

DG Realistic Clock



10:08

DG Realistic Clock

© 2025 Yong moon Kwon. All rights reserved.
Version 1.0.0

Latest Document :

<https://docs.google.com/document/d/1G8Sl9BdYxN4gpyfDzNCczPzV13SbhAguCbQAqTE9o8Q/edit?usp=sharing>

English

1. Overview

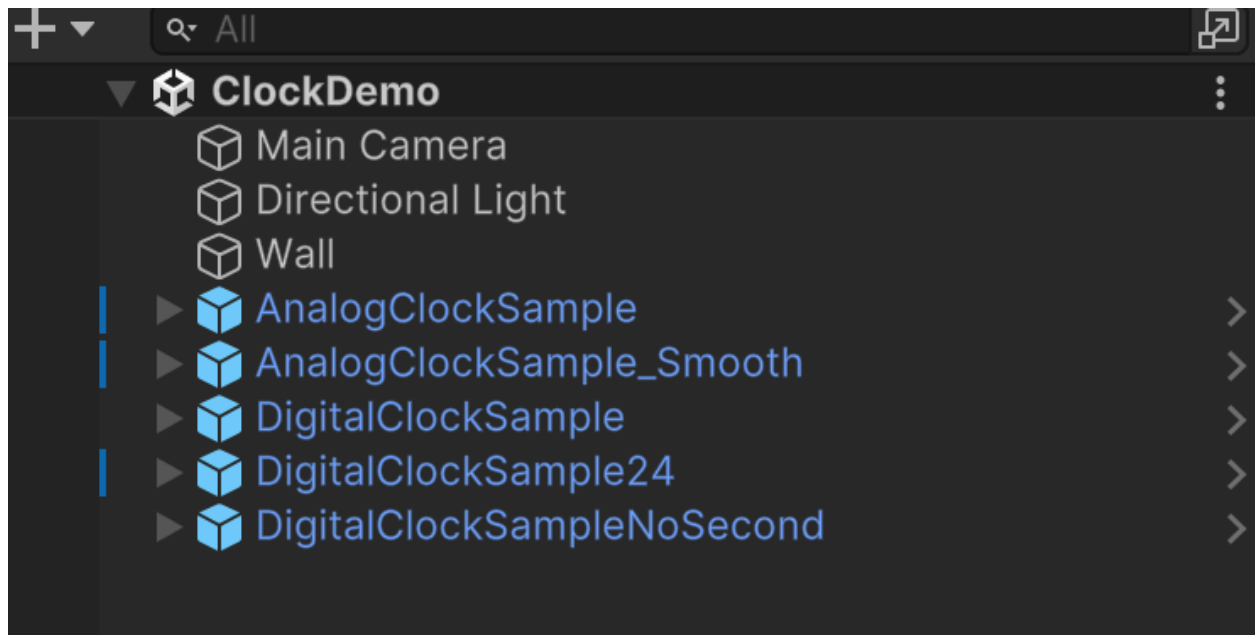
DG Realistic Clock provides analog and digital clocks that operate based on either real-time or a user-defined start time. Both clock types are implemented as components and can be attached to any GameObject. Simply configure the required references and it's ready to use.

2. Usage

Refer to the various clock prefabs included in the Demo for quick understanding.

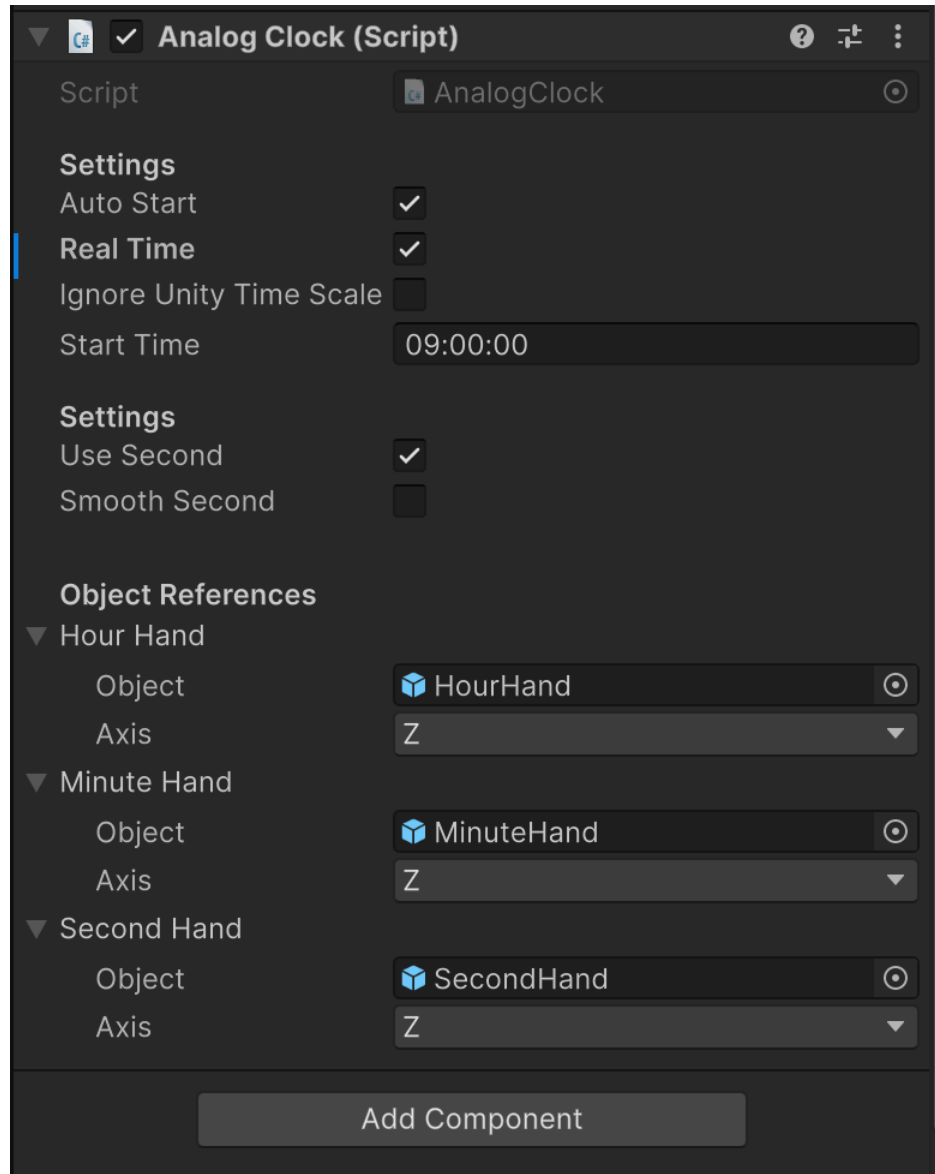
You may use them directly, but for better visuals, replacing the mesh may be desirable.

You can simply swap the mesh while keeping the prefab structure, or follow the guide below for full customization.



1. Analog Clock

1. Search for the **Analog Clock** component and add it to the root of your clock GameObject.
2. Adjust the necessary values in the **Settings** section.



- **Auto Start:** If enabled, the clock will start automatically at runtime. If disabled, you must call `Play()` manually in code.
- **Real Time:** If enabled, the clock uses real-world time. If disabled, it starts from the `startTime` field.
- **Ignore Unity TimeScale:** If true, the clock ignores Unity's timescale.
- **Start Time:** Used only when Real Time is false. Input a start time in 24-hour format.
- **Use Second:** Enables or disables the second hand.
- **Smooth Second:** If enabled, the second hand moves smoothly.

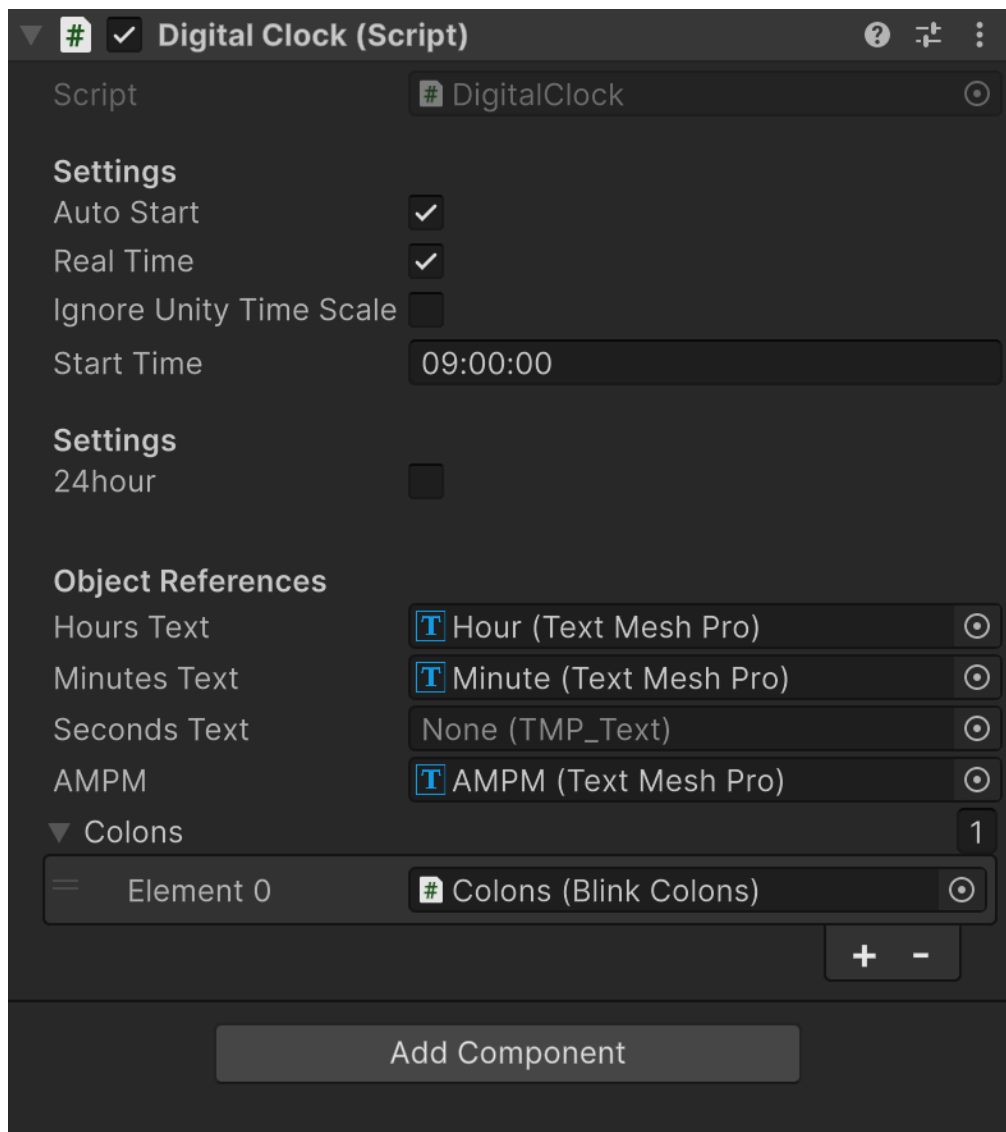
Then, set the **Object References**:

- Assign your hour, minute, and second hand objects.
- Set the **Axis** for each hand to determine the rotation axis (default: Z).

⚠ If "Use Second" is true but no second hand is assigned, an error will be thrown.

2. Digital Clock

Digital clocks share the same Settings section. Adjust them as needed.



- **24Hour:** If true, time is displayed in 24-hour format. If **AMPM** is assigned, it will still be hidden.

Assign Object References:

- **Hours Text**, **Minutes Text**, and **Seconds Text** each require their own TMP Text. This separation helps avoid unnecessary string allocations.
 - If **Seconds Text** is not assigned, seconds will not be shown.
 - Assign **AMPM** (optional TMP). If unassigned, AM/PM will not be displayed.
 - **Colons**: Assign a GameObject with **BlinkColons** component to enable blinking colons. If not needed, you may skip this or customize it freely.
-

3. Using Play via Code

Both AnalogClock and DigitalClock components expose a **Play()** method.

```
private void PlayClock()
{
    AnalogClock clock;

    clock.Play();
}
```

Call this method to start the clock.

Adjust Settings values before calling if needed.

```
private void PlayClock()
{
    AnalogClock clock;

    clock.SetSmoothSeconds(true);

    clock.RealTime = false;

    clock.SetStartTime("01:00:00");

    clock.Play();
}
```

Use **Stop()** to pause the clock.

```
private void PlayClock()
{
    AnalogClock clock;

    clock.Stop();
}
```

Various `Get~()` methods are available to retrieve the current time.

```
private void PlayClock()
{
    AnalogClock clock;

    clock.GetCurrentTimeSpan();

    clock.GetCurrentTimeString();

    clock.GetHours();

    clock.GetHours12();

    clock.GetMinutes();

    clock.GetSeconds();
}
```

+ Delegates

```
private void PlayClock()
{
    AnalogClock clock;

    clock.OnTimeChanged;
```

```
clock.OnStopped;  
  
clock.OnPlay;  
}
```

4. Feedback / Bug Reports

Please contact: gfr9905@gmail.com

日本語

1. 概要

DG Realistic Clock は、現実時間またはユーザー指定の開始時間に基づいて動作するアナログおよびデジタル時計を提供します。

どちらもコンポーネントとして提供され、任意のオブジェクトにアタッチし、必要な項目を設定するだけで即座に使用可能です。

2. 使用方法

デモに含まれている様々な時計プレハブを参考にすると、より早く理解できます。

プレハブをそのまま使っても構いませんが、見た目を改善するにはメッシュの差し替えが必要になるかもしれません。


構造を保ったままメッシュだけ差し替えるか、以下の手順で完全にカスタマイズすることも可能です。

1. アナログ時計

1. **Analog Clock** コンポーネントを検索し、時計オブジェクトのルートに追加します。
2. **Settings** セクションの各値を設定します。
 - **Auto Start**: 有効時、ゲーム開始と同時に自動で動作開始します。無効時は `Play()` をコードから呼び出す必要があります。
 - **Real Time**: 有効時は現実の時間を使用します。無効時は `startTime` に入力した時間から開始します。
 - **Ignore Unity TimeScale**: 有効時、Unity のタイムスケールを無視します。
 - **Start Time**: 24時間制で入力。Real Time が無効のときに使用されます。
 - **Use Second**: 秒針の使用有無を設定します。
 - **Smooth Second**: 有効時、秒針が滑らかに動きます。

Object References を設定します:

- 時針、分針、秒針のオブジェクトをそれぞれ割り当てます。
- 各針の回転軸 (Axis) を設定します。デフォルトは Z です。

 Use Second が true かつ秒針が割り当てられていない場合、エラーが発生します。

2. デジタル時計

設定項目はアナログ時計と共通です。用途に合わせて調整してください。

- **24Hour**: 有効時、24時間表記になります。AMPM オブジェクトが設定されていても表示されません。

Object References を設定:

- **Hours Text、Minutes Text、Seconds Text** にはそれぞれ別々の TMP Text を割り当ててください。
(文字列更新の無駄を避けるためです)
 - **Seconds Text** が未設定の場合、秒は表示されません。
 - **AMPM** を設定すると AM/PM を表示します(任意)。
 - **Colons** には **BlinkColons** コンポーネントが付いたオブジェクトを設定してください。未設定の場合は点滅しません。
-

3. コードからの起動

AnalogClock および **DigitalClock** は **Play()** メソッドを持っています。

これ呼び出すことで時計が起動します。

開始前に設定項目を変更できます。

停止が必要な場合は **Stop()** を呼び出します。

時間取得には **Get~()** 系の関数を使います。

4. バグ報告・ご意見・質問

gft9905@gmail.com までご連絡ください。

简体中文

1. 概要

DG Realistic Clock 提供模拟和数字时钟，可基于现实时间或用户定义的起始时间运行。两种时钟均作为组件提供，可附加到任意对象上，只需设置必要参数即可使用。

2. 使用方法

建议参考 Demo 中包含的各种时钟预制体，以快速理解使用方式。
你可以直接使用这些预制体，但为了视觉效果，建议根据需要替换模型。
你可以仅替换网格，保留结构，或按照下方说明进行完全自定义。

1. 模拟时钟

1. 搜索并将 **Analog Clock** 组件添加到目标对象的根节点。
2. 设置组件中的 **Settings** 项目：
 - **Auto Start**: 为 true 时，游戏开始自动启动时钟；否则需手动调用 `Play()`。
 - **Real Time**: 为 true 时使用现实时间；否则使用 `startTime`。
 - **Ignore Unity TimeScale**: 为 true 时忽略 Unity 的 TimeScale。
 - **Start Time**: 当 Real Time 为 false 时有效，需以 24 小时制格式输入。
 - **Use Second**: 是否启用秒针。
 - **Smooth Second**: 为 true 时，秒针平滑移动。

设置 **Object References**:

- 分别指定时针、分针、秒针的对象。
- 设置每个指针的旋转轴(默认 Z 轴)。

 若启用秒针但未设置对象，将出现错误信息。

2. 数字时钟

Settings 设置与模拟时钟相同，可根据需要调整。

- **24Hour**: 为 true 时使用 24 小时制，即使设置了 AMPM 也不会显示。

设置 **Object References**:

- 分别指定 **Hours Text**、**Minutes Text**、**Seconds Text** 的 TMP 组件。
分开设置可避免不必要的字符串更新。

- 若未指定 Seconds Text, 则不显示秒。
 - 设置 **AMPM** TMP 可显示 AM/PM(可选)。
 - **Colons** 需要挂载 **BlinkColons** 组件的对象才能实现闪烁效果。
-

3. 代码启动

AnalogClock 和 **DigitalClock** 都提供 **Play()** 方法。

调用此方法即可启动时钟。

你可以在调用前调整 Settings 参数。

如需停止, 使用 **Stop()** 方法。

使用 **Get~()** 方法获取当前时间信息。

4. 问题反馈

请发送邮件至 **gftr9905@gmail.com**

한국어 – 도큐먼트

1. 요약

DG Realistic Clock은 아날로그 시계와 디지털 시계를 실제 시간 또는 사용자가 지정한 시작 시간에 따라 구현할 수 있는 다용도 시계 시스템입니다.

두 시계 모두 컴포넌트 형태로 제공되며, 원하는 오브젝트에 부착 후 필수 요소만 설정하면 즉시 사용할 수 있습니다.


2. 사용 방법

데모 활용

Demo 씬에 포함된 다양한 Clock 프리팹을 참고하면 더 빠르게 이해할 수 있습니다. 프리팹을 그대로 사용해도 되지만, 시각적인 완성도를 위해 메쉬 교체가 필요할 수 있습니다. 기본 구조를 유지한 채 메쉬만 교체해도 되고, 완전히 커스터마이징하고 싶다면 아래 설명을 참고하세요.

1. 아날로그 시계

1. **Analog Clock** 컴포넌트를 검색하여 시계 오브젝트의 최상위 **GameObject**에 추가합니다.
2. **Settings** 항목을 설정합니다:
 - **Auto Start:** **True**일 경우 게임 시작 시 자동으로 시계가 작동합니다. **False**일 경우 **Play()** 함수를 직접 호출해야 합니다.
 - **Real Time:** **True**일 경우 실제 시간 기준으로 작동합니다. **False**일 경우 **StartTime** 값에서 시작합니다.
 - **Ignore Unity TimeScale:** **True**일 경우 유니티의 **Time.timeScale**을 무시하고 작동합니다.
 - **StartTime:** **Real Time**이 **False**일 때 사용되며, 24시간제 형식(**HH:mm:ss**)으로 입력해야 합니다.
 - **Use Second:** 초침 사용 여부를 설정합니다. **False**일 경우 초침이 무시됩니다.
 - **Smooth Second:** **True**일 경우 초침이 부드럽게 움직입니다.
3. **Object References** 항목을 설정합니다:
 - **Hour Hand, Minute Hand, Second Hand:** 각각 시, 분, 초침 **GameObject**를 할당합니다.
 - 각 핸들의 **Axis**는 회전 기준 축을 의미하며 기본값은 Z입니다.

 **Use Second**가 **True**인데 초침 오브젝트가 설정되지 않으면 에러가 발생합니다.

2. 디지털 시계

Settings 항목은 아날로그 시계와 동일하며, 필요에 따라 설정합니다.

- **24Hour**: **True**일 경우 24시간제로 표기되며, **AMPM** 항목이 설정되어 있어도 출력되지 않습니다.
 - 3. **Object References** 항목을 설정합니다:
 - **Hours Text, Minutes Text, Seconds Text**: 각각의 시간을 표시할 **TextMeshPro** 컴포넌트를 개별로 할당합니다.
→ 이는 불필요한 **string** 갱신을 줄이기 위한 구조입니다.
 - **Seconds Text**를 할당하지 않으면 초는 표시되지 않습니다.
 - **AMPM**: **TextMeshPro** 오브젝트를 할당하면 **AM/PM**이 출력됩니다. 미할당 시 표시되지 않습니다.
 - **Colons**: **BlinkColons** 컴포넌트가 부착된 **GameObject**를 할당하면 깜빡이는 콜론이 활성화됩니다.
필요 없거나 직접 구현하고 싶다면 자유롭게 대체할 수 있습니다.
-

3. 코드 기반 사용

- **AnalogClock** 및 **DigitalClock** 클래스에는 **Play()** 메서드가 포함되어 있으며 호출 시 시계가 동작합니다.
 - **Play()** 호출 전에는 **Settings** 값을 조절할 수 있습니다.
 - 시계를 정지하려면 **Stop()**을 호출합니다.
 - **Get~()** 메서드들을 통해 현재 시각 정보를 가져올 수 있습니다.
-

4. 버그 제보 및 문의

문제 발생 시 또는 기능 건의는 아래 메일로 연락주세요.

 gfr9905@gmail.com