
Laboratorium

Programowania Komputerów

Temat: *Program wykonujący określone zadanie*

Autor: Jan Olszański

Informatyk, sem.1, grupa 1, sekcja 2

Prowadzący: mgr inż. Grzegorz Kwiatkowski

Github: <https://github.com/ElderDrop/PK/tree/trunk/Projekt>

Dropbox: https://www.dropbox.com/sh/oo3lwr8tbyjr16q/AACApDyBQqD_IfCvEG0R6Rja?dl=0

1. Temat

Napisać program, umożliwiający symulację dwóch automatów komórkowych: *Gra w życie* oraz *Mrówka Langtona*.

2. Analiza, projektowanie

2.1. Algorytmy, struktury danych, ograniczenia specyfikacji

Program opiera się głównie na tablicach wielowymiarowych, na których program przeprowadza symulację. Występuje również lista jednokierunkowa, która zachowuje kolejne kroki mrówki przy automacie *Mrówka Langtona*.

Program używa standardowych algorytmów. W przypadku *Gra w życie*, aplikacja korzysta z następujących reguł Conwaya:

- martwa komórka, która ma dokładnie 3 żywych sąsiadów, staje się żywa w następnej jednostce czasu (rodzi się)
- żywa komórka z 2 albo 3 żywymi sąsiadami pozostaje nadal żywa; przy innej liczbie sąsiadów umiera (z „samotności” albo „zatłoczenia”)

Natomiast *Mrówka Langtona*:

- jeśli znajduje się na polu białym, to obraca się w lewo (o kąt prosty), zmienia kolor pola na czarny i przechodzi na następną komórkę;
- jeśli znajduje się na polu czarnym, to obraca się w prawo (o kąt prosty), zmienia kolor pola na biały i przechodzi na następną komórkę;
- porusza się na nieskończonej planszy, podzielonej na kwadratowe komórki (pola) w dwóch możliwych kolorach: czarnym i białym.

3. Specyfikacja zewnętrzna

3.1. Obsługa programu

Program jest odpalany z poziomu konsoli oraz wymaga odpowiednich parametrów wejściowych do działania. Pierwszym parametrem jest typ symulacji:

-g	<i>Gra w życie</i>
-m	<i>Mrówka Langtona</i>

W każdym przypadku jest oczekiwana inna ilość argumentów oraz inne parametry, podane już w dowolnej kolejności.

Gra w życie	
Ilość argumentów	5
-i	Plik wejściowy (dowolny format)
-o	Plik wyjściowy (dowolny format)
-c	Liczba cykli do przetworzenia w symulacji
-w	Szerokość planszy podanej w pliku wejściowym
-t	Wysokość planszy podanej w pliku wejściowym

Mrówka Langtona	
Ilość argumentów	6
-o	Plik wyjściowy (dowolny format)
-c	Liczba cykli do przetworzenia w symulacji
-w	Szerokość planszy podanej w pliku wejściowym
-t	Wysokość planszy podanej w pliku wejściowym
-x	Koordinat początkowy x (punkt startowy)
-y	Koordinat początkowy y (punkt startowy)

W przypadku podania złej ilości lub błędnych parametrów, program wyświetli komunikat pomocy (patrz 3.3).

3.2. Format danych wejściowych

Dane wejściowe przekazywane są do programu tylko w przypadku wybrania symulacji *Gra w życie*. Wymaga ona pliku, który zawiera tylko 0 i 1. Przykładowa zawartość pliku zawierająca plansze 10X10:

```
0100000000
0010000000
1110000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
```

3.3. Komunikaty

W programie występuje parę komunikatów. Komunikat pomocy występuje przy podaniu gdziekolwiek parametru *-h*, podaniu błędnej ilości argumentów lub błędnych parametrów i wygląda w następujący sposób:

'Pomoc:

Wybierz tryb gry poprzez -g dla gry w życie lub -m dla mrowki(Zawsze jako pierwszy argument)

-h wyświetlenie tego komunikatu

Gra w życie wymaga 5 argumentów plus argument wybory gry na początku !!:

-i plik wejściowy

-o plik wyjściowy

-c liczba cykli do przetworzenia w grze

-w szerokosc planszy

-t wysokosc planszy

PRZYKLADOWE UZYCIE:

-g -i input -o output -c 100 -w 10 -t 10

Mrowka Langtona wymaga 6 argumentow plus argument wybory gry na poczatku !!:

-o plik wyjsciowy

-c liczba cykli do przetworzenia w grze

-w szerokosc planszy

-t wyskosc planszy

-x koordynat poczatkowy x

-y koordynat poczatkowy y

PRZYKLADOWE UZYCIE:

-m -o outputAnt -x 3 -y 3 -c 50 -w 6 -t 6'

Kolejny komunikat może się pojawić podczas ładowania planszy:

'Bład przy ładowaniu pliku, sprawdź czy taki plik istnieje oraz czy jego ilosc kolumn i wierszy pokrywa sie z parametrami przekazanymi do programu'

Jest to błąd, który jest najprawdopodobniej spowodowany brakiem uprawnień, błędnym formatem pliku lub złymi parametrami podanymi podczas włączania projektu.

Możliwe rozwiązania tego błędu:

- Sprawdzenie czy podany plik istnieje i ma takie rozszerzenie jakie zostało podane podczas uruchamiania programu (patrz 3.2.) oraz czy zawiera poprawne dane (patrz 3.2.), a szczególnie czy na końcu nie ma pustej linijki.
- Sprawdzenie czy została podana poprawna ilość wierszy i kolumn do programu (patrz 3.1. i 3.2.).

Następnym komunikatem, który może wystąpić przy próbie zapisu do pliku, jest:

'Nie można otworzyć pliku, sprawdź czy masz uprawnienia do zapisywania plików lub/i tworzenia plików'

Błąd ten występuje przy próbie zapisu stanu gry lub zapisywaniu logu do pliku.

Możliwe rozwiązania tego błędu:

- Sprawdzenie, czy użytkownik posiada odpowiednie uprawnienia do zapisu i tworzenia plików. W przypadku nie posiadania uprawnień zgłosić się do administratora systemu.
- Sprawdzenie, czy inny program nie zajmuje pliku wyjściowego lub pliku z logami.

4. Specyfikacja wewnętrzna

Specyfikacja zewnętrzna znajduje się w pliku doc, w projekcie. Została ona wygenerowana za pomocą programu *Doxygen*.

5. Testowanie

5.1. Dane testowe

Program został przetestowany pod wieloma aspektami. Zostały podane błędne argumenty: pliki wejściowe (input_width oraz input_height) z niepoprawnymi danymi, czyli nie pokrywającymi się z wielkościami podanymi do wiersza poleceń. Został również przetestowany przypadek, kiedy podane w pliku wejściowym (input_diff) dane były spoza zakresu (patrz 3.2.). Pliki testowe wraz z programem znajdują się na githubie.

5.2. Wyniki

Program wykazał każdą nieprawidłowość i pokazał stosowne komunikaty przy każdym z wymienionych w 5.1. prób testowych. Aplikacja została przetestowana pod względem wycieków pamięci przez zewnętrzny program valgrind i nie wykazuje on żadnych wycieków pamięci w programie.

6. Wnioski

Całe zadanie okazało się dosyć łatwe na podstawowym poziomie. Dodanie wątku może nie zwiększyło znacząco wydajności, ale na pewno pozwoliło mi nauczyć się nowych i uniwersalnych rzeczy. Dzięki temu, że zadanie nie było skompilowane, mogłem skupić się na walidacji danych oraz na poprawnym formatowaniu tekstu, użyciu zewnętrznych narzędzi takich jak *Doxygen* albo zewnętrznych bibliotek np.: pthread.