Sky Arena Photon Tutorial Overview

Part 1: Part of this package (enhanced continuously) *AVAILABLE NOW*
Part 2: Part of this package (enhanced continuously) *AVAILABLE NOW*

Hi,

Thank you for purchasing the source code for this project. We will be able to make more tutorials just like these because of you, and we'll also be able to expand on this tutorial:

So it's a win for everyone ☺.

This text should serve as a small introduction into the project structure, so you will be able to find your way around. **The real documentation is happening right in the source code**. I've tried to write very clean code that should be easy to read and I've inserted comments wherever I wanted to explain my trains of thought.

Please let me know if there are sections that aren't very well documented and that could use more explanations. Please post your feedback here: http://forum.unity3d.com/threads/241158-New-Photon-Unity-Networking-Tutorial-Series

There are also **several YouTube Videos** that explain the Photon concepts and their implementation in this demo in more detail and they should help you get started with the most important Photon concepts.

You can find the videos of Part 1 here:

1 - OnPhotonSerializeView: http://youtu.be/8St-bqfmufA

2 - Synchronizing Player Ships: http://youtu.be/7hWuxxm6wsA

3 - RPCs: http://youtu.be/ozBmZ9FoN_o

4 - SceneObjects and buffered RPCs: http://youtu.be/Wn9P4d1KwoQ

And the videos of part 2 can be found here:

1 - Custom Properties and Room Browser: http://youtu.be/ahSxvClaLGc

2 - Custom Player Properties and Gamemodes: http://youtu.be/7hWuxxm6wsA

3 - Matchmaking: http://youtu.be/ozBmZ9FoN_o

4 - Chat and Friend List: http://youtu.be/Wn9P4d1KwoQ

Thank you again for supporting this project and stop by forum.exitgames.com and tell us about the great multiplayer games you are working on. We will also be able to help you with any questions about Photon in general, or this tutorial, over in our forums.

Take care,

Oliver

Setup

Before you can build this game for yourself, you need to update the Input Manager with the correct values; otherwise controller support and custom tags will break the game. We've included the correct project settings with this package, but we cannot set this up automatically. You have to open the zip file "./Assets/Setup/ProjectSettings.zip" and unpack its content to "./ProjectSettings/"

After you've restarted Unity, the project will build correctly.

Now you

Package Contents

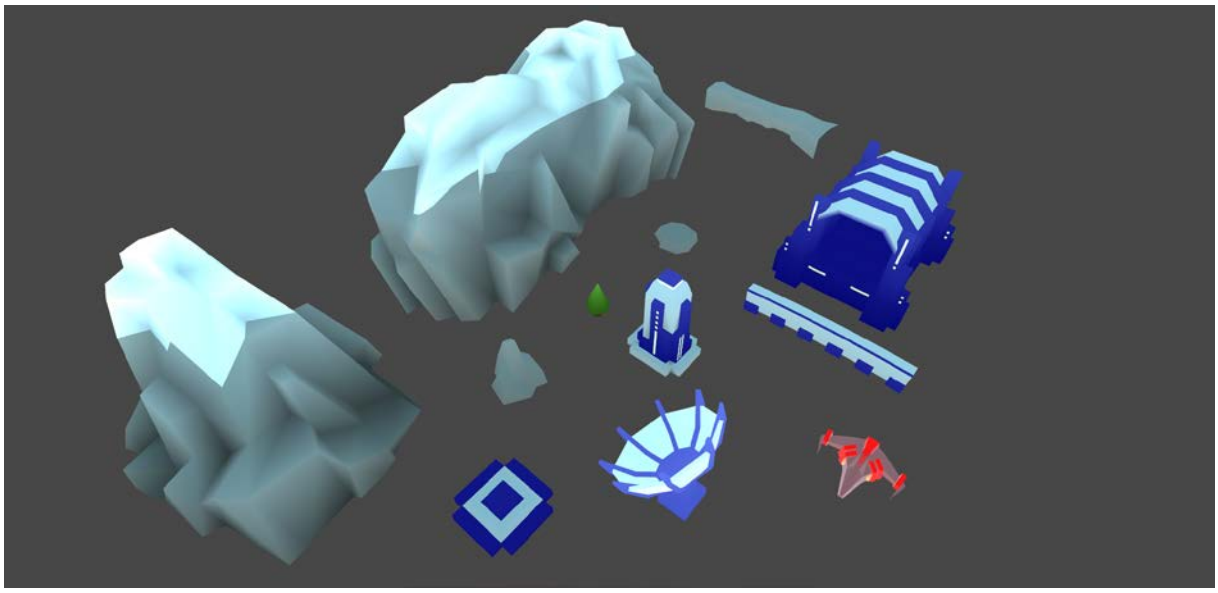
The content of this package is organized in "Parts". We finished Part 1 and Part 2 and they're both included in this package

Part 1 Content (Released! Part of this Package)

- OnPhotonSerializeView
    - Learn how to continually synchronize GameObjects between multiple online clients
    - Useful for objects like player characters, enemies, moving obstacles etc.
- RPCs
    - How to synchronize one-off events that happen at a specific time

    - This handles events like shooting lasers or picking up powerups from the scene
- SceneObjects
    - We show you how Photon differentiates between different types of GameObjects
    - Create important objects that down belong to a player like doors, pickups or enemies

Part 1 Assets



Part 2 Content (Released! Part of this package)

- Chat
    - Using Photon Chat
    - Independent of rooms and filtered by channel
    - Friend list that keeps track of your friends status
- Custom Properties

        - Can be used for rooms and players
        - They're used to synchronize data between all clients like player- or team scores
- Multiple Maps
        - How to sync loading
        - Using them in matchmaking
- Game Modes
        - Capture the Flag
        - Free For All Deathmatch
        - Team Deathmatch
- Join and Create Rooms
        - Multiple matchmkaing algorithms
        - Use custom properties or complex SQL filters to find the best server
        - Sortable room browser
        - Create your own room and customize its map queue

Project Structure

Assets/Editor/

- I didn't use any editor scripts in this demo, except for one very helpful one which I use in every project. It creates an "Open Scene" menu item in the top menu so I'm able to quickly access the most important scenes. Talk about a time saver ;-) This script is a gift from me for you

Assets/Fonts/

- I am using two very beautiful, futuristic fonts for Sky Arena. The best part: both are Open Source. Jura was created by Daniel Johnson can be downloaded here: http://www.google.com/fonts/specimen/Jura
  Orbitron, by Matt McInerney, can be found here: https://www.theleagueofmoveabletype.com/orbitron

Assets/Materials/

- Just a couple of materials I've created manually for the project

Assets/Models/

- All the 3d Assets are located in this folder

Assets/Models/Environment/

- The mountains, trees, rocks and buildings

Assets/Models/FX/

- Models used in some of the effects

Assets/Models/Pickup/

- Models for the pickups

Assets/Models/Projectiles/

- The laser model. Maybe we can add more projectiles in a future lesson

Assets/Models/Ship/

- The player ship

Assets/Photon Unity Networking/

- All the Photon related code. If you download the free Photon Package from the Unity Asset store, you will get this folder with all of its contents too. The Photon documentation can be found in here as well

Assets/Plugins/

- The compiled Photon plugins are put here

Assets/Prefabs/

- Here you'll find all the prefabs that don't need to be Instantiated at runtime by Photon. These are mostly effects prefabs and one side of the level. I've put the level into a prefab so that I can make it mirror very easily. I simple create two instances of this prefab and rotate one by 180°. Instant mirrored level ;-)

Assets/Resources/

- The prefabs in this folder will be instantiated by Photon at runtime. The Resources folder is a special Unity folder which allows you to access its contents at runtime with Resources.Load
  You can find more information here about this in the Unity documentation:
  https://docs.unity3d.com/Documentation/ScriptReference/Resources.html
- Note: Photon requires the prefabs it has to Instantiate to be in the root of a Resources folder

Assets/Scenes/

- The scenes that are used in the project

Assets/Scripts/

- This is the good stuff. All the game code is found here

Assets/Scripts/Camera/

- The camera follow script is found here

Assets/Scripts/Chat/

- Everything related to chat is found here

Assets/Scripts/FX/

- All effects and particle related stuff

Assets/Scripts/Gamemodes/

- The gamemode behaviours can be found here

Assets/Scripts/General/

- All the stuff that I was too lazy to organize into folders ☺

Assets/Scripts/GUI/

- All GUI related scripts

Assets/Scripts/Objects/

- Some helper scripts that can be used on level objects

Assets/Scripts/Pickups/

- There are health pickups and the team flags in this game, which you can find here

Assets/Scripts/Projectiles/

- Everything needed to create deadly weapons

Assets/Scripts/Ship/

- This is where the meat of the project is. The ship behaviours make up roughly 50% of all the code of this project ☺ Ship Movment, Collision, Shooting etc. I all tucked away here

Assets/Setup/

- You'll find a ProjectSettings.zip file in here. Unzip its contents into your projects ProjectSettings folder to setup the project properly. Without doing this you won't be able to play the project in your own editor.

Assets/Shader/

- I'm using a couple of custom shaders in this project to get the Sky Arena look. Most of them were created for the main Sky Arena game and tweaked for this demo. Enter at your own risk ☺

Assets/Standard Assets/

- I’m using the default Unity projectors for the shadows of the ships and the relevant assets are found here

Assets/Textures/

- All the 2D Glory. This folder is pretty self explanatory

Script Overview

Again, please feel free to browse the source code itself. I tried to make it as easy to read as possible.

The general flow of the game is quite simple:

1) We start in *MainMenu.cs* and wait for the user to press anything
2) We then connect to the server through *MultiplayerConnector.cs* and either Join or Create the "Tutorial" game, depending on if it exists or not
3) Once the level is loaded, the Script *PlayerSpawner.cs* creates a new ship instance for the local client, which is then automatically created on all the other connected clients as well, because we are using PhotonNetwork.Instantiate

During gameplay, the script *CaptureTheFlagGamemode.cs* is checking for the win conditions and asks the MasterClient to restart the match if one has finished. It is attached to the Main Camera.

We've dedicated YouTube videos to all other three big areas in this demo:

- Syncronizing the Player Ships Overview: http://youtu.be/8St-bqfmufA
- Syncronizing the Player Ships Implementation: http://youtu.be/7hWuxxm6wsA
- Shooting and hitting with Lasers: http://youtu.be/ozBmZ9FoN_o
- Picking up Health and the Flags: http://youtu.be/Wn9P4d1KwoQ

Roadmap

We want to expand upon this Tutorial to create a fully functioning multiplayer game, but we would like to hear your feedback to get an idea about what you would like to learn next. Please post your feedback to our forum post at http://forum.unity3d.com/threads/241158-New-Photon-Unity-Networking-Tutorial-Series

Again, thank you for reading and have fun creating your own multiplayer games.