

Advanced Save & Load



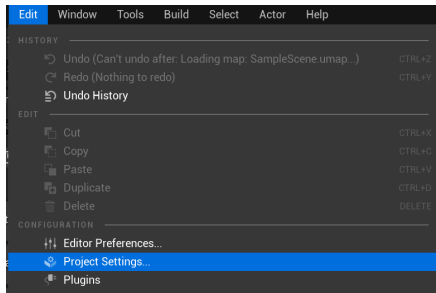
What is the system capable of:

- 100% Blueprint
- Thumbnail for every save slot
- Thumbnail quality control
- Save slot with character limitation control
- Date and time formats control
- Delete selected save slot
- Notification system with colors and text control
- Fade in and out on load
- Simple save game system
- Autosave with interval control
- Checkpoints based triggers
- Quick save
- Simple load game system
- Quick load
- Limit slots or have an unlimited amount
- Continue the game from the last point in the menu
- Save player location by default
- Custom checkpoint names
- Simple save component that controls save settings
- AI's example with location and life system using the system
- Player example stats (life, mana, XP) system using the system
- Super easy integration - plug and play (approximate 5 min)

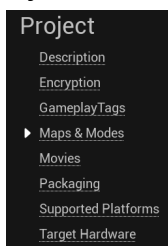
Activation With The Provided GI

To start the work with the system follow the next steps:

1) Go to the project settings



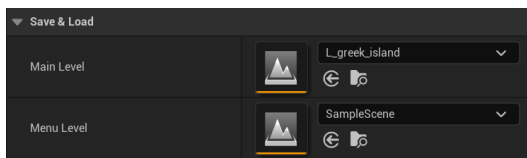
2) Under project select the option



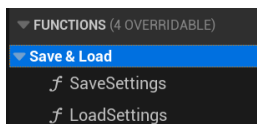
3) Select the Game Instance that you will use (The one provided)



4) To work properly we need to set the main menu level and the main level that will be played



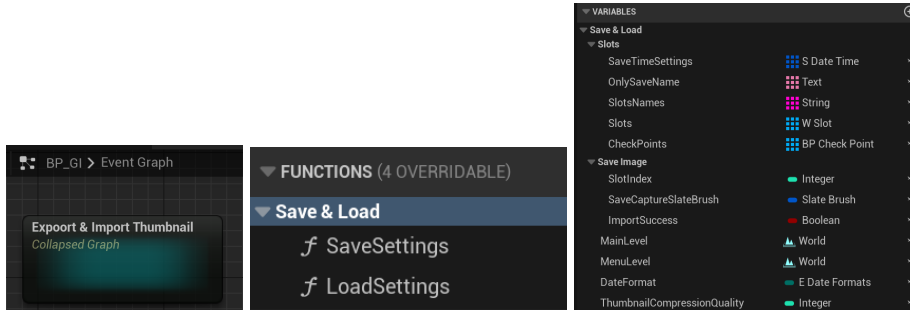
5) Start to set values



Activation With Own GI (Long Process)

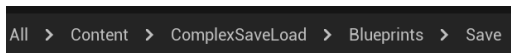
1) In case there is already created GI (game instance) we open our GI and the provided GI

2A) From the provided GI we will copy the next to our GI

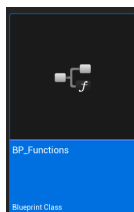


2B) Repeat steps 4-5 from **Activation With The Provided GI**

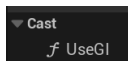
3) Go to this location



4) Open the blueprint



5) Create a cast function for easier communication that will be similar to the one provided, but with your GI

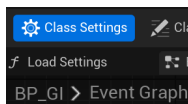


6) Search all over UseGI and replace with the new function created in section 5, in this case we need to replace the variables with the values around the save & load system with the new variables from our GI

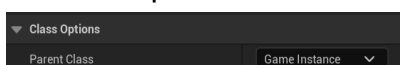
Activation With Own GI (Short Process)

1) In case there is already created GI (game instance) we open the provided GI

2) In **BP_GI** go to Class Settings



3) Under the Class Options section we click on Game Instance



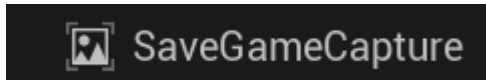
4) Now we choose the GI that we already have in our project (our own, the one that is already in use)

5) Repeat steps from **Activation With The Provided GI**

Player Setup

- 1) Open our main character and the provided character (**BP_ThirdPersonCharacter**)
- 2) Copy and paste the next things from the provided character to our main character in the next order

- Put it under the Camera Boom/ Spring Arm



- We can add it from the Add Component or just copy



- At the end copy and paste it into the Event Graph



How To Start

Player Setup

Inside the **AC_Save** we can find multiple control options:

Autosave Slot Name - will be shown on the screen when autosave

Quick Save Slot Name - will be shown on the screen when quick save

Check Point Slot Name - will be shown on the screen when checkpoint

Slots Limit - set the maximum amount of slots to save the game

Autosave Time Laps - the time interval in minutes between autosaves

Use Autosave - do we want to use Autosave or not

Use Notifications - do we want to use on-screen save system notifications or not

No Characters Warning - on save notify that the slot is empty

Max Characters Warning - on manual save notify that the slot is out of characters

Max Characters in Slot - the max number of characters in the slot name

Max Slots Amount - on save notify about slots amount limit

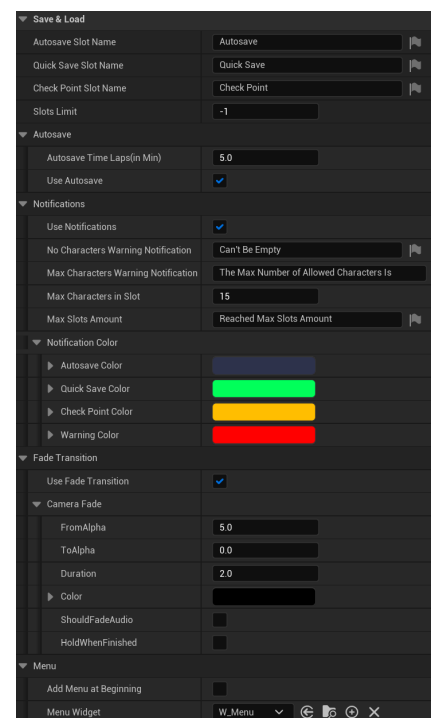
Notification Color - the colors of every notification

Use Fade Transition - do we want to have fade after loading between levels

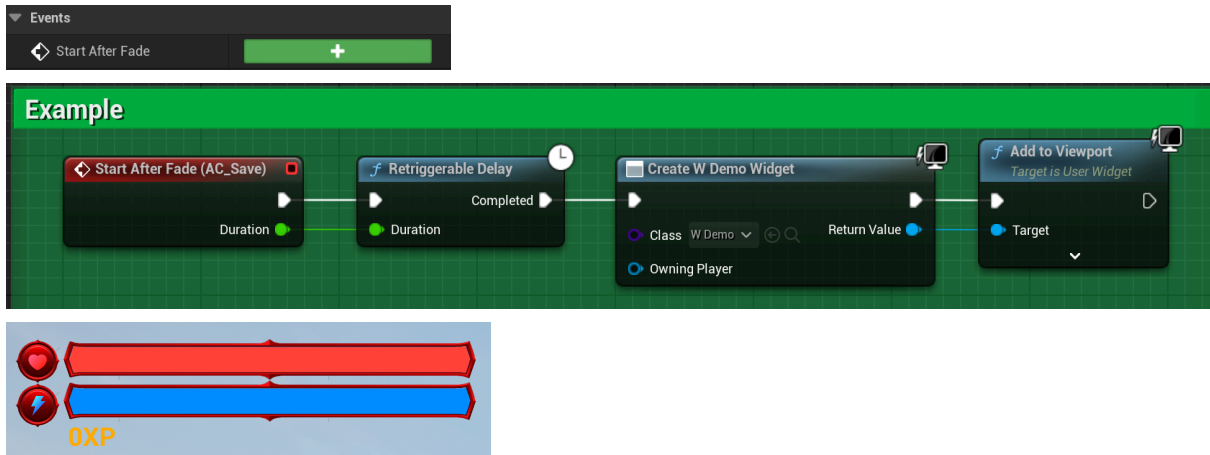
Camera Fade - settings for the player's camera fade when the main level is loaded

Add Menu at Beginning - Add pause menu after the load of the main level

Menu Widget - the pause menu widget to use



AC_Save contains an event that does something after the fade into the game, as example we provide a demo widget that shows the player's stats after the fade is finished



The demo stats will be in a couple of places, if we are talking about the player the player has variable/s that control the stats for example:

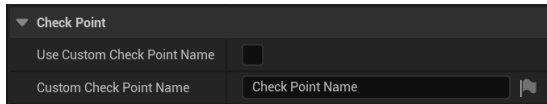


This is a struct that contains all player information

Default Value	
Demo Settings	
XP	0
Health	100.0
Mana	100.0

This information is spread over the project as any other variable and in this case, we use it in the demo widget to show the visual results

BP_CheckPoint



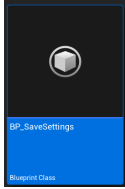
By default, the checkpoint name for all blueprints is “Check Point” or we can modify the name per every blueprint by clicking on the blueprint in the level, and by selecting it we can see in the settings the section from the image.

Use Custom Check Point Name - Enable or Disable custom checkpoint name

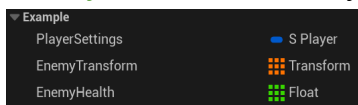
Custom Check Point Name - It’s the custom name for the checkpoint that will be in use only after Enabling “**Use Custom Check Point Name**”

Save & Load Setup

The project provides example variables that will be used in the system and the idea is to add your variables to it in the same way, in this case, we use **BP_SaveSettings**, and inside we insert our variables



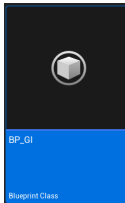
PlayerSettings is the same variable used in the player blueprint named DemoSettings
EnemyTransform is the array that has all our enemy transformations(Location, Rotation, Scale)
EnemyHealth is the array that has all our enemy health stats



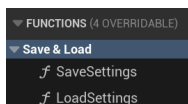
The enemy that is provided doesn't need a health array, but just a standard health variable



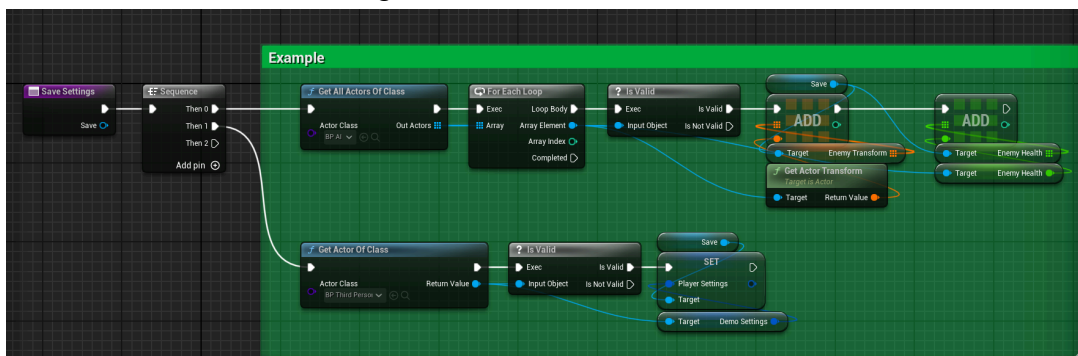
After we figured out what variables we needed now we set them in the system that can be found in **BP_GI**



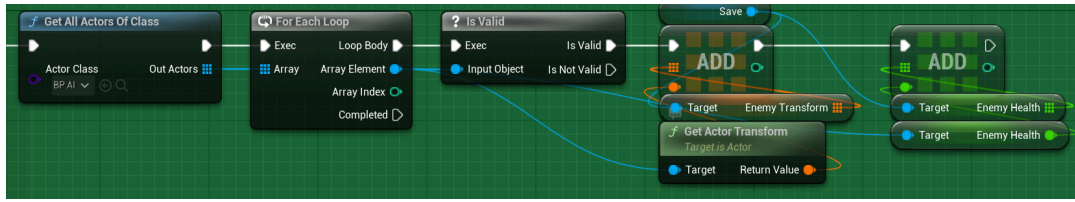
We have 2 functions to our use



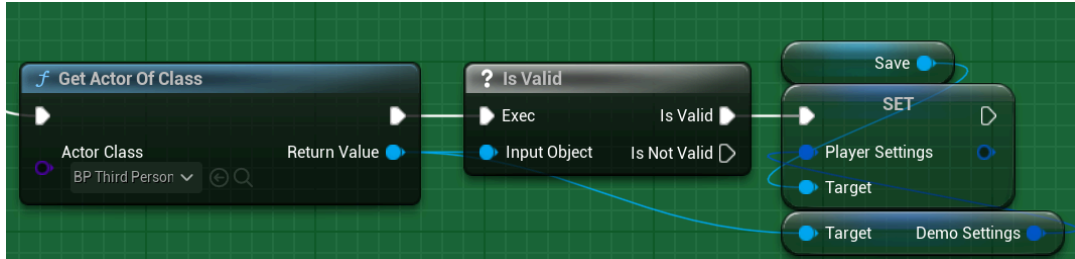
Let's start from SaveSettings



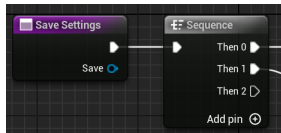
This is the setup of our enemies, we ran a loop over all our enemies from this class and save the transform and health values from our BP_SaveSettings



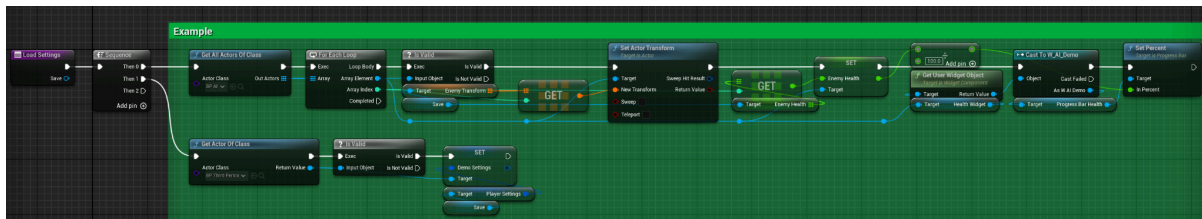
We are doing the same for our main character



In this way, we finish the process of saving values, and if we have more objects, actors or characters that we want to save, we just need to add a new pin in our sequence and recreate the process as it was shown for the enemies and the player



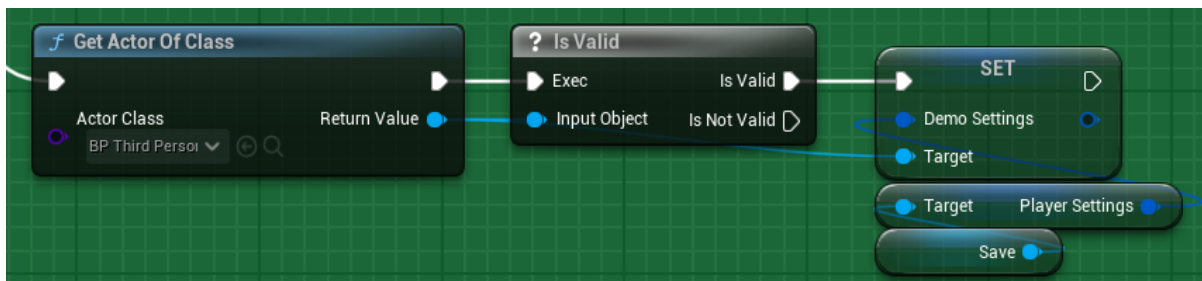
Now let's use LoadSettings



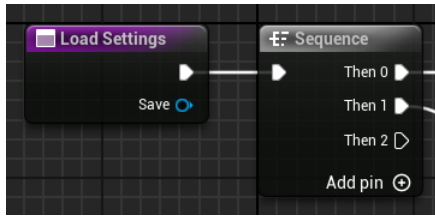
We restore all enemy's stats as transform and health and even set the health bar percent because in the demo it only updates when we hit the enemy so by restoring it from the load function we tell the system that on load finished update the widget as well



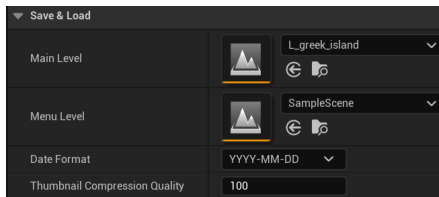
Doing the same to the player



In this way, we finish the process of loading values, and if we have more objects, actors or characters that we want to load, we just need to add a new pin in our sequence and recreate the process as it was shown for the enemies and the player



Use the same **BP_GI**, go to Class Defaults, and set the preferred Date Format you wish to use and the Thumbnail Compression Quality where 100 is the lowest quality and the lowest store cost and 0 is the highest quality and highest store cost

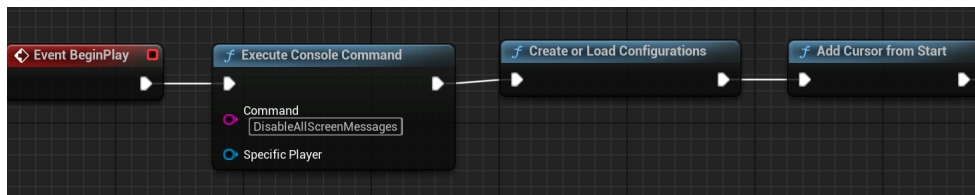


Notes:

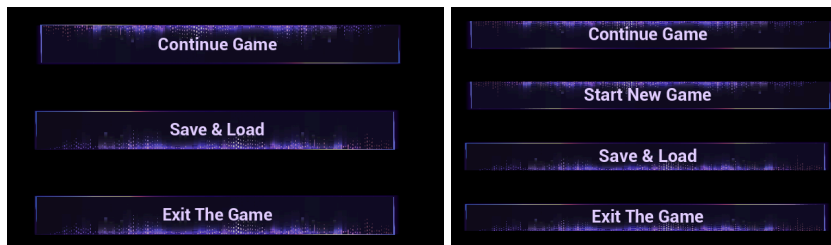
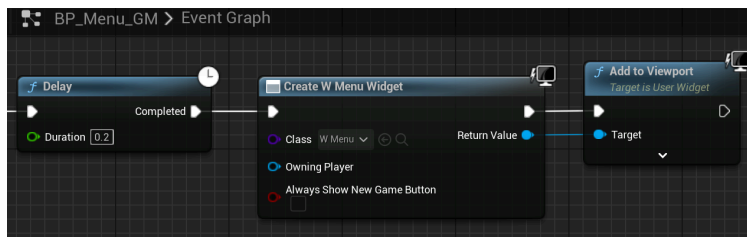
- Making sure that the system is working from time to time is good prevention to delete save files when developing, it can be good in cases where the system is not showing slots or does not update the saves.
- To find your saves we need to go into the project folders inside our Windows Explorer and then to a folder with the name Saved, we can see the folder with our saves with the name SaveGames.

Advanced Save & Load > Saved > SaveGames

- By deleting the files in SaveGames they will be recreated in the next run of the game
- To use your menu and not the provided one, insert the next code in the relevant Game Mode that contains “Create or Load Configurations”. This option will create the base save files that without them the system won’t be functional



- The slot limitation by default is -1 which means that there is no save slot limitation and if we set the value higher the limit will be 1 slot and higher
- “**Always Show New Game Button**” in W_menu will always show the new game button if it’s **True** in the main menu or hide the new game button after the first game save in the main menu if it’s **False** and the result can be shown in the images. To edit the value we can go to our Menu game mode called BP_Menu_GM or copy this piece of code into your game mode to take effect



- A short delay in BP_Menu_GM was needed to load settings so it's necessary!
- **Good to know:** The visual part of the provided menu is modular and can be changed to your desired style/view or integrated with additional buttons and functionality