Animation Effects using a Timer
Using Timers with Shaders

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

```glsl
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:

```c
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>Function Type</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single ramp 0.→1.</td>
<td>float t = Timer;</td>
</tr>
<tr>
<td></td>
<td>float t = Timer*Timer;</td>
</tr>
<tr>
<td></td>
<td>float t = Timer<em>Timer</em>Timer;</td>
</tr>
<tr>
<td></td>
<td>float t = 3.*Timer$^2$ – 2.*Timer$^3$;</td>
</tr>
<tr>
<td></td>
<td>float t = 10.*Timer$^3$ – 15.*Timer$^4$ + 6.*Timer$^5$</td>
</tr>
<tr>
<td>Double ramp 0.→1. →0.</td>
<td>float t;</td>
</tr>
<tr>
<td></td>
<td>if( Timer &lt;= .5 )</td>
</tr>
<tr>
<td></td>
<td>t = 2.*Timer;</td>
</tr>
<tr>
<td></td>
<td>else</td>
</tr>
<tr>
<td></td>
<td>t = 2. * ( 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

\[ y = \sin^2 \left( \frac{\pi}{2} x \right) \]

\[ y = 3x^2 - 2x^3 \]

\[ y = \frac{3}{2} \sqrt{x} \]

\[ y = \sqrt{x} \]

\[ y = x \]

\[ y = x^5 \]

\[ y = x^4 \]

\[ y = x^3 \]

\[ y = x^2 \]
Sidebar: Why Do These Two Curves Match So Closely?

\[ y = \sin^2 \left( \frac{\pi}{2} x \right) \]

\[ y = 3x^2 - 2x^3 \]

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

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

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

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