

## **UT.8 Generación de GUI con editores Visuales. Elementos Avanzados. MeshPro – Font Asset Creator.**

## Contenidos

**En esta Unidad de Trabajo vamos a tratar los siguientes puntos:**

- Introducción. *Diapositivas 3.*
- Componentes MeshPro – Font Asset Creator. *Diapositivas 4-*
- **Vídeos:**
  - Font Asset Creator.
  - ScrTextMeshPro.
- **Práctica:**
  - Práctica TextMeshPro.

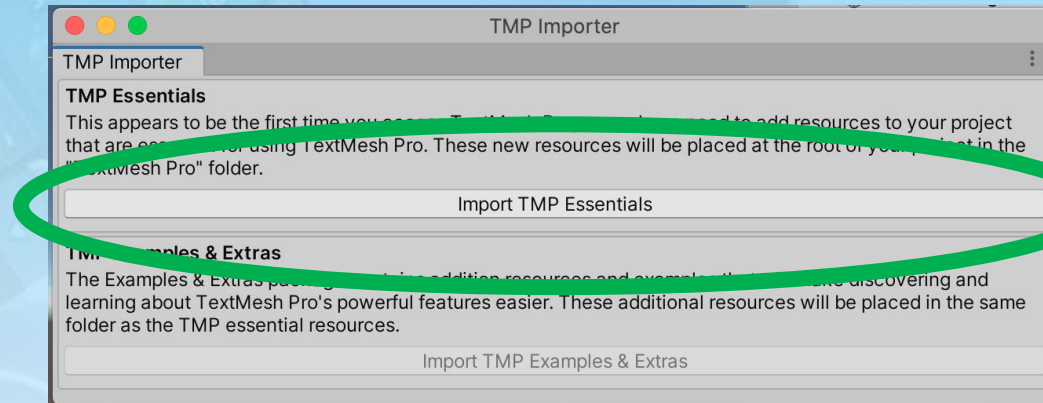
## Introducción

Una vez vistos los parámetros del componente MeshPro, nos cabe hacernos la pregunta, ¿podremos hacer nuestros tipos de fuentes, personalizando el color, sombras, iluminación, etc?.

La respuesta es, sí, TextMeshPro nos permite generar nuestras fuentes para trabajar sobre ellas. Para poder hacer esto, nos proporciona la herramienta **Font Asset Creator**.

A lo largo de esta presentación, vamos a trabajar sobre ella y ver diferentes propiedades que hacen de ella, uno de los mejores componentes de la UI de Unity.

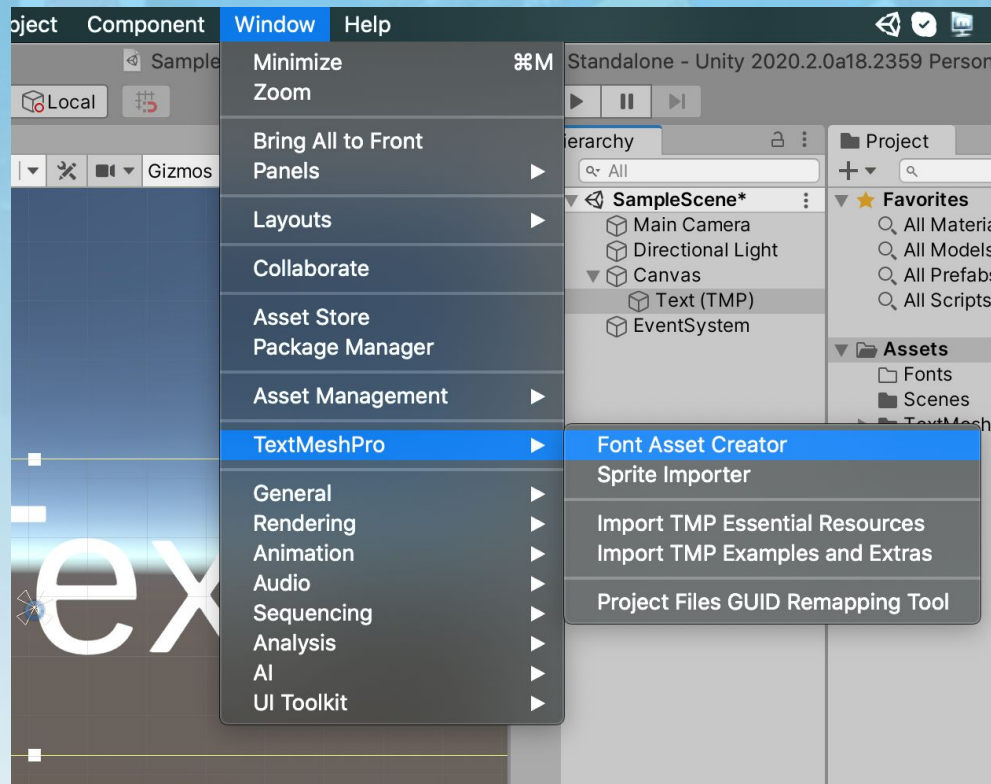
Para empezar a crear nuestras fuentes personalizadas con el componente TextMeshPro, obviamente, lo primero que debemos de hacer es cargar la herramienta TextMeshPro, de la forma que se indica en la presentación que precede a esta.



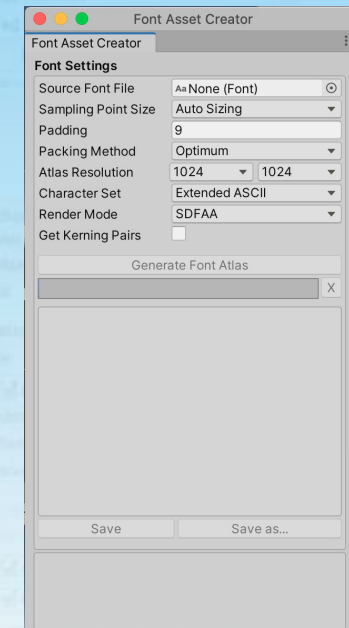
No nos hace falta importar los “TMP Examples & Extras”.



Lo siguiente que vamos a hacer es sacar la herramienta que nos proporciona TextMeshPro y que se denomina **Font Asset Creator**.



Window → TextMeshPro → Font Asset Creator

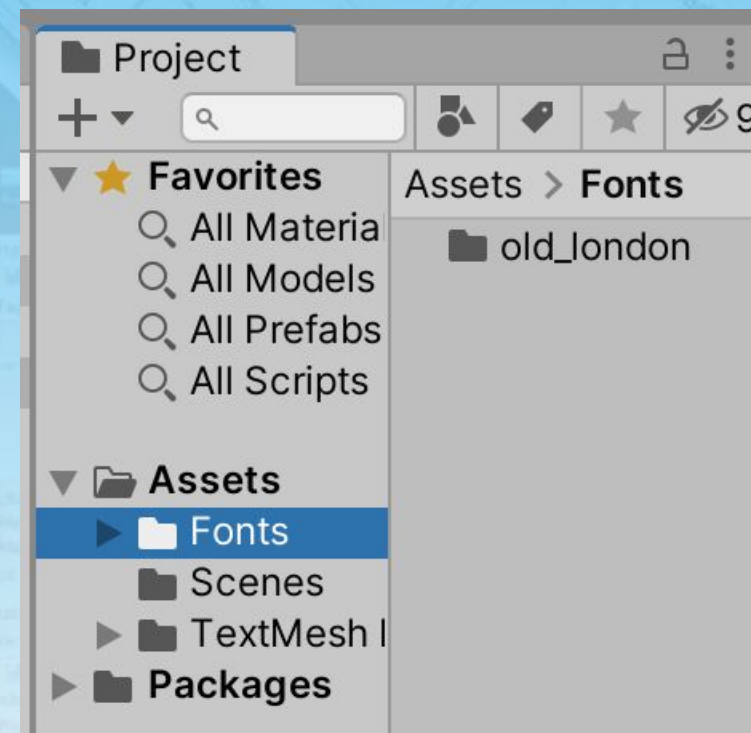
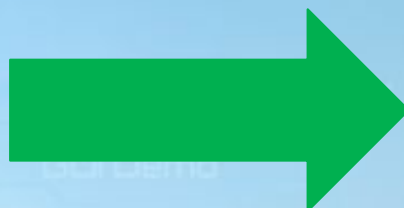
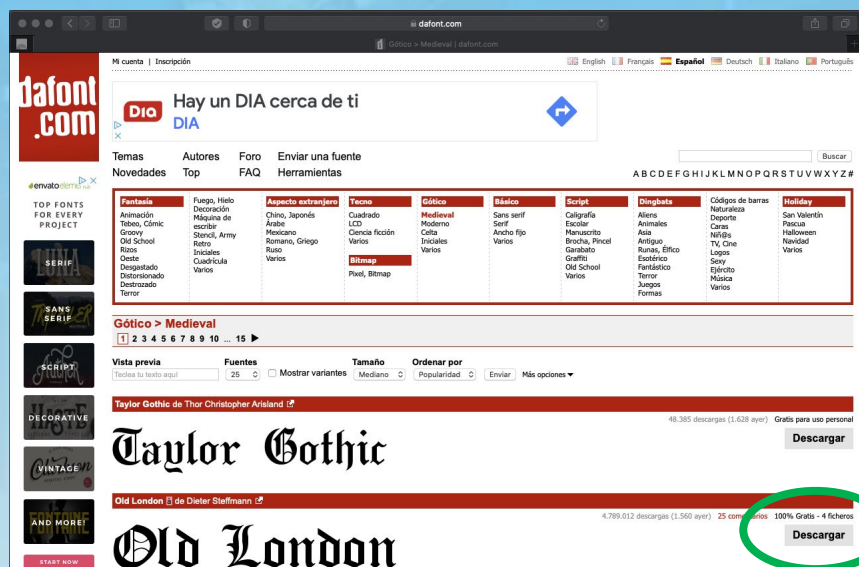


Para trabajar con esta herramienta nos hace falta una fuente base, que podemos crear o descargar desde diferentes páginas web, como:

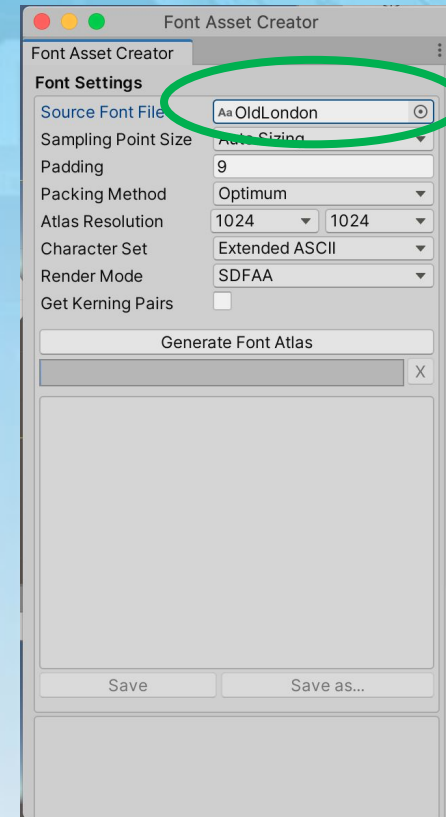
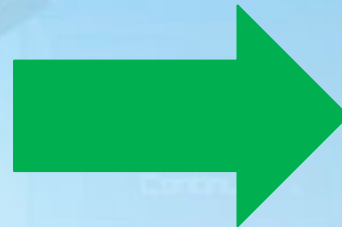
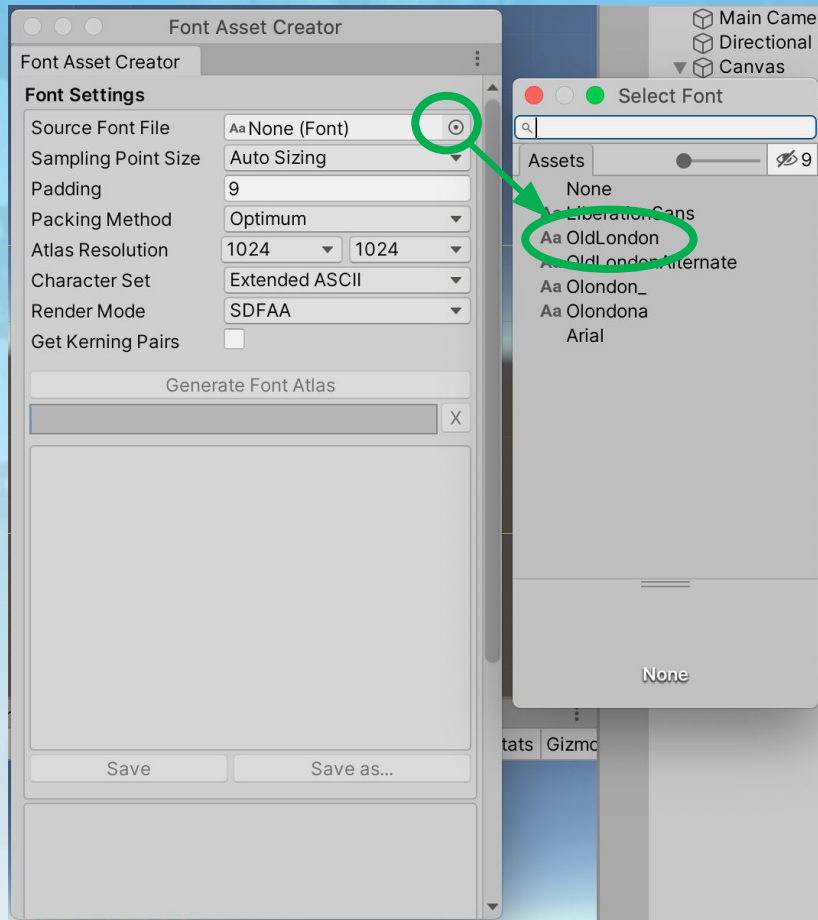
<https://fonts.google.com>

[www.dafont.com](http://www.dafont.com)

En este caso, vamos a ir a la página [www.dafont.com](http://www.dafont.com) y vamos a descargarnos una fuente, que importamos a nuestro proyecto dentro de una carpeta denominada Fonts.



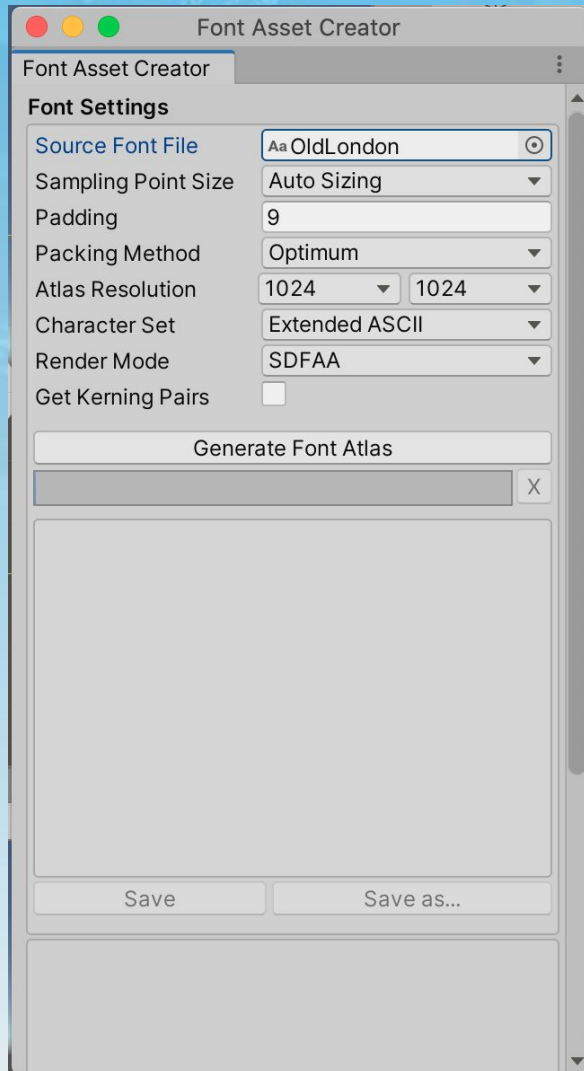
- Ya tenemos todo lo que necesitamos para crear nuestra fuente personalizada. Vamos a la ventana de Font Asset Create y cargamos la fuente que nos hemos descargado.





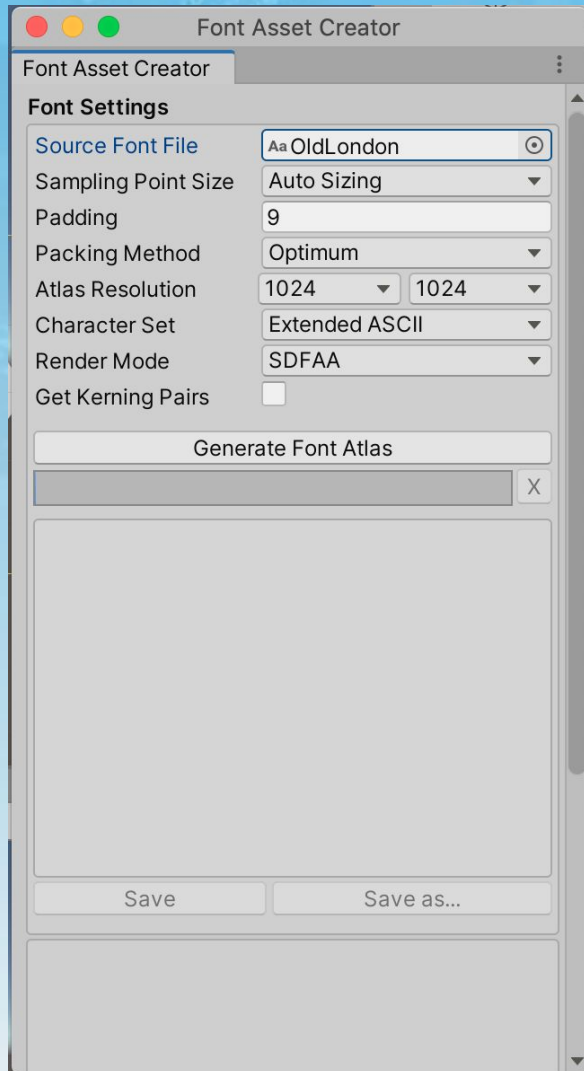
Nos encontramos los siguientes parámetros:

- **Source Font File:** Fichero con la fuente que nos hemos descargado de internet.
- **Sampling Point Size:** Indica el tamaño de cada uno de los caracteres de la fuente.
- **Padding:** Lo normal es tenerlo a 5. Esta variable está relacionada con la de **Atlas Resolution**. A más resolución, más valor habría que poner en este parámetro. Indica el porcentaje de relleno de la fuente. Si vamos a trabajar utilizando sombras o efectos como el underline, lo suyo es ponerlo a 9.
- **Packing Method:** Dos opciones, **Optimum** y **Fast**. la opción de Optimum, nos da más calidad, pero también hace la fuente más pesada.



Nos encontramos los siguientes parámetros:

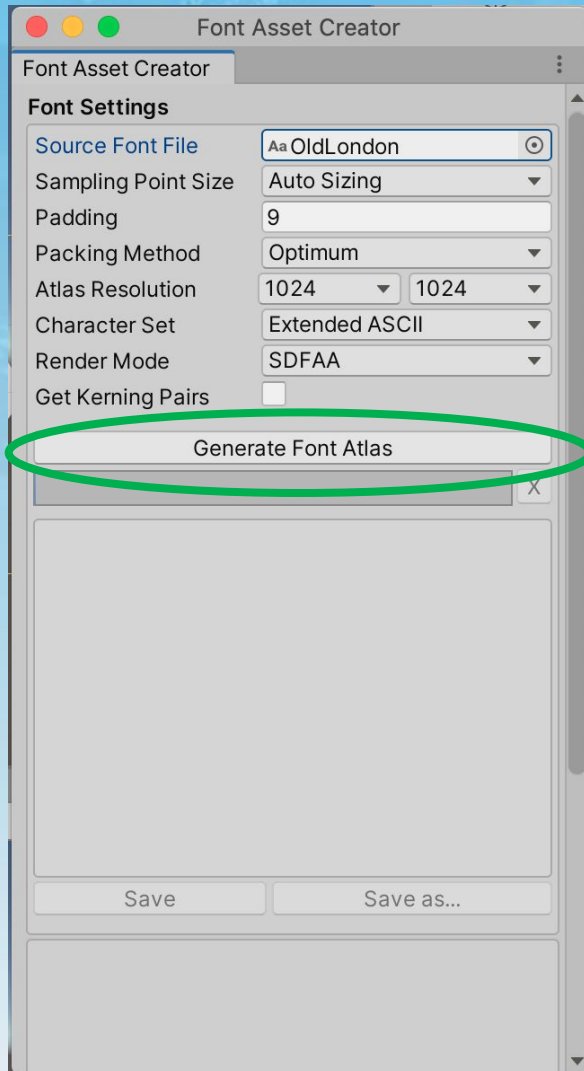
- **Atlas Resolution:** Resolución del texto. Lo normal es que esté a 512x512. Más resolución, mejor calidad y más pesado.
- **Character Set:** Por defecto aparece con ASCII. Si la fuente que nos hemos descargado, tiene letras con acentos o caracteres especiales, debemos de utilizar la opción Extended ASCII, para que nos los añada. Tenemos otras opciones que nos permiten cosas como añadir nosotros los caracteres especiales que nos hagan falta.
- **Render Mode:** La forma en la que se va a renderizar el texto. Recordar que el TextMeshPro, trabaja con un tipo de renderizado que nos proporciona más calidad que el GameObject Text de Unity. Tenemos desde tipos de renderizado más suaves y con menos calidad hasta algunos bastante costosos.



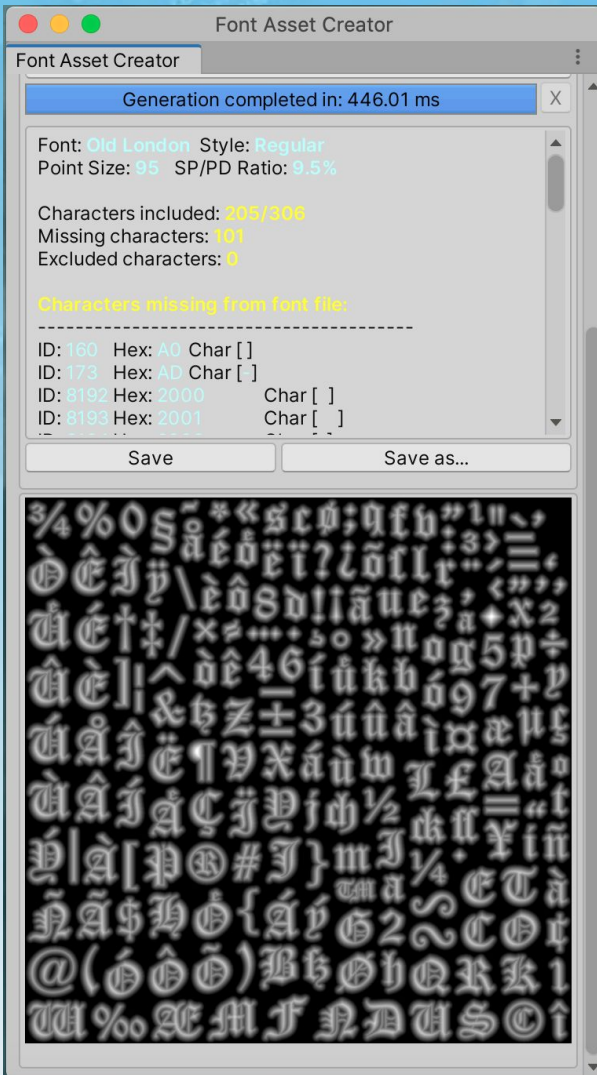
Nos encontramos los siguientes parámetros:

- **Get Kerning Pairs:** Si activamos esta opción vamos a trabajar sobre parejas de caracteres.

Una vez configurados estos parámetros, debemos pulsar sobre el botón **Generate Font Atlas**, lo que generará nuestro atlas de caracteres.



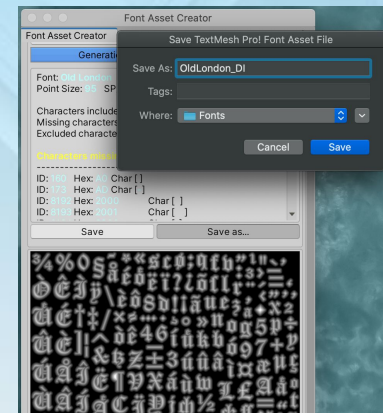




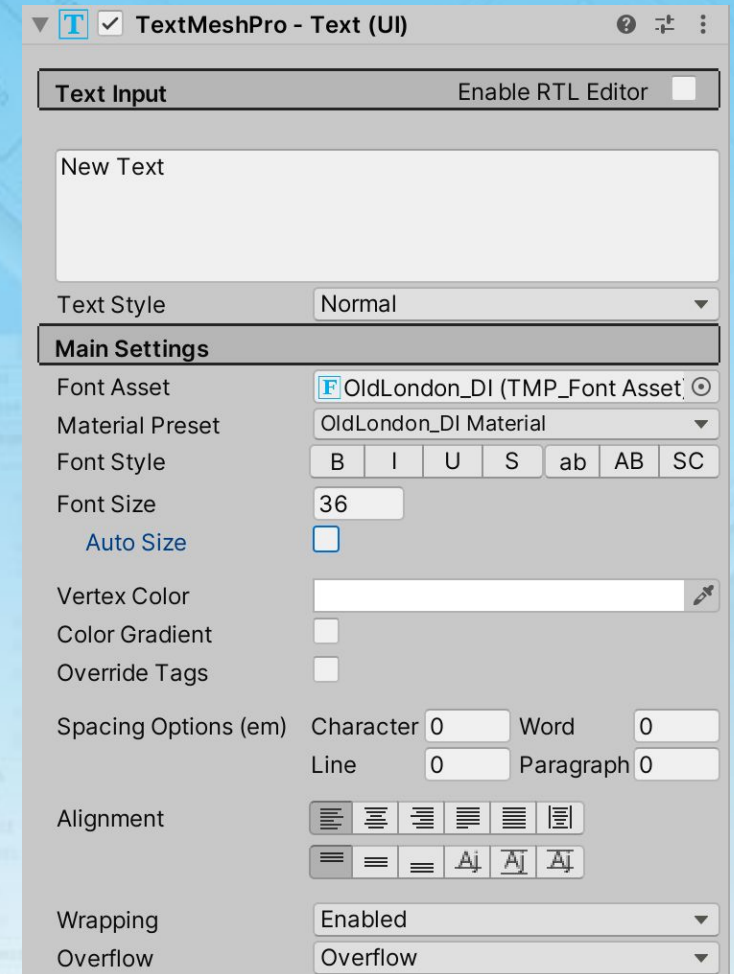
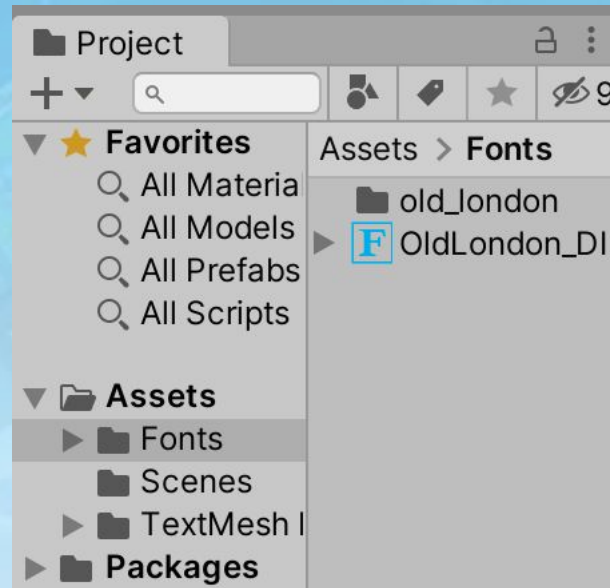
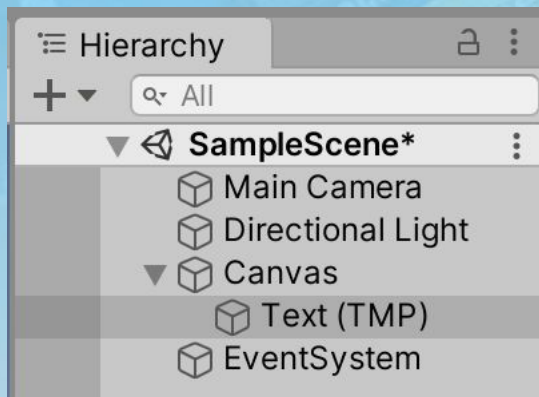
Por último, solo nos resta por hacer, pulsar sobre el botón **Save** o **Save As...**, para guardar la fuente, modificando directamente la de Old London (save) o generando una nueva a partir de Old London (Save As...).

En nuestro ejemplo, vamos a crear una nueva Fuente denominada **OldLondon\_DI**.

A partir de este momento, podremos utilizar esta fuente para cualquiera de nuestros GameObject de tipo TextMeshPro



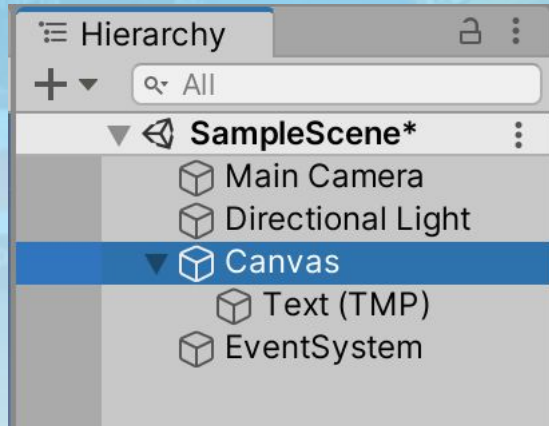




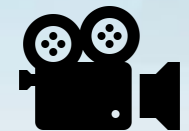
Ver el vídeo FontAssetCreator.mp4

## Introducción

Ya lo único que queda, es poner los parámetros vistos en la presentación precedente a esta, y darle la apariencia deseada. Estos cambios se quedan asociados a la fuente, con lo que cada vez que la utilicemos, tendrá los mismos valores en los parámetros y si hacemos algún cambio, afectará a todos los GameObject's de tipo MeshPro que utilicen nuestra fuente.



Desarrollo  
de Interfaces



[Ver el vídeo FontAssetCreator.mp4](#)

Por último y para terminar esta U.T., decir que para acceder a los parámetros de un componente TextMeshPro desde un script, se debe de hacer uso de la **librería TMPro**, y acceder al **componente TMProUGUI**.

```

1  //Author: Raquel Rojo y Mario Santos
2  //Unity Version: 2020.2.0a18.
3  //Description: working with TextMeshPro.
4  //Cycle: Desarrollo de Interfaces.
5  //Module: Desarrollo de Aplicaciones Multiplataforma.
6  //Date: January 23, 2021.
7  using System.Collections;
8  using System.Collections.Generic;
9  using UnityEngine;
10 //Librería necesaria para poder trabajar con TextMeshPro.
11 using TMPro;
12
13 public class ScrTextMeshPro : MonoBehaviour
14 {
15     private Color miColor;
16     private float transparencia;
17
18     private void Start()
19     {
20         miColor = GameObject.Find("Text (TMP)").GetComponent<TextMeshProUGUI>().color;
21         transparencia = 0.0f;
22     }
23     // Update is called once per frame
24     void Update()
25     {
26         miColor = new Color(miColor.r, miColor.g, miColor.b, transparencia);
27         GameObject.Find("Text (TMP)").GetComponent<TextMeshProUGUI>().color = miColor;
28         if(transparencia <= 1)
29             transparencia += 0.001f;
30     }
31 }

```



**Ver el vídeo ScrTextMeshPro.mp4**

Profesores: Raquel Rojo y Mario Santos.