

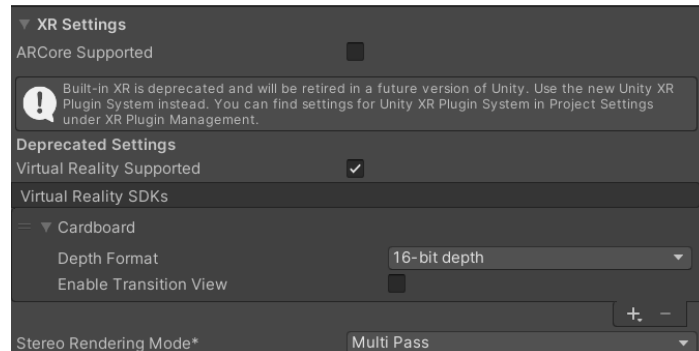
姓名：骆彦伊 Celine Loh Yan Yi

学号：520030990040

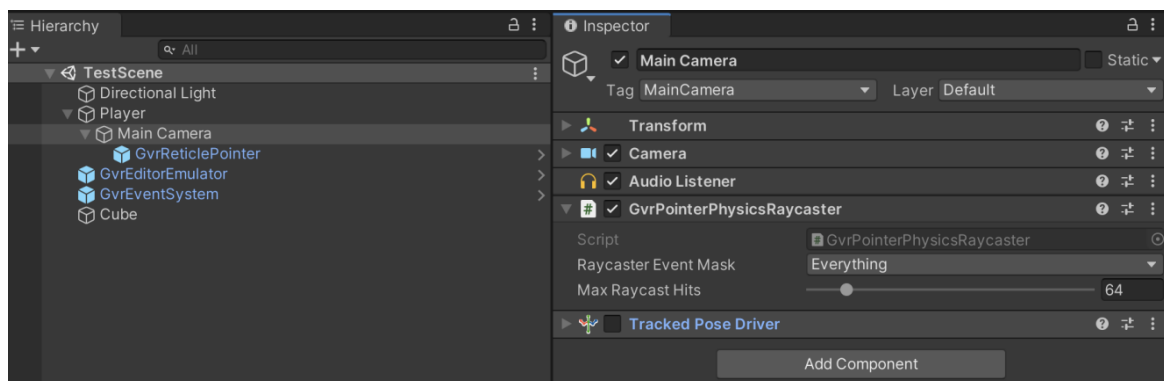
### 上机作业#8: Cardboard VR

#### A. Google VR SDK

1. 导入 Google VR SDK。
2. 在 Player Settings 中，Player > XR Settings 下选中 Virtual Reality Supported，并在 Virtual Reality SDKs 中添加 Cardboard。



3. 在 Player > Other Settings 下找到 Graphics APIs 并把 Vulkan 删除。
4. 新建空物体名为 Player，位于(0,0,0)，设置 Main Camera 为其子物体。注意 Main Camera -> Transform -> position = (0,0,0)。
5. 为 Main Camera 添加 GvrPointerPhysicsRaycaster 组件。该组件使之能和场景中的物体交互。
6. 在项目目录中搜索并添加 GvrEditorEmulator 预设体。该预设体使得 Unity Editor 可以模拟 VR 环境下的头部运动，便于在 Editor 中调试。
7. 在项目目录中搜索并添加 GvrEventSystem 预设体。该预设体使游戏能追踪点击、悬停等事件。
8. 在项目目录中搜索并添加 GvrReticlePointer 预设体，作为 Camera 的子物体。该预设体在游戏中显示为准星，使得用户能利用它在场景物体上点击、悬停。



## B. 游戏设计

### 1. 游戏名称：Tap-On-It

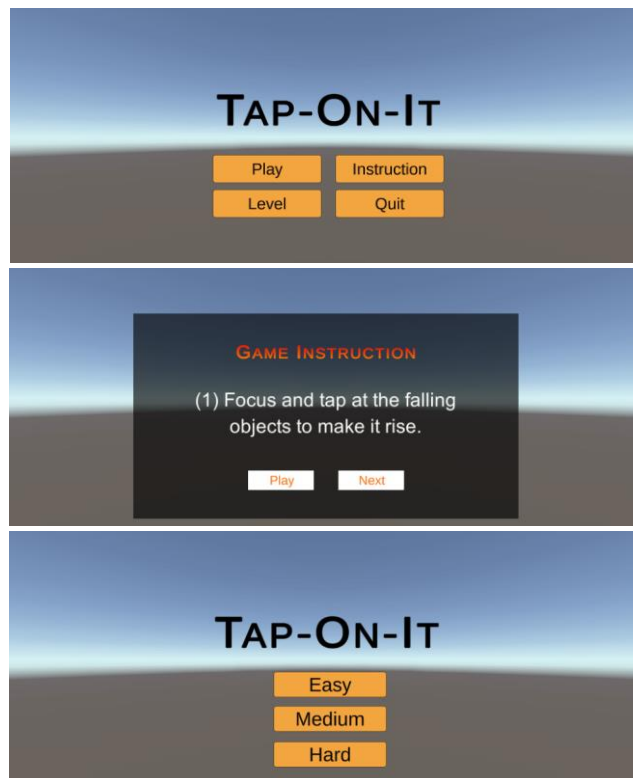
### 2. 游戏玩法：

- 游戏中包含许多不同类型的物体向下掉落。
- 当玩家点击物品时，该物品会往上升一段距离/ randomly teleported。
- 玩家需要保证在游戏时间内没有任何一个物体掉落在 Ground 上即可赢得游戏，否则输了游戏。

### 3. 设计思路：

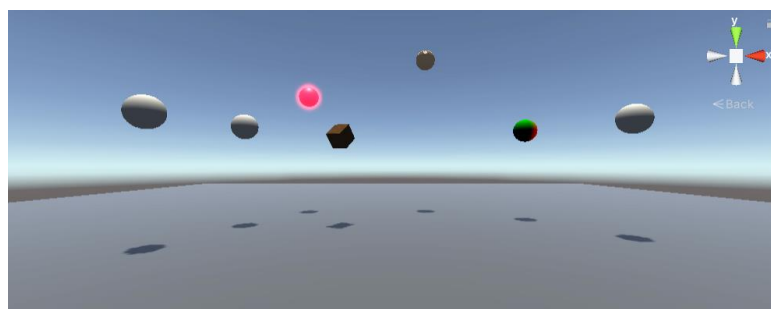
#### a. 首页（导航页）

- 选择 Play 可以直接进入 Level1。
- 选择 Instruction 可以查看游戏玩法。
- 选择 Level 可以任意选择不同游戏关卡，有 Easy、Medium 和 Hard Level。
- 选择 Quit 可以退出游戏。

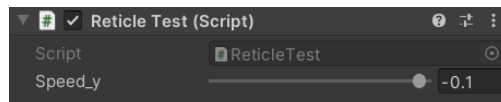


#### b. 游戏场景

- 添加不同物体，这里使用了之前作业内容里的 Basic Shader (Pure Colour Shader, Normal Shader, Texture Shader)、Custom Shader (Bloom Shader, Toon Shader) 以及 physics。



- 给各个物体绑定脚本，改变物体的 Speed\_y 可以改变物体的降落速度



- 获取头盔位姿信息：玩家视角会随着头盔 position & rotation 改变玩家视角

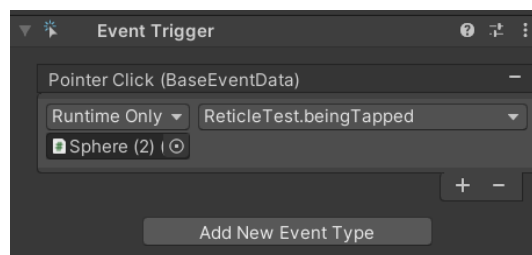
```
movementSpeed = 0.5f;

float zRotation = mainCamera.transform.rotation.eulerAngles.z;

if (zRotation > 180)
{
    zRotation -= 360;
}

if (zRotation >= -45 && zRotation < -15)
{
    transform.position += transform.right * movementSpeed * Time.deltaTime;
}
else if (zRotation > 15 && zRotation <= 45)
{
    transform.position -= transform.right * movementSpeed * Time.deltaTime;
}
else
{
    movementSpeed = 0f;
}
```

- 添加屏幕点击事件：给各个物体添加 Event Trigger Component



```
public void beingTapped()
{
    Vector3 position = gameObject.transform.position;

    // Get the individual x, y, and z values of the position
    float x = position.x;
    float y = position.y;
    float z = position.z;

    gameObject.transform.position = new Vector3(x, y + 2.0f, z);
}

0 references
public void RandomlyTeleport()
{
    var rad = Random.Range(0, 6.18f);
    var r = 2.5f;
    gameObject.transform.position = new Vector3(
        Mathf.Sin(rad) * r, Random.Range(5.0f, 6.0f), Mathf.Cos(rad) * r
    );
}
```

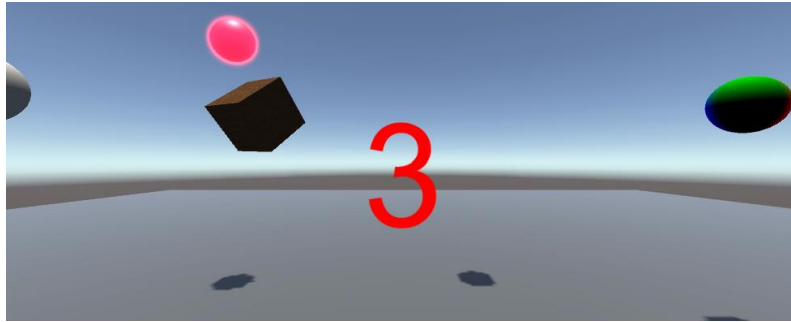
### c. 游戏计分机制

- 添加游戏时长倒计时，在游戏时长内不让物体碰到地面即可赢得游戏



d. 游戏开始

- 倒数 3, 2, 1 开始游戏



e. 游戏结束

- GAME OVER / YOU WIN!

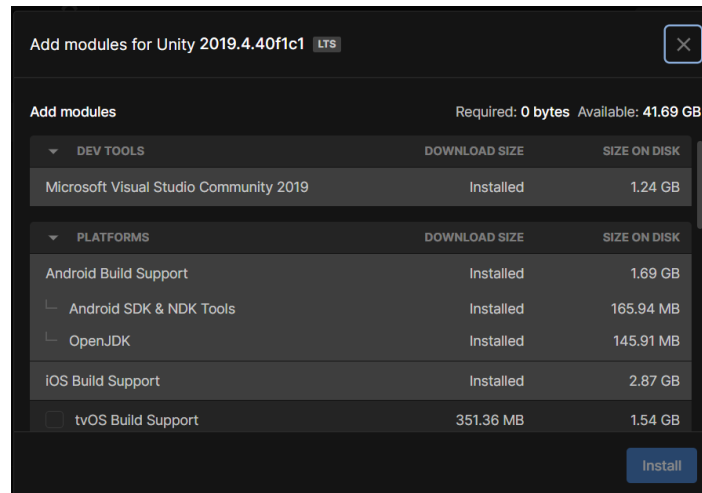


f. 游戏操作指南

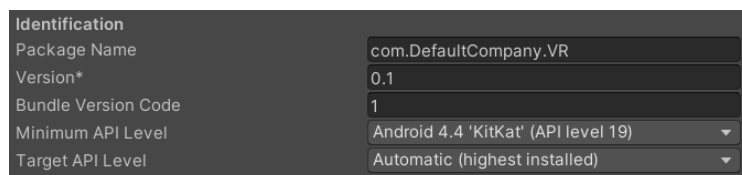
- 手机打开游戏，套上 Google Cardboard
- 旋转头盔即可改变玩家视角
- 当头盔向右压，视角会向右移，反之向左移
- 游戏中包含一个瞄准星，按下 Google Cardboard 上的按钮即可点击屏幕

## C. 配置移动端开发环境（本次作业使用 Windows 电脑 + Android 手机）

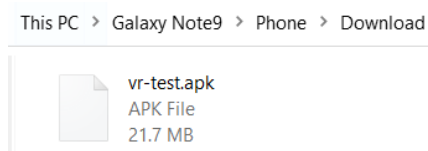
1. 使用 Unity Hub 安装 Android Build Support



2. 打开 project, 在 Build Settings 中, Switch Platform 到 Android
3. 在 Player Settings 中, 找到 Player > Android 下的 Other Settings, 修改 Identification 下的 Package Name 和 Minimum API Level



4. 在 Build Settings 中选中 Build 打包为 apk, 并在 Android 手机中运行
- 把 apk copy 到:



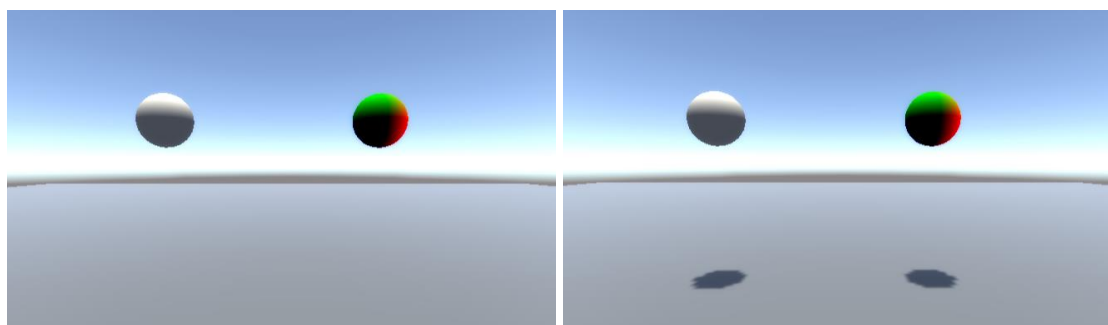
- 从手机打开: My Files > Installation files > Download > vr-test.apk

## D. 遇到的问题

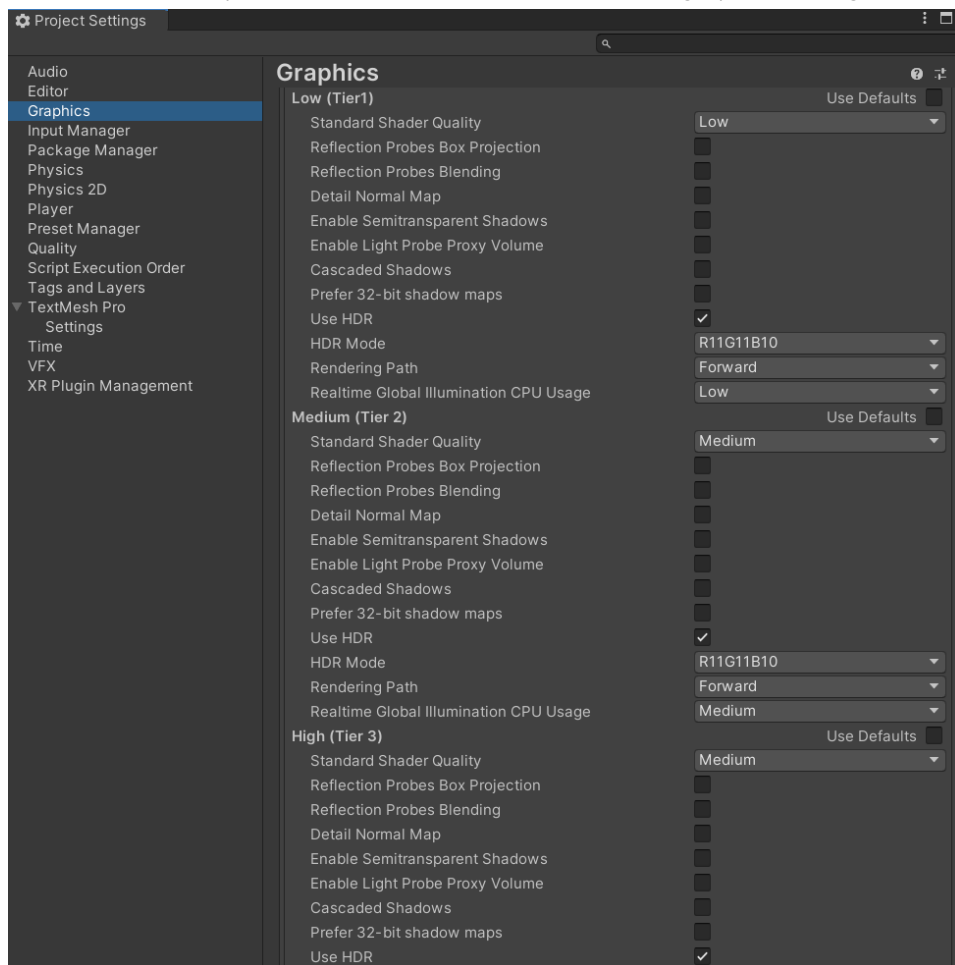
1. 之前作业里编写的 shader 没有 shadow effect, 这次把 shadow 加上去了, 只需要在 shader 里加上这行代码:

```
UsePass "Legacy Shaders/VertexLit/SHADOWCASTER"
```

效果: (Before & After)



2. Bloom Shader 在 scene view 中可以正常渲染，但是在 game view 中却不能。找到的原因是因为之前作业用的是 PC 端，这次 platform 换成了 Android，需要 enable graphics settings 中的 “Use HDR”。



效果：(Before & After)

