Simple Keytime Animation for CS 450/550

Approaches to Animation

Keyframing

The General Idea is to Interpolate the In-between Frames from the Smooth Curves Fit through the Key Frames

Keyframing involves creating certain key positions for the objects in the scene, and then the program later interpolating the animation frames in between the key frames.

In hand-drawn animation, the key frames are created by the senior animators, and the in-between frames are developed by the junior animators.

In our case, you are going to be the senior animator, and the computer will do the in-betweening.
Many Professional Animation Packages Make You Sculpt the Slopes (but we won’t . . .)

Blender:

The “Y vs. Frame” Curve Looks Like This

Do This Same Thing for the X, Y, and Z Translations and the X, Y, and Z Rotations

Instead of Key Frames, I Like Specifying Key Times Better

We created a C++ class to do the interpolation for you

```cpp
class Keytimes{
  public:
    void AddTimeValue(float time, float value);
    float GetFirstTime();
    float GetLastTime();
    int GetNumKeytimes();
    float GetValue(float time);
    void Init();
    void PrintTimeValues();
};
```
Instead of Key Frames, I Like Specifying Key Times Better

```c
int main( int argc, char *argv[] )
{
    Xpos.Init( );
    Xpos.AddTimeValue( 0.0, 0.000 );
    Xpos.AddTimeValue( 2.0, 0.333 );
    Xpos.AddTimeValue( 1.0, 3.142 );
    Xpos.AddTimeValue( 0.5, 2.718 );
    fprintf( stderr, "%d time-value pairs:\n", Xpos.GetNumKeytimes( ) );
    Xpos.PrintTimeValues( );
    fprintf( stderr, "Time runs from %8.3f to %8.3f\n", Xpos.GetFirstTime( ), Xpos.GetLastTime( ) );
    for( float t = 0.f; t <= 2.f; t += 0.1f )
    {
        float v = Xpos.GetValue( t );
        fprintf( stderr, "%8.3f\t%8.3f\n", t, v );
    }
}
```

Instead of Key Frames, I Like Specifying Key Times Better

Using the System Clock in Display( ) for Timing

```c
#define MSEC 10000 // i.e., 10 seconds
Keytimes Xpos, Ypos, Zpos;
Keytimes ThetaX, ThetaY, ThetaZ;

// in InitGraphics( ):
<< init the Keytime classes and add the keyframe values >>
.
.
.
// in Display( ):
// # msec into the cycle ( 0 - MSEC-1 ):
int msec = glutGet( GLUT_ELAPSED_TIME ) % MSEC;
// turn that into a time in seconds:
float nowSecs = (float)msec / 1000.f;
gPushMatrix( );
gTranslatef( Xpos.GetValue(nowSecs), Ypos.GetValue(nowSecs), Zpos.GetValue(nowSecs) );
gRotatef( ThetaX.GetValue(nowSecs), 1., 0., 0. );
gRotatef( ThetaY.GetValue(nowSecs), 0., 1., 0. );
gRotatef( ThetaZ.GetValue(nowSecs), 0., 0., 1. );
<< draw the object >>
gPopMatrix( );
```

Using the System Clock in Display( ) for Timing