UniCanvas

Supported OS:

- Windows
- Mac
- Linux

Native library files

UniCanvas is linked to following native libraries.

- Cairo graphics library
    - https://www.cairographics.org/

Windows

Native library files are contained in "Assets/UniCanvas/Plugins/Windows" folder.

- Cairo: libcairo.dll

Mac

Native library files are not contained.
Please install them using package manager.

Homebrew :

```
$ brew install cairo
```

Linux

Native library files are not contained.
Please install them using package manager.

Ubuntu or Debian-based Linux :

```
$ sudo apt install libcairo2
```

Other Platforms

Other platforms are not supported.
However you can use UniCanvas using your own dynamic link library files.
Please make sure your libraries are built with "cdecl" calling convention.

Each UniCanvas.dll in "Assets/UniCanvas/Plugins" folder are referencing different native library files.

- Windows: libcairo.*

- Mac: libcairo.2.*
- Linux: libcairo.so.2

Rename your library file to one of above name and set up UniCanvas.dll within "UniCanvas import setting" for your platform on Unity.

Canvas API

Only supported "2d" canvas context.

UniCanvas was implemented with reference to following documents.

- OffscreenCanvas - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/OffscreenCanvas
- CanvasRenderingContext2D - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/CanvasRenderingContext2D
- CanvasGradient - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/CanvasGradient
- CanvasPattern - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/CanvasPattern
- ImageData - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/ImageData
- ImageBitmap - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/ImageBitmap
- TextMetrics - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/TextMetrics
- Path2D - Web APIs | MDN
  - https://developer.mozilla.org/docs/Web/API/Path2D
- Geometry Interfaces Module Level 1
  - https://drafts.fxtf.org/geometry/

OffscreenCanvas

Supported

- width
- height
- getContext() : "2d" context only.
- transferToImageBitmap()

Unsupported

- convertToBlob()

UniCanvas

- Surface : Cairo ImageSurface object reference
- ToTexture2D() : Creates new UnityEngine.Texture2D object
- ApplyTo() : Applies to Texture2D

If an error occurs when calling ApplyTo() method,
set your Texture2D object as follows:

- format = TextureFormat.BGRA32
- width = canvas.width
- height = canvas.height

CanvasRenderingContext2D

Supported

- canvas
- currentTransform
- fillStyle
- filter
- font
- globalAlpha
- globalCompositeOperation
- imageSmoothingEnabled
- imageSmoothingQuality
- lineCap
- lineDashOffset
- lineJoin
- lineWidth
- miterLimit
- shadowBlur
- shadowColor
- shadowOffsetX
- shadowOffsetY
- strokeStyle
- textAlign
- textBaseline
- arc()
- arcTo()
- beginPath()
- bezierCurveTo()
- clearRect()
- clip()
- closePath()
- createImageData()
- createLinearGradient()
- createPattern()
- createRadialGradient()

- drawImage()
- ellipse()
- fill()
- fillRect()
- fillText()
- getImageData()
- getLineDash()
- isPointInPath()
- isPointInStroke()
- lineTo()
- measureText()
- moveTo()
- putImageData()
- quadraticCurveTo()
- rect()
- resetTransform()
- restore()
- rotate()
- save()
- scale()
- setLineDash()
- setTransform()
- stroke()
- strokeRect()
- strokeText()
- transform()
- translate()

Unsupported

- direction
- addHitRegion()
- clearHitRegions()
- drawFocusIfNeeded()
- drawWidgetAsOnScreen()
- drawWindow()
- removeHitRegion()
- scrollPathIntoView()

UniCanvas

- Context : Cairo Context object reference
- createPattern(Texture2D) : Creates canvas pattern from UnityEngine.Texture2D object
- drawImage(Texture2D) : Draws image from UnityEngine.Texture2D object

CanvasGradient

Supported

- addColorStop()

CanvasPattern

Supported

- setTransform()

ImageData

Supported

- new ImageData(array, width, height);
- new ImageData(width, height);
- data
- height
- width

ImageBitmap

Supported

- width
- height
- close()

TextMetrics

Supported

- width

Unsupported

- actualBoundingBoxLeft
- actualBoundingBoxRight
- fontBoundingBoxAscent
- fontBoundingBoxDescent
- actualBoundingBoxAscent
- actualBoundingBoxDescent
- emHeightAscent
- emHeightDescent
- hangingBaseline
- alphabeticBaseline
- ideographicBaseline

Path2D

Supported

- new Path2D()
- new Path2D(path)
- addPath()
- closePath()
- moveTo()
- lineTo()

- bezierCurveTo()
- quadraticCurveTo()
- arc()
- arcTo()
- ellipse()
- rect()

Unsupported

- new Path2D(d) : constructor with SVG path data argument

SVG support

You can draw SVG images using UniCanvas.Rsvg.

- https://bitbucket.org/uniuniworks/unicanvas.rsvg/

UniCanvas.Rsvg uses librsvg library internally.
It is licensed under GPL, please be careful it's usage.

Web compatibilities

You can use more JavaScript code compatibilities.
Please add following line to your MonoBehaviour based .cs file:

```
using UniCanvas.Web;
```

Rewrite your class to inherit from WebMonoBehaviour instead of MonoBehaviour.

```
public class MyClass : WebMonoBehaviour
```

Then you can use more web compatible functions.

For example:

- console object
  - console.log() etc.
- Math object
  - Math.PI, Math.sin() etc.
- Timer functions
  - setTimeout(), clearTimeout()
  - setInterval(), clearInterval()
  - requestAnimationFrame(), cancelAnimationFrame()
- Date time functions
  - Date object
  - performance.now()
- String converter functions

    - atob(), btoa()

You can copy and paste from more types of JavaScript code by using above functions.

Planning for the next version

- Add FreeType library (for emoji support)
- Add SVG support with svgren library
- WebGL, Android, iOS build support