Remoter2.0

设计开发文档

又来了一队学渣

设计思路：

以 wifi 作为媒介，将手机接收到的传感器信息（包括触屏信息，方向传感器信息，加速度传感器信息）通过 UDP 协议传递给电脑，电脑端根据这些信息进行模拟，以省去使用键鼠玩游戏或额外购买游戏手柄的麻烦。

开发人员：

姓名	电子邮件	项目分工
周奕	13302010032@fudan.edu.cn	成员协调、编程
郭一鸣	13302010020@fudan.edu.cn	产品演示、界面设计
张聪	13302010010@fudan.edu.cn	创意设计、编程
何天成	htc550605125@gmail.com	技术统筹、技术指导
田添星	13302010006@fudan.edu.cn	创意设计、软件测试

连接协议：

com.yag.remoter.messages

所有的连接协议都存在于该包内，并都实现标记接口 **Message**（继承自 Serialilable 接口，用于序列化信息使得信息可以传递）。

KeyMessage

int key 保存按键内容，使用 java.awt.event.KeyEvent 中的 KEY 对应的值。

int condition 保存按键状态，使用 java.awt.event.KeyEvent 中的按键状态值，标志按键属于按下还是松开状态。

MotionMessage

float dx,dy 保存触屏上手指掠过的横坐标、纵坐标差值。

MouseClickMessage

int key 保存按键的信息（左键，右键）。

EndingMessage

空类，标志连接中断，用于通知电脑端关闭服务。

电脑端结构：

com.yag.remoter.MainFrame

界面类，用于连接移动端，并显示当前连接情况。每次点击连接，开启新的线程调用 Core 类的方法。

com.yag.remoter.Core

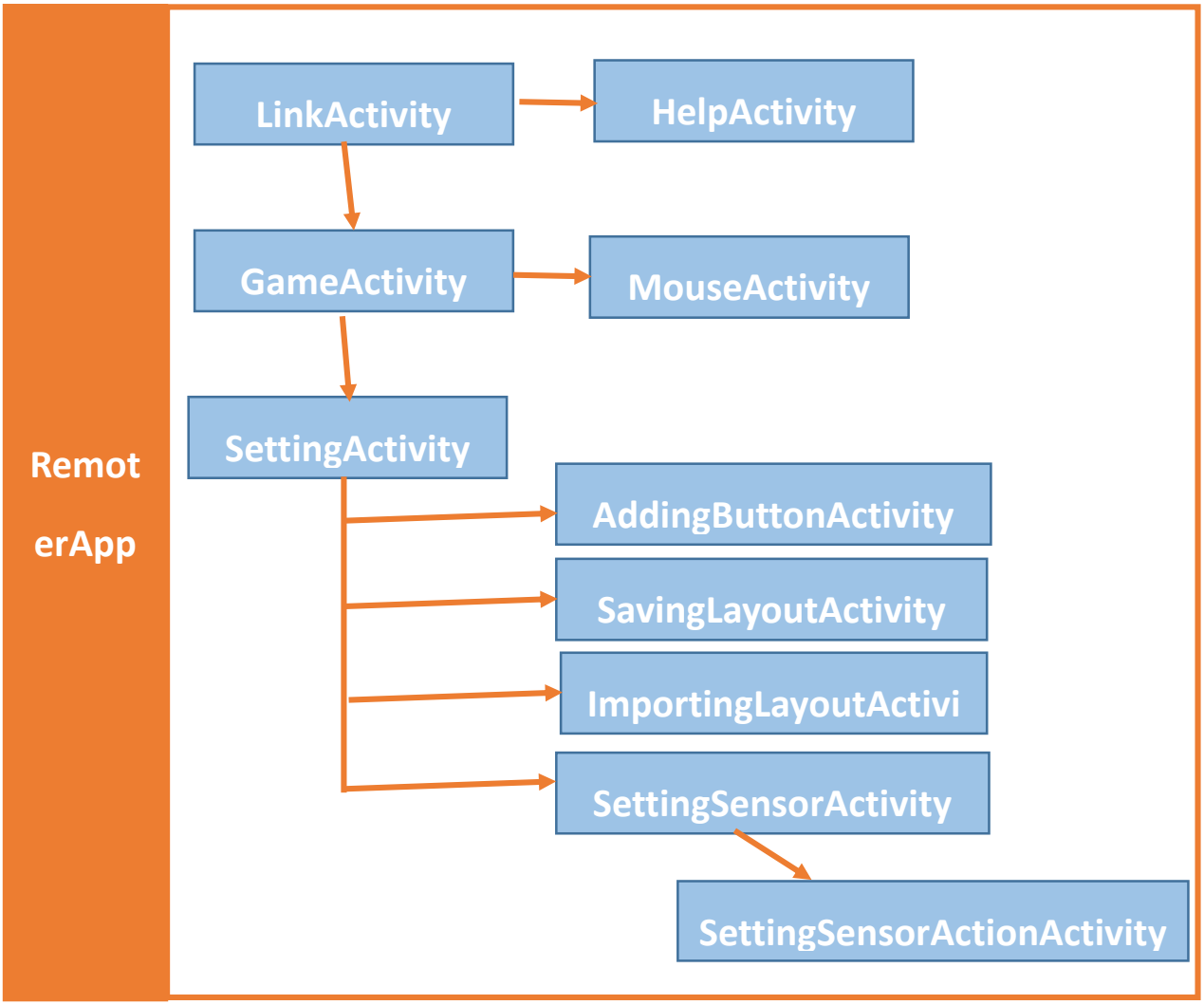
核心类，用于处理电脑端接受到的信息，使用 Robot 类，将各个 Message 转化为电脑操作。

void link() 网络连接方法，采用 UDP 连接协议，默认开启 12345 端口。。

void run() 主体运行方法，用于处理 Message 类。

String getIP() 返回当前计算机的 IP 地址，用于提供给手机端连接。

移动端结构：



RemoterApp

继承自 Application，用于保存网络连接、数据流等应用程序共享信息。

重要属性：

socket : DatagramSocket	UDP 连接
sendPacket : DatagramPacket	数据包
nameOfButton : String	新建的按钮的名称
addingButton : boolean	状态：新建按钮

deletingButton : boolean	状态：删除按钮
savingLayout : boolean	状态：保存布局
nameOfLayout : String	保存/导入的布局的名称
importingLayout : boolean	状态：导入布局
movingButton : boolean	状态：移动按钮
width : int	屏幕宽度
height : int	屏幕高度
upAndDown : String	方向传感器前后开关
leftAndRight : String	方向传感器左右开关
accelLeft : String	向左加速的 Key 值
accelRight : String	向右加速的 Key 值
accelUp : String	向上加速的 Key 值
accelDown : String	向下加速的 Key 值

Rutton

继承自 Button，使用于 GameActivity。

重要属性：

name:String	按钮名字

LinkActivity

连接界面，包含连接、帮助、退出三个功能。“连接”成功则跳转至 GameActivity，“帮助”跳转到 HelpActivity，用于显

示连接帮助。网络连接协议为 UDP 协议，以保证连接的快速、稳定，防止断线后出现异常。取得的连接保存在 RemoterApp。

GameActivity

游戏控制主界面。包含传感器信息的处理、按键信息的处理，用户界面的刷新等。

重要属性：

bg:HashMap<Rutton,LayoutParams>　当前按键布局

重要方法：

init() *执行初始化工作。*

初始化工作包括建立网络连接，初始化跳转按钮、设置按钮、方向按钮等。

onResume() *执行界面传感器注册、界面刷新等功能。*

1，如果 isMovingButton 为 true，此时按钮的触屏监听器将根据手势滑动而控制按钮移动。 游戏界面同时出现“确认”按钮，用于通知按钮移动完毕，即将 isMovingButton 置 false。

2，如果 isSavingLayout 为 true，则将界面保存在 HashMap 中，并序列化为输出流保存在本地。

3，如果 isImportingLayout 为 true，则从 RemoterApp 中取得

NameOfLayout，据此从本地读取布局文件，然后解析布局，写入界面。

onStop() 注销传感器

onTouchEvent() 拥有按钮移动的职能。

如果 isAddingNewButton 为 true，则从 RemoterApp 中取得 NameOfButton 并监听屏幕触摸事件，在触摸处添加 Button。

refresh() 用于刷新界面。

根据 HashMap 中的布局信息（Rutton 与 LayoutParams），清空当前界面，并刷新界面。

内部类：

MyOnTouchListener 按钮监听器

按钮监听器，根据 RemoterApp 中的状态参数确定当前动作。如果 isDeletingButton 为 true，则删除按钮；如果 isMovingButton 为 true，则移动按钮；如果二者皆非，则发送按钮信息。

MouseActivity

鼠标界面，含有模拟触摸版和左右按键。

HelpActivity

帮助界面，用于显示帮助信息。

SettingActivity

设置界面，包含添加/删除按钮，保存/导入布局，传感器设置，断开连接等功能。可跳转至 AddingButtonActivity、SavingLayoutActivity、ImportingLayoutActivity、SettingSensorActivity

AddingButtonActivity

添加按钮界面，继承自 ListActivity，用于在桌面触屏添加按钮。使用 KeyMessage 中定义的可选按钮 KEYS[]，设置 RemoterApp 中的 isAddingNewButton 参数。

SavingLayoutActivity

保存布局界面，用于保存当前按钮布局，设置 RemoterApp 中的 isSavingLayout 状态。将布局 HashMap 序列化，写入本地存储，以便下一次调用。

ImportingLayoutActivity

导入布局界面，设置 RemoterApp 中的 isImportingLayout 状态。用于导入存储过的布局，将布局文件解析，保存到布局 HashMap 中。

SettingSensorActivity

传感器设置界面，用于设置方向传感器、加速度传感器的信息。方向传感器用于控制前后左右（WASD），加速度传感器用于定制特殊按键（加速向前/后/左/右）。跳转到SettingSensorActionActivity。

SettingSensorActionActivity

传感器定制功能，用于显示可定制的按键，设置RemoterApp 中的 upAndDown、leftAndRight、accelUp 系列参数。基本复用了 AddingButtonActivity 的代码。

遇到问题以及解决方案：

1、问题：小组成员无安卓开发经验。

解决：寒假期间春节前进行开发自学。

2、问题：如何在不同的界面间方便地传输数据。

解决：初步使用 Intent，但试用后发现性能并不够优越，在多界面间都需要传递数据时，比较混乱，后来经过讨论并查询网络资料，决定使用 Application（即项目中的 RemoterApp）解决。

3、问题：模仿书上的案例，使用 Socket 连接电脑时，总是无法成功连接。

解决：组内成员讨论、查阅网络资料，下载网络连接的案例，发现高版本（SDK>9）不允许在主线程中进行网络连接，而默认版本是 SDK=18，于是将版本降低为 SDK=8。

4、问题：无线网连接时，常常出现连接不成功后断线无法重连，必须重启电脑端与手机端的软件，且有部分延迟。

解决：参阅图书馆资料，使用 UDP 协议替代 TCP 协议，由于 UDP 协议是无连接的协议，所以更快，也可以自动断线重连，在本项目中更稳定。同时，软件具备了多人对战的能力。服务端所能服务客户端的比例，从一对一向一对多转变。

5、问题：XML 难以实现定制按钮的功能。

解决：成员内讨论，查阅资料，放弃 XML 写组件，而使用 LayoutParams 通过硬代码实现。

6、问题：手机向某个方向甩动时，都会先加速再减速，所以无论如何使用加速度传感器，总是会伴着相反方向的干扰，如何选取最佳的传感器临界感应值。

解决：大量重复试验根据经验找到相对比较好的方案。

7、问题：测试极品飞车时，无法将手机翻动的数据传入电脑实现对电脑的控制，而↑↓←→四个虚拟按键也无法数据传入，而其他按键均可。

解决：经小组成员紧急商量，将方向传感器自动生成的KeyMessage 由↑↓←→四个按键换为游戏常用的 WASD 键位。

8、问题：后期测试发现无法实现多点触控。

解决：查阅网络资料，得出低版本不允许多点触控，而AndroidManifest 使用了低版本以保证网络可以连接。但是此时如果使用高版本则无法连接电脑，二者不可得兼，再次查阅网络资料，如果要在高版本中连接网络，必须创建新线程或者采取特殊代码的声明。项目中采取了后者。