

## SPRAWOZDANIE MRÓWKA\_LANGTONA

### 1) Idea działania

Idea działania mrówki Langtona polega na tym, że na dwuwymiarowej siatce znajduje się mrówka, która może wykonać 2 akcje:

- Jeżeli mrówka znajduje się w komórce białej, obraca się ona o 90 stopni w prawo oraz zmienia kolor komórki na przeciwny (z białego na czarny) oraz przesuwa się o jedną komórkę do przodu.
  - Jeżeli mrówka znajduje się w komórce czarnej, obraca się ona o 90 stopni w lewo oraz zmienia kolor komórki na przeciwny (z czarnego na biały) oraz przesuwa się o jedną komórkę do przodu.
- Domyślnie siatka jest wypełniona białymi komórkami, lecz istnieją warianty gdzie na mapę naniesione są czarne pola (przeszkody) lub gdzie nawet na mapie znajduje się kilka mrówek. Jak mrówka ma wyjść poza granice siatki, to pojawia się z jej drugiej strony.

W wyniku iteracji akcji mrówki powstaje biała mapa z wzorami z czarnych komórek na niej.

### 2) Wywołanie programu

Plik wykonywalny programu (po wykonaniu polecenia “make all” aby skompilować program) znajduje się w folderze bin/. Wywołanie programu ma następującą postać:

`./bin/mrowka (-m) (-n) (-i) (-f) (-d) (-l) (-r)`

Flagi które można podać:

- (-m) [liczbe wierszy planszy] (przy wczytywaniu i tak sami liczymy więc nie trzeba podawac)
- (-n) [liczbe kolumn planszy] (tak samo liczymy sami przy wczytywaniu)
- (-i) [liczbe iteracji ktore ma przejsc mrowka]
- (-f) [przedrostek plikow wynikowych name]
- (-d) [początkowy kierunek mrowki] (startowa lokacja jest przyjmowana za srodek)
- (-l) [wczytanie mapy o podanej nazwie]
- (-r) [generowanie mapy, gdzie wartosc oznacza procent zapelnienia czarnymi polami]

Argumenty można podać w dowolnej kolejności, a przy nie podaniu któregoś z argumentów program przyjmuje wartości domyślne.

### 3) Podział programu na moduły:

- getarg – moduł przetwarzający argumenty wywołania korzystając z funkcji getopt I przechowujący je w strukturze
- plansza – moduł zawierający funkcje ruchu mrówki oraz strukturę mrówki oraz siatki
- rys – moduł zajmujący się wczytywaniem mapy oraz jej rysowaniem do pliku
- operacja – moduł zarządzający działaniem mrówki
- main – moduł zarządzający całym programem I wywołujący pozostałe funkcje
- przeszkody – moduł służący do wypełniania siatki losowymi przeszkodami według podanego przez użytkownika procentu

### 4) Opis funkcji I struktur:

- struktura argumenty – przechowuje argumenty wywołania
- funkcja getarg – przetwarza I wczytuje podane argumenty wywołania, w razie nie podania wartości przez użytkownika przypisuje podstawowe wartości
- struktura komorka – przechowuje stan komórki (1 to kolor czarny, a 0 to kolor biały). Siatkę przechowujemy jako dwuwymiarową tablicę tej struktury
- struktura mrow – przechowuje położenie mrówki I jej obecny kierunek
- funkcja mrowka\_operacja – zajmuje się akcją mrówki (zmianą kierunku, stanu komórki oraz zmianą położenia mrówki)
- funkcja wczytaj – wczytuje podaną przez użytkownika mapę

- funkcja wypisz – rysuje kolejne iteracje działania mrówki. W wypadku podania przez użytkownika nazwy pliku, zostaną one zapisane do oddzielnych plików z odpowiednim przyrostkiem, inaczej zostaną wypisane na stdout.
- funkcja przejscie\_mrówki – wywołuje funkcje wypisz oraz mrowka\_operacja w pętli w celu wykonania odpowiedniej ilości iteracji działania mrówki
- funkcja przeszkody – wypełnia siatkę losowo przeszkodami według podanego przez użytkownika procentu wypełnienia

### 5) Przykładowe działania programu:

a) wywołanie bez argumentów (domyślne wartości: plansza 10x10, 10 iteracji)

```
./bin/mrowka
```

**Wynik:**

Iteracja 0:

[illegible]

Iteracja 1:

[illegible]

Iteracja 2:

[illegible]

Iteracja 3:

[illegible]

Iteracja 4:

[illegible]

Iteracja 5:

[illegible]

Iteracja 6:

[illegible]

Iteracja 7:

				■	▷				
				■		■			
					■	■			

Iteracja 8:

				■	■				
				■	▽	■			
					■	■			

Iteracja 9:

				■	■				
				◀	■	■			
					■	■			

Iteracja 10:

				■	■				
					■	■			
				▽	■	■			

b) zapisywanie do pliku

np. ./bin/mrowka -m 3 -n 3 -i 5 -f file

W folderze powstają pliki: file\_0iteracji.txt, file\_1iteracji.txt, ... file\_5iteracji.txt.

Pliki mają następującą zawartość:

-file\_0iteracji.txt

	△	

-file\_1iteracji.txt

	■	▷

-file\_2iteracji.txt

	■	■
		▽

-file\_3iteracji.txt

	■	■
	◁	■

-file\_4iteracji.txt

	▲	■
	■	■

-file\_5iteracji.txt

◁		■
	■	■

c) Wywołanie z 20% wypełnieniem przeszkodami

./bin/mrowka -m 3 -n 3 -i 5 -r 20

Iteracja nr. 0:

	△	■
■		■

Iteracja nr. 1:

	■	►
■		■

Iteracja nr. 2:

		△
	■	
■		■

Iteracja nr. 3:

▷		■
	■	
■		■

Iteracja nr. 4:

■		■
▽	■	
■		■

Iteracja nr. 5:

■		■
■	■	◁
■		■

6) Wnioski:

Mrówka Langtona z początku tworzy pozornie losowe wzory, jednak po ponad 10 tysiącach zaczyna tworzyć schematyczne struktury przypominające most, a po tym mrówka zaczyna poruszać się po spiralach. Zatem z pozornie chaotycznego ruchu wyłaniają się konkretne struktury. Mimo prostych zasad poruszania się mrówki, ciężko jest przewidzieć jej zachowanie w przypadku dużej liczby iteracji.