Multiplayer Team Deathmatch

Sviluppato in Unreal Engine 5.3

Focus

Il progetto è nato con l'obbiettivo di studiare aspetti specifici relativi i sistemi distruibiti all'interno del contesto di Unreal Engine 5, tra i quali:

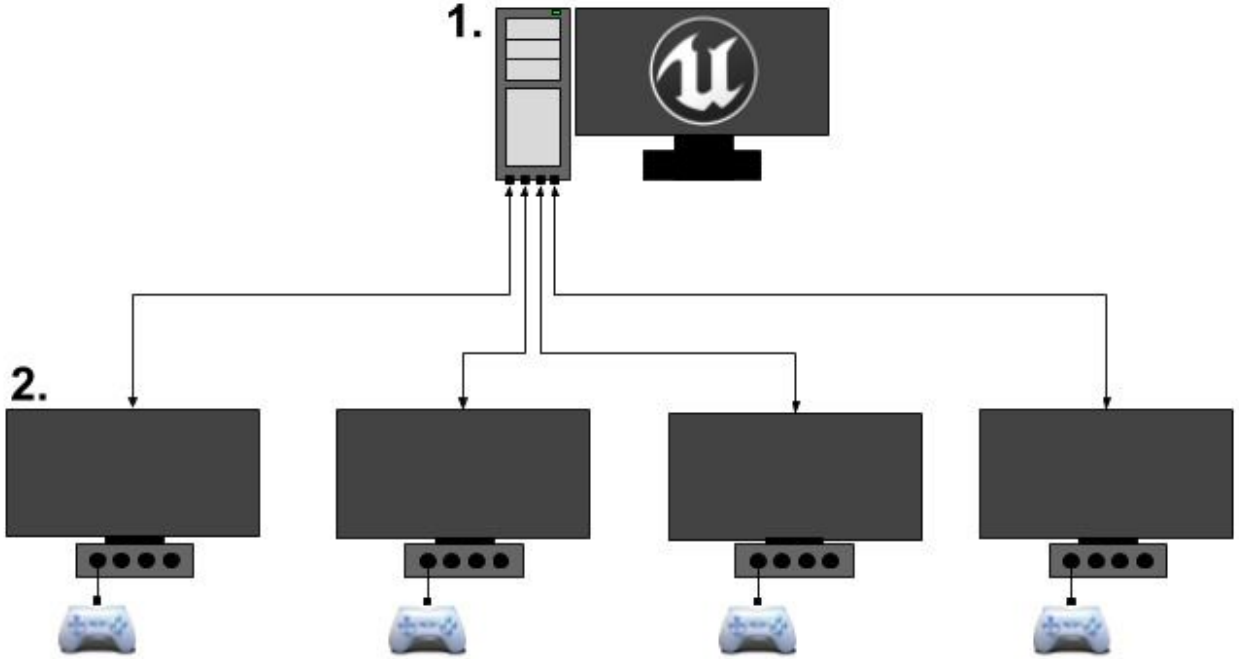
- Sistema di Replicazione tramite RepNotifiers.
- Uso di RPC.
- Concetto di Sessione, tramite Hosting e Join.
- Gestione dei client e del server.

Ciò che è stato implementato.

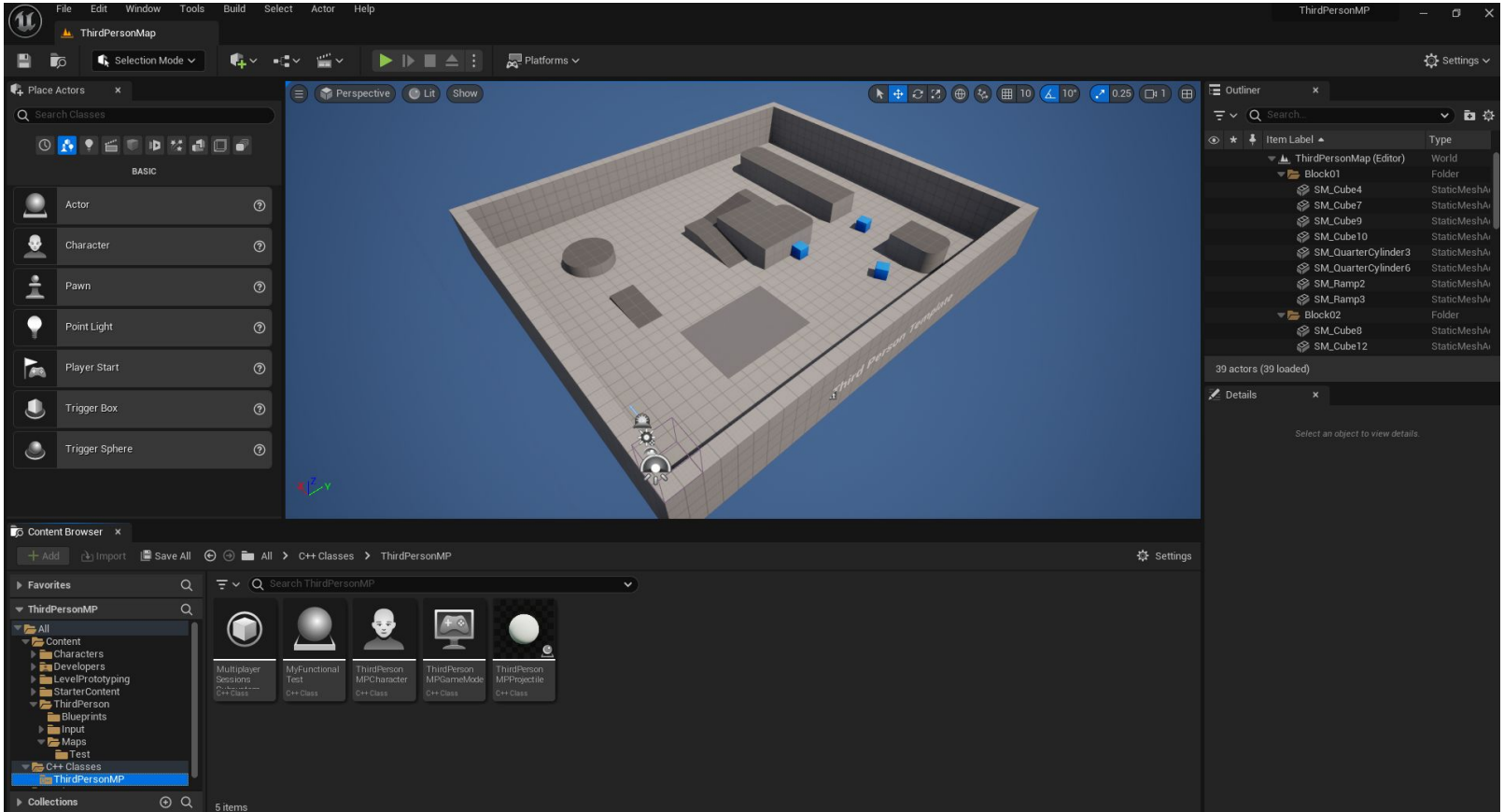
All’interno del gioco, sia per motivi di tempo che per rispettare i requisiti minimi per poter avere una base giocabile i seguenti meccanismi sono stati sviluppati:

- Integrazione di un Main menù.
- “Spawn” del player.
- Assegnamento di un player ad una squadra.
- Muoversi all’interno della scena di gioco.
- Sparare proiettili.
- Fare subire danni al giocatore.
- Morte del giocatore.
- Pulsante Exit.

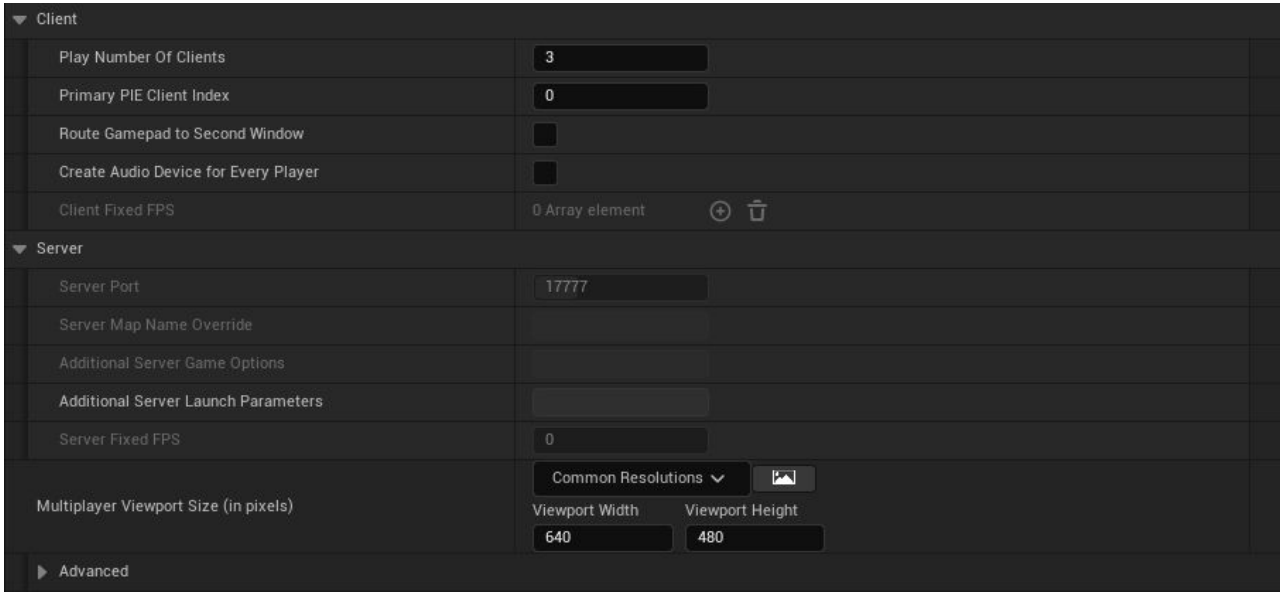
Architettura Client-Server



Unreal Engine 5 - Vista



Unreal engine e il multiplayer.



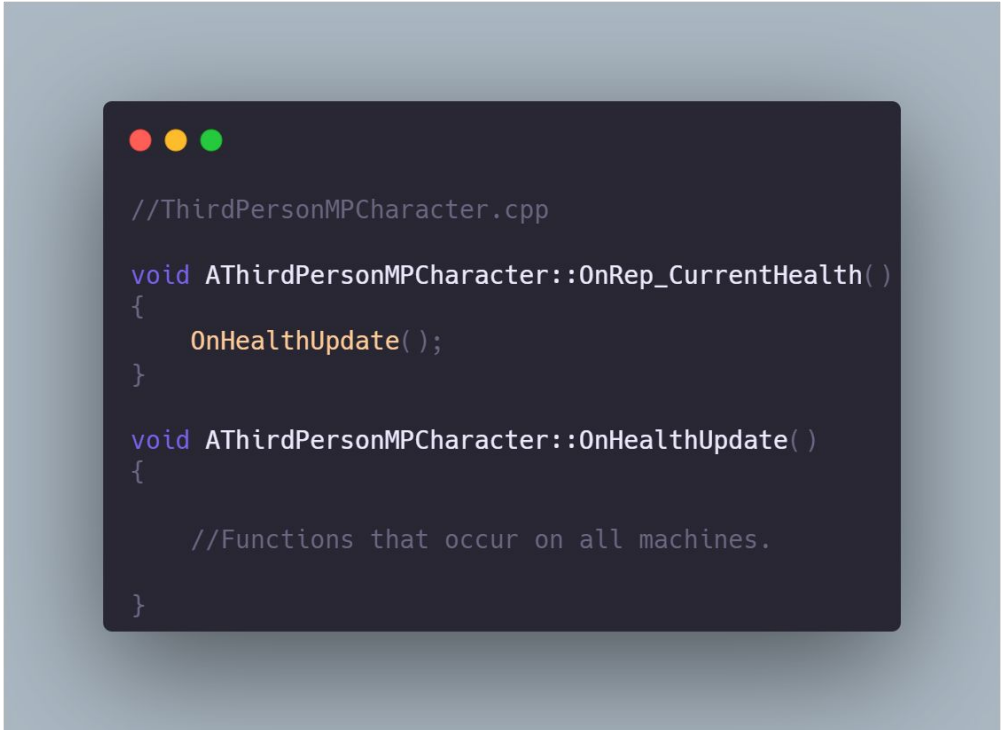
Implementazioni di rilievo - RepNotify

```
//ThirdPersonMPCharacter.h

/** The player's maximum health. This is the highest that their health can be, and the value that their
health starts at when spawned.*/
UPROPERTY(EditDefaultsOnly, Category = "Health")
float MaxHealth;

/** The player's current health. When reduced to 0, they are considered dead.*/
UPROPERTY(ReplicatedUsing=OnRep_CurrentHealth)
float CurrentHealth;

/** RepNotify for changes made to current health.*/
UFUNCTION()
void OnRep_CurrentHealth();
```

Implementazioni di rilievo - RepNotify

```
//ThirdPersonMPCharacter.cpp

void AThirdPersonMPCharacter::GetLifetimeReplicatedProps(TArray <FLifetimeProperty> & OutLifetimeProps)
const
{
    Super::GetLifetimeReplicatedProps(OutLifetimeProps);

    //Replicate current health.
    DOREPLIFETIME(AThirdPersonMPCharacter, CurrentHealth);
}
```

Implementazioni di rilievo - RepNotify



```cpp
//ThirdPersonMPCharacter.cpp

void AThirdPersonMPCharacter::OnRep_CurrentHealth()
{
    OnHealthUpdate();
}

void AThirdPersonMPCharacter::OnHealthUpdate()
{

    //Functions that occur on all machines.

}
```

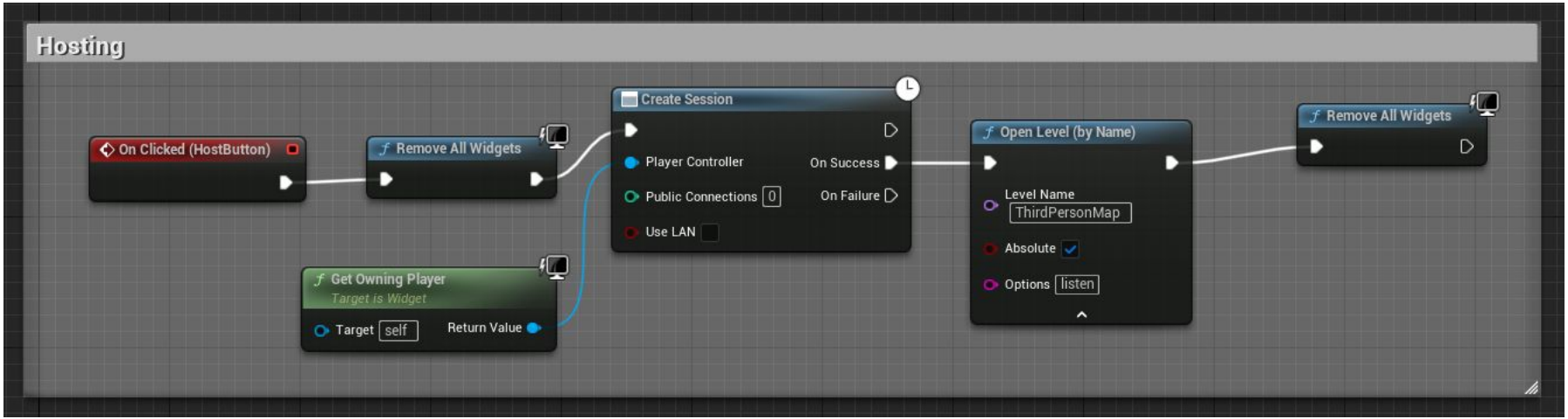
Implementazioni di rilievo - RPC

```
//ThirdPersonMPProjectile .h
UFUNCTION(Server, Reliable)
void HandleFire();

//ThirdPersonMPProjectile.cpp
void AThirdPersonMPCharacter::HandleFire_Implementation()
{
    FVector spawnLocation = GetActorLocation() + ( GetControlRotation().Vector()  * 100.0f ) +
                            (GetActorUpVector() * 50.0f);
    FRotator spawnRotation = GetControlRotation();

    FActorSpawnParameters spawnParameters;
    spawnParameters.Instigator = GetInstigator();
    spawnParameters.Owner = this;

    AThirdPersonMPProjectile* spawnedProjectile = GetWorld()->SpawnActor<AThirdPersonMPProjectile>
(spawnLocation, spawnRotation, spawnParameters);
}
```

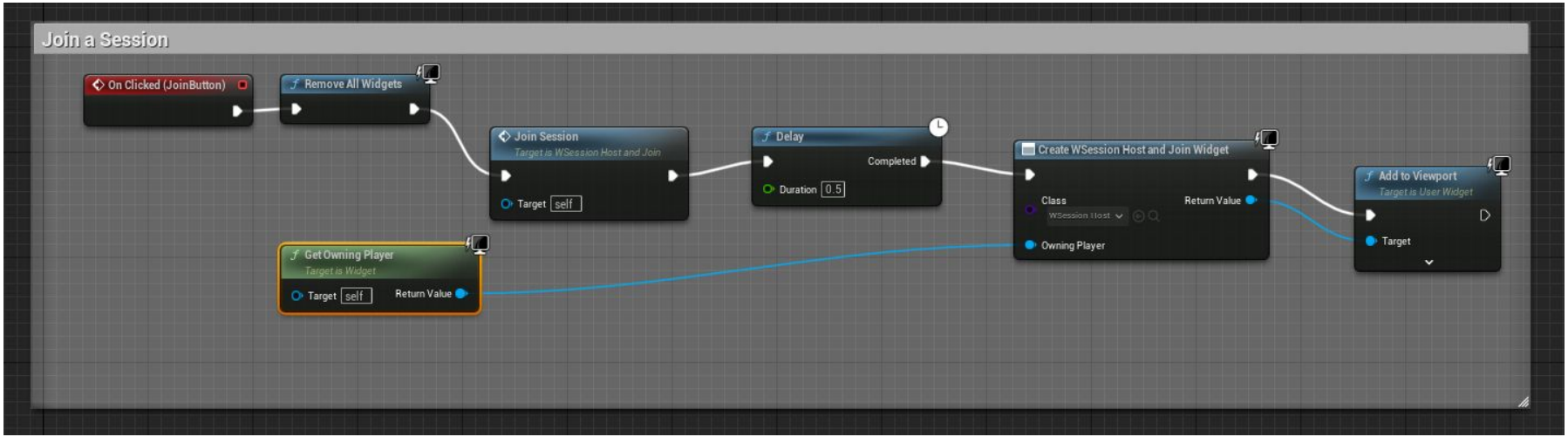
Blueprints


//UMultiplayerSessionsSubsystem.h
UFUNCTION(BlueprintCallable)
void CreateSession(FString ServerName);
UFUNCTION(BlueprintCallable)
void FindSession(FString ServerName);

Blueprints - Host



Blueprints - Join



Fine

Vedere la demo.