Specyfikacja funkcjonalna
automat komórkowy
"The game of life"

Jan Starczewski
Bartosz Michałowski

12 marca 2018

Spis treści

1 Opis ogólny

1.1 Nazwa programu

Nazwa programu: `life_game_generator`

1.2 Poruszany problem

Automatem komórkowym określa się system składający się z pojedynczych komórek, znajdujących się obok siebie. Każda z komórek może przyjąć jeden ze stanów, przy czym liczba stanów jest skończona, ale dowolnie duża. Stan komórek zmieniany jest synchronicznie w zależności od komórek otaczających oraz przyjętych reguł. Poruszanym problemem jest implementacja takowego automatu komórkowego **The game of life** autorstwa Johna Conwaya i stworzenie zestawu plików graficznych obrazujących kolejne generacje komórek począwszy od zapewnionej przez użytkownika generacji początkowej. Program jest napisany w **języku C**. Użytkownik powinien mieć możliwość wyboru typu sąsiedztwa spośród sąsiedztwa Moore'a i von Neumanna. Przyjęto zasadę że figury po napotkaniu krawędzi przechodzą przez nią i pojawiają się na przeciwległej stronie planszy.

2 Opis funkcjonalności

2.1 Jak korzystać z programu

Skompilowany program przeznaczony jest do uruchomienia z interfejsu tekstowego wraz z niezbędnymi argumentami tj. ścieżką do generacji początkowej w formie tekstowej(plik tekstowy), liczbą generacji, rodzajem sąsiedztwa i nazwą katalogu wyjściowego. Program tworzy folder o podanej nazwie wyjścia w katalogu `./output`. W przypadku niepowodzenia program zakończy pracę, a odpowiednie komunikaty zostaną zawarte w pliku log w katalogu głównym. Jeżeli utworzenie miejsca docelowego zakończy się sukcesem, zapisane do niego zostaną kolejne stany jako pliki o nazwie `gen_[NUM].png`, gdzie NUM to liczba porządkowa kolejnych generacji. W razie niepowodzenia jednego z zachodzących procesów, wszelkie komunikaty znajdą się w pliku `log` w katalogu wyjściowym. Katalog ten zawiera również pierwszą i ostatnią generację w postaci plików tekstowych o nazwach odpowiednio, `first_gen` i `last_gen`.

2.2 Uruchomienie programu

Wywołanie :`life_game_gen` [**ścieżka_generacji_początkowej**] [**liczba_generacji**] [**rodzaj_sąsiedztwa**] [**nazwa_katalogu_wyjściowego**]

[**ścieżka_generacji_początkowej**] : ścieżka do pliku zawierającego generacje początkową w formie tekstowej

[**liczba_generacji**] : oczekiwana liczba generacji

[**rodzaj_sąsiedztwa**] : wybór rodzaju sąsiedztwa:
`-M` sąsiedztwo Moore'a, `-N` sąsiedztwo von Neummana

[**nazwa_katalogu_wyjściowego**] : nazwa katalogu wyjściowego

Wywołanie przykładowe : `life_game_gen fgen.txt 10 -M 10gen`

2.3 Możliwości programu

Program wczytuje dane wejściowe, zawierające rozmiar planszy i konfigurację początkową. Bazując na pierwotnym ułożeniu komórek, tworzone są ich kolejne generacje. Odpowiednie stany są zapisywane w postaci plików, o rozszerzeniu PNG, do wytyczonego folderu. Użytkownik ma możliwość wyboru rodzaju sąsiedztwa (zasad), na podstawie których powstaną kolejne generacje.

3 Format danych i opis funkcjonalności

3.1 Słownik

Sąsiedztwo to otoczenie rozpatrywanej komórki. Sąsiedztwo Moore'a określa zbiór komórek stykających się bokami i rogami (osiem), z komórką rozpatrywaną, a von Neumanna, tylko bokami (cztery).

3.2 Struktura katalogów

Katalog główny zawierający plik wywołania. Podkatalog `./output` zawierający odpowiednio, kolejne katalogi wypełnione utworzonymi obrazami w formacie PNG, jak i pierwszą, ostatnią generację w formacie tekstowym.

3.3 Przechowywanie danych w programie

Stan komórek przechowywany jest w postaci jednowymiarowego wektora, deklarowanego po wczytaniu pliku zawierającego dane do pierwszej generacji. Istnieją dwie takie struktury danych, gdzie jedna zawiera w danej chwili stan początkowy, a druga stan kolejny.

3.4 Dane wejściowe

Program na wejściu otrzymuje plik tekstowy, w którym pierwsze dwie liczby oznaczają kolejno: szerokość i wysokość planszy. Kolejne liczby to komórki w stanie początkowym, w kolejności od lewej do prawej, wierszami od góry do dołu. Martwa komórka oznaczona jest przez cyfrę zero, żywa przez jeden.

3.5 Dane wyjściowe

Program generuje pliki o rozszerzeniu PNG, nazwach `gen_[NUM].png`, gdzie NUM to kolejne liczby całkowite, odpowiadające kolejnym generacjom. Pole białe oznacza komórkę martwą, czarne żywą. Pierwsza, jak i ostatnia generacja zostanie zapisana w plikach tekstowych, odpowiednio `first_gen` i `last_gen`. Utworzony zostanie także plik tekstowy `log.txt`, w którym znajdą się komunikaty od programu.

3.6 Komunikaty zwrotne

Przykładowe komunikaty zwrotne zawarte w plikach `log.txt`:

1. Błąd odczytu/zapisu:
   (a) Nie udało się wczytać pliku wejściowego - spowodowane nieistnieniem danego pliku lub brakiem uprawnień
   (b) Nie udało się utworzyć plików wynikowych - spowodowane brakiem uprawnień
   (c) Błędny format pliku wejściowego - plik wejściowy zawiera niedozwolone znaki
2. Błąd pamięci:
   (a) Niewystarczająca ilość pamięci - pamięć dostępna jest zbyt mała, aby utworzyć wektor o zadanym rozmiarze

4 Testowanie

4.1 Ogólny przebieg testowania

Każdy moduł programu zostanie przetestowany dla prostych, zarówno błędnych, jak i poprawnych danych wejściowych. Proste przykłady ułatwią weryfikację poprawności wyniku oraz identyfikację ewentualnych błędów. Obsługa błędów zostanie przetestowana korzystając z danych błędnych, zapewnionych dzięki analizie krytycznych fragmentów kodu. Główna uwaga zostanie skupiona na sprawdzeniu poprawności odczytu/zapisu danych oraz algorytmu tworzącego generacje.