Specyfikacja funkcjonalna – Wireworld

Krzysztof Dąbrowski i Jakub Bogusz

8 maja 2019

Spis treści

Rozdział 1

Cel projektu

Celem projektu jest implementacja automatu komórkowego Wire World w języku Java z interfejsem graficznym zaimplementowanym przy pomocy biblioteki JavaFX. Gotowy program ma pozwalać użytkownikowi przeprowadzać symulacje zgodne z określonymi zasadami. Parametrami generacji użytkownik będzie mógł sterować ręcznie przy pomocy graficznego interfejsu użytkownika.

Interfejs będzie wyświetlał na bierząco podgląd kolejnych generacji. Użytkownik będzie miał możliwość wstrzymania symulacji oraz zmianę stanu planszy przy pomocy narzędzi edycji.

Ponadto będzie istnieć również możliwość przełączenia trybu symulacji z Wire World na automat komórkowy Game of life.

Rozdział 2

Opis ogólny problemu

2.1 Wstęp

Projektu skupia się na realizacji 3 głównych aspektów problemu. Są to odpowiednio automaty komórkowe „Game of Life" i „Wireworld" oraz wizualna prezentacja działania tych automatów.

2.2 Wire World

Wire World jest automatem komórkowym wymyślonym przez Briana Silvermana w roku 1987. Jest często używany do symulacji elementów elektronicznych operujących na wartościach bitowych. Pomimo prostoty reguł, jakie nim rządzą, za pomocą Wireworld można nawet stworzyć działający komputer.

2.2.1 Symulacja

Stany Komórka może znajdować się w jednym z czterech stanów:

- pusta,
- głowa elektronu,
- ogon elektronu,
- przewodnik.

Pokolenie to zbiór stanów wszystkich komórek w danej chwili. Gdy stan pokolenia jest ustalony, możliwe jest utworzenie nowego (potomnego) pokolenia komórek, powstających według poniższych zasad.

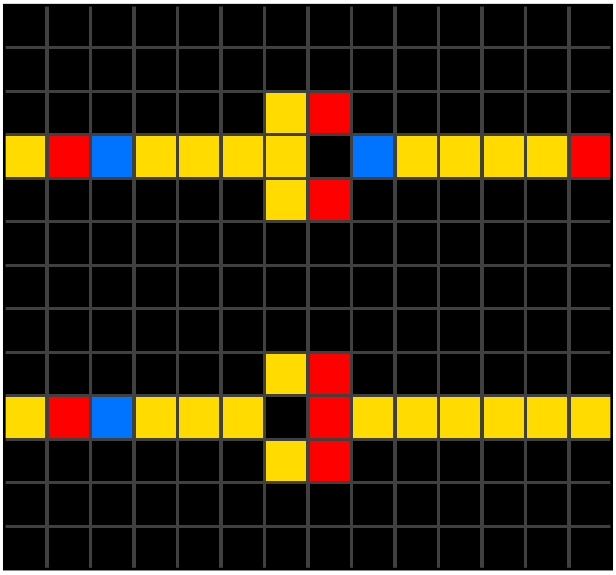
Reguły Następne pokolenie generowane jest zgodnie z regułami:

- Jeżeli komórka jest pusta, to pozostaje pusta niezależnie od jej otoczenia,

- Jeżeli komórka jest głową elektronu, to zmieni się w ogon elektronu,
- Jeżeli komórka jest ogonem elektronu, to zmieni się w przewodnik,
- Jeżeli komórka jest przewodnikiem i sąsiaduję z jedną lub dwoma komórkami będącymi głowami elektronu, to zmieni się w przewodnik.

2.2.2 Struktury

Symulacja przeprowadzona zgodnie z powyższymi regułami może prowadzić do powstania ciekawych obiektów zwanych strukturami.



Rysunek 2.1: Przykłady struktur - dioda przewodząca i nieprzewodząca

2.3 Game of life

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

2.3.1 Symulacja

Stany Komórka może znajdować się w jednym z dwóch stanów:

- żywa,
- martwa.

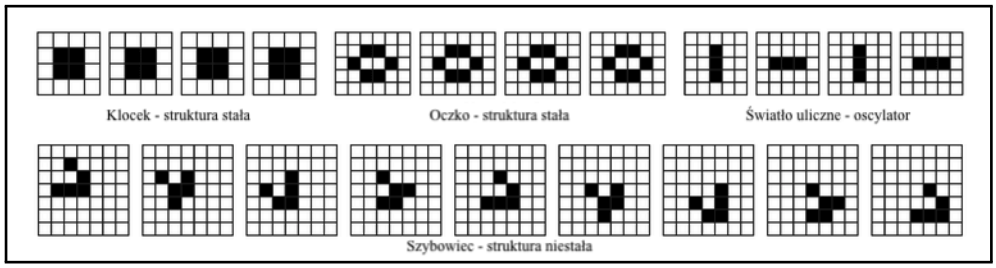
Pokolenie to stan wszystkich komórek w danej chwili. Gdy stan pokolenia jest ustalony, możliwe jest utworzenie nowego (potomnego) pokolenia komórek, powstających według poniższych zasad.

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.

2.3.2 Struktury

Symulacja przeprowadzona zgodnie z powyższymi regułami może prowadzić do powstania ciekawych obiektów zwanych strukturami.



Rysunek 2.2: Przykłady struktur

Reguły symulacji umożliwiają również tworzenie dużo bardziej skomplikowanych struktur (jak na przykład maszyna Turinga – https://youtu.be/My8AsV7bA94).

2.4 Interfejs użytkownika

Ważnym aspektem programu jest sposób interakcji z użytkownikiem. Realizowana aplikacja będzie obsługiwana poprzez graficzny interfejs użytkownika.

Interfejs pozwoli na poniższe funkcjonalności:

- Bieżący podgląd stanu symulacji
- Możliwość dostosowania prędkości symulacji
- Zapis aktualnego pokolenia do pliku
- Wyczytanie pokolenia z pliku
- Edycje poszczególnych komórek w trakcie symulacji
- Wstawianie figur wybranych z przybornika

Rozdział 3

Działanie programu

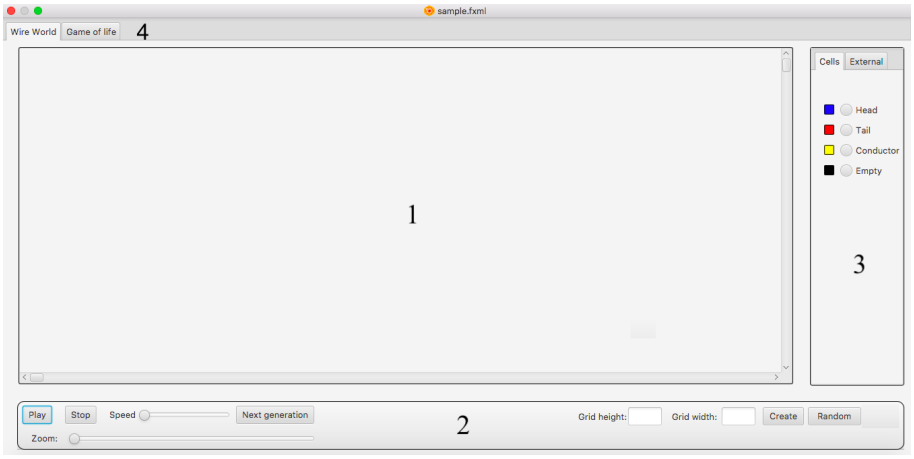
3.1 Opis komunikacji z użytkownikiem

Po uruchomieniu programu zostanie wyświetlony graficzny interfejs, który pozwoli użytkownikowi sterować parametrami w dowolny sposób, zgodny z regułami symulacji. W razie wystąpienia błędów program będzie informować użytkownika wyświetlając nowe okno zawierające opis błędu.

W ramach interfejsu użytkownik będzie mógł wykonywać czynności opisane w punkcie 2.4 oraz przełączać między automatami Game of Life i Wireworld.

3.2 Wygląd interfejsu użytkownika

Interfejs użytkownika będzie wyglądał następująco. Wyszczególnione obszary są zaznaczone numerami.



Rysunek 3.1: Całokształt interfejsu użytkownika

3.3 Opis elementów

1 Podgląd aktualnego stanu symulacji
W tej części będzie wyświetlane bieżące pokolenie symulacji aktualnego automatu komórkowego. Po bokach panelu znajdują się **suwak** pozwalające na zminę oglądanego obszaru planszy. Będą one szczególnie przydatne po przybliżeniu obrazu.

2 Ustawienia symulacji i planszy
Sekcja kontrolek umożliwiających sterowanie symulacją oraz zminę całości planszy.

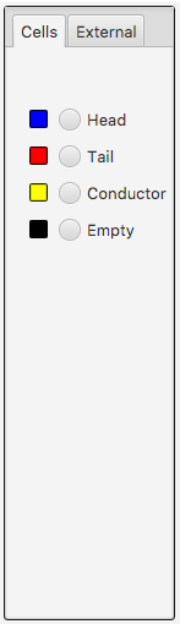
Po lewej stronie znajduje się grupa elementów wpływających bezpośrednio na podgląd symulacji.

- Play – uruchamia automatycznie przechodzenie między staniami symulacji,
- Stop – zatrzymuje automatycznie przechodzenie między staniami symulacji,
- Speed – zmienia tempo automatycznego przechodzenie między staniami symulacji,
- Next generation – ręczne przejście do następnego stanu symulacji,
- Zoom – zmienia skalę podglądu aktualnego stanu.

Po prawej stronie umieszczona jest grupa elementów pozwalających na reset całego stanu planszy.

- Grid height, Grid height – wprowadzenie rozmiaru planszy jaka zostanie utworzona,
- Create – Tworzy plansze o podanym rozmiarze, gdy plansza już istnieje i podany rozmiar jest równy planszy już istniejącej, opis przycisku zamienia się na "Clear", i wciśnięcie go zmienia stan wszystkich komórek na "Pusty" w przypadku WireWorld lub "Martwy" w przypadku Game of Life,
- Random – Nadanie wszystkim komórkom losowych stanów,

3 Przybornik
Boczny przybornik zawierał będzie dwie zakładki, które będą pozwalały na edycje fragmentów planszy.



(a) Przybornik Cells

(b) Przybornik External

Zakładka **Cells** umożliwi wybranie stanu komórki, a następnie nadanie komórkom wybranego stanu poprzez klikanie myszą komórek w oknie podglądu.

Zakładka **External** da dostęp do zestawu gotowych figur, które użytkownik będzie mógł wstawić w wybranym miejscu planszy. Użytkownik będzie mógł dodać własne figury do przybornika oraz zapisać i wczytać stan całej planszy przy pomocy guzików z tej zakładki.

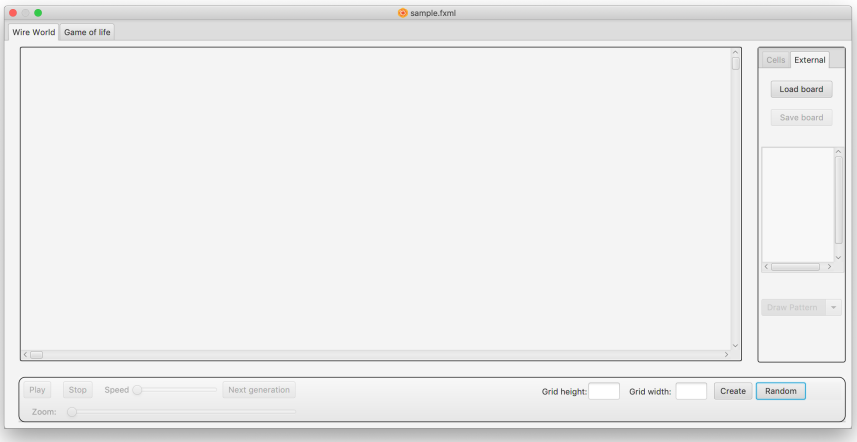
4 Wybór automatu
Wstążka pozwalająca na zmianę aktywnego automatu komórkowego.

3.4 Stany interfejsu

Podczas korzystania z aplikacji interfejs będzie ulegał drobnym zmianom. Głównie elementy odpowiadające za aktualnie niedostępne opcje będą zmieniać kolor na szary.

3.4.1 Po włączeniu programu

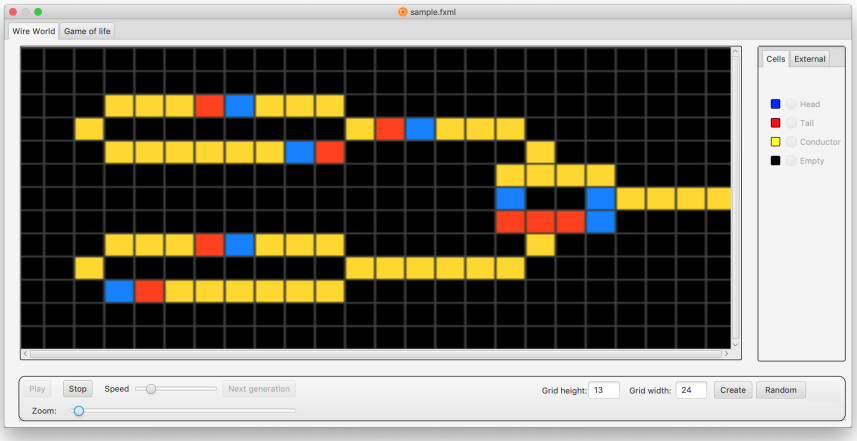
Większość kontrolek jest **wyłączonych** do momentu wczytania planszy z pliku lub wygenerowania losowej. Program domyślnie wyświetla okno symulacji automatu Wireworld.



Rysunek 3.3: Stan interfejsu po uruchomieniu aplikacji

3.4.2 W trakcie symulacji

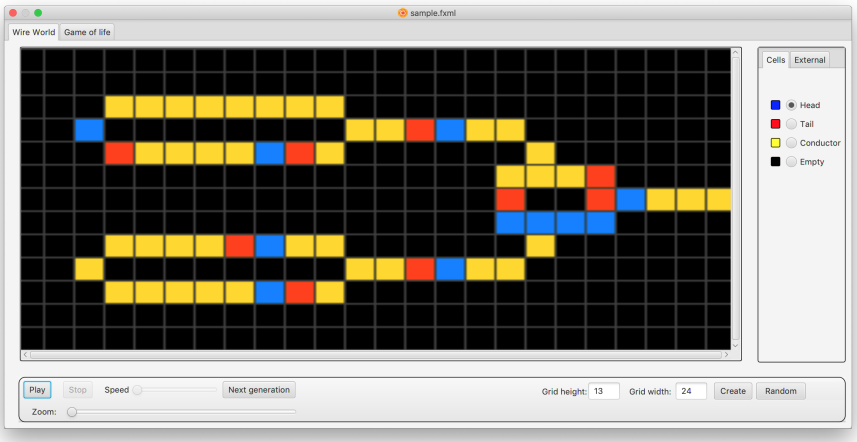
Można zmienić tempo symulacji oraz wstrzymać generację następnych pokoleń, co umożliwi edycję aktualnego pokolenia.



Rysunek 3.4: Stan interfejsu w trakcie symulacji

3.4.3 Symulacja wstrzymana

Możliwe jest edytowanie komórek przy pomocy narzędzie oraz wstawianie figur.



Rysunek 3.5: Stan interfejsu przy wstrzymanej symulacji

3.5 Tryb automatu Game of Live

Program obsługuje również automat komórkowy Game of Live. Przełączenie między aktualnym automatem jest realizowane przy pomocy wstążki na górze interfejsu. Większość elementów interfejsu pozostanie taka sama jak w automacie Wireworld. Inaczej będzie wyglądał przybornik do edycji planszy, ponieważ automat ten zawiera inne stany komórek.

Rozdział 4

Edycja planszy

Użytkownik ma kilka możliwości na wgranie planszy do programu lub edycję już istniejącej. Są one wymienione poniżej.

4.1 Plik wejściowy

Program pozwala na wczytanie całego stanu planszy (wraz z rozmiarem) z pliku. Dzięki temu możliwe jest łatwe odtworzenie stanu ze wcześniejszej symulacji lub prezentacja skomplikowanego układu.

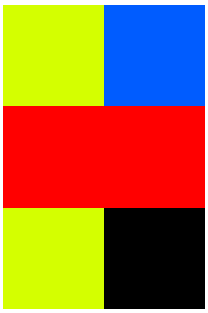
Program będzie w stanie obsłużyć różnie formaty plików wymienione poniżej.

4.1.1 Plansza w formacie JSON

W celu nadania danym przenośności, oraz ułatwienia generacji danych przy pomocy zewnętrznych narzędzi zostanie wykorzystany format JSON. Jest to **domyślny format** zapisu planszy.

Przykład

Przykład zapisu przedstawionej planszy.



Rysunek 4.1: Przykładowa plansza

Plik reprezentujący planszę:

```
{
  "width" : 2,
  "height" : 3,
  "states" : [ "HEAD", "TAIL", "HEAD", "HEAD", "HEAD", "EMPTY" ]
}
```

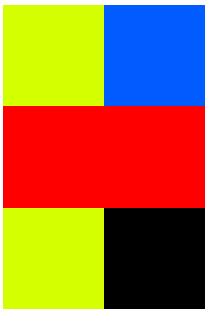
Wartości `width` i `height` określają wymiary planszy. `states` zawiera tablicę reprezentującą poszczególne stany komórek w kolejności od lewej do prawej oraz od góry do dołu. Element są najpierw jest pierwszy wiersz, potem drugi i tak aż do ostatniego.

4.1.2 Plansza w formacie XML

Możliwy jest również zapis stanu planszy do pliku XML. Z uwagi na mniejszą czytelność oraz większy rozmiar plików tego typu nie jest to domyślnie używany format. Użytkownik będzie miał możliwość zmiany formatu pliku, do jakiego chce zapisać planszę przy pomocy interfejsu graficznego.

Przykład

Przykład zapisu przedstawionej planszy.



Rysunek 4.2: Przykładowa plansza

Plik reprezentujący planszę:

```
<Board>
  <width>2</width>
  <height>3</height>
  <states>
    <states>CONDUCTOR</states>
    <states>HEAD</states>
    <states>TAIL</states>
    <states>TAIL</states>
    <states>CONDUCTOR</states>
    <states>EMPTY</states>
  </states>
</Board>
```

Wartości `width` i `height` określają wymiary planszy. `states` zawiera tablicę reprezentującą poszczególne stany komórek w kolejności od lewej do prawej oraz od góry do dołu. Element są najpierw jest pierwszy wiersz, potem drugi i tak aż do ostatniego.

4.2 Przybornik stanów

Jedną z części interfejsu użytkownika będzie przybornik pozwalający na wybór jednego ze stanów aktualnego automatu komórkowego. Po zaznaczeniu jednego ze stanów użytkownik będzie miał możliwość zamiany wybranych komórek aktualnego pokolenia na wybrany stan.

4.3 Wstawianie figur

Interfejs będzie zawierał zestaw typowych figur dla aktualnego automatu komórkowego. Możliwe będzie zaznaczenie wybranej figury oraz wstawienie jej w wybrane miejsce symulacji.

Użytkownik będzie mógł wybrać figurę z listy figur. Następnie po kliknięciu wybranego pola planszy figura zostanie wstawiona w to miejsce, tak by lewy górny róg figury znalazł się w wybranym miejscu.

4.3.1 Rozszerzanie listy figur

Użytkownik będzie miał możliwość dodania własnych figur do domyślnej listy. Interfejs będzie zawierał przycisk wywołujący edytor pojedynczej figury. Po wywołaniu edytora użytkownik będzie mógł narysować własną figurę, która po zatwierdzeniu zostanie dodana do listy. Figury rozszerzające podstawową listę będą zapisywanie w pliku konfiguracyjnym. Dzięki temu po uruchomieniu programu zostaną one ponownie załadowane.

4.4 Zapis stanu planszy

Program będzie w stanie zapisać aktualny stan planszy w formacie, który umożliwi późniejsze wczytanie planszy. Użytkownik będzie miał możliwość wyboru formatu pliku do jakiego chce zapisać stan.

4.5 Wczytanie stanu planszy

Program będzie w stanie wczytać stan planszy z pliku wskazanego przez użytkownika. Typ pliku zostanie rozpoznany po rozszerzeniu zapisanemu na końcu nazwy pliku (znaki po ostatniej kropce).

Rozdział 5

Sytuacje wyjątkowe

Czasem działanie programu może ulec zmianie na skutek nieprawidłowych danych wejściowych, niestandardowych ustawień wprowadzonych przez użytkownik lub z przyczyn losowych. Ten rozdział opisuje jak program zachowa się w takiej sytuacji, oraz co może ją wywołać.

5.1 Zmiana działania programu

W wyjątkowych sytuacjach działanie programu może ulec drobnym zmianom.

Nieoptymalna postać liczby Jeśli użytkownik wprowadzi liczbę o dłuższej postaci niż jest to koniecznie, po zatwierdzeniu pola liczba zostanie skrócona do optymalnej długości.

Na przykład po wprowadzeniu 00034 jako wysokość planszy, po zakończeniu edycji pola wartość ta zostanie zmieniona na 34.

5.2 Błędy

Opis błędów, które mogą wystąpić w trakcie działania programu. Komunikaty ich dotyczące będą wyświetlane w nowych oknach zawierających ich treść.

5.2.1 Błędy pliku wejściowego

Pusty plik W przypadku gdy plik wskazany przez użytkownika będzie pusty program powiadomi go o tym i nie wczyta planszy.

Rozmiar planszy nie będący liczbą Jeśli w miejscu rozmiaru planszy (szerokości lub wysokości) znajdować się będą wartości inne niż liczby, liczby ujemne lub liczby których iloczyn jest większy niż 100 000 program nie wczyta rozmiaru planszy. W takiej sytuacji wyświetli odpowiedni komunikat i nie wczyta planszy.

Niezgodności rozmiaru planszy z faktycznym stanem Jeśli podany rozmiar planszy nie będzie się zgadzać z ilością podanych w nim stanów komórek, program wyświetli odpowiedni komunikat i nie wczyta planszy.

Nieprawidłowy znak w pliku Jeśli podczas czytania pliku program napotka nierozpoznawany znak, wyświetli odpowiedni komunikat i nie wczyta planszy.

Błąd składniowy Jeśli plik będzie zawierał błąd składniowy (w formatowaniu JSON lub XML), program wyświetli odpowiedni komunikat i nie wczyta planszy.

5.2.2 Błędy podczas używania GUI

Zły rozmiar planszy Jeśli użytkownik w polu przeznaczonym na rozmiar planszy spróbuje wpisać:

- znaki nie będące cyframi,
- wartość mniejszą lub równą 0,
- wartości zbyt duże (ilość komórek przekraczająca 100 000)

to pole nie zmieni wartości.

Na przykład po próbie wpisania znaków znaków 5, 2, e, 4 jako szerokość planszy pole będzie zawierało kolejno wartości: 5, 52, 52, 524. Nieprawidłowy znak zostanie zignorowany.

Zły format wczytywanego pliku Jeśli program nie obsługuję wczytywania wybranego formatu pliku to powiadomi o tym użytkownika odpowiednim komunikatem w którym również przedstawi mu obsługiwane typy plików wejściowych.

5.2.3 Błędy losowe

Brak pamięci operacyjnej Gdyby w systemie zabrakło pamięci program nie będzie w stanie funkcjonować poprawnie. Program wyświetli komunikat o błędzie i przerwie symulację. Następnie poprosi o próbę wczytania mniejszej planszy lub uruchomienia programu później.