

**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^AT_EX

SUPERVISOR:
Grzegorz Rogus Ph.D

Krakow 2013

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

Spis treści

1. Wprowadzenie	6
1.1. Cele pracy	6
1.2. Zawartość pracy	6
2. Rys historyczny	7
3. State of art	8
3.1. Rodzaje map	8
3.1.1. Google Maps	8
3.1.2. Windows Maps	8
3.1.3. Yahoo Maps	9
3.1.4. Apple Maps	9
3.2. Google Earth	9
3.3. Time line	11
4. Definicja problemu	12
4.1. Co potrzeba	12
4.2. Problemy	12
5. Opis rozwiązania	13
5.1. Transmisja danych	13
5.2. Możliwości HTML5	13
5.2.1. Wykorzystanie HTML5	15
5.2.2. Storage	15
5.2.3. Web SQL Database	17
5.2.4. IndexedDB	18
5.3. Oś czasu	18
5.4. Przybliżanie wyników	19
5.5. Parser kml	19
5.6. Interferjs	19
6. Przykład użycia	20
7. Podsumowanie	21

1. Wprowadzenie

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

2-3 storny

wyraz τ , ϵ , χ

W rozdziale 3

obraz 3.6

w pliku `test.tex`.

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

1.1. Cele pracy

1.2. Zawartość pracy

2. Rys historyczny

Pomimo że w obecnych czasach, mapy nie są niczym niezwykłym, i wydają się codziennością nie zawsze tak było. W historii człowieka można znaleźć okresy czasu kiedy kartografia była nieznana, lub była bardzo niedokładna.

Pierwsze malowidła które można uznać za graficzną reprezentację otoczenia archeolodzy napotkali w okolicach Pavloc (Czechy), datowaną są one na 25 wiek przed naszą erą [Lam09] . Na 14 wiek p.n.e datowane są wykopaliska w Navarre(Hiszpania) obejmujące rysunki na piaskowcu [Cho09].

Jeszcze w XIV kartografia była bardzo ograniczona, mapy którymi posługiwano się nie były zbyt dokładne. 12 października 1410 gdy Krzysztof Kolumb dotarł do brzegów wyspy San Salvador uznając ją za jedną z wysp japońskich [Irv28]. Aby lepiej zrozumieć powód tej pomyłki wystarczy spojrzeć na mapę stworzoną w XV wieku przez kartografa Henricusa Martellusa 2.1. Widzimy na niej że m.in. obie ameryki nie były znane uwczesnym ludzior.



Rysunek 2.1: Mapa świata z 1489 roku.

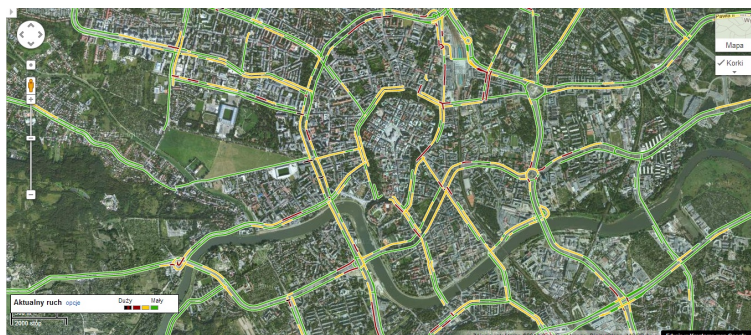
3. State of art

3.1. 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>

3.1.1. Google Maps

Rysunek 3.1 przedstawia obraz otrzymany w aplikacji Google Maps. Dodatkowo włączona opcja prezentacji natężenia ruchu jedynie potwierdza duże możliwości i łatwość obsługi. Przyjazny interfejs sprawia że praca jest prosta i pozwala na osiągnięcie bardzo dobrych wyników.



Rysunek 3.1: Google Maps.

3.1.2. Windows Maps

Rysunek 3.2 przedstawia obraz otrzymany wykorzystując Bing Maps. W tym konkretnym przykładzie widzimy znaczną różnicę kolorów, obecność chmur pomniejsza wartość tych zdjęć. Należy wspomnieć o znacznie uboższym interfejsie dostarczany użytkownikowi, interakcja jest w znacznym stopniu uboższa.

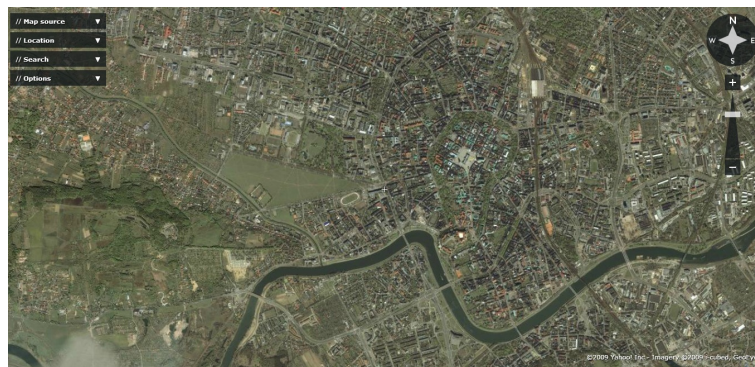


Rysunek 3.2: Bing Maps.

3.1.3. Yahoo Maps

Kolejnym dostarczycielem danych kartograficznych jest Yahoo, przykład znajduje się na rysunku 3.3. Interfejs jest zbliżony do Bing Maps, jednak obszar na którym możemy przedlądać zdjęcia jest mniejszy, obszary oddalone od większych miast nie są w pełni uwzglęnione.

tylko granice krakowa



Rysunek 3.3: Yahoo Maps.

3.1.4. 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.

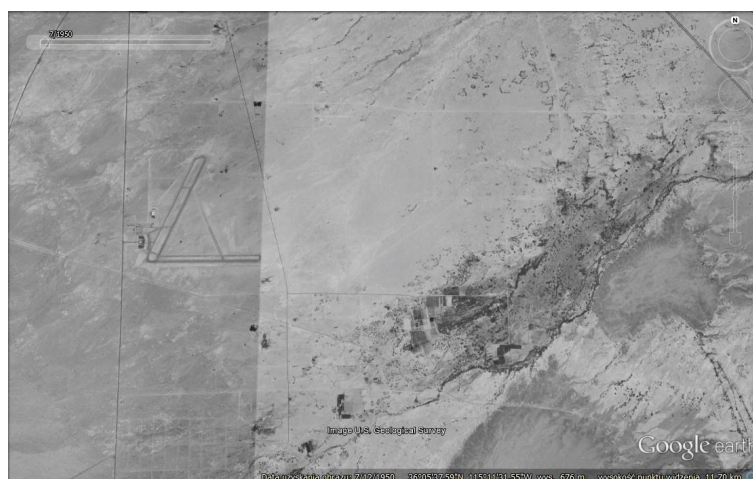
3.2. Google Earth

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 3.4, obraz terenu na którym powstanie miasto Las Vegas w roku 1950. Jak teren ten wyglądał w roku 1977 widzimy na rysunku 3.5, pomimo

widocznych zmian teren ten nadal w dużym stopniu jest pusty, dopiero na rysunku 3.6 widzimy aktualny stan miasta.

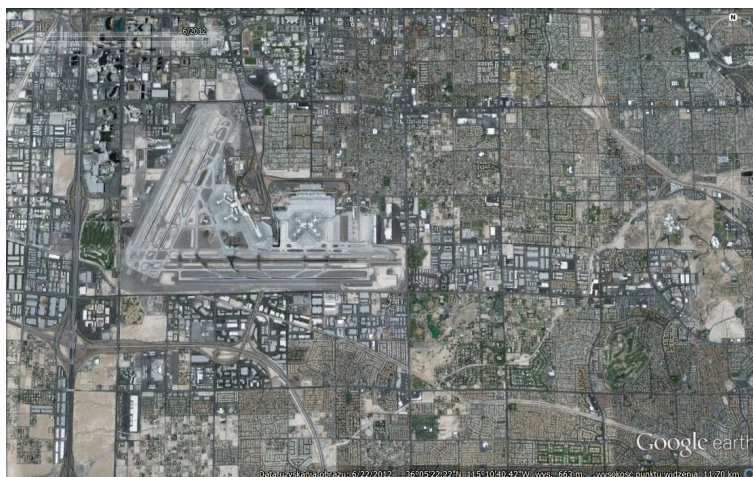
Dzięki funkcji zmiany punktu i kąta patrzenia, pokazywania ciekawych miejsc czy chociażby włączania trybu w którym budynki nabierają formy przestrzennej, 3D, możemy poprzez zabawę i wirtualne wycieczki poszerzać naszą wiedzę o otaczającym nas świecie.



Rysunek 3.4: Las Vegas w 1950 roku.



Rysunek 3.5: Las Vegas w 1950 roku.



Rysunek 3.6: Las Vegas w 1950 roku.

3.3. Time line

Jedną z większych uniedogonień podczas korzystania z map jest ich statyczność. Tradycyjne mapy pozwalają poznać jedynie jedną konkretną sytuację która panowała na danym obszarze, nawet jeśli nałożymy kilka informacji z różnych okresów na jeden obraz, wynik może przestać być czytelny. Korzystając z map dostępnych w internecie zazwyczaj istnieje opcja zmiany widoku. W jednej chwili możemy być na wysokości kilkuset kilometrów nad poziomem morza, aby w kolejnej znaleźć się kilkaset metrów nad ziemią.

Ciekawą opcją jest dostarczana przez Google Earth możliwość zmiany czasu wykonania zdjęć. Niestety jest ona nieprzydatna jeśli chcemy prezentować własne dane.

4. Definicja problemu

4.1. Co potrzeba

Aby móc w pełni pracować potrzeba

- Zapewnić szybkie działanie
- Przeprowadzić testy
-

4.2. Problemy

Przewidywane problemy

- Stworzyć interfejs

5. Opis rozwiązania

W poniższym rozdziale omówione zostaną kroki pracy.

5.1. Transmisja danych

Pierwszym aspektem który należy rozwiązać jest sposób przesyłania danych. Problem ten jest szczególnie istotny w omawianej pracy z uwagi na możliwość przesyłania informacji o granica lub innych liniach prezentowanych na mapie. Do opisu kwadratowego obszaru wymagane jest przesłanie informacji o 4 punktach. Jeżeli będziemy chcieli przekazać dokładniejszy zarys obszaru, zaprezentować granicę państwa lub linię frontu wojennego linia prosta w większości przypadków będzie zbyt ogólnym przybliżeniem, nie oddającym prawdziwej sytuacji.

Z raportu Akamai wynika że średnia przepływność łączy internetowych dla użytkowników korzystających z puli adresów IP przeznaczonych dla Polski w I kwartale 2012 r. wynosiła 5Mb/s <http://www.rp.pl/artykul/924483.html> (dostęp 13.04.2014). Jest to bardzo dobry wynik który plasuje Polskę w czołówce rankingu. Pomimo tego nie można pominąć faktu optymalizacji zapytań i danych przesyłanych, wymieniane dane pomiędzy użytkownikiem a serwerem powinny być jak najmniejsze. Duża popularność urządzeń mobilnych w których dostęp do internetu jest zapewniany często poprzez sieć bezprzewodową a dostęp do internetu nie jest jeszcze tak dogodny jak jest to w przypadku użytkowników stacjonarnych wymusza optymalizację.

Kolejnym powodem dla którego odpowiedzi serwera powinny być jak najlżejsze jest koszt pracy samego serwera. Jest to szczególnie widoczne w dużych aplikacjach mających wiele użytkowników, czas jaki jest przeznaczany dla pojedynczego użytkownika jest mnożony przez ich ilość. Z tego powodu zawsze podczas zwiększania ilości użytkowników korzystających z aplikacji następuje czas w którym należy zacząć korzystać z dodatkowego serwera. Celem programisty tworzącego kod który będzie wykorzystywał zasoby serwera (zarówno czas jak i pamięć) jest dbanie aby moment w którym niezbędne będzie korzystanie z większej ilości maszyn nastąpił przy jak największej ilości użytkowników.

xml - 729 557 json - 695 400

5.2. Możliwości HTML5

HyperText Markup Language, hipertekstowy język znaczników Pozwala na opisanie struktury informacji zawartych na stronie internetowej, to dzięki niemu przeglądarka może rozróżnić takie elementy

jak hiperłącze, akapit czy chociażby nagłówek.

Podobnie jak w przypadku XML, tak i tutaj wymagane jest aby wykorzystywane znaczniki umieszczane były w nawiasach ostrokatnych, a każdy z nich miał swoje domknięcie.

Poprawnym zapisem jest `<p>Wiadomość</p>` który oznacza pojedynczy akapit. Zapis `<p>Wiadomość<p>`, który różni się od poprzedniego brakiem znaku w drugim znaczniku, czyni to ten zapis niepoprawnym. Istnieje możliwość aby wykorzystać pojedynczy znacznik przykładem jest `
` określający nową linię która będzie widoczna na stronie.

Obecnie używany standart HTML4 ma niestety wiele ograniczeń, z tego powodu pracowano nad jego następnikiem. 22 stycznia 2008 W3C opublikował HTML5, wtedy jeszcze jako jedynie szkic.

Bardzo ciekawym i wartym zainteresowania dodanym elementem w nowej wersji jest obecność znaczniku canvas. Pozwala on na dynamiczne, skryptowe renderowanie kształtów i obrazów. Dzięki temu obiektowi możliwe stało się tworzenie animacji czy nawet gier działających w przeglądarce bez konieczności używania dodatkowych wtyczek czy programów.

<http://techtrendy.pl/title,Pierwsza-gra-3D-napisana-w-HTML5,wid,14102779,wia>

Przykładem wielkich możliwości jakie dostarcza udoskonalony język jest fakt iż już w roku 2011 powstała pierwsza trójwymiarowa gra stworzona w całości przy użyciu HTML5. Przykład grafiki widoczny jest na rysunku 5.1. Wcześniej takie efekty możliwe były jedynie przy wykorzystaniu technologii Flash, jej wadą była drudność edycji gotowego produktu, wyganało to specjalnego oprogramowania. Fakt tworzenia pliku wykonywalnego którego nie było możliwości edycji wymuszał dostęp do kodów źródłowych, czyli plików które tworzył autor programu.



Rysunek 5.1: Pierwsza gra 3D w html5.

Konkretną funkcjonalnością która wydaje się być przydatna w omawianym projekcie jest możliwość rysowania kształtów geometrycznych, również na innych obrazach. Przykładem wykorzystania tej technologii jest 5.2 na którym przy pomocy okręgu zaznaczono rynek główny w Krakowie i jego okolice, możliwość zmiany przejrzystości narysowanego kształtu pozwala aby obraz pod nim był nadal widoczny.



Rysunek 5.2: P.

```

1
2   var canvas = document.getElementById('myCanvas');
3   var context = canvas.getContext('2d');
4   var imageObj = new Image();
5
6   imageObj.onload = function() {
7       context.drawImage(imageObj, 69, 50);
8   context.beginPath();
9       context.arc(canvas.width / 2, canvas.height / 2, 90, 0, 2 * Math.PI, false);
10      context.fillStyle = "rgba(255, 0, 0, 0.5)";
11      context.fill();
12      context.stroke();
13  };
14  imageObj.src = './gm_1.jpg';

```

Listing 5.1: json

5.2.1. Wykorzystanie HTML5

Wstępne założenia zakładały wykorzystanie możliwości rysowania kształtów przy użyciu ogólnodostępnych map. Rozważania w sekcji 3.1 wskazały że najlepszym wyborem są Google Maps. Niestety opcja rysowania kształtów która dostarcza HTML5 wymaga użycia znacznika canvas, nie jest to kompatybilne z metodą wykorzystywania maps dostarczanych od omawianej firmy która wymaga wykorzystanie znacznika div. W momencie tworzenia tej pracy nie ma możliwości współpracy wybranych map z technologią rysowania kształtów geometrycznych. Sytuacja taka wymusza wykorzystanie innego rozwiązania do tworzenia interesujących nas warstw.

5.2.2. Storage

Inną ciekawą funkcją omówioną w podręczniku do HTML5 [Pil10] jest Storage. Pozwala on na przechowywanie danych w przeglądarce użytkownika. Różnicą w stosunku do ciasteczek które również potrafią przechowywać informacje o konkretnym użytkowniku jest:

- Większy rozmiar dostępnej pamięci m.in. Chrome 5MB , IE 10

- Informacje przechowywane są po stronie użytkownika, nie są przesyłane za każdym razem do serwera.
- Informacja może być przechowywana przez długi okres czasu.

Dodatkowo nie można pominąć faktu istnienia dwóch rodzajów tej pamięci.

- Session Storage Dane przechowywane są w kontekście sesji użytkownika, są one tracone w momencie zamknięcia okna przeglądarki.
- Local Storage Teoretycznie dane są przechowywane w nieskończoność, do momentu kiedy użytkownik nie usunie ich. Zamknięcie sesji nie powoduje czyszczenia danych.

Powodem który wymaga wykorzystania tego typu pamięci jest możliwość przechowywania danych nad którymi użytkownik pracuje w aktualnym czasie. Nie potrzebuje on informacji które były dla niego istotne podczas poprzedniej wizyty, sesji. Sytuacja ta jednoznacznie wskazuje że lepszym wyborem jest wybór pamięci sesyjnej (Session Storage).

Wadą tej pamięci jest jej interfejs. Obecnie przechowywany sposób danych to mapowanie w postaci napis->napis. Wymusza to aby każde dane które chcemy przechować muszą być w formie ciągu znaków. Przykład 5.2.2 przedstawia w jaki sposób możemy obiekt zawierający imię i nazwisko zapisać w pamięci. Linia 4 przedstawia obiekt w postaci której chcielibyśmy go przechować. Niestety zwykle przypisanie do zmiennej w pamięci powoduje że jedynie typ instancji zostaje zapisany. Aby móc zapisać w poprawnej formie dane musimy dokonać serializacji danych. Czynność tą możemy wykonać przy pomocy metody `stringify` z obiektu `JSON`, wynikiem jest ciąg znaków który możemy bez problemu zapisać w pamięci sesyjnej. Do odzyskania pierwotnego obiektu, odtworzenia go z zapisanego napisu wykorzystujemy metodę `parse` również z obiektu `JSON`.

```
1      uzytkownik={};
2      uzytkownik.imie='Jan'
3      uzytkownik.nazwisko='Kowalski'
4      //uzytkownik : Object {imie: "Jan", nazwisko: "Kowalski"}
5
6      sessionStorage.u1 = uzytkownik
7      //sessionStorage.u1 : "[object Object]"
8
9      sessionStorage.u2 = JSON.stringify(uzytkownik)
10     //sessionStorage.u2 : '{"imie":"Jan","nazwisko":"Kowalski"}'
11
12     uzytkownik2 = JSON.parse(sessionStorage.u2)
13     //uzytkownik2 : Object {imie: "Jan", nazwisko: "Kowalski"}
```

Listing 5.2: json

Wsparcie dla pamięci Storage nie jest obecne zazwyczaj w nowszych wersach przeglądarek. Na stronie <http://www.html5rocks.com/en/features/storage> możemy sprawdzić aktualny stan większości przeglądarek.

5.2.3. Web SQL Database

Koljenym ciekawym rozwiązaniem problemu przechowywania danych po stronie klienta jest Web Sql Database. Jest to baza danych która została umieszczona po stronie klienta. Jest to dodatkowa warstwa która korzysta z SQLite. Zaletą takiego rozwiązania jest przeżucenie kosztów obliczeniowych na stronę klienta, operacje takie jak sortowanie, filtrowanie na bazie danych wykonywane są przy wykorzystaniu zasobów komputera klienta.

Nie wątpliwą zaletą w stosunku do omówionej w rozdziale 5.2.2 pamięci jest możliwość ustalenia wielkości zalokowanej wielkości pamięci. Dzięki temu nie jesteśmy ograniczeni granicą maksymalnie 5MB w wielkości przeglądarek.

Przykład wykorzystania został zaprezentowany na przykładzie. Po stworzeniu bazy danych o nazwie ExampleDataBase i rozmiarze 2MB możemy korzystać z niej jak z każdej innej bazy danych. Możliwe są m.in. operacje tworzenia table (linie 3-5), wstawiania danych (parametry podajemy po treści zapytania, następnie możemy podać funkcje które zostaną wykonane w momencie sukcesu lub w przypadku wystąpienia błędu). Odczyt danych (linie 13-18) jest analogiczny do pozostałych operacji, dodatkowo pokazano w jaki sposób możliwa jest iteracja po otrzymanych wynikach.

Wadą tego rozwiązania jest jego popularność. Na stronie browser stats możemy sprawdzić jaka jest aktualnie popularność poszczególnych przeglądarek na świecie. Porównując te dane z dostępnością Web Sql dostępną na stronie <http://www.html5rocks.com/en/features/storage> (dostęp) możemy wnioskować że prawie 45% użytkowników (korzystających z IE i Firefox) nie mogą korzystać z tego rozwiązania. Fakt ten dyskwalifikuje je.

```
1      var db = window.openDatabase("ExampleDataBase", "1.0", "Description", 2 * 1024
2                                     * 1024);
3
4      db.transaction(function(tx) {
5          tx.executeSql("CREATE TABLE TableTest (id REAL UNIQUE, text TEXT)");
6      });
7
8      db.transaction(function(tx) {
9          tx.executeSql("INSERT INTO TableTest (id, text) VALUES (?, ?)", [1, 'test'],
10             onSuccess,
11             onError);
12      });
13
14      db.transaction(function(tx) {
15          tx.executeSql("SELECT * FROM TableTest", [], function(tx, result) {
16              for (var i = 0, item = null; i < result.rows.length; i++) {
17                  ...
18              }
19          });
20      });
```

Listing 5.3: json

5.2.4. IndexedDB

Trzecim rozwiązaniem jest indexDb.

5.3. Oś czasu

Tematem pracy oprócz wykorzystania i prezentacji danych na bazie informacji geograficznych jest ich połączenie z osią czasu, która pozwoliłaby na zmianę prezentowanych danych ze względu na interesujący nas okres czasu.

Ponieważ własna implementacja osi czasu która nieograniczałaby użytkownika, pozwala mu w pełni korzystać z możliwości pełnej aplikacji, która tylko w części składa się z możliwości manipulacji czasem postanowiono wykorzystać istniejące rozwiązanie.

Przeprowadzając badania rozwiązań głównymi aspektami na które zwrócono uwagę było:

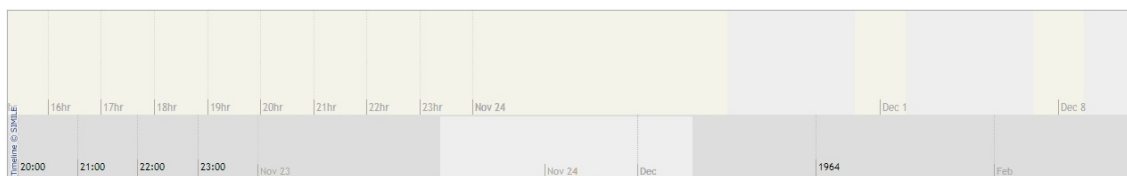
- Typ licencji Niezwykle ważne jest aby końcowy projekt był całkowicie otwarty dla dalszego rozwoju, pozwalał użytkownikom na adaptację rozwiązań dla swoich potrzeb. Z uwagi na to licencja musi pozwalać na nieograniczoną modyfikację kodu źródłowego.
- Wykorzystywane technologie Pamiętając że efektem końcowym powinien być framework odpowiadający szerokiemu gronowi odbiorców, będący jednocześnie darmowym ważne aby języki programowania wykorzystane przy jego tworzeniu też taki były. Oznacza to nie korzystanie z płatnych, wymagających licencji oprogramowań. Używane standardy powinny być rozpropagowane i używane przez szerokie grono obiorców.

Po długich i wnikliwych poszukiwaniach zdecydowano skorzystać z widжет-u o nazwie Timeline który został stworzony przez projekt SIMILE działający na uczelni MIT. Pozwala on na bardzo dużą konfigurowalność dzięki czemu jego modyfikacja, nawet bez ingerencji w kod źródłowy jest bardzo prosta. Projekt jest w stanie stworzyć kilka pasm które będą określały interwały czasu. Pozwala to aby wynik końcowy był przejrzysty bez względu czy prezentujemy wydarzenia które miały miejsce w okresie kilku minut czy setek lat.

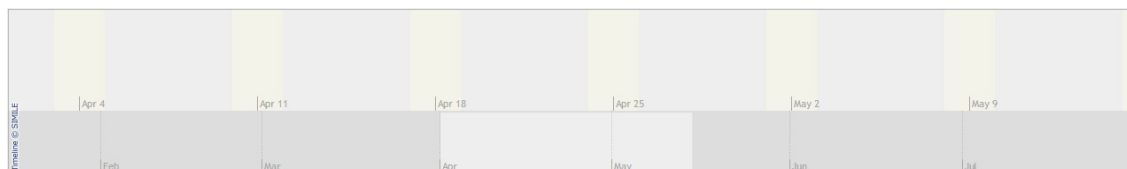
Problemem który pojawił się było połączenie wybranego sposobu prezentacji osi czasu z mapą i elementami które powinny zmieniać się wraz z upływem czasu.

Rysunek 5.3 prezentuje możliwości tego projektu. Pasma odpowiadają zakresom czasu, nie muszą on jednak być jednakowe w każdym miejscu. W zaprezentowanym przykładzie dzień 23 listopada uznany został za warty dokładniejszemu przyjrzeniu, łatwo możemy z dokładnością co do godziny zmieniać zakres czasu. Natomiast kolejny miesiąc może być o wiele mniej ciekawy, a obszar przeznaczony dla niego być mniejszy niż ten posiadany przez wymieniony powyżej dzień.

Dla porównania 5.4 prezentuje o wiele prostszą konfigurację w której dolne pasmo podzielone jest przez miesiące, natomiast górę przez tygodnie.



Rysunek 5.3: Timeline



Rysunek 5.4: Timeline 2

5.4. Przybliżanie wyników

Zmniejszanie ilości makretów w zależności od stopnia przybliżenia

5.5. Parser kml

Format KML jest używany do reprezentacji danych graficznych w aplikacjach firmy Google, takich jak Google Maps czy Google Earth. Oparty jest na standardzie XML, wrażliwy na wielkość liter, wymusza ścisłą poprawność danych z formatem. Oznacza to że każdy tag, element ma ściśle określone miejsce i nie może pojawić się nigdzie indziej.

Aby tworzona aplikacja miała możliwość współpracy ze stworzonymi wcześniej zbiorami danych, mapami mimo faktu iż informacje te nie będą w pełni wykorzystywały możliwości frameworku, stworzony został parser plików kml. Dzięki temu możliwy jest import danych, uzupełnienie ich o dodatkowe informacje i zapis do własnego formatu.

5.6. Interferjs

6. Przykład użycia

10

7. Podsumowanie

2-3 strony

Bibliografia

- [Cho09] C. Choi. *A palaeolithic map from 13,660 calBP: engraved stone blocks from the Late Magdalenian in Abauntz Cave (Navarra, Spain)*. Journal of Human Evolution, 2009.
- [chr]
- [Irv28] Washington Irving. *A History of the Life and Voyages of Christopher Columbus*. G. & C. Carvill, 1828.
- [Lam09] L. Lamport. *LaTeX system przygotowywania dokumentów*. Wydawnictwo Ariel, Krakow, 2009.
- [Pil10] M. Pilgrim. *Dive into HTML5*. 2010.