uResize

USER GUIDE

V1.00 (May 2018)

To view the latest online version of this document, visit

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

TABLE OF CONTENTS

INTRODUCTION

uResize is a Unity component which allows you to quickly and easily resize Unity UI elements at runtime by dragging the edges and/or corners.

Features:

- Simply add the uResize component to an element and it's ready to go
- Highly customizeable
- Provides the optional uResize_CursorController component which you can use to control the appearance of the mouse cursor
- No programming required
- Provides events (OnResizeBegin/OnResizeEnd/OnPointerEnterResizeListener/OnPointerExitResizeListener) which you can use to extend the functionality of uResize
- Works in all canvas render modes
- Documented source code included
- Compatible with XmlLayout

WORKING WITH URESIZE

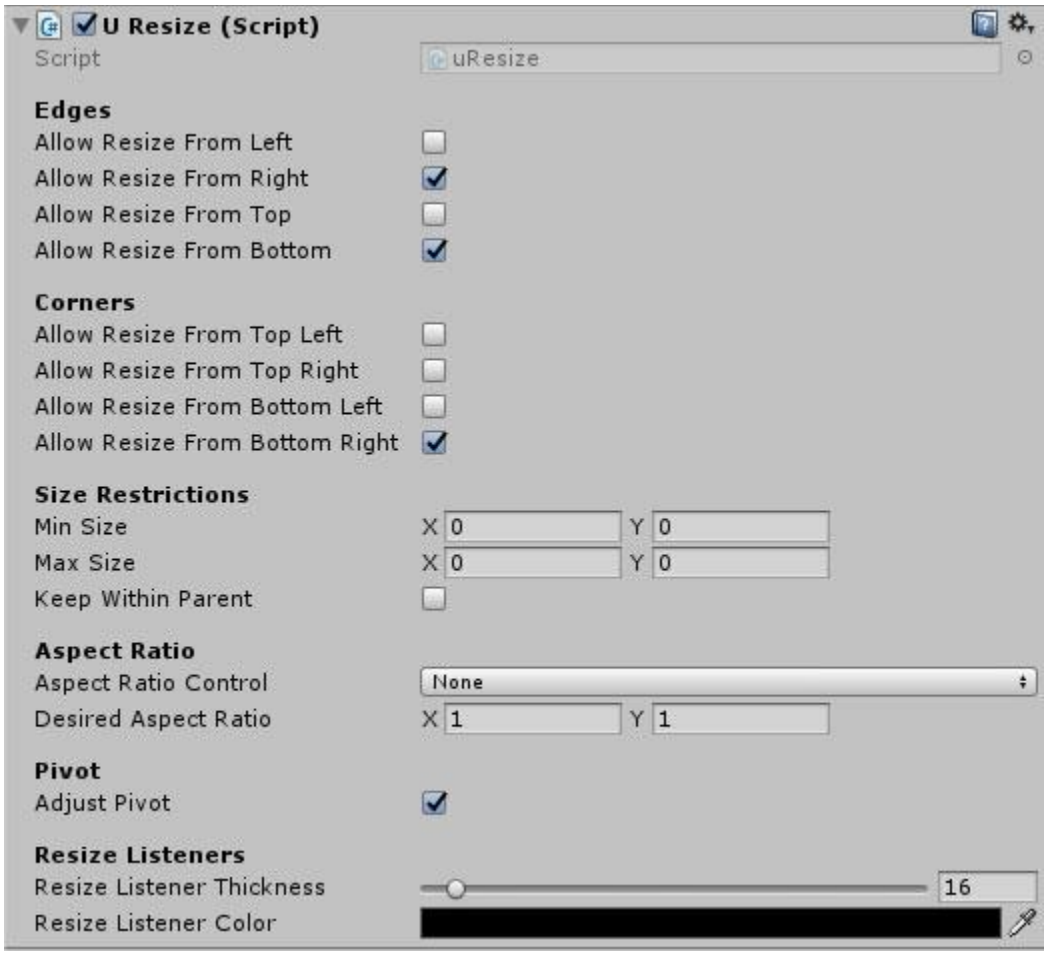
ADDING A NEW URESIZE COMPONENT

Adding a `uResize` component to an existing UI element is easy. Simply click the 'Add Component' button in the inspector, then select '**UI -> uResize**'.

Alternatively, you can also add it via code, e.g.

```
myUIElement.gameObject.AddComponent<DigitalLegacy.UI.Sizing.uResize>();
```

THE URESIZE COMPONENT



The uResize Component

- **Edges**
  These options allow you to specify from which sides the element will be resizeable.
- **Corners**
  These options allow you to specify from which corners the element will be resizeable.
  - **Size Restrictions**
    Min/Max Size: These options allow you to set an optional minimum and maximum size for the element.
    If left at zero, these options will be ignored.
  - Keep Within Parent: If this option is enabled, then the element will not allow its edges to be resized beyond the edges of its parent.

  Note: these options take precedence over anything else, including aspect ratios.

- **Aspect Ratio**
  These options allow you to determine whether or not resizing utilizes an aspect ratio, and how that ratio is controlled.

Aspect Ratio Modes

- None
- Auto: uResize will determine whether width controls height or vice versa automatically based on which edge is dragged.
- Width Controls Height: If this option is used, then increasing/decreasing the width will adjust the height using the ratio provided
- Height Controls Width: If this option is used, then increasing/decreasing the height will adjust the width using the ratio provided

Note: When applying the aspect ratio, Min and Max size values will still be respected, even if they don't match the ratio. Similarly, uResize may fail to keep the required aspect ratio if 'Keep Within Parent' is enabled.

- **Pivot**
  The 'Adjust Pivot' option allows you to specify how the element will appear to be resized.

  With 'Adjust Pivot' enabled, uResize will resize the element such that it appears that the edge(s) which are being dragged move, while all other edges remain in place.

  Width this option disabled, uResize will use whatever pivot had already been set by the element, e.g. if the pivot is (0.5,0.5) then the element will appear to grow or shrink from the center.

  Note: uResize will restore the original pivot value once the resize has been completed.

- **Resize Listeners**

  Resize listeners are essentially the borders of the element, and they are responsible for responding to drag events/etc.

  The 'Resize Listener Thickness' property allows you to adjust how thick the listeners are - the thicker they are, the easier it will be to grab the edges of the element.

  The 'Resize Listener Color' property allows you to make the listeners visible and specify what color they are. By default they are transparent, so the user cannot see them.

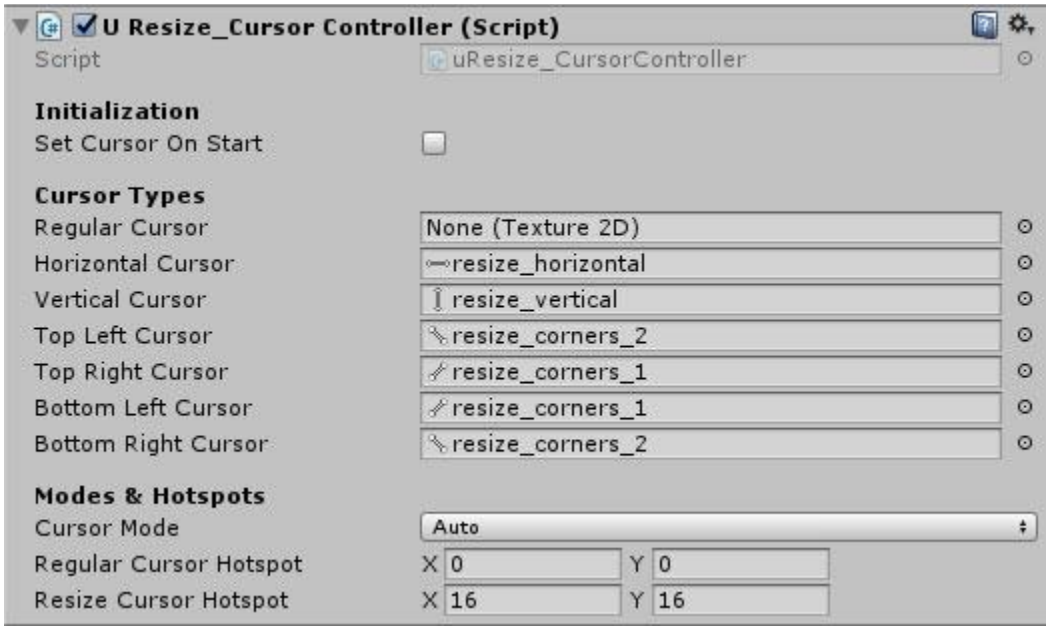
  Note: resize listeners are only created at runtime, so you won't be able to see any changes to these properties in edit mode.

THE URESIZE CURSOR CONTROLLER COMPONENT

The `uResize_CursorController` component allows you to specify cursors for uResize to use.

This component is completely optional. If you wish to use it, you can add it via the 'Add Component' menu (**UI -> uResize Cursor Controller**).

uResize_CursorController should be compatible with most custom cursor controllers, however, if necessary, you can modify your custom cursor controller to utilize uResize events (OnResizeBegin/OnResizeEnd/OnPointerEnterResizeListener/OnPointerExitResizeListener) in much the same way as uResize_CursorController does.



The uResize Cursor Controller Component

- **Initialization**

  The 'Set Cursor On Start' property allows you to determine whether or not the controller will attempt to control the cursor from the start of the scene.

- **Cursor Types**
  - Regular Cursor:
    Specifies the default cursor to be used when **not** resizing or hovering the mouse over a resize listener.

    By default, this is null. With this value, the system default cursor will be used instead.

  - Horizontal Cursor:
    Specifies the cursor to use for the left and right edges of the element.
  - Vertical Cursor:
    Specifies the cursor to use for the top and bottom edges of the element.

- Top Left / Top Right / Bottom Left / Bottom Right Cursor:
  Specifies the cursor to use for the elements corners.

- **Modes & Hotspots**
  - Cursor Mode
    - Auto: Allows Unity to use a hardware cursor where possible
    - Force Software: Forces Unity to use a software cursor
  - Regular Cursor Hotspot:
    Specifies the point of the regular cursor which is active. By default, this is the top-left corner.
  - Resize Cursor Hotspot:
    Specifies the point of the resize cursor which is active. By default, this is the center of the cursor.

uRESIZE EVENTS

If you wish, you can extend `uResize` by responding to events triggered by the component.

For example, the `uResize_CursorController` component responds to these events by changing the cursor.

The following example shows how the uResize_CursorController attaches event-listeners to the uResize event-handlers:

EXAMPLE

```
private void Start()
{
    // in this case, this component is on the same GameObject as uResize
    var uResize = this.GetComponent<uResize>();

    // OnpointerEnterListener(), OnPointerExitListener(), OnResizeBegin(), and OnResizeEnd()
    // are all methods in this component
    uResize.OnPointerEnterResizeListener.AddListener(OnPointerEnterListener);
    uResize.OnPointerExitResizeListener.AddListener(OnPointerExitListener);

    uResize.OnResizeBegin.AddListener(OnResizeBegin);
    uResize.OnResizeEnd.AddListener(OnResizeEnd);
}
```

uRESIZE EVENT TYPES

- **OnResizeBegin (eResizeListenerType type)**
  This event is called when uResize begins a resize event (when the user begins to drag an edge/corner).

- **OnResizeEnd ()**
  This event is called when uResize ends a resize event (when the user releases an edge/corner).

- **OnPointerEnterResizeListener (eResizeListenerType type)**
  This event is called whenever the pointer hovers over a resize listener.

- **OnPointerExitResizeListener (eResizeListenerType type)**
  This event is called whenever the pointer exits a resize listener.