**********ASSIGNMENT 3**********

NAME: RAMAVATH SANTHOSH ROLL NO: 22MCF1R40

BRANCH: MCA(CSE) COURSE: CGDA

QUESTION: Create a game SUPER MARIO (simple)

a) Perform the operations only two levels (Level-1-→ EASY , Level-2 --→Moderate)
b) Create a toy and the shirt color should be in red.

CODE:

PLAYERSCRIPT: ***PlayerMovement.cs***

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

public class PlayerMovement : MonoBehaviour
{
    public float speed=6f;
    private AudioSource audioManager;
    private Rigidbody2D myBody;
    private Animator anim;

    public Transform groundCheckPosition;
    public LayerMask groundLayer;

    private bool isGrounded;
    private bool jumped;

    public float jumpPower= 12f;
    // Start is called before the first frame update
```

```
void Awake()
{
    myBody = GetComponent<Rigidbody2D>();
    anim = GetComponent<Animator>();
    audioManager = GetComponent<AudioSource> ();

}

void Start()
{

}



// Update is called once per frame
void Update()
{
    CheckIfGrounded();
    PlayerJump();

}
void FixedUpdate()
{
    PlayerWalk();

}
void PlayerWalk()
{
    float h= Input.GetAxisRaw("Horizontal");

    if(h>0){
        myBody.velocity = new Vector2(speed,myBody.velocity.y);
        ChangeDirection(1);
```

```
        }


        else if(h<0){
            myBody.velocity = new Vector2(-speed,myBody.velocity.y);
            ChangeDirection(-1);
        }
        else{
            myBody.velocity= new Vector2(0f,myBody.velocity.y);
        }

        anim.SetInteger("Speed",Mathf.Abs((int)myBody.velocity.x));

    }

    void ChangeDirection(int direction){
        Vector3 tempScale=transform.localScale;
        tempScale.x=direction;
        transform.localScale=tempScale;
    }
    void CheckIfGrounded(){
        isGrounded= Physics2D.Raycast
(groundCheckPosition.position,Vector2.down,0.1f,groundLayer);

        if(isGrounded){
            //we jumped before
            if(jumped){
                jumped=false;
                anim.SetBool("Jump",false);
            }
        }
    }
    void PlayerJump(){
        if(isGrounded){
            if(Input.GetKey (KeyCode.Space)){
```

```
            jumped =true;
            audioManager.Play();
            myBody.velocity = new Vector2(myBody.velocity.x,jumpPower );
            anim.SetBool ("Jump",true);
        }
    }
  }
}
```

MAIN MENU SCRIPT: *MainMenuController.cs*

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

public class MainMenuController : MonoBehaviour
{
    public void PlayGame(){
        SceneManager.LoadScene ("Gameplay");
    }
}
```

HELPER SCRIPT: *Mytags.cs*

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

public class MyTags : MonoBehaviour
```

```
{
   public static string PLAYER_TAG = "Player";
}
```

CAMERA SCRIPT: *CameraFollow.cs*

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

public class CameraFollow : MonoBehaviour
{

    public float resetSpeed= 0.5f;
    public float camerSpeed= 0.3f;
    public Bounds cameraBounds;
    private Transform target;
    private float offsetZ;
    private Vector3 lastTargetPosition;
    private Vector3 currentVelocity;
    private bool followsPlayer;

    void Awake()
    {
        BoxCollider2D myCol= GetComponent<BoxCollider2D>();
        myCol.size= new Vector2(Camera.main.aspect*2f*Camera.main.orthographicSize,15f);
        cameraBounds= myCol.bounds;
    }
    // Start is called before the first frame update
    void Start()
    {
        target= GameObject.FindGameObjectWithTag (MyTags.PLAYER_TAG).transform;
        lastTargetPosition =target.position;
        offsetZ=(transform.position - target.position).z;
        followsPlayer=true;
```

```
    }

    // Update is called once per frame
    void FixedUpdate()
    {
        if(followsPlayer){
            Vector3 aheadTargetPos =target.position+ Vector3.forward*offsetZ;

            if(aheadTargetPos.x >= transform.position.x){
                Vector3 newCameraPosition = Vector3.SmoothDamp(transform.position,aheadTargetPos,
                ref currentVelocity,camerSpeed);

                transform.position =new Vector3 (newCameraPosition.x,
transform.position.y,newCameraPosition.z);

                lastTargetPosition =target.position;
            }
        }

    }
}
```

BGSCRIPT: *ScaleBaground.cs*

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

public class ScaleBackground : MonoBehaviour
{
    // Start is called before the first frame update
```

```
    void Start()
    {
        SpriteRenderer sr= GetComponent<SpriteRenderer> ();
        transform.localScale= new Vector3 (1,1,1);

        float width = sr.sprite.bounds.size.x;
        float height = sr.sprite.bounds.size.y;

        float worldHeight = Camera.main.orthographicSize*2f;
        float worldWidth =  worldHeight/Screen.height*Screen.width;

        Vector3 tempScale = transform.localScale;
        tempScale.x = worldWidth/width+0.1f;
        tempScale.y = worldHeight/height+0.1f;

        transform.localScale=tempScale;

    }

}   //class
```