**《虚拟现实技术》课程作业**

Unity坦克游戏

院系：软件学院

班级：1804

学号：U201816816

姓名：杨子越

**Unity坦克游戏**

说明文档

# 一、介绍

该项目以虚拟现实技术课程中老师所教授UnityChan相关 人物角色动画控制为基础，引入室外场景模型并添加光照和阴影；添加交互性元素如 发射炮弹，键盘按钮移动坦克；实现第一、三人称视角切换。

开发平台为Unity 2018.4.21f1（64-bit），操作系统为Windows 10（64位）。

# 二、操作说明

使用Visual Studio编译这个游戏，然后运行。

## 1、这个是游戏刚开始进入时的场景



## 2、移动人物漫游场景

WSAD键用来控制坦克的位置，行走用户以第三人称视角在整个场景中进行移动。

## 3、移动鼠标切换视角

可以移动鼠标，来切换视角



## 4、鼠标点击发射炮弹

点击鼠标的时候，可以发射炮弹到一个位置，观看到炸弹爆炸的效果





# 三、探索的技术

## （1） 多摄像机切换

using UnityEngine;

public class CameraControl : MonoBehaviour

{

public float m\_DampTime = 0.2f;

public float m\_ScreenEdgeBuffer = 4f;

public float m\_MinSize = 6.5f;

[HideInInspector] public Transform[] m\_Targets;

private Camera m\_Camera;

private float m\_ZoomSpeed;

private Vector3 m\_MoveVelocity;

private Vector3 m\_DesiredPosition;

private void Awake()

{

m\_Camera = GetComponentInChildren<Camera>();

}

private void FixedUpdate()

{

Move();

Zoom();

}

private void Move()

{

FindAveragePosition();

transform.position = Vector3.SmoothDamp(transform.position, m\_DesiredPosition, ref m\_MoveVelocity, m\_DampTime);

}

private void FindAveragePosition()

{

Vector3 averagePos = new Vector3();

int numTargets = 0;

for (int i = 0; i < m\_Targets.Length; i++)

{

if (!m\_Targets[i].gameObject.activeSelf)

continue;

averagePos += m\_Targets[i].position;

numTargets++;

}

if (numTargets > 0)

averagePos /= numTargets;

averagePos.y = transform.position.y;

m\_DesiredPosition = averagePos;

}

private void Zoom()

{

float requiredSize = FindRequiredSize();

m\_Camera.orthographicSize = Mathf.SmoothDamp(m\_Camera.orthographicSize, requiredSize, ref m\_ZoomSpeed, m\_DampTime);

}

private float FindRequiredSize()

{

Vector3 desiredLocalPos = transform.InverseTransformPoint(m\_DesiredPosition);

float size = 0f;

for (int i = 0; i < m\_Targets.Length; i++)

{

if (!m\_Targets[i].gameObject.activeSelf)

continue;

Vector3 targetLocalPos = transform.InverseTransformPoint(m\_Targets[i].position);

Vector3 desiredPosToTarget = targetLocalPos - desiredLocalPos;

size = Mathf.Max (size, Mathf.Abs (desiredPosToTarget.y));

size = Mathf.Max (size, Mathf.Abs (desiredPosToTarget.x) / m\_Camera.aspect);

}

size += m\_ScreenEdgeBuffer;

size = Mathf.Max(size, m\_MinSize);

return size;

}

public void SetStartPositionAndSize()

{

FindAveragePosition();

transform.position = m\_DesiredPosition;

m\_Camera.orthographicSize = FindRequiredSize();

}

}

1. 坦克生命值切换

using UnityEngine;

using UnityEngine.UI;

public class TankHealth : MonoBehaviour

{

public float m\_StartingHealth = 100f;

public Slider m\_Slider;

public Image m\_FillImage;

public Color m\_FullHealthColor = Color.green;

public Color m\_ZeroHealthColor = Color.red;

public GameObject m\_ExplosionPrefab;

private AudioSource m\_ExplosionAudio;

private ParticleSystem m\_ExplosionParticles;

private float m\_CurrentHealth;

private bool m\_Dead;

private void Awake()

{

m\_ExplosionParticles = Instantiate(m\_ExplosionPrefab).GetComponent<ParticleSystem>();

m\_ExplosionAudio = m\_ExplosionParticles.GetComponent<AudioSource>();

m\_ExplosionParticles.gameObject.SetActive(false);

}

private void OnEnable()

{

m\_CurrentHealth = m\_StartingHealth;

m\_Dead = false;

SetHealthUI();

}

public void TakeDamage(float amount)

{

m\_CurrentHealth -= amount;

SetHealthUI();

if (m\_CurrentHealth <= 0f && !m\_Dead)

{

OnDeath();

}

}

private void SetHealthUI()

{

m\_Slider.value = m\_CurrentHealth;

m\_FillImage.color = Color.Lerp(m\_ZeroHealthColor, m\_FullHealthColor, m\_CurrentHealth / m\_StartingHealth);

}

private void OnDeath()

{

m\_Dead = true;

m\_ExplosionParticles.transform.position = transform.position;

m\_ExplosionParticles.gameObject.SetActive(true);

m\_ExplosionParticles.Play();

m\_ExplosionAudio.Play();

gameObject.SetActive(false);

}

# 四、参考资源

**Unity Standard Assets**