INSTITUTO DE INFORMÁTICA
Universidade Federal de Goiás

Jogos Multiplayer e Sistemas Distribuídos

Felipe Gabriel Fraga Pinto
Rafael Langaro Passarinho
Rodrigo Resende Moreira

2022

INSTITUTO DE INFORMÁTICA
UFG

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

Agenda

- Arquiteturas Disponíveis
- Ferramentas

Arquiteturas Disponíveis

Multiplayer local

1. Hotseat
2. Split screen
3. LAN (Local Area Network)

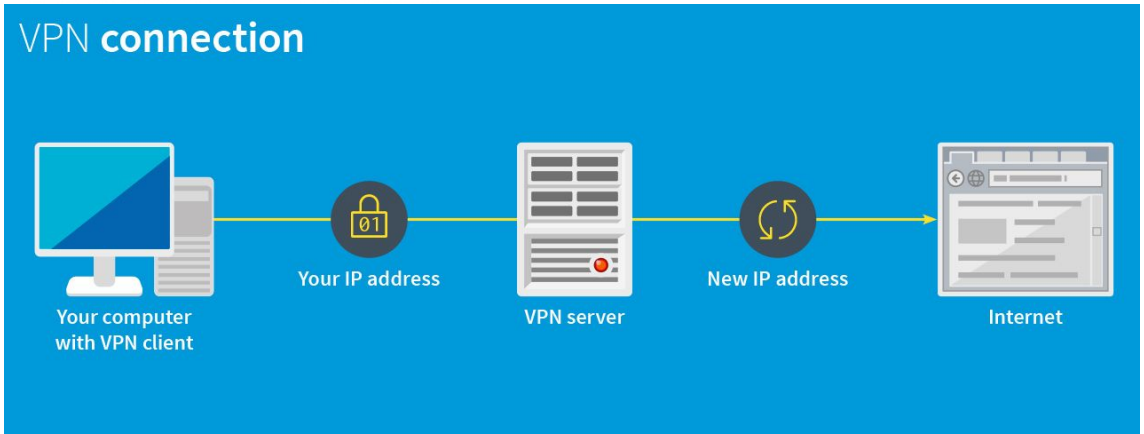
Multiplayer local

Layout: Simple right

VPN *(Virtual private network)*

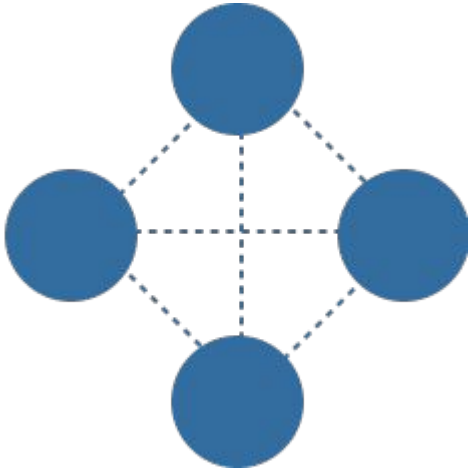
Garena

LogMeIn Hamachi

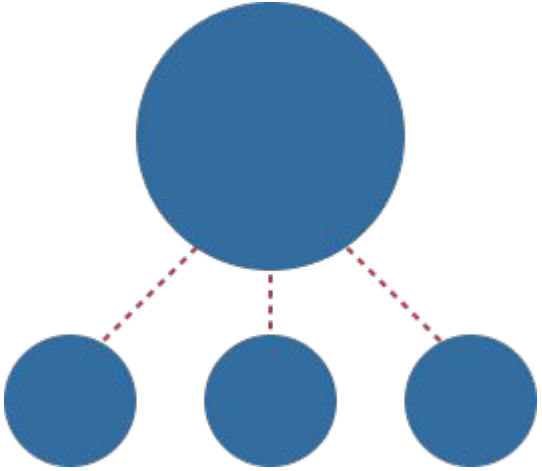


P2P *(Peer-to-peer)*

Direct



Player-Hosted

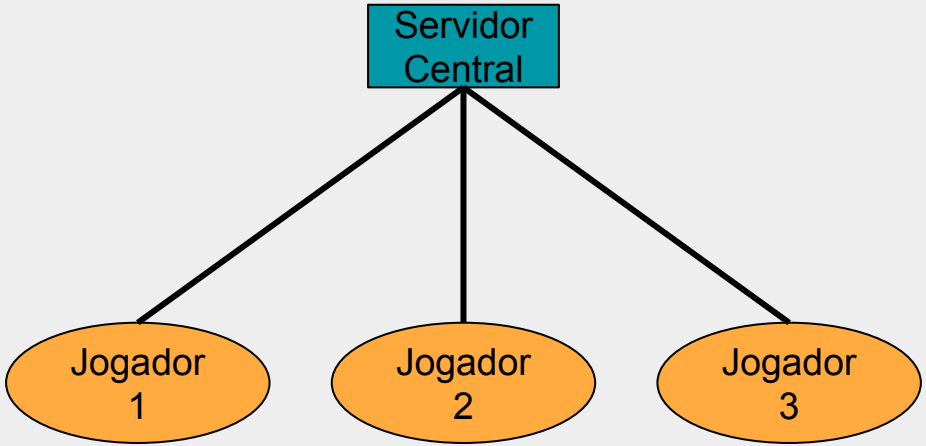


Servidor Dedicado

PLAYFAB

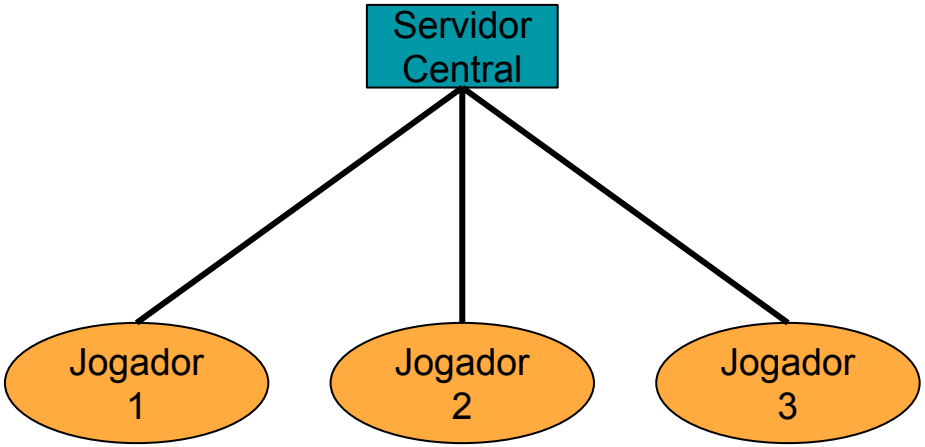
Comparação

Multiplayer local/LAN	P2P
Baixa latência	Baixo custo
Baixo custo	Maior alcance/escal.
Alcance limitado	Baixa segurança (cheat)
Sem escalabilidade	Vantagem Host



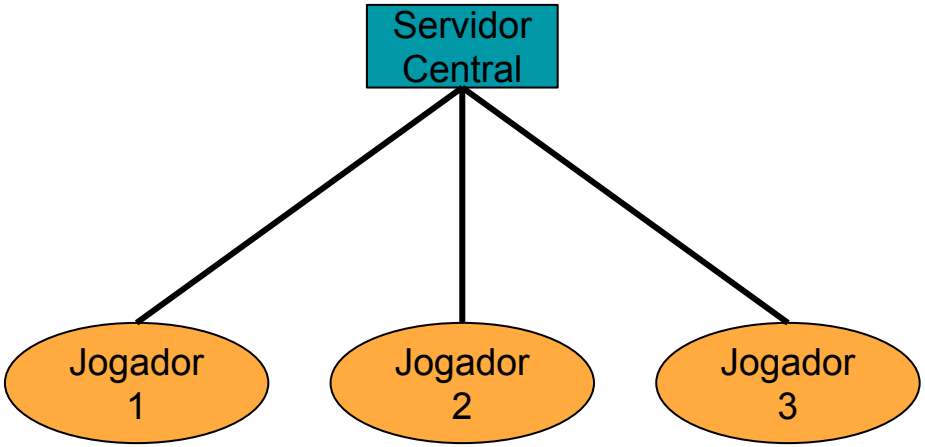
Cliente Servidor

Cliente-Servidor

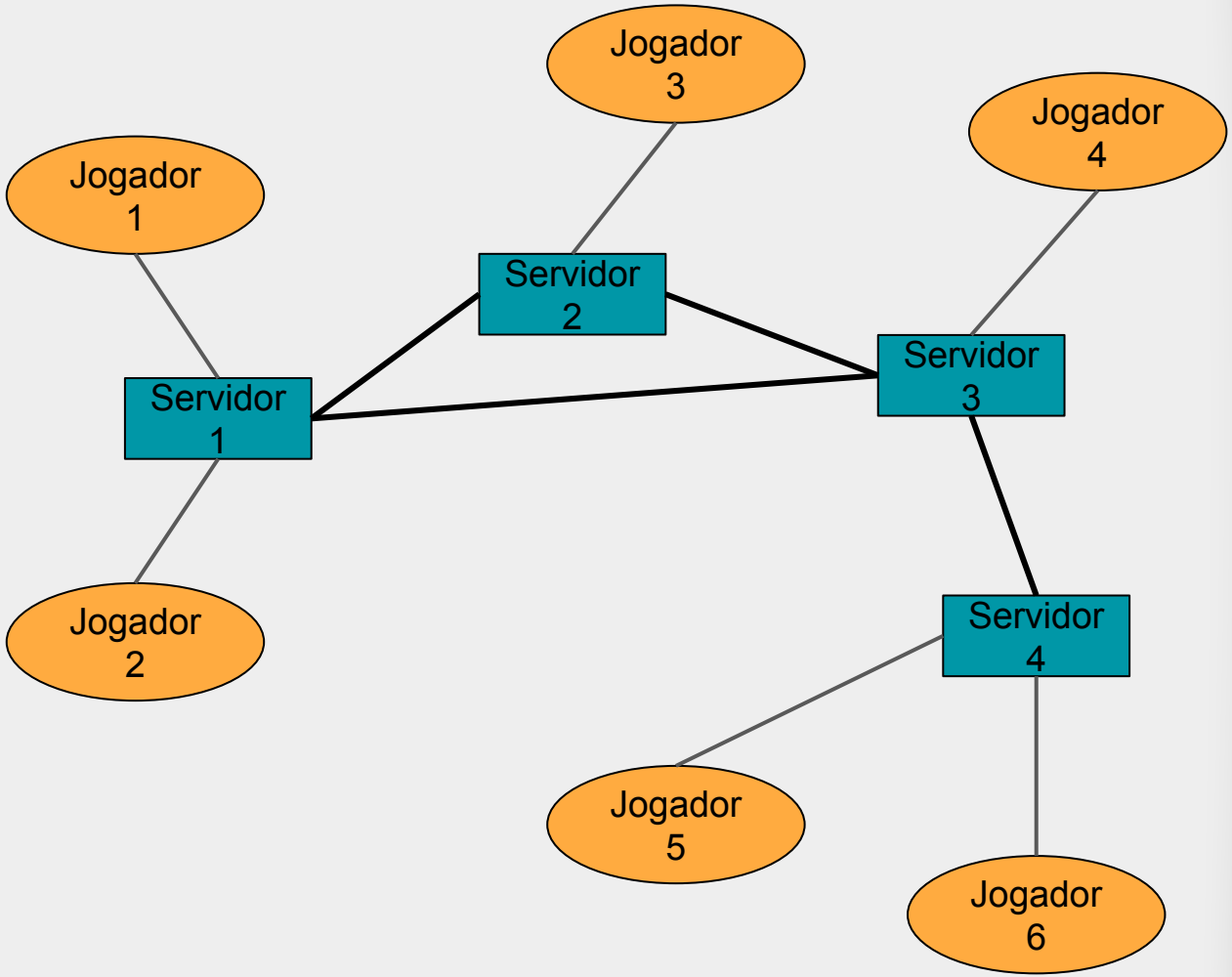


- Todas as funções e informações são centralizadas em um único Servidor com um grupo de Clientes conectados ao Servidor para enviar e receber dados;
- Como apenas o Servidor tem autoridade dentro de todo o jogo, não há problemas de consistência no sistema;

Cliente-Servidor

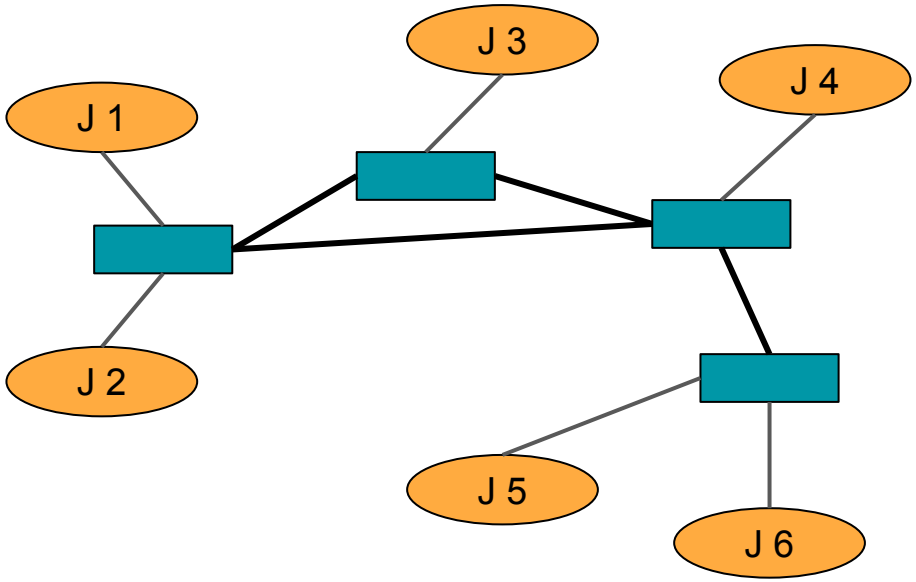


- Como todos os Clientes estão conectados a um Servidor, cada comando vai ter de ser enviado ao Servidor e depois enviado novamente para outros Clientes, adicionando latência;
- Como toda a comunicação entre Clientes têm que ser transferida pelo Servidor, possui uma alta carga de serviço e o Servidor se torna um gargalo.

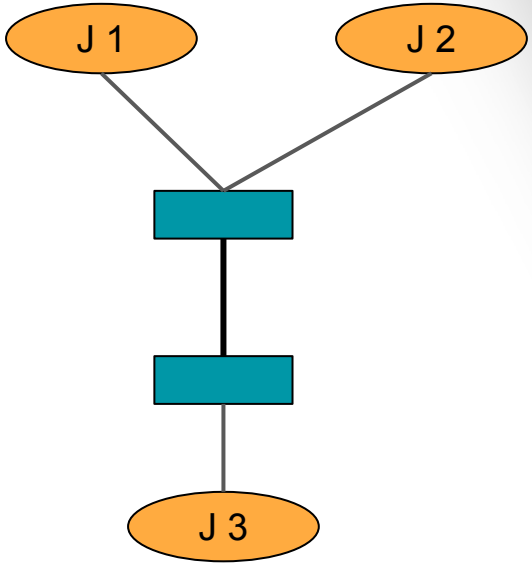


Servidor Distribuído

Servidor Distribuído



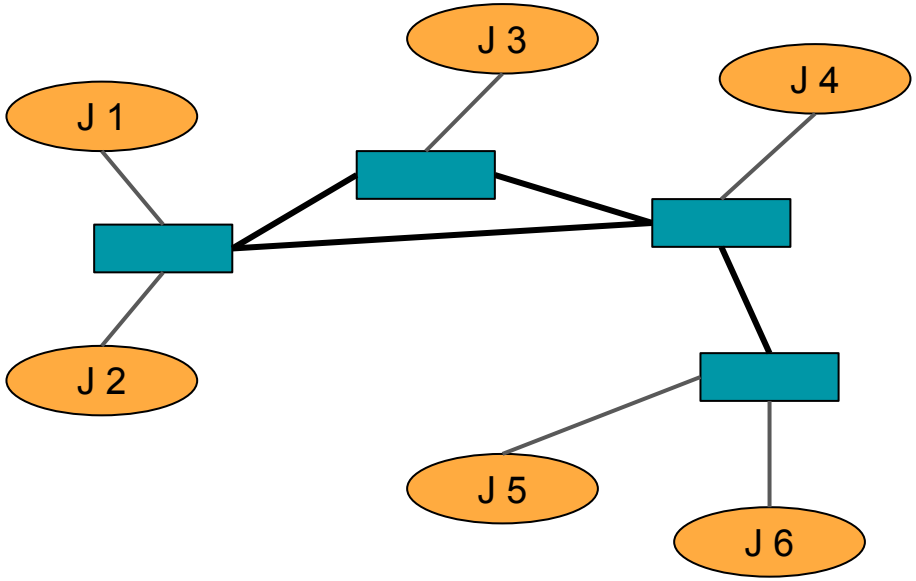
Servidor Distribuido



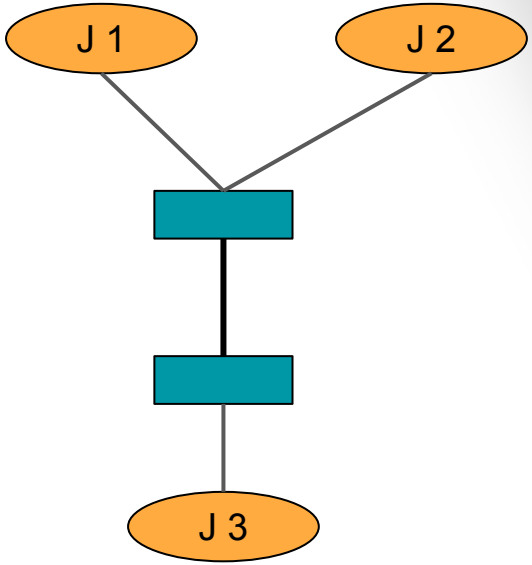
Servidor Distribuído Espelhado

- Como todos os Servidores estão geograficamente distribuídos, todo Cliente pode escolher um Servidor que forneça a menor latência;

Servidor Distribuído



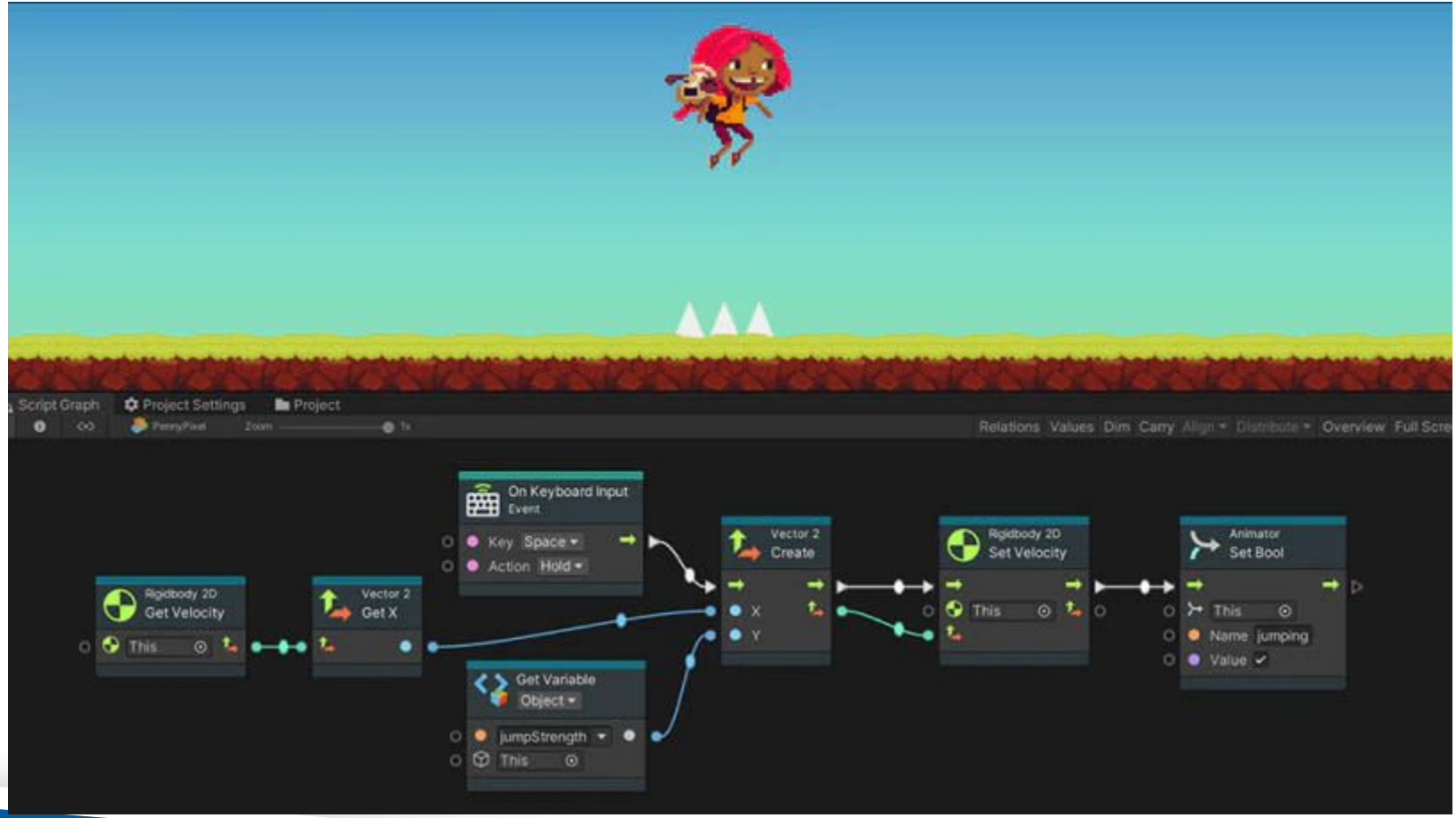
Servidor Distribuido



Servidor Distribuído Espelhado

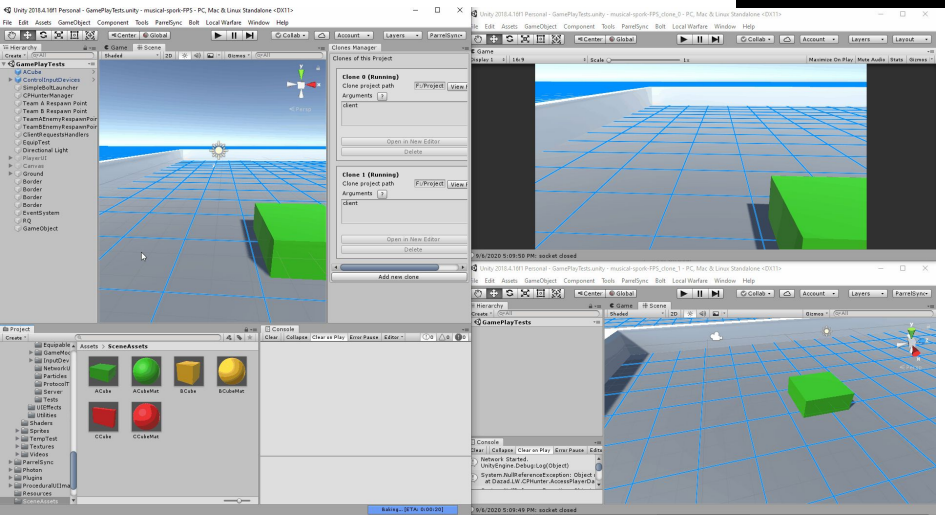
- Além da estrutura multi-servidor distribuir a carga do Servidor, esta estrutura também distribui o risco de falhas no Sistema.

Ferramentas


Script Graph
Project Settings
Project
Relations
Values
Dim
Carry
Overview
On Keyboard Input
Event
Key Space
Action Hold
Rigidbody 2D
Get Velocity
This
Vector 2
Get X
Vector 2
Create
X
Y
Rigidbody 2D
Set Velocity
This
Animator
Set Bool
This
Name jumping
Value
Get Variable
Object
jumpStrength
This

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DemoScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

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

    }
}
```

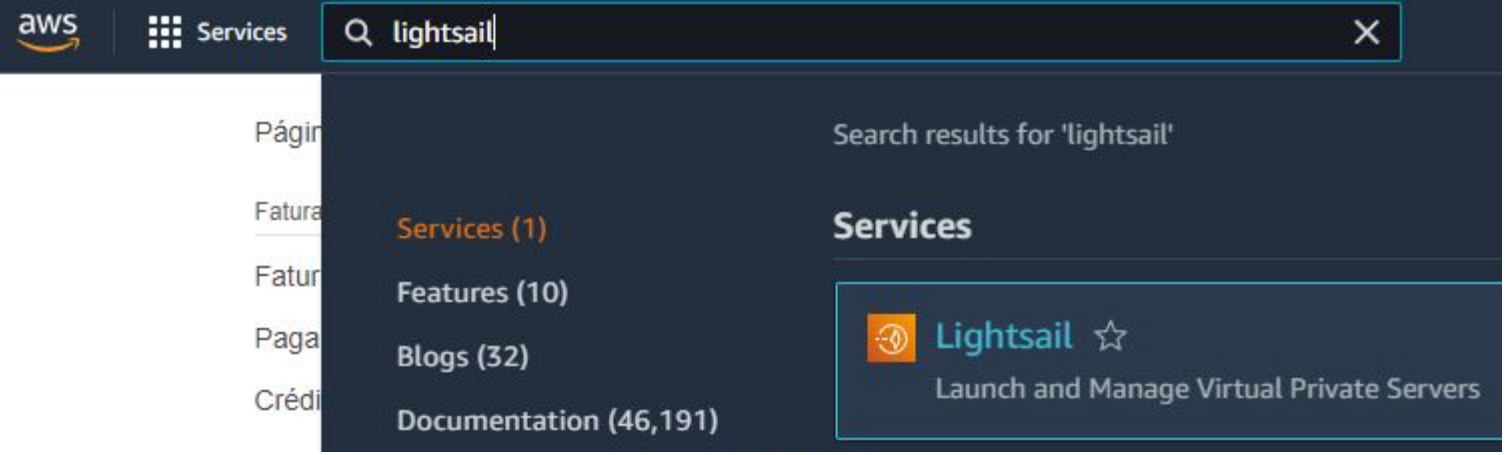
Bibliotecas



https://github.com/VeriorPies/ParrelSync/

https://mirror-networking.gitbook.io/docs/


photon



Sort by Date

Create instance

servidor-teste

1 GB RAM, 1 vCPU, 40 GB SSD

Running

3.238.66.184

2600:1f18:425e:d700:2269:aa9a:45e0:87b4

Virginia, Zone A

Instance location

You are creating this instance in **Virginia, Zone A** (us-east-1a)

Change AWS Region and Availability Zone

Pick your instance image

Select a platform

Linux/Unix
28 blueprints

Microsoft Windows
4 blueprints

Select a blueprint

Apps + OS | OS Only

Amazon Linux 2
2.0.20220805.0

Amazon Linux
2018.03.0.202...

Ubuntu
20.04 LTS

Ubuntu
18.04 LTS

Ubuntu
16.04 LTS

Debian
10.8

Debian
9.13

Debian
8.7

FreeBSD
12.3

openSUSE
15.2

CentOS
8 2004-01

CentOS
7 2009-01

servidor-teste

1 GB RAM, 1 vCPU, 40 GB SSD

Amazon Linux 2

Virginia, Zone A (us-east-1a)

Stop Reboot

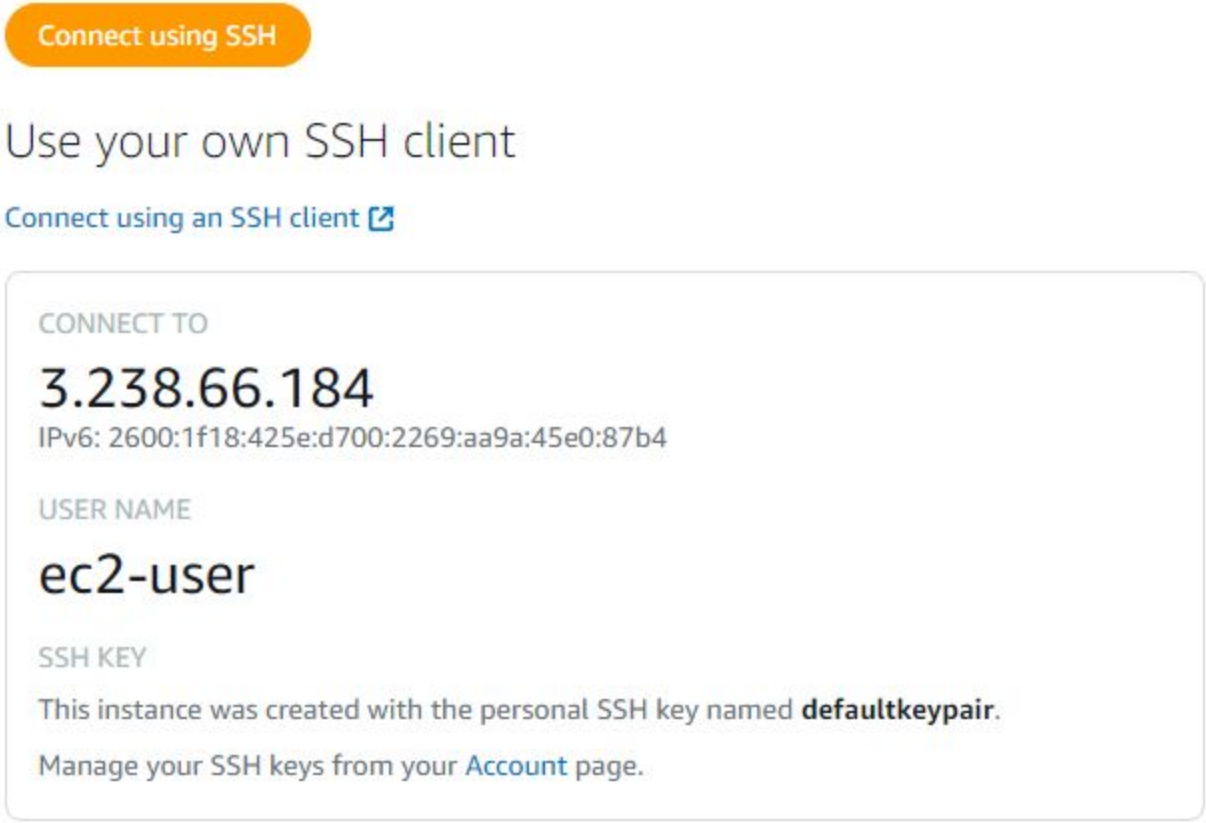
Status: **Running**

Public IP: **3.238.66.184**

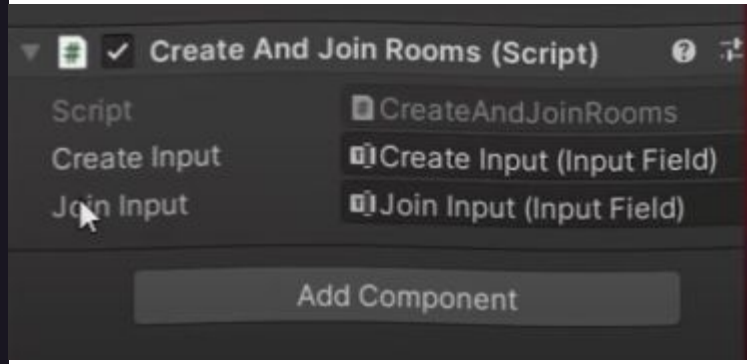
Private IP: 172.26.7.144

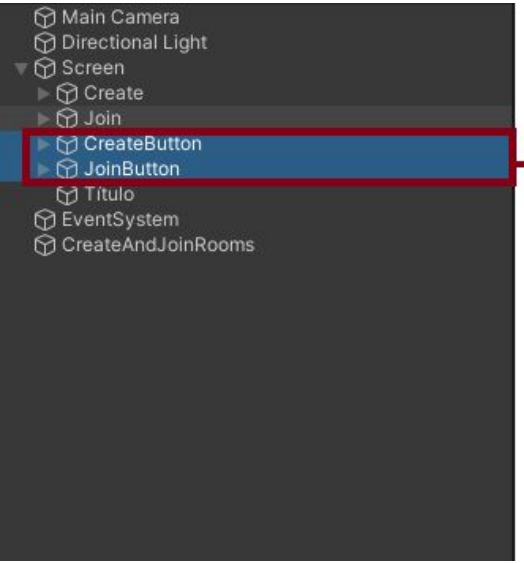
Public IPv6: 2600:1f18:425e:d700:2269:aa9a:45e0:87b4

Learn more about IPv6


Connect using SSH
Use your own SSH client
Connect using an SSH client
CONNECT TO
3.238.66.184
IPv6: 2600:1f18:425e:d700:2269:aa9a:45e0:87b4
USER NAME
ec2-user
SSH KEY
This instance was created with the personal SSH key named defaultkeypair.
Manage your SSH keys from your Account page.

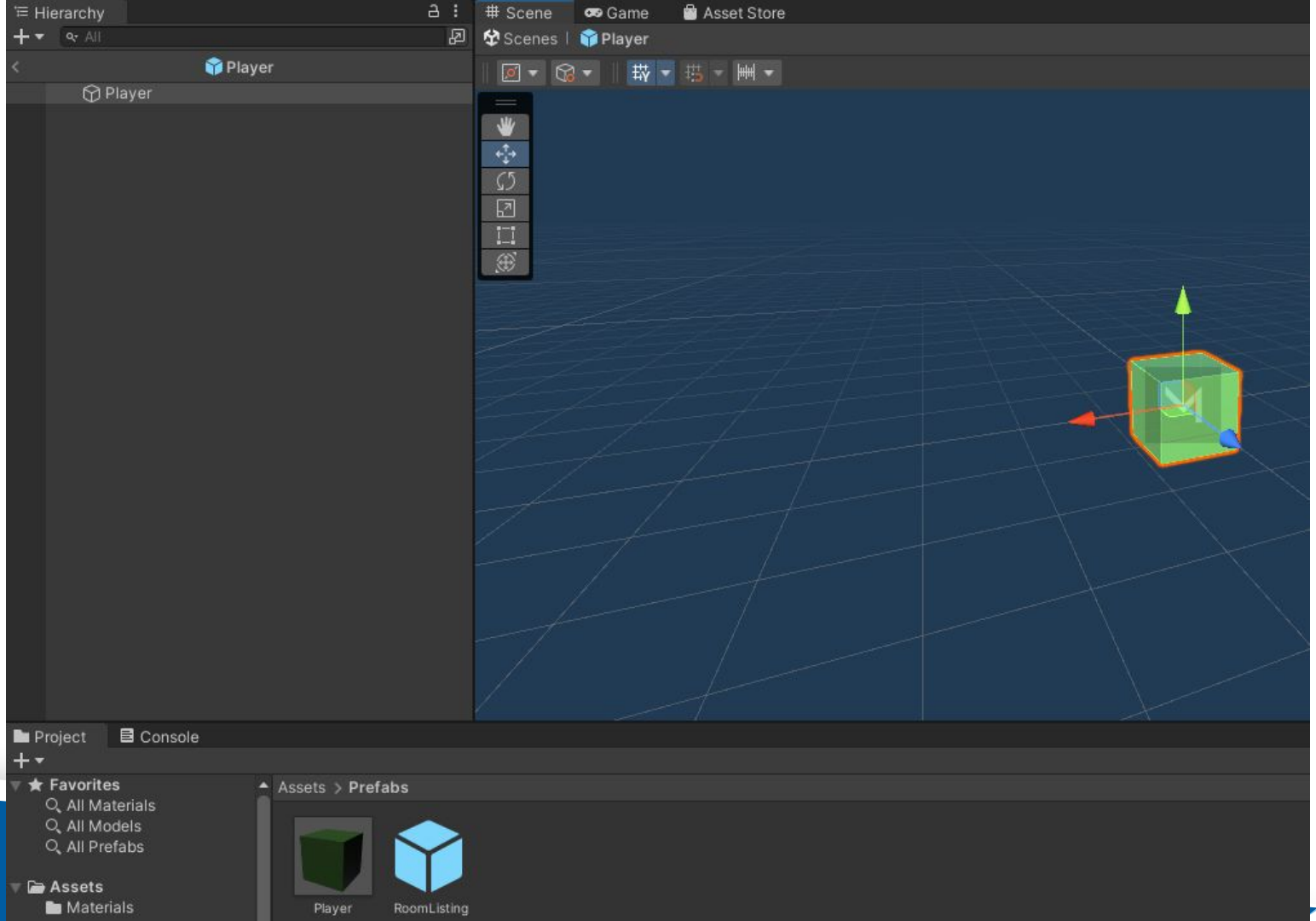
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using Photon.Pun;

public class CreateAndJoinRooms : MonoBehaviourPunCallbacks
{
    public InputField createInput;
    public InputField joinInput;

    public void CreateRoom()
    {
        PhotonNetwork.CreateRoom(createInput.text);
    }

    public void JoinRoom()
    {
        PhotonNetwork.JoinRoom(joinInput.text);
    }

    public override void OnJoinedRoom()
    {
        PhotonNetwork.LoadLevel("Game");
    }
}
```



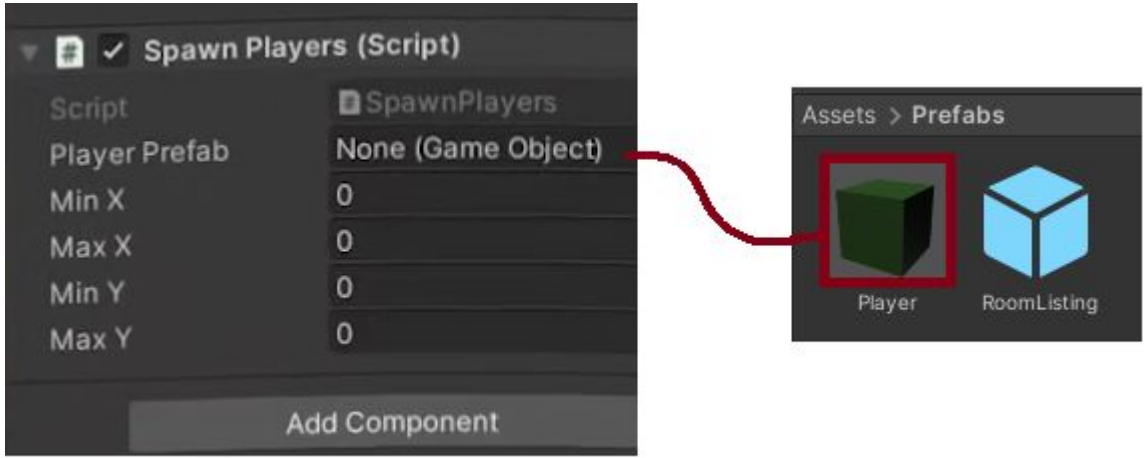


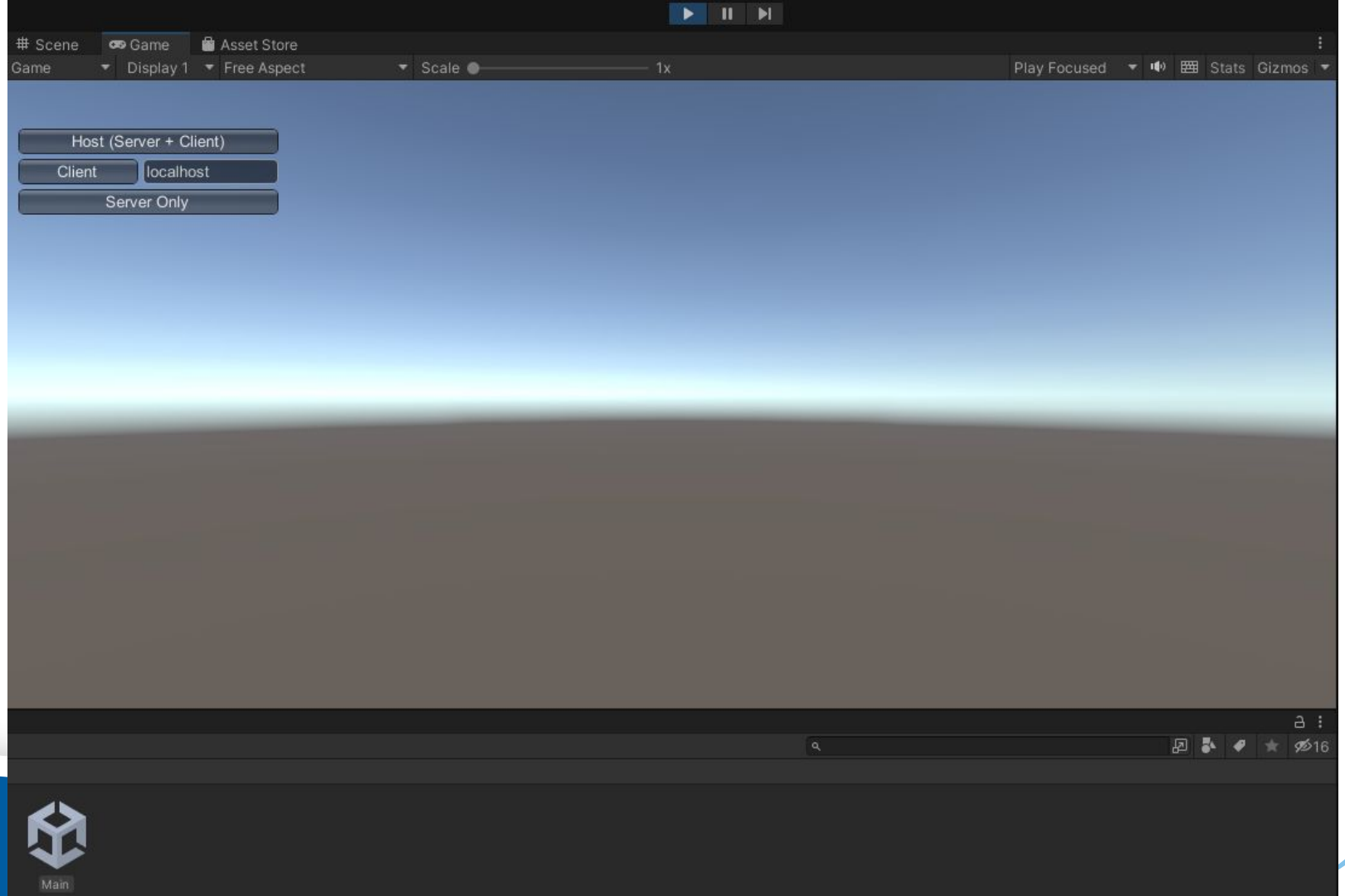
Sistemas Distribuídos - Jogos Multiplayer

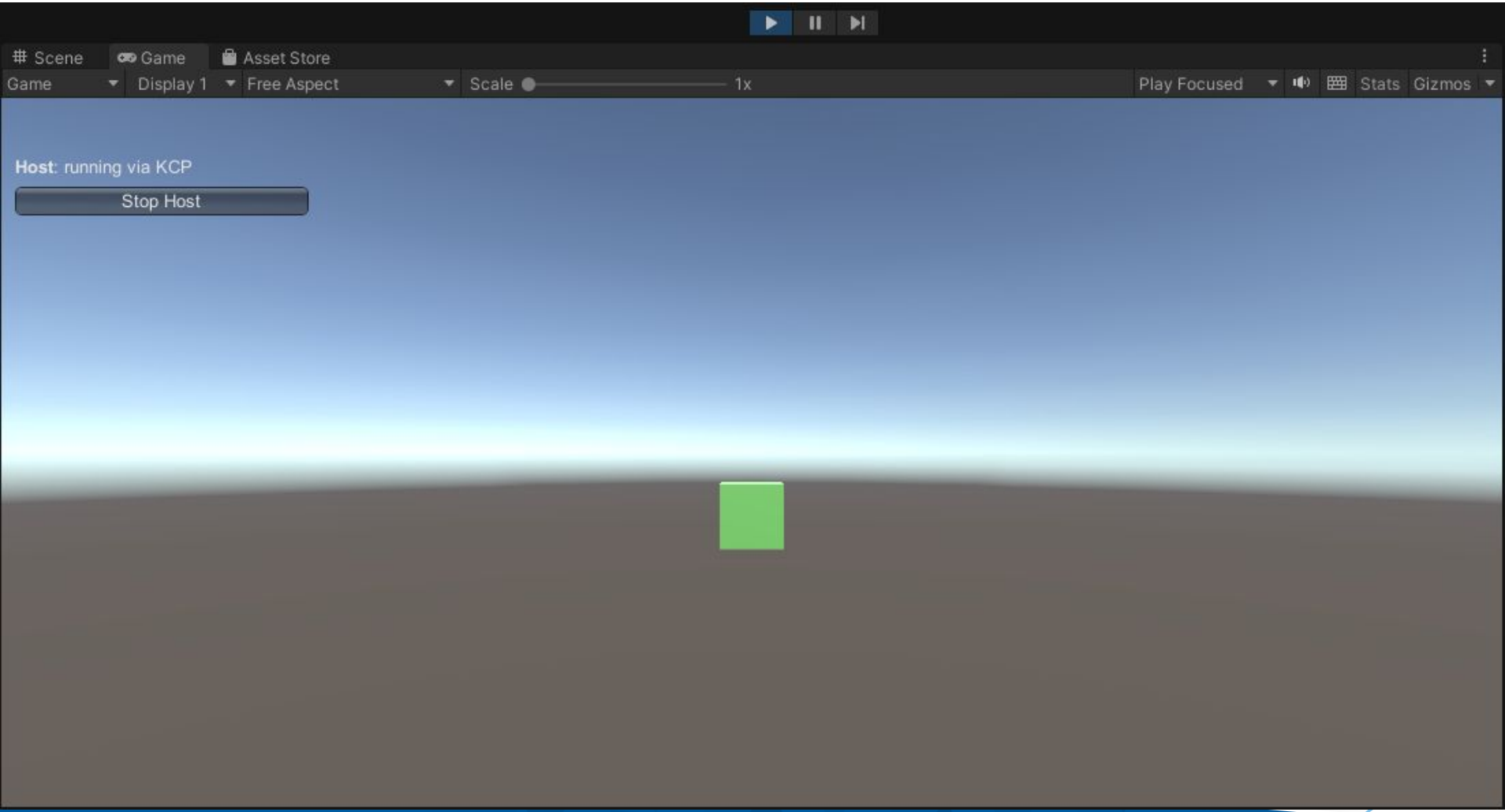
Digite nome

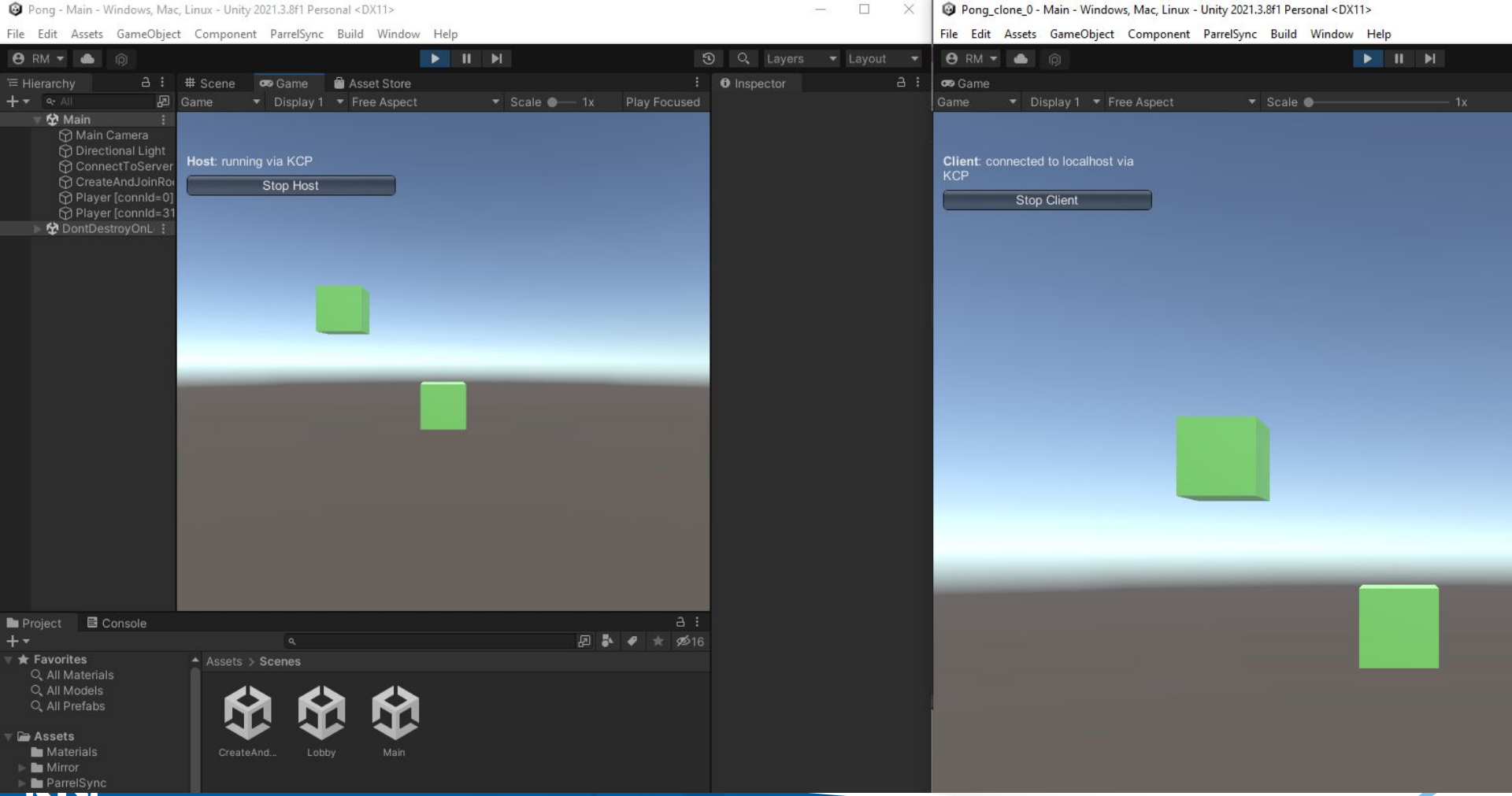

Hierarchy
All
Player
Player
Scene
Game
Asset Store
Scenes
Player
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Materials
Assets
Prefabs
Player
RoomListing

```
public class SpawnPlayers : MonoBehaviour
{
    public GameObject playerPrefab;

    public float minX;
    public float maxX;
    public float minZ;
    public float maxZ;
    // Start is called before the first frame update
    void Start()
    {
        Vector3 randomPosition = new Vector3(Random.Range(minX, maxX), 0, Random.Range(minZ, maxZ));
        PhotonNetwork.Instantiate(playerPrefab.name, randomPosition, Quaternion.identity);
    }
```


Spawn Players (Script)
Script
SpawnPlayers
Player Prefab
None (Game Object)
Min X
0
Max X
0
Min Y
0
Max Y
0
Add Component
Assets > Prefabs
Player
RoomListing

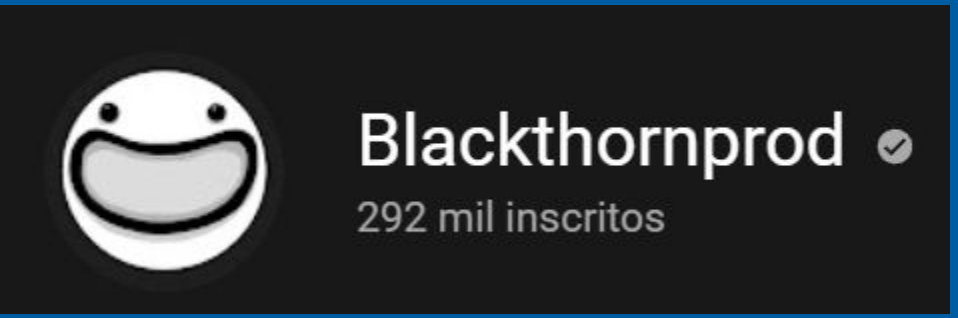

Scene
Game
Asset Store
Game
Display 1
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Host (Server + Client)
Client
localhost
Server Only
16
Main

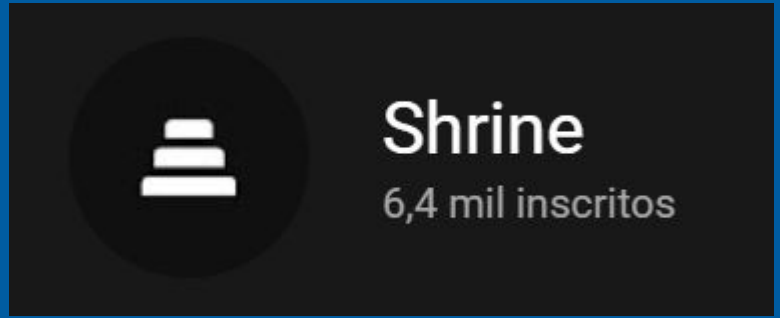

Scene
Game
Asset Store
Game
Display 1
Free Aspect
Scale
1x
Play Focused
Stats
Gizmos
Host: running via KCP
Stop Host



https://www.loom.com/share/d9e89d96a39c4b24972c842c56a7c7a2

Obrigado





Dúvidas ou **sugestões**?

INSTITUTO DE INFORMÁTICA UFG

UFG UNIVERSIDADE FEDERAL DE GOIÁS