

Specyfikacja funkcjonalna

Automat komórkowy - WireWorld

Michał Sut

1 Opis ogólny

1.1 Nazwa programu

Automat komórkowy *WireWorld*

1.2 Wstęp teoretyczny

WireWorld to rodzaj automatu komórkowego stworzony przez Briana Silvermana. Ogólna zasada działania automatów komórkowych jest następująca: Dana jest siatka - dla uproszczenia dwuwymiarowa w postaci "kratownicy". Siatka ta składa się z przylegających do siebie komórek, które mogą znajdować się w jednym z kilku możliwych stanów.

Komórki zmieniają swój status na podstawie swojego obecnego stanu oraz stanu komórek sąsiadujących, według pewnych określonych reguł. Stan wszystkich komórek w danej chwili nazywamy generacją. Niniejszy program jest takim właśnie automatem, opartym o zbiór zasad stworzonych przez Briana Silvermana, zwanym *WireWorld*.

1.3 Zasady działania automatu *WireWorld* Briana Silvermana

Komórka może znajdować się w jednym z czterech stanów:

1. Pusta,
2. Głowa elektronu,
3. Ogon elektronu,
4. Przewodnik

Stany oznaczane są kolorami, zwykle biały, czerwony, żółty, czarny. Kolejne generacje budowane są z wykorzystaniem zestawu pięciu zasad:

- Komórka pozostaje Pusta, jeśli była Pusta.
- Komórka staje się Ogonem elektronu, jeśli była Głową elektronu.
- Komórka staje się Przewodnikiem, jeśli była Ogonem elektronu.
- Komórka staje się Głową elektronu tylko wtedy, gdy dokładnie 1 lub 2 sąsiadujące komórki są Głowami Elektronu.
- Komórka staje się Przewodnikiem w każdym innym wypadku.

W *WireWorld* stosuje się sąsiedztwo Moore'a¹.

¹Sąsiedztwo 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).

2 Opis funkcjonalności

2.1 Jak korzystać z programu

Program uruchamiany będzie za pomocą pliku wykonywalnego z rozszerzeniem ***.jar**. Początkową planszę utworzyć będzie można na dwa sposoby:

1. wczytując dane z pliku tekstowego,
2. używając interfejsu graficznego poprzez zaznaczanie pól lub wstawiając gotowe elementy np. diody.

Generacje wyświetlane będą w oknie programu w czasie rzeczywistym po kliknięciu przycisku **"Play/Stop"**

2.2 Przykładowe użycie programu

1. Wczytanie pliku z danymi.
2. Podanie liczby generacji.
3. Wybór zapisywania generacji do plików graficznych (w ustawieniach).
4. Zatrzymanie programu w dowolnym momencie.
5. Wyeksportowanie aktualnego stanu planszy do pliku tekstowego.
6. Zakończenie programu.

2.3 Możliwości programu

Podstawowe:

- Utworzenie planszy poprzez wczytanie pliku tekstowego lub za pomocą interfejsu graficznego.
- Edytowanie wczytanych z pliku danych w oknie programu.
- Podgląd w czasie rzeczywistym na przeprowadzane generacje.
- Wstawianie struktur takich jak diody i bramki logiczne.
- Zapisanie aktualnej generacji do pliku tekstowego.

Dodatkowe:

- Zapisywanie generacji do plików graficznych.
- Możliwość zmian ustawień programu, np.: kolorów komórek, nazw plików wyjściowych, zapętlania generacji.

3 Format danych i struktura plików

3.1 Struktura katalogów

- WireWorld – katalog główny projektu
 - src – katalog z plikami źródłowymi
 - * img – katalog z grafikami wykorzystywanymi w programie
 - * test – testy poszczególnych klas oraz całego programu
 - data – katalog z danymi tekstowymi
 - pictures (opcjonalnie) – katalog z generacjami w postaci obrazów

3.2 Struktura plików

Ten podrozdział dotyczy zarówno danych wejściowych jak i wyjściowych, ponieważ ich struktura jest taka sama. Ma to na celu umożliwienie używania danych wyjściowych jako wejść dla kolejnych uruchomień programu.

Plik tekstowy powinien w każdym wierszu zawierać informację o tym jaki element będzie wstawiany, a następnie parę liczb będącą współrzędnymi elementu na planszy. W przypadku struktur, które zajmują więcej niż jedno pole, współrzędne oznaczają jej lewy górny róg. Za początek układu współrzędnych uznaje się komórkę w lewym górnym rogu planszy. Liczby generacji nie podaje się w pliku – będzie ona wpisywana w oknie programu.

Przykładowe dane wejściowe:

```
diodenormal 0 3
diodereversed 0 9
electronhead 1 3
electronhead 1 9
electrontail 0 3
electrontail 0 9
```

Program będzie rozpoznawał następujące elementy:

- **empty** – pusta komórka. Element opcjonalny - plansza domyślnie wypełniana jest komórkami pustymi,
- **electronhead** – „głowa” elektronu,
- **electrontail** – „ogon” elektronu,
- **wire** – przewód,
- **diodenormal** – normalna dioda,
- **diodereversed** – odwrócona dioda,
- **orgate** – bramka OR (alternatywa),
- **xorgate** – bramka XOR (alternatywa wykluczająca),
- **andgate** – bramka AND (koniunkcja),
- **notgate** – bramka NOT (negacja).

4 Scenariusz działania programu

4.1 Scenariusz ogólny

1. Użytkownik uruchamia program.
2. Użytkownik tworzy nową planszę wczytując plik tekstowy lub ręcznie ustawiając elementy.
3. W przypadku wybrania opcji tworzenia planszy z pliku, program sprawdza poprawność danych². Jeśli są poprawne, zostaną wczytane do aplikacji.
4. Użytkownik zadaje liczbę generacji i klika przycisk rozpoczynający działanie automatu komórkowego.

²Oznacza to, że program sprawdza czy nie podano niedozwolnych znaków (liczb ujemnych bądź liter i symboli specjalnych) oraz czy nazwy elementów są rozpoznawalne.

5. Program przeprowadza zadaną liczbę generacji, wyświetlając je w czasie rzeczywistym.
6. Ostatni stan planszy zapisywany jest do pliku tekstowego.

4.2 Obsługa błędów, nieprawidłowych danych oraz sytuacji wyjątkowych

4.2.1 Nieprawidłowy format danych wejściowych

Dane uznane zostaną za błędne jeśli:

- występuje nieznan element,
- współrzędne nie są liczbą całkowitą dodatnią,
- współrzędne nie są parą liczb (tzn. występują więcej lub mniej niż 2 liczby po nazwie elementu).

Dla takich danych program wyświetli komunikat o błędzie i pozwoli wybrać inny plik.

4.2.2 Błędne ścieżki plików wyjściowych

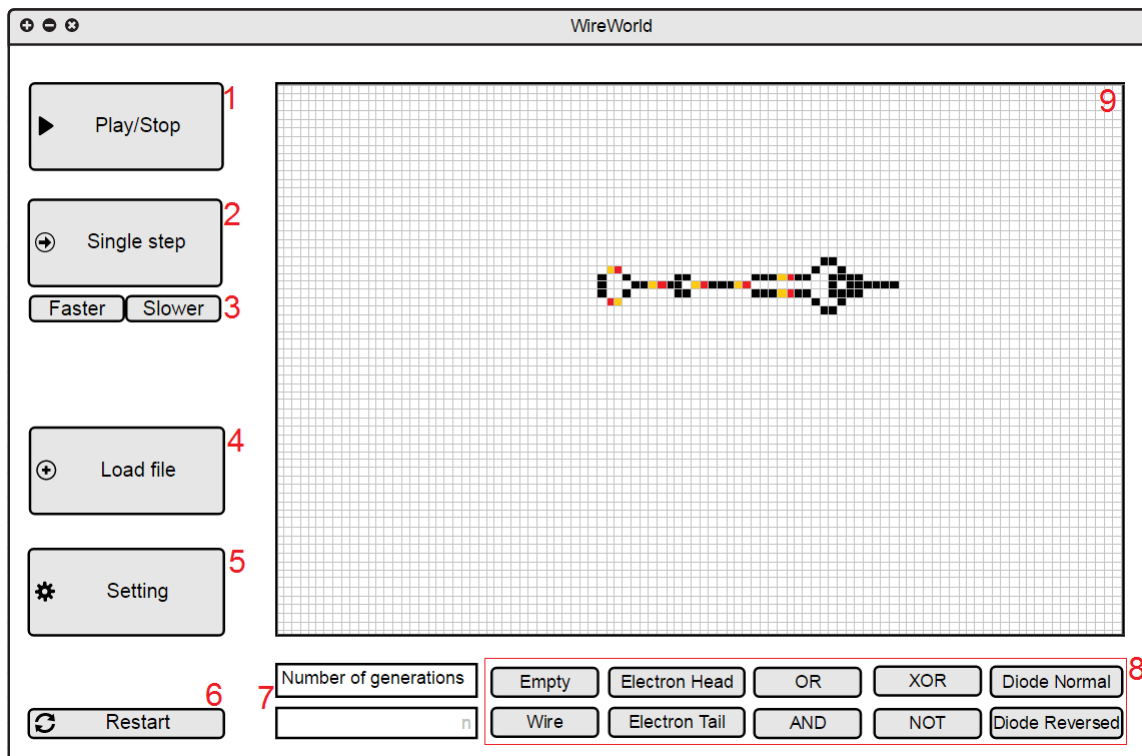
W oknie ustawień, podając nazwy do plików można podać także pełne ścieżki. Jeśli nazwa pliku zawiera niedozwolone znaki, tj. \ / : * ? | < > " lub podana ścieżka nie istnieje, pojawi się komunikat informujący o tym.

4.2.3 Nieprawidłowa wartość w polu z liczbą generacji

Jeśli wprowadzona liczba generacji nie jest liczbą całkowitą dodatnią, wówczas pole zostanie wyczyszczone i pojawi się informacja o błędzie wraz z podpowiedzią prawidłowego uzupełnienia pola.

4.3 Ekran działania programu

4.3.1 Okno główne

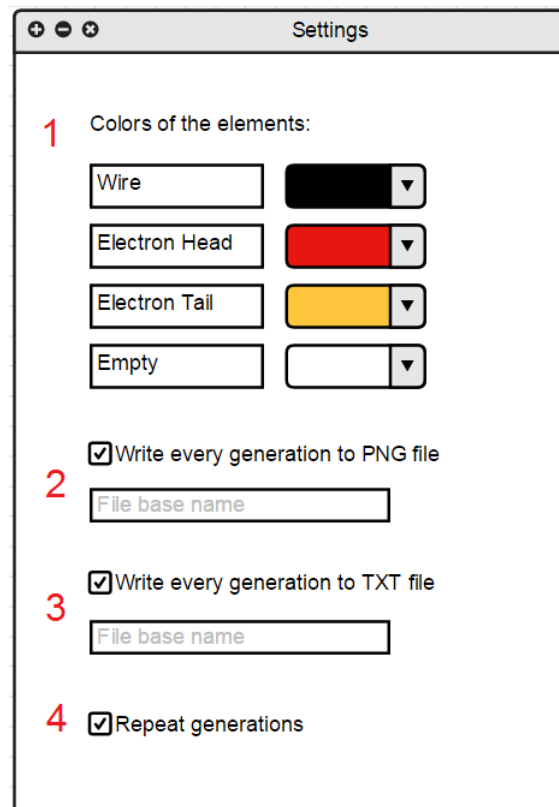


Rysunek 1: Główne okno programu

Objaśnienia:

1. Rozpoczynanie(wznawianie) i zatrzymywanie działania automatu.
2. Wykonanie jednej generacji.
3. Wydłużenie lub skrócenie czasu pomiędzy wyświetlanymi stanami planszy.
4. Wczytanie pliku z danymi wejściowymi.
5. Ustawienia.
6. Restart. Usuwa aktualne dane, czyści planszę.
7. Liczba generacji. Pole to aktywne jest jedynie przed pierwszym uruchomieniem generacji.
8. Panel z dostępnymi do wstawienia elementami.
9. Plansza, na której wyświetlane będą generacje.

4.3.2 Ustawienia



Rysunek 2: Okno ustawień

Objaśnienia:

1. Panel wyboru kolorów elementów.
2. Wybór czy chcemy zapisywać każdą kolejną generację jako obraz.
3. Wybór czy chcemy zapisywać każdą generację do pliku tekstowego.
4. Opcja, która pozwala zdecydować czy program ma się powtarzać po wykonaniu określonej liczby generacji.