Multi-VR Users Tutorial

IDVR 2018

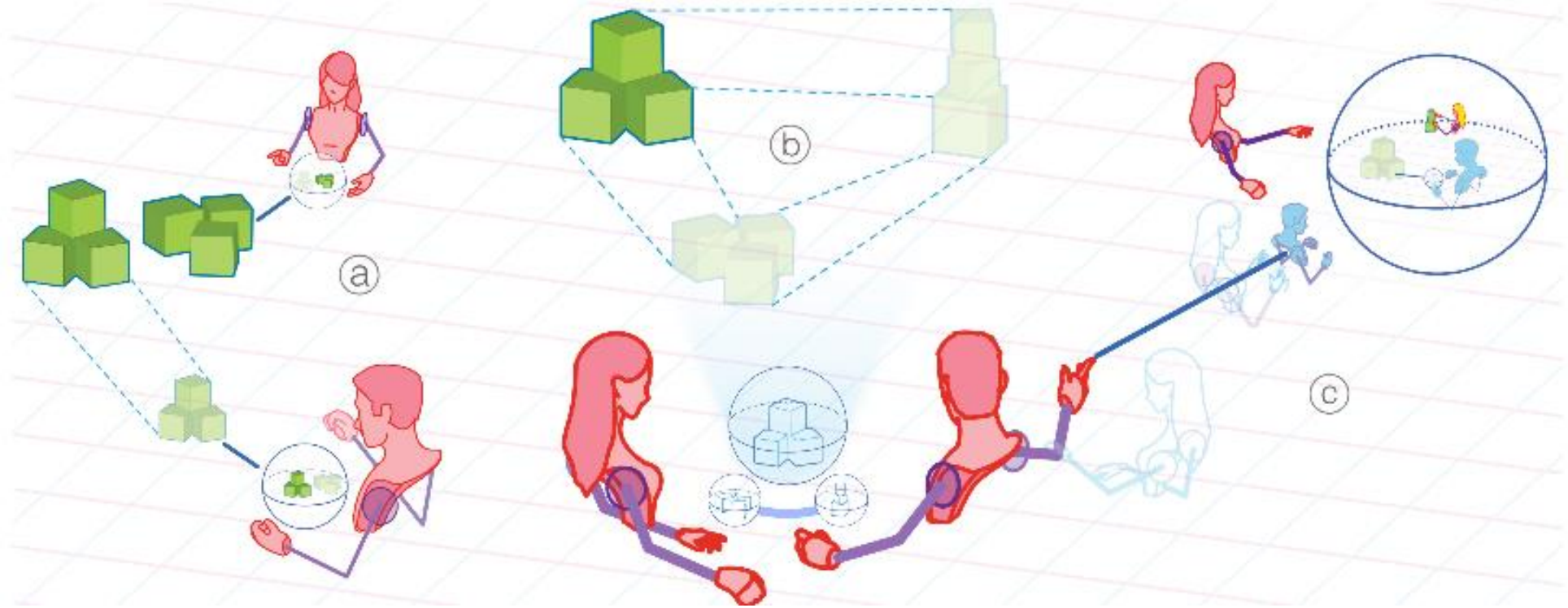
Why Multi-VR Users?

Social Media (Facebook Social VR)

Entertainment

Development

- Spacetime



Haijun Xia, Sebastian Herscher, Ken Perlin, and Daniel Wigdor. 2018. Spacetime: Enabling Fluid Individual and Collaborative Editing in Virtual Reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (UIST '18). ACM, New York, NY, USA, 853-866. DOI: https://doi.org/10.1145/3242587.3242597

Photon

photon

PRODUCTS

SDKs Documentation Sign In

The world's #1 independent networking engine and multiplayer platform — Fast, reliable, scalable.
Made for anyone: indies, professional studios and AAA productions.

We Make Multiplayer Simple

Join 315,197
Developers Using Photon

TRY PHOTON **FREE**

EXIT GAMES

PUN 2 - FREE

FREE

★★★★☆ 17 user reviews

Add to My Assets

Note: Contains breaking changes to existing PUN Classic (v1) projects. Please read our PUN 2 migration notes. In doubt: keep PUN Classic.

Photon Unity Networking 2:
NEW demos, cleaner API & structure, clear separation of PUN and Realtime API

Export to all platforms: mobile, desktop, consoles (including Playstation, Xbox & Nintendo Switch), TV, VR, AR & web.

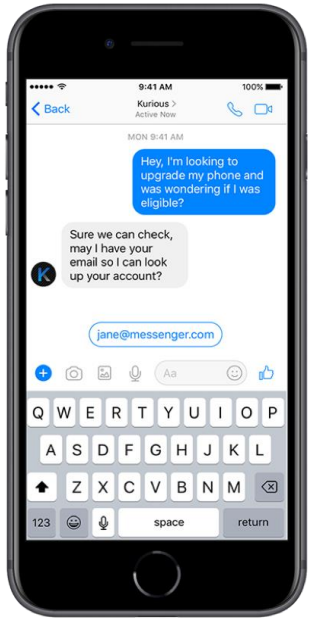
PUN 2 is all you need to easily add multiplayer to your games and launch them globally.
Forget about hosting, connection issues and latency.

>> Global low latency: Photon Cloud hosting centers in North & South America, Europe, Asia & Australia provide low

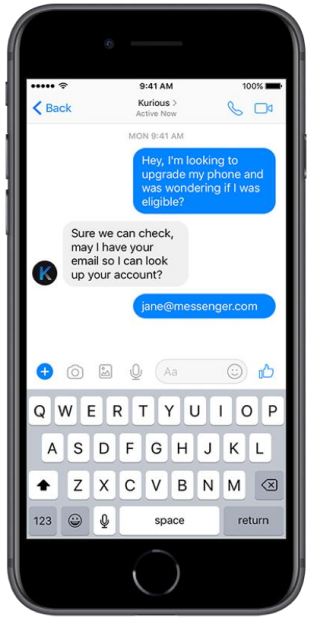
Photon Unity Networking (PUN) 2.0 Migration:
https://doc.photonengine.com/en-us/pun/v2/getting-started/migration-notes

Demo: Snowball Fight

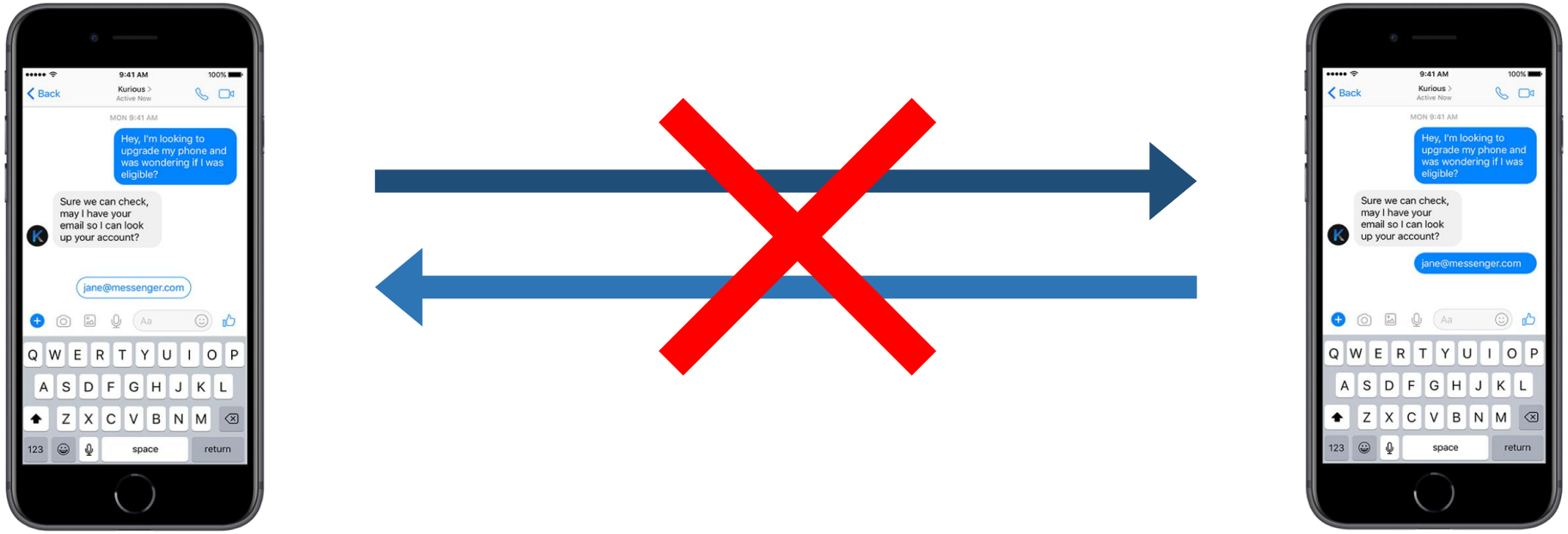
Basic Multiuser Network Structure


9:41 AM
100%
Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com
Aa
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return

A

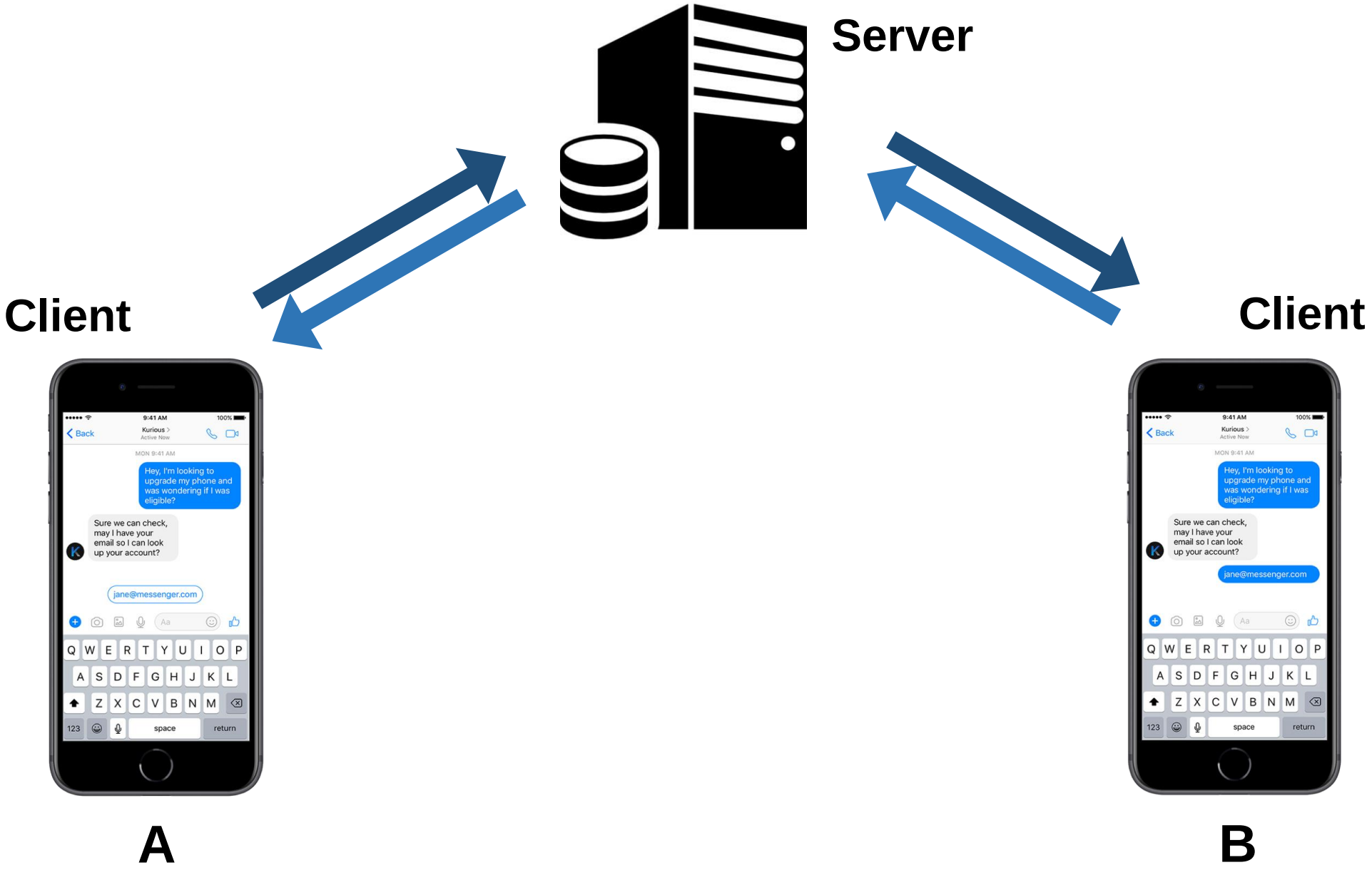

9:41 AM
100%
Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com
Aa
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return

B

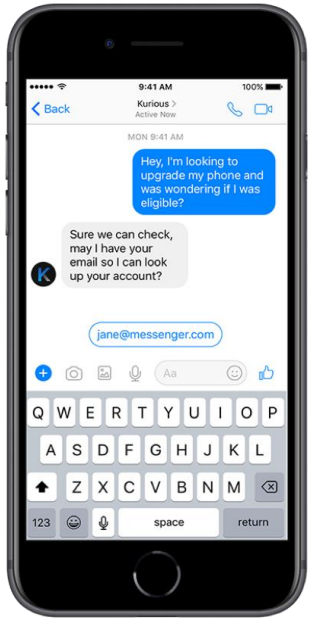

9:41 AM
100%
Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com
Aa
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return

A

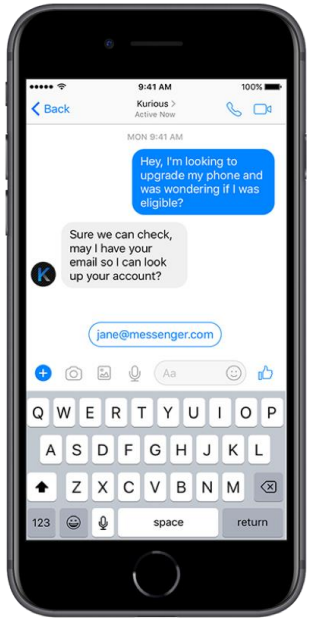
B


Server
Client
Client
Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com
A
B

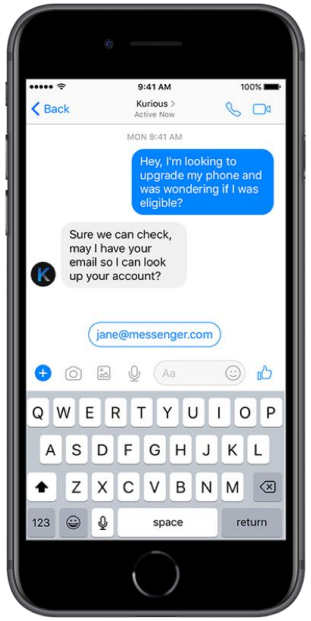
Server


Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com

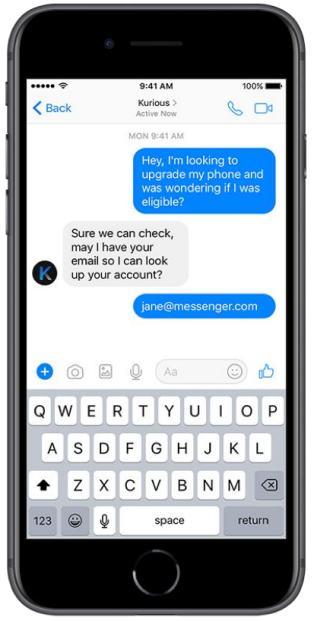
A


Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com

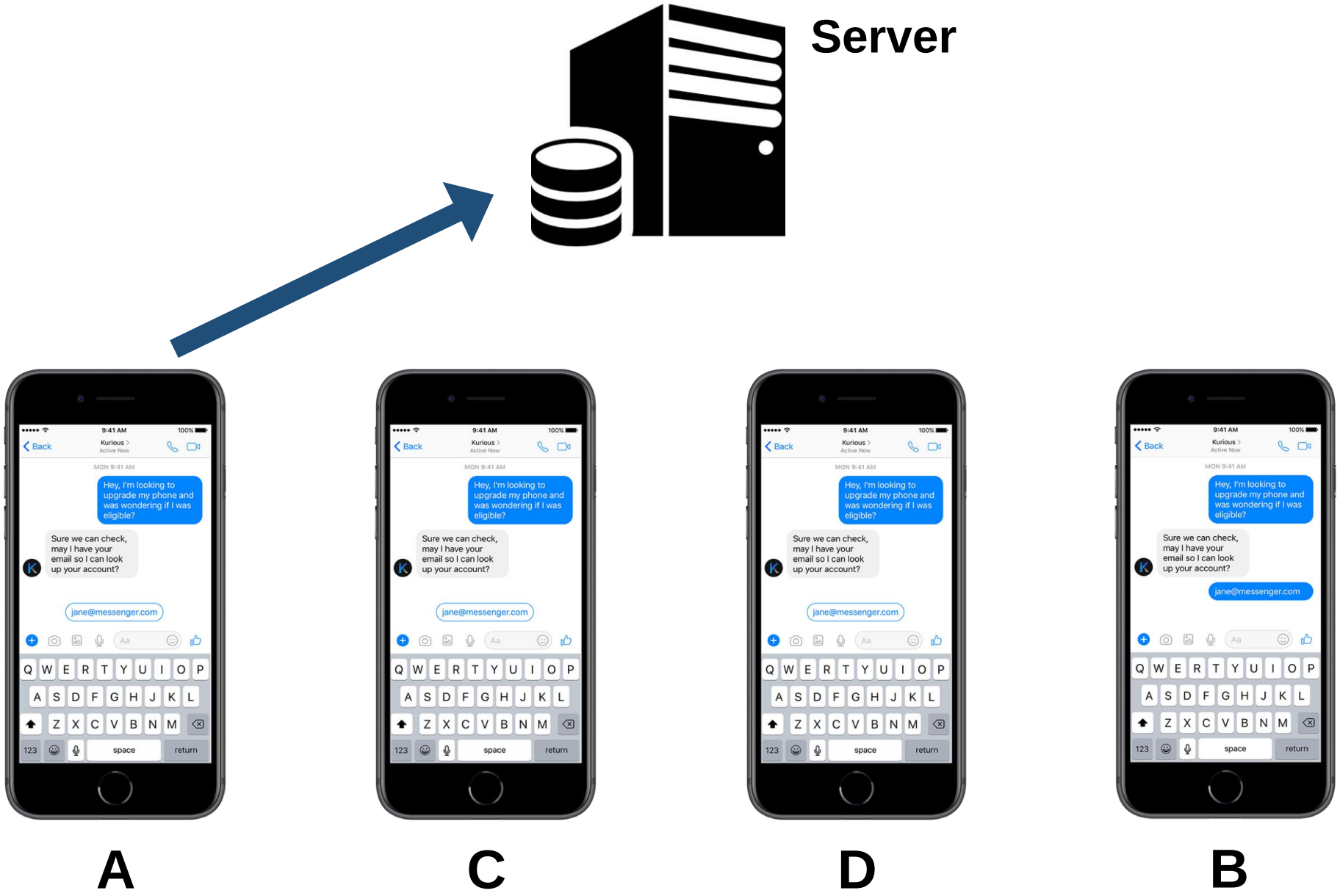
C

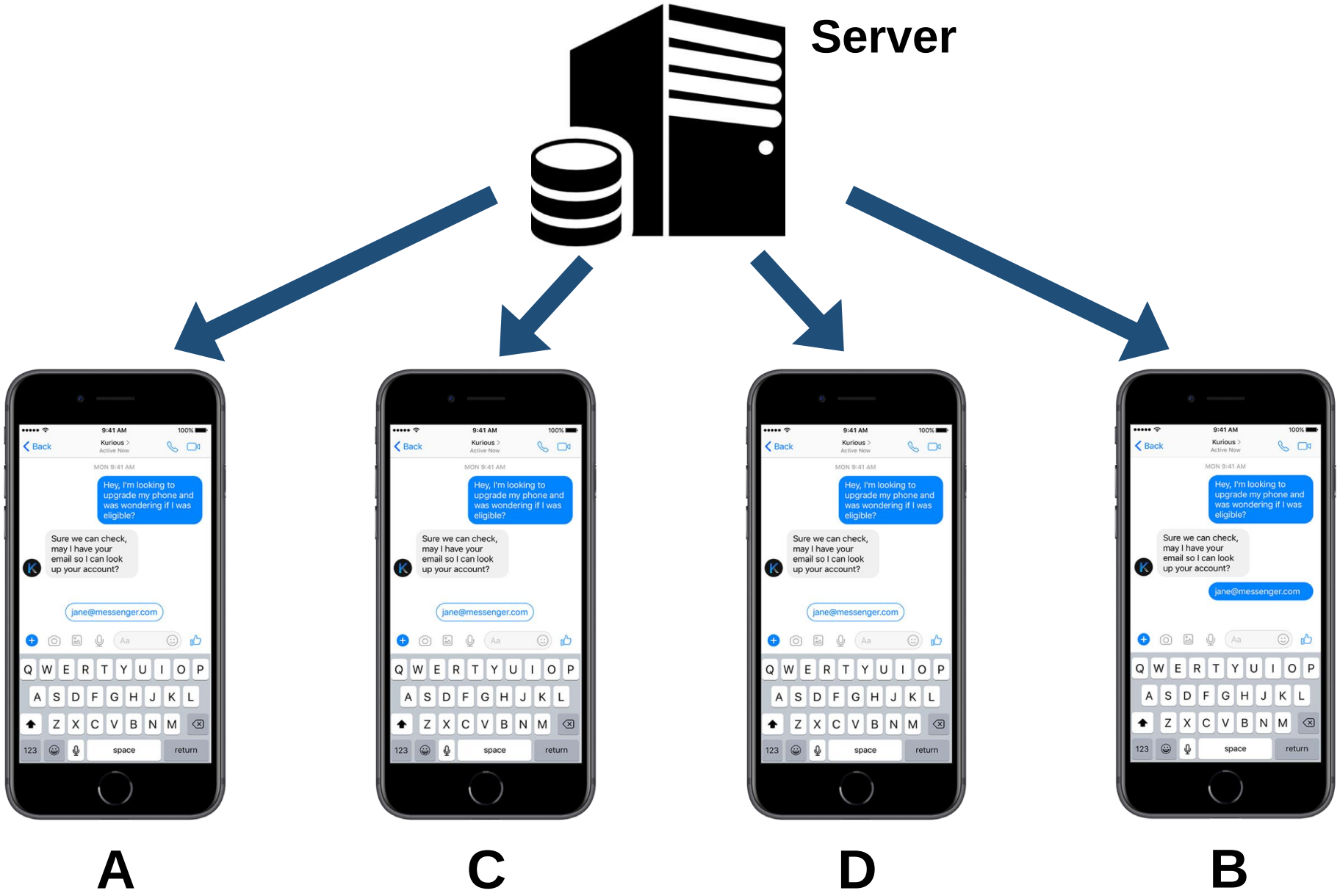

Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com

D

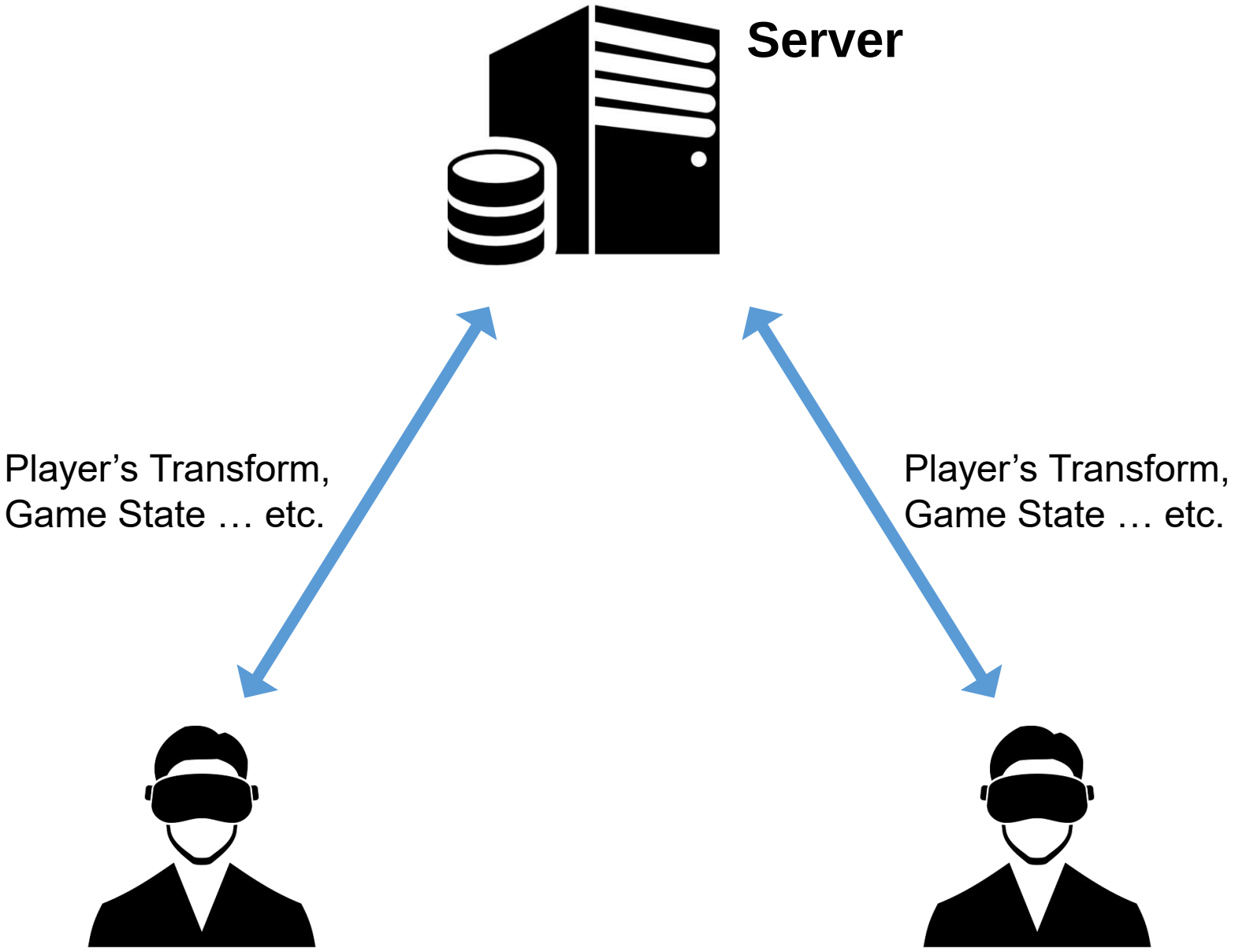

Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com

B

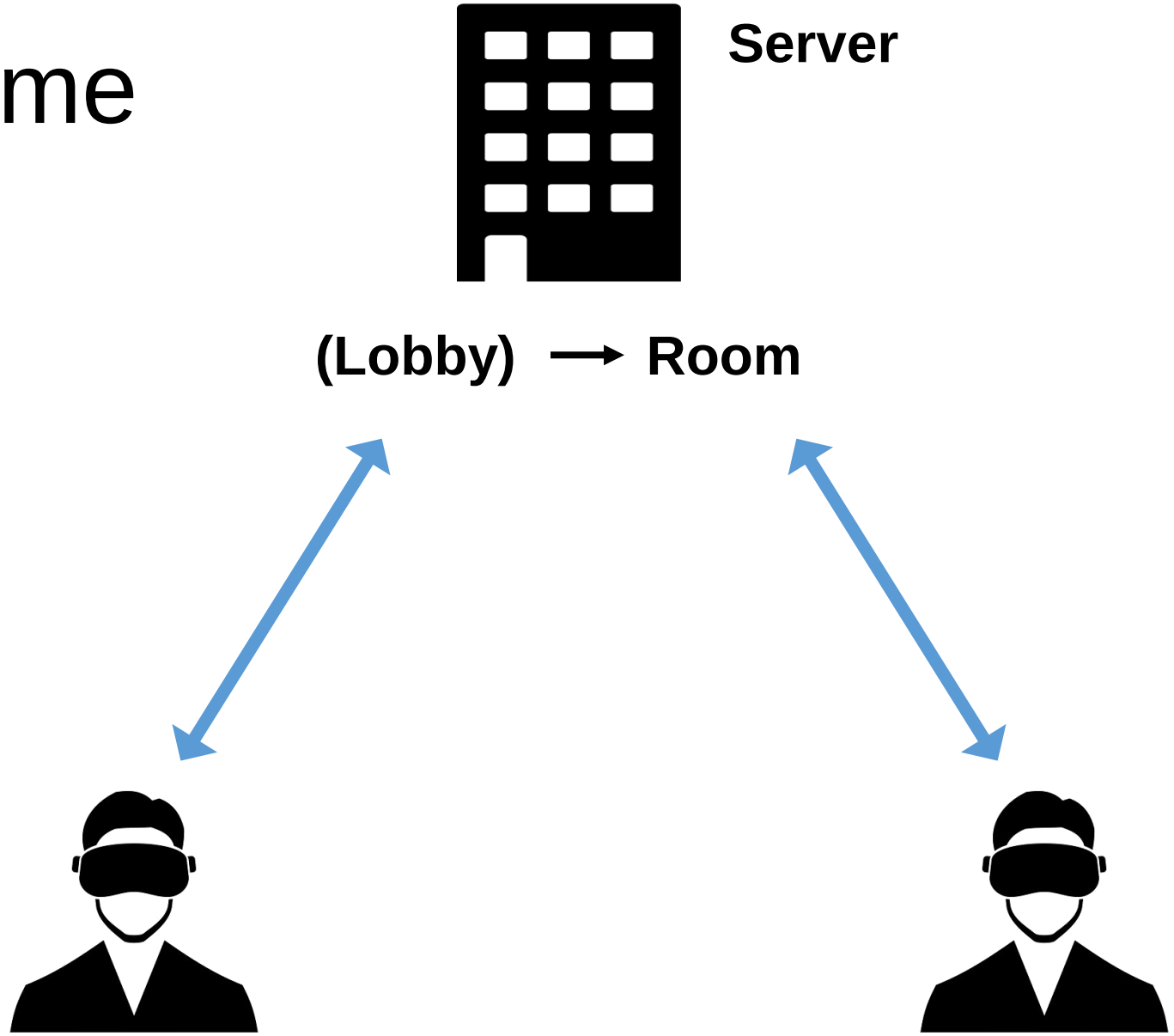

Server
Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com
A
C
D
B


Server
Back
Kurious
Active Now
MON 9:41 AM
Hey, I'm looking to upgrade my phone and was wondering if I was eligible?
Sure we can check, may I have your email so I can look up your account?
jane@messenger.com
A
C
D
B

Back to Photon


Server
Player's Transform,
Game State … etc.
Player's Transform,
Game State … etc.

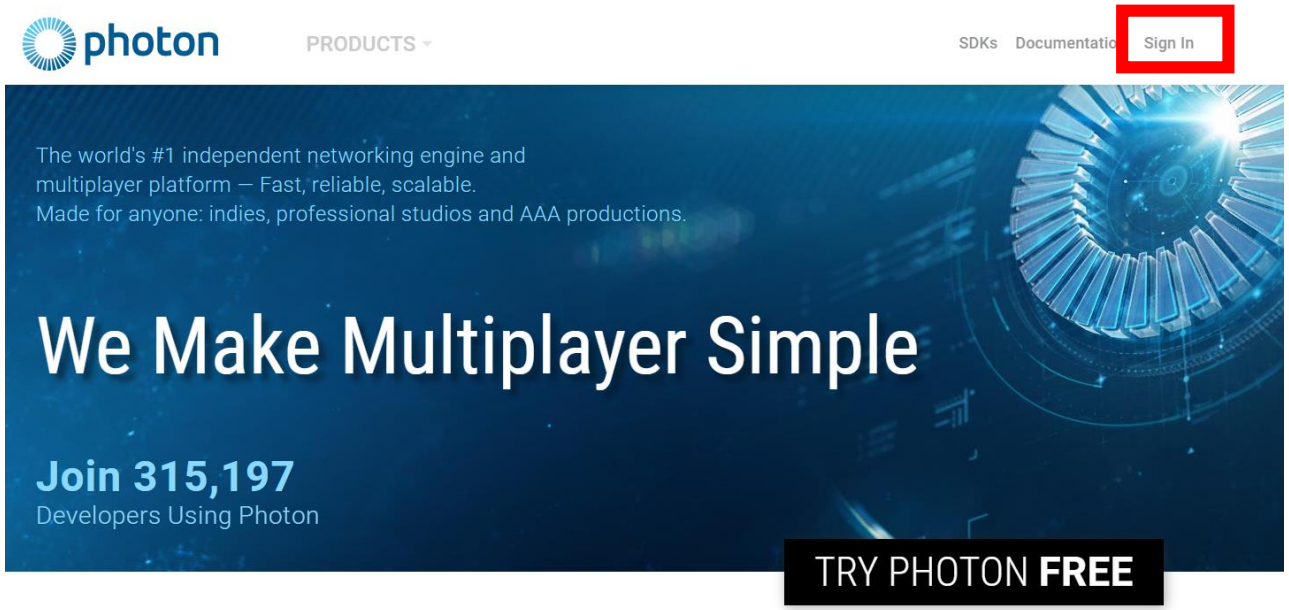
Join a Game

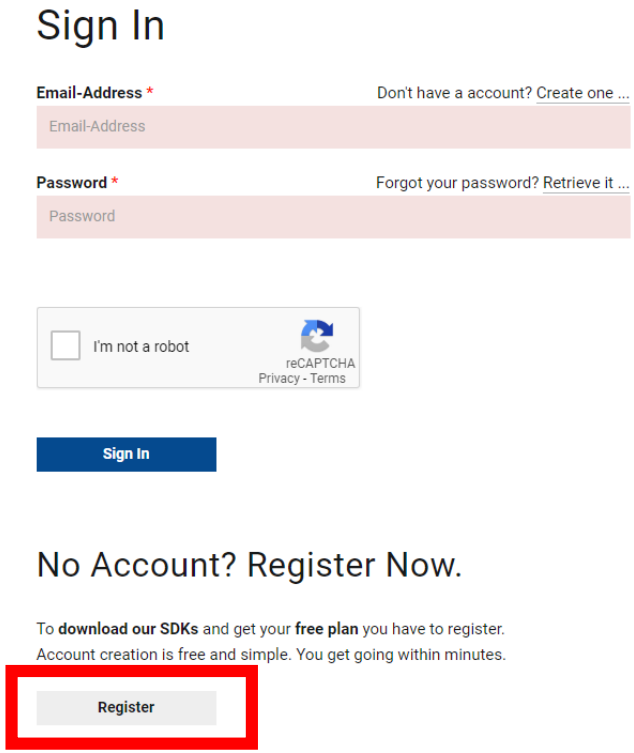


PUN Tutorial

Photon Tutorial

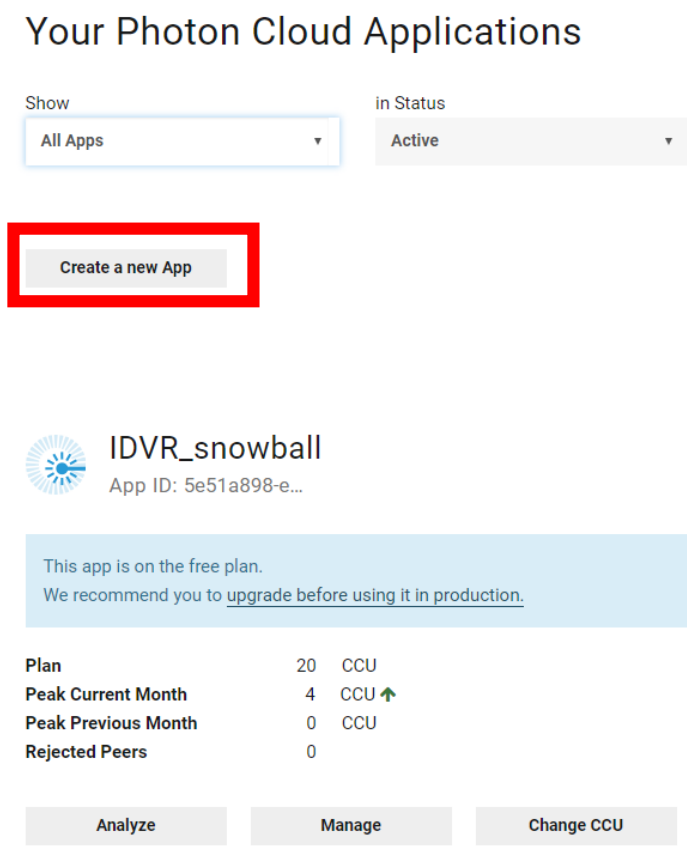
- Register a photon account
  - Photon Engine: https://www.photonengine.com/en-US/

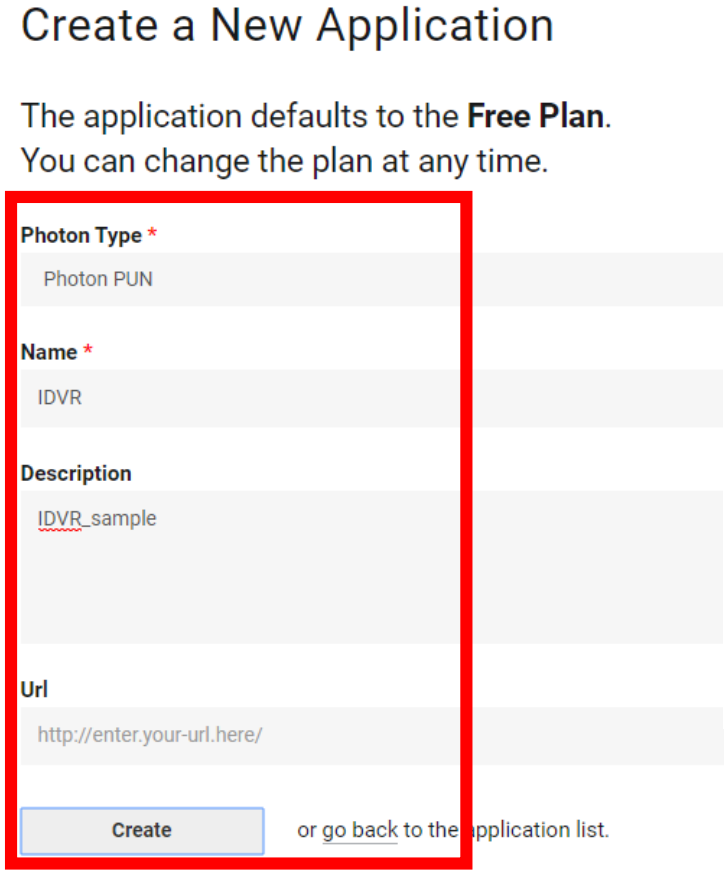


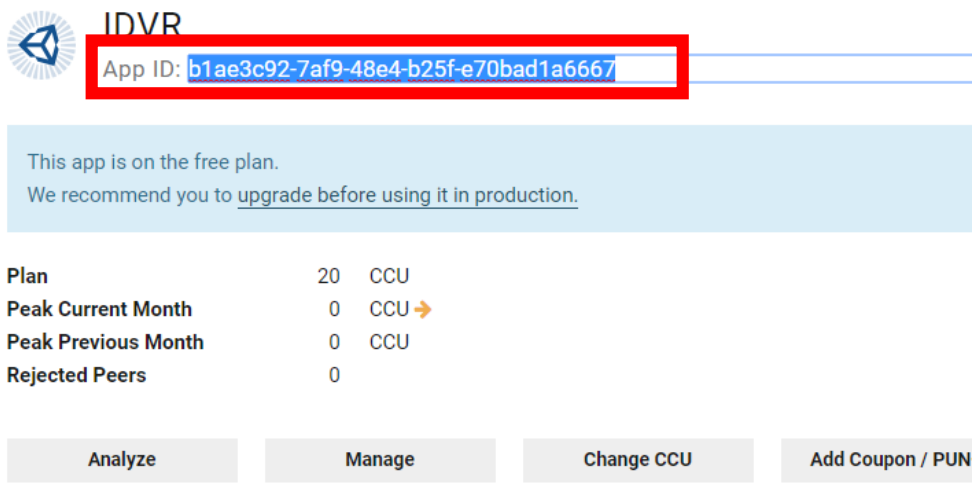


Photon Tutorial

- Create a app after you register an account (mark down the App ID)

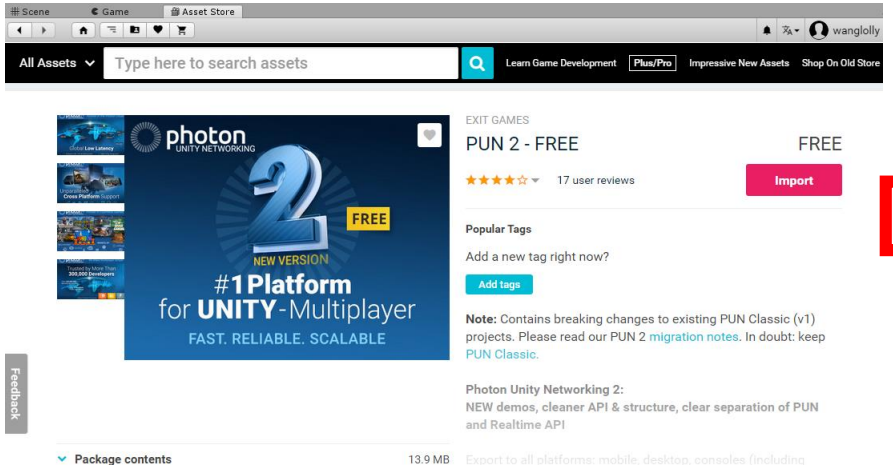


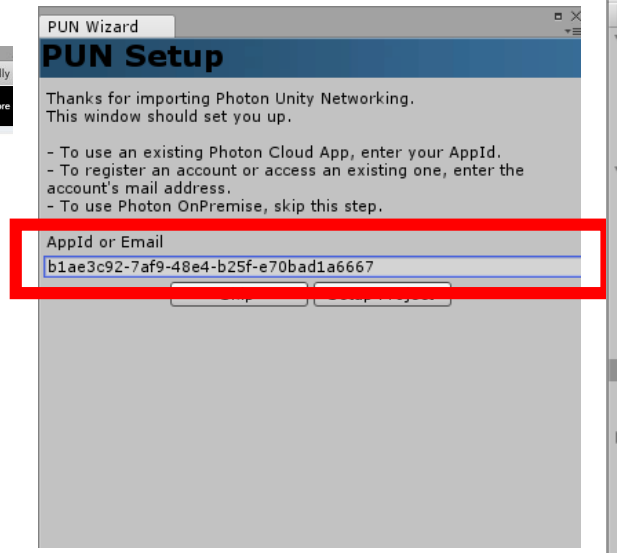


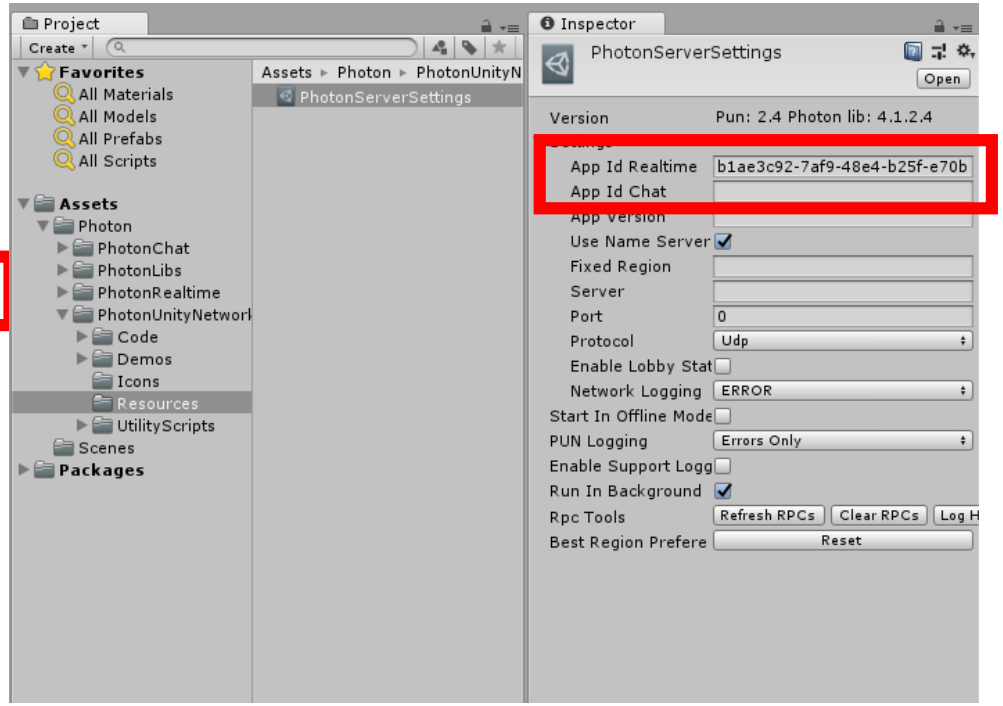


Photon Tutorial

- Go to Unity Assets Store. Download and import PUN 2.0 (free)
- Type in your App ID in the PUN Wizard panel or the PhotonServerSettings.







Photon Tutorial

- Establish a connection

Connect to Server → (Join a Lobby) → Join a Room

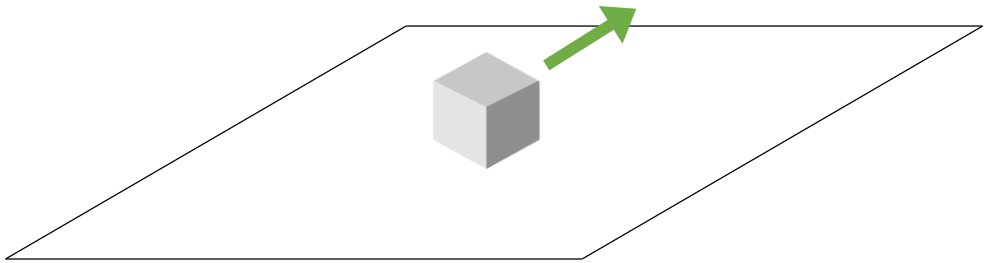
PhotonNetwork.ConnectUsingSettings();

PhotonNetwork.JoinLobby();

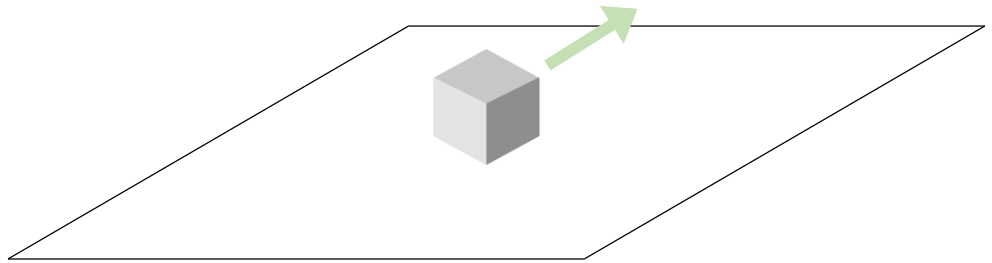
PhotonNetwork.JoinRoom();
PhotonNetwork.JoinRandomRoom();
PhotonNetwork.CreateRoom();

Photon Tutorial

- GameObject Synchronization



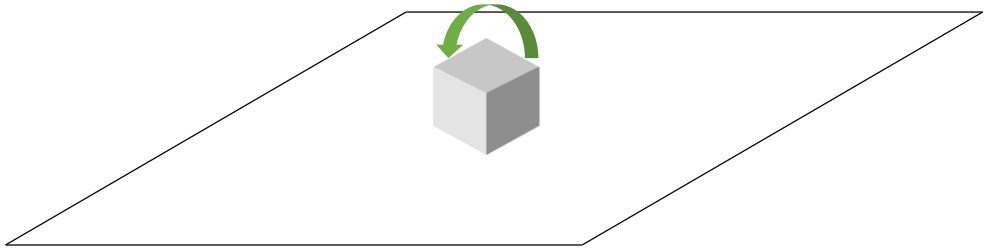
VR Player A Scene



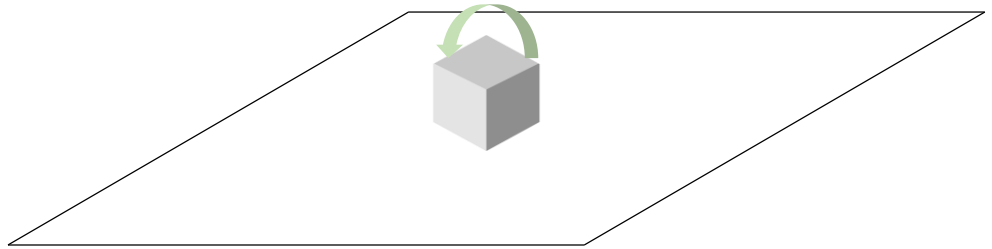
VR Player B Scene

Photon Tutorial

- GameObject Synchronization



VR Player A Scene



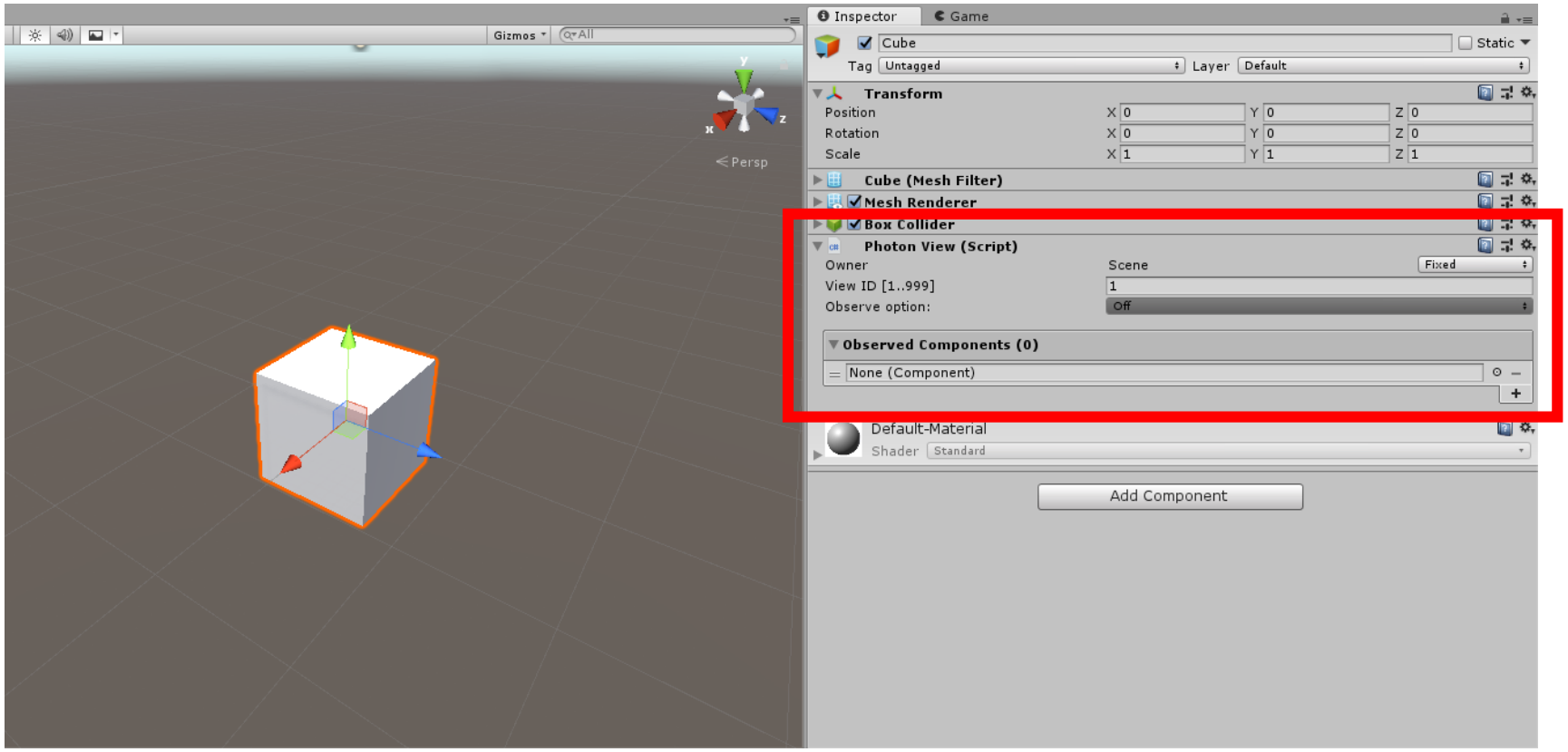
VR Player B Scene

Photon Tutorial

- GameObject Synchronization

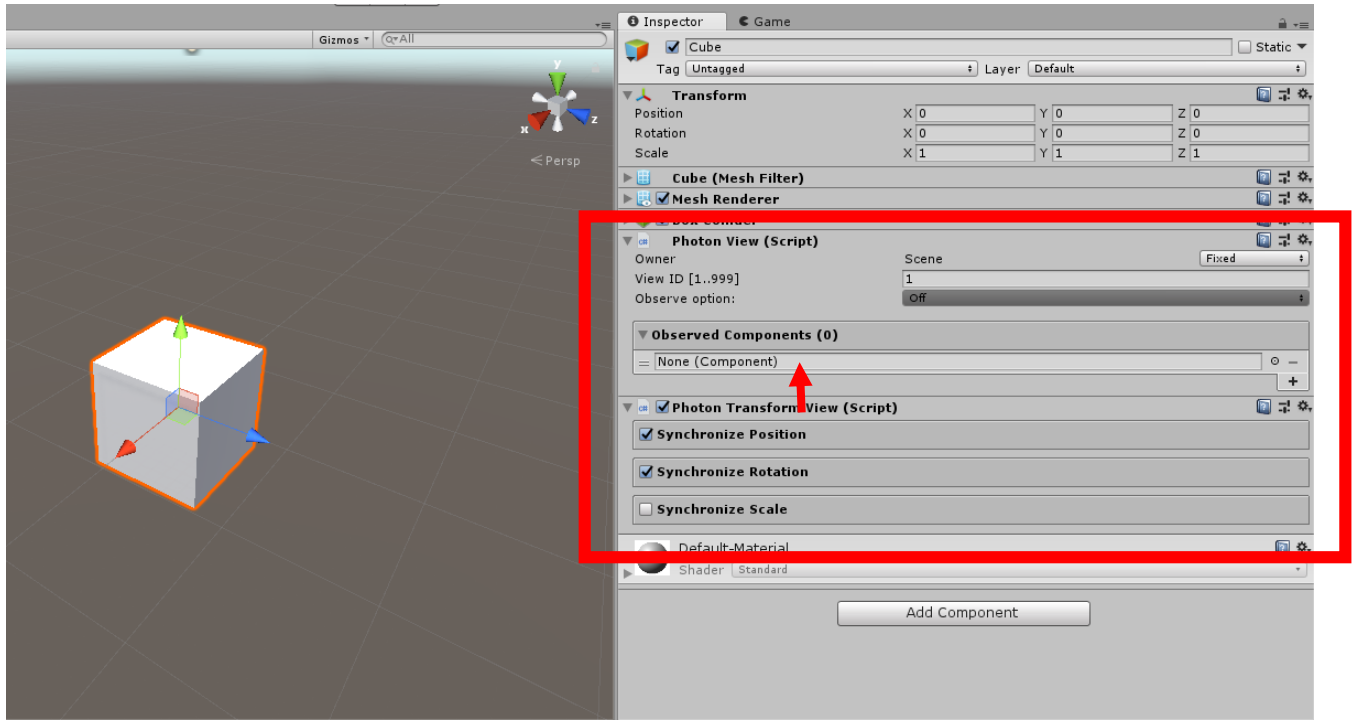
Photon Tutorial

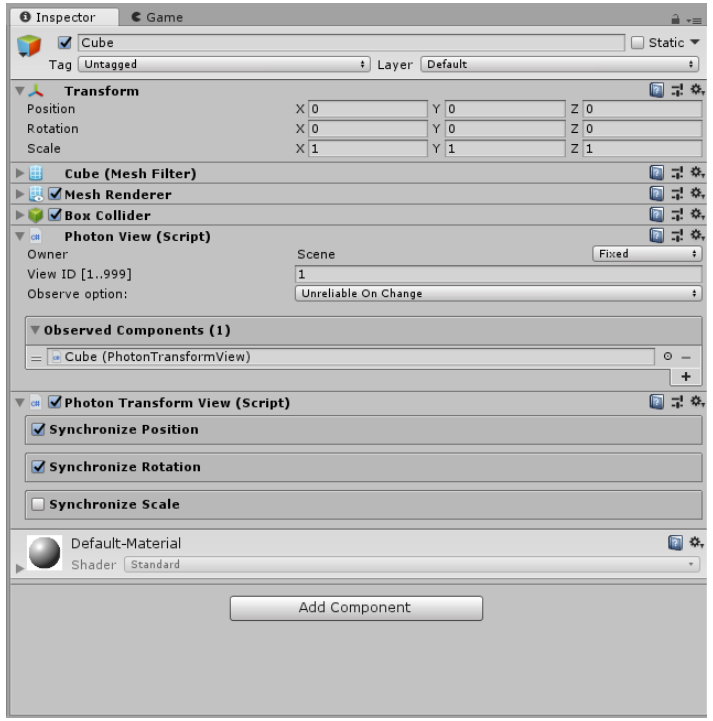
- GameObject’s Transform Synchronization
  - Attach “Photon View” on the gameobject for synchronization



Photon Tutorial

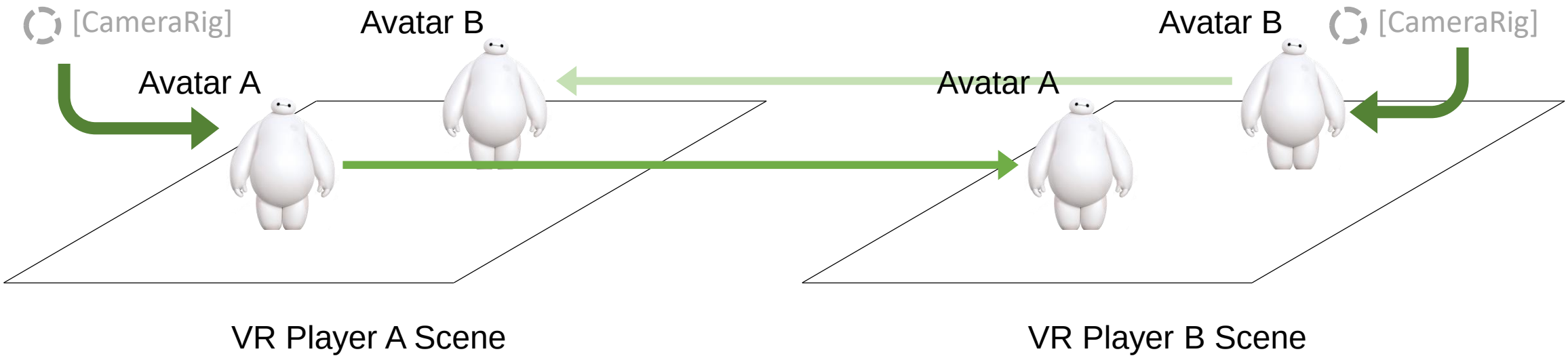
- GameObject’s Transform Synchronization
  - Attach “Photon Transform View” on the gameobject for transform’s synchronization
  - Drag the “Photon Transform View” to “Observed Component” in “Photon View”.





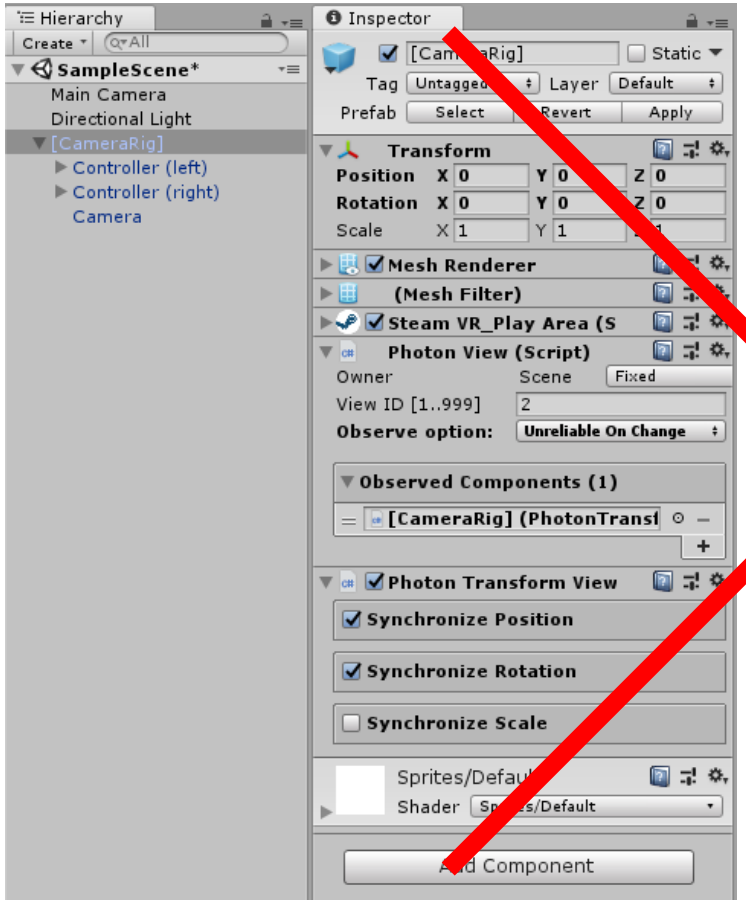
Photon Tutorial

- GameObject Synchronization

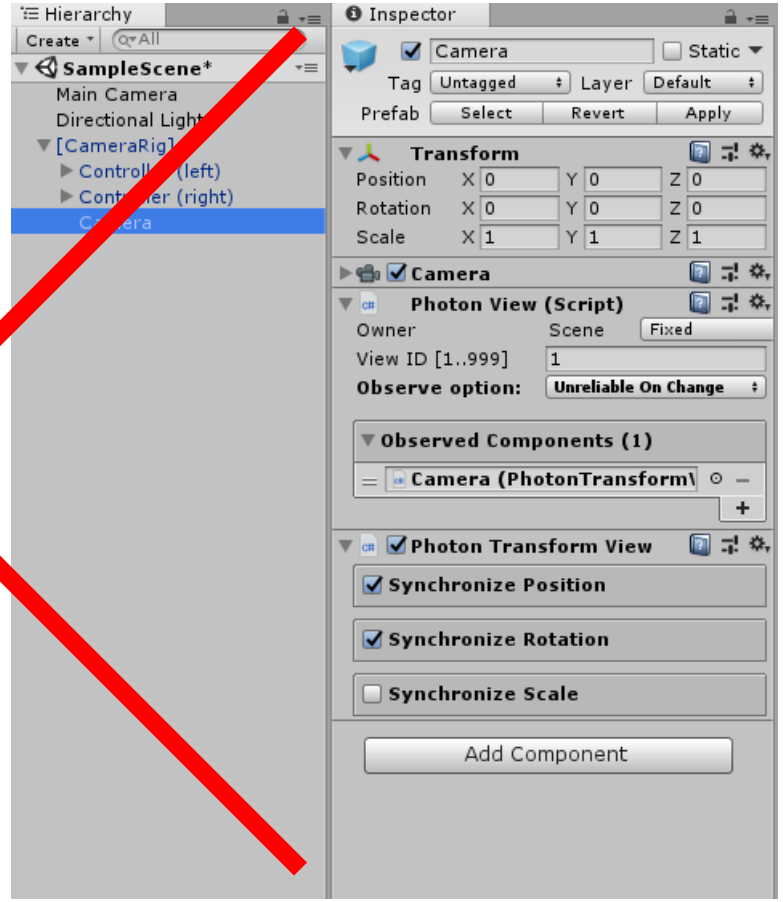


Photon Tutorial

- GameObject Synchronization

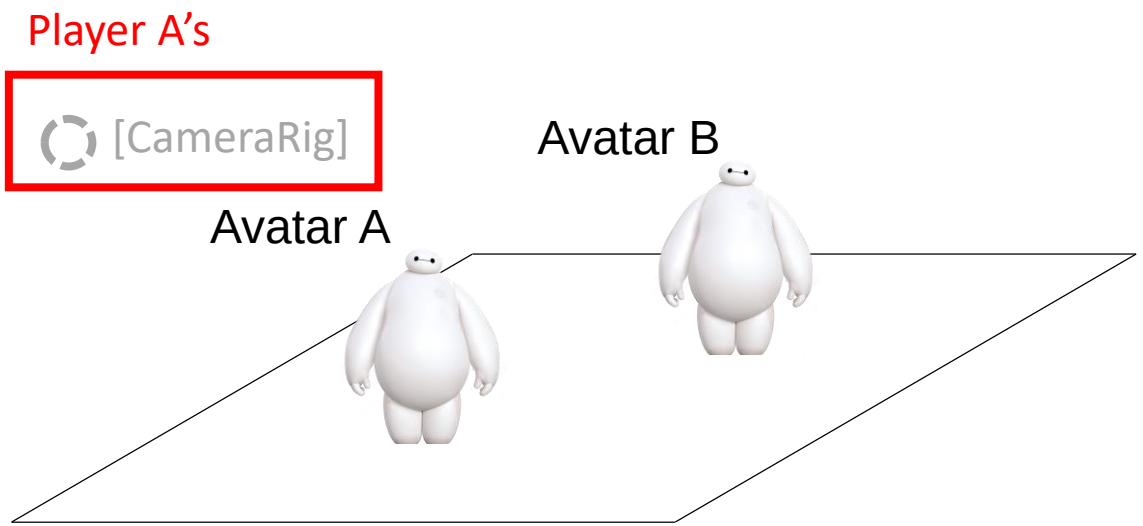


OR

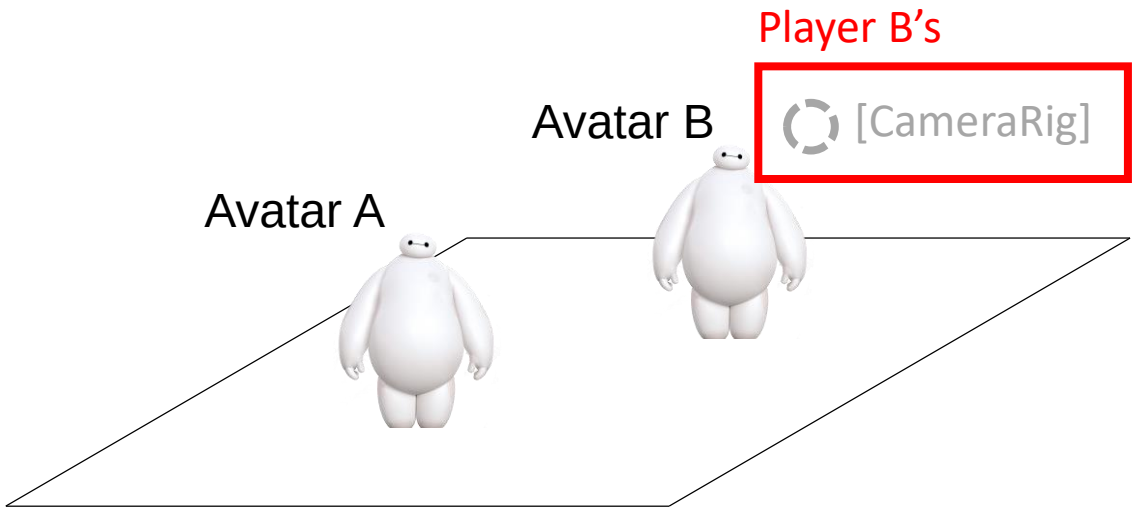


Photon Tutorial

- GameObject Synchronization



VR Player A Scene

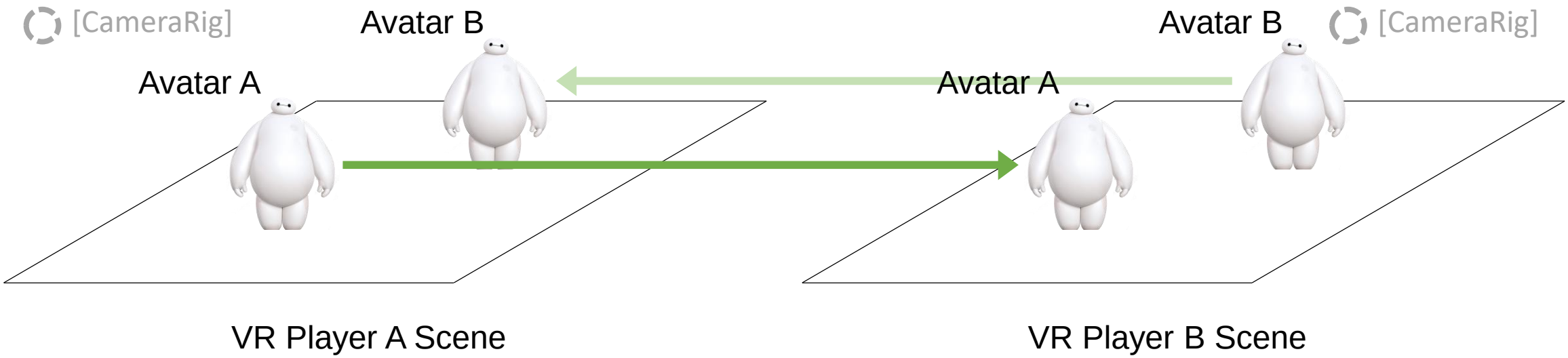


VR Player B Scene

We have only one [CameraRig] in each player's scene....

Photon Tutorial

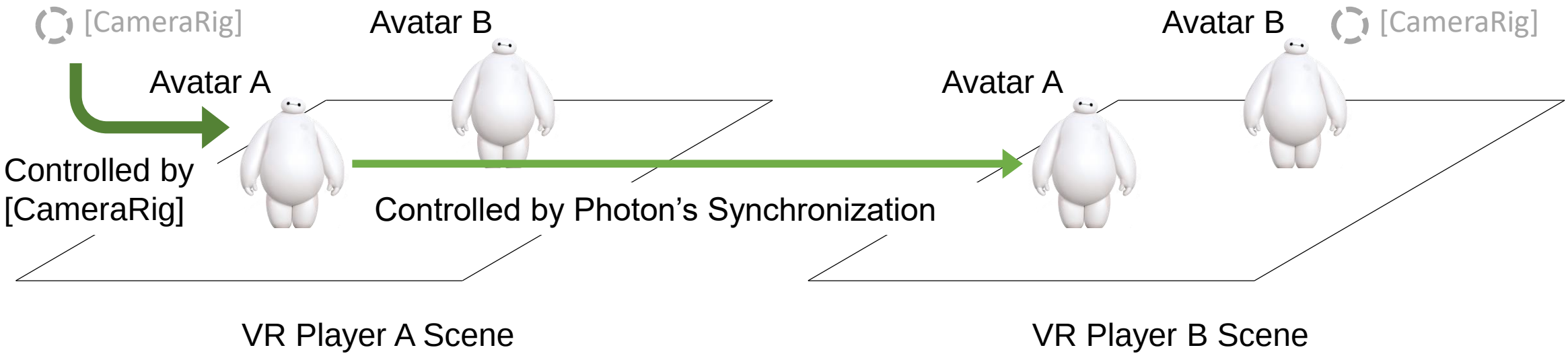
- GameObject Synchronization



We synchronize the player’s avatar, which is also gameobject in the scene (your model).

Photon Tutorial

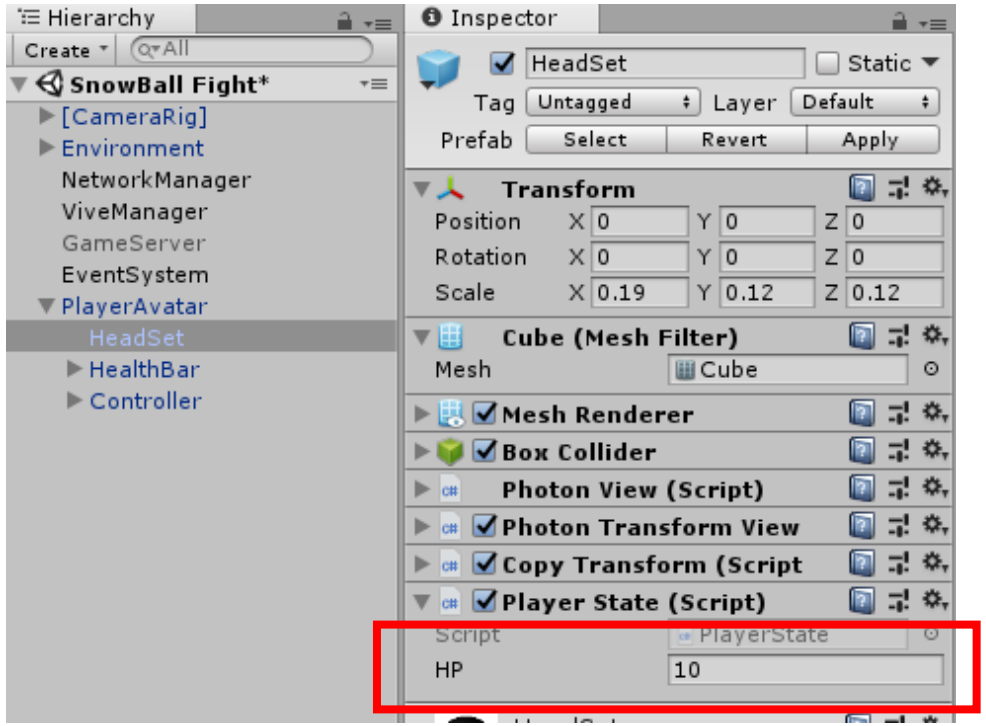
- GameObject Synchronization



We synchronize the player's avatar, which is also gameobject in the scene (your model).

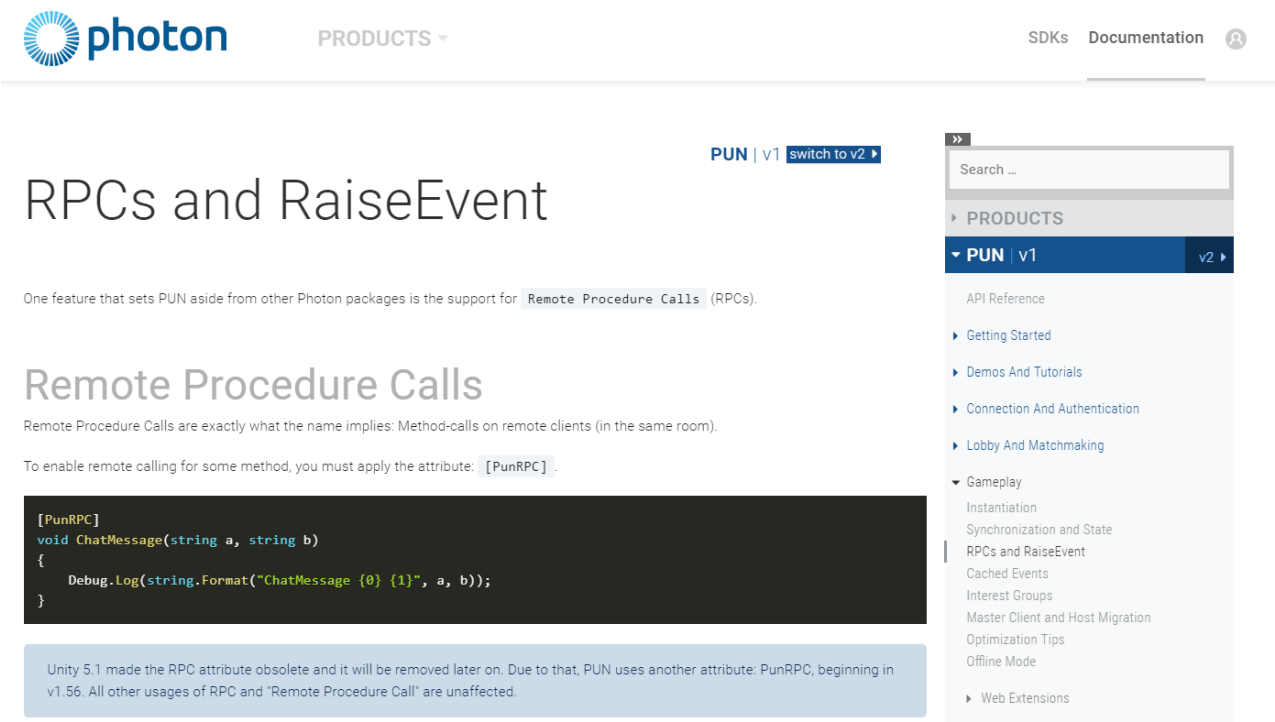
Photon Tutorial

- Game State Synchronization
  - Game's state: Player's HP, Player's State, Weapon's value…etc.
  - These "values" are often a variable in your script, which is not a "gameObject's transform".



Photon Tutorial

- Game State Synchronization
  - RPC (Remote Procedure Call): RPC is a protocol that allows a program to call a subroutine in another computer without programmer explicitly coding the details for the remote interaction
  - Photon supports RPC as well.



Game

Game Structure

- Game Logic Server (Master) & Game Client
  - Game Logic Server: Handle the Game Logic. In Photon Cloud, we call it “Master”.

    (Game Server ≠ Network Server)

  - Game Client: Handle the player’s character state

Game Structure

- Game Logic Server (Master) & Game Client

Game Logic Server (Master)

- Generate Health Pack
- Game Logic (GameManager)

Game Client

- Update the Player's Avatar
- Generate a Snowball
- Maintain Player's state (HP)

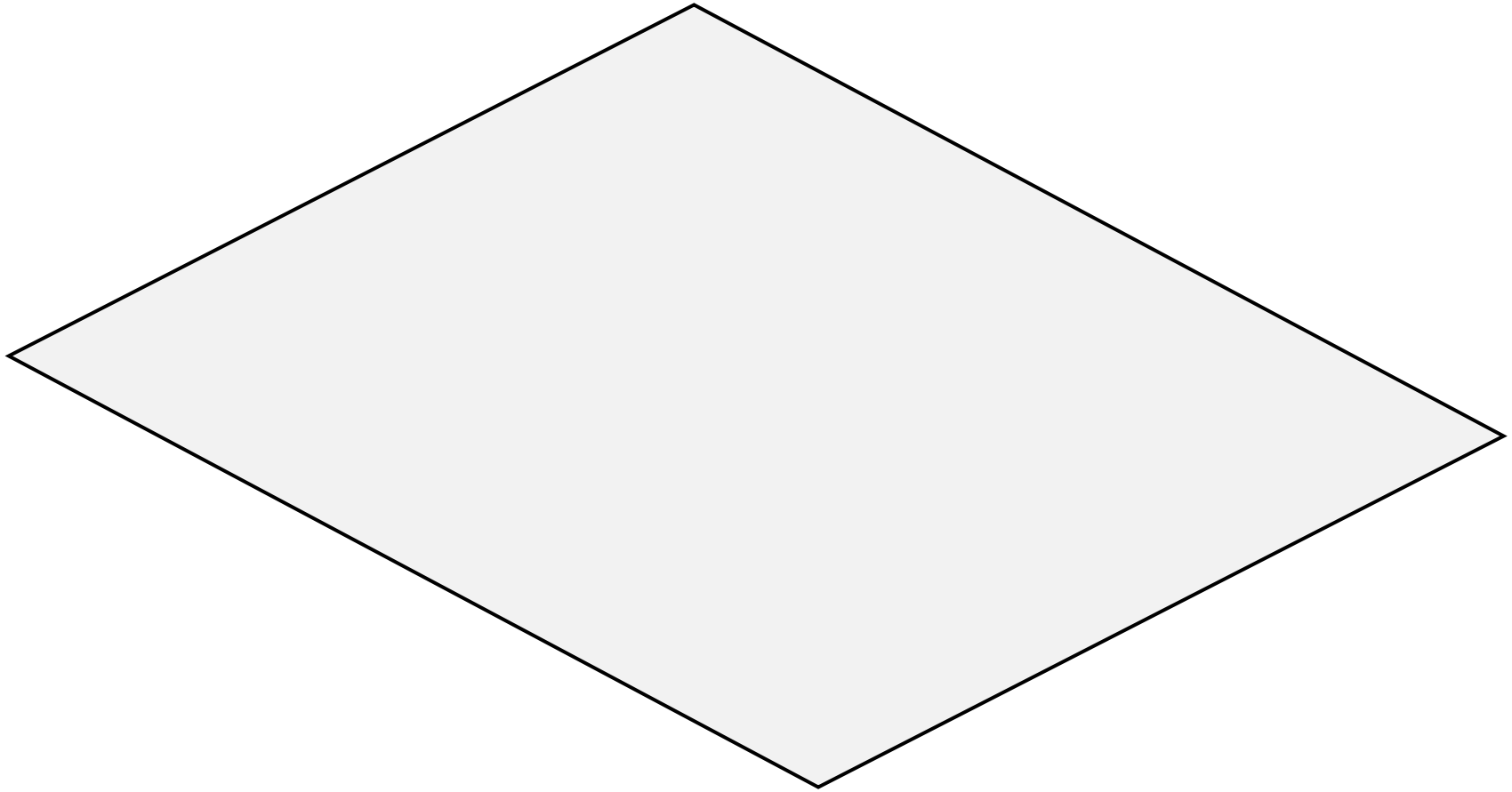
Suggestion: Make a similar chart for your project before you start it. It will be much better for you to develop your game.

Code Analysis

Snowball Fight

Environment

Transform
Renderer



Snowball Fight

NetworkManager

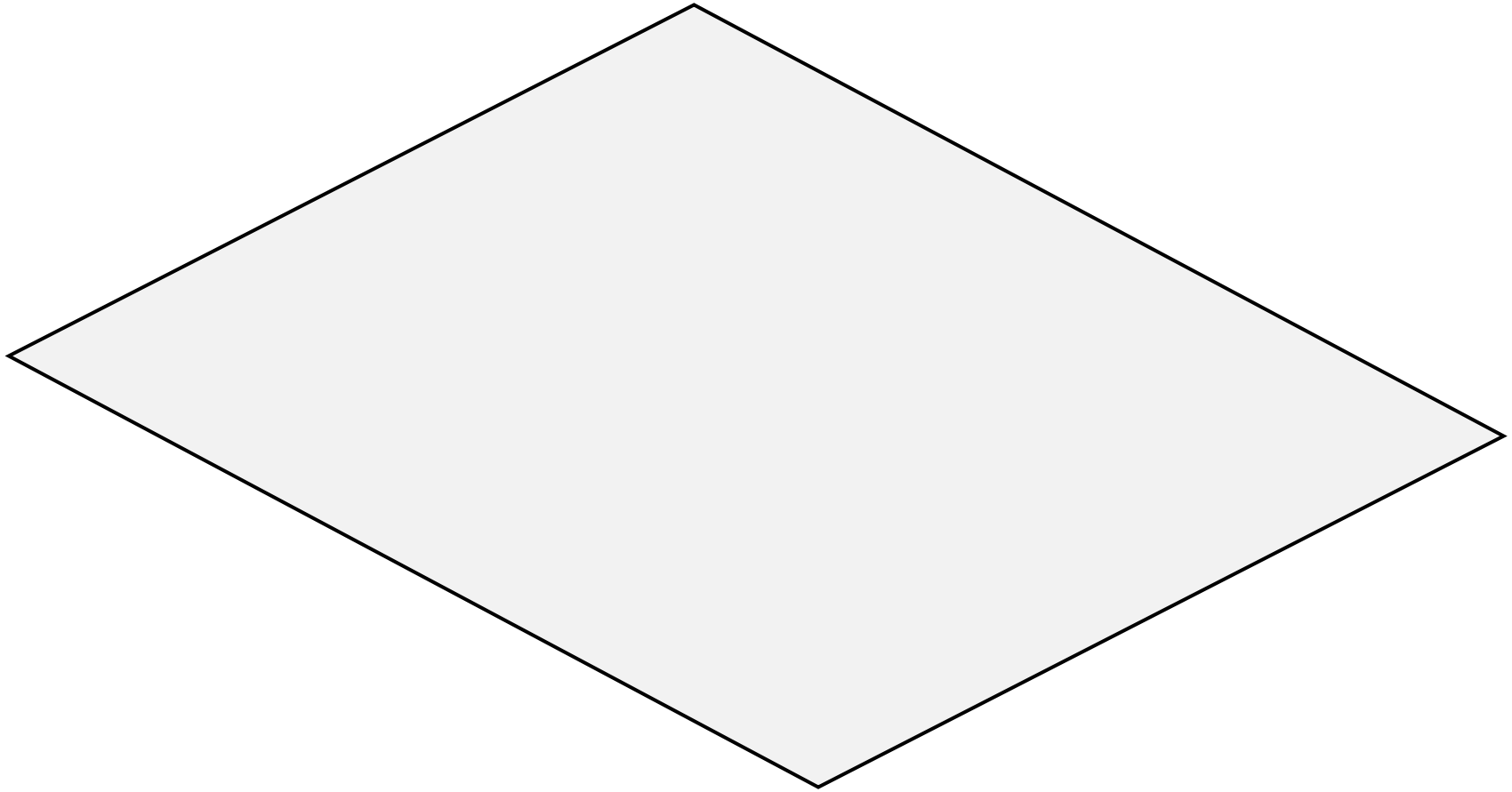
Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

Transform
Renderer



Snowball Fight

NetworkManager

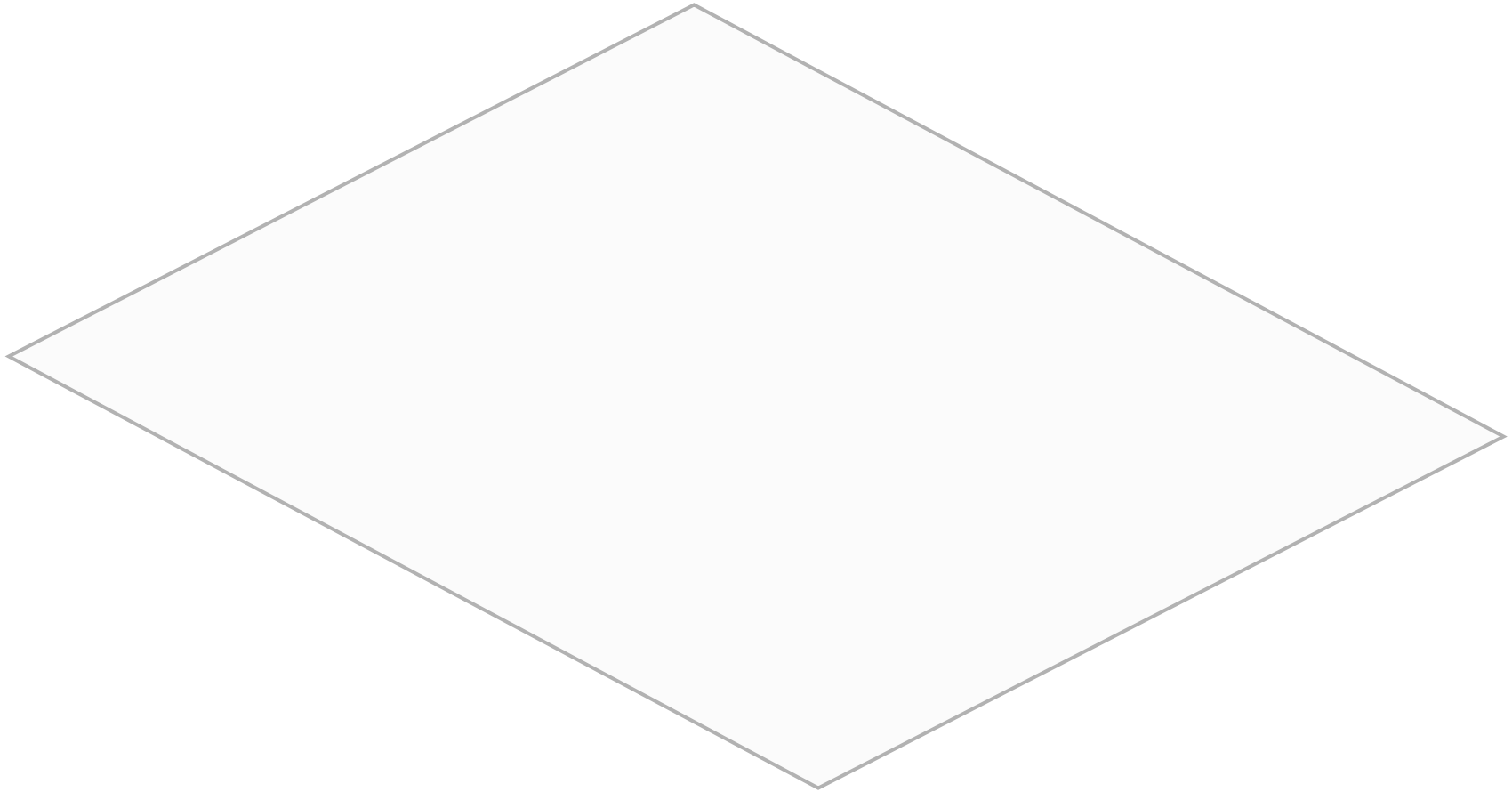
Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

Transform
Renderer



NetworkManager.cs

- Help game client connect to the room.
- Instantiate PlayerAvatar.

- Note: the prefabs you want to generate by PhotonNetwork.Instantiate() have to be in Prefabs > Resources directory.

Snowball Fight

NetworkManager

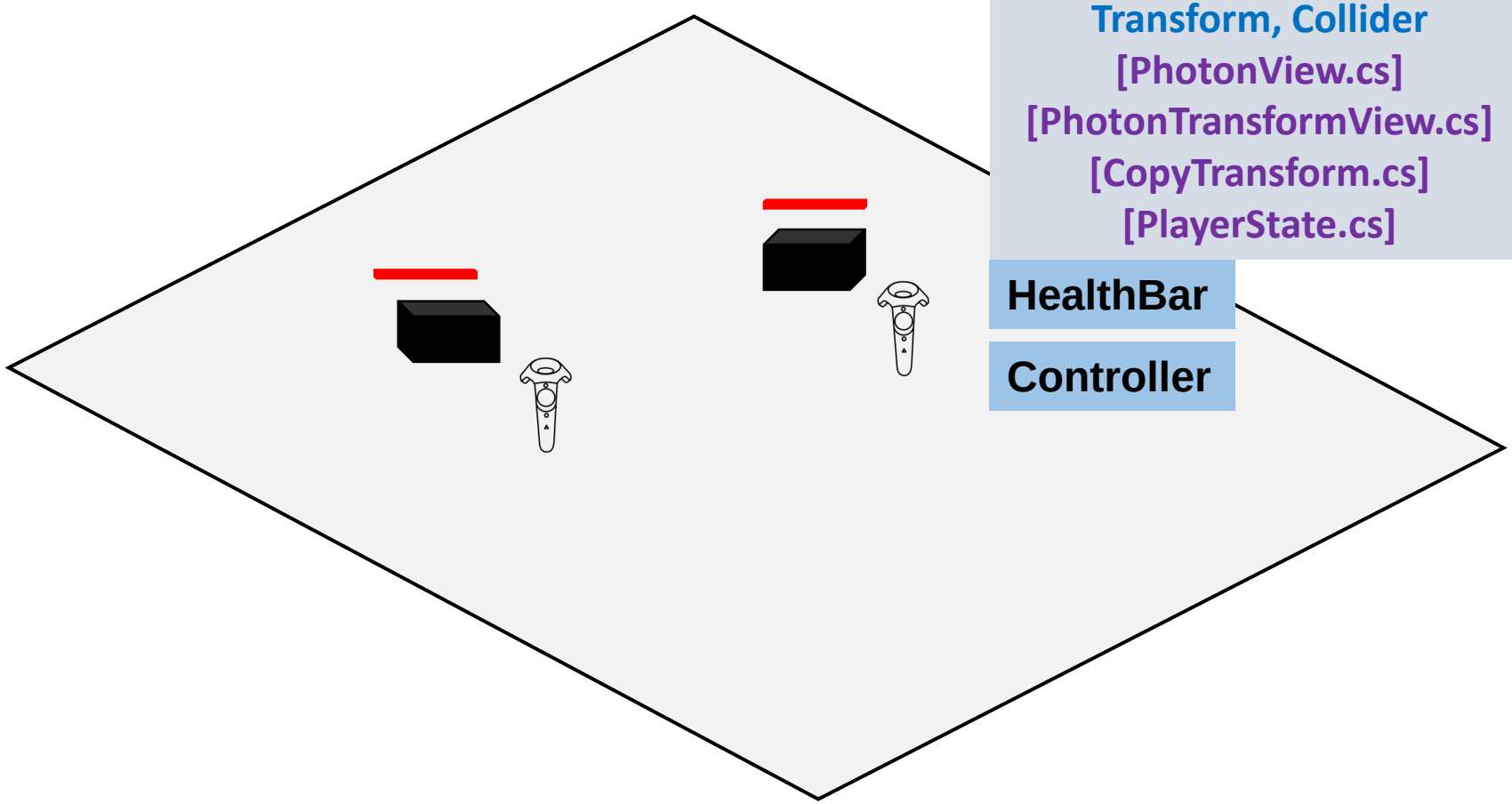
Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

Transform
Renderer



PlayerAvatar

HeadSet

Transform, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CopyTransform.cs]
[PlayerState.cs]

HealthBar

Controller

Snowball Fight

NetworkManager

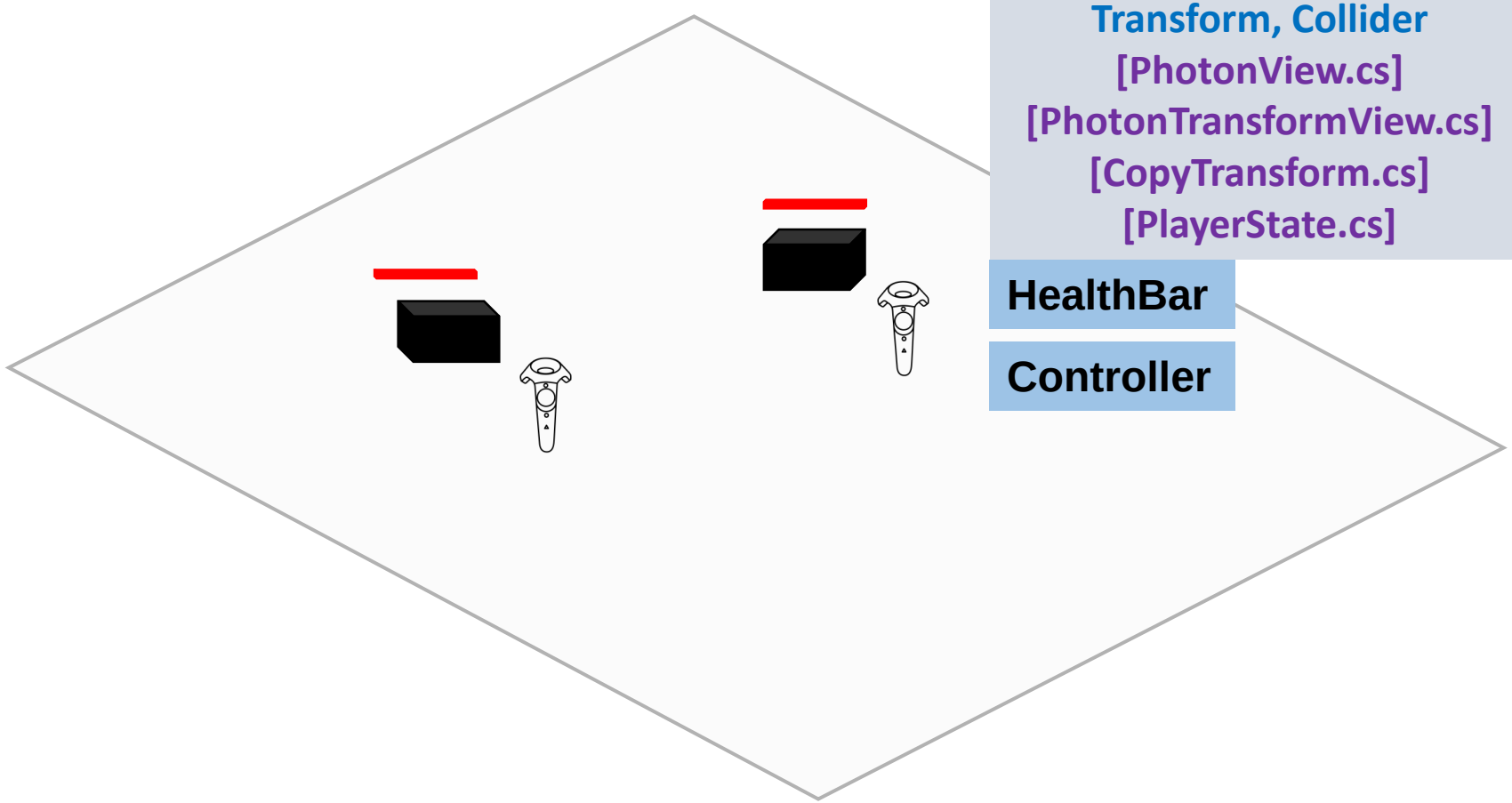
Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

Transform
Renderer



PlayerAvatar

HeadSet

Transform, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CopyTransform.cs]
[PlayerState.cs]

HealthBar

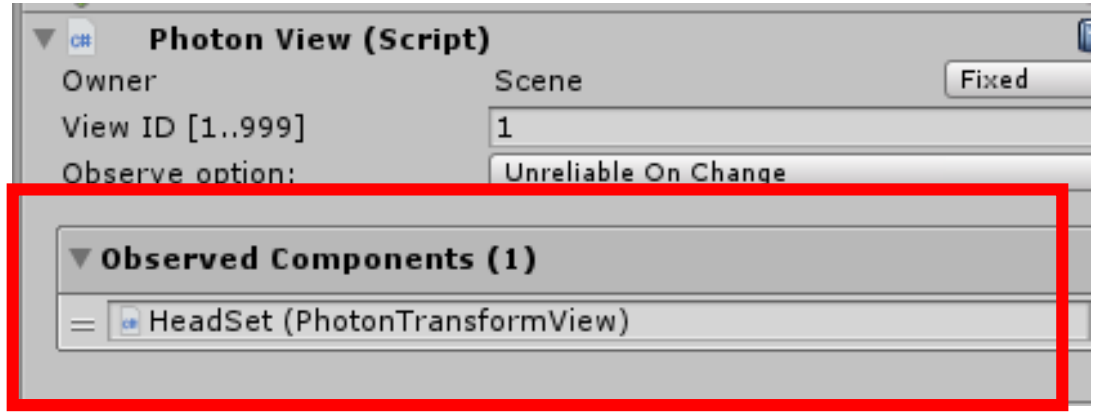
Controller

Synchronized GameObject

- The GameObject we have to update their information to all game clients.
  - Game players
  - Snowball
  - Health bar
  - Health pack (will talk about this at the end)
- PhotonView.cs and PhotonTransformView.cs

PhotonView.cs

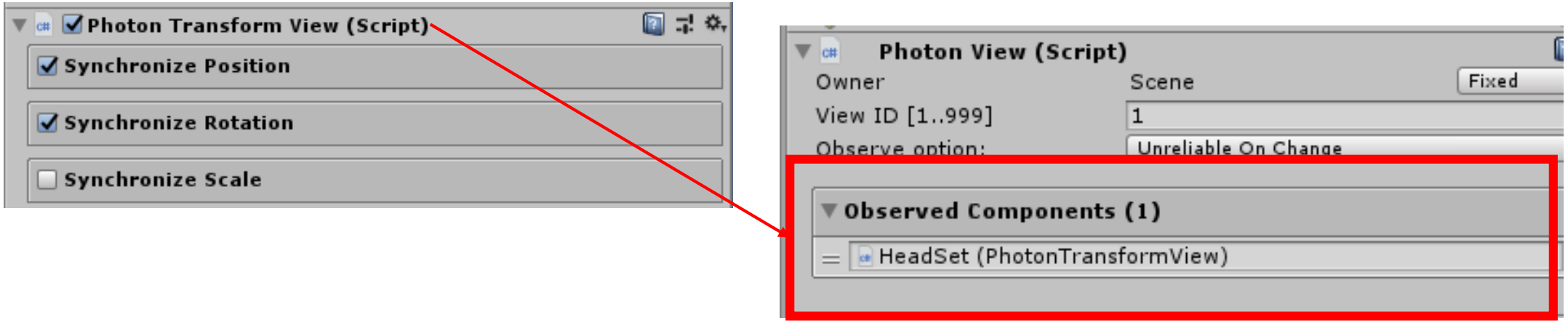
- A script provided in Photon package.
- This is used to synchronized the GameObjects or states related to the game.



Drag the components you want to synchronize over here

PhotonTransformView.cs

- A script provided in Photon package.
- This is used to synchronized the Transform of a GameObject



CopyTransform.cs

- Copy the position and the rotation from the vive CameraRig if it is generated in local. Otherwise, the transform should be controlled from the network

```
if (this.GetComponent<PhotonView>().IsMine)
{
    if (GameObjectType == type.head)
    {
        this.transform.position = ViveManager.Instance.head.transform.position;
        this.transform.rotation = ViveManager.Instance.head.transform.rotation;
    }
    if (GameObjectType == type.leftHand)
    {
        this.transform.position = ViveManager.Instance.leftHand.transform.position;
        this.transform.rotation = ViveManager.Instance.leftHand.transform.rotation;
    }
    if (GameObjectType == type.rightHand)
    {
        this.transform.position = ViveManager.Instance.rightHand.transform.position;
        this.transform.rotation = ViveManager.Instance.rightHand.transform.rotation;
    }
}
```

PlayerState.cs

- Update()
  - Update HealthBar
  - check if the player is dead
- OnTriggerEnter()
  - Check if hit by a snowball, update HP, destroy snowball

PlayerState.cs

- synchronizeHP()
- After updating HP on local, the value has to be synchronized in other game clients.

```
[PunRPC]
void synchronizeHP(int value)
{
    GetComponent<PhotonView>().RPC("setHP", RpcTarget.Others, value);
}

[PunRPC]
void setHP(int v)
{
    HP = v;
}
```

PlayerState.cs

- synchronizeHP()
- After updating HP on local, the value has to be synchronized in other game clients.

```
[PunRPC]
void synchronizeHP(int value)
{
    GetComponent<PhotonView>().RPC("setHP", RpcTarget.Others, value);
}

[PunRPC]
void setHP(int v)
{
    HP = v;
}
```

Other game clients

Snowball Fight

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

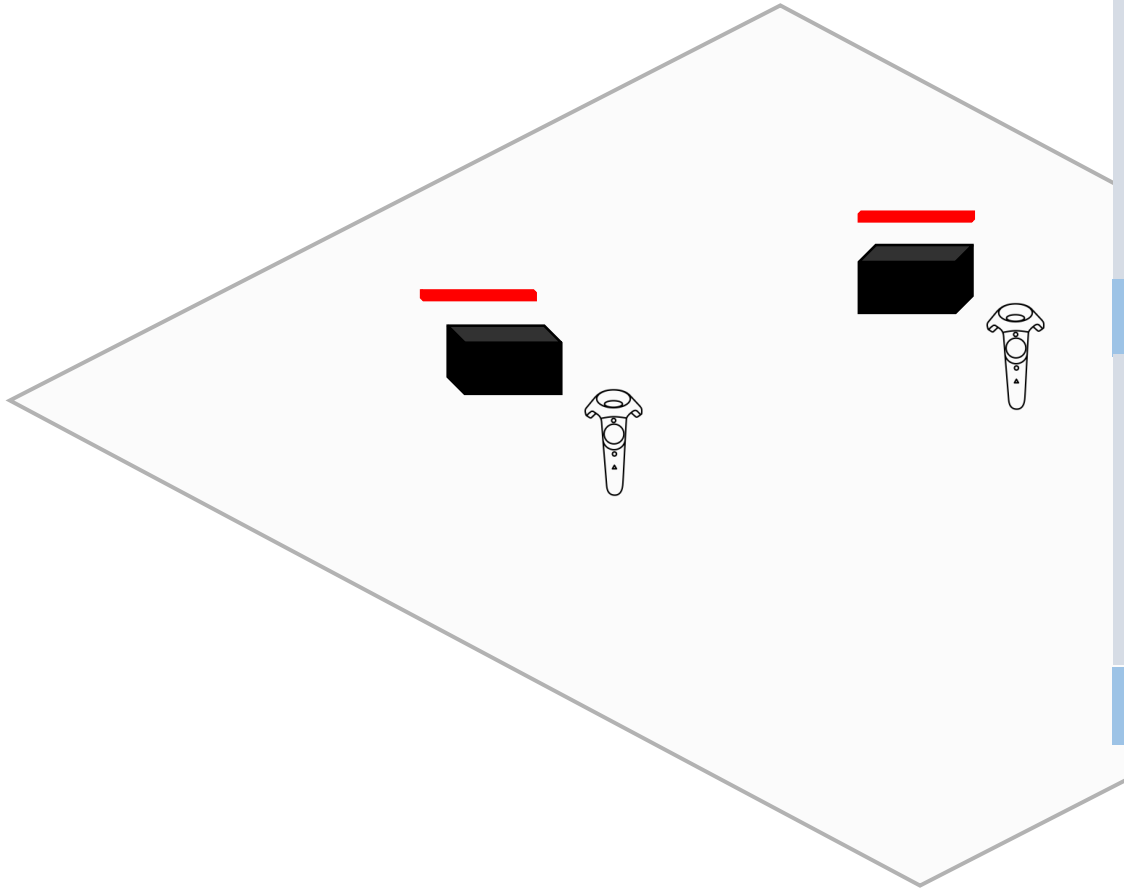
Transform
[ViveManager.cs]

Master

Transform
[GameManager.cs]
[HealthPackControl.cs]

Environment

Transform
Renderer



PlayerAvatar

HeadSet

Transform
[PhotonView.cs]
[PhotonTransformView.cs]
[CopyTransform.cs]
[PlayerState.cs]

HealthBar

Transform
[HealthBarControl.cs]
[PhotonView.cs]
[PhotonTransformView.cs]
[HealthBarTransform.cs]

Controller

HealthBar

- HealthBarControl.cs
  - Update the health bar
- HealthBarTransform.cs
  - Set the position and rotation of health bar

Snowball Fight

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

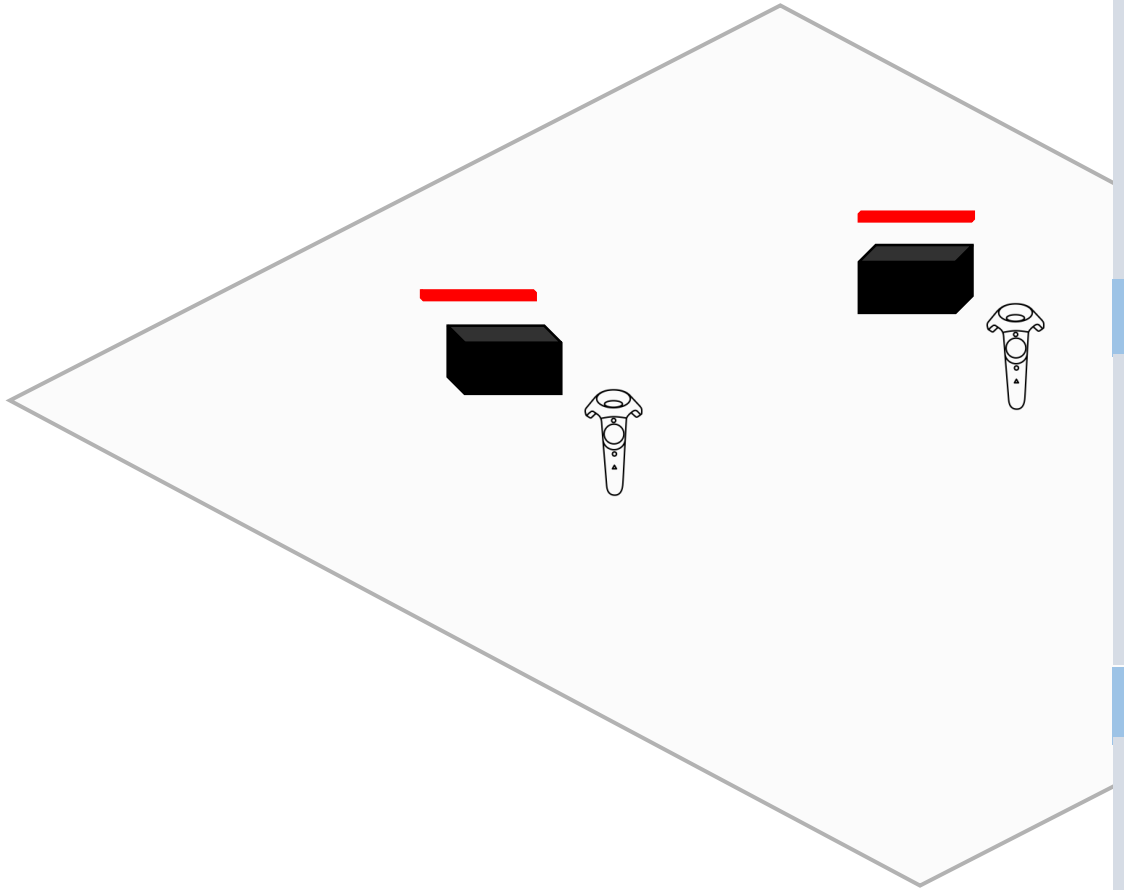
Transform
[ViveManager.cs]

Master

Transform
[GameManager.cs]
[HealthPackControl.cs]

Environment

Transform
Renderer



PlayerAvatar

HeadSet

Transform
[PhotonView.cs]
[PhotonTransformView.cs]
[CopyTransform.cs]
[PlayerState.cs]

HealthBar

Transform
[HealthBarControl.cs]
[PhotonView.cs]
[PhotonTransformView.cs]
[HealthBarTransform.cs]

Controller

Transform
[PhotonView.cs]
[PhotonTransformView.cs]
[CopyTransform.cs]
[PlayerController.cs]

PlayerController.cs

- The controller script are much similar to the HandController.cs in Dinosaur Game.
- Instantiate the snowball prefab with PhotonNetwork.Instantiate().

```
//Generate the Snowball if the user presses his/hers own controller trigger button
if (this.GetComponent<PhotonView>().IsMine)
{
    if (!isPressing)
    {
        if (handType == HandType.LeftHand)
        {
            if (ViveManager.Instance.leftHand.GetComponent<SteamVR_TrackedController>().triggerPressed)
            {
                //Generate a snowball
                snowBall = PhotonNetwork.Instantiate(snowBallPrefab.name, this.transform.position, this.transform.rotation);
                //Makes the snow follow the controller
                snowBall.transform.parent = this.transform;
                isPressing = true;
            }
        }
```

Snowball Fight

Environment

Transform
Renderer

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

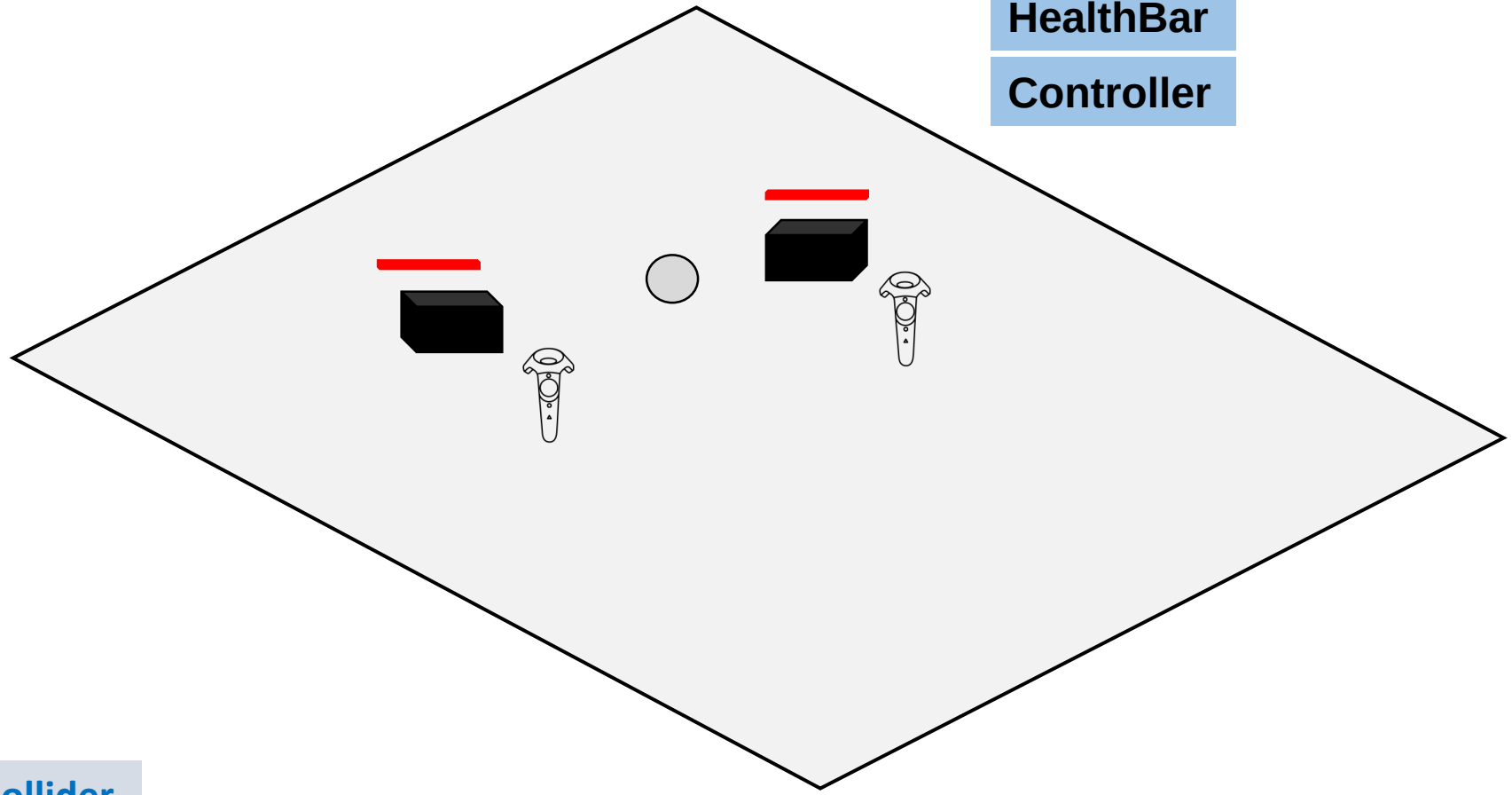
Transform
[ViveManager.cs]

PlayerAvatar

HeadSet

HealthBar

Controller



Snowball

Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappera.cs]

Snowball Fight

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

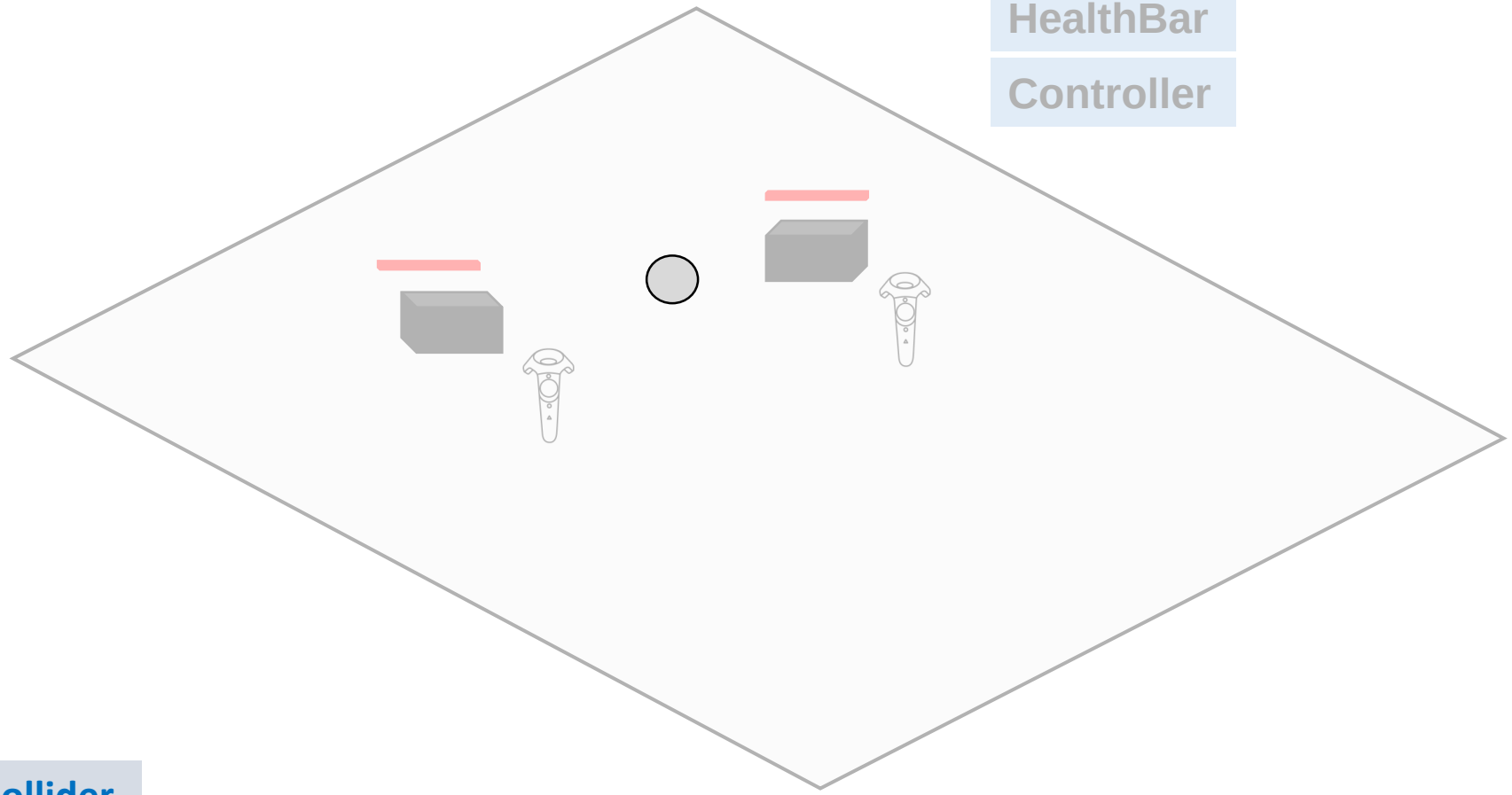
Transform
Renderer

PlayerAvatar

HeadSet

HealthBar

Controller



Snowball

Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappear.cs]

CountDownDisappear.cs

- Destroy the snowball if it exist in the scene more than 10 secs.

```
public float countDownTimer = 10.0f;
public bool startCountDown = false;
private float timer = 0;

// Update is called once per frame
void Update () {

    if (startCountDown)
    {
        timer += Time.deltaTime;
        if (timer > countDownTimer)
            PhotonNetwork.Destroy(this.gameObject);
    }
}
```

Snowball Fight

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

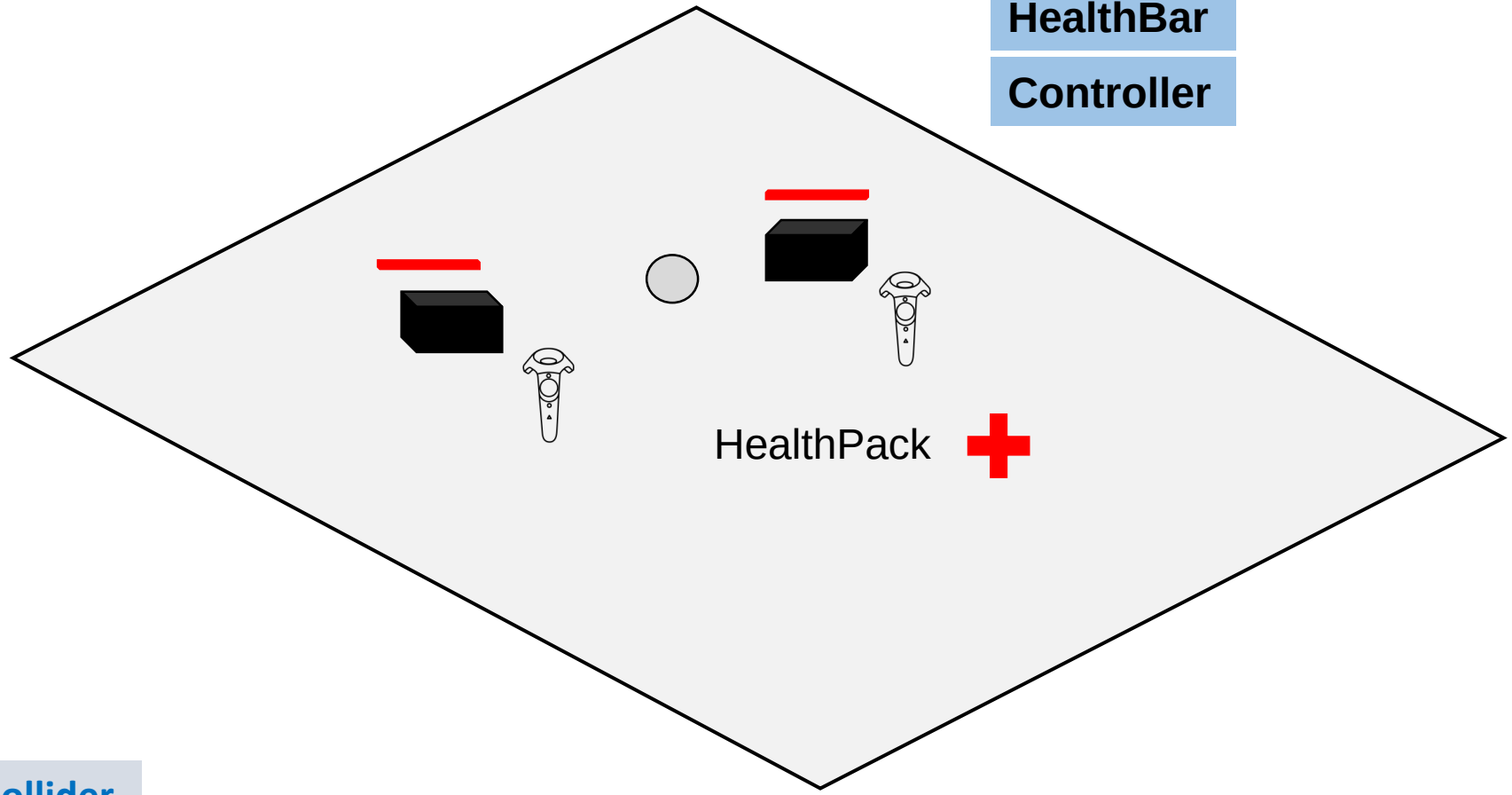
Transform
Renderer

PlayerAvatar

HeadSet

HealthBar

Controller



Snowball

Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappear.cs]

Snowball Fight

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Environment

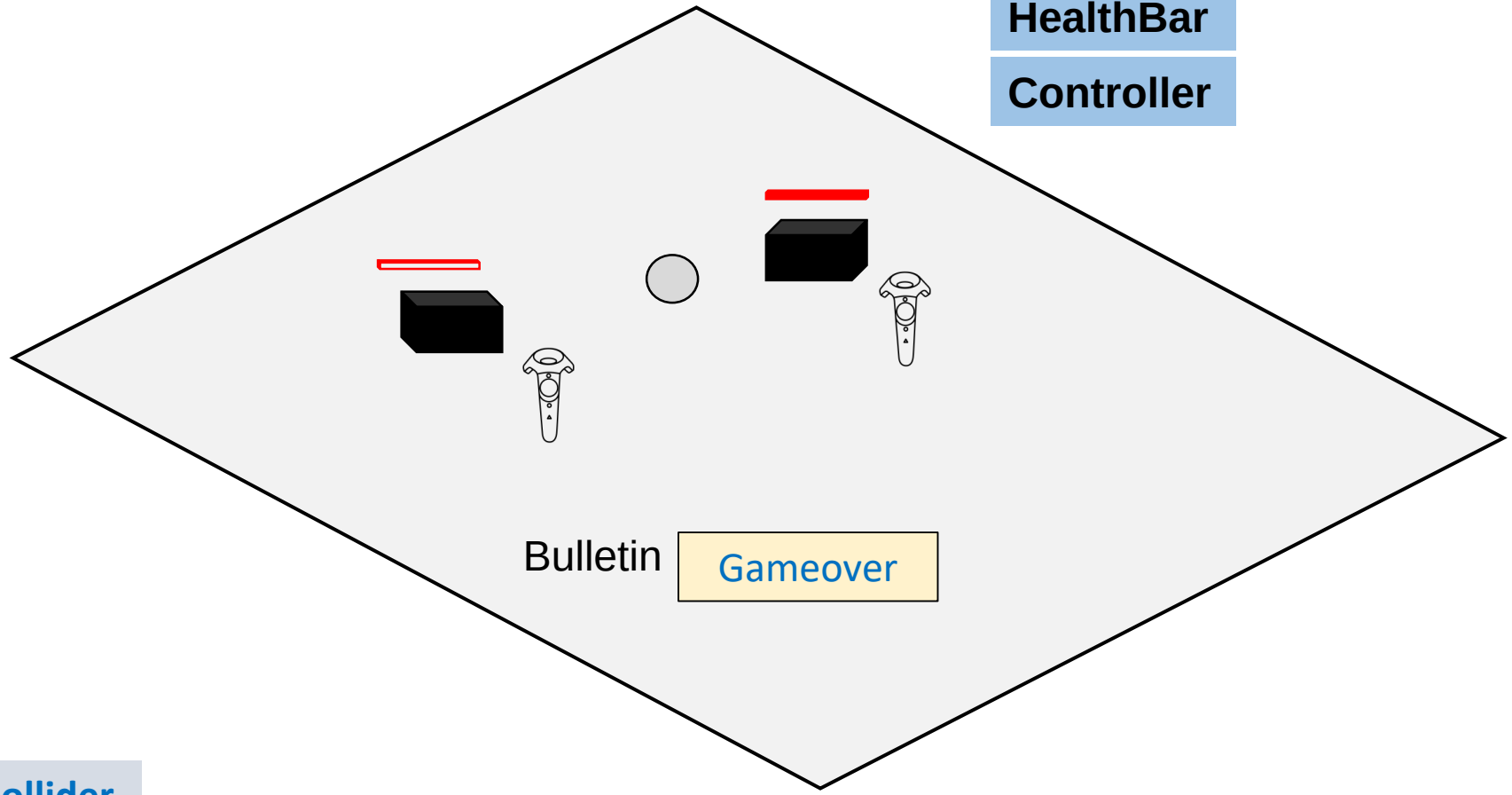
Transform
Renderer

PlayerAvatar

HeadSet

HealthBar

Controller



Snowball

Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappear.cs]

Snowball Fight

Environment

Transform
Renderer

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Master

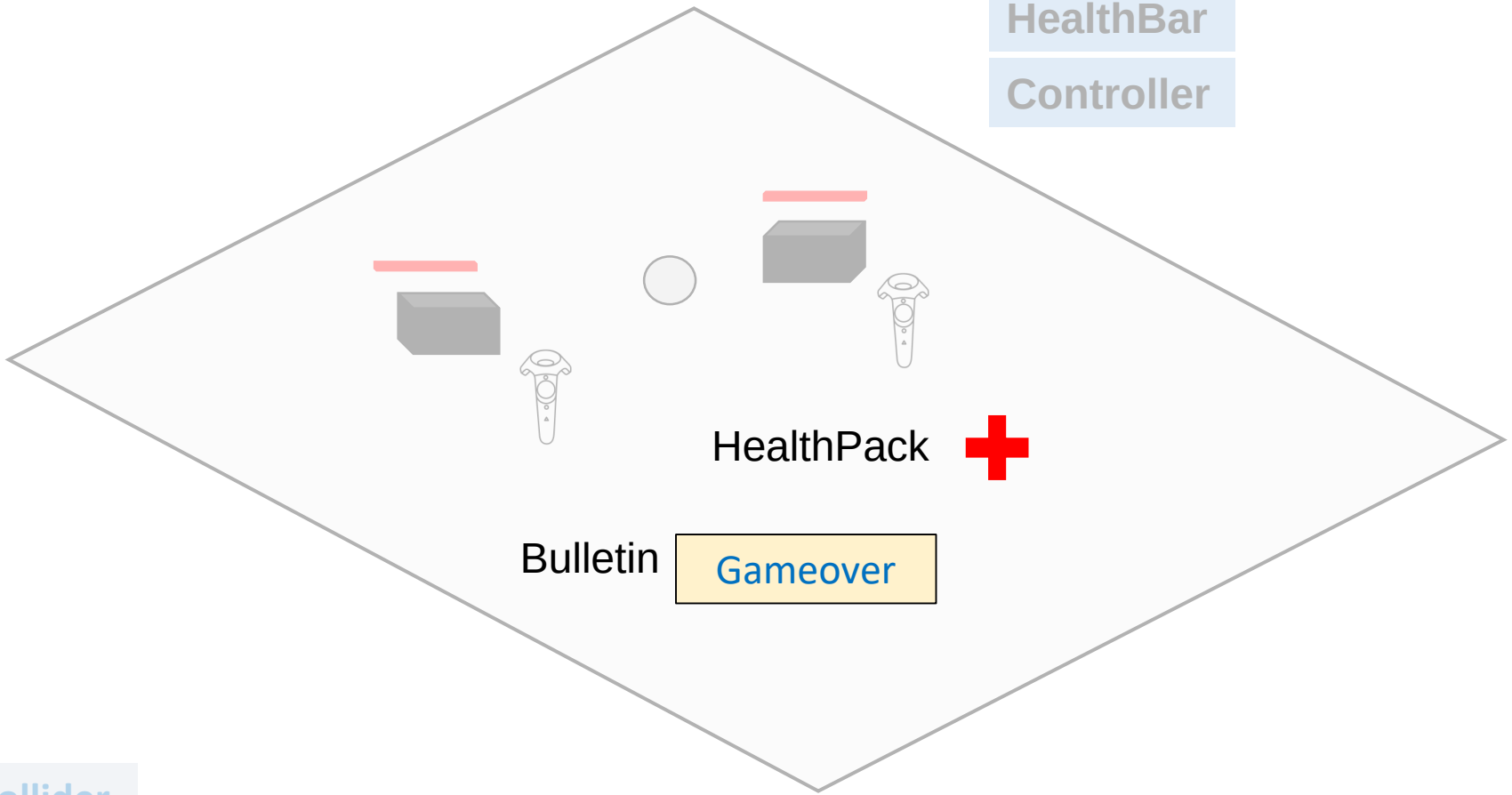
Transform
[GameManager.cs]
[HealthPackControl.cs]

PlayerAvatar

HeadSet

HealthBar

Controller



Snowball

Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappear.cs]

Back to NetworkManager.cs

- We set the first game client as the master.
- Master manage the global states, some of the GameObjects, and game logic of a game.

```
public override void OnJoinedRoom()
{
    Debug.Log("OnJoinedRoom() called by PUN. Now this client is in a room. From here on, your game wo
    local_player = PhotonNetwork.Instantiate(avatar_prefab.name, Vector3.zero, Quaternion.identity);
    //Check if the player is the master client
    if(PhotonNetwork.IsMasterClient)
    {
        //If he/she is, then activate the server gameobject
        //Which means that he/she will play as the server as well
        Master.SetActive(true);
    }
}
```

Master

- GameManager.cs
  - GAMEOVER()

```
public GameObject gameOverMessage;

public void GAMEOVER()
{
    PhotonNetwork.Instantiate(gameOverMessage.name, new Vector3(0, 0.5f, 0), Quaternion.identity);
    GetComponent<HealthPackControl>().enabled = false;
}
```

Master

- HealthPackControl.cs
  - PhotonNetwork.Instantiate() the health pack repeatedly
  - PhotonNetwork.destroy() the health pack if exist too long

Snowball Fight

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

Transform
[ViveManager.cs]

Master

Transform
[GameManager.cs]
[HealthPackControl.cs]

Snowball

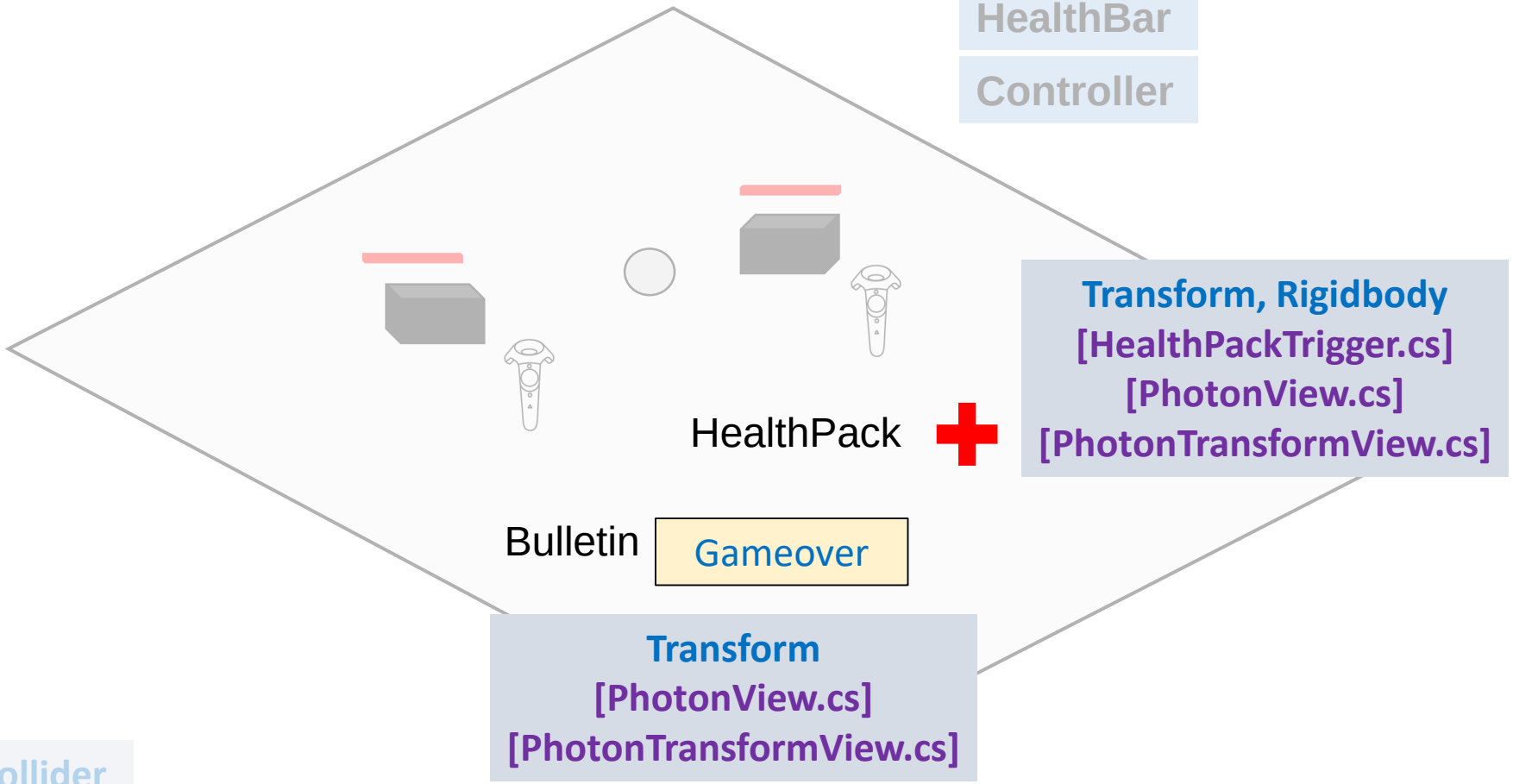
Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappear.cs]

Environment

Transform
Renderer

PlayerAvatar

HeadSet
HealthBar
Controller



HealthPackTrigger.cs

```
void OnTriggerEnter(Collider other)
{
    //if the health pack is hit by a snowball
    if (other.CompareTag("SnowBall"))
    {
        if (other.GetComponent<PhotonView>().IsMine)
        {
            NetworkManager.Instance.local_player.transform.GetChild(0).gameObject.GetComponent<PlayerState>().AddHealth(1);
            //Destroy the snowball
            PhotonNetwork.Destroy(other.gameObject);
        }
        HealthPackControl.time = 0.0f;
        HealthPackControl.isPackExist = false;
        //destroy the health pack
        GetComponent<PhotonView>().TransferOwnership(PhotonNetwork.LocalPlayer);
        PhotonNetwork.Destroy(this.gameObject);
    }
}
```

Why TransferOwnership?

- OnTriggerEnter is only activated once, although there are multi-VR users’ scene.
- In PlayerState.cs:
  - Player A instantiates a snowball
  - That snowball hits player B
  - Only player A ‘s PlayState.cs enters the OnTriggerEnter() function

- You can only destroy the GameObject you created.

HealthPackTrigger.cs

- The health pack is created by Master
- So the game client has to transfer the ownership of health pack first in order to destroy it.

```
    //destroy the health pack
    GetComponent<PhotonView>().TransferOwnership(PhotonNetwork.LocalPlayer);
    PhotonNetwork.Destroy(this.gameObject);
}
```

Snowball Fight

Environment

Transform
Renderer

NetworkManager

Transform
[NetworkManager.cs]

ViveManager

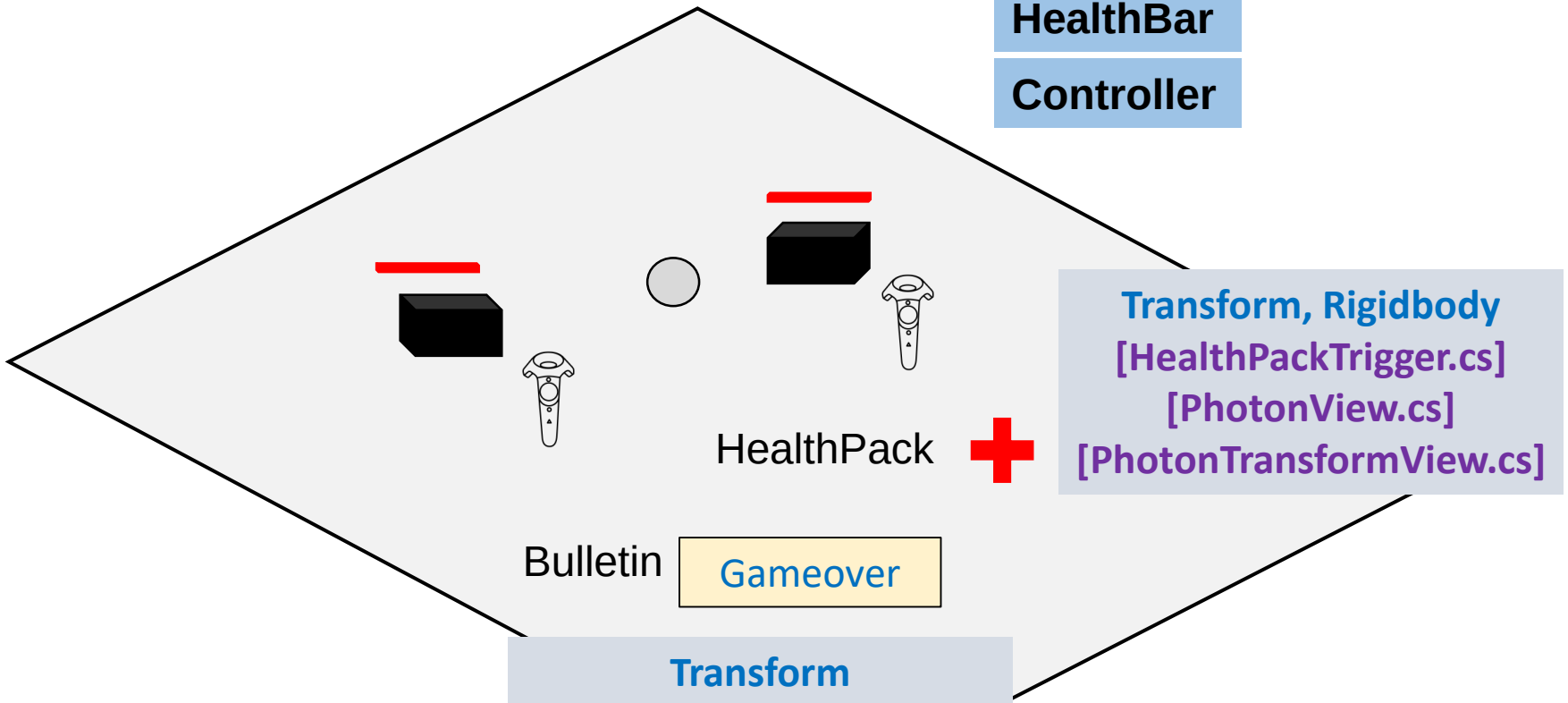
Transform
[ViveManager.cs]

Master

Transform
[GameManager.cs]
[HealthPackControl.cs]

Snowball

Transform, Rigidbody, Collider
[PhotonView.cs]
[PhotonTransformView.cs]
[CountDownDisappear.cs]



Collocated Setup

Collocated Setup

- To make to VR collocated in the same lighthouse setting.
- Copy two configuration files from one of the project.

- Steam > config > chaperone_info.vrchap
- Steam > config > lighthouse > lighthousedb.json

Collocated Setup

:all (C:) › Program Files (x86) › Steam › config

Name	Date modified	Type	Size
gamepad	8/30/2018 5:29 PM	File folder	
lighthouse	11/18/2018 6:43 PM	File folder	
oculus	8/30/2018 5:29 PM	File folder	
oculus_legacy	8/30/2018 5:29 PM	File folder	
vrappconfig	8/30/2018 5:30 PM	File folder	
appconfig.json	11/14/2018 10:57 ...	JSON File	1 KB
chaperone_info.vrchap	11/15/2018 12:31 ...	VRCHAP File	9 KB
config.vdf	11/14/2018 11:22 ...	VDF File	16 KB

ll (C:) › Program Files (x86) › Steam › config › lighthouse

Searc

Name	Date modified	Type	Size
lighthousedb.json	11/18/2018 6:43 PM	JSON File	33 KB
lhr-0fa7b2a6	11/14/2018 12:24 AM	File folder	
lhr-ffed1947	9/25/2018 1:32 PM	File folder	
lhr-ffe65f45	9/12/2018 1:04 PM	File folder	
lhr-ffd71942	9/5/2018 9:07 PM	File folder	
lhr-01b5e13b	9/3/2018 12:16 PM	File folder	
lhr-1ec37302	9/3/2018 12:15 PM	File folder	
lhr-f48b4ffb	8/30/2018 5:30 PM	File folder	
lhr-f7eebd40	8/30/2018 5:30 PM	File folder	
lhr-fdaf7bc3	8/30/2018 5:30 PM	File folder	

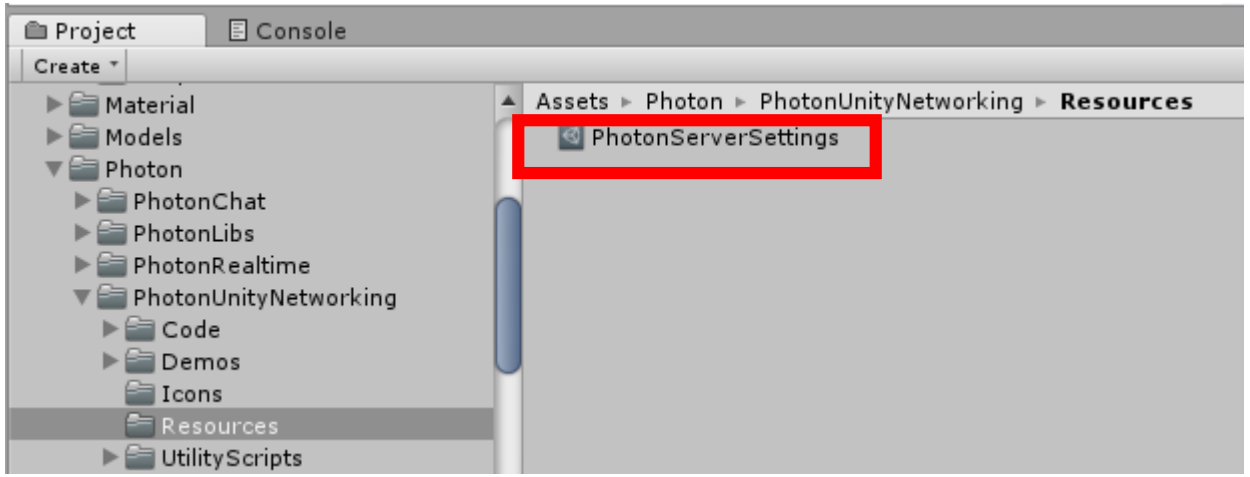
Note: Region Setting

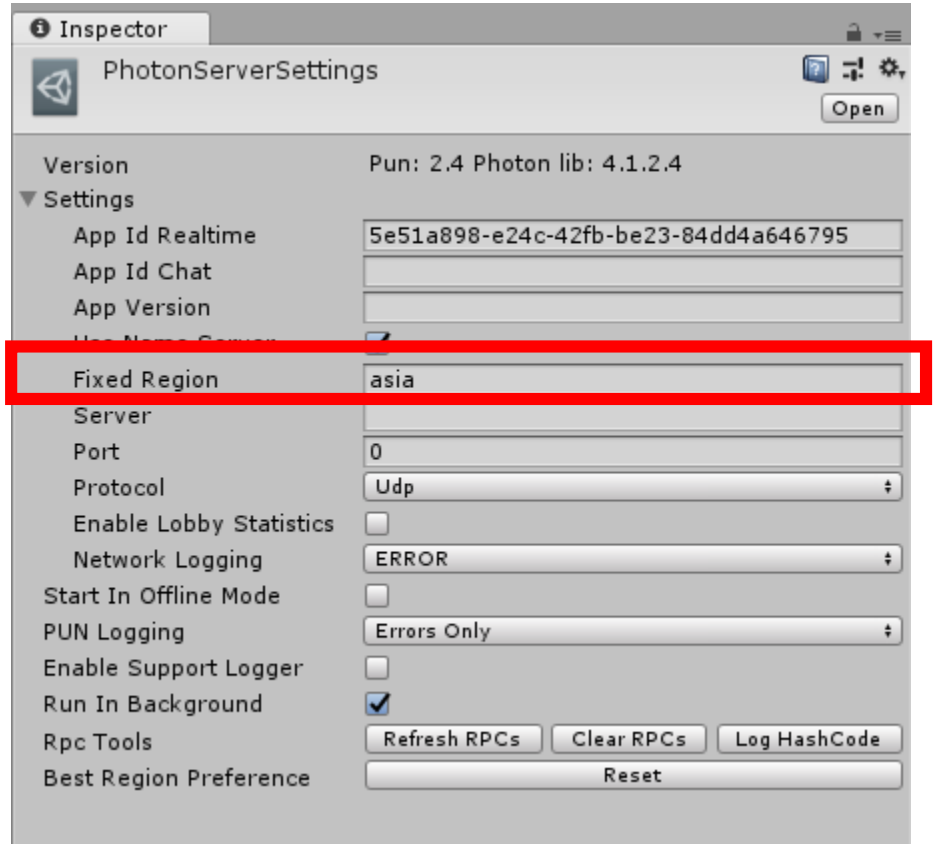
- A list of available regions and tokens

Region	Hosted in	Token
Asia	Singapore	asia
Australia	Melbourne	au
Canada, East	Montreal	cae
Chinese Mainland (See Instructions)	Shanghai	cn
Europe	Amsterdam	eu

Note: Region Setting

- Set Fixed region of your projects to the same one (e.g. asia)





References

- PUN Documentation (v1.9)
- PUN 2.0 Migration
- Photon Forum