Before anything else…

INSTANT GOOD DAY

Thank you.

Thanks for choosing Instant Good Day to be used on your project. It has been a real pleasure for me to do this, my first Unity3d package.

Since I started this project, my Love and respect for nature has grown exponentially since it is a real challenge trying to mimic its greatness, its complexity, its balance and its perfection, so I have put all my dedication to achieve a good result but also to keep it as simple as I could.

I hope for you the best of success on all of your projects using Instant Good Day, I will feel very proud to know I helped in some way on this regard.

Amable Rodríguez

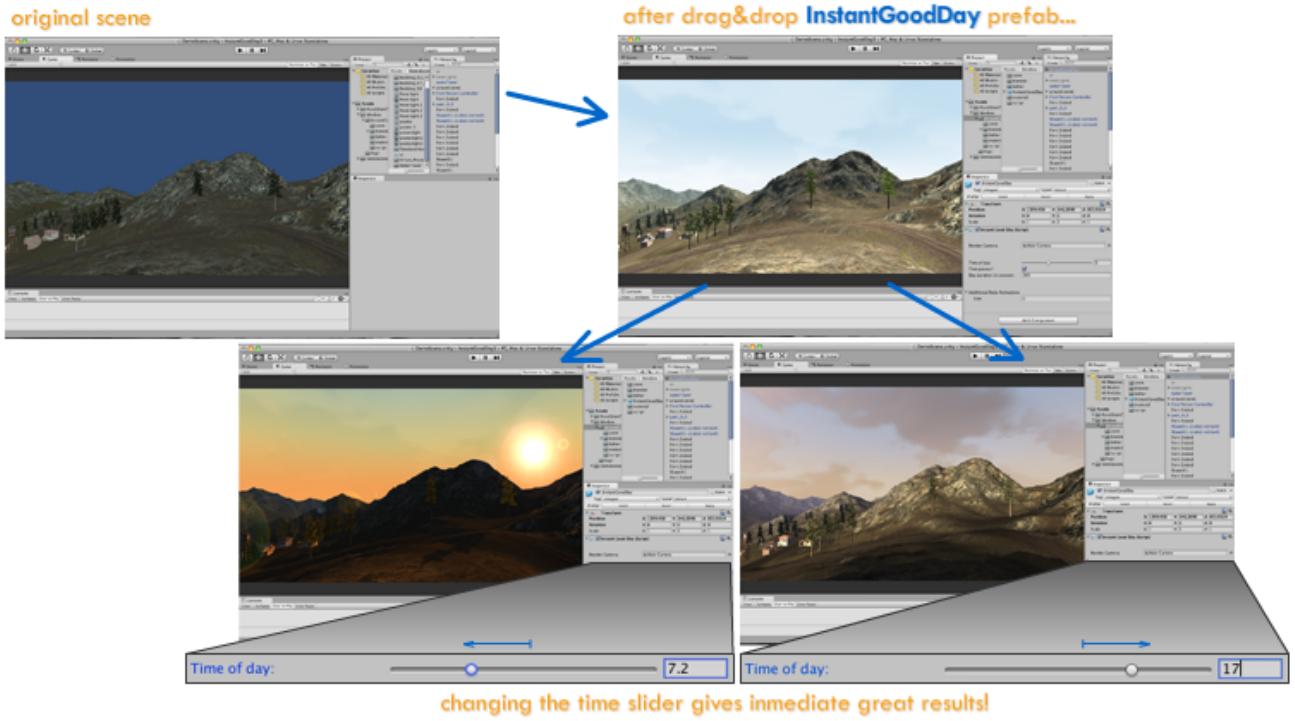
Table of Content

INSTANT GOOD DAY

Workflow	4
Editor	6
Editor : Render Camera	7
Editor : Time of Day	8
Editor : Time passes?	9
Editor : Day duration in seconds	10
Editor : Additional Daily Animations	11
Using scripts to control the prefab	13
API Reference	14
FAQ	16
Troubleshooting	18
Epilog	19

Workflow

Add a realistic day/night cycle only by drag-and-dropping the prefab to your scene, change the hour by moving a simple slider and you will see the result instantly on the editor.



You will obtain instant realistic outdoor environment with sky, clouds, sun, moon, stars, fog and environmental lights all harmoniously coordinated on a smooth day/night animation, super easy to use.

Workflow

INSTANT GOOD DAY

It is really easy and fast to obtain a high quality result:

1) drag-and-drop the InstantGoodDay prefab to your scene, it will automatically detect your camera (you can also assign it manually)

2) change the hour of the day and the 24h cycle duration (you can also stay on a fixed hour if you don't want the time to pass)

3) hit play and have an Instant Good Day!

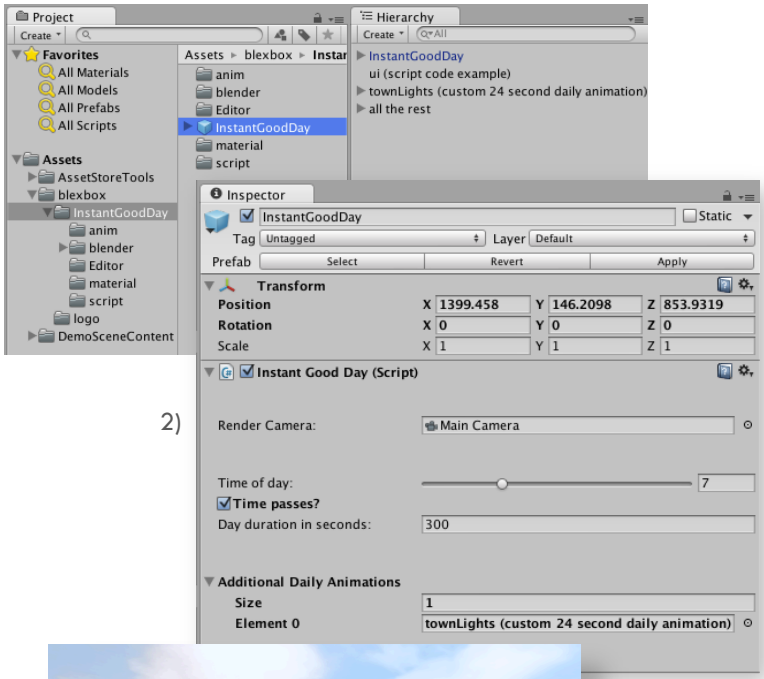
(optional) :

- add your own custom daily animation to the list so it will be synced with the day/night cycle allowing you to do things at specific time of the day (example: turning the street lights on at night and off during the day like on the demo).

- you can also check for the current time of the day with a simple line of code! (example: you can ask what time is it at any moment like it is done to show the clock on the demo)

You can see a video of this workflow at this address:
https://www.youtube.com/watch?v=wQw5b_pGgzE

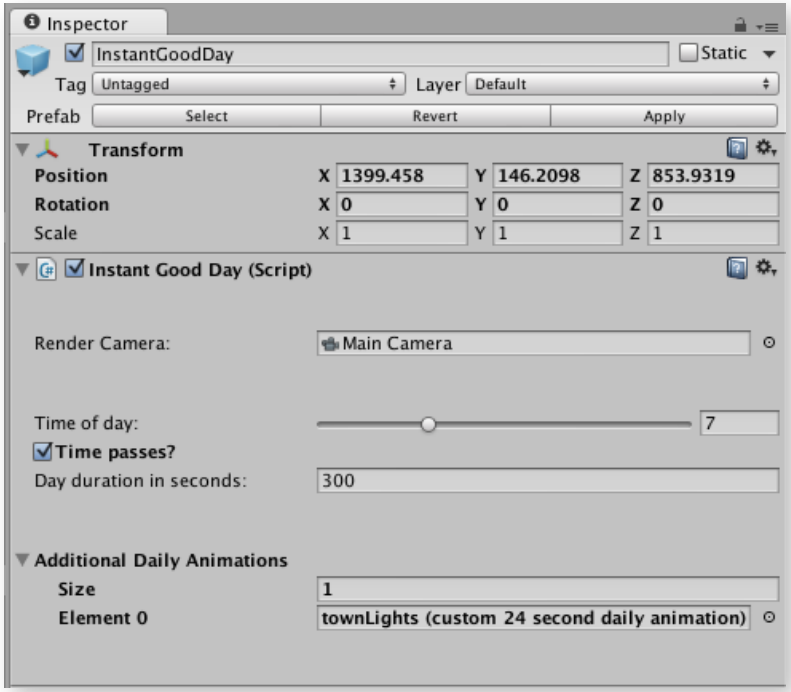
1)



2)

3)

Editor



From the editor we can set-up the prefab once it is dropped to your scene. You will notice that it has only the minimal controls needed.

It has been made the simplest way possible to ensure a minimal impact on your project and to assure that anyone can use it.

We can identify 3 groups of controls:

- specify the Camera
- control the time
- assign the daily animations

Note: you can also control all this properties at run-time from your scripts (see API Reference section for more detail)

Now let's see each option in more detail.

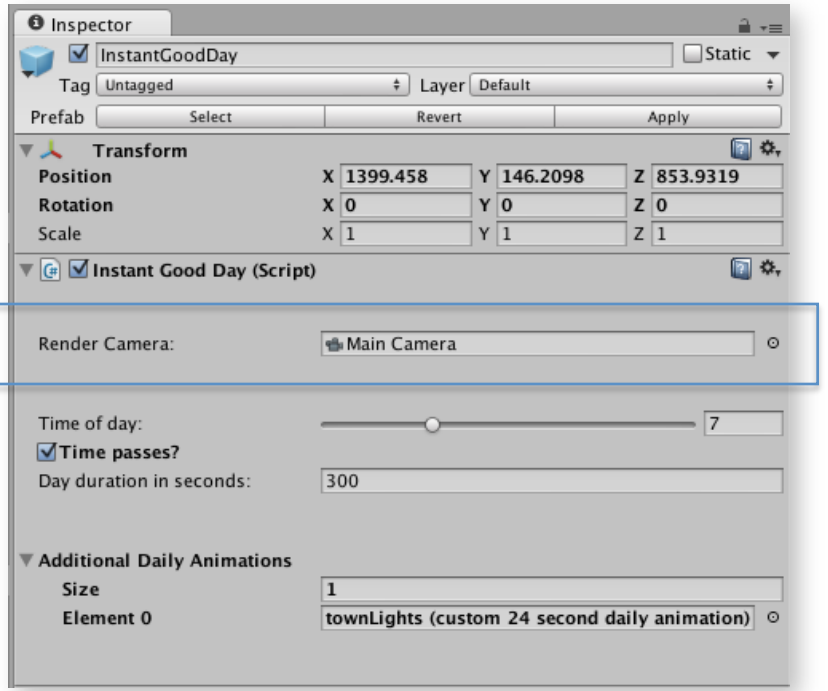
Editor : Render Camera

INSTANT GOOD DAY

Drag-and-drop the InstantGoodDay prefab to your scene, it will automatically detect your camera by searching all the cameras having a name like "Main Camera", "MainCamera", "Camera" or "Camara".

If the desired Camera to be used by you have another name and it couldn't be found automatically or if you just want to specify it manually, you can assign it by drag-and-drop your camera to the Render Camera property.

You can also have access to this property using the proper method called from your script (see API Reference section for more detail).



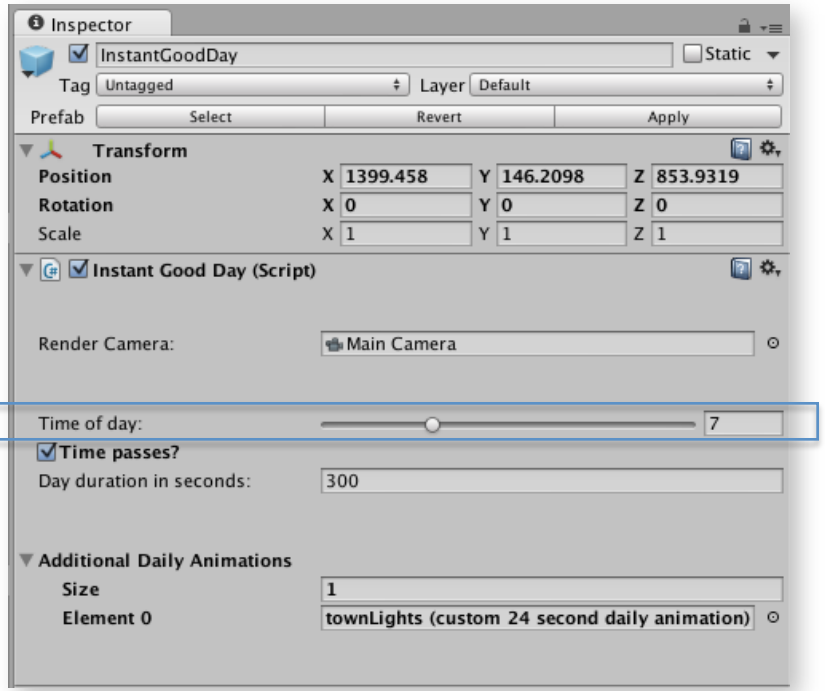
Editor : Time of Day

INSTANT GOOD DAY

Change the hour of the day by using the editor slider or assign a value directly on the value square next to the slider.

This is a numeric value so for example to represent 7:30am you will set it to 7.5 instead.

You can set and get the value on your scripts with couple of methods that work with this numeric value but also you have two others that works in military format HH:MM, we make use of this on the demo in order to show the current time on the ui (see API Reference section for more detail).



Editor : Time passes?

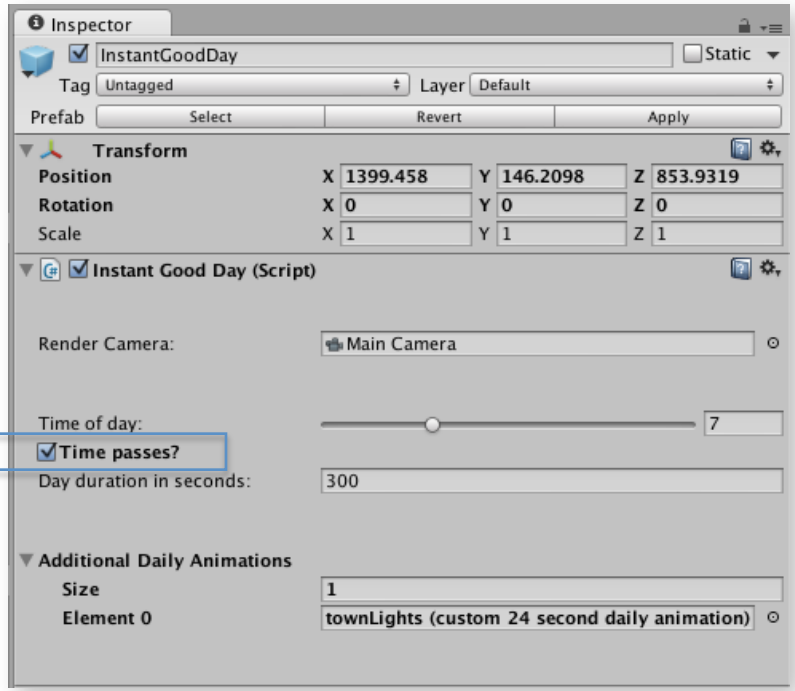
INSTANT GOOD DAY

As default this option comes enable so the time pass and we see the day turn into night, etc.

You can also stay fixed on a particular time of the day by disabling the "Time passes?" option so everything except the clouds and stars will be static.

Clouds and stars will keep its natural movement no matter if the time remains static or, if you set it to pass, no matter the duration value you choose.

You can also access this property in your scripts at any moment at run-time (see API Reference section for more detail).

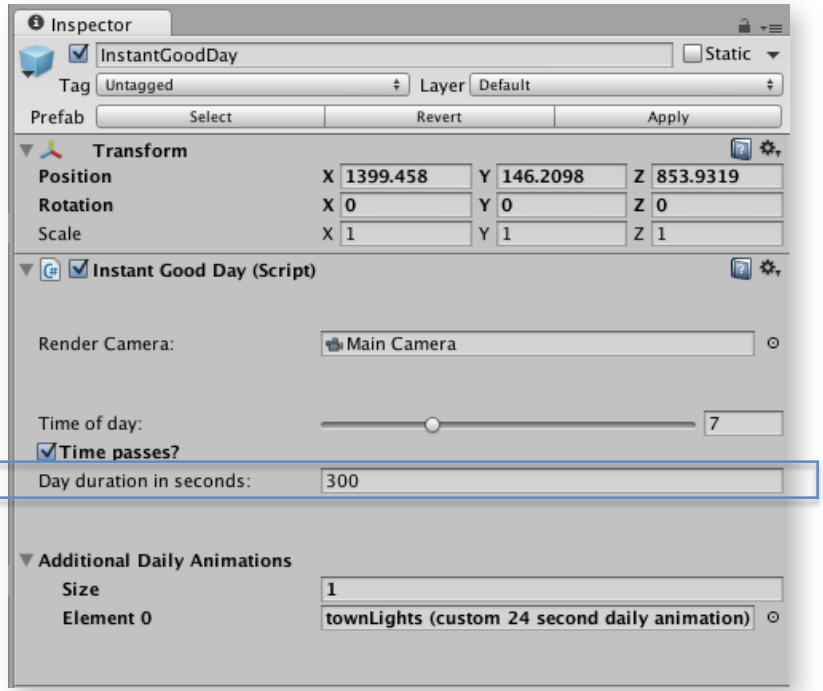


Editor : Day duration in seconds

In order to make the time pass and see the day progressing over the time, you need to enable the "Time passes?" option, but once enabled you can specify how much real-time will it take for a 24h day to complete its cycle.

This value is expressed in seconds, so for example a value of 300 is equivalent to 5min. This value will be used to adjust the animation speed.

You can also access this property in your scripts at any moment at run-time (see API Reference section for more detail).



Editor : Additional Daily Animations

INSTANT GOOD DAY

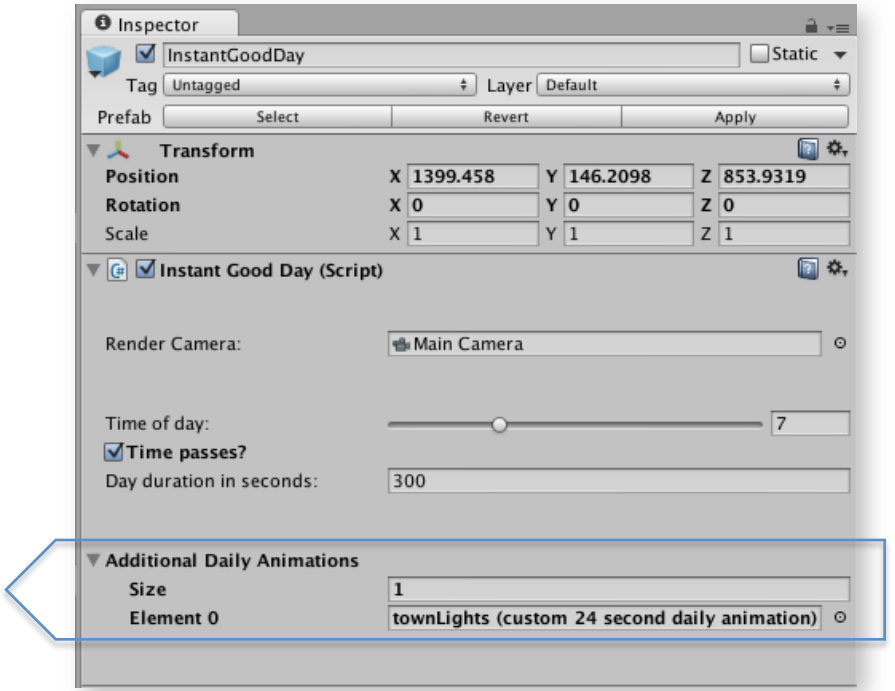
You can add you own custom animations to the list so they will be played in sync with the day cycle. This animations have to be 24 seconds long representing the 24 hours of a day.

In order to do this, select your GameObject and click Add Curve button on the Unity's animation panel and do the animation been sure that it is 24 seconds long (refer to Unity3d documentation for more detail about how to animate a GameObject).

After your GameObject is animated, change the Size of the list in order to have a new element and then just drag and drop your animated Game Object to the list.

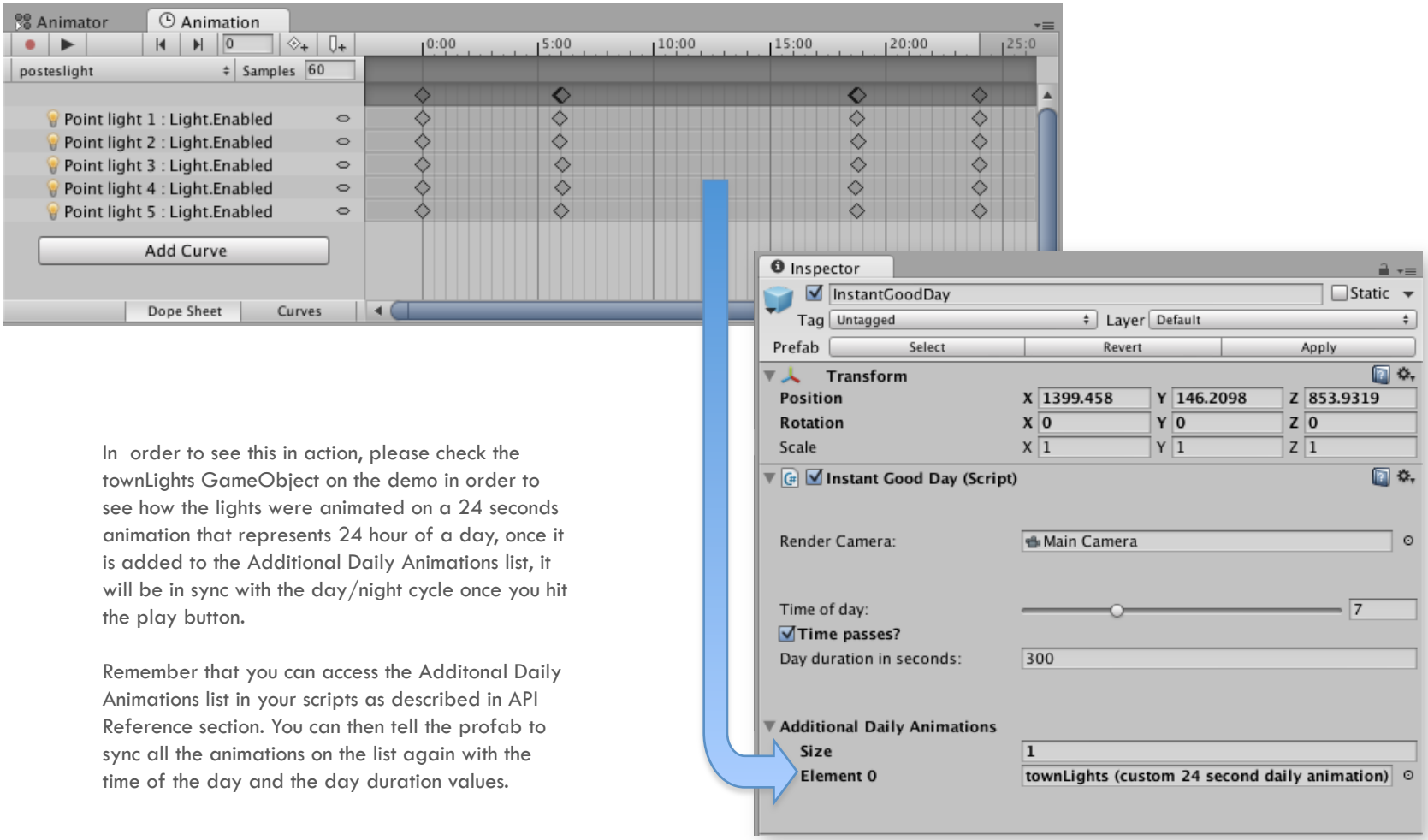
So if for example as in the demo, you want to turn on the lights at 18:30pm, you will set a keyframe on your 24 seconds animation at 18:30 with your lights on. Once you have completed the animation as you want it and you be sure it is 24 seconds long, you change the size of the list and drag and drop the animated GameObject into the "Additional Daily Animations" list on the editor.

You can also have access at run-time to this list on your scripts (see API Reference section for more detail).



Editor : Additional Daily Animations

INSTANT GOOD DAY



In order to see this in action, please check the townLights GameObject on the demo in order to see how the lights were animated on a 24 seconds animation that represents 24 hour of a day, once it is added to the Additional Daily Animations list, it will be in sync with the day/night cycle once you hit the play button.

Remember that you can access the Additonal Daily Animations list in your scripts as described in API Reference section. You can then tell the profab to sync all the animations on the list again with the time of the day and the day duration values.

Using scripts to control the prefab

Even when you can use this prefab with no need of doing a single line of code, you can for sure manipulate every property from your scripts at run-time.

Let's take the following examples:

- let's say that in your application you need to have several cameras on a race game, all you need to do is inform the prefab about what is the current camera once it changes, so you will set this value with the selected camera once the change has been done informing the prefab about it, the prefab will then adjust to that camera from now on so the render images produced by it will be right as it supposed to be

- let's say that in your game the player is going to sleep and it supposed to wake-up 8 hours later, so you could get the current time value, add the 8 hours and set it back again, changes will be immediately visible

- let's say that in your game you have a little animal character controlled by some AI Agent, so let's say that it supposed to be active during the day and sleeping during the night, implementing this will imply that the character will ask the prefab about what time is it, then it will make decisions regarding its behavior according to the time of the day

- let's say that you hit pause button, so then we demand the prefab to stop time from passing, later on the un-pause event we would tell the prefab to let the time pass again

- let's say that in your game the player can choose between experiencing a normal time or a fast time simulation like in many RTS games, so you could have a duration value associated to each speed and depending of the speed demanded you set the new amount of time taken by the day to complete, this will change the animation speed right away

- let's say that in your game you have a factory randomly creating animated flowers that have a 24 second animation representing its behavior during the day, and let's say that you just have create a new one, so in this case you will ask the prefab about the current Custom Animation List, then you will add the new element and demand the animations to be sync. Let's say that one of this flowers just died and you need to removed it from the scene, so in this case you just ask for the list and then remove the element.

All this possible scenarios can be done by using the methods described on the API Reference section. Take a look at the ui class file which is part of the demo for more concrete examples.

Controlling the Camera property

void SetRenderCamera(Camera camera)
This method set the current Render Camera value, it is needed in order to adjust the scale of the prefab to fit the Camera's clipping planes far distance and also to adjust to the Camera position at all times.

Camera GetRenderCamera()
This method returns the current Camera value.

Controlling the Time of Day in numeric format

void SetNumericHour(float value)
This method is used in order to set the current hour of the day. Valid values are between 0 and 23.99. Since it is in numeric format, the minutes part of the hour are base 100 and not base 60 as on the clock, so for example if we want to set the time to 7:30am, we will use 7.5 as value instead.

float GetNumericHour()
This method is used to obtain the current value for the hour of the day in numeric format.

Note: Accessing the hour in a numeric format is the most efficient method since it doesn't employ any conversion, use this if you don't expressly need to show the time in the formatted way.

Controlling the Time of Day in string format

void SetMilitaryHour(string value)
This method is used in order to set the current hour of the day. Valid value is an string with the HH:MM format.

string GetMilitaryHour()
This method returns the current hour of the day as an string with HH:MM format.

Note: This is ideal when there is the express need to use a human friendly format (for example to display the time), but since it is internally transformed to its numeric equivalent, it is not recommended to use it on your logic unless you really need it. Using the numeric equivalent instead is preferred for performance.

Play and Stop the time

void PassTime()
This method is used to let the day/night animation to run freely.

void StopTime()
This method is used to stop the day/night animation. Note that the Stars and Clouds animations will keep its fixed movement.

24h Day Duration in real time

void SetDayDurationInSeconds(int value)
Use this method to establish the day cycle duration. Given value has to be between 1 and int.MaxValue. This value represents the amount of seconds in real time that will be needed to complete a 24 hours cycle in-game. So setting a value of 300 will make the prefab to adjust its speed to do a day in 5min, also all the other animations will be sync with the main day/night animation.

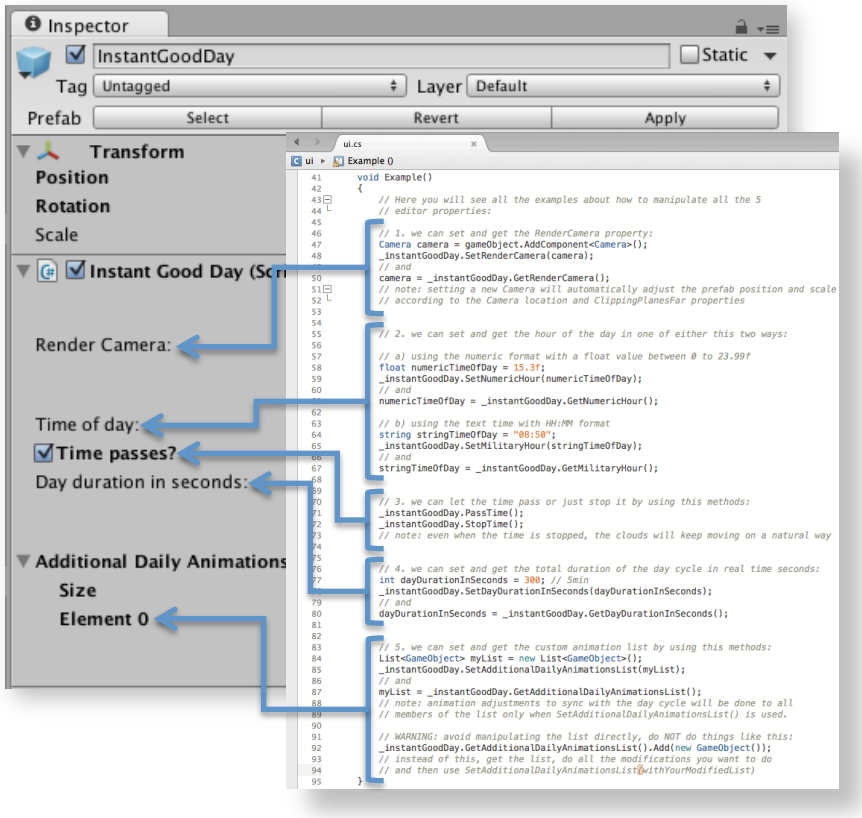
int GetDayDurationInSeconds()
This method returns the current day cycle duration in seconds.

Additional Daily Animations list

List<GameObject> GetAdditionalDailyAnimationsList()
Use this method to obtain the additional daily animation list. This list contains animated GameObjects, this animations are 24 seconds long and represent the 24 hours of a day.

Note: if you do some modification to this list just remember to call SyncDailyAnimations() method right after so all the animations will be in sync.

void SyncDailyAnimations()
Use this method to force re-sync all the additional daily animations with the main daily animation.



Please take a look at examples of usage on the ui class in the demo, there you will find a method named Test() where you will find call examples in order to manipulate all the same editor properties by code.

FAQ

Q. **How much time did I saved buying this prefab vs. doing it all from scratch by myself?**

A. Well, of course that depends on your experience, the scope of what you want to achieve and the approach you will use to make it.

I took all the time needed to do something that I really like. My goal was to end up with something that was really high quality but also practical. So that is the main gain on time that we have using this prefab.

Time wasted on doing and re-doing and tweaking and re-tweaking the environmental elements each time on each new project wasn't and option. This time is saved and you can use it on your real project.

I spent a lot of time looking at the sky and trying to mimic the things I saw, I did a lot of different tests, try-error long loops, until I found, finally, the balance that I was looking for. This took me time for sure, but I had a clear picture of what I wanted as result and I didn't stop until I found it. And this took me time. So let's say that you have saved that time yourself.

There is an important amount of time that you have saved already by using this prefab instead of some others on the market: this is the most simple to use by far and with excellent results despite of this. Too many controls on it and you could end up with a super complex monster not always easy to tweak and not necessarily obtaining a good realistic result either.

So, at the end I can't really answer this question with a measure but, what I can tell you is this: I took all the time needed to arrive to something that is really simple made, good to see and easy to use, and now you save all the time that you would otherwise would use creating something like this by yourself, so now you can concentrate on your real project where you will be just adding this prefab as an environmental effect.

Q. **Is there rain or other kind of weather?**

A. Not on this version since the objective was to create a general purpose single day/night animation with the strict minimal controls on it and the weather will make it way more complex in all senses.

In my experience at least, 80% of the times we need to do some kind of outside environment effect is just about having a clear day or night setting since it will be just a surrounding part of the real game or application and not the focus, since the real focus of the player is on the core of your project instead.

Q. **Can I modify the stars and clouds animation speed?**

A. You can alter the prefab at your own risk in order to play with the animations to adapt them to your needs, in this particular case you will have to alter the Stars and Clouds animation manually.

This is not big deal but it was just out of the scope of this prefab.

So, there is no editor control made for this parameters for the moment. Feel free to modify the prefab always knowing that you can return to the original if something goes wrong.

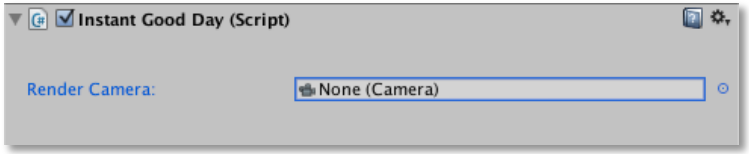
Troubleshooting

"I drag-and-drop the prefab to my scene but my Camera is not been auto-detected."

R. The name used on your Camera is not something near to "Main Camera". In fact, "Main Camera", "MainCamera", "Camera" or "Camara" are the names that the prefab will use in order to try to guess what is the camera it supposed to be attached to.

If the desired Camera to be used by you have another name and it couldn't be found automatically or if you just want to specify it manually, you can assign it by drag-and-drop your camera to the Render Camera property.

You can always assign it directly on the editor but also at run-time using the proper script call (see API Reference section for more detail).



"I see a weird circle on the sky (and I don't like it!)"

R. The Camera generating this kind image is not the one specified on the Render Camera property. You must tell Instant Good Day prefab what is the Camera to be attached to.

You can change this value even at run-time using the scripts method (see API Reference section for more detail), once it is set correctly, the prefab will stay adjusted to that Camera's clipping planes far distance and position on the scene so you will see its rendered image correctly.

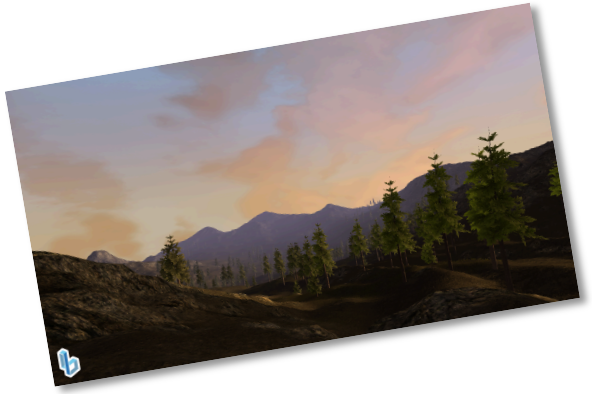
Troubleshooting

"I change the parameters on the editor but when I hit play I don't see the changes."

R. Maybe in some part of the code you are changing this values using some of the calls described on the API Reference, so when the play button is pressed, the code is executed overwriting the options entered on the editor window.

"I only get NaN:NaN as return value when calling GetMilitaryHour(), and the animation doesn't move"

R. This is most likely due to a zero value in the Time of Day property. Please use a value bigger than 0.



"I get negative values when calling the GetNumericHour() and the animation doesn't move"

R. This is most likely due to a negative value in the Time of Day property. Please use a value bigger than 0.

"I have an error Ambient component not found"

R. The internal link between the main prefab script and its ambient component is broken somehow, if you are not able to plug it back by yourself you will have to revert the prefab to its original state. This doesn't happen unless we modify the prefab manually.

Epilog

Instant Good Day prefab has been made the simplest way possible to ensure a minimal impact on your project. No complex scripts, no expensive shaders. You don't need any programming skills at all. You don't need to adjust hundreds of confusing editor options when all you need is a simple instant good day setup that just works right from the box.

Each animation curve, each color transition, each light intensity variation during each single moment of the day has been manually tweaked creating a natural behavior of real life day elements to obtain the most realistic experience that generates a balanced mood during the day.

All this work has generated this prefab that is yours today. I hope you find this a good fit for your project's needs.

Best of lucks and, have an **Instant Good Day**!