python + uiautomatorX 自动化培训

作者：　　李金洲
日期：2017/8/11

概要：python函数；uiautomatorX，API，Demo
目标：环境搭建，熟练使用api
作业：使用atx遍历触控板一级菜单
详细：

- **参考网址**

ATX github：https://github.com/NetEaseGame/ATX
ATX API：http://atx.readthedocs.io/en/latest/?badge=latest
ATX 社区：https://testerhome.com/topics/node78

- **install opencv**

For Win32

```
        $ pip install http://o8ookdsx.qnssl.com/opencv_python-2.4.12-cp27-none-win32.whl
```

For AMD64

```
$ pip install http://o8ookdsx.qnssl.com/opencv_python-2.4.12-cp27-none-win_amd64.whl
```

- **install uiautomatorX**

```
$ pip install --upgrade --pre atx
# install opencv dependencies
pip install opencv_contrib_python
```

- **改写脚本**

```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-

import os
import sys
import unittest
import time
from mylogging import log
# uiautomatorX 模块
import atx

# 主设备待测app包名和activity
appPackage = 'com.android.calculator2'

class Common(object):
    def __init__(self, driver,log):
        self._driver = driver
        self._logger = log

    def start_app(self):
        self._driver.press.home()
        self._driver.start_app(appPackage)
        if self._driver(resourceId="com.android.calculator2:id/formula").wait.exists(timeout=5000):
            time.sleep(1)
            self._logger.debug("start cal success")
            return True
        else:
            self._logger.error("start cal timeout")
            return False

    def cal(self, formula, expect):
        """
        1+2, 3

        """
        self.clear()

        # 循环点击输入的内容
        for op in formula:
            if op is "*":
                op = "×"
            if op is "/":
                op = "÷"
            self._driver(text=op).click()
            time.sleep(1)
            self._logger.debug("Input {0} success".format(op)) # 判断

        self._driver(text="=").click()
        self._logger.info("Expect: {0}".format(expect)) # 打印期望结果
        time.sleep(2)
        result = self._driver(resourceId="com.android.calculator2:id/formula").get_text() # 获取计算结果

        if result == expect:
            self._logger.info("Result: {0} - Success".format(result))
            return True
        else:
            self._logger.error("Result: {0} - Failed".format(result))
            return False


    def exit_app(self):
        self._driver.stop_app(appPackage)
        self._driver.press.home()
        if not self._driver(resourceId="com.android.calculator2:id/formula").wait.exists(timeout=3000):
            time.sleep(1)
            self._logger.debug("exit cal success")
            return True
        else:
            self._logger.error("exit cal failed")
            return False

    def clear(self):
        if self._driver(resourceId="com.android.calculator2:id/clr").exists:
            self._driver(resourceId="com.android.calculator2:id/clr").click()
            time.sleep(1)
        if not self._driver(resourceId="com.android.calculator2:id/clr").exists:
            self._logger.debug('clear success')
            return True
        else:
            self._logger.error('clear failed')
            return True


class AndroidTestCases(unittest.TestCase):
    def setUp(self):
        self._logger = log
        self._driver = atx.connect()
        # self._sdriver=Device("")
        self._mdriver = Common(self._driver, self._logger)
        if not self._mdriver.start_app():
            exit(-1)

    def tearDown(self):
        if not self._mdriver.clear():
            exit(-1)
        if not self._mdriver.exit_app():
            exit(-1)

    def test_one_plus_two(self):
        if not self._mdriver.cal("1+2", "3"):
            exit(-1)

    def test_click_imgage(self):
        self._logger.debug('click the image 1')
        self._driver.click_image('btn/1.jpg')
        time.sleep(3)
```