Systems Library

- Documentation -

Features **1**
Interface system 1
Game settings system 2
Input Mapping system 2
Localization system 2
Travel system 3
Setup systems **3**
Interface system 3
Game settings system 4
Input mapping system 4
Localization system 4
Travel system 5
Tips and tricks **5**
File names 5
Number of repetition with a timer 6

1. Features

1.1. Interface system

The interface system is used to create widget with mouse/keyboard/gamepad input. You can control the logic of your widget easily.

FL_InterfaceSystem	This library contains everything you need to use the interface system
Widget_Base	This class is used to create interface that can manage mouse/keyboard/gamepad input
InteractiveContent_Base	This class is used to create interactive component. They are useful to create logic inside the widget / be bound to others systems / ...
InformativeContent_Base	This class is used to create informative component. They are useful to inform your player about somethings / be bound to others systems / ...
Panel	Custom panel, they can receive input from the player and send the input to the appropriate component or used it for his own logic.
Overlay	Overlay is used in special interactive content. They are useful when you expect an answer from your player.

1.2. Game settings system

The game settings system work closely with the interface system. This system is useful to create an options menu, your player will be able to change your game settings like graphic, sound, gameplay and more ...

FL_GameSettingsSystem	This library contains everything you need to use the game settings system
DT_GS_Settings	This data table is used to create settings just by adding a row
SG_GS_Settings	Useful to save your game settings

1.3. Input Mapping system

Input mapping system work closely with the interface system. Input are regrouped by category that mean you can't have the same input in one category. If your player enter the same input in a category, the first input mapping will be reset to "None". You can create one category for each game mode you have (example : player movement input, player fighting input, ….)

FL_InputMappingSystem	This library contains everything you need to use the input mapping system
InputMapping_Base	This class is used to create a widget with a detection of input from your player.

1.4. Localization system

The localization system is made with tables : different tables composed with columns for each languages. English is the native language and for now, only french has been added. You can easily translate a text using the "translate" function accessible everywhere.

FL_LocalizationSystem	This library contains everything you need to use the localization system
STT_LS_TextTable	Used this structure to create data table.

1.5. Travel system

The travel system allow you to load and unload a level and his sublevel. The travel system use a "loading screen" when you load different level/sublevel.

A_TravelSystem	Must be placed in the persistent_level and a reference must be created in the game instance
TS_LoadingScreen	Example of a loading screen.
E_TS_Name	List of your levels.

2. Setup systems

You can find in “SystemsLibrary/Example_Setup” example of ‘how to setup all systems / create widget / ...’

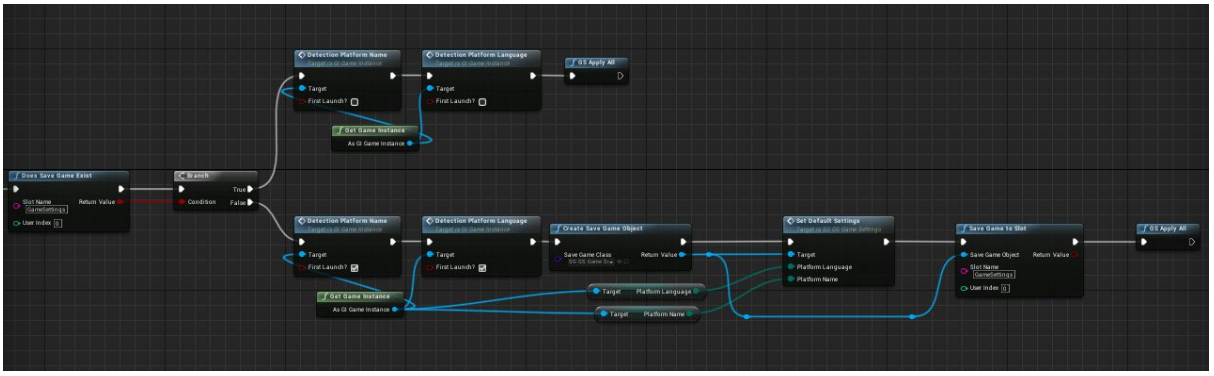
2.1. Interface system

- open ‘SG_GS_GameSettings’
- fill up the map variable ‘IS_GamepadInput’
- open ‘EnglishKeyboard’ function
- fill up the ‘English’ local variable
- do the same thing for all languages

- use ‘Widget_Base’ to create a new widget
- add your new menu in ‘E_INT_ListInteractiveMenu’
- add your new menu in ‘STV_INT_InteractiveUMG’
- open FL_InterfaceSystem
- connect ‘select’ nodes with your new menu

2.2. Game settings system

- load your default settings (see image below)



blueprint to add in your persistent level

- add ‘BI_GS_LogicMenu’ in your widget that handle game settings
- add ‘BI_GS_LogicContent’ in your interactive content

- to add a new settings, add a row in ‘DT_GS_Settings’
- create an apply function in ‘FL_GameSettingsSystem’
- add the default value of your new setting in the ‘SG_GS_GameSettings’

2.3. Input mapping system

- add ‘BI_GS_LogicMenu’ in your widget that handle input mapping
- add ‘BI_GS_LogicContent’ in your interactive content

- create your action/axis mapping in the project setting → input

2.4. Localization system

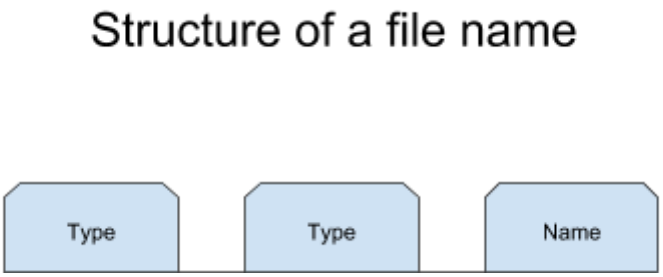
- add a new language in ‘E_LS_Language’
- add your new language in ‘STT_LS_TextTable’
- open ‘FL_LocalizationSystem’
- connect your new language to the ‘select’ node

2.5. Travel system

- add the ‘A_TravelSystem’ in your persistent level
- create a reference variable in your game instance and set it at the start of your persistent level
- set ‘TS_Levels’ with all levels/sublevels your have
- add your levels in the ‘E_TS_LevelName’

3. Tips and tricks

3.1. File names



A file can have one or two types.

A : Actor
BI : Blueprint Interface
C : Content
DT : Data Table
E : Enumeration
FL : Function Library
GI : Game Instance
GS : Game Settings
INT : Interface
IM : Input Mapping
LS : Localization System
M : Material
MD : Mouse Design
S : Setup
SG : Save Game
STT : Structure Table
STV : Structure Variable
T : Texture
TS : Travel System

3.2. Number of repetition with a timer

If you set your timer with 0.1 that means your timer will be launched 10 times in one second.

Time	Number of repetition
1	1
0.1000	10
0.0500	20
0.0250	40
0.0170	60
0.0100	100
0.0085	120
0.0050	200
0.0025	400