Lecture 21 of 41

Animation Basics
Lab 4: Modeling & Rigging in Maya

William H. Hsu
Department of Computing and Information Sciences, KSU

Public mirror web site: http://www.kddresearch.org/Courses/CIS636
Instructor home page: http://www.cis.ksu.edu/~bhsu

Readings:

Today: Flash animation handout
Next class: Chapter 17, esp. §17.1 – 17.2, Eberly 2e – see http://bit.ly/ieUq45
Reference: http://www.learning-maya.com
Lecture Outline

- Reading for Last Class: §10.4, 12.7, Eberly 2e, Mesh handout
- Reading for Today: §11.1 – 11.6 Eberly 2e (736), Flash handout
- Reading for Next Class: §17.1 – 17.2, Eberly 2e
- Last Time: Curves & Surfaces
  - Piecewise polynomial curves (aka splines) and their properties
  - Hermite vs. Bézier curves: manipulation vs. display (rendering)
  - DeCasteljau’s algorithm: recursive linear interpolation
  - Other representations: Bernstein basis functions, matrix form
  - Bicubic surfaces
  - Bilinear interpolation
- Today: Maya & Animation Preliminaries – Ross Tutorials
  - Maya interface: navigation, menus, tools, primitives
- Next Class: Animations 2 – Rotations, Dynamics & Kinematics
# Where We Are

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Topic</th>
<th>Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Lab 4a: Animation Basics</td>
<td>Flash animation handout</td>
</tr>
<tr>
<td>22</td>
<td>Animation 2: Rotations, Dynamics, Kinematics</td>
<td>Chapter 17, esp. §17.1–17.2</td>
</tr>
<tr>
<td>23</td>
<td>Demos 4: Modeling &amp; Simulation, Rotations</td>
<td>Chapter 10, 13, §17.3–17.5</td>
</tr>
<tr>
<td>24</td>
<td>Collisions 1: axes, OBBS, Lab 4b</td>
<td>§2.4.3, 8.1, GL handout</td>
</tr>
<tr>
<td>25</td>
<td>Spatial Sorting: Binary Space Partitioning</td>
<td>Chapter 6, esp. §6.1</td>
</tr>
<tr>
<td>26</td>
<td>Demos 5: More CGA: Picking, HW/Exam</td>
<td>Chapter 7; §8.4</td>
</tr>
<tr>
<td>27</td>
<td>Lab 5a: Interaction Handling</td>
<td>§8.3–8.4; 4.2, 5.0, 5.6, 9.1</td>
</tr>
<tr>
<td>28</td>
<td>Collisions 2: Dynamic, Particle Systems</td>
<td>§9.1, particle system handout</td>
</tr>
<tr>
<td>29</td>
<td>Exam 2 review: Hour Exam 2 (evening)</td>
<td>Chapters 5–6, 7–8, 12, 17</td>
</tr>
<tr>
<td>30</td>
<td>Lab 5b: Particle Systems</td>
<td>Particle system handout</td>
</tr>
<tr>
<td>31</td>
<td>Animation 3: Control &amp; IK</td>
<td>§8.3, CGA handout</td>
</tr>
<tr>
<td>32</td>
<td>Ray Tracing 1: intersections, ray trees</td>
<td>Chapter 14</td>
</tr>
<tr>
<td>33</td>
<td>Lab 6a: Ray Tracing Basics with POV-Ray</td>
<td>RT handout</td>
</tr>
<tr>
<td>34</td>
<td>Ray Tracing 2: advanced topic survey</td>
<td>Chapter 15, RT handout</td>
</tr>
<tr>
<td>35</td>
<td>Visualization 1: Data (Quantities &amp; Evidence)</td>
<td>Tuffe handout (1)</td>
</tr>
<tr>
<td>36</td>
<td>Lab 6b: More Ray Tracing</td>
<td>RT handout</td>
</tr>
<tr>
<td>37</td>
<td>Visualization 2: Objects</td>
<td>Tuffe handout (2 &amp; 4)</td>
</tr>
<tr>
<td>38</td>
<td>Color Basics; Term Project Prep</td>
<td>Color handout</td>
</tr>
<tr>
<td>39</td>
<td>Lab 7: Fractals &amp; Terrain Generation</td>
<td>Fractals/Terrain handout</td>
</tr>
<tr>
<td>40</td>
<td>Visualization 3: Processes; Final Review 1</td>
<td>Tuffe handout (3)</td>
</tr>
<tr>
<td>41</td>
<td>Project presentations 1: Final Review 2</td>
<td>–</td>
</tr>
<tr>
<td>42</td>
<td>Project presentations 2</td>
<td>–</td>
</tr>
<tr>
<td>43</td>
<td>Final Exam</td>
<td>Ch. 1 – 8, 10 – 15, 17, 20</td>
</tr>
</tbody>
</table>

Lightly-shaded entries denote the due date of a written problem set, heavily-shaded entries, that of a machine problem (programming assignment), blue-shaded entries, that of a paper review, and the green-shaded entry, that of the term project.

Green, blue and red letters denote exam review, exam, and exam solution review dates.
Acknowledgements:
Curves & Surfaces

Steve Rotenberg
Visiting Lecturer
Graphics Lab
University of California – San Diego
CEO/Chief Scientist, PixelActive
http://graphics.ucsd.edu

Barry McCaul
Lecturer
School of Computing
Dublin City University
http://www.computing.dcu.ie/~bmccaul/

Ken Hawick
Professor
Institute of Information and Mathematical Sciences (IIMS)
Massey University – Albany
http://www.massey.ac.nz/~kahawick/
Acknowledgements:
Splines

Jim Foley
Professor, College of Computing &
Stephen Fleming Chair in
Telecommunications
Georgia Institute of Technology

Andy van Dam
T. J. Watson University Professor of
Technology and Education &
Professor of Computer Science
Brown University

Steve Feiner
Professor of Computer Science &
Director, Computer Graphics and User
Interfaces Laboratory
Columbia University

John F. Hughes
Associate Professor of Computer
Science
Brown University

James D. Foley
Georgia Tech

Andries van Dam
Brown University
http://www.cs.brown.edu/~avd/

Steven K. Feiner
Columbia University
http://www.cs.columbia.edu/~feiner/

John F. Hughes
Brown University
http://www.cs.brown.edu/~jfh/
Review [1]: Vector Polynomials (Curves)

- Linear: \( f(t) = at + b \)
- Quadratic: \( f(t) = at^2 + bt + c \)
- Cubic: \( f(t) = at^3 + bt^2 + ct + d \)

We usually define the curve for \( 0 \leq t \leq 1 \)
Review [2]:
Linear Interpolation

- Linear interpolation (Lerp) is a common technique for generating a new value that is somewhere in between two other values.
- A ‘value’ could be a number, vector, color, or even something more complex like an entire 3D object...
- Consider interpolating between two points a and b by some parameter t.

\[ \text{Lerp}(t, a, b) = (1 - t)a + tb \]

Adapted from slides © 2003 – 2006 S. Rotenberg, UCSD
Review [3]: Hermite Curves

- Polylines are linear (1st order polynomial) interpolations between points
  - Given points \( P \) and \( Q \), line between the two is given by the parametric equation:
    \[ x(t) = (1-t)P + tQ, \quad 0 \leq t \leq 1 \]
  - \((1-t)\) and \( t \) are called weighting functions of \( P \) and \( Q \)
- Splines are higher order polynomial interpolations between points
  - Like linear interpolation but with higher order weighting functions allowing better approximations/smoother curves
- One representation - Hermite curves (interpolating spline):
  - Determined by two control points \( P \) and \( Q \), an initial tangent vector \( v \) and a final tangent vector \( w \).

\[
\begin{align*}
  y(t) &= (2t^3 - 3t^2 + 1)P + (-2t^3 + 3t^2)Q \\
  &\quad + (t^3 - 2t^2 + t)v + (t^3 - t^2)w
\end{align*}
\]

- Satisfies:
  - \( y(0) = P \)
  - \( y(1) = Q \)
  - \( y'(0) = v \)
  - \( y'(1) = w \)

Adapted from slides © 2010 van Dam et al., Brown University
Review [4]: Bézier Curves

- Bezier representation is similar to Hermite
  - 4 points instead of 2 points and 2 vectors ($P_1 \ldots P_4$)
  - Initial position $P_1$, tangent vector is $P_2 - P_1$
  - Final position $P_4$ tangent vector is $P_4 - P_3$
  - This representation allows a spline to be stored as a list of vertices with some global parameters that describe the smoothness and continuity

- Bezier splines are widely used (Adobe, Microsoft) for font definition


Adapted from slides © 2010 van Dam et al., Brown University http://bit.ly/hiSt0f Reused with permission.
Review [5]: De Casteljau’s Algorithm

\[ x = \text{Lerp}(t, r_0, r_1) \]
\[ r_0 = \text{Lerp}(t, q_0, q_1) \]
\[ q_0 = \text{Lerp}(t, p_0, p_1) \]
\[ r_1 = \text{Lerp}(t, q_1, q_2) \]
\[ q_1 = \text{Lerp}(t, p_1, p_2) \]
\[ q_2 = \text{Lerp}(t, p_2, p_3) \]
\[ p_3 \]

Adapted from slides © 2003 – 2006 S. Rotenberg, UCSD
Review [6]: Bernstein Polynomials – Matrix Form

\[ x = at^3 + bt^2 + ct + d \]

\[ a = (-p_0 + 3p_1 - 3p_2 + p_3) \]
\[ b = (3p_0 - 6p_1 + 3p_2) \]
\[ c = (-3p_0 + 3p_1) \]
\[ d = (p_0) \]

\[ x = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \]
Review [7]: \( G^i \) vs. \( C^i \) Continuity

- **Geometric Continuity:** \( G^i \)
  - Guarantees that direction of \( i^{th} \) derivative equal
  - \( G^0 \): curves touch at join point
  - \( G^1 \): curves also share common tangent direction at join point
  - \( G^2 \): curves also share common center of curvature at join point

- **Mathematical Continuity:** \( C^i \)
  - Guarantees that direction, magnitude of \( i^{th} \) derivative equal
  - \( C^0 \equiv G^0 \): curves touch at join point
  - \( C^1 \): curves share common tangent direction / magnitude at join point
  - \( C^2 \): curves share common second derivative at join point

Review [8]:
Parametric Bicubic Surfaces

- **Parametric Bicubic Surface**: Generalization of Parametric Cubic Curve
  
  \[ P(u, v) = [x(u, v), y(u, v), z(u, v)] \quad 0 \leq u \leq 1 \quad 0 \leq v \leq 1 \]

- **From Curves to Surfaces**
  
  - Let one parameter (say \( v \)) be held at constant value
  - Above will represent a curve
  - Surface generated by sweeping all points on boundary curve, e.g., \( P(u, 0) \), through cubic trajectories, defined using \( v \), to boundary curve \( P(u, 1) \)
Review [9]:
Curves & Surfaces

- **Curves**
  - Bézier: easier to scan convert (DeCasteljau)
  - Hermite: easier to control via GUI (tangent)

- **Bicubic patches**
  - Bilinear interpolation
  - Control patch aka Coons patch

- Acknowledgments - thanks to Eric McKenzie, Edinburgh, from whose Graphics Course some of these slides were adapted.

---

Sinbad: Legend of the Seven Seas
© 2003 Dreamworks, SKG
Trailer: [http://youtu.be/1KCX0pFPRwk](http://youtu.be/1KCX0pFPRwk)
Eris scene: [http://youtu.be/w1r8_vByXW4](http://youtu.be/w1r8_vByXW4)
Acknowledgements:
Maya Character Rigging

Aaron Ross
Founder, Digital Arts Guild
http://dr-yo.com
http://www.youtube.com/user/DigitalArtsGuild

Jim Lammers
President
Trinity Animation
http://www.trinity3d.com

Larry Neuberger
Associate Professor, Alfred State SUNY College of Technology
Online Instructor, Art Institute of Pittsburgh
http://poorhousefx.com
Resources [1]:
Basic Maya Tutorials – Ross
Resources [2]: Animation Tutorials – Lammers

Resources [3]: Examples Online

"Maya Animation" at Animation Arena © 2004 – 2011 G. Nakpil, Toronto, CANADA
http://bit.ly/gXXQTG

Student art gallery for Maya 4 Fundamentals (http://amzn.to/eOld3Q)
Lab 4 [1]:
Rigging “Tin Can Man”, Unreal Wiki

© 2003 – 2008 Unreal Wiki
Lab 4 [2]:
Part A – Modeling

http://bit.ly/h9IRmT
Lab 4 [3] : Part B - Rigging

Character Modeling in Maya [1]: Muscle Models & Deformations

Fig. 1.

Fig. 2.

Fig. 3.

Alfred State College, State University of New York
Character Modeling in Maya [2]:
Deform ★ Blend Shape

Fig. 4.

Fig. 5.

Fig. 6.

Alfred State College, State University of New York
Character Modeling in Maya [3]:
Animate ● Set Driven Key ● Set

Fig. 7.
Fig. 8.
Fig. 9.
Fig. 10.

Alfred State College, State University of New York
Character Modeling in Maya [4]: Driver

Fig. 11.  

Fig. 12.  

Fig. 13.  

Fig. 14.

Alfred State College, State University of New York
Character Modeling in Maya [5]: Blend Shape Deformation Setup

The **driver** is the elbow. This is saying that whenever the elbow joint rotates around the Y-axis, the arm deformation will take place.

The **driven** is the blend shape. This is what will be deformed when the driver.

We have the window to the left set up saying that when the elbow joint rotates around the Y-axis, the rightArmFlex blend shape will deform to my specifications.
Character Modeling in Maya [6]: Inverse Kinematics (IK)

Fig. 16.

Fig. 17.

Fig. 18.

Alfred State College, State University of New York
Character Modeling in Maya [7]: Controlling Deformation & Rotation

Fig. 19.

Fig. 20.

Fig. 21.

Alfred State College, State University of New York
Cloth Modeling in Maya [1]:
More Driven Keys & Blend Shape

© 2002 - 2003 PlanetSack Tutorials
http://bit.ly/hocnu1
Cloth Modeling in Maya [2]: Output

That’s it! Now you just have to repeat steps 6 - 8 for all joints that will cause wrinkles in the clothing. Finally, the finished effect (Quicktime, double-click to play):

You can see how driven keys and BlendShape nodes can really enhance your character setup. You could also use this technique to create other effects like bulging muscles. The possibilities are endless!
Summary

- Reading for Next Class: §17.1 – 17.2, Eberly 2e
- Last Time: Curves & Surfaces
  - Piecewise linear, quadratic, cubic curves and their properties
  - Interpolation: subdivision (DeCasteljau’s algorithm)
  - Bicubic surfaces & bilinear interpolation
  - Maya interface: navigation, menus, tools, primitives
  - GUI & objects (Ross 1); viewports, transforms, & hotkeys (Ross 2)
  - Nodes & attributes (Ross 3); UI, channel box & deformers (Ross 4)
  - Modeling, scene creation, materials (Ross 5)
- Previous Videos (#3): Morphing & Other Special Effects (SFX)
- Next Set of Videos (#4): Modeling & Simulation
- Next Class: Animations 2 – Rotations, Dynamics & Kinematics
Terminology

- **Piecewise Polynomial Curves** *aka* Splines
- **Continuity:** Geometric (G<sup>i</sup>), Mathematical (C<sup>i</sup>)
- **Bicubic Surfaces** including NURBS Surfaces
- **Maya Software** for 3-D Modeling & Animation
  - **Shelves** – groups of tools & action icons; compare palettes, toolbars
  - **Hotkeys** – key combos for common functions; compare macros
  - **Viewports** – scene views for editing: orthographic, perspective
  - **Channel box** – GUI for accessing position, rotation, scale, history
  - **Deformers** – tools for controlling complex vertex meshes
- **Rigging Character Models:** Defining Components of Articulated Figure
  - **Joints** – axis of rotation, angular *degree(s)* of freedom (DOFs)
  - **Bones** – attached to joints, rotate about joint axis