

Specyfikacja funkcjonalna – Gra w życie

Krzysztof Dąbrowski i Jakub Bogusz

27 lutego 2019

Spis treści

1	Cel projektu	2
2	Opis ogólny problemu	3
3	Działanie programu	4
3.1	Komunikacja z użytkownikiem	4
3.2	Tryb wsadowy	4
3.3	Plik wejściowy wejściowego	5
3.3.1	Przykład	5
3.3.2	Format pliku	5
4	Wyniki działania programu	7

Rozdział 1

Cel projektu

Celem projektu jest implementacja gry w życie w języku C. Gotowy program ma przeprowadzać symulację kolejnych pokoleń oraz generować na ich podstawie pliki graficzne przedstawiające etapy symulacji.

Rozdział 2

Opis ogólny problemu

Gra w życie jest automatem komórkowym wymyślonym przez brytyjskiego matematyka John Horton Conway w 1970 roku. Polega na symulacji kolejnych pokoleń życia komórek według następujących zasad.

Stany Komórka może znajdować się w jednym z dwóch stanów:

- żywa
- martwa

Reguły Następne pokolenie generowane jest zgodnie z regułami:

- Jeżeli komórka była martwa i miała dokładnie 3 żywych sąsiadów, w następnym pokoleniu staje się żywa,
- Jeżeli komórka była żywa to pozostaje żywa jeśli miała dwóch lub trzech żywych sąsiadów. W przeciwnym razie staje się martwa.

Rozdział 3

Działanie programu

3.1 Komunikacja z użytkownikiem

Program działa w trybie wsadowym. Oznacza to, że użytkownik podaje jedynie argumenty początkowe dla symulacji wraz z uruchomieniem programu (w formie flag), a następnie program automatycznie przetwarza dane i generuje wyniki.

3.2 Tryb wsadowy

Argumenty

- `-h / --help`
Wyświetlenie pomocy
- `-f [nazwa pliku] / --file plik=[nazwa pliku]`
Plik z wejściowym stanem planszy zgodny z [formatem](#).
- `-o [ścieżka] / --output_dest=[ścieżka]`
Ścieżka do folderu, w którym zostaną zapisane wyniki symulacji. Domyślnie pliki graficzne nie będą generowane.
- `-t (gif | png) / --type (gif | png)`
Typ generowanych rezultatów. Domyślnie gif.
- `-n [liczba] / --number_of_generations=[liczba]`
Ilość pokoleń do wygenerowania. Domyślnie 15
- `-p [liczba] / --step=[liczba]`
Wybór co który stan symulacji będzie zapisywany. Domyślnie 1
- `-s [liczba] / --size=[liczba]`
Losowe generowanie planszy początkowej o podanym rozmiarze. Wyklucza się z `-f`

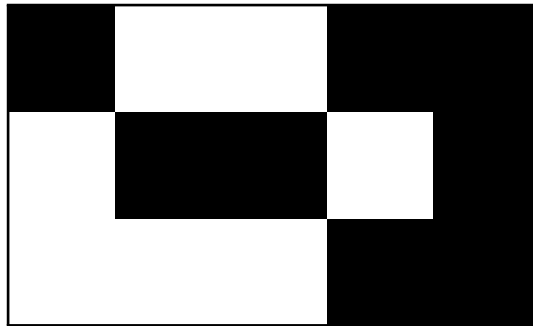
- `-d [liczba] / --delay=[liczba]`
Podanie tego argumentu spowoduje wyświetlanie w konsoli kolejnych generacji symulacji. Wartość argumentu [liczba] oznacza czas w milisekundach między wyświetleniem poszczególnych pokoleń. Domyślnie 1000.

3.3 Plik wejściowy wejściowego

Plik wejściowy pozwala na wczytanie planszy bez konieczności wprowadzania przez użytkownika informacji o komórkach jedna po drugiej.

3.3.1 Przykład

5 3	– rozmiar (x y)
1 0 0 1 1	– Wartości poszczególnych komórek
0 1 1 0 1	– 1 - żywa
0 0 0 1 1	– 0 - martwa



Rysunek 3.1: Grafika wygenerowana na podstawie przykładowej planszy

3.3.2 Format pliku

Kodowanie

Ponieważ plik powinien zawierać tylko liczby arabskie i odstępy możliwe jest dowolne kodowanie kompatybilne z ASCII.

Sugerowane kodowania to: ASCII, UTF-8, ISO 8859, Windows-1250

Opis formatu

Plik w pierwszej linii powinien zawierać 2 liczby. Pierwsza z nich oznacza rozmiar planszy w poziomie, druga w pionie.

Następnie plik powinien zawierać tyle linii jaki został podany rozmiar w pionie.
W każdej z tych linii powinno być tyle 0 lub 1 ile wynosi rozmiar w poziomie.
Zero oznacz komórkę martwą, a jeden komórkę żywą.

Rozdział 4

Wyniki działania programu

Wyniki działania programu będą zależeć od preferencji użytkownika. W trybie [interaktywnym](#) program zapyta o docelowy format wyniku, a w trybie z argumentami wiersza poleceń, wynik zależeć będzie od wartości [argumentu -t](#) oraz [argumentu -o](#). W obu przypadkach użytkownik będzie mógł:

- wyświetlić wybraną ilość generacji w konsoli,
- wygenerować wybraną ilość plików .png z reprezentacjami graficznymi kolejnych pokoleń
- wygenerować plik .gif przedstawiający życie cywilizacji