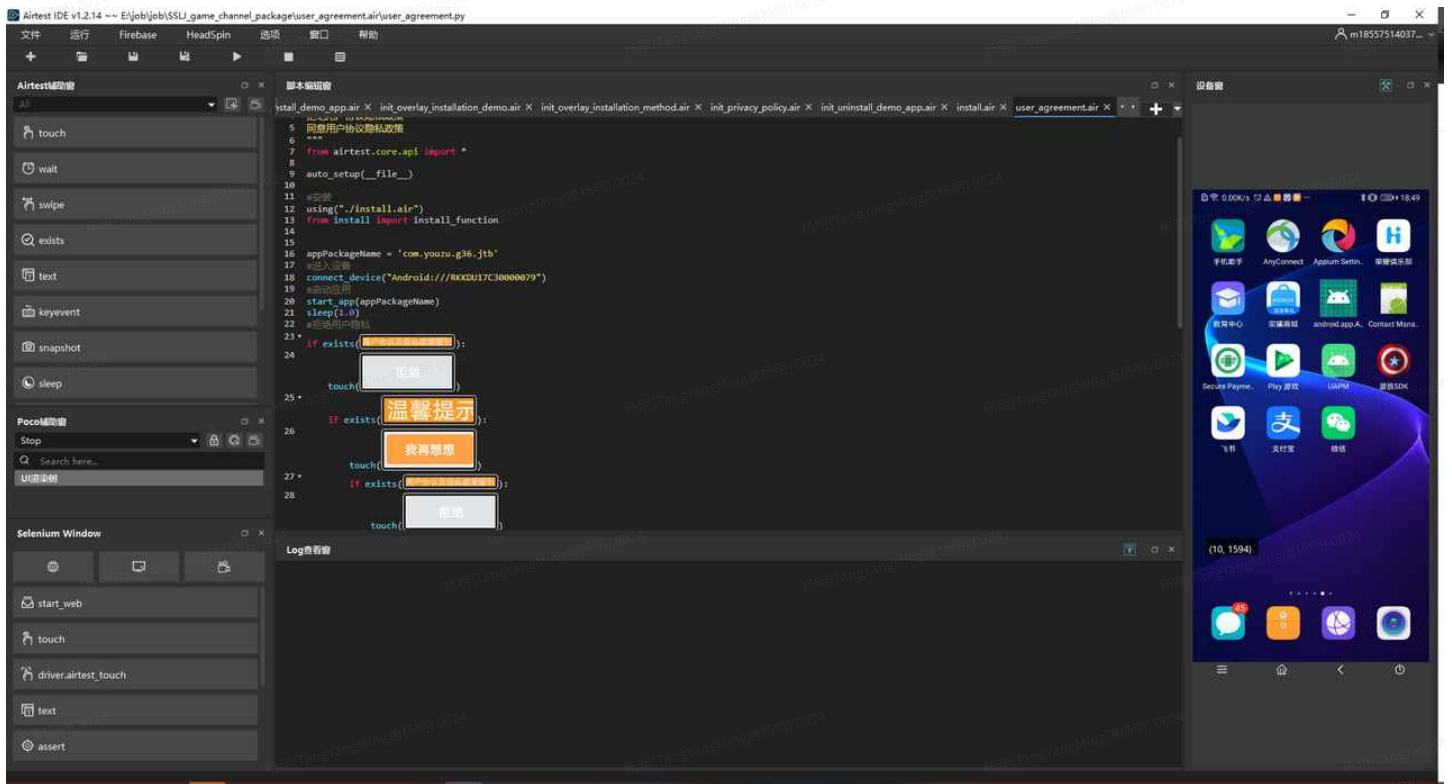


UI自动化测试-Airtest-IDE工具使用

1. AirtestIDE主界面介绍



默认情况下，AirtestIDE的界面主要由1个顶部菜单栏和5个窗口组成。分别是，顶部菜单栏、Airtest和Poco辅助窗、脚本编辑窗、log查看窗和设备窗口。其实还有1个airtest-selenium的辅助窗，需要在顶部菜单栏的窗口选项里面勾选上，才会显示在IDE的界面中。此外，初始布局中的"设备窗"是工具的设备连接交互区。

接下来我们来详细介绍一下主要使用到的功能吧！

2. 顶部菜单栏

顶部菜单栏（用于新建脚本、运行脚本、生成运行报告等一系列操作，还可以设置脚本相关的配置）详细介绍一下常用的功能

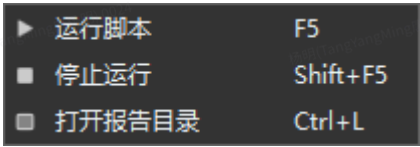
1) 文件

“文件”主要用于一系列的脚本文件的操作，比如新建、打开、保存和另存等等



2) 运行

“运行”则用于运行脚本的一系列操作，我们还可以在这里打开报告目录



3) 选项

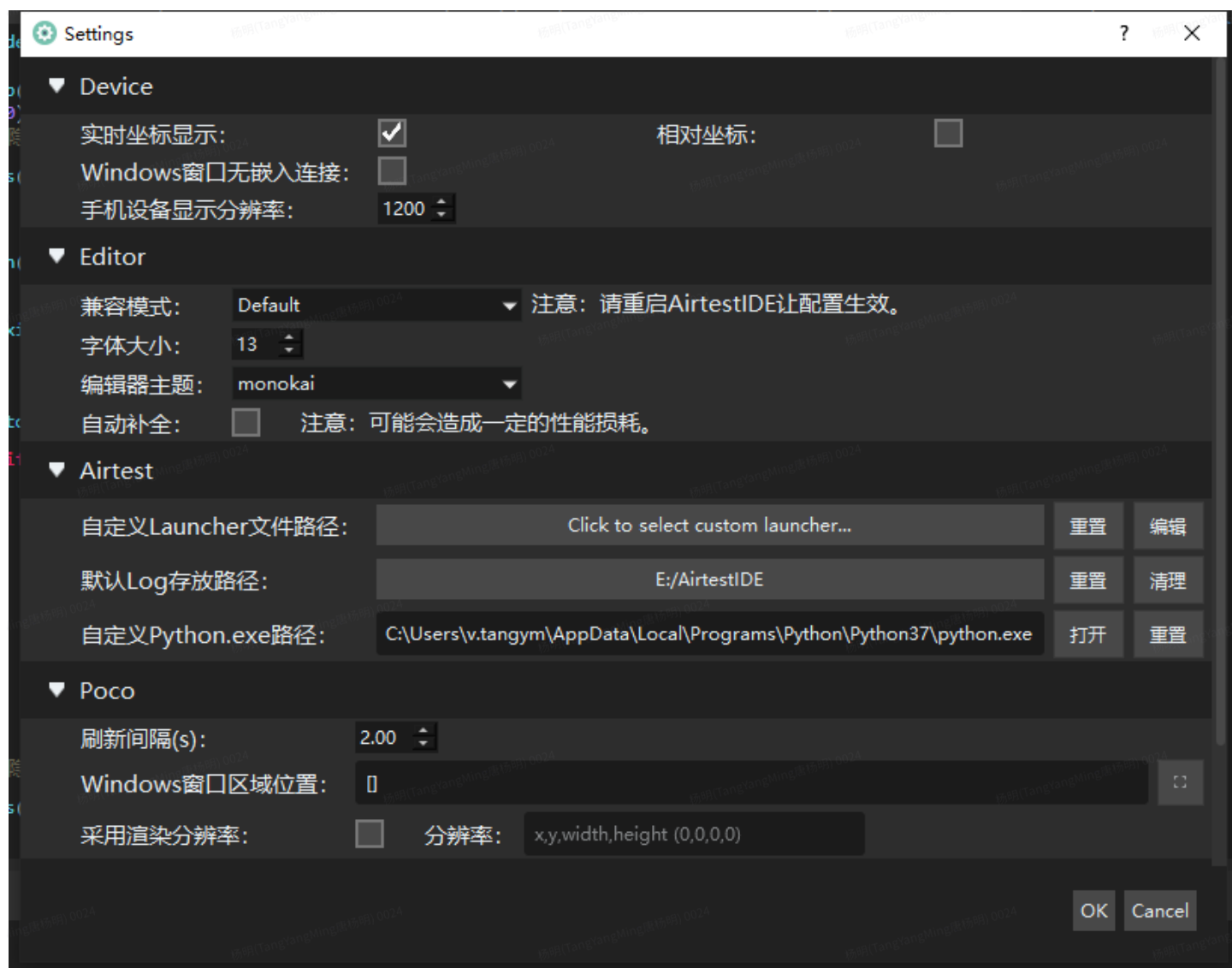
“选项”的功能比较丰富。可以对Airtest、Poco和运行环境等做一系列设置，还可以设置IDE的语言和皮肤。主要用到的是设置



我们在IDE的使用过程中，会经常遇到需要在选项里进行设置的情况，比如设置使用本地Python环境，不使用AirtestIDE自带的环境来跑脚本

设置

在AirtestIDE的顶部菜单栏中，**选项** 这一设置包含了非常多实用的功能，需要详细了解的如下



1) Device

在Device下，我们可以设置显示坐标、选择Windows无嵌入模式以及设置设备窗口显示的手机分辨率等等：



① 实时坐标显示与相对坐标

仅勾选 **实时坐标显示** 选项，可以实时在手机屏幕画面上显示出鼠标位置的绝对坐标，方便大家查看以及获取坐标位置。此时 点击鼠标右键，还可以自动将当前坐标信息复制到剪贴板中，在代码里只需要直接粘贴即可插入点击位置的坐标。

在勾选了 **实时显示坐标** 的情况下，再勾选 **相对坐标** 选项，将会以(0, 0)到(1, 1)为范围显示出相对坐标。使用相对坐标可以避免跨分辨率的操作点超出屏幕的问题，使坐标操作兼容性更好。

如下图显示的是绝对坐标：



如下图显示的是相对坐标：



② 手机设备显示分辨率

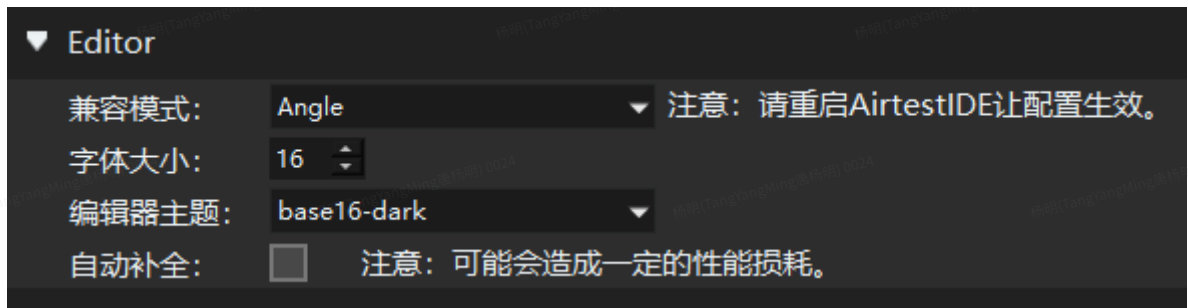
当已经连接好 Android 手机时，AirtestIDE会在Device窗口显示出手机实时同步画面。如果觉得清晰度较低，可以修改 手机设备显示分辨率 的值，数字越大，清晰度越高。默认为800，取值范围为300-2000之间。

手机设备显示分辨率: 1200

如果还是觉得手机实时画面清晰度不够高，可以在连接手机之前，在下拉菜单中勾选 Use Javacap 模式（不勾选的情况下，默认使用Minicap连接手机），再进行连接。在Javacap模式下，能够看到清晰度更高的手机画面，但是延迟会比默认的Minicap模式要稍微高一些。

2) Editor

在Editor下，我们可以设置IDE的启动模式、字体大小、编辑器主题以及开启自动补全等等。



① 兼容模式

有些少部分用户由于显卡兼容性问题，可能在启动AirtestIDE时，不能看到脚本编辑窗口和log窗口中的文字，或者是启动后会闪退，但是通过双击文件夹中的 `兼容性模式启动.bat` 脚本能够正常启动AirtestIDE。

为了能够无需每次启动都要双击脚本，我们在设置中新增了一个选项：



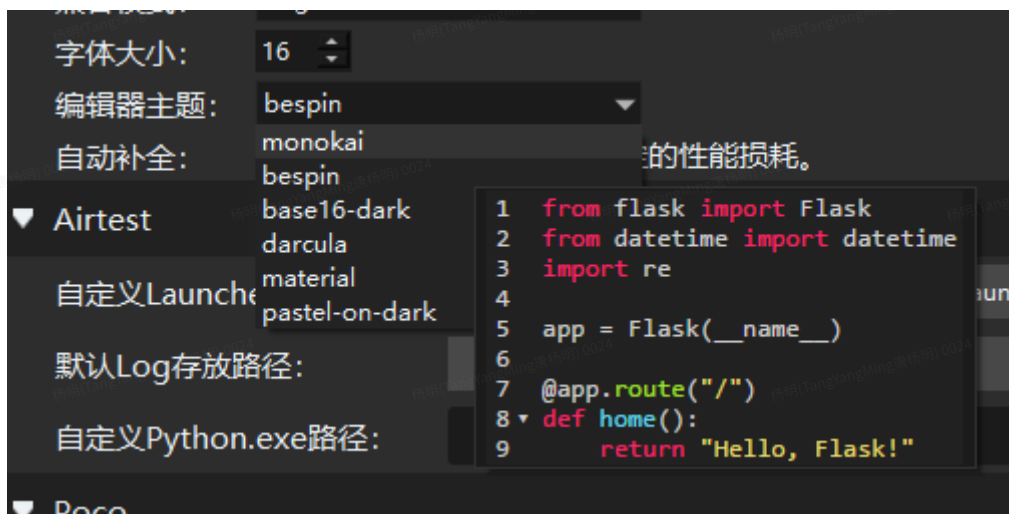
默认情况下会选择 `default` 模式，如果启动有问题的用户，可以尝试选中下拉菜单中的另外三个选项，并且重启AirtestIDE，查看接下来是否能正常显示脚本编辑窗口和log查看窗口。如果能够生效，未来启动AirtestIDE时都将会以这个配置项来启动，无需每次都手工双击 `兼容性模式启动.bat` 脚本来启动AirtestIDE了。

② 字体大小

可以调节编辑窗口和Log窗口内的文字大小，默认是 `14px`。在AirtestIDE中，还可以通过按住 `ctrl+鼠标滚轮` 来实现改变文字大小。

③ 编辑器主题

我们根据 `AirtestIDE` 提供的两款不同配色的皮肤，分别添加了几种对应的编辑框代码主题可供选择，当鼠标悬停在主题名称上面时，可以看到预览图片：



④ 自动补全

自动补全 提供了一个编辑窗中的python代码自动补全功能，默认为不勾选此功能，因为在部分机器上可能会造成性能降低。

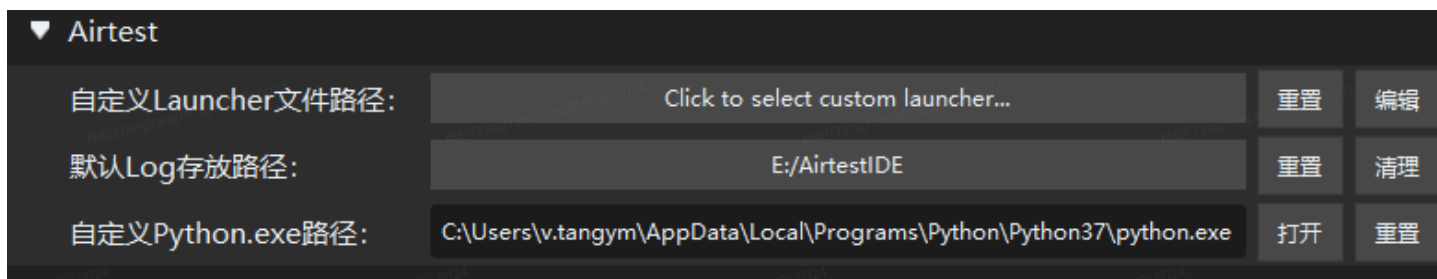
打开此选项后，可以在AirtestIDE中编辑代码时，享受到 `airtest` 和 `poco` 库的代码自动补全提示，如果希望能够对更多python库进行更好的自动补全，建议参考 [python环境部署](#)，在部署了本地Python环境以后，在IDE设置的 **自定义Python.exe路径** 选项里设置好对应的python.exe路径，即可使用到更强大的python代码自动补全效果。

如果修改了 **自定义Python.exe路径** 的路径，需要重启IDE后才能让新的代码补全生效。



3) Airtest

在Airtest部分，我们可以设置自定义的Launcher文件路径、可以设置默认Log存放路径，还可以设置IDE使用本地的Python环境：



① 自定义LAUNCHER文件路径

自定义Launcher文件路径 选项可以设置一个自定义的 `launcher.py` 文件的路径，通常情况下这个选项无需设置

② 默认LOG存放路径

默认Log存放路径 用于指定在AirtestIDE中运行脚本时的log目录，IDE将会把产生的log文件、截图数据默认放置在这个目录下。

默认Log存放路径: E:/AirtestIDE **重置** **清理**

点击 **重置** 按钮将会把log目录设置为系统默认的temp目录，点击 **清理** 按钮将会自动帮忙清空该目录下的文件，释放硬盘空间，请操作前确认好没有需要保留的重要log文件。

③ 自定义PYTHON.EXE路径

自定义Python.exe路径 允许使用本地的 `python.exe` 来跑在AirtestIDE里写好的脚本。由于AirtestIDE内置了一个python3.6.5环境，假如你希望能够使用自己的本地python环境（可以跑一些本地安装了的第三方库等等），可以通过设置这个选项来改变运行脚本默认使用的 `python.exe`。

点击 **打开** 可以打开文件选择窗口，请务必选择一个本地安装的 `python.exe` 路径。点击 **Reset** 将会将填写内容清空。

在mac系统下，需要设置python可执行文件路径，如果使用的是系统python，使用 `which python` 指令来获取python可执行文件所在路径（如果是python3，需要使用 `which python3`）：

```
1 >which python3
2 /Library/Frameworks/Python.framework/Version/3.7/bin/python3
```



将该路径填写到AirtestIDE的设置窗口即可，请注意填写的是 `bin/` 目录下的可执行文件，而不是 `python` 所在目录。

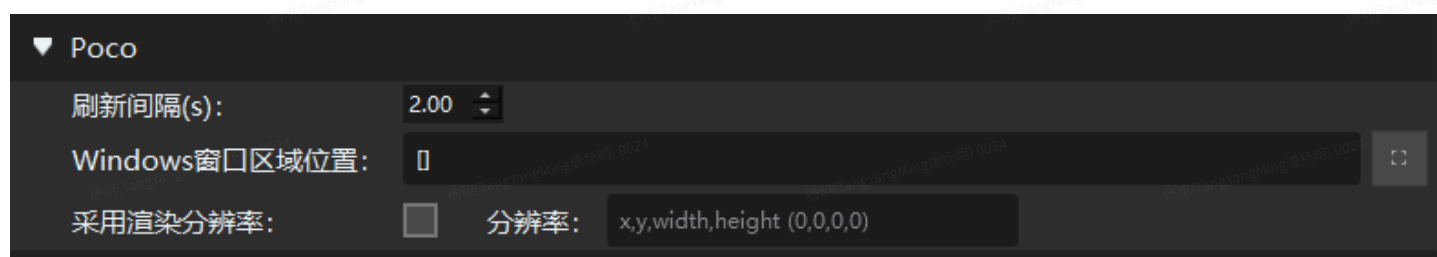
假如使用了虚拟环境，就需要填写虚拟环境目录下的python可执行文件路径了，例如在mac系统中可能是 `/Users/username/.virtualenvs/your_env/bin/python3`。

注意：

- 如果希望使用本地Python环境运行脚本，需要事先部署airtest和pocoui相关环境，请参考[运行脚本](#)里的 `本地python环境运行脚本` 的内容进行环境部署。
- AirtestIDE设置了本地的 `python.exe` 路径后，会默认使用 `AirtestIDE/sample/custom_launcher.py` 来启动脚本，如果使用了自己的 `launcher.py`，请在AirtestIDE中设置自己的 `launcher.py` 的脚本路径。关于 `launcher.py` 的介绍，请查看[Airtest启动器介绍](#)的内容。

4) Poco

在Poco部分，我们可以设置UI树的刷新间隔时间、框选Windows窗口的区域位置和自定义渲染分辨率等等：



① 刷新间隔 (S)

`刷新间隔 (s)` 选项用于配置Poco辅助窗中刷新Poco-UI树的间隔时间，单位为秒，默认为2s。

② WINDOWS窗口区域位置

在IDE连接了Windows游戏窗口后（该Windows游戏已经提前嵌入了Poco-sdk），如果想要使用Poco来查看界面元素，可以通过 `Windows窗口区域位置` 来设置窗口位置，这是为了能够在Windows窗口上显示poco元素标记用的，不设置不会对脚本运行产生任何影响。

③ 采用渲染分辨率

AirtestIDE 可适应90%以上全面屏不适配的问题, 对于剩下未支持的10%手机, IDE支持用户自定义设置渲染分辨率:



勾选采用渲染分辨率,然后输入竖屏模式下的渲染分辨率,点击OK即可。渲染分辨率为用逗号隔开的四个数字,数字分别代表竖屏模式下的 offset_x, offset_y, offset_width, offset_height.

比如本例中在 1080*2220 分辨率下的大鱼来了,它在手机上会有两个黑边,上黑边高度为 100px,画面高度为2020px,下黑边高度为 100px,所以它的渲染分辨率则为(0,100,1080,2020)



竖屏模式下

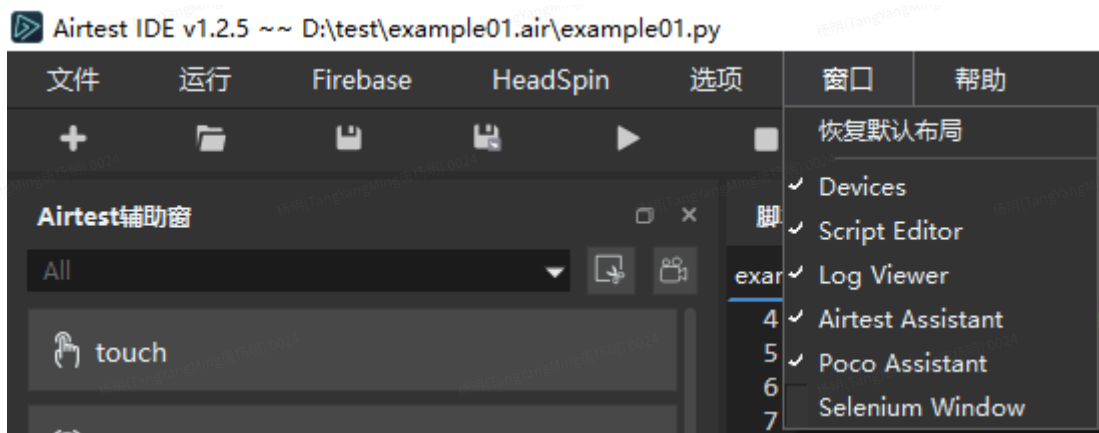
上黑边高度为 100 纵向画面高度为 2020
(0, 100, 1080, 2020)
左侧没有黑边, 宽度为 0 横向画面宽度为 1080

注意

以上设置, 修改之后均需要点击右下角的OK按钮保存设置, 修改设置之后直接关闭设置窗口将不生效。

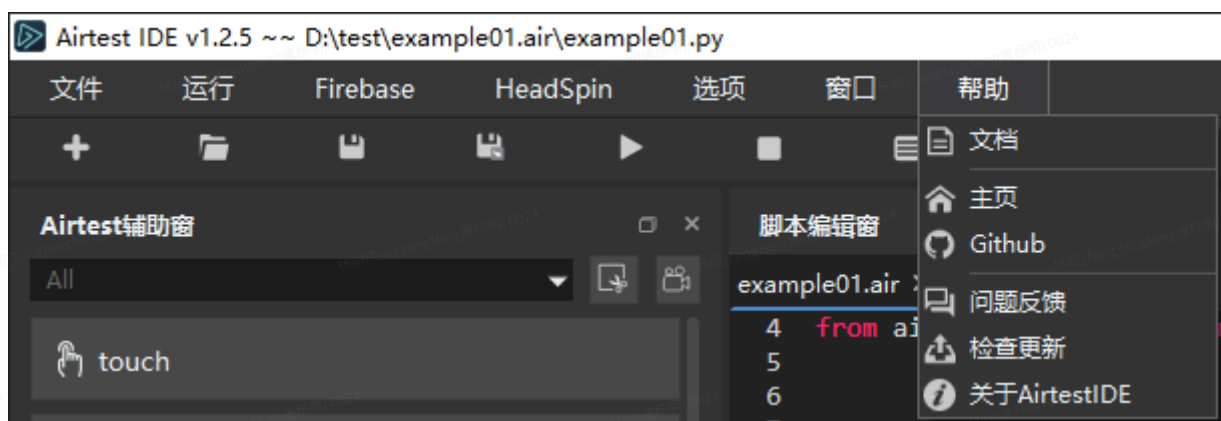
4) 窗口

“窗口”可以则对IDE的窗口显示做一些改动, 默认情况下, IDE有5个窗口, selenium 窗口默认是不展示的。如果你不小心拖动了一些窗口或者不小心把某些窗口关掉了, 可以使用“恢复默认布局”把窗口的布局恢复到最初始的状态。



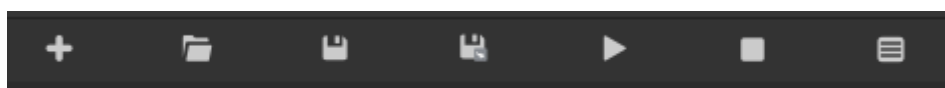
5) 帮助

最后这个“帮助”，可以帮助我们快速跳转到官方文档、官方主页、AirtestProject的GitHub项目上，还在做一些问题反馈和更新操作。



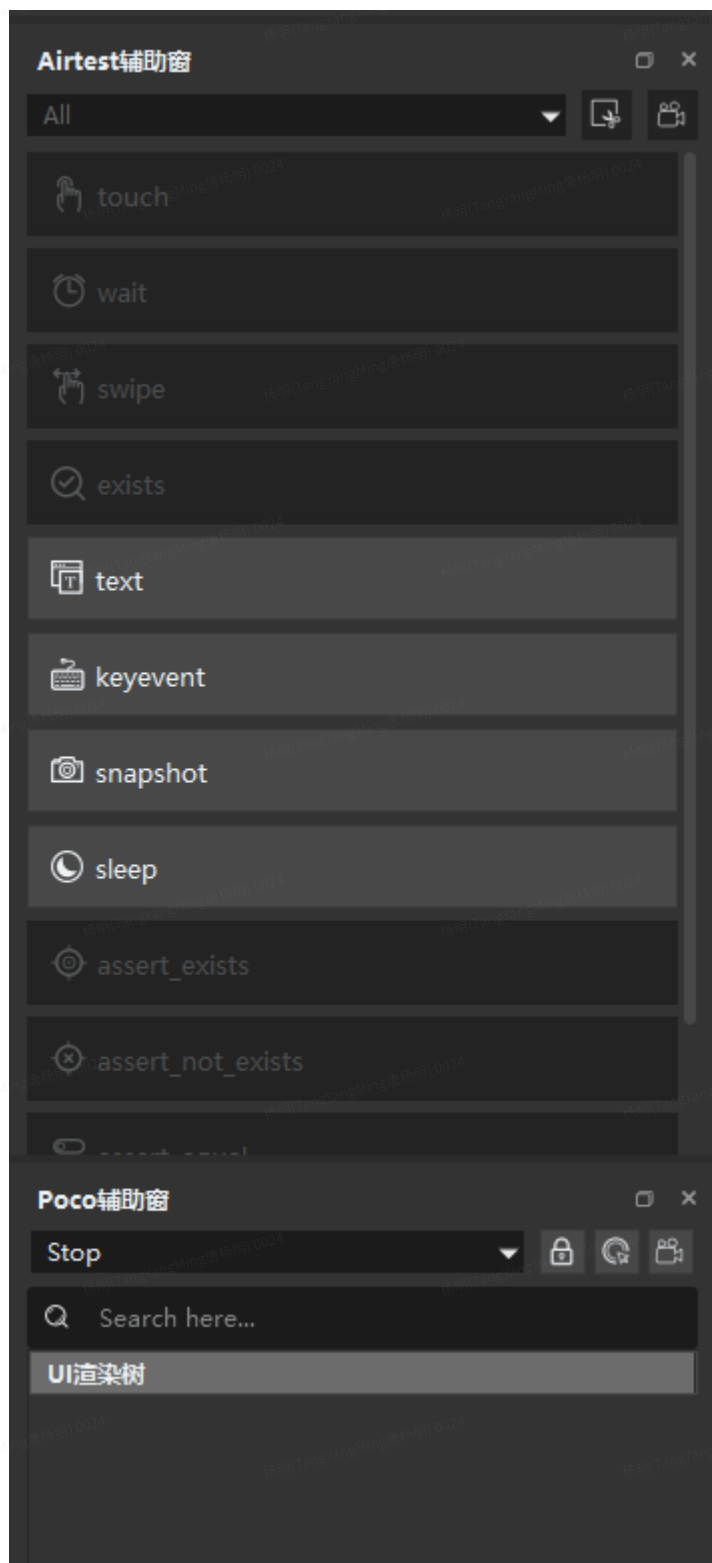
6) 快捷键

顶部菜单栏还包含一行的快捷按钮，主要是针对脚本的快捷操作，包含新建、打开、保存、另存、运行、停止运行和生成报告：

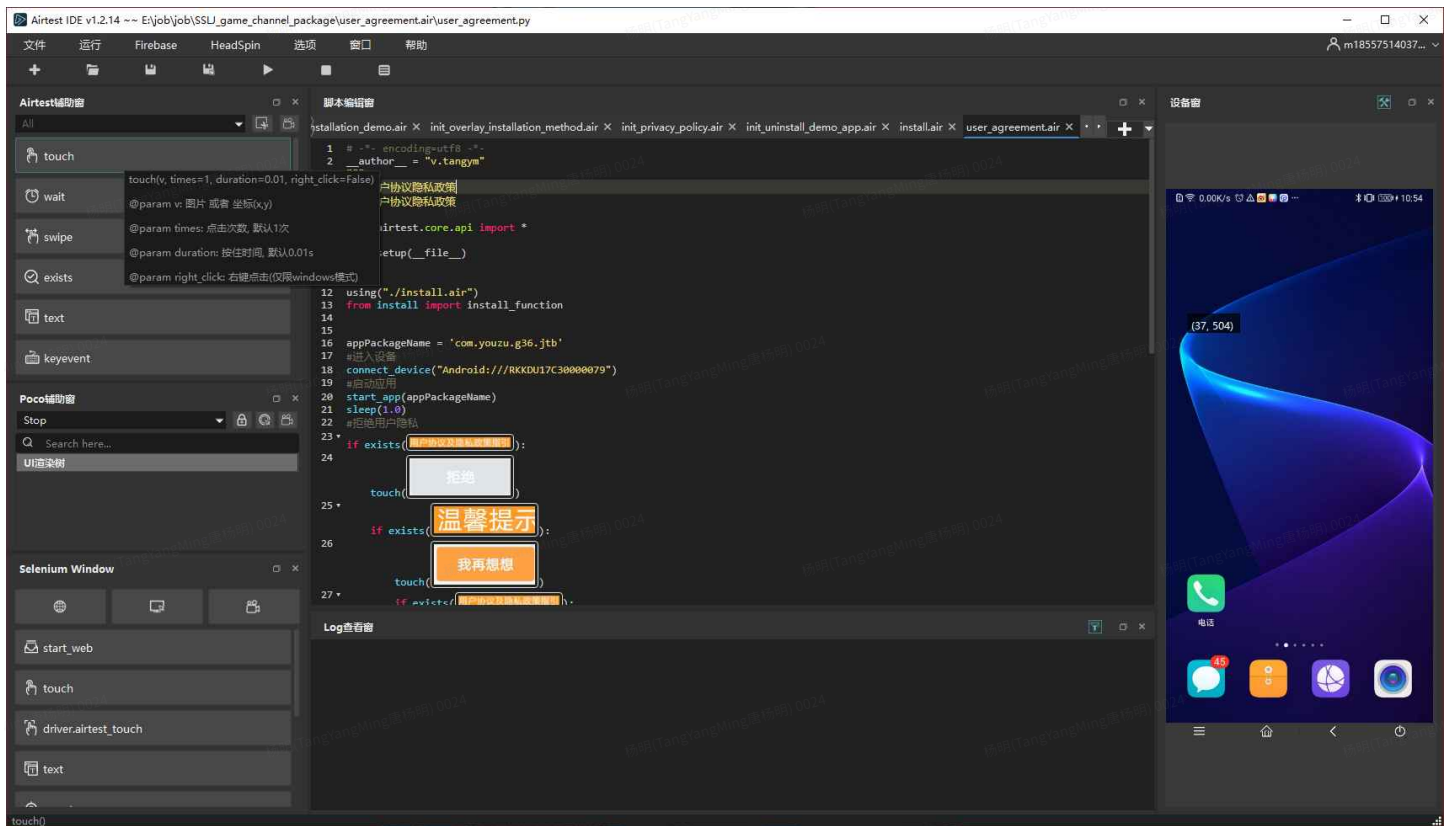


3. Airtest和Poco辅助窗

IDE左侧的2个窗口，Airtest和Poco的辅助窗，使用这两个窗口可以帮助我们快速录制airtest和poco的自动化脚本。在没有连接设备的情况下，airtest辅助窗中，需要截图的一些操作的快捷键是不可用的，poco辅助窗中也不显示任何UI渲染树的信息。

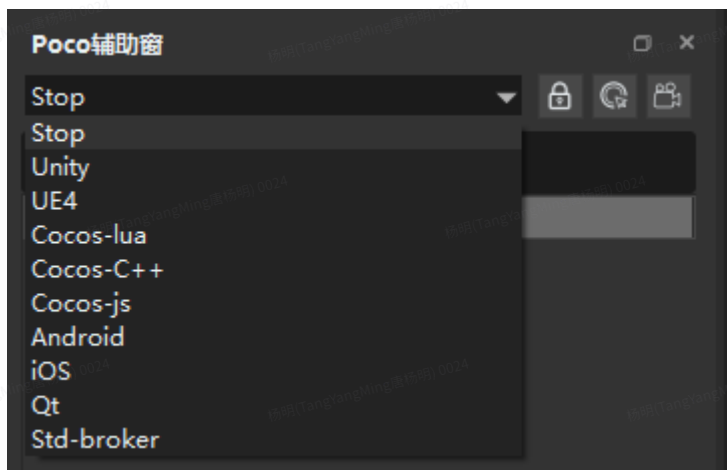


当我们在设备窗口连接上待测设备之后，Airtest辅助窗里所有的快捷键都变成可用的，把鼠标移动到对应的快捷键上，还可以查看该方法的参数详情。右上角还给我们提供了一个截图和录制脚本的快捷键。



了解：（Poco目前我们的项目不支持）

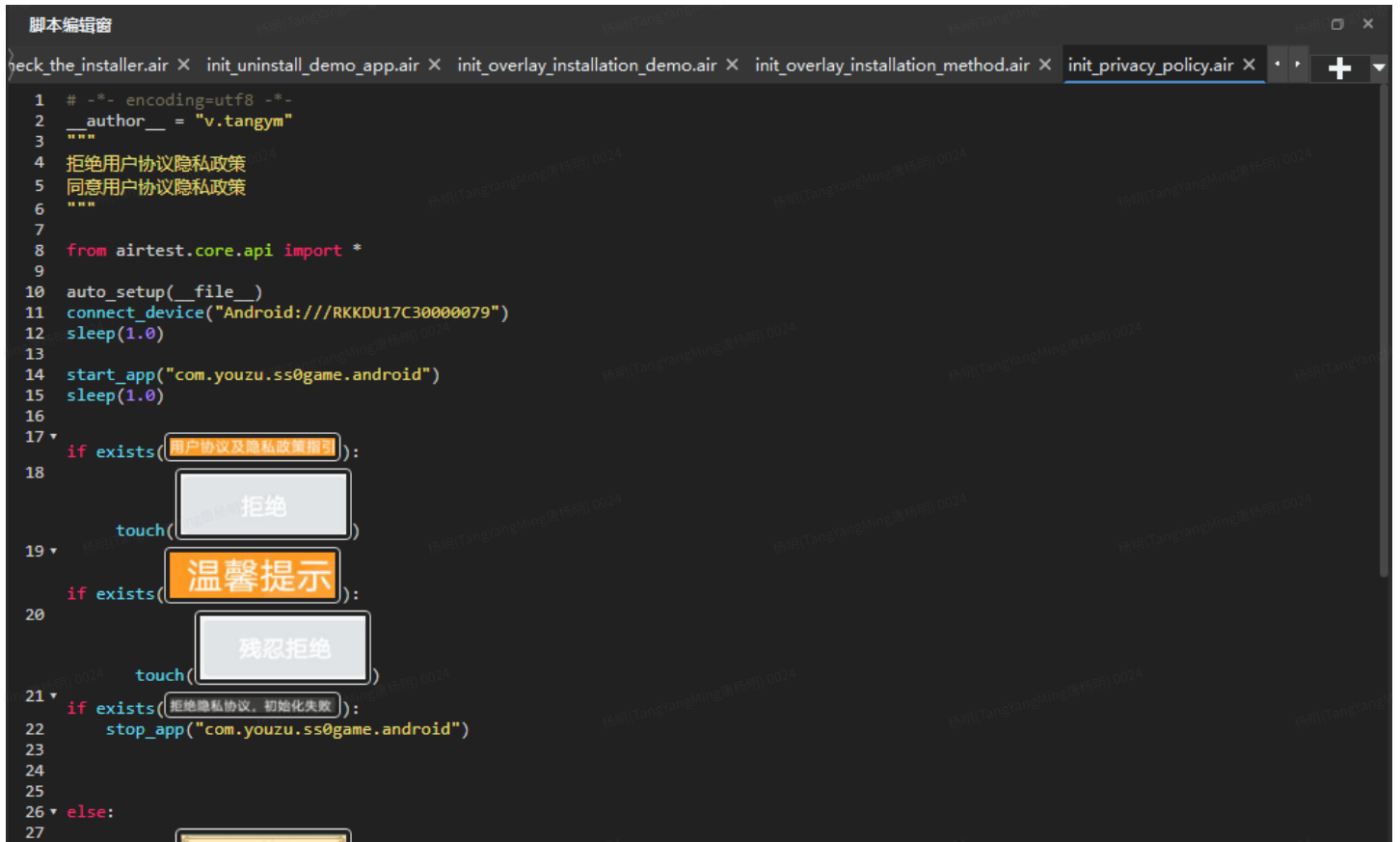
Airtest是基于图像识别的，而poco是基于控件搜索的，所以连接设备之后，我们就可以在poco辅助窗内，选择对应的模式，然后查看设备的UI渲染树，以此来对设备上的控件做一些操作



比如此时我们连接的是一台Android设备，测试的是1个unity游戏项目，我们就可以下拉选择unity模式，稍等一会，UI树就被刷新出来了。点击右上角的三个按钮，我们就可以锁定某个控件或者录制poco脚本了。如果测试的是安卓原生的一些应用，则选择android模式即可

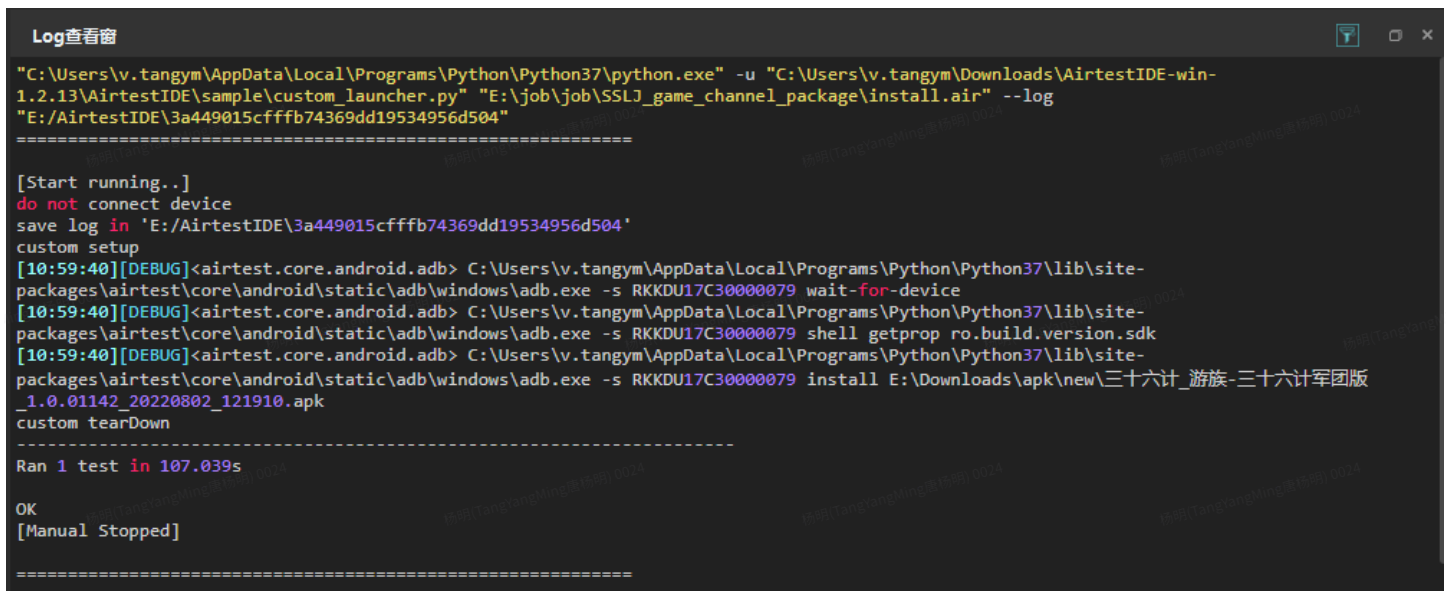
4. 脚本编辑窗

就像上文演示的那样，录制功能可以帮助我们在脚本编辑窗自动生成airtest和poco的脚本，我们也可以自己手动在编辑窗内编写自动化脚本



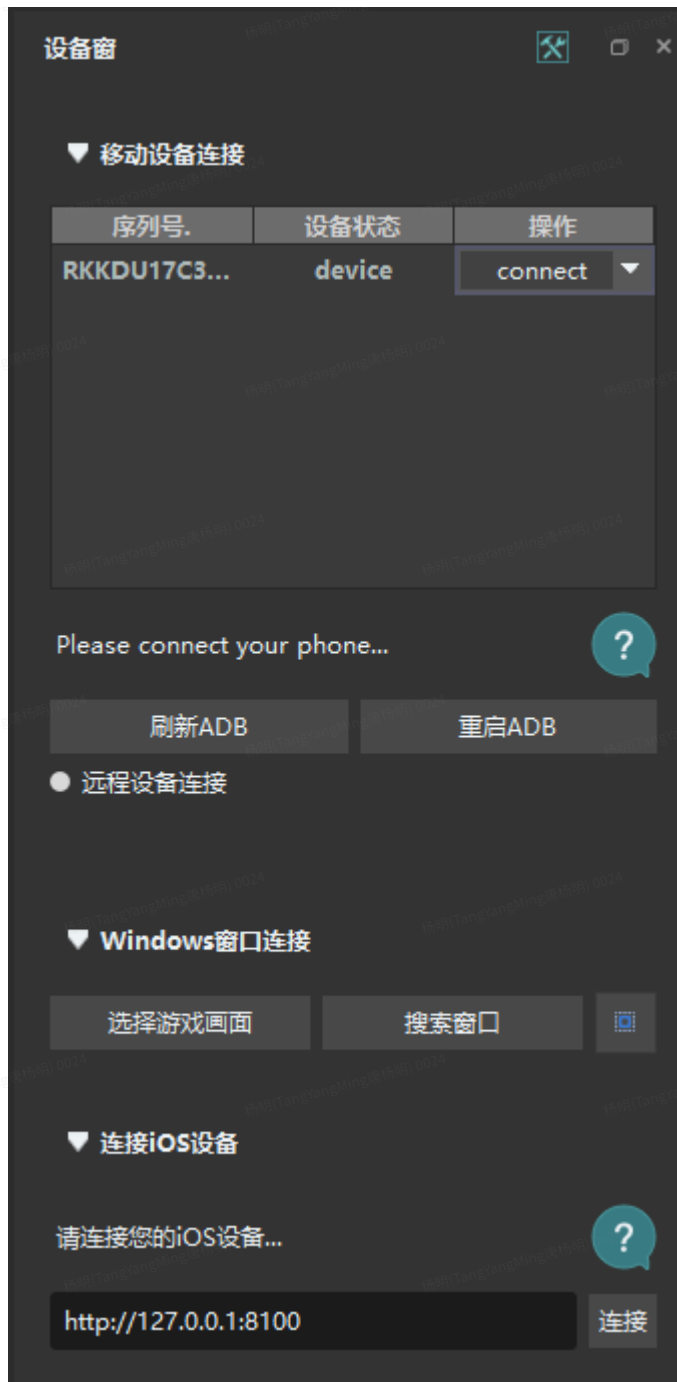
5. log查看窗

log查看窗有2个功能，1个是在检索控件的时候，查看控件的属性详情，另一个就是在运行脚本的时候查看运行log（主要是查看运行log，排查问题）



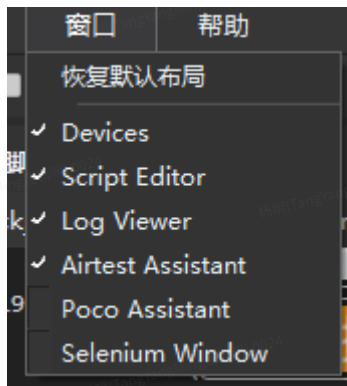
6. 设备连接窗口

在设备连接窗口，我们可以连接Android真机、模拟器、iOS真机、Windows窗口等多种类型的待测设备。具体的连接方式我们可以参看官方IDE使用教程的[设备连接章节](#)。



7. AirtestIDE的主界面交互

AirtestIDE的主界面是由多个可dock窗口组合而成的，我们可以按照自己的开发习惯和实际需要进行灵活拖拽布局，并可以通过下拉菜单"窗口"-"恢复默认布局"将内部布局恢复为默认状态。



布局信息会在软件关闭时会自动保存，下次重新启动软件/下载新版本软件使用时，之前的布局记录会自动延续。

恭喜你 你已经对AirtestIDE有一个初步的了解了