Sikuli Guide

This guide provides a compressed version of Sikuli’s official documentation. This guide should provide you with enough knowledge to solve all the lab and homework tasks. It is also intended to be not overbeared with technical information and to have important information in easy-to-understand form.

Sikuli forum in case of questions or problems - https://answers.launchpad.net/sikuli

Basics

`find(img_string OR Pattern)` - searches the given region (or the whole screen) for the best match of the given image (can be string or Pattern(string)). Throws FindFailed exception if the match was not found.

`exists(img_string OR Pattern)` - searches for the match with similar argument as in `find()`. If match was found - returns the match, if not - returns None.

`click(img_string OR Pattern OR Match OR Location)` - finds the match of an image (if string or Pattern are used as an argument); performs a left-click on the position of the match.
E.g. `click(image) = click(Pattern(image)) = click(find(Pattern(image))) = click(Match) = click(Location)

`doubleClick(img_string OR Pattern OR Match)` - same as `click()`, but clicks twice.

`rightClick(img_string OR Pattern OR Match)` - same as `click()`, but right-click instead of left-click()

`dragDrop(img_string OR Pattern OR Match, img_string OR Pattern OR Match)` - finds the match of images (if necessary); simulates dragging and dropping from location of the first match to the second.

Image settings

Matching Preview - shows all matches that will be found for the current image. In this tab you can modify the similarity of the image with a slider. This can also be done with `Pattern(path_to_image).similar(int)

Target Offset - shows current offset of the image. Can also be done with `Pattern(path_to_image).targetOffset(x, y)`
**Region**

Region is a rectangular area on the Screen, that is defined by x, y, w, h (x and y being the coordinates for the upper left corner of the region, w - width in pixels, h - height in pixels). Basic methods like click(), find(), dragDrop() etc. are used on a region to narrow the testing (e.g. someReg.click()). If a method is used without a region then the default region (the whole screen) will be used. Region itself is taken from Screen class, however if you don’t use multi monitor setup, you won’t need to use it (everything will be called on default Screen, which is the only one).

**Constructing a region:**

\[ r = \text{Region}(x, y, w, h) \]

Match that is returned by find() can also be used as a region. Coordinates of the match (or region) can be extracted as follows: m.x, m.y.

All of the following methods are region methods.

**findAll(img_string OR Pattern)** - same as find() but return an iterator object of all matches that were found. Throws FindFailed if none were found. Example:

```python
matches = findAll(...) for m in matches: m.click()
```

**findAllList(img_string OR Pattern)** - same as findAll(), but returns a list instead of an iterator object. Doesn’t throw FindFailed if no matches were found. Instead returns an empty list.

**wait(img_string OR Pattern, seconds)** - wait for a given amount of time until a given image appears in the region (if amount of time is not specified, waits forever).

**waitVanish(img_string OR Pattern, seconds)** - same as wait(), but waits until a given image disappears from the region.

**highlight(seconds)** - highlights a region giving it a red border for specified amount of time. Good for debugging your tests.

**hover(img_string OR Pattern)** - same as dragDrop(), but doesn’t perform any clicks.

**above(range), below(range), left(range), right(range)** - creates a new region that is defined with respect to the given region. In case of above(): the new bottom edge will be the next pixel row of given region’s top edge (if range is not given).
grow(range) - expands (range > 0) or shrinks (range < 0) the given region in all 4 directions. Can also be used as grow(width, height) and grow(left, right, top, bottom).

Text manipulation
Those are all region methods.

collectWords() - extracts the text contained in the region as a list of matches in order from top left to bottom right. To get the text from the match use .getText(). NB! Words here mean character sequences, so anything that looks like a symbol and is separated by spaces, will be a word.

collectLines() - same as collectWords(), but finds lines of words.

collectWordsText() and collectLinesText() does the same, but returns a list of strings instead of matches.

type(string) - simulates typing a sequence of keys on the keyboard.
paste(string) - pastes the string at once. Can be useful to “empty” the clipboard.

Other classes and functions:

Pattern(img_string) - object that is created from an image and passed to functions like find(), click() etc.
Location(x, y) - can be used to click(), dragDrop() etc. to specific point. When click(“img.png”) is used, Location() is created from Match’es x and y and then used to click.
Env.getMouseLocation() - returns current location of the mouse as Location().
Env.getClipboard() - returns currently copied string.
Possible problems that you might encounter

You will get this error if you use a region that is way too small.

*CvException: CvException [org.opencv.core.CvException: cv::Exception: OpenCV(3.4.2)
C:\build\3_4_winpack-bindings-win64-vc14-
static\opencv\modules\imgproc\src\templmatch.cpp:1107: error: (-215:Assertion failed)
_img.size().height <= _templ.size().height && _img.size().width <= _templ.size().width in
function 'cv::matchTemplate'

FindFailed error when the image is clearly on screen:
After changes on screen it is usually recommended to use time.sleep() (even for 0.2
seconds) or wait(), because changes seem to mess with the Sikuli’s image recognition
algorithm.