mc-autofisher Documentation

Release 1.1.1

Robert Best

Jan 25, 2019

Contents

Automated Fishing Program for Minecraft

CHAPTER 1

Command Line Usage

```
python -m mc-autofisher [--levdist x] [--tesspath x] [-s|--new-bbox] [...]
```

Arguments:

`--levdist x`: Integer (default 5) The maximum Levenshtein distance allowed for text matching, accounts for slight errors in Tesseract's text recognition, i.e. 'Fishing Gobber' instead of 'Fishing Bobber'

`--tesspath x`: String The path to your Tesseract installation, usually unnecessary unless you have some special setup

`--new-bbox` or `-s` Tells the program you want to run the screengrabber and record the new bounding box in the config file. The screengrabber will run by default if there is not a bounding box already recorded in the config file and you have not manually provided coordinates on the command line

Other Parameters: If four flagless arguments are present they will be treated as manually-entered coordinates of the top left and bottom right corners of the bounding box and will take precedence over `--new-bbox` and the coordinates in the config file

Examples:

```
$ python -m mc-autofisher                              # Runs with␣
↪parameters from config file

$ python -m mc-autofisher 50 50 100 100                # Runs with␣
↪specified bounding box, all other parameters from config file

$ python -m mc-autofisher --levdist 1                  # Runs with a␣
↪maximum Levenshtein distance of 1

$ python -m mc-autofisher --tesspath /home/Desktop     # Why would you␣
↪install Tesseract on your Desktop?

$ python -m mc-autofisher --new-bbox                   # Runs the␣
↪screengrabber first to allow selection of a new bounding box
```

(continues on next page)

(continued from previous page)

CHAPTER 2

mc_autofisher.fisher module

Automates Minecraft fishing by watching a portion of the screen for the subtitle text “Fishing Bobber splashes”, and double clicking the right mouse button when it is found, allowing it to reel in and recast the line every time a fish appears.

mc_autofisher.fisher.**match**(*screenText*, *targetText=’Fishing Bobber’*, *threshold=5*)

Takes in some text and iterates through it line by line, checking to see if any of the lines contain something similar to the target text using Levenshtein distance

Parameters

- **screenText** (*str*) – The text to search for a match in
- **targetText** (*string*, optional) – The text to search for, defaults to “Fishing Bobber”
- **threshold** – (*int*, optional) The maximum acceptable Levenshtein distance to be considered a match, defaults to 5

Returns True if a match was found, False otherwise

Examples

```
>>> match("Fishing Bobber")
True
>>> match("Fishing Gobber")
True
>>> match("Fishing")
False
>>> match("Some text Fishing Bobber some more text")
True
```

mc_autofisher.fisher.**start**(*bbox*, *allowed_error=5*, *tesspath=”*)

Takes in bounding box coordinates and begins watching that section of the screen for the text “Fishing Bobber splashes”, double clicking the right mouse button when it sees it to reel in the fish and recast the line. Press enter to quit.

Parameters

- **bbox** (*list*) – A list of 4 numbers (x1, y1, x2, y2) obtained from the screengrabber program, where (x1, y1) is the top left corner of the bounding box and (x2, y2) is the lower right corner
- **allowed_error** (*int*, optional) – The margin of error (measured in Levenshtein distance) allowed for text matching, i.e. 'Fishing Gobber splashes' has an error of 1
- **tesspath** (*str*, optional) – The path to your Tesseract installation, leave blank to use default

CHAPTER 3

mc_autofisher.screengrabber module

Allows the user to select an area of the screen that the autofisher will watch for subtitles. When the *grab()* method is called, it will open a Tkinter window where the user can click and drag a rectangular area whose coordinates will then be returned

`mc_autofisher.screengrabber.`**`grab`**`()`

Opens a Tkinter window and allows the user to click and drag a rectangular selection, then returns the coordinates of the selection

Returns A list of 4 numbers (x1, y1, x2, y2), where (x1, y1) is the top left corner and (x2, y2) is the lower right corner

Return type list

Python Module Index

Index