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?
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.

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

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

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

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

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

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

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

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

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

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

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

```
4 time-value pairs
Time runs from 0.000 to 2.000

0.000 0.000
0.100 0.232
0.200 0.806
0.300 1.538
0.400 2.334
0.500 2.718
0.600 2.989
0.700 3.170
0.800 3.258
0.900 3.250
1.000 3.142
1.100 2.935
1.200 2.646
1.300 2.302
1.400 1.924
1.500 1.539
1.600 1.169
1.700 0.840
1.800 0.574
1.900 0.397
2.000 0.333
```
Forward Kinematics: Change Parameters – Connected Things Move (All children understand this)

Forward Kinematics: Transformation Hierarchies

Determine Object Locations?

Ground

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:

- Emit
- Random Number Generator
- Display
- Update

Particle Systems Examples

Particle Systems Examples

Chuck Evans

particles.mp4
Particle Systems Examples

The Lion King (2019) – Disney

A Particle System to Simulate Colliding Galaxies in Cosmic Voyage

Cosmic Voyage

Particles Don’t Actually Have to Be “Particles”

Avatar

The Lion King

Mulan
Multiple Animation Techniques Can Be Combined

A Particle System coming from a moving keyframed object in Blender:

Animating using Rigid-body Physics

Newton's second law:

\[ \text{force} = \text{mass} \times \text{acceleration} \]

or

\[ \ddot{x} = \frac{\text{force}}{\text{mass}} \]

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.

Animation using Fluid Physics

\[ x(T) = \int_{T=0}^{T=T} \ddot{x} dt \approx \sum \ddot{x} \Delta t \]

Animating using Physics

\[ k = \text{spring stiffness} \]

\[ (D - D_0) = \frac{F}{k} \]

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

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

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

Simulating a Bouncy String

Simulating a Bouncy String

Placing a Physical Barrier in the Scene
Functional Animation:
Make the Object Want to Move Towards a Goal Position

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

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

Functional Animation

Functional Animation

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

Motion Capture is for Hands and Faces Too
Even Animals can be MoCapped

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

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

Tron I —
They probably should have used physics, but didn’t

Card Trick

Pixar Animated Shorts

Rob Russ

Pixar