# 论文第五版

## 系统实现

### 系统的实现过程

### 主要问题及解决方法

传感器监听SensorEventListener回收的时机

为了实现可以通过点头的方式触发视野检测，调用了Android硬件传感器中的加速度传感器Sensor.TYPE\_ACCELEROMETER与地磁传感器Sensor.TYPE\_MAGNETIC\_FIELD实现获取Android设备x,y,z轴的偏转角度，但是传感器的监听并不会因为程序退出而释放资源，所以需要主动去注销，那么注销与创建的时机尤为重要，为了不浪费过多的资源，监听的注销应当是在检测开始之后，而为了使可以通过点头的方式触发检测，监听的注册应当是在检测之前，那么建议将注册写在onclick中，每次触发选取眼睛的点击事件，且连接有控制器，则注册一个新的传感器监听，但Android无法主动获取unity 中Camera对象的状态（至少我没查到，可以在调研调研），所以如果在跳转到UnityPalyerActiviy之后立即注销掉监听，那么Camera对象可能还没被创建出来，这是如果直接向unity发送消息调用开始检测的方法，肯定是找不到对象的，这样，程序就会卡在这，造成异常，解决办法就是，由unity主动调用Android的注销方法，当unity中的Camera对象创建好后，会主动向Andoid发送消息，调用注销，此时Camera对象已经建立，不会存在Android向unity发送消息找不到对象的问题

结果图是否应该被加载

这可能算是程序上的一个bug，每次刷新fragment时，都会去本地加载结果图，因为结果图是持久化的，所以当检测过一次后，第二次加载如果不做限制，结果图必定会被加载到，而文本数据又不是持久化的，每次退出app后，就重置了，所以这样就会引起歧义，这里有两种解决方式，一是将文本数据也进行持久化，就是把数据写到文件里，每次app启动都要先去读这个文件，二是对结果图的加载条件做制约，如果是没有文本数据的情况下，就不让他进行加载

对于检测过程中有效响应时间的判定

在程序的设计中，视标的显示时间只有短短的0.2秒，如果仅在这0.2秒内同时完成发现，响应这两个动作是极为不合理的，所以在视标消失的空隙时间里，应有一部分时间被视为是有效的响应时间

检测冲突的处理

为了加快检测进度，对控制器的按钮添加了加速方法，每隔2秒或者每点击一次加速，都会调用都一次checking（），但这有极大的概率两个方式会产生冲突，解决的办法就是加上锁，当调用一次完成后，所才会被释放，下一次checking（）才会执行

图像位置处理

由于unity在vr模式下会主动开器陀螺仪（可能是因为我找到的禁用陀螺仪的方法不起作用），故眼前的图形不会是固定不动的，这将对测试产生很大的影响，为了解决这个问题，将将unity所有的组件都绑定到了Camera上，这样，不管检测者怎样移动，图像始终是在眼前，但是随着测试者移动，组件的坐标确是在不断发生变化的，假如还是以固定的坐标作为视标的出现的位置，那么当测试者移动后，将看不到视标，故为了解决这个问题，将视标的位置改为相对于Background的位置，每次显示前，都回去刷新Background的位置

结果图的存储问题

unity中的内容在打包之后是只可读不可写的，即你所生成的结果图无法被写入到你在unity下的文件夹里，所以需要自指定到persistentDataPath 文件夹

/////////////////////////////扩充//////////////////////////

#### 1. 第一种 路径加载文件Resources，当你在Unity工程的Asset路径下创建一个Resources文件夹，然后将一些资源文件放到该文件夹下，那么可以用脚本来直接加载Resources文件夹下的资源。

#### Resources.Load("文件名字，注：不包括文件后缀名");这样Load方法需要传入一个string类型的参数，然后返回Resources文件夹下的对应物体的GameObject对象。要注意的是用这种方法可以在PC端和Android加载资源，但是不能修改其中的资源，也就是说只可以读不可以写。所以一般都放一些预设体，当从PC端打包到android端的时候，这个文件夹中的资源会把没有用的东西都自己舍去只留下项目中真正使用的部分，相当于做了一定的优化。

#### 2.第二种 直接放在项目根路径下来保存文件，把XML文件直接创建到Asset或者Asset目录下的其他文件夹中，在PC平台中可以使用Application.dataPath+路径 直接方法对应文件，但是在移动平台这样是不能访问的。Application.dataPath返回的其实就是Asset的根目录。

#### 3.第三种 在Asset目录下创建StreamingAssets文件夹，这个文件夹中存放的资源在PC端可以实现可读可写操作，但是在移动端只能只能读，不能写，当我们通关后想把一些信息存起来是不可以的。

#### string filepath = Application.dataPath +"/StreamingAssets"+"/my.xml";

#### 4.第四种persistentDataPath 该路径下的文件是存放在手机沙盒当中的，在手机端是可读可写的，当我们想要用一个文件来存放每关的信息时，这个方式是可以的，但是由于这个文件我们无法再开发中直接创建，所以在项目的文件数据管理模块中，应用代码来生成这个文件比如说XML、Json这样生成的文件可以在移动端自由读写。

Android平台

Application.dataPath /data/app/xxx.xxx.xxx.apk

Application.streamingAssetsPath jar:file:///data/app/xxx.xxx.xxx.apk/!/assets

Application.persistentDataPath /data/data/xxx.xxx.xxx/files

Application.temporaryCachePath /data/data/xxx.xxx.xxx/cache

IOS平台

Application.dataPath Application/xxx-xx/xxx.app/Data

Application.streamingAssetsPath Application/xxxxx-xxxx/xxx.app/Data/Raw

Application.persistentDataPath Application/x-xxxx-xxx/Documents

Application.temporaryCachePath Application/xxxxxx-xxx/Library/Caches

Windows Web Player

Application.dataPath file:///D:/MyGame/WebPlayer

(即导包后保存的文件夹，html文件所在文件夹)

Application.streamingAssetsPath

Application.persistentDataPath

Application.temporaryCachePath

/////////////////////////没写完，大概吧////////////////////////////