Gra w życie

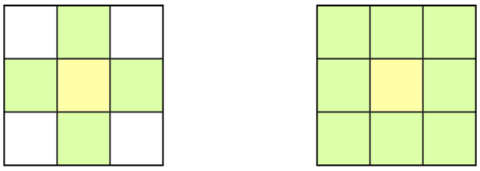
Trzaskowski Mikołaj 279075
Adam Wołonkiewicz 279078

1 Wstęp teoretyczny

Automat komórkowy jest to obiekt składający się z sieci komórek, które posiadają pewien stan z określonego zbioru oraz algorytmu zbioru reguł, który określa stan danej komórki, najczęściej w zależności od stanów komórek sąsiednich. Brzmi to skomplikowanie, ale okaże się za chwilę, że w praktyce w prosty sposób osiągnąć będzie można zadziwiające efekty.
Każda z komórek może posiadać dwa stany: żywy i martwy. Autor tego automatu, angielski matematyk John Conway, założył następujące reguły:

1. żywa komórka umiera z samotności, gdy posiada jednego lub żadnego sąsiada (inną żywą komórkę)
2. żywa komórka posiadająca dwóch lub trzech sąsiadów podtrzymuje życie
3. żywa komórka umiera, gdy posiada więcej niż trzech sąsiadów
4. martwa komórka posiadająca 3 sąsiadów ożywa

Ważnym elementem programu jest również ustalenie sąsiedztwa. Może być definiowane na różne sposoby, najczęściej jest to sąsiedztwo Moore'a (8 komórek dookoła) lub sąsiedztwo von Neumanna (4 komórki przylegające bokiem).



Rysunek 1: sąsiedztwo von Neumanna i sąsiedztwo Moore'a

2 Warunki brzegowe

W naszej wersji programu wykorzystamy periodyczne (przenikające) warunki brzegowe. Definiują one zamkniętą siatkę w taki sposób, że np. symulując poruszającą się cząstkę po dojściu do krawędzi pojawi się ona z drugiej strony. Komórka znajdująca się na brzegu siatki ma za sąsiada komórkę leżącą po drugiej stronie siatki. Ten typ przejść bardzo dobrze opisuje nasze otoczenie – środowisko, w którym wszystko również odbywa się na periodycznej przestrzeni, jaką jest torus.

3 Uruchomienie programu

Aby uruchomić program, należy najpierw skompilować plik poleceniem 'make'. Powoduje to utworzenie wykonywalnego pliku 'life'. Kolejnym krokiem jest uruchomienie pliku 'life' i podanie trzech argumentów w następujący sposób:

```
./life [plik_z_siatką] [reguła_sąsiedztwa] [liczba_generacji]
```

1. Pierwszy argument, czyli plik zawierający siatkę, musi być plikiem tekstowym, gdzie w pierwszych dwóch liniach posiadać dwie liczby całkowite (int), gdzie pierwsza z nich oznacza liczbę wierszy, a druga liczbę kolumn. Następnie plik musi zawierać siatkę, w której wartości są tylko jedynkami (1) lub zerami (0), np:

   ```
   5
   5
   0 0 0 0 0
   0 1 0 0 0
   0 0 0 1 0
   0 1 1 1 0
   0 0 0 0 0
   ```

2. Drugi argument określa, która reguła sąsiedztwa ma zostać użyta. Aby program wykorzystał sąsiedztwo:
   Moora - podaj '-m'
   Neumanna - podaj '-n'

3. Trzeci argument określa liczbę generacji, którą program ma wykonać. Musi być on liczbą całkowitą i większą od zera.

4 Funkcjonalność

Stworzony program umożliwia wyświetlenie na standardowe wyjście sekwencji przejść automatu komórkowego Johna Conwaya dla określonych danych

wejściowych.
Tworzy on również pliki wynikowe (tekstowe i graficzne) generowane po każdym przejściu. Pliki tekstowe przechowują rozmiary siatki i ich aktualną generację (mogą być one ponownie wczytane jako plik wejściowy), natomiast pliki graficzne w formacie png są wizualizacją tej generacji.

5 Format danych i struktura plików

Program podzielony został na następujące moduły:
- output.c i output.h - odpowiadają za wyświetlanie sekwencji generacji oraz wypisanie ich do plików;
- logic.c i logic.h - odpowiadają za wyznaczanie kolejnych generacji automatu;
- png_gen.c png_gen.h - odpowiada za generację obrazów png;
- main.c - główna funkcja programu;
- Makefile - odpowiada za automatyczną kompilację plików i usunięcie poprzednich wyników.
Powyższe pliki znajdują się w głównym katalogu programu.

Plik 'Makefile' umożliwia również usunięcie plików wytworzonych podczas kompilacji (pliki z rozszerzeniem '.o') oraz katalogów z plikami wynikowymi. W tym celu należy wprowadzić polecenie:

```
'make clean'
```

Jeżeli program zostanie właściwie uruchomiony, zostają utworzone dwa katalogi do przechowywania danych wyjściowych z pliku:
- generations_txt - do przechowywania plików tekstowych;
- generations_png - do przechowywania plików graficznych.

W programie struktura zawierająca komórki będzie dwuwymiarową tablicą. Pamięć jej musi być alokowana dynamicznie, aby program mógł działać dla siatek o różnych rozmiarach.
Wartości komórek określone będą za pomocą zer i jedynek, gdzie 0 - komórka martwa, 1 - komórka żywa.

6 Scenariusz działania programu

Po uruchomieniu, program wykonuje następujące czynności:

1. sprawdza, czy została podana prawidłowa liczba argumentów, jeżeli nie, na standardowe wyjście wypisywany jest komunikat jak uruchomić program;

2. program sprawdza czy plik wejściowy istnieje, jeżeli tak, to wczytuje dane dotyczące rozmiaru siatki;

3. program alokuje pamięć na siatkę i wczytuje do niej wartości komórek z pliku;

4. dla każdej komórki program wylicza sumę żywych sąsiadów, na podstawie której określany jest następny stan komórki, po wyznaczeniu stanów dla wszystkich komórek wyznaczana jest następna generacja, kolejnie wypisywanie na standardowe wyście kolejnych przejść, utworzenie plików wyjściowych i zapisanie ich do poszczególnych katalogów

5. powyższy krok powtarzany jest liczbę razy, określoną w argumencie wywołania

6. ostatnim krokiem jest zwolnienie pamięci przeznaczonej na dwuwymiarową tablicę, a następnie zakończenie programu.