

Specyfikacja Funkcjonalna Projektu „Gra W Życie”

Andrzej Czechowski, Bartosz Zakrzewski

Data utworzenia: 29.02.2020

Data ostatniej modyfikacji: 10.03.2020

1 Cel Projektu

Program ma na celu tworzenie kolejnych generacji tablic/tabelek dwuwymiarowych według zasad „Gry w życie” Johna Conwaya. Otrzymane wyniki będziemy zapisywać do plików tekstowych oraz do plików PNG.

2 Opis Teoretyczny

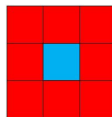
Uznajemy naszą tablicę dwuwymiarową za zbiór komórek. Każda z komórek przyjmuje stan 1 albo 0. Cyfra 1 oznacza, że komórka jest żywa, natomiast 0, że jest martwa. W zależności od ilości żywych sąsiadów, każda komórka w nowej generacji (w stanie wszystkich komórek po jednym „przejściu”/jednostce czasowej) przybiera nowy stan lub pozostaje w tym samym.

Zasady przeżywania/umierania komórki:

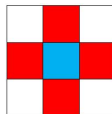
- Martwa komórka, która ma dokładnie 3 żywych sąsiadów, staje się żywa w następnej generacji.
- Żywa komórka, która ma 2 lub 3 żywych sąsiadów pozostaje nadal żywa; przy innej liczbie sąsiadów umiera.

W naszym programie pierwszym do implementacji sąsiedztwem jest sąsiedztwo Moore’a (komórka ma ośmiu sąsiadów).

Rodzaje sąsiedztwa (Kolor niebieski - rozpatrywana komórka, kolor czerwony - jej sąsiedzi):



caption 1: sąsiedztwo Moore’a



caption 2: sąsiedztwo von Neumanna

3 Opis funkcjonalności

3.1 Podstawowe zdolności programu

- wczytywanie tablicy z określonego z góry formatu pliku .txt,
- wykonanie podanej liczby generacji danej tablicy i zapisanie wyniku,
- wynik ma zostać zapisany w pliku .txt - w takim samym formacie jak plik wejściowy, aby mógł być także wczytany,
- wynik ma zostać zapisany zarówno w pliku TXT jak i w formacie PNG,
- program ma wykonywać różne zadania, w zależności od wartości argumentów wywołania użytkownika,
- program ma się kompilować i uruchamiać na serwerze „jimp.iem.pw.edu.pl”,
- program zostanie napisany w języku C, podzielony na pliki z poszczególnymi funkcjami,
- utworzony zostanie plik makefile, który zapewni kompilację plików,
- uznajemy obszar tablicy za „świat” zamknięty - oznacza to, że skrajne lewe komórki nie mają sąsiadów po lewej stronie, komórki w rogach mają tylko trzech sąsiadów itd.

3.2 Dane wejściowe

Aby program wczytał poprawnie dane, użytkownik musi stworzyć plik .txt w podanym formacie. Dane mają zostać podane w kolejności: Liczba wierszy, liczba kolumn tabeli, numer generacji, tabela składająca się z 0 i 1.

Przykładowe dane początkowe:

```
4 4
0
1 1 0 0
1 0 0 0
0 0 1 0
0 0 1 1
```

3.3 Dane wyjściowe

Format pliku wyjściowego txt ma być taki sam jak format pliku wejściowego, aby było możliwe użycie go ponownie do wczytania.

Gdy użytkownik wybierze opcję krok po kroku (-outgen n -SBS) zostaną utworzone pliki wejściowe ze wszystkimi generacjami o nazwach wynik1, wynik2, ... wynikn (w przypadku gdy zaczynamy od generacji 0, n - numer generacji) będą zapisane w dwóch folderach, w których zostaną zapisane:

- generacje w formacie txt zgodnie z wzorcem pliku wejściowego,
- te same generacje w formacie PNG jako czarno-biała plansza.

Przykładowy plik wyjściowy:

```
4 4
1
0 0 0 0
0 1 0 0
0 0 0 1
0 0 0 0
```

4 Rozwój programu

Program można rozwinąć o:

- zmianę sąsiedztwa - użytkownik może wybrać rodzaj sąsiedztwa - Moore’a lub von Neumanna,
- zapisywanie wszystkich generacji w jednym pliku,
- opcję zmiany kryteriów umierania, ożywiania komórek,
- stworzenie osobnej funkcji, która będzie tworzyła losową planszę,
- możliwość zmiany „rodzaju świata” (świat otwarty tablicy to taki, że np. lewy sąsiad skrajnie lewej komórki to skrajnie prawa komórka itp.

5 Ograniczenia

- Funkcja przeszukująca sąsiadów może nie być najbardziej optymalna, dlatego dla zbyt dużej liczby generacji (będziemy próbować ją wyszacować, przykładowo 100) program może zająć zbyt dużo czasu, albo w ogóle się nie zakończyć.
- Zbyt duży rozmiar planszy także może mieć takie konsekwencje jak opisane powyżej (przykładowo 100 x 100).
- Użytkownikowi zostaje narzucony format danych wejściowych oraz nazwy plików wyjściowych.

6 Argumenty wywołania programu

- -input filename - plik wejściowy,
- -output dirname - folder do którego zapisane zostaną dane (opcjonalnie),
- -h - wyświetlenie help,
- -outgen n - określenia liczbę generacji.

Dodatkowe opcje outgen:

- domyślnie - do pliku TXT i PNG zostanie zapisana ostatnia generacja,
- -SBS - zapisane zostaną wszystkie generacje od s (początkowa generacja) do n (włącznie), do plików: wyniki, ...wynikn,
- x,y,z - wybrane generacje x,y,z ($\leq n$) zostaną zapisane do poszczególnych plików.

7 Komunikaty błędów / Sytuacje wyjątkowe

Błędy związane z argumentami wywołania:

- brak argumentów wywołania - uruchomi się help:
kod błędu: -1,
- nie wpisanie wymaganych flag:
kod błędu: -2,
- podanie nie istniejącego pliku wejściowego:
kod błędu: -3,
- podanie źle sformatowanego pliku wejściowego:
kod błędu: -4,
- wpisanie za dużej liczby generacji:
kod błędu: -5;
- zbyt duży rozmiar tablicy w pliku wejściowym:
kod błędu: -6;
- przy wpisaniu argumentu wywołania -gen -n x,z,y, któryś z x,z,y będzie większy niż n (ilość generacji):
kod błędu -7;
- nie utworzenie katalogu lub pliku wyjściowego:
kod błędu -8;
- przepełnienie tablicy, błąd mallocowania:
kod błędu -9.

8 Przykładowe uruchomienie programu

Będąc w folderze z plikiem makefile wpisujemy w terminalu „make” i to skompiluje wszystkie potrzebne do działania programu pliki z rozszerzeniem .c. Następnie wpisujemy: ./out i podajemy argumenty wywołania.

Przykładowe uruchomienie programu:

make

./out -input dane2 -outgen 1

Plik wejściowy:

```
4 4
0
1 1 0 0
1 0 0 0
0 0 1 0
0 0 1 1
```

Plik wyjściowy (wynik1.txt):

```
4 4
1
1 1 0 0
1 0 0 0
0 1 1 1
0 0 1 1
```

9 Testowanie

- Program będzie testowany i uruchamiany przeważnie w środowiskach Linuxowych.
- Początkowo, poprawność działania programu będzie testowana na prostych danych, gdzie sami potrafimy ocenić, jak zmieni się dana generacja.
- Do znajdowania błędów w kodzie posłuży nam program gdb.
- Dzięki programowi valgrind będziemy mogli sprawdzić czy udało nam się prawidłowo zwolnić pamięć.

10 Źródła

- Opis projektu „Automat Komórkowy” B. Chaber - ISOD
- Tobiasz Siemiński - opis specyfikacji funkcjonalnej: <https://sortris.blogspot.com/2010/05/jak-pisac-specyfikacje-funkcjonalna.html>
- Interpunkcja w wyliczeniach <https://www.ekorekta24.pl/interpunkcja-w-wyliczeniach-przecinki-sredniki-a-moze-nic/>
- Ten dokument został utworzony w LaTeX’ie za pomocą strony <https://www.overleaf.com>
- Jest to pierwszy z kolei dokument dotyczący projektu „Gra w Życie”