Politechnika Warszawska
Wydział Elektryczny

SPECYFIKACJA FUNKCJONALNA
PROJEKT: WIREWORLD

Autorzy:

Anna Głowińska
numer indeksu: 291070
anna.glowinska98@gmail.com

Adam Czajka
numer indeksu: 291063
czajka.adam147@gmail.com

Praca wykonana w ramach przedmiotu: Języki i metody programowania 2

26 kwietnia 2018

Spis treści

1 Opis ogólny **2**
1.1 Nazwa programu 2
1.2 Podstawowe pojęcia 2
1.3 Poruszany problem 2

2 Wygląd programu **3**
2.1 Wygląd GUI 3
2.2 Opis wyglądu GUI 4

3 Opis funkcjonalności **4**
3.1 Korzystanie z programu 4
3.2 Uruchomienie programu 4
3.3 Możliwości programu 4

4 Format danych i struktury plików **5**
4.1 Struktura katalogów 5
4.2 Przechowywanie danych 5
4.3 Dane wejściowe 6
4.4 Dane wyjściowe 6

5 Scenariusz działania programu **6**
5.1 Scenariusz ogólny 6
5.2 Scenariusz szczegółowy 7

6 Testowanie **7**
6.1 Ogólny przebieg testowania 7

1 Opis ogólny

1.1 Nazwa programu

Wireworld - automat komórkowy Wireworld Briana Silvermana zbudowany w języku `Java`.

1.2 Podstawowe pojęcia

Komórka - najmniejsza jednostka automatu komórkowego, która może znajdować się w jednym z czterech stanów (przypisujemy im odpowiednie kolory): pusta - czarna, głowa elektronu - niebieska, ogon elektronu - czerwony, przewodnik - żółty.

Automat komórkowy - uporządkowany zbiór komórek, z których każda znajduje się w jednym z kilku dozwolonych stanów. Komórki przylegają do siebie tworząc siatki.

Siatka - dwuwymiarowy fragment płaszczyzny o danych wymiarach, podzielony liniami tworzącymi macierz, składającą się z pojedynczych komórek.

Generacja - stan wszystkich komórek w danej chwili, ale też czynność zmiany ich stanów.

Sąsiedztwo Moore'a - graniczenie komórki z innymi komórkami, warunkujące nowy stan tej komórki, na podstawie 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).

1.3 Poruszany problem

Zadaniem automatu komórkowego *Wireworld* jest symulacja elementów elektronicznych operujących na wartościach bitowych. Symulacja toczy się na dwuwymiarowej płaszczyźnie podzielonej na kwadratowe komórki. Zgodnie z przyjętymi zasadami każda komórka ma ośmiu sąsiadów (sąsiedztwo Moore'a), czyli komórki przylegające do niej bokami i rogami. Każda komórka może znajdować się w jednym z czterech stanów: pusta, głowa elektronu, ogon elektronu albo przewodnik).

Stany komórek zmieniają się przy każdej generacji. Dana generacja jest używana do obliczenia następnej generacji. Po obliczeniu wszystkie komórki

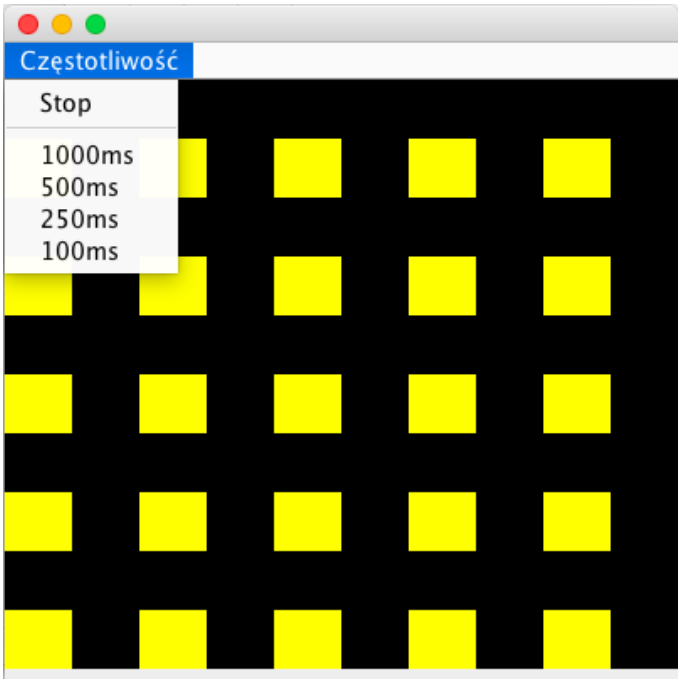
zmieniają swój stan dokładnie w tym samym momencie. Stan komórki po generacji zależy tylko od otaczających ją sąsiadów i jej obecnego stanu oraz zmienia się zgodnie z ustalonymi zasadami.

Zestaw reguł przy tworzeniu nowej generacji (przejścia do następnego stanu) jest następujący:

- komórka pusta → komórka pusta,
- głowa elektronu → ogon elektronu,
- przeprowadzenie zadanej liczby generacji,
- ogon elektronu → przewodnik,
- przewodnik → głowa elektronu, ale tylko wtedy, gdy dokładnie 1 lub 2 komórki sąsiadujące są głowami elektronu.

2 Wygląd programu

2.1 Wygląd GUI



Rysunek 1: Wygląd GUI.

2.2 Opis wyglądu GUI

Graficzny interfejs użytkownika programu *Wireworld* jest reprezentowany przez okno dialogowe. Przebieg generacji odbywa się w centralnej części okna, gdzie komórki przyjmują odpowiednie kolory w zależności od ich stanu. Powyżej znajduje się pasek menu, w którym użytkownik programu dobiera częstotliwość wyświetlania kolejnych generacji oraz w wybranym momencie zatrzymuje działanie programu i zapisuje bieżącą generację do pliku TXT.

3 Opis funkcjonalności

3.1 Korzystanie z programu

Program *Wireworld* jest pisany, kompilowany oraz uruchamiany za pomocą programu `IntelliJ IDEA 2017.3.5 (Community Edition)`.

3.2 Uruchomienie programu

Program `IntelliJ` wywoływany jest automatycznie po naciśnęciu przycisku `run`.

3.3 Możliwości programu

Zadania jakie wykonuje program:

- wczytywanie do programu początkowej konfiguracji generacji z pliku w formacie TXT,
- wczytanie danych wejściowych z wbudowanej w program biblioteki,
- przeprowadzenie kolejnych generacji,
- wyświetlanie kolejnych generacji w czasowych odstępach o długości wybranej przez użytkownika użytkownika,
- zatrzymanie wyświetlania kolejnych generacji,
- zapisywanie bieżącej generacji do pliku TXT, który może zostać potem wczytany.

4 Format danych i struktury plików

4.1 Struktura katalogów

Szkielet projektu został przedstawiony na poniższym schemacie:

```
wireworld
+— data
|    +— data.txt
|    +— library
|    +— results
+— src
|    +— ww
|        +— tests
+— out
|    +— production
|        +— wireworld
|            +— ww
```

gdzie:
`wireworld` - katalog projektu,
`data` - katalog zawierający dane testowe (dane.txt) oraz katalogi `library` i `results`,
`library` - katalog zawierający przykłady danych wejściowych w plikach TXT,
`results` - katalog w którym zostają zapisane bieżące generacje w formacie TXT,
`src` - katalog zawierający pliki z kodem programu,
`tests` - katalog zawierający pliki z kodem poszczególnych testów,
`out` - katalog zawierający pliki wykonywalne.

4.2 Przechowywanie danych

Na dysku: Pliki z danymi tekstowymi przedstawiającymi stany wybranych generacji są zapisywane w katalogu `wireworld/data/results`.

W programie: Zostanie utworzona klasa `Matrix` przechowująca dwuwymiarową tablicę obiektów klasy `Cell` (poszczególnych komórek przechowujących swoje stany).

4.3 Dane wejściowe

Program będzie operował na pliku wejściowym o formacie TXT, gdzie dane przedstawione są w postaci:

```
x y
3 0 1 1 ...
0 2 3 1 ...
2 1 3 0 ...
... ... ...
```

gdzie:

- `x`, `y` to odpowiednio podane szerokość oraz długość dwuwymiarowej płaszczyzny na której odbędzie się symulacja *Wireworld*,
- `0` - odpowiada komórce pustej, `1` - głowie elektronu, `2` - ogonowi elektronu, `3` - przewodnikowi.

4.4 Dane wyjściowe

Program dzięki swojej pracy generuje pliki TXT zawierające informacje o danej generacji.

Te pliki zostaną zapisane w katalogu: `wireworld/data/results`.

5 Scenariusz działania programu

5.1 Scenariusz ogólny

Przebieg działania programu:

1. Kompilacja i uruchomienie.
2. Przebieg kolejnych generacji.
3. Zmiana częstotliwości wyświetlania generacji po ustawieniu tego przez użytkownika.
4. Zakończenie działania programu.

5.2 Scenariusz szczegółowy

Szczegółowy przebieg działania programu:

1. Uruchomienie programu przebiega automatycznie w środowisku `IntelliJ`.
2. Program oferuje możliwość wyboru między częstotliwości, zatrzymania programu oraz przechodzenia manualnie do następnej generacji.
3. Po zatrzymaniu wyświetlania kolejnych generacji program umożliwia zapisanie generacji do pliku TXT.
4. Program kończy działanie po zamknięciu okienka w którym wyświetlane są generacje.

6 Testowanie

6.1 Ogólny przebieg testowania

Testowanie programu *Wireworld* odbywa się z pomocą frameworków `Mockito` oraz biblioteki `AssertJ`. Pozostałe elementy programu - menu w GUI oraz wygląd okienka zostaną przetestowane metodą dynamiczną.

Pliki zawierające kod poszczególnych testów będą się znajdować w katalogu `wireworld/src/ww/tests`.