TimersMadeEasyLite

Overview

Intro:
TimersMadeEasyLite is a quick and easy way to set up customised timers with a few clicks. It has some customisation to get the look of the timer you really want, set up with some ready made prefabs ready to drop in to your project. This project has been set up to work in the canvas with GUI elements.

This documentation includes all the paid version features so you can see everything this and the paid version can do.

Paid version features – *Highlighted in orange*
1. Days timer, go up to a maximum of 364 days, 23 hours, 59 minutes and 59 seconds
2. Display Milliseconds
3. Remove leading zero's
4. Percentage display
5. Clock display
6. Adjust timer speed
7. Separator type adjustment, use / or . instead of the default ;
8. Pause and resume time functions
9. Add and remove time functions
10. Run a function at timer start as well as timer end

The paid version can be found here on the Unity Asset Store

Usage:
To quickly get started -

A note, make sure to be working within the canvas if you are displaying a visual timer.

1.There are three main ways to apply the timer:

1.1 - The first is to simply add the timer.cs script as a component on a text, textmeshpro, slider, or a radial image set to 'Filled'.

1.2 - The second is to have a main Timer control object with the script applied to that object. You can then open up display and then output type and select the desired output type which are the same options as above.

1.3 - The third is simply using the prefabs in the prefabs folder which have been automatically set up to run on start with 2 seconds of time.

2. Simply set the desired time you want under Set time. This will calculate the total seconds for you. A maximum of 23 hours, 59 minutes and 59 seconds is available in the free version.

3. Check start at runtime under customise to begin the timer on play. If you want to set it to start at a custom time, simply call the 'Timer' class method StartTimer(). This will use the time you have set in the inspector.

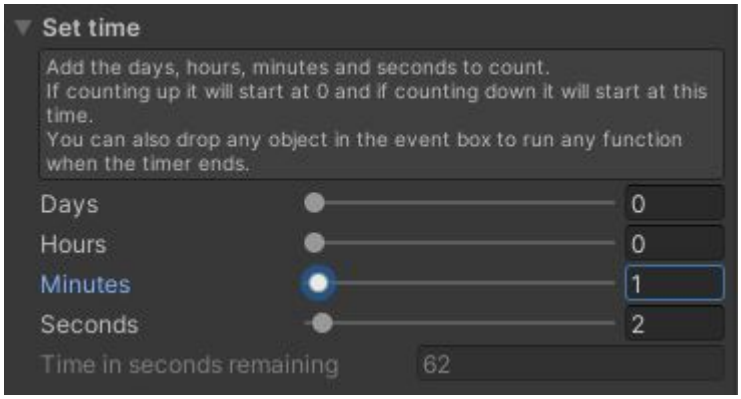
There are a few customisation options to choose from, most are self explanatory, so just feel free to have a play around with them, or continue to read this document.

Features

Set time:

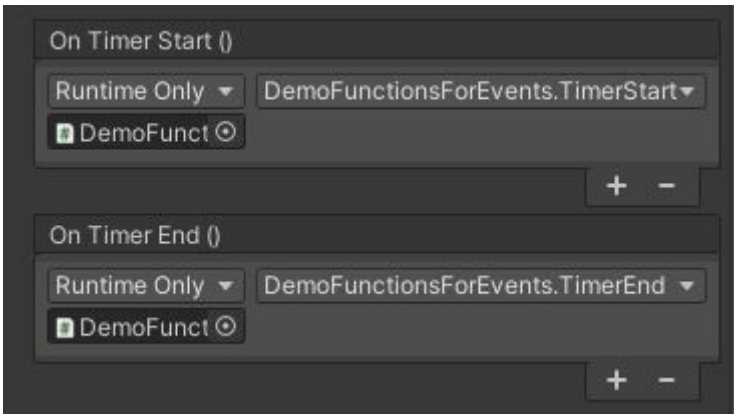
Days in paid version

Under the set time drop-down you can select the amount of time to set. This will be reflected in the Time in Seconds. The timer is set up to do all the calculations in seconds, using Time.deltaTime. This keeps it frame rate independent when being used of many different systems.



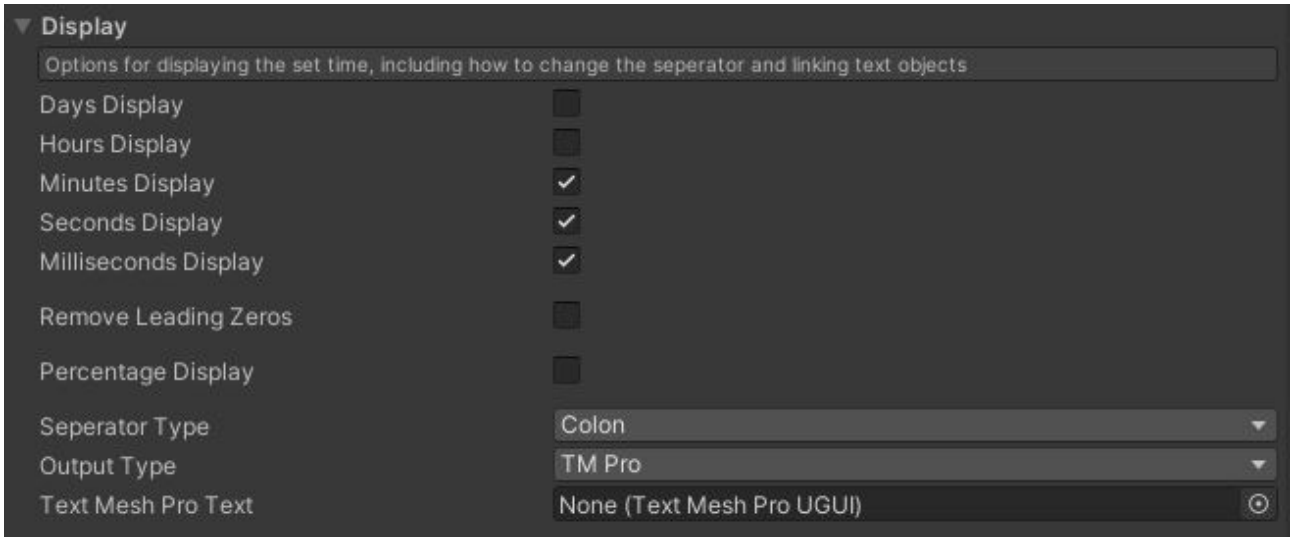
OnTimerStart only in paid

There are two built in custom events for the timer, which run on timer start, or at timer end. Simply link up your object, make sure your custom function is set to Public and select it from the class.



Display:

Under the display drop-down there are options to not display certain parameters such as days or *milliseconds*. Depending on how you want to view the timer, you can select any combination of days, hours, minutes, seconds and *milliseconds*, however, the other display options have some limitations. Please note, if you are hiding something that is set, it will still be calculated, just not displayed. For example, if you have 3 days set in the Set time section, and choose not display days, it will still calculate them but not show them. Visa versa if you do not set days or hours, and choose to display them, they will just appear as '0's.



Remove Leading Zero's – By default the display will be set with leading zeros, for example; 01:07:147. In some situations such as counting down with only seconds for a countdown to start the level, you would probably only want to display a single digit for the seconds. Check this box and check only Seconds Display and you will have a single digit countdown.

Percentage Display – This option simply converts your time remaining into a percentage of the Set time. So for example if you set 10 seconds counting down, it will display 50% when at 5 seconds.

Separator Type – This is an option to change the Separator. By default it is set to use ':' in between the selected display parameters like so 0:00:00:00:000. You can choose from '.' or '/' resulting in 0.00.00.00.000 or 0/00/00/00/000.

Output Type - The output type is asking how you would like to display your time. There are currently 5 options available (None, standard text, text mesh pro text, slider, and dial). Choose one from the drop-down menu and then simply drag the object you would like to use for display making sure it matches the required Class Type.

None – Will not use an output type, meaning you can still use this timer to count in the background, and then execute functions on timer start or end.

Text & TextMeshPro – Will simply display the set time using the selected display options.

Slider – This option will take any slider with the component Slider on it. It uses the Set time

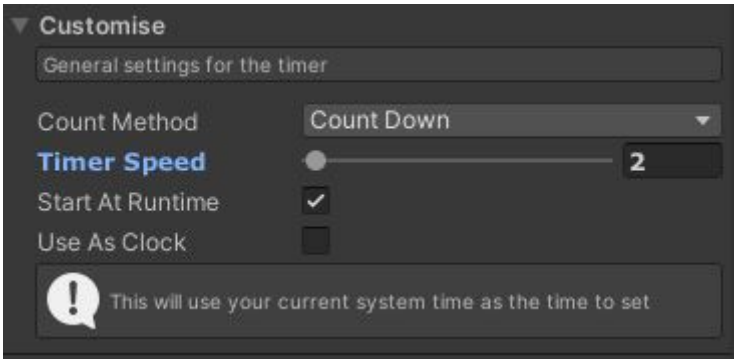
parameters to set the slider.maxValue and slider.value to iterate the remaining time.

Dial – This option requires an object with an Image component, or use one of those provided in the Sprites folder. Make sure the mode of the Image component is set to 'Filled' and the fill amount to 1. This uses some Lerping to work out how much of the dial should show.

There are prefabs of all of these objects ready to just drag and drop into the project and customise. The dials use white sprites so you can easily manage colours.

Customise:

Under the customise drop-down there are a few options to change how the timer functions.



Count Method – An option to count down or up. If you choose to count down, the time on time start is the time you set in Set time. If you choose to count up, the time will start at 0 and end at the time you set in Set time.

Timer Speed **–** A simple slider to control how fast the timer runs. Default is 1. If you want to half the speed, so 4 seconds will take 8 seconds, use a value of 0.5. If you want it to run double as quick, so 4 seconds will actually take 2 seconds, use a value of 2.

Start At Runtime – If checked, runs the timer when pressing play. If unchecked, the timer will not run unless you call the public method StartTimer(). See under scripting for details.

Use As Clock **–** This will use your system time to count from, it will also automatically change your count method to count up, and disable the display of days and milliseconds. You can still count down if you want to. Please note, this option is designed to be used with text objects and will not work with sliders or dials.

Scripting:

There are several public methods that can be called from the Timer script.

First of all make sure to get a reference to the Timer class in your preferred method.

Public Methods -

GetRemainingSeconds()
This will return the number of remaining seconds. For example, you can make a check every second to see if the remaining time is equal to halfway through.

StartTimer()
This method simply starts the timer, with the time set in the inspector. You have to stop the timer or wait for it to end before calling this method again.

StopTimer()
This will stop the timer running, and reset its values according to the count method. If counting up, your reset value will be 0, if counting down, your reset value will be the values set in Set time.

PauseTimer()
Pauses the timer, stopping all counting without resetting anything. If the timer is paused, you can either resume or stop the timer. Please note you cannot start a new timer while it is paused.

ResumeTimer()
Resumes the timer if it was paused. This method only works if the timer is paused.

AddTime(float seconds)
Add time to the timer in seconds. You can use the inspector Set time section to work out how many seconds to add. This will only work if the time to add will not take you beyond the Set time if counting down. Please note, this method by its nature works in reverse when you change the count method. For example, trying to add time to a timer counting down from 10 seconds, will make the timer last longer. When adding time to counting up, it will making the timer end quicker.

RemoveTime(float seconds)
Remove time from the timer in seconds. You can use the inspector Set time section to work out how many seconds to remove. This will only work if the time to remove will not take you less than 0 if counting up. Please note, this method by its nature works in reverse when you change the count method. For example, trying to add time to a timer counting down from 10 seconds will make the timer last longer. When adding time to counting up, it will make the timer end quicker.

float ReturnTotalSeconds()
This will return the number of seconds set in the inspector under Set time. This does not count, it simply returns the total number of seconds set.

double ConvertToTotalSeconds(float days, float hours, float minutes, float seconds)
You can call this function with your desired days, hours, minutes and seconds to receive these values returned as seconds.

string DisplayFormattedTime(double remainingSeconds)
This method handles the display input and output. If you put a total number of seconds into it, it will convert it and return a string of the selected display options set in the inspector. This method is used in conjunction with the remaining seconds to output the remaining time in a text format.

If you have any questions or queries please do not hesitate to contact us at support@golemitegames.com