

河北水利电力学院

《数字娱乐产品开发》

课程设计说明书

题目：塔防游戏的设计与实现

专业： 数字媒体技术

班级:

学号：

姓名:

指导教师:

成绩:

2021 年 6 月 10 日

目录

- 一、作品简介..... 1
- 二、设计与实现..... 1
 - 1. 游戏场景.....1
 - 2. 塔的信息类.....2
 - 3. 塔的创建类.....2
 - 4. 塔的控制类.....2
 - 5. 子弹控制类.....3
 - 6. 敌人信息类.....3
 - 7. 敌人诞生类.....3
 - 8. 敌人控制类.....4
 - 9. 相机移动类.....4
 - 10. 路径管理类.....4
- 三、技术难点及解决方案.....4
- 四、待改进的问题.....5
- 五、设计总结.....6

一、作品简介

这款游戏是一款单机类塔防游戏，在导入的场景文件的塔基上创建相应的武器模型，通过键盘上的 WSAD 键和方向键控制摄像机移动和鼠标右键控制摄像机旋转。当鼠标移动到塔基的位置产生阴影通过数字键盘 1、2 键切换武器模型，当点击左键时创建塔阴影相对应的武器模型。键盘的 1 键模型为弓弩，键盘的 2 键模型切换为大炮，当武器模型检测到敌人时，则根据当前自身的攻击的类型来进行攻击。弓弩为跟随效果，单一伤害，大炮为抛物线效果，范围伤害。当游戏运行 5 秒后批量产生 20 个敌人，每个敌人之间间隔 2 秒产生，当创建达到 20 个敌人后，间隔 5 秒再依次创建另一种类型的敌人。当敌人被创建出来时，设置敌人速度让敌人沿着当前场景中路点依次进行移动播放动画。并设置敌人当前血量、最大血量并实现 3D 血条效果，设置方法供子弹控制脚本进行传递伤害，当血量为 0 时销毁当前敌人。当有敌人走到最后一个路点时，视为游戏失败弹出游戏失败框。当产生的敌人全部被消灭，视为游戏成功弹出游戏成功框。开始游戏后，游戏场景里面有菜单栏可对游戏进行暂停、继续、退出、重新开始等操作。

二、设计与实现

1. 游戏场景





场景描述：图一：菜单界面：开始游戏、退出游戏、游戏关于；图二：游戏主界面：主界面、暂停\继续游戏、退出游戏；图三：交锋场面。图四，成功界面。

2. 塔的信息类

塔的信息类 (TowerBaseInfo), 该类的主要功能是代表每座塔的信息。

主要属性：定义枚举获取塔 `public TowerBase_Type type = TowerBase_Type.None;`

方法说明：`public enum TowerBase_Type` 代表塔类型的枚举设置-1 表示空塔、0 表示弓弩、1 表示大炮。

3. 塔的创建类

塔的创建类 (CreateTower), 该类的主要功能是当鼠标放在塔基的位置应当创建阴影当点击左键时创建塔阴影对应的模型 (弓弩, 大炮) 当按下键盘的 1 2 键切换创建的模型。

主要属性：`GameObject[] towerPrefabs` 用来存放从内存中获取到的塔；`GameObject[] towerInstanceShadowPrefabs` 用来存放实例化到场景中的塔阴影；`GameObject[] towerShadowPrefabs` 用来存放从内存中获取到的塔阴影；`GameObject[] towerShows` 用来存放场景中已经存在的显示模型；`int currentShowIndex = 0` 当前显示塔的索引，当前索引为 0 时表示弓弩，索引为 1 时表示大炮；`Ray ray` 声明射线类型；射线返回的信息

RaycastHit hit; 可以被检测层级对象 public LayerMask layer; 塔的高度 float high = 13.5f。

方法说明：Resources.Load 方法可以通过在该文件夹中根据传入的路径进行加载游戏对象，参数是 string 类型 要的是该文件下具体到加载对象名字的路径。void SetTowerCurrentIndex(int index) 设置塔当前显示索引。

4. 塔的控制类

塔的控制类 (TowerController)，该类的主要功能是每隔一个攻击间隔时间进行检测范围内是否有敌人如果有敌人，则根据当前塔攻击的类型来决定攻击方式，攻击方式：弩为跟随效果，单一伤害，大炮为抛物线，范围伤害。

主要属性：GameObject[] bullets 子弹预制体；string[] bulletNames = { "Bullet_Fire", "Bullet_Ice" } 子弹预制体的名字；public float Interval = 2 间隔时间；Transform firePoint 发射点；public BulletType bulletType 子弹类型的枚举；public LayerMask layer 代表敌人的层级；Collider[] targets 检测到的敌人碰撞碰撞器数组；public float atkRange = 80 检测的攻击半径。

方法说明：transform.Find 在当前的子物体下方根据名字去查找的方法。IEnumerator AutoCheckRangeEnemy() 协程自动检测范围内的敌人的方法。public Transform GetEnemy() 查找敌人的方法通过球形检测范围内的敌人。

5. 子弹控制类

子弹控制类 (BulletsController)，该类的主要功能是子弹诞生后，获取当前塔检测到的敌人目标后进行攻击逻辑划分为跟随和抛物线效果。

主要属性：public AtkType atkType 可以被攻击的层级；public float moveSpeed = 10 子弹的移动速度；public BulletType bulletType 公开子弹类型枚举供编辑器选择子弹类型；Transform target 由塔的控制脚本进行赋值代表就是攻击对象的位置。

方法说明：定义协程 IEnumerator AutoFire() 方法让子弹进行攻击，并开启协程 StartCoroutine(AutoFire())。public void SetTarget(Transform enemyTF) 由塔的控制脚本进行访问并将塔获取的攻击对象传递进来。public enum BulletType 子弹的类型的枚举。public enum AtkType 攻击的类型的枚举。

6. 敌人信息类

敌人信息类 (EnemyInfo)，该类的主要功能是设置当前血量最大血量并实现敌人血条效果，对外提供受伤 (Damage) 方法，供子弹控制脚本进行传递伤害，当血量为 0 时销

毁当前敌人，并进行计数，当销毁的数量与生成的数量相等时，意为敌人全军覆没，弹出成功消息框。

主要属性：`public float maxHP = 100` 最大血量的变量；`private float currentHP` 当前血量的变量；`Slider slider` 定义 `slider` 组件用来获取血条。

方法说明：对外提供受伤 `Damage` 方法，进行扣血，当血量为 0 时销毁当前敌人。

7. 敌人诞生类

敌人诞生类（`SpwanEnemy`），该类的主要功能是挂载到场景 `ScriptsManager` 物体上游戏运行 5 秒后批量产生 20 个敌人，每个敌人之间间隔 2 秒产生，当创建达到 20 个敌人后，间隔 5 秒再依次创建另一种类型的敌人。

主要属性：`GameObject[] Enemys` 用来存放从内存中获取到的敌人。`string[] EnemyName = { "Enemy01", "Enemy02" }` 敌人的名字。

方法说明：使用协程方法产生敌人控制波数和个数，并在 `Start()` 方法中开启协程。

8. 敌人控制类

敌人控制类（`EnemyController`），该类的主要功能是将脚本挂载到敌人预制体身上，当敌人被创建出来时，沿着当前场景中路点依次进行移动播放动画等。当敌人到达最后一个路点时，意为游戏失败，弹出游戏失败框，并销毁对象，使得销毁敌人的数量与生成敌人的数量不对等，消除 Bug；游戏成功框的弹出。

主要属性：定义 `public int index = 0` 定义索引用来获取地图上路点的索引；定义 `public float speed = 15f` 敌人的速度。

方法说明：定义一个敌人移动的方法 `void Moving()` 供其调用。

9. 相机移动类

相机移动类（`CameraController`），该类的主要功能是使用键盘水平和垂直轴值实现相机平按住鼠标右键滑动鼠标实现相机左右旋转。

主要属性：`public float moveSpeed` 定义移动的速度；`public float rotateSpeed` 定义旋转的速度。

方法说明：在 `Update()` 方法中获取键盘水平垂直，做成 `Vector3`，然后根据 `v3` 进行移动，当按住鼠标右键获取鼠标水平轴值影响摄像机旋转，根据鼠标左右滑动旋转摄像机。

10. 路径管理类

路径管理类（`PathManager`），该类的主要功能是在开始时找到所有的路点依次存储，然后进行绘制路线需要 `LineRender` 组件才能绘制在 `Game` 视窗。

主要属性: `Vector3[] points` 定义路点数组; `LineRenderer line` 线性渲染组件。

方法说明: 在 `Start()` 方法中分配路点的数组空间, 获取父物体下的每一个路点存储到路点数组中。获取线性组件, 将绘制点的数量赋值给线性组件, 根据路点数组绘制一根线。

三、技术难点及解决方案

1. 如何使用键盘控制实现相机平移?

使用键盘水平和垂直轴实现相机平移, 通过 `Input.GetAxis("Horizontal")` 和 `Input.GetAxis("Vertical")` 获取键盘水平垂直, 通过 `transform.Translate(moveDir * Time.deltaTime * moveSpeed, Space.World)` 进行移动。将 `y` 值实时赋值为 0 防止向下运动 `moveDir.y = 0`。

2. 如何获取路点数组?

可以将 `PathManager` 里面的路点数组设置为静态的, 加载 `Path` 路径下的 `Points` 点, 最后定义数组获取。

2. 如何为敌人添加血条并在诞生时一起出现?

将敌人拖到场景中, 在敌人的子级下面添加一个 `Canvas`, 在其中添加一个 `Slider` 组件, 调整属性, 设置其为血条将其放到敌人头上的合适的位置。将设置好的敌人设置成预制体, 挂上相对应的脚本将会在诞生时一起出现。

3. 如何自动检测范围内的敌人?

创建协程, 创建死循环每次循环都要等待一个间隔时间 再去搜索敌人, 当搜索到敌人时, 将创建在塔基上的武器看向敌人, 将方法放入 `Update()` 方法中进行实时监测, 并利用 `Interval` 函数设置监测的时间间隔。

4. 游戏成功框与游戏失败框的弹出。

首先将两个框创建出来, 并创建成预制体, 分别在敌人移动脚本和敌人受伤脚本中, 通过定义变量用来从内存中获取预制体数组 `GameObject[] Canvas`; 获取加载 `Canvas[i]=Resources.Load<GameObject>("Enemies/" + CanvasName[i])` 与 诞生 `Instantiate(Canvas[0])`。

四、待改进的问题

1. 界面不美观, 游戏场景和武器模型未更换。改进思路: 自行设计塔防游戏的主题, 利用 `3Ds max` 软件、`C4D` 等软件进行模型创建贴题, 再通过 `AI`、`PS` 修图软件进行主题图片的设计, 全部制作完成后, 陆续替换到塔防游戏的项目中。

2. 未设置关卡，模型和素材搭配不合理，功能较为简单，可玩性不高，改进思路：编写相应的控制器脚本例如：敌人血量控制器、敌人移动速度控制器和武器装备的攻击速度和攻击范围控制器，通过设置敌人和武器的各项属性，依次设置游戏的关卡，提高游戏的可玩性。

3. 继续游戏的功能未实现，改进思路：编写继续游戏按钮的脚本，使用 `Time.timeScale` 实现游戏的时间的暂停与继续。

4. 用户名账号登录功能未实现，改进思路：创建新的场景，进行用户登录页面的设计，利用 unity 插件查询已登录用户的信息，跳转的相应等级的游戏的场景，提升用户体验，大大增加了用户的粘性。

5. 未实现联机，现在的游戏目前属于一款单机游戏，如果实现联机功能，会大大提高可玩性，改进思路：查询 Unity 手册的联网 API 添加到项目中，并创建相应的脚本 `ConnectNetwork` 类职责：实现联机，并对联机用户所闯关卡积分制进行大排名。

五、设计总结

在设计开发开始之前，本人大致了解简单的游戏项目开发，但是由于没有系统学习脚本编写，所以在动手制作游戏过程中，经常卡在脚本的编写上。例如场景设计中，路点获取并绘制路线的脚本就成为开发中一个很大的难题，在查阅资料和视频之后成功将一些问题解决，但是薄弱点依然存在，同学和老师在游戏开发中给予本人很大的帮助，推荐了一些有用的资料和教程让游戏制作成功。在今后的学习中，本人将继续努力学习，在游戏设计开发中，遇到的一些问题和游戏本身存在的一些问题将是本人今后一段时间内继续努力学习方向，同时还将用心学好 C# 这门编程语言和 Unity3D 游戏开发这款游戏引擎。通过这次塔防游戏的设计，对游戏设计思路更加清晰，塔防类游戏的核心模块为塔的攻击和子弹的攻击脚本，通过老师带领我们搭建并且编写脚本最后到自己实现功能并添加一些个性化的拓展，让我们明白从项目方案敲定到需求设计转变到代码实现的全部流程，从而感觉到游戏开发设计的魅力，为后期进一步学习游戏开发相关的知识打下坚实的基础，也大大激起了我对游戏设计开发的兴趣。最后呢，本人将自己开发的这款塔防游戏的最初版本 1.0 已经开源到 Github 社区，为大家的游戏交流提供便利。自己也会将这款塔防游戏的后续版本继续开发下去，完善这款游戏的功能的同时对代码算法进行及时的优化与更新提升用户体验的同时也增加了游戏的稳定性与鲁棒性。