

Mrówka Langtona

Opis algorytmu

Algorytm mrówki Langtona opiera się na śledzeniu, jak zachowuje się określona plansza po i iteracjach. Mrówka chodząc po planszy skręca w lewo gdy będzie na polu białym oraz skręca w prawo gdy natrafi na pole czarne. Poruszając się w ten sposób, jednocześnie zmieniając kolor pola, znajdującego się pod nią, algorytm rysuje chaotyczny, jednak deterministyczny rysunek, aby następnie stworzyć odnogę, którą mrówka będzie się poruszała do końca iteracji programu, bądź gdy natrafi na przeszkodę.

Wywołanie

Program jest uruchamiany z pliku wykonywalnego, stworzonego przez kompilator c. Plik ten nazwany jest „mrowka” i znajduje się on w folderze bin. Można go zaktualizować wywołując plik „Makefile”, znajdujący się w folderze głównym programu.

Plik wykonywalny jesteśmy w stanie uruchomić za pomocą polecenia „./bin/mrowka [n] [m] [i] [dir] [%]”, gdzie dane wejściowe odpowiadają: n – liczba wierszy, m – liczba kolumn, i – liczba iteracji, dir – kierunek początkowy mrówki [od 0 do 3], % - procent zapełnienia planszy czarnymi polami.

Podział modułowy

Program jest podzielony na 3 pliki z rozwinięciem „.c” oraz 2 pliki z rozwinięciem „.h”.

Plik „main.c” zawiera funkcję „main”, która odpowiada za czytanie danych wejściowych oraz wywołanie programu.

Plik „ant.c” odpowiada za tworzenie, destrukcję(zwolnienie z pamięci) oraz ruch mrówki. Odpowiadają za to odpowiednio funkcje: „create_ant”, „destroy_ant”, „step”. Powiązany z tym plikiem jest plik „ant.h” zawierający prototypy funkcji zdefiniowanych w „ant.c” oraz definicje stałych, które będzie używał ten program.

Plik „board.c” odpowiada za tworzenie, destrukcję(zwolnienie z pamięci) oraz wypisanie planszy po której będzie poruszać się mrówka. Za te działania odpowiadają odpowiednio funkcje: „create_board”, „destroy_board”, „save_board”. Powiązany plikiem jest „board.h”. Zawiera on prototypy funkcji zdefiniowanych w „board.c”

W całym programie znajduje się jedna własna struktura nazwana „Ant”. Zawiera ona 3 zmienne int. Oznaczają one odpowiednio: wiersz, kolumnę oraz kierunek; mrówki.

Przykłady

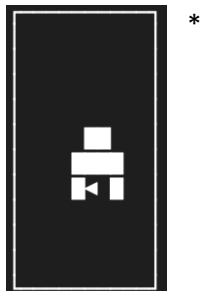
1.

Dla wywołania „./bin/mrowka 10 10 20 0 0”, stworzy się 20 plików w folderze „out”

```
MaciejK@MacBook-Pro-4 out % ls
1.txt  11.txt 13.txt 15.txt 17.txt 19.txt 20.txt 4.txt  6.txt  8.txt
10.txt 12.txt 14.txt 16.txt 18.txt 2.txt  3.txt  5.txt  7.txt  9.txt
```

Te pliki odpowiadają odpowiednim iteracjom przejścia mrówki i zawierają plansze wypełnione białymi oraz czarnymi polami. Zaznaczona jest też w nich pozycja mrówki.

Przykładowy wygląd pliku(16.txt):



2.

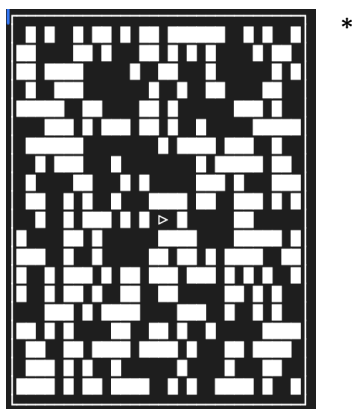
Gdy program zostanie wywołany z innymi danymi „20 30 30 1 50”, można zauważyć, że tworzy się 30 plików tak jak zakłada program.

```
MaciejK@MacBook-Pro-4 out % ls
1.txt  12.txt 15.txt 18.txt 20.txt 23.txt 26.txt 29.txt 4.txt  7.txt
10.txt 13.txt 16.txt 19.txt 21.txt 24.txt 27.txt 3.txt  5.txt  8.txt
11.txt 14.txt 17.txt 2.txt  22.txt 25.txt 28.txt 30.txt 6.txt  9.txt
```

Każdy z plików wygenerowany przez ten program będzie miał 20 wierszy oraz 30 kolumn.

Pierwsza iteracja czyli plik nazwany „1.txt” zawiera już czarne pola, co jest definiowane przez ostatni argument, tworząc planszę początkową z 50% czarnymi polami. Sama mrówka obrócona jest w prawo co definiuje 2 argument.

Przykładowy wygląd pliku(1.txt):



* - czarne kwadraty są zamienione na białe i na odwrót. Jest to spowodowane ciemnym motywem edytora tekstu

Wnioski

Algorytm Mrówki Langtona obrazuje jak chaotyczne ruchy mrówki zmieniają się w uporządkowane, po ok. 10000 iteracji. Jest to o tyle ciekawe, że jest to algorytm deterministyczny, czyli przy każdym wywołaniu, będzie wyglądał tak samo (nie wliczając 5 argumentu, który tworzy planszę losową).

Mimo ilości możliwych iteracji jakie może wykonać program wykonuje się on relatywnie szybko dla stosunkowo małej liczby wierszy i kolumn. Algorytm ten wykonuje się w złożoności $n*m*i$.