Readme - Advanced Color Picker 1.3.3

Advanced Color Picker is a powerful, easy-to-use color pickers bundle made for Unity. Works for all platforms and supports picking Colors in different ways.

You can use the color pickers by calling 2 - 3 APIs. Or create and design your own picker by starting with the included prefabs and component scripts. Use it out of the box, or making a new one? You can always have choices!

We present you with the professional-looking, accurately calculated UI, yet highly customizable color picker! No matter what Unity player platform, mobile or desktop applications. This is the color picker package you need!

Index

(1) Features

- Pick Color from the palette, easy to add/replace palette textures (Editor or Runtime)
- Adjust the RGBA sliders to pick Color
- RGBA input-fields, set Color by entering the RGBA values
- Hex code input-field, convert hex code to Color
- Picker Rect Component: an advanced color dropper for picking colors from the screen, adjustable picker rect size for picking an area of the pixels, and calculate the nearest color of that area looks like, optional to save the picked area as an image (JPG/PNG)
- Preset Colors, auto/manual manage, save & load the preset colors. Easy, flexible, configurable layout
- Support multiple color pickers running at the same time, making your color picker more extensible, and flexible for more complicated use cases
- Ready to use and easy to customize UI design (5 ready to use color picker templates included, you can create more if you want)
- Show the picked Color, RGB/RGBA values, show Hex color code, show the picked area texture
- Highly customizable settings and concise API

Highly Customizable

Enable/Disable dragging for the color picker
Enable/Disable alpha setting
Enable/Disable color dropper
Adjustable color dropper size
Flexibly change the palette texture (runtime & editing time)
Auto-fit with screen, also has extra settings for handling panel size
Scalable panel (by setting a scale factor)
Options for auto manage(save/load) preset colors to the PlayerPrefs, or by manual operations

Support Platform

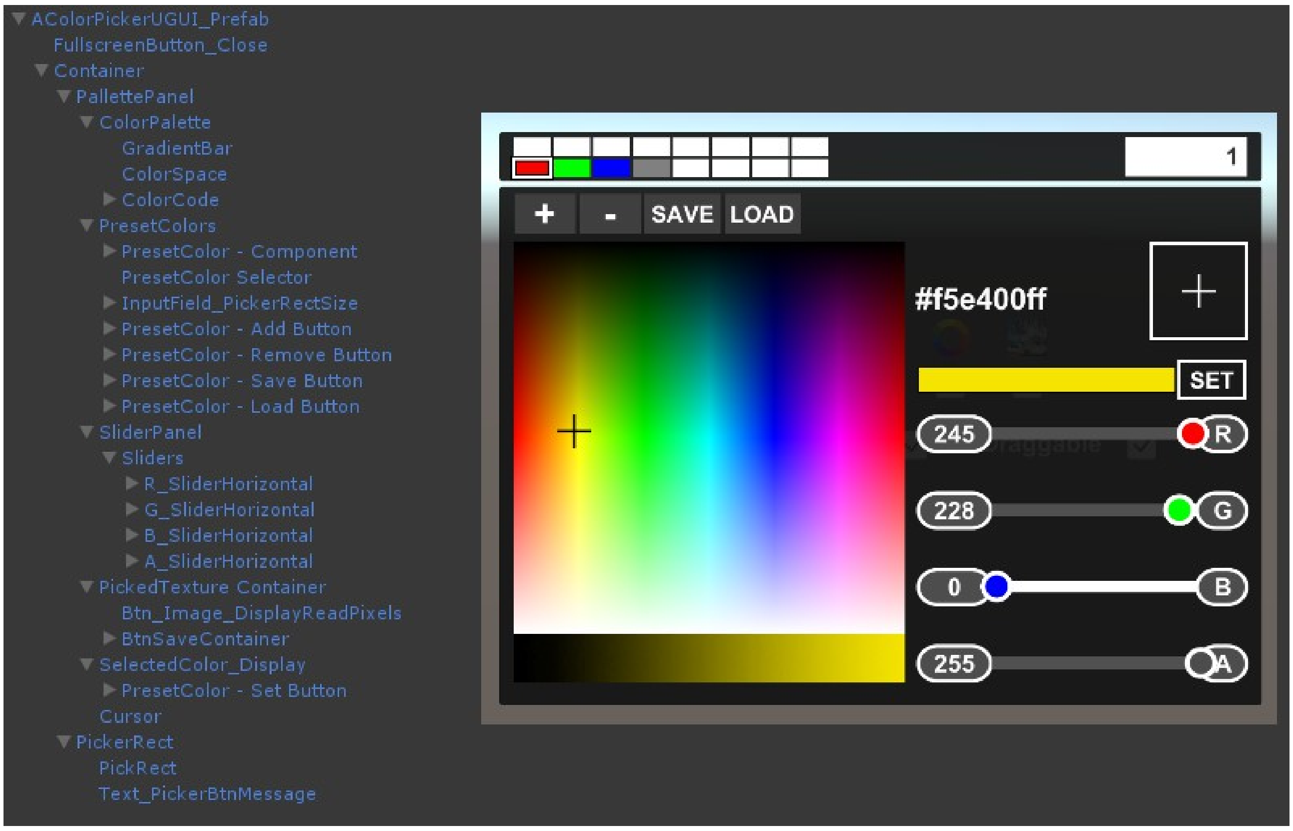
Both Legacy InputManager & New InputSystem
New Unity UI (compatible with other UI systems)
Unity 5 and newer
All platforms, Mobile and Desktop

(2) Fast Guide

Demo scenes and carefully designed color picker prefabs are provided in the package.
Try them out and you will know how they work and know how to use the provided scripts to do more you want. The provided scripts and UI are carefully designed to make this package easy to work with, flexible to customize and highly reusable!

Just give some different settings to the provided prefabs to make some more variations. You can change their settings at runtime, or change the settings on the provided prefabs in the Unity inspector.

(3) Prefab Structure



```
▼ AColorPickerUGUI_Prefab
    FullscreenButton_Close
  ▼ Container
    ▼ PallettePanel
      ▼ ColorPalette
          GradientBar
          ColorSpace
        ▶ ColorCode
      ▼ PresetColors
        ▶ PresetColor - Component
          PresetColor Selector
        ▶ InputField_PickerRectSize
        ▶ PresetColor - Add Button
        ▶ PresetColor - Remove Button
        ▶ PresetColor - Save Button
        ▶ PresetColor - Load Button
      ▼ SliderPanel
        ▼ Sliders
          ▶ R_SliderHorizontal
          ▶ G_SliderHorizontal
          ▶ B_SliderHorizontal
          ▶ A_SliderHorizontal
      ▼ PickedTexture Container
          Btn_Image_DisplayReadPixels
        ▶ BtnSaveContainer
      ▼ SelectedColor_Display
        ▶ PresetColor - Set Button
          Cursor
    ▼ PickerRect
        PickRect
        Text_PickerBtnMessage
```

* More color picker prefab templates are provided in the asset, the structure looks similar in the Hierarchy window, but the UI looks so different(please refer to the screenshots on our website or AssetStore page)

(4) Coding Examples

The **AColorPicker.cs** script holds most of the color picker functions, variables and callbacks. Below are some useful code snippets for easy integrating the color picker in your app. You can also find these examples in the demoscene scripts.

To instantiate a color picker:
AColorPicker picker = AColorPicker.Create(string: prefabName);
or
AColorPicker picker = AColorPicker.Create(Transform: parentTransform, string: prefabName);

Call the Setup method for some basic setup:
picker.Setup(GradientBarUsage: gradientBarUsage, Action: onCloseAction);
or
picker.Setup(GradientBarUsage: gradientBarUsage, bool: hasPickerRect,
Action: onCloseAction);
or
picker.Setup(GradientBarUsage: gradientBarUsage, int: pickerRectSize,
Action: onCloseAction);
or
picker.Setup(Action: onCloseAction);

Set the callback to receive picked color:
picker.SetOnColorChangeCallback(MethodName_Or_Action);

Set the callback to receive picked texture:
picker.SetOnPickedSampleTextureCallback(MethodName_Or_Action);

Toggle draggable feature:
picker.ToggleDraggable(bool: isDraggable);

Toggle picker rect(dropper) feature:
picker.TogglePickerRect(bool: enableDropper);

Set the panel position:
picker.SetPanelPosition(float: X, float: Y);

Set the picker rect(dropper) position:
picker.SetPickerRectPosition(float: X, float: Y);

Set the color picker scale:
picker.SetContainerScale(float: scaleFactor);

Set the color picker scale by canvas size:
picker.SetScaleByCanvasSize(int: canvasSize);

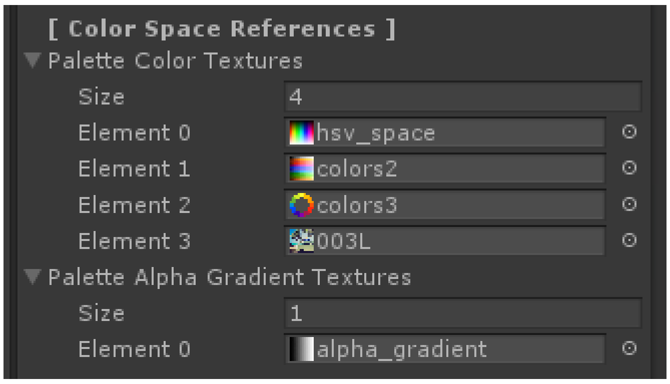
Set the picker rect(dropper) size:
picker.SetPickerRectSize(int: size);

Set the panel width with a constant width value:
Picker.SetPanelWidth(int: constantWidth);

Set the panel width with a fill-rate with respect to the screen width:
Picker.SetPanelWidth(float: fillRate);

Enable or Disable the Gradient bar and Alpha, set to use the gradient bar as Alpha Gradient or Color Gradient:
picker.SetGradientBar(GradientBarUsage: gradientBarUsage);
GradientBarUsage enum options:
Disable: Hide gradient bar, hide alpha settings
AlphaGradient: Show gradient bar, use it as alpha gradient, show alpha settings
ColorGradient: Show gradient bar, use it as color gradient, show alpha settings
ColorGradient_NoAlpha: Show gradient bar, use it as color gradient, hide alpha settings
Disable_HasAlpha: Hide gradient bar, show alpha settings

You can also change the palette textures at runtime.
Below shows the code and prepared texture references in the prefab.



[Color Space References]
Palette Color Textures

Size	4
Element 0	hsv_space
Element 1	colors2
Element 2	colors3
Element 3	003L

Palette Alpha Gradient Textures

Size	1
Element 0	alpha_gradient

Change the palette color texture (by index):
picker.ChangePaletteColorTexture(int: index);

Change the palette color texture with a new texture:
picker.ChangePaletteColorTexture(Texture2D: newTexture);

Change the palette alpha gradient texture (by index):
picker.ChangePaletteColorTexture(Texture2D: newTexture);

Change the palette alpha gradient texture with a new texture:
picker.ChangePaletteAlphaGradientTexture(Texture2D: newTexture);

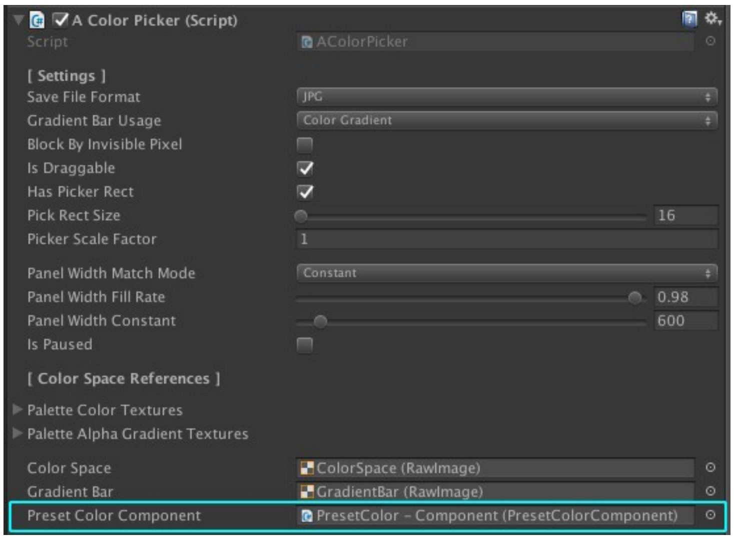
< Extra >
Make an UI object(e.g. Image, RawImage, Text) as draggable:
DDraggableUI.SetDraggable(GameObject: targetUI, bool: isEnabled);

Set canvases to be managed by the DcanvasLayerHandler, for automatically handle the canvas layers when the user drag/tap on the UI:
DCanvasLayerHandler.iAddCanvas(Canvas: targetCanvas);

(5) Preset Color Component

Preset Colors:

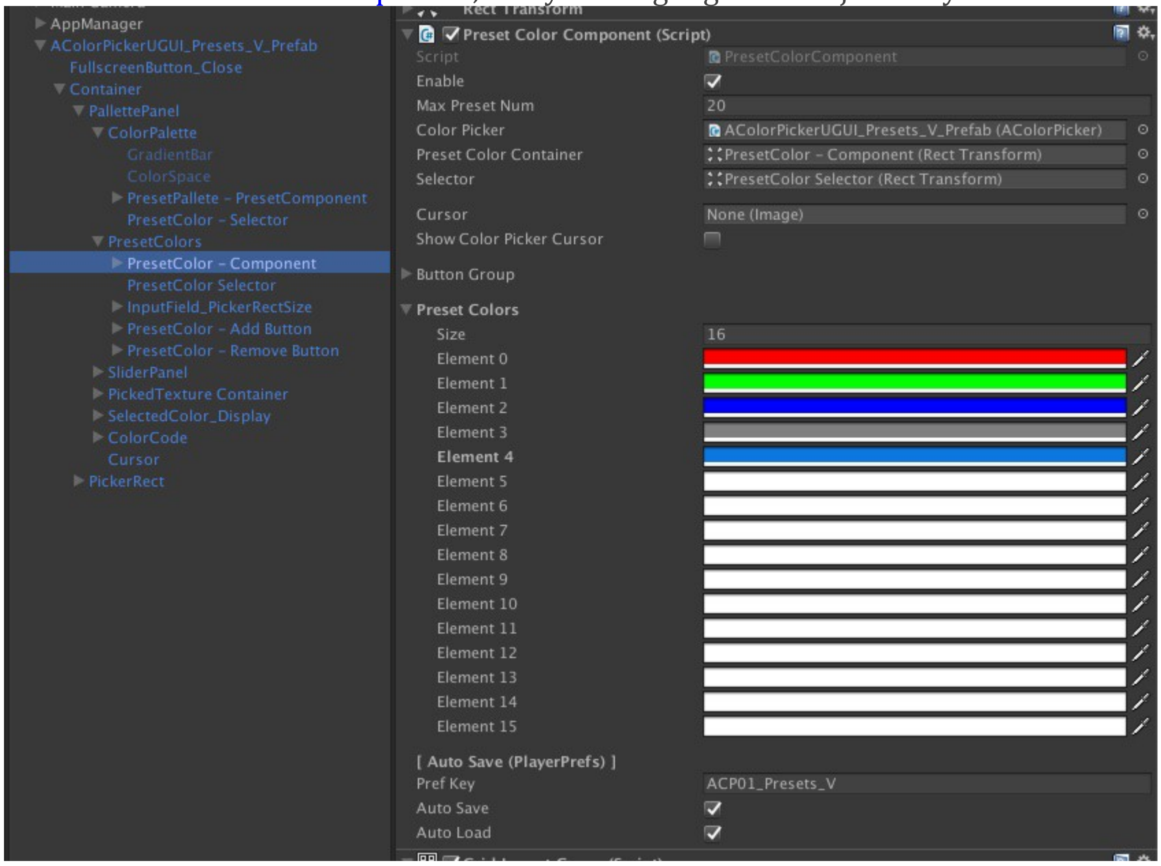
The PresetColorComponent is initially designed for setting up preset colors that can be set and pick at runtime. Users can store picked colors to the presets and pick from them later.
To configure the PresetColorComponent in the provided color picker prefabs, please open/drag the prefab to the root of your scene, select the parent transform of the prefab, now you can see the referenced component in the inspector view of AColorPicker:



A Color Picker (Script)

Script	AColorPicker
[Settings]	
Save File Format	JPG
Gradient Bar Usage	Color Gradient
Block By Invisible Pixel	
Is Draggable	✓
Has Picker Rect	✓
Pick Rect Size	16
Picker Scale Factor	1
Panel Width Match Mode	Constant
Panel Width Fill Rate	0.98
Panel Width Constant	600
Is Paused	
[Color Space References]	
Palette Color Textures	
Palette Alpha Gradient Textures	
Color Space	ColorSpace (RawImage)
Gradient Bar	GradientBar (RawImage)
Preset Color Component	PresetColor - Component (PresetColorComponent)

Click on the PresetColorComponent, Unity will highlight the object for you. Select the object:



- AppManager
- AColorPickerUGUI_Presets_V_Prefab
  - FullscreenButton_Close
  - Container
    - PallettePanel
      - ColorPalette
        - GradientBar
        - ColorSpace
      - PresetPallete - PresetComponent
      - PresetColor - Selector
      - PresetColors
        - PresetColor - Component
        - PresetColor Selector
      - InputField_PickerRectSize
      - PresetColor - Add Button
      - PresetColor - Remove Button
    - SliderPanel
    - PickedTexture Container
    - SelectedColor_Display
    - ColorCode
    - Cursor
  - PickerRect

Preset Color Component (Script)

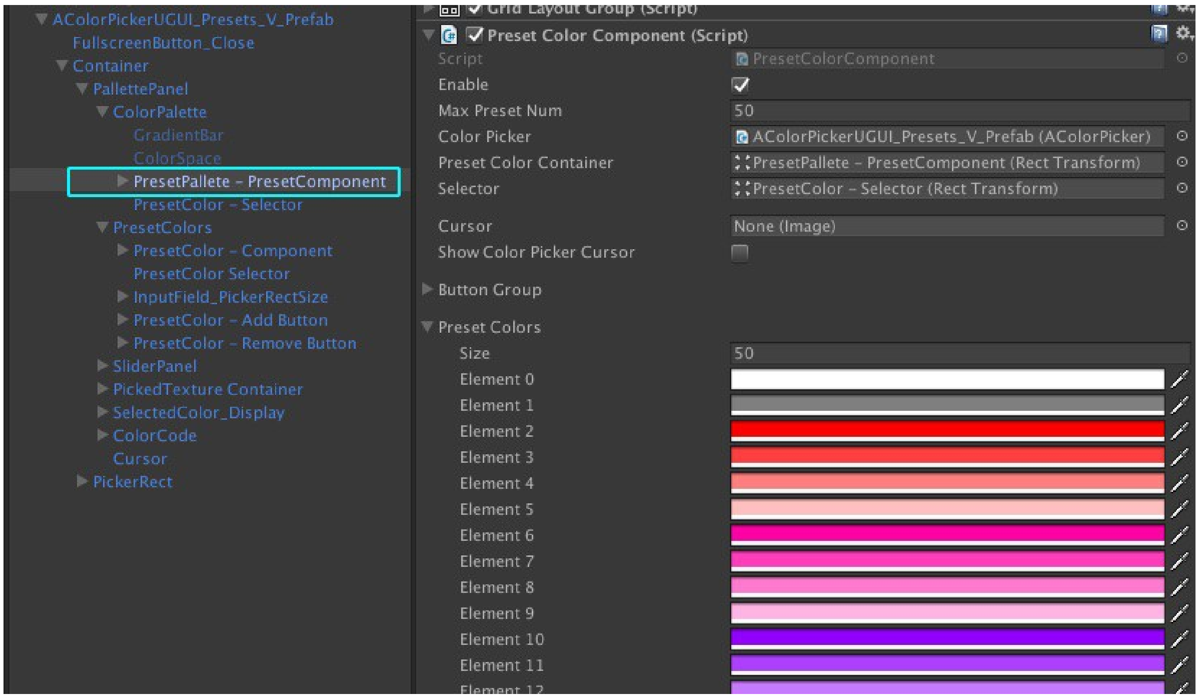
Script	PresetColorComponent
Enable	✓
Max Preset Num	20
Color Picker	AColorPickerUGUI_Presets_V_Prefab (AColorPicker)
Preset Color Container	PresetColor - Component (Rect Transform)
Selector	PresetColor Selector (Rect Transform)
Cursor	None (Image)
Show Color Picker Cursor	
Button Group	
Preset Colors	
Size	16
Element 0	🟥
Element 1	🟩
Element 2	🟦
Element 3	⬜
Element 4	🟦
Element 5	⬜
Element 6	⬜
Element 7	⬜
Element 8	⬜
Element 9	⬜
Element 10	⬜
Element 11	⬜
Element 12	⬜
Element 13	⬜
Element 14	⬜
Element 15	⬜
[Auto Save (PlayerPrefs)]	
Pref Key	ACP01_Presets_V
Auto Save	✓
Auto Load	✓

As you can see that the variables are quite self-explanatory by their names, or you can hover your mouse on them for more tips.
It is time for you to try it out, make changes and tests until you are satisfied with the prefab.

Use Preset Colors as Palette:

Except for using a palette texture for picking colors, the PresetColorComponent is also possible to use as a palette.

Here we show you where and how to find the example in our prefab. So you can follow it to begin your modifications and test it faster. For your convenient, we have created a color picker prefab using preset colors.
Please open/drag the prefab: ***AColorPickerUGUI_Presets_V_Prefab*** to the root of your scene.



```
▼ AColorPickerUGUI_Presets_V_Prefab
    FullscreenButton_Close
  ▼ Container
    ▼ PallettePanel
      ▼ ColorPalette
          GradientBar
          ColorSpace
        ▶ PresetPallete - PresetComponent
          PresetColor - Selector
      ▼ PresetColors
        ▶ PresetColor - Component
          PresetColor Selector
        ▶ InputField_PickerRectSize
        ▶ PresetColor - Add Button
        ▶ PresetColor - Remove Button
      ▶ SliderPanel
      ▶ PickedTexture Container
      ▶ SelectedColor_Display
      ▶ ColorCode
        Cursor
    ▶ PickerRect
```

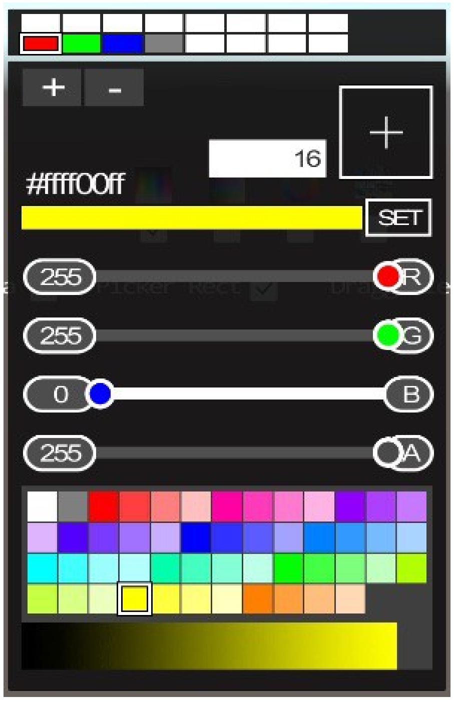
Grid Layout Group (Script)

Preset Color Component (Script)

Script	PresetColorComponent
Enable	✓
Max Preset Num	50
Color Picker	AColorPickerUGUI_Presets_V_Prefab (AColorPicker)
Preset Color Container	PresetPallete - PresetComponent (Rect Transform)
Selector	PresetColor - Selector (Rect Transform)
Cursor	None (Image)
Show Color Picker Cursor	☐
▶ Button Group	
▼ Preset Colors	
Size	50
Element 0	⬜
Element 1	▨
Element 2	🟥
Element 3	🟥
Element 4	🟥
Element 5	🟥
Element 6	🟪
Element 7	🟪
Element 8	🟪
Element 9	🟪
Element 10	🟪
Element 11	🟪
Element 12	

Similar settings like the normal preset colors we have shown you before.
The main difference is, it has no buttons(Add/Remove/Save/Load) enabled in the Button Group, and it doesn't need the Auto Save flag to be enabled because we don't want the user to change the main palette. Again, try it out youself now^^

The looking of the PresetColors picker at runtime:



Preset Colors

Use Preset Colors as a palette

(6) Make Your Own Color Picker Prefab

How if you still need some custom designed color picker for your app?

We suggest to clone from the provided prefabs, give the new prefab(s) an unique name.

Put the new prefab(s) in the ‘Resources’ folder. (Btw, you can move the unused prefabs out of the 'Resources' folder to avoid including them in the builds)

Modify the new prefab as you want.

Instantiate the new color picker in your script:

```
AColorPicker picker = AColorPicker.Create(string: YourColorPickerPrefabName);
```

or

```
AColorPicker picker = AColorPicker.Create(Transform: parentTransform,
string: YourColorPickerPrefabName);
```

THANK YOU

Thank you for using this package!
For any question, suggestion and bug report please contact us at swan.ob2@gmail.com.
Remember to rate this asset on the Asset Store. Your review is always appreciated, and very important to the development of this asset!

Review And Rating

Visit our asset page to find out more!
https://www.swanob2.com/assets

SWAN DEV