Animation

Animation is the process of giving motion to your geometric models. Before animating, there are questions you need to ask first:

- Why am I doing this?
- Do I want the animation to obey the real laws of physics? Partially? Which elements?
- Am I willing to "fake" the physics to get the objects to move in a way that I tell it?
- Do I have specific key positions I want the objects to pass through no matter what?
- Do I want to simply record the motion of a real person, animal, etc., and then play it back?

Keyframe Animation

These icons refer to explanatory videos on the class web site.

For my own work, instead of Key Frames, I like specifying Key Times better. And, so, I created a C++ class to do it for you.

Here's Some Code that Lets You Create DIY Keyframe Animations

```cpp
class Keytimes:
    void AddTimeValue( float time, float value );
    float GetFirstTime( );
    float GetLastTime( );
    int GetNumKeytimes( );
    float GetValue( float time );
    void Init( );
    void PrintTimeValues( );

for my own work, instead of Key Frames, I like specifying Key Times better.
And, so, I created a C++ class to do it for you.

Instead of Key Frames, I Like Specifying Key Times Better

Keytimes Xpos,

Here’s Some Code that Lets You Create DIY Keyframe Animations

For my own work, instead of Key Frames, I like specifying Key Times better. And, so, I created a C++ class to do it for you.

Here’s Some Code that Lets You Create DIY Keyframe Animations

For my own work, instead of Key Frames, I like specifying Key Times better. And, so, I created a C++ class to do it for you.

Here’s Some Code that Lets You Create DIY Keyframe Animations

For my own work, instead of Key Frames, I like specifying Key Times better. And, so, I created a C++ class to do it for you.

Here’s Some Code that Lets You Create DIY Keyframe Animations

For my own work, instead of Key Frames, I like specifying Key Times better. And, so, I created a C++ class to do it for you.

Here’s Some Code that Lets You Create DIY Keyframe Animations

For my own work, instead of Key Frames, I like specifying Key Times better. And, so, I created a C++ class to do it for you.
Instead of Key Frames, I Like Specifying Key Times Better

<table>
<thead>
<tr>
<th>Time</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>(0.00, 0.00)</td>
</tr>
<tr>
<td>0.000</td>
<td>(1.00, 0.142)</td>
</tr>
<tr>
<td>0.100</td>
<td>(0.00, 0.232)</td>
</tr>
<tr>
<td>0.500</td>
<td>(0.50, 0.2718)</td>
</tr>
<tr>
<td>1.000</td>
<td>(1.00, 3.142)</td>
</tr>
</tbody>
</table>

Using the System Clock in Display() for Timing

```
#define MSEC 10000 // i.e., 10 seconds

Keytimes Xpos, Ypos, Zpos;
Keytimes ThetaX, ThetaY, ThetaZ;
.
.
if( AnimationIsOn )
{
    // # msec into the cycle (0 - MSEC-1):
    int msec = glutGet( GLUT_ELAPSED_TIME ) % MSEC;
    // turn that into a time in seconds:
    float nowTime = (float)msec / 1000.;
    glPushMatrix( );
    glTranslatef( Xpos.GetValue( nowTime ), Ypos.GetValue( nowTime ), Zpos.GetValue( nowTime ) );
    glRotatef( ThetaX.GetValue( nowTime ), 1., 0., 0. );
    glRotatef( ThetaY.GetValue( nowTime ), 0., 1., 0. );
    glRotatef( ThetaZ.GetValue( nowTime ), 0., 0., 1. );
    // draw the object
    glPopMatrix( );
}
```

Forward Kinematics: Change Parameters – Connected Things Move
(All children understand this)

Inverse Kinematics (IK): Things Need to Move to a Particular Location – What Parameters Will Make Them Do That?

Of course, there will always be target locations that can never be reached. Think about that spot in the middle of your back that you can never scratch!
Particle Systems:
A Cross Between Modeling and Animation?

The basic process is:

1. Emit
2. Random Number Generator
3. Display
4. Update

Particle Systems Examples

The Lion King (2019) -- Disney
A Particle System to Simulate Colliding Galaxies in Cosmic Voyage

Particles Don’t Actually Have to Be “Particles”

Multiple Animation Techniques Can Be Combined

Animating using Rigid-body Physics

Newton's second law:
\[ \text{force} = \text{mass} \times \text{acceleration} \]
or
\[ \ddot{x} = \text{acceleration} = \frac{\text{force}}{\text{mass}} \]

The Lion King
Mulan
Avatar

In order to make this work, you need to supply physical properties such as mass, center of mass, moment of inertia, coefficients of friction, coefficients of restitution, etc.

Animating using Fluid Physics

\[ F = k \left( D - D_0 \right) \]

D₀ = unloaded spring length
\[ \frac{(D - D_0)}{k} = \frac{F}{k} \]
k = spring stiffness in Newtons/meter or pounds/inch

Or, if you know the displacement, the force exerted by the spring is:

This is known as Hooke’s Law
Animating using the Physics of a Mesh of Springs

Simulating a Bouncy String

Placing a Physical Barrier in the Scene

Animating Cloth

Cloth Examples
**Functional Animation:**

Make the Object Want to Move Towards a Goal Position

\[ m\ddot{x} + c\dot{x} + kx = 0 \]

While Making it Want to Move Away from all other Objects

\[ m\ddot{x} = \sum F \text{ repulsive} \]

**Total Goal – Make the Free Body Move Towards its Final Position While Being Repelled by the Other Bodies**

\[ m\ddot{x} + c\dot{x} + kx = \sum F \]

Increasing the Stiffness

Stiffness = 9

Stiffness = 3

Stiffness = 6
Increasing the Repulsion Coefficient

Repulse = 10
Repulse = 30

Motion Capture ("MoCap") as an Input for Animation

Even Animals can be MoCapped

My cats would never have put up with this...

https://www.youtube.com/watch?v=cxyzq_lQr9p0

Motion Capture is for Hands and Faces Too

Natural Point

Even Animals can be MoCapped

They probably should have used physics, but didn’t

Tron I –

MAGI
Card Trick

Pixar Animated Shorts

Pixar