PascalGL User Guide

Unit – pglTime.pas

Summary

This unit’s purpose is to provide the user of the PascalGL library with time-keeping and event functionality. The primary class, **TPGLClock**, is intended to allow for control of the “main loop” iteration duration, as well as provide access to system (CPU) time and real-world time, elapsed time, tracking of Frames-Per-Second, as well as other features. The unit also provides a **TPGLTimer** which allows the user to set and control “duration” events. Finally, the **TPGLTimeTrigger** class provides “time-targeted” event functionality.

Types and Classes

* pglTimeEvent – Procedure Type
* TPGLTimeStruct – Record Type
* TPGLTimer – Class Type
* TPGLTimeTrigger – Class Type
* TPGLClock – Class Type

pglTimeEvent

type pglTimeEvent = procedure();

A simple procedure type that accepts no parameters. PglTimeEvent is accepted as a parameter by TPGLTimer and TPGLTimeTrigger methods to be used as a callback procedure/function pointer. The procedure is executed when the TPGLTimer and TPGLTimeTrigger objects reach their specified duration/target.

***Example***

*In this example, we create a simple procedure with no parameters, a TPGLClock and TPGLTimer instance and a main procedure. We initialize Clock and Timer, and assign TimerProc to the Timer. We then assign Timer to Clock and begin our main loop that executes as long as Clock is running. After 5 seconds, Timer’s pglTimeEvent executes, which stops the clock and allows the main loop to exit. Clock and Timer are then destroyed before the program terminates.*

procedure TimerProc(); register; // this is our procedure that will be passed as a pglTimeEvent

procedure Main(); register;

var

Clock: TPGLClock;

Timer: TPGLTimer;

procedure TimerProc();

begin

Clock.Stop();

end;

procedure Main();

begin

Clock := TPGLClock.Create();

Clock.SetIntervalInFPS(60);

Timer := TPGLTimer.Create(TimerProc,5,False);

Timer.SetActive(True);

Clock.AssignTimer(Timer);

Clock.Start();

while Clock.Running = True do begin

Clock.Wait();

end;

Clock.Free();

Timer.Free();

end;

TPGLTimeStruct

type TPGLTimeStruct = record

Seconds: Double;

Minutes: Int32;

Hours: Int32;

Class Operator Initialize(Out Dest: TPGLTimeStruct); Register;

Procedure AddTime(ATime: Double); Register;

Function ToString(): String; Register;

End;

A record structure that specifies a time or duration in HH:MM:SS format. Functionality is provided to add to the time values and return the values as a formatted string.

Fields

Seconds: Double – Specifies the seconds portion of the time/duration.

Minutes: Int32 – Specifies the minutes portion of the time/duration.

Hours: Int32 – Specifies the hours portion of the time/duration.

Methods

AddTime(ATime: Double) – Increase the time/duration of the record by ATime seconds. Minutes and Hours are updated accordingly and Seconds is rolled over to 0 if Seconds exceeds 59.99. The user can provide a negative value to ATime and time/duration will be decreased. Seconds, Minutes and Hours will not be allowed to decrease below 0.