Desktop Unity Net.Rumble Demo Game with PlayFab Party and Multiplayer

(GDK UWP with Xbox Live login variant)

*\* This sample has been developed with Unity 2020.3.24f1 and the PlayFab Unity SDKs on Windows.*

# Description

This is a simple multiplayer game. It is expected that developers should be familiar with the basics of using PlayFab Unity SDKs and PlayFab API as well as creating and configuring Xbox Live game titles. They also need to create a PlayFab game title for this project in PlayFab developer portal, and know its Title ID. Additionally, they also need to create an Xbox Live game title for this project in Xbox Partner Dev Center, and know its configuration parameters: XBL Title ID, StoreID, SCID, Package Identity Name, AppID, etc. They will need to configure their Xbox Live game title to use a Sandbox accessible by the Xbox user accounts they intend to use with this app. This sample app is intended to demonstrate how developers can use PlayFab Unity API and Microsoft GDK Unity API to perform the following functions:

* Login to Xbox Live using an Xbox user account.
* Login to the PlayFab server using Xbox Live authentication token.
* Get the avatar, username, and user ID of the Xbox login user.
* Find game lobbies, join game lobbies, create game lobbies, customize lobby attributes, and use game matchmaking using PlayFab Multiplayer Unity API.
* Create a Party network, have players join the Party network for message/data exchange and voice chat using PlayFab Party Unity API.
* After joining the lobby, display player names, other player information, update player information, synchronize information of players in the lobby, and change ship color, ship type.
* After joining the lobby, start a multiplayer game and Party voice chat.

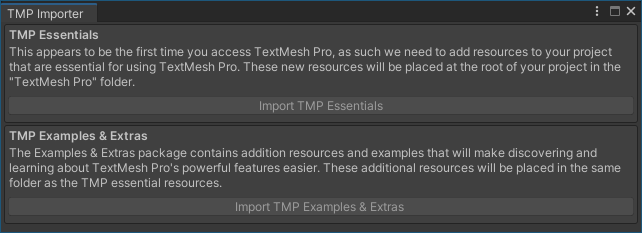
In this sample app, PlayFab Unity API is used to log in, find the game lobby, join the game lobby, create the game lobby, customize the lobby attributes, match the game, send and receive messages from lobby users, provide voice chat with lobby users and communicate with lobby users in real time. The script file path of the game code making calls into PlayFab Unity API is: Assets\Sample\Scripts\PlayFabLogic.

Xbox Live user login and token management, user profile picture and name are provided by GDK Unity API. The script file path of the game code making calls into GDK Unity API is: Assets\Sample\Scripts\XboxLive.

# Building the Sample

This example depends on the PlayFab Unity SDKs and GDK Unity SDK. Developers need to download the PlayFab Unity SDKs and GDK Unity SDK and import them into Unity to satisfy the SDK dependencies required by the example.

## Developers need to download and import the following SDK

* PlayFab Party Unity SDK download link: [PlayFab/PlayFabPartyUnity (github.com)](https://github.com/PlayFab/PlayFabPartyUnity) (release 1.7.0.0-main.0 was used at the moment of writing this document)
* PlayFab Multiplayer Unity SDK download link: [PlayFab/PlayFabMultiplayerUnity: PlayFab Multiplayer Unity SDK (github.com)](https://github.com/PlayFab/PlayFabMultiplayerUnity) (release 1.1.0.0-main.0 was used at the moment of writing this document)
* PlayFabEditorExtensions Unity plugin download link: <https://aka.ms/PlayFabUnityEdEx>
* GDK Unity SDK (also known as GDK Unity Package) download link: [microsoft/gdk-unity-package: The Unity package for PC GDK developers. (github.com)](https://github.com/microsoft/gdk-unity-package) (release 2104.0.20220209 (gdk-pc.2104.0.20220209.unitypackage) was used at the moment of writing this document)
* In opened Unity project, navigate to Assets/Import Package/Custom Package in the main menu to import the GDK Unity SDK package.
* To use the GDK Unity SDK, make sure that macro symbols "USE\_MS\_GAMECORE" and "USE\_UNITY\_GAMECORE" are defined in Player à Other Settings à Script Compilation à Scripting Define Symbols fields of Project Settings screen.
* Make sure that “Allow ‘unsafe’ Code”is enabled (checked) in Player à Other Settings à Script Compilation section of Project Settings screen.
* The example has only one "SampleScene.unity" scene and the path is: "Assets/Sample/Scenes".
* Upon opening the scene, you may be prompted to import TextMesh Pro resources in a popup window, please import both essentials and extras:  
  
* Please check whether the PlayFabEnvInitializer object exists in SampleScene. This is a prefab, add it to the scene if necessary. Please check its Inspector panel if the script is mounted, to see if the PlayFabMultiPlayerManagerPrefab and PlayfabMultiPlayerEventProcessorPrefab of the PlayFabEnvInitializer script are empty. If they are empty, please add them.

PlayFabMultiPlayerManagerPrefab local path： Assets\PlayFabPartySDK\Prefabs\PlayFabMultiPlayerManagerPrefab.prefab

PlayfabMultiPlayerEventProcessorPrefab local path：Assets\PlayFabMultiplayerSDK\Prefabs\PlayfabMultiplayerEventProcessor.prefab

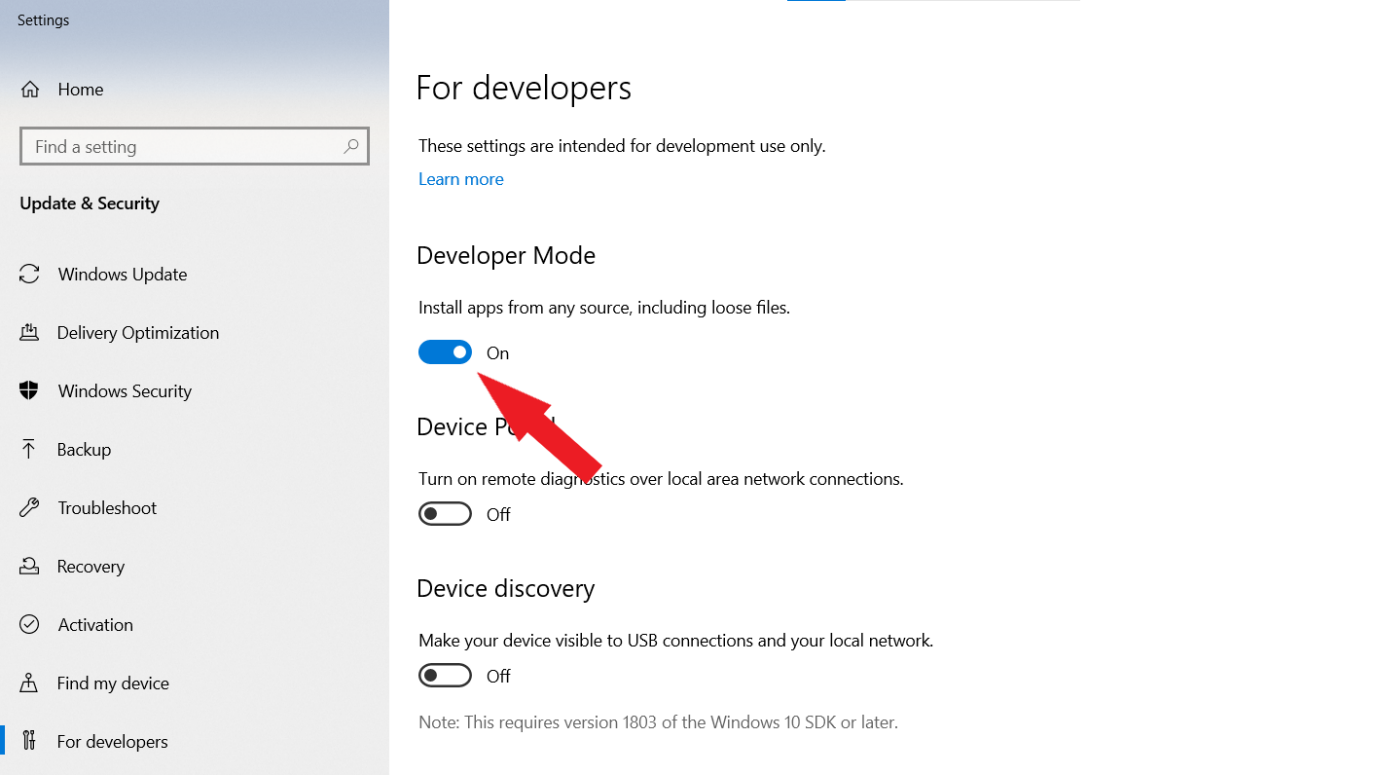
* Open the PlayFabEnvInitializer script, modify TITLE\_ID to set your PlayFab Title ID. PlayFabEnvInitializer local path: Assets\Sample\Scripts\PlayFabLogic\PlayFabEnvInitializer.cs

Install GDK and configure the runtime environment

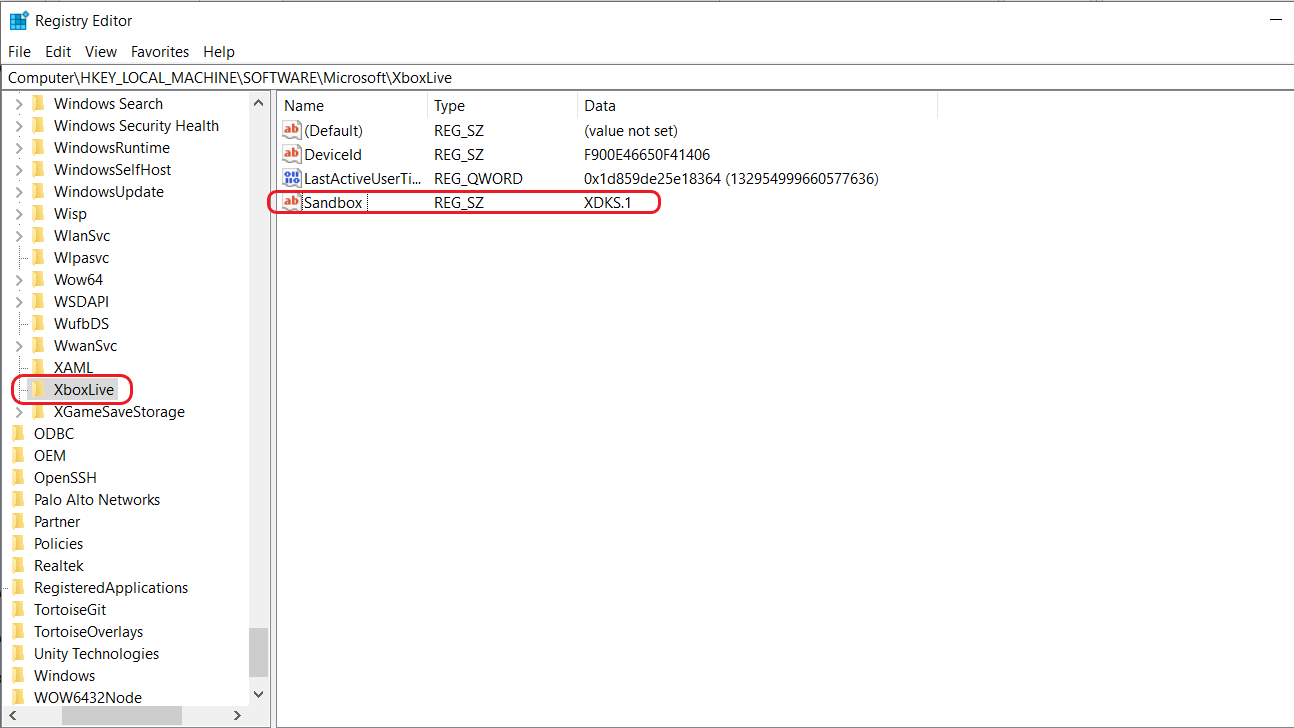
Download a recent version of Microsoft GDK (for example, **2021.04 – 10.0.19041.6078** at the moment of writing this document) from Microsoft GDK portal (<https://aka.ms/gdkdl>). Install the downloaded GDK. Make sure to select all optional components during installation. It will install the GDK and necessary VS build tools.

Before launching and testing the application, you need to configure your PC environment：

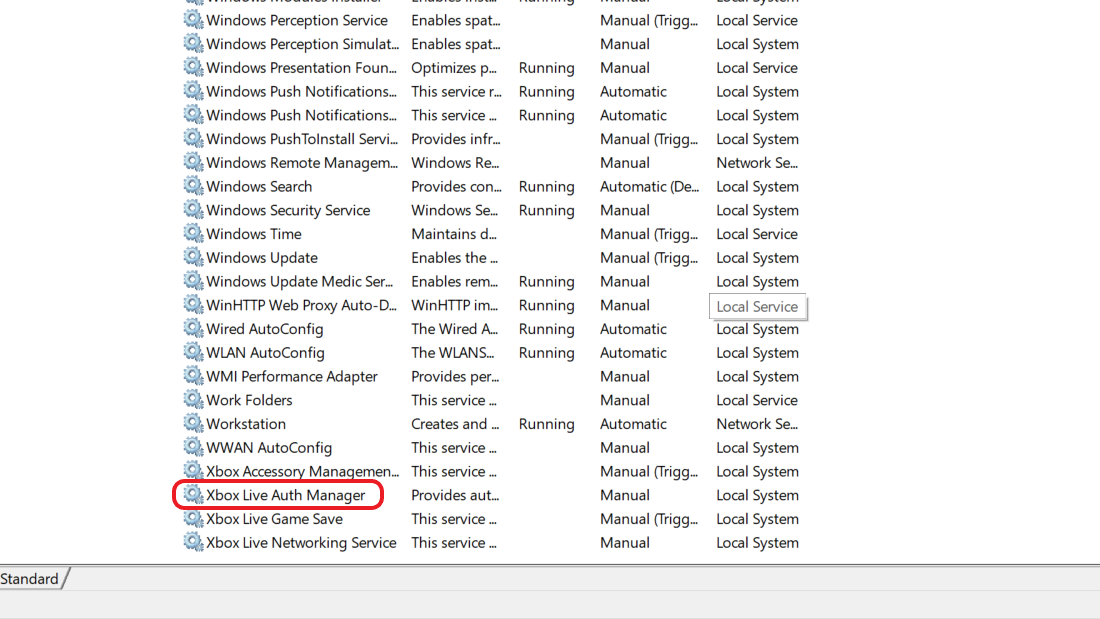
1. Make sure your Windows PC is set to Developer Mode：



1. Make sure your Windows PC is set to the right Xbox Live Sandbox depending on the configuration of your Xbox Live game title. For example, let’s assume it is“**XDKS.1**”just for reference in the following steps:
   1. Create/Modify the registry setting Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\XboxLive\Sandbox if needed (string value):



* 1. You may restart this Windows service after updating Sandbox to make it take effect (an alternative option is to reboot your PC) :



# Configure MicrosoftGame.config

The file GdkMetadata\MicrosoftGame.Config needs to be updated with configuration information specific to your Xbox Live game title from Xbox Partner Dev Center before building, installing, and running the app.

It is expected that a developer should be familiar with the process of building and configuring applications for Microsoft Game Core (GDK). Please refer to corresponding documentation or sample code as necessary.

On the screenshot below blue stickers indicate the data that needs to be provided:

A picture containing graphical user interface

Description automatically generated

Identity Name: Package Identity Name from Xbox Live configuration.

Identity Publisher: Package Identity Publisher from Xbox Live configuration.

StoreId, TitleId and MSAAppId are parameters from Xbox Live configuration.

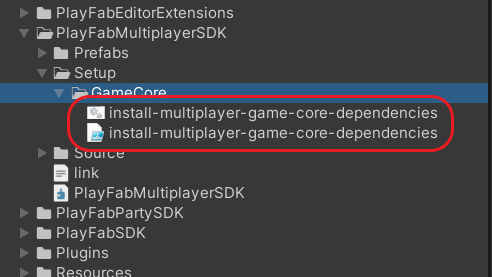
Executable Name and DefaultDisplayName should typically be the same, this is the application name and the name of game’s executable file (without the extension).

PublisherDisplayName: Publisher Display Name from Xbox Live configuration.

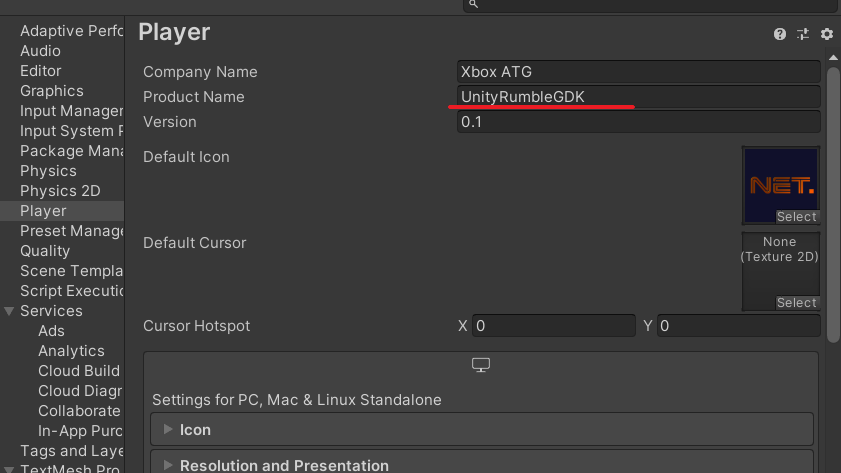
Please, also remember to specify a required value in SandboxIds, it should match the Sandbox(es) in Xbox Live configuration.

# Start Build and Install

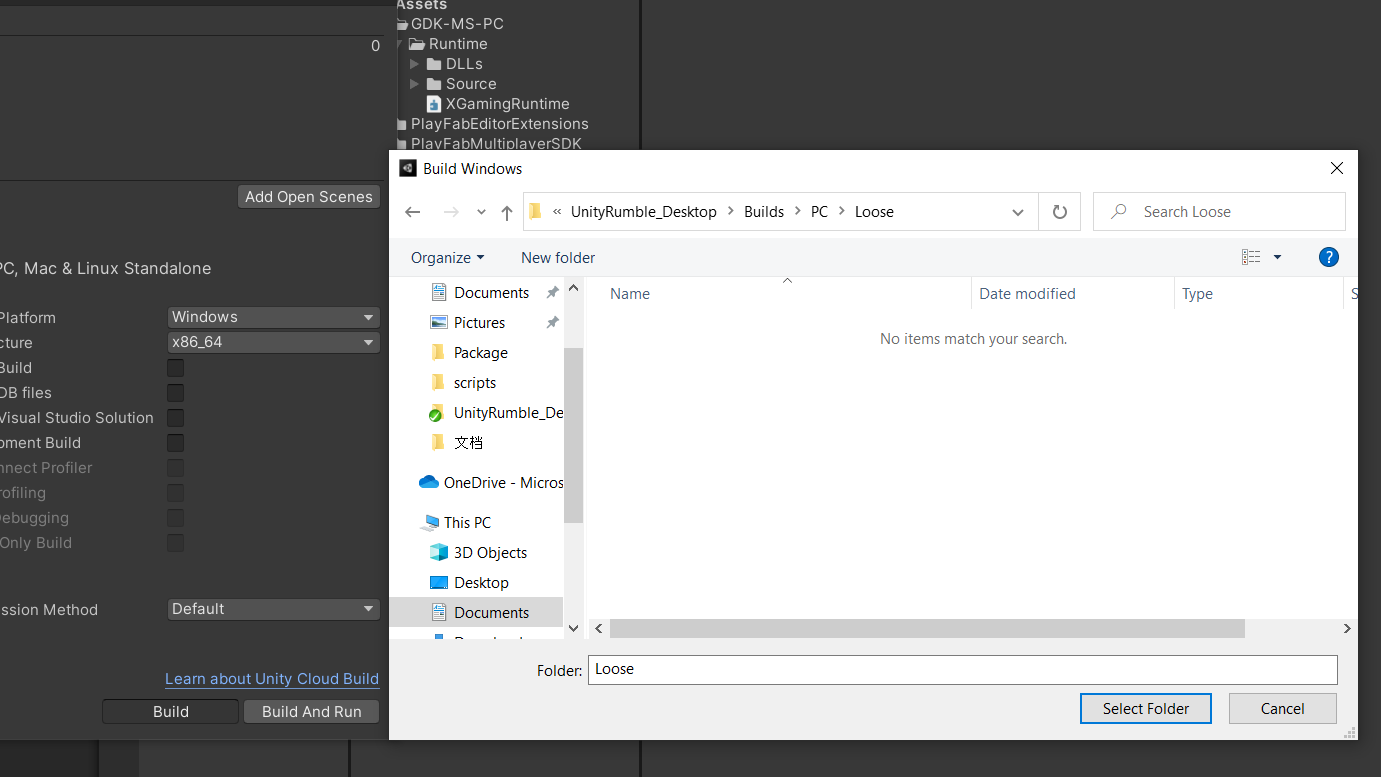
Run “install-multiplayer-game-core-dependencies.cmd” and “install-multiplayer-game-core-dependencies.ps1”. They may copy additional files to Unity project:



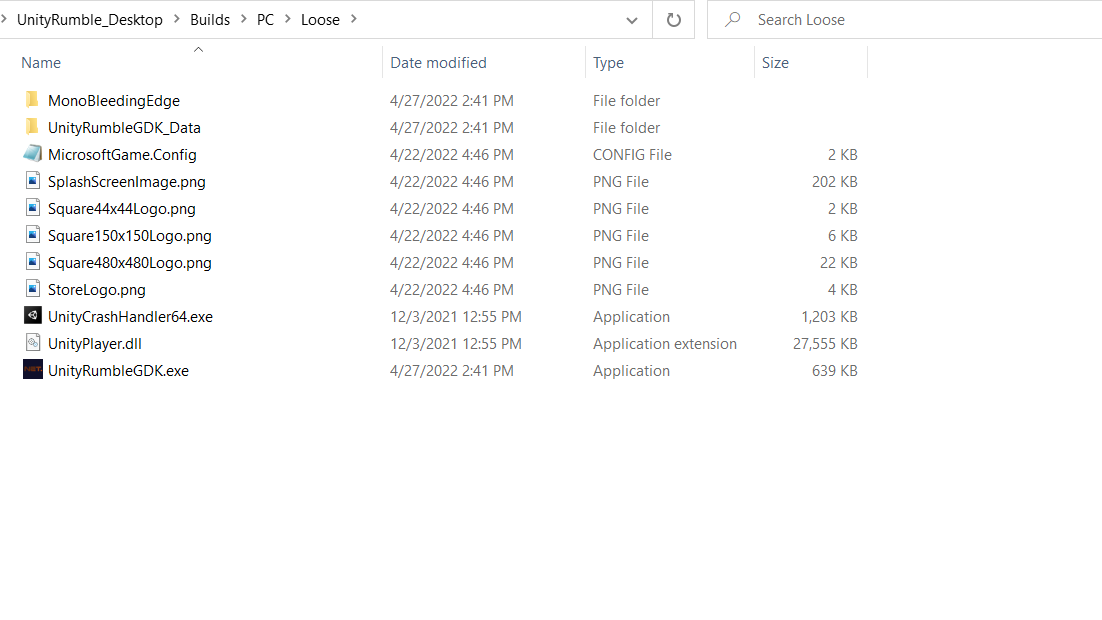
Set Company Name and Product Name that match corresponding values in your modified MicrosoftGame.config file. Company Name can be set to the same value as PublisherDisplayName, and Product Name should be set to the same value as Executable Name (without the extension):



All of the Unity, Xbox Live, and PlayFab configurations required to build the sample should already be set once all of the SDKs have been unpacked, and the sample is loaded into Unity. The sample should be built within the Unity IDE using the “Build” button:



This sample currently provides a “Builds” folder and expects the build output to go into a folder path “Builds/PC/Loose”, like so：



Please copy the“MicrosoftGame.config” file and all other files (PNG images) from the“GdkMetadata” folder to the “Builds/PC/Loose”folder. These are supporting files needed by a GDK application.

Once Unity has completed its build process, and the metadata files copied to the build output folder, “side load” the built GDK application into Windows: using the “Gaming VS 2019 Command Prompt” one can execute the “wdapp register Builds\PC\Loose” command from the sample root folder. If necessary, please refer to corresponding GDK documentation on the information about how to build and run GDK applications on a PC.

# Running the sample

When you successfully build and register/install the app onto your developer Windows PC, you should see an app icon with your application name within the Windows Start menu. Clicking on the app icon should launch the sample app.

It is important to remember that the PC should be running in the right Xbox Live sandbox that the title has been configured for. A GDK tool named “XBLPCSandbox” (should run within “Gaming VS 2019 Command Prompt”) -- can assist with switching the active sandbox. You will also need Xbox Live test user accounts who can login to the sandbox.

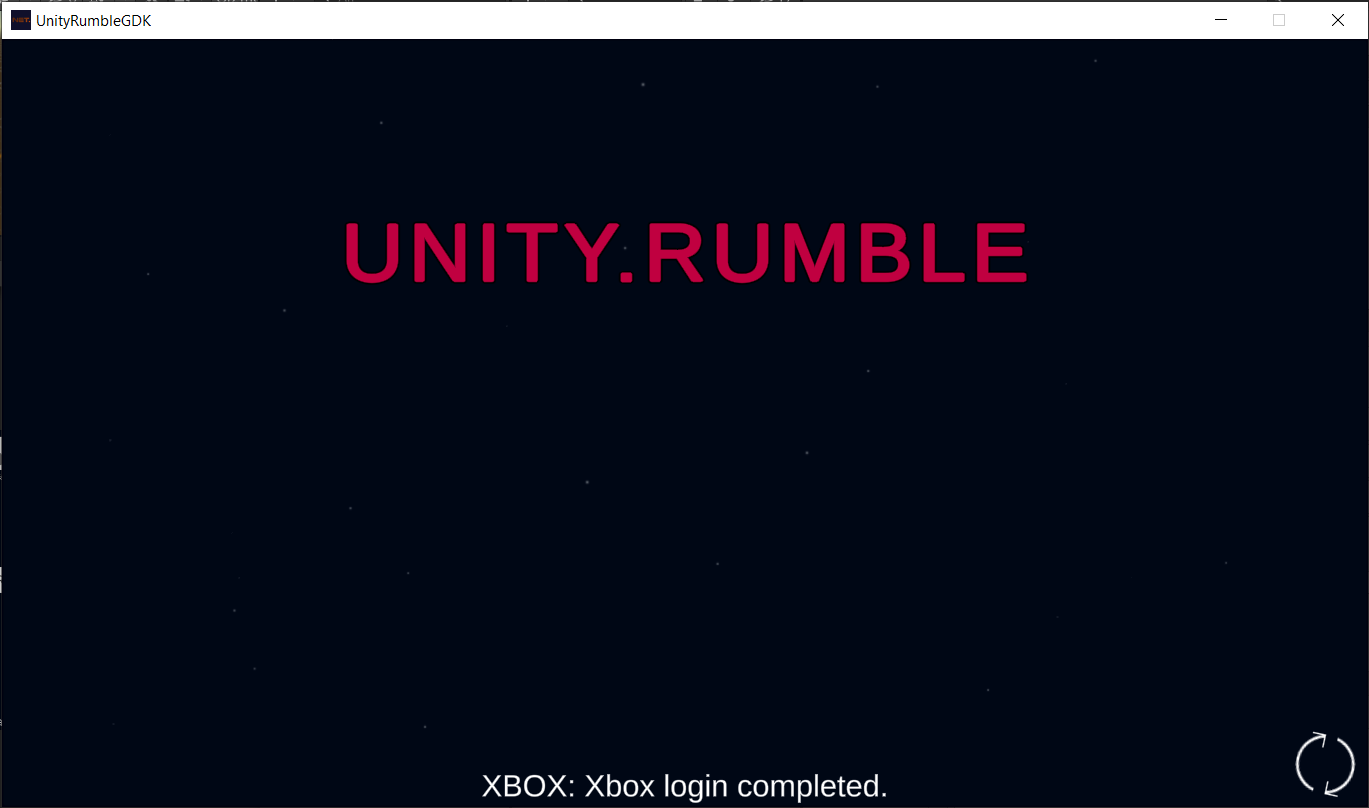
# Sample Start Screen



The screenshot shows the first interface when the sample is launched. Click the "Start" button to log into the PlayFab server. If any of the login steps fail, a failure message is displayed at the bottom of the screen and the screen remains on the start screen.

Common failure scenarios may be that users are unable to access their current Xbox account, their Internet connection is disconnected, or PlayFab is experiencing some form of service disruption.

# Login to PlayFab with Xbox



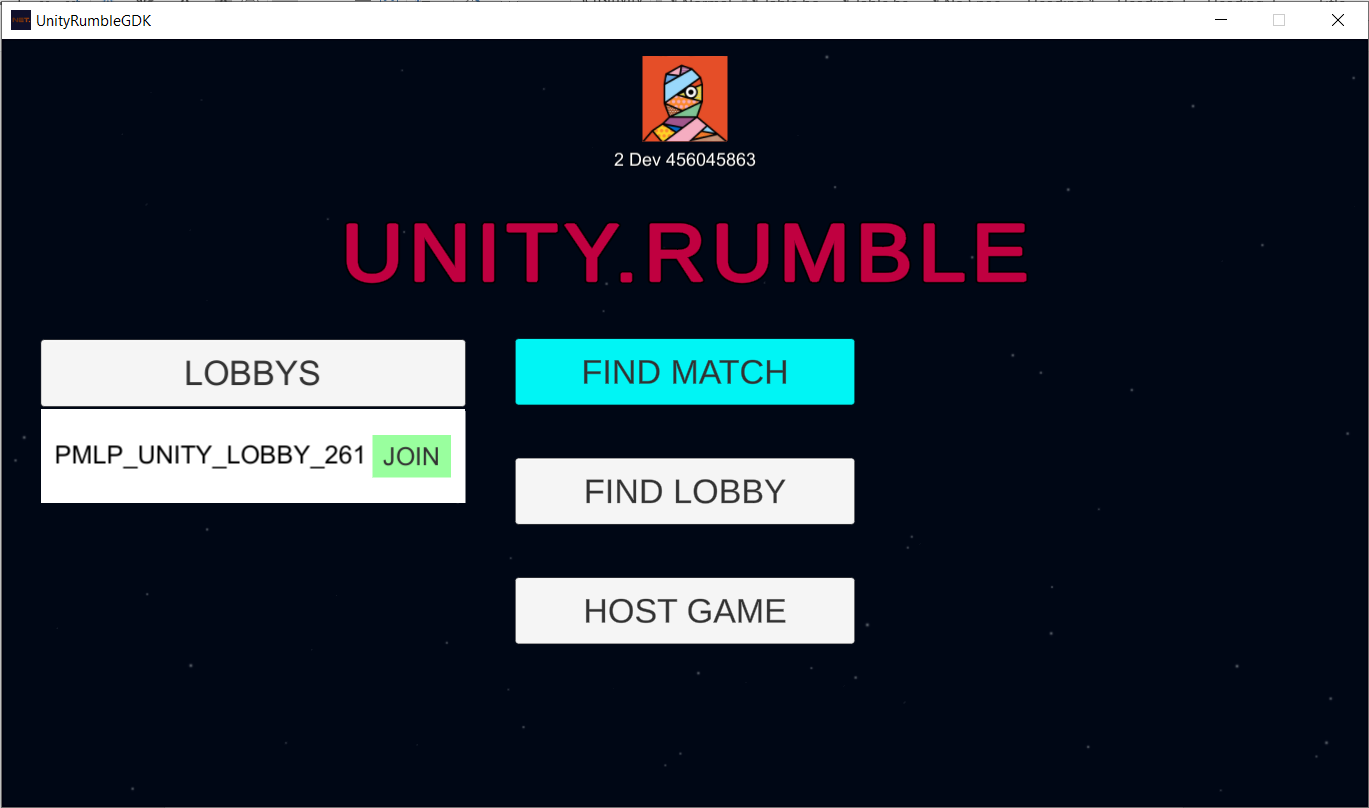
The screenshot above shows the next screen that appears when the sample launches. PlayFab user login process that is ongoing after clicking the start button. The bottom of the screen shows that you are logging in.

# Sample Main Menu Screen

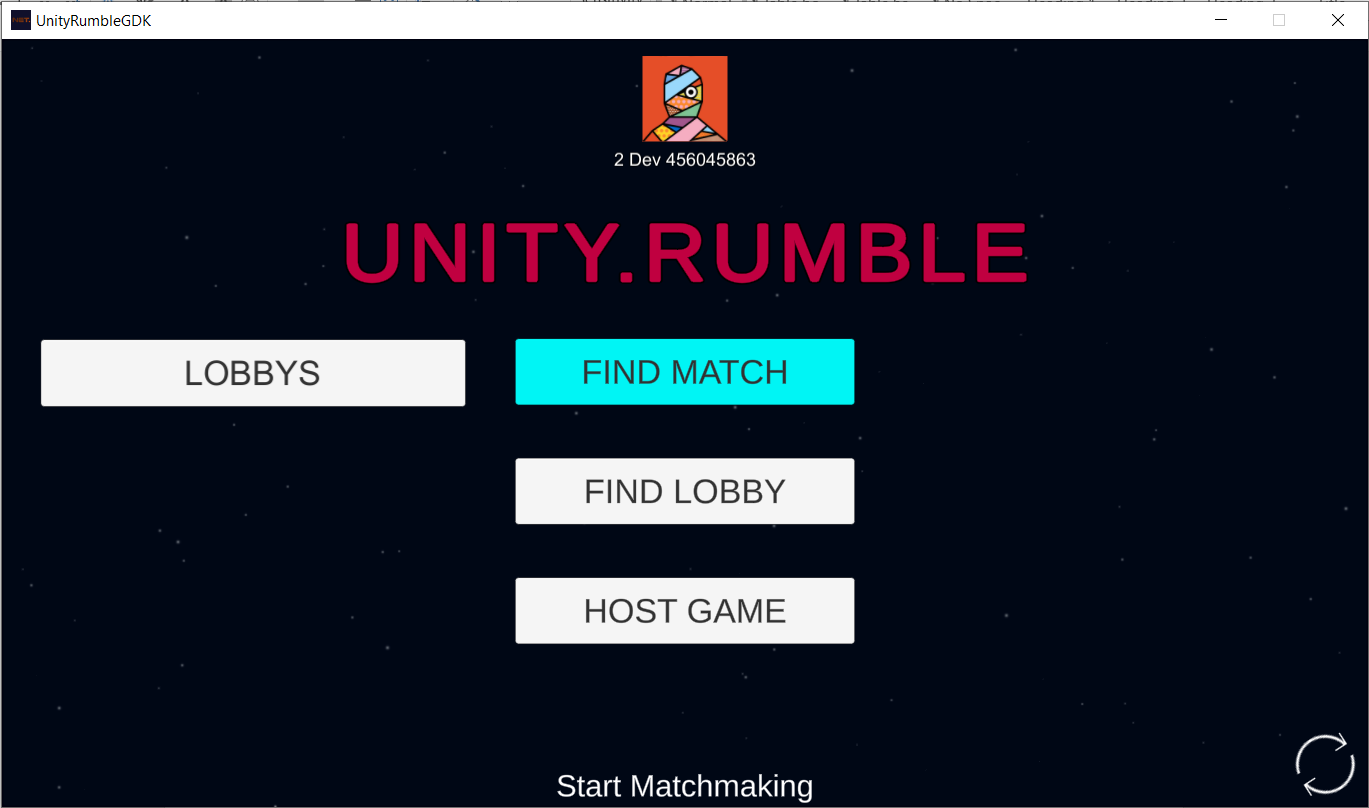
When the user successfully logs into PlayFab, the Main Menu screen for the sample is displayed. The four main features provided are:

1. Click the "FIND MATCH" button to start matching the game.
2. Click the "HOST GAME" button to create a game lobby which will make the lobby available for other players to find and join it.
3. Click the "FIND LOBBY" button to display all the available game lobbies. After a lobby has been found all available lobbies will be displayed under "LOBBIES"

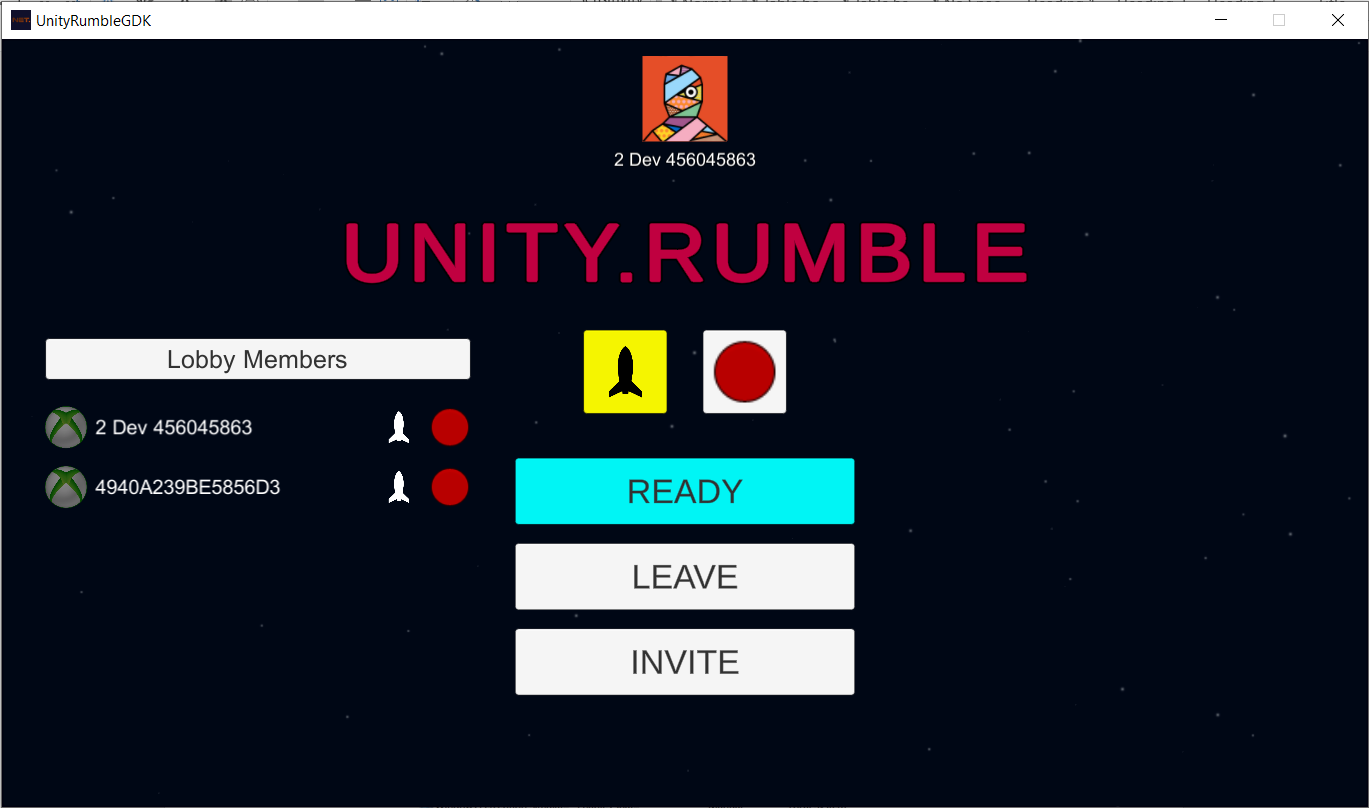
If any of the above functions fail, the error message is displayed at the bottom of the screen and the user is left on the Main Menu screen：



# Find Match Screen

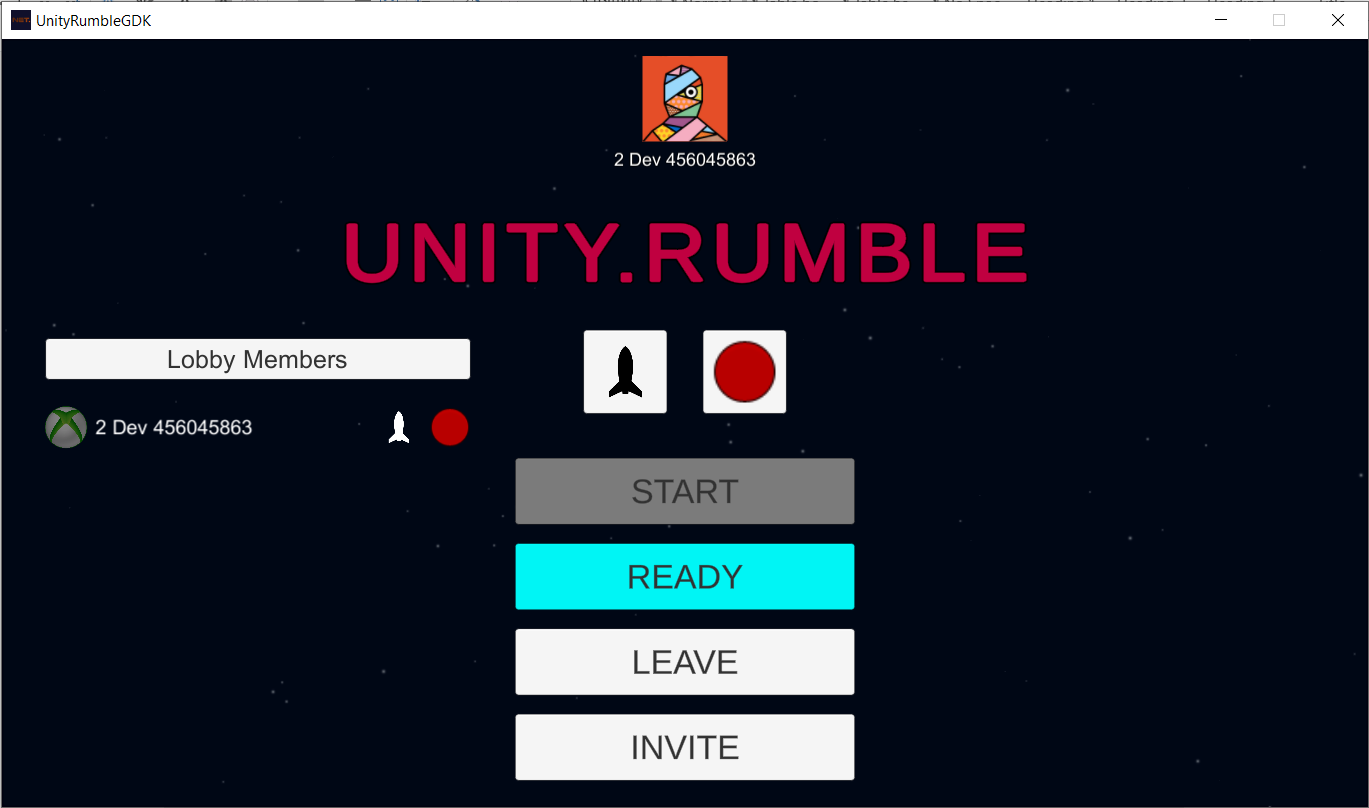
When the "FIND MATCH" button is clicked, the player starts matching the game, as shown below:

When the player successfully matches the game, the lobby screen opens, as shown below:



# The Game Lobby Screen

Click the "HOST GAME" button and PlayFab will create a Lobby for the player, which can be found by other players. Once the lobby is created, the GAME Lobby Screen will open, as shown below:



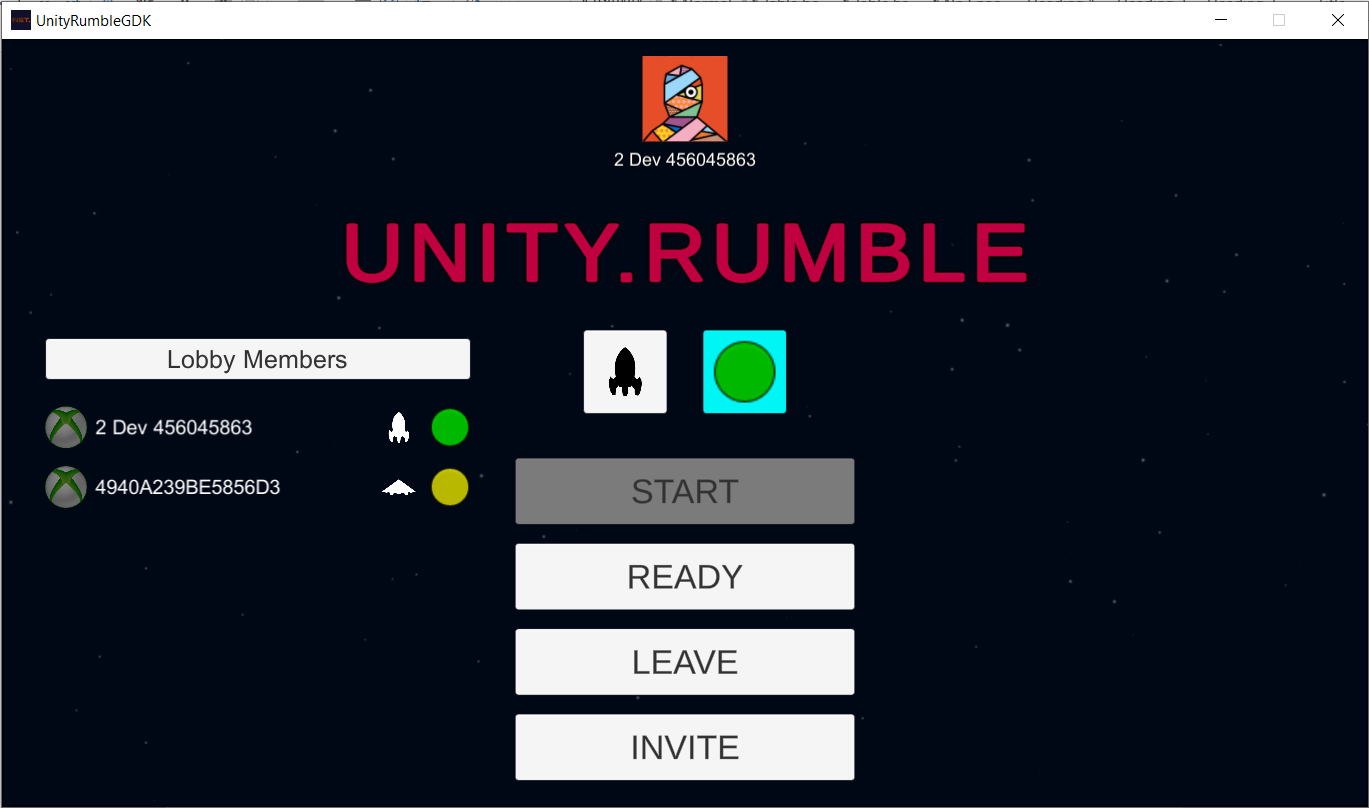
# The Ship Style and Color

Lobby members can change the style and color of their ships by clicking the "Ship" button and the “Color” button.

Ship Button：

Color Button：

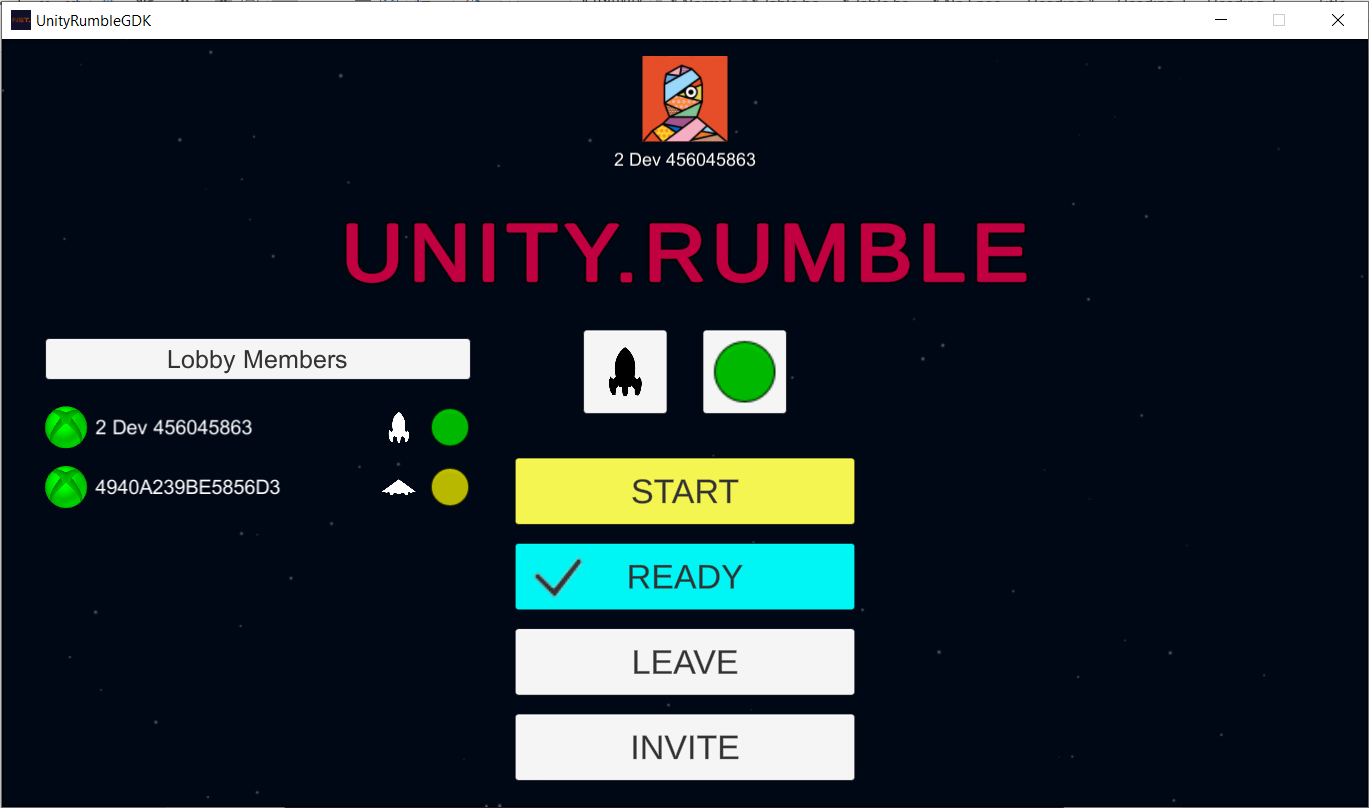
This example has four different ship styles and six different ship colors. The following picture shows the interface after clicking the ship button and the color button:



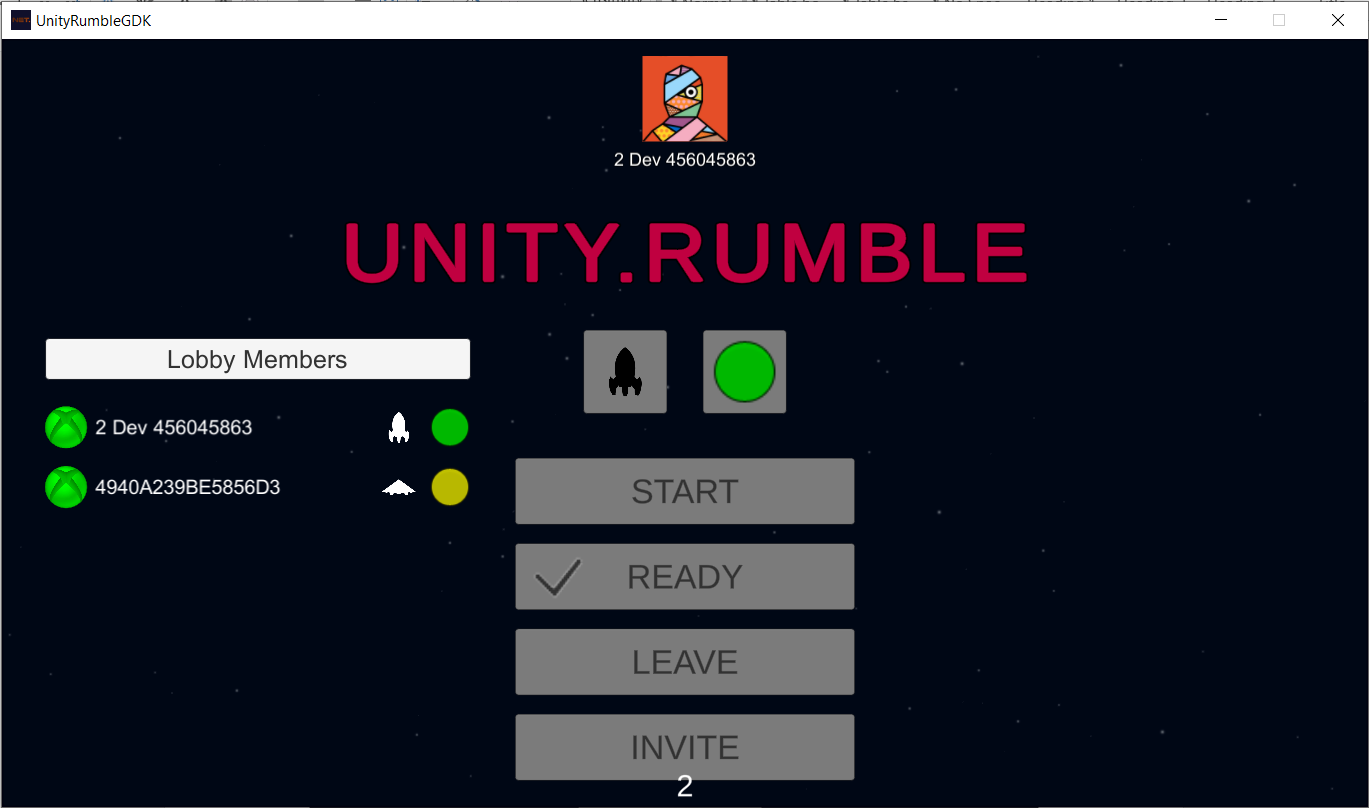
# The Lobby All Ready

When the members of the Lobby are ready to begin playing, they can change their state of ready by clicking the "Ready" button will synchronize the state with other members of the Lobby. When all players are ready, the player who created the Lobby can click the "Start" button to start the game and notify all members of the lobby that the game is beginning. All members of the lobby will see a countdown at the bottom of the screen indicating that the game is about to begin. The ready status is displayed before each username. Gray indicates that the user is not ready, and green indicates that the user is ready.

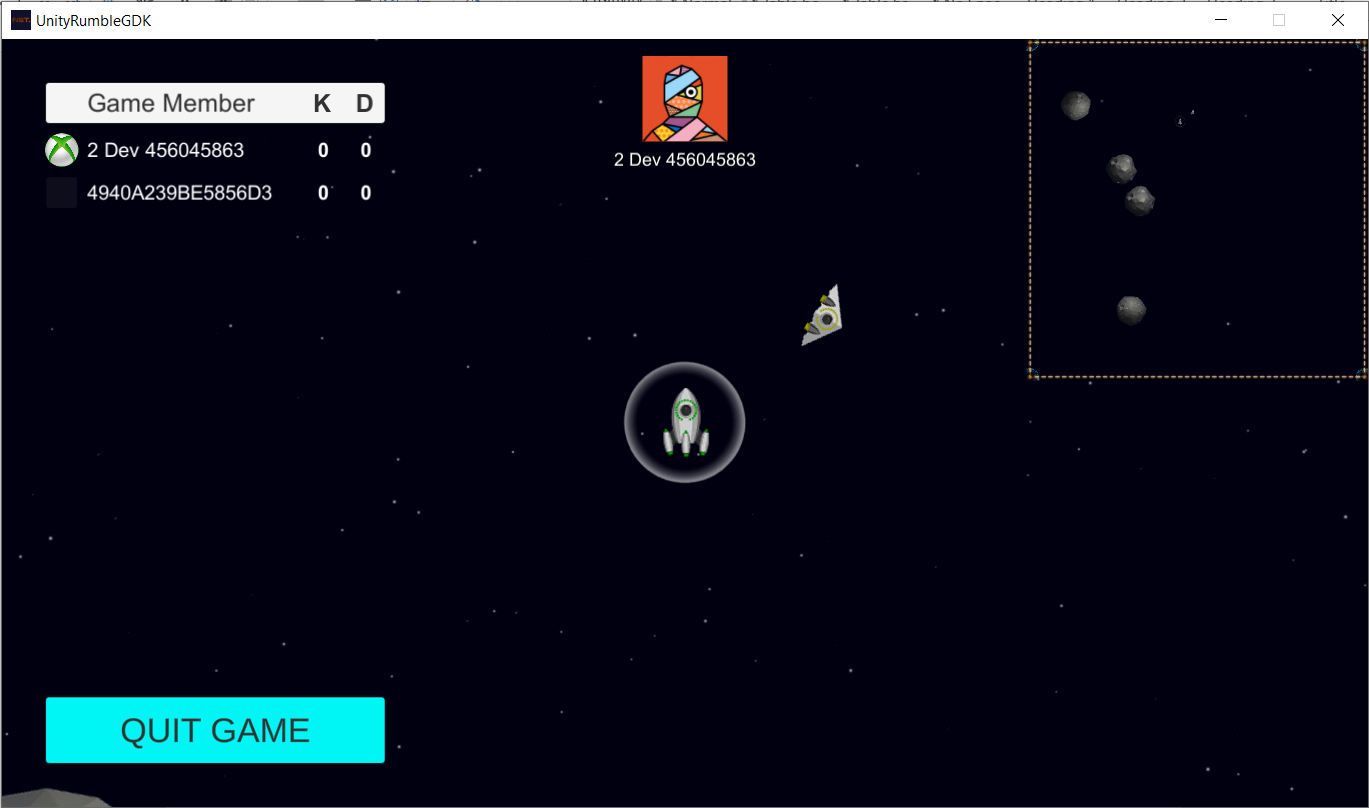
The following picture shows all users ready to play:



The picture below shows the countdown to start the game：

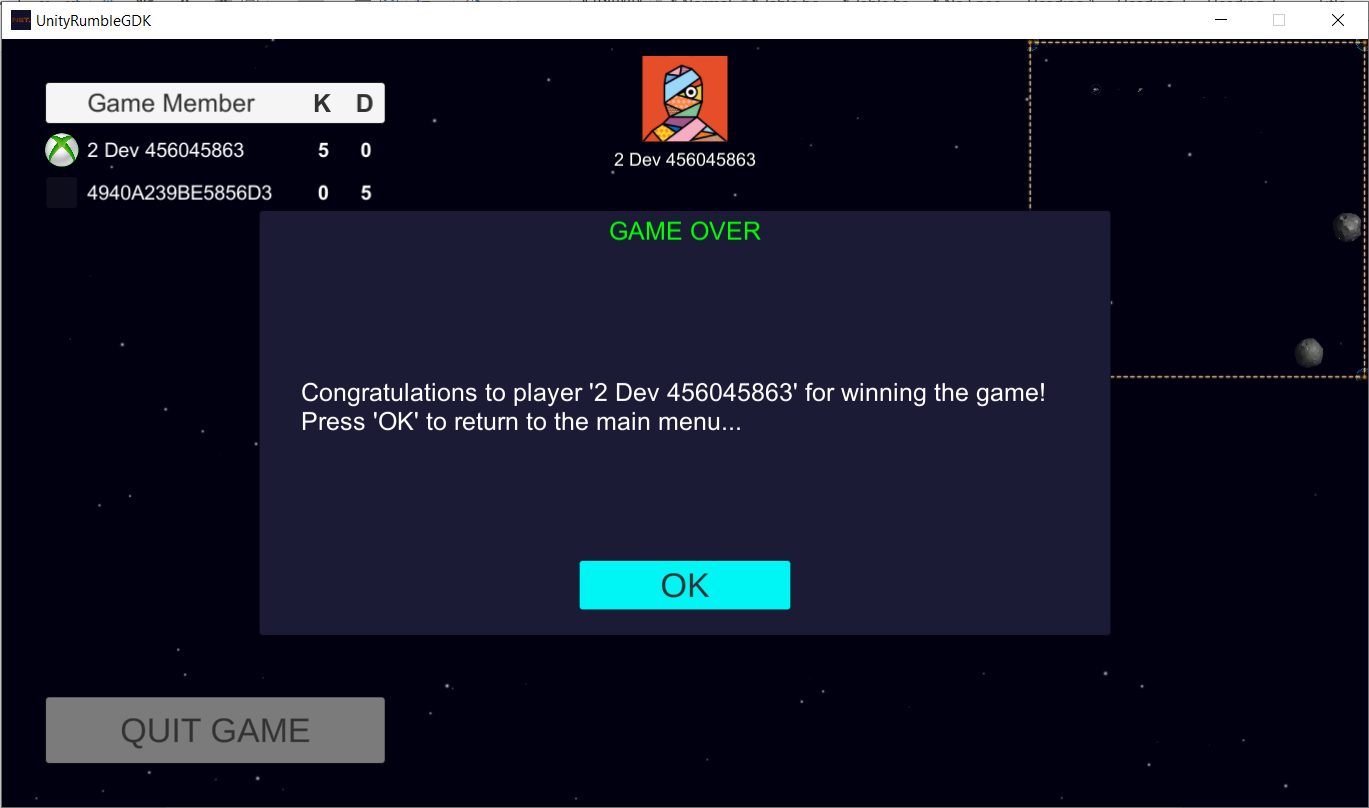


# The Game Play Screen

 The upper left corner of the Game Play screen shows the current game's player, with username and "kill" and "death" statistics.

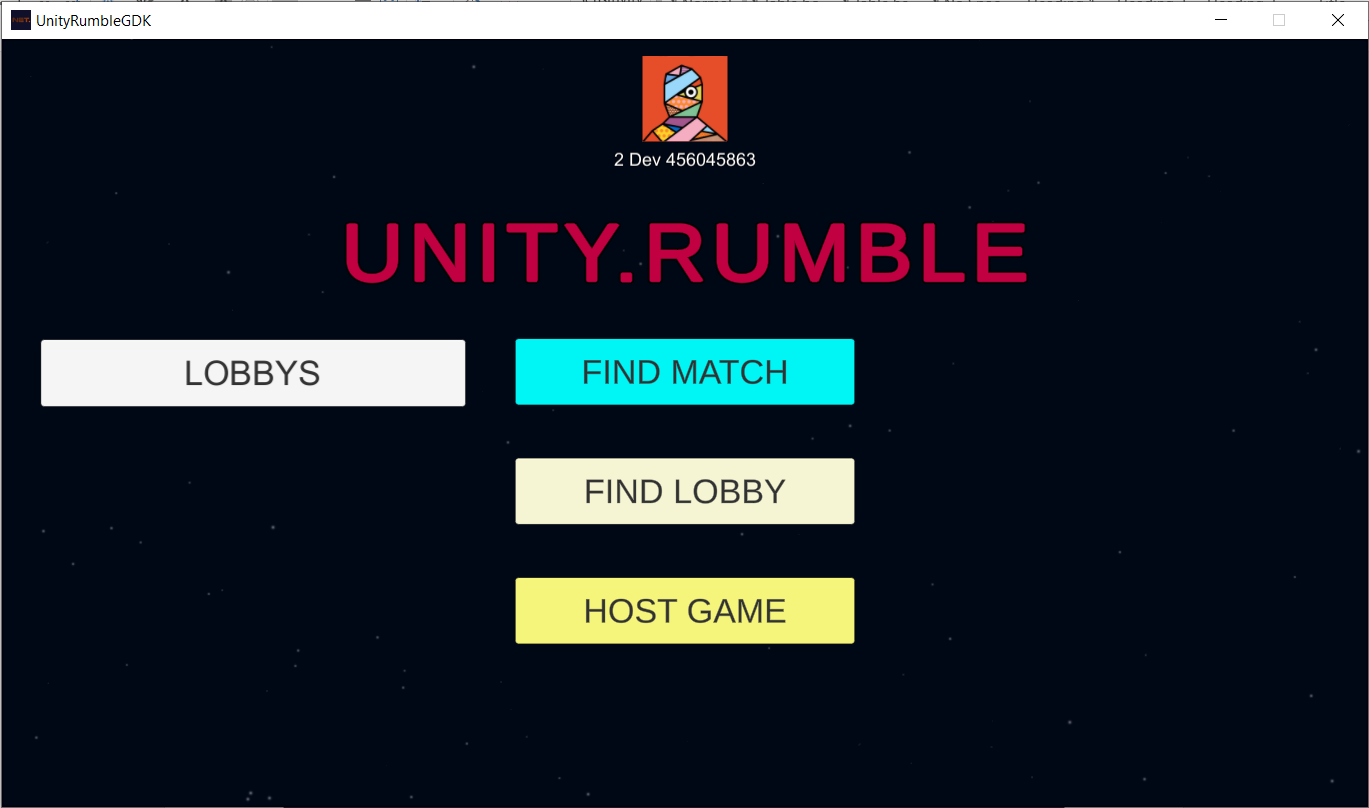
The Xbox icon to the left of the host's name indicates whether it is the owner. If a user clicks the "Quit Game" button to quit the game, he will disappear from other player's player lists. Players can make real-time voice calls after joining the game.

The game ends when a player has a kill score of 5. At this point, the game end screen will pop up, and click the "OK" button to return to the Main Menu screen, as shown below:



# The Profile Screen

Almost all screens display the player information, including their profile picture and username, as shown below:



## Implementation notes

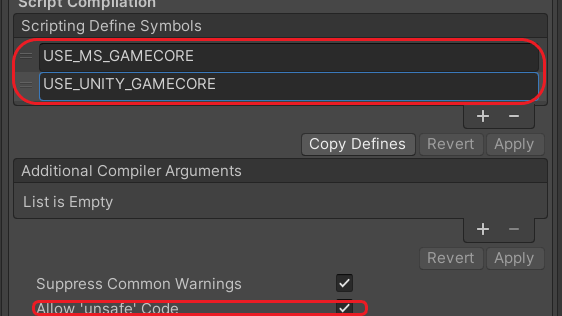
You can find the sample scripts in the “Assets/Sample/Scripts" folder, which contains "UI/View"-related code and core logic.

• Xbox profile management logic is in Assets\Sample\Scripts\Xbox Live

• PlayFab Unity SDKs-based code for Party, Lobby, Matchmaking and voice chat are in Assets\Sample\Scripts\PlayFabLogic

• PlayFab automatically enables voice chat between players when they create a Party network.

•To use the GDK Unity SDK and the PlayFab Unity SDKs, make sure "USE\_MS\_GAMECORE" and " USE\_UNITY\_GAMECORE " are defined in Scripting Define Symbols, as shown in the image below：



## Known issues

This example is based on the latest version of PlayFab Unity SDKs and GDK Unity SDK (GDK Unity Package) at the time of development. Using newer versions or integrating their different versions may cause incompatibilities with the current code.

## Privacy Statement

For more information about Microsoft’s privacy policies in general, see the [Microsoft Privacy Statement – Microsoft privacy](https://privacy.microsoft.com/en-us/privacystatement/)