Performant Localization System

Performant Localization System (PLS) permite manejar las traducciones de tu juego, pudiendo elegir entre optimizar el uso de CPU o el uso de memoria.

Las traducciones se leen de uno o varios archivos `.csv`, según cómo se configure. Puedes subdividir estos archivos por `Zonas` (utilizando una columna especial dentro del mismo) a fin de mejorar el rendimiento.

Features

- Soporta **uno o varios** archivos de traducción (formato CSV).
- **Compatible con Excel** (usando formato CSV).
- Permite elegir entre **optimizar uso de CPU** u **optimizar uso de Memoria**.
- Pensado para mantener el código **limpio y legible**.
- Integración con **TextMesh Pro**.
- Olvidate de las horribles traducciones usando claves. ¡Traduce directamente sobre texto en tu idioma nativo!

Carpetas

Performant Localization System debería encontrarse en:

```
Assets/Plugins/Performant Localization System
```

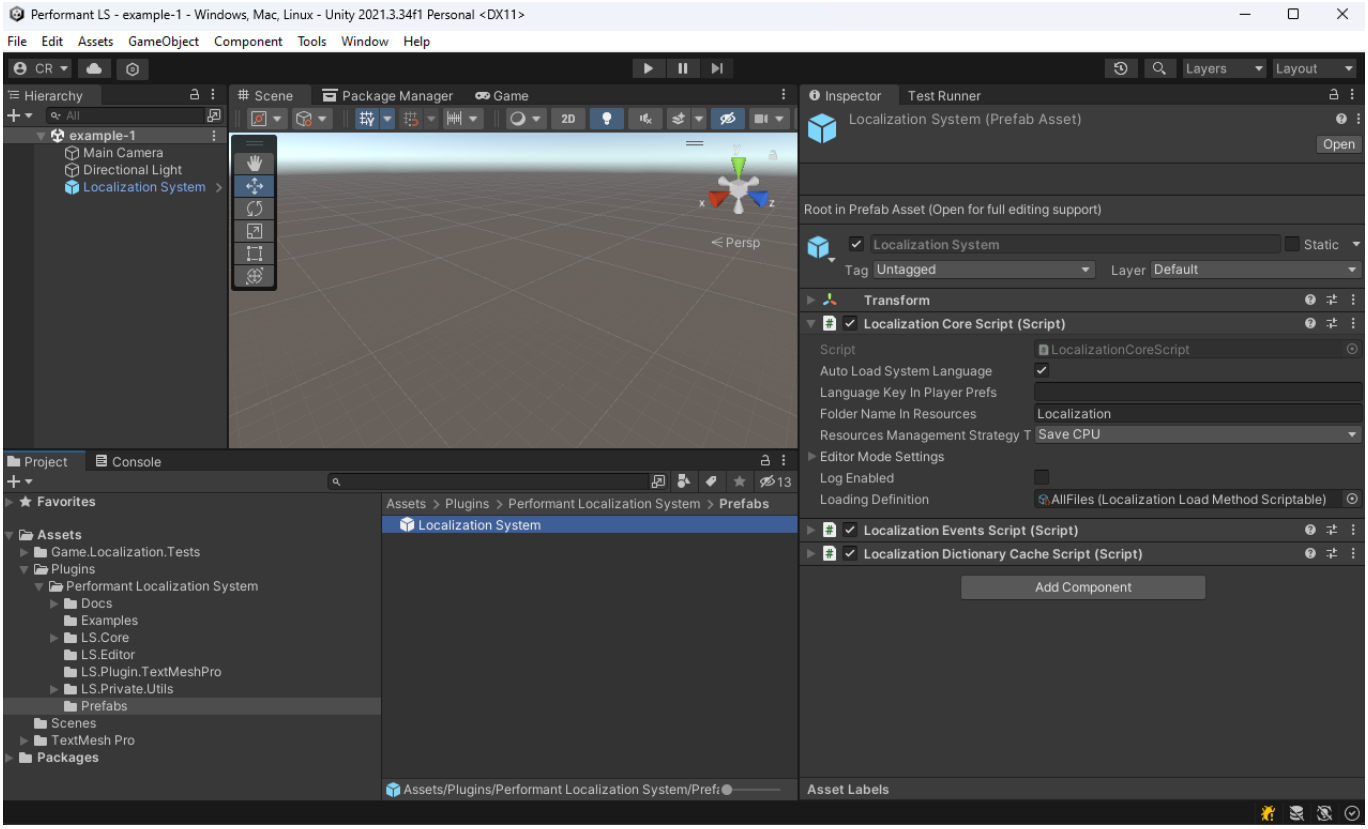
Dentro, tenemos la siguiente estructura:

- **Docs**: Documentación del asset.
- **Examples**: Escenas con ejemplos.
- LS.Core: Código principal del plugin. Se recomienda no modificar.
- LS.Editor: Código para el editor de Unity. Se recomienda no modificar.
- **LS.Plugin.TextMeshPro**: Scripts adicionales para facilitar la traducción con TextMesh Pro.
- LS.Private.Utils: Código de soporte para el plugin. Se recomienda no modificar.
- **LS.Scripts**: Scripts para que utilices en el juego.
- **Prefabs**: Objetos útiles para facilitar la integración.

Getting Started

1. Agregar el prefab `Localization System` a la escena.

Arrastra el `Prefab` llamado `Localization System` dentro de la escena.



Este prefab incluye tres scripts:

- `Localization Core Script`: Script principal. En él se encuentran todas las funciones de traducción.
- `Localization Events Script`: Proveedor de Eventos. Este Script ofrece un evento al que puede subscribirse para detectar cuando el lenguaje cambia.
- `Localization Dictionary Cache Script`: Almacén en memoria de términos cargados.

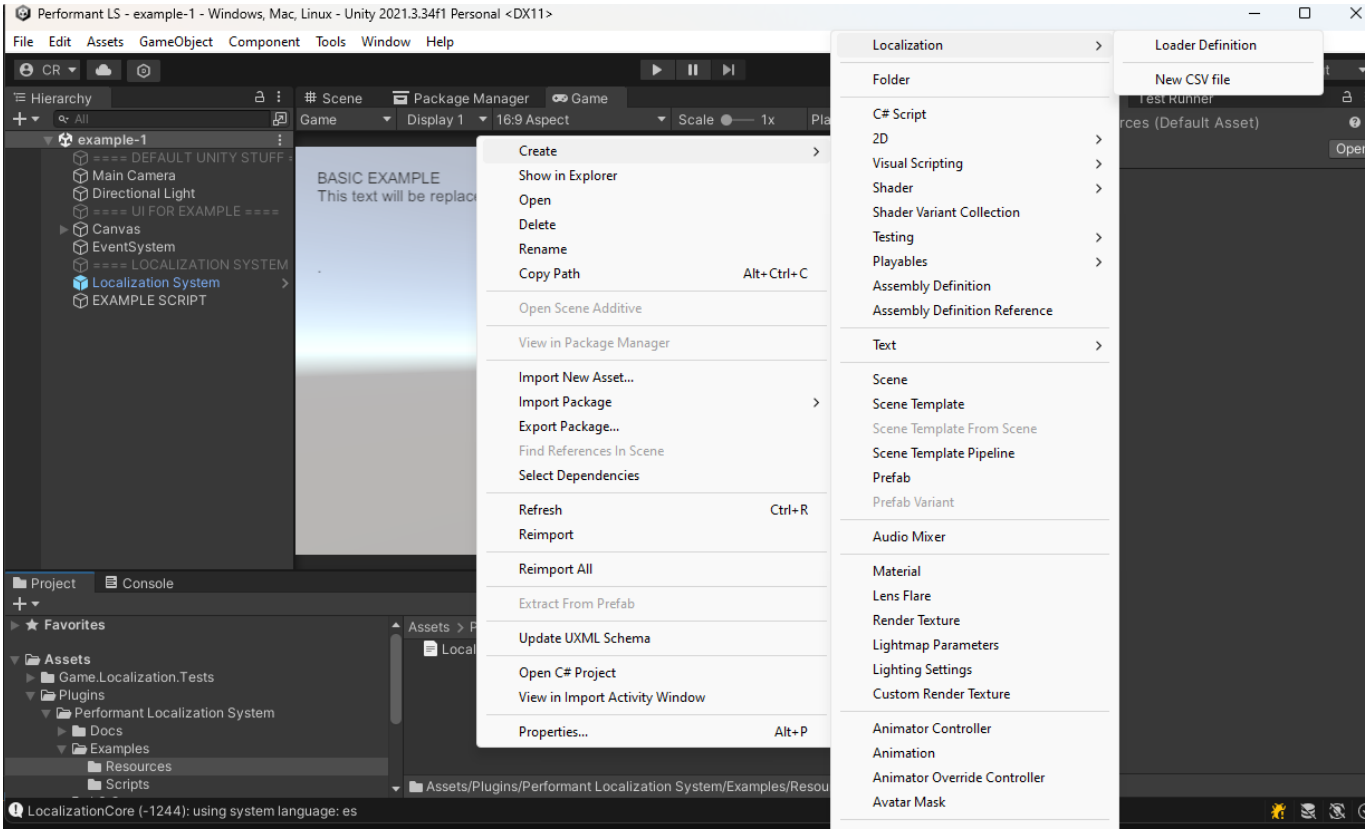
---

2. Crear archivo o archivos de traducción

Genere el o los archivos de traducción que necesite. Es recomendable que se encuentren en alguna subcarpeta de `Resources`.

Para crear un archivo nuevo, simplemente vaya a una carpeta, y haga:

```
Right click > Create > Localization > New CSV file
```



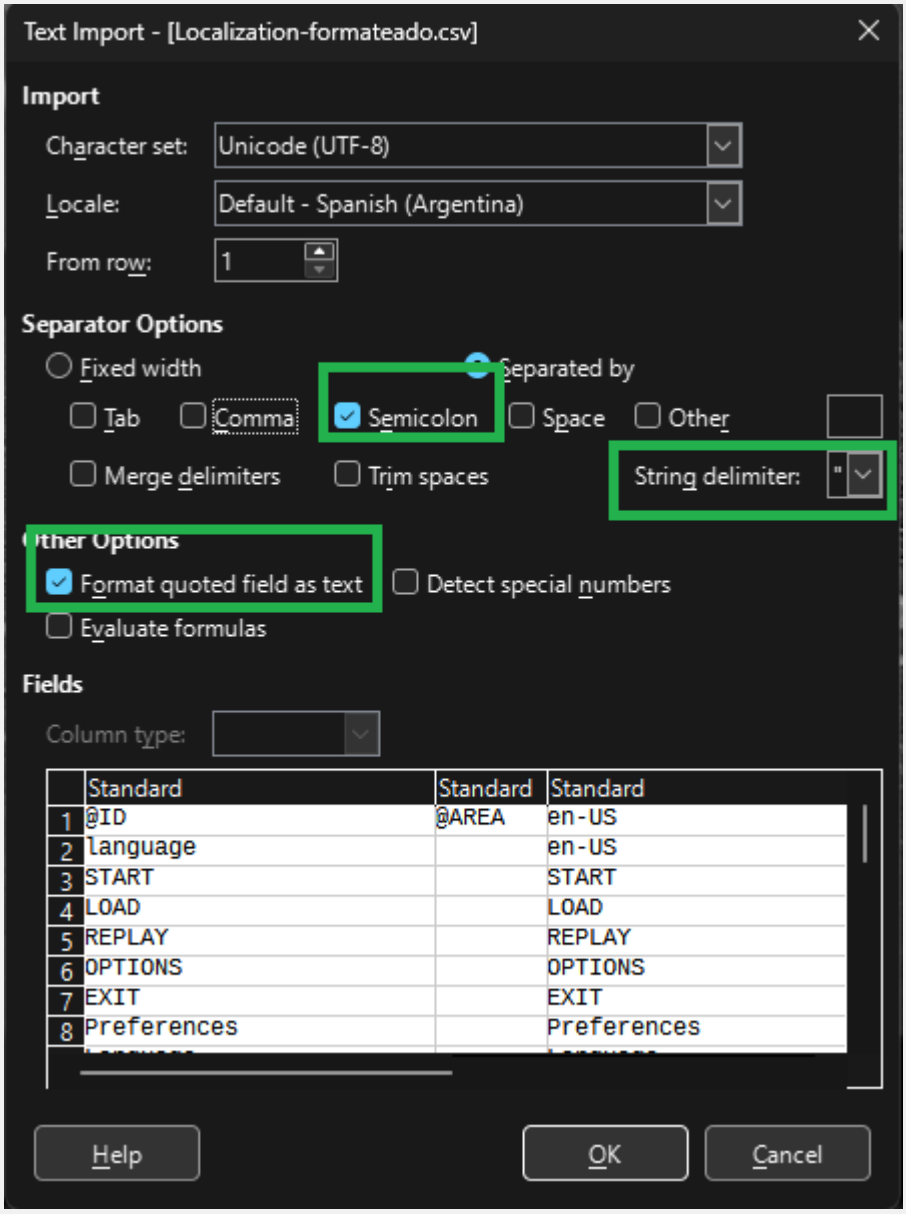
Esto hará que automáticamente se genere un archivo `.CSV` con encoding `UTF-8` en su carpeta.

Puede editar el archivo con cualquier editor de hoja de cálculos (recomendado), o con cualquier editor de texto.

> Le recomendamos fuertemente utilizar `LibreOffice Calc`. Es *Open Source*, multiplataforma, y funciona perfectamente. Si desea saber más de `LibreOffice Calc`, le dejamos un enlace en la sección de **Recursos Adicionales**.

El **formato default** de los archivos `.CSV` en PLS es: celdas con *comilla doble (")*, separación de celdas con *punto y coma` (*😉

Configuración default recomendada para Libre Office



.

Luego, dentro del archivo, debería usar la primera columna para clave, y las demás para los distintos idiomas.

Los headers de las columnas se definen en la primera fila. El idioma debe estar expresado en ISO 639-1.
Ejemplos: `en`, `en-US`, `en-EN`, `es`, `es-ES`, `es-AR`.

Archivo CSV de Ejemplo 1

A	B	C	D
@ID	@AREA	en-US	es-AR
language		English	Español
BASIC EXAMPLE			EJEMPLO BÁSICO
In this example, the language will change every five seconds. You can see the code for this in the script 'Example1Script.cs'.			En este ejemplo se cambiará el idioma cada cinco segundos. Pueden ver el código del mismo en el script 'Example1Script.cs'.
Hello world!			¡Hola mundo!
Selected language: {0}			Idioma elegido: {0}
Language code: {0}			Código del idioma: {0}

.

---

3. Cargar un Idioma

Este paso es opcional. Si dejas la opción `autoLoadSystemLanguage` en `true` dentro del Script `LocalizationCoreScript`, automáticamente detectará el idioma del sistema y elegirá en base a ese.

Carga Manual de Idioma

Si aún así desea realizar la carga manual del idioma, deberá desactivar `autoLoadSystemLanguage`, y luego llamar al método `LoadLanguage`.

Ejemplo:

```
LocalizationCoreScript.Instance.LoadLanguage("en");
```

> Nota: Si usted tiene definidos el idioma "en-US" dentro de su archivo CSV, aún así puede llamarlo como LoadLanguage("en"). Performant Localization System lo encontrará sin problema.

---

4. String Localization

En el paso 4, ya procedemos a traducir los strings que necesitemos.

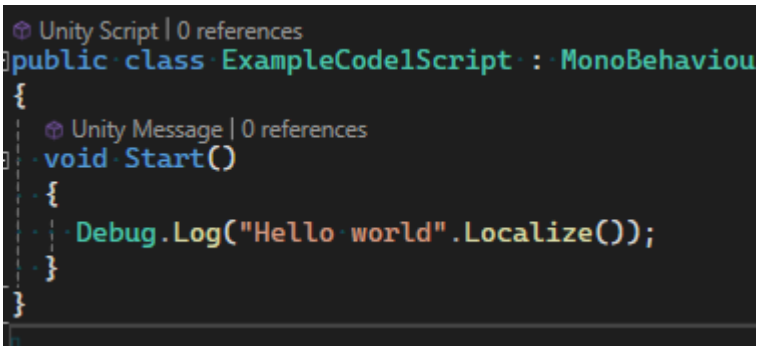
4.1. Extension Methods

Los `Extension Methodos` son la mejor forma de localizar strings para mantener el código limpio. Estos métodos extienden la clase `string` para poder acceder fácilmente a su traducción.

Para usar estos métodos, debes incluir el siguiente `using`:

```
using PLS.Extensions;
```

Para localizar strings, en la mayoría de los casos, simplemente basta con llamar al extension method `Localize()`.



Recomendamos que en vez de usar claves codificadas, como "hello-world", directamente utilizes de clave los textos a traducir. De esta forma el código queda más natural, más facil de leer.

Lista de Extension Methods

- Localize()

    - Intenta traducir el texto, y si no encuentra la traducción, devuelve el mismo texto.
- LocalizeOrDefault(defaultValue)
    - Intenta traducir el texto, y si no encuentra la traducción, devuelve el valor que se indique en el parámetro `defaultValue`.
- LocalizeOnlyIfExists()
    - Intenta traducir el texto, y si no encuentra la traducción, devuelve `null`.
- LocalizeIfKeyLengthGT(lengthLimit)
    - Realiza la traducción solo si el texto supera del largo indicado en el parámetro de `lengthLimit`. Esto es muy útil para no traducir botones de mando, como `A`, `B`, `X`, `Y`, etc.
- LocalizedReplace(key)
    - Intenta traducir todas las ocurrencias de `key` en el texto de entrada. Esto utiliza el método `String.Replace`, por lo que puede generar algo de `garbage`.
- SmartLocalize()
    - Este es el método más pesado de traducción. Recomiendo usarlo lo menos posible. `SmartLocalize` intenta traducir un párrafo de forma "inteligente" (por un algoritmo) utilizando todo lo que tiene disponible en el diccionario. Dicho esto, pueden imaginar que no viene sin costo.

4.2. Llamadas Directas a LocalizationCoreScript

Si bien recomendamos utilizar los métodos explicados en la `sección 4.1`, también pueden llamar directamente a las funciones que se encuentran en el script `LocalizationCoreScript`.

Este además tiene properties adicionales, que le permiten por ejemplo saber el idioma actual cargado.

```
7 references
public interface ITranslator
{
    6 references
    bool IsLoaded { get; }

    4 references
    string Language { get; }

    1 reference
    string NormalizedLanguage { get; }

    3 references
    List<string> Languages { get; }

    11 references
    void LoadLanguage(string language);

    1 reference
    void LoadLanguageByZone(string language, params string[] zones);

    9 references
    string Localize(string key);

    3 references
    string LocalizeOnlyIfExists(string key);

    4 references
    string LocalizeOrDefault(string key, string defaultValue);

    5 references
    string Replace(string textToLocalize, string id);

    6 references
    string SmartLocalize(string text);
}
```

4.3. Scripts

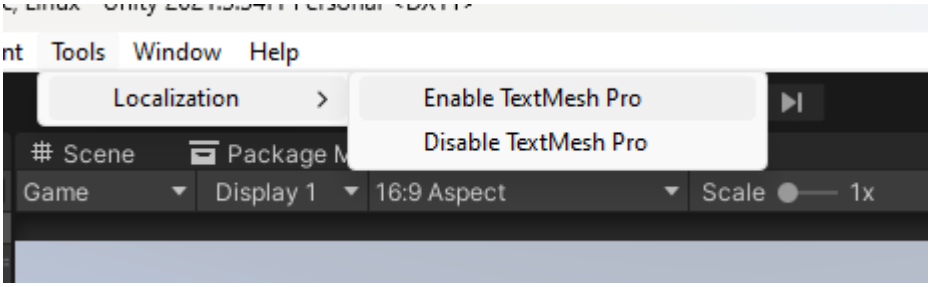
- AudioTranslationScript
- EnableObjectByLanguageScript
- LanguageChangerScript
- TextTranslatorScript

4.4. Plugins / Auto-Localization Scripts

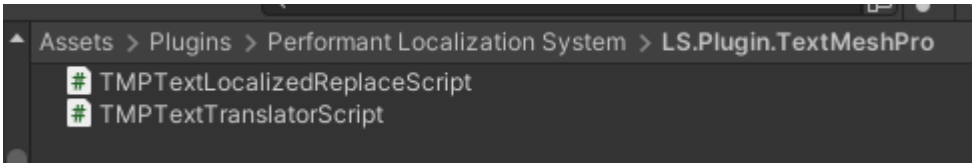
En la carpeta `LS.Plugin.TextMeshPro` encontrará Scripts para traducir automáticamente interfaces con TextMeshPro.

Para poder utilizarlos, deberá tener instalado `TextMesh Pro` en su proyecto.

Luego de esto, simplemente debe ir a `Tools > Localization > Enable TextMesh Pro`.



Esto le permitirá hacer uso de los siguientes Scripts:



Scripts

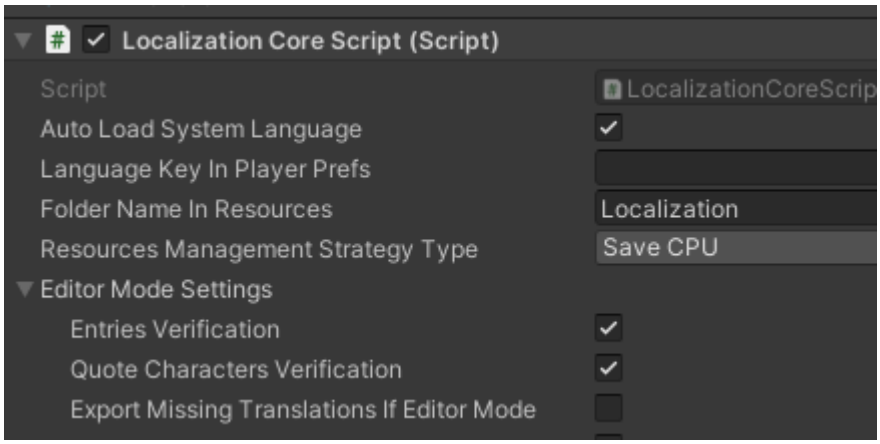
- TMPTextLocalizedReplaceScript
  - Traduce automáticamente utilizando el método `LocalizeReplace`.
- TMPTextTranslatorScript
  - Traduce automáticamente utilizando el método `Localize`.

Migración

La forma de manejar PLS es muy sencilla. Nuestro principal objetivo es lograr que la migración para usar el sistema de traducción sea simple y poco invasiva, evitando tener códigos ilegibles para los textos del juego o tener que centralizar todo en una clase.

1. **Habilita la exportación de traducciones faltantes**

- Puedes habilitar la opcion `Export Missing Translations If Editor Mode`. Esta generará un archivo `.CSV` en la carpeta de `Assets` por cada vez que una llamada a un método `Localize` no encuentre una traducción.



- Lo mejor de todo es que esto no afecta al rendimiento final de tu juego en producción, dado que sólo se habilita si estás en el editor de Unity.

2. **Identificar Puntos de Traducción**

- Comienza identificando las partes de tu código donde se necesita la localización del texto. Esto puede incluir elementos de la interfaz de usuario, diálogos en el juego y cualquier otro contenido textual.

3. **Preparar Archivos de Traducción**

- Crea y organiza tus archivos de traducción. Asegúrate de que cada idioma y sus respectivas traducciones estén correctamente formateadas y guardadas en una subcarpeta de alguna carpeta `Resources` de tu proyecto.

3. **Actualizar tu Código**

- Reemplaza los strings hardcodeados con llamadas a los métodos de localización proporcionados por PLS.
- Ejemplo antes de la migración:

```
myTextComponent.text = "Hello, World!";
```

Ejemplo después de la migración:

```
myTextComponent.text = "Hello, World!".Localize();
```

4. **Probar tu Localización**

- Después de actualizar tu código, prueba exhaustivamente el proyecto para asegurarte de que todo el texto esté correctamente traducido. Recuerda verifica diferentes idiomas.

5. **Optimizar y Refinar**

- Revisa tu implementación para buscar oportunidades de optimización.

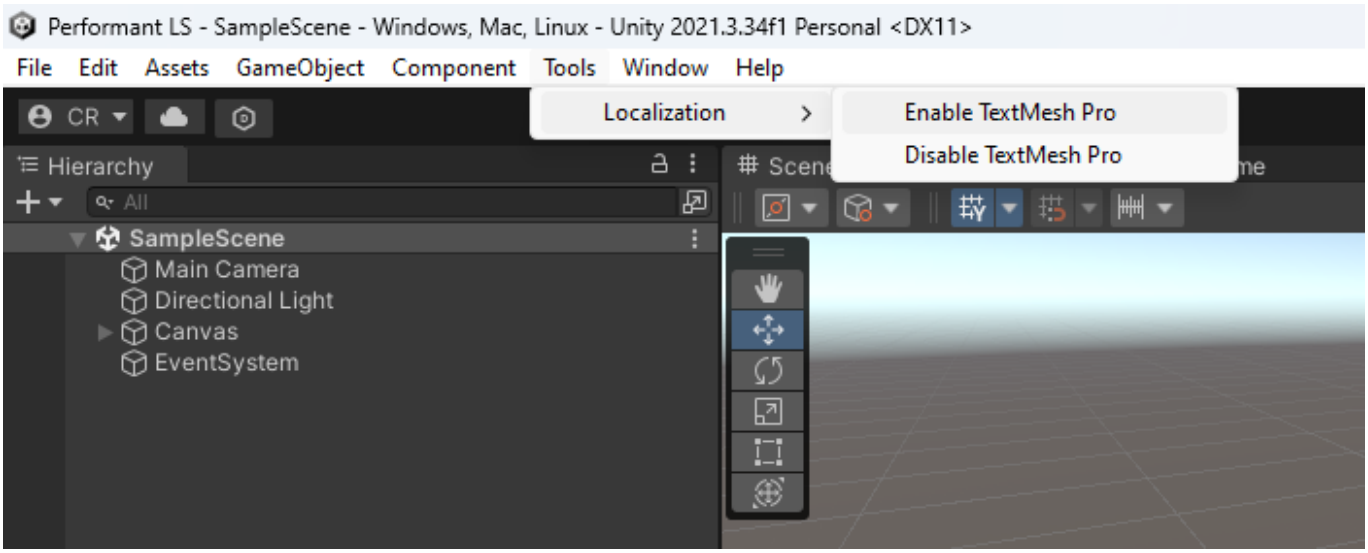
Consejos Adicionales

- **Respaldar Tu Proyecto**: Antes de comenzar la migración, asegúrate de hacer una copia de seguridad de tu proyecto para evitar perder trabajo existente.
- **Migración Incremental**: Considera migrar en etapas para manejar los cambios de manera más efectiva y detectar problemas tempranamente.
- **Buscar Apoyo**: Si encuentras desafíos, no dudes en contactar a la comunidad de soporte o consultar la documentación para obtener ayuda.

Plugins

TextMesh Pro

Para habilitar los scripts que interactuan con TextMesh Pro, debe ur al menú `Tools > Localization > Enable TextMesh Pro`.



FAQ

¿Este sistema es la segunda versión de "Localization System"?

Este sistema es totalmente independiente de nuestro querido "Localization System".

Localization System es un Asset que viene evolucionando desde hace muchísimos años. Es sólido y fue usado en por varios estudios, incluyendo un juego AAA.

Ese sistema permite leer tanto archivos CSV como XML.

Performant Localization System en cambio:

- Es un sistema de localización completamente **nuevo y diferente**.
- Está pensado para generar la **menor cantidad de** `garbage` **posible**.
- También se pensó para que las traducciones por código sean **totalmente limpias**.
- **Sólo soporta formato CSV**.

Referencias

- Asset store page: https://assetstore.unity.com/publishers/1881
- Official website: https://forjagames.com
- Youtube channel: https://youtube.com/@ForjaGames
- Support (discord): https://discord.com/invite/dqGdSpVcAc

Recursos Adicionales

- LibreOffice official website: https://www.libreoffice.org

Soporte

Puede obtener soporte en tiempo real por medio de nuestro Discord:

https://discord.com/invite/dqGdSpVcAc

Por favor, si le gusta este Asset no olvide calificarlo. Es muy importante para nosotros, y nos ayuda a crecer.

¡Muchas gracias!