

Timer Manager For Unity API

TimerManager.GetInstance

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

返回

`TimerManager` 时间管理器单例

描述

获取时间管理器单例，如果不存在，则创建它

TimerManager.Instance

```
public static TimerManager Instance;
```

描述

时间管理器单例，如果还未被创建，则为 `null`

TimerManager.SetTimer

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

参数

<code>inHandle</code>	计时器句柄
<code>inDelegate</code>	计时器委托
<code>inDuration</code>	计时器持续时间
<code>inbLoop</code>	是否循环
<code>inTimeScale</code>	时间尺度，被钳制在区间[0.001, 1000]
<code>inFirstDelay</code>	首次持续时间

描述

创建一个计时器，并返回计时器句柄 `inHandle`，如果输入的 `inHandle` 指向一个仍然存活的计时器则先销毁它然后被覆盖。计时器持续 `inDuration` 秒后，执行 `inDelegate` 委托并自动销毁。如果 `inbLoop` 为 `true`，那么计时器将以间隔为 `inDuration` 秒执行一次 `inDelegate` 委托，直到被显式销毁。如果 `inFirstDelay` 不为负数，那么计时器首次持续时间将被替换为 `inFirstDelay` 秒。`inTimeScale` 可以此计时器的时间流速速率。

TimerManager.SetTimerForNextUpdate

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

参数

<code>inDelegate</code>	计时器委托
-------------------------	-------

返回

`TimerHandle` 被创建计时器的句柄

描述

创建一个计时器，并返回计时器句柄 `inHandle`。计时器将在下一帧执行 `inDelegate` 委托并自动销毁

TimerManager.ClearTimer

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

参数

<code>inHandle</code>	计时器句柄
-----------------------	-------

描述

销毁 `inHandle` 指向的计时器

TimerManager.ClearAllTimersForObject

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

参数

obj 使用计时器的对象

描述

销毁与 **obj** 对象有关的所有计时器，在 **obj** 被销毁时，需要调用此方法来清理任何依赖于此对象的计时器

TimerManager.PauseTimer

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

参数

inHandle 计时器句柄

描述

暂停 **inHandle** 指向的计时器

TimerManager.UnPauseTimer

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

参数

inHandle 计时器句柄

描述

唤醒 **inHandle** 指向的计时器

TimerManager.IsTimerExists

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

参数

inHandle 计时器句柄

返回

bool **inHandle** 指向的计时器是否存在

描述

inHandle 指向的计时器是否存在

TimerManager.GetTimerRemaining

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

参数

inHandle 计时器句柄

返回

float **inHandle** 指向的计时器剩余的真实时间

描述

获取 **inHandle** 指向的计时器剩余的真实时间，如果计时器不存在，返回-1

TimerManager. GetTimeScale

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

参数

inHandle 计时器句柄

返回

float **inHandle** 指向的计时器时间尺度

描述

获取 **inHandle** 指向的计时器时间尺度，如果计时器不存在，返回-1

TimerManager. SetTimeScale

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

参数

inHandle 计时器句柄

inTimeScale 新的时间尺度

描述

设置 **inHandle** 指向的计时器时间尺度

Timer Handle 语法糖

TimerHandle.Clear

```
public void Clear();
```

描述

销毁 `inHandle` 指向的计时器

TimerHandle.Pause

```
public void Pause();
```

描述

暂停 `inHandle` 指向的计时器

TimerHandle.UnPause

```
public void UnPause();
```

描述

唤醒 `inHandle` 指向的计时器

TimerHandle.IsExists

```
public bool IsExists();
```

返回

`bool` `inHandle` 指向的计时器是否存在

描述

`inHandle` 指向的计时器是否存在

TimerHandle.Remaining

```
public float Remaining;
```

描述

获取 `inHandle` 指向的计时器剩余的真实时间

TimerHandle.TimeScale

```
public float TimeScale;
```

描述

获取/设置 `inHandle` 指向的计时器的时间尺度