Automat Komórkowy (Java)

Patryk Cholewa, Michał Mitros (razem robiliśmy również poprzedni projekt)

June 7, 2017

1 Wstep

Nasza aplikacja jest symulatorem pewnych ogólnych automatów komórkowych. Do jej działania należy wybrać zasady, według jakich symulator ma działać, warunki graniczne oraz metode wyliczania sasiedztwa. Stany komórek w tablicy da sie ustawiać klikajac je w generowanej dynamicznie tablicy. Najważniejsza funkcjonalnościa jest symulacja automatu "Wireworld", ale da sie symulować znacznie ogólniejsze.

2 Ustawienia automatu

2.1 Sasiedztwo

2.1.1 "Moore"

Z komórka granicza wszystkie komórki stykajace sie rogami lub bokami.

2.1.2 "Neumann"

Z komórka granicza wszystkie komórki stykajace sie rogami lub bokami.

2.2 Granica

2.2.1 "Void"

Wszystkie potencjalne wartości komórek poza plansza wynosza 0 (stan: 0).

2.2.2 "Planet"

Wartości potencjalne komórki poza plansza to wartości pierwszych komórek po przeciwległej stronie.

2.3 Zasady

Posiadaja one ściśle określona budowe. Jest to lista stringów z których każdy string odpowiada kolejnemu stanowi komórki. Stringi odpowiadaja wyrażeniom warunkowym o nastepujacej budowie: "Jeżeli komórka ma stan A i ma sasiadów o stanie B w ilości C, to w nastepnej generacji bedzie mieć stan D, w przeciwnym wypadku bedzie mieć stan E". W praktyce stringi te maja budowe : B/C/D/E. Jak łatwo zauważyć brakuje tu stanu A, gdy ten jest zawarty w indeksie danego stringa na liście.

A - stringów nie może być mniej, niż zadeklarowanej liczby stanów

B - może być dowolna liczba naturalna mniejsza od liczby stanów

C - sa to zapisane rosnaco liczby sasiadów spełniajace opisywany warunek; sa to cyfry obok siebie. np. 23 odpowiada liczbie sasiadów 2 lub 3. Warto zauważyć, że komórka nigdy nie bedzie mieć wiecej, niż 8 sasiadów, wiec zapisanie samej cyfry 9 oznacza, że warunek nigdy nie bedzie spełniony, a wartość D nic nie znaczy.

D - stan nie może być wiekszy od liczby stanów

E - jak wyżej

Przykłady:

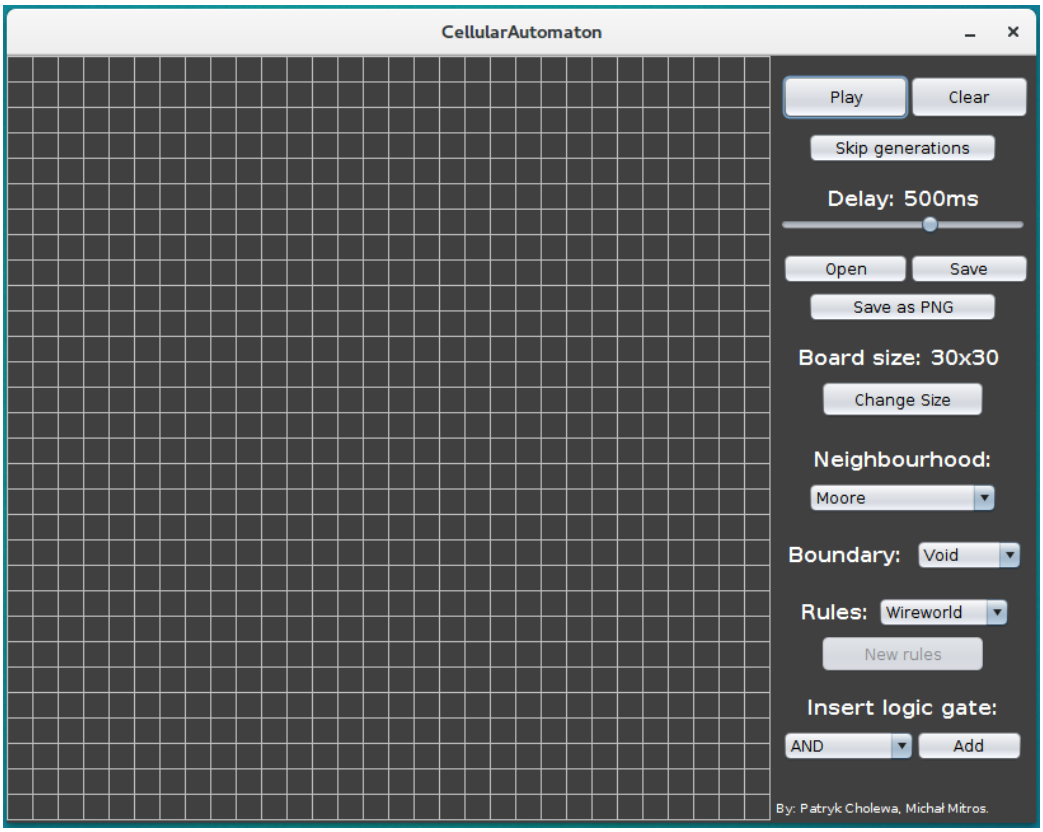
2.3.1 Gra w życie Conway'a

0: 1/3/1/0
1: 1/23/1/0

2.3.2 Wireworld

0: 0/9/1/0
1: 1/9/1/2
2: 2/9/1/3
3: 1/12/1/3

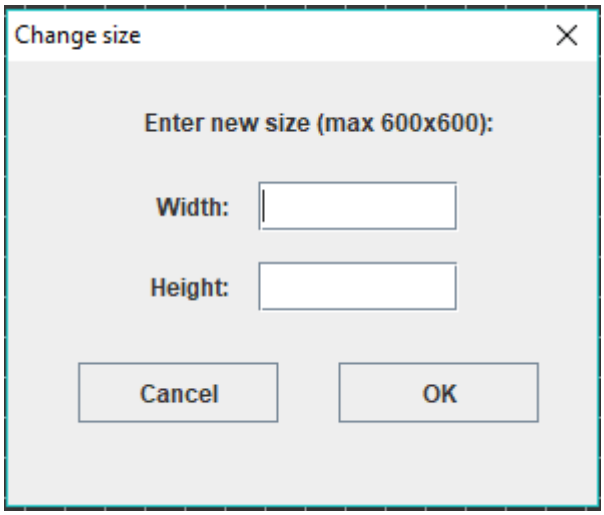
3 Instrukcja użytkownika



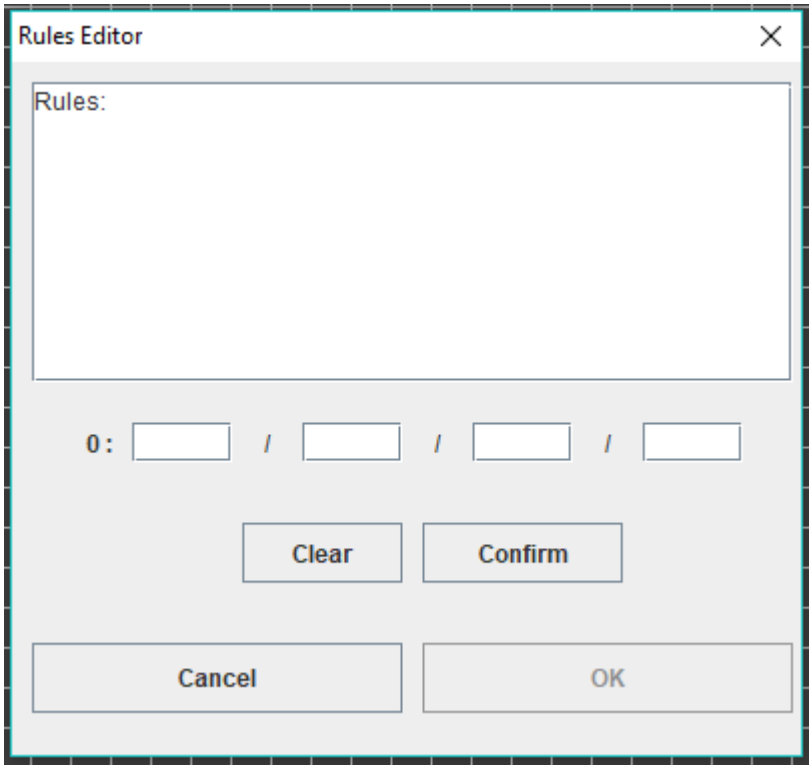
Obraz 1. Wyglad aplikacji po uruchomieniu

Aplikacja dzieli sie na tablice reprezentujaca generacje automatu i boczne menu. Aby zmienić stan komórki o +1 należy nacisnać na komórke za pomoca LPM, o -1 PPM.Symulacje uruchamiamy i przerywamy przyciskiem "**Play/Pause**". Przycisk "**Clear**" czyści plansze, a przycisk "**Skip generations**" otwiera okno umożliwiajace pominiecie wybranej ilości generacji.. Suwak "**Delay**" ustawia opóźnienie symulacji. Przycisk "**Open**" ustawia tablice na taka uzyskana przyciskiem "**Save**" zgodnie z pewna notacja (3.1.). "**Save as PNG**" zapisuje plik PNG ze zrzutem tablicy. Przycisk "**Change size**" otwiera okno zmiany rozmiarów planszy (wymiary nie moga być zerowe, ujemne, ani wieksze niż 600x600 komórek). Jeżeli chociaż jeden z wymiarów planszy bedzie wiekszy niż 150 komórek, to nie zostana wygenerowane linie ułatwiajace reczna zmiane wartości komórek. Listy rozwijalne "**Neighbourhood**", "**Boundary**" i "**Rules**" pozwalaja ustawić działanie automatu. Po wybraniu zasad "**Own Rules**" pojawia sie możliwość naciśniecia

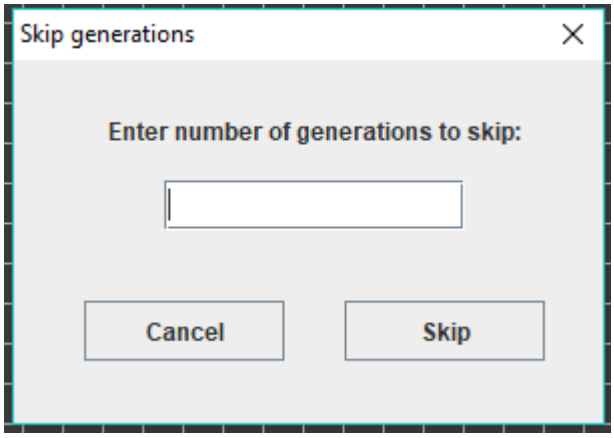
przycisku “**New rules**”. Otwiera on specjalne nowe okno bedace edytorem zasad. Przycisk “**Add**” w podmenu Insert logic gate umożliwia dodanie wybranej bramki logicznej Wireworlda (3.2.). Funkcja dodana z obowiazku projektowego.



Obraz 2. Okno zmiany wymiarów tablicy



Obraz 3. Edytor zasad. Zasady tworzone zgodnie z (2.3.).



Obraz 4. Okno pominiecia generacji.

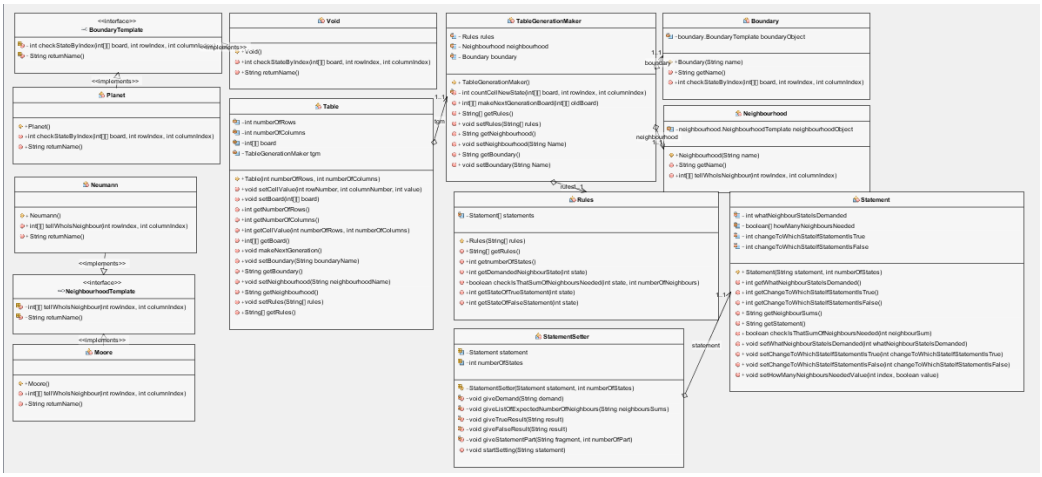
3.1 Notacja zapisu tablicy

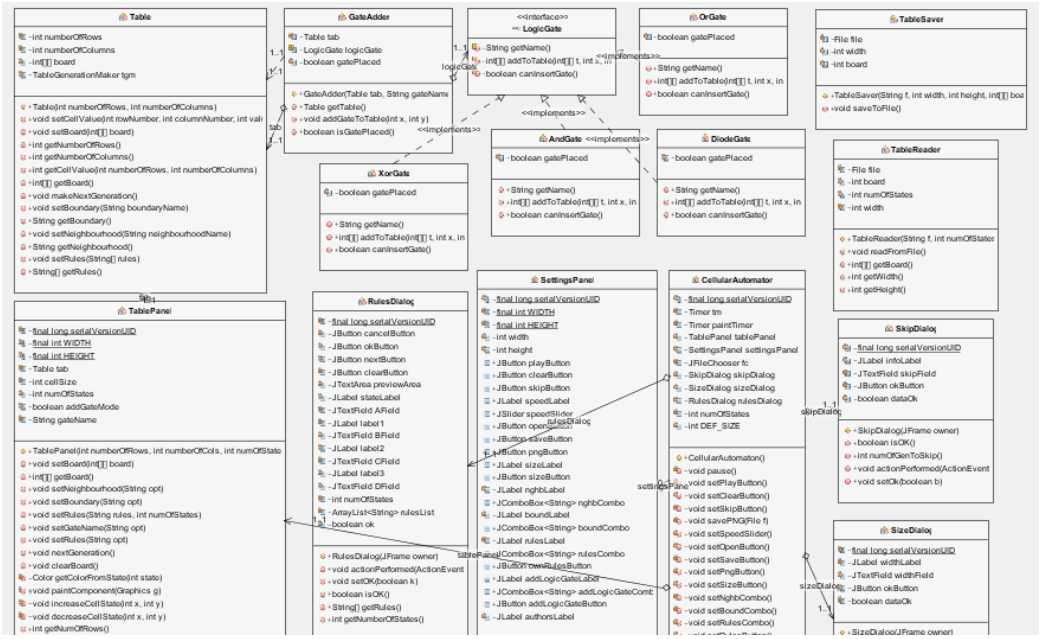
W pliku pierwsza linia sa wymiary tablicy oddzielone spacja. Kolejnych linii jest tyle, ile w tablicy znajduje sie komórek o stanie niezerowym. Każda kolejna linia zawiera oddzielone kolejno spacjami: współrzedna odcieta, współrzedna rzedna oraz stan danej komórki. Komórki, których stanu nie ustawiono, przyjmowane sa za komórki o stanie zerowym. Stan komórki nieobsługiwany przez obecnie ustawione w automacie zasady przyjmowany jest za stan zerowy.

3.2 Dodawanie bramek logicznych

Opcja wklejania bramek logicznych dostepna jest tylko wtedy, kiedy zasady ustawione sa na Wireworld. Wciśniecie przycisku "**Add**" spowoduje zmiane trybu wprowadzania komórek do automatu. Pierwsze klikniecie w obszarze tablicy wklei wybrana bramke logiczna. Po nim automat wróci do klasycznego trybu wprowadzania po jednej komórce naraz. Komórka wskazywana przez kursor jest lewa-górna komórka wklejanego obszaru, dlatego, jeżeli na prawo i pod kursorem nie ma wystarczajacej ilości dostepnych komórek, bramka nie zostanie wklejona. Wklejone bramki maja wejście/wejścia z lewej strony i wyjście z prawej strony. Automat umożliwia szybkie wklejanie czterech bramek logicznych: AND, OR, DIODE oraz XOR (Extreme OR).

4 Zależność klas





5 Kluczowe konstrukcje

5.1 Fasada "Table"

Klasa Table stanowi fasade ukrywajaca złożoności silnika automatu. Zastosowaliśmy ja przede wszystkim, aby jednoznacznie rozdzielała podział zadań miedzy nami (programujacymi), ale też, aby zmiany wewnetrzne systemu nie wymuszały zmian w obsłudze.

5.2 Strategia "Neighbourhood" i "Boundary"

Klasy Neighbourhood i Boundary odpowiadaja za podpiecie wybranych przez użytkownika klas zawierajacych wykluczajace sie metody. Zastosowanie takiego rozwiazania umożliwia zmiane ustawień automatu bez resetowania programu, a także łatwość rozbudowy o nowe algorytmy.

5.3 Interpreter "Rules"

Klasa Rules odpowiada za interpretowanie zbioru wyrażeń jej przekazanych (tu: zasad automatu). Wywołuje ona klase Statement, która interpretuje pojedyncze wyrażenie. Ten wzorzec został zastosowany, aby umożliwić stworzenie prostego jezyka edycji zasad. Edytor daje możliwość tworzenia znacznie

bardziej skomplikowanych automatów, jednocześnie jezyk zasad pozostaje prosty do posługiwania sie nim.

5.4 Fabryka "GateAdder"

Zadaniem klasy GateAdder jest delegowanie zapisania w tablicy bramki logicznej do odpowiedniej klasy. Zastosowaliśmy ten model po to aby możliwe było łatwe dodawanie nowych bramek logicznych oraz zarzadzanie nimi.