

Basic QuickChart.io URL Generator

Basic QuickChart.io URL Generator es una herramienta simple pero poderosa que te permite generar fácilmente URLs para crear gráficos personalizados utilizando QuickChart.io.

A diferencia de la instalación de la librería QuickChart.io, esta herramienta proporciona una solución rápida y sin complicaciones para crear URLs de gráficos simples sin la necesidad de instalar librerías externas. Con esta herramienta, puedes crear gráficos de barras, gráficos Lineales, gráficos de Radar, entre otros, simplemente proporcionando los datos y ajustando las opciones según tus necesidades. Ya sea que necesites visualizar datos estadísticos, mostrar tendencias o crear informes visuales, Basic QuickChart.io URL Generator te ofrece una manera rápida y eficiente de crear gráficos personalizados sin la necesidad de instalar otras dependencias.

¡Optimiza tu flujo de trabajo y agiliza la creación de gráficos con esta práctica herramienta!

Contenido

Basic QuickChart.io URL Generator	1
Componentes	2
Instrucciones de Uso.....	4
Requisitos.....	6
Notas de la Versión	6
Licencia	6
Créditos.....	6
Contacto y soporte.....	6

Componentes

La herramienta se compone de 4 scripts:

ChartStyle: Contiene información sobre el estilo de una gráfica, como colores, tamaño, visualización de los ejes.

Atributos;

- private ChartType type;
 - Bar
 - Line
 - Radar
 - Pie
 - Doughnut
 - PolarArea
- private Color categoryLabelsFontColor
- private float categoryLabelsFontSize
- private Color fontXColor
- private float fontXSize
- private Color fontYColor
- private float fontYSize
- private bool displayAxeX
- private bool displayAxeY
- private int width
- private int height

CharDataset: Contiene información del conjunto de datos de la gráfica como el nombre, los datos, y colores.

Atributos:

- string Label
- float[] Data
- bool Fill
- Color BorderColor
- Color[] BackgroundColor

Chart: Representa una Gráfica, necesita las etiquetas de categoría de la gráfica (las que suelen ir en el eje x) un Estilo de gráfica y al menos un Conjunto de datos, también contiene la versión pero esto raramente necesita ser modificado.

Atributos:

- `string[] CategoryLabelsX`
- `ChartDataset[] Datasets`
- `ChartStyle Style`
- `string version`

Funciones:

- `string ChartURL()`: Convierte todos los datos de una gráfica en una string que representa una URL que permite obtener la gráfica.
- `ColorsToStringFormatRGBA(Color[] colors)`: convierte un conjunto de colores en un string con el formato `['rgba(0,0,0,0)','rgba(0,0,0,0)']`
- `ColorToStringFormatRGBA(Color color)`: convierte un color en un string con el formato `'rgba(0,0,0,0)'`

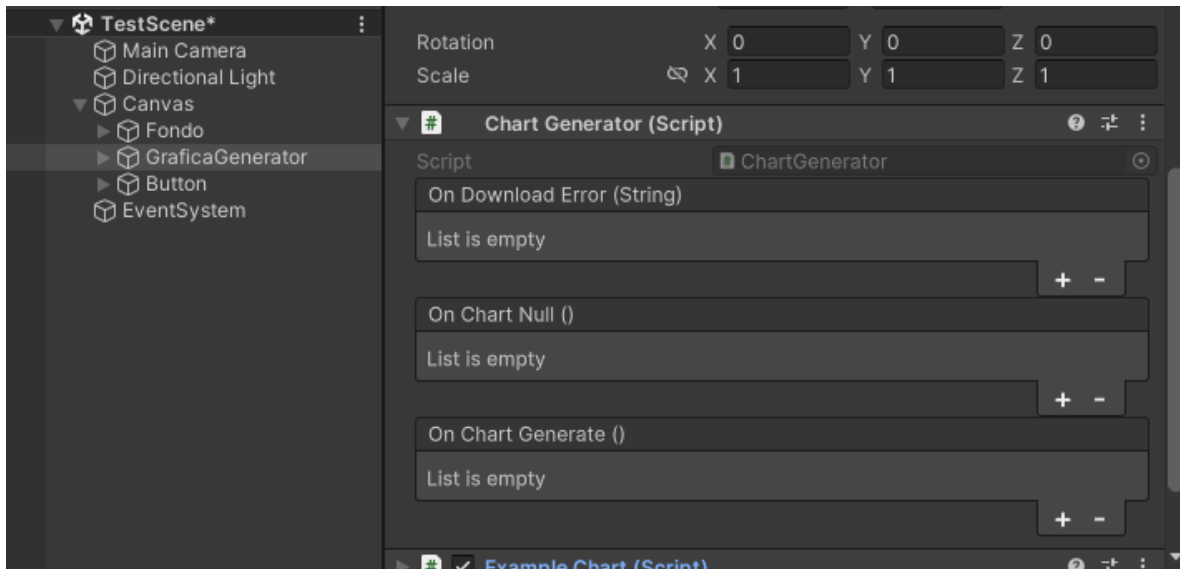
ChartGenerator: Se encarga de generar la imagen.

Funciones:

- `GenerateChartImage(Chart chart, Action<Texture2D> onTextureReady)`: Envía la información de la url a una co-rutina.
- `IEnumerator DownloadChartImage(string url, Action<Texture2D> onTextureReady)`: co-rutina que se encarga de enviar la url y esperar la imagen, una vez la imagen ha sido descargada exitosamente ejecuta la función que haya sido pasada como parámetro.

Instrucciones de Uso

Primero es necesario agregar ChartGenerator a nuestra escena.



En el mismo GameObject Agregamos nuestro script donde crearemos nuestra grafica y la instanciaremos con su información.

```
public void CreateChart()  
{  
    //Create a Chart for the amount of fruit that i have seen in my dreams in the week  
    // Create ChartStyle whit editor Variables  
    ChartStyle chartStyle = new ChartStyle(  
        chartType,  
        colorCategoryLabels,  
        sizeCategoryLabels,  
        colorfontX,  
        sizefontX,  
        colorfontY,  
        sizefontY,  
        displayfontX,  
        displayfontY,  
        width,  
        height  
    );  
  
    //Category Labels  
    string[] categoryLabels = new string[]{"Lunes","Martes","Miercoles","Jueves","Viernes"};  
    // data for datasets  
    float[] data1 = new float[] { 50f, 60f, 70f, 180f, 190f };  
    float[] data2 = new float[] { 100f, 200f, 300f, 400f, 500f };  
    float[] data3 = new float[] { 150f, 130f, 200f, 600f, 450f };  
    //datasets for chart  
    ChartDataset dataset1 = new ChartDataset("Sandias", data1, fillDataset1, colorBorderDataset1, colorsBackgDataset);  
    ChartDataset dataset2 = new ChartDataset("Melones", data2, fillDataset2, colorBorderDataset2, colorsBackgDataset);  
    ChartDataset dataset3 = new ChartDataset("Guayabas", data3, fillDataset3, colorBorderDataset3, colorsBackgDataset);  
    // Create chart  
    Chart chart = new Chart(categoryLabels, new ChartDataset[] { dataset1,dataset2 }, chartStyle);  
}
```

Después llamaremos a la función que se encuentra en ChartGenerator GenerateChartImage pasando como parámetros la gráfica y una función que reciba un Texture2D como parámetro, que se ejecutará cuando la imagen este lista.

```
// Call function to generate image from url  
GetComponent<ChartGenerator>().GenerateChartImage(chart, ChartTextureReady);
```

```
private void ChartTextureReady(Texture2D chartTexture){  
    OnChartTextureReady?.Invoke(chartTexture);  
}
```

Requisitos

Tener los 4 Scripts en el proyecto.

Notas de la Versión

Versión 1.0: Permite crear algunas graficas como: Barras, Línea, Radar, Pie, Dona, Área polar.

Licencia

GNU: Esta herramienta, así como cualquier modificación debe ser libre de uso para el público, puede ser usada en productos de carácter comercial siempre y cuando este no sea parte integral o comprenda la mayor parte del proyecto.

Créditos

QuickChart.io

Programador: Gustavo Adolfo Hernandez Elizondo.

Contacto y soporte

Correo: tavo98a@gmail.com