

Animazione e game experience

Accorgimenti per una pipeline d'animazione davvero professionale

Daniele Pezzella



Ministero delle Imprese
e del Made in Italy





TRASH

TALE OF WASTETOWN

GAMES for MOVIES,
MOVIES for GAMES



DANIELE PEZZELLA

Daniele Pezzella lavora in animazione dal 2011 con esperienze nazionali e internazionali, sia cinematografiche che videoludiche. Partito da un background in Commercials e Vfx, nel tempo si è spostato sul settore dell'animazione per videogames, affascinato dapprima dai giochi narrativi e cinematografici per focalizzarsi poi sull'animazione per gameplay con l'intento di favorire l'esperienza del giocatore dal punto di vista del game design e della responsività.

In Trash: Tale of Wastetown si è occupato della realizzazione e supervisione della maggior parte delle animazioni di gameplay e della direzione di diverse sequenze cinematiche



GAMES
FOR MOVIES
MOVIES
FOR GAMES

A1 AL1
INTERACTIVE

Animation

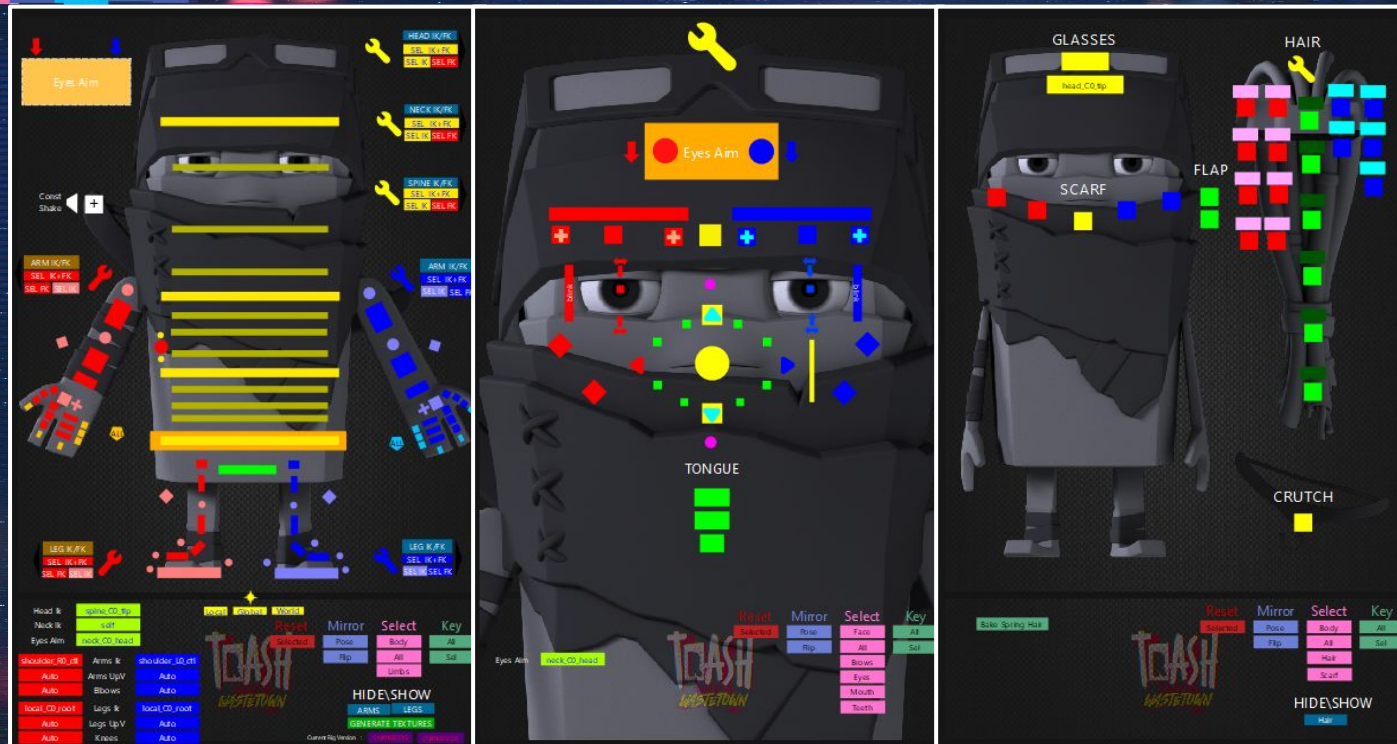
- Migliori Amici: Anim Pickers
- Animbot / Studio Library
- Loop di dinamiche: guida pratica



Migliori Amici: Anim Pickers

GAMES
FOR MOVIES
MOVIES
FOR GAMES

Costruire anim pickers per velocizzare il processo di animazione. I pickers, possono comprendere anche pulsanti che lanciano scripts costruiti ad-hoc per ogni esigenza del personaggio.



- Codifica colore
- Sezioni separate fra pulsanti aggregatori, space switch e utilità;
- Icone di impostazioni
- Grafica accattivante (eh sì, non riuscivamo a resistere...)

Sono solo alcune delle caratteristiche che rendono i picker solidi e ben navigabili.

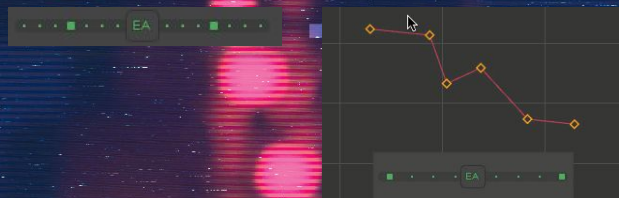
Un tripudio di picker





Eh si, nella storia dei tool salvavita c'è un prima e un dopo Animbot. Qui di seguito uno showcase delle funzioni più usate (da me)

Ease In / Out

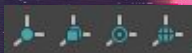


Tweener world space



- Selection sets
- Copy/Paste animation (anche fra maya diversi)
- Copy/Paste Pose (anche fra maya diversi)
- Mirror Animation
- Copy xform relationship
- Copy xform world space
- Selection Attribute space switcher

Aux Pivots



- Temp Pivot
- Temp Pivot to last object
- Temp Pivot Centered
- Temp Pivot World space



Global Offset



Motion Trails



- Apply Smart Euler Filter
- Clear Animation
- Crop Animation
- Remove Redundant Keys
- Remove Static Anim Curves
- Reverse Animation
- Smart Snap Keys

Tumble around selection



Studio Library



Caratteristiche Principali

Hub centralizzato per tutto:

- Cinematiche, Meccaniche, singoli animatori, Riggers
- Accessibile tramite API;
- Layout customizzabile;
- Scelta Namespace (Multi-Character)
- Key, Mirror, Additive
- Explorer Folder Structure
- UI Customizzabile (Grandezze, Icone, Etc.)

Sono solo alcune delle caratteristiche che rendono la studio library uno strumento essenziale.

Loop di dinamiche (mGear)



1. Trovare una combinazione funzionante di stiffness e dumping, (impostare playback a play every frame, max realtime);
2. Estendere il range di timeline al doppio della durata dell'animazione (sia in testa che in coda) individuando il frame di loop anche sulle estensioni (1-33 avrà IN-frame a 1-32 = -31, OUT-frame a 33+32 = 65)
3. Impostare Pre/Post cycle su tutte le chiavi dell'animazione su infinito;
4. bakare la "dinamica" con pre/post roll da -31 a +65 (In questo modo, la dinamica arriverà al frame 1 già morbida senza dover partire impulsivamente da 1)
5. restringere di nuovo il frame range a 1-33;
6. Aggiungere un animation layer contenente tutti i controlli dinamici bakati;
7. Copiare il frame 1 sul layer al frame 33;
8. Attraverso i 33 frames il primo frame viene portato sull'ultimo, realizzando un match di posa e mantenendo il loop dinamico senza perdere momentum.

Spring chain intensity	0
damping_0	0.3
damping_1	0.3
damping_2	0.3
damping_3	0.3
damping_4	0.3
damping_5	0.3
stiffness_0	0.2
stiffness_1	0.2
stiffness_2	0.2
stiffness_3	0.2
stiffness_4	0.2
stiffness_5	0.2

Rigging

- Slim Tape: Un tool di debug visuale
- Dinamiche dei personaggi
- Rigging in UE

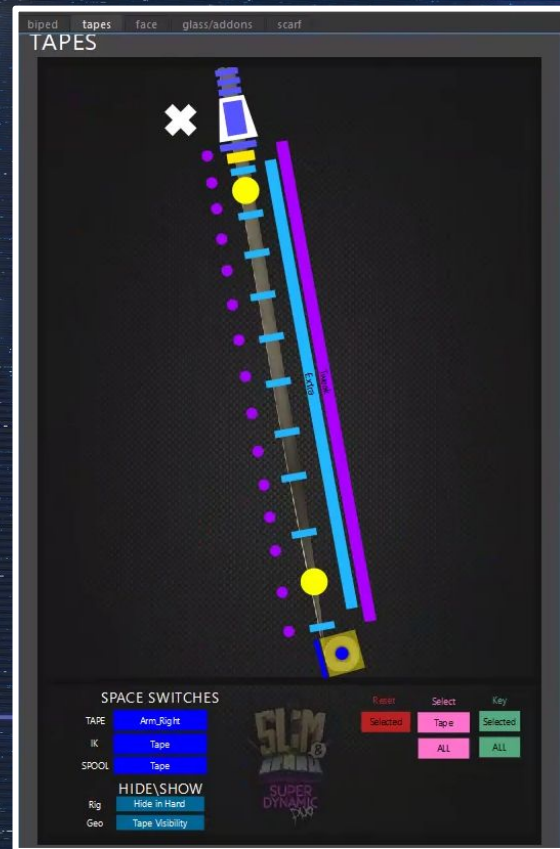
Slim Tape: Un tool di debug visuale

Animare il Tape di Slim significa avere a che fare con molteplici controlli concorrenti che si muovono molto veloce in scena, spesso dando luogo a torsioni invisibili o difficili da animare.

Molto spesso sulla catena Ik/Fk di bind, si ottenevano dei flip di 180° molto difficili da vedere a occhio, solo nello uAsset emergevano poi, ad animazione completata, dei flip e gimbal lock molto vistosi, che rendevano animare il nastro in maya un processo senza possibilità di prevederne i difetti.

Questa difficoltà e i numerosi retake iniziali di animazione, hanno portato a dover creare una soluzione mid-ground di rigging al servizio dell'animazione.

Il setup finale è costituito da 39 controlli di forma e uno di setup. I controlli gialli sono di tipo IK, mentre i blu e i viola sono dei tweaker di forma intermedia.



Slim Tape: Un tool di debug visuale



Quando si anima una lunga catena di joints stretchabili su una lunga distanza in viewport, è molto difficile tenerla in vista tutta insieme, perché bisognerebbe zoomare indietro di molto la viewport, perdendo il dettaglio su cosa avviene sulla catena stessa. Quando poi questa è bakata e esportata in fbx spesso si verificano vistosi gimbal locks in engine che flippano il nastro in maniera imprevedibile.

- All'inizio facemmo una grande catene IK/FK ibrida piena di controlli (partendo da un nastro esteso) → approccio poco controllabile dalle estremità perché deformava male quando si comprimeva. Serviva quindi qualcosa di più controllabile dalla punta;
- Quindi provammo la via del Braccio IK/FK e usammo il "polso" come nostra punta del rocchetto → anche questa via però, nonostante fosse flessibile, dava problemi quando esportavamo per unreal, per colpa di flip (soprattutto quando il nastro diventava corto);



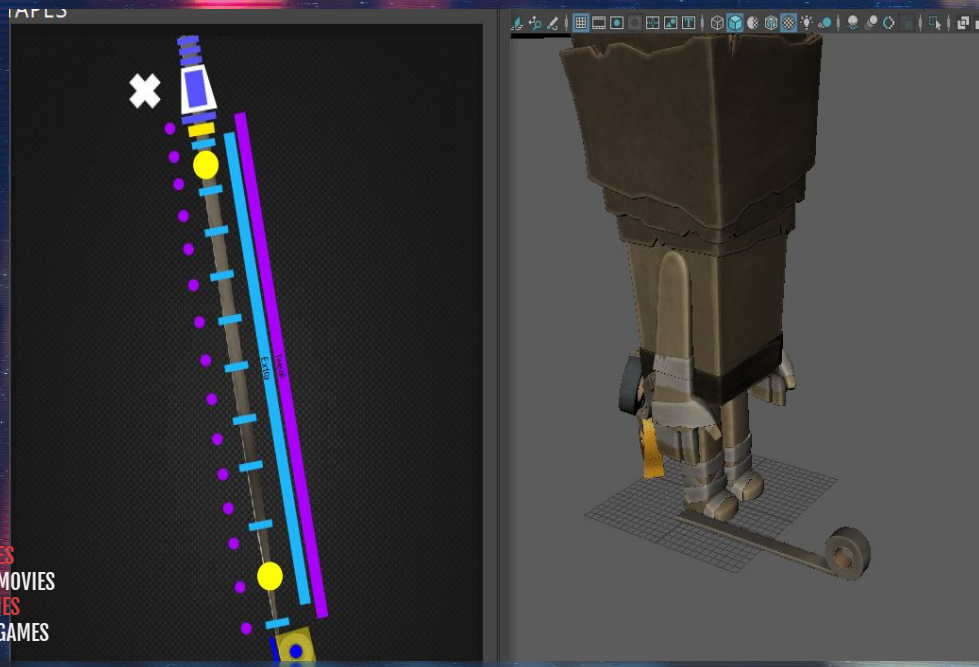
E' soprattutto nelle stealth kill che le problematiche sul tape sono emerse, perchè le distanze da coprire sono maggiori e il comportamento del tape dalla camera/viewport non è sempre riconoscibile in termini di forma.

- Quindi optammo per fare il nastro molto corto all'inizio (con una grande densità poligonale) e un setup full-IK con tangenti e piccoli tweak individuali.

Slim Tape: Rigging Visual Debug

Do's:

Creare un debugger visivo che permetta di visualizzare in movimento, anche su azioni molto veloci o molto distanti, il comportamento del tape riggato, evidenziandone gli orientamenti tramite degli attributi di debugging ausiliari, è stata la chiave di volta.



----- VISIBILITY	-----
Tape Color	white
ctl vis	on
Extra Vis	off
Tweak Vis	off
Preview	off
Gizmo	1
Target	1
Tape	0
Tape Proxy	off
----- SETTINGS	-----
Tape Follow	Arm_Right
IK Follow	Tape
Roll Follow	Tape
--- DEFORMER	-----
Wide	0
Thick	0
Sine Def	0
Amplitude	2
Wavelength	1
Offset	0
Offset	0
Wavelength	1
Amplitude	2

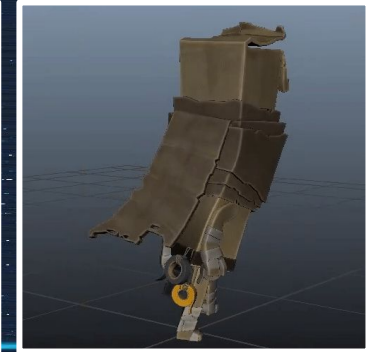
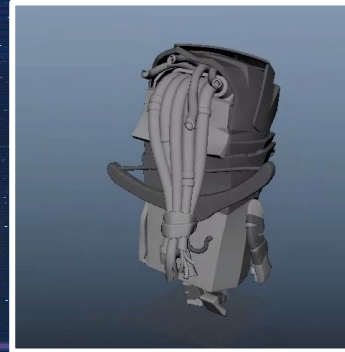
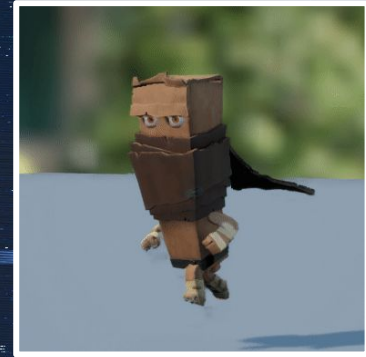
Dinamiche dei personaggi

□ In TTOW, abbiamo creato un sistema dinamico di controlli che ne permette la simulazione già in fase di animazione, per poi esportare un baking frame per frame delle dinamiche simulate.

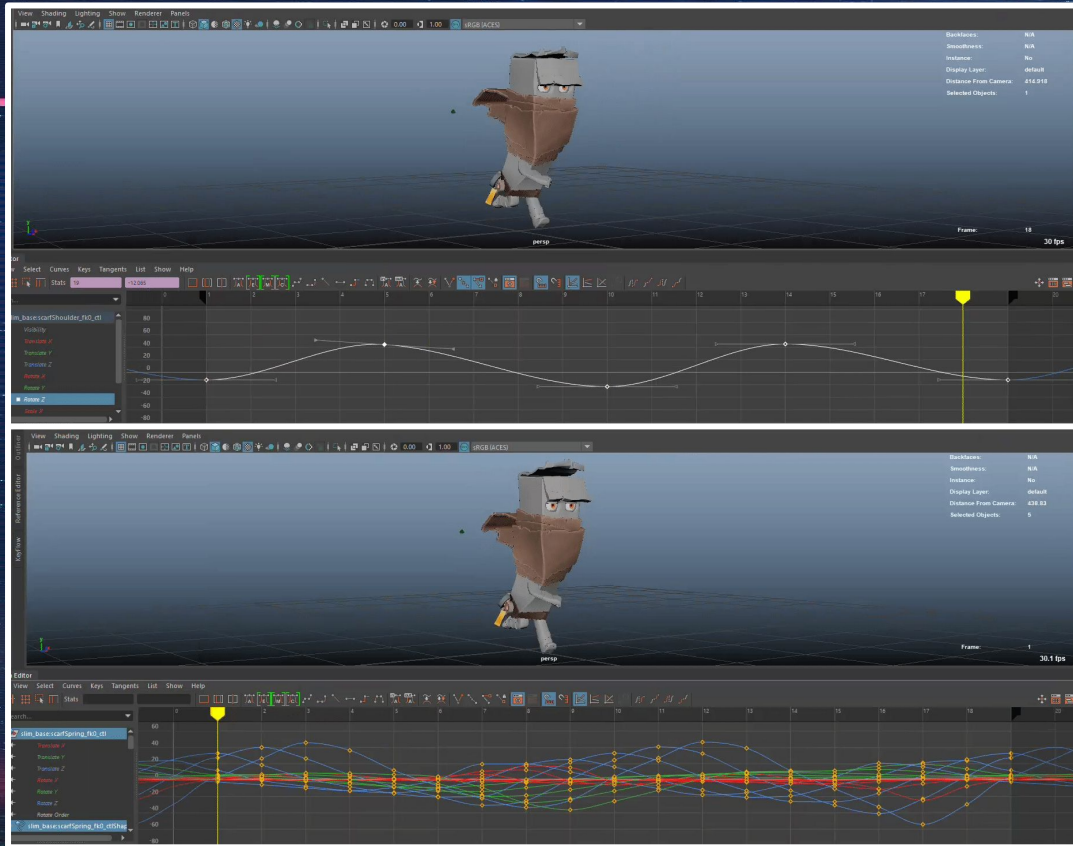
- Abbiamo usato il framework di mgear per le dinamiche. Al contrario di n-cloth, quelle di mGear non sono basate sul playback e funzionano anche semplicemente muovendo i controlli da fermo.
- Normalmente mGear fa il bake delle chiavi di animazioni su gruppi offset, ma visto che a noi servivano dei loop precisi di animazione, la "casualità" della dinamica, doveva essere controllabile tramite rig. Abbiamo quindi fatto tramite script un trasferimento (cut/paste) dagli offset ai controlli e agganciato la funzionalità al pulsante di bake spring nel picker.



Le dinamiche in engine possono essere attraenti ma anche rappresentare un boomerang se non supervisionate a dovere, tendono a incastrarsi e collidere facilmente e possono dar luogo a glitch di vario tipo.



Dinamiche dei personaggi



FK-Driven: Una caratteristica chiave del setup delle catene simulabili è che possono essere simulate sulla base di keyframes fatti a mano.

A tale scopo è stato usato un setup del tipo:

IK chain → FK driver chain (key poses) → FK spring chain

In questo modo si possono creare delle pose base a mano, ad esempio l'oscillazione del controllo padre della sciarpa, e su quello lanciare la dinamica di tutta la catena cinematica.

Così i loop sono facilmente animabili, mentre per le cutscenes, dove l'animazione è più complessa e non ripetitiva, si può simulare a partire da una serie di key-poses fatte usando tutti i controlli della sciarpa.

Rigging in Unreal Engine

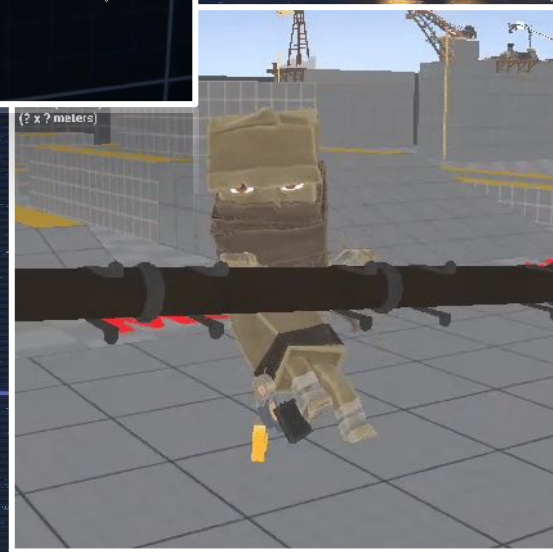
□ In Unreal abbiamo creato un Full-Body IK Rig per adattare gambe e braccia alle variazioni di terreno e appigli che si possono verificare rispetto alle più neutre condizioni di animazione.

In gioco il rig è stato usato:

- nella meccanica di climbing, per agganciare mani e piedi alle diverse pendenze degli appigli;
- nella locomotion di base, per orientare i piedi e le gambe rispetto alle asperità del terreno.
- Per orientare l'upper body rispetto alla camera in ogni momento.

Il rig di Slim in TTOW, originariamente e per una serie di fattori, discende da quello usato originariamente nel film di trash.

Per questo motivo, la gerarchia di joints di Slim, il protagonista, tutta sviluppata in Maya, non aveva particolare importanza dal punto di vista dell'ordine dei joints, fintanto che si restava nell'ambiente DCC di Maya.



Rigging in Unreal Engine



Quando abbiamo setuppato il Full-body IK control rig in Unreal, però, abbiamo sbattuto contro un muro: Unreal si aspettava un determinato ordine gerarchico dei joints, ma il personaggio era già stato largamente animato in maya e non potevamo rifare tutte le animazioni.

Il solo personaggio di Slim (gli altri furono creati già con questo accorgimento), all'interno contiene due gerarchie:

Quella iniziale, derivata dal film

- shoulder
 - twist 1
 - twist 2
 - elbow
 - twist 1
 - twist 2
 - wrist

Quella ottimale per unreal

- shoulder
 - elbow
 - wrist
 - twist 1
 - twist 2
 - twist 1
 - twist 2

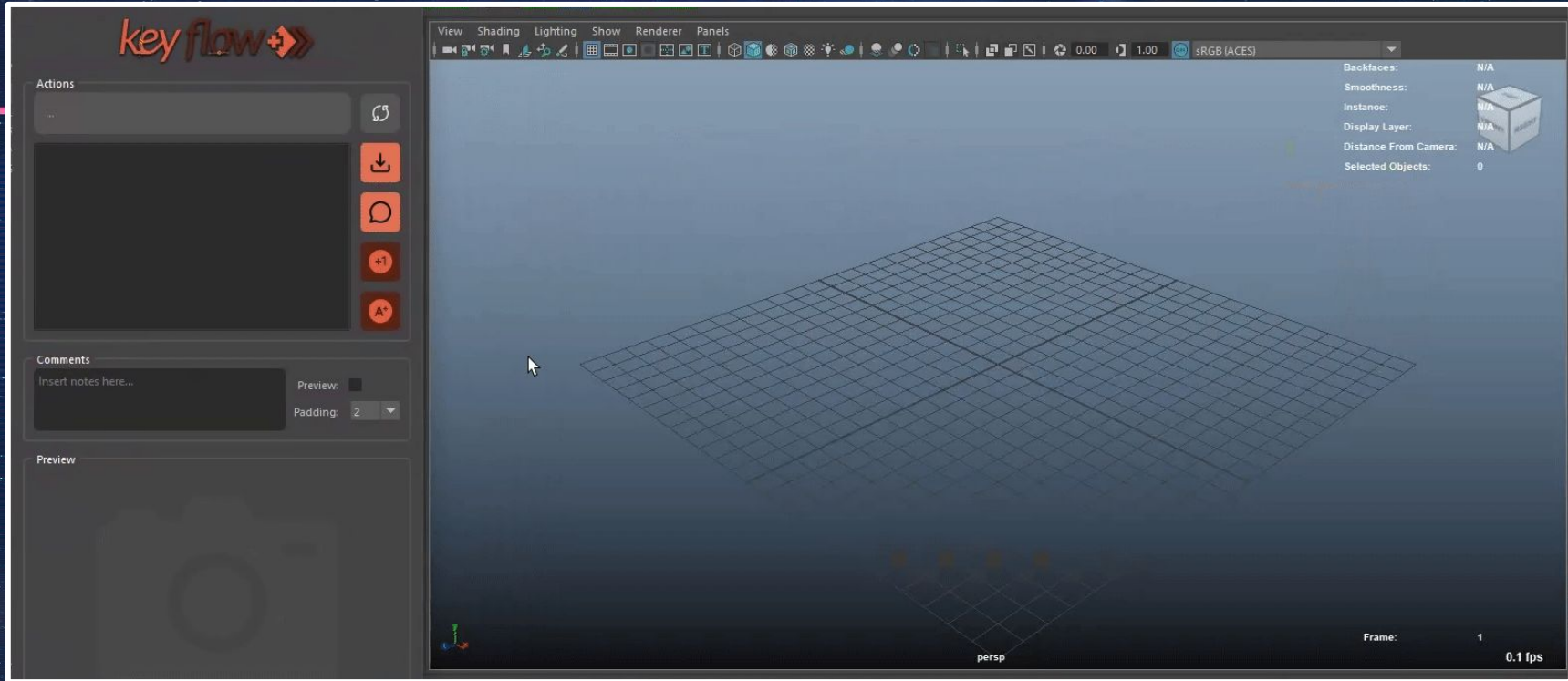
Quindi inizialmente avevamo tutti i joint tutti uno dopo l'altro mentre unreal vuole le 3 giunzioni ben separate del tipo BASE-MID-TIP e twist joints invece parentati nel MID e TIP joint. Con il primo setup Unreal ignorava i twist joints per come era impostata la gerarchia e costruiva il setup sulla triade shoulder - twist1 - twist2, rompendo le animazioni.



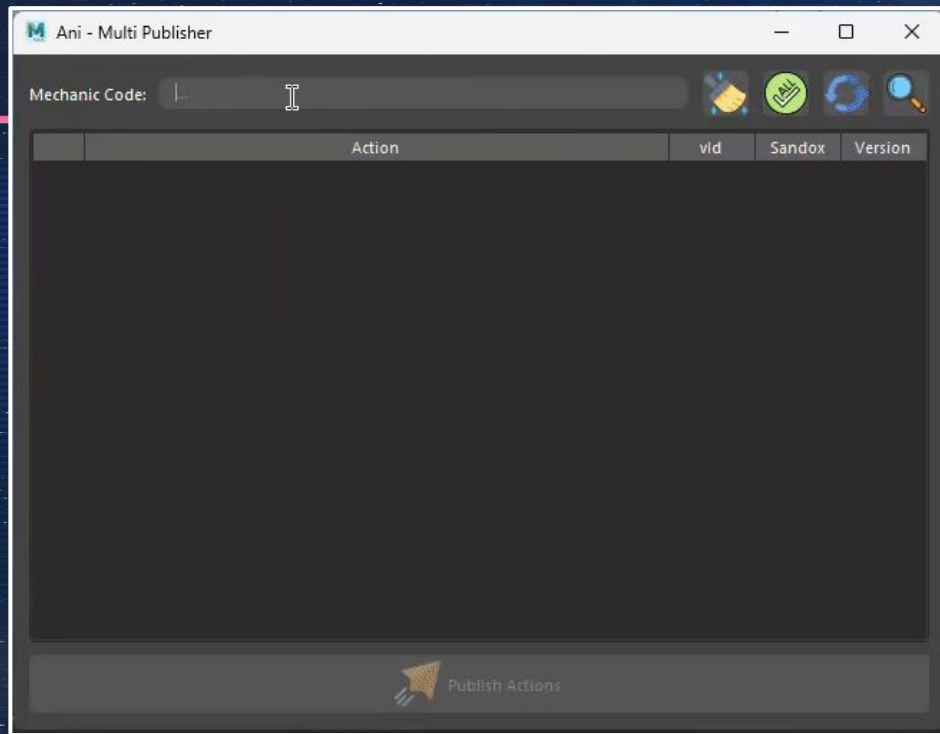
Pipeline

- KeyFlow, un simpatico Action Browser
- Animation Multi-Publisher
- Env Exporter per cinematiche

Keyflow, un simpatico Action Browser



Animation Multi-Publisher



At-a-glance solution: Per vari motivi può capitare che alcune modifiche fatte in corsa non vengano pubblicate per dimenticanza.

Il multi publisher è una soluzione a colpo d'occhio per avere un'overview dello stato di pubblicazione delle azioni nelle meccaniche rispetto alla versione di Sandbox.

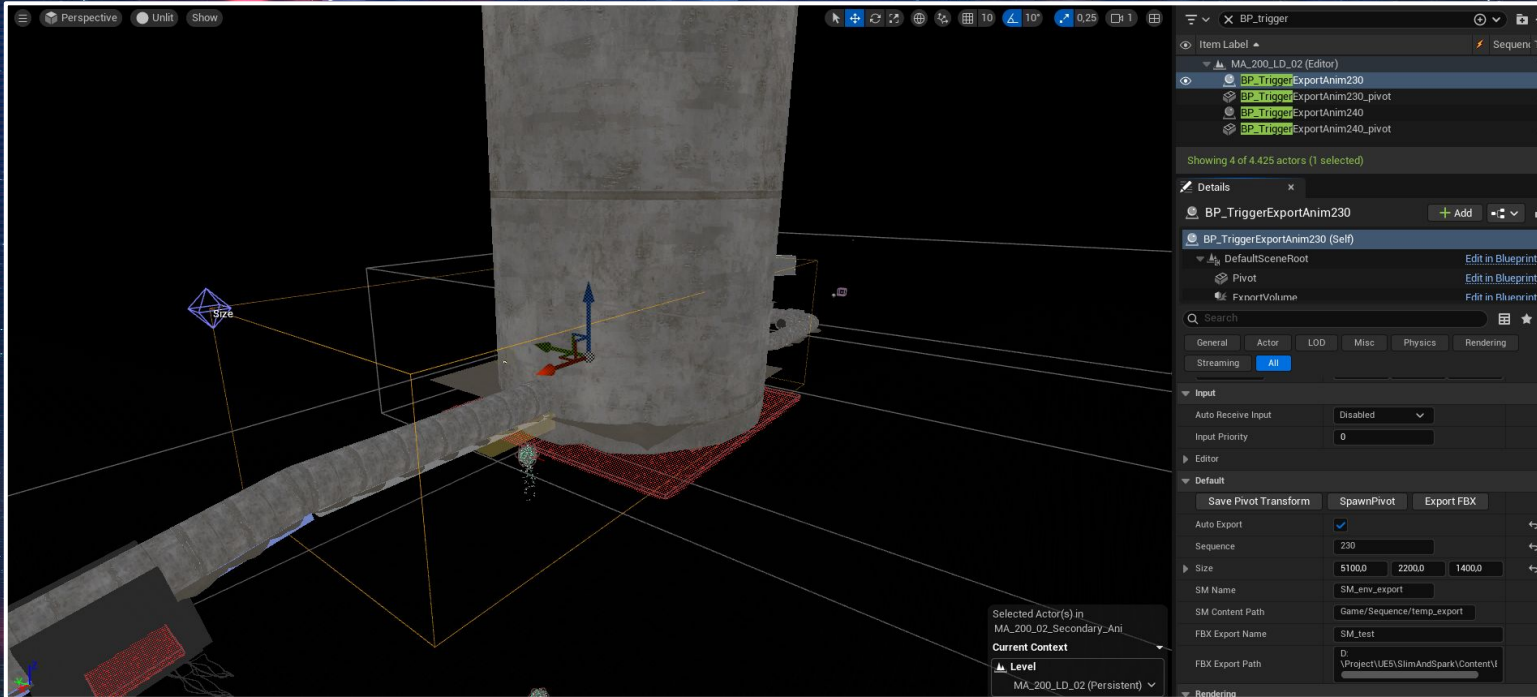
Avere le animazioni sempre aggiornate significa averle sempre pronte alla versione più recente per qualsiasi batch-tool di aggiornamento che venga lanciato dalla console di pipeline.

Funzionamento: Differenzia tra versione in sandbox e pubblicata, apertura scena in maya, salvataggio progressivo, pubblicazione scena salvata, export in fbx dell'animazione e propagazione degli aggiornamenti verso shotgrid e database collegati.

Env Exporter per cinematiche. UE <--> Maya

GAMES
FOR MOVIES
MOVIES
FOR GAMES

□ Gli environment sono delle entità in continua evoluzione a volte anche dopo il lock dei contenuti, rischiando di diventare un boomerang per animazione se non si è flessibili nel loro aggiornamento. La creazione di un tool di esportazione automatizzata ha permesso di velocizzare di molto il processo di esportazione e aggiornamento.



Env per cinematiche. UE <--> Maya

GAMES
FOR MOVIES
MOVIES
FOR GAMES



Il concetto centrale è avere un pivot di hookup in unreal che diventa lo zero del mondo in maya. Fatto questo, una volta che i personaggi sono animati in maya ed esportati, basterà in unreal fare l'attach delle skeletal mesh dentro il sequencer al pivot generato dal tool per gli environment e i personaggi snapperanno nel posto giusto.



Env per cinematiche. UE <--> Maya

☐ Esportare le coordinate di traslazione di tutti i pivot degli environment per le cinematiche in un database ci ha permesso di averli a disposizione quando necessario, anche tramite linea di comando. Anche lo script di importazione della cinematiche interroga il database e in automatico aggancia tutti i props e personaggi animati al valore di traslazione assoluto nella mappa, che però è relativo al BP_animTrigger che è stato creato per una sequenza.

```
P:\projects\SLIMESPARK\tools>cinepos -seq 230
Sequence: 230
Positions: [10981.180739661158, 1174.8214014990133, -484.4088509194621]
```

Save Pivot Transform		SpawnPivot		Export FBX	
Auto Export	☒				
Sequence	230				
▶ Size	5100,0	2200,0	1400,0		
SM Name	SM_env_export				
SM Content Path	Game/Sequence/temp_export				
FBX Export Name	SM_test				
FBX Export Path	D: \Project\UE5\SlimAndSpark\Content\{ _____				

Usando un trigger box per acquisire un pezzo di environment, abbiamo sempre una scena maya relativamente piccola e centrata sull'origine, invece di averla a migliaia di centimetri di distanza dall'origine (cioè con i valori assoluti della mappa di gioco), che rende la navigazione della viewport in maya molto difficile.

Tools Shelf

AL1 Animation Shelf è dove risiedono la maggior parte degli script che abbiamo implementato, vediamo quali sono:

Gameplay Tools

KeyFlow

Multi
Publisher

Mechanics
Swapper

Anim
Publisher

Quick
Exporter

Save
Progressive

Cutsenes Tools

Asset
Loader

Measure
Box

Cinematic
Shot
Opener

Environment
Updater

Shot
Initializer

Shot
Publisher

Cinematics
Swapper



Pre-Alpha

GAMES
FOR MOVIES
MOVIES
FOR GAMES

A1 A1
INTERACTIVE

THAT'S
ALL
FOLKS

GAMES
FOR MOVIES
MOVIES
FOR GAMES

A1 A1
INTERACTIVE