Izabela Bargieł
Oliwia Woźniak

SPRAWOZDANIE

Mrówka Langtona

1. Idea działania

Mrówka Langtona to automat komórkowy, wymyślony przez amerykańskiego informatyka Christophera Langtona. Wirtualny owad skierowany jest w jedną z czterech stron (w naszym programie są to G – do góry, D – do dołu, P – w prawo i L – w lewo). Mrówka porusza się po dwuwymiarowej przestrzeni podzielonej na komórki według prostego schematu: jeżeli stanie na czarnej komórce wykonuje obrót o 90* w prawo, w przeciwnym wypadku o 90* w lewo, następnie zmienia kolor komórki, w której się znajduje na przeciwny i przesuwa się o 1 pole do przodu, tworząc w ten sposób wzór.

2. Wywołanie programu

Program uruchamia się przy pomocą komendy:

./ant <tu umieścić flagi oraz odpowiadające im parametry>

Przy wywołaniu program możliwe jest wprowadzenie siedmiu różnych parametrów, gdzie:

A) 4 z nich sa obowiązkowe do uruchomienia programu:

-m liczba wierszy;

-n liczba kolumn;

-k początkowy kierunek mrówki;

-i liczba iteracji;

B) 3 z nich są opcjonalne:

-f nazwa pliku zawierającego mapę;

-r prefix do nazwy plików wynikowych;

-p procent zapełnienia losowo wygenerowanej mapy, należy podać liczbę z zakresu <0, 100> (może być zmiennoprzecinkowa);

UWAGA! Jeżeli zostaną podane za razem nazwa pliku i procent zapełnienia planszy, to mapa pobrana z pliku będzie priroytetyzowana, a procent w tym przypadku będzie bezużyteczny

3. Moduły, podstawowe funkcje, struktury

Najważniejsze struktury:

struct mrowka

- struktura przechowująca parametry mrówki: współrzędne położenia oraz kierunek, w który jest skierowana

char** plansza

- tablica dwuwymiarowa przechowująca aktualny stan planszy i modyfikowana przez mrówkę

Funkcja main

- pobiera parametry do uruchomienia programu za pomocą flag
- po kolei uruchamia potrzebne funkcje
- przechodzi przez kolejne iteracje mrówki
- w zależności od potrzeb dodatkowo może:
  - generować losowy procent zapełnienia mapy
  - stworzyć i nazwać pliki uwzględniając podany prefix

Moduły:

Moduł 1 – fun1.c

Funkcje:

- *mrowka gen_mrówka(int m, int, n, char kier_wej) ;*

- generuje mrówkę za pomocą structury mrowka
- ustawia mrówkę na środku planszy
- ustawia kierunek mrówki

- *void rys_plansza(FILE &plik, int m, int n, char **plansza, mrowka ant);*
  - wpisuje aktualną iteracje mapy wraz z mrówka do pliku
- *void rys_plansza_stdout(int m, int n, char **plansza, mrowka ant);*
  - wypisuje aktualną iteracje mapy wraz z mrówką na standardowe wyjście

Moduł 2 – funkcje2.c

Funkcje:

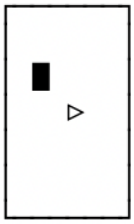
- *mrowka ruch (char** mapa, mrowka mr, int iteracje, int m, int n);*
  - funkcja odpowiedzialna za ruch mrówki
  - korzysta z funkcji przesun_do_przodu, aby zmienić współrzędne mrówki w zależności od jej kierunku
  - korzysta z funkcji skret_w_lewo oraz skret_w_prawo do zmiany kierunku mrówki
  - uaktualnia kolor planszy
  - zwraca strukturę mrowka z danymi po wykonaniu ruchu
- *char** gen_mapa(int m, int n, int procent);*
  - zwraca tablicę o wymiarach m x n z podanym procentem komórek zamalowanym na czarno (zmienna procent)
  - zaokrągla ilości zamalowanych komórek zawsze do dołu
- *char** czytaj_mape_z_pliku(int m, int n, FILE* plik);*
  - wczytuje mapę z podanego pliku do formatu używanego w programie
  - pobiera współrzędne początkowe mrówki z pliku, jeśli są podane

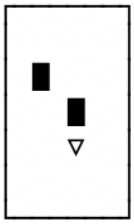
4. Przykładowe działanie programu

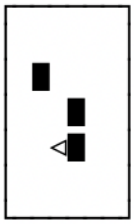
Wywołanie:

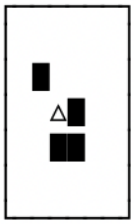
./ant -m 5 -n 6 -i 5 -k P

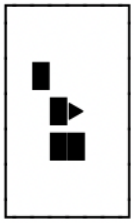
```
MacBook-Pro-3:mrowka_l oliwiawozniak$  ./ant -m 5 -n 6 -i 5 -k P
Parametry startowe:
wiersze: 5
kolumny: 6
kierunek: P
iteracje: 5
plik: (null)
prefix: (null)
proc: 5.658856
```











5. Wnioski

Praca nad największym z dotychczasowych projektów była pod wieloma względami ciekawym doświadczeniem.

Po pierwsze, ze względu na grupowy charakter pracy niesamowicie pomocnym narzędziem okazał się być Git Hub, usprawniając symultaniczne pisanie kodu i wprowadzając porządek w kontrolę wersji.

Podczas realizacji zadania poznałyśmy również nowe funkcjonalności języka C, między innymi niezwykle przejrzystą i skuteczną funkcję getopt służącą do pobierania argumentów wywołania oraz napotkałyśmy nową przeszkodę, wide chary, które po licznych próbach udało nam się pokonać.

Zagadnienie realizowane przez projekt, wizualizacja algorytmu, pokazało nam, jak wiele satysfakcji może dać wykorzystanie programowania do zautomatyzowania procesów. Możliwość obserwowania niezwykle zjawiskowych efektów swojej pracy była zdecydowanie elementem napędzającym nas do dalszego działania.