, selezionate tutte le facce che vedete nella vista laterale, e, premendo U, scegliete *From Window* dal menù *UV Calculation* che sarà intanto comparso.

Adesso, come avrete capito dando uno sguardo all'area dell'UV Editor, dovete, con quanta precisione potete, far combaciare la sagoma tratteggiata con quella dell'immagine, cercando di fare come si vede in questa figura:



Adesso, la stessa identica cosa va fatta per le facce che compongono l'altro lato del modello,

ruotatelo e quindi selezionatele. L'unica accortezza, dopo aver scelto il tipo di calcolo degli assi UV dal menù, è quello di fare un flip orizzontale del tratteggio, che altrimenti sarà simmetricamente speculare all'immagine.

**Nota:** Nella versione 2.31 di Blender, l'opzione *From Window* del menù *UV Calculation* non funziona. Dovrete usare un'altra versione per compiere questo passaggio, tornando eventualmente alla 2.31 solamente dopo averlo esplicitato.

**Nota:** Vi ho detto poco sopra di non essere riuscito in alcun modo a far funzionare l'*UV\_Export.py*, a quanto mi è parso di capire è stata colpa di un mio erroneo utilizzo dello stesso, in realtà bisogna prima mappare le facce dell'oggetto nell'*UV/Image Editor* (anche se non si ha ancora a disposizione la texture) e poi esportare la tga di riferimento con lo script. Provatevi voi ed eventualmente mi fate sapere.

La mappatura che avete appena fatto varrà per ogni texture che voi vorrete mappata secondo l'UV Mapping.

Questo vuol dire che il lavoro appena fatto, varrà per entrambe le texture che ci siamo creati, l'unica cosa da fare è definire effettivamente quali saranno le texture che comporranno il materiale da assegnare al modello.

Createvi un nuovo materiale e poi due texture, chiamate quest'ultime in modo che vi facciano ricordare che una conterrà la colormap, ed un'altra la mappa del bump.

Per la texture della colormap, andate nella tab delle texture (F6), selezionate come tipo *image*, e quindi caricate la texture che ci siamo creati prima.

Ora, tornate alla tab dei materiali (F5) e nella sezione Texture scegliete UV come *Map Input* e Col nella tab denominata *Map To*.

Ripetete la stessa cosa per l'altra texture, cambiando solamente l'ultimo passaggio, invece di selezionare Col, premete su Nor e su Spec, ed spostate lo slider Nor verso un valore maggiore. Fate delle prove finché il risultato non vi piacerà.

## Animazione (1)

L'animazione che ho in mente per questo piccolo tutorial è qualcosa di semplice, senza dubbio, ma che comunque evidenzierà alcune cose interessanti, come l'animazione per fotogrammi chiave e per cinematica diretta.

Questa può essere pensata come composizione di due movimenti indipendenti, ma fusi tra loro, che accadono in contemporanea, per uno useremo la prima tecnica, per l'altro prevalentemente la seconda.

Il primo è il movimento del pesce rispetto alla telecamera ed alle luci, l'altro è il movimento di alcuni vertici che compongono il corpo del pesce, relativamente alle loro posizioni originali, o allo stato di riposo, si potrebbe dire.

Mi rendo conto di essere stato poco chiaro, quindi mi appresto a mostrarvi effettivamente cosa ho in mente.

L'animazione che costruiremo non è nulla di particolare, il pesciolino "nuoterà" lungo una sola direzione, entrando nel campo della telecamera dopo qualche fotogramma, sosterrà per un po', verso la metà del filmato, e poi sparirà dal lato opposto a quello dove all'inizio è comparso.

Il nostro occhio non si farebbe impressionare da un movimento simile, abituato com'è alla realtà ed alla sua rocambolesca, colorita e stravagante vitalità, ma se ci fermiamo un secondo ad analizzare il fenomeno - il fulcro dell'attività dell'artista è sempre stato l'osservare - ci rendiamo



conto che si tratta di una *traslazione* lungo un solo asse di un oggetto, che oltre a cambiare velocità, in fisica diremmo che ha un'accelerazione non nulla, cambia anche forma, noi diremmo che nuota, ma non so quanta differenza faccia al programma.

Nel suo moto e nel suo cambio di forma non c'è arbitrarietà, il moto è costante, poi decelerato, poi accelerato e di nuovo costante, mentre il suo cambiar forma sembra si possa collegare a rotazioni e traslazioni particolari di alcuni dei suoi vertici.

Sì, c'è da uscire pazzi, concordo, ma Blender ci verrà in aiuto, ve lo garantisco. E' basilare ricordarsi che non si deve mai perdere di vista l'analisi che abbiamo fatto, non dobbiamo farci soverchiare dal caos e dalla pluralità delle combinazioni, dobbiamo tenere a mente la scomposizione del movimento in unità più piccole e semplici da gestire.

Il primo dei due movimenti che abbiamo identificato all'inizio è il più facile da ricreare.

Vi accorgete della potenza dei fotogrammi chiave e delle curve di interpolazione, due cose che ritroverete anche nel secondo tipo di movimento.

Carichiamo il modello in Blender e passiamo al primo schermo, quello chiamato *Animation*, ora assicuriamoci di avere da un lato la vista della telecamera (basterà premere "0" dal tastierino numerico) e dall'altro l'*IPO Curve Editor*, lo strumento che ci permetterà di manipolare le curve di interpolazione del programma.

Sì, ho capito, non mi state seguendo più, devo fare un'altra digressione esplicativa, ben venga allora.

Il concetto di fotogramma chiave e quello di interpolazione sono strettamente correlati, è vitale poter disporre di queste due funzioni, altrimenti dovremmo definire ogni fotogramma come fa il fumettista per le sue tavole, le disegna tutte!

Il computer invece ci viene in aiuto, il grafico penserà a definire gli attributi del movimento (traslazione, rotazione, cambio di dimensione) solo relativamente a certi fotogrammi "chiave", mentre quelli intermedi verranno generati matematicamente. Il concetto è che se è possibile far fare al computer i lavori più regolari e ripetitivi, allora che così sia! L'artista è più libero di pensare alle "irregolarità" del movimento, mentre ne lascia la ricostruzione regolare all'elaboratore.

Un esempio vi aiuterà a capire, se devo far muovere una sfera da un punto A ad un punto B, e voglio che il movimento sia rettilineo uniforme, cioè a velocità costante, mi basterà definire il punto di partenza e quello di arrivo, e poi andare a definire il tipo di interpolazione per essere sicuri che sia *lineare*, cioè una retta.

Tutti vi ricorderete, dai tempi della scuola, che in un grafico spazio-tempo, il moto rettilineo uniforme è rappresentato da una retta, da una funzione lineare per l'appunto, "spazi uguali in tempi uguali", in quel particolare moto lo spazio percorso è direttamente proporzionale al tempo trascorso.

Spero che adesso vi sia più chiaro quello che andremo a fare, e soprattutto le potenzialità dell'*IPO Curve Editor*, che è proprio lo strumento con il quale definiremo i punti chiave e la funzione grafica interpolata.

Vi assicuro che non c'è bisogno di alcuna conoscenza fisico-matematica per riuscire a ricreare tutti i movimenti che vi passano per la testa, ma solamente molte prove, intuizione e le capacità di analisi che vi ho illustrato all'inizio di questo lungo discorso.

Ora che avete il programma aperto, il modello caricato, e le finestre disposte come ci fa più comodo, è arrivato il momento di definire i fotogrammi chiave, ce ne servono due, il primo che stabilirà il punto di partenza del pesce, ed il secondo, come avete intuito, il punto di arrivo. Prima però, definiamo la lunghezza dell'animazione, io ho deciso di crearne una di 10s, dal

momento che ci sono 25fps nel formato PAL tradizionale, i frame totali saranno 250, impostiamo questo numero nel *Render Tab* (F10), nel pulsante con vicino scritto *End:*, nella sezione *Anim*.

Usando la vista della telecamera come riferimento, sistemiamo il pesciolino sulla sinistra, in modo che sia appena fuori dal rettangolo tratteggiato più esterno (che ci mostra il campo di ripresa) e premiamo I.

Istantaneamente apparirà il menù *Insert Key*, che, come avrete capito, ci permette di impostare un fotogramma chiave, non solo, ci da anche la possibilità di scegliere quali parametri tenere in considerazione, a noi interessa solamente la posizione, scegliamo quindi la prima opzione, *Loc*. Guardate cosa succede nella finestra di destra, nell'*IPO Curve Editor*, sono comparse tre linee diritte colorate, rappresentano il grafico tempo-coordinata spaziale, ce n'è una per ogni dimensione, *LocX*, *LocY* e *LocZ*.

A questo punto, aggiungiamo un altro fotogramma chiave per vedere cosa succede, per prima cosa, raggiungiamo il fotogramma finale, poi spostiamo il pesce lungo una sola direzione ed in modo che sia appena fuoriuscito alla destra del campo della telecamera, premiamo ancora una volta I ed impostiamo un'altra chiave di tipo *Loc*.

Visto cos'è successo? Sono comparsi altri 3 punti chiave sulle nostre 3 curve, 2 di esse sono ancora delle rette, ma la terza, la *LocX*, presenta una variazione di altezza tra il frame 0 ed il frame 250.

Spero sia chiara a questo punto la stretta correlazione tra il grafico rappresentato dalle curve ed i parametri del moto dell'oggetto. Avendo noi mosso il pesce lungo un solo asse, l'unico parametro che varia durante l'animazione è la sua coordinata rispetto all'asse X, è questo che ci vuole dire quella curva!

Ma le sorprese non finiscono qui, mettendoci a giocare con i punti chiave, ci accorgiamo come esse siano in realtà punti di controllo di curve Spline, e pertanto presentano i diversi tipi di maniglie per controllarne la forma, non solo, proprio come faremmo per modellare una Spline, premendo CTRL possiamo aggiungere altri punti e variare con ancora più precisione la forma della curva rispetto ai nostri bisogni!

Immagino che ormai abbiate capito della potenza demiurgica che è al vostro servizio, e che vogliate imparare a servirvene per i vostri scopi divini. :-)

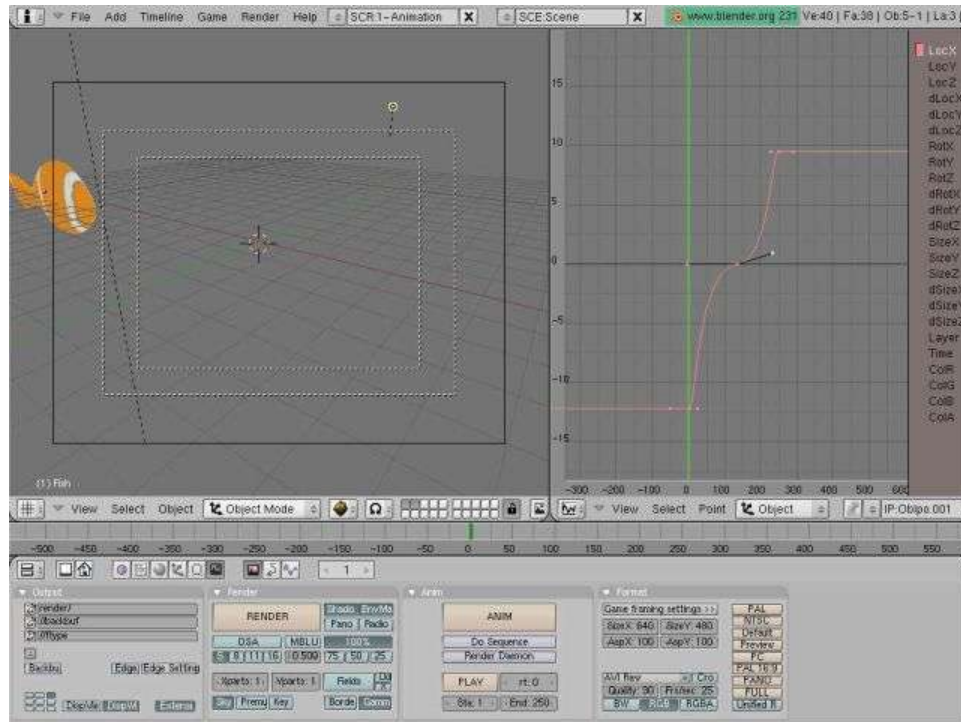
Fate tutte le prove che volete, sembra sempre tutto molto chiaro all'inizio, ma ce ne vuole finché riusciate a mettervi in testa la relazione tra quei punti e quelle curve e l'effettivo movimento delle vostre creazioni.

Mmm, vediamo se avete capito... aggiungete un terzo punto, verso il mezzo dell'animazione, ed editate la curva in modo che il modello sembri fermarsi e poi riprendere, con una discreta accelerazione, il suo moto verso destra.

Potete in ogni momento verificare l'animazione con l'anteprima che Blender vi mette a disposizione premendo ALT-A quando avete il focus su una finestra di tipo *3D View*.

Ah, dal momento che non c'è alcuna variazione lungo l'asse Y ed Z, potete cancellare le rispettive curve.

La situazione che vi si parerà davanti sarà simile a questa:



Direi che la prima parte della fase animativa è completata, ora ci attende la seconda, forse più interessante e divertente, ma anche più lunga e stancante, dobbiamo dotare il modello di un movimento suo intrinseco, di un'alterazione nella forma, altrimenti, così com'è, potrebbe anche essere confuso con un manico di scopa ;-).

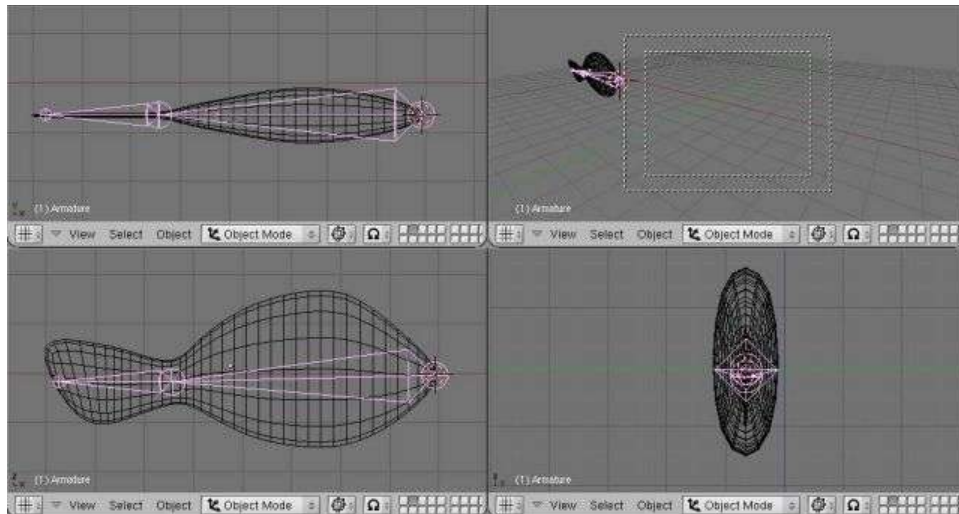
## Animazione (2)

Ciò che faremo in questa fase sarà il definire un'armatura di ossa che ci aiuterà a controllare la deformazione del pesce, in maniera che il suo movimento traslatorio sembri dovuto al movimento della sua pinna.

Ognuna di queste ossa potrà essere collegata alla precedente, tramite un punto comune, una specie di articolazione, e, soprattutto, ognuna di queste ossa controllerà il movimento di un certo gruppo di vertici.

Posizioniamo il cursore 3D dove si trova la testa del pesce, ed aggiungiamo l'oggetto armatura. Fatto questo noteremo la comparsa del nostro primo osso, il quale, partendo dal cursore 3D, seguirà la posizione del puntatore. Premendo il tasto sinistro del mouse nei pressi dell'inizio della coda fisseremo l'estremità finale di questo primo osso e contemporaneamente quella iniziale del secondo, il quale, ripetendo le stesse azioni di prima, lo fisseremo alla fine della coda del nostro pesce.

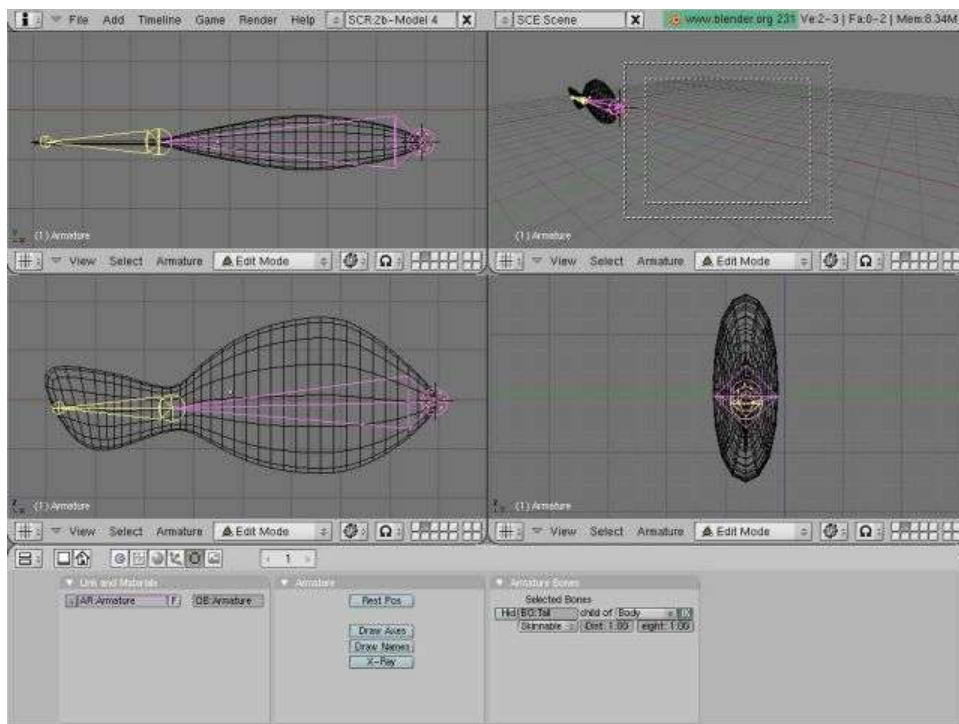
Aiutatevi con questa figura:



Definita l'armatura, è tempo di andare a vedere quali vertici verranno influenzati dal movimento di quali ossa.

Per fare questo, dobbiamo dare il nome a quest'ultime ed ai gruppi di vertici, vediamo come fare la prima di queste due cose.

Selezioniamo l'armatura, premiamo TAB per andare in *Edit Mode*, poi selezioniamo i due estremi del primo osso e, andando nell'*Edit Tab*, cambiamo la scritta "Bone", nella sezione *Armature Bones* in qualcosa tipo "Body", o "Corpo", o come volete. Ora facciamo la stessa cosa per il secondo osso, lo selezioniamo e cambiamo la scritta "Bone.001" in "Tail", "Coda" o altro. Notiamo questa volta il rapporto di dipendenza tra le due ossa, c'è scritto infatti "Tail child of Body", il secondo osso è "figlio" del primo.

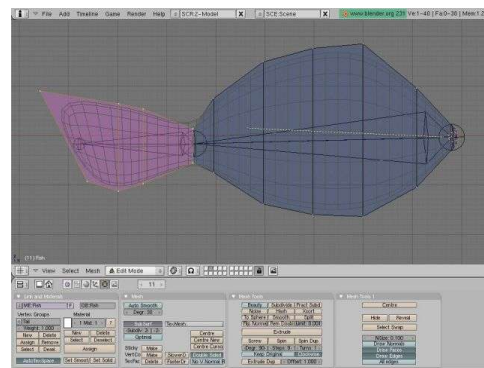
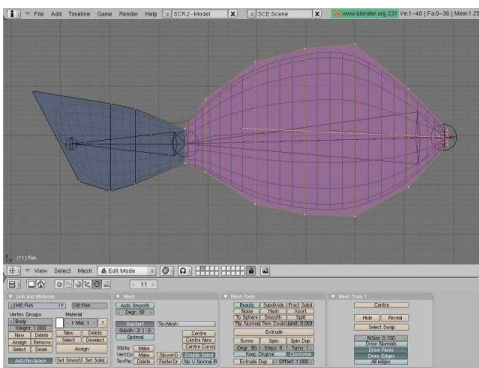


Per creare i *Vertex Groups* selezioniamo il modello del pesce, premiamo TAB e spostiamoci ancora una volta nell' *Edit Tab*, selezioniamo tutti i vertici tranne quelli che compongono la coda, poi, nella sezione *Link and Materials*, premiamo *New* sotto la scritta *Vertex Groups*, scriviamo il nome, poi premiamo su *Assign*.

Per creare il secondo gruppo di vertici, deselectioniamoli tutti con A, poi selezioniamo solo quelli della coda e ripetiamo le azioni precedenti.

I nomi assegnati non devono essere arbitrari, devono invece rispecchiare quelli dati alle ossa dell'armatura.

Notate, nelle due figure sottostanti, come alcuni vertici appartengano a due gruppi diversi, questo vuol dire che verranno influenzati da due diverse ossa contemporaneamente.



Ora l'ultimissima cosa, per sancire definitivamente l'unione tra armatura e modello, e verificare con mano il nostro lavoro, dobbiamo creare un rapporto gerarchico tra queste due entità.

Selezioniamo prima il pesce (è importante l'ordine!), schiacciamo SHIFT per selezionare contemporaneamente l'armatura e premiamo CTRL-P.

Comparirà il menù *Make Parent*, scegliamo da questo l'opzione *Use Armature*, comparirà un secondo menù, *Create Vertex Groups?*, dal quale sceglieremo *Don't Create Groups* (abbiamo infatti già creato prima i gruppi di vertici).

Adesso che l'armatura è "genitore" del modello, possiamo verificare le deformazioni selezionandola e ruotando in *Pose Mode* le singole ossa.

Un'ultima cosa da aggiustare, dobbiamo trasferire la *IPO Curve* definita prima per la traslazione del pesce sull'armatura, altrimenti questa rimarrà ferma, e le deformazioni non saranno corrette.

Allo stesso tempo eliminiamo il collegamento tra questa IPO e il modello, visto che quest'ultimo subirà le stesse trasformazioni di moto del suo "genitore", ovvero l'armatura.

Tornando allo schermo *I-Animation*, selezioniamo il pesce e premiamo la grande X a fianco della scritta "IP:ObIpo.001", poi selezioniamo l'armatura e scegliamo dal menù la IPO di prima. Abbiamo così trasferito la curva del moto da un oggetto all'altro, come volevamo.

Premendo ALT-A però, noteremo una piccola sfasatura, ciò è dovuto ai differenti centri dei due oggetti, l'armatura ha il centro sulla prima estremità del primo osso, mentre il nostro modello, probabilmente lo avrà da qualche altra parte.

Credo che ormai ve la possiate cavare da soli nel sistemare questo piccolo inconveniente.

Verificato che tutto sia ok, è il momento di creare le diverse pose che ci serviranno.

Vi dico da subito che saranno solo tre le pose chiave, tutte le altre saranno interpolate a partire da

queste, seguendo lo stesso principio lungamente trattato nella fase precedente.

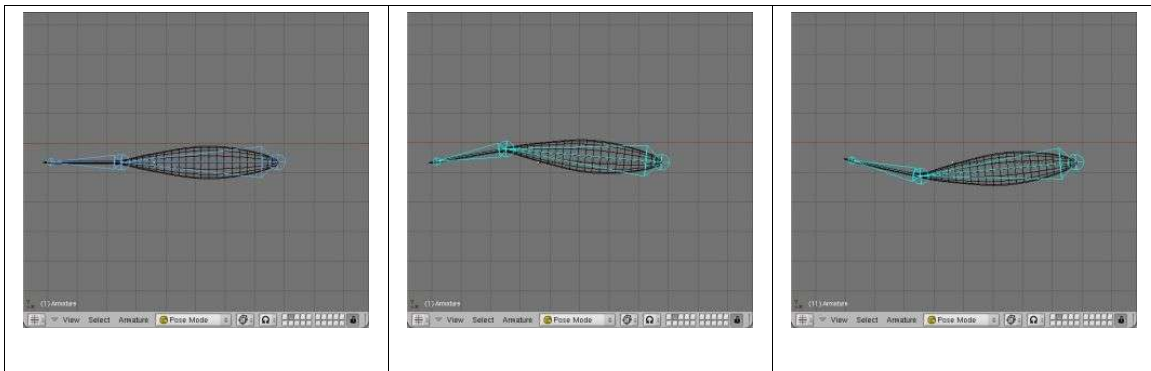
Selezionate l'armatura ed andate in *Pose Mode*, disponete le ossa in modo che il pesce sia dritto, come se non fosse stato mai deformato, e premete I, dal menù *Insert Key* scegliete *Rot*, dal momento che due sole ossa congiunte possono unicamente ruotare una rispetto all'altra.

Cambiate l'*IPO Curve Editor* nell'*Action Editor*, noterete la presenza di due strisce col nome delle ossa implicate nel movimento e due rettangolini grigi che rappresentano la posa chiave di queste.

Selezioniamo entrambi i rettangolini e spostiamoli con G, proprio come fossero oggetti, verso la metà dell'animazione, infatti, in corrispondenza dell'arresto della traslazione, vi sarà anche uno stato di riposo delle pinne.

Ora invece definiamo le altre due pose chiave, quelle della pinnata verso destra, e quella simmetrica verso sinistra, in questo movimento, non solo la pinna, ma anche il corpo seguirà l'azione.

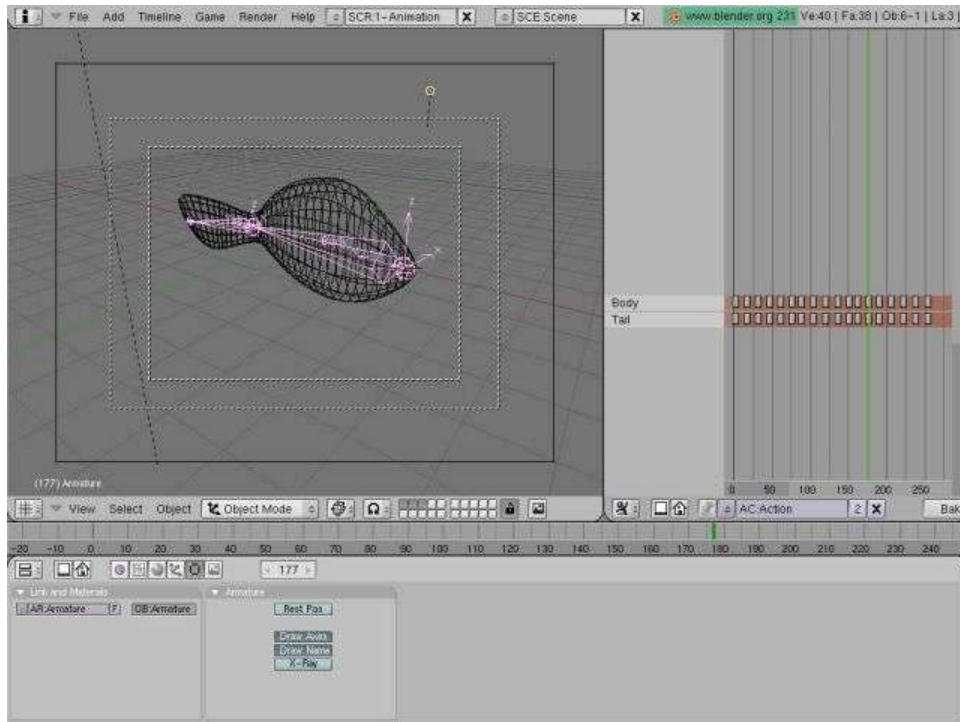
Osservate qui sotto le tre posizioni:



Ora il lavoro più duro in assoluto ci attende... cercare di far combaciare l'accelerazione e la velocità del moto traslato del pesce con la frequenza della sua pinnata, qui purtroppo nessuno strumento ci aiuta, forse la simulazione fisica dei fluidi di Maya Unlimited, ma, ehm, credo questo esuli leggermente la trattazione di questa mia guida. :-)

Come dicevo, non ci viene in aiuto nulla di particolare, ma solamente il copiare le tre pose chiave in diversi punti della timeline dell'animazione ed il ponderare con attenzione la loro distanza (che poi si traduce nella frequenza della pinnata) in corrispondenza della curva del moto, che invece ci indica lo spostamento del corpo e di riflesso (in analisi matematica diremmo per derivazione :-)) la sua velocità ed accelerazione.

Ecco un'altra immagine per capire un po' cosa aspettarsi alla fine:



E' un lavoro iterativo fatto di molte, moltissime prove, cambiamenti, controlli, revisioni, etc... Quando l'anteprima con ALT-A vi darà soddisfazione, e quando l'inquadratura, le luci e tutto il resto, sarà stato reso come più vi agrada, è tempo di leggere la conclusione.

## Conclusioni

Uff, è finita! Ora è solo tempo di impostare le preferenze relative al rendering (la risoluzione, il formato, l'antialiasing, etc.) nel *Render Tab* e poi premere il grande tasto *ANIM* per far partire il calcolo dei singoli frame della nostra animazione.

E' stato un lavoro lungo, stancante, a tratti magari anche frustrante, lo riconosco, io stesso ho sudato le proverbiali sette camicie e più volte ho avuto la tentazione di gettare la spugna.

Ma non è stato tutto inutile, perché è garantito che d'ora in avanti avrete più lucidità e consapevolezza nel cercare di conseguire i vostri obiettivi grafici con Blender. :-)

Ovviamente è d'obbligo la lettura in toto del manuale e di molti tutorial e la pratica continua per progredire in questo campo.

**Nota:** Questa guida è frutto di un lungo lavoro, è stata riletta e revisionata, ma è ovviamente sempre possibile che ci siano errori più o meno grandi o veri strafalcioni, non esitate a farmelo sapere!

Per correzioni, critiche e commenti: [angeloth@tin.it](mailto:angeloth@tin.it).