

Specyfikacja funkcjonalna programu lifeGameEmulator

Ciszewski Jakub, Rahachevich Aliaksandra

28 Luty 2021

1 Cel projektu

Program Life ma na celu symulowanie działania gry w życie Johna Conwaya. Program działa w trybie nieinteraktywnym/wsadowym. Program pozwala na symulację dowolnej ilości iteracji zmian planszy. Program generuje pliki wyjściowe w formacie tekstowym lub graficznym.

2 Dane wejściowe

Program w celu poprawnego działania przyjmuje:

- Plik zawierający opis wyglądu początkowego planszy. Opis składa się z zapisanej po sobie ilości kolumn oraz wierszy na planszy oraz symboli "0" i "1" oznaczających kolejno komórki martwe i komórki żywe. Zarówno ilość kolumn, jak i wierszy powinna być większa od 2.

Przykład pliku wejściowego:

```
5 4
0 0 1 1 0
0 0 0 0 1
0 0 1 1 0
0 0 1 0 1
```

- Opcjonalne dodatkowe argumenty wywołania wpływające na działanie programu.

3 Argumenty wywołania programu

Program Life akceptuje następujące argumenty wywołania:

- `-dataFile filePath` ścieżka do pliku zawierającego opis planszy tak jak powyżej. W przypadku braku podania tego argumentu wczytana zostaje domyślna struktura.

- *-flatArea false | true* powoduje zmianę trybu interpretacji planszy przez algorytm. Kiedy argument przyjmuje wartość false plansza jest zaokrąglana- komórki znajdujące się na brzegach są ze sobą zaokrąglane. Argument true powoduje, że plansza jest płaska- komórki znajdujące się na brzegach nie są ze sobą porównywane. Domyślna wartość to false.
- *-help* powoduje wyświetlenie informacji o sposobie działania programu.
- *-iterations n* Ilość iteracji gry jaką wykona program w trakcie swojego działania.
- *-outputFilename name* ustawia nazwę pliku do którego zostaną zapisane dane.
- *-proximity moore | neumann* wybiera typ sąsiedztwa. Domyślnym algorytmem jest sąsiedztwo moore-a.
- *-saveAsPicture false | true* decyduje o tym czy iteracja/iteracje mają być zapisywane w formacie graficznym. Domyślna wartość to false.
- *-saveAsTxt false | true* decyduje o tym czy iteracja/iteracje mają być zapisywane w formacie tekstowym. Domyślna wartość to true.
- *-saveEveryIteration 0 | 1 | 2* decyduje o tym czy iteracja ma zostać zapisana do pliku. 0 - niezapisuje żadnej iteracji, 1 - zapisuje tylko ostatnią iterację, 2 - zapisuje każdą iterację. Domyślna wartość to 0.
- *-SBS false|true* wybiera tryb działania programu. Czy program ma zatrzymywać się po każdym przejściu, czy wykonywać wszystkie iteracje. Podczas działania programu w trybie SBS można go opuścić naciskając przycisk *ń*". Domyślna wartość false.

Przykładowe wywołania programu:

- `lifeGameEmulator -data lifeGameEmulatorData.txt -flatArea false -iterations 10 -outputFilename Wyniki -proximity moore -saveAsPicture false -saveAsTxt true -saveEveryIteration 2 -SBS true`

Efektem będzie analiza pliku *lifeGameEmulator.txt*. Program symuluje 10 iteracji z typem sąsiedztwa *moore-a*. Program po kolei wykonuje *wszystkie* iteracje *przerywając i oczekując na akcje użytkownika*. Algorytm korzysta z sąsiedztwa *moore-a* oraz *zaokrąglenia planszy*. Wynik każdej iteracji jest zapisywany w formacie *tekstowym* do pliku *Wyniki.txt*.

4 Teoria

Program generuje płaszczyznę o określonych wymiarach, która składa się z komórek. Komórka jest martwa (0) lub żywa (1).

W następnych iteracjach program generuje płaszczyznę z następnymi iteracjami gry według zasad:

- Martwa komórka, która ma dokładnie 3 żywych sąsiadów, staje się żywa
- Martwa komórka która miała liczbę sąsiadów różną od 3 pozostaje martwa.
- Żywa komórka, która ma 2 albo 3 żywych sąsiadów, pozostaje nadal żywa
- Żywa komórka, która posiada inną liczbę sąsiadów od 2 lub 3 staje się martwa.

Program może wykorzystywać różne rodzaje sąsiedztwa:

- – *sąsiedztwo Moore’a*: 8 przylegających komórek (znajdujących się: na południu, na południowym-zachodzie, na zachodzie, na północnym-zachodzie, na północy, na północnym-wschodzie, na wschodzie i na południowym-wschodzie).
- – *sąsiedztwo von Neumanna*: 4 przylegające komórki (na południu, zachodzie, północy i wschodzie).

5 Komunikaty błędów

Podczas działania programu użytkownik może napotkać następujące błędy:

- Nieprawidłowe dane wejściowe: ***Dane wejściowe zawarte w podanym pliku mają nieprawidłowy format. Zalecamy poprawienie danych wejściowych, lub podanie innego pliku.***
Komunikat wyświetlany jest w sytuacji, kiedy dane zawarte w pliku wejściowym są nieprawidłowe.
- Nieprawidłowa nazwa pliku wejściowego: ***Plik o podanej nazwie nie istnieje. Należy podać nazwę istniejącego pliku wejściowego***
Komunikat wyświetlany jest wtedy, gdy nie istnieje plik o podanej nazwie.
- Nieprawidłowy argument wywołania: ***Następujące argumenty wywołania są niepoprawne: {Lista niepoprawnych argumentów}. Sprawdź dostępne argumenty wywołania za pomocą -help.*** Komunikat wyświetlany w przypadku, gdy podana nazwa argumentu wywołania nie istnieje.
- Nieprawidłowa wartość argumentu wywołania: ***Argument {nazwa argumentu wywołania} ma nieprawidłową wartość. Prawidłowa wartość to {Prawidłowa wartość argumentu wywołania}.*** Komunikat wyświetlany w przypadku użycia nieprawidłowej wartości dla argumentu wywołania.