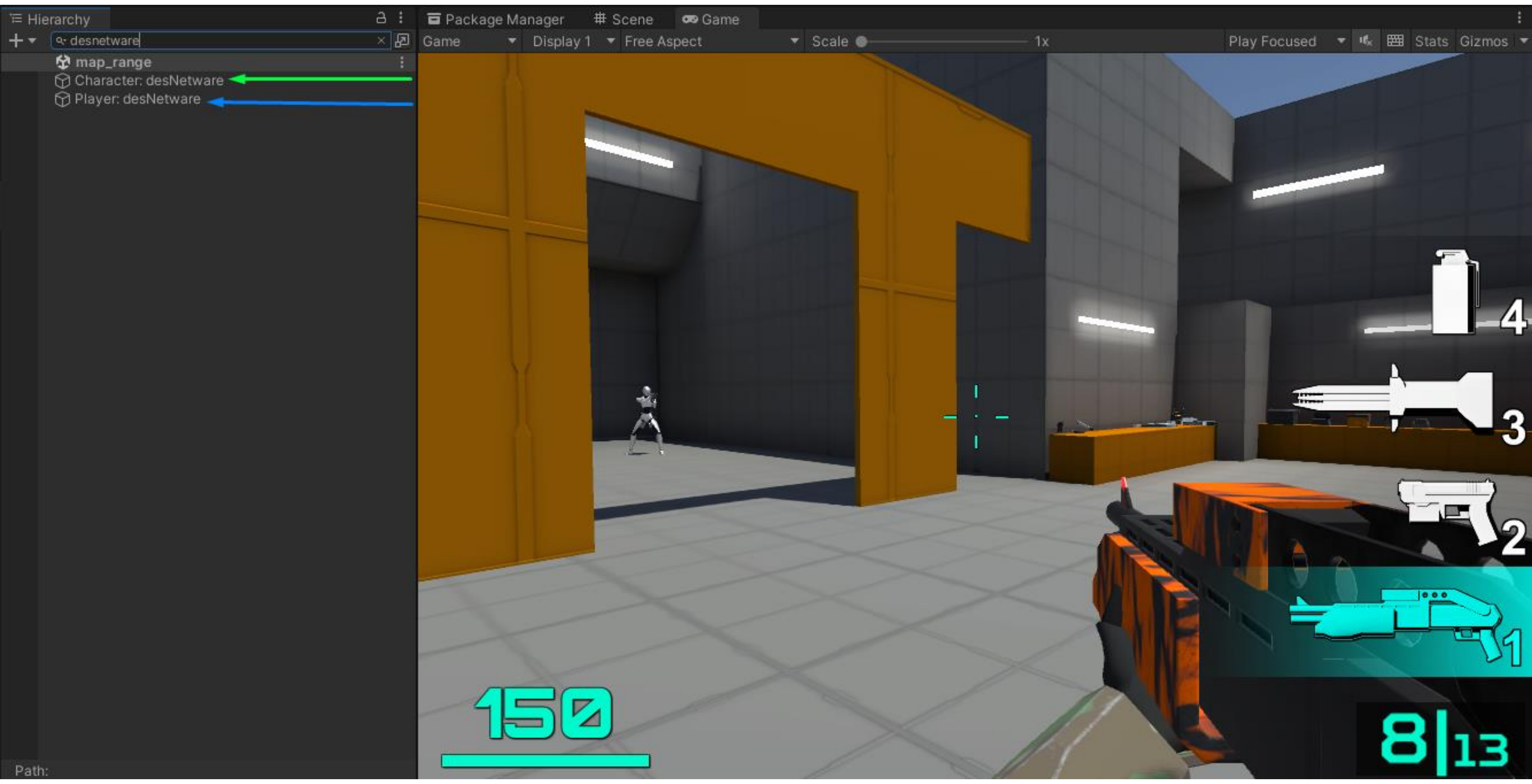
MultiFPS systems documentation

Contents

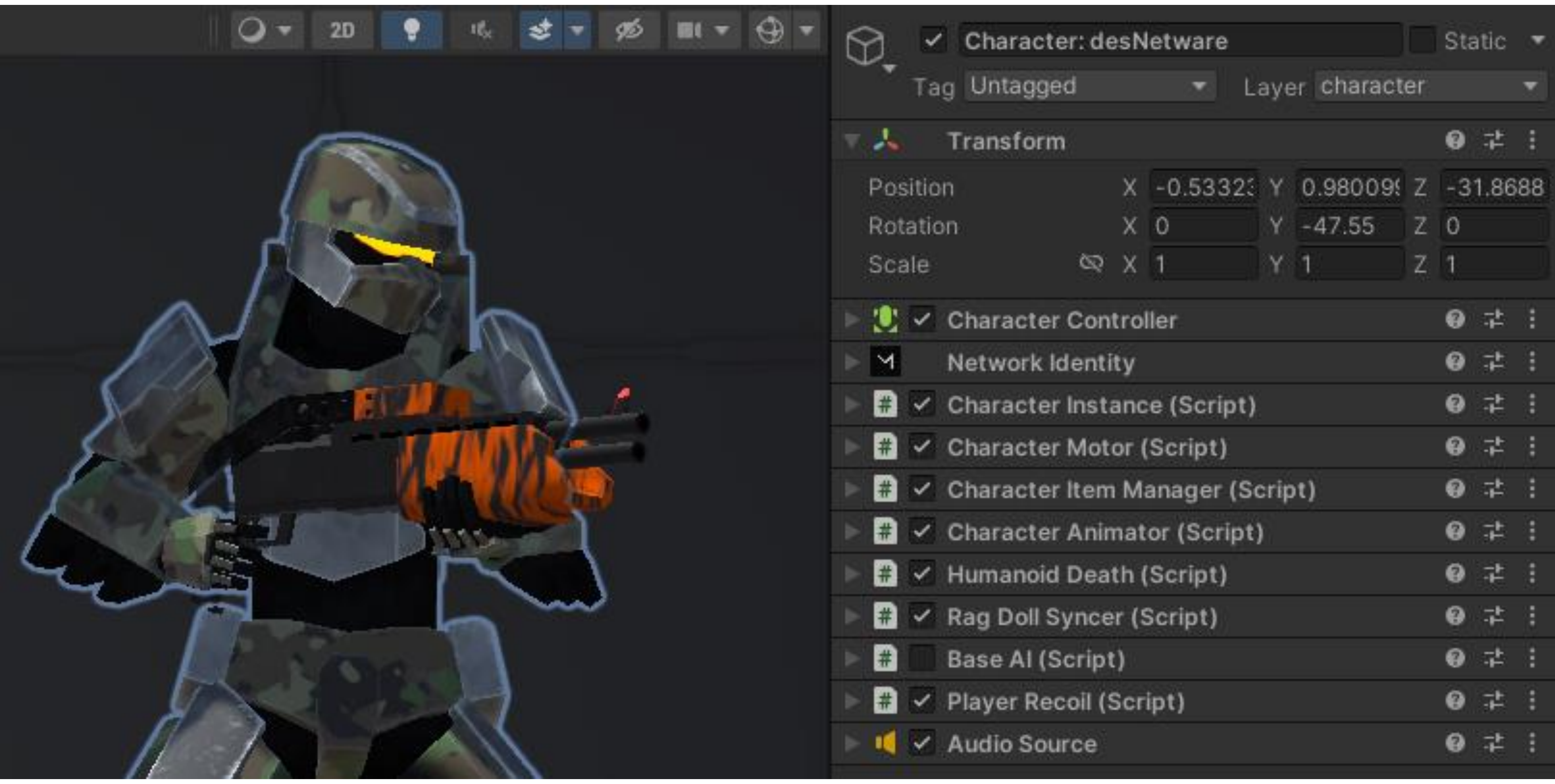
1.Players management



Each connected player (and bot) is represented by empty object with component PlayerInstance . This object is named “Player” with player nickname after that. From this objects players can communicate by chat and send spawn requests to the server, which can be accepted or not depending on game state and gamemode. This object exists as long as player is present in the game.

Then, when this object already exists, server will spawn character for given player, if certain conditions for given gamemode and gamestate are met. For example, if we play Team Deathmatch, spawn player after he chooses his team, or respawn player if he died and waited a couple seconds. Or don’t let them spawn, and make them wait for next round to respawn, like it is in gamemodes like TeamEliminations or Defuse.

2. Player prefab



Character: desNetware
Static
Tag Untagged
Layer character

Transform
Position X -0.53323 Y 0.98009 Z -31.8688
Rotation X 0 Y -47.55 Z 0
Scale X 1 Y 1 Z 1

✓ Character Controller
Network Identity
✓ Character Instance (Script)
✓ Character Motor (Script)
✓ Character Item Manager (Script)
✓ Character Animator (Script)
✓ Humanoid Death (Script)
✓ Rag Doll Syncer (Script)
Base AI (Script)
✓ Player Recoil (Script)
✓ Audio Source

Components list with explanation:

CharacterInstance Responsible for basic character management. Like setting perspective from which character will be rendered, sending player input to server and clients, lerping position and more. Everything that character is controlled by players needs to have. This class also inherits Health class, what makes character able to be damaged and killed.

Health Responsible for receiving damage, and synchronizing it on server. Includes callbacks for being damaged and killed, specific for clients and server

CharacterMotor Responsible for character movement. Bots are also using this for their movement.

CharacterItemManager Responsible for managing player inventory, equipping items, putting them down, dropping them and picking new ones.

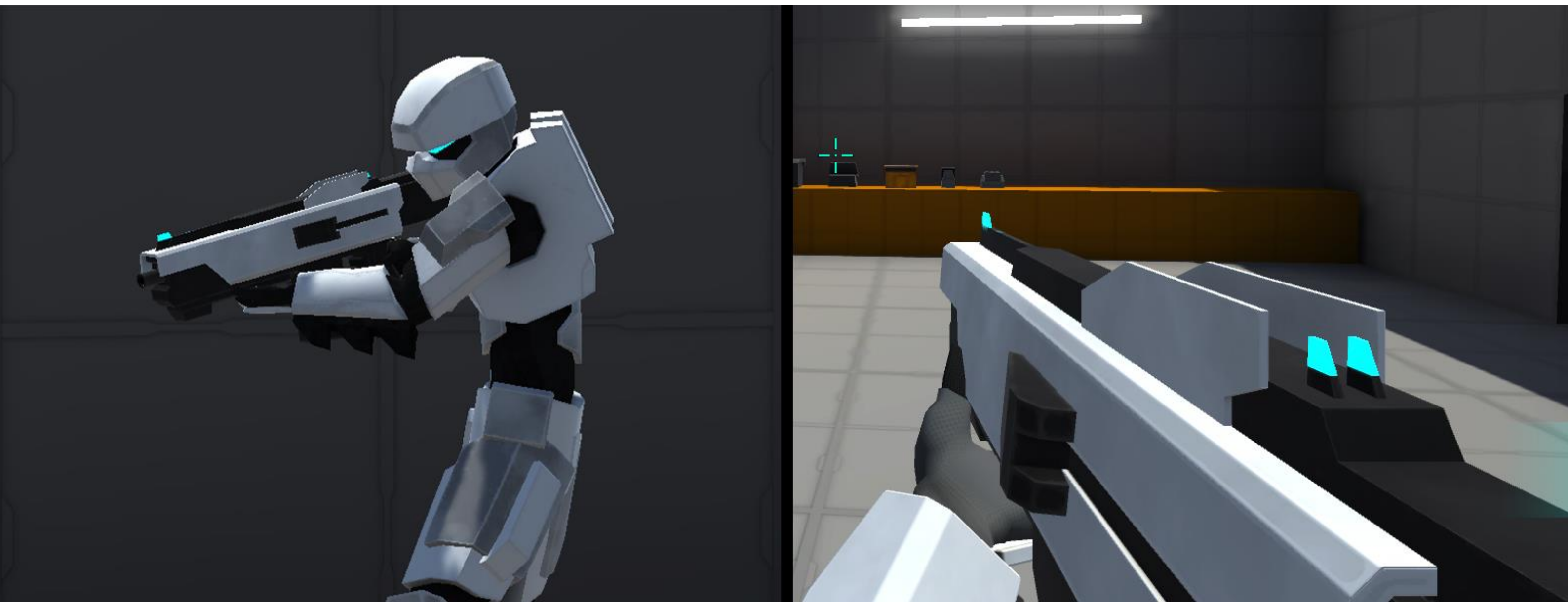
CharacterAnimator Based on player inputs stored in CharacterInstance component, animates character accordingly, thanks to that we don't need to use Mirror's CharacterAnimator, which saves us network bandwidth.

HumanoidAnimator Just spawns ragdoll on death

RagdollSyncer Responsible for sending to clients ragdoll information that is calculated on server so dead body lands for every client the same way.

BaseAI Enabled only for bots. This scripts is bot mind. It decides what to do and sends input to CharacterInstance. Go there, look there, shoot there, crouch in random moments. After sending input to CharacterInstance everything is handled as if it was normal player, no exceptions.

3. Item system



MultiFPS for both FPP and TPP perspective uses same item prefabs, so each item has only one prefab in game files. Difference lays in animations animation sets.

Item system consists of two scripts: Item (and classes that inherits Item) and CharacterItemManager

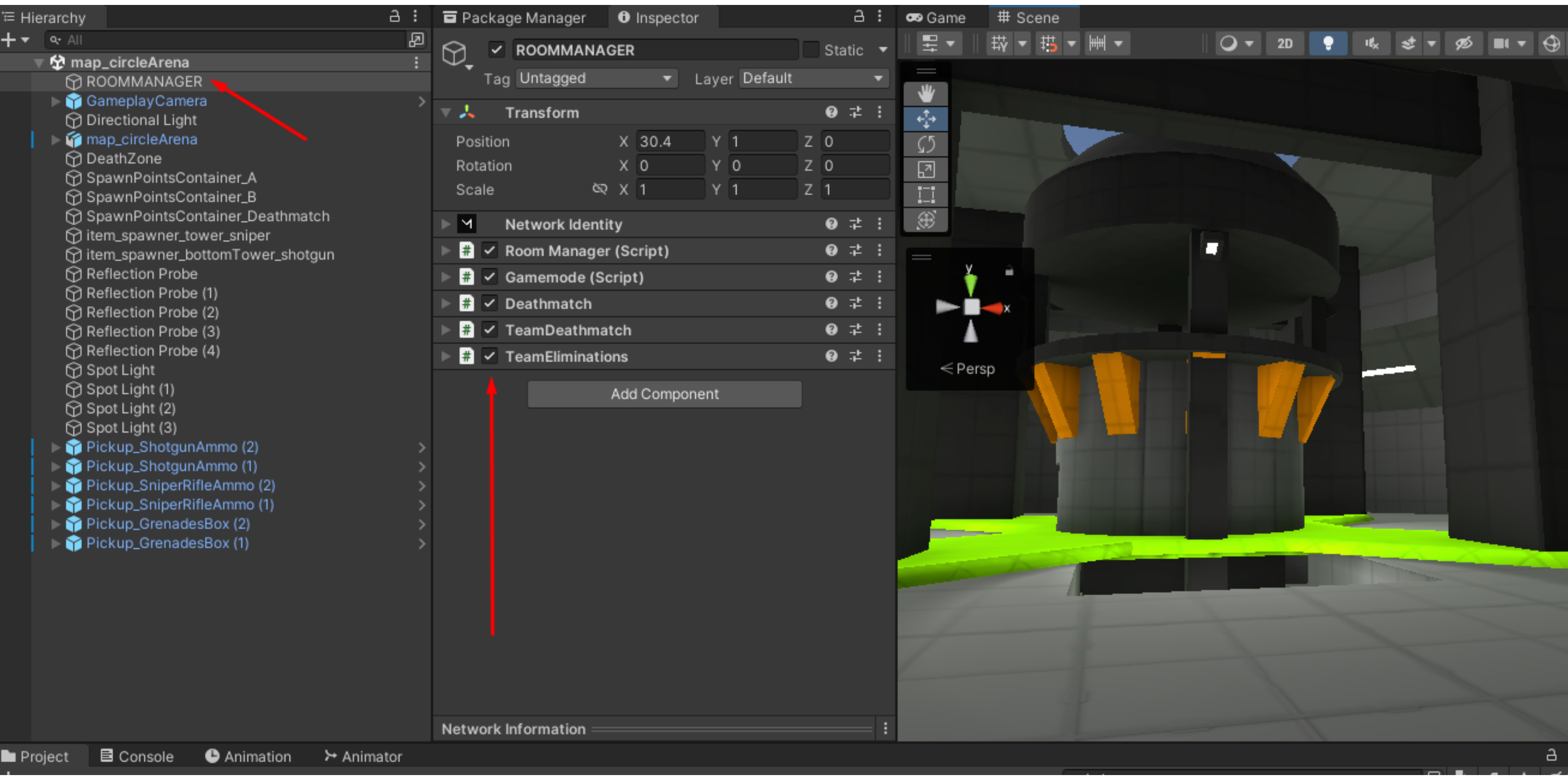
Every single item in the game that is usable for player inherits from Item. This class contains functions for picking, dropping, taking, and putting down item. It also synchronizes this functions for all players connected to the game.

CharacterItemManager manages player equipment. Thanks to this class we can take item, pickup it, drop, drop all of our items on death etc. It is also responsible for synchronization, so if we decide to change weapon, every client will see that.

4. Gamemodes

Gamemode system consists of those scripts: : RoomManager Gamemode and scripts that inherits from Gamemode: Deathmatch TeamDeathmatch TeamEliminations and Defuse

All of those scripts are present in map scene, on gameobject "ROOMMANAGER"



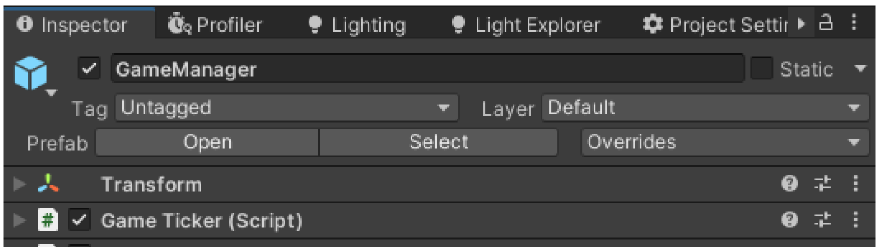
```
Hierarchy
map_circleArena
  ROOMMANAGER
  GameplayCamera
  Directional Light
  map_circleArena
  DeathZone
  SpawnPointsContainer_A
  SpawnPointsContainer_B
  SpawnPointsContainer_Deathmatch
  item_spawner_tower_sniper
  item_spawner_bottomTower_shotgun
  Reflection Probe
  Reflection Probe (1)
  Reflection Probe (2)
  Reflection Probe (3)
  Reflection Probe (4)
  Spot Light
  Spot Light (1)
  Spot Light (2)
  Spot Light (3)
  Pickup_ShotgunAmmo (2)
  Pickup_ShotgunAmmo (1)
  Pickup_SniperRifleAmmo (2)
  Pickup_SniperRifleAmmo (1)
  Pickup_GrenadesBox (2)
  Pickup_GrenadesBox (1)

Inspector
ROOMMANAGER                Static
Tag Untagged     Layer Default
Transform
  Position  X 30.4  Y 1  Z 0
  Rotation  X 0     Y 0  Z 0
  Scale     X 1     Y 1  Z 1
Network Identity
Room Manager (Script)
Gamemode (Script)
Deathmatch
TeamDeathmatch
TeamEliminations
Add Component
Network Information
```

Each gamemode is placed in its own component. RoomManager component activates gamemode previously selected by user in hub scene (or loads it by properties if it is server headless instance). If map supports certain gamemodes, then their components have to be present and set up in this in object "ROOMMANAGER".

Base Gamemode class is also responsible for spawning bots if they are needed.

5. Networking

For networking MultiFPS uses Mirror, open-source network solution for server authoritative multiplayer games on Unity3D engine. For managing networking Mirror provides NetworkManager script, MultiFPS CustomNetworkManager inherits from that to add more functionality. This "GameManager" gameobject is located on "hub" scene and will be present also on gameplay maps



```
Inspector
GameManager                Static
Tag Untagged     Layer Default
Prefab  Open  Select  Overrides
Transform
Game Ticker (Script)
```

Offline scene is just scene that mirror will go to when client get disconnected, online scene is managed by

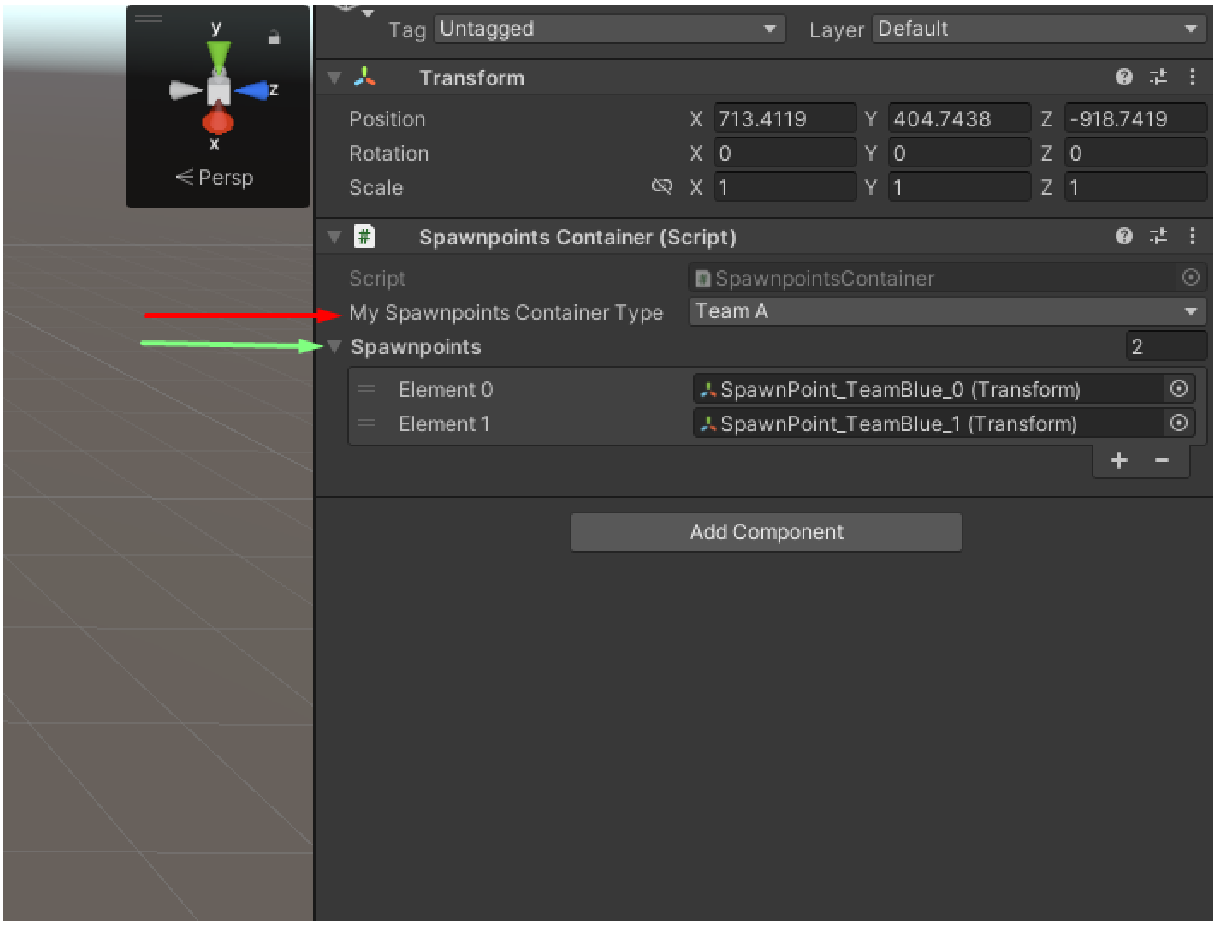
If You add any new object that have to be spawned in game for all participating players then you must add it to list “Registered Spawnable Prefabs”. First object in this list is always player prefab, so if you want to replace it, it must be also first prefab in this list.

6. How to add your custom map

There is also youtube tutorial that shows this process here

1. Create unity scene
2. Place your map in this scene (don’t forget to bake navigation map for it so bots will be able to navigate on your map, since they use Unity build-in navmesh system)
3. Create spawnpoints, as empty objects, their names does not matter.

4. MultiFPS categorizes spawnpoints for three types, and to group them apart we use another empty object with SpawnpointsContainer script attached to it. For gamemodes that use teams (for example Defuse, TeamDeathmatch, TeamEliminations) we will use group types TeamA and TeamB, to differentiate spawnpoints for each team. Example below shows setup for spawnpoints for team A (team blue in game)

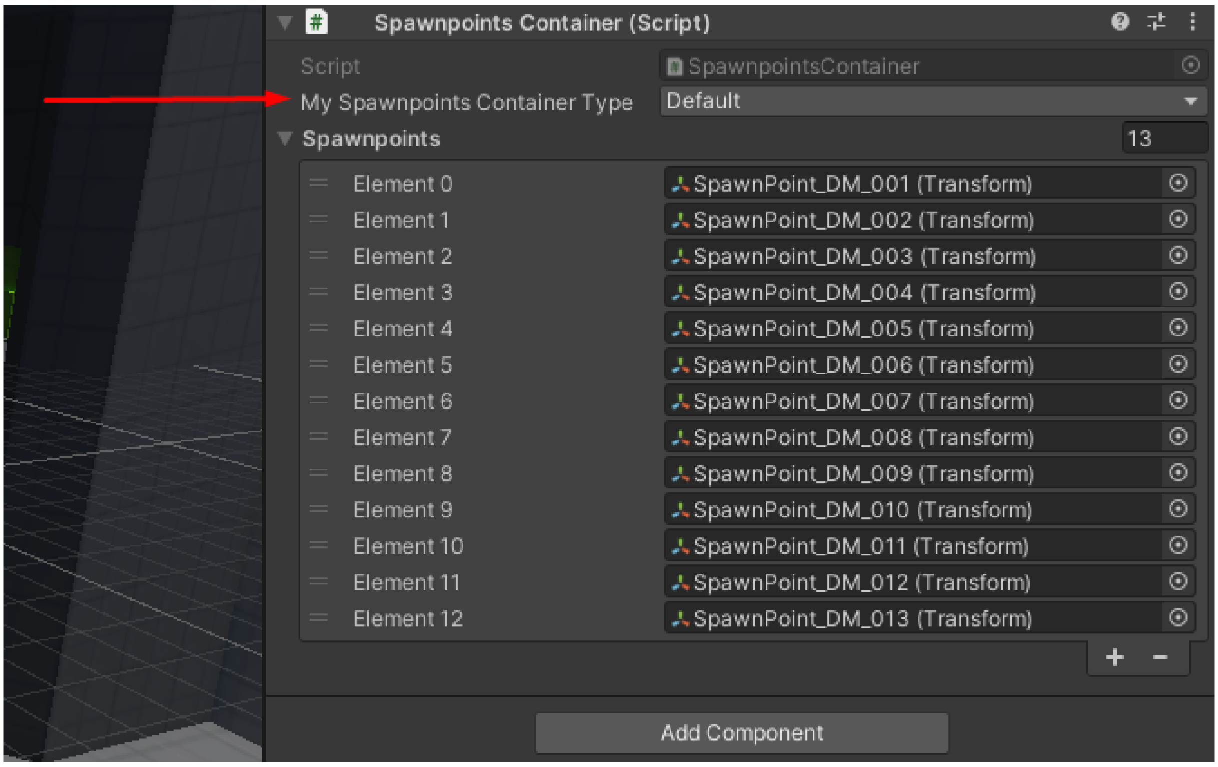


Tag: Untagged   Layer: Default

Transform
Position X 713.4119 Y 404.7438 Z -918.7419
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Spawnpoints Container (Script)
Script: SpawnpointsContainer
My Spawnpoints Container Type: Team A
Spawnpoints: 2
Element 0: SpawnPoint_TeamBlue_0 (Transform)
Element 1: SpawnPoint_TeamBlue_1 (Transform)

For gamemodes that don't have teams (regular Deathmatch) we will use type "Default". This will tell deathmatch gamemode that these are the spawnpoints for every player present in match


Spawnpoints Container (Script)

Script	SpawnpointsContainer
My Spawnpoints Container Type	Default
Spawnpoints	13
Element 0	SpawnPoint_DM_001 (Transform)
Element 1	SpawnPoint_DM_002 (Transform)
Element 2	SpawnPoint_DM_003 (Transform)
Element 3	SpawnPoint_DM_004 (Transform)
Element 4	SpawnPoint_DM_005 (Transform)
Element 5	SpawnPoint_DM_006 (Transform)
Element 6	SpawnPoint_DM_007 (Transform)
Element 7	SpawnPoint_DM_008 (Transform)
Element 8	SpawnPoint_DM_009 (Transform)
Element 9	SpawnPoint_DM_010 (Transform)
Element 10	SpawnPoint_DM_011 (Transform)
Element 11	SpawnPoint_DM_012 (Transform)
Element 12	SpawnPoint_DM_013 (Transform)

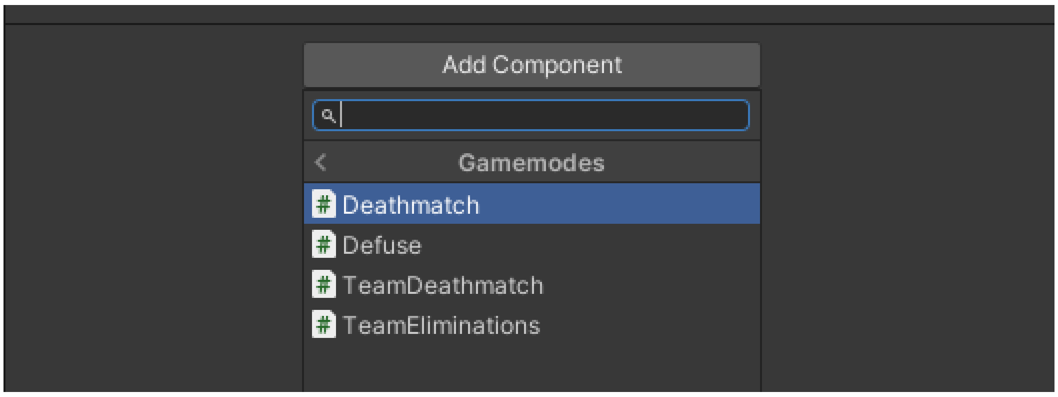
Add spawnpoints and their containers according to your needs, You can also have all three spawnpoints type present if Your map support multiple types of gamemodes at the same time.

Maps "BombBuilding" and "CircleArena" have all three types of spawnpoints, since they both support team based, and non-team based gamemodes.

5. Crate empty gameobject. Name it "RoomManager", name is not crucial.

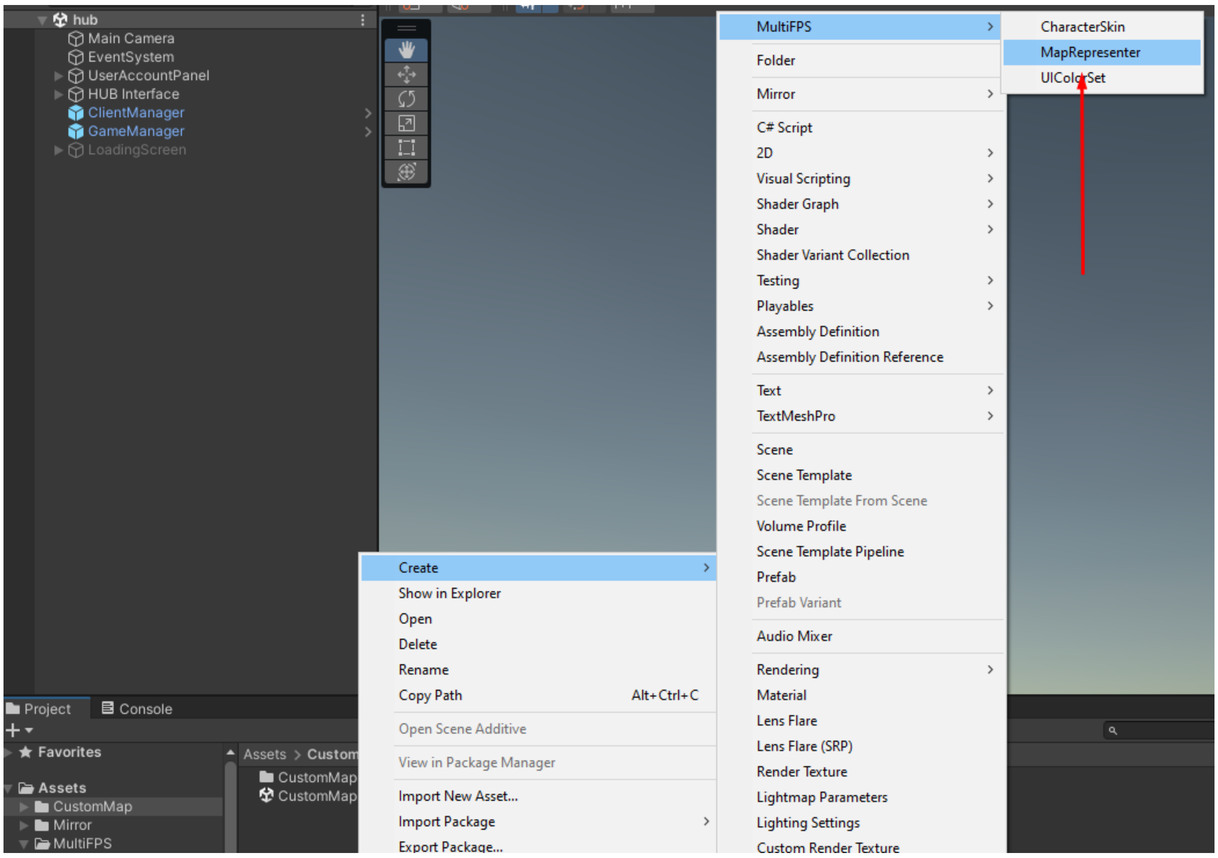
Add to this object RoomManager script and your desired gamemodes, they can be easily found by clicking:

Add Component -> MultiFPS -> Gamemodes



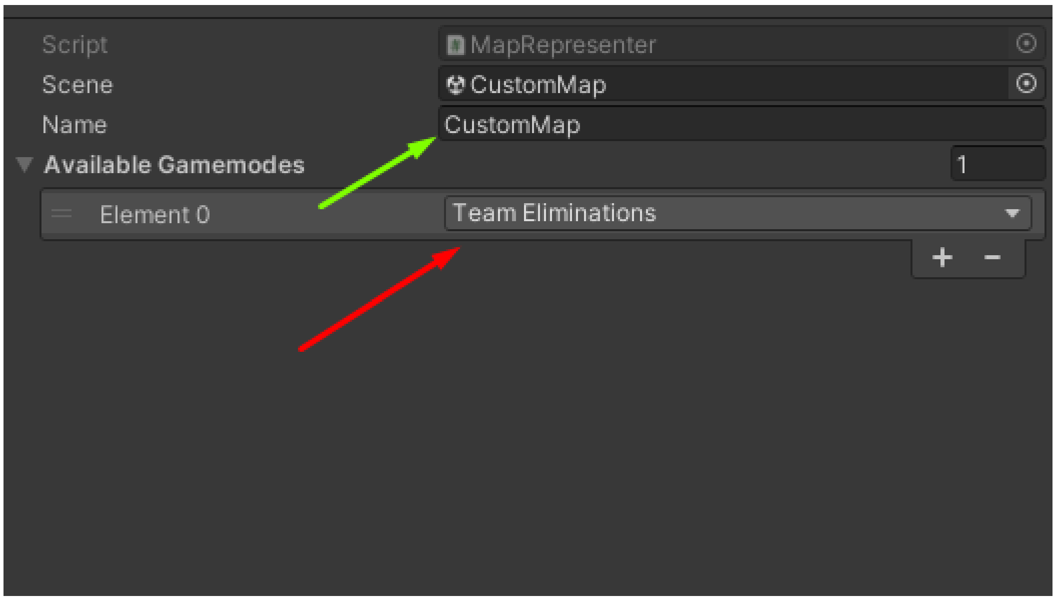
RoomManager will read properties set by player in main menu and apply them.

6. Map is ready, now we have to make it accessible for user to choose, to do that we need to create new file



Right click on project tab, and go select it Create->MultiFPS->MapRepresenter

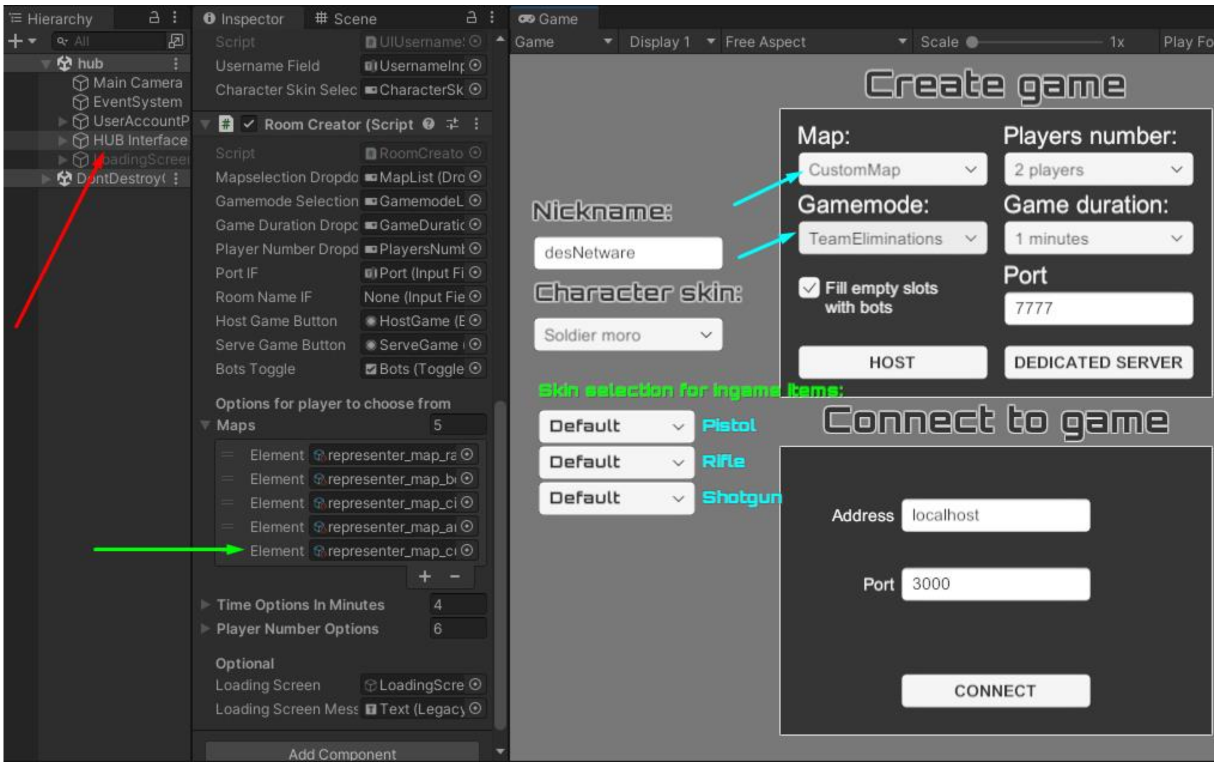
In the inspector of this object setup information about Your map



Script	MapRepresenter
Scene	CustomMap
Name	CustomMap
Available Gamemodes	1
Element 0	Team Eliminations

Name field is just for user, what he will see in map list, and available gamemodes list is what he will be able to choose for this map, so add here only gamemodes that we set in step 5.

7. Load hub scene (this is scene where player creates game)
   Go to HUB Interface gameobject, and in script RoomCreator add map representer created in step 6 to Maps list. Now if we hit play we will be able to see our new map and gamemodes that we can choose for it. After selecting your map click HOST button to launch the game.



Hierarchy: hub — Main Camera, EventSystem, UserAccountP, HUB Interface, LoadingScree, DontDestroy(

Inspector — Room Creator (Script)

Script	UIUsername
Username Field	UsernameInp
Character Skin Selec	CharacterSk
Script	RoomCreato
Mapselection Dropdo	MapList (Dro
Gamemode Selection	GamemodeL
Game Duration Dropc	GameDuratic
Player Number Dropd	PlayersNumt
Port IF	Port (Input Fi
Room Name IF	None (Input Fie
Host Game Button	HostGame (E
Serve Game Button	ServeGame
Bots Toggle	Bots (Toggle

Options for player to choose from
Maps 5
Element representer_map_ra
Element representer_map_bi
Element representer_map_ci
Element representer_map_ai
Element representer_map_ci
Time Options In Minutes 4
Player Number Options 6
Optional
Loading Screen LoadingScre
Loading Screen Mess Text (Legacy

Game — Display 1 — Free Aspect — Scale 1x

Create game
Map: CustomMap
Players number: 2 players
Gamemode: TeamEliminations
Game duration: 1 minutes
Fill empty slots with bots ✓
Port 7777
Nickname: desNetware
Character skin: Soldier moro
HOST
DEDICATED SERVER
Skin selection for ingame items:
Default Pistol
Default Rifle
Default Shotgun
Connect to game
Address localhost
Port 3000
CONNECT

Thank You for using MultiFPS

Contact:

Email: desnetware@gmail.com

Website

Youtube channel

AssetStore publisher page