

Timer Manager For Unity API

TimerManager.GetInstance

```
public static TimerManager GetInstance();
```

Returns

TimerManager Time Manager Singleton

Description

Get the Time Manager Singleton, create it if it doesn't exist

TimerManager.Instance

```
public static TimerManager Instance;
```

Description

Time Manager Singleton, **null** if not yet created

TimerManager.SetTimer

```
public void SetTimer(ref TimerHandle inHandle, TimerDelegate inDelegate, float inDuration, bool inbLoop = false, float inTimeScale = 1f, float inFirstDelay = -1f);
```

Parameters

inHandle Timer handle
inDelegate Timer delegate
inDuration Timer duration
inbLoop Whether to loop
inTimeScale Time scale, clamped to the interval [0.001, 1000]
inFirstDelay First Delay

Description

Create a timer and return the timer handle **inHandle**. If **inHandle** points to a still active timer, destroy it first and then override it. The timer lasts for **inDuration** seconds, executes **inDelegate**, and automatically destroys itself. If **inbLoop** is true, the timer will execute **inDelegate** at intervals of **inDuration** seconds until explicitly destroyed. If **inFirstDelay** is not negative, the initial duration of the timer will be replaced with **inFirstDelay** seconds. **inTimeScale** can be used to adjust the time flow rate of this timer.

TimerManager.SetTimerForNextUpdate

```
public TimerHandle SetTimerForNextUpdate(TimerDelegate inDelegate);
```

Parameters

inDelegate Timer delegate

Returns

TimerHandle Handle of the created timer

Description

Create a timer and return the timer handle **inHandle**. The timer will execute **inDelegate** in the next frame and automatically destroy itself.

TimerManager.ClearTimer

```
public void ClearTimer(in TimerHandle inHandle);
```

Parameters

inHandle Timer handle

Description

Destroy the timer pointed to by **inHandle**

TimerManager. ClearAllTimersForObject

public void **ClearAllTimersForObject**(object **obj**);

Parameters

obj The object using the timers

Description

Destroy all timers associated with the **obj** object. When **obj** is destroyed, this method needs to be called to clean up any timers that depend on **obj**.

TimerManager. PauseTimer

public void **PauseTimer**(TimerHandle **inHandle**);

Parameters

inHandle Timer handle

Description

Pause the timer pointed to by **inHandle**

TimerManager. UnPauseTimer

public void **UnPauseTimer**(TimerHandle **inHandle**);

Parameters

inHandle Timer handle

Description

Unpause the timer pointed to by **inHandle**

TimerManager. IsTimerExists

public bool **IsTimerExists**(in TimerHandle **handle**);

Parameters

inHandle Timer handle

Returns

bool Whether the timer pointed to by **inHandle** exists

Description

Whether the timer pointed to by **inHandle** exists

TimerManager. GetTimerRemaining

public float **GetTimerRemaining**(in TimerHandle **inHandle**);

Parameters

inHandle Timer handle

Returns

float Remaining real time of the timer pointed to by **inHandle**

Description

Get remaining real time of the timer pointed to by **inHandle**. Return -1 if the timer does not exist

TimerManager. GetTimeScale

public float **GetTimeScale**(in TimerHandle **inHandle**);

Parameters

inHandle Timer handle

Returns

float Time scale of the timer pointed to by **inHandle**

Description

Get time scale of the timer pointed to by **inHandle**. Return -1 if the timer does not exist

TimerManager. SetTimeScale

public void **SetTimeScale**(in TimerHandle **inHandle**, float **inTimeScale**);

Parameters

inHandle Timer handle

inTimeScale New time scale

Description

Set time scale of the timer pointed to by **inHandle**

Timer Handle Syntactic sugar

TimerHandle. Clear

public void **Clear**();

Description

Destroy the timer pointed to by **inHandle**

TimerHandle. Pause

public void **Pause**();

Description

Pause the timer pointed to by **inHandle**

TimerHandle. UnPause

public void **UnPause**();

Description

Unpause the timer pointed to by **inHandle**

TimerHandle. IsExists

public bool **IsExists**();

Returns

bool Whether the timer pointed to by **inHandle** exists

Description

Whether the timer pointed to by **inHandle** exists

TimerHandle. Remaining

public float **Remaining**;

Description

Get remaining real time of the timer pointed to by **inHandle**. Return -1 if the timer does not exist

TimerHandle. TimeScale

public float **TimeScale**;

Description

Get/Set time scale of the timer pointed to by **inHandle**