# 3D平衡球游戏

# 需求分析与概要设计

## 项目说明

3D平衡球是我们小时候玩的一款经典游戏，再加上最近在b站刷到了大佬玩平衡球的视频，我们小组决定用c#和unity实现这款游戏，也作为我们用unity实现的第一个项目。

## 项目目标：

1.在主菜单界面可以选择不同的关卡进行挑战，同时还有背景音乐。

2.开始挑战后，玩家要在限定时间内控制小球收集完全部金币并抵达终点。其中地图上会有不同功能的板，例如循环移动的木板，上升踏板，传送板，还有会发射导弹的npc等。

循环移动的木板：该木板会按照固定的路径循环移动。

上升踏板：小球接触到该踏板后，会受到一个向上的力，使小球腾空。

传送板：小球接触到该传送板后，会被传送到另一个传送板旁边。

会发射导弹的npc：小球会被npc发射的导弹击退甚至击飞。

3.若在挑战期间掉落或超时，则失败。

## 软硬件环境需求

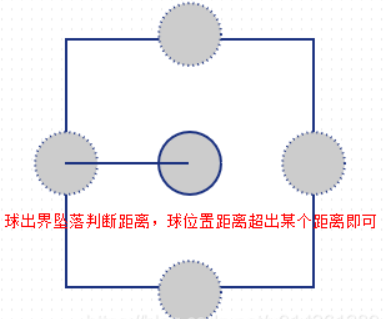
Visual Studio 写主体代码。

Unity负责前端相关模型的导入，以及最后生成文件。

## 使用的关键技术：

3D平衡球的实现主要要完成以下四点。

1. 小球控制：创建Player脚本，Input.GetKey(KeyCode.W);添加四个判断（wasd），小球具有加速度，同时用鼠标控制小球运动的视角，使运动更真实。
2. 相机跟随：创建Camera follow脚本，相机始终在小球的后方，相差一个固定的值。同时也创建了Camera look脚本，相机会跟随鼠标的移动转变视角。
3. 制作道具：游戏的魅力在于就是有各种各样的关卡和道具。本游戏中，制作了循环移动的木板，上升踏板，传送板等道具。
4. 掉落判定：球心与边缘位置距离超出设定值即可。



## 需求分析

## 系统用例

玩家用键盘wasd控制小球的运动，用鼠标控制小球的视角。

## 业务流程 Y

主界面

N

关卡挑战

Y

挑战失败

挑战成功

是否返回主菜单

是否返回主菜单

N

下一关挑战

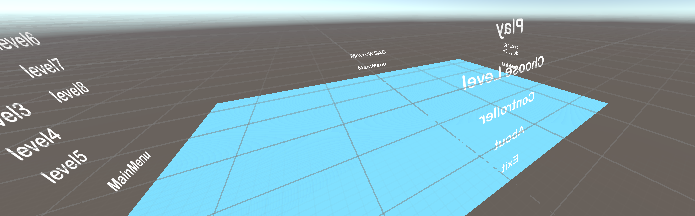
图 2游戏流程图

## 概要设计

## 功能模块设计

3.1.1 界面切换：

在菜单界面内，不采用2D普通界面，而是在3D空间内，在4方位的空中展示菜单，通过镜头视角的转换完成菜单界面的切换。（菜单界面与游戏场景的切换仍是不同scenes之间的切换）



3.1.2 传送门（关卡内传送门、关卡间传送门即胜利门）

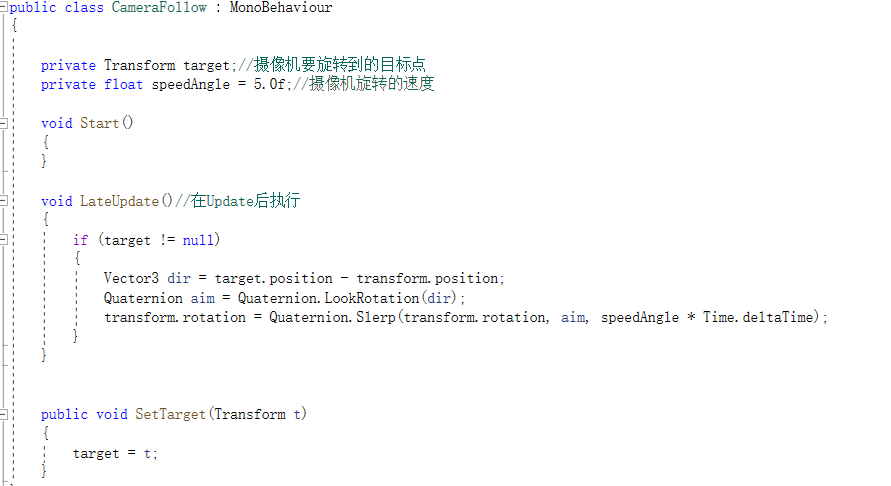
关卡内传送门需记录两两传送门之间的对应关系，通过方位的记录完成传送。关卡间传送门直接变化scenes，主要需要胜利条件判定（金币的拾取）

3.1.3 小球移动

小球移动的操作难度是游戏难度的主要决定因素，操作过于简单一是不符合“平衡球”的游戏设定。小球的移动方向通过获取键盘输入WASD来控制，利用unity物理引擎，朝键盘输入指示方向施加力。

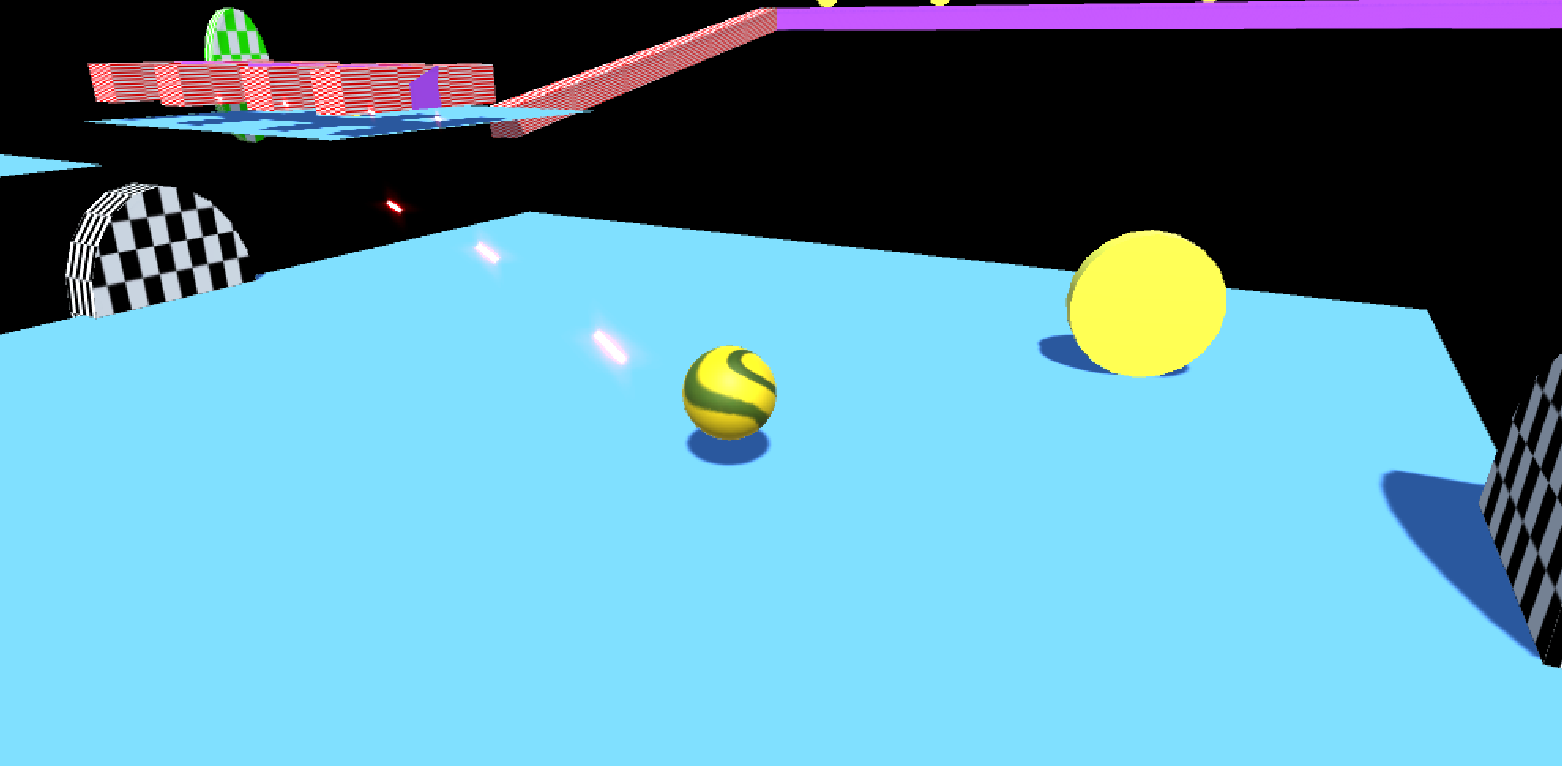
3.1.4 镜头跟随、移动

在菜单界面中，镜头位置固定在场景中间，镜头方位有4个可动变化。在游戏场景中，镜头固定在平衡球的后上方（使镜头位置始终等于小球位置加上某一个偏移量），镜头朝向可随着鼠标移动改变，将鼠标移动的画面帧数通过公式计算出视角的移动角度。



3.1.5 NPC、跟踪子弹

NPC在地图的两个位置反复移动，朝平衡球发射子弹。NPC子弹在发射时，首先获取平衡球坐标，通过公式计算出发射位置和小球之间的角度，并向其发射。随后每空相同的时间间隔，获取一次小球的位置，并以与子弹发射类似的方法，使子弹略微偏向平衡球位置（后续根据反复实验，找到合适的时间间隔与偏移角度，使子弹的转向不会因为小球移动过快而十分突兀，同时不会出现子弹掉头等夸张机动）



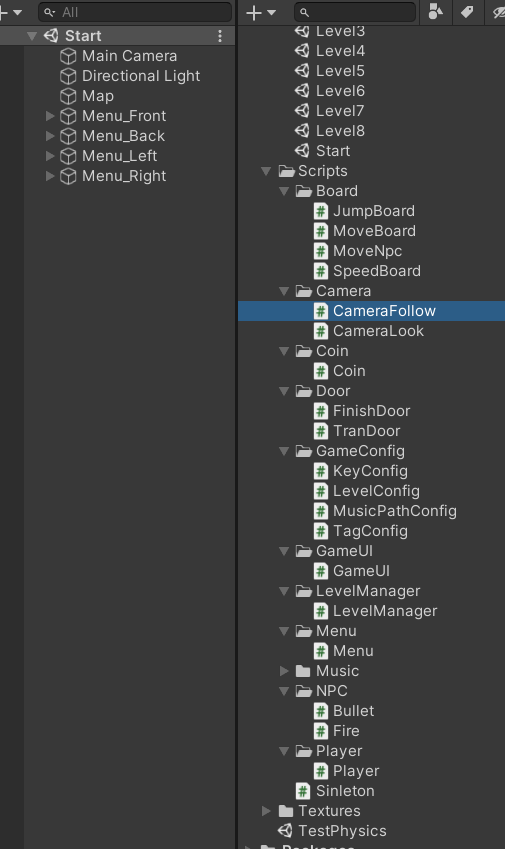
3.1.6 移动地板

地板移动在两点之间移动，进行加速匀速减速3过程，此移动与小球移动一样，只是其移动方向固定。

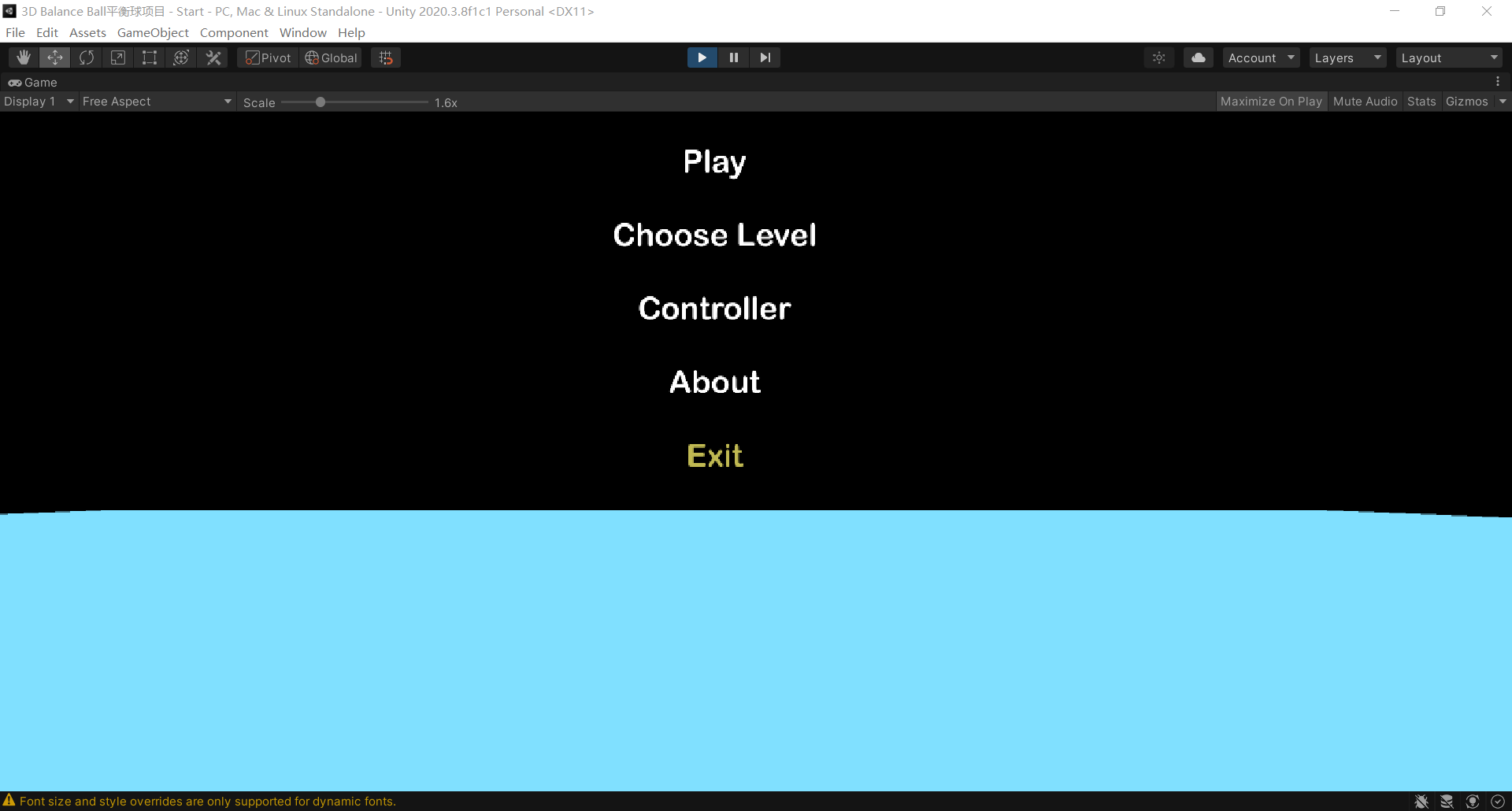
3.1.7 金币

触碰触发，使金币数+1，与最终胜利条件有关

## 核心类图



## 界面设计



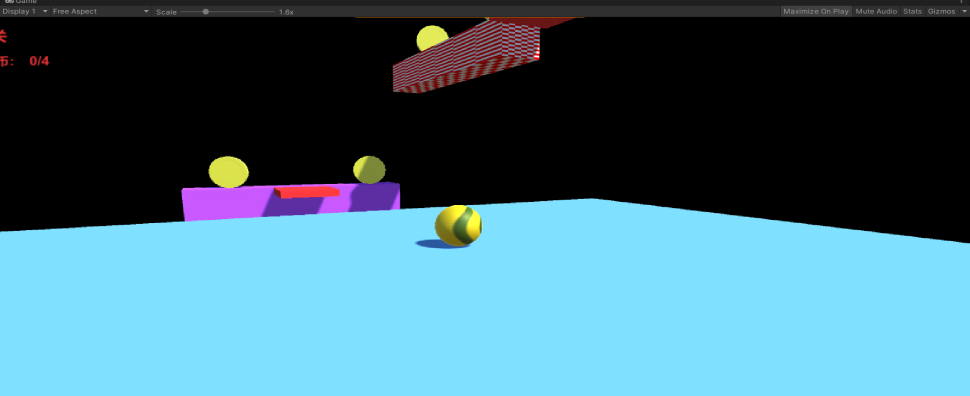


图 5 系统界面