

Specyfikacja Funkcjonalna

Paweł Skiba, Jakub Świder

5 maja 2019

Spis treści

1	Cel projektu	2
2	Teoria	2
2.1	WireWorld	2
2.2	Game of Life	3
3	Graficzny interfejs użytkownika	4
3.1	WireWorld	4
3.2	Game of Life	5
4	Interakcja użytkownika z programem	5
4.1	Elementy interfejsu użytkownika	5
4.2	Plik wejściowy	8
4.3	Ostrzeżenia	9
5	Komunikaty błędów	10

1 Cel projektu

Celem projektu jest stworzenie aplikacji okienkowej o nazwie **CA Simulator** z wykorzystaniem języka programowania JAVA. Podczas tworzenia graficznego interfejsu użytkownika będzie wykorzystywana biblioteka graficzna SWING. Aplikacja ma na celu generowanie w czasie rzeczywistym obrazów przedstawiających kolejne generacje określonych automatów komórkowych. Jednym z nich będzie **WireWorld**, który został stworzony przez Briana Silvermana. Kolejnym automatem obsługiwanym przez aplikację będzie **Game of Life** Johna Conwaya.

2 Teoria

Automat komórkowy - jest to uporządkowany zbiór komórek, z których każda znajduje się w jednym z kilku dozwolonych stanów, np. włączony/wyłączony. Komórki przylegają do siebie tworząc zwykłe siatki, np. dwuwymiarowe w postaci kratownicy.

Stan wszystkich komórek w danej chwili nazywamy generacją. Gdy stan generacji jest ustalony, możliwe jest utworzenie nowej generacji komórek. Nowy stan danej komórki zależy od jej obecnego stanu oraz od obecnych stanów jej sąsiadów. Dla każdego automatu istnieje zbiór reguł określających zmiany stanu. W automatach komórkowych przyjęło się, że różne stany przedstawia się za pomocą różnych kolorów komórek. W zależności od rodzaju automatu bierze się pod uwagę różne rodzaje sąsiedztwa, np. w siatkach dwuwymiarowych:

- 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).

2.1 WireWorld

W automacie komórkowym WireWorld zostanie zastosowana dwuwymiarowa siatka oraz sąsiedztwo Moore'a.

Komórka może znajdować się w jednym z czterech stanów, którym odpowiadają poszczególne kolory:

- pusta - czarny;
- przewodnik - żółty;
- głowa elektronu - niebieski;
- ogon elektronu - czerwony.

Zasady obowiązujące przy tworzeniu nowej generacji:

1. komórka pozostaje *pusta*, jeśli była *pusta*.
2. komórka staje się *ogonem elektronu*, jeśli była *głową elektronu*.
3. komórka staje się *przewodnikiem*, jeśli była *ogonem elektronu*.
4. komórka staje się *głową elektronu* tylko wtedy, gdy dokładnie 1 lub 2 sąsiadujące komórki są *głowami elektronu*;
5. komórka staje się *przewodnikiem* w każdym innym wypadku.

2.2 Game of Life

W automacie komórkowym Game of Life zostanie zastosowana dwuwymiarowa siatka oraz sąsiedztwo Moore'a.

Komórka może znajdować się w jednym z dwóch stanów, którym odpowiadają poszczególne kolory:

- żywa - biały;
- martwa - czarny.

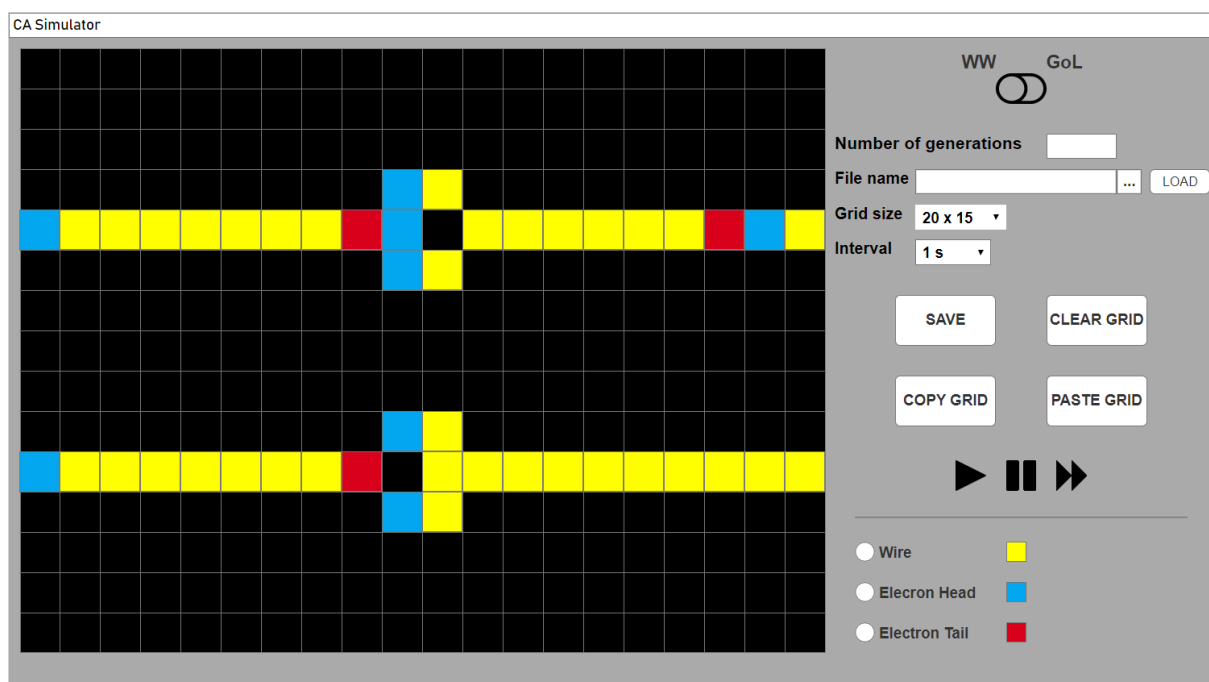
Zasady obowiązujące przy tworzeniu nowej generacji:

1. jeżeli komórka jest *martwa* i ma dokładnie 3 żywych sąsiadów to w następnej generacji staje się *żywa*;
2. jeżeli komórka jest *żywa* oraz posiada 2 lub 3 żywych sąsiadów to w następnej generacji pozostaje *żywa*, w innym wypadku komórka staje się *martwa*.

3 Graficzny interfejs użytkownika

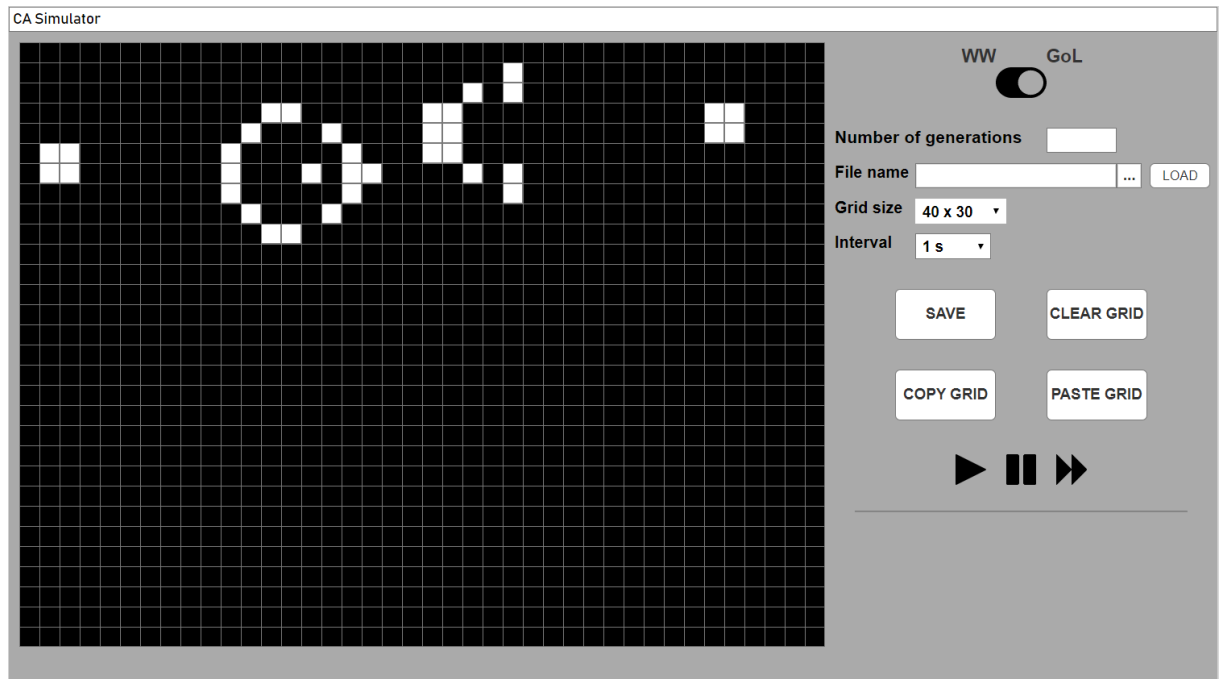
Poniżej zostaną zaprezentowane dwie odsłony graficznego interfejsu użytkownika. Za pomocą przełącznika WW/GoL można zmienić automat, który będzie symulowany przez program.

3.1 WireWorld



Rysunek 1: Graficzny interfejs użytkownika - WireWorld

3.2 Game of Life



Rysunek 2: Graficzny interfejs użytkownika - Game of Life

4 Interakcja użytkownika z programem

4.1 Elementy interfejsu użytkownika

Użytkownik z systemem może komunikować się za pomocą poszczególnych elementów graficznego interfejsu użytkownika, których opis można znaleźć poniżej.

- Plansza *Grid* - jest to główny element programu, który jest odpowiedzialny za utworzenie wstępnego obrazu generacji. Plansza jest podzielona na określoną ilość kwadratów, a każdy z nich może być w jednym z określonych stanów dla danej gry. W zależności od automatu komórkowego (WireWorld lub Game of Life) obowiązują różne metody tworzenia wstępnego obrazu. Jeżeli mówimy o WireWorld, użytkownik musi zaznaczyć odpowiednie pole wyboru (Wire, Electron Head lub Electron

Tail), a następnie lewym przyciskiem myszy zaznaczyć konkretny kwadrat, aby zmienić jego stan na ten, który użytkownik wybrał w polu wyboru. Ponowne naciśnięcie na zaznaczony już element zmieni jego stan na pusty. Nastomiast w Game of Life użytkownik po kliknięciu lewym przyciskiem myszy na dany kwadrat zmieni jego stan z żywego na martwy lub odwrotnie;

- Przełącznik *WW/GoL* - jest on odpowiedzialny za przełączanie pomiędzy automatami komórkowymi WireWorld i Game of Life. W momencie, gdy użytkownik będzie chciał przełączyć daną grę, a plansza będzie uzupełniona, system wygeneruje ostrzeżenie dotyczące utraty stanu planszy (patrz *Rysunek 3*);
- Pole tekstowe *Number of generation* - jest to pole do wprowadzenia liczby generacji. Program oczekuje wprowadzenia liczby całkowitej równej zero lub większej od zera. W pierwszym przypadku zostanie wywołana symulacja, która będzie wykonywana do momentu, gdy jej dwa kolejne stany powtórzą się lub do momentu kliknięcia przez użytkownika przycisku *Pause*. Gdy zostanie wprowadzana liczba całkowita większa od zera program wykona określoną przez użytkownika liczbę generacji. W innym przypadku wystąpi błąd opisany w następnym rozdziale;
- Pole tekstowe *File name* - jest to pole przeznaczone do wprowadzenia ścieżki do pliku z początkową generacją;
- Przycisk *...* - jest to przycisk, który otwiera okno systemowe. Dzięki temu użytkownik może odszukać interesujący go plik z początkową generacją dużo szybciej;
- Przycisk *Load* - jest odpowiedzialny za załadowanie wybranego wcześniej pliku zawierającego początkową generację do planszy. Zanim program zostanie uruchomiony można edytować planszę;
- Lista rozwijalna *Grid size* - jest to lista rozwijalna za pomocą której użytkownik może zmienić ilość obiektów (kwadratów) w planszy. Dostępne wymiary to 20x15, 40x30, 80x60, 160x120. W momencie, gdy plansza będzie uzupełniona, a użytkownik będzie chciał zmienić rozkład planszy, system wygeneruje ostrzeżenie dotyczące utraty stanu siatki (patrz *Rysunek 4*);
- Lista rozwijalna *Interval* - jest to lista rozwijalna odpowiedzialna za wybranie przez użytkownika opóźnień między kolejnymi generacjami.

Wartości, które mogą zostać wybrane z listy to 1/8 sekundy, 1/4 sekundy, 1/2 sekundy, 1 sekunda, 2 sekundy;

- Przycisk *SAVE* - odpowiada za zapisanie obecnego stanu planszy do pliku, który będzie mógł być później otworzony przez program, aby przywrócić zapisany stan planszy. Po wciśnięciu pojawi się okno pozwalające wybrać użytkownikowi nazwę pliku oraz folder docelowy;
- Przycisk *CLEAR GRID* - spowoduje przywrócenie planszy do wyjściowego stanu, to zanczy takiego w którym wszystkie komórki planszy są puste/martwe;
- Przycisk *COPY GRID* - spowoduje skopiowanie obecnego stanu planszy do schowka, aby można go było potem łatwo przywrócić. Schowek opróżnia się po wyjściu z programu, bądź przełączeniu się pomiędzy automatami komórkowymi;
- Przycisk *PASTE GRID* - powoduje przywrócenie stanu planszy ze schowka;
- Przycisk *Play* - powoduje urochomienie symulacji automatu komórkowego i generowania automatycznie kolejnych plansz, na podstawie ustalonego interwału czasowego oraz wybranego automatu komórkowego. Generacje będą prezentowane w taki sposób, że każda kolejna generacja zostanie nadpisana na planszy. Symulacja zatrzyma się po wpisanej przez użytkownika liczbie generacji, w momencie gdy nie będzie można wygenerować nowej planszy, lub po wciśnięciu przez użytkownika przycisku *Pause*;
- Przycisk *Pause* - w przypadku gdy program jest w trakcie generowania kolejnych plansz wciśnięcie spowoduje zatrzymanie generacji, w przeciwnym przypadku przycisk nie zadziała;
- Przycisk *Fast forward* - spowoduje wygenerowanie kolejnej planszy na podstawie obecnego stanu planszy;
- Pole wyboru *Wire* - jest to pole dostępne dla użytkownika tylko przy automacie komórkowym WireWorld. Po wybraniu pola wyboru *Wire* każde kliknięcie lewym przyciskiem myszy w dany obiekt (kwadrat) na planszy zmieni jego kolor na żółty, a gdy obiekt (kwadrat) ma kolor żółty, jego stan zostanie zmieniony na czarny;

- Pole wyboru *Electron Head* - jest to pole dostępne dla użytkownika tylko przy automacie komórkowym WireWorld. Po wybraniu pola wyboru *Electron Head* każde kliknięcie lewym przyciskiem myszy w dany obiekt (kwadrat) na planszy zmieni jego kolor na niebieski, a gdy obiekt (kwadrat) ma kolor niebieski, jego stan zostanie zmieniony na czarny;
- Pole wyboru *Electron Tail* - jest to pole dostępne dla użytkownika tylko przy automacie komórkowym WireWorld. Po wybraniu pola wyboru *Electron Tail* każde kliknięcie lewym przyciskiem myszy w dany obiekt (kwadrat) na planszy zmieni jego kolor na czerwony, a gdy obiekt (kwadrat) ma kolor czerwony, jego stan zostanie zmieniony na czarny.

4.2 Plik wejściowy

Użytkownik może przygotować plik tekstowy opisujący układ planszy, który będzie mógł być wczytany przez program. Plik powinien na samej górze zawierać nazwę automatu komórkowego, następnie planszę ułożoną z:

- 0 - komórka pusta w WireWorld i komórka martwa w Game of Life;
- 1 - przewodnik w WireWorld i komórka żywa w Game of Life;
- 2 - głowa elektronu w WireWorld;
- 3 - ogon elektronu w WireWorld.

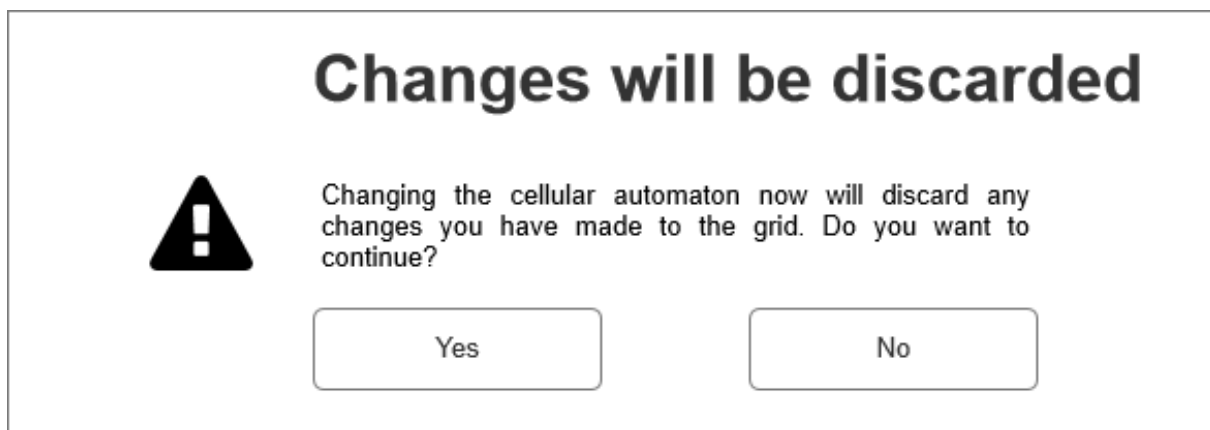
Następnie użytkownik może wprowadzić następujące sformułowania określające dane elementy (x i y odpowiadają określonej pozycji pierwszej komórki danej struktury na planszy):

- Wire(x,y) - odpowiada za umieszczenie komórki przewodnika;
- ElectronHead(x,y) - odpowiada za umieszczenie komórki głowy elektronu;
- ElectronTail(x,y) - odpowiada za umieszczenie komórki ogonu elektronu;
- Diode(x,y ,Normal) - odpowiada za umieszczenie diody w kierunku przewodzenia;
- Diode(x,y ,Reversed) - odpowiada za umieszczenie diody w kierunku zaporowym;
- AND(x,y) - odpowiada za umieszczenie bramki logicznej AND;

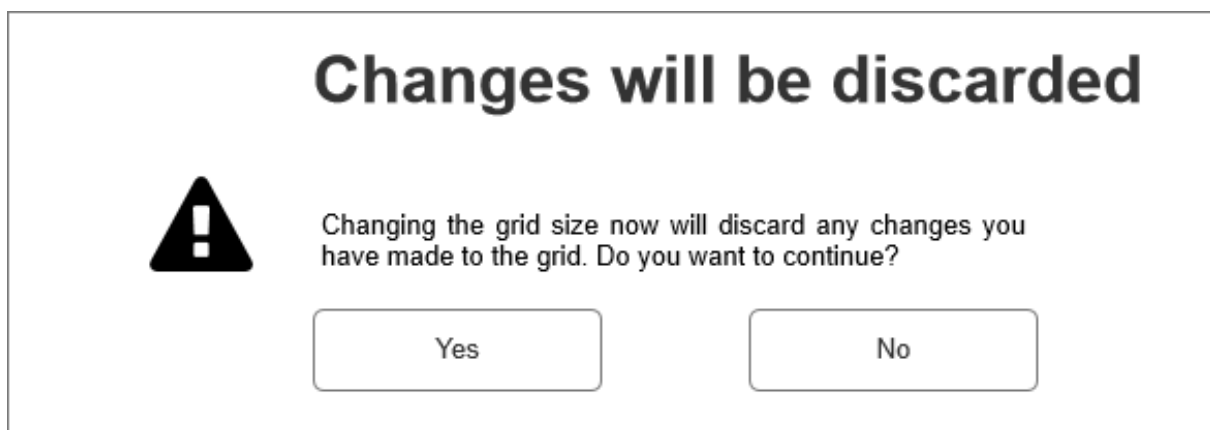
- $OR(x,y)$ - odpowiada za umieszczenie bramki logicznej OR;
- $XOR(x,y)$ - odpowiada za umieszczenie bramki logicznej XOR.

Plik może zawierać jednocześnie opis siatki i struktur. W takim wypadku opis siatki musi być umieszczony w pliku powyżej opisu elementów.

4.3 Ostrzeżenia



Rysunek 3: Okno dialogowe ostrzegające przed utratą wprowadzonych zmian



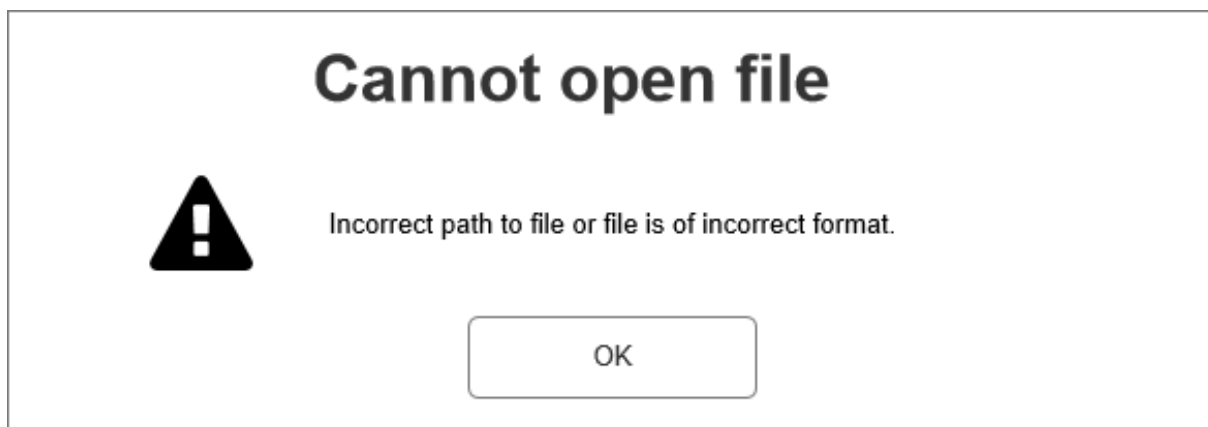
Rysunek 4: Okno dialogowe ostrzegające przed utratą wprowadzonych zmian

5 Komunikaty błędów

Program zakomunikuje błędy w następujących sytuacjach

1. Błędny plik lub ścieżka do pliku

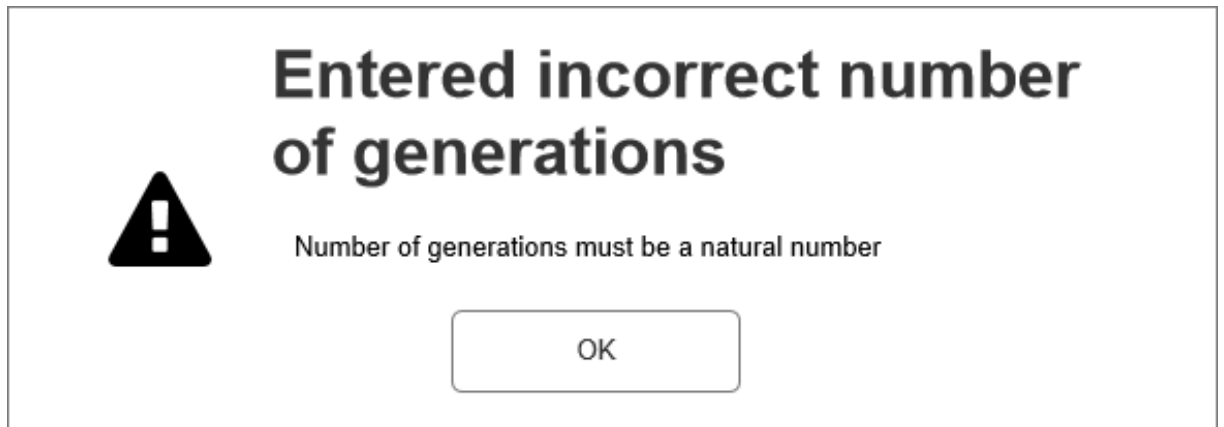
W przypadku gdy użytkownik w oknie *File name* poda błędną ścieżkę, ścieżkę do pliku o złym formacie, lub nie poda ścieżki i wciśnie przycisk *Load* wyświetli się okno z komunikatem błędu (patrz *Rysunek 4.*);



Rysunek 5: Okno komunikatu błędu dotyczącego pliku, lub ścieżki do pliku

2. Błędna liczba generacji

W przypadku wpisania w oknie *Number of generations* liczby nie będącej liczbą naturalną i wciśnięciu przycisku *Play* wyświetli się okno z komunikatem błędu (patrz *Rysunek 5.*);.



Rysunek 6: Okno komunikatu błędu dotyczącego liczby generacji