

使用Python语言实现 Appium自动化测试

APPIUM WITH PYTHON

DAY10

内容

上午	09:00 ~ 09:30	模拟操作2
	09:30 ~ 10:20	
	10:30 ~ 11:20	
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



模拟操作2

模拟操作2

手势操作

TouchAction与MultiAction

轻触

短按

长按

滑动

连续滑动

多点触控

放大和缩小

放大

缩小

手势操作



TouchAction与MultiAction

- TouchAction与MultiAction：模拟手势操作



轻触



短按



放大



多点触控



缩小



等待



长按



滑动

TouchAction与MultiAction (续1)

- TouchAction : 模拟手势操作 , 例如手指轻触屏、短按、长按、按住等待、移动手指、释放手指等操作 , perform执行被调用时执行该动作。
- MultiAction : MultiAction是针对多点触控操作的 , 是TouchAction的一个补充模块 , MultiAction对象是手势动作 (TouchAction) 的集合。
- 多点触控手势只有两种方法 , 即添加和执行 :
 - add用于添加另一个手势操作到动作集合。
 - 当perform执行被调用时 , 添加到多点触控的所有手势动作都被发送到App , 并执行 , 就像它们同时发生一样。
 - appium首先执行所有手势动作的第一个事件 , 然后执行第二个 , 等等。



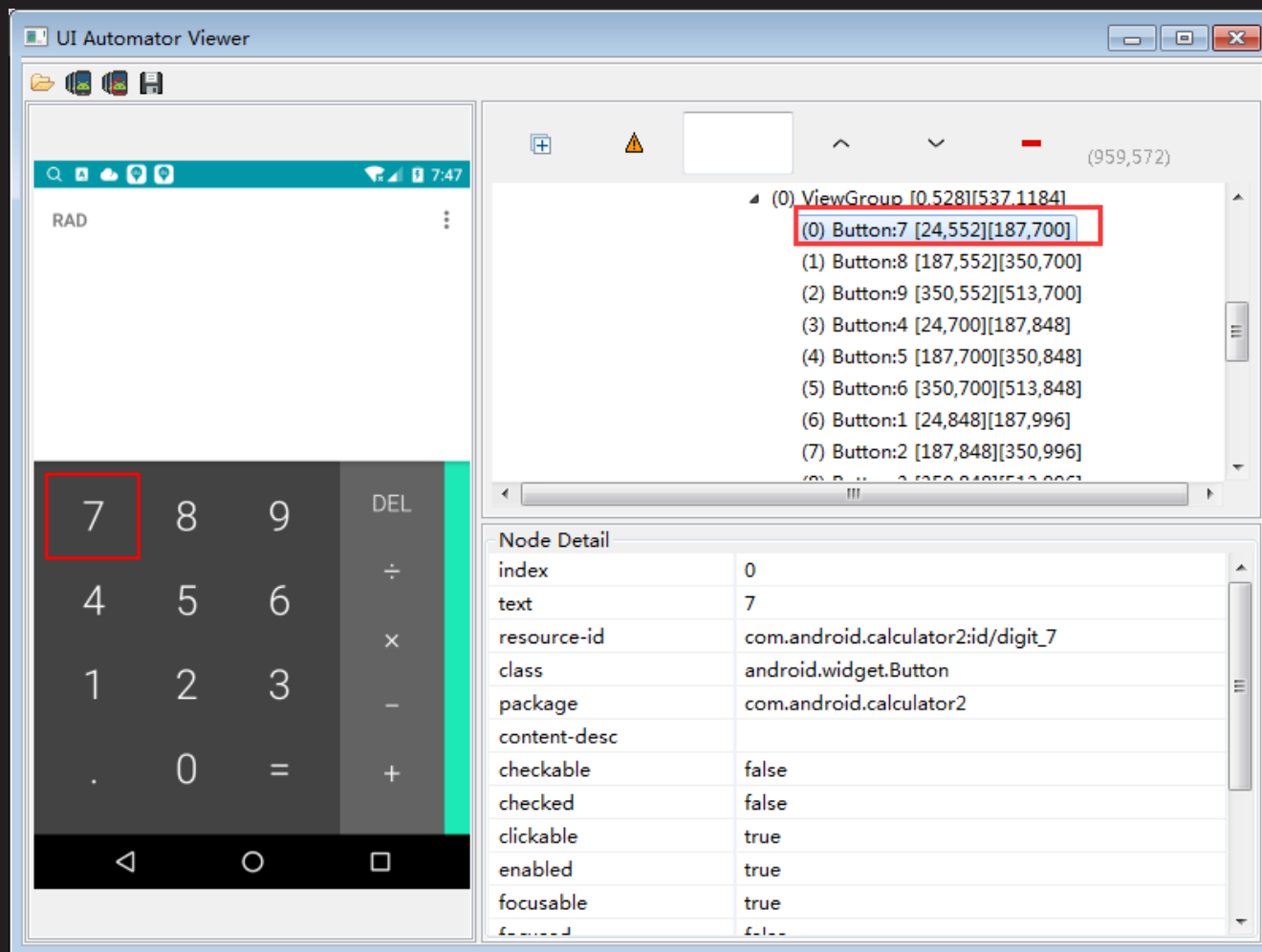
轻触

- 模拟手指轻触屏幕上指定的控件，或指定坐标点位置，或指定控件中偏移位置，使用tap方法。
- 用法一：在控件的中心点上轻触一下
 - tap(WebElement el)
- 用法二：在 (x,y) 点轻触一下
 - tap(int x, int y)
- 用法三：以控件el的左上角为基准，x轴向右移动x单位，y轴向下移动y单位，在该点上轻触
 - tap(WebElement el, int x, int y)



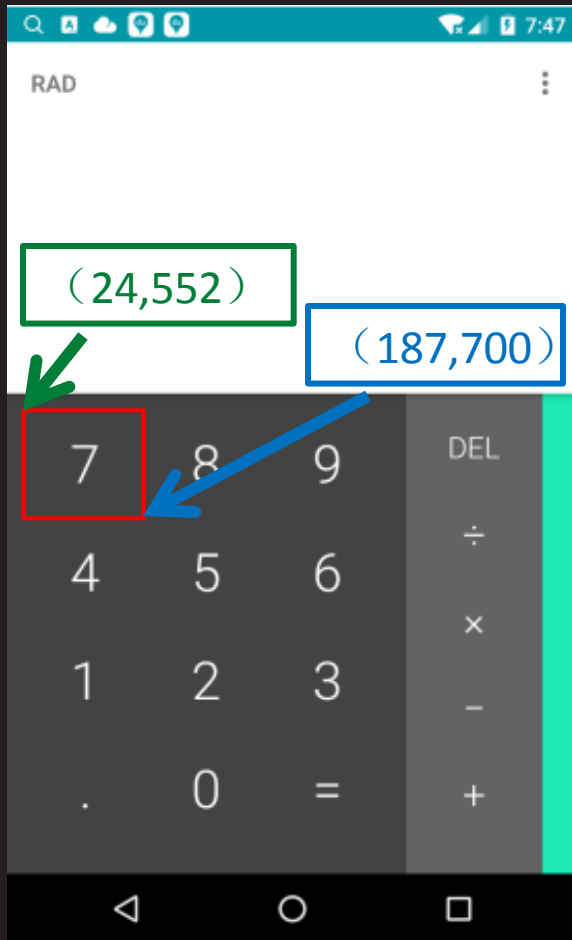
轻触（续1）

- 示例：模拟手指轻触屏幕上7按钮



轻触（续2）

- 示例用法一：轻触页面元素（默认在中间位置）



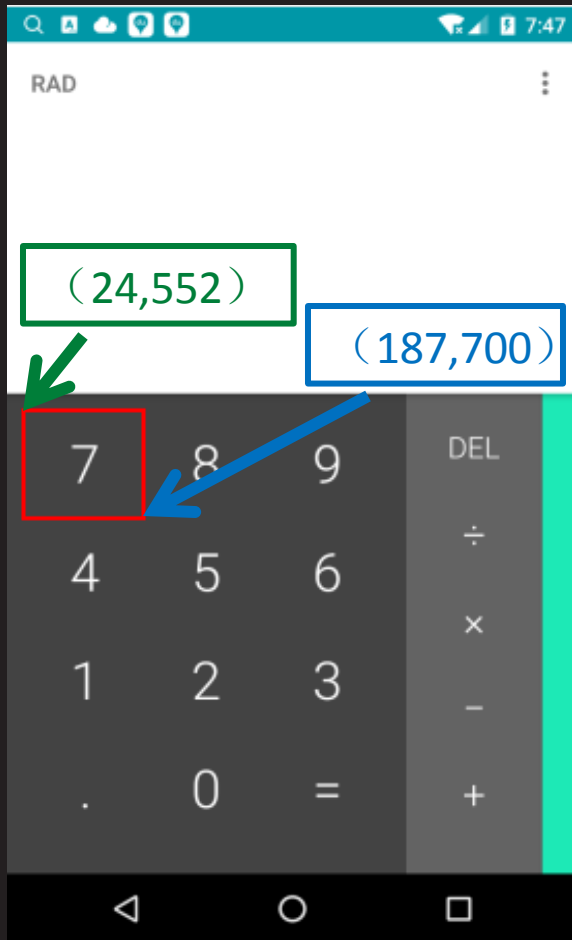
轻触按钮7的中间点位置

```
num7=driver.find_element_
by_id("com.android.calculat
or2:id/digit_7")
```

```
TouchAction(driver).tap(num
7).perform()
```

轻触（续3）

- 示例用法二：按坐标位置轻触页面元素的中间点位置



轻触按钮7的中间点位置

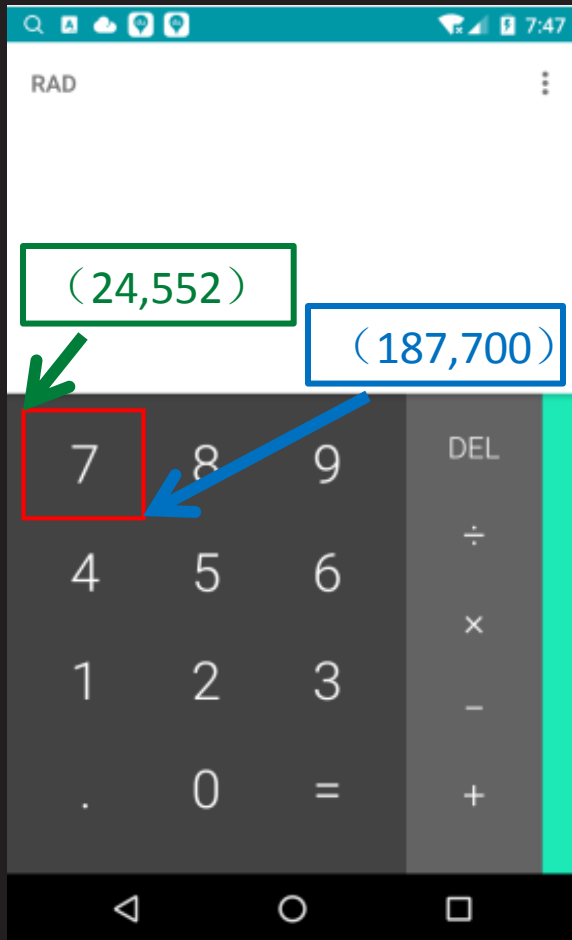
```
num7=driver.find_element_
by_id("com.android.calculat
or2:id/digit_7")
```

```
TouchAction(driver).tap(x=2
4+(187-24)/2,y=552+(700-
552)/2).perform()
```

注意：坐标不能超过屏幕分辨率xy值

轻触（续4）

- 示例用法三：轻触页面元素的指定偏移量的中间点位置



轻触按钮7的中间点位置

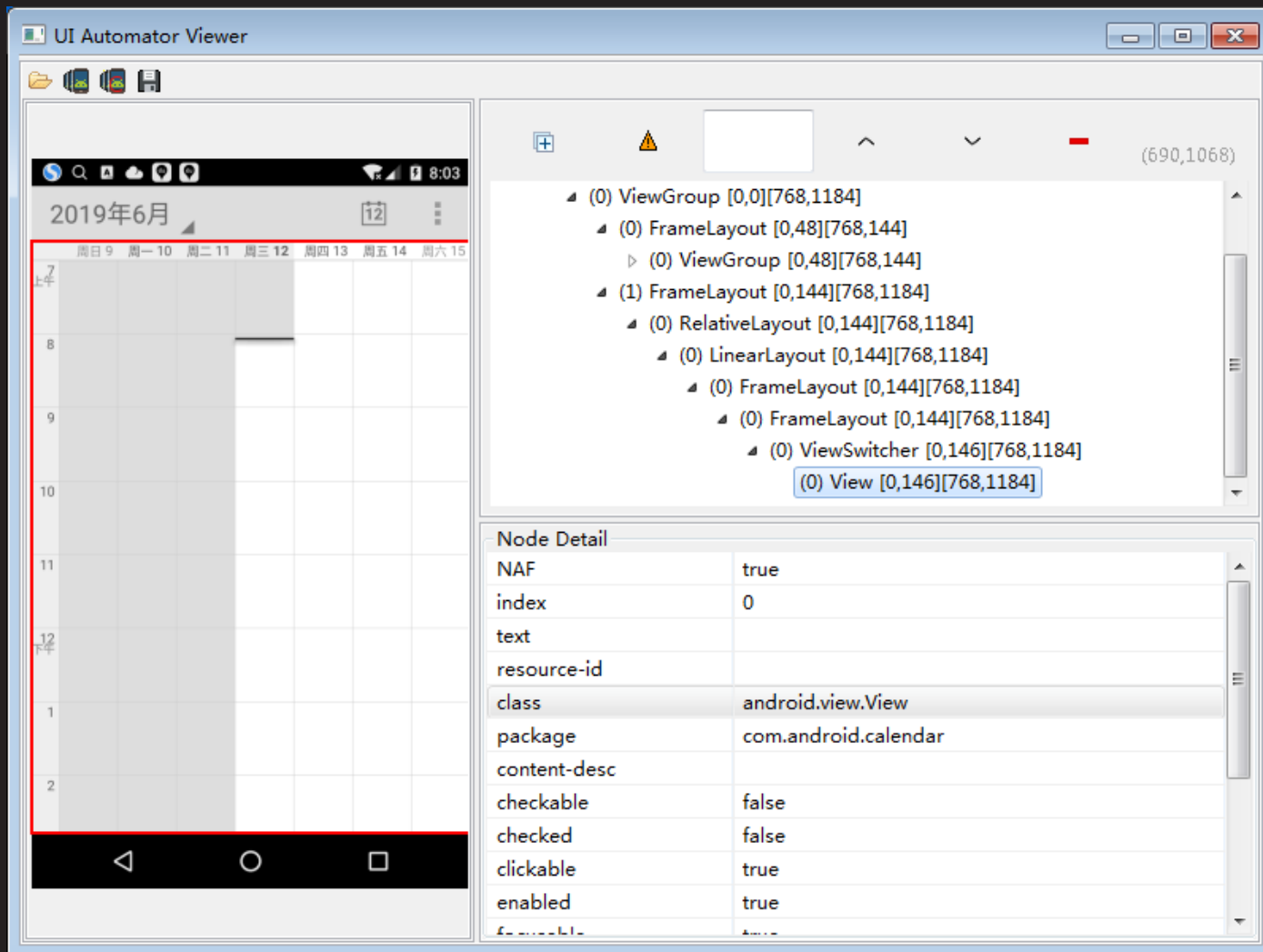
```
num7=driver.find_element_
by_id("com.android.calculat
or2:id/digit_7")
```

```
TouchAction(driver).tap(num7,(187-24)/2,(700-552)/2).perform()
```

注意：坐标不能超过实际页面元素的最大偏移范围

短按

- 短按：press方法



短按（续1）

- 示例用法一：短按日历中指定日期的单元格



```
# 日期单元格(第2行第6列)
a=driver.find_element_by_class_name("android.view.View")
TouchAction(driver).press(a).perform()
```

说明：位置按照x和y坐标值来确定位置
注意：坐标不能超过实际范围



短按（续2）

- 示例用法二：短按日历中指定日期的单元格



```
# 日期单元格(第2行第6列)
a=driver.find_element_by_class_name("android.view.View")
TouchAction(driver).press(x=(768-0)/7*6,y=(1184-146)/8*2).perform()
```

说明：位置按照x和y坐标值来确定位置
注意：坐标不能超过实际范围

短按（续3）

- 示例用法三：短按日历中指定日期的单元格



```
# 日期单元格(第2行第6列)
a=driver.find_element_by_class_name("android.view.View")
TouchAction(driver).press(a,
(768-0)/7*6,(1184-146)/8*2).perform()
```

说明：位置按照x和y坐标值来确定位置
注意：坐标不能超过实际范围

- 使用TouchAction里的方法，长按某元素位置。



说明：

- 1、通过driver可以直接调出long_press_keycode方法，但是这个方法是长按手机上某个按钮。
- 2、长按某个元素或者长按屏幕上某个点，就需要用到TouchAction里面提供的long_press操作了



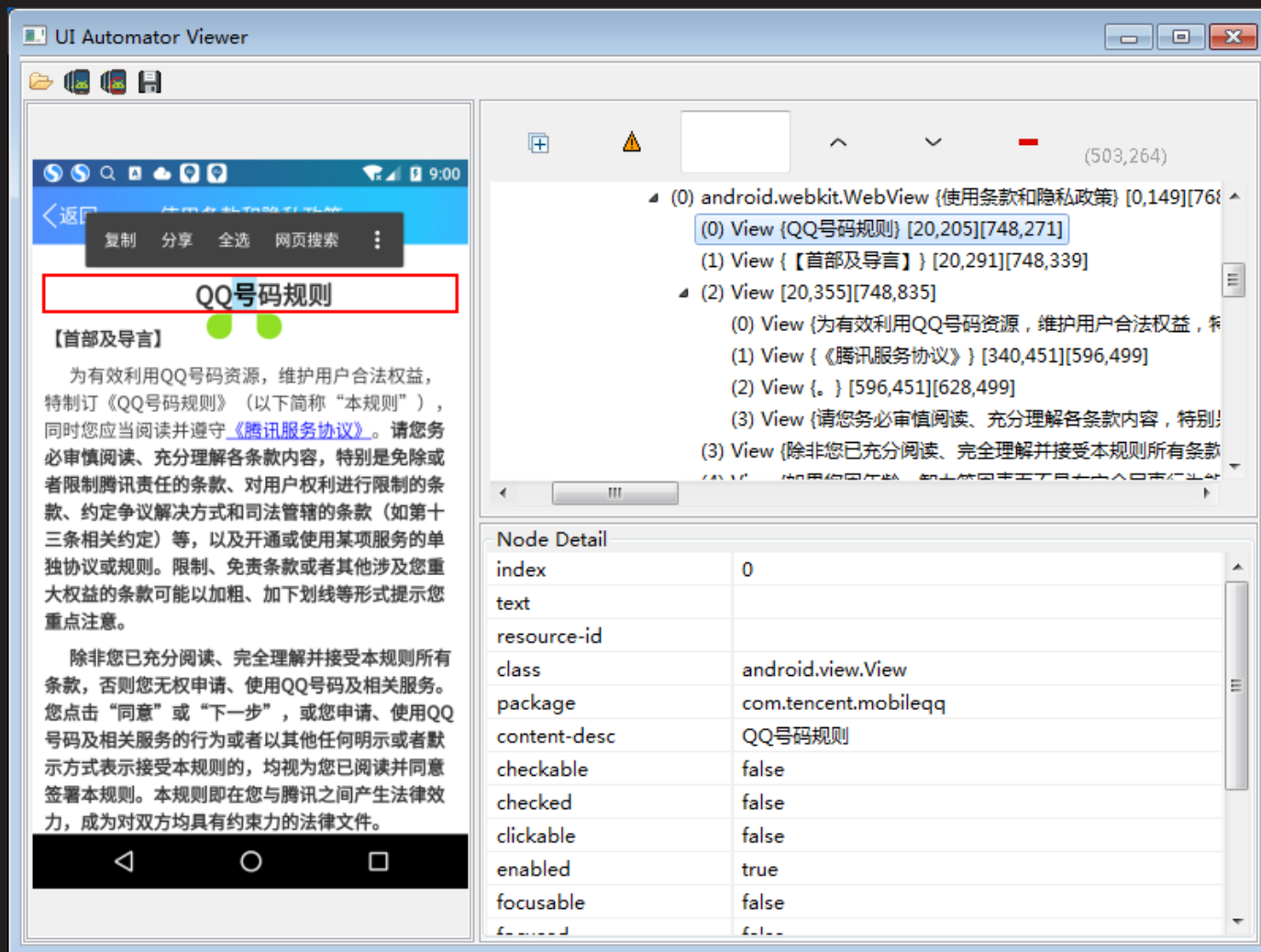
长按（续1）

- 长按：long_press方法
- 注意：坐标可以省略代表长按页面元素中心点，如果有坐标值作为参数，不是绝对坐标，是偏移量。
- 示例：
 - # 长按标题文字中间位置
t=driver.find_element_by_accessibility_id("QQ号码规则")
 - TouchAction(driver).long_press(t).perform()
 - TouchAction(driver).long_press(x=(748-20)/2,y=(271-205)/2).perform()
 - TouchAction(driver).long_press(t,(748-20)/2,(271-205)/2).perform()



长按 (续2)

- 长按：long_press方法



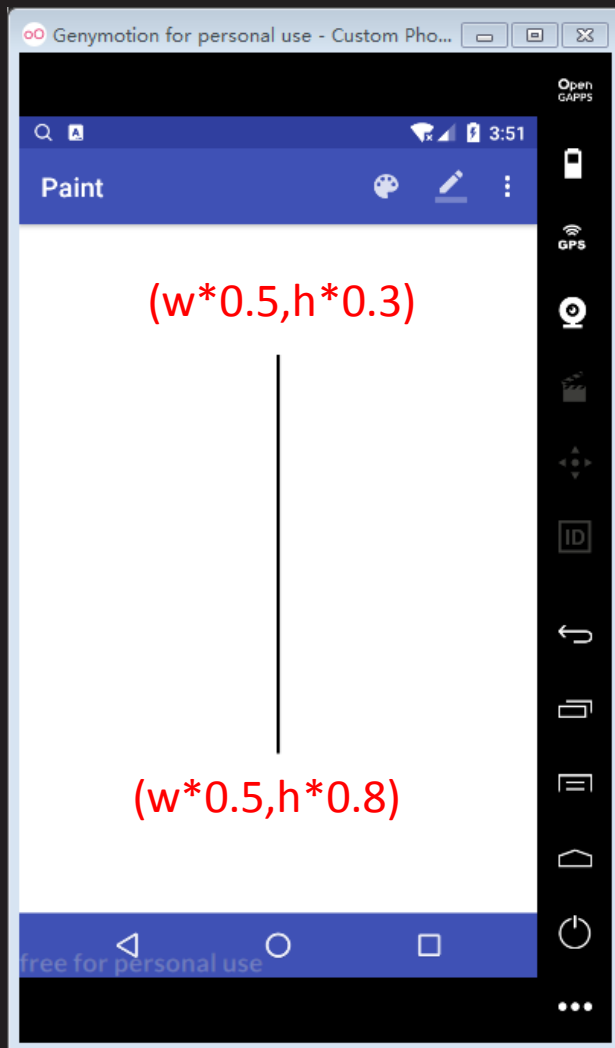
滑动

- 使用TouchAction里的方法，从某点滑到某点后释放。
 - 步骤1：短按开始点坐标位置（press）
 - 步骤2：等待1000毫秒（wait）
 - 步骤3：移动到指定坐标位置（move_to）
 - 步骤4：等待1000毫秒（wait）
 - 步骤5：释放（release）
 - 步骤6：执行这一系列动作（perform）
- 说明：如果连续滑动，可以反复执行步骤3和步骤4



滑动（续1）

- 示例1：在画图板里，从上向下滑动一条直线



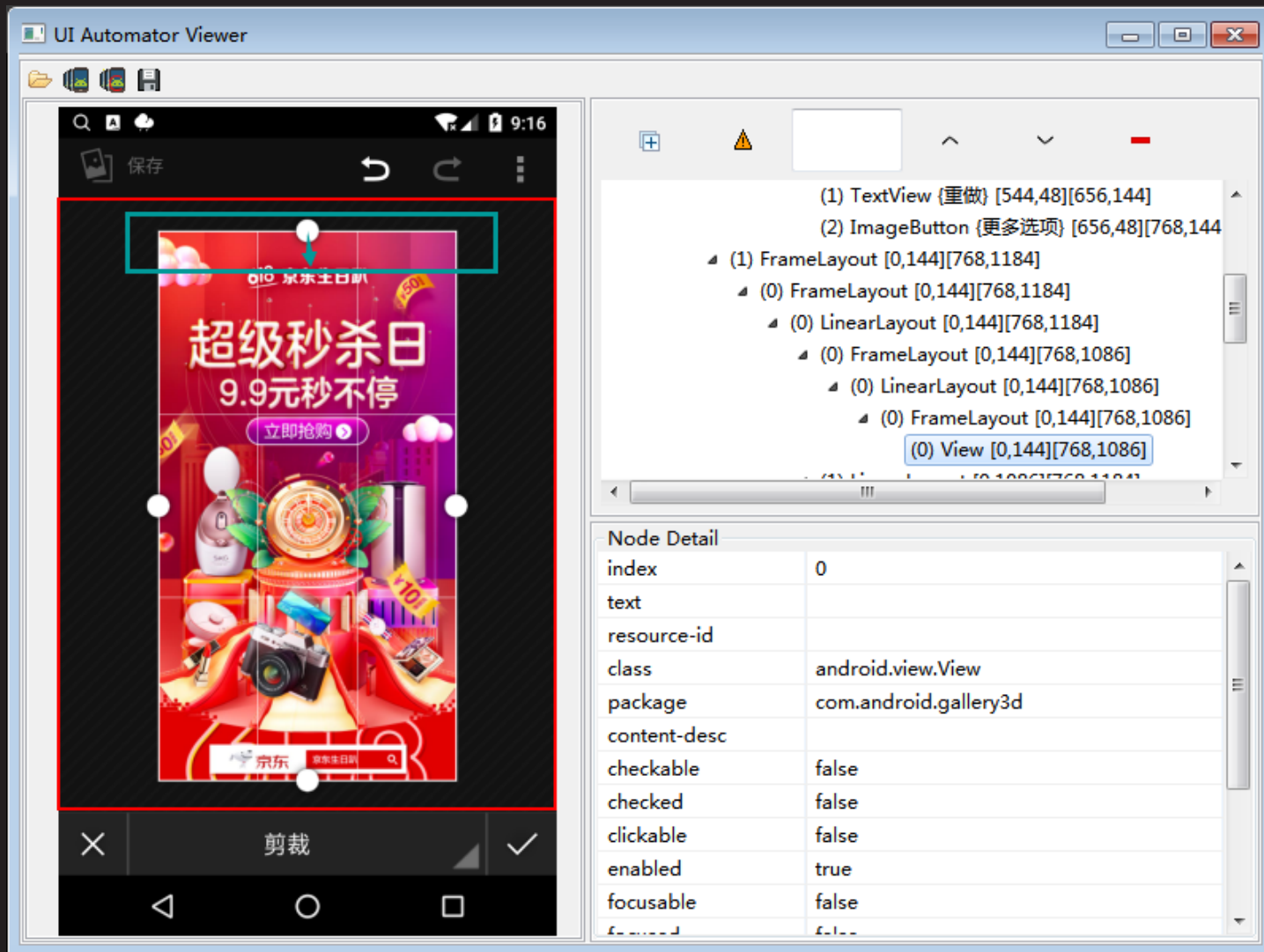
滑动（续2）

- 示例1代码：从上向下滑动一条直线
 - from appium.webdriver.common.touch_action import TouchAction
 -
 - w = driver.get_window_size()['width']
 - h = driver.get_window_size()['height']
 - TouchAction(driver).press(x=w*0.5, y=h*0.3).wait(1000).move_to(x=w*0.5, y=h*0.8).wait(1000).release().perform()



滑动（续3）

- 示例2：向下滑动400px来，修改图库中图片的高度



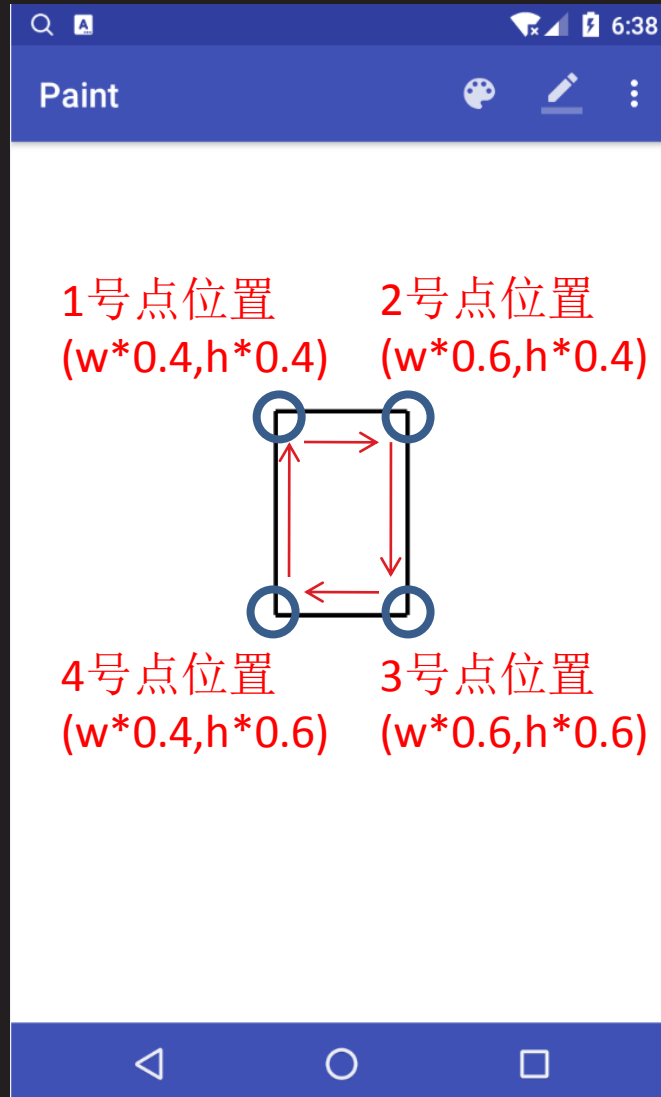
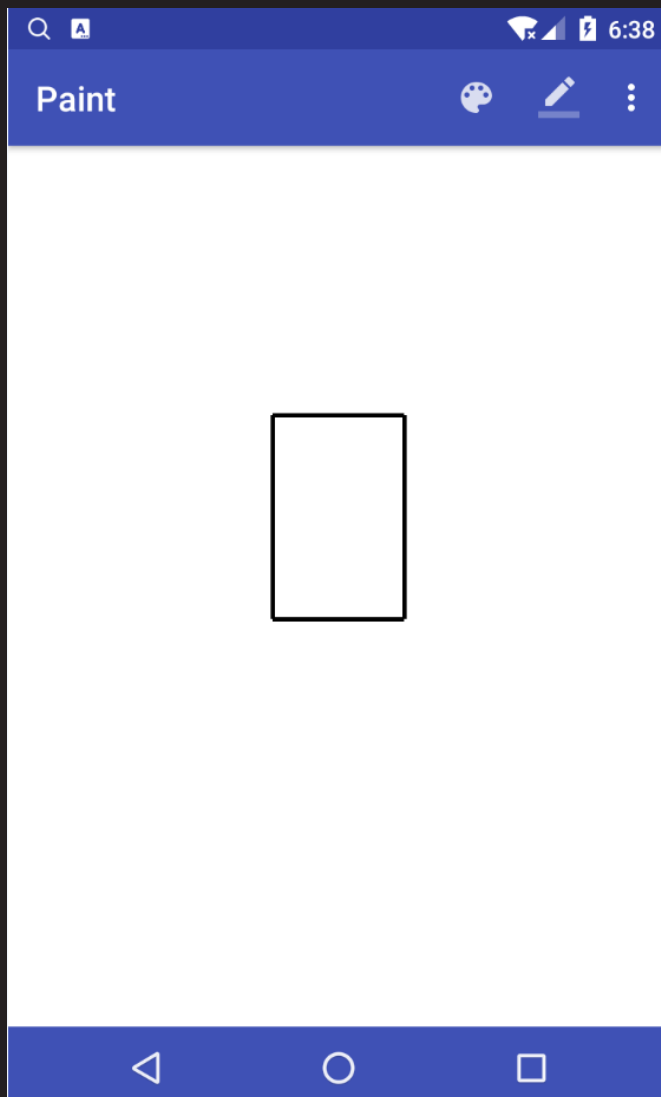
滑动（续4）

- 示例2代码：向下滑动400px
 - from appium.webdriver.common.touch_action import TouchAction
 -
 - w = driver.get_window_size()['width']
 - h = driver.get_window_size()['height']
 - TouchAction(driver).press(x=768/2, y=160).wait(1000).move_to(x=768/2, y=200).wait(1000).release().perform()



连续滑动

- 在画图板中，画中心区域的一个长方形（1-2-3-4-1）



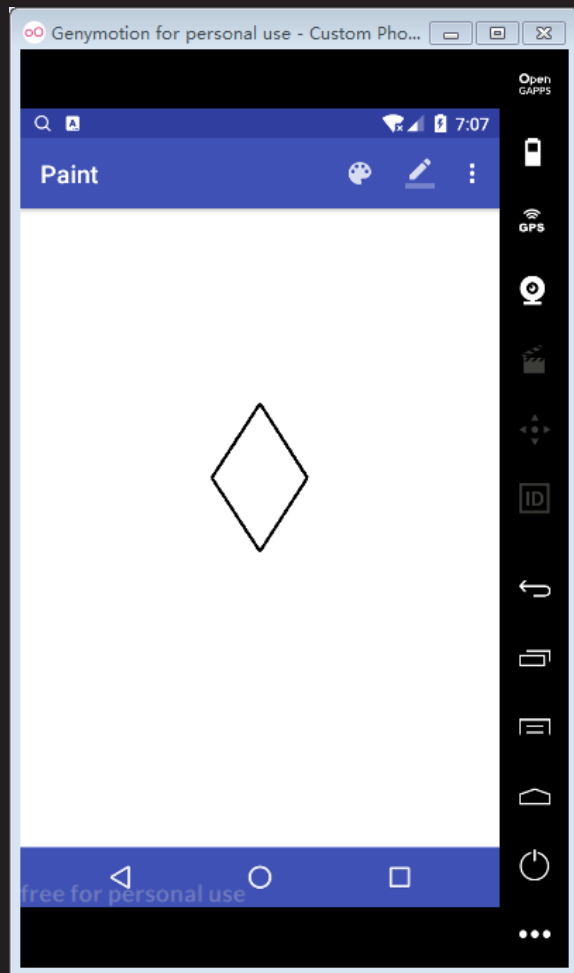
连续滑动（续1）

- 画长方形代码示例：
 - from appium.webdriver.common.touch_action import TouchAction
 -
 - w = driver.get_window_size()['width']
 - h = driver.get_window_size()['height']
 - a3=TouchAction(driver).press(x=w*0.4, y=h*0.4).wait(1000)
 - for i,j in zip([6,6,4,4],[4,6,6,4]):
 - a3.move_to(x=w*0.1*i, y=h*0.1*j).wait(1000)
 - a3.release()
 - a3.perform()



画菱形

- 在画图板中，画中心区域的一个菱形



多点触控

- 思考：在手机屏幕上多个手指同时操作，如何实现？
- 例如：水果忍者游戏

知识讲解



多点触控（续1）

- 多点触控是在同一显示界面上的多点或多用户的交互操作模式。
- 例如：水果忍者
 - 单点触控一次只能一根手指切
 - 多点触控就可以几根手指一起切
- 说明：一般手机的多点触控最多支持10点，说明可以支持十个接触点同时使用（也就玩水果忍者游戏可以十个手指头同时划）。



多点触控（续2）

- MultiAction有两个方法，add(TouchAction action)和perform()。实现的就是将多个TouchAction添加进来，在执行perform()后同时执行这些操作，实现多点触控操作。

MultiAction

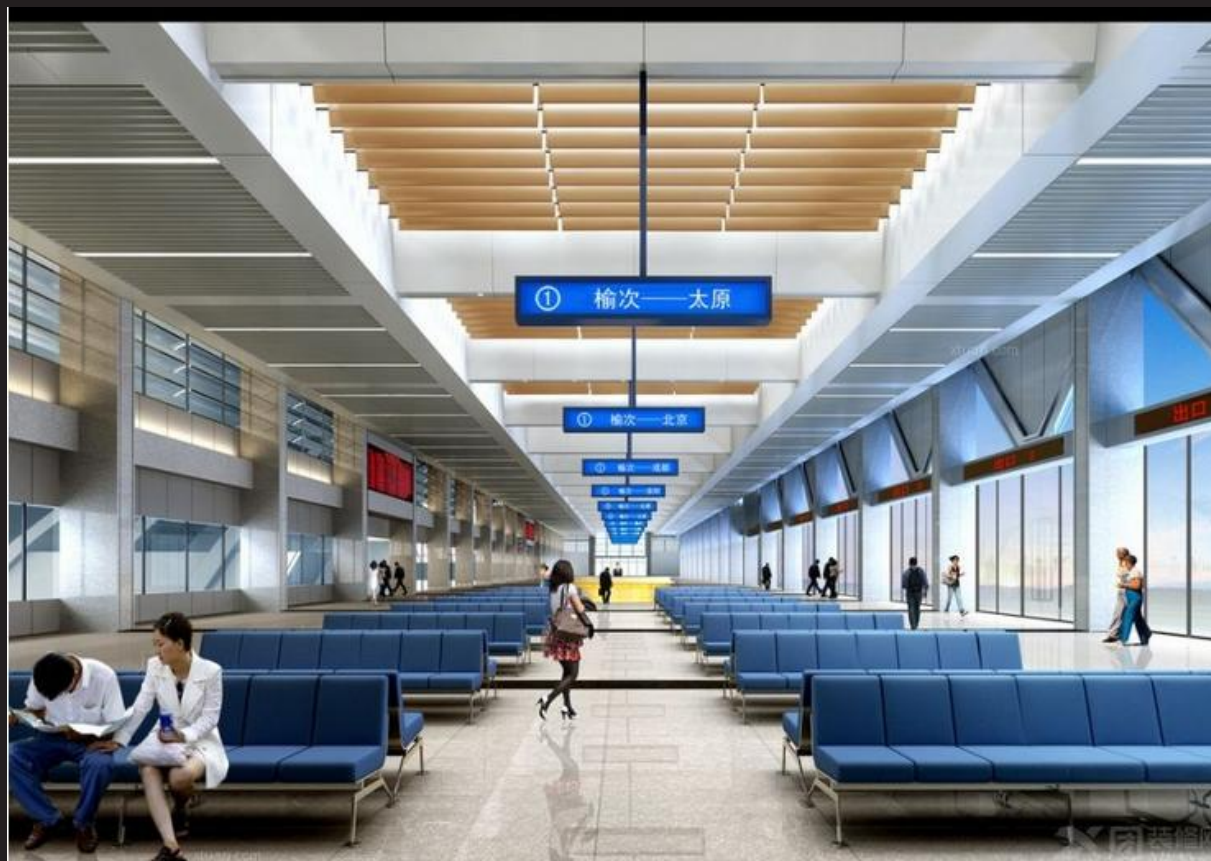
TouchAction
1

TouchAction
2



多点触控（续3）

- MultiAction与火车站候车室的功能类似
 - add方法：加入候车室等待人群
 - perform方法：检票放行，所有人一起出发



多点触控（续4）

- 步骤：导包、新建TouchAction、添加到集合、批量执行

步骤1：导包

TouchAction

MultiAction

步骤2：新建TouchAction

action1

action2

.....

步骤3：添加到集合

add(action1,action2,.....)

步骤4：批量执行

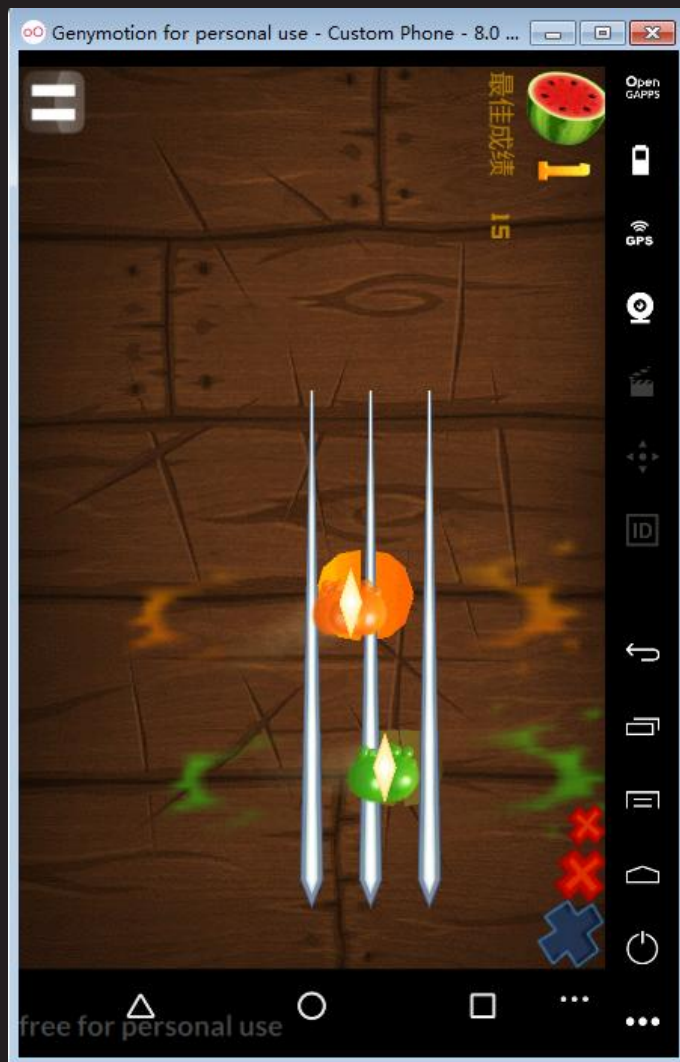
perform

多点触控（续5）

- 语法：
 - from appium.webdriver.common.touch_action import TouchAction
 - from appium.webdriver.common.multi_action import MultiAction #步骤1、导入MultiAction包
 - action1=TouchAction(driver)
action2=TouchAction(driver)
 - ma = MultiAction(driver)
action1.press.....动作1
action2.tap.....动作2 #步骤2、新建TouchAction
 - # 调用add方法将多个动作添加到动作集合，统一执行
ma.add(action1, action2,.....) #步骤3、添加TouchAction
ma.perform() #步骤4、批量执行

多点触控（续6）

- 示例：



多点触控（续7）

- 步骤1：导包
 - from appium.webdriver.common.touch_action import TouchAction
 - from appium.webdriver.common.multi_action import MultiAction
- 步骤2：新建TouchAction
 - a1=TouchAction(driver)
 - a2=TouchAction(driver)
 - a3=TouchAction(driver)
 - a1.press(x=w*0.1,
y=h*0.3).wait(1000).move_to(x=w*0.9,
y=h*0.3).wait(1000).release()



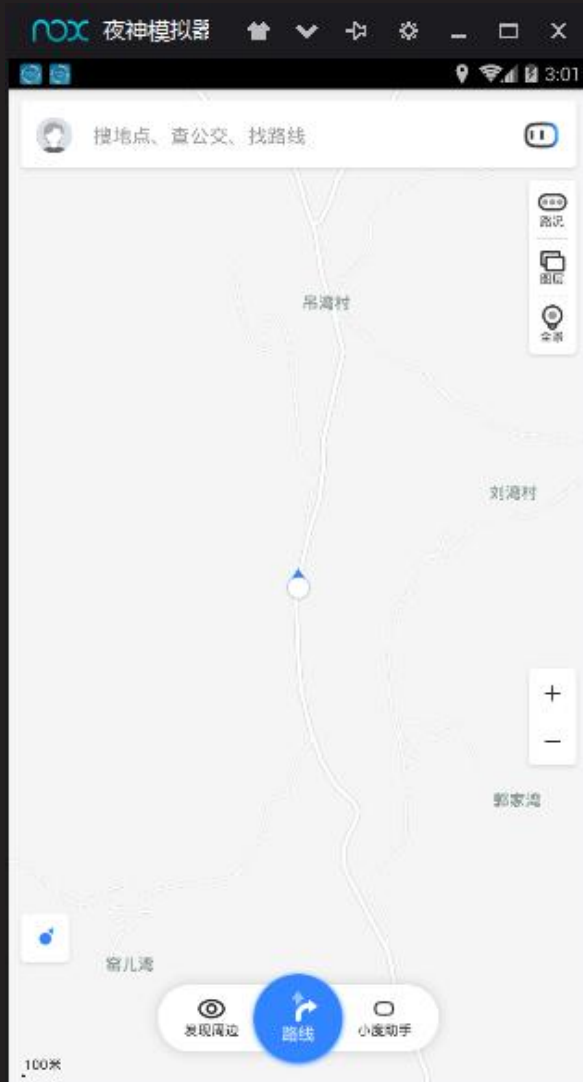
多点触控（续8）

- 步骤2：新建TouchAction（接上页）
 - a2.press(x=w*0.1, y=h*0.4).wait(1000).move_to(x=w*0.9, y=h*0.4).wait(1000).release()
 - a3.press(x=w*0.1, y=h*0.5).wait(1000).move_to(x=w*0.9, y=h*0.5).wait(1000).release()
- 步骤3：添加TouchAction到集合
 - ma=MultiAction(driver)
 - ma.add(a1,a2,a3)
- 步骤4：批量执行
 - ma.perform()



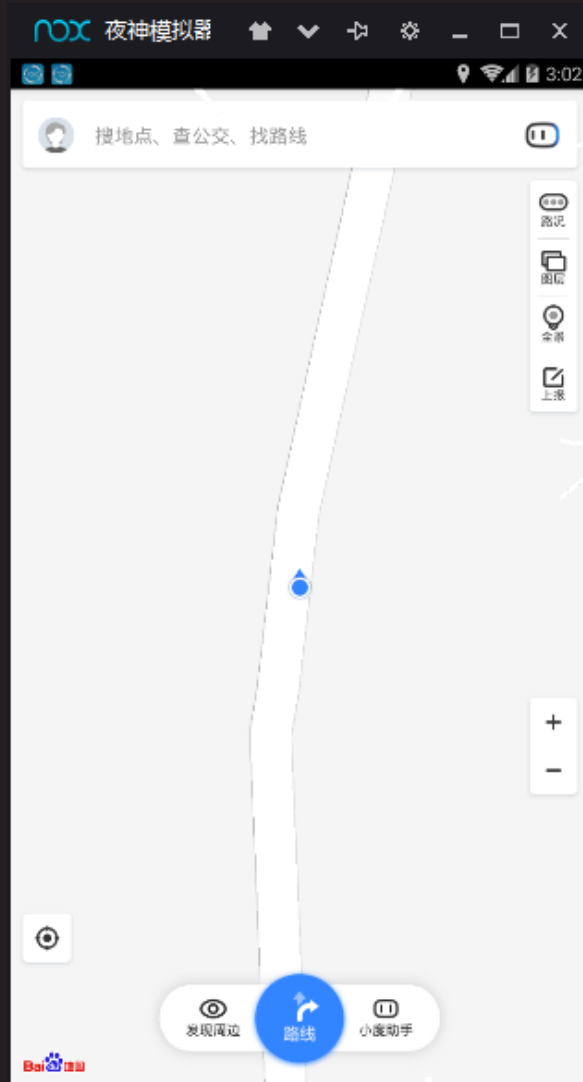
放大和缩小

- 示例：地图功能里，有放大和缩小功能



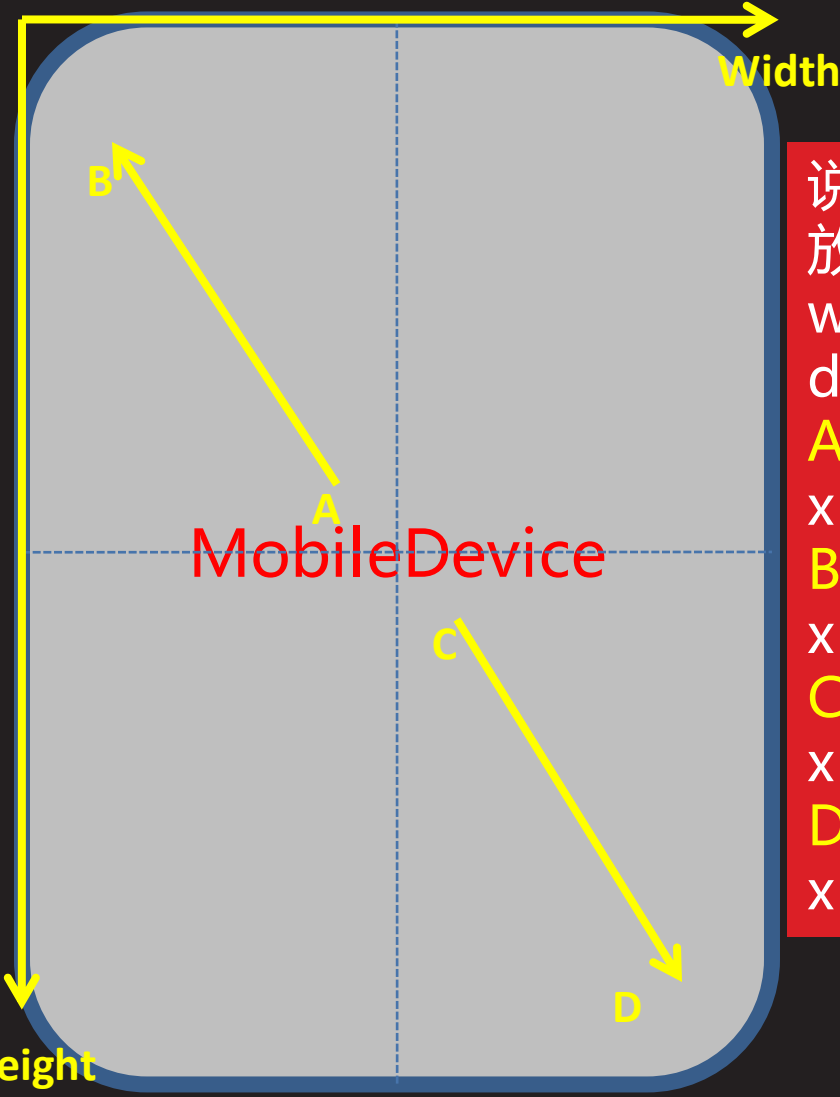
放大

缩小



放大

- 放大(Zoom)：屏幕 (双指往外移动来放大屏幕)



说明：

放大 $A \rightarrow B, C \rightarrow D$

$w = \text{driver.get_window_size()}['width']$

$d = \text{driver.get_window_size()}['height']$

A :

$x = w * 0.4, y = h * 0.4$

B :

$x = w * 0.1, y = h * 0.1$

C :

$x = w * 0.6, y = h * 0.6$

D :

$x = w * 0.9, y = h * 0.9$

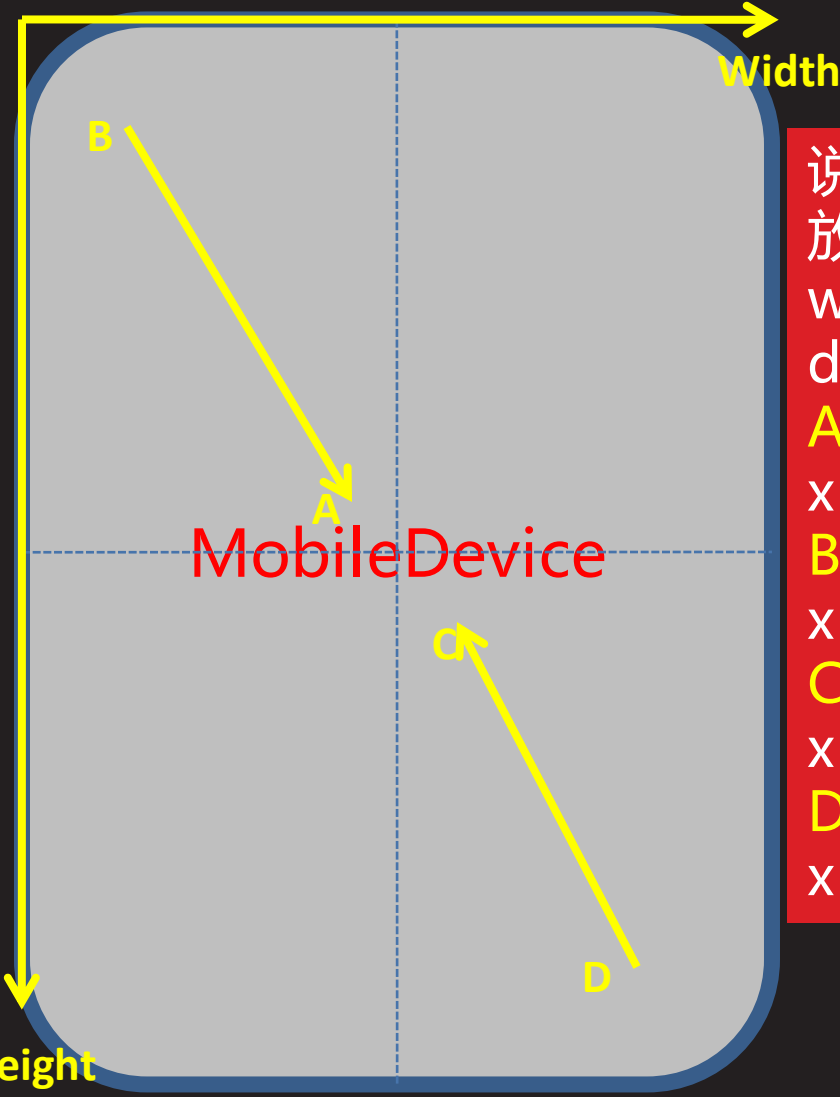
放大（续1）

- `action1=TouchAction(driver)`
`action2=TouchAction(driver)`
- `ma = MultiAction(driver)`
`action1.press(x=w * 0.4, y=h * 0.4).wait(1000).move_to(x=w * 0.1, y=h * 0.1).wait(1000).release()#动作1`
`action2.press(x=w * 0.6, y=h * 0.6).wait(1000).move_to(x=w * 0.9, y=h * 0.9).wait(1000).release()#动作2`
- `# 调用add将两个动作加入集合`
`ma.add(action1, action2)`
`ma.perform()#同时进行`



缩小

- 缩小(Pinch)：屏幕 (双指往内移动来缩小屏幕)



说明：

放大 $B \rightarrow A, D \rightarrow C$

$w = \text{driver.get_window_size()}['width']$

$d = \text{driver.get_window_size()}['height']$

A：

$x = w * 0.4, y = h * 0.4$

B：

$x = w * 0.1, y = h * 0.1$

C：

$x = w * 0.6, y = h * 0.6$

D：

$x = w * 0.9, y = h * 0.9$

缩小 (续1)

- action1=TouchAction(driver)
action2=TouchAction(driver)
- ma = MultiAction(driver)
action1.press(x=w * 0.1, y=h * 0.1).wait(1000).move_to(x=w * 0.4, y=h * 0.4).wait(1000).release()#动作1
action2.press(x=w * 0.9, y=h * 0.9).wait(1000).move_to(x=w * 0.6, y=h * 0.6).wait(1000).release()#动作2
调用add加入两个动作到集合
- ma.add(action1, action2)
- ma.perform()#同时进行

百度地图放大缩小

- 百度地图先缩小5次，再放大3次。
- 【详见COOKBOOK】



总结和答疑

