

Thanks for buying

MFPS

(**M**ultiplayer FPS)

1.0.5

Content:

- About.
- Get starting with MFPS.
- Explanation of the scenes.
- Functions of the buttons on the main menu.
- Class List.
- Put a new weapon.
- Put a new player model. (WIP)
- Create a new scene.
- Contact.
- Change Log

-About:

-MFPS is an advanced multiplayer FPS system with multiple features, containing everything you need to start your own multiplayer game with an advanced base.

Features:

- Menu / Lobby.
- Search and create new rooms.
- AFK Detection. (If a player is inactive for a desert time, this will automatically return to the lobby.
- Kill Camera.
- Player Class
- Super Player Jump
- Basic examples, delegates internal events
- Advanced example of multiplayer event
- MedKit Pickup (Sync)
- Ammo Pickup
- All Code in C#
- Advance KillFeed in corner (WhoKilledWho)
- HeatShot Detect
- Team selection and Auto-Team selection
- Games for rounds or only one round.
- Room Time Sync.
- Smooth weapon Movements when is running.
- Damage Indicator.
- Advance Kit Call and Pick Up(Like CoD AW)
- Advanced weapon system (Bullet penetration, Bullet Physical or Raycast Spread, Impact strength, range).
- Shot types (Auto, Burst and Semi-Automatic).
- Real GunBob Movement.
- Real HeatBob Movement.
- TilWeapon Movement.
- Player Model Animation in Network.
- Weapon Fire and Bullet Sync.
- Network Gun Sync.
- Fall Damage.
- Kill Notification.
- Draw the Dynamic Name Above Head.
- Network Footstep.
- Tree game modes (Free For All, Team Death Macht and CaptureThe Flag)
- Different Player Model for each team.
- Multiplayer Chat.
- ScoreBoard (Kills,Deaths,Score,Ms Server(Ping)) for each player.
- Options menu (in Lobby and Rooms).

Get starting with MFPS

Required :

- Get Photon PUN Plugin

-Once imported the package, you need to download and import the Photon Unity Networking package from the asset Store:

<https://www.assetstore.unity3d.com/en/#!/content/1786>

- Now you are ready to start with MFPS.

- Get Your AppID
- Register a Photon Cloud Account: <https://cloud.exitgames.com/Account/SignUp>
- Get your AppID from the Dashboard

2. Paste Your AppID

- In unity editor, Open [Window] -> [Photon Unity Networking] in the menu bar.
- Push [Setup] button, and paste your AppID to [Your AppID] field.
- Choose your region you want to set. You can select from EU/US/Asia/Japan.
- Push [Save] button

3. Play Game!

- Have fun!

Required Layers and Tags

:Tags

1:Enemy

2:WeaponCam

4:Projectile

5:BodyPart

6:GameManager

7:Mouse

8:Remote

9:WeaponManager

10:ItemManager

11:Ladder

12:Concrete

13:Metal

14:Flag

:Layer

8:Weapon

- Explanation of the scene "Menu"

- Open the scene named "Menu"
- Then play the scene
- Now a small menu will appear, where you will write the name of your player.
- A look you have written your name, click on the button called "Continue".
- Now another menu will appear where you will have several buttons with various options.

Functions of the buttons on the Lobby.

- Search Room:
to open a menu where you may find a new server you join.
- Host Room:
to create and host a new server to which other players can join by opening this menu, the options appear in the room which you could configure.
- Settings
will open the menu with the options of the game, where you adjust to your liking, and then keep in the room charge.
- Change server:
will open a menu with photon cloud servers, enabled to connect, entering here you may not return to the main menu to choose a server to connect to.
- Quit
will open a small window to confirm that you want to quit the game (only works when you already built the game).

Host a new Room:

- For create a new room in Lobby, go to "Host Room" Option.
- Now you will have some options that may change your disposition:
- Host Name:
The name of room, this is the name that appears in the list of available Room.
- Max Players:
Maximum capacity of players who can enter the room.
- Max Time:
the time duration of the round or rounds.
- GameMode:
the mode of game to be played
- Auto Team Selection:
if true, players will be automatically placed to the team with less players.
- Game Per Rounds:
if true, the end of a round, automatically start another, if false, the end of the round is returned to the lobby.

Class list

[bl_PhotonConnection](#).

functional components:

the main script that allows us to establish the connection in Network Manager with [photon cloud \(PUN\)](#), contains everything needed to establish the connection to our server in the cloud.

This component requires (PhotonView) to function

Class Component:

- [string](#) AppID
The id of our application in photon cloud, unless you have a subscribe here to get a: [appID](#)
- [string](#) AppVersion
version of our game, this is to prevent players having problems with old versions.
- [Int](#) port
address "Port" to which we will establish the connection (default is 5055)
- [Bool](#) ShowPhotonStatus
show the status of the connection in the menu?
- [Bool](#) ShowPhotonStatics
shows the statistics of our game in the menu?
- [List<string>](#) GameModes = [new List<string>\(\)](#)
list of reference names of our game modes.
- [List<int>](#) MaxPlayers = [new List<int>\(\)](#)
list of options available to the maximum capacity of players in a room
- [List<int>](#) RoomTimes = [new List<int>\(\)](#)
list with the different options in the room maximum time (in seconds)

BI_GameManager

This is the script that handles the logic of the player spawn contains the functions to instantiate our players in different spawnpoints to others to handle the camera to preview the room.

functional components:

- `GameObject` `Player_Team_1`
the prefabs instantiate the player for Team 1 / Delta
- `GameObject` `Player_Team_2`
the prefabs instantiate the player for Team 2 / Recon
- `GameObject` `RommCamera`
The camera preview the room or chamber, which will deactivate once the player is instantiated
- `Transform[]` `AllSpanwPoints`
All spawnpoints to instantiate the player in the FFA mode
- `Transform[]` `DeltaSpanwPoints`
All spawnpoints to instantiate the player in the TDM mode for Deltan team
- `Transform[]` `ReconSpanwPoints`
All spawnpoints to instantiate the player in the TDM mode for Recon team

```
public void SpawnPlayer(Team t_team)
```

Call this when you need to instantiate your player.

```
public void DestroyPlayer()
```

Call this when you need to Destroy your player.

BI_RoomMenu

handles the logic of the room like GUI scoreboard, score, teams, etc ... to other contains functions to choose the game mode and choose the team.

functional components:

- **Bool** AutoTeamSelection
choose the team or automatic selection
- **Int** MaxPing
Maximum ping before displaying the message
- **String** MsnMaxPing
when the ping exceeds MaxPing then display this warning text
- **Bool** RotateCamera
want to rotate the camera preview
- **Float** RotSpeed
the speed with which the camera rotate

public int GetPlayerInReconCount

Call this method to get the number of players in the Recon team.

public int GetPlayerInDeltaCount

Call this method to get the number of players in the Delta team.

public static IEnumerator FadeIn()

coroutine call this to get the effect of Fade In

public static IEnumerator FadeOut(float t_time)

coroutine call this to get the effect of Fade Out.

■

bl_PlayerSettings

use this script to configure when player is local and when it is remote (when is ours and when not).

functional components:

String LocalTag

Player tag, where is local (when is ours)

String RemoteTag

Player tag, where is Remote (when is not ours)

Team m_Team

Player Team enum

List<MonoBehaviour> Local_DisabledScript

list of script that will disable when our player

List<MonoBehaviour> Remote_DisabledScript

list of script that will disable when is not our player

List<GameObject> Local_DesactiveObjects

list of GameObject that will disable when our player

List<GameObject> Remote_DesactiveObjects

list of GameObject that will disable when is not our player

List<Messages> m_SendMessage

when the player is not our player, and do not want to completely disable the script, then send a message to disable some functions only .

class Hands

contains the necessary variables to change textures hands for the respective equipment.

public void RemotePlayer()

is called when the player is not ours (the remote)

public void LocalPlayer()

is called when the player is ours (the local)

bl_PlayerDamageManager.cs

this contains all the logic of the player health This is enabled locally and remotely.

Functional Component:

bool LocalPlayer

it serves as a benchmark for determining whether the actor is local or remote

string LocalPlayerName

it serves as a reference to identify the player's name, this is automatically assigned in RunTime.

float Health

health with players begin in each Spawn.

float MaxHelath

health allowable limit each player can be obtained, it can be greater than Health.

float SpawnTime

time to take back to appear after death.

float FadeSpeed

time it will take to fading the texture Paint and Blood HUD.

float UIIntensity

multiplier to increase the intensity of the alpha textures of Blood and Paint HUD

float HitFade

time when fading the Hitmarker .

float ShakeSmooth

speed with which the camera will return to original position after movement.

float ShakeTime

time duration of the movement before returning to the original position.

float ShakeAmount

Shake amount of movement, more amount = more movement.

Vector2 RandomScore

the range of experience that will be added in personal score for each time you get a Kill injury.

public void GetDamage(bl_OnDamageInfo e)

This call to exert damage to the player who has this, it will automatically sync, just be sure to submit all the required variables (see bl_OnDamageInfo)

public void Suicide()

Call this to automatically kill the player that contains the script.

This is a simple base class
to us serve as an extension of Photon.Monobehaviour default

```
public GameObject FindPlayerRoot(int view)  
int view  
the viewId of PhotonView of player we want to find
```

Call this when you want to find the GameObject player through its viewId, functional when you want to get a component in the player.

```
public PhotonView FindPlayerView(int view)  
int view  
the viewId of PhotonView of player we want to find
```

Call this when you want to find the PhotonView player through its viewId, functional when you want to send a RPC or just call the player.

```
public bool isMine  
True if the PhotonView is "mine" and can be controlled by this client.  
PUN has an ownership concept that defines who can control and destroy each PhotonView. True in case the owner matches the local PhotonPlayer. True if this is a scene photonview on the Master client.
```

```
public bool isConnected  
if true, we are connected to Photon
```

```
public GameObject FindPhotonPlayer(PhotonPlayer p)  
PhotonPlayer p  
the PhotonPlayer from which we obtain the GameObject
```

Call this when you want to get the GameObject a player by PhotonPlayer, usefull when you want to get the component of the player.

```
public string GetTeam(PhotonPlayer p)  
PhotonPlayer p  
player we want to get the team.
```

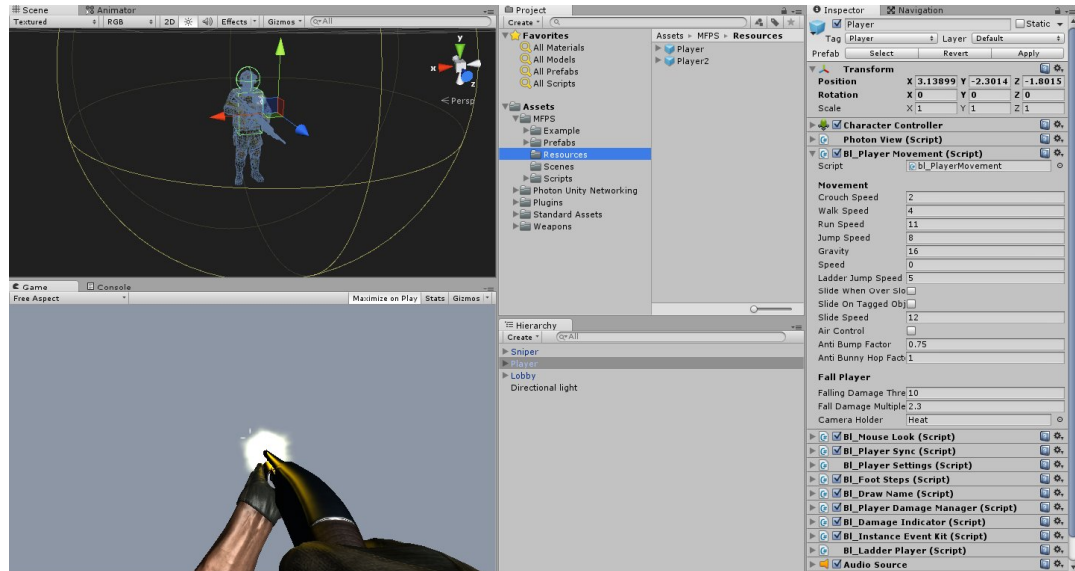
Call this when you want to get the team some specific player.

```
public GameMode GetGameMode  
Call this when you want to obtain the actual game mode of Room.
```

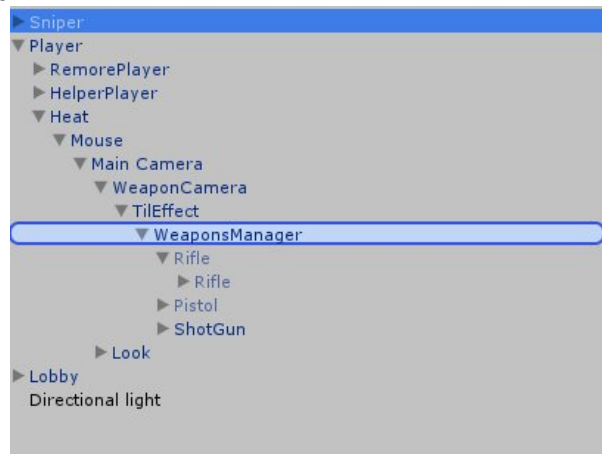
```
public string LocalName  
Call this when you want to get the name of our local player.
```

Put a new weapon [Tutorial]

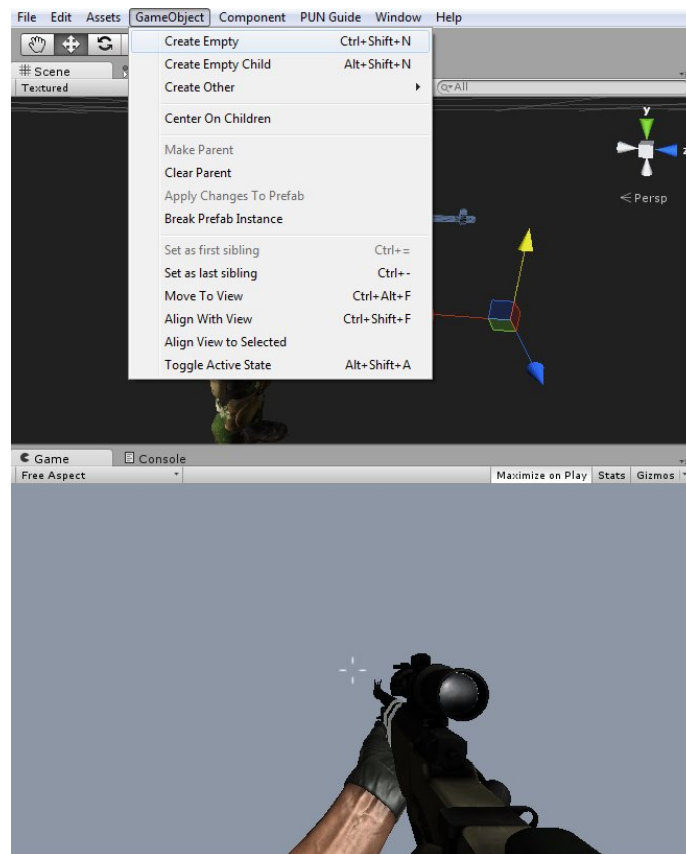
to add a new weapon available to players, first drag the prefabs "Player" from the "Resource" folder in the scene where we are.



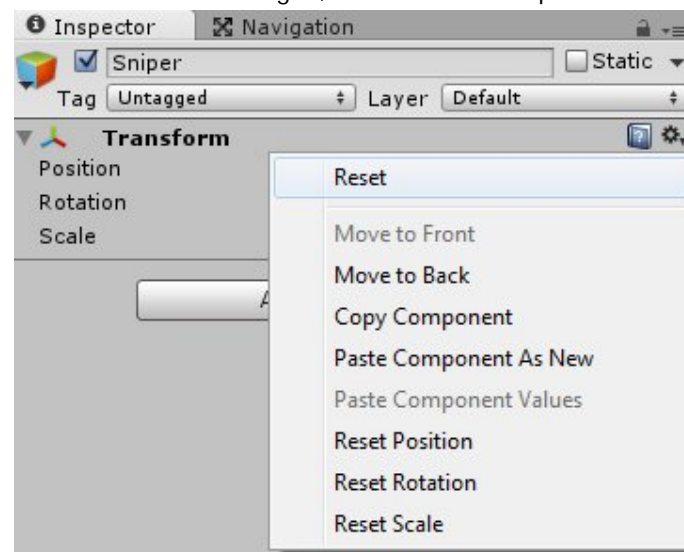
-Then drag the model of the new weapon in the scene too, and we do son of "WeaponManager".



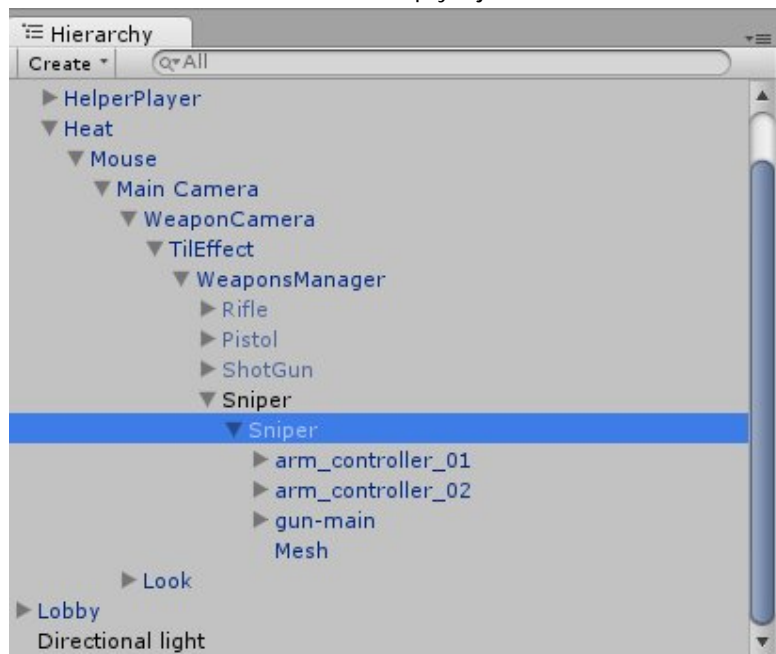
- put the gun securely in place, for it to look good on camera (see window, preview the game), then create an empty object.



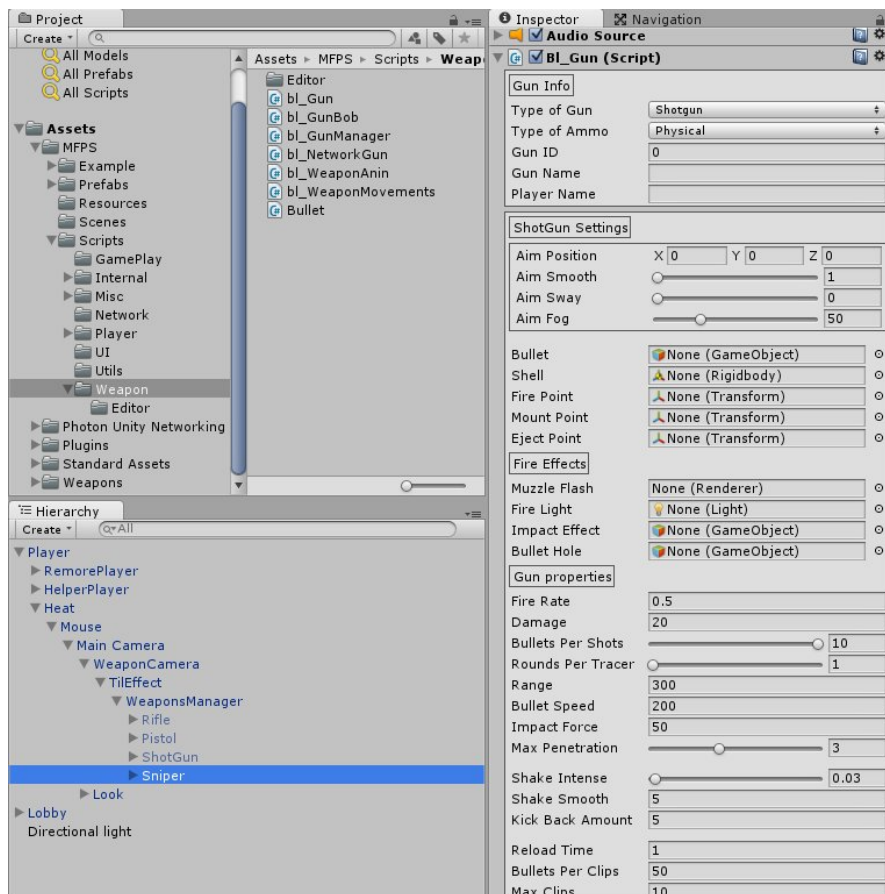
then rename the empty object with the name of his new weapon, then Do it son of the new model of the gun, and this resets it's position.



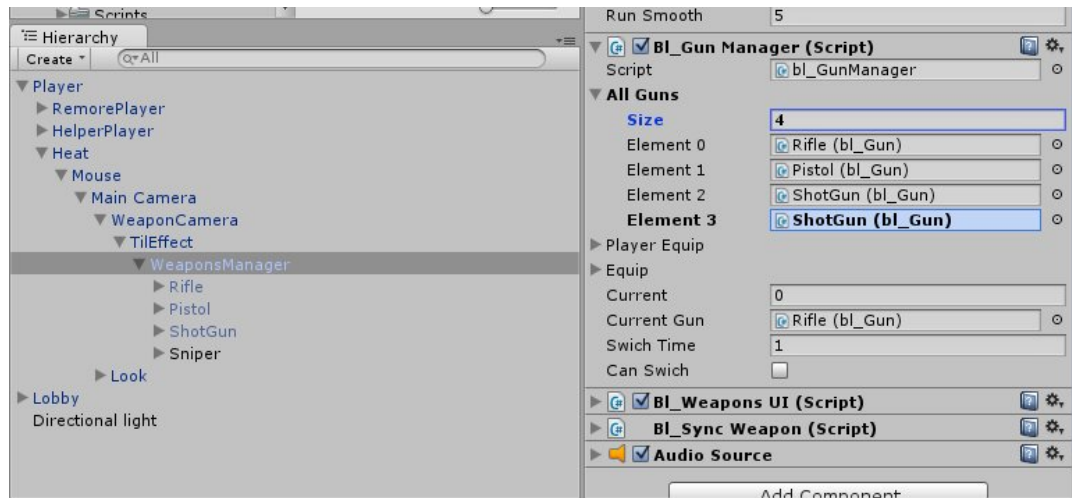
then again make the empty object, son of "WeaponManager", then the new model gun
Do it son, emptyobject.



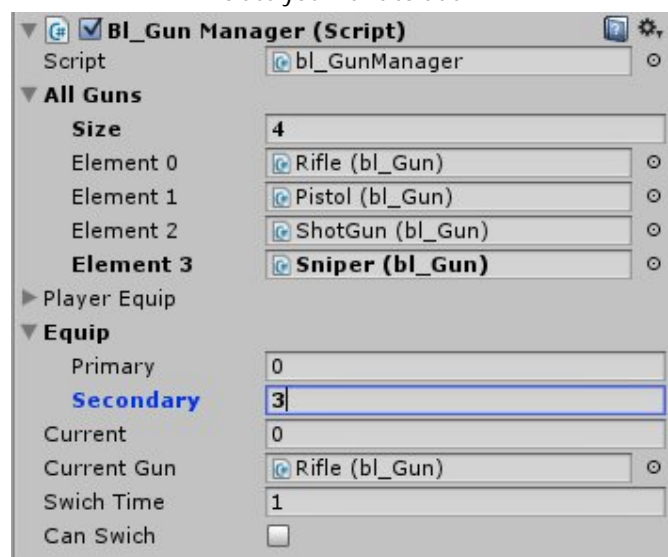
then the "empty object" add "bl_Gun.cs" script, and modify and fill in all the variables
that are there.



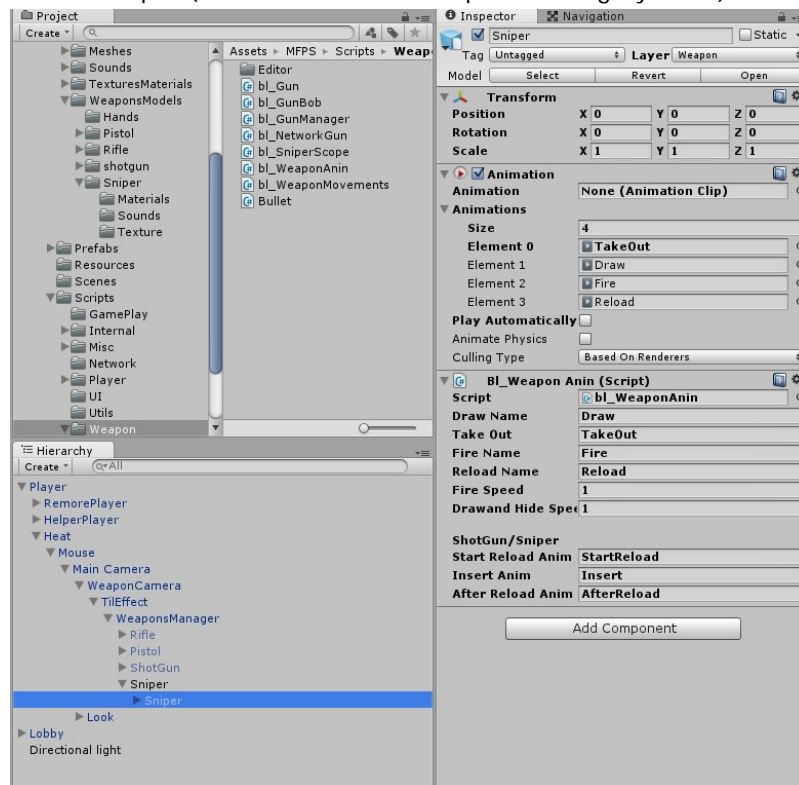
once filled all variables, add new weapon to the list of "bl_GunManager" call "AllGuns".



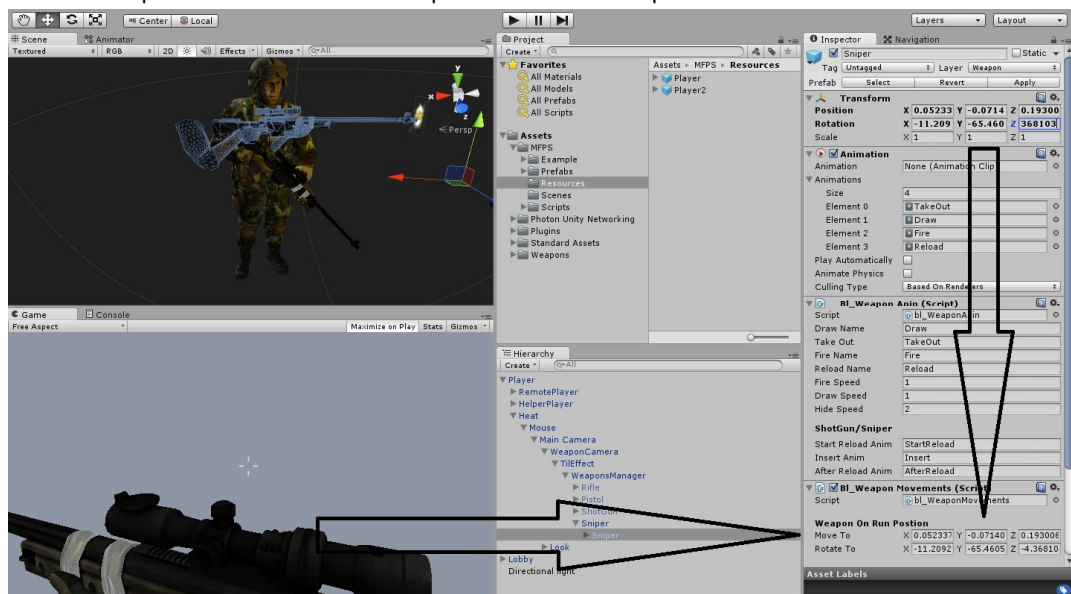
if you want to make this a primary or secondary weapon of some PlayerClass, you only put the "Element Num" of "AllGuns" list occupies the weapon in the variable of the class you want to add.



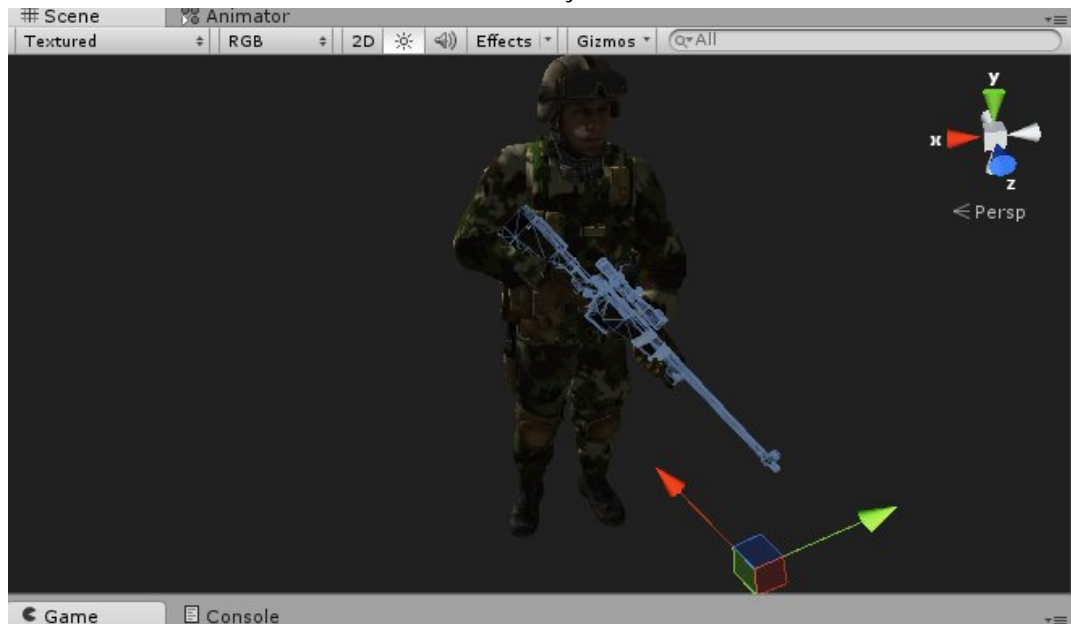
then in the model of the weapon, which contains the component "Animation", add the "WeaponAnim.cs" script and put the name of the respective animations of your new weapon (the model needs to be operate in Legacy mode)



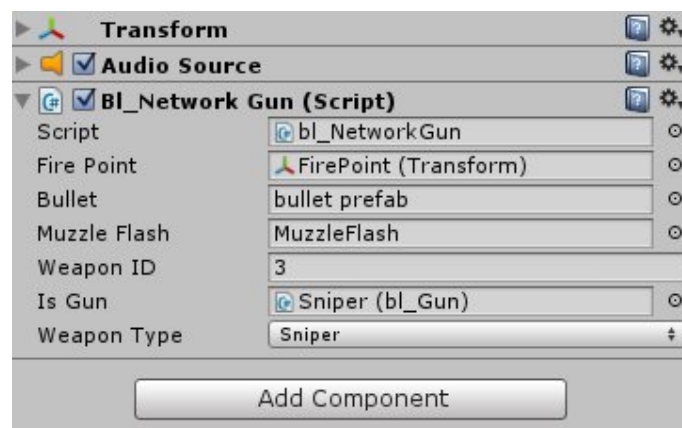
if you want to add the effect of rotation of the weapon, add the script called "bl_WeaponMovements.cs" and to place the correct position you can do so from the inspector, just put the gun in the gun position will move when running, and then copy the position and rotation and paste it into the script, the same for both.



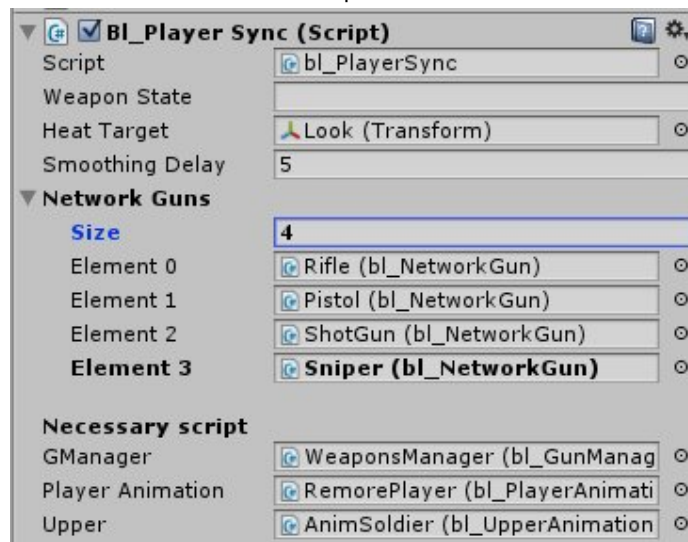
then the model of the third-party player, put the gun third person in the position of the hands correctly.



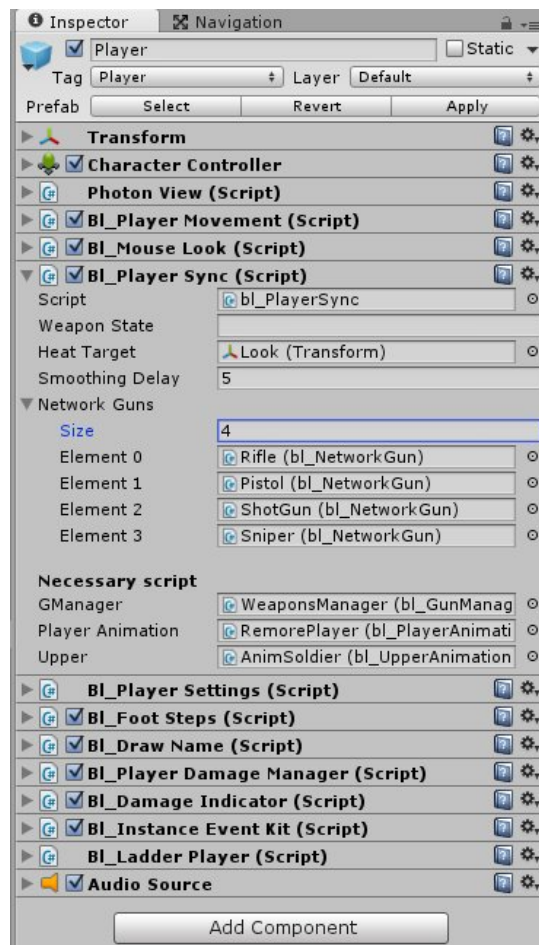
then add the script called "bl_NetworkGun.cs" and fill in all the fields correctly, note that the variable "isGun", you should place the same gun but in local mode, as this will be where to take the information.



then "bl_PlayerSync.cs" in the list called "NetworkGuns" add the new weapon of the third person.



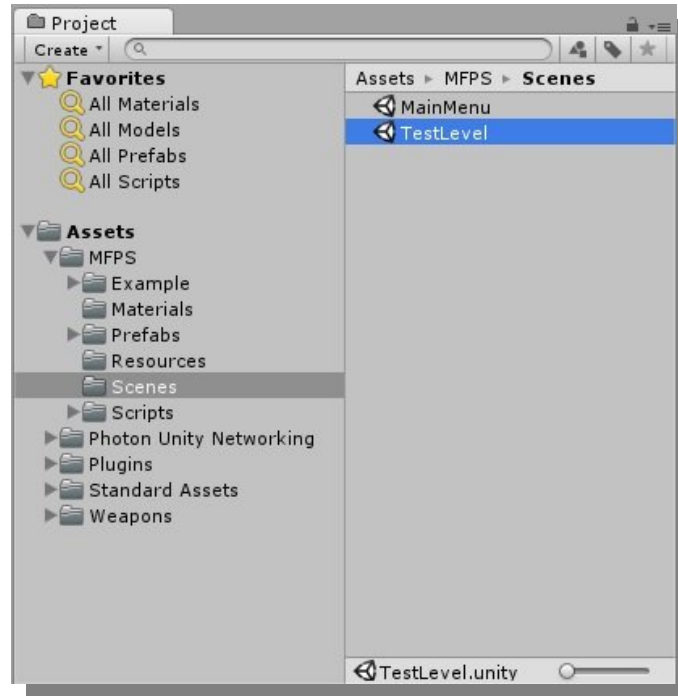
and finally update the player prefabs from both teams, just create a copy of prefabs in scena and rename "Player2" and place it in the prefabs of player2.



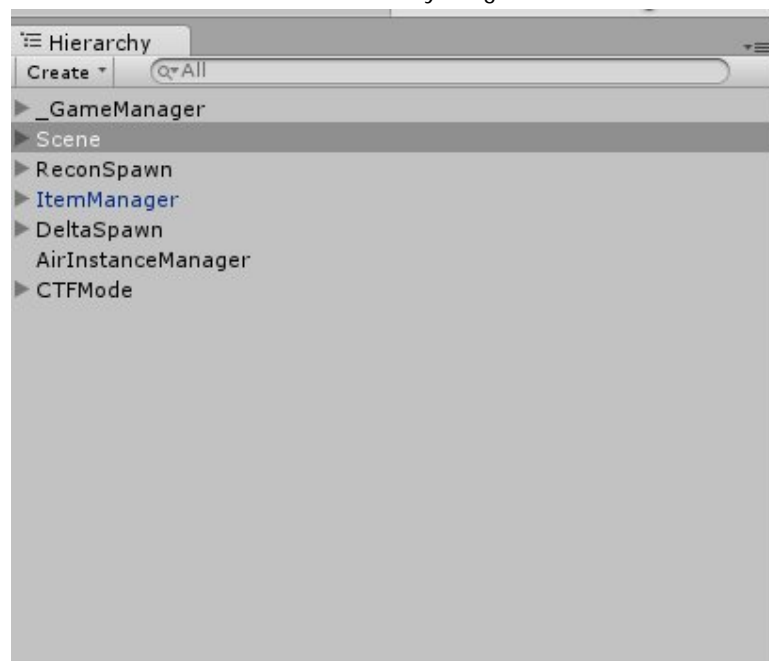
Create a new scene [Tutorial]

to create new multiplayer scene, follow these step:

- first, is to open the package sample scene called "TestLevel" in the scene folder.



- then when we're in the scene, in the "Hierarchy" window, delete the gameobject called "Scene", because it contains all the ambience of the scena.
- do not delete anything else.



The screenshot shows the Unity Inspector window with the 'GameObject' menu open. The menu options are as follows:

MenuItem	Shortcut
Create Empty	Ctrl+Shift+N
Create Empty Child	Alt+Shift+N
Create Other	
Center On Children	
Make Parent	
Clear Parent	
Apply Changes To Prefab	
Break Prefab Instance	
Set as first sibling	Ctrl+=
Set as last sibling	Ctrl+-
Move To View	Ctrl+Alt+F
Align With View	Ctrl+Shift+F
Align View to Selected	
Toggle Active State	Alt+Shift+A

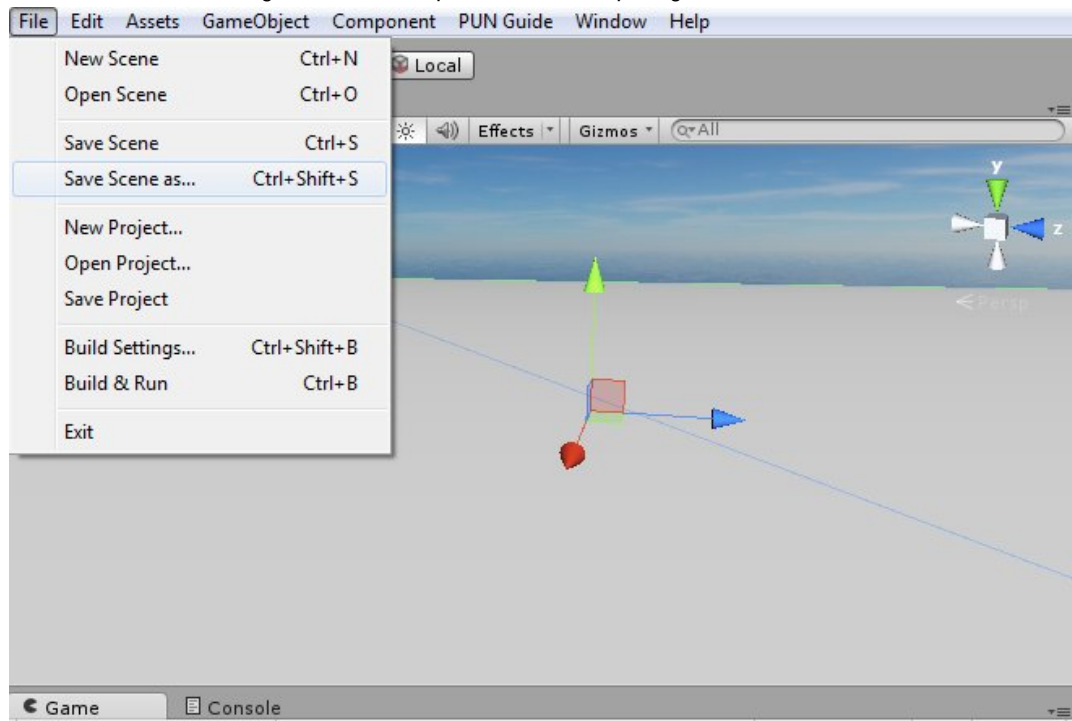
The background shows the Unity Hierarchy panel with a 'Textured' object selected, and the main viewport displaying a 3D scene with a grid floor and a blue sky. The 'Persp' view mode is indicated in the bottom right corner of the viewport.

The screenshot displays the Unreal Engine 4 interface. The main viewport shows a third-person view of a character standing on a grassy field with mountains in the background. The top bar includes the 'Game' menu, 'Console', and 'Nomico v1 Play' button. The right side features the 'Hierarchy' panel, which lists the game object structure, including 'GameManager', 'ReconGame', 'ReconGame', 'DataSpan', 'AirframeManager', 'CTMode', and 'Scene'. The 'Outliner' panel is also visible, showing a list of assets and a search bar.

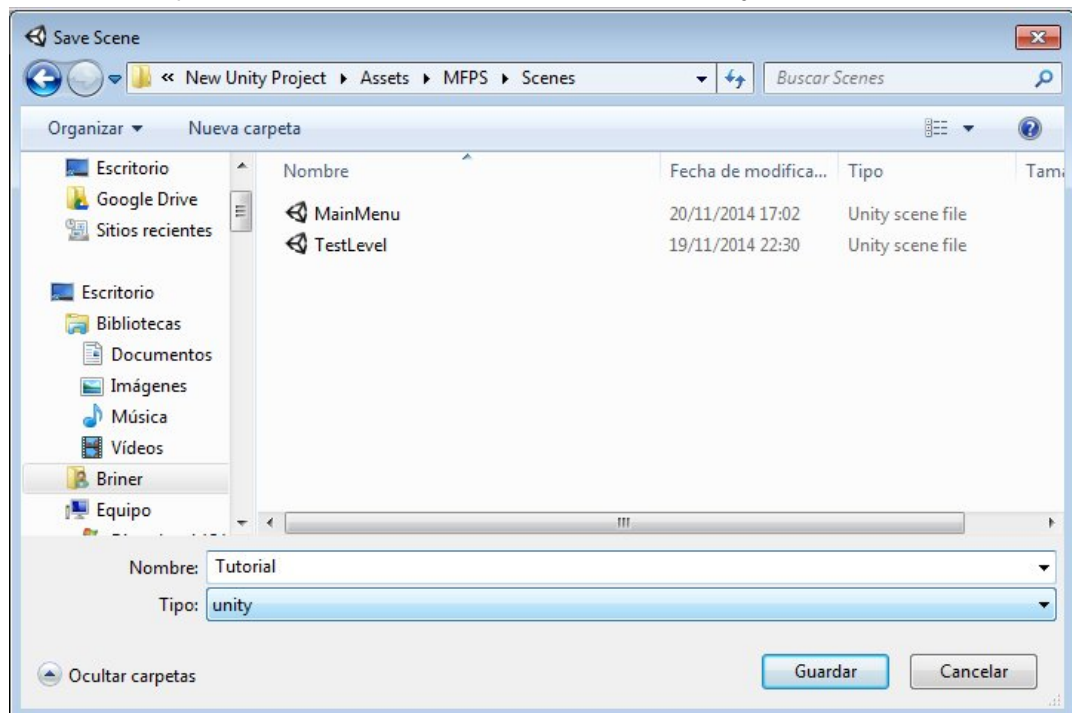
- Must have some geometry for both teams (for TDM and CTF mode) have the same opportunities.

- Avoid packing items you do not need, since it's a multiplayer game, when you're in a room with up to 12 players, resources taking pc gaming are very high, so just put what it takes to save resource pc .

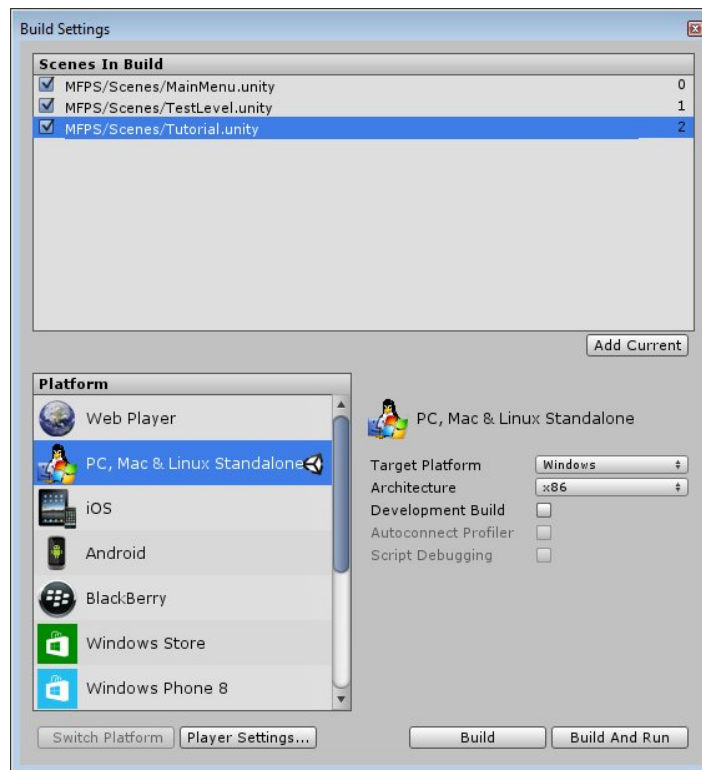
once completed the new level of the scena ,do not forget to reposition the spawn points, go to the "File" window, and keep the scena in the option "Save As", this not having to create and place all the scripts again scene.



keep the scene in the scenes folder, with the name of your level.



Now in the "Build Settings", add the new scena, to add to the project and compile when building the game.

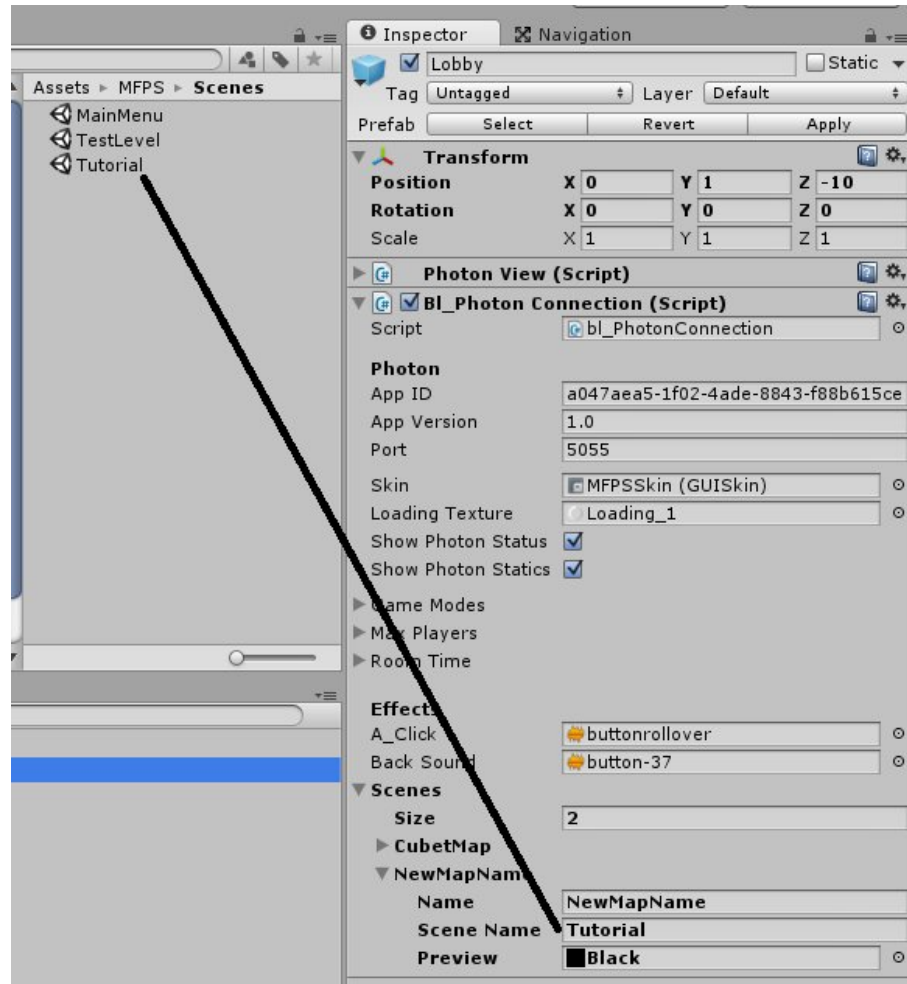


now in the "MainMenu" scene in "bl_PhotonConnection.cs" script in the list called "Scenes", add a new field.



Now fill the new field variables like this:

- Name: The name that will be displayed in the GUI.
- Scene Name :The name of the project scene
- Preview : Texture Scena preview



and with this we conclude, now you have a new scene.

Change Log:

- version 1.0:-----

- Initial version

Version 1.0.1-----

- Remove Button BestCloud from Changer server menu
- Improved FallDamage
- Added GUI for Flag
- different sensitivity when is Aimed.
- different Bullet spread when is Aimed.
- fixed small bugs.

Version 1.0.2-----

- Improved: Menu Lobby
- Add: more options for GUISkin
- Added: custom shader "Bumped Specular MFPS"
- Added: custom shader "MFPS Lighting"
- Fix: shotgun low bullet GUI
- Improved: when round is finished, the player no take damage
- Fix: when the player joined the room, the score was reset.
- Fix: Game Per Round now is working
- Improved: UI player now support Unity 4.6 UI
- Add: Advanced Custom Shader for SkyBox with Sun
- Add: Blood Particle for remote and local player
- Add: New Skin for player2
- Add: Atlas sprite for UI 4.6
- Improved: Shotgun and sniper Impact force
- Improved: Gravity force
- Improved: Fire animation speed when is Aimed
- Add: Show only scoreboard with press tab

Version 1.0.3-----

- Change all weapons and arms models with new Animations,these can be used for commercial use.
- Impromved: Spawn Player, now chooses a random spawn point and add a more random position.
- Fix: when creating a new Room this present without connection.
- Add: Suicide notification
- Fix: Burts Mode Animation and Camera Shake
- Added: New UIAnimations
- Added: Swicht Type of Fire when press "B" (Full,Semi,Single).
- Added: Health Slider UI.
- Re-Writte: Chat System.
- Impromoved: bl_PhotonConnection.cs logic.
- Added: New fonts type.
- Added: Now you can change Player Class in game, when press "M" and select the new class (Apply in next Spawn).
- Added: Heat Animations for a best effect.
- Added: Suicide Button, when you press button, them you death, is you suicide more than 3 times you receive kick of room.
- Adeed: sleeves and gloves texture for each team.
- Added: Now Sleeves and gloves textures depend of team (different for the team Delta and Recon).
- Fix: Some small bugs in game.
- Added: New Reload Sound System, for greater coordination reload sounds with animation.
- Added: When you are death or not playing, UI is not visible
- Added: New Static bool bl_GameManager.isAlive this determines whether the player is currently alive or dead.
- Added: New Static int bl_GameManager.m_view this contains the current actorId, of our client.

Version 1.0.4-----

- Impromoved: complementary effect of the slider of bullets and health is now smoother.
- Impromoved: Bullet Text now with smoother effect.
- Fix: Bug hand models that would disappear.
- Added: Icons for KillFeed (optional).
- Fix: Vignette disappears when another player dies.
- Fix: Registered OnAmmoEvent more than 1 time.
- Impromoved: Weapons sounds with new Free to use.
- Added: Now you can't fire when round is end.
- Impromoved: Ragdoll.
- Fix: Scoreboard Tab, now is show with "N" key.
- Added: Hit Player sound, when receive damage.
- Fix: Select Class and disconnect buttons for FFA.

Version 1.0.5-----

- Added: New spawn effect in gloves and sleeves materials (Optional).
- Added: New script "bl_DeathZone.cs": add a collider, and serves to declare restricted areas when you are in one of these areas for more than 5 seconds, you will die.
- Fix: Suicide button now is enable only when is alive.
- Fix: Singletons Scripts instance more than one time.
- Impromoved: Window Chat now use uGUI.
- Fix: problems with collisions, the player not detected OnTriggerExit.
- Added: New Base component script "bl_PhotonHelper.cs" an extended version of "Photon.MonoBehaviour".
- Added: New script "bl_OnDamageInfo.cs", use this as a reference for information needed to send a new injury.
- Added: New Option for ADS "useSmooth" in "bl_Gun.cs" if true for a soft effect, if false for a snap effect.
- Added: New Script "bl_BodyPartManager.cs", This script helps us manage and configure easily our remote player hitboxes.
- Added: New Script "bl_Grenade.cs" contains the logic of Granada type weapon.
- Added: New Script "bl_Blast.cs" contains the logic of explosions.
- Impromoved: Now Launcher type grenade is working.
- Added: New weapon "Grenade".
- Added: New Static Extension for player popierties, this facilitates access to properties more authoritatively for each photon player, ej: [PhotonNetwork.player.GetKills\(\)](#);
- Fix: damage direction of fall damage.
- Impromoved: Damage Indicator, now is more accurate.
- Impromoved: Player Sync is now more advanced and more variants, with a new option of "Extrapolate" for calculate the next position of the player.
- Added: New slot for all class "Special", change this with Keycode.Alpha3
- Fix: Shotgun and Sniper Delay Fire sound interrupts the firing sound.