How to Freeze and Unfreeze an Animation and Maintain Time Continuity

Mike Bailey
mjb@cs.oregonstate.edu

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License
At the Top of the Program:

```cpp
const int MS_PER_CYCLE = 10000;       // 10000 milliseconds = 10 seconds
float TimeFrozen;                     // when animation was frozen
float TimeUnfrozen;                   // when animation was unfrozen
float TimeElapsed;                    // how much time elapsed between freezing and unfreezing
```

In Reset( ):

```cpp
TimeElapsed = 0.f;
```
In Keyboard( )

case 'f':
case 'F':
    Freeze = ! Freeze;
    if( Freeze )
    {
        glutIdleFunc(NULL);
        TimeFrozen = Time - TimeElapsed;
        if( TimeFrozen < 0. )
            TimeFrozen = TimeFrozen + 1.f;  // wrap-around
    }
    else
    {
        glutIdleFunc(Animate);
        int ms = glutGet(GLUT_ELAPSED_TIME);
        ms %= MS_PER_CYCLE;  // the value of ms is between 0 and MS_PER_CYCLE-1
        Time = (float)ms / (float)MS_PER_CYCLE;  // makes the value of Time [0.,1.)
        TimeUnfrozen = Time;
        TimeElapsed = TimeUnfrozen - TimeFrozen;
        if( TimeElapsed < 0. )
            TimeElapsed = TimeElapsed + 1.f;  // wrap-around
    }
    break;
When Drawing

```c
float time = Time - TimeElapsed;
if( time < 0. )
    time = time + 1.f;       // wrap-around
```

When drawing, now use `time` in the same way you used `Time` before. For example:

```c
    glRotatef( 360.f * time, 0., 1., 0. );
    or
    float y = Amplitude * sinf( 2.f * F_PI * time );
```