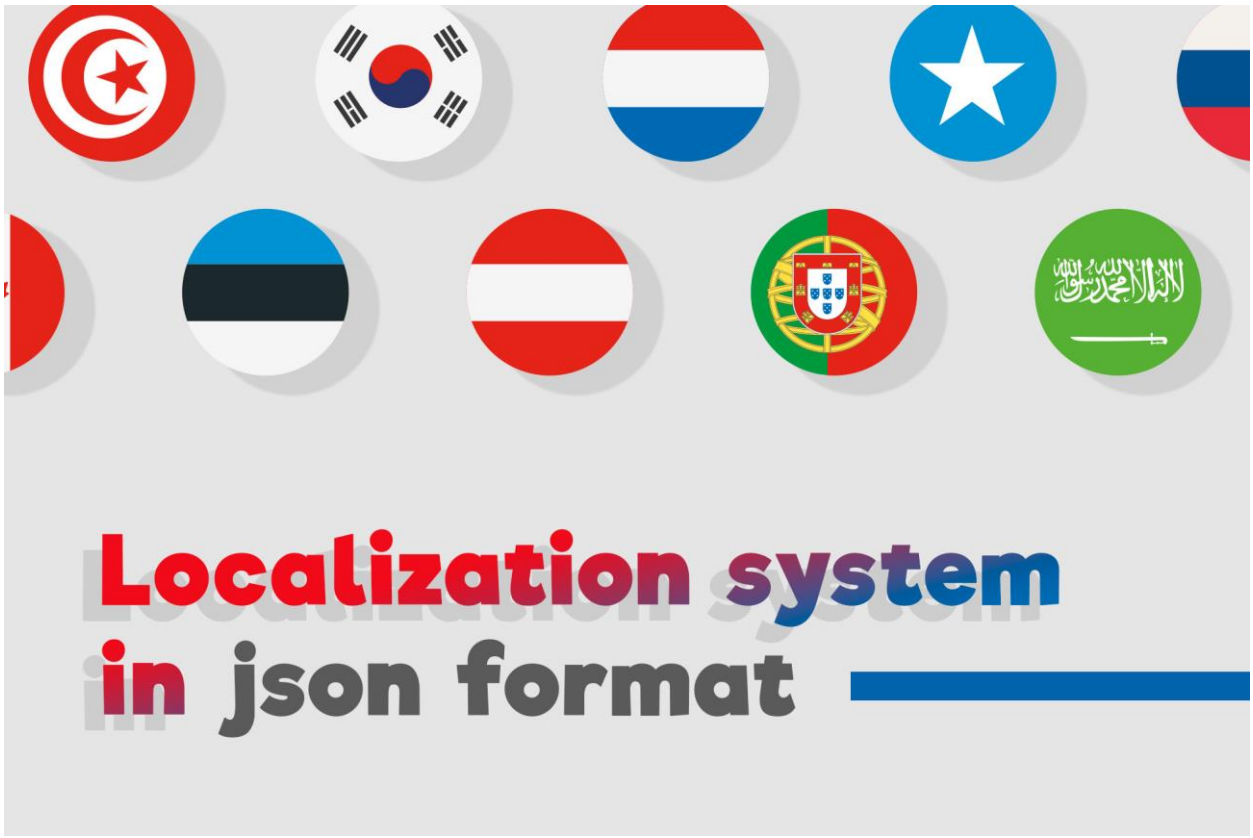
localization system in json format



Thank you for purchasing my asset!

Introduction

Localization system for your project. All languages are loaded from json files located in the root of the project. This allows you to scale multilingualism already in the assembled project. All you need is to add the required json file to the desired folder and this language will appear in the project automatically.

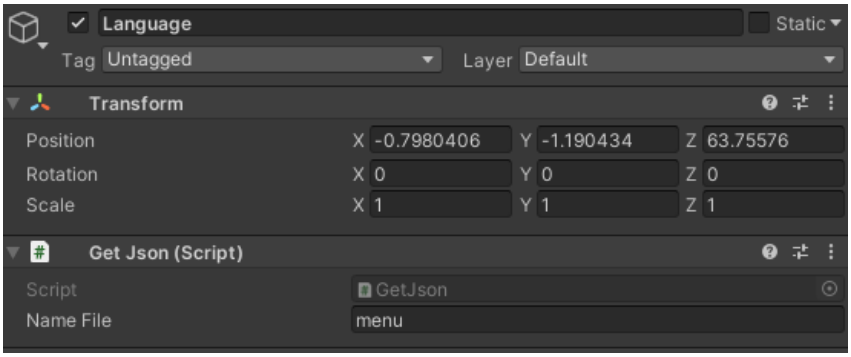
The main advantages of this approach to project localization:

1. Unlimited number of languages added;
2. A simple and flexible system for changing the texts in the json file already in the assembled project or in the unity editor;
3. Algorithm for automatic detection of the system language and memorizing the selected language;
4. Ability to add languages already in the assembled project;
5. A system for binding keys to text fields in the project to automatically detect the required translation.

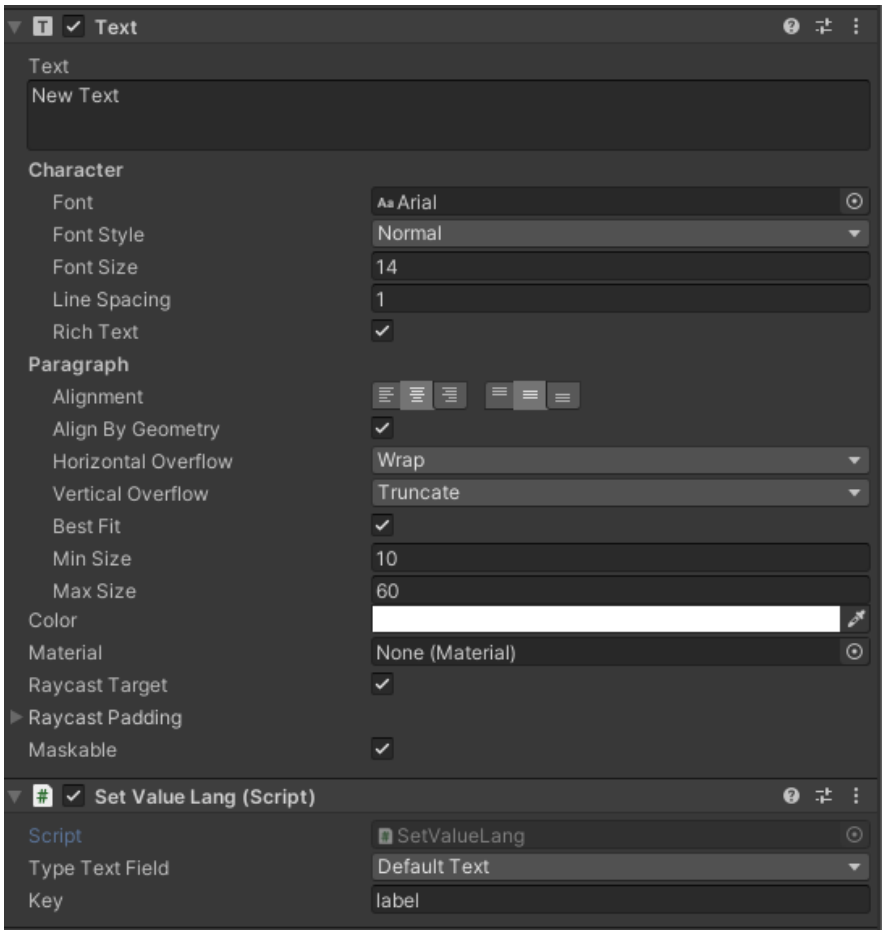
Setting up an asset

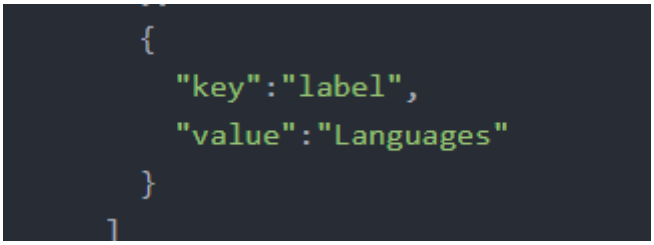
After you have installed this asset, you need to make some settings:

1. The "**StreamingAssets**" directory must be moved to the project root in the Assets folder. This is a reserved name from Unity so that after building the project, there is a folder "**StreamingAssets**" in the root, which contains our translations of all languages.
2. Add the "**GetJson**" script to your scene. This script determines which language is currently on your system or the language saved in the registry (PlayerPrefs), the script also accesses the folder with all your translations and fetches all keys from the json file. Also, in the "**NameFile**" field, you must specify the name of the file with the translation for a specific scene. It is important to indicate the name without the format and path to the file.



3. For each TextField in your Canvas panel, you need to add a "**SetValueLang**" script. This script makes a selection of the required translation from the already loaded json file for a specific field. This script is an example where it substitutes translation for a standard text field or TextMeshPro. In the "**Key**" field, you must specify the key from the json file, where the value will be assigned to this text field. In the drop-down list "**TypeTextField**" you can select the appropriate type of text field, which is currently used in your project.



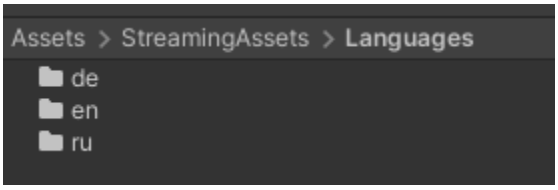


After such basic settings in your project, the translation will already work if the json translations are in the root of the project. It remains to figure out how to add translations and change the current language of your project.

How to add a language and your translation?

In order to add a new language to your project, follow these steps:

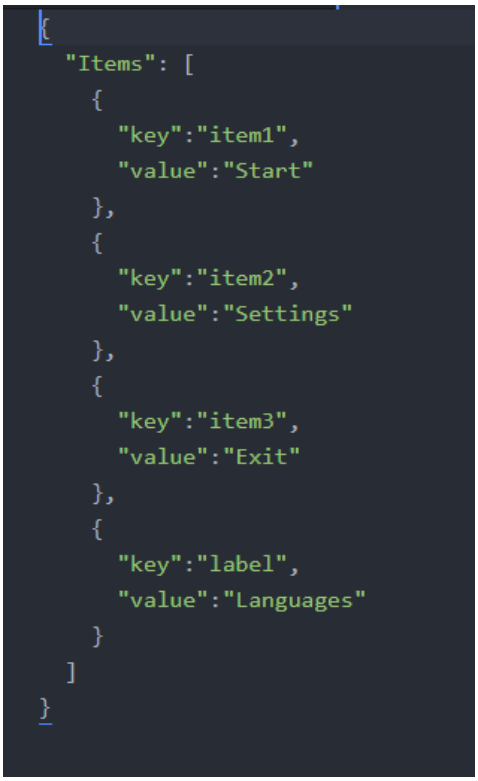
1. The path of all languages is as follows "**Assets/StreamingAssets/Languages/**". This folder contains all languages. If you have a great path, then you need to make sure that the path matches the above path.
2. In the "**Languages**" folder, add a short name (2-3 letters) of the language in Latin.



3. In the folder of each language it is necessary to create a json file "**config.json**", it will contain the basic settings and method data of a specific language.

The key "**key**" is always "**config**". Key "**title**" contains the name of the language, it will be displayed in your project on the button to change the language. The key "**key_lang**" contains the value of the multiple name of the language. It is important that this key matches the name of the folder in which this settings file is located.

After the done manipulations, you can add json files of your translations to the current folder. The file name can be assigned whatever you want, the main thing is to specify this name in your project in the settings of the "**GetJson**" script. The json file structure for your translation should look like this:



Each translation has a key and a meaning. The key is used in the project to identify the translation of a specific field.

How do I switch languages in a project?

In order to select a language, you need to call the "**SwitchLanguage**" method in the "**ConfigButtLang**" script. Typically this method is called when a button is clicked.

This script has a variable "**key_lang**", which should take the value of the short name of the language to which your project will switch.

For everything to work correctly, this script must be hung on each button that triggers the language change event. This is not entirely correct, I know! You can improve this functionality or wait for the asset update! Anyway, I would appreciate your feedback in AssetsStore!

An illustrative example in the demo scene

I added a demo scene to the project, where three languages are already configured and after launch, you can change the language and see how it works. But the main message of this scene is to show how I can improve my localization system.

The demo scene contains functionality that automatically adds all translations from the "**Languages**" folder, and also automatically instantiates buttons with a choice of translation on Canvas. The script that is responsible for this is called "**GetLanguages**". I will not go into detail on how this algorithm works, since we will go into the discussion of a completely different topic! But if you have any questions, please write to my mail **denis@proger.xyz**! I will gladly help you!

Support

If you are confused by something, you can write to denis@proger.xyz and we will do our best to contact you immediately. We hope you enjoy our asset and good luck in creating your games!