Hi,
Thank you for purchase,
I hope, my package will help you.

It is a template for creating multiplayer PvP games like Clash Royale with deterministic game mechanics.

Note!
Package is required PUN 2 you need to instal this package in your project

Features
- Only send "hero creating" input between players
- Deterministic game mechanics

How the package works

The package has two scenes - Lobby and Game
On the Lobby scene there is a two scripts for Network part
Class PunNetworkManager working with PUN 2
This script is used for connecting to the server PhotonNetwork.ConnectUsingSettings() , creating or joining room PhotonNetwork.JoinOrCreateRoom and sending messages between players public void RPC(string methodName, RpcTarget target, params object[] parameters)

The class NetworkMessenger   receive messages from the network and call his events that are processed by GameManager

The class GameManager on the "Game" scene creates two groups "class Group" with Towers "TowerManager" and Heroes "HeroManager" and sets them on the arena groups[i].Init2Group(arena, i);

The mechanics of the game is deterministic because all game logic (heroes creating time and heroes moving) working with long (Fixed-point arithmetic)

When user want to create "Hero" , the "Hero" will be added to the waiting list

 uiManager.OnCreateHero += AddMyHeroToWaitingList; local user creating and
PunNetworkManager.NetworkManager.Messenger.OnAddOtherHeroToWaitingList += AddOtherHeroToWaitingList;   creating from other user through the network

long addTime
MyGroup.AddHeroToWaitingList(index, addTime, position);
OtherGroup.AddHeroToWaitingList(index, addTime, position);

After adding "Hero" to the waiting list, in the private void UpdateGame()
The "Heroes" will be created at a time that is equal to the time which saved in the
"HeroesWaitingList"
long GameTime
if (groups[0].HeroesWaitingList.ContainsKey(GameTime))
if (groups[1].HeroesWaitingList.ContainsKey(GameTime))

How you can modify the contents

Please look at the clases

class HeroShooterManager, class SimpleHeroManager and class
SimpleTowerManager which is override the abstract classes
HeroManager and TowerManager

As you can see, you can override the methods
public abstract void OnFight();
public abstract bool CanBeAsTarget(HeroManager target);

So, you can create your own "Hero Manager" and for example play his animation when
"Hero" is fighting and determine is any "Hero" can be as target "Hero" for your new
"Hero" (Can your new hero damage present "HeroManager target" or not?)