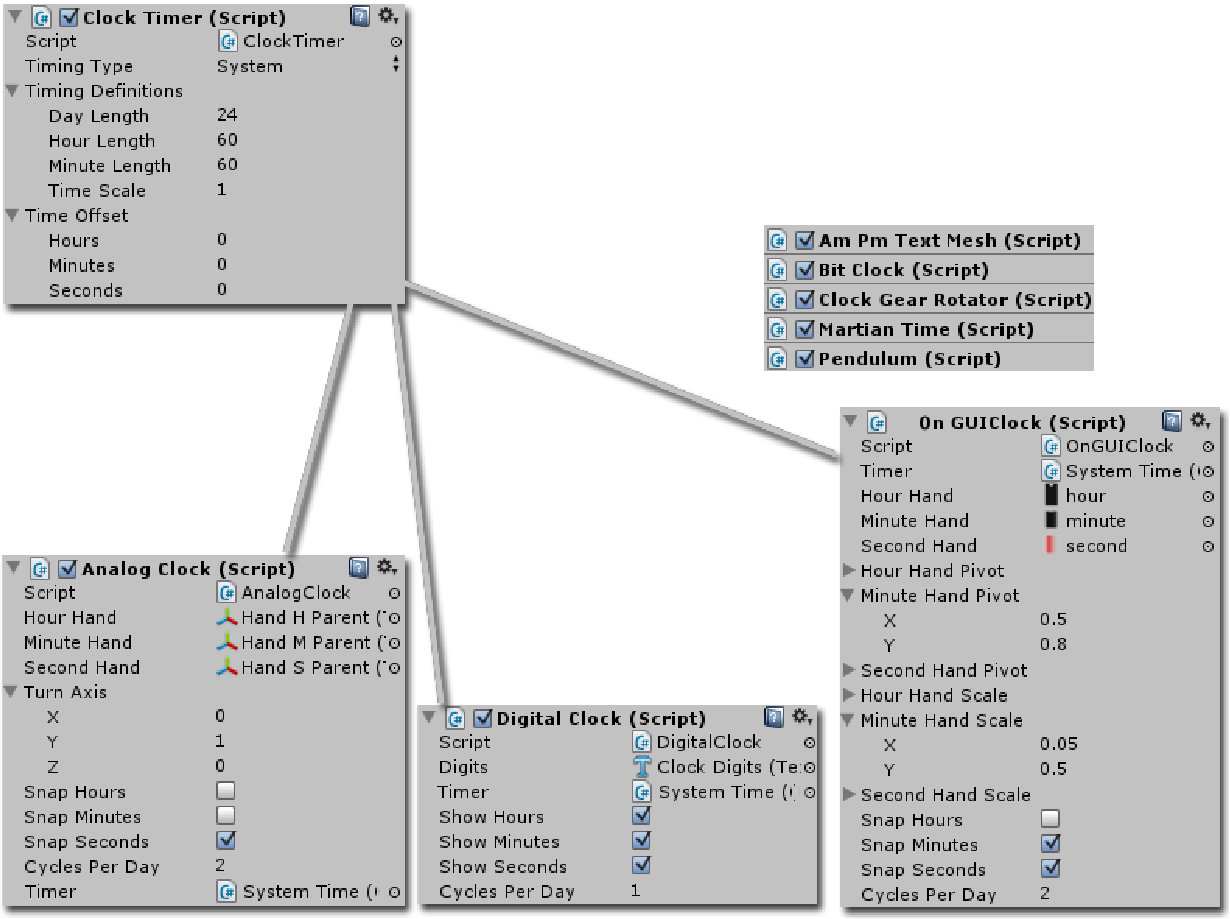
Clock Script Pack

for Unity

1 Overview

Clock Script Pack is a package of scripts, use examples and associated assets for Unity for the purpose of creating different types of in-game clocks.

The heart of the package is the ClockTimer class, which holds the actual time, and the rest of the scripts are used to render the clocks on screen. ClockTimer has various settings, and the game can have an arbitrary number of different timers at different settings (or just one that is used by all clocks)



☑ Clock Timer (Script)

Script	ClockTimer
Timing Type	System
Timing Definitions	
Day Length	24
Hour Length	60
Minute Length	60
Time Scale	1
Time Offset	
Hours	0
Minutes	0
Seconds	0

☑ Am Pm Text Mesh (Script)
☑ Bit Clock (Script)
☑ Clock Gear Rotator (Script)
☑ Martian Time (Script)
☑ Pendulum (Script)

On GUIClock (Script)

Script	OnGUIClock
Timer	System Time ('
Hour Hand	hour
Minute Hand	minute
Second Hand	second
Hour Hand Pivot	
Minute Hand Pivot	
X	0.5
Y	0.8
Second Hand Pivot	
Hour Hand Scale	
Minute Hand Scale	
X	0.05
Y	0.5
Second Hand Scale	
Snap Hours	□
Snap Minutes	☑
Snap Seconds	☑
Cycles Per Day	2

☑ Analog Clock (Script)

Script	AnalogClock
Hour Hand	Hand H Parent ('
Minute Hand	Hand M Parent ('
Second Hand	Hand S Parent ('
Turn Axis	
X	0
Y	1
Z	0
Snap Hours	□
Snap Minutes	□
Snap Seconds	☑
Cycles Per Day	2
Timer	System Time ('

☑ Digital Clock (Script)

Script	DigitalClock
Digits	Clock Digits (Te
Timer	System Time ('
Show Hours	☑
Show Minutes	☑
Show Seconds	☑
Cycles Per Day	1

2 Scripts

2.1 ClockTimer

Keeps track of time in one of various modes:

- **System** - Takes the time from the user’s operating system
- **Custom** - Accumulated Time.deltaTime. Shouldn’t be used if high precision or consistency is required
- **UTC** - Universal Time Coordinated, analogous to GMT
- **SinceLevelLoad** - Game time passed since current level was loaded
- **SinceGameStarted** - Game time passed since game was started
- **RealTimeSinceGameStart** - Real time since game was started, unaffected by Time.timeScale

Timing definitions:

A **time scale** can be applied to the timer, resulting in a faster or slower pace. Timescale can also be 0 or negative. Day, hour and minute lengths can be set to custom values if other than the ordinary 24/60/60 values are needed.

Time Offset:

An offset can be set to the clock. The offset is unaffected by timeScale. For example, to have the timer keep track of time in New York, NY, set the timing type to UTC and hours offset to -5.

2.2 AnalogClock

Rotates the hands of an analog clock based on a time read from a ClockTimer. Hour, minute and second hands can be supplied, but they are not mandatory - you can have a clock without hand for seconds, for example. The **turnAxis** variable defines the local axis over which to rotate the clock hands, and the hands can be set to **snap** to whole values. A **cyclesPerDay** variable (default: 2) defines how many times the hour hand rotates during a day.

2.3 DigitalClock

Creates and updates the time read from a ClockTimer as a string. If a TextMesh is connected to the **digits** variable, the text of the TextMesh is updated accordingly. Will format the text with the corrent number of leading zeroes and separating characters between hours, minutes and seconds. If **cyclesPerDay** variable is set to 1, the clock behaves as a 24h clock, and set to 2, as a 12h clock. Optionally, the text can be set to exclude hours, minutes or seconds. **GetTimeText()** function returns the current time string, if you wish to render it in some fashion other than TextMesh (for example on OnGUI).

2.4 OnGUIClock

Renders clock hands on OnGUI. Usage:

1. Add OnGUIClock component to a gameobject
2. Initialize the public variables
    a. timer - the ClockTimer used
    b. hourHand, minuteHand and secondHand textures - vertical, upward pointing textures for the clock hands
    c. hour, minute and secondHandPivot that define the pivot points for the hands, typical value is (0.5, 1)
    d. hour, minute and secondHandScale that define how large the textures are rendered in relation to the Rect that defines the clock draw area, typical value is (0.05, 0.5)
    e. snap variables define whether the hands snap to their values or progress smoothly

    f. cyclesPerDay defines the number of rotations the hour hand does in a day (typically analog clocks are 12h clocks, so the hand makes 2 cycles)

3. In the class where you make your OnGUI calls to draw the GUI, add a reference to the OnGUIClock component, and call it’s Draw(Rect position) function

Example call:

```
public OnGUIClock myClock;
void OnGUI()
{
	myClock.Draw(new Rect(0, 0, 300, 300));
}
```

2.5 BitClock

Converts the time read from a ClockTimer into a binary value, and displays it on a TextMesh if one is connected. The binary value can have up to 31 digits, and can be grouped into groups separated by a chosen character or string. BitClock advances in ticks of one bit, where the full range of the binary value represents a full day, so for example in a 16-bit clock the full range is .

Examples of output:

bitGroups	bitsPerGroup	time	output
1	8	00:00:00	00000000
4	4	12:00:00	1000.0000.0000.0000
2	4	23:59:59	1111.1111
2	4	01:30:00	0001.0000

If you wish to render the text in some other way than a TextMesh (such as on OnGUI) you can get the output string from the BitClock object with GetBitString() function.

2.6 Utility Scripts

2.6.1 AmPmTextMesh

Set the text on a TextMesh to “AM” or “PM” depending on time read from a ClockTimer. The text can be customized to something other than “AM” or “PM”.

2.6.2 MartianTime

Initialized a ClockTimer to represent MTC - Mars Time Coordinated. In effect, multiplies the ClockTimer’s timeScale by 1 / 1.027491, sets the timing type to RealTimeSinceGameStart and sets the correct offset. If you wish to set the ClockTimer to a spesific Martian time zone, just set the ClockTimer’s time offset accordingly.

2.6.3 Pendulum

Swings a transform back and forth according to time read from a ClockTimer.

2.6.4 ClockGear

Rotates on object over time in one of three modes:

- Default. Simply rotates the object over time.
- SeeSaw. Rotates back and forth between -360 and 360 degrees.
- Snappy. Rotates in increments, and pauses between the movements.

Can be connected to a ClockTimer, in which case the rotation speed takes the timer’s timeScale in to account.

3 Prefabs

The following clock prefabs can be found with the package. The all require a ClockTimer to be connected. Their main purpose is to serve as examples and templates for your own, awesome in-game clocks.

- Analog Clock 2D - a typical analog clock on a 2D plane
- Analog Clock 2D Gear - a steampunk-esque analog clock on 2D planes
- BitTextClock - a TextMesh that displays the time in binary format
- Digital Clock 2D - digital 24h clock in 2D
- Digital Clock 3D - a digital 12h (am/pm) alarm clock
- Tall Clock - a grandfather clock with a swingin pendulum
- Tall Clock 2 - another grandfather clock
- Three Floating Dials - a scifi clock made of three separate dials that rotate (instead of having hands that rotate)

In addition, the Prefabs/Timers folder contains a group of ClockTimer prefabs in various settings.

4 FAQ

Q: Will this work on mobile?

A: Yes (tested on iPad 1st gen and iPhone 4), but as always, watch your draw calls and avoid drawing large transparent objects on the screen when not necessary. Another optimization tip is to keep the amount of ClockTimers to a minimum - most games only run in one time, so they only need one ClockTimer. The overhead per timer is not much, though, and the clock scripts themselves are pretty lightweight.

5 Contact

Clock Script Pack is released for Unity Asset Store and maintained by Ifelse Media Ltd. We are happy to provide support for our products and to answer any questions.

Support email: support@ifelsemedia.com

Website: http://ifelsemedia.com/
Development blog: http://ifelsemedia.tumblr.com/
Facebook: https://www.facebook.com/IfelseMedia
Twitter: https://twitter.com/IfelseMedia
YouTube Channel: http://www.youtube.com/user/IfelseMedia

The developer responsible for this product is Timo Moisio, and can be contacted directly at timo.moisio@ifelsemedia.com

Do you think this pack is missing something vital? Don't hesitate to send us email, we'd love to hear from you on how we could improve this package!