2D SHAPER

2D Shaper is a **full featured, high performance** 2D drawing library.

It provides essential 2D drawing capabilities for IMGUI (the Immediate Mode GUI system) or for any other script that can leverage OpenGL 2D drawing like camera's OnPostRender events, achieving outstanding results for UI, charts, HUDs, editor customization, etc.

Features:

- Simple shapes drawing: points, lines, rects, quads circles, arcs, triangles, polygons, arrows...
- Fully configurable line style, including: line thickness, color and dashed line pattern
- Full support for sprite rendering with configurable size, orientation and overlay color
- Full transparency support
- Optional fill texture for different patterns
- Easy text drawing using Bitmap Fonts with configurable size, color and orientation, or with fully customizable text style using GUI labels

Other utilities:

- Get world position and size of a UI element
- Coordinates conversion between GUI, screen and OpenGL spaces

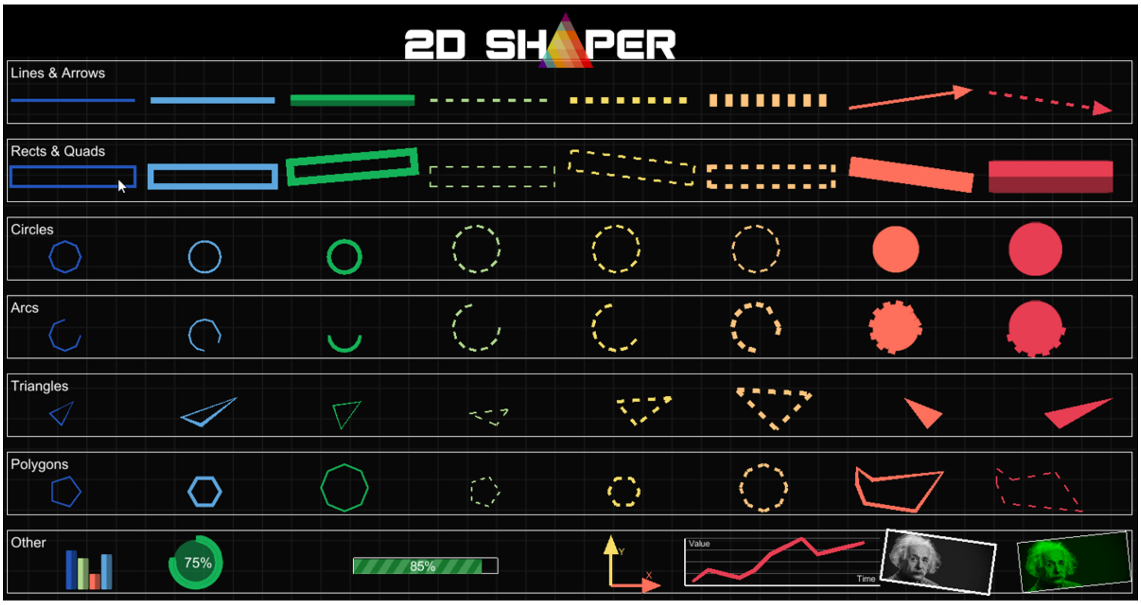
Please note: This package is not a component that can be added to a game object in the inspector. It's a code library you can use from a script (typically from methods like OnGui or OnPostRender). It includes sample scenes showing the most frequent use cases.

Support

If you need any support with the package, please contact: iayucar@simax.es

Examples

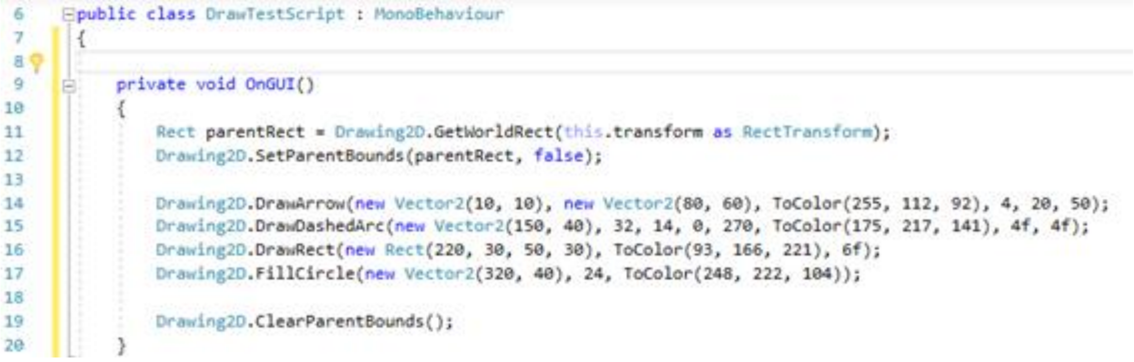
2D Shaper can draw and fill most common 2D shapes:



Sample Code

The following code shows a basic use case in the OnGUI method of a panel, inside a Canvas.

It first sets the bounds of the parent panel where the draw will be done, and then calls several drawing methods:



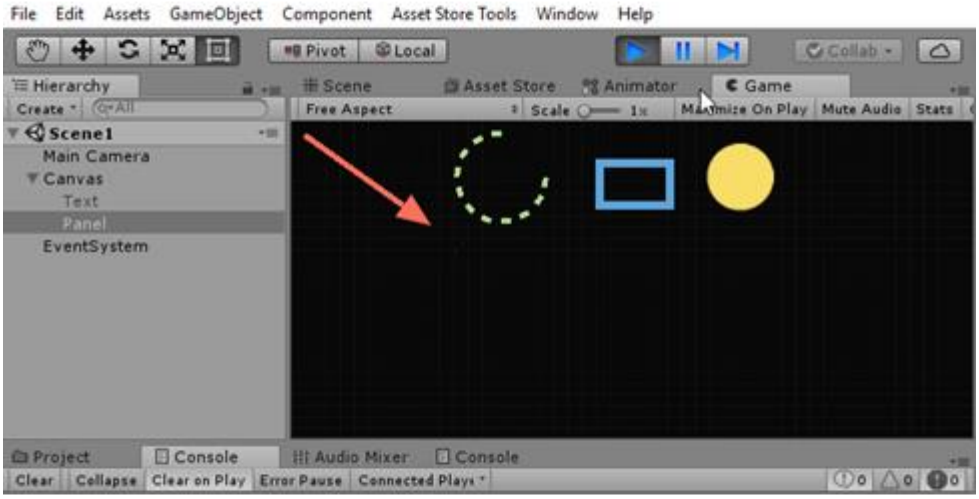
```
public class DrawTestScript : MonoBehaviour
{

    private void OnGUI()
    {
        Rect parentRect = Drawing2D.GetWorldRect(this.transform as RectTransform);
        Drawing2D.SetParentBounds(parentRect, false);

        Drawing2D.DrawArrow(new Vector2(10, 10), new Vector2(80, 60), ToColor(255, 112, 92), 4, 20, 50);
        Drawing2D.DrawDashedArc(new Vector2(150, 40), 32, 14, 0, 270, ToColor(175, 217, 141), 4f, 4f);
        Drawing2D.DrawRect(new Rect(220, 30, 50, 30), ToColor(93, 166, 221), 6f);
        Drawing2D.FillCircle(new Vector2(320, 40), 24, ToColor(248, 222, 104));

        Drawing2D.ClearParentBounds();
    }
```

That code produces the following result:



Appendix 1 – Reference

Namespace: GraphicDNA
Class: Drawing2D

Property ScreenWidth

Screen Width, in pixels. If no custom value is set, unity's Screen.Width will be used as default

⊟ **Syntax**

C#

```
public static float ScreenWidth
```

Property ScreenHeight

Screen Height, in pixels. If no custom value is set, unity's Screen.Height will be used as default

⊟ **Syntax**

C#

```
public static float ScreenHeight
```

Method Dictionary

This is a cache to store circle points already generated, to speed up drawing circles.

⊟ **Syntax**

C#

```
private static readonly Dictionary<int, IList<Vector2>> m_circleCache = new Dictionary<int, I
```

Property ProjectionMatrix

Builds a projection matrix for OpenGL, that sets the 0,0 in the TopLeft corner of the screen, just like in the GUI

⊟ **Syntax**

C#

```
private static Matrix4x4 ProjectionMatrix
```

Property MaterialColorOnly

Returns the default material to fill shapes

⊟ **Syntax**

C#

```
public static Material MaterialColorOnly
```

Property MaterialColorAndTexture

Returns the default material to fill shapes

⊟ **Syntax**

C#

```
public static Material MaterialColorAndTexture
```

Property DefaultDashedLineTexture

Default pixel texture used for filling dashed shapes

Syntax

C#

```
public static Texture2D DefaultDashedLineTexture
```

Property DefaultCircleTexture

Default circle texture used for filling round shapes

Syntax

C#

```
public static Texture2D DefaultCircleTexture
```

Property DefaultTriangleTexture

Default Triangle texture used for filling triangle shapes

Syntax

C#

```
public static Texture2D DefaultTriangleTexture
```

Property DefaultTextStyle

Default text style to render text

Syntax

C#

```
public static GUIStyle DefaultTextStyle
```

Method LoadProjectionMatrix

Loads the appropiate projection matrix

Syntax

C#

```
private static void LoadProjectionMatrix()
```

Method ClearFrameBuffer

Clears the frame buffer window to the specified color

Syntax

C#

```
public static void ClearFrameBuffer(Color pClearColor)
```

Parameters

pClearColor

Method BuildGLVertexBuffer

Transforms a list of 2D points in screen GUI coordinates, to Vector3 points valid for GL rendering with an Orthogonal projection matrix

Syntax

C#

```
private static Vector3[] BuildGLVertexBuffer(IList<Vector2> pPoints)
```

Parameters

pPoints

Return Value

Method GUIPointToGLVertex

⊟ Syntax

C#

```
private static Vector3 GUIPointToGLVertex(Vector2 pGUIPoint)
```

Parameters

pGUIPoint

Return Value

Method GetLineQuad

Returns the vertices of a quad that represents a line with thickness The way GL expects quad vertices is (0,0), (0,1), (1,1), (1,0)

⊟ Syntax

C#

```
private static Vector2[] GetLineQuad(Vector2 pointA, Vector2 pointB, float pLineWidth)
```

Parameters

pointA
Vector in screen coords (absolute coords, parents or groups are ignored in this operation)

pointB
Vector in screen coords (absolute coords, parents or groups are ignored in this operation)

pLineWidth
Line thickness, in pixels

Return Value

Method SetMaterial

Sets the material to be used

⊟ Syntax

C#

```
private static void SetMaterial(Texture2D pOverrideTexture = null)
```

Parameters

pOverrideTexture

Method SetParentBounds

Sets the coordinate system for drawing operations so the (0,0) is the top-left corner of the group. If pClip is enabled, all controls are clipped to the group. Groups CANNOT be nested Please note: Do not use GUI.BeginGroup to achieve this, as that might result in a malfunction of the drawing features

⊟ Syntax

C#

```
public static void SetParentBounds(Rect pRect)
```

Parameters

pRect

Parent rectangle to use as a reference

pClip

True to enable clipping to this rectangle

Method ClearParentBounds

Clears any parent bounds or coordinate system. After a call to this method, clipping will be deactivated and all coordinates will be relative to screen's origin.

Syntax

C#

```csharp
public static void ClearParentBounds()
```

Method GetRectInParent

Gets the rect relative to parent coords

Syntax

C#

```csharp
private static Rect GetRectInParent(Rect pRect)
```

Parameters

pRect

Return Value

Method GetPosInParent

Gets the Pos relative to parent coords

Syntax

C#

```csharp
private static Vector2 GetPosInParent(Vector2 pPt)
```

Parameters

pPt

Return Value

Method GetWorldRect

Returns the world position and size of a UI element (provided its RectTransform) The returned position corresponds to the Top-Left corner of the element.

Syntax

C#

```csharp
public static Rect GetWorldRect(RectTransform pTransform)
```

Parameters

pTransform

RectTransform of the UI Element

Method Rect

// Returns a rectangle to mimic a line with thickness //

Syntax

C#

```
//private static Rect GetLineRect(Vector2 p1, Vector2 p2, float lineWidth, out float distance
        //
```

Parameters

p1
p2
lineWidth
distance
angle

Method DrawPoint

Draws a point

⊟ Syntax

C#

```
public static void DrawPoint(Vector2 pPoint, Color color)
```

Parameters

pPoint
Coordinates of the point (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the point

Method DrawPoint

Draws a point

⊟ Syntax

C#

```
public static void DrawPoint(Vector2 pPoint, Color color, float pSize)
```

Parameters

pPoint
Coordinates of the point (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the point

pSize
Size of the point, in pixels

Method DrawTexture

Draws a texture

⊟ Syntax

C#

```
public static void DrawTexture(Rect pRect, Color color, Texture2D pTexture)
```

Parameters

pRect
Rectangle in the screen to draw the texture (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color that will be multiplied to the texture (white for no effect)

pTexture
Texture to draw

Method DrawLine

Draws a line between two points. GUI.ContentColor is assumed to draw the line

⊟ Syntax

C#

```
public static void DrawLine(Vector2 pointA, Vector2 pointB)
```

Parameters

pointA
First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB
Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

Method DrawLine

Draws a line between two points.

⊟ Syntax

C#

```
public static void DrawLine(Vector2 pointA, Vector2 pointB, Color color)
```

Parameters

pointA
First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB
Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the line

Method DrawLine

Draws a line between two points. GUI.ContentColor is assumed to draw the line

⊟ Syntax

C#

```
public static void DrawLine(Vector2 pointA, Vector2 pointB, float width)
```

Parameters

pointA
First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB
Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

width
Line thickness, in pixels

Method DrawLine

Draws a line between two points.

⊟ Syntax

C#

```
public static void DrawLine(Vector2 pointA, Vector2 pointB, Color color, float width)
```

Parameters

pointA
First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB
Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the line

width
Line thickness, in pixels

Method DrawLine

Draws a line between two points, using a customized base texture to fill the segment

⊟ Syntax

C#

```
public static void DrawLine(Vector2 p1, Vector2 p2, Color color, float lineWidth, Texture2D p
```

Parameters

p1
First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

p2
Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the line

lineWidth
Line thickness, in pixels

pOverrideTexture
Custom texture to use as base fill (null to use default fill)

pTilingMultiplier
Number of repetitions of the texture in U,V (null to use default, with 1 repetition in each direction)

Method DrawLines

Draws lines with thickness == 1, using GUI 2D points

⊟ Syntax

C#

```
public static void DrawLines(Vector2[] pPointPairs, Color pColor)
```

Parameters

pPointPairs
Pairs of points, with the start-end points for each line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pColor
Color of all lines

Method DrawLines

Draws lines with thickness == 1, using OpenGL Vector3 points in homogeneous coords. Please use the BuildGLVertexBuffer method to convert from GUI coordinates to OpenGL homogeneous coordinates

⊟ Syntax

C#

```
public static void DrawLines(Vector3[] vertexBuffer, Color pColor)
```

Parameters

vertexBuffer
Pairs of points in homogeneous coordinates, valid for OpenGL rendering with an Ortho projection matrix

pColor
Color of all lines

Method void

// Welds segment vertices together to fix joints //

⊟ Syntax

C#

```
//private static void FixLineCorners(List<Vector2> vertices)
        //
```

Parameters

vertices

Method DrawLines

Draws lines, using OpenGL Vector3 points in homogeneous coords. Please use the BuildGLVertexBuffer method to convert from GUI coordinates to OpenGL homogeneous coordinates

⊟ Syntax

C#

```
public static void DrawLines(Vector2[] pPointPairs, Color pColor, float lineWidth, Texture2D p
```

Parameters

pPointPairs
Pairs of points, with the start-end points for each line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pColor
Color of all lines

lineWidth
Line thickness, in pixels

pOverrideTexture
Texture to overlay (null to use only color)

pTilingMultiplier
Tiling multiplier or null for default tiling (1, 1)

pFixLineCorners
If true, fixes line corners considering the line to be continuous. Looks better but it's significantly slower (default = true)

Method DrawDashedLine

Draws a dashed line between two points. GUI.ContentColor is assumed to draw the line

⊟ Syntax

C#

```
public static void DrawDashedLine(Vector2 pointA, Vector2 pointB)
```

Parameters

pointA

First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB

Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

Method DrawDashedLine

Draws a dashed line between two points.

⊟ Syntax

C#

```
public static void DrawDashedLine(Vector2 pointA, Vector2 pointB, Color color)
```

Parameters

pointA

First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB

Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the line

Method DrawDashedLine

Draws a dashed line between two points. GUI.ContentColor is assumed to draw the line

⊟ Syntax

C#

```
public static void DrawDashedLine(Vector2 pointA, Vector2 pointB, float width)
```

Parameters

pointA

First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pointB

Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

width

Line thickness, in pixels

Method DrawDashedLine

Draws a dashed line between two points.

⊟ Syntax

C#

```
public static void DrawDashedLine(Vector2 p1, Vector2 p2, Color color, float lineWidth, float
```

Parameters

p1
First point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

p2
Second point of the line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the line

lineWidth
Line thickness, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method void

// Draws lines, using OpenGL Vector3 points in homogeneous coords. Please use the BuildGLVertexBuffer method to convert from GUI // coordinates to OpenGL homogeneous coordinates //

⊟ Syntax

C#

```
//public static void DrawDashedLines(Vector2[] pPointPairs, Color pColor, float lineWidth, boo
        //
```

Parameters

pPointPairs
Pairs of points, with the start-end points for each line (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pColor
Color of all lines

lineWidth
Line thickness, in pixels

pOverrideTexture
Texture to overlay (null to use only color)

pTilingMultiplier
Tiling multiplier or null for default tiling (1, 1)

pFixLineCorners
If true, fixes line corners considering the line to be continuous. Looks better but it's significantly slower (default = true)

Method DrawArrowTip

⊟ Syntax

C#

```
private static void DrawArrowTip(Vector2 p1, Vector2 p2, Color color, float pTipWidth, float p
```

Parameters

p1
p2
color
lineWidth
pTipWidth

pTipLength

Method DrawArrow

Draws an arrow

⊟ Syntax

C#

```
public static void DrawArrow(Vector2 p1, Vector2 p2, Color color, float lineWidth, float pTip
```

Parameters

p1
First point of the arrow (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

p2
Second point of the arrow (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the arrow

lineWidth
Line thickness, in pixels

pTipWidth
Width of the tip, in pixels

pTipLength
Length of the tip, in pixels

Method DrawDashedArrow

Draws an arrow

⊟ Syntax

C#

```
public static void DrawDashedArrow(Vector2 p1, Vector2 p2, Color color, float lineWidth, floa
```

Parameters

p1
First point of the arrow (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

p2
Second point of the arrow (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the arrow

lineWidth
Line thickness, in pixels

pTipWidth
Width of the tip, in pixels

pTipLength
Length of the tip, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method CreateArc

Creates a list of points that builds an arc (plain points, not point pairs)

⊟ Syntax

C#

```
private static List<Vector2> CreateArc(float radius, int sides, float startingAngleDeg, float
```

Parameters

radius

sides

startingAngleDeg

degrees

Return Value

Method DrawArc

Draws a Arc. GUI.ContentColor is assumed to draw the circle

⊟ Syntax

C#

```
public static void DrawArc(Vector2 center, float radius, int sides, float pStartAngleDeg, flo
```

Parameters

center

Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius

Radius of the circle, in Pixels

sides

Number of sides (more sides, more detail and less performance)

pStartAngleDeg

Degrees where arc starts (being 0º == 15h in a clock and 270º == 12h)

pDegrees

Number of degrees to rotate, starting from pStartAngleDeg

Method DrawArc

Draws a Arc.

⊟ Syntax

C#

```
public static void DrawArc(Vector2 center, float radius, int sides, float pStartAngleDeg, flo
```

Parameters

center

Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius

Radius of the circle, in Pixels

sides

Number of sides (more sides, more detail and less performance)

pStartAngleDeg

Degrees where arc starts (being 0º == 15h in a clock and 270º == 12h)

pDegrees

Number of degrees to rotate, starting from pStartAngleDeg

color

Color of the circle

Method DrawArc

Draws a Arc.

Syntax

C#

```
public static void DrawArc(Vector2 center, float radius, int sides, float pStartAngleDeg, floa
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

pStartAngleDeg
Degrees where to start the arc, being 0º = 15h on a clock, 90º = 18h, 180º = 21h, etc

pDegrees
Number of degrees to rotate, starting from pStartAngleDeg

color
Color of the circle

lineWidth
Line thickness, in pixels

Method DrawArc

Draws a Arc.

Syntax

C#

```
public static void DrawArc(Vector2 center, float radius, int sides, float pStartAngleDeg, floa
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

pStartAngleDeg
Degrees where to start the arc, being 0º = 15h on a clock, 90º = 18h, 180º = 21h, etc

pDegrees
Number of degrees to rotate, starting from pStartAngleDeg

color
Color of the circle

lineWidth
Line thickness, in pixels

pLineTexture
Custom base texture to fill the segments (null to use default fill)

Method DrawDashedArc

Draws a dashed Arc.

⊟ Syntax

C#

```
public static void DrawDashedArc(Vector2 center, float radius, int sides, float pStartAngleDe
```

Parameters

center

Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius

Radius of the circle, in Pixels

sides

Number of sides (more sides, more detail and less performance)

pStartAngleDeg

Degrees where to start the arc, being 0º = 15h on a clock, 90º = 18h, 180º = 21h, etc

pDegrees

Number of degrees to rotate, starting from pStartAngleDeg

color

Color of the circle

lineWidth

Line thickness, in pixels

pDashMultiplier

Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method CreateCircle

Returns a plain list of points that build a circle (not point pairs, just points)

⊟ Syntax

C#

```
private static IList<Vector2> CreateCircle(int sides, bool pSaveCache = true, bool pCloseCirc
```

Parameters

sides

pSaveCache

Return Value

Method CreateCircleTexture

⊟ Syntax

C#

```
private static Texture2D CreateCircleTexture(int pTextureWidthHeight)
```

Return Value

Method DrawCircle

Draws a circle. GUI.ContentColor is assumed to draw the circle

⊟ Syntax

C#

```
public static void DrawCircle(Vector2 center, float radius, int sides)
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

Method DrawCircle

Draws a circle.

⊟ Syntax

C#

```
public static void DrawCircle(Vector2 center, float radius, int sides, Color color)
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

color
Color of the circle

Method DrawCircle

Draws a circle.

⊟ Syntax

C#

```
public static void DrawCircle(Vector2 center, float radius, int sides, Color color, float lin
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

color
Color of the circle

lineWidth
Line thickness, in pixels

Method DrawCircle

Draws a circle.

⊟ Syntax

C#

```
public static void DrawCircle(Vector2 center, float radius, int sides, Color color, float lin
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

color
Color of the circle

lineWidth
Line thickness, in pixels

pLineTexture
Custom base texture to fill the segments (null to use default fill)

Method DrawDashedCircle

Draws a circle.

⊟ Syntax

C#

```
public static void DrawDashedCircle(Vector2 center, float radius, int sides, Color color, floa
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

sides
Number of sides (more sides, more detail and less performance)

color
Color of the circle

lineWidth
Line thickness, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method FillCircle

Fills a circle, using GUI.Content color

⊟ Syntax

C#

```
public static void FillCircle(Vector2 center, float radius)
```

Parameters

center

Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

Method FillCircle

Fills a circle

⊟ Syntax

C#

```
public static void FillCircle(Vector2 center, float radius, Color color)
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

color
Color of the circle

Method FillCircle

Fills a circle

⊟ Syntax

C#

```
public static void FillCircle(Vector2 center, float radius, Color color, Texture2D pCircleTex
```

Parameters

center
Center of the circle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius of the circle, in Pixels

color
Color of the circle

pCircleTexture
Custom base texture to perform the fill (must be round texture to keep the circle shape)

Method GetQuadCorners

⊟ Syntax

C#

```
private static Vector2[] GetQuadCorners(Vector2 pCenter, float pWidth, float pHeight, float? p
```

Parameters

pCenter
pWidth
pHeight
pOrientationDeg

Return Value

Method DrawQuad

Fills a quad with an arbitrary orientation

⊟ Syntax

C#

```
public static void DrawQuad(Vector2 pCenter, float pWidth, float pHeight, Color pColor, float
```

Parameters

pCenter
Center of the quad (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pWidth
Width of the quad, in pixels

pHeight
Height of the quad, in pixels

pColor
Color of the quad

pLineWidth
Line thickness, in pixels

pOrientation
Orientation of the quad, in degrees, or null to use default

pOverrideTexture
Fill texture to overlay, or null to use color only

Method DrawDashedQuad

Fills a quad with an arbitrary orientation

⊟ Syntax

C#

```
public static void DrawDashedQuad(Vector2 pCenter, float pWidth, float pHeight, Color pColor,
```

Parameters

pCenter
Center of the quad (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pWidth
Width of the quad, in pixels

pHeight
Height of the quad, in pixels

pColor
Color of the quad

pLineWidth
Line thickness, in pixels

pOrientation
Orientation of the quad, in degrees, or null to use default

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method FillQuad

Fills a quad with an arbitrary orientation

⊟ Syntax

C#

```
public static void FillQuad(Vector2 pCenter, float pWidth, float pHeight, Color pColor, float
```

Parameters

pCenter
Center of the quad (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pWidth
Width of the quad, in pixels

pHeight
Height of the quad, in pixels

pColor
Color of the quad

pOrientation
Orientation of the quad, in degrees, or null to use default

pOverrideTexture
Fill texture to overlay, or null to use color only

pTilingMultiplier
Tiling multiplier or null for default tiling (1, 1)

Method FillQuads

Fills quads that can have any orientation. Each quad is defined by its 4 vertices in the following order: BL, TL, TR, BR

⊟ Syntax

C#

```
public static void FillQuads(Vector2[] pVertices, Color color)
```

Parameters

pVertices
Vertices of the Quad in the following order: BL, TL, TR, BR (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the quad

pOverrideTexture
Texture to overlay, if any

pTilingMultiplier
Tiling multiplier or null for default tiling (1, 1)

Method FillQuads

Fills quads that can have any orientation. Each quad is defined by its 4 vertices in the following order: BL, TL, TR, BR

⊟ Syntax

C#

```
public static void FillQuads(Vector2[] pVertices, Color color, Texture2D pOverrideTexture, Ve
```

Parameters

pVertices
Vertices of the Quad in the following order: BL, TL, TR, BR (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the quad

pOverrideTexture
Texture to overlay, if any

pTilingMultiplier
Tiling multiplier or null for default tiling (1, 1)

Method FillQuads

Fills quads that can have any orientation. Each quad is defined by its 4 vertices in the following order: BL, TL, TR, BR

⊟ Syntax

C#

```
public static void FillQuads(Vector2[] pVertices, Vector2[] pTexCoords, Color color, Texture2
```

Parameters

pVertices
Vertices of the Quad in the following order: BL, TL, TR, BR (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the quad

pOverrideTexture
Texture to overlay, if any

pTilingMultiplier
Tiling multiplier or null for default tiling (1, 1)

Method FillQuadFullScreen

WIP

⊟ Syntax

C#

```
public static void FillQuadFullScreen(Color color)
```

Parameters

previewRect

Method FillQuadFullScreen

WIP

⊟ Syntax

C#

```
public static void FillQuadFullScreen(Color color, Texture2D pOverrideTexture = null, Vector2
```

Parameters

previewRect

Method DrawRect

Draws a rectangle (draws the four sides of it). GUI.ContentColor is assumed

⊟ Syntax

C#

```
public static void DrawRect(Rect pRect)
```

Parameters

pRect

Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

Method DrawRect

Draws a rectangle (draws the four sides of it)

⊟ Syntax

C#

```
public static void DrawRect(Rect pRect, Color color)
```

Parameters

pRect

Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the rectangle.

Method DrawRect

Draws a rectangle (draws the four sides of it)

⊟ Syntax

C#

```
public static void DrawRect(Rect pRect, Color color, float lineWidth)
```

Parameters

pRect

Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the rectangle.

lineWidth

Line thickness, in pixels

Method DrawRect

Draws a rectangle (draws the four sides of it)

⊟ Syntax

C#

```
public static void DrawRect(Rect pRect, Color color, float lineWidth, Texture2D pTexture)
```

Parameters

pRect

Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the rectangle.

lineWidth

Line thickness, in pixels

pTexture

Custom texture to use as base fill (null to use default fill)

Method DrawDashedRect

Draws a rectangle (draws the four sides of it)

⊟ Syntax

C#

```
public static void DrawDashedRect(Rect pRect, Color color, float lineWidth, float pDashMultip
```

Parameters

pRect
Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the rectangle.

lineWidth
Line thickness, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method DrawRects

Draws Rects using OpenGL, ideal for performance critical situations. Please note: Rects drawn using OpenGL are not affected by clipping

⊟ Syntax

C#

```
public static void DrawRects(IList<Rect> pRects, Color pColor, bool pClearFrameBufferColor =
```

Parameters

pRects
List of rectangles to draw (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pColor
Color of all rects

pClearFrameBufferColor
True to clear the whole frame buffer, false otherwise

Method FillRect

Fills a rectange. GUI.ContentColor is assumed

⊟ Syntax

C#

```
public static void FillRect(Rect pRect)
```

Parameters

pRect
Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

Method FillRect

Fills a rectange.

⊟ Syntax

C#

```
public static void FillRect(Rect pRect, Color color)
```

Parameters

pRect
Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the rectangle

Method FillRect

Fills a rectange.

⊟ Syntax

C#

```
public static void FillRect(Rect pRectangle, Color pColor, Texture2D pTexture = null, Vector2
```

Parameters

pRectangle
Rectangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pColor
Color of the rectangle

pTexture
Custom texture to use as base fill (null to use default fill)

pTilingMultiplier
Number of repetitions of the texture in U,V (null to use default (1, 1))

Method CreateTriangleTexture

⊟ Syntax

C#

```
private static Texture2D CreateTriangleTexture(int pTextureWidthHeight)
```

Return Value

Method DrawTriangle

Draws a Triangle, defined by 3 points

⊟ Syntax

C#

```
public static void DrawTriangle(Vector2 pA, Vector2 pB, Vector2 pC, Color color, float lineWi
```

Parameters

pA
First point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pB
Second point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pC
Third point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the triangle

lineWidth
Line thickness, in pixels

Method DrawTriangle

Draws a Triangle, defined by 3 points

⊟ Syntax

C#

```
public static void DrawTriangle(Vector2 pA, Vector2 pB, Vector2 pC, Color color, float lineWi
```

Parameters

pA
First point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pB
Second point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pC
Third point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the triangle

lineWidth
Line thickness, in pixels

pLineTexture
Custom texture to use as base fill (null to use default fill)

Method DrawDashedTriangle

Draws a Triangle, defined by 3 points

⊟ Syntax

C#

```
public static void DrawDashedTriangle(Vector2 pA, Vector2 pB, Vector2 pC, Color color, float
```

Parameters

pA
First point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pB
Second point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pC
Third point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the triangle

lineWidth
Line thickness, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method FillTriangle

Syntax

C#

```
public static void FillTriangle(Vector2 pA, Vector2 pB, Vector2 pC, Color color, Texture2D pO
```

Parameters

pA

First point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pB

Second point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pC

Third point of the triangle (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the Triangle

pOverrideTexture

Texture to overlay (null to use color only)

pTilingMultiplier

Number of repetitions of the texture in U,V (null to use default (1, 1))

Method DrawPolygon

Draws a closed polygon (draws a line between each pair of vertices, in order, and one closing line between the last and the first) Assumes GUI.ContentColor

Syntax

C#

```
public static void DrawPolygon(Vector2[] pVertices)
```

Parameters

pVertices

Array of vertices of the polygon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

Method DrawPolygon

Draws a closed polygon (draws a line between each pair of vertices, in order, and one closing line between the last and the first)

Syntax

C#

```
public static void DrawPolygon(Vector2[] pVertices, Color color)
```

Parameters

pVertices

Array of vertices of the polygon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color

Color of the polygon

Method DrawPolygon

Draws a closed polygon (draws a line between each pair of vertices, in order, and one closing line between the last and the first)

⊟ Syntax

C#

```
public static void DrawPolygon(Vector2[] pVertices, Color color, float lineWidth)
```

Parameters

pVertices
Array of vertices of the polygon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the polygon

lineWidth
Line thickness, in pixels

Method DrawPolygon

Draws a closed polygon (draws a line between each pair of vertices, in order, and one closing line between the last and the first)

⊟ Syntax

C#

```
public static void DrawPolygon(Vector2[] pVertices, Color color, float lineWidth, Texture2D p
```

Parameters

pVertices
Array of vertices of the polygon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the polygon

lineWidth
Line thickness, in pixels

pLineTexture
Custom texture to use as base fill (null to use default fill)

Method DrawPolygon

Draws a closed or open polygon (draws a line between each pair of vertices, in order, and if pLeaveOpen is false, draws a closing line between the last and the first)

⊟ Syntax

C#

```
public static void DrawPolygon(Vector2[] pVertices, Color color, float lineWidth, Texture2D p
```

Parameters

pVertices
Array of vertices of the polygon, plain list, not point pairs (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the polygon

lineWidth
Line thickness, in pixels

pLineTexture
Custom texture to use as base fill (null to use default fill)

pLeaveOpen
True to leave the polygon open. False to close it with a final line between the last and first vertices

pFixLineCorners
If true, fixes line corners considering the line to be continuous. Looks better but it's significantly slower (default = true)

Method DrawDashedPolygon

Draws a closed or open, dashed polygon (draws a dashed line between each pair of vertices, in order, and if pLeaveOpen is false, draws a closing line between the last and the first)

⊟ Syntax

C#

```
public static void DrawDashedPolygon(Vector2[] pVertices, Color color, float lineWidth, bool p
```

Parameters

pVertices
Array of vertices of the polygon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

color
Color of the polygon

lineWidth
Line thickness, in pixels

pLineTexture
Custom texture to use as base fill (null to use default fill)

pLeaveOpen
True to leave the polygon open. False to close it with a final line between the last and first vertices

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method DrawPentagon

Draws a pentagon

⊟ Syntax

C#

```
public static void DrawPentagon(Vector2 center, float radius, Color color, float lineWidth)
```

Parameters

center
Center of the pentagon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

Method DrawPentagon

Draws a pentagon

⊟ Syntax

C#

```
public static void DrawPentagon(Vector2 center, float radius, Color color, float lineWidth, Te
```

Parameters

center
Center of the pentagon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

pLineTexture
Custom texture to use as base fill (null to use default fill)

Method DrawDashedPentagon

Draws a dashed pentagon

⊟ Syntax

C#

```
public static void DrawDashedPentagon(Vector2 center, float radius, Color color, float lineWi
```

Parameters

center
Center of the pentagon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method DrawHexagon

Draws a Hexagon

⊟ Syntax

C#

```
public static void DrawHexagon(Vector2 center, float radius, Color color, float lineWidth)
```

Parameters

center
Center of the Hexagon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

Method DrawHexagon

Draws a Hexagon

⊟ Syntax

C#

```
public static void DrawHexagon(Vector2 center, float radius, Color color, float lineWidth, Te
```

Parameters

center

Center of the Hexagon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius

Radius, in pixels

color

Color of the pentagon

lineWidth

Line thickness, in pixels

pLineTexture

Custom texture to use as base fill (null to use default fill)

Method DrawDashedHexagon

Draws a dashed Hexagon

⊟ Syntax

C#

```
public static void DrawDashedHexagon(Vector2 center, float radius, Color color, float lineWid
```

Parameters

center

Center of the Hexagon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius

Radius, in pixels

color

Color of the pentagon

lineWidth

Line thickness, in pixels

pDashMultiplier

Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method DrawOctogon

Draws a Octogon

⊟ Syntax

C#

```
public static void DrawOctogon(Vector2 center, float radius, Color color, float lineWidth)
```

Parameters

center

Center of the Octogon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

Method DrawOctogon

Draws a Octogon

⊟ Syntax

C#

```
public static void DrawOctogon(Vector2 center, float radius, Color color, float lineWidth, Te
```

Parameters

center
Center of the Octogon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

pLineTexture
Custom texture to use as base fill (null to use default fill)

Method DrawDashedOctogon

Draws a dashed Octogon

⊟ Syntax

C#

```
public static void DrawDashedOctogon(Vector2 center, float radius, Color color, float lineWid
```

Parameters

center
Center of the Octogon (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

radius
Radius, in pixels

color
Color of the pentagon

lineWidth
Line thickness, in pixels

pDashMultiplier
Dash frequency multiplier (min = 0.1, max = 10, default = 3)

Method DrawText

Draws text using BitmapFonts (compatible with OnGUI and OnPostRender methods)

⊟ Syntax

C#

```
public static void DrawText(string pText, Vector2 pPos, int pFontSize, Color pColor, BitmapFo
```

Parameters

pText
Text to draw

pPos
Coords of the TopLeft corner of the text (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pFontSize
Font size

pColor
Color of the text

Method DrawText

Draws text

⊟ Syntax

C#

```
public static void DrawText(string pText, float pX, float pY, int pFontSize, Color pColor, Bi
```

Parameters

pText
Text to draw

pX
X coord of the TopLeft corner of the text (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pY
Y coord of the TopLeft corner of the text (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pFontSize
Font size

pColor
Color of the text

Method DrawText

Draws text

⊟ Syntax

C#

```
public static void DrawText(string pText, float pX, float pY, float pRotationDegs, float pTex
```

Parameters

pText
Text to draw

pX
X coord of the TopLeft corner of the text (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pY
Y coord of the TopLeft corner of the text (use relative coordinates to parent only if SetParentBounds has been used, use absolute screen coords otherwise)

pFontSize

Font size

pColor
Color of the text

Method DrawGUIText

Draws text using GUI Label capabilities and TextStyles (only valid from OnGui methods)

⊟ Syntax

C#

```
public static void DrawGUIText(string pText, Vector2 pPos, int pFontSize, Color pColor)
```

Parameters

pText
Text to draw

pPos
Coords of the TopLeft corner of the text (use relative coordinates to parent only if BeginGroup has been used, use absolute screen coords otherwise)

pFontSize
Font size

pColor
Color of the text

Method DrawGUIText

Draws text using GUI Label capabilities and TextStyles (only valid from OnGui methods)

⊟ Syntax

C#

```
public static void DrawGUIText(string pText, float pX, float pY, int pFontSize, Color pColor)
```

Parameters

pText
Text to draw

pX
X coord of the TopLeft corner of the text (use relative coordinates to parent only if BeginGroup has been used, use absolute screen coords otherwise)

pY
Y coord of the TopLeft corner of the text (use relative coordinates to parent only if BeginGroup has been used, use absolute screen coords otherwise)

pFontSize
Font size

pColor
Color of the text

Method DrawGUIText

Draws text using GUI Label capabilities and TextStyles (only valid from OnGui methods)

⊟ Syntax

C#

```
public static void DrawGUIText(string pText, float pX, float pY, GUIStyle pTextStyle)
```

Parameters

pText

Text to draw

pX
X coord of the TopLeft corner of the text (use relative coordinates to parent only if BeginGroup has been used, use absolute screen coords otherwise)

pY
Y coord of the TopLeft corner of the text (use relative coordinates to parent only if BeginGroup has been used, use absolute screen coords otherwise)

pTextStyle
Custom Text Style