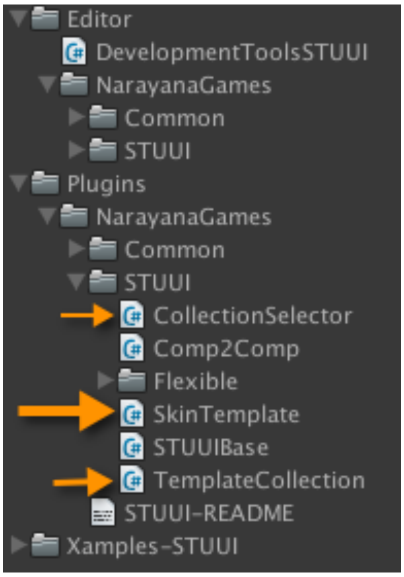
First Steps with STUUI



After you have downloaded and imported STUUI to your project, you will see three main folders:

- **Editor:** Contains *NarayanaGames/Common* and *NarayanaGames/STUUI*
- **Plugins:** Contains *NarayanaGames/Common* and *NarayanaGames/STUUI*
- **Xamples-STUUI:** Contains several example scenes, skins, and fonts

The reason the core files are stored in Plugins is so that you can immediately use them from Boo and JavaScript (as STUUI is written in C#, that works automatically). The *Common* folders contain generally useful classes, and *STUUI* obviously contains everything specific to *STUUI - Skinning and Templating for Unity UI*. All examples are contained in a separate folder that you can delete to clean up your project once you are familiar with STUUI.

The files you'll work with are highlighted in green in the screenshot above:

- **SkinTemplate component:** The core component that STUUI provides and that you always will want to use is SkinTemplate. This is attached both to the objects that need to pull styling information, as well as those providing that information. That way, you can easily copy the first few GUI-objects that you have ready to create your styling templates. All you need to do then is assign the "Reference" of the object in your scene that pulls the information to the object that provides the information (usually in a prefab but could as well be another object in your scene). For most scenarios, this is actually all you need.
- **TemplateCollection component: I**f you want to switch the skin on the fly, during runtime, you'll also use the TemplateCollection-component which provides a set of SkinTemplates and a way to easily exchange one TemplateCollection for another. This follows the same approach: You have one main TemplateCollection to which each SkinTemplate of that collection refers to (by having that collection assigned to the Collection slot). Then, you can have several specific TemplateCollections which need to have exactly the same structure (same game

object names but different styling settings) so they can be assigned to the Current Collection slot.

- **CollectionSelector component:** For this, STUUI also provides the CollectionSelector which gives you an easy to use API to switch the collections on the fly. See Example_03 for a full example of using this.
- **Folder Flexible with Flexible- and SimpleComponents:** In addition to the main components that are supported (**Graphic** for colors and materials, **Image** for sprites and relevant settings and **Text** for all text settings), which always apply all supported properties, there's also several “**FlexibleComponents**": **FlexibleText** is similar to Text but offers a checkbox for each property, and only when that checkbox is checked will that property be applied to the SkinTemplate that’s pulling the style information.
  In addition, there's also **SimpleColor** (only a color that is applied to any Graphic), **SimpleSprite** (just a sprite that can be assigned to any Image), **SimpleText** (only the text-string that will be applied to a Text-component) and **SimpleAudioClip** (which has an audio clip that can be assigned to any AudioSource). **SimpleText** and **SimpleAudioClip** are primarily for **localization**: By using those, you can easily build **localization packages** that can then be assigned like other skins to switch to another language, including any spoken language (this is illustrated by Example_05). For this purpose, you can, of course, also use SimpleSprite to change any images, and FlexibleText in case you want to change the font or font size for specific languages.