基于 HoloLens 的台球游戏设计

盖中会、汤君豪

（一）概述

我们的设计为基于 HoloLens 的桌球游戏，游戏的目标是将所有的数字球击入桌角的球带中，避免白球入洞。当所有数字球都被击入洞时，视野中央会有获胜特效；当白球入洞时，白球位置会复原（出现在场景中央），并产生烟雾的特效。玩家可以通过控制面板控制游戏的开始、重置。

我们所采用的交互方式为：使用一个白球和一个击球器，二者都可以被玩家的手所移动，玩家利用这两者所形成的直线，将白球击出，以射击其他数字球。

（二）实现内容

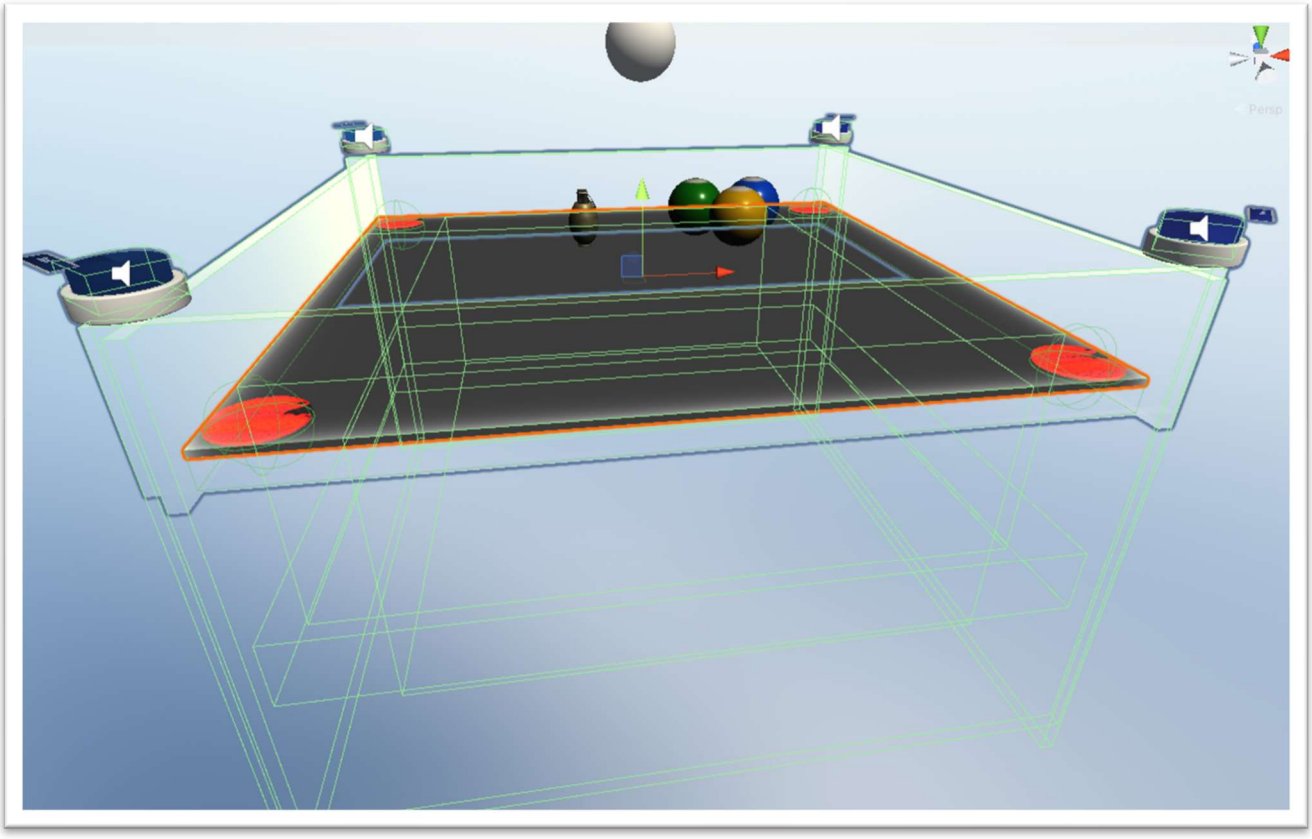
1.基本的游戏环境，包括球台、台球（数字球和白球）、发射器
2.球台上物品与玩家的交互，玩家可以用手移动其位置
3.跟随玩家视角的 UI 界面，可以控制游戏开始、重置和发射
4.桌角的4个按钮，用于向对角方向击球
5.按钮按下时、击球、白球入洞和游戏获胜时的特效
6.游戏获胜、失败的判定

（三）实现思路

1.游戏环境的布置

游戏的桌面部分需要一个平面、4堵墙、4个球洞。其中4堵墙使用4个盒状的碰撞体即可；球台需要使用两层碰撞体，上层的碰撞体用于表示桌面，所以需要留出4个角落使球到达边缘时可以落下，下层的碰撞体用于接住球，使球不至于落出视野外，达到落入球洞的效果。

桌面上的东西需要实现碰撞效果，所以需要添加刚体组件。需要实现反弹的效果，所以需要添加物理材质，调整参数设置至效果合适。



2.物品与玩家手的交互

在白球和击球器上添加组件 NearInteractionGrabbable，将交互类型设置为移动和旋转。

3.跟随玩家视角的 UI 界面

添加管理游戏全局状态的 GameManager 脚本，并将其添加到场景的空对象中，添加接口函数 GameManager::Start()，GameManager::Reset()，GameManager::Shot()，在场景中使用按钮组的预制体，将按钮按下的事件绑定到对应的函数上即可。

4.桌角的4个按钮

使用按钮预制体，将其放置在4个桌角，便携 ButtonShot 脚本，实现 Shot()方法，将按钮的按下事件绑定的该函数上，在此函数中，我们需要给白球的刚体组件一个力（击球），并播放特效。

5.特效的实现

在需要播放特效的地方使用 Instance(GameObject,Transform)生成特效预制体，特效预制体需要提前设置：大小适合场景、自动开始、不循环。这样生成特效后就会自动播放，并且播放结束后会自动 Destroy()。

6. 游戏获胜、失败的判定

同样是使用全局管理的脚本 GameManager 来管理、判定游戏状态，我们在 GameManager::Update()函数中我们就可以判定是否全部数字球都入洞、白球是否入洞，根据判定结果调用对应函数（获胜或者白球入洞），以管理游戏状态。

（四）核心代码

1.击球的动作

```
public void Boom()
{
    Vector3 direction = new Vector3(target.transform.position.x - transform.position.x,
        0, target.transform.position.z - transform.position.z);
    direction = direction.normalized * I;

    //击球
    target.GetComponent<Rigidbody>().AddForce(direction);

    //生成爆炸的特效
    Transform fire_transform = this.transform;
    fire_transform.forward = new Vector3(0, 1, 0);
    Instantiate(boom_fire, fire_transform);
}
```

2.游戏状态的判定、管理

```
void Update()
{
    bool all_falled = true;
    for(int i=0;i<=2;i+=1)
    {
        if(!Ball_Falled(i))
        {
            all_falled = false;
            break;
        }
```

```
    }

    if(!showed&&all_falled)
    {
        showed = true;
        Instantiate(Win_Effect, Win_Transform);
    }
}

//判断球是否入洞
bool Ball_Falled(int index)
{
    return balls[index].transform.position.y < -0.605;
}
```