Adam Jeż

Projekt „Mrówka Langtona”

Celem projektu była implementacja automatu komórkowego Chrisa Langtona *„Mrówka Langtona”* na zasadach określonych w treści zadania:

- mrówka porusza się na planszy o określonym rozmiarze złożonej z pikseli, gdzie każdy piksel jest komórką w kolorze czarnym lub białym

- mrówka będąc na komórce czarnej, wykonuje obrót w prawo o kąt prosty, zmienia kolor pola na biały i przechodzi na następną komórkę

- mrówka będąc na komórce białej, wykonuje obrót w lewo o kąt prosty, zmienia kolor pola na czarny i przechodzi na następną komórkę

- gdy mrówka znajduje się na granicy planszy i próbuje za nią wyjść to przechodzi automatycznie na losową sąsiednią komórkę leżącą na planszy

Użytkownik wybiera po jakiej planszy ma się poruszać mrówka. Do wyboru dostępna jest jedna z trzech opcji:

a) plansza całkowicie biała o podanych rozmiarach przez użytkownika

b) plansza czarno-biała, gdzie użytkownik podaje rozmiar i prawdopodobieństwo wygenerowania czarnego piksela

c) plansza czarno-biała, którą użytkownik może załadować jako plik w formacie PNG.

Wyjściem programu jest seria obrazów przedstawiających ruch mrówki po planszy zapisanych w folderze „obrazy”. Użytkownik podaje ilość kroków, jakie ma wykonać mrówka.

W programie występuje klasa – *Ant*. Ma następujące atrybuty:

- osobno podawane współrzędne położenia mrówki na planszy (x, y), gdzie punktem początkowym planszy (0, 0) jest lewa, górna komórka

- kolor piksela na którym znajduje się mrówka (składający się z krotki trzech współrzędnych kolorów RGB (x, y, z))

- kierunek, w którym zwrócona jest mrówka (w postaci ciągu znaków), domyślnym kierunkiem początkowym jest góra („up”)

Głównymi metodami klasy jest pivot() i move_forward(), które służą odpowiednio do obrotu (zmiany kierunku) i przesuwania się (zmiany współrzędnych położenia).

Po uruchomieniu pliku startującego *interface_project_ant.py* wyświetla się nazwa projektu i trzy opcje działania programu po wpisaniu do konsoli terminala odpowiedniej cyfry. Użytkownik ma do wyboru:

1- Stwórz białą planszę o podanych rozmiarach
2- Zaimportuj gotowy obraz PNG, po którym mrówka będzie się poruszać
3- Stwórz czarno-białą planszę o podanych rozmiarach i prawdopodobieństwie wystąpienia czarnego piksela

Po wpisaniu cyfry *1* użytkownik jest proszony o podanie wymiarów planszy (dwóch liczb naturalnych oddzielonych spacją, nie mniejszych niż 2). Zalecane jest podawanie liczb z przedziału pomiędzy 50 a 1000 (większe wartości przydają się do stworzenia tak zwanej „*autostrady*”). (*) Następnie użytkownik podaje liczbę kroków, jakie ma wykonać mrówka. Po tym, użytkownik podaje co ile kroków mrówki ma być zapisany obraz do folderu (nazwy tych plików są numerowane kolejno przy którym kroku mrówki zostały zapisane). Po zakończeniu kroków mrówki użytkownik wpisuje cyfrę *1*, gdy chce

stworzyć animację GIF i następnie podaje prędkość z jaką będą się zmieniały obrazy (ilość obrazów na sekundę) lub wpisuje dowolny inny znak, jeśli chce od razu zakończyć działanie programu. Program się kończy, rezultaty „spaceru" są dostępne w folderze „*obrazy*". Program działa analogicznie, gdy po uruchomieniu użytkownik wpisze cyfrę *3*. Dodatkowo, na początku, należy podać prawdopodobieństwo (liczbę z przedziału [0,1]) z jakim przy tworzeniu planszy, wygenerowany piksel będzie koloru czarnego. W przypadku podania cyfry *2* użytkownik podaje nazwę pliku (bez cudzysłowu) w formacie *PNG* (składający się tylko z białych lub czarnych pikseli), który ma być użyty jako plansza (przykładowy obraz, dostępny w ramach sprawdzenia tej funkcjonalności to *image_mock.png*). Następnie podaje kolejne parametry od (*). Program usuwa zawartość folderu *obrazy* po każdym uruchomieniu.

Projekt posiada plik konfiguracyjny *config_project_ant.py* w którego skład wchodzą:

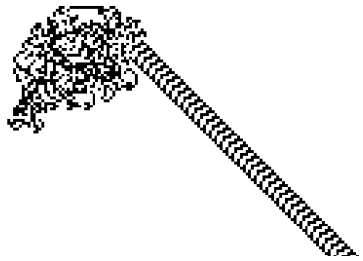
- dwie krotki zawierające współrzędne RGB kolorów (dla białego (0, 0, 0), dla czarnego (255, 255, 255) (odp. *WHITE, BLACK*)

- lista ciągów znaków opisujących kierunki mrówki (*directions*)

Refleksje:

Tworzenie projektu było rozwijającym i pouczającym doświadczeniem. Wykorzystanie nowo poznanych bibliotek , takie jak *numpy, PIL, imageio* pozwoliło połączyć algorytm chodzenia mrówki po planszy z jego wizualizacją w postaci obrazów. Dodatkowo zaimplementowana opcja tworzenia animacji, w łatwiejszy i atrakcyjniejszy sposób przedstawia typowe, jak również i nietypowe „wzory" tworzone przez mrówkę. Warto wspomnieć o powstawaniu tak zwanej „*autostrady*", która występuje po około 15 tysiącach, wydawałoby się, chaotycznych ruchów mrówki (widoczna na obrazie poniżej). Współrzędne początkowe mrówki są zawsze losowo wybierane, a więc, gdy mrówka będzie chciała wyjść poza planszę to wybiera losowy inny kierunek. Dodatkowa opcja, by zapisywać co daną liczbę kroków obraz, pozwala na wykonanie bardzo dużej ilości kroków bez potrzeby zapisu bardzo wielu plików. Również możliwość usuwania zawartości folderu służy lepszemu zarządzaniu obrazami.

Program, oczywiście, można by było rozwinąć np. o bibliotekę *argparse*, która mogłaby uprościć podawanie parametrów.



Obraz „autostrady" stworzony przez mrówkę