Simple Crosshair Generator Scripting API

This document describes the API that Simple Crosshair Generator provides.

Contents

SimpleCrosshair - Class 1

Properties 1

Methods 1

InitialiseCrosshairImage 2

GenerateCrosshair 2

SetColor 2

SetThickness 2

SetSize 2

SetGap 3

DrawCrosshair 3

GetSize 3

GetThickness 4

GetGap 4

GetColor 4

GetCrosshair 4

Code example 4

Crosshair – Class 5

Properties 5

CrosshairColorChannel – Enum 6

Values 6

SimpleCrosshair - Class

To edit a crosshair at runtime, you can use the scripting API for the SimpleCrosshair component. This component includes methods to modify crosshair properties and redraw the crosshair.

Properties

Property	Description
m_crosshairImage	The Image in your Scene that this script draws the crosshair to.

Methods

The following is the list of methods in the SimpleCrosshair class.

InitialiseCrosshairImage

Creates a Canvas and an Image in your Scene and sets them up with the correct properties. The Image that this method creates becomes the target that the crosshair generator draws the crosshair to.

GenerateCrosshair

Generates a crosshair from the currently defined crosshair properties and then paints the crosshair to the target Image in the Scene.

SetColor

Changes the color of the crosshair.

Parameters

Parameter	Type	Description
channel	CrosshairColorChannel	The color channel to set the value for.
value	int	The value to set the color channel to. Set this to a value between 0 and 255.
redrawCrosshair	bool	Specifies whether to redraw the crosshair with the updated value. If you set this to true, the generator redraws the crosshair. If you set this to false, you can still use GenerateCrosshair to draw the crosshair with the updated value.

or

Parameter	Type	Description
color	Color	The color to set the crosshair to. This uses Unity's Color struct. For more information on what values you can use, see Unity's documentation on Color.
redrawCrosshair	bool	Specifies whether to redraw the crosshair with the updated value. If you set this to true, the generator redraws the crosshair. If you set this to false, you can still use GenerateCrosshair to draw the crosshair with the updated value.

SetThickness

Changes the thickness of the crosshair. This is the width (in pixels) of each crosshair line.

Parameters

Parameter	Type	Description
newThickness	int	The thickness to set the crosshair to. If you use a value lower than **1**, this method sets the width to **1**.
redrawCrosshair	bool	Specifies whether to redraw the crosshair with the updated value. If you set this to true, the generator redraws the crosshair. If you set this to false, you can still use GenerateCrosshair to draw the crosshair with the updated value.

SetSize

Changes the size of the crosshair. This is the length (in pixels) of each crosshair line.

Parameters

Parameter	Type	Description
newSize	int	The size to set the crosshair to. If you use a value lower than **1**, this method sets the size to **1**.
redrawCrosshair	bool	Specifies whether to redraw the crosshair with the updated value. If you set this to true, the generator redraws the crosshair. If you set this to false, you can still use GenerateCrosshair to draw the crosshair with the updated value.

SetGap

Changes the gap size of the crosshair. This is the distance (in pixels) from the center of the crosshair to the start of each crosshair line.

Parameters

Parameter	Type	Description
newGap	int	The gap to set the crosshair to have. If you use a value lower than **0**, this method sets the gap to **0**.
redrawCrosshair	bool	Specifies whether to redraw the crosshair with the updated value. If you set this to true, the generator redraws the crosshair. If you set this to false, you can still use GenerateCrosshair to draw the crosshair with the updated value.

DrawCrosshair

Takes the crosshair that you pass into this method, draws it to a Texture2D, and returns the Texture2D.

Parameters

Parameter	Type	Description
crosshair	Crosshair	The crosshair to draw and return as a Texture2D. If you do not pass in a crosshair as a parameter, this method draws the current crosshair and returns it as a Texture2D.

Returns

Type	Description
Texture2D	A Texture2D that contains the drawn crosshair.

You can use the Texture2D that this method returns to display a crosshair in your Scene. For example, you can display a crosshair in an Image component. For this, create a Sprite and assign it to the Image:

- To create the Sprite, use the following code example, where **crosshairTexture** is the Texture2D that this method returns:
```
Sprite crosshairSprite = Sprite.Create(crosshairTexture,
    new Rect(0, 0, crosshairTexture.width, crosshairTexture.height),
    Vector2.one / 2);
```
- To assign the Sprite to an Image, get a reference to the Image in your Scene and set its **sprite** property to the Sprite you created. Use the following code example, where **crosshairImage** is a reference to an Image in your Scene:
```
crosshairImage.sprite = crosshairSprite;
```

GetSize

Gets the current size of the crosshair lines.

Returns

Type	Description
int	The current length (in pixels) of the crosshair lines.

GetThickness

Gets the current thickness of the crosshair lines.

Returns

Type	Description
int	The current width (in pixels) of the crosshair lines.

GetGap

Gets the current gap at the center of the crosshair.

Returns

Type	Description
int	The current distance (in pixels) from the center of the crosshair to the beginning of each crosshair line.

GetColor

Gets the current color of the crosshair.

Returns

Type	Description
Color	The current color of the crosshair lines.

GetCrosshair

Gets the Crosshair instance that defines the current crosshair.

Returns

Type	Description
Crosshair	The Crosshair instance that defines the current crosshair.

Code example

For an example on how to use the API, see `SimpleCrosshairAPIExample.cs`. If you add this script to a GameObject in your Scene, when you enter Play Mode, you can use the keyboard to change the properties of the crosshair. The controls are:

- Q: Increase gap size.
- A: Decrease gap size.
- W: Increase thickness.
- S: Decrease thickness.
- E: Increase size.
- D: Decrease size.
- Spacebar: Set the crosshair to a random color.

```
using UnityEngine;

public class SimpleCrosshairAPIExample : MonoBehaviour
{
    public SimpleCrosshair simpleCrosshair;

    private void Start()
    {
        if(simpleCrosshair == null)
        {
            Debug.LogError("You have not set the target SimpleCrosshair. Disabling.");
            enabled = false;
        }
    }

    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.Q))
        {
            int curGap = simpleCrosshair.GetGap();
            simpleCrosshair.SetGap(curGap + 2, true);
        }
        else if (Input.GetKeyDown(KeyCode.A))
        {
            int curGap = simpleCrosshair.GetGap();
            simpleCrosshair.SetGap(curGap - 2, true);
        }
        if (Input.GetKeyDown(KeyCode.W))
        {
            int curThickness = simpleCrosshair.GetThickness();
            simpleCrosshair.SetThickness(curThickness + 2, true);
        }
        else if (Input.GetKeyDown(KeyCode.S))
        {
            int curThickness = simpleCrosshair.GetThickness();
            simpleCrosshair.SetThickness(curThickness - 2, true);
        }
        if (Input.GetKeyDown(KeyCode.E))
        {
            int curSize = simpleCrosshair.GetSize();
            simpleCrosshair.SetSize(curSize + 2, true);
        }
        else if (Input.GetKeyDown(KeyCode.D))
        {
            int curSize = simpleCrosshair.GetSize();
            simpleCrosshair.SetSize(curSize - 2, true);
        }
        if (Input.GetKeyDown(KeyCode.Space))
        {
            simpleCrosshair.SetColor(Random.ColorHSV(), true);
        }
    }
}
```

Crosshair – Class

A class that stores properties to define the shape and color of a crosshair.

Properties

Property	Description

Size	The length (in pixels) of each crosshair line.
Thickness	The width (in pixels) of each crosshair line.
Gap	The distance (in pixels) from the center of the crosshair to the beginning of each crosshair line.
Color	The color of the crosshair.
SizeNeeded	Specifies the minimum size (in pixels) that the crosshair texture needs to be. This is equal to whichever value is largest out of (Gap * 2) + (Size * 2) or Thickness.

CrosshairColorChannel – Enum

An enum that the generator uses to differentiate between color channels.

Values

Value	Description
RED	Refers to the red color channel.
GREEN	Refers to the green color channel.
BLUE	Refers to the blue color channel.
ALPHA	Refers to the alpha color channel.