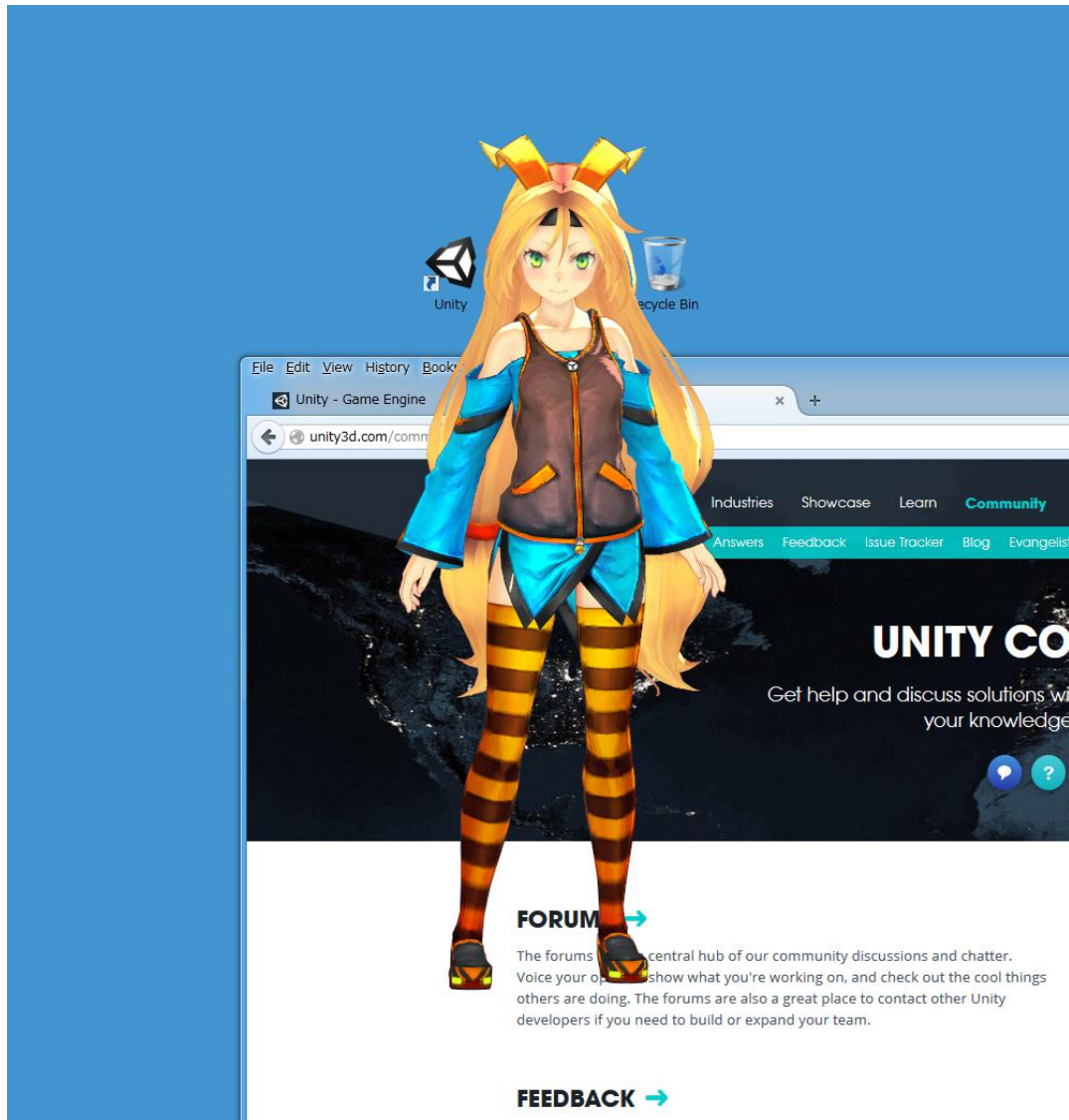


Desktop Mascot Maker Reference Manual

Version 1.9.0 2018/01/14 by PanzerSoft



Desktop Mascot Maker is a tool for creating your mascot character with Unity.

You can make any 3D/2D mascot with extremely easy steps!

Limitations

Before you use this asset, please read the followings.

- **The target platform of this asset is Windows only.**

Features

Before you use this asset, please read the followings.

- **Ready for Personal edition and Professional edition**
- **Ready for uGUI (Image and Text only)**
- **Full control of the mascot from Unity**
- **Lots of event handlers**
- **Detect mouse hovering**
- **Mascot's opacity change**
- **Unity's main window opacity change**
- **Multiple mascots display (with MascotMakerMulti component)**
- **Source code included (MascotMaker.cs, MascotMakerMulti.cs)**

Contacts

<http://www.assetstore.unity3d.com/en/#!/publisher/9484>

Table of Contents

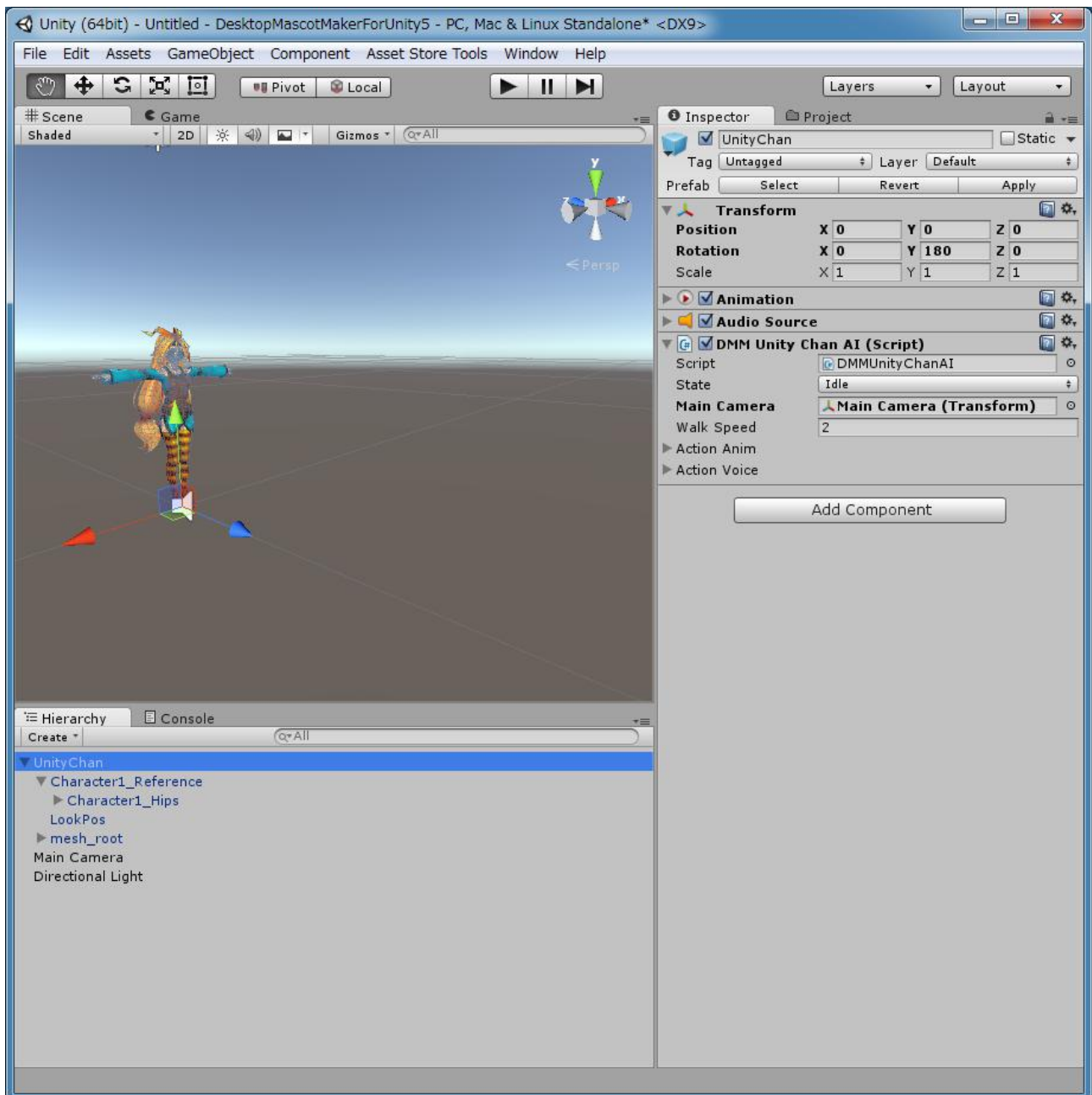
Quick Start.....	4
Chroma Key Compositing.....	11
Display of Multiple Mascots.....	13
Usage of uGUI.....	14
How to hide Unity's main window.....	15
MascotMaker Class Reference.....	16
MascotMakerMulti Class Reference.....	22
Frequently Asked Questions.....	29

Quick Start

Place Your Mascot

Create new scene and place your mascot in the scene.

Tweak the directional light intensity.

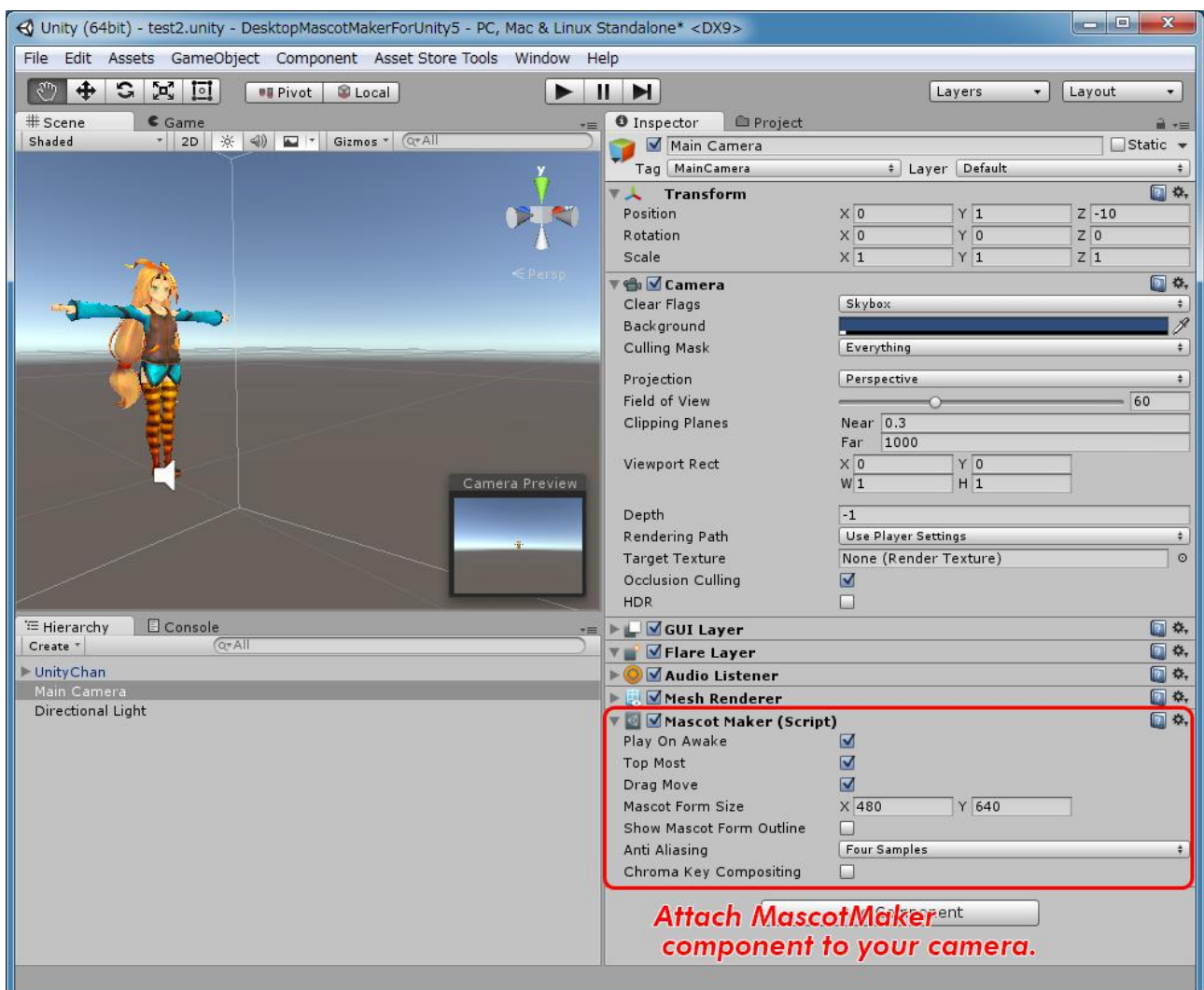
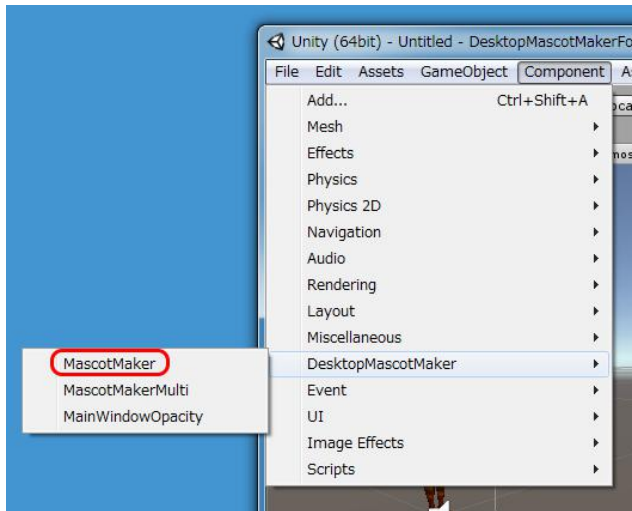


Attach MascotMaker Component to Camera

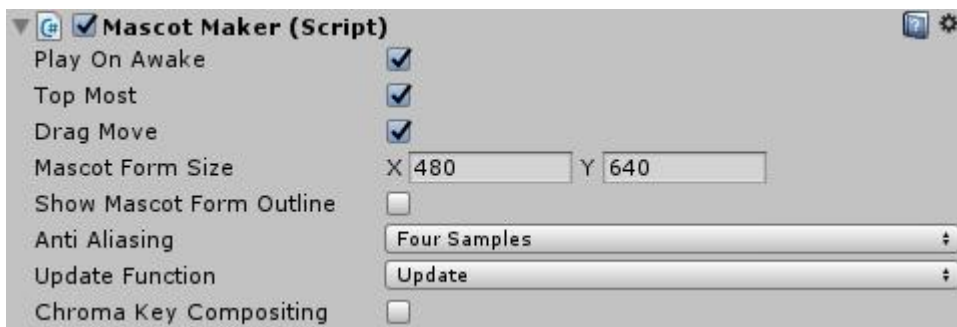
Attach MascotMaker component to the Main Camera.

You can add MascotMaker component through

Menu > Component > DesktopMascotMaker > MascotMaker



MascotMaker component settings



Play On Awake (bool value)

If set to true, the mascot will automatically show on awake.

Top Most (bool value)

If set to true, the mascot will be displayed as a topmost form.

Drag Move (bool value)

If set to true, the mascot can be moved by left mouse button dragging.

Mascot Form Size (Vector2 value)

You can change Mascot Form's size here. (NOTE : can also change at runtime)

Show Mascot Form Outline (bool value)

This check box is for debug.

If set to true, Mascot Form's outline will be visible and you can check its size.

Anti Aliasing (enum value)

Specify anti-aliasing level here.

Update Function (enum value)

Specify update function type (Update or LateUpdate).

In most cases, Update is OK. Only when the rendering is not good, try LateUpdate.

Chroma Key Compositing

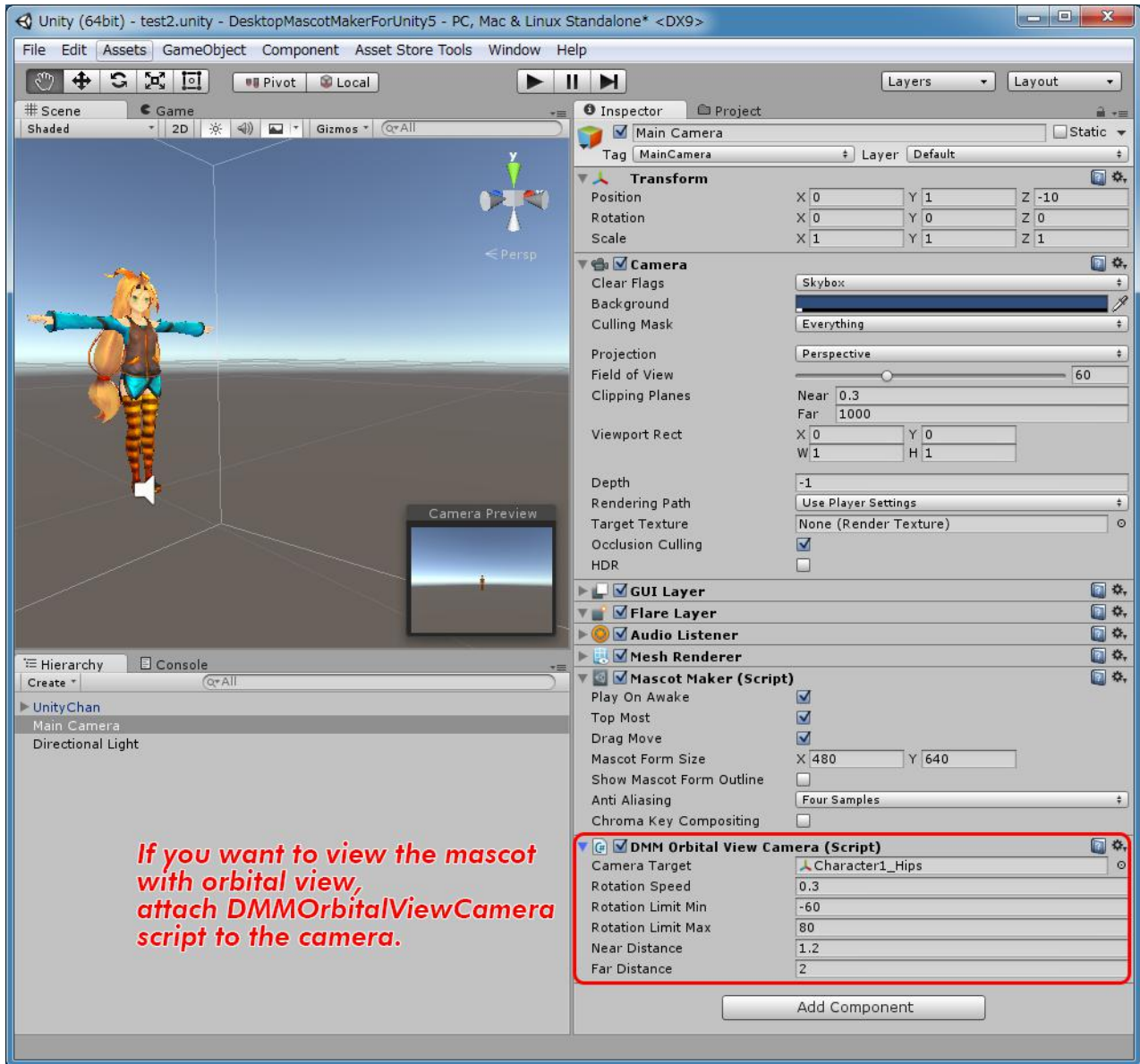
If set to true, Chroma Key Compositing mode will be ON.

You can also change these settings via scripting

```
bool MascotMaker.Instance.PlayOnAwake           // bool value
bool MascotMaker.Instance.TopMost                // bool value
bool MascotMaker.Instance.DragMove               // bool value
Vector2 MascotMaker.Instance.MascotFormSize      // Vector2 value
bool MascotMaker.Instance.ShowMascotFormOutline  // bool value
MascotMaker.AntiAliasingType MascotMaker.Instance.AntiAliasing // enum value
bool MascotMaker.Instance.ChromaKeyCompositing   // bool value
```

Attach Camera Script to Camera

If you want to view your mascot with orbital view, attach DMMOrbitalViewCamera.cs script to your camera. Then assign something to Camera Target variable. In this example, UnityChan's Character1_Hips is assigned. This script is optional. You don't have to attach this script if you don't want orbital view.



DMM Orbital View Camera Settings

Camera Target : Camera's target to look at.

Rotation Speed : Camera's rotation speed.

Rotation Limit Min : Minimum limit of vertical rotation .

Rotation Limit Max : Maximum limit of vertical rotation.

Min Size : Minimum size of orthographic view. Only for orthographic camera.

Max Size : Maximum size of orthographic view. Only for orthographic camera.

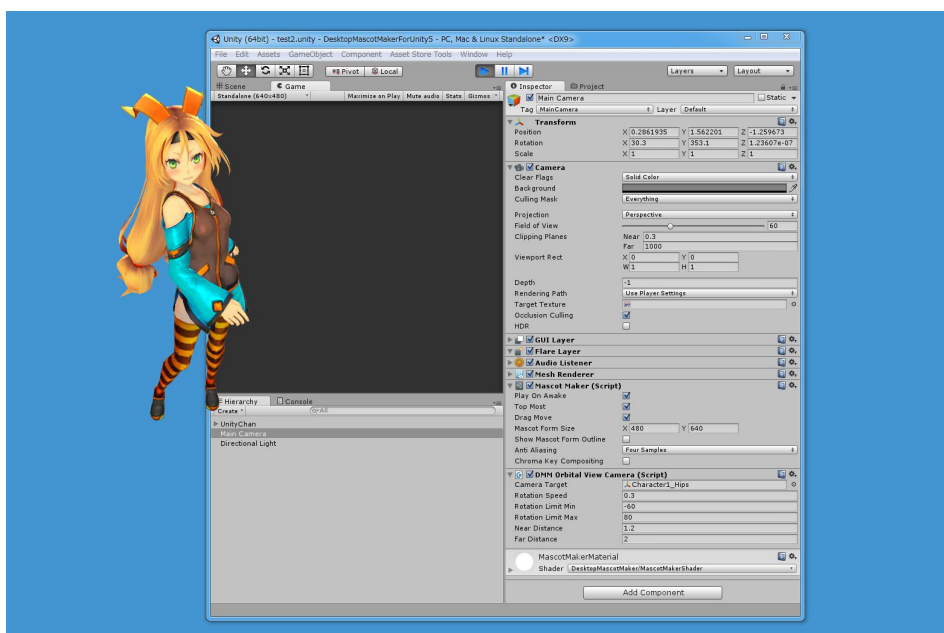
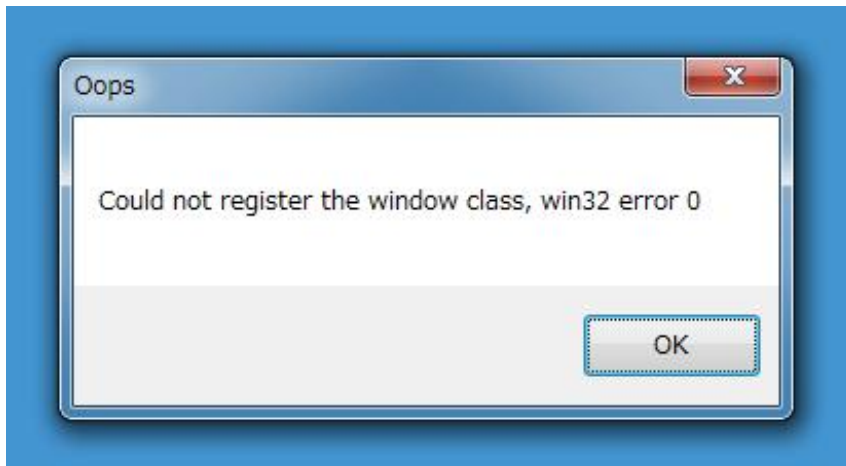
Near Distance : Minimum distance between camera and target. Only for perspective camera.

Far Distance : Maximum distance between camera and target. Only for perspective camera.

That's it !!

Press play button and check your mascot.

(NOTE: If an error window pops up, simply press OK. See the Frequently Asked Questions section of this document for further information on why this occurs.)



For More Advanced Usage

Please see Assets/DesktopMascotMaker/Examples folder.

If you want to interact with your mascot, please see Demo01_uGUI.unity, Demo02_Event.unity and Demo03_Physics.unity.

Demo01_General.unity demonstrates how to move and interact with the mascot.

Demo02_Event.unity demonstrates how to use MascotMaker's events.

Demo03_Physics.unity demonstrates how to apply Unity's physics to the mascot.

If you want to use uGUI's Image/Text, see Demo04_uGUI.unity.

Demo04_uGUI.unity demonstrates how to use uGUI with Desktop Mascot Maker.

If you want to hide Unity's main window, see Demo05_HideMainWindow.unity.

Demo05_HideMainWindow.unity demonstrates how to hide/show Unity's main window.

If you want to display multiple mascots, see Demo06_MultipleMascots.unity.

Demo06_MultipleMascots.unity demonstrates how to display multiple mascots.

Chroma Key Compositing

When you use Transparent Shader, rendering problems sometimes happen.

In such a case, please use Chroma Key Compositing mode.

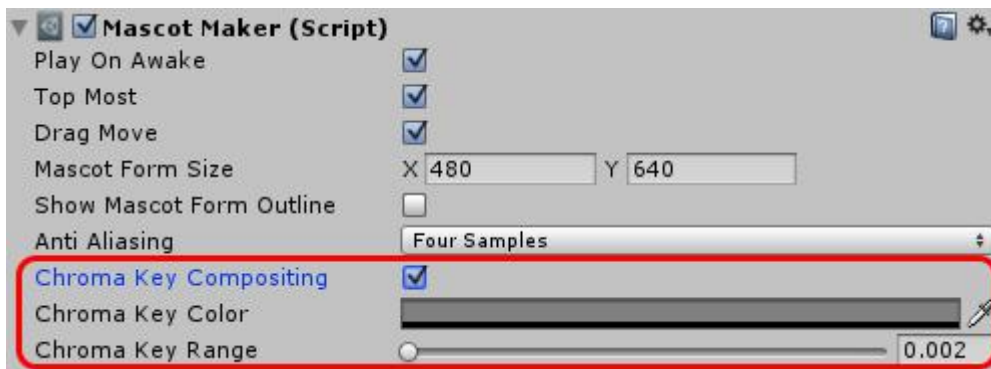
If you don't use Transparent Shader, I recommend to check off Chroma Key Compositing mode.



Chroma Key Compositing Off



Chroma Key Compositing On



Chroma Key Compositing Settings

Chroma Key Compositing (bool value)

If set to true, Chroma Key Compositing mode will be ON.

Chroma Key Color (Color value)

Chroma key background color. This color will be transparent.

Chroma Key Range (float value)

Chroma key compositing intensity. [0.002 - 0.5]

You can also change these settings via scripting

```
bool MascotMaker.Instance.ChromaKeyCompositing // bool value  
Color MascotMaker.Instance.ChromaKeyColor // Color  
float MascotMaker.Instance.ChromaKeyRange // float value [0.002-0.5]
```

Display of Multiple Mascots

Desktop Mascot Maker has two primary components, MascotMaker component and MascotMakerMulti component.

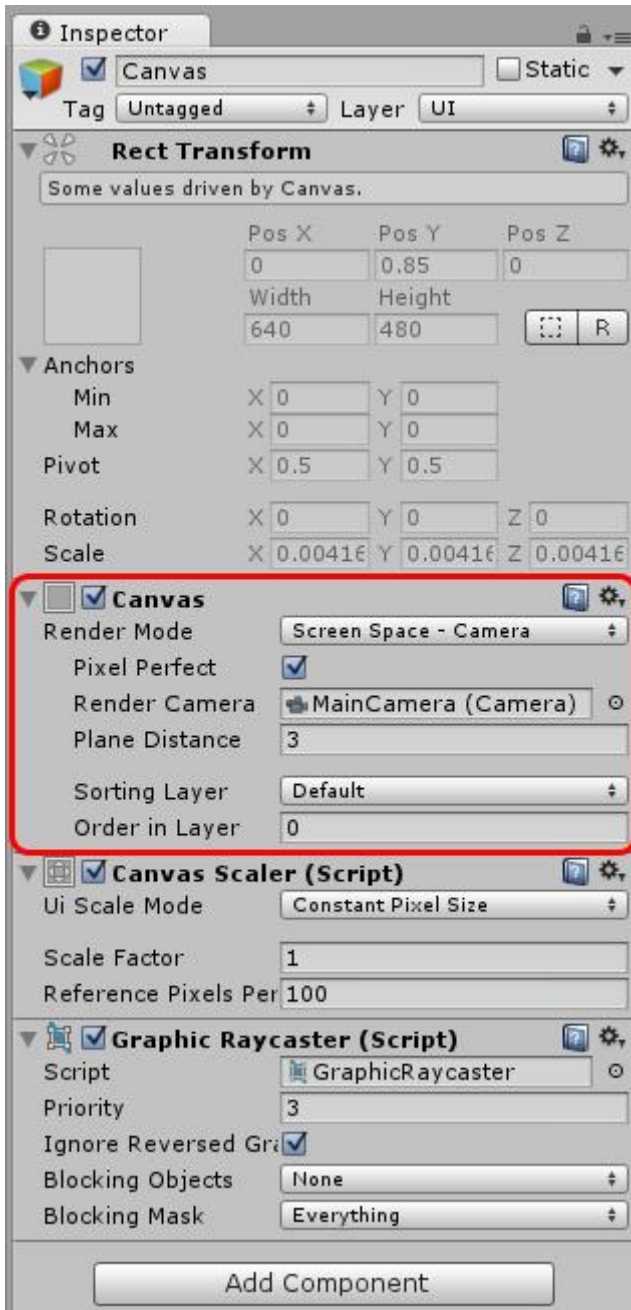
MascotMaker component is implemented as singleton design pattern, so this component is allowed to exist only one per one scene. On the other hand, MascotMakerMulti is implemented as normal MonoBehaviour component, so more than one MascotMakerMulti components can exist per one scene.

If you want to display multiple mascots per one scene, please use MascotMakerMulti components.

For the details, please see the section of MascotMakerMulti Class Reference in this document and check the demo (Demo06_MultipleMascots.unity).

Usage of uGUI

In order to use uGUI's Image/Text with DesktopMascotMaker, set Canvas's Render Mode "Screen Space - Camera" and set Render Camera like following:



Please see the demo (*Demo04_uGUI.unity*) for the details.

How to hide Unity's main window

Sometimes may be you want to show only mascot character and hide Unity's main window. In such a case, please attach MainWindowOpacity.cs script anywhere in your scene and set Main Window Opacity valude to 0.

You can add MainWindowOpacity component through

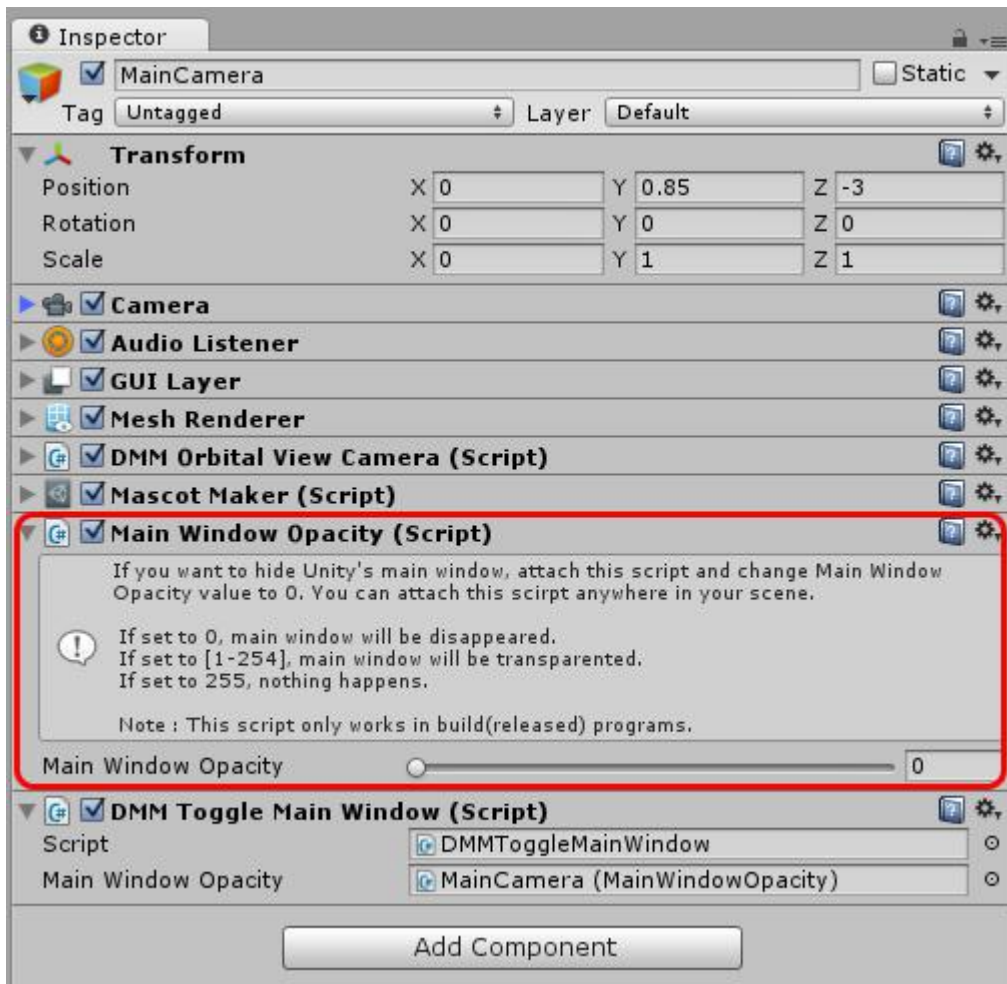
Menu > Component > DesktopMascotMaker > MainWindowOpacity

If set to 0, main window will be disappeared.

If set to [1-254], main window will be transparented.

If set to 255, nothing happens, just normal window.

NOTE : This script only works in build(released) programs. Please build your scene to check !



Please see the demo (*Demo05_HideMainWindow.unity*) for the details.

MascotMaker Class Reference

Access Through Singleton Instance

MascotMaker component is implemented by singleton design pattern.

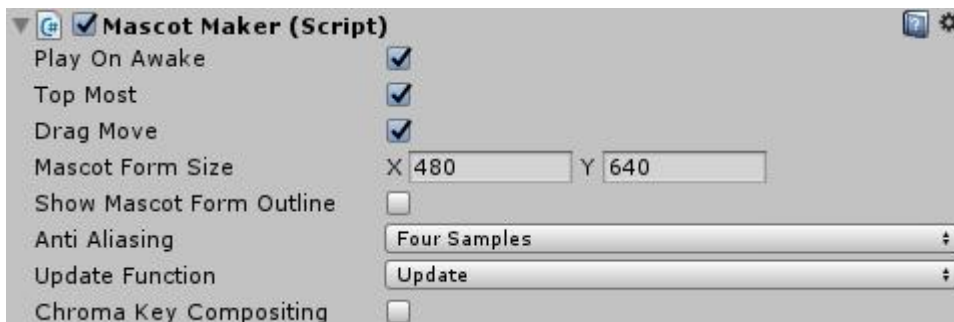
So you can access MascotMaker's variables through `MascotMaker.Instance` .

```
using DesktopMascotMaker;           // use this namespace
...
MascotMaker.Instance.xxx = ...      // like this
```

When you attach this component to your camera, the view of camera will be displayed through Mascot Form.

MascotMaker component can exist only one instance because MascotMaker component is implemented as singleton. If you want to display multiple mascots, please use MascotMakerMulti component. MascotMakerMulti component is implemented as normal (non-singleton) MonoBehaviour component. For the details, please see the section of MascotMakerMulti Class Reference in this document and check the demo (Demo06_MultipleMascots.unity).

Setting Variables



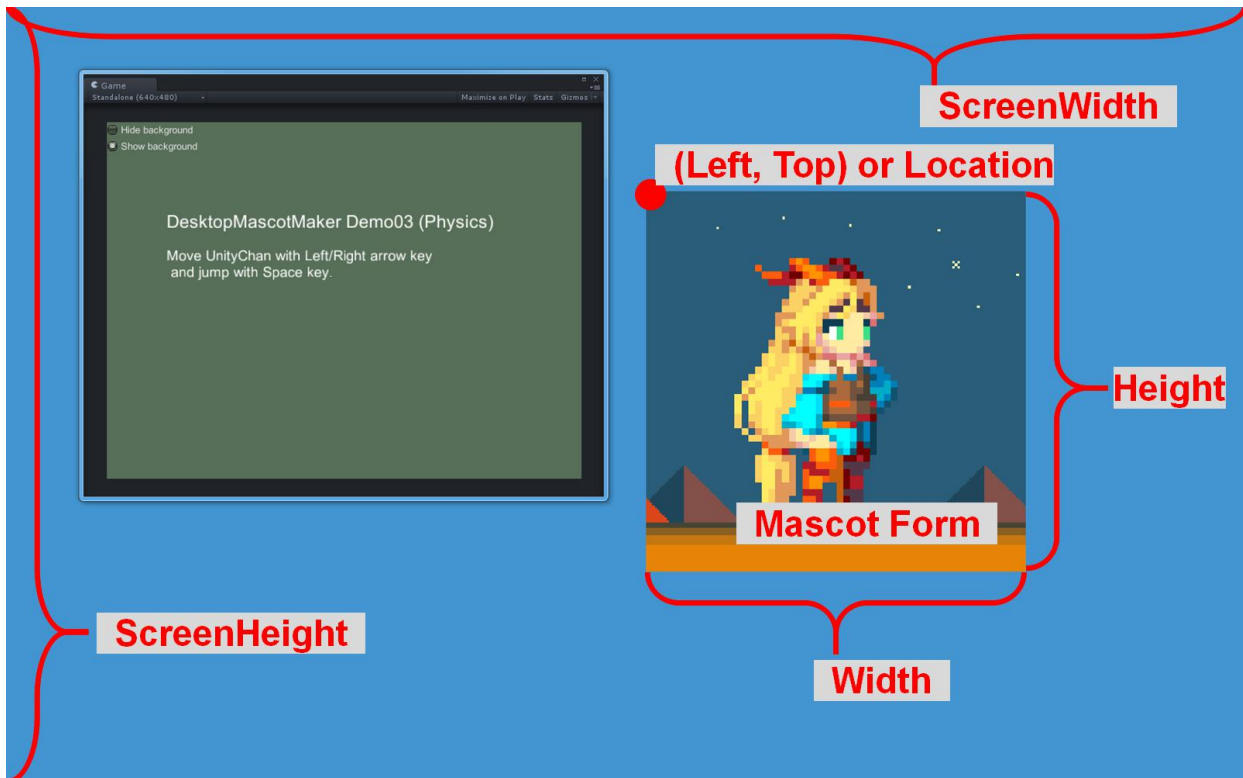
You can change MascotMaker component's setting variables via scripting.

```
bool MascotMaker.Instance.PlayOnAwake           // bool value
bool MascotMaker.Instance.TopMost                // bool value
bool MascotMaker.Instance.DragMove              // bool value
Vector2 MascotMaker.Instance.MascotFormSize     // Vector2 value
bool MascotMaker.Instance.ShowMascotFormOutline // bool value
MascotMaker.AntiAliasingType MascotMaker.Instance.AntiAliasing // enum value
MascotMaker.UpdateFuncType MascotMaker.Instance.UpdateFunc      // enum value
bool MascotMaker.Instance.ChromaKeyCompositing  // bool value
```


Mascot Form Variables

You can move Mascot Form with Left, Top (or Location) variables.

```
int MascotMaker.Instance.Left  
int MascotMaker.Instance.Top  
System.Drawing.Point MascotMaker.Instance.Location  
int MascotMaker.Instance.Width  
int MascotMaker.Instance.Height  
int MascotMaker.Instance.ScreenWidth // read only  
int MascotMaker.Instance.ScreenHeight // read only
```



For example, if you set

```
MascotMaker.Instance.Left = 0;  
MascotMaker.Instance.Top = 0;
```

then the Mascot Form's top left corner will be aligned with the top left corner of the screen.

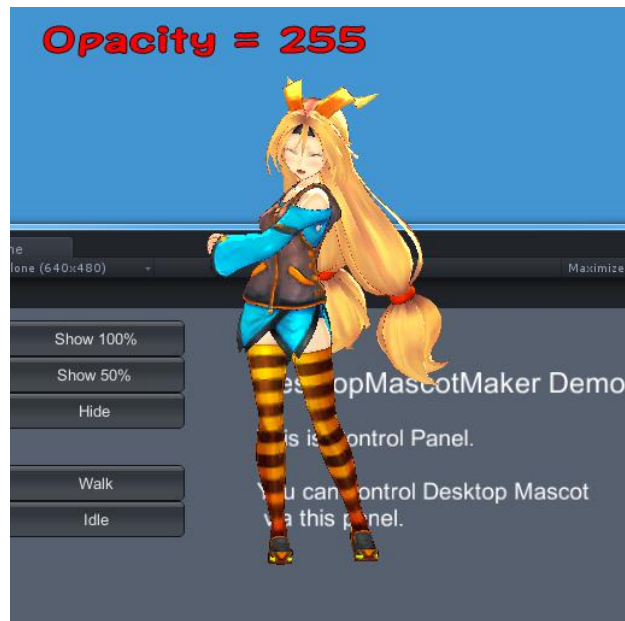
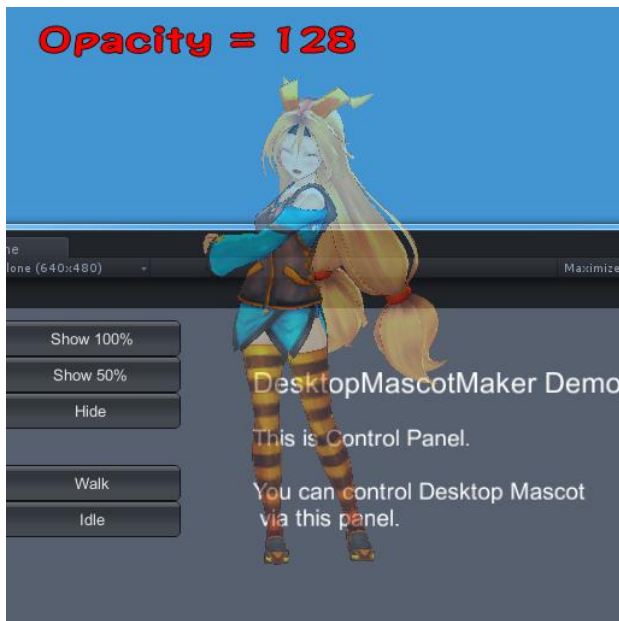
If you want to move the mascot at runtime, change `MascotMaker.Instance.Left/Top` value in Unity's `Update()` function. Like this:

```
void Update()
{
    MascotMaker.Instance.Left += Time.deltaTime * speed;
}
```

Mascot Opacity

You can change mascot's opacity as follows.

```
MascotMaker.Instance.Opacity = 128; // int value. [0-255]  
MascotMaker.Instance.Hide(); // Hide mascot completely  
MascotMaker.Instance.Show(); // Show again
```



If you set opacity [1-254], the mascot will be translucent.

If you want to hide the mascot, please use Hide() function.

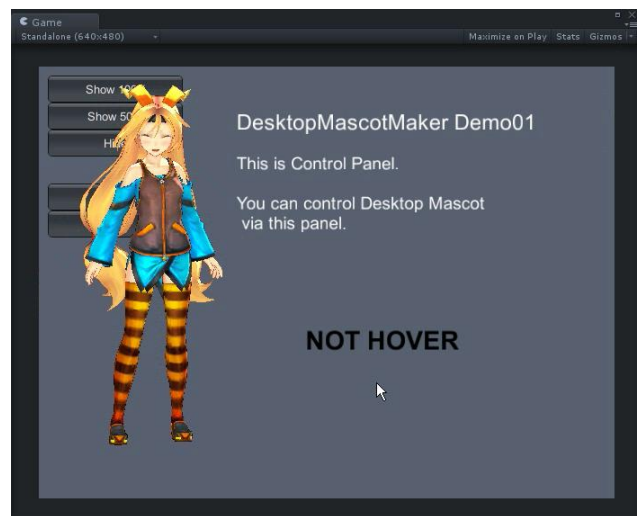
MascotMaker.Instance.Hide() hides the mascot completely and keeps CPU load low. On the other hand, if you hide mascot by using *MascotMaker.Instance.Opacity = 0*, CPU load is relatively high because the MascotMaker still tries to display mascot internally.

Is Mouse Hovering?

Check whether the mouse cursor is hovering over the mascot.

If the mouse cursor is on the mascot, `MascotMaker.Instance.IsMouseHover` will be true, otherwise it will be false.

```
bool mouseHover = MascotMaker.Instance.IsMouseHover; // read only
```



Please see the demo (Demo01_General.unity) for the details.

Event Settings

You can use MouseEvents, Activated/DeactivateEvents, KeyEvents, DragDropEvents, MoveEvents. Add your custom event function to MascotMaker's event handler like this:

```
// Add function to MascotMaker's EventHandler
MascotMaker.Instance.OnLeftMouseDown += MouseDown;
...
void MouseDown(object sender, MouseEventArgs e)
{
    // Some code here
    Debug.Log("Clicked!");
}
```

This is MascotMaker's event list.

Event	Event Handler Type	
OnLeftMouseDown	MouseEventHandler	Occurs when the Left mouse button is pressed.
OnLeftMouseUp	MouseEventHandler	Occurs when the Left mouse button is released.
OnLeftDoubleClick	MouseEventHandler	Occurs when the Left mouse is double clicked.
OnRightMouseDown	MouseEventHandler	Occurs when the right mouse button is pressed.
OnRightMouseUp	MouseEventHandler	Occurs when the right mouse button is released.
OnRightDoubleClick	MouseEventHandler	Occurs when the right mouse button is double clicked.
OnMiddleMouseDown	MouseEventHandler	Occurs when the middle mouse button is pressed.
OnMiddleMouseUp	MouseEventHandler	Occurs when the middle mouse button is released.
OnMiddleDoubleClick	MouseEventHandler	Occurs when the middle mouse button is double clicked.
OnMouseWheel	MouseEventHandler	Occurs when the mouse is wheeled.
OnActivated	EventHandler	Occurs when the Mascot Form is activated.
OnDeactivate	EventHandler	Occurs when the Mascot Form is deactivated.
OnKeyDown	KeyEventHandler	Occurs when a key is pressed while Mascot Form has focus.
OnKeyUp	KeyEventHandler	Occurs when a key is released while Mascot Form has focus.
OnDragDrop	DragEventHandler	Occurs when a drag-and-drop operation is completed.
OnDragEnter	DragEventHandler	Occurs when an object is dragged into Mascot Form's bounds.
OnDragLeave	EventHandler	Occurs when an object is dragged out of Mascot Form's bounds.
OnDragOver	DragEventHandler	Occurs when an object is dragged over Mascot Form's bounds.
OnMove	EventHandler	Occurs when the Mascot Form is moved.

Please see the demo(*Demo02_Event.unity*) for the details.

MascotMakerMulti Class Reference

If you want to display multiple mascots, please use MascotMakerMulti components. When you attach this component to your camera, the camera will display the camera view as a Mascot Form. You can add this component through the menu **Menu > Component > DesktopMascotMaker > MascotMakerMulti**.

MascotMaker component is implemented as Singleton design pattern and allows only one instance per scene.

On the other hand, MascotMakerMulti component implements as normal MonoBehaviour component so you can use multiple components per one scene.

The difference between MascotMaker and MascotMakerMulti component is the access method to their variables. When you use MascotMaker component, you access the variables like this :

```
MascotMaker.Instance.Left = 100;
```

On the other hand, when you use MascotMakerMulti component, you have to get its instance with some methods (Inspector assignment, GetComponent function, etc) like this :

```
// declare a variable  
public MascotMakerMulti mascotMakerMulti;  
  
void Start ()  
{  
    // get instance by GetComponent method  
    mascotMakerMulti = GetComponent<MascotMakerMulti>();  
    // access MascotMakerMulti's variables like this  
    mascotMakerMulti.Left = 100;  
}
```

All MascotMakerMulti's variables are functional equivalent to MascotMaker's variables.

For example,

The *mascotMakerMulti.Left* is functional equivalent property to *MascotMaker.Instance.Left*.

The *mascotMakerMulti.Show()* is functional equivalent to *MascotMaker.Instance.Show()*.

The *mascotMakerMulti.OnLeftMouseDown* is functional equivalent event handler to

MascotMaker.Instance.OnLeftMouseDown.

That is to say, if you want to use MascotMakerMulti instead of MascotMaker, replace *MascotMaker.Instance* to *mascotMakerMulti* (*mascotMakerMulti* is *MascotMakerMulti*'s instance).

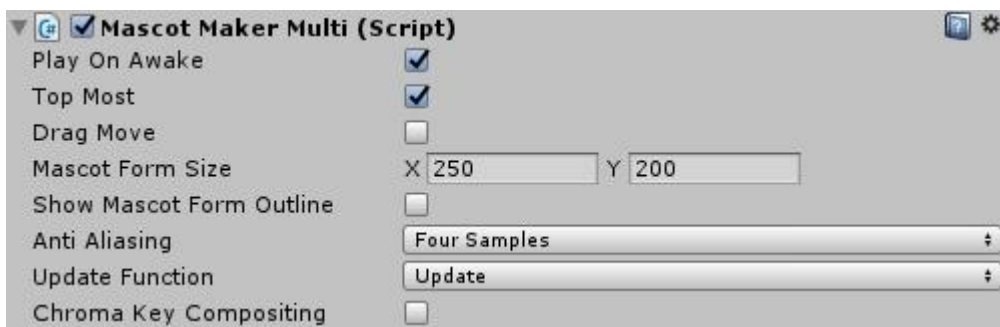
For the details, please see the demo(Demo06_MultipleMascots.unity), especially please see the script

DMMUnityChan2DAIForMulti.cs to learn how to use MascotMakerMulti component.

NOTE : You can use one MascotMaker component and multiple MascotMakerMulti components same time in the same scene.

MascotMakerMulti Settings

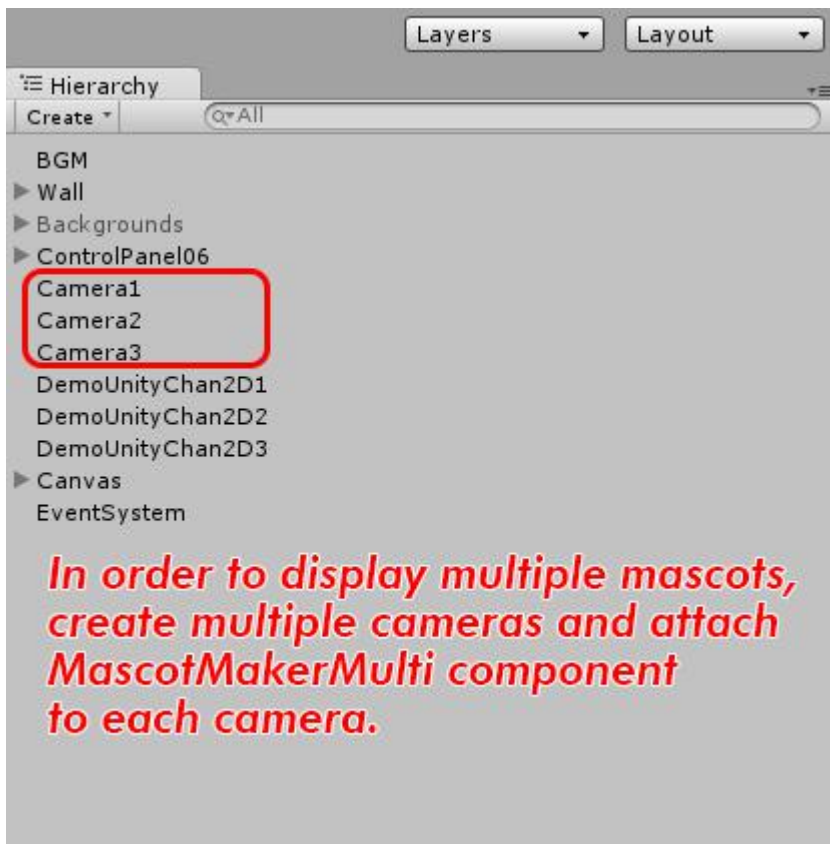
MascotMakerMulti component's settings are same as MascotMaker component's. Please see Quick Start for details.



MascotMakerMulti's setting variables can be accessed via scripting.

```
// mascotMakerMulti is instance of MascotMakerMulti component  
bool mascotMakerMulti.PlayOnAwake // bool value  
bool mascotMakerMulti.TopMost // bool value  
bool mascotMakerMulti.DragMove // bool value  
Vector2 mascotMakerMulti.MascotFormSize // Vector2 value  
bool mascotMakerMulti.ShowMascotFormOutline // bool value  
MascotMakerMulti.AntiAliasingType mascotMakerMulti.AntiAliasing // enum value  
MascotMakerMulti.UpdateFuncType mascotMakerMulti.UpdateFunc // enum value  
bool mascotMakerMulti.ChromaKeyCompositing // bool value
```

MascotMakerMulti component can be attached to multiple cameras per one scene. For example, if you want to display three independent mascots, create three cameras in the scene and attach each of three MascotMakerMulti components to each camera.

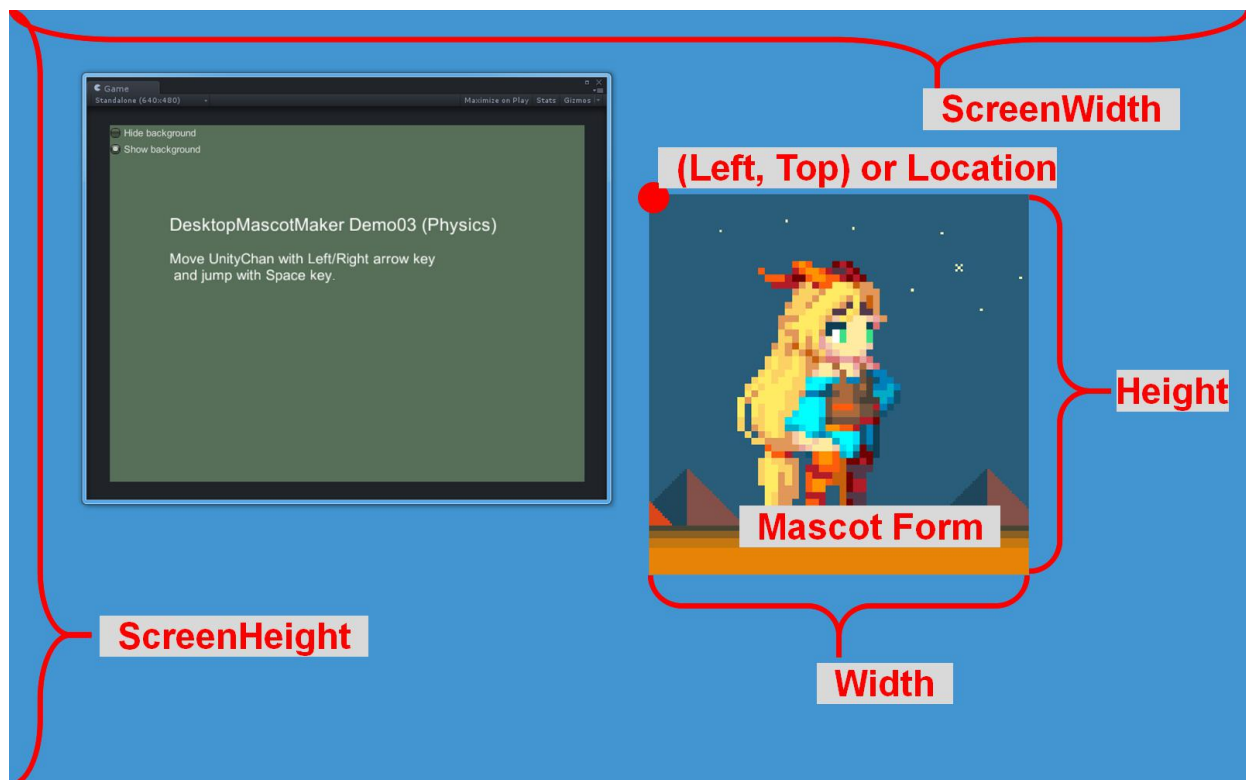


Please See the demo (Demo06_MultipleMascots.unity) for details.

Mascot Form Variables

You can move Mascot Form with Left, Top (or Location) variables.

```
// mascotMakerMulti is instance of MascotMakerMulti component  
int mascotMakerMulti.Left  
int mascotMakerMulti.Top  
System.Drawing.Point mascotMakerMulti.Location  
int mascotMakerMulti.Width  
int mascotMakerMulti.Height  
int mascotMakerMulti.ScreenWidth // read only  
int mascotMakerMulti.ScreenHeight // read only
```



For example, you set

```
// mascotMakerMulti is instance of MascotMakerMulti component  
mascotMakerMulti.Left = 0;  
mascotMaker.Instaince.Top = 0;
```

then the Mascot Form's top left conrner will be aligned with the top left corner of screen.

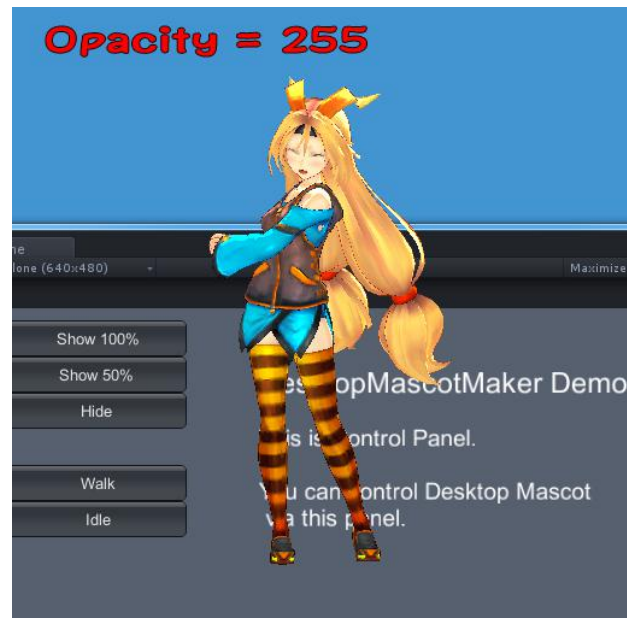
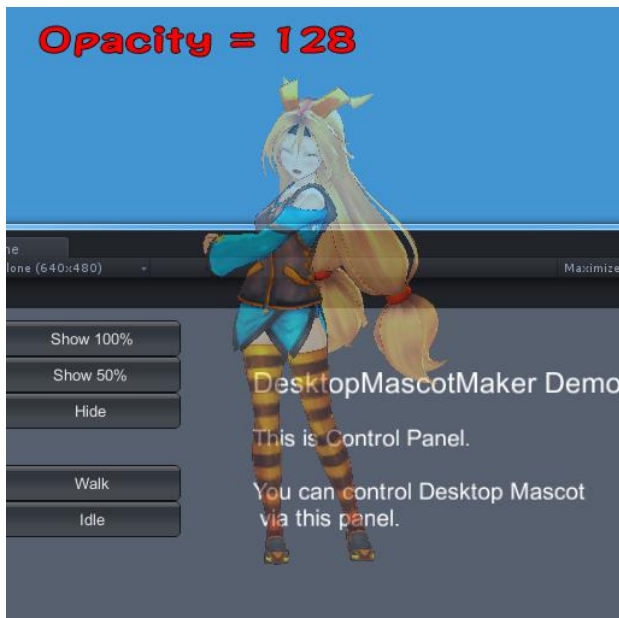
In order to move the mascot at runtime, change `mascotMakerMulti.Left/Top` value in Unity's `Update()` function. Like this:

```
void Update()
{
    // mascotMakerMulti is instance of MascotMakerMulti component
    mascotMakerMulti.Left += Time.deltaTime * speed;
}
```

Mascot Opacity

Change mascot's opacity as follows.

```
// mascotMakerMulti is instance of MascotMakerMulti component
mascotMakerMulti.Opacity = 128; // [0-255]
mascotMakerMulti.Hide(); // Hide mascot completely
mascotMakerMulti.Show(); // Show again
```



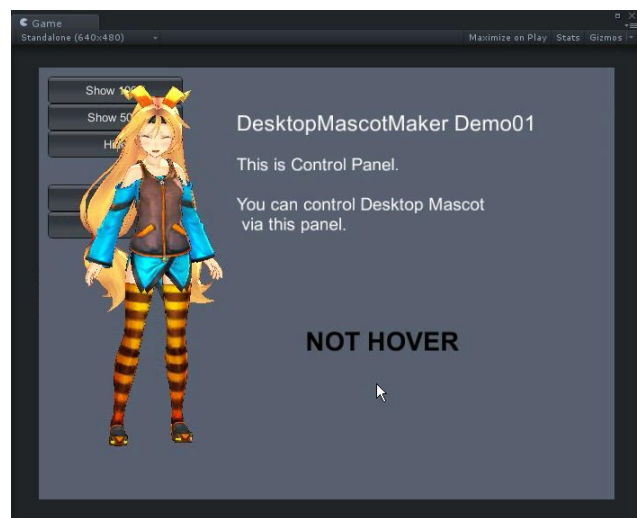
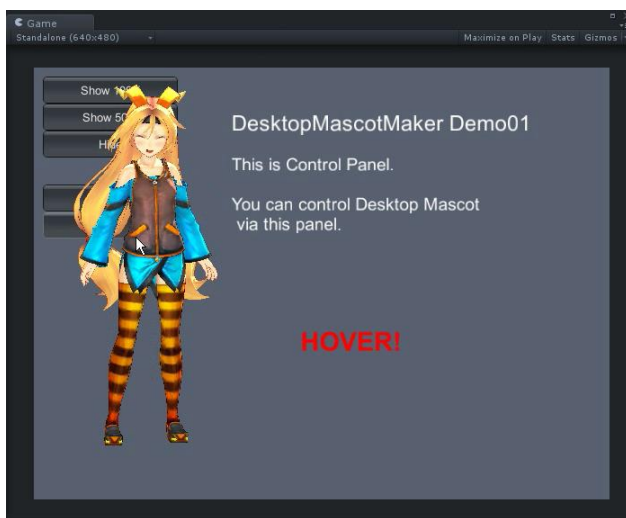
If you set the Opacity [1-254], the mascot will be translucent. In order to hide the mascot completely, please use `mascotMakerMulti.Hide()` function. In `mascotMakerMulti.Hide()` function, no display processing occurs internally and keeps CPU load relatively low. On the other hand, if you hide mascot by using `mascotMakerMulti.Opacity = 0`, CPU load is relatively high because the `MascotMakerMulti` still tries to display mascot internally.

Is Mouse Hovering?

Check whether the mouse cursor is hovering over the mascot.

If the mouse cursor is on the mascot, `mascotMakerMulti.IsMouseHover` will be true, otherwise it will be false.

```
// mascotMakerMulti is instance of MascotMakerMulti component
bool mouseHover = mascotMakerMulti.IsMouseHover; // read only
```



Event Settings

You can use `MouseEvents`, `Activated/DeactivateEvents`, `KeyEvents`, `DragDropEvents`, `MoveEvents`.

Add your custom event function to `MacotMakerMulti`'s event handler like this:

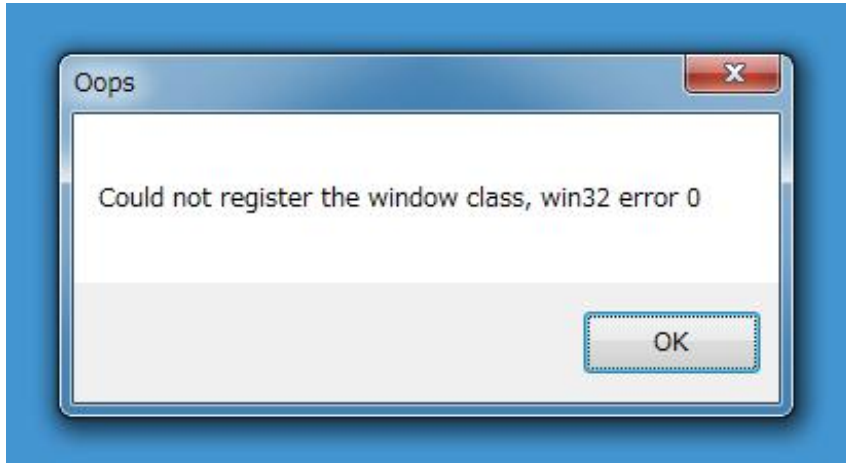
```
// Add function to MascotMakerMulti's EventHandler
// mascotMakerMulti is instance of MascotMakerMulti component
mascotMakerMulti.OnLeftMouseDown += MouseDown;
...
void MouseDown(object sender, MouseEventArgs e)
{
    // Some code here
    Debug.Log("Clicked!");
}
```

This is MascotMakerMulti's event list. (same as MascotMaker)

Event	Event Handler Type	
OnLeftMouseDown	MouseEventHandler	Occurs when the Left mouse button is pressed.
OnLeftMouseUp	MouseEventHandler	Occurs when the Left mouse button is released.
OnLeftDoubleClick	MouseEventHandler	Occurs when the Left mouse is double clicked.
OnRightMouseDown	MouseEventHandler	Occurs when the right mouse button is pressed.
OnRightMouseUp	MouseEventHandler	Occurs when the right mouse button is released.
OnRightDoubleClick	MouseEventHandler	Occurs when the right mouse button is double clicked.
OnMiddleMouseDown	MouseEventHandler	Occurs when the middle mouse button is pressed.
OnMiddleMouseUp	MouseEventHandler	Occurs when the middle mouse button is released.
OnMiddleDoubleClick	MouseEventHandler	Occurs when the middle mouse button is double clicked.
OnMouseWheel	MouseEventHandler	Occurs when the mouse is wheeled.
OnActivated	EventHandler	Occurs when the Mascot Form is activated.
OnDeactivate	EventHandler	Occurs when the Mascot Form is deactivated.
OnKeyDown	KeyEventHandler	Occurs when a key is pressed while Mascot Form has focus.
OnKeyUp	KeyEventHandler	Occurs when a key is released while Mascot Form has focus.
OnDragDrop	DragEventHandler	Occurs when a drag-and-drop operation is completed.
OnDragEnter	DragEventHandler	Occurs when an object is dragged into Mascot Form's bounds.
OnDragLeave	EventHandler	Occurs when an object is dragged out of Mascot Form's bounds.
OnDragOver	DragEventHandler	Occurs when an object is dragged over Mascot Form's bounds.
OnMove	EventHandler	Occurs when the Mascot Form is moved.

Frequently Asked Questions

Q : What's this MessageBox?



A : Sorry its mono's internal message. But this does not affect your application, ignore it.

This error message occurs in mono's dll inside unity.

(...¥Unity¥Editor¥Data¥Mono¥lib¥mono¥2.0¥System.Windows.Forms.dll)

Please check this code line 894.

<https://github.com/mono/mono/blob/mono-2-10/mcs/class/Managed.Windows.Forms/System.Windows.Forms/XplatUIWin32.cs>

This message occurs because the function Win32RegisterClass returns false in unity editor environment.

I can't modify this mono code inside unity because it may affect all unity projects.

This error message occurs only in unity editor environment and **it does not affect your final compiled (released) .exe programs**. So could you ignore this message? Sorry for the inconvenience.

Q : How can I do performance tuning?

A : Change Mascot Form's size as small as possible.

First of all, the performance of Desktop Mascot Maker depends on Mascot Form's size. So please make Mascot Form's size as small as possible. Second, if you hide the mascot, please use Hide() function instead of Opacity = 0. This is also efficient for good performance tuning.

Q : I played demo but I couldn't find Unity-Chan, where is Unity-Chan!?

A : Please restart project.