1

# Animation Effects using a Timer


Oregon State University

Mike Bailey

mjb@cs.oregonstate.edu






2

## Using Timers with Shaders

*glman* has a built-in Timer variable. You just need to declare it:

**uniform float Timer;**

Then, just use it in your code.
It goes from 0. to 1. in 10 seconds, and then instantly back to 0.

Or, you can program a Timer yourself in your .cpp program:

```
float Timer;                                    // global variable
const  int  MS_PER_CYCLE  =  10*1000;           // 10,000 ms = 10 seconds
. . .
void
Animate( )
{
        int ms = glutGet( GLUT_ELAPSED_TIME );
        ms %= MS_PER_CYCLE;
        Timer = (float)ms / (float)MS_PER_CYCLE;    // 0. to 1. in 10 seconds
        glutSetWindow( MainWindow );
        glutPostRedisplay( );
}

void
InitGraphics( )
{
. . .
        glutIdleFunc( Animate );
}
```

## Fun With Zero-to-One:
## There are many ways to map 0.→1. to a different function

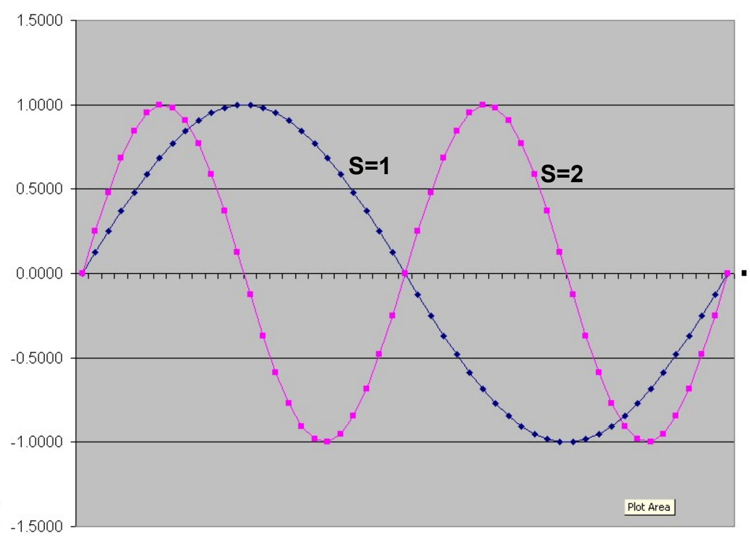
| | |
|---|---|
| Single ramp 0.→1. | float t = Timer;<br>float t = Timer*Timer;<br>float t = Timer*Timer*Timer;<br>float t = 3.*Timer$^2$ – 2.*Timer$^3$;<br>float t = 10.*Timer$^3$ – 15.*Timer$^4$ + 6.*Timer$^5$ |
| Double ramp 0.→1. →0. | float t;<br>if( Timer <= .5 )<br>t = 2.*Timer;<br>else<br>t = 2. * ( 1. – Timer ); |
| Smooth oscillation -1. → 1. → -1. | float t = sin( 2.*π*Timer ); |
| Smooth oscillation 0. → 1. → 0. | float t = .5 + .5*sin(2.*π*Timer ); |
| Faster oscillation | float t = sin( 2.*π*S*Timer ); |
| Bigger oscillation | float t = Mag * sin( 2.*π*S*Timer ); |





## float t = sin( 2.*π*S*Timer );

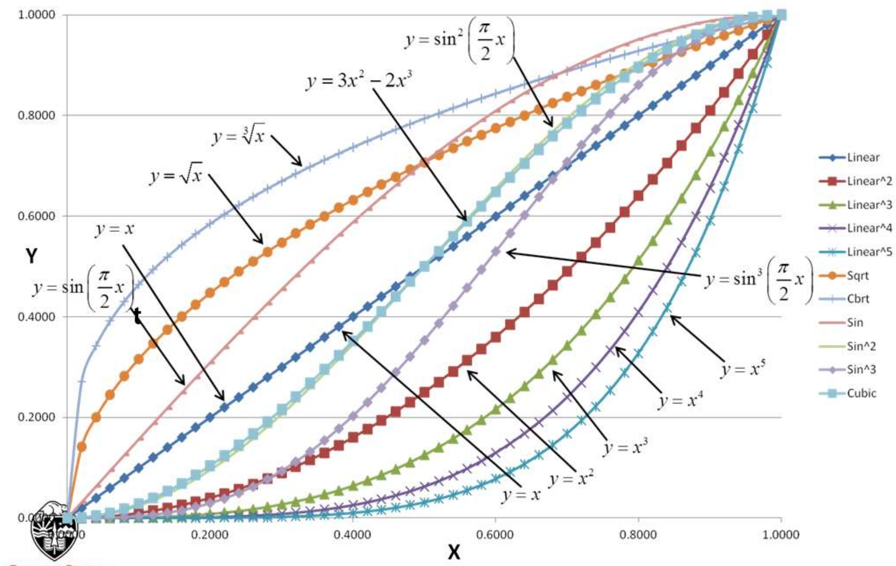
t




Timer

## Fun-With-Zero-To-One

5

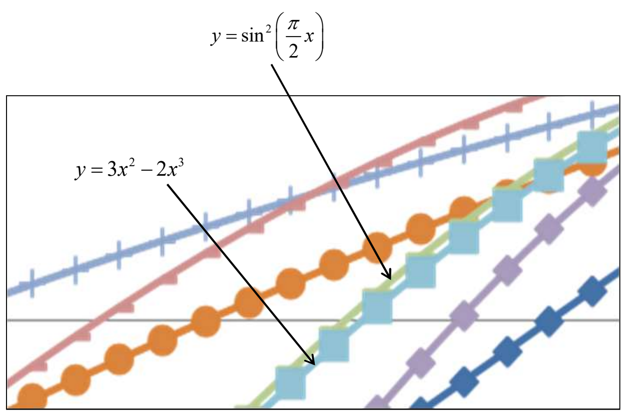



Oregon State University
Computer Graphics

mjb – December 4, 2022

## Sidebar: Why Do These Two Curves Match So Closely?

6




The Taylor Series expansion of $y = \sin^2\left(\frac{\pi}{2}x\right)$ around x=0.5 is:

$$y = \left(\frac{1}{2} - \frac{\pi}{4} + \frac{\pi^3}{96}\right) + x\left(\frac{\pi}{2} - \frac{\pi^3}{16}\right) + x^2\left(\frac{\pi^3}{8}\right) - x^3\left(\frac{\pi^3}{12}\right)$$

$$= .038 - .37x + 3.88x^2 - 2.58x^3$$

which is somewhat close to: $y = 3x^2 - 2x^3$

Oregon State University
Computer Graphics

mjb – December 4, 2022