

**Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie**

---

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

KATEDRA AUTOMATYKI



**PRACA MAGISTERSKA**

**ŁUKASZ ZIEŃKOWSKI**

**INTERAKTYWNA MAPA CZASU Z DODATKOWĄ OSIĄ CZASU**

PROMOTOR:  
dr inż. Grzegorz Rogus

Kraków 2013

## **OŚWIADCZENIE AUTORA PRACY**

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMIENIONE W PRACY.

.....

PODPIS

**AGH**  
**University of Science and Technology in Krakow**

---

Faculty of Electrical Engineering, Automatics, Computer Science and Electronics

DEPARTMENT OF AUTOMATICS



**MASTER OF SCIENCE THESIS**

**ŁUKASZ ZIEŃKOWSKI**

**THESIS IN L<sup>A</sup>T<sub>E</sub>X**

SUPERVISOR:  
Grzegorz Rogus Ph.D

Krakow 2013

Serdecznie dziękuję ... tu ciąg dalszych  
podziękowań np. dla promotora, żony,  
sąsiada itp.

## Spis treści

<b>1. Wprowadzenie</b>	6
1.1. Cele pracy	6
1.2. Zawartość pracy	6
<b>2. State of art</b>	7
2.1. Rodzaje map	7
2.1.1. Google Maps	7
2.1.2. Windows Maps	7
2.1.3. Yahoo Maps	8
2.1.4. Apple Maps	8
2.2. Google Earth	8
2.3. Time line	10
2.4. Kompilacja	10
2.5. Narzędzia	11
2.6. Przygotowanie dokumentu	11
<b>3. Definicja problemu</b>	12
3.1. Co potrzeba	12
3.2. Problemy	12
<b>4. Opis rozwiązania</b>	14
4.1. Transmisja danych	14
4.1.1. XML	14
4.1.2. JSON	15
4.2. Interferjs	16
<b>5. Podsumowanie</b>	17
<b>6. Podsumowanie</b>	18

# 1. Wprowadzenie

2-3 storny

cytat ([NN10])

wyraz  $\tau$ ,  $\epsilon$ ,  $\chi$

W rozdziale 2

obraz 2.6

95%

w pliku `test.tex`.

Na stronie <http://kile.sourceforge.net/screenshots.php>

## 1.1. Cele pracy

## 1.2. Zawartość pracy

## 2. State of art

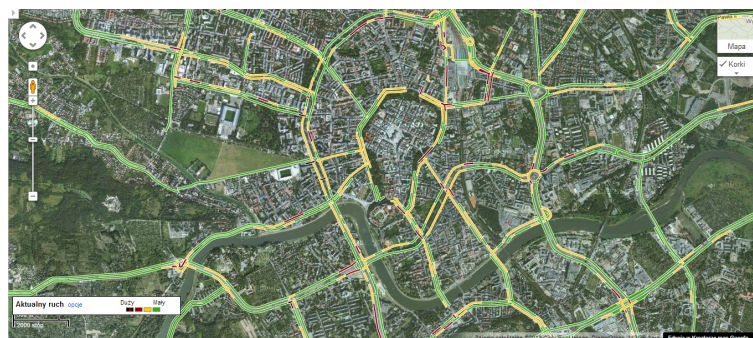
20-30 W rozdziale tym przedstawiono podstawowe informacje dotyczące struktury prostych plików  $\text{\LaTeX}$ a. Omówiono również metody kompilacji plików z zastosowaniem programów *latex* oraz *pdflatex*.

### 2.1. Rodzaje map

Tworząc aplikację która ma dostarczać informacji korzystających z map należy zapoznać się dostępnymi źródłami. Z powodu szrokiego wyboru poniżej omówione zostaną jedynie aplikacje które dostarczają informacji ogólnoświatowych. Na polskim rynku dostępnych jest kilka rozwiązań, ich główną wadą jest ograniczenie do terytorium Polski, dodatkowo często nie dostarczają one obrazów satelitarnych, są to m.in. <http://zumi.pl>

#### 2.1.1. Google Maps

2.1



Rysunek 2.1: Google Maps.

#### 2.1.2. Windows Maps

<http://www.flashearth.com/>

nie wszystko

2.2

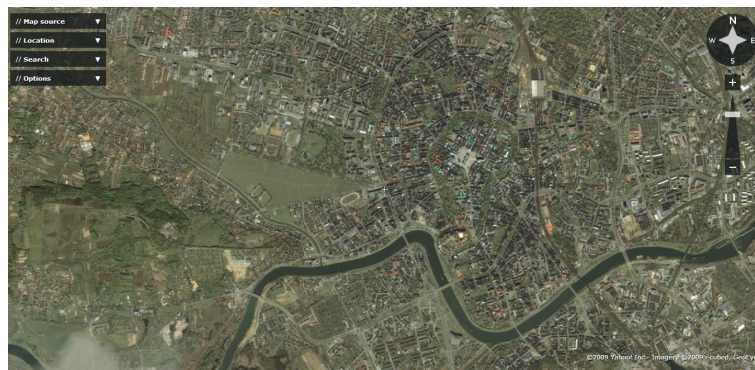


Rysunek 2.2: Bing Maps.

### 2.1.3. Yahoo Maps

2.3

tylko granice krakowa



Rysunek 2.3: Yahoo Maps.

### 2.1.4. Apple Maps

Kolejną dużą marką która dostarcza informację jest Apple. Niestety nie ma wersji która pozwalałaby na dostęp do tej usługi z powszechnie używanych komputerów stacjonarnych.

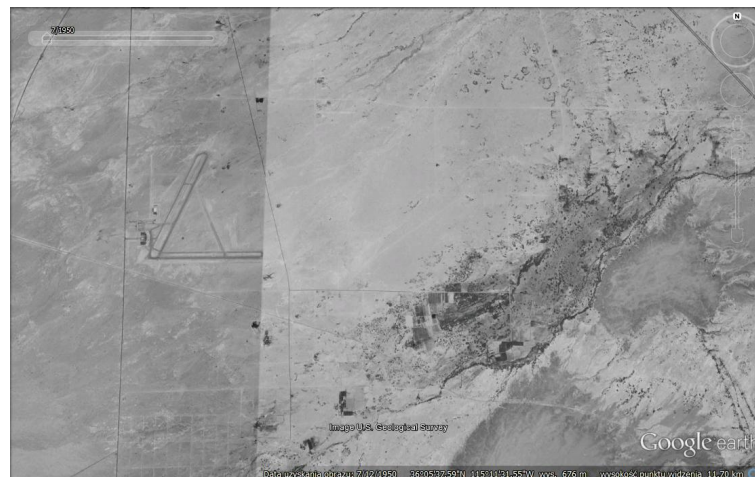
## 2.2. Google Earth

Ciekawe wykorzystanie obrazów satelitarnych i wskaźnika czasu zostało zaprezentowane w programie Google Earth. W aplikacji tej możemy zobaczyć nie tylko najbardziej aktualne zdjęcia, ale jesteśmy w stanie cofnąć się w czasie i zobaczyć jak wyglądał obszar na który patrzymy w przeszłości.

Przykład takiej sytuacji został przedstawiony na rysunku 2.4, obraz terenu na którym powstanie miasto Las Vegas w roku 1950. Jak teren ten wyglądał w roku 1977 widzimy na rysunku 2.5, pomimo widocznych zmian teren ten nadal w dużym stopniu jest pusty, dopiero na rysunku 2.6 widzimy aktualny stan miasta.



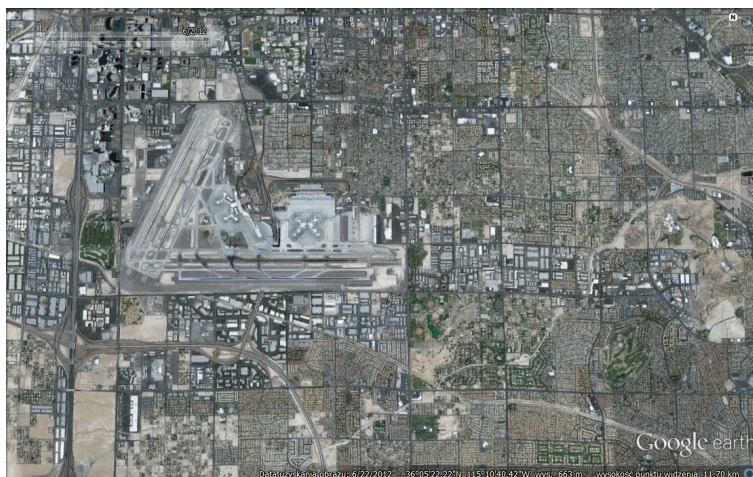
Dzięki funkcji zmiany punktu i kąta patrzenia, pokazywania ciekawych miejsc czy chociażby włączania trybu w którym budynki nabierają formy przestrzennej, 3D, możemy poprzez zabawę i wirtualne wycieczki poszerzać naszą wiedzę o otaczającym nas świecie.



Rysunek 2.4: Las Vegas w 1950 roku.



Rysunek 2.5: Las Vegas w 1950 roku.



Rysunek 2.6: Las Vegas w 1950 roku.

## 2.3. Time line

Plik  $\text{\LaTeX}$ owy jest plikiem tekstowym, który oprócz tekstu zawiera polecenia formatujące ten tekst (analogicznie do języka HTML). Plik składa się z dwóch części:

1. Preambuły – określającej klasę dokumentu oraz zawierającej m.in. polecenia dołączającej dodatkowe pakiety;
2. Części głównej – zawierającej zasadniczą treść dokumentu.

```

1 \documentclass[a4paper,12pt]{article} % preambuła
2 \usepackage[polish]{babel}
3 \usepackage[utf8]{inputenc}
4 \usepackage[T1]{fontenc}
5 \usepackage{times}
6
7 \begin{document} % część główna
8
9 \section{Sztuczne życie}
10
11 % treść
12 % ąśęźźćńłóĘŚĄŻŻĆŃÓŁ
13
14 \end{document}

```

## 2.4. Kompilacja

```

1 latex test.tex
2 dvips test.dvi -o test.ps
3 ps2pdf test.ps

```

lub za pomocą PDF $\LaTeX$ :

```
1 pdflatex test.tex
```

## 2.5. Narzędzia

- Edit shortcuts – definiowanie własnych klawiszy skrótów;
- Line Tools – dodatkowe operacje na liniach tekstu;

## 2.6. Przygotowanie dokumentu

Plik źródłowy  $\LaTeX$ a jest zwykłym plikiem tekstowym. Przygotowując plik źródłowy warto wiedzieć o kilku szczegółach:

- Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja. Aby uzyskać *twardą spację*, zamiast znaku spacji należy użyć znaku *tyldy*.
- Znakiem końca akapitu jest pusta linia (ilość pustych linii nie ma znaczenia), a nie znaki przejścia do nowej linii.
- $\LaTeX$  sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.

## **3. Definicja problemu**

### **3.1. Co potrzeba**

Aby móc w pełni pracować potrzeba

- Stworzyć interfejs
- Zapewnić szybkie działanie
- Przeprowadzić testy
- Stworzyć interfejs
- Zapewnić szybkie działanie
- Przeprowadzić testy
- Stworzyć interfejs
- Zapewnić szybkie działanie
- Przeprowadzić testy

### **3.2. Problemy**

Przewidywane problemy

- Stworzyć interfejs
- Zapewnić szybkie działanie
- Przeprowadzić testy
- Stworzyć interfejs
- Zapewnić szybkie działanie
- Przeprowadzić testy
- Stworzyć interfejs

- Zapewnić szybkie działanie
- Przeprowadzić testy

## 4. Opis rozwiązania

50

### 4.1. Transmisja danych

xml - 729 557 json - 695 400 [NN10] [BL12]

#### 4.1.1. XML

```
1 <?xml version="1.0" ?>
2 <map>
3   <title>Tove</title>
4   <description>Jani</description>
5   <from>123444</from>
6   <to>123444</to>
7   <markers>
8     <marker>
9       <x>123.344</x>
10      <y>123.344</y>
11      <z>123.344</z>
12      <from>123444</from>
13      <to>123444</to>
14      <description>Jani</description>
15      <icon>Jani</icon>
16    </marker>
17    <marker>
18      <x>123.344</x>
19      <y>123.344</y>
20      <z>123.344</z>
21      <from>123444</from>
22      <to>123444</to>
23      <description>Jani</description>
24      <icon>Jani</icon>
25    </marker>
26    <marker>
27      <x>123.344</x>
28      <y>123.344</y>
29      <z>123.344</z>
```

```
30 <from>123444</from>
31 <to>123444</to>
32 <description>Jani</description>
33 <icon>Jani</icon>
34 </marker>
35 </markers>
36 </map>
```

Listing 4.1: caption

### 4.1.2. JSON

```
1 {
2   "title": "Tove",
3   "description": "Jani",
4   "from": 123444,
5   "to": 123444,
6   "markers": [
7     {
8       "x": "123.344",
9       "y": "123.344",
10      "z": "123.344",
11      "from": "123444",
12      "to": "123444",
13      "description": "Jani"
14      "icon": "Jani"
15    },
16    {
17      "x": "123.344",
18      "y": "123.344",
19      "z": "123.344",
20      "from": "123444",
21      "to": "123444",
22      "description": "Jani"
23      "icon": "Jani"
24    },
25    {
26      "x": "123.344",
27      "y": "123.344",
28      "z": "123.344",
29      "from": "123444",
30      "to": "123444",
31      "description": "Jani"
32      "icon": "Jani"
33    },
34  ]
35 }
```

Listing 4.2: json

## 4.2. Interferjs



## **5. Podsumowanie**

10

## **6. Podsumowanie**

2-3 strony

## Bibliografia

- [BL12] Xu Chen Yingying Yu Boci Lin, Yan Chen. Comparision between json and xml in applications on ajax. *CrossTalk*, 2012.
- [NN10] Reynolds R. Izurieta C. Nurseitov N., Paulson M. Comparison of json and xml data interchange formats: A case study. *CrossTalk*, 2010.