# AnchorExtensions Class Reference

## Static Public Member Functions

static Vector2 **GetAnchorsPosition** (this RectTransform tr, bool centeredPivot=false)

Get the anchors position. The intent of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent. More...

static Vector2 **GetAnchorsSize** (this RectTransform tr)

Get the anchors size. This just returns tr.anchorMax - tr.anchorMin because size is determined by anchorMax. Note: If the anchors are placed together this returns (0,0). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent. More...

static void **SetAnchorsPosition** (this RectTransform tr, Vector2 targetPos, bool centeredPivot=false)

Move the anchors, moving anchors also moves the image as you may know. This is the same than moving the anchors holding control + shift key in the Unity editor. To set the position of another object you must pass: SetPosition(tr.anchorMin.x, tr.anchorMin.y) because these are the values that determines the anchors position of the object. For a more simple sintax use SetPosition(targetTransform.GetPosition()) or SetPosition(targetTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent. More...

static void **SetAnchorsPosition** (this RectTransform tr, RectTransform targetPosFromOtherObject)

Move the anchors, moving anchors also moves the image as you may know. This is the same than moving the anchors holding control + shift key in the Unity editor. To set the position of another object you must pass: SetPosition(tr.anchorMin.x, tr.anchorMin.y) because these are the values that determines the anchors position. For a more simple sintax use SetPosition(targetTransform.GetPosition()) or SetPosition(targetTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent. More...

static void **SetAnchorsSize** (this RectTransform tr, Vector2 targetSize)

Resize the anchors, resizing anchors also resizes the image as you may know. This is the same than moving the anchors holding shift key in the Unity editor. To set the size of another object you must pass: SetSize(targetTransform.GetSize()) Or for a more simple sintax use SetSize(targetRectTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent. More...

static void **SetAnchorsSize** (this RectTransform tr, RectTransform targetPosFromOtherObject)

Resize the anchors, resizing anchors also resizes the image as you may know. This is the same than moving the anchors holding shift key in the Unity editor. To set the size of another object you must pass: SetSize(targetTransform.GetSize()) Or for a more simple sintax use SetSize(targetTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent. More...

static Vector2 **ScreenSpaceToAnchorCoordinates** (this RectTransform tr, Vector2 screenSpacePoint)

Converts a screen space coordinate (pixels) to a range between 0 to 1 (used by the anchors), I call it "anchor space". An example: To make the anchors follow the mouse, place the anchors in the corners of the image and

| | |
|---|---|
| | add this line of code to the Update function of a component in the target UI object: transform.SetAnchorsPosition(transform.ScreenSpaceToAnchorSpace(Input.mousePosition)); It's not a good idea to make a mouse pointer with this, but it's usefull in other cases like a dragable object. Note: This probably will not work as you expect when the canvas is configured as "Word Space" because everithing becomes 3D. More... |
| static Vector2 | **AnchorCoordinatesToScreenSpace** (this RectTransform tr, Vector2 anchorSpacePoint)<br>Converts a range between 0 to 1 (used by the anchors), I call it "anchor space" to a screen space coordinate (pixels). Note: This probably will not work as you expect when the canvas is configured as "Word Space" because everithing becomes 3D. More... |
| static Vector2 | **ThisAnchorCoordinatesToCanvasAnchorCoordinates** (this RectTransform tr, Vector2 anchorSpacePoint)<br>An anchor space point at the level of this object to be converted to an anchor space point at the level of the canvas. Example: If this object is inside a container which is a child of the canvas and is located at a position of (0.5, 0.5), this method will return (0.5, 0.5) when you pass (0, 0). More... |
| static Vector2 | **CanvasAnchorCoordinatesToThisAnchorCoordinates** (this RectTransform tr, Vector2 anchorSpacePoint)<br>A canvas anchor space point to be converted to an anchor space point at the level of this object. Example: If this object is inside a container which is a child of the canvas and is located at a position of (0.5, 0.5), this method will return (-0.5, -0.5) when you pass (0, 0). More... |

## Static Public Attributes

| | |
|---|---|
| static bool | **ThrowExeptions** = true |

## Detailed Description

Definition at line **5** of file **AnchorExtensions.cs**.

## Member Function Documentation

**static Vector2 AnchorExtensions.AnchorCoordinatesToScreenSpace ( this RectTransform** **tr,**
**Vector2** **anchorSpacePoint**
**)**
static

Converts a range between 0 to 1 (used by the anchors), I call it "anchor space" to a screen space coordinate (pixels). Note: This probably will not work as you expect when the canvas is configured as "Word Space" because everithing becomes 3D.

**Parameters**

| | |
|---|---|
| **tr** | |
| **anchorSpacePoint** | The anchor space point at the level of this object to be converted to screen space coordinate point (pixel) |

**Returns**

Definition at line **157** of file **AnchorExtensions.cs**.

**static Vector2 AnchorExtensions.CanvasAnchorCoordinatesToThisAnchorCoordinates ( this RectTransform tr,**
**Vector2 anchorSpacePoint**
**)** static

A canvas anchor space point to be converted to an anchor space point at the level of this object. Example: If this object is inside a container which is a child of the canvas and is located at a position of (0.5, 0.5), this method will return (-0.5, -0.5) when you pass (0, 0).

**Parameters**

- **tr**
- **anchorSpacePoint** Point to be converted to canvas anchor space point

**Returns**

Definition at line **226** of file **AnchorExtensions.cs**.

**static Vector2 AnchorExtensions.GetAnchorsPosition ( this RectTransform tr,**
**bool centeredPivot =** `false`
**)** static

Get the anchors position. The intent of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent.

**Parameters**

- **tr**
- **centeredPivot** If false the pivot is in the down left corner, if true the pivot is in the center

**Returns**

Definition at line **17** of file **AnchorExtensions.cs**.

**static Vector2 AnchorExtensions.GetAnchorsSize ( this RectTransform tr )** static

Get the anchors size. This just returns tr.anchorMax - tr.anchorMin because size is determined by anchorMax. Note: If the anchors are placed together this returns (0,0). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent.

**Parameters**

- **tr**

**Returns**

Definition at line **35** of file **AnchorExtensions.cs**.

**static Vector2 AnchorExtensions.ScreenSpaceToAnchorCoordinates ( this RectTransform tr,**
**Vector2 screenSpacePoint**
**)** static

Converts a screen space coordinate (pixels) to a range between 0 to 1 (used by the anchors), I call it "anchor space". An example: To make the anchors follow the mouse, place the anchors in the corners of the image and add this line of code to the Update function of a component in the target UI object: transform.SetAnchorsPosition(transform.ScreenSpaceToAnchorSpace(Input.mousePosition)); It's not a good idea to make a mouse pointer with this, but it's usefull in other cases like a dragable object. Note: This probably will not work as you expect when the canvas is configured as "Word Space" because everithing becomes 3D.

**Parameters**

- **tr**
- **screenSpacePoint** The screen space point (pixel) to convert to anchor space point at the level of this object

**Returns**

Definition at line **143** of file **AnchorExtensions.cs**.

**static void AnchorExtensions.SetAnchorsPosition ( this RectTransform tr,**
**Vector2 targetPos,**
**bool centeredPivot = `false`**
**)** static

Move the anchors, moving anchors also moves the image as you may know. This is the same than moving the anchors holding control + shift key in the Unity editor. To set the position of another object you must pass: SetPosition(tr.anchorMin.x, tr.anchorMin.y) because these are the values that determines the anchors position of the object. For a more simple sintax use SetPosition(targetTransform.GetPosition()) or SetPosition(targetTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent.

**Parameters**

- **tr**
- **targetPos**
- **centeredPivot** If false the pivot is in the down left corner, if true the pivot is in the center

Definition at line **53** of file **AnchorExtensions.cs**.

**static void AnchorExtensions.SetAnchorsPosition ( this RectTransform tr,**
**RectTransform targetPosFromOtherObject**
**)**

static

Move the anchors, moving anchors also moves the image as you may know. This is the same than moving the anchors holding control + shift key in the Unity editor. To set the position of another object you must pass: SetPosition(tr.anchorMin.x, tr.anchorMin.y) because these are the values that determines the anchors position. For a more simple sintax use SetPosition(targetTransform.GetPosition()) or SetPosition(targetTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent.

**Parameters**

| | |
|---|---|
| **tr** | |
| **targetPosFromOtherObject** | Copy the position from another RectTransform, can be an object inside another container or even another canvas, the position will be translated |

Definition at line **77** of file **AnchorExtensions.cs**.

**static void AnchorExtensions.SetAnchorsSize ( this RectTransform tr,**
**Vector2 targetSize**
**)**

static

Resize the anchors, resizing anchors also resizes the image as you may know. This is the same than moving the anchors holding shift key in the Unity editor. To set the size of another object you must pass: SetSize(targetTransform.GetSize()) Or for a more simple sintax use SetSize(targetRectTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent.

**Parameters**

| | |
|---|---|
| **tr** | |
| **targetSize** | |

Definition at line **102** of file **AnchorExtensions.cs**.

**static void AnchorExtensions.SetAnchorsSize ( this RectTransform tr,**
**RectTransform targetPosFromOtherObject**
**)** static

Resize the anchors, resizing anchors also resizes the image as you may know. This is the same than moving the anchors holding shift key in the Unity editor. To set the size of another object you must pass: SetSize(targetTransform.GetSize()) Or for a more simple sintax use SetSize(targetTransform). The aim of this function is manipulating anchors with only a single Vector2, specially usefull for tweens and clean code. In most cases it's a good idea to move and resize images using anchors because everything becomes screen size independent.

**Parameters**

- **tr**
- **targetPosFromOtherObject** Copy the size from another RectTransform, can be an object inside another container or even another canvas, the position will be translated

Definition at line **119** of file **AnchorExtensions.cs**.

**static Vector2**
**AnchorExtensions.ThisAnchorCoordinatesToCanvasAnchorCoordinates ( this RectTransform tr,**
**Vector2 anchorSpacePoint**
**)** static

An anchor space point at the level of this object to be converted to an anchor space point at the level of the canvas. Example: If this object is inside a container which is a child of the canvas and is located at a position of (0.5, 0.5), this method will return (0.5, 0.5) when you pass (0, 0).

**Parameters**

- **tr**
- **anchorSpacePoint**

**Returns**

Definition at line **171** of file **AnchorExtensions.cs**.

---

The documentation for this class was generated from the following file:

- **AnchorExtensions.cs**