

Structs

TimerData

Description

Structure representing timer data containing various properties related to the timer's state and progress.

Properties

TimeElapsed	The elapsed time (in seconds) since the timer started.
TimeRemaining	The remaining time (in seconds) until the timer completes its duration.
RatioElapsed	The ratio of elapsed time to the total duration (expressed as a fraction between 0 and 1).
RatioRemaining	The ratio of remaining time to the total duration (expressed as a fraction between 0 and 1).
WorldTime	The current world time (in seconds) when the TimerData is updated.
FinishTime	The expected time (in seconds) when the timer will finish its duration.
TimeDelta	The time interval (in seconds) between the last update and the current update of the TimerData.

Classes

TimerEvent

Description

A class representing an event associated with a specific time during the execution of a Timer.

Properties

KeyTime	The time (in seconds) when this event should be triggered during the Timer's execution.
---------	---

KeyEvent	The UnityEvent that will be invoked when the associated time is reached during the Timer's execution.
----------	---

Private Properties

isCalled	A flag indicating whether the event has already been called during the current Timer execution.
----------	---

Public Methods

IsCalled	Checks if the event has been called during the current Timer execution.
CallEvent	Calls the associated UnityEvent, triggering the event's actions with the given TimerData.
Reset	Resets the state of the TimerEvent, allowing it to be called again in future Timer executions.

Timer

Description

Properties

Duration	The total duration (in seconds) of the timer.
IsLooped	A flag indicating whether the timer should be looped/restarted after it completes.
IsCompleted	A flag indicating whether the timer has completed its duration.
IsRunning	A flag indicating whether the timer is currently running.
UsesRealTime	A flag indicating whether the timer uses real-time or game-time. Real-time is unaffected by changes to the timescale of the game (e.g., pausing, slow-mo), while game-time is affected by these changes.

Private Properties

startTime	The time (in seconds) when the timer started or was last resumed.
lastUpdatedTime	The time (in seconds) when the TimerData was last updated or when the timer was last resumed.

timeBeforeStop	The time (in seconds) when the timer was stopped. Null if the timer is not stopped.
timeBeforePause	The time (in seconds) when the timer was paused. Null if the timer is not paused.
onFinish	The UnityEvent invoked when the timer completes its duration.
onUpdate	The UnityEvent invoked each frame with updated TimerData during the timer's execution.
timerEvents	List of TimerEvents representing events to be triggered during the timer's execution.

Public Methods

Timer	Constructor for the Timer class that initializes the timer with the provided parameters.
Stop	Stops the timer's execution. If the timer is already finished, it has no effect.
Pause	Pauses the timer's execution. If the timer is already paused or finished, it has no effect.
Resume	Resumes the timer's execution if it was previously paused. If the timer is not paused or finished, it has no effect.
AddEvent	description
AddEvents	Adds a TimerEvent to the list of events associated with the timer.
AddEvents	Adds a list of TimerEvents to the existing list of events associated with the timer.
GetTimeElapsed	Returns the elapsed time (in seconds) since the timer started. If the timer has finished or the world time exceeds the finish time, it returns the total duration of the timer to prevent negative values.
GetTimeRemaining	Returns the remaining time (in seconds) until the timer completes its duration.
GetRatio	Returns the ratio of elapsed time to the total duration (as a fraction between 0 and 1). Optionally, the ratio can be returned as a percentage if 'inPercentage' is true.
GetRatioRemaining	Returns the ratio of remaining time to the total duration (as a fraction between 0 and 1). Optionally, the ratio can be returned as a percentage if 'inPercentage' is true.

GetWorldTime	Returns the current world time (in seconds) based on whether the timer uses real-time or game-time.
GetFinishTime	Returns the expected time (in seconds) when the timer will finish its duration.
GetTimeDelta	Returns the time interval (in seconds) between the last update and the current update of the TimerData.
Update	Updates the timer's state and executes associated events. This method should be called each frame to keep the timer active and to trigger events at the appropriate times.