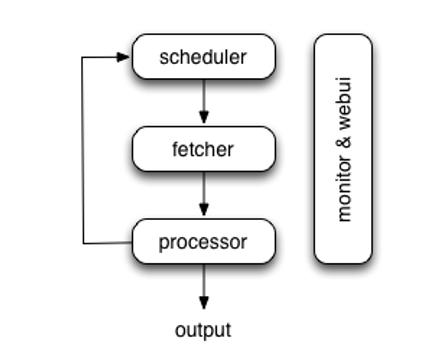
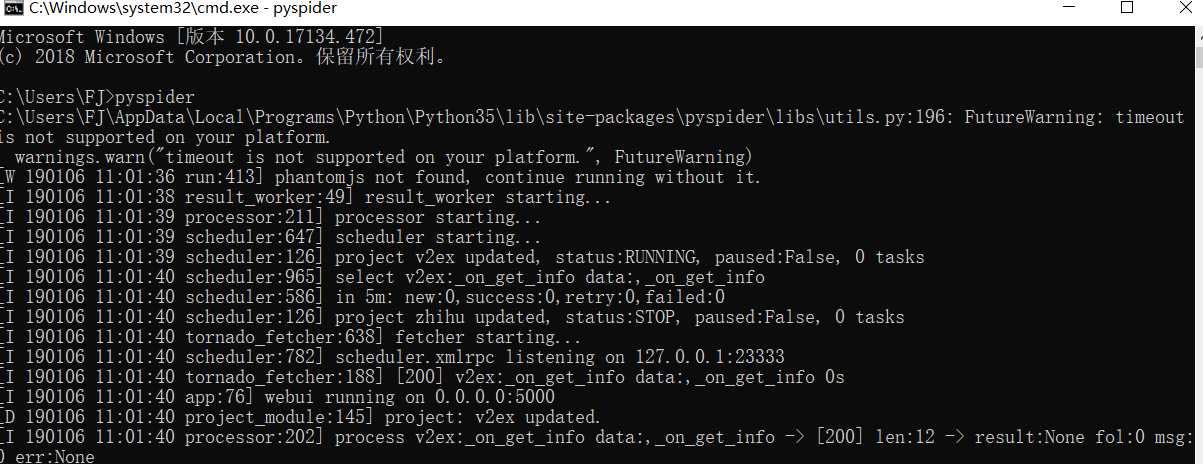
# Pyspider爬取网站信息

框架架构如下图所示

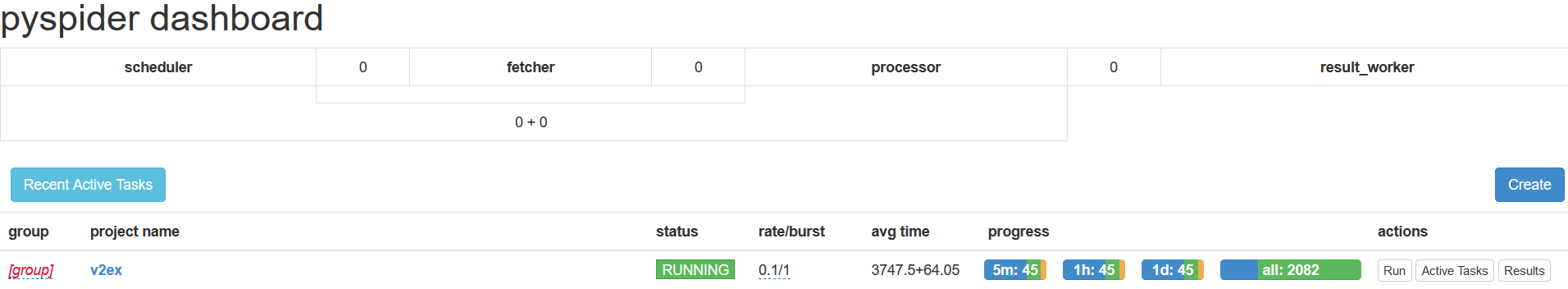


本次实践的目标是爬取网站V2EX网站的问题及评论信息，首先将网站的首页设置为爬虫的入口地址，分析所有版块的地址特点，用css选择器找出所有版块的地址链接，然后进入这些版块，再从版块下的标题列表中进入网站详情页，从中提取出问题和评论信息，将其下载下来存储到本地数据库中。

在cmd中直接输入PySpider启动。

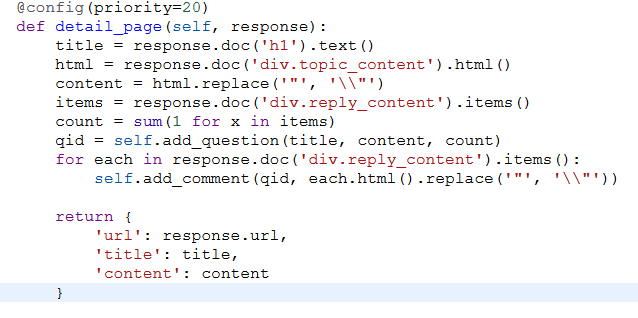


然后在PySpider中创建项目。

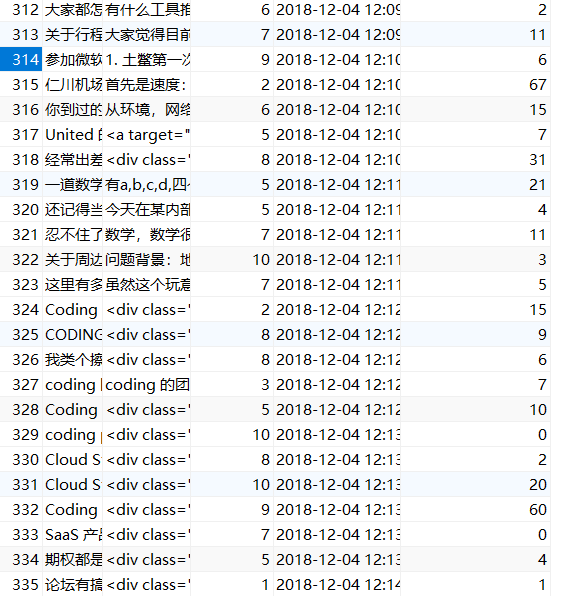


·具体代码及实现如下所示。









# 手机实现对鼠标的控制

## 1.需求分析

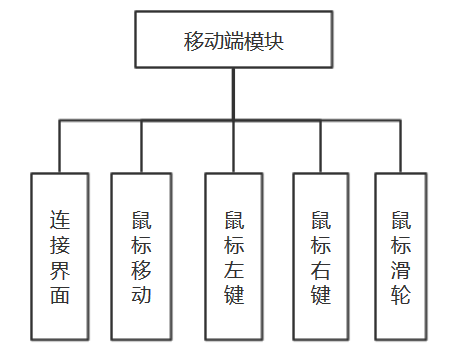
#### 连接界面功能

用户在启动应用程序前，首先要保证pc端网络正常连接，设置好手机的移动网络，保证手机能访问电脑的ip。用户启动应用程序后，首先出现的是连接界面。该界面主要包括指定服务器的ip地址，端口号，序的帮助信息和退出程序。向服务器发送连接请求，建立通讯连接，跳转到模拟鼠标界面。

#### 鼠标模拟操作功能

鼠标功能模块是在Android系统上添加对鼠标的软件支持。在鼠标功能模块界面上设计出一块触摸测试区域，用该区域来模拟手机屏幕的触摸屏功能。用户手指在触摸区域滑动，通过滑动事件处理获取手指滑动前后的坐标差值,发送给服务器。而在服务器端绘制出一个鼠标的光标，光标随着客户端发送的鼠标差值而移动，当用户按下点击鼠标按键，启动服务器光标对于区域的应用或者组件,从而实现一个模拟鼠标的功能。

## 2.结构设计



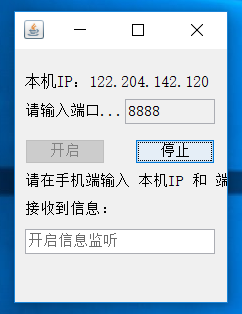
## 3.详细设计

客户端主要负责用户交互、监听与捕捉事件、与服务器端进行数据通信，向服务器发送请求和接收服务器的反馈信息。Android默认是没有鼠标支持的，但在Android系统会应用到其他的地方，如平板电脑、电视中，很多时候就需要鼠标的支持。鼠标的支持原理为：Android中有原生的触摸(Touch)事件，鼠标的各种事件都可以转换为各种Touch事件，将鼠标事件走向转换成Touch事件相关的路径就可以实现。

鼠标的滑动过程其实很简单，就是不断改变鼠标的位置坐标，然后重新将鼠标光标绘制在屏幕上。不过这种坐标的位置改变和绘制过程是通过一定逻辑来控制实现的。鼠标的移动就是通过改变坐标位置来实现的。改变了坐标再重新绘制，给我们的感觉是鼠标在移动。如果鼠标水平向左移动，X坐标减小；如果鼠标水平向右移动，X坐标增大。如果鼠标垂直向上移动，Y坐标减小；如果鼠标垂直向下移动，Y坐标增大。 在界面设计上，在界面顶端预留出一大快空白区域，作为模拟测试区域，采 用ListView控件来作为该区域。设置监听器，监听手指的活动。区域下方设计为鼠标控制键，当按下此按键时，相当于点击了鼠标键位。

## 4.系统实现

首先运行服务端代码，会显示出本机的IP地址，并设置一个端口进行监听。



在手机端输入服务端的IP地址和端口号进行连接



在主页面的上半部通过滑动手指可以控制鼠标移动，并实现了鼠标左右键点击和滑轮时间，手机根据不同事件发送对应字符串，服务器端解析字符串并调用相应的操作。



## 5.系统测试

经过测试，服务端可以正常启动，手机端输入正确的ip地址和端口号即能成功连接，通过滑动手机触摸屏可以控制鼠标移动，左右键点击和滑轮事件都得到了正确响。

具体代码如下

**public class** ControlActivity **extends** Activity {  
 **private static float** *mx* = 0; *// 发送的鼠标移动的差值* **private static float** *my* = 0;  
 **private static float** *lx*; *// 记录上次鼠标的位置* **private static float** *ly*;  
 **private static float** *fx*; *// 手指第一次接触屏幕时的坐标* **private static float** *fy*;  
 **private static float** *lbx* = 0; *// 鼠标左键移动初始化坐标* **private static float** *lby* = 0;  
 Handler **handler** = **new** Handler();  
 Runnable **leftButtonDown**;  
 Runnable **leftButtonRealease**;  
 Runnable **rightButtonDown**;  
 Runnable **rightButtonRealease**;  
  
 **private** FrameLayout **leftButton**;  
 **private** FrameLayout **rightButton**;  
   
 **private** String **volumedownkey** = **"leftButton"**;  
 **private** String **volumeupkey** = **"leftButton"**;  
   
 **private boolean flag**= **true**;*//屏蔽home键* **private** DatagramSocket **socket**;  
   
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 *//* ***TODO Auto-generated method stub* super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***control***);  
 **try** {  
 **socket** = **new** DatagramSocket();  
 } **catch** (SocketException e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
 initTouch();  
 }  
   
 @Override   
 **public void** onAttachedToWindow() {   
 */\*if(flag) {  
 this.getWindow().setType(WindowManager.LayoutParams.TYPE\_KEYGUARD);   
 } \*/* **super**.onAttachedToWindow();   
 }   
  
 **private void** initTouch() {  
 FrameLayout touch = (FrameLayout) **this**.findViewById(R.id.***touch***);  
  
 *// let's set up a touch listener* touch.setOnTouchListener(**new** View.OnTouchListener() {  
 **public boolean** onTouch(View v, MotionEvent ev) {  
 **if** (ev.getAction() == MotionEvent.***ACTION\_MOVE***)  
 onMouseMove(ev);  
 **if** (ev.getAction() == MotionEvent.***ACTION\_DOWN***)  
 onMouseDown(ev);  
 **if** (ev.getAction() == MotionEvent.***ACTION\_UP***)  
 onMouseUp(ev);  
 **return true**;  
 }  
 });  
  
 **leftButton** = (FrameLayout) **this**.findViewById(R.id.***leftButton***);  
 **leftButton**.setOnTouchListener(**new** View.OnTouchListener() {  
 **public boolean** onTouch(View v, MotionEvent ev) {  
 **if** (ev.getAction() == MotionEvent.***ACTION\_DOWN***) {  
 onLeftButton(**"down"**);  
 **handler**.post(**leftButtonDown**);  
 }  
 **if** (ev.getAction() == MotionEvent.***ACTION\_UP***) {  
 onLeftButton(**"release"**);  
 *lbx* = 0;  
 *lby* = 0;  
 **handler**.post(**leftButtonRealease**);  
 }  
 **if** (ev.getAction() == MotionEvent.***ACTION\_MOVE***)  
 moveMouseWithSecondFinger(ev);  
 **return true**;  
 }  
 });  
  
 **rightButton** = (FrameLayout) **this**.findViewById(R.id.***rightButton***);  
 **rightButton**.setOnTouchListener(**new** View.OnTouchListener() {  
 **public boolean** onTouch(View v, MotionEvent ev) {  
 **if** (ev.getAction() == MotionEvent.***ACTION\_DOWN***) {  
 onRightButton(**"down"**);  
 **handler**.post(**rightButtonDown**);  
 }  
 **if** (ev.getAction() == MotionEvent.***ACTION\_UP***) {  
 onRightButton(**"release"**);  
 **handler**.post(**rightButtonRealease**);  
 }  
 **return true**;  
 }  
 });  
  
 FrameLayout middleButton = (FrameLayout) **this** .findViewById(R.id.***middleButton***);  
 middleButton.setOnTouchListener(**new** View.OnTouchListener() {  
 **public boolean** onTouch(View v, MotionEvent ev) {  
 **if** (ev.getAction() == MotionEvent.***ACTION\_DOWN***)  
 onMiddleButtonDown(ev);  
 **if** (ev.getAction() == MotionEvent.***ACTION\_MOVE***)  
 onMiddleButtonMove(ev);  
 **return true**;  
 }  
  
 });  
  
 **this**.**leftButtonDown** = **new** Runnable() {  
 **public void** run() {  
 drawLeftButtonDown(**leftButton**);  
 }  
  
 **private void** drawLeftButtonDown(FrameLayout fl) {  
 fl.setBackgroundResource(R.drawable.***zuoc***);  
 }  
 };  
  
 **this**.**rightButtonDown** = **new** Runnable() {  
 **public void** run() {  
 drawButtonDown(**rightButton**);  
 }  
  
 **private void** drawButtonDown(FrameLayout fl) {  
 fl.setBackgroundResource(R.drawable.***youc***);  
 }  
 };  
  
 **this**.**leftButtonRealease** = **new** Runnable() {  
 **public void** run() {  
 drawLeftButtonRealease(**leftButton**);  
 }  
  
 **private void** drawLeftButtonRealease(FrameLayout fl) {  
 fl.setBackgroundResource(R.drawable.***zuo***);  
  
 }  
 };  
  
 **this**.**rightButtonRealease** = **new** Runnable() {  
 **public void** run() {  
 drawButtonRealease(**rightButton**);  
 }  
  
 **private void** drawButtonRealease(FrameLayout fl) {  
 fl.setBackgroundResource(R.drawable.***you***);  
  
 }  
 };  
  
 }  
  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 **super**.onCreateOptionsMenu(menu);  
 getMenuInflater().inflate(R.menu.***menu***, menu);  
 **return true**;  
 }  
  
 */\*\*  
 \* 捕捉菜单事件  
 \*/* @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **switch** (item.getItemId()) {  
  
 **case** R.id.***exit***:  
 doExit();  
 **return true**;  
 }  
 **return false**;  
 }  
  
 **private void** moveMouseWithSecondFinger(MotionEvent event) {  
 **int** count = event.getPointerCount();  
 **if** (count == 2) {  
 **if** (*lbx* == 0 && *lby* == 0) {  
 *lbx* = event.getX(1);  
 *lby* = event.getY(1);  
 **return**;  
 }  
 **float** x = event.getX(1);  
 **float** y = event.getY(1);  
 sendMouseEvent(**"mouse"**, x - *lbx*, y - *lby*);  
 *lbx* = x;  
 *lby* = y;  
 }  
 **if** (count == 1) {  
 *lbx* = 0;  
 *lby* = 0;  
 }  
  
 }  
  
 **private void** onMouseDown(MotionEvent ev) {  
 *lx* = ev.getX(); *// 当手机第一放入时 把当前坐标付给lx  
 ly* = ev.getY();  
 *fx* = ev.getX();  
 *fy* = ev.getY();  
 }  
  
 **private void** onMouseMove(MotionEvent ev) {  
 **float** x = ev.getX();  
 *mx* = x - *lx*; *// 当前鼠标位置 - 上次鼠标的位置  
 lx* = x; *// 把当前鼠标的位置付给lx 以备下次使用* **float** y = ev.getY();  
 *my* = y - *ly*;  
 *ly* = y;  
 **if** (*mx* != 0 && *my* != 0)  
 **this**.sendMouseEvent(**"mouse"**, *mx*, *my*);  
  
 }  
  
 **private void** onMouseUp(MotionEvent ev) {  
 **if** (*fx* == ev.getX() && *fy* == ev.getY()) {  
 sendMessage(**"leftButton:down"**);  
 sendMessage(**"leftButton:release"**);  
 }  
  
 }  
  
 **private void** sendMouseEvent(String type, **float** x, **float** y) {  
 String str = type + **":"** + x + **","** + y;  
 sendMessage(str);  
 }  
  
 **private void** onLeftButton(String type) {  
 String str = **"leftButton"** + **":"** + type;  
 sendMessage(str);  
  
 }  
  
 **private void** onRightButton(String type) {  
 String str = **"rightButton"** + **":"** + type;  
 sendMessage(str);  
 }  
  
 **private void** sendMessage(String str) {  
 **try** {  
 *// 首先创建一个DatagramSocket对象  
   
 // 创建一个InetAddree* InetAddress serverAddress = InetAddress.*getByName*(Settings.*ipnum*);  
 **byte** data[] = str.getBytes();  
 *// 创建一个DatagramPacket对象，并指定要讲这个数据包发送到网络当中的哪个地址，以及端口号* DatagramPacket packet = **new** DatagramPacket(data, data.**length**,  
 serverAddress, Settings.*scoketnum*);  
 *// 调用socket对象的send方法，发送数据* **socket**.send(packet);  
 } **catch** (Exception e) {  
 *//* ***TODO Auto-generated catch block*** e.printStackTrace();  
 }  
 }  
  
 **private void** onMiddleButtonDown(MotionEvent ev) {  
 *ly* = ev.getY();  
  
 }  
  
 **private void** onMiddleButtonMove(MotionEvent ev) {  
 *// count++;* **float** y = ev.getY();  
 *my* = y - *ly*;  
 *ly* = y;  
 **if** (*my* > 3 || *my* < -3) { *// 减少发送次数 滑轮移动慢点* String str = **"mousewheel"** + **":"** + *my*;  
 sendMessage(str);  
 }  
  
 }  
   
 **private void** downsetting(){  
 **final** String[] items = {**"鼠标左键"**,**"鼠标右键"**};  
 **new** AlertDialog.Builder(**this**)   
 .setTitle(**"选择按键"**) *//此处 this 代表当前Activity* .setItems(items, **new** DialogInterface.OnClickListener() {   
 **public void** onClick(DialogInterface dialog, **int** item) {   
 Toast.*makeText*(getApplicationContext(), items[item], Toast.***LENGTH\_SHORT***).show(); *//将选中的文本内容按照土司提示 方式显示出来, 此处的getApplicationContext() 得到的也是当前的Activity对象，可用当前Activity对象的名字.this代替（Activity.this）* **switch** (item) {  
 **case** 0:  
 **volumedownkey**=**"leftButton"**;  
 **break**;  
 **case** 1:  
 **volumedownkey**=**"rightButton"**;  
 **break**;   
  
  
 }  
 }   
 }).show();*//显示对话框* }  
   
 **private void** upsetting(){  
 **final** String[] items = {**"鼠标左键"**,**"鼠标右键"**};  
 **new** AlertDialog.Builder(**this**)   
 .setTitle(**"选择按键"**) *//此处 this 代表当前Activity* .setItems(items, **new** DialogInterface.OnClickListener() {   
 **public void** onClick(DialogInterface dialog, **int** item) {   
 Toast.*makeText*(getApplicationContext(), items[item], Toast.***LENGTH\_SHORT***).show(); *//将选中的文本内容按照土司提示 方式显示出来, 此处的getApplicationContext() 得到的也是当前的Activity对象，可用当前Activity对象的名字.this代替（Activity.this）* **switch** (item) {  
 **case** 0:  
 **volumeupkey**=**"leftButton"**;  
 **break**;  
 **case** 1:  
 **volumeupkey**=**"rightButton"**;  
 **break**;   
  
 }  
 }   
 }).show();*//显示对话框* }  
   
 @Override  
 **public boolean** onKeyDown(**int** keyCode, KeyEvent event) {  
  
 **if** (keyCode == KeyEvent.***KEYCODE\_VOLUME\_DOWN***) {  
 **if**(**volumedownkey**.equals(**"leftButton"**)){  
 sendMessage(**"leftButton:click"**);  
   
 } **else if**(**volumedownkey**.equals(**"rightButton"**))  
 sendMessage(**"rightButton:click"**);  
 **else** sendMessage(**"keyboard:key,"**+**volumedownkey**+**",click"**);  
 **return true**;  
  
 } **else if** (keyCode == KeyEvent.***KEYCODE\_VOLUME\_UP***) {  
  
 **if**(**volumeupkey**.equals(**"leftButton"**))  
 sendMessage(**"leftButton:click"**);  
 **else if**(**volumeupkey**.equals(**"rightButton"**))  
 sendMessage(**"rightButton:click"**);  
 **else** sendMessage(**"keyboard:key,"**+**volumeupkey**+**",click"**);  
 **return true**;  
  
 }**else if**( keyCode== KeyEvent.***KEYCODE\_HOME***){  
 **return true**;  
 } **else if**( keyCode== KeyEvent.***KEYCODE\_BACK***){  
 **return true**;  
 }   
   
 **else** {  
  
 **return super**.onKeyDown(keyCode, event);  
  
 }  
  
 }  
   
 *// 退出程序* **protected void** doExit() {  
 **new** AlertDialog.Builder(**this**)  
 .setMessage(**"程序退出"**)  
 .setPositiveButton(**"确认"**,  
 **new** DialogInterface.OnClickListener() {  
 **public void** onClick(  
 DialogInterface dialoginterface, **int** i) {  
 finish();  
 }  
 })  
 .setNeutralButton(**"取消"**,  
 **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface arg0, **int** arg1) {  
 }  
  
 }).show();  
  
 }  
  
}