

OOTP Update

2024/07/09

Update since 2024/07/04

With the assist of Optical Character Recognition to verify that each step is performed correctly

- `pip install -r requirements.txt`

Ignore the error if there is a problem installing `pytesseract`

Click

- Install `pytesseract` package (by homebrew)
 - Windows OS can ask 鄭竹淇
 - Mac (in cmd):
 - `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
 - `brew install tesseract`
- Update `pytesseract.pytesseract.tesseract_cmd` @ `utils.py` line 151
 - Mac: use `which tesseract` in cmd (Apple silicon: `"/opt/homebrew/bin/tesseract"`)
- `auto_click`: click and verify
- `normal_click`: only click, use it when nothing really change (need `sleep`)
- `while not ... enter()`: ref. `create_game`, `nice` & `ok` in `utils.py`. A little logic is required here (the opposite logic from `auto_click`), but it shouldn't be too hard

```
class Click:  
    """  
    3 Instance attributes (item.position, item.region, item.word)  
    2 Instance method  
    1 Terminate  
    """  
  
    def __auto_click(self):  
        """  
        same as before, but you might need to change the setting to that fit  
        your computer the most  
        """  
  
        ...  
  
    def __identify_word(self):  
        """  
        Use pytesseract to identify the word in item.region  
        """
```

```

if self.word == self.identify_word:
    pass
else:
    click again
.....
.

def terminate_simulator(self):
    .....
    if identify fail more than max_failures (Default 10, tunable),
    terminate code automatically
    .....
.
.
```

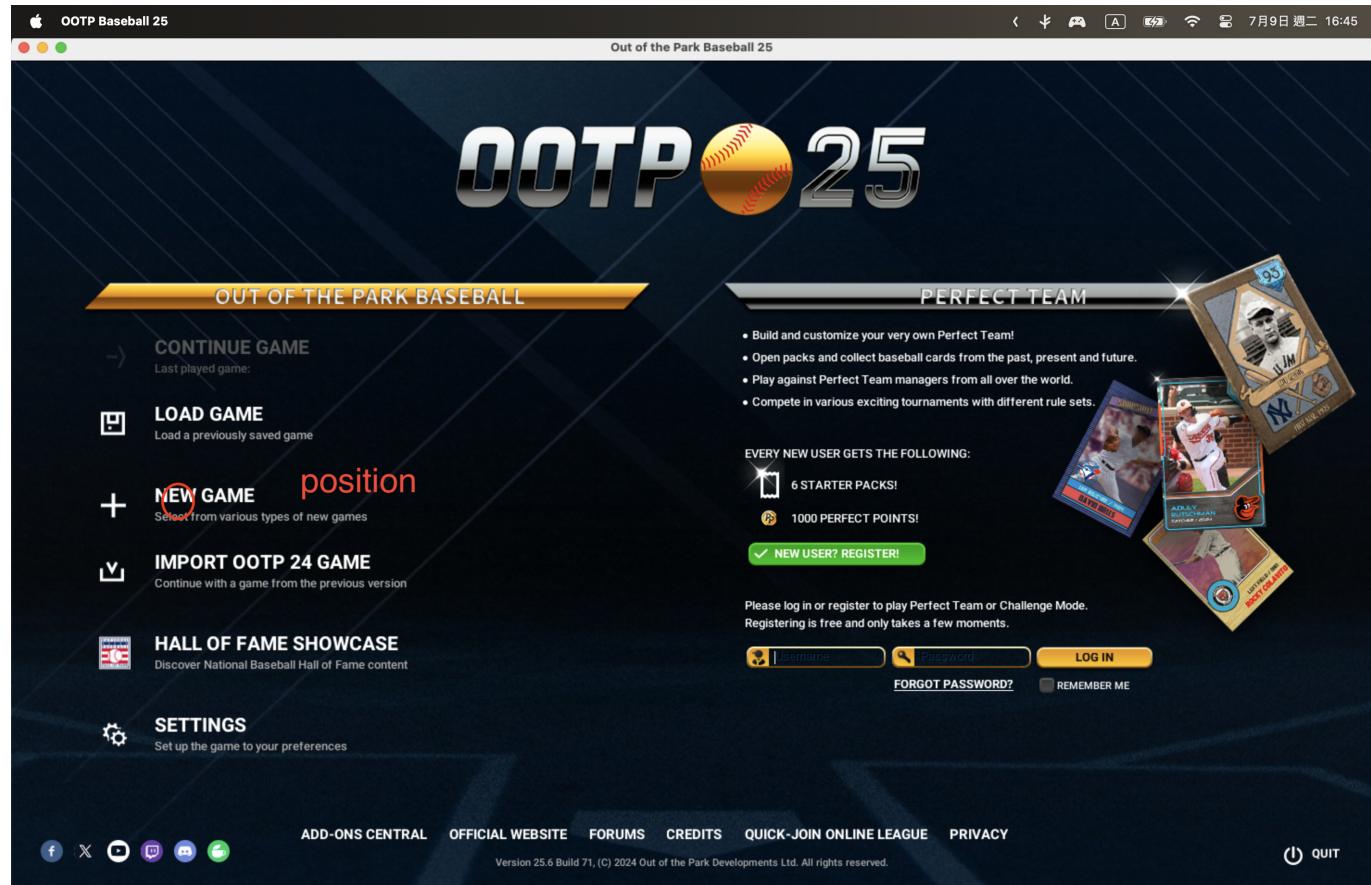
Module

Represent each item to click, contain position, region, and word

```

class module:
    def __init__(self, position, region, word) -> None:
        self.position = position
        self.region = region
        self.word = word
.
```

`module.position`: stands for where to click (same as before)

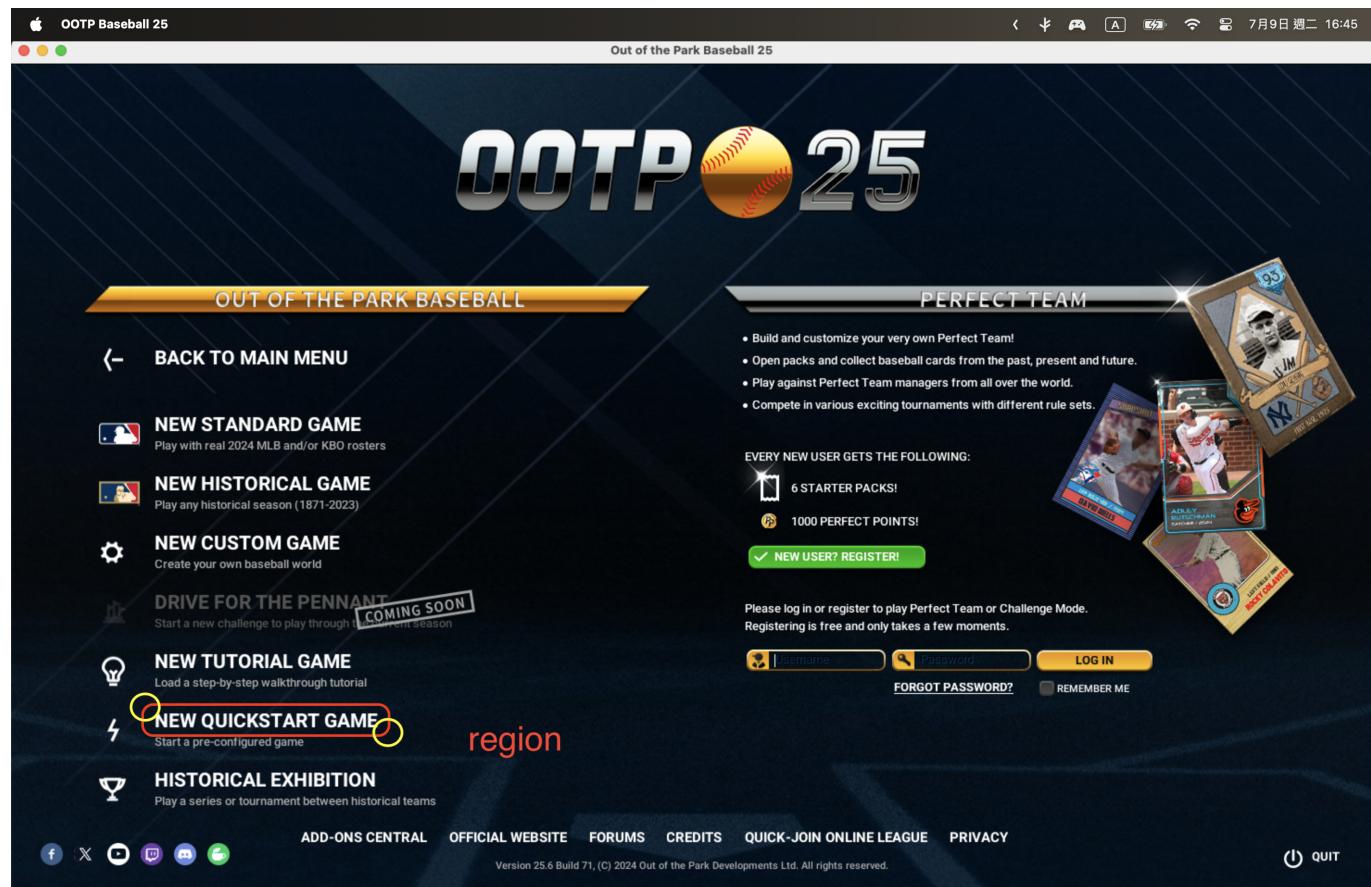


module.region:

- stands for the region to verify whether **click** has properly executed. It should contain **words that only appear after click**
- **(left_x, top_y, right_x, bottom_y)**: the x&y of that two yellow points in Example 1
- It is **safer** to get the region as **big** as possible, as long as not containing other irrelevant words

module.word: stands for the **words** inside the region

```
# Example 1
FirstGame_new_game = module(
    position = (181, 532),
    region = (147, 757, 416, 784), ## Get NEW QUICKSTART GAME -> which
means FirstGame_new_game had successfully click
    word = "NEW QUICKSTART GAME"
)
```



```
# Example 2
play_button = module(
    position = (901, 85),
    region = (834, 142, 919, 160), ## Get "Finish Today!" after "Play"
was clicked
    word = "Finish Today!"
)
```

The screenshot shows the Minnesota Twins lineup for a game against the Detroit Tigers. The lineup table includes columns for POS, #, Name, Inf, Mor, Age, Nationality, Height, Weight, Bats, Throws, and O. A context menu is open, highlighting the 'Finish Today!' option. The interface also shows various game management and simulation tools.

POS	#	Name	Inf	Mor	Age	Nationality	Height	Weight	Bats	Throws	O
C	27	Ryan Jeffers			26	USA	6' 4"	230 lbs	Right	Right	★
C	8	Christian Vázquez			33	PUR	5' 9"	195 lbs	Right	Right	★
1B	47	Edouard Julien			24	CAN	6' 2"	195 lbs	Left	Right	★
2B	50	Willi Castro			26	PUR	6' 1"	205 lbs	Switch	Right	★
3B	23	Royce Lewis			24	USA	6' 2"	200 lbs	Right	Right	★
SS	4	Carlos Correa			29	PUR	6' 4"	215 lbs	Right	Right	★
LF	38	Matt Wallner			26	USA	6' 5"	220 lbs	Left	Right	★
CF	25	Byron Buxton			30	USA	6' 2"	190 lbs	Right	Right	★
RF	26	Max Kepler			31	GER	6' 4"	220 lbs	Left	Left	★

Get region

- You can use `get_position_region.py` to get the region

```
# get_position_region.py
```

- Point at the Top Left corner
-> Position get
- Move to the Right Bottom corner
-> Position get

```
return region (tuple of 4)
```

- You can use `test_region.py` to check what will the computer get in that region

Change batter sequence

The place to change batter sequence has changed.

The screenshot shows the Out of the Park Baseball 25 software interface. The top navigation bar includes FILE, GAME, JIM SMITH (with a red notification dot), MLB, MIN, PLAY, PRESEASON (MON. MAR. 18TH, 2024), and MINNESOTA (0-0, .000 PCT, - GB - 2nd in the AL Central Division). Below the navigation is a green 'CONTINUE' button with 'Finish today.' and a dropdown menu for CUSTOM LINEUPS and ACTIONS... The main content area displays the Minnesota Twins lineup for a game against a Right-handed Pitcher and Designated Hitter (RHP+DH). The lineup table includes columns for POS, #, Name, Inf, Mor, Age, Nationality, Height, Weight, Bats, Throws, Overall Potential, Salary, Years left, MLY, and Scout Acc. The players listed are Ryan Jeffers, Christian Vázquez, Edouard Julien, Willi Castro, Royce Lewis, Carlos Correa, Matt Wallner, Byron Buxton, and Max Kepler. A red box highlights the first four players in the lineup: Willi Castro, Byron Buxton, Matt Wallner, and Royce Lewis.

#	B	Lineup vs. RHP+DH	POS	Depth Starter	Utility 1	Starts	Utility 2	Starts	Defense Sub.	#	Pinch Hitters
1	S	Willi Castro	2B	C. Vázquez	R. Jeffers	Ev. 15th Game				1	
2	R	Byron Buxton	CF	E. Julien						2	
3	L	Matt Wallner	LF	W. Castro						3	
4	R	Royce Lewis	3B	R. Lewis						4	
5	L	Edouard Julien	1B	C. Correa	W. Castro	If Starter tired				#	Pinch Runners
6	R	Carlos Correa	SS	M. Wallner						1	
7	L	Max Kepler	RF	B. Buxton						2	
8	R	Ryan Jeffers	DH	M. Kepler						3	
9	R	Christian Vázquez	C	R. Jeffers						4	

Notice

1

If you are not familiar with the process, you can print everything out in cmd. Please ref. 終端機輸出記錄.txt

```
## Sucess at the first try
Sucessful identify NEW QUICKSTART GAME
```

Word: Attention!

Identify word: ## Fail to identify words at the first try (Game is still loading) -> click again

Sucessful identify Attention! ## Sucess at the second try

2

I removed almost all sleep because OCR perform solid on my computer, but feel free to add it back.