Multi Language

Config file

All the UI texts in this project is configured in a json file:

“Assets/RoomBuildingStarterKit/Assets/Config/GameText.json”

The json file consists of key value pairs, as in Fig.1. We choose one of them, such as:

```
“START_GAME”: {
    “ENGLISH”: “Start Game”,
    “CHINESE”: “开始游戏”
}
```

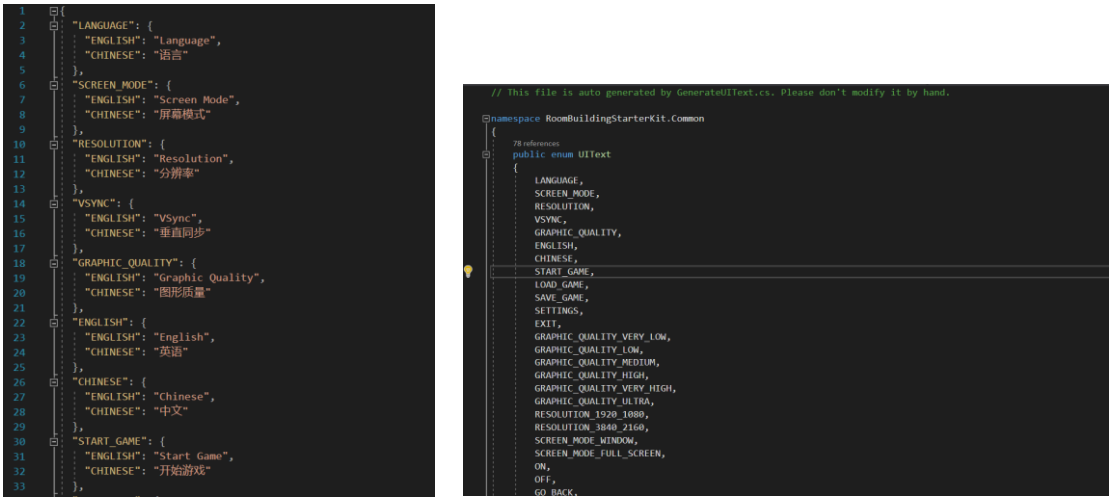


Figure.1 GameText.json and UIText.cs

The key **“START_GAME”** is an UIText enum value. UIText defines IDs for different texts. **“ENGLISH”** and **“CHINESE”** are different language identifiers. **UIText.cs** is generated by editor script in Fig.2. Every time you modify GameText.json, you need to click **Tools > GenerateUIText**. Then UIText.cs will be updated with your changes in GameText.json file.

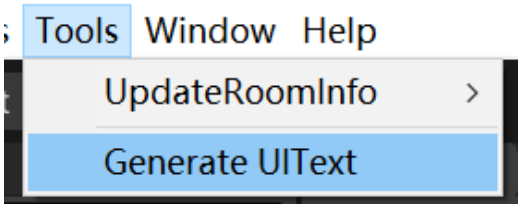


Figure.2 GenerateUIText

TMP UGUI Wrapper

In this project, we use Text Mesh Pro to display game texts. I write a Wrapper for Text Mesh Pro UGUI named **Text Mesh Pro UGUI Wrapper** which could response to language switch in real time. The TMP wrapper will find texts in GameText.json by UIText enum value and current language setting. If you want your Text gameObject support multi-language, you should remove the original Text Mesh Pro UGUI component then add component **"Text Mesh Pro UGUI Wrapper"** in Fig.3. The inspector of Text Mesh Pro UGUI Wrapper component is exactly same with original TMP component, all you need to do is fill the Text.Input with prefix "UITEXT." + UIText enum value, in this example, we should write "UITEXT.START_GAME" as in Fig.4.

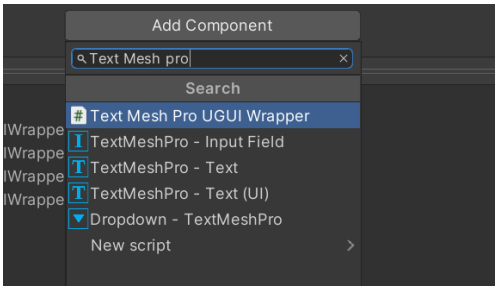


Figure.3 Text Mesh Pro UGUI Wrapper component

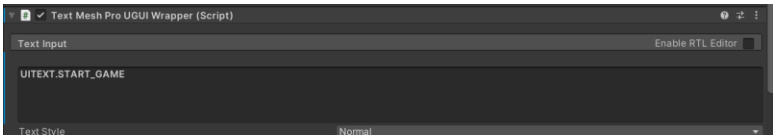
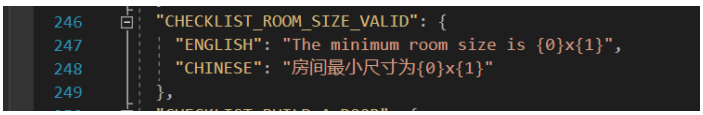


Figure.4 Text Input

The Text Mesh Pro UGUI Wrapper also supports parameters placeholder. For example, in Fig.5, the **"CHECKLIST_ROOM_SIZE_VALID"** enum defines the text **"The minimum room size is {0}x{1}"**. The {0} and {1} are two placeholders for the input parameters. When fill the string parameters in script, you should first acquire your **TextMeshProUGUIWrapper** component, then use its **SetGlobalText** API with the UIText enum and a string array consists of the two parameters:

```
this.Text.SetGlobalText(
    UIText.CHECKLIST_ROOM_SIZE_VALID,
    new string[]{ "3", "5", }
);
```

Finally, the ui text will displayed as **"The minimum room size is 3x5"**



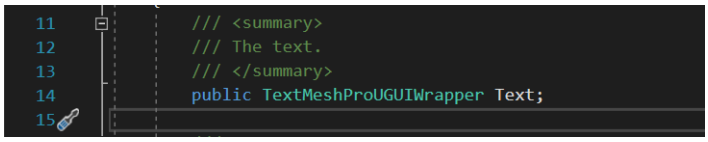


Figure.5 Parameter placeholders

If I left something unclear or if you encounter any problem to this project. Please don't hesitate to contact with me with:

Email: hurrywon@163.com
Discord: https://discord.gg/J7JTGqcf3v