Unity scripts

* Audio 1,2
* Bullet 1,2,3
* Level 1,2
* Player 1,2,3
* Enemy 1,2,3,4,5,6,7,8

Audio

1,Audio Manager

using UnityEngine.Audio;

using System;

using UnityEngine;

public class AudioManager : MonoBehaviour

{

    public Sound[] sounds;

    void Awake(){

        foreach(Sound s in sounds)

        {

            s.source = gameObject.AddComponent<AudioSource>();

            s.source.clip = s.clip;

            s.source.volume = s.volume;

            s.source.pitch = s.pitch;

        }

    }

    public void Play(string name)

    {

        Sound s = Array.Find(sounds, sound => sound.name == name);

        if(s == null){

            Debug.LogWarning("That sound " + name + " is not found");

            return;

        }

        s.source.Play();

    }

}

2,Sound

using UnityEngine.Audio;

using UnityEngine;

[System.Serializable]

public class Sound

{

    public string name;

    public AudioClip clip;

    [Range(0f,1f)]

    public float volume;

    [Range(0.1f,3f)]

    public float pitch;

    [HideInInspector]

    public AudioSource source;

}

Bullets

1,Player Bullet

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Bullet : MonoBehaviour

{

    public float speed = 10f;

    public Rigidbody2D rb;

    public int damage = 1;

    public float lifeTime;

    public GameObject deathEffect;

    void Start()

    {

        Invoke("DestroyProjectile", lifeTime);

    }

    void Update()

    {

        rb.velocity = transform.right \* speed;

        //transform.Translate(transform.right \* speed \* Time.deltaTime);

    }

    void OnTriggerEnter2D(Collider2D hitInfo)

    {

        Enemy enemy = hitInfo.GetComponent<Enemy>();

        Enemy2 enemy2 = hitInfo.GetComponent<Enemy2>();

        Enemy3 enemy3 = hitInfo.GetComponent<Enemy3>();

        Enemy4 enemy4 = hitInfo.GetComponent<Enemy4>();

        EnemyAi enemy5 = hitInfo.GetComponent<EnemyAi>();

        EnemyPatrol enemy6 = hitInfo.GetComponent<EnemyPatrol>();

        Enemy6 enemy7 = hitInfo.GetComponent<Enemy6>();

        if(enemy != null)

        {

            enemy.TakeDamage(damage);

        }

        if(enemy2 != null)

        {

            enemy2.TakeDamage(damage);

        }

        if(enemy3 != null)

        {

            enemy3.TakeDamage(damage);

        }

        if(enemy4 != null)

        {

            enemy4.TakeDamage(damage);

        }

        if(enemy5 != null)

        {

            enemy5.TakeDamage(damage);

        }

        if(enemy6 != null)

        {

            enemy6.TakeDamage(damage);

        }

        if(enemy7 != null)

        {

            enemy7.TakeDamage(damage);

        }

        if (!hitInfo.CompareTag("Player"))

        {

            DestroyProjectile();

        }

    }

    void DestroyProjectile()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        FindObjectOfType<AudioManager>().Play("BulletDestroyed");

        Destroy(gameObject);

    }

    }

2,Enemy Bullet

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Bullet1 : MonoBehaviour

{

    public float speed = 10f;

    //public Rigidbody2D rb;

    public int damage = 1;

    public float lifeTime;

    public Transform player;

    private Vector2 target;

    public GameObject deathEffect;

    void Start()

    {

        player = GameObject.FindGameObjectWithTag("Player").transform;

        target  = new Vector2(player.position.x,player.position.y);

        Invoke("DestroyProjectile", lifeTime);

    }

    void Update()

    {

        transform.position =  Vector2.MoveTowards(transform.position ,target,speed\*Time.deltaTime);

        if(transform.position.x == target.x){

            DestroyProjectile();

        }

    }

        void OnTriggerEnter2D(Collider2D other){

        if(other.CompareTag("Player")){

            DestroyProjectile();

        }

        if (!other.CompareTag("Enemy"))

        {

            if(!other.CompareTag("GroundEnemy"))

            {

                if(!other.CompareTag("Bullet"))

                {

                    DestroyProjectile();

                }

            }

        }

        }

    void DestroyProjectile()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    }

3,Enemy Bullet

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Bullet2 : MonoBehaviour

{

    public float speed = 10f;

    //public Rigidbody2D rb;

    public int damage = 1;

    public float lifeTime;

    public Transform player;

    private Vector2 target;

    public GameObject deathEffect;

    void Start()

    {

        player = GameObject.FindGameObjectWithTag("Player").transform;

        target  = new Vector2(player.position.x,player.position.y);

        Invoke("DestroyProjectile", lifeTime);

    }

    void Update()

    {

        transform.position =  Vector2.MoveTowards(transform.position ,target,speed\*Time.deltaTime);

        if(transform.position.x == target.x){

            DestroyProjectile();

        }

    }

        void OnTriggerEnter2D(Collider2D other){

        if(other.CompareTag("Player")){

            DestroyProjectile();

        }

        if (!other.CompareTag("Enemy"))

        {

            if(!other.CompareTag("GroundEnemy"))

            {

            DestroyProjectile();

            }

        }

        }

    void DestroyProjectile()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    }

Level Scripts

1,level ending

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class EndingLevel : MonoBehaviour

{

    public int Respawn;

    void OnTriggerEnter2D(Collider2D other)

    {

        if (other.CompareTag("Player"))

        {

            SceneManager.LoadScene(Respawn);

        }

    }

}

2, Level generaion random

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class LevelGeneration : MonoBehaviour

{

    public GameObject[] objects;

    void Start()

    {

        int rand = Random.Range(0 , objects.Length);

        Instantiate(objects[rand], transform.position ,Quaternion.identity);

    }

}

Player Scripts

1,Kill Player

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class KillPlayer : MonoBehaviour

{

    public int Respawn;

    // Start is called before the first frame update

    void Start()

    {

    }

    // Update is called once per frame

    void Update()

    {

    }

    void OnTriggerEnter2D(Collider2D other)

    {

      if (other.CompareTag("Player"))

        {

            SceneManager.LoadScene(Respawn);

        }

    }

}

2, Move Player

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class MovingPlayer : MonoBehaviour

{

    public float speed;

    public float jump;

    private float move;

    private Rigidbody2D rb;

    private bool isJumping ;

    // Dashing

    private bool canDash = true;

    private bool facingRight;

    private bool isDashing;

    public float dashingPower = 24f;

    public float dashingTime = 0.2f;

    private float dasingCooldown = 1f;

    [SerializeField] private TrailRenderer tr;

    private float coyoteTime = 0.2f;

    private float coyoteTimeCounter;

    private float JumpBufferTime = 0.2f;

    private float JumpBufferTimeCounter;

    public float fallDistance = -50f;

    public int Respawn;

    public Transform player;

    public GameObject dashEffect;

    public ParticleSystem Jumpdust;

    public ParticleSystem Movedust;

    void Start()

    {

        rb = GetComponent<Rigidbody2D>();

        player = GameObject.FindGameObjectWithTag("Player").transform;

    }

    // Update is called once per frame

    void Update()

    {

        if(!isJumping)

        {

            coyoteTimeCounter = coyoteTime;

        }

        else

        {

            coyoteTimeCounter -= Time.deltaTime;

        }

        if(player.position.y <= fallDistance)

        {

            SceneManager.LoadScene(Respawn);

        }

        if(Input.GetButtonDown("Jump"))

        {

            JumpBufferTimeCounter = JumpBufferTime;

        }

        else

        {

            JumpBufferTimeCounter -= Time.deltaTime;

        }

        if(isDashing){

            return;

        }

        move = Input.GetAxisRaw("Horizontal");

        if(move < 0 ){

            facingRight = false;

            if(!isJumping)

            {

                CreateDust2();

            }

        }

        if(move > 0 ){

            facingRight = true;

            if(!isJumping)

            {

                CreateDust2();

            }

        }

        rb.velocity = new Vector2(move \* speed, rb.velocity.y);

        if (JumpBufferTimeCounter > 0f && coyoteTimeCounter > 0f)

        {

            FindObjectOfType<AudioManager>().Play("JumpSound");

            rb.AddForce(new Vector2(rb.velocity.x, jump));

            JumpBufferTimeCounter = 0f;

            isJumping = true;

            CreateDust();

        }

        if(Input.GetButtonDown("Jump") && rb.velocity.y > 0f)

        {

            rb.velocity = new Vector2( rb.velocity.x , rb.velocity.y \* 0.5f);

            coyoteTimeCounter = 0f;

        }

        if (Input.GetKeyDown(KeyCode.LeftShift) && canDash){

            StartCoroutine(Dash());

        }

    }

    private void FixedUpdate(){

        if(isDashing){

            return;

        }

    }

    void OnCollisionEnter2D(Collision2D other)

    {

        if (other.gameObject.CompareTag("Ground"))

        {

            isJumping = false;

        }

        if(other.gameObject.CompareTag("GroundEnemy"))

        {

            isJumping = false;

        }

    }

    private IEnumerator Dash(){

        canDash = false;

        isDashing = true;

        float originalGravity = rb.gravityScale;

        rb.gravityScale = 0f;

        if(facingRight){

            FindObjectOfType<AudioManager>().Play("DashSound");

            Instantiate(dashEffect, transform.position, Quaternion.identity);

            rb.velocity = new Vector2(transform.localScale.x \* dashingPower, 0f);

        }

        if(!facingRight){

            FindObjectOfType<AudioManager>().Play("DashSound");

            Instantiate(dashEffect, transform.position, Quaternion.identity);

            rb.velocity = new Vector2(transform.localScale.x \* -dashingPower, 0f);

        }

        tr.emitting = true;

        yield return new WaitForSeconds(dashingTime);

        tr.emitting = false;

        rb.gravityScale = originalGravity;

        isDashing = false;

        yield return new WaitForSeconds(dasingCooldown);

        canDash = true;

    }

    void CreateDust(){

        Jumpdust.Play();

    }

    void CreateDust2(){

        Movedust.Play();

    }

}

3, Player weapon

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Weapon : MonoBehaviour

{

    public Transform firePoint;

    public GameObject bulletPrefab;

    public float offset;

    private float timeBtwShots;

    public float startTimBtwShots;

    public GameObject BulletEffect;

    void Update()

    {

        Vector3 difference = Camera.main.ScreenToWorldPoint(Input.mousePosition) - transform.position;

        float rotZ = Mathf.Atan2(difference.y, difference.x) \* Mathf.Rad2Deg;

        transform.rotation = Quaternion.Euler(0f, 0f, rotZ + offset);

        if (timeBtwShots <= 0)

        {

            if (Input.GetButtonDown("Fire1"))

            {

                Shoot();

            }

        }

        else

        {

            timeBtwShots -= Time.deltaTime;

        }

    }

    void Shoot()

    {

        Instantiate(BulletEffect, firePoint.position, Quaternion.identity);

        Instantiate(bulletPrefab, firePoint.position, transform.rotation);

        timeBtwShots = startTimBtwShots;

        //Instantiate(bulletPrefab, firePoint.position, firePoint.rotation);

        //Debug.Log("Hello");

    }

}

Enemy Scripts

1

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Enemy : MonoBehaviour

{

    public float speed;

    public float AgroRange;

    private Rigidbody2D rb2d;

    public Transform player;

    void Start()

    {

        player = GameObject.FindGameObjectWithTag("Player").transform;

        rb2d = GetComponent<Rigidbody2D>();

    }

    void Update()

    {

        float distToPlayer = Vector2.Distance(transform.position, player.position);

        if(distToPlayer < AgroRange)

        {

            ChasePlayer();

        }

        else

        {

            rb2d.velocity = new Vector2(0,rb2d.velocity.y);

        }

    }

    public int health = 6;

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    public GameObject deathEffect;

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    void ChasePlayer()

    {

        if (transform.position.x < player.position.x)

        {

            rb2d.velocity = new Vector2(speed, rb2d.velocity.y);

        }

        else if (transform.position.x > player.position.x)

        {

            rb2d.velocity = new Vector2(-speed, rb2d.velocity.y);

        }

    }

}

2,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Enemy2 : MonoBehaviour

{

    public float speed;

    public float AgroRange;

    public float retreatDistance;

    private Rigidbody2D rb2d;

    private float timeBtwShots;

    public float startTimeBtwShots;

    public Transform player;

    public GameObject projectile;

    void Start()

    {

        player = GameObject.FindGameObjectWithTag("Player").transform;

        rb2d = GetComponent<Rigidbody2D>();

        timeBtwShots = startTimeBtwShots;

    }

    void FixedUpdate()

    {

        float distToPlayer = Vector2.Distance(transform.position, player.position);

            if(distToPlayer < AgroRange)

        {

            if(distToPlayer > retreatDistance){

            ChasePlayer();

            }

            else

            {

            rb2d.velocity = new Vector2(0, rb2d.velocity.y);

            }

        }

        else

        {

            rb2d.velocity = new Vector2(0, rb2d.velocity.y);

        }

    if(timeBtwShots <= 0){

        Instantiate(projectile,transform.position,Quaternion.identity);

        timeBtwShots = startTimeBtwShots;

    }

    else{

        timeBtwShots -= Time.deltaTime;

    }

    }

    public int health = 6;

    public GameObject deathEffect;

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    void ChasePlayer()

    {

        if (transform.position.x < player.position.x)

        {

            rb2d.velocity = new Vector2(speed, rb2d.velocity.y);

        }

        else if (transform.position.x > player.position.x)

        {

            rb2d.velocity = new Vector2(-speed, rb2d.velocity.y);

        }

    }

}

3,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Enemy3 : MonoBehaviour

{

    public float speed;

    public float AgroRange;

    public float retreatDistance;

    private Rigidbody2D rb2d;

    private float timeBtwShots;

    public float startTimeBtwShots;

    public Transform player;

    public GameObject projectile;

    void Start()

    {

        player = GameObject.FindGameObjectWithTag("Player").transform;

        rb2d = GetComponent<Rigidbody2D>();

        timeBtwShots = startTimeBtwShots;

    }

    void FixedUpdate()

    {

        float distToPlayer = Vector2.Distance(transform.position, player.position);

            if(distToPlayer < AgroRange)

        {

            if(distToPlayer > retreatDistance){

            ChasePlayer();

            }

            else

            {

            rb2d.velocity = new Vector2(0,rb2d.velocity.y);

            }

        }

        else

        {

            rb2d.velocity = new Vector2(0, rb2d.velocity.y);

        }

    if(timeBtwShots <= 0){

        Instantiate(projectile,transform.position,Quaternion.identity);

        timeBtwShots = startTimeBtwShots;

    }

    else{

        timeBtwShots -= Time.deltaTime;

    }

    }

    public int health = 6;

    public GameObject deathEffect;

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    void ChasePlayer()

    {

        if (transform.position.x < player.position.x)

        {

            rb2d.velocity = new Vector2(speed, rb2d.velocity.y);

        }

        else if (transform.position.x > player.position.x)

        {

            rb2d.velocity = new Vector2(-speed, rb2d.velocity.y);

        }

    }

}

4,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Enemy4 : MonoBehaviour

{

    public Transform player;

    public float AgroRange;

    public float speed;

    private Rigidbody2D rb2d;

    //private bool FollowPlayer = true;

    // For dashing

    private bool canDash = true;

    private bool facingRight;

    private bool isDashing;

    public float dashingPower = 24f;

    public float dashingTime = 0.2f;

    public float dasingCooldown = 1f;

    [SerializeField] private TrailRenderer tr;

    public int health = 6;

    public GameObject deathEffect;

    void Start()

    {

        rb2d = GetComponent<Rigidbody2D>();

    }

    void Update()

    {

        //distance to player

        float distToPlayer = Vector2.Distance(transform.position,player.position);

        if(distToPlayer < AgroRange){

            // code to chase player

            ChasePlayer();

            if(canDash){

            StartCoroutine(Dash());

            }

        }

        else

        {

            StopChasePlayer();

        }

            if(isDashing){

            return;

    }

    }

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void FixedUpdate(){

            if(isDashing){

            return;

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    void ChasePlayer(){

        //Enemy left of palyer , To move right

        if(transform.position.x < player.position.x)

        {

            //rb2d.velocity = new Vector2(speed,0);

            rb2d.velocity = new Vector2(speed, rb2d.velocity.y);

            facingRight = true;

        }

        else

        {

        //Enemy right of player, to move left

            //rb2d.velocity = new Vector2(-speed,0);

            rb2d.velocity = new Vector2(-speed, rb2d.velocity.y);

            facingRight = false;

        }

    }

    void StopChasePlayer()

    {

        rb2d.velocity = new Vector2(0,rb2d.velocity.y);

    }

    private IEnumerator Dash(){

        canDash = false;

        isDashing = true;

        float originalGravity = rb2d.gravityScale;

        rb2d.gravityScale = 0f;

        if(facingRight){

            //rb2d.velocity = new Vector2(transform.localScale.x \* dashingPower, 0f);

            rb2d.AddForce(new Vector2(rb2d.velocity.x, 400));

        }

        if(!facingRight){

           // rb2d.velocity = new Vector2(transform.localScale.x \* -dashingPower, 0f);

            rb2d.AddForce(new Vector2(rb2d.velocity.x, 400));

        }

        tr.emitting = true;

        yield return new WaitForSeconds(dashingTime);

        tr.emitting = false;

        rb2d.gravityScale = originalGravity;

        isDashing = false;

        yield return new WaitForSeconds(dasingCooldown);

        canDash = true;

    }

}

5,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyPatrol : MonoBehaviour

{

    public float speed;

    private float distance = 2f;

    private bool movingRight = true;

    public Transform GroundDetection;

    public int health = 6;

    public GameObject deathEffect;

    void Update(){

        transform.Translate(Vector2.right \* speed \* Time.deltaTime);

        RaycastHit2D groundInfo = Physics2D.Raycast(GroundDetection.position, Vector2.down, distance);

        if(groundInfo.collider == false){

            if(movingRight == true){

                transform.eulerAngles = new Vector3(0,-180,0);

                movingRight = false;

            }

            else{

                transform.eulerAngles = new Vector3(0,0,0);

                movingRight = true;

            }

        }

    }

    void OnCollisionEnter2D(Collision2D other){

        if(other.gameObject.CompareTag("Obstacle") || other.gameObject.CompareTag("Enemy")){

            if(movingRight == true){

                transform.eulerAngles = new Vector3(0,-180,0);

                movingRight = false;

            }

            else{

                transform.eulerAngles = new Vector3(0,0,0);

                movingRight = true;

                }

        }

    }

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

}

6,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Enemy6 : MonoBehaviour

{

    public float speed;

    public float AgroRange;

    public float retreatDistance;

    private Rigidbody2D rb2d;

    private float timeBtwShots;

    public float startTimeBtwShots;

    public Transform player;

    private bool movingRight = true;

    public Transform SpawnPosition;

    public GameObject projectile;

    public GameObject projectile2;

    public int NoOfEnemy ;

    void Start()

    {

        player = GameObject.FindGameObjectWithTag("Player").transform;

        rb2d = GetComponent<Rigidbody2D>();

        timeBtwShots = startTimeBtwShots;

    }

    void FixedUpdate()

    {

        float distToPlayer = Vector2.Distance(transform.position, player.position);

            if(distToPlayer < AgroRange)

        {

            if(distToPlayer > retreatDistance){

            ChasePlayer();

            }

            else

            {

            rb2d.velocity = new Vector2(0, rb2d.velocity.y);

            }

        }

        else

        {

            rb2d.velocity = new Vector2(0, rb2d.velocity.y);

        }

    if(timeBtwShots <= 0){

        //rb2d.velocity = new Vector2( rb2d.velocity.y, 10f);

        if(NoOfEnemy < 6){

            if(movingRight == true){

            Instantiate(deathEffect, SpawnPosition.position, Quaternion.identity);

            Instantiate(projectile,SpawnPosition.position,Quaternion.identity);

            NoOfEnemy ++;

            }

            else

            {

            Instantiate(deathEffect, -SpawnPosition.position, Quaternion.identity);

            Instantiate(projectile,-SpawnPosition.position,Quaternion.identity);

            NoOfEnemy ++;

            }

            }

        else{

            Instantiate(projectile2,transform.position,Quaternion.identity);

            }

        timeBtwShots = startTimeBtwShots;

    }

    else{

        timeBtwShots -= Time.deltaTime;

    }

    }

    public int health = 10;

    public GameObject deathEffect;

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    void ChasePlayer()

    {

        if (transform.position.x < player.position.x)

        {

            rb2d.velocity = new Vector2(speed, rb2d.velocity.y);

            movingRight = false;

        }

        else if (transform.position.x > player.position.x)

        {

            rb2d.velocity = new Vector2(-speed, rb2d.velocity.y);

            movingRight = true;

        }

    }

}

7,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyAi : MonoBehaviour

{

    public Transform player;

    public float AgroRange;

    public float speed;

    private Rigidbody2D rb2d;

    //private bool FollowPlayer = true;

    // For dashing

    private bool canDash = true;

    private bool facingRight;

    private bool isDashing;

    public float dashingPower = 24f;

    public float dashingTime = 0.2f;

    public float dasingCooldown = 1f;

    [SerializeField] private TrailRenderer tr;

    public int health = 6;

    public GameObject deathEffect;

    void Start()

    {

        rb2d = GetComponent<Rigidbody2D>();

    }

    void Update()

    {

        //distance to player

        float distToPlayer = Vector2.Distance(transform.position,player.position);

        if(distToPlayer < AgroRange){

            // code to chase player

            ChasePlayer();

            if(canDash){

            StartCoroutine(Dash());

            }

        }

        else

        {

            StopChasePlayer();

        }

            if(isDashing){

            return;

    }

    }

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void FixedUpdate(){

            if(isDashing){

            return;

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

    void ChasePlayer(){

        //Enemy left of palyer , To move right

        if(transform.position.x < player.position.x)

        {

            rb2d.velocity = new Vector2(speed,rb2d.velocity.y);

            facingRight = true;

        }

        else

        {

        //Enemy right of player, to move left

            rb2d.velocity = new Vector2(-speed,rb2d.velocity.y);

            facingRight = false;

        }

    }

    void StopChasePlayer()

    {

        rb2d.velocity = new Vector2(0,rb2d.velocity.y);

    }

    private IEnumerator Dash(){

        canDash = false;

        isDashing = true;

        float originalGravity = rb2d.gravityScale;

        rb2d.gravityScale = 0f;

        if(facingRight){

            rb2d.velocity = new Vector2(transform.localScale.x \* dashingPower, 0f);

        }

        if(!facingRight){

            rb2d.velocity = new Vector2(transform.localScale.x \* -dashingPower, 0f);

        }

        tr.emitting = true;

        yield return new WaitForSeconds(dashingTime);

        tr.emitting = false;

        rb2d.gravityScale = originalGravity;

        isDashing = false;

        yield return new WaitForSeconds(dasingCooldown);

        canDash = true;

    }

}

8,

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyPatrol : MonoBehaviour

{

    public float speed;

    private float distance = 2f;

    private bool movingRight = true;

    public Transform GroundDetection;

    public int health = 6;

    public GameObject deathEffect;

    void Update(){

        transform.Translate(Vector2.right \* speed \* Time.deltaTime);

        RaycastHit2D groundInfo = Physics2D.Raycast(GroundDetection.position, Vector2.down, distance);

        if(groundInfo.collider == false){

            if(movingRight == true){

                transform.eulerAngles = new Vector3(0,-180,0);

                movingRight = false;

            }

            else{

                transform.eulerAngles = new Vector3(0,0,0);

                movingRight = true;

            }

        }

    }

    void OnCollisionEnter2D(Collision2D other){

        if(other.gameObject.CompareTag("Obstacle") || other.gameObject.CompareTag("Enemy")){

            if(movingRight == true){

                transform.eulerAngles = new Vector3(0,-180,0);

                movingRight = false;

            }

            else{

                transform.eulerAngles = new Vector3(0,0,0);

                movingRight = true;

                }

        }

    }

    public void TakeDamage (int damage)

    {

        health -= damage;

        if(health <= 0)

        {

            Die();

        }

    }

    void Die()

    {

        Instantiate(deathEffect, transform.position, Quaternion.identity);

        Destroy(gameObject);

    }

}