

Uniwersytet Jagielloński w Krakowie
Wydział Fizyki, Astronomii i Informatyki Stosowanej

Łukasz Kostrzewa

Nr albumu: 1080514

Wizualizacja, edycja i przetwarzanie grafów on-line

Praca magisterska
na kierunku Informatyka stosowana

Praca wykonana pod kierunkiem
dr hab. Barbary Strug
Zakład Projektowania i Grafiki Komputerowej

Kraków 2017

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyploma została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Kraków, dnia

Podpis autora pracy

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Kraków, dnia

Podpis kierującego pracą

Spis treści

Wstęp	4
1 Wprowadzenie	5
1.1 Czym są grafy	5
1.2 Znane grafy	5
1.3 Zastosowania grafów	5
2 Wymagania	6
2.1 Tworzenie grafów	6
2.1.1 Importowanie grafów	6
2.1.2 Generowanie grafów	8
2.2 Wizualizacja	9
2.3 Edycja	9
2.4 Przetwarzanie	9
2.5 Eksportowanie	10
3 Istniejące rozwiązania	11
3.1 Aplikacje internetowe	11
3.2 Aplikacje desktopowe	11
4 Projekt i analiza	12
4.1 Biblioteki do wizualizacji grafów w JavaScript	12
4.1.1 Cytoscape.js	12
4.1.2 sigma.js	12
4.1.3 VivaGraph.js	12
4.1.4 Linkurious.js	12
4.2 Interfejs użytkownika	12
4.3 Część serwerowa	12
5 Implementacja	13
6 Testy	14

7 Wnioski	15
A Instrukcje dla użytkowników	16
B Instrukcje dla programistów	17
C Użyte narzędzia	18
Bibliografia	19

Wstęp

„This question is so banal, but seemed to me worthy of attention in that geometry, nor algebra, nor even the art of counting was sufficient to solve it¹”. Tak w 1736 roku pisał Leonhard Euler w liście do Giovanniego Marinoniego, włoskiego matematyka i inżyniera, o jednym z pierwszych problemów w teorii grafów – problemie mostów królewskich. Banalny, ale warty uwagi.

W dzisiejszych czasach teoria grafów rozwiązuje wiele nietrywialnych problemów, a część z nich nadal pozostaje otwarta. Grafy znalazły praktyczne zastosowanie w wielu różnorodnych dziedzinach nauki, takich jak informatyka, ekonomia, socjologia, jak również chemia, lingwistyka, geografia czy nawet architektura. Bez wątpienia teoria grafów jest dziedziną matematyki i informatyki, która zasługuje na uwagę, co postaram się w niniejszej pracy przedstawić.

Głównym celem mojej pracy jest stworzenie aplikacji służącej do wizualizacji i edycji grafów w przeglądarce. W przeciągu kilku ostatnich lat mogliśmy zaobserwować gwałtowny wzrost znaczenia aplikacji internetowych. Co dziwne, na dzień dzisiejszy w sieci praktycznie nie ma rozwiązania, które pozwalałoby wczytać graf, wyświetlić, w łatwy sposób przetworzyć, a następnie wyeksportować do znanego formatu. Praca ta jest odpowiedzią na ów deficyt.

W pracy dokonam również przeglądu i analizy bibliotek JavaScript oraz technologii służących do wizualizacji grafów w przeglądarce.

¹Cytat zaczerpnięty z [HW04], wyróżnienie własne.

Rozdział 1

Wprowadzenie

1.1 Czym są grafy

1.2 Znane grafy

1.3 Zastosowania grafów

Rozdział 2

Wymagania

Rozdział ten zawiera wszystkie wymagania funkcjonalne, które powinna spełniać aplikacja, aby praca z grafami była możliwie przystępna.

2.1 Tworzenie grafów

Podstawowym i oczywistym wymaganiem jest, aby użytkownik mógł stworzyć nowy, pusty graf. Ponadto użytkownik powinien mieć możliwość zaimportowania istniejącego grafu oraz wygenerowania znanego grafu, np. cyklu lub grafu pełnego o zadanej ilości wierzchołków.

2.1.1 Importowanie grafów

Istnieje wiele formatów służących do opisu grafów. Do najpopularniejszych należą [MB04; Gep]

- GraphML – *Graph Markup Language*
- GEXF – *Graph Exchange XML Format*
- JGF – *JSON Graph Format*
- DOT – format programu Graphviz
- GML – *Graph Modeling Language*
- DGML – *Directed Graph Markup Language*
- XGMML – *eXtensible Graph Markup and Modeling Language*

Użytkownik powinien móc wczytać graf w formatach GraphML, GEXF oraz JGF.

Graph Markup Language (GraphML)

Listing 2.1: Przykład grafu w formacie GraphML

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
    http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <graph id="G" edgedefault="undirected">
    <node id="1"/>
    <node id="2"/>
    <edge source="1" target="2"/>
  </graph>
</graphml>
```

Graph Exchange XML Format (GEXF)

Listing 2.2: Przykład grafu w formacie GEXF

```
<?xml version="1.0" encoding="UTF-8"?>
<gexf xmlns="http://www.gexf.net/1.2draft" version="1.2">
  <graph mode="static" defaultedgetype="directed">
    <nodes>
      <node id="0" label="Hello" />
      <node id="1" label="Word" />
    </nodes>
    <edges>
      <edge id="0" source="0" target="1" />
    </edges>
  </graph>
</gexf>
```


JSON Graph Format (JGF)

Listing 2.3: Przykład grafu w formacie JGF

```
{
  "graph": {
    "nodes": [{
      "id": "A",
    },
    {
      "id": "B",
    }
  ],
  "edges": [{
    "source": "A",
    "target": "B"
  }]
}
```

DOT Graphviz

Listing 2.4: Przykład grafu w formacie DOT

```
graph graphname {
  a -- b -- c;
  b -- d;
}
```

2.1.2 Generowanie grafów

Użytkownik powinien mieć możliwość wygenerowania znanych grafów, dla zadanych parametrów wejściowych:

- Graf pusty
- Graf liniowy

- Graf cykliczny
- Koło
- Graf pełny (lub turniej)
- Graf pełny dwudzielny
- Graf Petersena
- Drzewa

Definicje i przykłady powyższych grafów znajdują się w sekcji 1.2. Ponadto przydatnym dodatkiem w aplikacji będzie możliwość wygenerowania grafu losowego – o danej ilości wierzchołków oraz parametrem prawdopodobieństwa określającym, czy pomiędzy dwoma wierzchołkami istnieje krawędź.

2.2 Wizualizacja

oddalanie, przybliżanie
 layouts (grid, circle, concentric, bfs)
 samodzielne ustawianie wierzchołków i force layout
 różne typy wierzchołków / kolory / ikony w wierzchołkach
 style / kolor krawędzi
 wyszukiwanie po danych

2.3 Edycja

osobny tryb edycji
 dodawanie/usuwanie wierzchołków/krawędzi
 dodawanie etykiet / własności
 grupowanie wierzchołków

2.4 Przetwarzanie

Podstawowe algorytmy:

- Wyszukiwanie najkrótszej ścieżki
- Minimalne drzewo rozpinające
- Page rank
- Spójne składowe
- Cykl Eulera
- Cykl Hamiltona

2.5 Eksportowanie

Użytkownik powinien mieć możliwość wyeksportowania do formatów, które zostały przedstawione w podsekcji 2.1.1. Ponadto przydatną funkcjonalnością będzie możliwość wyeksportowania obecnego widoku do pliku graficznego, np. PNG lub JPG.

Rozdział 3

Istniejące rozwiązania

3.1 Aplikacje internetowe

Graph Creator – Illuminations

<http://illuminations.nctm.org/Activity.aspx?id=3550>

<http://graphonline.ru/en/>

<https://dl.dropboxusercontent.com/u/4189520/GraphJS/graphjs.html>

<https://yiboyang.github.io/graphrel/>

<https://visualgo.net/en/dfsdfs>

<http://crunchbase.linkurio.us/demo/>

3.2 Aplikacje desktopowe

Rozdział 4

Projekt i analiza

4.1 Biblioteki do wizualizacji grafów w JavaScript

	Cytoscape.js	Sigma	VivaGraphJS
Licencja	MIT	MIT	BSD 3
Rozmiar	294	112,9	60,4
Renderowanie			
SVG	•	tak	•
HTML5 Canvas	•	tak	•
WebGL Canvas	•	tak	•
Obsługiwane formaty	•	•	•
Rozszerzalność	•	•	•
•	•	•	•

4.1.1 Cytoscape.js

4.1.2 sigma.js

4.1.3 VivaGraph.js

4.1.4 Linkurious.js

4.2 Interfejs użytkownika

4.3 Część serwerowa

Rozdział 5

Implementacja

Rozdział 6

Testy

Rozdział 7

Wnioski

Dodatek A

Instrukcje dla użytkowników

Dodatek B

Instrukcje dla programistów

Dodatek C

Użyte narzędzia

Bibliografia

- [HW04] Brian Hopkins i Robin Wilson. „*The Truth about Königsberg*”. W: *College Mathematics Journal* 35 (maj 2004), s. 198–207. URL: https://www.maa.org/sites/default/files/pdf/upload_library/22/Polya/hopkins.pdf (term. wiz. 29.04.2017).
- [MB04] S. Mohammed i M. Bernard. *Graph File Formats*. Spraw. tech. Mona, Kingston, Jamajka: Department of Mathematics and Computer Science, The University of the West Indies, 2004. URL: http://www2.sta.uwi.edu/~mbernard/research_files/fileformats.pdf (term. wiz. 29.04.2017).
- [Gep] Gephi. *Supported Graph Formats*. The Gephi Consortium. URL: <https://gephi.org/users/supported-graph-formats/> (term. wiz. 29.04.2017).