Analog Cloks Vol.1

-

Introduction

This package contains three different models of analog clocks including an alarm clock.
All prefabs can use preset date and time or the system date and time. Therefore they can be easily adjusted according to a global clock or used as a global clock.

Details

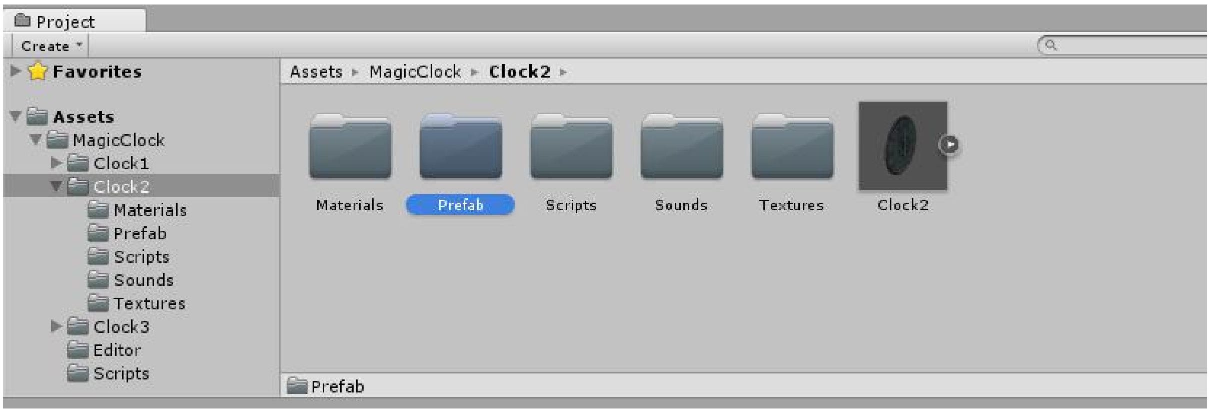
All models of this package were modeled using Blender 2.78a and contain **approximately** the following number of tris:

- Clock1: 1480;
- Clock2: 1858;
- Clock3: 2558.

The textures are in the size 2048x2048 and contain the maps, Diffuse, Normal, Ambient Occlusion and Specular.

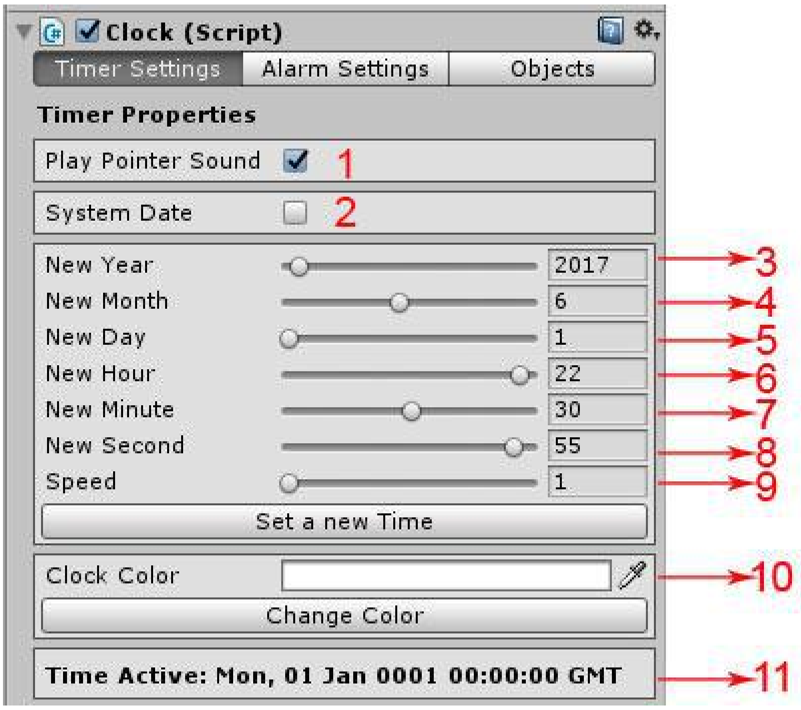
Using the Clocks:

Within each asset folder, we have a folder called Prefab. This folder contains the pre-configured clock asset and its scripts. Use these assets in your scene.



Setting the Date and Time:

With your clock in the scene, you will see in the inspector a flap with the image below:



Timer Properties

Play Pointer Sound ☑

System Date ☐

New Year 2017
New Month 6
New Day 1
New Hour 22
New Minute 30
New Second 55
Speed 1

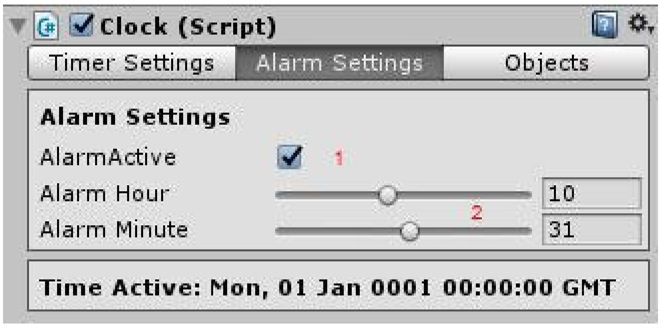
Clock Color

Time Active: Mon, 01 Jan 0001 00:00:00 GMT

1. If it is checked, it will play the sound of the second hand. It is useful to uncheck it if you increase the speed of your watch too much.
2. If checked, your clock will be set to system time and date, and changes to the date and time will be unavailable.
3. Adjusts the current clock year.
4. Sets the current clock month.
5. Sets the day of the current clock.
6. Sets the current clock time.
7. Adjusts the minutes of the current clock.
8. Sets the seconds of the current clock.
9. Adjusts the speed at which time will pass on your watch in a range of 1 to 100.
10. Available in this package only for the alarm clock. Changes the color of your alarm clock.
11. Shows the current time of your watch.

* After setting the date and time, you must click "**Set a new Time**" for the changes to take effect.
* After setting the new alarm color, you must click "**Change color**" to set the new color of your alarm clock.

Alarm Settings:



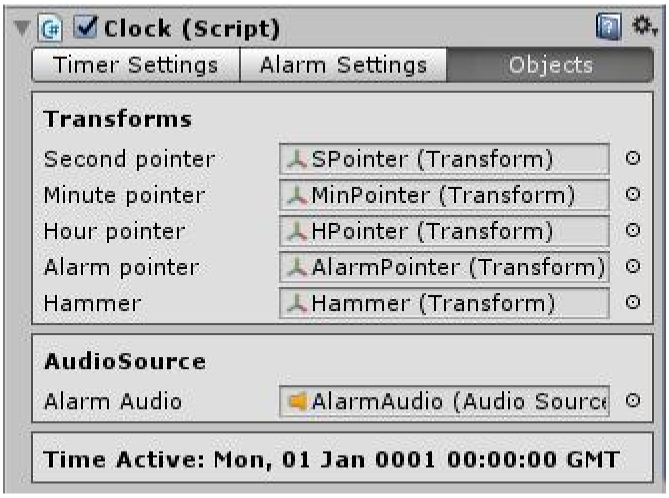
Alarm Settings

AlarmActive ☑
Alarm Hour 10
Alarm Minute 31

Time Active: Mon, 01 Jan 0001 00:00:00 GMT

1. Sets whether the alarm is active or not. If the alarm is waking up and you want to stop it, uncheck this box.
2. Hour and minutes for the alarm to start waking up.

Transforms:



Transforms

Second pointer: SPointer (Transform)
Minute pointer: MinPointer (Transform)
Hour pointer: HPointer (Transform)
Alarm pointer: AlarmPointer (Transform)
Hammer: Hammer (Transform)

AudioSource

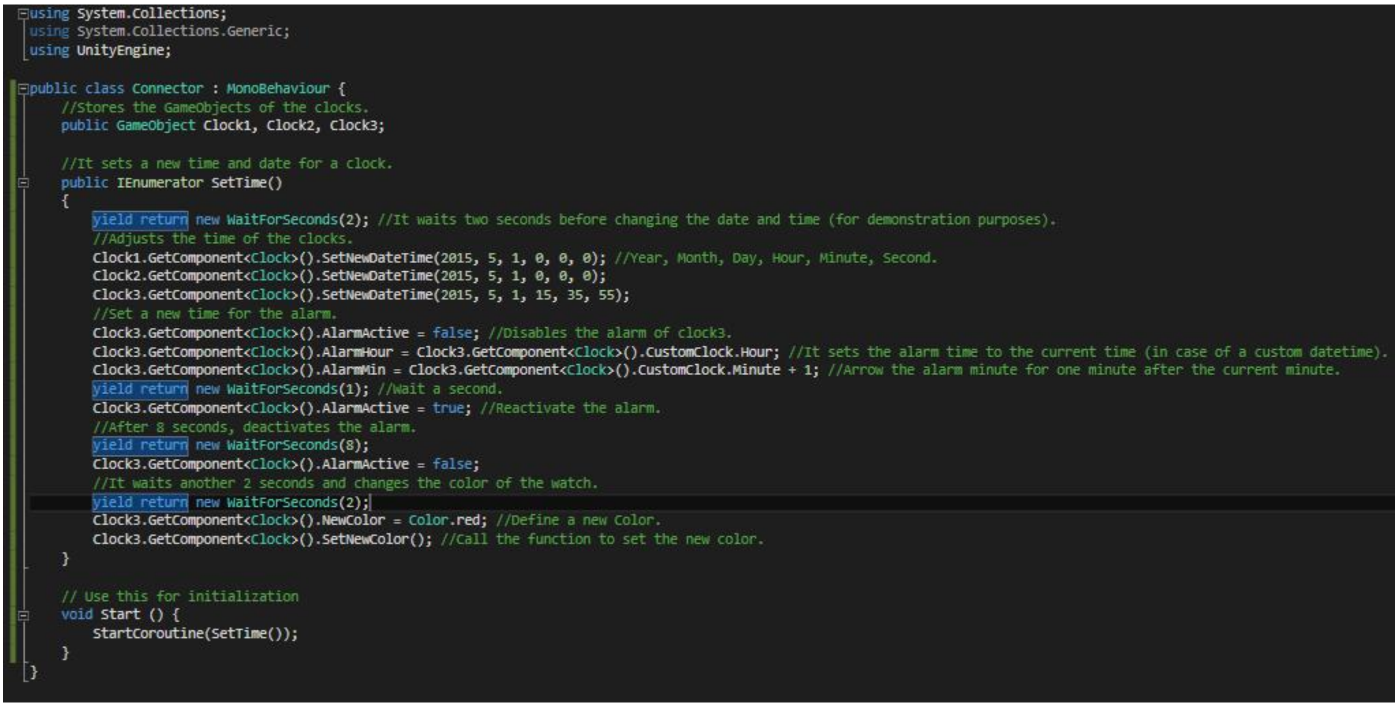
Alarm Audio: AlarmAudio (Audio Sourc

Time Active: Mon, 01 Jan 0001 00:00:00 GMT

This section defines the transforms of the clock hands and alarm clock (in the case of an alarm clock). The prefabs contained in the package already come with this configured session, but it can be useful if you want to add functionality in a custom clock model.

Scripting:

You can also change the date and time, alarm and alarm color via script.
In your assets folder, under "**SampleScene-> SampleScripts**" there is a script called **Connector.cs**.
This script explains in an explanatory way how to change the properties of your clocks via script.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Connector : MonoBehaviour {
    //Stores the GameObjects of the clocks.
    public GameObject Clock1, Clock2, Clock3;

    //It sets a new time and date for a clock.
    public IEnumerator SetTime()
    {
        yield return new WaitForSeconds(2); //It waits two seconds before changing the date and time (for demonstration purposes).
        //Adjusts the time of the clocks.
        Clock1.GetComponent<Clock>().SetNewDateTime(2015, 5, 1, 0, 0, 0); //Year, Month, Day, Hour, Minute, Second.
        Clock2.GetComponent<Clock>().SetNewDateTime(2015, 5, 1, 0, 0, 0);
        Clock3.GetComponent<Clock>().SetNewDateTime(2015, 5, 1, 15, 35, 55);
        //Set a new time for the alarm.
        Clock3.GetComponent<Clock>().AlarmActive = false; //Disables the alarm of clock3.
        Clock3.GetComponent<Clock>().AlarmHour = Clock3.GetComponent<Clock>().CustomClock.Hour; //It sets the alarm time to the current time (in case of a custom datetime).
        Clock3.GetComponent<Clock>().AlarmMin = Clock3.GetComponent<Clock>().CustomClock.Minute + 1; //Arrow the alarm minute for one minute after the current minute.
        yield return new WaitForSeconds(1); //Wait a second.
        Clock3.GetComponent<Clock>().AlarmActive = true; //Reactivate the alarm.
        //After 8 seconds, deactivates the alarm.
        yield return new WaitForSeconds(8);
        Clock3.GetComponent<Clock>().AlarmActive = false;
        //It waits another 2 seconds and changes the color of the watch.
        yield return new WaitForSeconds(2);
        Clock3.GetComponent<Clock>().NewColor = Color.red; //Define a new Color.
        Clock3.GetComponent<Clock>().SetNewColor(); //Call the function to set the new color.
    }

    // Use this for initialization
    void Start () {
        StartCoroutine(SetTime());
    }
}
```

In the example scene there is a GameObject containing this script and you can test its features.
Keep in mind that in this particular package, the only asset in which color can be modified is Clock3. This is already being fixed for future packages.

Getting Help:

If you have any questions or suggestions about this package, please contact us at contact.renan@gmail.com.

Thank you for purchasing this package!