Animation Effects using a Timer

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

Computer Graphics

Using Timers with Shaders

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

\[ \text{uniform float Timer;} \]

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:

```cpp
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

<table>
<thead>
<tr>
<th></th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single ramp 0.→1.</td>
<td>float t = Timer; float t = Timer<em>Timer; float t = Timer</em>Timer*Timer;</td>
</tr>
<tr>
<td></td>
<td>float t = 3.*Timer^2 – 2.*Timer^2; float t = 10.*Timer^3 – 15.*Timer^4 + 6.*Timer^5</td>
</tr>
<tr>
<td>Double ramp 0.→1. →0.</td>
<td>float t; if (Timer &lt;= .5) t = 2.<em>Timer; else t = 2.</em> (1. – Timer);</td>
</tr>
<tr>
<td>Smooth oscillation -1. → 1. → -1.</td>
<td>float t = sin(2.<em>π</em>Timer);</td>
</tr>
<tr>
<td>Smooth oscillation 0. → 1. → 0.</td>
<td>float t = .5 + .5*sin(2.<em>π</em>Timer);</td>
</tr>
<tr>
<td>Faster oscillation</td>
<td>float t = sin(2.<em>π</em>S*Timer);</td>
</tr>
<tr>
<td>Bigger oscillation</td>
<td>float t = Mag * sin(2.<em>π</em>S*Timer);</td>
</tr>
</tbody>
</table>

float t = sin(2.*π*S*Timer);
Fun-With-Zero-To-One

Sidebar: Why Do These Two Curves Match So Closely?

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

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

\[
= 0.038 - 0.37x + 3.88x^2 - 2.58x^3
\]

which is somewhat close to: \( y = 3x^2 - 2x^3 \)