**HƯỚNG DẪN SETUP PROJECT AUTO TEST**

1. **Cài đặt môi trường**

* Cài đặt Python (3.3 trở lên): <https://python.org/downloads>
* Cài đặt AirtestIDE: <https://airtest.netease.com>
* Yêu cầu Developer:
  + Thêm WebSocketServer vào base client, thêm poco sdk vào source code javascript của game cần test
  + Thêm server riêng cho việc dev auto để không ảnh hưởng tới việc test build bình thường
  + Server thêm lib để có thể Cheat Time từ admin tool
  + Đặt tên các object cần thiết trong game
  + Commit config json lên svn
* Tìm hiểu thêm về Airtest và Poco tại:
  + <https://airtest.readthedocs.io/en/latest/>
  + <https://poco.readthedocs.io/en/latest>

1. **Cài thêm một số lib của python:**

Mở command line của window, sử dụng cú pháp: **pip install tên\_lib** để cài đặt thư viện (cài từng thư viện một). Các thư việc cần cài: **–U airtest, pocoui, xlrd, pandas, xlsxwriter**

Sau đó mở Setting của Airtest, setup đường dẫn Python path trỏ đến file python.exe đã cài ở mục 1.

A screenshot of a computer

Description automatically generated

1. **Sửa file Parser.py trong lib của airtest**

Copy file **Parser.py** trong Main.air/lib và paste đè vào file Parser.py trong lib của Airtest ở đường dẫn: Thư\_mục\_cài\_python\Lib\site-packages\airtest\cli\parser.py

Ex: C:\Users\pc2\AppData\Local\Programs\Python\Python38\Lib\site-packages\airtest\cli\parser.py

1. **Bật chế độ debug của Device hoặc LD player (giả lập android):**

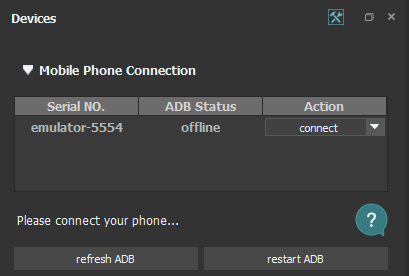
Mặc định LDPlayer sẽ không bật tính năng USB debug và cho connect ra bên ngoài, cần phải thực hiện 2 bước theo hình dưới để bật tính năng này.

A screenshot of a computer

Description automatically generated with medium confidence

1. **Connect device**

* Mở LDPlayer hoặc kết nối device thật với máy tính
* Mở Airtest sẽ thấy device tự động hiển thị trong list Mobile. Click connect ở device tương ứng.



1. **Set up các test case cần chạy:**

Mở file **Testcase.xls**, chỉnh sửa các thông tin trong sheet TestCase để chạy multi test case. Cụ thể:

* **Device:** Thứ tự của device khi chạy test case từ nhỏ đến lớn tương ứng với tên của device cũng từ nhỏ đến lớn.
* **Tên các cột tiếp theo:** Điền theo tên function của feature/event cần test
* **Các giá trị True/False**: Đánh dấu có thực hiện run test feature/event đó hay không

Ví dụ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Device** | **Login** | **VIP** | **DailyBonus** | **EventWC** | **EventWC\_1** | **EventWC\_2** |
| 1 | FALSE | FALSE | FALSE | TRUE | FALSE | FALSE |
| 2 | FALSE | FALSE | FALSE | FALSE | TRUE | FALSE |
| 3 | FALSE | FALSE | FALSE | FALSE | FALSE | TRUE |

Khi config như trên thì sẽ thực hiện test Event WC trên 3 device với 3 kịch bản khác nhau. Nếu muốn các device chạy cùng 1 kịch bản thì chỉ cần config trên 1 cột.

1. **Kịch bản test**

Kịch bản test chi tiết của từng feature/event sẽ được list riêng ở từng sheet với tên tương ứng với từng cột trong sheet TestCase. Như sau:

Table

Description automatically generated with medium confidence

Mỗi dòng tương ứng với 1 function được gọi từ project. Chỉ những function nào có giá trị Test = True mới được gọi. Mỗi function ở đây, cần được bắt đầu ở Lobby và kết thúc cũng ở Lobby để đảm bảo việc call 1 function bất kì hay nhiều function liên tiếp không bị lỗi.

1. **Cách thức ghi log**

Mỗi khi muốn ghi log vào file report, tùy vào mục đích ghi log có thể call 1 trong 3 function nằm trong Features.py như sau:

* **CheckImgExists** (caseId, step, content, isPoco = True, isExists = True, timeWait = 1): Kiểm tra 1 hình ảnh có tồn tại trên màn hình hay không.
  + **caseId:** Tương ứng với mỗi trường hợp cần test trong kịch bản test.
  + **Step:** Nội dung ghi log gợi nhớ bước đó thực hiện hành động gì.
  + **Content:** Tên hình ảnh cần check.
  + **isPoco:** Hình ảnh (content) đang check có phải là content Poco hay là content ảnh chụp. Vì việc kiểm tra sự tồn tại (exists) của 2 loại content này dùng 2 function khác nhau.
  + **isExists:** là kết quả mong muốn của QC.
    - True là mong đợi hình ảnh đó CÓ show trên màn hình.
    - False là mong đợi hình ảnh đó KHÔNG show trên màn hình.
  + **timeWait:** Thời gian chờ airtest kiểm tra content**.**
* **CheckTxtExists** (txtInGame, txtCompare, caseId, step): Kiểm tra các nội dung không phải là hình ảnh (văn bản, số) có đúng hay không.
  + **txtInGame:** Là nội dung hiển thị trong game hoặc giá trị của các param trên server.
  + **txtCompare:** Là kết quả mong muốn của QC.
  + **caseId:** Tương ứng với mỗi trường hợp cần test trong kịch bản test.
  + **step:** Nội dung ghi log gợi nhớ bước đó thực hiện hành động gì.
* **WriteLogRunning** (caseId, step, content, isImg, isPass, txtInGame = ""): Thực hiện việc ghi log khác mà không so sánh hình ảnh hay nội dung gì. Khi đó cần truyền vào **content = “”, isImg = False**. Các param còn lại tương tự 2 function trên.

Tham khảo thêm cách call các function này trong project demo.

1. **Run & Report:**

Bật device test lên, mở apk và login vào game, sau đó chạy file run.bat trong thư mục project hoặc F5 từ **Start.py** trong Airtest. Tool sẽ chạy các test case được config ở phần 5.

Report sau mỗi lần Test sẽ đc lưu ở file excel trong thư mục **Report** với format tên: **Device XX\_DDMM\_HHMMSS**. Trong đó XX là id của Device test, DDMM HHMMSS lần lượt là ngày tháng giờ phút giây start test.

Text

Description automatically generated

Graphical user interface, table

Description automatically generated

Hình ảnh bug sẽ được lưu trong thư mục **Snapshot** với format tên tương tự như file Report.