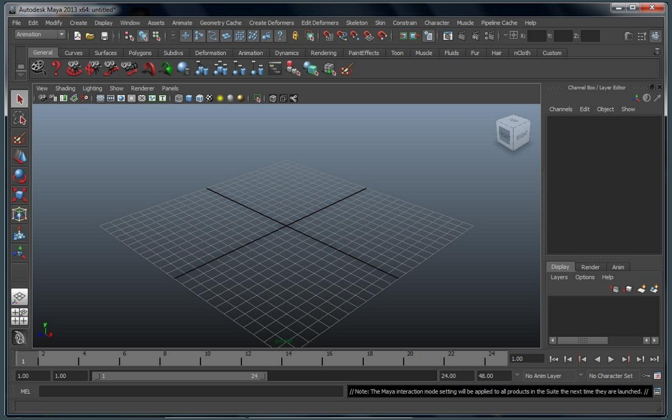
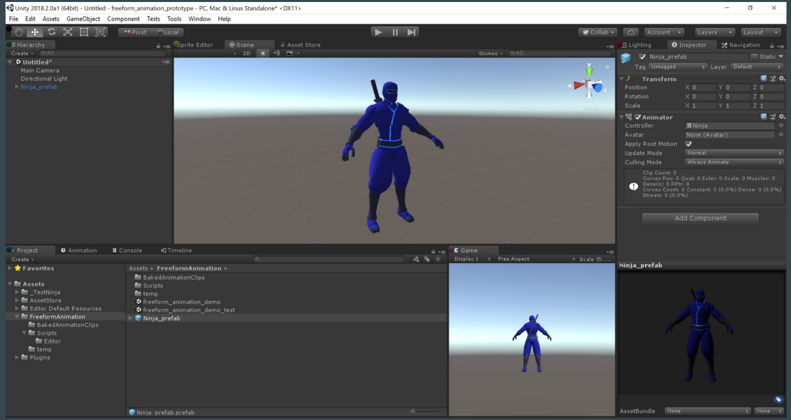
Real-Time Animation Workflows

UW Animation Capstone
Dave Hunt, 2018

DCC to Engine Pipeline

Export from DCC* (Maya) → Import to Engine (Unity)





Definitions:
*DCC = Digital Content Creation (tools)

DCC to Engine Pipeline

Export from DCC* (Maya) → Import to Engine (Unity)

- **Basic** (whole scene)
  - Export FBX from Maya
  - Open in Unity (and cross your fingers)

Definitions:
*DCC = Digital Content Creation (tools)

DCC to Engine Pipeline

Export from DCC* (Maya) → Import to Engine (Unity)

- **Basic** (whole scene)
  - Export FBX from Maya
  - Open in Unity (and cross your fingers)

(UW Summer Class VR video…)

Definitions:
*DCC = Digital Content Creation (tools)

DCC to Engine Pipeline

Export from DCC* (Maya) → Import to Engine (Unity)

- **Basic** (whole scene)
  - Export FBX from Maya
  - Open in Unity (and cross your fingers)
- **Advanced** (per-object)
  - Separate exports for Render Model and Skeletal Animations
  - Prepare the scene for export (pre-export script)
    - Delete everything except for animated skeletons
  - Export FBX (options)
  - Import into Unity and reassemble the scene (scripted)

Definitions:
*DCC = Digital Content Creation (tools)

Adapting Film Animation to Run-Time

Off-Line Rendered Film Animation

- Linear content
- Single, long animation clips
- Final result is usually one clip per-character, per-shot
- Layout/animatic animation is often multiple shots per sequence

Adapting Film Animation to Run-Time

Off-Line Rendered Film Animation -vs- Run-Time Animation

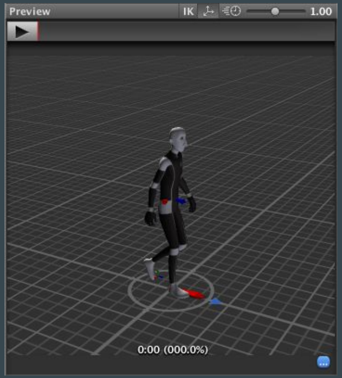
- Linear content
- Single, long animation clips
- Final result is usually one clip per-character, per-shot
- Layout/animatic animation is often multiple shots per sequence

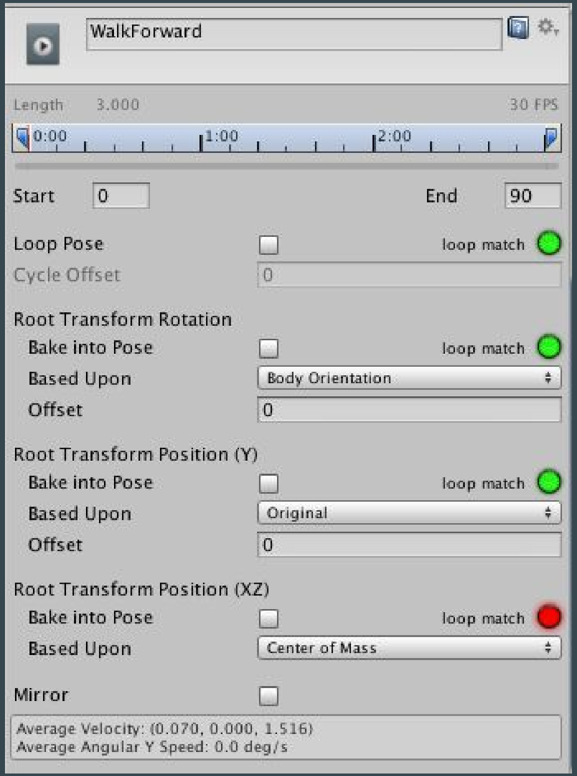
- Options for non-linear content
- Short clips and looping cycles
- Blending with layers: additive, overlay, blend-screens, etc.
- Inverse-Kinematics (IK)
- Often controlled by state machines
- Procedural motion: physics, math

Character Locomotion

Walk cycles can be moved in engine using...

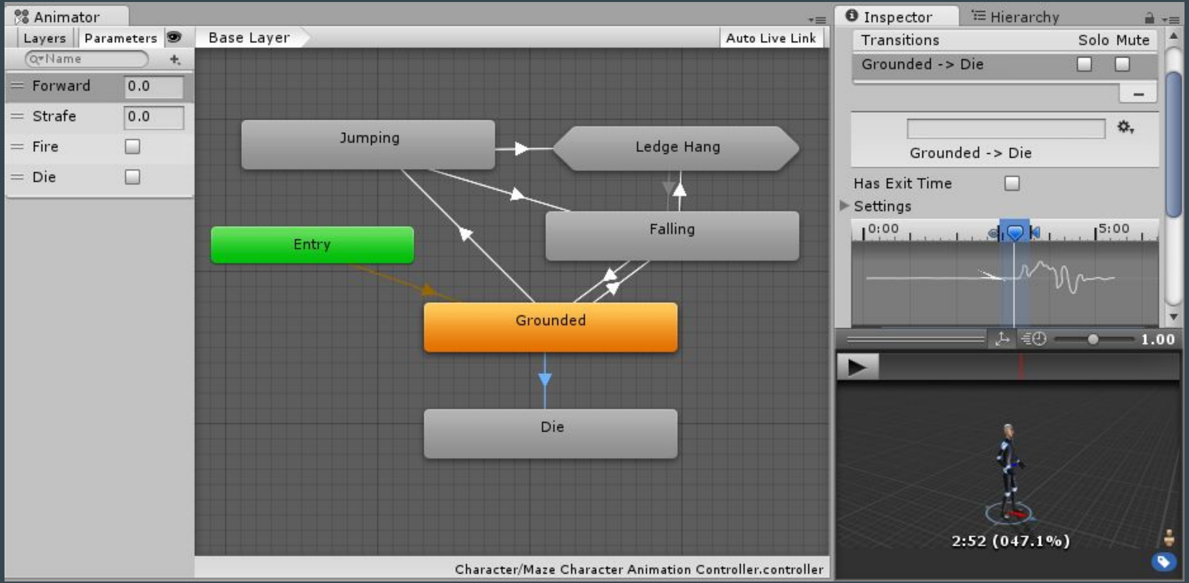
- Keyframed root motion
- Animated in place, moved by code
- Path animation
- Physics
- Motion Fields (advanced)





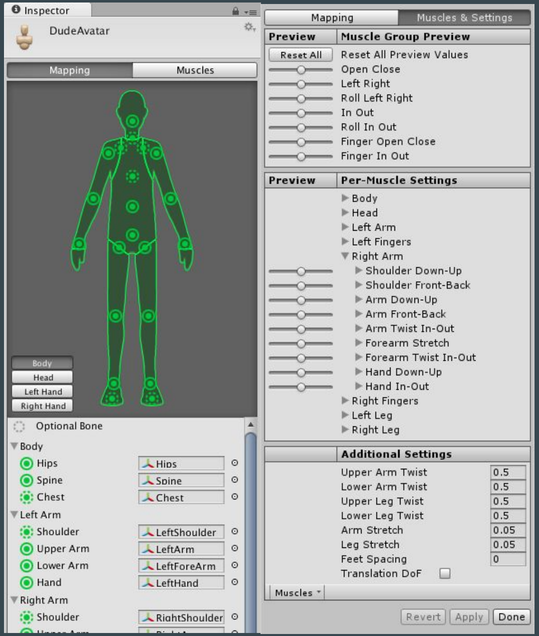
https://unity3d.com/learn/tutorials/topics/animation/authoring-root-motion?playlist=17099

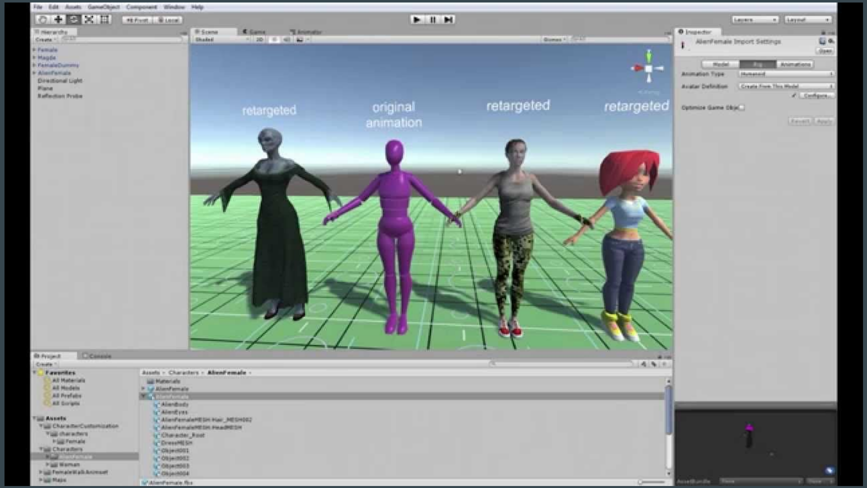
Animation State Machines



https://unity3d.com/learn/tutorials/topics/animation/animate-anything-mecanim

Run-Time Animation Retargeting



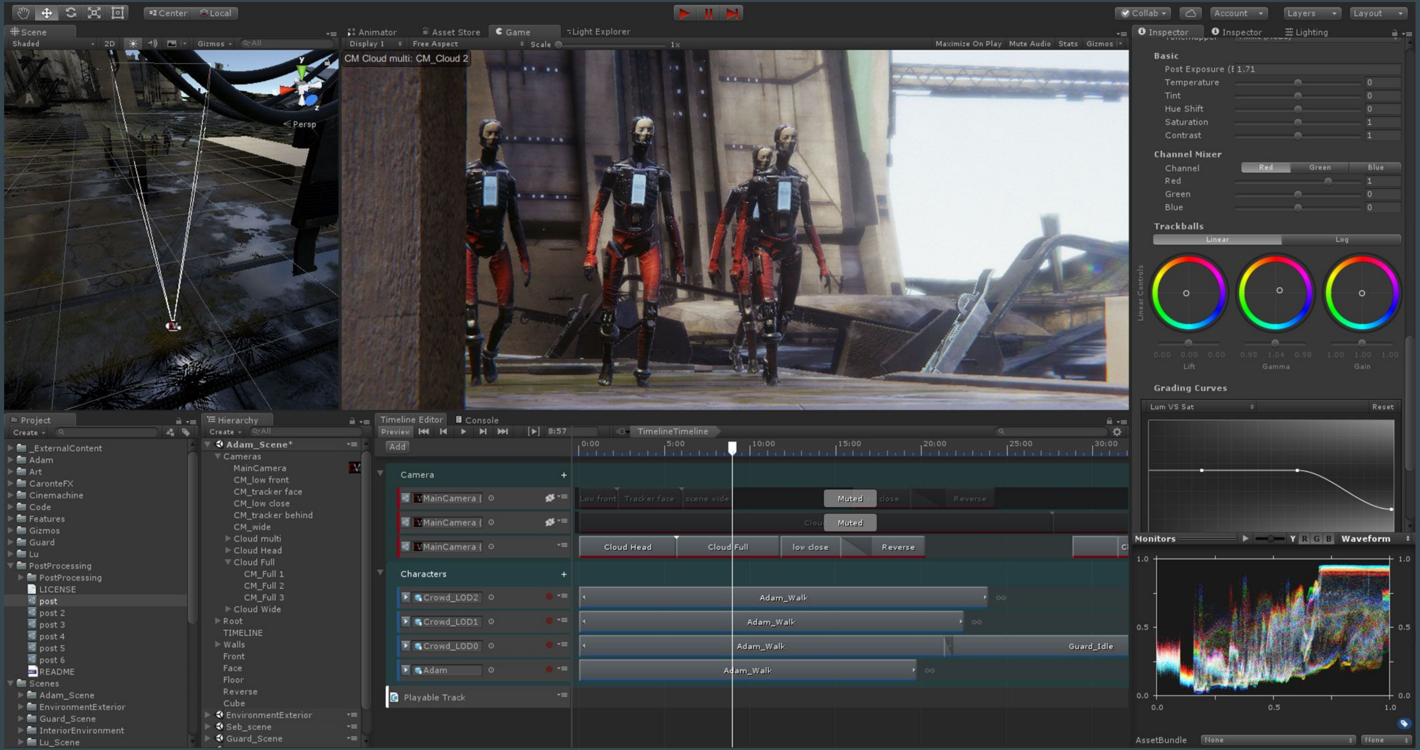


https://unity3d.com/learn/tutorials/topics/animation/humanoid-avatars

Run-Time Track Sequencing

Unity Timeline tool and Cinemachine

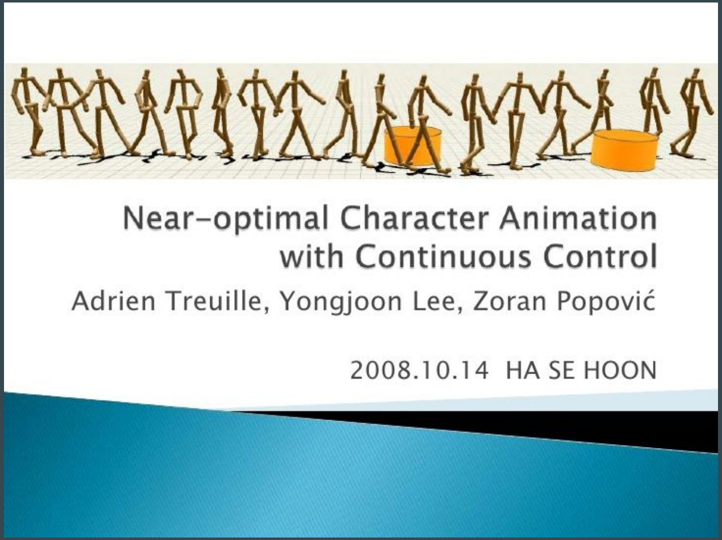
https://youtu.be/1WMMJWbXD6A



https://unity3d.com/learn/tutorials/topics/animation/using-timeline-overview?playlist=17099

Future of Run-Time Animation...

Motion Fields - UW Computer Science & Engineering



Similarity and neighborhoods Central to our definition of a motion field is the notion of the *similarity* between motion states. Given a motion state m, we compute a *neighborhood* $\mathcal{N}(m) = \{m_i\}_{i=1}^{k}$ of the k most similar motion states via a k-nearest neighbor query over the database [Mount and Arya 1997]. In our tests we use $k = 15$. We calculate the (dis-)similarity by:

$$d(m, m') = \sqrt{\begin{array}{rr} \beta_{\text{root}} \|v_{\text{root}} - v'_{\text{root}}\|^2 & + \\ \beta_0 \|q_0(\hat{u}) - q'_0(\hat{u})\|^2 & + \\ \sum_{i=1}^{n} \beta_i \|p_i(\hat{u}) - p'_i(\hat{u})\|^2 & + \\ \sum_{i=1}^{n} \beta_i \|(q_i p_i)(\hat{u}) - (q'_i p'_i)(\hat{u})\|^2 & + \end{array}} \quad (1)$$

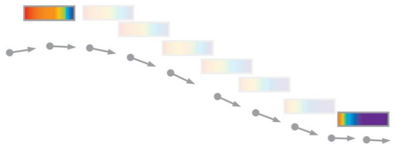


Figure 4: Value function with temporal compression. *The value functions at intermediate motion states are interpolated by the neighboring 'anchor' motion states that have explicitly stored value functions.*

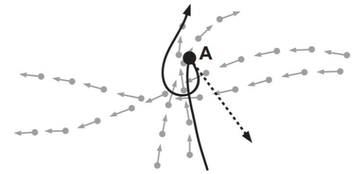


Figure 5: Responding to external perturbation. *When external force (dashed vector) is applied at state* **A** *causing a discontinuous change of behavior, the system can immediately find a new path around the motion fields to naturally recover from the impact.*

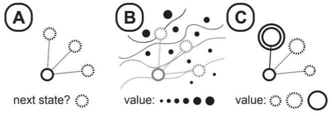


Figure 2: Action search using a value function. *(A) At every state, we have several possible actions (dashed lines) and their next states (dashed circles). (B) We interpolate the value function stored at database states (black points) to determine the value of each next state. (C) We select the highest value action to perform.*

http://grail.cs.washington.edu/projects/motion-fields/motion-fields.pdf

Motion Matching: UbiSoft's For Honor

http://www.gameanim.com/2016/05/03/motion-matching-ubisofts-honor/