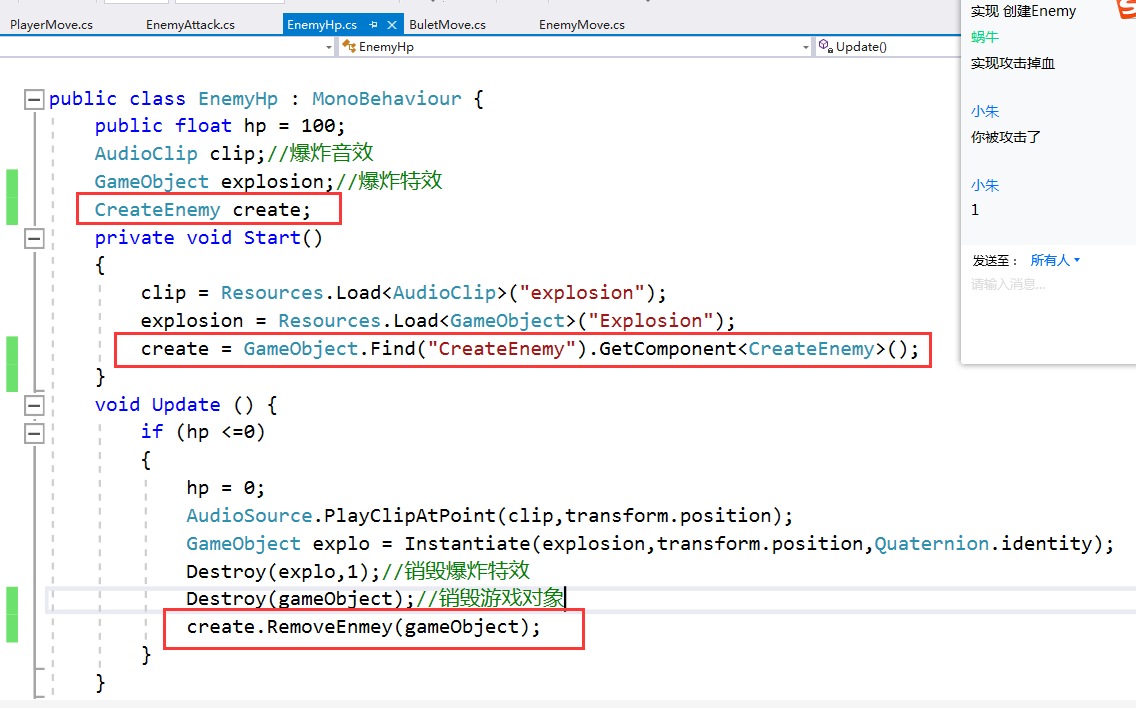
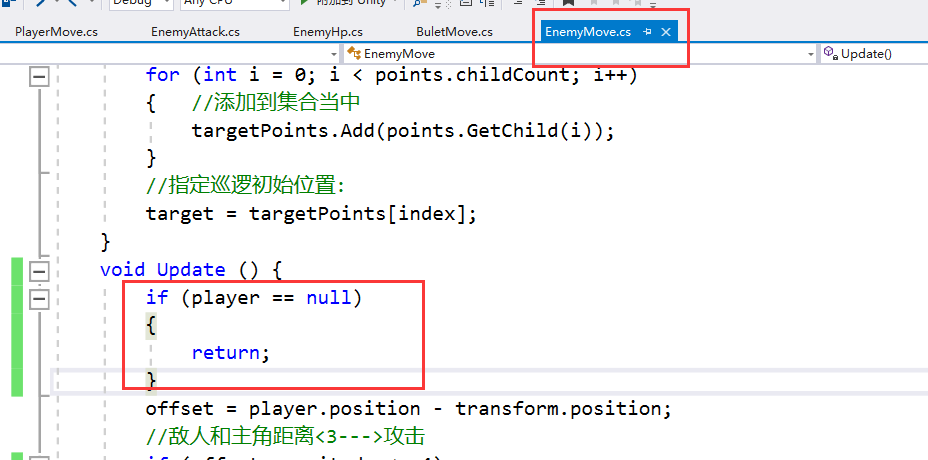


**实现主角移动,按下鼠标发射子弹,子弹2秒销毁,并且产生音效,主角死亡,游戏暂停,并且产生爆炸音效**





**代码：**

public class BulletMove : MonoBehaviour

{

Rigidbody rigi;

void Start()

{

rigi = GetComponent<Rigidbody>();

}

void Update()

{

//transform.position += transform.forward \* Time.deltaTime \* 10f;

rigi.velocity = transform.forward \* 10f;

Destroy(gameObject, 3f);

}

private void OnCollisionEnter(Collision collision)

{

if (collision.gameObject.name == "player")

{

collision.gameObject.GetComponent<PlayerMove>().BeiAttack(20);

Destroy(gameObject);

}

if(collision.gameObject.tag=="enemy")

{

collision.gameObject.GetComponent<EnemyHp>().TakeDamage(10);

Destroy(gameObject);

}

}

}

public class EnemyAttack : MonoBehaviour

{

//实例化子弹：

GameObject bullet;

Transform firepoint;

AudioClip clip;//音乐片段

void Start()

{

bullet = Resources.Load<GameObject>("bullet");

clip = Resources.Load("weapon\_enemy") as AudioClip; ;

firepoint = transform.GetChild(1).GetChild(0);

}

public void CreateBullet()

{

GameObject obj = Instantiate(bullet, firepoint.position, transform.rotation);

//PlayClipAtPoint:播放音效的方法，不需要手动添加AudioSource组件

//参数1：要播放的音效

//参数2：播放音效的位置

AudioSource.PlayClipAtPoint(clip, transform.position);

Destroy(obj, 2f);

}

}

public class PlayerMove : MonoBehaviour

{

GameObject bullet;

Transform fire;

AudioClip explosion;

AudioClip weapomClip;

GameObject Explosion;

void Start()

{

bullet = Resources.Load<GameObject>("bullet");

weapomClip = Resources.Load("weapon\_enemy") as AudioClip;

fire = transform.GetChild(1).GetChild(0);

Explosion= Resources.Load("Explosion") as GameObject;

explosion= Resources.Load<AudioClip>("explosion");

}

public int hp=200;

void Update()

{

float hor = Input.GetAxis("Horizontal");

float ver = Input.GetAxis("Vertical");

transform.position += transform.forward \* ver \* Time.deltaTime \* 10f;

transform.Rotate(transform.up \* hor \* Time.deltaTime \* 60f);

if (Input.GetMouseButtonDown(0))

{

//生成子弹

GameObject obj = Instantiate(bullet, fire.position, transform.rotation);

AudioSource.PlayClipAtPoint(weapomClip, transform.position);

Destroy(obj, 2f);

}

if (hp <= 0)

{

hp = 0;

GameObject obj = Instantiate(Explosion, fire.position, transform.rotation);

AudioSource.PlayClipAtPoint(explosion, transform.position);

Destroy(obj, 1f);

Destroy(gameObject);

Time.timeScale = 0;//游戏暂停

}

}

public void BeiAttack(int vaule)

{

hp -= vaule;

}

}

public class EnemyHp : MonoBehaviour

{

public int hp = 100;

AudioClip clip;

GameObject explosion;//爆炸特效

CreateEnemy create;

void Start()

{

clip = Resources.Load<AudioClip>("explosion");

explosion = Resources.Load<GameObject>("Explosion");

create = GameObject.Find("CreateEnemy").GetComponent<CreateEnemy>();

}

void Update()

{

if (hp <= 0)

{

hp = 0;

AudioSource.PlayClipAtPoint(clip, transform.position);

GameObject explo = Instantiate(explosion, transform.position, Quaternion.identity);

Destroy(explo, 1f);//销毁爆炸特效

Destroy(gameObject);

create.RemoveEnemy(gameObject);

}

}

public void TakeDamage(int vaule)

{

hp -= vaule;

}

}

public class CreateEnemy : MonoBehaviour

{

List<GameObject> tanklist = new List<GameObject>();

public int maxCount = 5;

float timer = 0;

GameObject enemyobj;

GameObject enemy;

void Start()

{

enemy = Resources.Load<GameObject>("Enemy");

}

void Update()

{

if (tanklist.Count >= maxCount)

{

return;

}

else

{

timer += Time.deltaTime;

if (timer >= 4f)

{

timer = 0;

enemyobj = Instantiate(enemy, new Vector3(Random.Range(-8f, 8f), 0, Random.Range(-8f, 8f)), Quaternion.identity);

tanklist.Add(enemyobj);

}

}

}

public void RemoveEnemy(GameObject obj)

{

tanklist.Remove(obj);

}

}

//实现敌人巡逻攻击，敌人AI

public class EnemyMove : MonoBehaviour

{

//巡逻位置管理器：

List<Transform> targetPoints = new List<Transform>();

//巡逻体的父物体：

Transform points;

//目标位置：

Transform target;

Transform player;

int index = 0;

Vector3 offset;

//移动速度

public float moveSpeed = 5f;

//旋转速度

public float rotateSpeed = 45f;

EnemyAttack attack;

float timer = 0;

void Start()

{

attack = GetComponent<EnemyAttack>();

player = GameObject.Find("player").transform;

points = GameObject.Find("points").transform;

for(int i = 0; i < points.childCount; i++)

{

//添加到集合中

targetPoints.Add(points.GetChild(i));

}

//指定巡逻初始位置

target = targetPoints[index];

}

void Update()

{

if (player == null)

{

return;

}

offset = player.position - transform.position;

//敌人和主角距离<3--->攻击

if (offset.magnitude <= 3)

{ //强制看向人物

transform.LookAt(player.position);

//攻击:

timer += Time.deltaTime;

if (timer >= 0.5f)

{

timer = 0;

attack.CreateBullet();

}

}

else if (offset.magnitude > 3 && offset.magnitude <= 8)//追击

{

Quaternion qua = Quaternion.LookRotation(offset);

transform.rotation = Quaternion.Lerp(transform.rotation, qua, 0.5f);

transform.position = Vector3.MoveTowards(transform.position, player.position, Time.deltaTime \* moveSpeed);

}

else

{

//旋转:

Quaternion qua = Quaternion.LookRotation(target.position - transform.position);

transform.rotation = Quaternion.Lerp(transform.rotation, qua, 0.5f);

//巡逻

transform.position = Vector3.MoveTowards(transform.position, target.position, Time.deltaTime \* moveSpeed);

}

//判断角色和目标的距离

if (Vector3.Distance(transform.position, target.position) < 0.02f)

{//切换目标:

target = targetPoints[++index % targetPoints.Count];

}

}

}