Praca z ekranem

Twoje programy stosujące automatyzację GUI nie muszą ślepo klikać i wprowadzać treści. Moduł `pyautogui` ma możliwość wykonania zrzutu ekranu, czyli utworzenia pliku obrazu na podstawie aktualnej zawartości ekranu. Funkcjonalność powoduje również zwrot obiektu typu `Image` przedstawiającego bieżący wygląd ekranu. Jeżeli tylko przeglądałeś zawartość tej książki, powinieneś zapoznać się z rozdziałem 17. i zainstalować moduł `pillow`, zanim będziesz kontynuować pracę i wykonywać przykłady przedstawione w tym podrozdziale.

W systemie Linux konieczne jest zainstalowanie programu `scrot`, aby można było używać oferowanej przez moduł `pyautogui` funkcji tworzenia zrzutu ekranu. Instalację tego programu możesz przeprowadzić w powłoce za pomocą polecenia **`sudo apt-get install scrot`**. Jeżeli używasz systemów Windows lub macOS, pomiń ten krok i kontynuuj lekturę.

Wykonanie zrzutu ekranu

W celu wykonania zrzutu ekranu w Pythonie należy wywołać funkcję `pyautogui.screenshot()`. W powłoce interaktywnej wprowadź przedstawione poniżej polecenia.

```
>>> import pyautogui
>>> im = pyautogui.screenshot()
```

W powyższym fragmencie kodu zmienna `im` zawiera obiekt typu `Image` przedstawiający wykonany zrzut ekranu. Teraz w zmiennej `im` możesz wywoływać metody obiektu `Image`, podobnie jak w każdym innym obiekcie tego typu. W powłoce interaktywnej wprowadź przedstawione poniżej polecenia.

```
>>> im.getpixel((0, 0))
(176, 176, 175)
>>> im.getpixel((50, 200))
(130, 135, 144)
```

Funkcji `getpixel()` przekazujemy krotkę współrzędnych, taką jak (0, 0) lub (50, 200), a wartością zwrotną będzie kolor piksela znajdującego się na obrazie w punkcie o podanych współrzędnych. Wartość zwrotna funkcji `getpixel()` to krotka RGB składająca się z trzech liczb całkowitych określających ilość składowej koloru czerwonego, zielonego i niebieskiego w danym pikselu. (Nie ma tutaj czwartej wartości kanału alfa, ponieważ obraz zrzutu ekranu nie zawiera przezroczystości). W taki sposób program może „zobaczyć", co aktualnie znajduje się na ekranie.

Analiza zrzutu ekranu

Przyjmujemy założenie, że jednym z kroków w programie stosującym automatyzację GUI jest kliknięcie szarego przycisku. Przed wywołaniem metody `click()` możesz wykonać zrzut ekranu i spojrzeć na piksel, który ma być kliknięty przez skrypt. Jeżeli nie będzie to taki sam kolor szary jak na przycisku do kliknięcia, wówczas program „wie", że coś jest nie w porządku. Być może okno zostało nieoczekiwanie przesunięte lub wyskakujące okno dialogowe przysłoniło przycisk. Na tym etapie zamiast kontynuować operację — i prawdopodobnie kliknąć nieprawidłowy element — program może „zobaczyć", że kliknięty będzie niepoprawny element i zatrzymać swoje działanie.

Funkcja `pixelMatchesColor()` modułu `pyautogui` zwróci wartość `True`, jeśli piksel w punkcie o podanych współrzędnych X i Y na ekranie ma określony kolor. Pierwszy i drugi argument tej funkcji to liczby całkowite definiujące współrzędne X i Y. Trzecim argumentem jest krotka trzech liczb całkowitych określających kolor RGB punktu na ekranie, który ma być dopasowany przez piksel. W powłoce interaktywnej wprowadź przedstawione poniżej polecenia.

```
>>> import pyautogui
>>> im = pyautogui.screenshot()
>>> im.getpixel((50, 200)) ❶
(130, 135, 144)
>>> pyautogui.pixelMatchesColor(50, 200, (130, 135, 144)) ❷
True
>>> pyautogui.pixelMatchesColor(50, 200, (255, 135, 144)) ❸
False
```

Po wykonaniu zrzutu ekranu i użyciu funkcji `getpixel()` w celu pobrania krotki RGB dla piksela znajdującego się w punkcie o wskazanych współrzędnych ❶ te same współrzędne i krotkę koloru RGB przekazujemy funkcji `pixel↪MatchesColor()` ❷, która powinna zwrócić wartość `True`. Następnie zmieniamy wartość w krotce RGB i ponownie wywołujemy funkcję `pixelMatchesColor()` wraz z tymi samymi współrzędnymi ❸. Wartością zwrotną powinno być `False`. Metoda okaże się użyteczna, gdy program stosujący automatyzację GUI będzie miał wywołać funkcję `click()`. Pamiętaj, że kolor w punkcie o podanych współrzędnych musi być *dokładnie* dopasowany. Jeżeli będzie nawet minimalnie inny, na przykład `(255, 255, 254)` zamiast `(255, 255, 255)`, wtedy wartością zwrotną funkcji `pixelMatchesColor()` będzie `False`.

Projekt — rozbudowa programu mouseNow.py

Możemy teraz rozbudować utworzony wcześniej w rozdziale program *mouseNow.py*, aby wyświetlał nie tylko współrzędne X i Y bieżącego położenia kursora myszy, ale również podawał kolor RGB piksela wskazywanego przez kursor myszy.

Kod zdefiniowany w pętli `while` programu *mouseNow.py* zmodyfikuj w pokazany poniżej sposób.

```
#! python3
# mouseNow.py — Wyświetla bieżące położenie kursora myszy.
--cięcie--
        positionStr = 'X: ' + str(x).rjust(4) + ' Y: ' + str(y).rjust(4)
        pixelColor = pyautogui.screenshot().getpixel((x, y))
        positionStr += ' RGB: (' + str(pixelColor[0]).rjust(3)
        positionStr += ', ' + str(pixelColor[1]).rjust(3)
        positionStr += ', ' + str(pixelColor[2]).rjust(3) + ')'
        print(positionStr, end='')
--cięcie--
```

Kiedy teraz uruchomisz program *mouseNow.py*, wygenerowane dane wyjściowe będą zawierały także wartość RGB piksela wskazywanego przez kursor myszy.

```
Naciśnij klawisze Ctrl+C, aby zakończyć działanie programu.
X:  406 Y:   17 RGB: (161,  50,  50)
```

Te informacje wraz z funkcją `pixelMatchesColor()` powinny ułatwić Ci dodanie do skryptów stosujących automatyzację GUI kodu sprawdzającego kolor piksela.

Rozpoznawanie obrazu

Co zrobić w sytuacji, jeśli wcześniej nie wiadomo, w którym punkcie trzeba będzie kliknąć? W takim przypadku można użyć rozpoznawania obrazu. Przekaż modułowi `pyautogui` obraz przeznaczony do kliknięcia i pozwól mu na ustalenie współrzędnych.

Jeżeli na przykład wcześniej wykonałeś zrzut ekranu w celu przygotowania przechowywanego w pliku *submit.png* obrazu przycisku *Prześlij*, wówczas funkcja `locateOnScreen()` zwróci współrzędne miejsca na ekranie, w którym został znaleziony ten obraz. Aby zobaczyć, jak działa funkcja `locateOnScreen()`, wykonaj zrzut niewielkiej części ekranu, zapisz plik obrazu na dysku, a następnie w powłoce interaktywnej wprowadź przedstawione poniżej polecenia. Nie zapomnij o zastąpieniu *submit.png* nazwą, pod którą zapisałeś plik.

```
>>> import pyautogui
>>> pyautogui.locateOnScreen('submit.png')
(643, 745, 70, 29)
```

Czteroelementowa krotka zwrócona przez funkcję `locateOnScreen()` zawiera informacje o współrzędnej X lewej krawędzi, współrzędnej Y górnej krawędzi, szerokości i wysokości pierwszego miejsca na ekranie, w którym został znaleziony