

LAMPIRAN



KUESIONER TUGAS AKHIR – 1461505293 MUHAMMAD IMAM ROJIB

PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : Moch Sholeh

Usia : 20

Status Sosial : MAHASISWA

A. KARAKTERISTIK RESPONDEN

1. Apakah Anda Pengguna Game Android

a. Ya b. Tidak

B. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (y)

SS = Sangat Setuju S = Setuju C = Cukup

TS = Tidak Setuju STS = Sangat Tidak Setuju

NO	Parameter	Penilaian				
		STS	TS	C	S	SS
Parameter Antarmuka						
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
Parameter Performan dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan		✓			
5	Aplikasi menampilkan perpindahan menu dengan lancar		✓			
6	Aplikasi berjalan dengan lancar		✓			
Parameter Gameplay						
1	Gameplay aplikasi jarang ditemui di aplikasi lain			✓		
2	Gameplay aplikasi membuat pemain betah bermain		✓			

C. Kritik dan Saran

- Buat fitur team
- Kontrol bola kurang baik

Surabaya, 10 November 2022

Moch Sholeh



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : Ramadhan Prandu N.
Usia : 16 thn
Status Sosial : SMA

D. KARAKTERISTIK RESPONDEN

2. Apakah Anda Pengguna Game Android
 a. Ya b. Tidak

E. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (y)

SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

NO	Parameter	Penilaian				
		STS	TS	C	S	SS
Parameter Antarmuka						
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
Parameter Performan dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan					✓
5	Aplikasi menampilkan perpindahan menu dengan lancar				✓	
6	Aplikasi berjalan dengan lancar				✓	
Parameter Gameplay						
1	Gameplay aplikasi jarang ditemui di aplikasi lain				✓	
2	Gameplay aplikasi membuat pemain betah bermain					✓

F. Kritik dan Saran

.....
..... Kontroler bola kurang bagus

Surabaya, 10 November 2022

..... Ramadhan Prandu N.



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : RYAN ABUHAJI
Usia : 10
Status Sosial : SD

G. KARAKTERISTIK RESPONDEN

3. Apakah Anda Pengguna Game Android
Ya b. Tidak

H. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (y)
SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

Table with 6 columns: NO, Parameter, and Penilaian (STS, TS, C, S, SS). Rows include categories like Parameter Antarmuka, Parameter Performan dan Kenyamanan, and Parameter Gameplay.

I. Kritik dan Saran

Handwritten feedback: 'tampilan cuma 1 tampilan'

Surabaya, 10 November 2022

Handwritten signature

RYAN ABUHAJI



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : MUHAMMAD IMAM ROJIB
Unit : IS
Status Sosial : SMP

J. KARAKTERISTIK RESPONDEN

4. Apakah Anda Pengguna Game Android
 Ya Tidak

K. PENILAIAN TERHADAP APLIKASI

Isilah table dibawah ini dengan menggunakan tanda (√)

SS = Sangat Sesuai S = Sesuai C = Cukup
TS = Tidak Sesuai TSS = Sangat Tidak Sesuai

NO	Parameter	Positium				
		STS	TS	C	S	SS
Parameter Antarmuka						
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	✓
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai					✓
Parameter Performansi dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan					✓
5	Aplikasi menampilkan penyusunan menu dengan lancar				✓	
6	Aplikasi berjalan dengan lancar				✓	
Parameter Gameplay						
1	Gameplay aplikasi yang ditawarkan di aplikasi lain				✓	
2	Gameplay aplikasi membuat pemain betah bermain					✓

L. Kritik dan Saran

Kemudahan yang baik

Surabaya, 10 November 2022

[Signature]



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : SADAM APRILIA NTO

Usia : 30

Status Sosial : Pekerja

M. KARAKTERISTIK RESPONDEN

5. Apakah Anda Pengguna Game Android

Ya b. Tidak

N. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (y)

SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

NO	Parameter	Penilaian				
		STS	TS	C	S	SS
Parameter Antamuka						
1	Aplikasi memiliki tampilan dan desain yang menarik					✓
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai					✓
Parameter Performan dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan					✓
5	Aplikasi menampilkan perpindahan menu dengan lancar				✓	
6	Aplikasi berjalan dengan lancar					✓
Parameter Gameplay						
1	Gameplay aplikasi jarang ditemui di aplikasi lain				✓	
2	Gameplay aplikasi membuat pemain betah bermain				✓	

O. Kritik dan Saran

.....
..... Sering Bug botaknya

Surabaya, 10 November 2022

SADAM



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : Muhammad Sony Setiawan
 Usia : 16
 Status Sosial : SMA

BB. KARAKTERISTIK RESPONDEN

10. Apakah Anda Pengguna Game Android
 j. Ya Tidak

CC. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (y)
 SS = Sangat Setuju S = Setuju C = Cukup
 TS = Tidak Setuju STS = Sangat Tidak Setuju

NO	Parameter	Penilaian				
		STS	TS	C	S	SS
Parameter Antarmuka						
1	Aplikasi memiliki tampilan dan desain yang menarik				L	
2	Aplikasi memiliki menu yang mudah digunakan				L	
3	Aplikasi memiliki tata letak tombol yang sesuai				L	
Parameter Performan dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan				L	
5	Aplikasi menampilkan perpindahan menu dengan lancar			L		
6	Aplikasi berjalan dengan lancar				L	
Parameter Gameplay						
1	Gameplay aplikasi jarang ditemui di aplikasi lain				L	
2	Gameplay aplikasi membuat pemain betah bermain			L		

DD. Kritik dan Saran

.....

Surabaya, 12 November 2022

M. Sony Setiawan



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : Ham Anugrah
Usia : 11
Status Sosial : SD

Y. KARAKTERISTIK RESPONDEN

9. Apakah Anda Pengguna Game Android
i. Ya Tidak

Z. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (y)

SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

NO	Parameter	Penilaian				
		STS	TS	C	S	SS
Parameter Antarmuka						
1	Aplikasi memiliki tampilan dan desain yang menarik					L
2	Aplikasi memiliki menu yang mudah digunakan					L
3	Aplikasi memiliki tata letak tombol yang sesuai					L
Parameter Performan dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan					L
5	Aplikasi menampilkan perpindahan menu dengan lancar					L
6	Aplikasi berjalan dengan lancar					L
Parameter Gameplay						
1	Gameplay aplikasi jarang ditemui di aplikasi lain					L
2	Gameplay aplikasi membuat pemain betah bermain					L

AA. Kritik dan Saran

.....
.....
.....
.....

Surabaya, 12 November 2022

Ham Anugrah

Ham Anugrah



PENGEMBANGAN APLIKASI GAME MULTIPLAYER STREET SOCCER DENGAN UNITY 3D

Identitas Responden

Nama : Handam perbalingga

Usia : 21

Status Sosial : MAHASISWA

V. KARAKTERISTIK RESPONDEN

8. Apakah Anda Pengguna Game Android
 h. Ya Tidak

W. PENILAIAN TERHADAP APLIKASI

Istilah table dibawah ini dengan menggunakan tanda (v)

SS = Sangat Setuju S = Setuju C = Cukup

TS = Tidak Setuju STS = Sangat Tidak Setuju

NO	Parameter	Penilaian				
		STS	TS	C	S	SS
Parameter Antarmuka						
1	Aplikasi memiliki tampilan dan desain yang menarik			✓	✓	
2	Aplikasi memiliki menu yang mudah digunakan			✓		
3	Aplikasi memiliki tata letak tombol yang sesuai			✓		
Parameter Performan dan Kenyamanan						
4	Aplikasi nyaman untuk dimainkan				✓	
5	Aplikasi menampilkan perpindahan menu dengan lancar			✓		
6	Aplikasi berjalan dengan lancar			✓		
Parameter Gameplay						
1	Gameplay aplikasi jarang ditemui di aplikasi lain			✓		
2	Gameplay aplikasi membuat pemain betah bermain			✓		

X. Kritik dan Saran

.....

Surabaya, 12 November 2022

Handam

Handam D.

SCRIPT

BILLBOARD

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

public class Billboard : MonoBehaviour
{
    Camera cam;

    void Update()
    {
        if(cam == null)
            cam = FindObjectOfType<Camera>();

        if(cam == null)
            return;

        transform.LookAt(cam.transform);
        transform.Rotate(Vector3.up * 180);
    }
}
```

CONTROLLER

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

[RequireComponent(typeof(Rigidbody),typeof (BoxCollider))]
public class Controller : MonoBehaviour
{
    [SerializeField] private Rigidbody _rigidbody;
    [SerializeField] private FixedJoystick _joystick;
    [SerializeField] private Animator _animator;
    [SerializeField] private float _moveSpeed;
```

```

        private void FixedUpdate()
        {
            _rigidbody.velocity = new
Vector3(_joystick.Horizontal * _moveSpeed,
_rigidbody.velocity.y, _joystick.Vertical * _moveSpeed);

            if( _joystick.Horizontal !=0 || _joystick.Vertical
!=0 )
            {
                transform.rotation =
Quaternion.LookRotation(_rigidbody.velocity);
            }
        }
}

```

LAUNCHER

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Photon.Pun;
using TMPro;
using Photon.Realtime;
using System.Linq;

public class Launcher : MonoBehaviourPunCallbacks
{
    public static Launcher Instance;
    [SerializeField] TMP_InputField roomNameInputField;
    [SerializeField] TMP_Text errorText;
    [SerializeField] TMP_Text roomNameText;
    [SerializeField] Transform roomListContent;
    [SerializeField] Transform playerListContent;
    [SerializeField] GameObject roomListItemPrefab;
    [SerializeField] GameObject PlayerListItemPrefab;
    [SerializeField] GameObject startGameButton;
    [SerializeField] GameObject reloadGameButton;
    void Awake()
    {
        Instance = this;
    }

    // Start is called before the first frame update

```

```

void Start()
{
    Debug.Log("Joined to Server");
    PhotonNetwork.ConnectUsingSettings();
    PhotonNetwork.AutomaticallySyncScene = true ;
}

public override void OnConnectedToMaster()
{
    Debug.Log("Joined to Master");
    PhotonNetwork.JoinLobby();
    PhotonNetwork.AutomaticallySyncScene = true ;
}

public override void OnJoinedLobby()
{
    MenuManager.Instance.OpenMenu("title"); // dari
menumanager.instanc.aktifkan class openmenu dengan string
tittle
    Debug.Log("Joined Lobby");
    PhotonNetwork.NickName = "Player " + Random.Range(0,
1000).ToString("0000");
}

public void CreateRoom()
{
    if(string.IsNullOrEmpty(roomNameInputField.text))
    {
        return;
    }
    PhotonNetwork.CreateRoom(roomNameInputField.text);
    MenuManager.Instance.OpenMenu("loading");
}

public override void OnJoinedRoom()
{
    MenuManager.Instance.OpenMenu("room");
    roomNameText.text = PhotonNetwork.CurrentRoom.Name;

    Player[] players = PhotonNetwork.PlayerList;
    foreach (Transform child in playerListContent)

```

```

        {
            Destroy(child.gameObject);
        }

        for(int i = 0; i < players.Count(); i++)
        {
            Instantiate(PlayerListItemPrefab,playerListContent).
GetComponent<PlayerListItem>().Setup(players[i]);
        }
        startGameButton.SetActive(PhotonNetwork.IsMasterClient
);
    }

    public override void OnMasterClientSwitched(Player
newMasterClient)
    {
        startGameButton.SetActive(PhotonNetwork.IsMasterClient
);
    }

    public override void OnCreateRoomFailed(short returnCode,
string message)
    {
        errorText.text = "Room Creation Failed : " + message;
        MenuManager.Instance.OpenMenu("error");
    }

    public void StartGame()
    {
        PhotonNetwork.LoadLevel(1);
    }
    public void ReloadGame()
    {
        PhotonNetwork.LoadLevel(1);
    }
    public void LeaveRoom()
    {
        PhotonNetwork.LeaveRoom();
        MenuManager.Instance.OpenMenu("loading");
    }

    public void JoinRoom(RoomInfo info)

```

```

    {
        PhotonNetwork.JoinRoom(info.Name);
        MenuManager.Instance.OpenMenu("loading");
    }
    public override void OnLeftRoom()
    {
        MenuManager.Instance.OpenMenu("title");
    }

    public override void OnRoomListUpdate(List<RoomInfo>
roomList)
    {
        foreach(Transform trans in roomListContent)
        {
            Destroy(trans.gameObject);
        }
        for (int i = 0; i < roomList.Count; i++)
        {
            if(roomList[i].RemovedFromList)
                continue;
            Instantiate(roomListItemPrefab,
roomListContent).GetComponent<RoomListItem>().Setup(roomList[i])
;
        }
    }

    public override void OnPlayerEnteredRoom(Player newPlayer)
    {
        Instantiate(playerListItemPrefab,playerListContent).GetC
omponent<PlayerListItem>().Setup(newPlayer);
    }
}

```

MENU

```

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

public class Menu : MonoBehaviour
{
    public string menuName;
}

```

```

public bool open;
public void Open()
{
    open = true;
    gameObject.SetActive(true);
}

public void Close()
{
    open = false;
    gameObject.SetActive(false);
}
}

```

MENU_MANAGER

```

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

public class MenuManager : MonoBehaviour
{
    public static MenuManager Instance;
    [SerializeField] Menu[] menus;

    void Awake()
    {
        Instance = this;
    }

    public void OpenMenu(string menuName)
    {
        for(int i = 0; i < menus.Length; i++)
        {
            if (menus[i].menuName == menuName) // list menu
            canvas == title
            {
                menus[i].Open(); // open/aktiv
            }
            else if (menus[i].open) //jika udah aktif
            {
                CloseMenu(menus[i]); // close
            }
        }
    }
}

```

```

    }
}

public void OpenMenu(Menu menu)
{
    for(int i = 0; i < menus.Length; i++) // cek
    {
        if (menus[i].open) // jika sudah
        {
            CloseMenu(menus[i]); // tutup
        }
    }
    menu.Open(); // open tunggal
}

public void CloseMenu(Menu menu)
{
    menu.Close();
}
}

```

PLAYER_CONTROLLER

```

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

public class PlayerController : MonoBehaviour
{
    Rigidbody rb;
    PhotonView PV;
    Animator anim;
    public float speed;
    public float smooth;
    public float rotationSpeed;
    [SerializeField] private Transform transformPlayer;
    [SerializeField] private Transform BallP;
    [SerializeField] private Transform BallPosition;
    bool stickToPlayer ;
    private Vector2 previousLocation;
    private float speedRotateBall;
}

```

```

private GameObject Bola;

void Awake()
{
    rb = GetComponent<Rigidbody>();
    PV = GetComponent<PhotonView>();
    anim = GetComponentInChildren<Animator>();
}

void Start()
{
    if(!PV.IsMine)
    {
        Destroy(GetComponentInChildren<Camera>().gameObject)
;
        Destroy(rb);
    }
    // posisiBall.instance.Dribble();
}

void Update()
{
    if(!PV.IsMine)
    {
        return;
    }
    Move();
    Shoot();
}

void Move()
{
    float horizontal = Input.GetAxis("Horizontal");
    float vertical = Input.GetAxis("Vertical");

    Vector3 direction = new
Vector3(horizontal,0,vertical).normalized;

```



```

transform.Translate(direction * speed * Time.deltaTime);

if (direction != Vector3.zero)
{
    anim.SetBool ("isRun", true);

}
else
{
    anim.SetBool ("isRun", false);
}
transform.Rotate(0, horizontal * rotationSpeed *
smooth, 0);

}

void Shoot()
{

    /* if (!stickToPlayer) // bola belum menpel ada pemain
    {
        Bola = GameObject.Find("Ball");
        // setting jarak player dengan bola
        float distanceToPlayer =
Vector3.Distance(transformPlayer.position,
Bola.transform.position);
        Debug.Log(distanceToPlayer);
        if (distanceToPlayer < 3) // jika jarak sudah dekat
maka bool aktif
        {
            stickToPlayer = true;
        }
    }
    else
    {
        // set posisi bola agar didepan pemain
        // Vector2 currentLocation = new
Vector2(Bola.transform.position.x, Bola.transform.position.z);
        //speedRotateBall =
Vector2.Distance(currentLocation, previousLocation) /
Time.deltaTime;//lokasi saat ini dan lokasi selanjutnya

```

```

        //Bola.transform.position = BallPosition.position;

        // memutar bola sesuai arah player berjalan
        //Bola.transform.Rotate(new
Vector3(transform.right.x, 0, transform.right.z),
speedRotateBall, Space.World);
        //previousLocation = currentLocation;
    } */

    if (Input.GetKeyDown("space"))
    {

        anim.SetBool ("isRun", false);
        anim.SetTrigger("shoot");
        print("Shoot");
        stickToPlayer = false;
    }
}
}
}

```

PLAYER_LIST_ITEM

```

using Photon.Realtime;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using Photon.Pun;
using System.Linq;

public class PlayerListItem : MonoBehaviourPunCallbacks
{
    [SerializeField] TMP_Text text;
    Player player;
    public void SetUp(Player _player)
    {
        player = _player;
        text.text = _player.NickName;
    }

    public override void OnPlayerLeftRoom(Player otherPlayer)
    {

```

```

        if (player == otherPlayer)
        {
            Destroy(gameObject);
        }
    }

    public override void OnLeftRoom()
    {
        Destroy(gameObject);
    }
}

```

PLAYER_MANAGER

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Photon.Pun;
using System.IO;

public class PlayerManager : MonoBehaviour
{
    PhotonView PV;
    Animator anim;
    GameObject controller;

    void Awake()
    {
        PV = GetComponent<PhotonView>();
        anim = GetComponent<Animator>();
    }

    void Start()
    {
        if(PV.IsMine)
        {
            CreateController();
        }
    }
}

```

```

void CreateController()
{
    Transform spawnpoint =
SpawnManager.Instance.GetSpawnpoint();
    controller =
PhotonNetwork.Instantiate(Path.Combine("PhotonPrefabs","PlayerCo
ntroller"),spawnpoint.position,spawnpoint.rotation,0,new
object[]){PV.ViewID});

}
}

```

PLAYER_NAME_MANAGER

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using Photon.Pun;

public class PlayerNameManager : MonoBehaviour
{
    [SerializeField]TMP_InputField usernameInput;

    public void OnUsernameInputValueChanged()
    {
        PhotonNetwork.NickName = usernameInput.text;
    }
}

```

ROOM_LIST_ITEM

```

using Photon.Realtime;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class RoomListItem : MonoBehaviour

```

```

{

    [SerializeField] TMP_Text text;
    public RoomInfo info;
    public void SetUp(RoomInfo _info)
    {
        info = _info;
        text.text = _info.Name;
    }

    public void OnClick()
    {
        Launcher.Instance.JoinRoom(info);
    }
}

```

ROOM_MANAGER

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Photon.Pun;
using Photon.Realtime;
using System.Linq;
using UnityEngine.SceneManagement;
using System.IO;

public class RoomManager : MonoBehaviourPunCallbacks
{

    public static RoomManager Instance;

    void Awake()
    {
        if(Instance)
        {
            Destroy(gameObject);
            return;
        }
        DontDestroyOnLoad(gameObject);
        Instance = this;
    }
}

```

```

public override void OnEnable()
{
    base.OnEnable();
    SceneManager.sceneLoaded += OnSceneLoaded;
}

public override void OnDisable()
{
    base.OnDisable();
    SceneManager.sceneLoaded -= OnSceneLoaded;
}

void OnSceneLoaded(Scene scene, LoadSceneMode loadSceneMode)
{
    if(scene.buildIndex == 1)
    {
        PhotonNetwork.Instantiate(Path.Combine("PhotonPrefabs", "PlayerManager"), Vector3.zero, Quaternion.identity);
    }
}
}

```

SCORE_MANAGER

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using Photon.Pun;
using TMPro;

public class ScoreManager : MonoBehaviour
{
    public static ScoreManager instance;

    private void Awake()
    {
        instance = this;
    }

    public static int p1Score = 0;
}

```

```

    public static int p2Score = 0;
    public TextMeshProUGUI p1ScoreText;
    public TextMeshProUGUI p2ScoreText;

    private void update()
    {
        ResetPositions();
        p1ScoreText.text = p1Score.ToString();
        p1ScoreText.text = p2Score.ToString();
    }

    public void ResetPositions()
    {
        PhotonNetwork.LoadLevel(1);
    }
}

```

SPAWN_MANAGER

```

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

public class SpawnManager : MonoBehaviour
{
    public static SpawnManager Instance;
    Spawnpoint[] spawnpoints;

    void Awake()
    {
        Instance = this;
        spawnpoints = GetComponentsInChildren<Spawnpoint>();
    }

    public Transform GetSpawnpoint()
    {
        return
spawnpoints[Random.Range(0,spawnpoints.Length)].transform;
    }
}

```

SPAWN_POINT

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

public class Spawnpoint : MonoBehaviour
{
    [SerializeField] GameObject graphics;

    void Awake()
    {
        graphics.SetActive(false);
    }
}
```

USERNAME_DISPLAY

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

public class UsernameDisplay : MonoBehaviour
{
    [SerializeField] PhotonView playerPV;
    [SerializeField] TMP_Text text;

    void Start()
    {
        text.text = playerPV.Owner.NickName;
    }
}
```