

开始教程

可以GAutomator脚本可前往<http://wetest.qq.com/cloud/index.php/phone/blrooike> 下载。示例：sample/sample.py中的**test**函数,示例apk:sampel/wetest_demo.apk。wetest_demo.apk已经集成了wetest sdk。

1.1 Simple Usage

已经安装好python及依赖库后，可以使用pycharm（请下载社区版，社区版免费）直接打开工程，你可以下面的代码开始我们的测试

```
#import lib path,only use in this demo
#import sys,os
#sys.path.append(os.path.abspath(os.path.join(os.getcwd(), "..")))

import wpyscripts.manager as manager
from testcase.tools import *

def test():
    engine=manager.get_engine()
    logger=manager.get_logger()

    version=engine.get_sdk_version()
    logger.debug("Version Information : {}".format(version))

    scene=engine.get_scene()
    logger.debug("Scene : {}".format(scene))

    sample_button=engine.find_element("/Canvas/Panel/Sample")
    logger.debug("Button : {}".format(sample_button))
    #engine.click(sample_button)
    screen_shot_click(sample_button)

test()
```

上面的代码可以保存为sample_test.py,然后运行

```
python sample_test.py
```

请确保，wetestdemo游戏已经拉起，GAutomator库能够查找到。sample_test.py保存在与main.py同级目录下，就能搜索到所有的库。

1.2 实例详解

wpyscripts.manager模块提供了自动化测试所需的所有功能，提供与引擎、手机、报告相关的内容，也提供了日志实现。testcase.tools里面封装了常用的函数，如screen_shot_click截图标记红点并点击，find_elment_wait获取element直到出现为止。

```
import wpyscripts.manager as manager
from testcase.tools import *
```

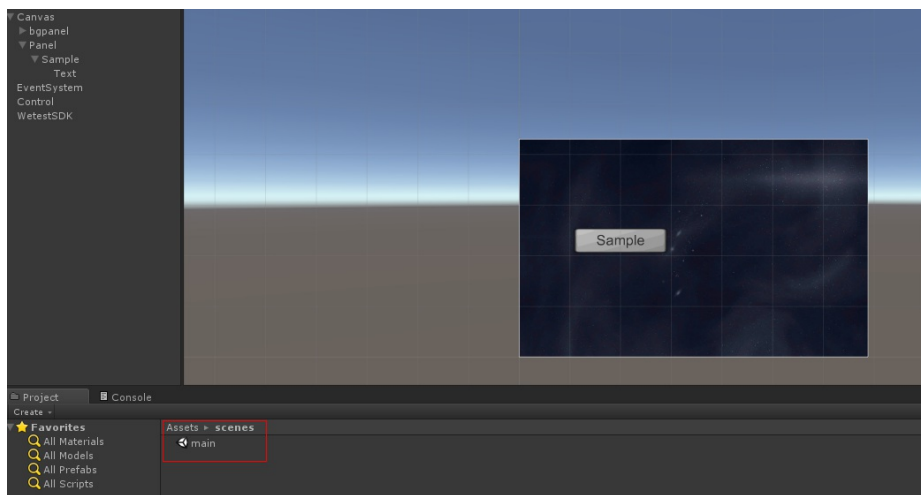
下一步，创建engine和日志实例

```
engine=manager.get_engine()
logger=manager.get_logger()
```

`engine.get_sdk_version()`能够获取Unity版本信息、Wetest sdk版本信息，能够获取该信息时，证明脚本已经成功连上游戏。如果获取失败，则会抛出Socket Error异常,抛出该异常可能是手机USB线没有连好或者手机开发者选项未打开，游戏并没有拉起等原因照成的。
`logger.debug("")`输出对应日志，请使用manager.get_logger()获取的实例，避免脚本在云端wetest.qq.com使用时出错。

```
version=engine.get_sdk_version()
logger.debug("Version Information : {}".format(version))
```

`engine.get_scene()` 获取当前游戏界面对应scene名称，wetestdemo游戏的第一个界面名称为main



`engine.find_element("/Canvas/Panel/Sample")` 查找当前界面中路径为/Canvas/Panel/Sample的节点，如果存在则返回Element，不存在则返回None。find_element直接使用Unity `GameObject.Find`查找当前游戏中的game object。

查找到的节点samle_button (Element)，有两个属性object_name,instance。object_name是节点的全路径，instance是节点实例的编号（`GameObject.GetInstanceID()`获取）instance在当前游戏中一定是唯一的。

`engine.click(sample_button)` 尝试点击samle_button这个GameObject的中心点。

```
sample_button=engine.find_element("/Canvas/Panel/Sample")
logger.debug("Button : {0}".format(sample_button))
engine.click(sample_button)
```

wpyscripts包含4大接口

```
engine=manager.get_engine()
reporter=manager.get_reporter()
device=manager.get_devcie()
logger=manager.get_logger()
```

- engine:Unity相关内容，主要包括控件获取，游戏操作
- reporter:云端报告相关，截图、标记操作过程、性能数据打标签
- device:手机设备相关，如屏幕长宽高、转向，也包括QQ登录等。
- logger:日志输出接口，保证本地与云端输出的日志都能获取到

1.3 wetest云端兼容测试

GAutomator编写好的测试脚本，只需要非常简单的修改，就能wetest云端上做兼容测试。云端几千台手机，按照脚本执行游戏。wetest能够发现兼容问题，同时高度还原执行现场，包括手机日志、崩溃信息、截图、执行过程等。

云端执行脚本时，会执行testcase.runner下的run函数，只需要把自己的业务逻辑加入到这个函数中即可

```
import traceback

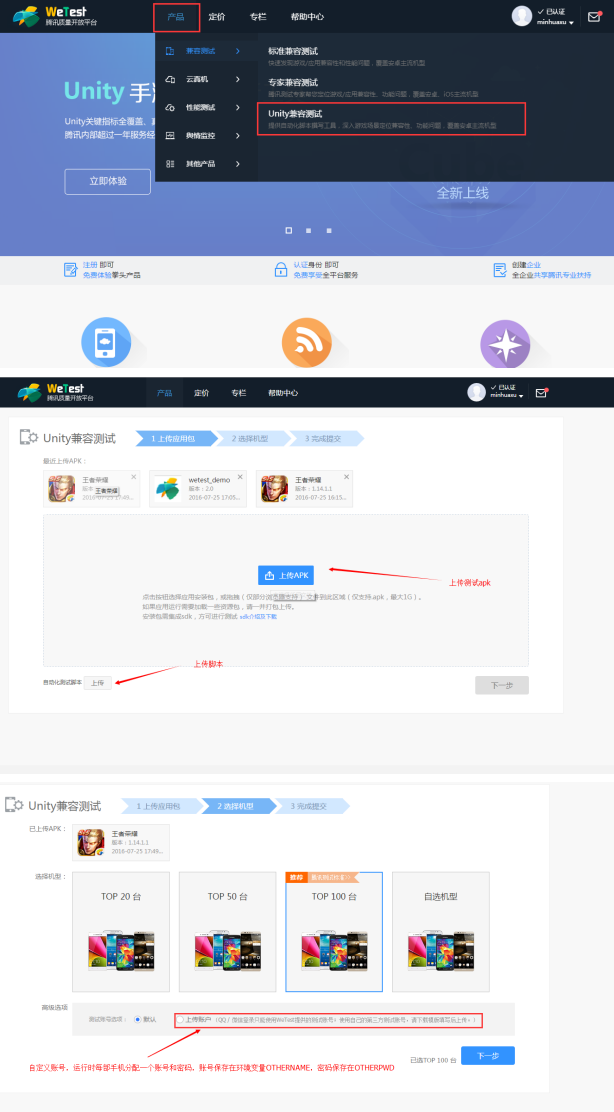
try:
    from sample.sample import *
except Exception,e:
    traceback.print_exc()

def run():
    """
    业务逻辑的起点
    """
    try:
        test()
    except Exception,e:
        traceback.print_exc()
```

然后，运行scripts目录下的，build.py

python build.py

会在scripts目录下产生一个wpyscripts_upload.zip。只有企业用户才可以使用云端测试，请登录wetest.qq.com，联系工作人员了解详情。



1.4 本地运行

注：调试时手动启动游戏，运行到指定界面，运行对应的脚本即可，如调试大厅界面的代码，游戏跑到大厅界面，再运行自动化测试逻辑。不需要从main.py启动

GAutomator支持一台PC在多台android手机上同时测试。首先需要设置游戏的包名，在main.py开头处，进行修改设置local_package。

```
local_package = os.environ.get("PKGNAME", "") # 你需要测试的包名,可以设置默认值

#local_package ="com.tencent.wetest.demo"
```

一般一个工程通过main方式启动，只能测试一个游戏，所以直接在main.py里面写死，也避免参数传入的麻烦。

1、测试一台手机，如果PC上USB只连接一台手机，直接启动main.py即可

```
python main.py
```

2、测试多台手机，如果PC上USB连接超过一台手机，需要通过命令行的方式启动

```
adb devices #查看当前手机序列号

saaaweadf      device
asdfadfadf     device
```

获取到当前PC连接的手机序列号之后，通过命令行的方式控制脚本在指定的手机上进行测试。

```
python main.py --qqname=2952020110 --qpwd=wetestpwd --engineport=50031 --uiport=19000 --serial=saaaweadf
python main.py --qqname=2952020111 --qpwd=wetestpwd --engineport=50032 --uiport=19001 --serial=asdfadfadf
```

上面的命令分别代表，在序列号"saaaweadf"手机上测试，测试时使用的QQ号为2952020110,密码为wetestpwd，与引擎建立映射的网络端口号为50031,与UIAutomator服务建立映射的网络端口为19000。第二条命令类似。

命令行参数含义如下：

```
--qqname:qq账号，每部手机应该都不一样
--qpwd:qq密码
--wechataccount:微信账号
--wechatpwd:微信密码
--othername:其他任何账号
--otherpwd:其他任何账号的密码
--engineport:与手机端的sdk服务建立网络映射，填入的为本地的网络端口号（如,50031），不同手机之间要确保不同
--uiport:与手机端的UIAutomator服务建立网络映射，填入的为本地的网络端口号（如,19008），不同手机之间要确保不同
--serial:adb devcies能够查看手机的序列号，不同的序列号代表不同的手机
```