UIOBJECT3D

Render 3D models on any Unity Canvas

USER GUIDE

V1.00 (May 2017)

To view the latest online version of this document, visit

http://www.digital-legacy.co.za/UIObject3D/Documentation

TABLE OF CONTENTS

INTRODUCTION

UIObject3D allows you to render 3D objects on Unity UI canvases as if they were Unity image objects.

Features:

- Render any 3d model or prefab easily on any Unity UI Canvas
- Adjust the target object's rotation, offset, and distance from the camera
- Adjust lighting for the target object

WORKING WITH UIOBJECT3D

HOW UIOBJECT3D WORKS

UIObject3D works by creating an instance of your prefab or model in your scene, aiming a camera at it, and rendering what that camera sees to a sprite which is then rendered by a Unity UI Image component.

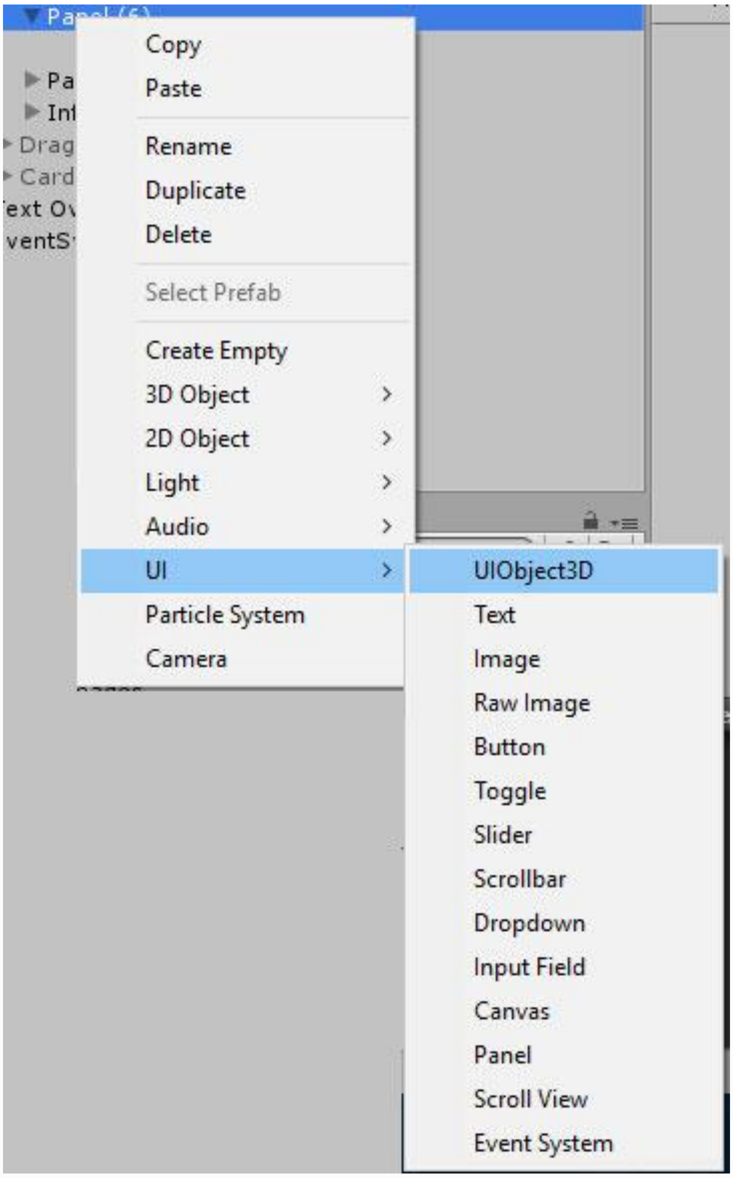
The target object and its camera are in a separate layer to the rest of your scene, so they are not visible to your regular cameras, nor is the rest of your scene visible to the target camera.

UIObject3D only renders an updated image when its properties change (rather than every frame), so it only has a minimal impact on the performance of your game or application. Additionally, the target object and camera will be cleaned up by UIObject3D whenever it is disabled or destroyed.

`</HowUiobject3dWorks>`

ADDING A NEW UIOBJECT3D TO YOUR UI

To add a new UIObject3D object, right click the desired location in the hierarchy window, and select **UI -> UIObject3d**



Add New UIObject3D

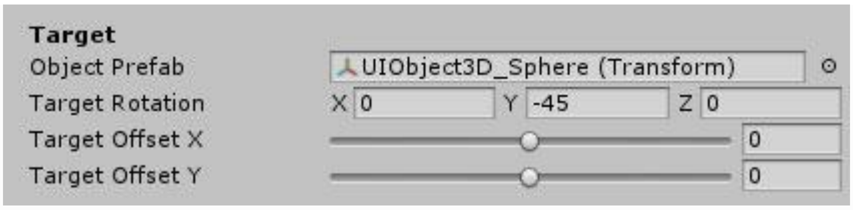
</AddingANewUiobject3dToYourUI>

Setting the Target Object

To specify which model or prefab to render, simply drag the object from the project window into the 'Object Prefab' field.

</SettingTheTargetObject>

Manipulating the Target Object



Target

Object Prefab: UIObject3D_Sphere (Transform)

Target Rotation: X 0 Y -45 Z 0

Target Offset X: 0

Target Offset Y: 0

UIObject3D Target properties

- **Target Rotation**

  This specifies the targets desired rotation using euclidean angles (x,y,z).

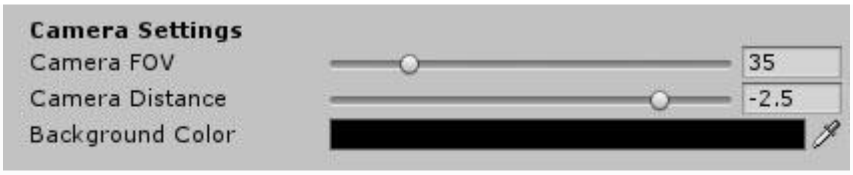
  Please note that this rotation is applied after the base prefab or model's rotation, so if your prefab is already set to your desired rotation, then you won't need to manipulate this property.

- **Target Offset X / Y**

  These properties allow you to move the target object left/right (X) and up/down (Y) relative to the center of the image and accepts values between -1 and 1, where 0 is in the center.

</ManipulatingTheTargetObject>

MANIPULATING THE TARGET CAMERA



Camera Settings

Camera FOV 35

Camera Distance -2.5

Background Color

UIObject3D Camera properties

- **Camera FOV**

  This adjusts the field-of-view of the target camera. (default value: 35)

- **Camera Distance**

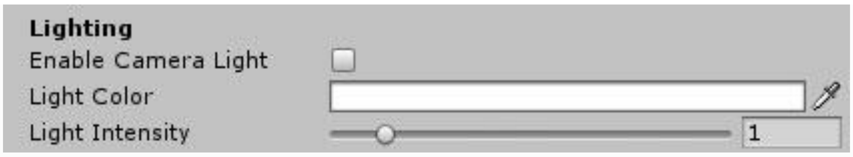
  This adjusts how far away the camera is from the target object. The lower the value, the further away the camera will be and the smaller the object will appear to be (default value: -3.5)

- **Background Color**

  This specifies an optional background color for the image. The default value is transparent.

</ManipulatingTheTargetCamera>

MANIPULATING LIGHTING FOR THE TARGET OBJECT



Lighting

Enable Camera Light □

Light Color

Light Intensity 1

UIObject3D Lighting properties

- **Enable Camera Light**

  If this property is enabled, then a Unity Light object will be attached to the camera, lighting up the target object with the chosen color and intensity.

- **Light Color**

  Specifies the color of the light object (if enabled).

- **Light Intensity**

  Specifies the intensity of the light object (if enabled).

Regard directional lights:

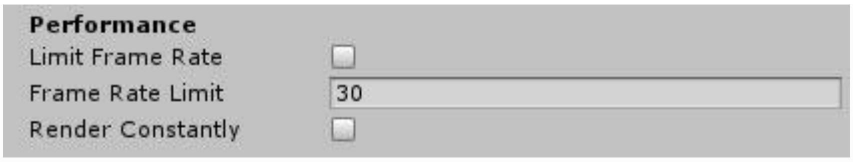
By default, UIObject3D targets will be lit by directional lighting within your scene (if any directional lights are present).

If you'd like to prevent this, then you will need to adjust the directional light's **Culling Mask** and remove the *UIObject3D* layer.

`</ManipulatingLightingForTheTargetObject>`

PERFORMANCE

By default, UIObject3D will only render an update image if any of its properties change (whether that change is made via the inspector, or in code), which helps keep its performance impact to a minimum. However, in some cases you may wish to take advantage of some additional performance-enhancing features UIObject3D provides, such as when you are updating large numbers of UIObject3D objects each frame.



Performance
Limit Frame Rate □
Frame Rate Limit 30
Render Constantly □

UIObject3D Performance properties

- **Limit Frame Rate**

  If this property is enabled, then UIObject3D will never exceed the frame rate limit specified by the 'Frame Rate Limit'. If this property is disabled, then it will render at (up to) the global frame rate.

- **Frame Rate Limit**

  Specifies the maximum number of frames per second at which to render the target object.

- **Render Constantly**

  If this property is enabled, then the UIObject3D image will be rendered (updated) constantly (using the frame rate limit, if enabled), even if none of the UIObject3D properties change.

  This may be useful if, for example, your prefab has animations which you wish to be visible. However, enabling this option will cause UIObject3D to render a lot more frequently than it normally does, which may have a performance impact.

</Performance>

ADDITIONAL EFFECTS

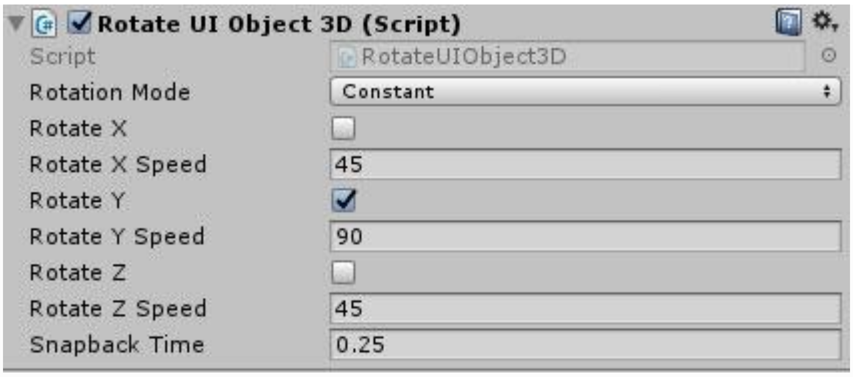
As UIObject3D produces a regular sprite which is rendered by a Unity UI Image component (specifically, it uses 'UIObject3DImage' which extends 'Image'), you can manipulate the result using any effects which would normally work on an 'Image' component, for example:

- You can manipulate the image's 'color' property to apply an overlay effect.
- You can use 'Shadow' and 'Outline' components as you would with a regular image.
- You can change the images type from 'Simple' to 'Filled' and manipulate it's 'Fill Method', 'Origin', 'Fill Amount', etc. properties

`</AdditionalEffects>`

THE ROTATEUIOBJECT3D COMPONENT

In order to make rotating objects (a common scenario) easier, UIObject3D provides the 'RotateUIObject3D' component which can be added to the same GameObject as the UIObject3D.



☑ Rotate UI Object 3D (Script)

Script	RotateUIObject3D
Rotation Mode	Constant
Rotate X	□
Rotate X Speed	45
Rotate Y	☑
Rotate Y Speed	90
Rotate Z	□
Rotate Z Speed	45
Snapback Time	0.25

RotateUIObject3D properties

- **Rotation Mode**

  This property specifies when the object should be rotated. The options are:

  - Constant : this object will be rotated constantly.
  - When Mouse Is Over : This object will only be rotated when the mouse is over it.
  - When Mouse Is Over Then Snap Back : This object will only be rotated when the mouse is over it. When the mouse is moved away, it will snap back to its original position.

- **Rotate X / Y / Z**

  Specifies that the object should be rotated in the specified Euclidean plane.

- **Rotation Speed X / Y / Z**

  Specifies how fast the object should be rotated in the specified plane. Negative values are acceptable.

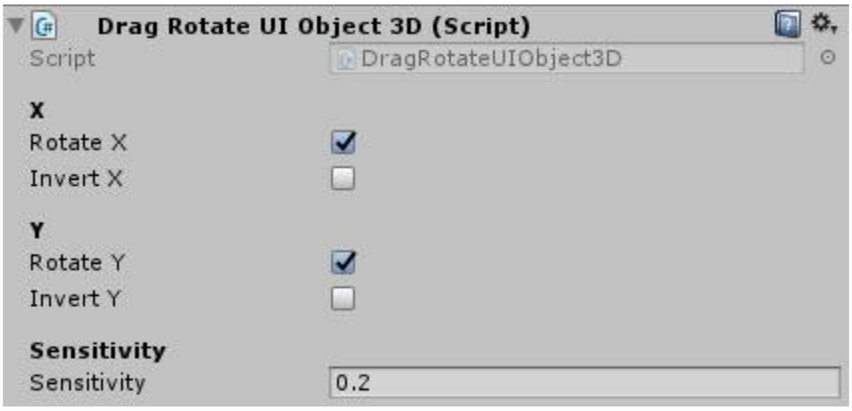
- **Snapback Time**

  How long it should take for the object to return to its original position when 'When Mouse Is Over Then Snap Back' rotation mode is used.

</TheRotateuiobject3dComponent>

THE DRAGROTATEUIOBJECT3D COMPONENT

The DragRotateUIObject3D component allows you to drag the target object and rotate it.



Drag Rotate UI Object 3D (Script)

Script DragRotateUIObject3D

X

Rotate X ☑

Invert X ☐

Y

Rotate Y ☑

Invert Y ☐

Sensitivity

Sensitivity 0.2

DragRotateUIObject3D properties

- **Rotate X / Y**

  Allow rotation in the specified direction (X - Left/Right, Y - Up/Down)

- **Invert X / Y**

  If these properties are enabled, then the object will rotate in the opposite direction on the specified plane.

- **Sensitivity**

  Specifies how responsive the object is to being dragged. A lower value will result in the object taking more effort to drag.