Dokumentacja KOŃCOWE "WireWorld" DLA JĘZYKA PROGRAMOWANIA JAVA

Spis treści

1 Wprowadzone zmiany

Program "WireWorld" został podzielony na następujące klasy:

- `Main`,
- `Canvas`,
- `NewLife`,
- `Process`,
- `ProcessOfLife`,
- `ReadFile`,
- `SaveFile`.

1.1 Klasa Main

W klasie `Main` jest inicjalizacja głównej klasy `Process` i wywołanie metody `go`.

1.2 Klasa Canvas

Do klasy `Canvas` został dodany konstruktor `public Canvas` zawierający tablice, wysokość, szerokość, canvasPanel i radius dla planszy:

```
public Canvas(int[][] tabl, int widht, int hegth,
Canvas canvasPanel, int POINT_RADIUS) {
          this.widht = widht;
         this.hegth = hegth;
         this.tabl = tabl;
         this.canvasPanel = canvasPanel;
         this.POINT_RADIUS = POINT_RADIUS;
}
```

1.3 Klasa NewLife

Do klasy `NewLife` został dodany konstruktor `public NewLife` zawierający tablice, wysokość, szerokość, canvasPanel dla planszy:

```
public NewLife(int[][] tabl, int widht, int hegth,
Canvas canvasPanel) {
         this.tabl = tabl;
         this.widht = widht;
         this.hegth = hegth;
         this.canvasPanel = canvasPanel;
}
```

1.4 Klasa ProcessOfLife

Do klasy `ProcessOfLife` został dodany konstruktor `public ProcessOfLife` zawierający 2 tablicy, wysokość, szerokość, canvasPanel dla planszy:

```
public ProcessOfLife(int[][] tabl, int[][] tabl2,
int widht, int hegth, Canvas canvasPanel) {
        this.widht = widht;
        this.hegth = hegth;
        this.tabl = tabl;
        this.tabl2 = tabl2;
        this.canvasPanel = canvasPanel;
}
```

1.5 ReadFile

Do klasy `ReadFile` został dodany konstruktor `public ReadFile` zawierający 2 tablicy, wysokość, szerokość, canvasPanel dla planszy:

```
public ReadFile(int[][] tabl, Canvas canvasPanel) {
        this.tabl = tabl;
        this.canvasPanel = canvasPanel;
}
```

Został wprowadzony `JFileChooser` dla wybrania miejsca,z którego użytkownik chcę odczytać plik.

1.6 SaveFile

Do klasy `SaveFile` został dodany konstruktor `public SaveFile` zawierający 2 tablicy, wysokość, szerokość, canvasPanel dla planszy:

```
public SaveFile(int[][] tabl, int widht, int hegth) {
         this.widht = widht;
         this.hegth = hegth;
         this.tabl = tabl;
         for (int x = 0; x < widht; x++) {
             for (int y = 0; y < hegth; y++) {
                 for (int i = 0; i < tabl.length; i++) {
                     this.tabl[i] = tabl[i].clone();
                }
            }
        }
    }
```

Został wprowadzony `JFileChooser` dla zapisu pliku wynikającego.

1.7 Process

Klasa `Process` można powiedzieć jest główną klasą programu. Wprowadzone następujące zmiany:

- dodany `public class DiodaListener implements ActionListener` dla uruchomienia przycisku z `Diodą`, mamy teraz lewą i prawą diodę.
- został podpisany `slider`, dla tego, żeby było widać w którą stronę jaka prędkość generacji.

1.8 Pakiet

Pakiety:

1. `gra` - cały kod,
2. `META-INF` - wersja programu,
3. `target` - wynik po działaniu `Mavenu`, powstaje plik z rozszerzeniem `.JAR`,
4. `test` - kod dla testów jednostkowych.

2 Działania i wyniki programu

Po napisaniu koda są następujące wyniki:

2.1 Uruchomienie

Ekran działania programu jest przedtawiony na `Rysunku 1`

Ekran działania programu po uruchomeniu jest przedstawiony na `Rysunku 2`

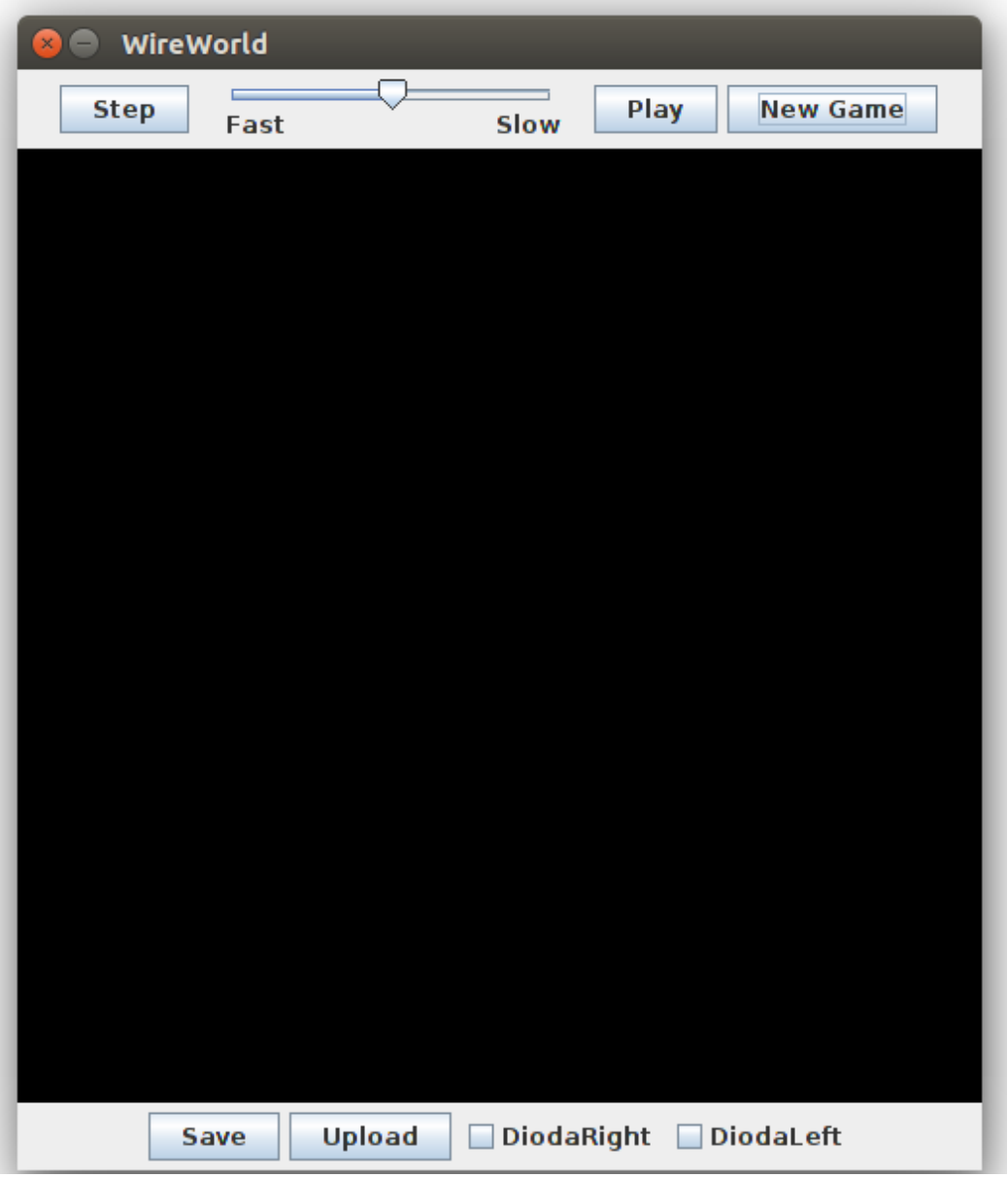
Ekran prawej diody jest przedstawiony na `Rysunku 3`

Ekran lewej diody jest przedtawiony na `Rysunku 4`

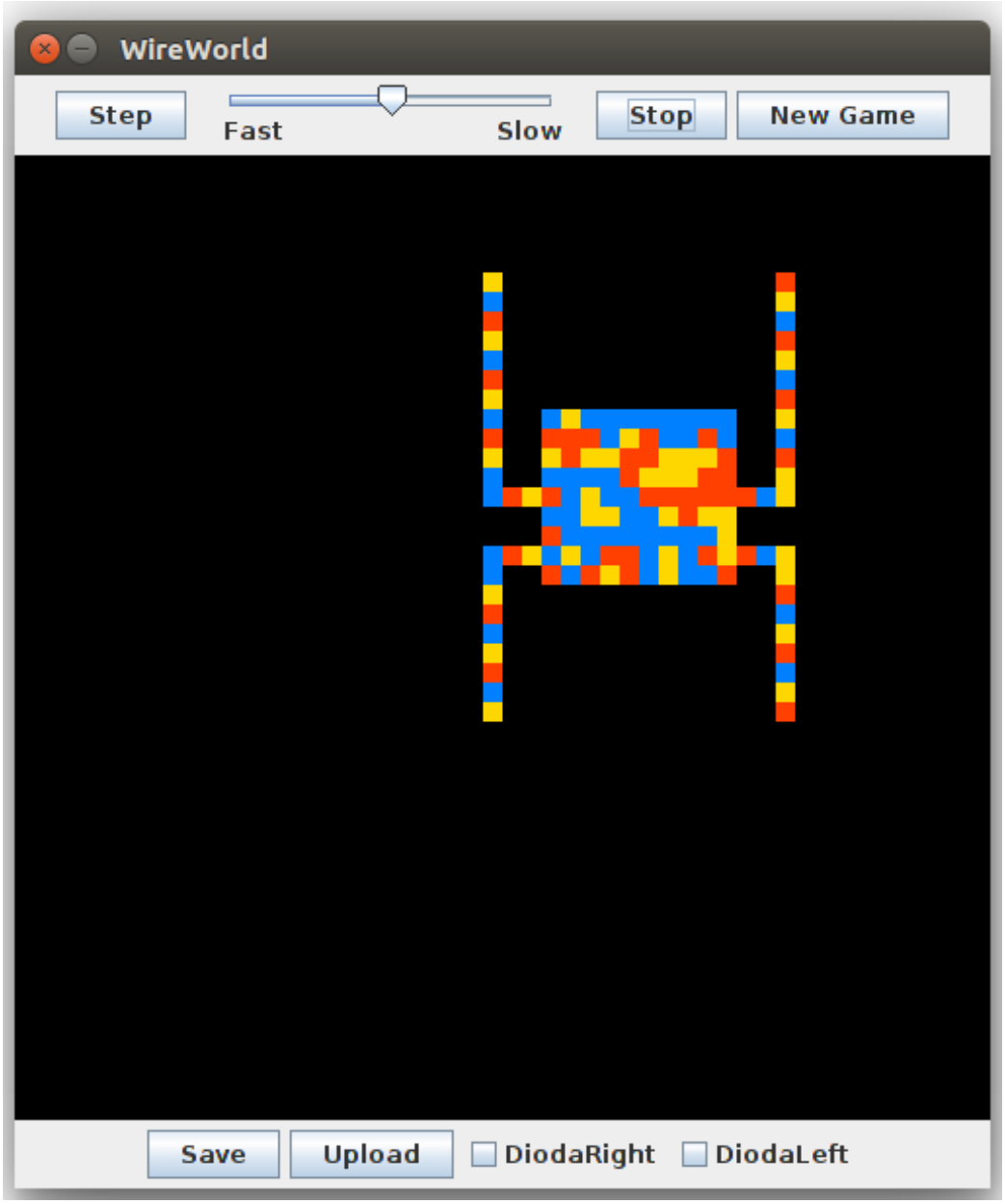
Plik `save.txt` zawierające dane o plansze i o programie jest przedstawiony na `Rysunku 5`

3 Wnioski

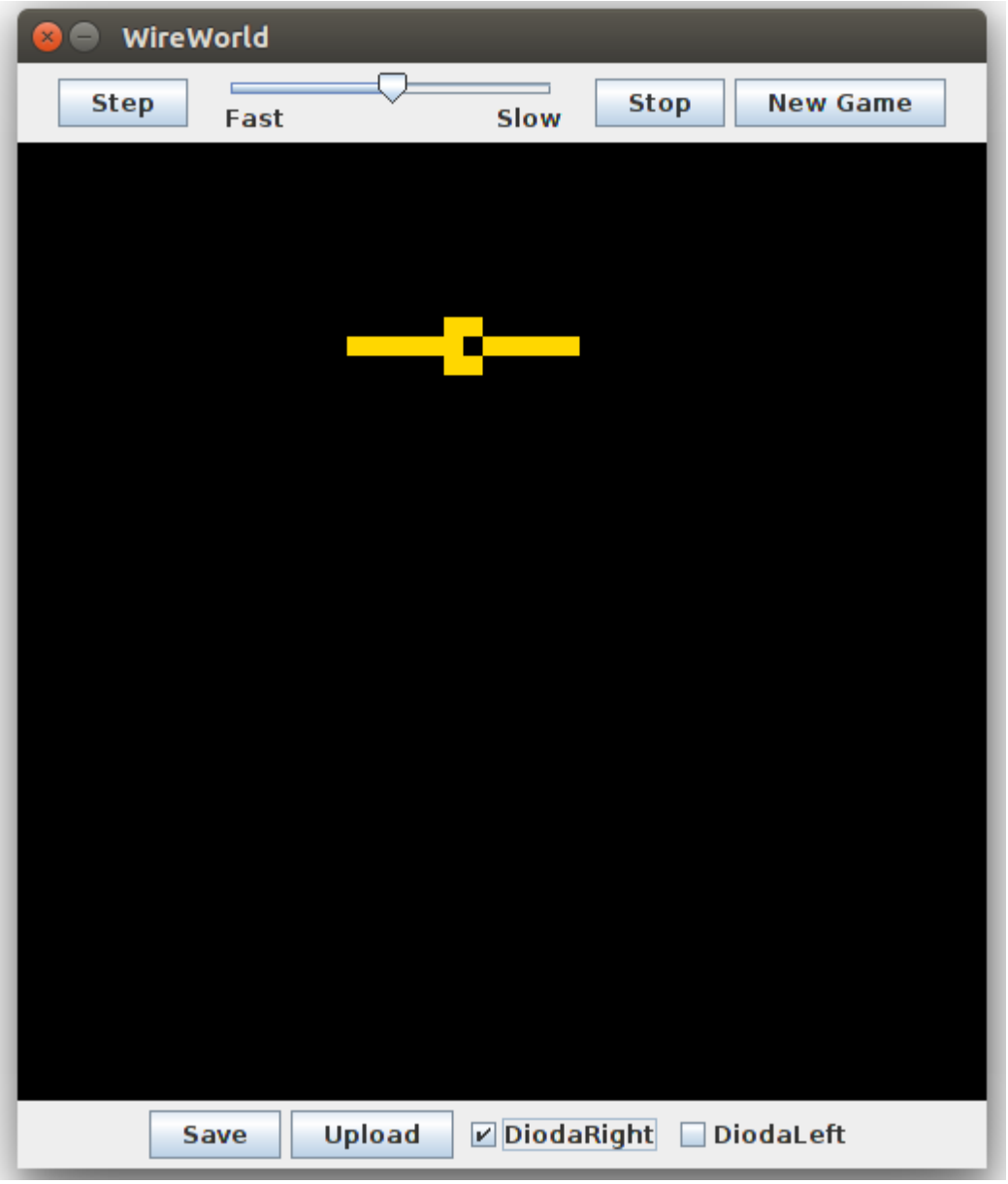
Podczas implementacji programu zaszło kilka zmian względem założeń ze specyfikacji implementacyjnej. Wynikały one głównie z lepszego rozeznania problemu, a także z zauważania oddzielnych i niezależnych metodach w większych metodach. Zostały też przeprowadzone zmiany kosmetyczne dotyczące nazw metod i wygładu programu, mające na celu wyeliminowanie nieścisłości kodu oraz poprawę jego czytelności. Wówczas podstawowe i najważniejsze założenia pochodzące ze specyfikacji implementacyjnej zostały zgodne z pierwotnymi ustaleniami, co potwierdza dobrze przemyślaną koncepcję systemu zawartą w specyfikacji funkcjonalnej oraz implementacyjnej.



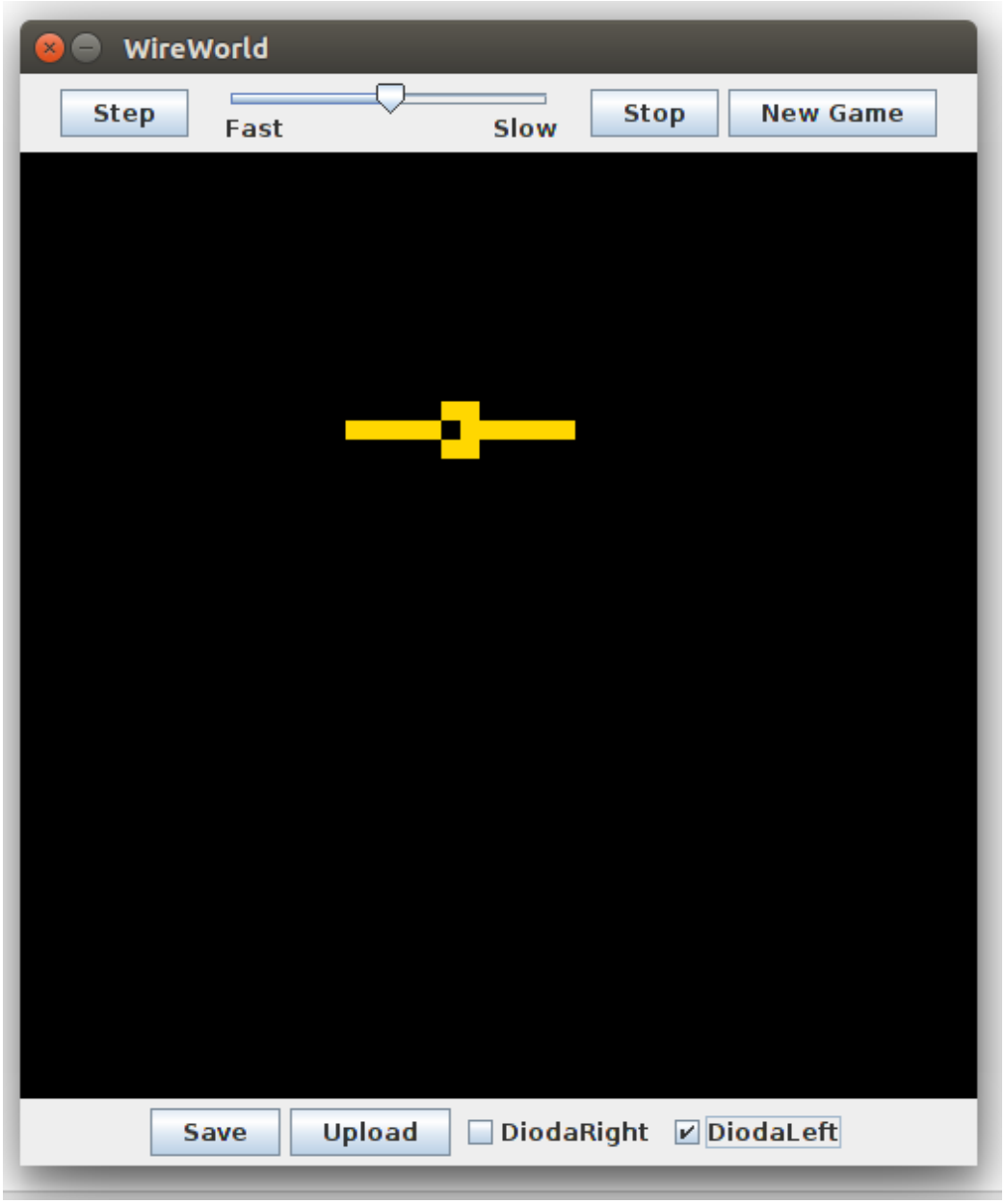
Rysunek 1: ekran programu



Rysunek 2: ekran programu po uruchomieniu



Rysunek 3: ekran prawej diody



Rysunek 4: ekran lewej diody

```
 1 ElectronTail: 24, 6;
 2 Field: 24, 7;
 3 ElectronHead: 24, 8;
 4 ElectronTail: 24, 9;
 5 Field: 24, 10;
 6 ElectronHead: 24, 11;
 7 ElectronTail: 24, 12;
 8 Field: 24, 13;
 9 ElectronHead: 24, 14;
10 ElectronTail: 24, 15;
11 Field: 24, 16;
12 Field: 24, 17;
13 Field: 24, 20;
14 Field: 24, 21;
```

Rysunek 5: plik save.txt