Conectarnos con Steam, Epic Launcher, Cbox, etc. Ellos lo tienen montado de forma que todo esto sea abstracto (da igual el subsistema que haya por detrás).

IOnlineSubsystem\*: interfaz (puede ser de Steam), de aquí podemos pedirle interfaces.

**IOnlineSubsystem\* sub = Online::GetSubsystem(getWorld())**

Con el subsistema podemos pedirle distintas interfaces:

**Sub->GetIdentity();**

**IOnlineIdentityInterfacePtr**

Manager para crear cosas. Y así juntamos todo el código y evitamos el spaguetti

Source / MCVGame / Managers / MCV\_OnlineManager.h y .cpp

Van a ser subsystem de GameInstance

A screen shot of a computer

Description automatically generated with medium confidence

Podemos ver los subsistemas

A screenshot of a computer

Description automatically generated

subsistemas de Android, Facebook, null(para LAN), oculus, steam,

Esta por ahí también el del GameInstance que es el include que necesitamos.

Y vemos en el .h que tiene las funcione sinitialize y deinitialize

A screen shot of a computer program

Description automatically generated with medium confidence

En el mcvGame.Build tenemos que añadir los susbsistemas

A screen shot of a computer program

Description automatically generated with medium confidence

En el initialize cogemos el IonlineSubsystem

A picture containing text, screenshot, font, line

Description automatically generated

Miramos el .h del OnlineSubssytem y vemos las funciones. Por ejemplo para un sistema de achievements. Usando esa función nos sirve tanto para steam como para otra plataforma. Hoy vamos a usar el de las sessions:

A screenshot of a computer screen

Description automatically generated with low confidence

Cogemos la interfaz y tendrá métodos para gestionar.

A screen shot of a computer program

Description automatically generated with low confidence

Y esto se encarga de hacer el login.

Al hacer login podemos registrarnos al OnLoginDelegate

A screen shot of a computer program

Description automatically generated with low confidence

Si por ejemplo cramos una sesión seria igual, nos registramos al delegate.

Cuando se hace el login llamamos a la función OnLoginComplete, que recibe unos parámetros que tenemos que ver en el .h

A screen shot of a computer code

Description automatically generated with low confidence

Así podemos trabajar con este tipo de interfaces.

Vamos a crear un menú de sesiones para el multiplayer.

UI / OnlineMenu: creamos widget

Crea un botón y en el onclicked

A screenshot of a computer

Description automatically generated with medium confidence

Y si se presiona el FindSessions:

A screen shot of a computer

Description automatically generated with medium confidence

Y para cada sesión añadimos un nuevo widget de tipo WBP\_Session Result

Ponemos dispatcher para unirnos a la sesión cuando clicken el botón (este tipo de botón ira en el FindSessions)

A screenshot of a computer

Description automatically generated with medium confidence

Y nos queda algo así el onlineMenu:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Ya nos funciona para local

Descanso y luego steam para los de casa

Activamos el plugin de steam

A screenshot of a computer

Description automatically generated with medium confidence

En el Game.Build le tenemos que poner una dependencia, Tenemos en Google una documentación para verla:

A screenshot of a computer

Description automatically generated

En el DefaultEngine.ini tenemos que configurar algún valor:

A picture containing text, screenshot, font, graphics

Description automatically generated

Por defecto está el Null, nosotros tenemos que usar Steam

A screen shot of a computer

Description automatically generated with medium confidence

Para desarrolladores usamos el 480

A screenshot of a computer program

Description automatically generated with medium confidence

Con steam si haces SHIFT + A sale Steam overlay.