Easy Online System Multiplayer

EOSM is small yet easily modifiable template. Purpose of this template/Boiler Plate is to provide any developer good starting point and easy to modify code to their needs. Having to right all new code again and again lead to problem and time it take to right code that should be basic need of game.

This pack include the Basic Online System where you can login to the Epic service (only epic for now).

I have made widget system that use level name to get and set all the widgets so all the widget can be maintain in one place and called automatically.

Note: This is my way of doing thing and all blueprint are independent so widget and online system can be delete or use separately.

Basic needs:

- You will need epic games account.
- Must be Logged in to Epic Online Service and set up product in there.
- Active internet connection.
- Advanced Sessions Plugin (open source on GitHub).
- EOS Plugin to be enable(All plugin with EOS name).
- Replicated Graph Plugin to be Enable (For bulk data to be formatted in graph and easy to handle packages. Good for big player base games).
- Fill out eos Details in Plugin panal.

All the Logic is in Game Instance Blueprint "GI_EOSM" that control the online system and All the Game UI/HUD is control from the Widget "Main_HUD".

Advance sessions plugin is must to have so C++ can be skipped and only blueprints can be use. It provide with many great new function that can be use with stream too.

GI_EOSM:

This blueprint is baseline of all the online system and is alive as long as the game/instance is running. So your login details and all the online system is available to you in any blueprint in editor.

All the event in this blueprint are explained here.

Event init: Init is use only for when game is launch so the auto login system can be Fire. This use the already save login data (*not given publically) to login.
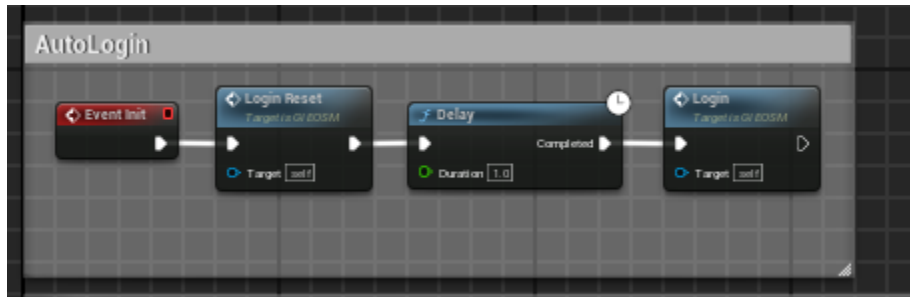

*This data can be extracted if need (Online > Advance Identity > "Any filed you need").

Event Login: Login use Two Method to login to epic service first is to auto Login/Persistence and second is Web Portal Login. If Old login data can be found then Persistence will auto login user to that epic account and if not then web portal will be open and game will be listening to that authentication call to verify login. If login is verify by the epic service then level "Main_manu_" will be opened.
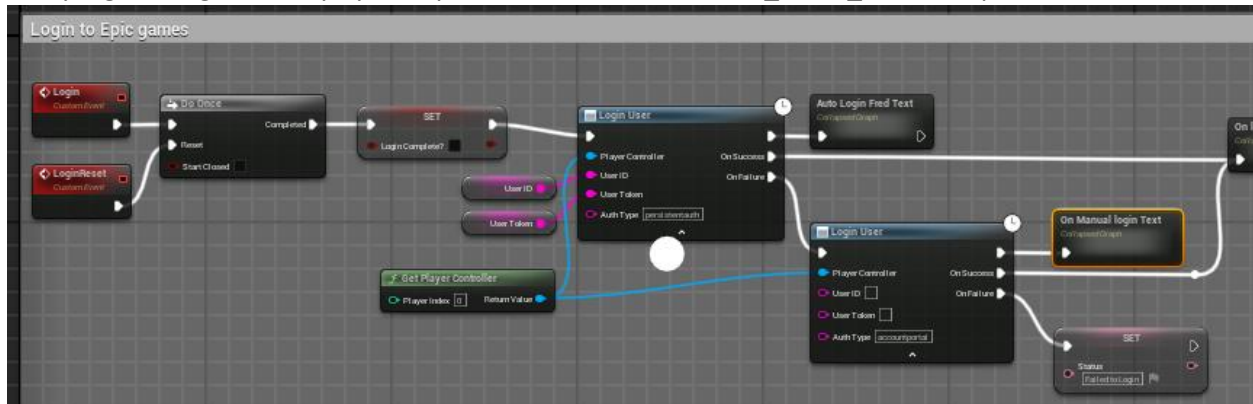

Event Create Session: This event make new Sessions all the Sessions details/Settings can be adjusted from the blueprint node "Create Advance Session".
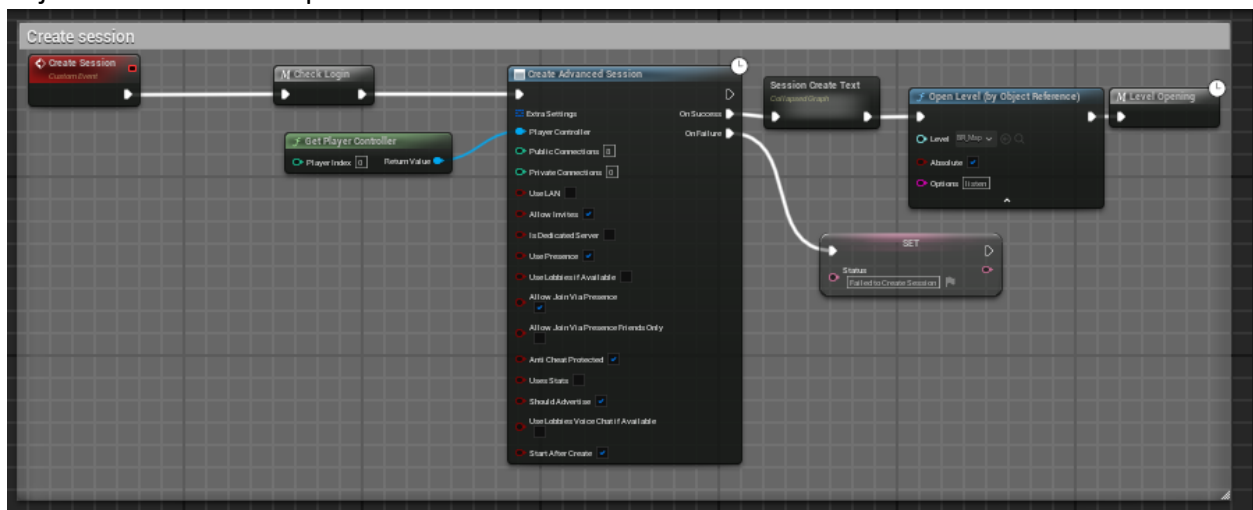
Event Join Session: This event let use to join the session. Ten sessions are set in max results which will find max ten session before joining one of them. "Find Session Advance" Node provide all the function/options needs to adjust session search to your needs. Then "Join Session" Node will join then last found session and travel to the world.
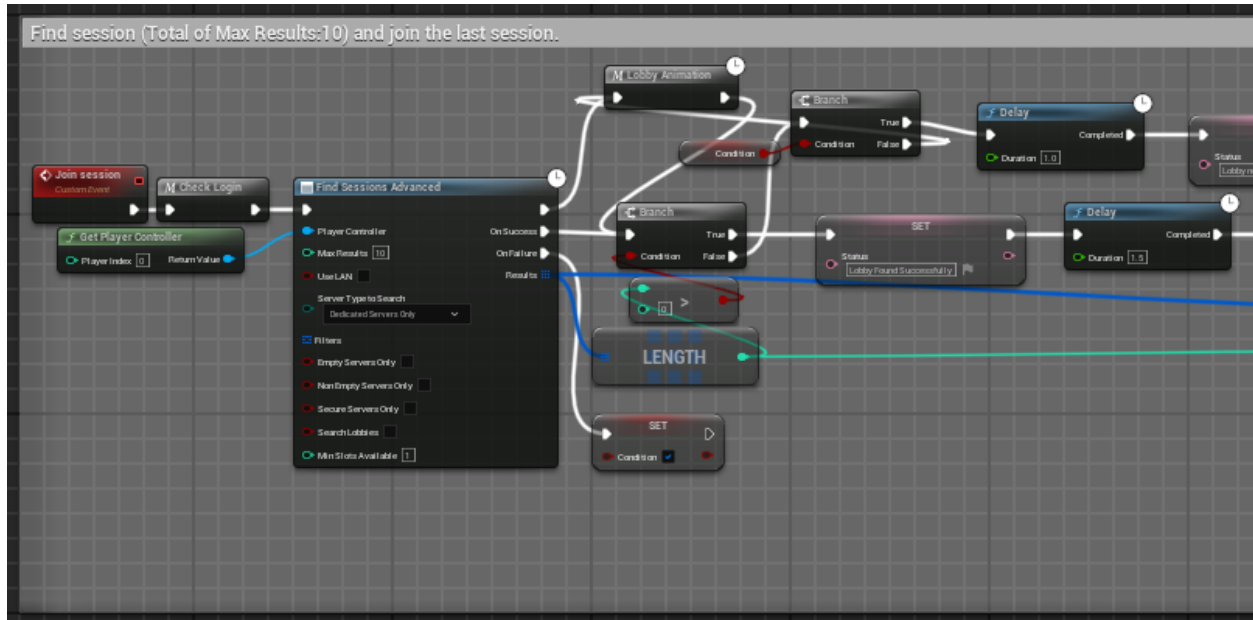

Event End Session: This end any session you are on and open level "Main_Menu_".
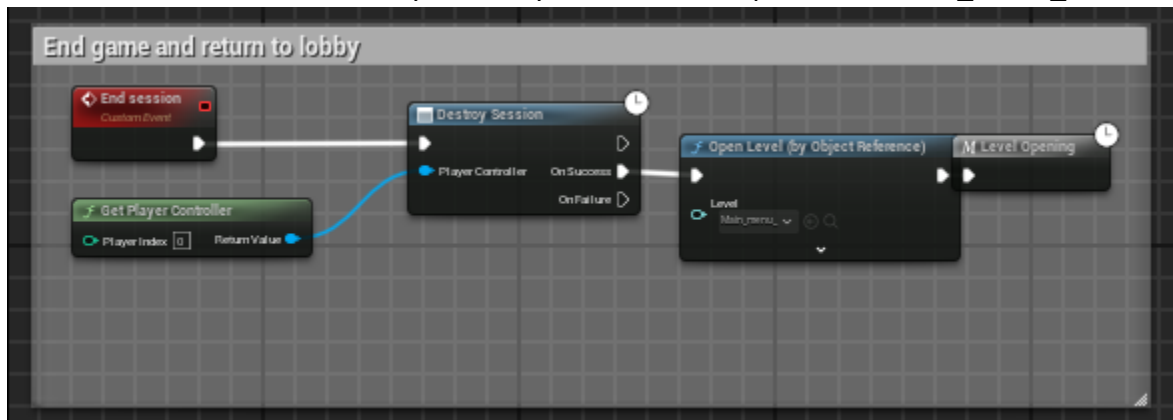
Event Set User Data: If you need to have User id and User Token be saved Call this function and it will be save as String. Then this info can be fed to the login system or user to identify user for block/invite.
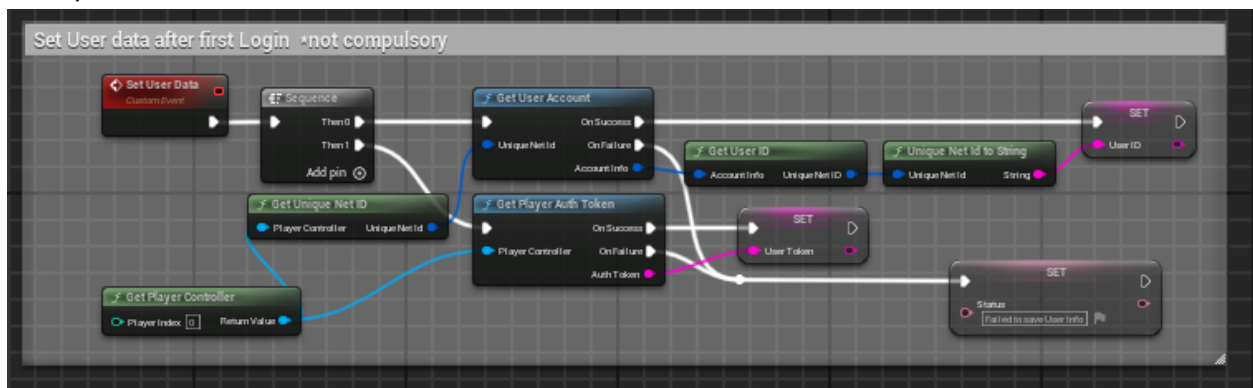

Event Logout: Logout will delete current user info and open Level "Login" So new User can login to the game.
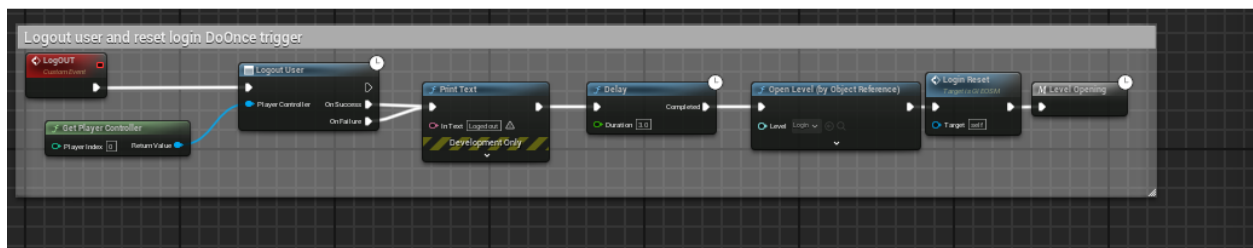
Macro Lobby Animation: This macro use delays to make motion text

Macro Level Opening: This Macro use 5 second delay and set "Level Transition?" Boolean to true and then false so widget can be updated as by new level name.

Macro Check Login: This macro verify that user is login and calling the create/join session so success call be made. Otherwise it will logout or simple open login page again.
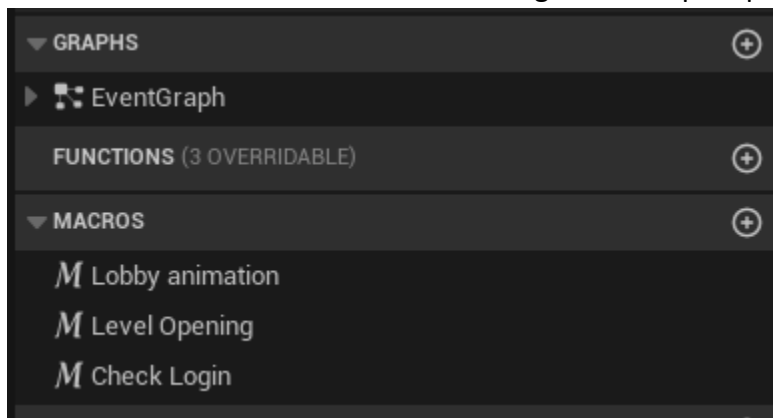
Main_HUD:

This Blueprint Control all the widget that are been shown on the screen at any given time of game/instance lifecycle. This use name and number system where level name verify and then Branch node is use to select the correct widget to show. This provide one place to handle all the ui things now new level can be added and only update is going to be to add that last level name in branch node in last of all branches and it will work as intended. As shown in image below.