GAutomator与Cube结合 说明文档

GAutomator与Cube性能自动化测试结合的使用说明。

GAutomator与Cube结合后，通过GAutomator控制手机端的Cube服务，完成对游戏运行过程中的性能自动化测试，用户将不再需要直接操作Cube。

使用环境

- Python 2.7
- Android手机（需Root、开启USB调试模式）
- 安装CubeX服务（CubeX.apk）

文件介绍

Cube的相关内容主要集中在`/cube/`目录下，包括了协议、通信、操作、管理、异常等文件。

- socket_client.py ： 提供与Cube服务通信的Socket连接支持。
- cube_protocol.py ： 提供与Cube服务通信过程中所需的协议Code，除此，也包括了Cube处理过程中的相关Code定义。
- cube_exceptions.py : 提供了Cube相关的异常定义。
- **cube_client.py ： 核心，提供了CubeClient对象，用于完成Cube相关测试流程操作**
- cube_manager.py : 提供了相关的公用管理操作函数，包括拉起Cube服务、获取CubeClient对象单例等。
- main_peroform.py : GAutomator与Cube结合的主要结合点，大部分Cube测试的流程在此完成（但不局限于此）。
- test.py : 简单的测试文件。
- **test_scene*sample*：*示例文件，场景测试需要根据该文件模式进行编写，并在cube*manager.py中引入、调用。**

操作流程

GAutomator与Cube结合后，游戏的拉起、结束等操作不再需要手动操作，而是交由Cube完成。因此，在脚本里使用时，部分处理流程，与原GAutomator使用有所区别。

GAutomator中负责控制整个Cube性能测试流程。

单次Cube性能测试流程

主要包括如下阶段：

- Cube准备阶段： Cube服务拉起、建立与Cube服务的Socket通信
- Cube测试阶段：开始测试、场景标记
- Cube结束阶段：停止测试、上传或删除报告

整体测试流程

主要包括如下阶段：

- 环境准备：获取帐号信息、包信息等
- Cube测试阶段
- 结束阶段

单次Cube性能测试流程

样例程序：`test_scene_sample.py`

`test_scene_sample.py`是一个测试场景的样例，其中主要包括了Cube准备阶段、Cube测试阶段、Cube结束阶段的内容。

Cube准备阶段

Cube准备阶段包括了拉起Cube服务、建立与Cube服务之间的通信等。

相关的操作通过使用`cube/cube_manager`内的函数进行处理。

拉起Cube服务

```
def start_service(service_name)
```

通过传入Cube服务的名称，完成服务的拉起，Cube服务名称为：

```
com.tencent.cubex/com.tencent.cubex.service.CubeService
```

获取Cube客户端

Cube客户端在GAutomator采用单例模式提供，需要引入cube_manager模块，使用其中定义的`cube_client`变量。

通过该变量，将能够获取`CubeClient`对象单例，因此，可以在脚本的任何需要使用Cube的地方通过该方法获取到`CubeClient`对象。

启动测试

Cube测试分成4种模式，包括：

- 0： 轻度测试
- 1： 重度测试
- 2： 资源测试
- 3： mono内存

代码示例：

```
from cube_manager import *

ret = cube_client.begin_cube_test(local_package, scene_name, CubeTestLevel.LIGHT, CubeSnapshots.OPEN)
```

Cube测试阶段

Cube测试阶段主要完成了具体的测试。

Cube测试阶段中，由GAutomator自动化测试为主，在关键的测试点中，可以进行**打标记**操作。

打标记

Cube性能测试过程中，可以在重要的位置进行**打标记**处理，以便后续能够**快速定位特殊测试点**的性能数据。

打标记通过CubeClient对象的`mark_tag()`函数实现。

代码示例：

```
from cube_manager import *

cube_client.mark_tag()
```

Cube结束阶段

停止测试

游戏测试结束后，使用Cube提供的停止测试函数，结束游戏进程，在这期间，Cube收集相关数据。

代码示例：

```
from cube_manager import *
```

```
cube_client.stop_cube_test()
```

上传或删除报告

游戏测试结束后，需要根据需要，上传报告。

代码示例：

```
from cube_manager import *

cube_client.upload_record()
```

整体测试流程

整体测试流程为GAutomator控制的流程，与原GAutomator自动化测试基本一致，这部分内容仅作了解，用户只需要关注**“Cube测试阶段”**相关内容即可。

环境准备

环境准备阶段主要完成了参数解析、信息处理等，如帐号信息、端口信息等。

Cube测试阶段

Cube测试阶段包括了多个场景测试，每个场景测试都是独立的、完整的。

场景测试内的处理流程前文已经详细介绍。在这里，主要是要严格的根据 Cube Web配置内容，安排场景测试的执行。只有正确的执行了每一个测试场景，才能够完成当次测试，并获得邮件报告的推送，否则将有可能无法出发邮件报告发送规则。

结束阶段

主要进行异常处理等，完成整个GAutomator测试。

其他

如有问题，请联系开发。