Piña Colada with AutoCoconut

Lukáš Růžička

August 18, 2020

What is AutoCoconut?

AutoCoconut is a mouse and keyboard events logging and screenshots taking application that helps to easily record a certain **workflow**.

How does it work?

The application . . .

1. runs in the background
2. records mouse and key events, takes screenshots
3. saves all such information in a json file (raw)
4. interprets the recorded data
5. wraps them into a file for output (json, adoc, html, openqa)

Parts of the application

- `autococonut.py`
- `event_handler.py`
- `screenshot_grabber.py`
- `interpreter.py`
- `translator.py`

event_handler.py

- uses pynput to monitor the input
- records each single event (mouse and key)
- classifies keys into groups (alphanumeric, special, modifiers)
- matches events and screenshots
- creates a json file with “raw” events

The ‘raw’ json format

```
"1597741812.6215603": {
    "type": "mouse",
    "action": "click",
    "button": "left",
    "coordinates": [1805,1418],
    "screens": [
        "1597741812.6215603.png",
        "1597741811.6215603.png"]
},
```

interpreter.py

- iterates over the ‘raw’ json file
- searches for patterns
- converts ‘raw‘ events into **super events**
- creates a new json with super events

Interpreter super events

- click and double click
- scroll and drag
- typing
- special keys
- key combinations

The super events json file

```
"1597745423.9703004": {
    "type": "typing",
    "subtype": "text",
    "action": "type",
    "key": "o",
    "screens": [null,null],
    "combined": null,
    "text": "useradd pkolinko",
    "reason": "special key pressed"
},
```

translator.py

- iterates over the super events json file
- retouches some of the events to comply with the selected output
- creates overlays for the screenshots
- creates needle json files for OpenQA output

autococonut.py

- runs the application
- deals with console arguments
- produces output
- uses the **Killer Ninja** templates[1] to produce selected output.

[1] I know they are called *Jinja*.

Possible output (currently)

- raw json
- super events json
- Asciidoc file
- HTML file
- OpenQA test subroutine
- possibly more

How to use it?

1. run the script
2. prepare your workplace
3. press the **stop key** to start listening
4. do your workflow
5. press the stop key to stop listening
6. get the results

Useful CLI options 1

--stopkey a dedicated key to start and stop listening (F10)

--offset a time offset in seconds to delay or precede the screenshots (1)

--caption the title of the output document (AutoCoconut – workflow report)

Useful CLI options 2

- **--file** the name of the output file and directory to save the screenshots (none)
- **--output** one of the possible outputs (adoc)

Search for Audi A4

- look for an available used A4
- clicking, typing, special keys
- asciidoc output
- overlain screenshots

Add to Gnome Calendar

- add an event into the Gnome Calendar
- clicking, typing, special keys, key combinations
- openqa output
- openqa needles
- successfully tested in OpenQA

Test basic calculations

- test basic operations
- mouse clicking, special keys
- openqa output
- openqa needles
- successfully tested in OpenQA

Q&A

Speak now or forever hold your peace.