SOURCE CODES.

backManager.cs

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

public class backManager : MonoBehaviour {

    public AudioSource menuSound;

    public void GoBack(){
        Application.LoadLevel ("MainMenu");
    }

    // Use this for initialization
    void Start () {
        menuSound.Play ();
    }
}
```

GameManager.cs

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

public class GameManager : MonoBehaviour {

    public static GameManager instance;

    [SerializeField]
    private GameObject player;

    [SerializeField]
    private GameObject platform;

    private float minX = -2.24f, maxX = 2.24f, minY = -5.03f, maxY = -
3.79f;


    private float lerpTime = 1.5f;
    private bool lerpCamera;
    private float lerpX;
    void Awake(){
        makeInstance ();
        CreateInitialPlatform ();
    }

    void makeInstance(){
        if (instance == null)
            instance = this;
    }

    void Update(){
        if (lerpCamera) {
            lerpTheCamera ();
        }
    }


    public void CreateInitialPlatform(){
        Vector3 temp = new Vector3 (Random.Range (minX, minX + 1.2f),
Random.Range (minY, maxY), 0);
        Instantiate (platform, temp, Quaternion.identity);
        temp.y += 2f;
        Instantiate (player, temp, Quaternion.identity);

        temp = new Vector3 (Random.Range (maxX, maxX - 1.2f), Random.R
ange (minY, maxY), 0);

        Instantiate (platform, temp, Quaternion.identity);

    }

    public void lerpTheCamera(){

        float x = Camera.main.transform.position.x;
        x = Mathf.Lerp (x, lerpX, lerpTime * Time.deltaTime);
        Camera.main.transform.position = new Vector3 (x, Camera.main.t
ransform.position.y, Camera.main.transform.position.z);
        if (Camera.main.transform.position.x >= (lerpX - 0.07f)) {
            lerpCamera = false;
        }
    }

    public void CreateNewPlatformAndLerp(float lerpPosition){

        CreateNewPlatform ();
        lerpX = lerpPosition + maxX;
        lerpCamera = true;
    }

    void CreateNewPlatform(){

        float cameraX = Camera.main.transform.position.x;
        float newMaxX = (maxX * 2) + cameraX;
        Instantiate (platform, new Vector3 (Random.Range (newMaxX, new
MaxX - 1.2f), Random.Range (minY, maxY), 0), Quaternion.identity);
    }



}
```

GameOverManager.cs

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

public class GameOverManager : MonoBehaviour {

    public static GameOverManager instance;
    private GameObject gameOverPanel;
    private Animator gameOverAnim;
    private Button retryBtn,MenuBtn;
    private Text score;

    private GameObject scoreText;
    public Text highScore;

    void Awake(){
        makeInstance ();
        InitializeVariable ();
    }

    public void makeInstance(){
        if (instance == null)
            instance = this;
    }

    public void showGameOverPanel(){
        scoreText.SetActive (false);
        gameOverPanel.SetActive (true);
        ScoreManager.instance.GetHighScore ();
        score.text = "Score:\n" + "" + ScoreManager.instance.GetScore
();
        highScore.text ="High Score:\n" +  PlayerPrefs.GetInt ("highSc
ore").ToString ();
        LeaderBoardManager.instance.AddScoreToLeaderBoard ();
        gameOverAnim.Play("GameOverPanel fadeIn");
        AdManager.instance.ShowAd ();

    }

    void InitializeVariable(){
        gameOverPanel = GameObject.Find ("GameOverPanel Holder");
        gameOverAnim = gameOverPanel.GetComponent<Animator> ();
        retryBtn = GameObject.Find ("RetryButton").GetComponent<Button
>();
        MenuBtn = GameObject.Find ("MenuButton").GetComponent<Button>
();

        retryBtn.onClick.AddListener (() => PlayAgain ());
        MenuBtn.onClick.AddListener (() => Menu ());

        scoreText = GameObject.Find ("ScoreText");
        score = GameObject.Find ("Text").GetComponent<Text> ();

        gameOverPanel.SetActive (false);
    }

    public void PlayAgain(){
        Application.LoadLevel (Application.loadedLevelName);
    }

    public void Menu(){
        Application.LoadLevel ("MainMenu");
    }
}
```

jumpButton.cs

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

public class jumpButton : MonoBehaviour,IPointerDownHandler,IPointerUpHandler {


    public void OnPointerDown(PointerEventData data){
        if (playerJumpScript.instance != null) {
            playerJumpScript.instance.SetPower (true);
        }
    }

    public void OnPointerUp(PointerEventData data){
        if (playerJumpScript.instance != null) {
            playerJumpScript.instance.SetPower (false);
        }


    }

}
```

MenuManager.cs

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

public class MenuManager : MonoBehaviour {

    public AudioSource menuSound;

    public void PlayGame(){
        Application.LoadLevel ("Gameplay");
    }
    // Use this for initialization
    void Start () {
        menuSound.Play ();
    }

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

    }
    public void Help(){
        Application.LoadLevel ("howTo");
    }

//    public void showLeaderBoard(){
//        LeaderBoardManager.instance.ShowLeaderBoard ();
//    }
}
```

playerJumpScript.cs

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

public class playerJumpScript : MonoBehaviour {

    public static playerJumpScript instance;
    private Rigidbody2D rb;
    private Animator anim;



     public AudioSource landsound,deadSound;

    [SerializeField]
    private float forceX,forceY;
    private bool didJump,setPower;
    private float tresholdX = 7f;
    private float tresholdY = 14f;

    private Slider powerBar;
    private float powerBarTreshold = 10f;
    private float powerBarValue = 0f;

    void  Awake(){
        makeInstance ();
    }

    void Update(){
        SetPower ();
        Initialize ();
    }
    void Initialize(){
        powerBar = GameObject.Find ("PowerBar").GetComponent<Slider> (
);
        rb = GetComponent<Rigidbody2D> ();
        anim = GetComponent<Animator> ();
        powerBar.minValue = 0f;
        powerBar.maxValue = 10f;
        powerBar.value = powerBarValue;

    }

    public void makeInstance(){
        if (instance == null)
            instance = this;
    }

    void SetPower(){
        if (setPower && !didJump) {
            forceX += tresholdX * Time.deltaTime;
            forceY += tresholdY * Time.deltaTime;


            if (forceX < 3f ) {
                forceX = 3f;
            }


            if (forceX > 6.5f ) {
                forceX = 6.5f;
            }

            if (forceY < 3f) {

                forceY = 3f;
            }


            if (forceY > 13.5f) {

                forceY = 13.5f;
            }

            powerBarValue += powerBarTreshold * Time.deltaTime;
            powerBar.value = powerBarValue;
        }
    }
    public void SetPower(bool setPower){
        this.setPower = setPower;

        if (!setPower && !didJump) {
            jump ();
        }

    }

    void jump(){
        rb.velocity = new Vector2 (forceX, forceY);
        forceX = forceY = 0;
        didJump = true;
        anim.SetBool ("jump", didJump);

        powerBarValue = 0f;
        powerBar.value = powerBarValue;
    }

    void OnTriggerEnter2D(Collider2D target){
        if (didJump) {
            didJump = false;
            anim.SetBool ("jump", didJump);

            if (target.tag == "platform") {
                if (GameManager.instance != null) {
                    GameManager.instance.CreateNewPlatformAndLerp (tar
get.transform.position.x);
                    landsound.Play();


                }

                if (ScoreManager.instance != null) {
                    ScoreManager.instance.IncrementScore ();
                    AchievementManager.instance.CheckAchievements ();
                }

            }


        }

        if (target.tag == "dead") {
```

```
                    if (GameOverManager.instance != null) {
                        GameOverManager.instance.showGameOverPanel ();
            }
            Destroy (gameObject);

        }

    }
}
```

ScoreManager.cs

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

public class playerJumpScript : MonoBehaviour {

    public static playerJumpScript instance;
    private Rigidbody2D rb;
    private Animator anim;



     public AudioSource landsound,deadSound;

    [SerializeField]
    private float forceX,forceY;
    private bool didJump,setPower;
    private float tresholdX = 7f;
    private float tresholdY = 14f;

    private Slider powerBar;
    private float powerBarTreshold = 10f;
    private float powerBarValue = 0f;

    void  Awake(){
        makeInstance ();
    }

    void Update(){
        SetPower ();
        Initialize ();
    }
    void Initialize(){
        powerBar = GameObject.Find ("PowerBar").GetComponent<Slider> (
);
        rb = GetComponent<Rigidbody2D> ();
        anim = GetComponent<Animator> ();
        powerBar.minValue = 0f;
        powerBar.maxValue = 10f;
        powerBar.value = powerBarValue;

    }

    public void makeInstance(){
        if (instance == null)
            instance = this;
    }

    void SetPower(){
        if (setPower && !didJump) {
            forceX += tresholdX * Time.deltaTime;
            forceY += tresholdY * Time.deltaTime;


            if (forceX < 3f ) {
                forceX = 3f;
            }


            if (forceX > 6.5f ) {
                forceX = 6.5f;
            }

            if (forceY < 3f) {

                forceY = 3f;
            }


            if (forceY > 13.5f) {

                forceY = 13.5f;
            }

            powerBarValue += powerBarTreshold * Time.deltaTime;
            powerBar.value = powerBarValue;
        }
    }

    public void SetPower(bool setPower){
        this.setPower = setPower;

        if (!setPower && !didJump) {
            jump ();
        }

    }

    void jump(){
        rb.velocity = new Vector2 (forceX, forceY);
        forceX = forceY = 0;
        didJump = true;
        anim.SetBool ("jump", didJump);

        powerBarValue = 0f;
        powerBar.value = powerBarValue;
    }

    void OnTriggerEnter2D(Collider2D target){
        if (didJump) {
            didJump = false;
            anim.SetBool ("jump", didJump);

            if (target.tag == "platform") {
                if (GameManager.instance != null) {
                    GameManager.instance.CreateNewPlatformAndLerp (tar
get.transform.position.x);
                    landsound.Play();


                }

                if (ScoreManager.instance != null) {
                    ScoreManager.instance.IncrementScore ();
                    AchievementManager.instance.CheckAchievements ();
                }

            }


        }

        if (target.tag == "dead") {
```

```
                    if (GameOverManager.instance != null) {
                        GameOverManager.instance.showGameOverPanel ();
            }
            Destroy (gameObject);

        }

    }
}
```