1. **BulletScript**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BulletScript : MonoBehaviour {

public float velX = 5f;

float velY = 0f;

Rigidbody2D rb;

//void OnCollisionEnter(Collision collision)

//{

// GameObject.Destroy(gameObject);

//}

// public void OnCollisionEnter(Collision node)

//{

// if (node.gameObject.tag == "Asteroid")

// {

// Destroy(node.gameObject);

// }

// }

/\* void OnTriggerEnter(Collider col)

{

Destroy(col.gameObject);

}

void OnTriggerEnter()

{

Destroy(gameObject);

} \*/

/\*void OnTriggerEnter(Collider other)

{

Destroy(gameObject);

}\*/

void OnTriggerEnter2D(Collider2D other)

{

if (other.gameObject.tag == "Asteroid")

{

Destroy(gameObject);

}

else if (other.gameObject.tag == "AI\_ship")

{

Destroy(gameObject);

}

}

// Use this for initialization

void Start () {

rb = GetComponent<Rigidbody2D>();

}

// Update is called once per frame

void Update()

{

rb.velocity = new Vector2(velX, velY);

Destroy(gameObject, 5f);

}

}

1. **SpawnRoids**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SpawnRoids : MonoBehaviour {

public Transform[] SpawnPoints;

public float spawnTime = 1.5f;

public GameObject Asteroids;

public GameObject AIShip;

int flag = 0;

// Use this for initialization

void Start () {

InvokeRepeating("Spawning", spawnTime, spawnTime);

}

// Update is called once per frame

void Update () {

/\*

if (AIShip.activeSelf)

{

CancelInvoke("Spawning");

flag = 0;

}

else if(!AIShip.activeSelf)

{

if(flag == 0)

{

InvokeRepeating("Spawning", spawnTime, spawnTime);

}

flag = 1;

}

\*/

}

void Spawning()

{

int spawnIndex = Random.Range(0, SpawnPoints.Length);

Instantiate(Asteroids, SpawnPoints[spawnIndex].position, SpawnPoints[spawnIndex].rotation);

}

}

1. **MoveScript**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityStandardAssets.CrossPlatformInput;

using UnityEngine.SceneManagement;

public class MoveScript : MonoBehaviour {

float directionY;

Rigidbody2D rb;

public GameObject bulletToRight;

Vector2 bulletPos;

public float fireRate = 0.5f;

float nextFire = 0.0f;

void OnTriggerEnter2D(Collider2D other)

{

//if (other.gameObject.tag == "Asteroid")

// {

if (other.gameObject.tag == "AI\_bullet" || other.gameObject.tag == "Asteroid")

{

//Destroy(gameObject);

//SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

healthbar.health -= 10f;

if (healthbar.health == 0)

{

Destroy(gameObject);

SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

}

}

}

void Start()

{

// If you want the min max values to update if the resolution changes

// set them in update else set them in Start

rb = GetComponent<Rigidbody2D>();

}

void Update()

{

directionY = CrossPlatformInputManager.GetAxis("Vertical");

rb.velocity = new Vector2(0, directionY \* 100);

//CrossPlatformInputManager.GetButtonDown("Fire1");

//if (Input.GetButtonDown("Fire1")&& Time.time > nextFire)

//CrossPlatformInputManager.GetButtonDown("Shoot1");

if (CrossPlatformInputManager.GetButtonDown("Shoot1") && Time.time > nextFire)

{

nextFire = Time.time + fireRate;

fire();

}

}

void fire()

{

bulletPos = transform.position;

bulletPos += new Vector2(+1f, -0f);

Instantiate(bulletToRight, bulletPos, Quaternion.identity);

}

}

1. **Score**

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class Score : MonoBehaviour

{

public Transform player;

public Text scoreText;

private int now = 3;

public int score = 0;

public int highScore = 0;

string highScoreKey = "HighScore";

void Start()

{

//scoreLabel.text = score.ToString();

scoreText.text = score.ToString();

//get high score if it is there, otherwise 0

highScore = PlayerPrefs.GetInt(highScoreKey, 0);

}

void Update()

{

InvokeRepeating("AdToScore", 0, 10);

}

void AdToScore()

{

if (now > 0)

{

score = score + 1;

scoreText.text = score.ToString();

}

}

void OnDisable()

{

//if score greater than high score, set new high score and save

if (score > highScore)

{

PlayerPrefs.SetInt(highScoreKey, score);

PlayerPrefs.Save();

}

}

}