# Modern Dark/Blue GUI Skin for Unity 3D

## General description

The Modern Dark/Blue UI is a GUI skin for Unity 3D 4.6 or higher. It uses the new Unity GUI (uGUI).

Provided are C# classes for creating a demo scene, and all necessary prefabs and sprites. Included is a tooltip system, that displays a tooltip text when the user hovers over an ui element.

### The Tooltip System - How It Works

The tooltip works basically the same as in some other software, i.e. Firefox.

If the user moves the mouse over an ui element and keeps it still for 0.5 seconds, the tooltip text is faded in. Once the users moves the mouse again, the tooltip text is quickly faded out.

The tooltip is always placed below the ui element.

### Tooltip System - Implementation

The tooltip system is implemented in the class 'TooltipManager.cs'. The prefab 'TooltipCanvas' is provided for the tooltip canvas, background image and text. The class 'UITooltipCanvas.cs' encapsulates the construction of this tooltip. It is instantiated once in the scene and puts a reference of itself in a static class variable of 'UIElement.cs', the base class for all ui elements.
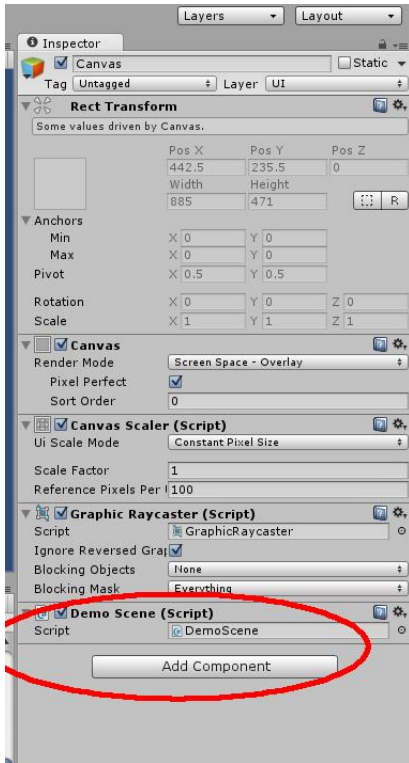
'UIElement.cs', has a public method 'AddTooltip(string text)', which allows to set the tooltip text individually for any ui element.

### Tooltip System - Demo Scene

The demo scene consists only of a Canvas, a Main Camera and the EventSystem:
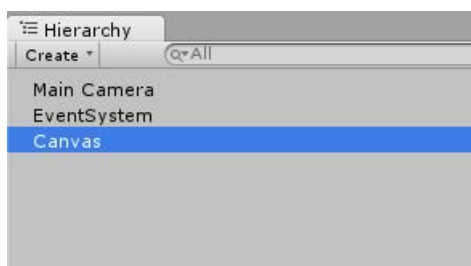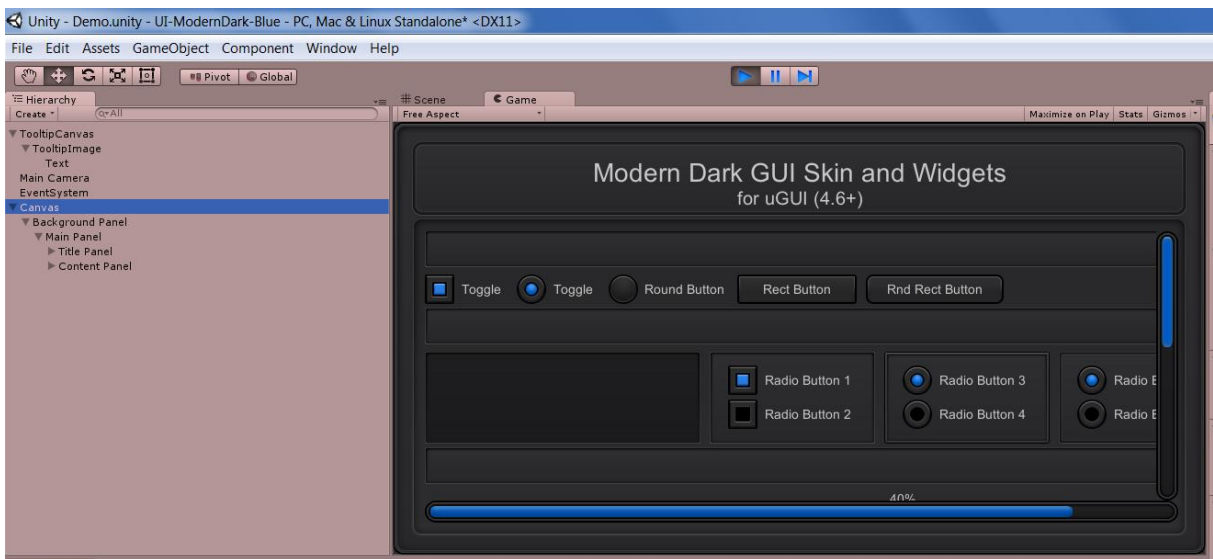


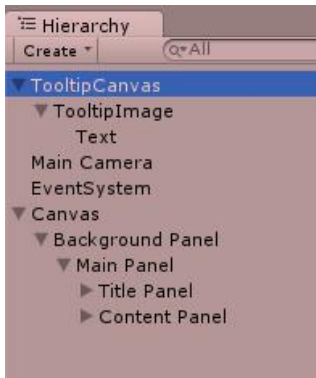The Canvas has the 'DemoScene.cs' script attached, which is executed automatically when the scene is run:

The GUI is completely created from the 'DemoScene.cs' script.

This is how the scene looks when it is run:

A TooltipCanvas has been created on the same level as the GUI Canvas:



The TooltipCanvas holds a single image, the background image for all tooltips, and a text element for the tooltip text. There is only one background image and text, because there is only one tooltip displayed at any one time.

The Sort Order of the TooltipCanvas is set to a high value, in order to show it always above all ui elements in the GUI Canvas:
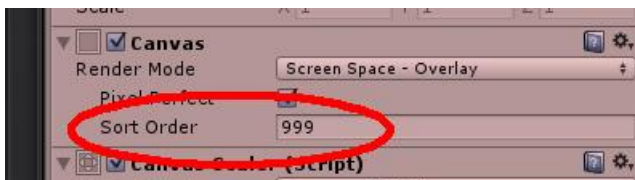


The Render Mode of the TooltipCanvas is normally set to 'Screen Space - Overlay', but you can set to any of the other available modes as well:
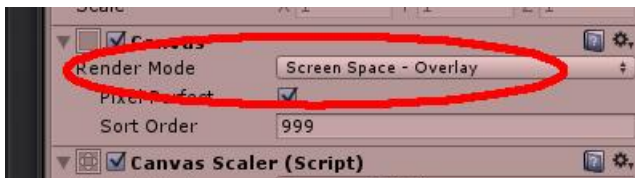


If you use Screens Space - Camera or World Space you must assign the same Camera as in your GUI Canvas, and tag the Camera as 'MainCamera'!

The tooltip background image has a CanvasGroup attached, and the Alpha value must initially be set to 0, such that the tooltip is normally hidden:
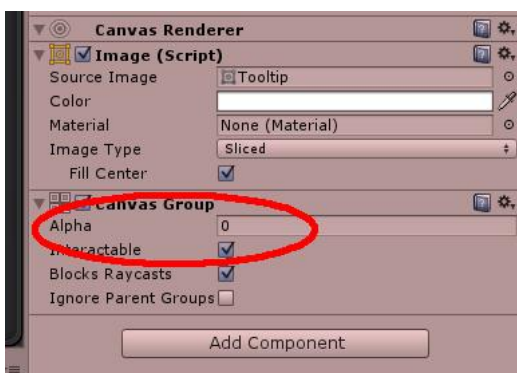
The 'TooltipManager.cs' script uses this alpha value to fade the tooltip in and out.

Individual ui elements, such as as button, have the 'TooltipManager' script attached:
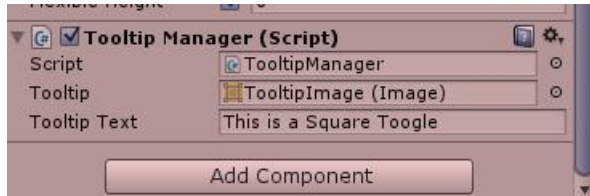


In the 'Tooltip' field, the tooltip background image (the one under the Tooltip Canvas) is attached, in 'Tooltip Text' the text to be displayed is entered. If you use the class framework provided with the demo scene, all this is automatically done for you by a call to 'UIElement.AddTooltip(text)'.

If you try the tooltip system in the Unity editor, make sure that the Game Window is active (by clicking once somewhere in the Game Window), otherwise the tooltip system does not receive mouse clicks.

## Tooltip System - Customization

In 'TooltipManager.cs' you can customize some parameters (under the comment 'Parameters, can be changed'). You can change the fade in and fade out times for the tooltip, the trigger delay (the time the user needs to hold the mouse still, before the tooltip is displayed), the tooltip offset (distance from the bottom of the ui element where the tooltip is shown) and the text inset (inset of text from the borders of the tooltip image).

## Illustrator Script 'RadialScale.jsx'

If you want to change the scale on the Radial Knob, you can use the Illustrator script 'RadialScale.jsx' which is provided in the 'Illustrator' folder.

To use the script, create an empty document in Illustrator, **set the fill color to black** if it isn't already and start the script (from File -> Scripts -> Other Script and navigate to the 'Illustrator' folder under this Unity package).

The scripts displays an options panel and creates a default radial scale in the document:
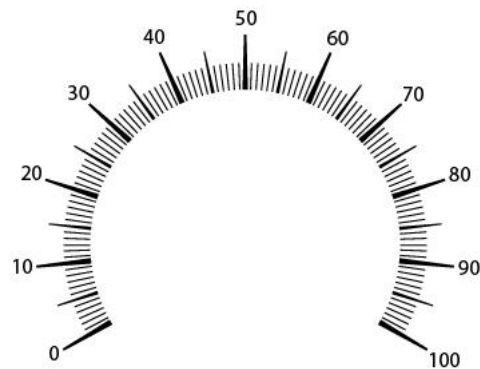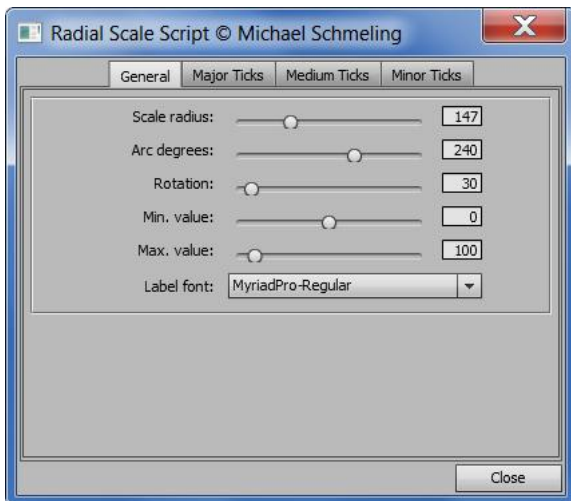
(If you don't see the radial scale it may be because your fill color is set to white on a white background or because the options panel is hiding it!)

Use the four tabs on the options panel to customize the appearance of the radial scale.
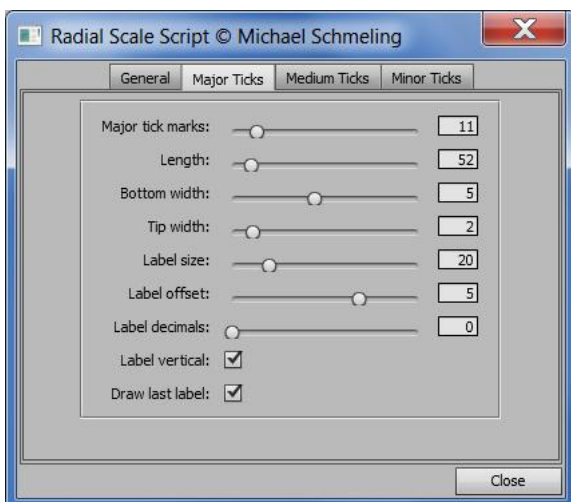


When you close the options panel, all settings are stored in a Tag of the 'Radial Scale' group in the document, and when you reopen the options panel, you'll see the same options as before. (The settings are gone if you delete the 'Radial Scale' group.)

Here are a few of the different radials scale you can create:
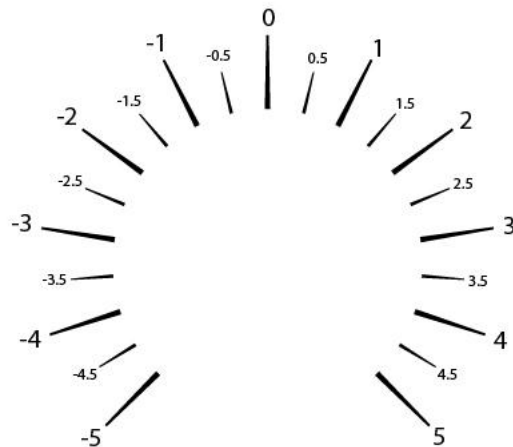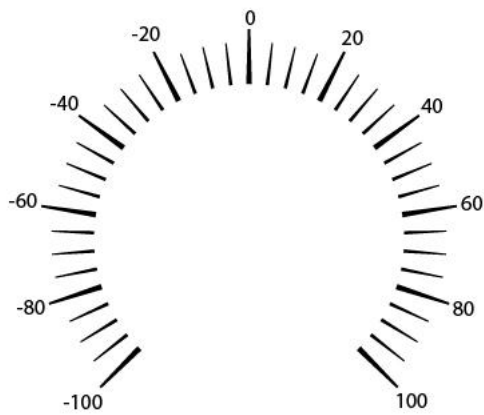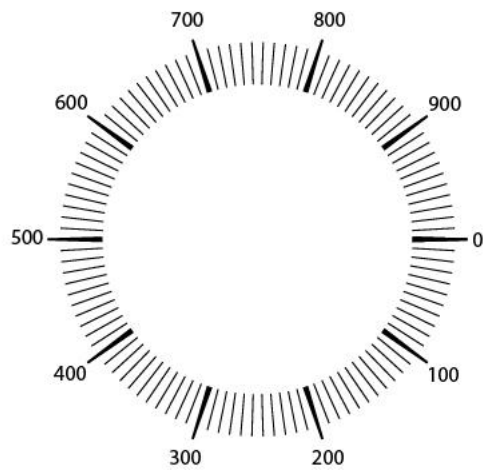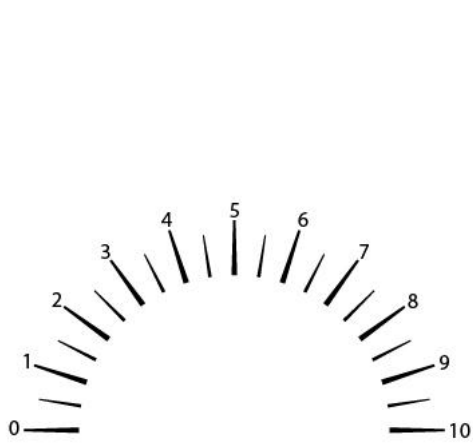
## Sprites

All sprites have the packing tag 'ModernDarkGUISkin' set.

## Sprite Sources

All sprite sources are available in the 'Sources' folder. They come as Photoshop CS4 .psd files (and one Illustrator .ai file for the Radial Scale). The .psd files are layered and use vector shapes and dynamic gradients, so they can be easily resized without loss of detail.

## Mobile Devices

This GUI package has not been tested on mobile devices, only on the PC/Desktop.

## Contact

Don't hesitate to contact me if you find any problems with the product or if you have any questions or suggestions for improvements.

Email: sales@aridocean.com