Automaty komórkowe
Projekt 1: rozszerzony automat Life

Arkadiusz Kasprzak

Streszczenie

Niniejszy dokument stanowi sprawozdanie z pierwszego projektu realizowanego w ramach przedmiotu *Automaty komórkowe*. Tematem projektu było opracowanie interaktywnej aplikacji internetowej implementującej automat 2D (tzw. *Life-like*) na siatce kwadratowej z periodycznymi warunkami brzegowymi i otoczeniem Moore'a. W sprawozdaniu omówione zostaną podstawowe zagadnienia teoretyczne opisujące implementowany automat, podstawy implementacji oraz testy działania - w tym zaobserwowane charakterystyczne struktury.

Spis treści

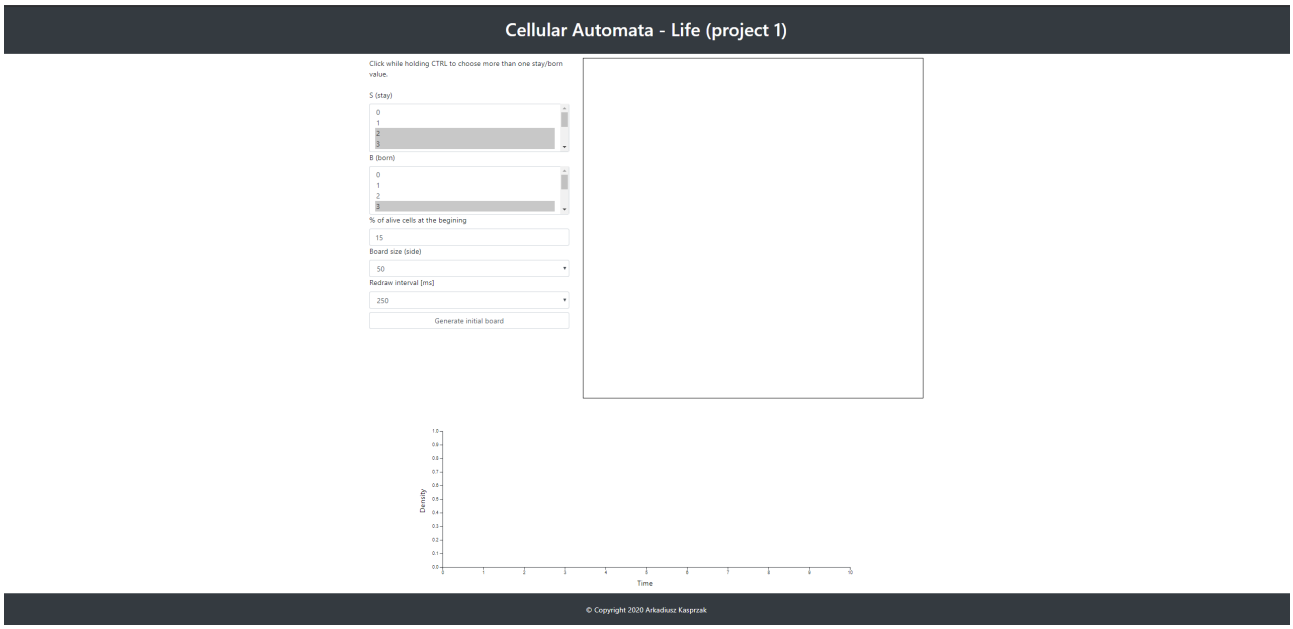
1 Wstęp

Celem projektu było stworzenie aplikacji implementującej dwuwymiarowy automat komórkowy stanowiący rozszerzenie popularnego automatu *Life*. W swojej podstawowej postaci automat ten jest zdefiniowany na dwuwymiarowej sieci kwadratowej, gdzie każda z komórek może znajdować się w jednym z dwóch stanów (żywa/martwa) [1]. Automat ten opiera się na regule 23/3 (jeśli w otoczeniu żywej komórki są 2 lub 3 inne żywe komórki, to pozostaje ona żywa oraz jeśli w otoczeniu martwej komórki znajdują się 3 komórki żywe, to komórka ta staje się żywa). Zarówno w klasycznym automacie *Life*, jak i w prezentowanej w ramach projektu wersji stosuje się otoczenie Moore'a (czyli obejmujące komórki mające z komórką aktualnie przetwarzaną wspólne zarówno krawędzie, jak i wierzchołki). Implementowana w ramach projektu wersja automatu pozwala na stosowanie reguł innych niż 23/3 (dlatego stanowi rozszerzenie oryginalnego automatu - ang. *Life-like*). Projekt stosuje periodyczne warunki brzegowe (sieć nie jest więc, w przeciwieństwie do automatu oryginalnego, nieskończona).

2 Implementacja i opis interfejsu

Projekt zaimplementowany został w formie prostej strony internetowej (klient) z użyciem technologii HTML5, CSS3 oraz języka JavaScript. Wizualizacja działania automatu oparta została o animację wykonaną za pomocą elementu `<canvas>`. W celu szybkiego przygotowania strony wykorzystana została biblioteka Bootstrap. Wykres gęstości wykonany został natomiast z pomocą biblioteki D3 (Data-Driven Documents). Projekt nie wymaga instalowania lub pobierania żadnych dodatkowych zależności - wszystkie potrzebne pliki bibliotek zawarte zostały w katalogu `resources`.

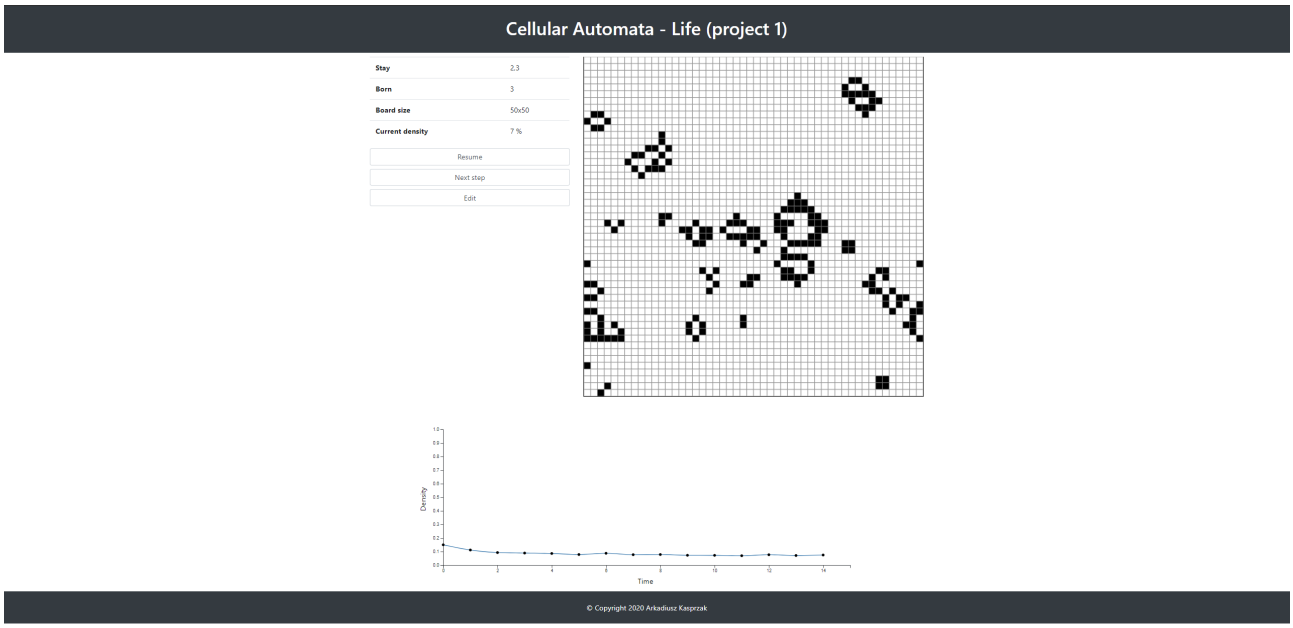
Projekt korzysta z funkcjonalności wprowadzonych w standardzie ECMAScript 6 - do jego działania może więc być konieczna stosunkowo nowa wersja przeglądarki. Projekt został przetestowany na przeglądarkach: Google Chrome (wersja 80.0.3987.132) oraz Firefox (wersja 73.0.1).



Rysunek 1: Interfejs przygotowanej aplikacji zaraz po uruchomieniu. Po lewej widoczne menu konfiguracji automatu, po prawej prostokąt na którym wyświetlana będzie symulacja, w dolnej części natomiast wykres gęstości żywych komórek.

Rysunek 1 stanowi zrzut ekranu przedstawiający aplikację w stanie zaraz po uruchomieniu. Widoczne na nim są trzy podstawowe elementy aplikacji - menu pozwalające na konfigurację działania automatu widoczne - po lewej, prostokąt, w którym wyświetlana będzie animacja - po prawej oraz wykres gęstości żywych komórek w dziedzinie czasu. Użytkownik ma możliwość wyboru reguł (*stay/born*) działania automatu, początkowej gęstości żywych komórek, rozmiaru siatki, na której prowadzona jest symulacja oraz szybkości działania symulacji.

Wybór odpowiedniej konfiguracji przenosi użytkownika do kolejnego ekranu, gdzie możliwe jest wygenerowanie odpowiednich warunków początkowych (poprzez kilkukrotne powtarzanie operacji losowania lub manualną zmianę stanu każdej z komórek poprzez jej naciśnięcie). Po zakończeniu tej czynności użytkownik może rozpocząć symulację. Działającą symulację można w dowolnym momencie zatrzymać a następnie wznowić (rys. 2).



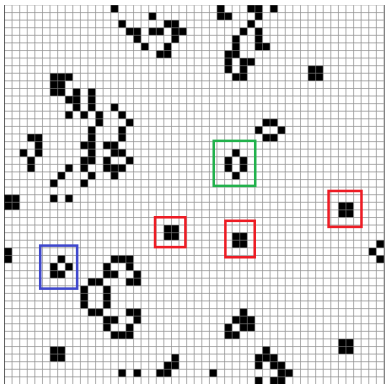
Rysunek 2: Interfejs przygotowanej aplikacji po zatrzymaniu działania symulacji. Widoczne możliwe opcje wznowienia symulacji, wykonania pojedynczego jej kroku oraz powrotu do konfiguracji.

3 Testy działania

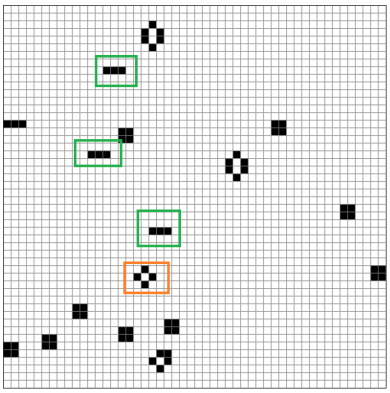
Dowolność w specyfikacji reguły początkowej pozwala na zaobserwowanie wielu interesujących przypadków działania automatu. Domyślnie testy przeprowadzane były na sieci o rozmiarze 50×50 (z wyjątkiem kilku sytuacji, w których inny rozmiar zostanie wyszczególniony w sprawozdaniu). Niniejsze sprawozdanie zawiera opis jedynie kilku wybranych testów przeprowadzonych przez autora - ilość interesujących struktur możliwych do uzyskania za pomocą stworzonego programu jest znacznie większa.

Testy rozpoczęte zostały od klasycznej reguły **23/3**. Wygenerowana została początkowa plansza o gęstości 0.6. Przebieg symulacji zaobserwować można na rysunku 3. Przedstawiony tam został stan symulacji w chwili początkowej - widoczna znaczna gęstość żywych komórek, po około 470 krokach oraz po osiągnięciu stanu stabilnego - po około 650 krokach. Bardzo charakterystyczny jest fakt. iż już w pierwszym kroku symulacji stosunkowo znaczna gęstość 0.6 zredukowana została do około 0.15. Na rysunku 3b zaobserwować można niektóre ze struktur charakterystycznych dla automatu *Life* - blok (oznaczony kolorem czerwonym), kryształ (nazywany też ulem - oznaczony kolorem zielonym) oraz łódź (kolor niebieski). Cechą charakterystyczną tego typu struktur jest ich stabilność (bez zewnętrznej ingerencji pozostają identyczne niezależnie od kroku czasowego).

(a) Krok 0



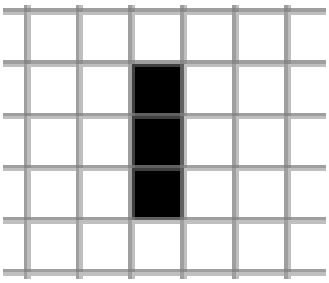
(b) Krok 470



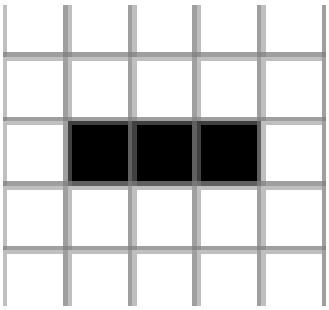
(c) Krok 650

Rysunek 3: Przebieg działania automatu dla reguły 23/3 w trzech wybranych chwilach czasowych.

Na rysunku 3c, przedstawiającym sytuację po ustabilizowaniu się automatu, poza wcześniej wymienionymi zaobserwować można jeszcze dwie inne interesujące struktury: stabilną strukturę koniczyny (kolor pomarańczowy) oraz najprostszy z tzw. oscylatorów - *blinker* (oznaczony kolorem zielonym). Cechą charakterystyczną oscylatorów jest to, iż zmieniają swój stan w sposób okresowy. Rysunek 4 przedstawia proces ewolucji *blinker'a*.



(a) Pierwszy możliwy stan

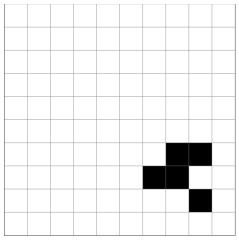


(b) Drugi możliwy stan

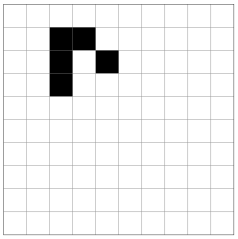
Rysunek 4: Dwa możliwe stany *blinker'a*. Struktura ta w każdym kroku czasowym zmienia swój stan na przeciwny.

Opisane struktury są oczywiście tylko niektórymi z możliwych do zaobserwowania podczas symulacji automatu z regułą 23/3. Podczas przeprowadzania innych symulacji pojawiło się ich znacznie więcej. Część z nich widoczna jest na rysunku 5.

(a) Bochenek - statyczny



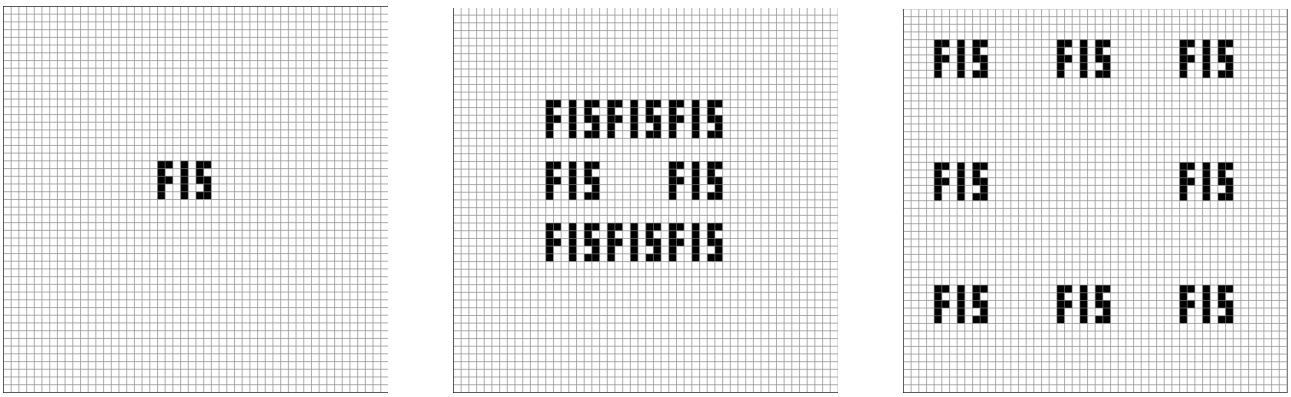
(b) Szybowiec - początek lotu



(c) Szybowiec - po upływie kilku kroków czasowych

Rysunek 5: Pozostałe struktury zaobserwowane podczas wielokrotnych symulacji - statyczny bochenek oraz tzw. szybowiec (ang. *glider*) - struktura okresowa poruszająca się po siatce (tutaj siatka 10×10).

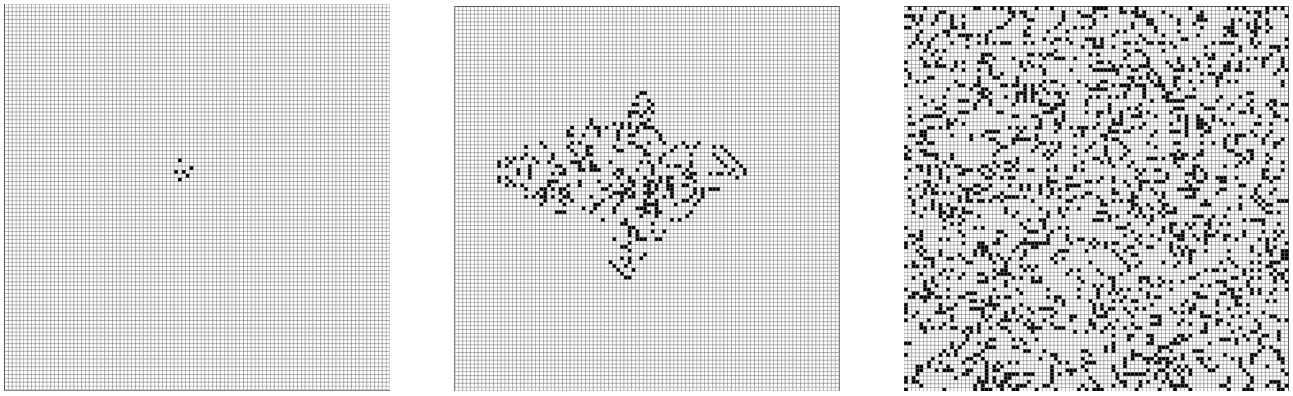
Kolejną testowaną regułą była reguła **1357/1357**. Charakteryzuje się ona pewną szczególną cechą - dowolny początkowy wzór jest, po pewnej liczbie kroków czasowych, powielany (stąd nazwa - replikator). Przykład działania widoczny na rysunku 6.



(a) Krok 0 (b) Krok 8 (c) Krok 16

Rysunek 6: Przykład działania automatu o regule 1357/1357 - w ósmym kroku czasowym początkowy wzór jest powielany osiem razy.

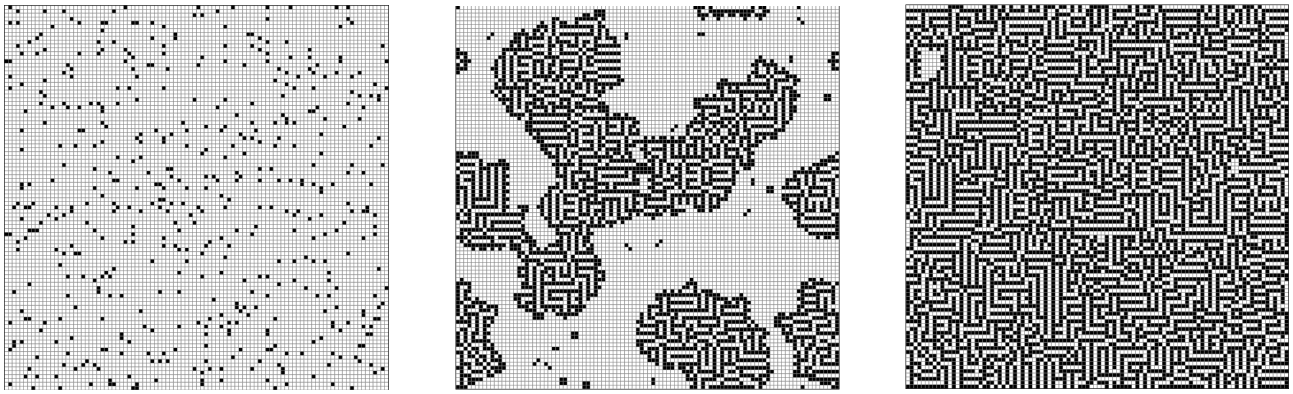
Testom poddana została również reguła **/2** - jest ona interesująca, ponieważ nie zawiera warunku pozwalającego żywym komórkom przeżyć przejścia do kolejnego kroku czasowego - tzn. wszystkie komórki żywe w kroku czasowym t_n umierają przy przejściu do kroku t_{n+1}. Wynik przykładowej symulacji zawarty został na rysunku 7 - w tym wypadku, aby możliwe było dokładne zaobserwowanie działania automatu, symulacja przeprowadzona została na sieci o rozmiarze 100×100. Zaobserwować można znaczny wzrost gęstości żywych komórek - od wartości początkowej poniżej 0.01 do stabilnej wartości około 0.2. Jest to skutkiem niskiej wartości warunku na ożywanie komórki - konieczne są jedynie dwie żywe komórki w otoczeniu. Powstały wzór ma chaotyczną naturę, po pewnym czasie rozprzestrzenia się po całej sieci.



(a) Krok 0 (b) Krok 40 (c) Krok 120

Rysunek 7: Przykład działania automatu o regule /2 - widoczny znaczny wzrost gęstości żywych komórek (od wartości poniżej 0.01 do około 0.2) pomimo braku warunku na przeżycie komórek.

Ostatnią regułą poddaną testom w ramach niniejszego dokumentu była reguła **12345/3**. Wyniki przykładowej symulacji przedstawione zostały na rysunku 8. Symulacja przeprowadzona została na sieci o rozmiarach 100×100. Gęstość początkowa żywych komórek wyniosła w tym przypadku 0.05. W kolejnych krokach symulacji widocznych na rysunku widoczne jest tworzenie się przypominającej labirynt struktury. Wewnątrz tej struktury stany komórek nie ulegają zmianom. Sytuacja stabilizuje się po zajęciu przez strukturę całej siatki - od tego momentu nie następują już żadne dalsze zmiany. Końcowa gęstość żywych komórek miała wartość 0.54.



(a) Krok 0 (b) Krok 31 (c) Krok 130

Rysunek 8: Przykład działania automatu o regule 12345/3. Widoczne tworzenie się przypominającej labirynt struktury. Rys. 8c (krok 130) przedstawia sytuację ustabilizowaną (brak dalszych zmian, gęstość komórek żywych 0.54).

4 Wnioski

- udało się stworzyć poprawnie działającą implementację dwuwymiarowego automatu komórkowego będącego rozszerzeniem klasycznego automatu *Life*
- testy przeprowadzone dla klasycznej konfiguracji 23/3 pozwoliły zaobserwować charakterystyczne struktury, takie jak: blok, bochenek, *blinker* czy szybowiec.
- testy innych konfiguracji pozwoliły zaobserwować charakterystyczne wzorce, takie jak replikator czy labirynt
- dla klasycznego automatu *Life* testy wykazały, że podanie bardzo niskiej (od około 0 do 0.1) lub bardzo wysokiej (od około 0.8 do 1.0) początkowej gęstości żywych komórek prowadzi do bardzo niskich (nawet zerowych) wartości gęstości w kolejnych krokach czasowych.
- klasyczny automat *Life* zwykle dąży do stosunkowo niskich wartości wynikowej gęstości.
- w przypadku reguły 12345/3 utworzenie wzoru labiryntu możliwe jest nawet dla niskich (rzędu 0.03) wartości początkowej gęstości.
- dla gęstości z przedziału od około 0.2 do 0.7 automat o regule 12345/3 ma tendencje do bardzo szybkiego tworzenia stabilnego labiryntu
- wzór powstały w wyniku zastosowania reguły /2 ma tendencję do rozprzestrzeniania się po całej sieci

Literatura

[1] dr hab. inż. Krzysztof Malarz, prof. AGH, *wykład prowadzony w ramach przedmiotu Automaty komórkowe*, http://home.agh.edu.pl/~malarz/dyd/ak/ (dostęp: 17.03.2020)

[2] Wikipedia, the free encyclopedia, *Life-like cellular automaton*, https://en.wikipedia.org/wiki/Life-like_cellular_automaton (dostęp: 17.03.2020)

[3] Wikipedia, the free encyclopedia, *Seeds (cellular automaton)*, https://en.wikipedia.org/wiki/Seeds_(cellular_automaton) (dostęp: 17.03.2020)

[4] Luis Carlos Castillo, *Swing Game of Life*, http://luisccastillo.github.io/Swing-GameOfLife/ (dostęp: 17.03.2020)

[5] d3noob, *Simple d3.js tooltips*, https://bl.ocks.org/d3noob/a22c42db65eb00d4e369 (dostęp: 17.03.2020)

[6] Yan Holtz, *Most basic line chart in d3.js*, https://www.d3-graph-gallery.com/graph/line_basic.html (dostęp: 17.03.2020)

[7] *Dokumentacja biblioteki Bootstrap w wersji 4.4*, https://getbootstrap.com/docs/4.4/getting-started/introduction/ (dostęp: 17.03.2020)