

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Vizualizace kartografických dat na webu

Daria Silivonenko

Vedoucí: RNDr. Ondřej Žára

Obor: Web a multimédia

Studijní program: Softwarové technologie a management

Květen 2018



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Silivonenko** Jméno: **Daria** Osobní číslo: **434891**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Softwarové technologie a management**
Studijní obor: **Web a multimedia**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Vizualizace kartografických dat na webu

Název bakalářské práce anglicky:

Geographic Visualization on the web

Pokyny pro vypracování:

Seznamte se s nejčastěji používanými formáty geografických a kartografických dat: WKT/WKB, Shapefile, GeoJSON, TopoJSON. Dále se seznamte s problematikou geografických projekcí (Robinsonova, Mercatorova, Winkel tripel, ortografická).

Naimplementujte následně nástroj, který bude geografická data projektovat a zobrazovat pomocí tří odlišných webových technologií - SVG, HTML Canvas a WebGL. Porovnejte různé aspekty (komplexita, výkon, náročnost na hardware, softwarová podpora) těchto technologií. Uvažte, jak povaha vstupních dat ovlivňuje volbu technologie použité pro vizualizaci v prostředí HTTP klienta.

Do výsledného nástroje přidejte podporu pro interaktivitu (ovládání posunu a měřítka myši). Volitelně přidejte i ovládání dotykem a následně otestujte, je-li hardware současných mobilních zařízení dostatečně vhodný pro jeho používání. Shrňte možnosti prezentace obsáhlých geografických/kartografických dat na webu a navrhněte, jakým směrem by se mělo ubírat další vylepšování naprogramované aplikace.

Seznam doporučené literatury:

Scott Murray: Interactive Data Visualization for the Web, O'Reilly Media, ISBN 9781491921289
John P. Snyder: Flattening the Earth, The University of Chicago Press, ISBN 9780226767475
<https://en.wikipedia.org/wiki/Geovisualization>
https://en.wikipedia.org/wiki/Map_projection
<https://medium.com/@mbostock/command-line-cartography-part-1-897aa8f8ca2c>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

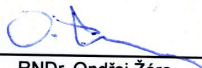
RNDr. Ondřej Žára, Katedra počítačové grafiky a interakce

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:


Datum zadání bakalářské práce: **28.02.2018**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**


RNDr. Ondřej Žára
podpis vedoucí(ho) práce

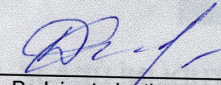

podpis vedoucí(ho) ústavu/katedry


prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

9.05.2018
Datum převzetí zadání


Podpis studentky

Průběh: Web a multimedia
Studijní program: Softwarové technologie a management
Katedra počítačové grafiky a interakce
Fakulta elektrotechnická
Číslo: 434891

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce: Visualizace kartografických dat na webu

Název bakalářské práce anglicky: Geographic Visualization on the web

Podmínky pro vypracování:
Zároveň se s největší možnou mírou účastí na přednáškách a konzultacích. Dále se seznamím s odbornou literaturou a vypracuji práci. Dále se seznamím s odbornou literaturou a vypracuji práci. Dále se seznamím s odbornou literaturou a vypracuji práci.

Seznam doporučené literatury:
Scott Murray: Interactive Data Visualization for the Web, O'Reilly Media, ISBN 978-1-449-31989-2
John F. Shneiderman: Interactive Data Visualization, The University of Chicago Press, ISBN 978-0-262-08557-5
http://en.wikibooks.org/wiki/Interactive_data_visualization
<http://medium.com/@michaelcommand-line-visualization-part-1-807a930c>

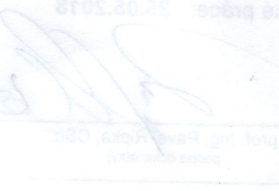
Jméno a pracovní vedoucí(n) bakalářské práce: RNDr. Ondřej Žár, Katedra počítačové grafiky a interakce

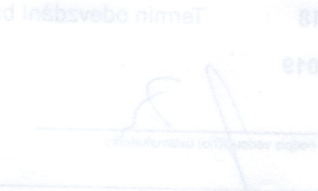
Jméno a pracovní (druhý) vedoucí(n) nebo konzultant(ka) bakalářské práce: _____

Datum zadání bakalářské práce: 30.09.2018

Plnost zadání bakalářské práce: 30.09.2018

Termín odevzdání bakalářské práce: 30.05.2019


Podpis vedoucího práce


Podpis studentky

Poděkování

Děkuji všem, kteří mě podporovali při psaní této práce. Zejména vedoucímu práce RNDr. Ondřeji Žárovi za odbornou pomoc, čas a trpělivost. Děkuji také rodině a přátelům za jejich podporu a motivaci.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala sama s přispěním vedoucího práce a že jsem v práci uvedla veškeré použité informační zdroje. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry Počítačové grafiky a interakce na FEL.

V Praze, 25. Května 2018

Abstrakt

Cílem dané práce je seznámit se s nejčastěji používanými formáty geografických a kartografických dat: WKT/WKB, Shapefile, GeoJSON, TopoJSON. Dále se seznámit s problematikou některých geografických projekcí jako Robinsonova, Mercatorova, Winkel Tripel a ortografická. Dalším úkolem práce je na základě prostudovaných materiálů naimplementovat nástroj, který bude geografická data projektovat a zobrazovat pomocí tří odlišných webových technologií - SVG, HTML Canvas a WebGL. Důvodem vytváření toho nástroje je porovnání různých aspektů použitých technologií, takových jako komplexita, výkon, náročnost na hardware, softwarová podpora.

Klíčová slova: projekce, webové technologie, formát

Vedoucí: RNDr. Ondřej Žára

Abstract

The goal of this thesis is familiarize ourselves with some of the most popular storage formats for geospatial data: WKT/WKB, Shapefile, GeoJSON, TopoJSON. Then study some of the popular mapping projections such as Robinson, Mercator, Winkel Tripel and orthographic projections. Next thesis's goal is implement a software tool that is able to project and visualize geospatial data using three distinct web technologies - SVG, HTML Canvas and WebGL. The reason of creating this tool is comparing different aspects of used technologies: complexity, performance, hardware requirements, software support.

Keywords: projection, storage format, web technologies

Title translation: Geographic Visualization on the web

Obsah

| | |
|---|-----------|
| 1 Úvod | 1 |
| 2 Teoretická část | 3 |
| 2.1 Problematika datových formátů . | 3 |
| 2.1.1 WKT/WKB | 4 |
| 2.1.2 Shapefile | 6 |
| 2.1.3 GeoJSON | 8 |
| 2.1.4 TopoJSON | 9 |
| 2.2 Principy geografických projekcí . | 10 |
| 2.2.1 Robinsonova projekce | 10 |
| 2.2.2 Mercatorova projekce | 12 |
| 2.2.3 Winkel Tripel | 13 |
| 2.2.4 Ortografická projekce | 13 |
| 2.3 Webové technologie pro prezenaci kartografických dat | 14 |
| 2.3.1 SVG | 14 |
| 2.3.2 HTML Canvas | 15 |
| 2.3.3 WebGL | 16 |
| 3 Praktická část | 19 |
| 3.1 Implementace | 19 |
| 3.2 Struktura a funkce | 20 |
| 3.3 Analýza | 22 |
| 3.3.1 Komplexita | 22 |
| 3.3.2 Výkon | 23 |
| 3.3.3 Náročnost na hardware | 24 |
| 3.3.4 Softwarová podpora | 24 |
| 3.3.5 Výsledek analýzy | 24 |
| 4 Závěr | 25 |
| Literatura | 27 |
| A Zkratky | 29 |
| B Obsah příloženého CD | 31 |

Obrázky

| | |
|---|----|
| 2.1 WKT reprezentace různých typů geometrií..... | 4 |
| 2.2 WKT reprezentace souřadnicového systému..... | 5 |
| 2.3 schéma WKB reprezentace geometrie..... | 6 |
| 2.4 Příklad zápisu geometrické kolekce pomocí GeoJSON..... | 9 |
| 2.5 Příklad zápisu geometrické kolekce pomocí TopoJSON..... | 10 |
| 2.6 Robinsonova projekce..... | 12 |
| 2.7 Mercatorova projekce..... | 12 |
| 2.8 Winkel Tripel projekce..... | 13 |
| 2.9 Ortografická projekce..... | 14 |
| 2.10 Příklad zvětšení SVG obrázku (vlevo) a HTML Canvas obrázku (vpravo)..... | 15 |
| 2.11 Příklad vytváření čtverce pomocí SVG..... | 15 |
| 2.12 Příklad vytváření čtverce pomocí HTML Canvas..... | 16 |
| 2.13 Vytváření shaderů, jejich kompilace a aplikace na program ve WebGL..... | 17 |
| 3.1 Struktura naimplementované aplikace..... | 20 |
| 3.2 Menu aplikace..... | 21 |
| 3.3 Příklad vykreslení mapy Afriky danou aplikací..... | 21 |
| 3.4 Reakce aplikace na klikání..... | 22 |
| 3.5 Příklad zápisu funkcí mouseover a mousout..... | 23 |

Tabulky

| | |
|--|----|
| 2.1 Hodnoty pro označení typu geometrie v .shp souboru..... | 7 |
| 2.2 Robinsonovy tabulkové souřadnice | 11 |
| 3.1 Tabulka uvádějící přibližný čas potřebný na vykreslování konkrétního množství prvků objektu..... | 23 |



Kapitola 1

Úvod

V dnešní době existuje mnoho nástrojů pro zobrazování geografických a kartografických dat, jako Google maps, Mapbox, ArcGIS a další. Dokonce jsou k dispozici i webové aplikace umožňující uživatelům vytvářet vlastní mapy a geografické objekty. K takovým aplikacím můžeme zařadit OpenLayers, Mapshaper apod. Samozřejmě různé nástroje používají i různé technologie pro vizualizaci geodat. Úkolem dané bakalářské práce je naimplementovat speciální nástroj pro vizualizaci geografických a kartografických dat různými webovými technologiemi a zjistit která z těchto technologií je vhodná nebo naopak není vhodná pro takovou vizualizaci a jakým způsobem je lépe prezentovat obsáhlá geografická/kartografická data na webu.

Práce se skládá ze tří částí: teoretické, praktické a závěrečné.

V teoretické části probereme hlavní okruhy, se kterými je třeba se seznámit, abychom zjistili, co zahrnuje proces vizualizace. Je nutno vědět, v jakém formátu se ta data zapisují, jakou projekci je lepší použít pro prezentaci a nakonec jaké existují webové technologie umožňující takovou prezentaci.

Praktická část je zaměřená právě na vytváření webového nástroje, který bude schopen vizualizovat geodata různými způsoby. Díky danému nástroji budeme způsobilý porovnat rozličné aspekty využitých technologií a po jejich analýze určit jejich výhody a nevýhody a případně vybrat tu nevhodnější pro vizualizaci geografických/kartografických dat technologií.

V závěru shrneme možnosti prezentace geodat na webu a navrhneme, jakým směrem by se mělo ubírat další vylepšování naprogramované aplikace.

Kapitola 2

Teoretická část

V dané kapitole probereme několik nejvíc známých datových formátů pro ukládání kartografických dat, také principy geografických projekcí a nakonec vzhledneme na existující webové technologie vhodné pro vizualizaci geografických/kartografických dat.

2.1 Problematika datových formátů

Existuje docela velké množství formátů pro ukládání geografických a kartografických dat. Ta data mohou být organizována dvěma základními modely: rastrovým a vektorovým. Pro praktickou část své bakalářské práce jsem rozhodla používat vektorový model a v důsledku toho i formát, který bude vhodný pro taková data. Daný výběr je zdůvodněn několika příčinami:

- Vektorová data jsou jednodušší v aktualizaci a údržbě, zatímco budeme chtít aktualizovat rastrový obrázek, on bude muset být zcela překreslován.
- Vektorová data mohou být snadněji zvětšována/zmenšována a reprojektována, což může zjednodušit kombinování vektorových vrstev z různých zdrojů.
- Vektorová data mohou být zobrazena jako vektorová grafika, zatímco rastrová data se budou zobrazovat jako obraz, který může mít blokový vzhled pro hranice objektu. (v závislosti na rozlišení rastrového souboru)
- Vektorová data umožňují vizuálně hladkou a snadnou implementaci překryvných operací, zejména pokud jde o grafické a tvarově řízené informace, jako jsou mapy, trasy a vlastní písma, které jsou s rastrovými daty obtížnější.

Jelikož bylo řešeno používat vektorová data, vzhledneme na několik formátů, které umožňují ukládat určitý typ dat. Podrobněji jsem se zastavila na těchto formátech: WKT/WKB, Shapefile, GeoJSON a TopoJSON.

2.1.1 WKT/WKB

WKT (Well-known text) je textový značkovací jazyk, který se používá pro reprezentaci vektorových geometrických objektů na mapě a pro reprezentaci prostorových souřadnicových systémů prostorových objektů.

- Reprezentace vektorových geometrií:

WKT může být použit jak k vytváření nových instancí, tak i k převodu již existujících instancí do textové podoby pro alfanumerické zobrazení. Geometrické objekty, které lze znázornit pomocí WKT, zahrnují: body, čáry, mnohoúhelníky, multigeometrii reprezentující více než jednu geometrii stejné dimenze v jednom objektu, a také geometrické kolekce, které ukládají geometrie různých velikostí. WKT se skládá ze tří částí: typ geometrie, typ souřadnic a seznam souřadnic. Parametr Typ souřadnic určuje, zda má geometrie souřadnici Z (souřadnice označující výšku) a/nebo lineární prostorový souřadnicový systém (M). Pokud nemá určeny žádný z těchto typů souřadnic, tak zůstává parametr typ souřadnic prázdný. V případě, že má definováno Z souřadnici, tak je typ souřadnice nastaven na Z, pokud má lineární prostorový souřadnicový systém - je nastaven na M, a pokud má obě, tak ZM. Parametr Seznam souřadnic definují vrcholy geometrie. Položky seznamu souřadnic jsou rozděleny čárkami a celý seznam je uzavřen závorkami. Pokud geometrie obsahuje více komponent, musí se zaokrouhlovat každá část například:

$$\text{MultiPoint}((10 \ 10), (20 \ 20))$$

Pokud je geometrie prázdná, po záznamu typu geometrie následuje klíčové slovo EMPTY.

```

GEOMETRYCOLLECTION ( POINT (4 6), LINESTRING (4 6, 7 10))
POINT ZM (1 1 5 60)
POINT M (1 1 80)
POINT EMPTY
CIRCULARSTRING (1 5, 6 2, 7 3)
COMPOUNDCURVE(CIRCULARSTRING (0 0, 1 1, 1 0), (1 0, 0 1))
CURVEPOLYGON (CIRCULARSTRING (-2 0, -1 -1, 0 0, 1 -1, 2 0, 0 2, -2 0), (-1 0, 0 0.5, 1 0, 0 1, -1 0))
MULTICURVE ((5 5, 3 5, 3 3, 0 3), CIRCULARSTRING(0 0, 2 1, 2 2))
TRIANGLE ((0 0 0, 0 0 1, 0 1 0, 0 0 0))
TIN (((0 0 0, 0 0 1, 0 1 0, 0 0 0)), ((0 0 0, 1 1 0, 0 0 0)))
POLYHEDRALSURFACE Z (
  ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)),
  ((0 0 0, 0 1 0, 0 1 0, 0 0 1, 0 0 0)),
  ((0 0 0, 1 0 0, 1 1 0, 0 0 1, 0 0 0)),
  ((1 1 1, 1 0 1, 0 0 1, 0 1 1, 1 1 1)),
  ((1 1 1, 1 0 1, 1 0 0, 1 1 0, 1 1 1)),
  ((1 1 1, 1 1 0, 0 1 0, 0 1 1, 1 1 1)),
)

```

Obrázek 2.1: WKT reprezentace různých typů geometrií.

- Reprezentace prostorových souřadnicových systémů:

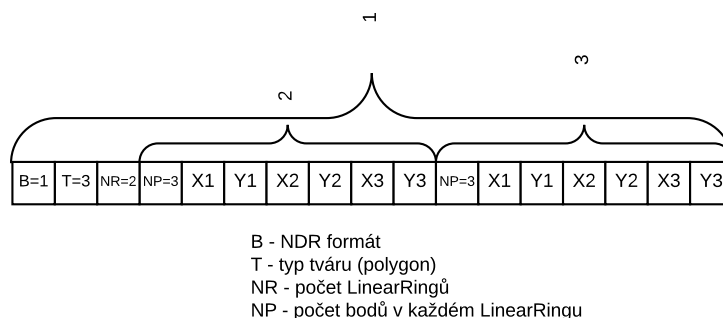
Prostorový souřadnicový systém může být projekční souřadnicový systém (definován X, Y souřadnicemi), geografický (definován zeměpisnou šířkou a délkou) nebo geocentrický (definován souřadnicemi X, Y, Z). Souřadnicový systém datového souboru je identifikován klíčovým slovem PROJCS - pokud jsou data uvedeny v projekčních souřadnicích, GEOGCS - pokud jsou v geografických souřadnicích, nebo GEOCCS - pokud jsou v geocentrických souřadnicích (geocentrický souřadnicový systém má počátek ve středu Země). U projekčních souřadnic za klíčovým slovem PROJCS následují všechny komponenty, které definují projekční souřadnicový systém: název souřadnicového systému, geografický souřadnicový systém, 0 nebo více parametrů, lineární jednotka měření. Geografický souřadnicový systém je definován jménem, datem, vztahným bodem a úhlovou jednotkou měření. Geocentrický souřadnicový systém je podobný geografickému, je definován jménem, datem, vztahným bodem a lineární jednotkou měření. Každá komponenta systému má klíčové slovo (např. DATUM nebo UNIT) a po něm následují definované parametry v hranatých závorkách oddělené čárkou. Některé komponenty se skládají ze subkomponent, výsledkem v tomto případě bude vnořená struktura. Příklad zápisu souřadnicového systému ve formátu WKT lze vidět na obrázku 2.2.

```
PROJCS["PCS_Lambert_Conformal_Conic",
GEOGCS["GCS_North_American_1983",
DATUM["D_North_American_1983",
SPHEROID["GRS_1980",6378137.0,298.257222101]],
PRIMEM["Greenwich",0.0],
UNIT["Degree",0.0174532925199433]],
PROJECTION["Lambert_Conformal_Conic"],
PARAMETER["False_Easting",6200000.0],
PARAMETER["False_Northing",3000000.0],
PARAMETER["Central_Meridian",-91.86666666666666],
PARAMETER["Standard_Parallel_1",49.0],
PARAMETER["Standard_Parallel_2",77.0],
PARAMETER["Latitude_Of_Origin",63.390675],
UNIT["Meter",1.0]]
```

Obrázek 2.2: WKT reprezentace souřadnicového systému.

WKB (Well-known Binary) je binární ekvivalent WKT. WKB dovoluje vyměňovat geometrické objekty mezi SQL/CLI-klientem a SQL-implemencí v binárně formě. První byte ve streamu identifikuje, jak jsou binární hodnoty reprezentovány, a to buď pomocí NDR reprezentace (Network Data Representation) nebo XDR (eXtended Data Representation). Tyto způsoby reprezentace se liší v pořadí bytů. NDR kódování je „little endian“, což znamená že nejméně významný byte je ukládán jako první, XDR kódování – „big endian“ - nejvýznamnější byte je ukládán jako první. Další část streamu

udává typ geometrie. Ten může být definován hodnotami od 1 do 7 označující Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon a GeometryCollection. Pokud se geometrie skládá z více geometrií, další byty udávají, kolik dalších geometrií se ve streamu nachází. Další sada bytů označuje počet bodů v první geometrii a pak následují souřadnice X a Y každého bodu. Pro každou další geometrii se postup opakuje.



Obrázek 2.3: schéma WKB reprezentace geometrie.

2.1.2 Shapefile

Shapefile formát ukládá netopologickou geometrii a atributy prostorových prvků do datasetu. Geometrie je uložena jako tvar obsahující sadu vektorových souřadnic. Daný formát je vyvinut a řízen společností Esri (Environmental Systems Research Institute) jako otevřená specifikace pro interoperabilitu dat mezi produkty Esri a dalšími softwarovými produkty GIS. Shapefile formát může prostorově popsat vektorové prvky: body, čáry a mnohoúhelníky. Každá položka má obvykle atributy, které ji popisují, například název nebo teplotu.

2D uspořádání souřadnicových dat předpokládá kartézský souřadnicový systém za použití souřadnic X, Y nebo Easting Northing. Toto uspořádání je konzistentní i pro geografické souřadnicové systémy, kde je uspořádání podobné (používají zeměpisnou šířku a délku). Geometrii mohou také podporovat 3- nebo 4-rozměrné souřadnice Z a M pro výšku a měřítko. Souřadnice Z ukládá výšku každé souřadnice v 3D prostoru, který lze použít pro analýzu nebo vizualizaci geometrie pomocí 3D počítačové grafiky. Uživatelsky definovaná dimenze M může být použita pro jednu z mnoha funkcí, jako je například ukládání lineárních referenčních měření nebo relativní čas funkce do 4D prostoru.

ESRI Shapefile se skládá z hlavního souboru, indexového souboru a dBASE tabulky. Hlavní soubor je soubor s přímým přístupem, soubor s proměnnou délkou záznamu, ve kterém každý záznam popisuje geometrie pomocí seznamu vrcholů. V indexovém souboru obsahuje každý záznam offset (posun) odpovídající záznamu hlavního souboru od začátku hlavního souboru. Tabulka

dBASE obsahuje atributy vlastností s jedním záznamem pro každou funkci. Vztah "one-to-one" mezi geometrií a atributy je založen na čísle záznamu. Záznamy atributů v souboru dBASE musí být ve stejném pořadí jako záznamy v hlavním souboru.

Hlavní soubor, indexový soubor a soubor dBASE mají stejnou předponu. Předpona musí začínat alfanumerickým znakem (a-Z, 0-9), následujícím nulami nebo sedmi znaky (a-Z, 0-9, -, _). Přípona hlavního souboru je .shp. Přípona indexového souboru je .shx. Přípona pro tabulku dBASE je .dbf. Všechna písmena v názvu souboru jsou malá, protože operační systémy jsou citlivé na velikost písmen u názvů souborů.

Organizace hlavního souboru. Hlavní soubor (.shp) obsahuje záhlaví souboru (100 bytů) s pevnou délkou a záznamy s proměnnou délkou. Každý záznam o proměnné délce je tvořen záhlavím záznamu o pevné délce a dále je následován obsahem záznamu s proměnnou délkou. Všechny geometrie v shapefile, které nejsou Null, musejí mít stejný typ geometrie. Hodnoty typů geometrie jsou následující:

| Hodnota | Typ geometrie |
|---------|---------------|
| 0 | null |
| 1 | Point |
| 3 | PolyLine |
| 5 | Polygon |
| 8 | MultiPoint |
| 11 | PointZ |
| 13 | PolyLineZ |
| 15 | PolygonZ |
| 18 | MultiPointZ |
| 21 | PointM |
| 23 | PolyLineM |
| 25 | PolygonM |
| 28 | MultiPointM |
| 31 | MultiPatch |

Tabulka 2.1: Hodnoty pro označení typu geometrie v .shp souboru.

Záhlaví každého záznamu obsahuje číslo záznamu a délku obsahu záznamu. Záznam záhlaví má pevnou délku 8 bytů.

Obsah záznamu formátu shapefile se skládá z typu geometrie a dále následují geometrická data pro danou geometrii. Délka obsahu záznamu závisí na počtu částí a vrcholů geometrie. Pro každý typ geometrie nejdříve musí být popsána geometrie a pak jeho mapování na obsah záznamu na disk.

Organizace indexového souboru. Indexový soubor (.shx) obsahuje záhlaví délky 100 bytů a následují záznamy pevné délky 8 bytů. Celkem je struktura souboru indexu velmi podobná struktuře hlavního souboru. Délka souboru je uložena v záhlaví indexu a je to celková délka indexového souboru v 16-bitových slovech (padesát 16-bitových slov záhlaví plus čtyřnásobek počtu záznamů). Délka obsahu, která je uložena v záznamu indexu, je stejná jako hodnota, která je uložena v hlavičce záznamu hlavního souboru.

Organizace dBASE souboru. Soubor dBASE (.dbf) obsahuje libovolné atributy nebo klíče atributů, ke kterým mohou být připojeny další tabulky. Tento formát je standardní DBF soubor používaný velkým množstvím tabulkových aplikací v systémech WindowsTM a DOS. Existují čtyři požadavky, které musejí být splněny pro dBASE soubor:

- název souboru musí mít stejnou předponu jako shapefile a indexový soubor. Jeho přípona musí být .dbf;
- tabulka musí obsahovat jeden záznam pro každou geometrii;
- pořadí záznamu musí být stejné jako pořadí geometrických prvků v hlavním (* .shp) souboru;
- hodnota roku v hlavičce dBASE musí být po roce 1900.

2.1.3 GeoJSON

GeoJSON je otevřený formát vytvořený pro reprezentaci jednoduchých geografických prvků spolu s jejich neprostorovými atributy. Tento formát je založen na JSON (JavaScript Object Notation). Mezi prvky, které můžeme použít, patří: body, line stringy, polygony a také vícedílné kolekce těchto typů. GeoJSON formát se liší od ostatních standardů GIS v tom, že byl napsán internetovou pracovní skupinou vývojářů, a není proto udržován formální normalizační organizací.

GeoJSON je podporován řadou mapovacích a GIS softwarových balíčků, včetně OpenLayers, Leaflet, MapServer, Geoforge software, GeoServer, GeoDjango, GDAL a CartoDB. Je také možné použít GeoJSON s PostGIS a Mapnik, které zpracují formát přes knihovnu konverze GDAL OGR. Bing Maps, Yahoo! a Google také podporují služby GeoJSON ve svých API službách.

Reprezentace GeoJSON geometrických typů jsou analogické s binární (WKB) a textovou (WKT) reprezentací. Referenční souřadnicový systém pro všechny souřadnice GeoJSONu je geografický souřadnicový systém. Využívá World Geodetic System 1984 (WGS 84) se vztaheným bodem a s jednotkami délky a šířky uvedených v desetinných stupních.

Aplikační rozhraní Google Maps JavaScript v3 přímo podporuje integraci datových vrstev GeoJSON od 19. března 2014. Pro jazyk Julia je k dispozici balíček GeoJSON.jl.


```

{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [102.0, 0.5]
      },
      "properties": {
        "prop0": "value0"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]
        ]
      },
      "properties": {
        "prop0": "value0",
        "prop1": 0.0
      }
    }
  ]
}

```

Obrázek 2.4: Příklad zápisu geometrické kolekce pomocí GeoJSON

GitHub také podporuje vykreslování GeoJSONu a export Potrace GeoJSON.

Geojson.io podporuje vykreslování a úpravu GeoJSON ve webovém prohlížeči.

Pozoruhodným potomkem GeoJSONu je TopoJSON, což je rozšíření GeoJSONu, které kóduje geoprostorovou topologii a které typicky poskytuje menší velikost souborů.

■ 2.1.4 TopoJSON

TopoJSON je rozšíření GeoJSON, které kóduje topologii. Geometrie v TopoJSON souborech je spojena dohromady ze segmentů sdílených částí nazývaných oblouky. Tyto oblouky jsou sekvence bodů, zatímco řetězce (line strings) a polygony jsou definovány jako sekvence oblouků. Každý oblouk je definován pouze jednou, ale může být několikrát odkazován různými geometrií, což snižuje redundanci a tím snižuje velikost souboru.

Pro další snížení velikosti souboru TopoJSON používá kvantifikované delta-kódování pro celočíselné souřadnice. To se podobá zaokrouhlení hodnot souřadnic (např. LilJSON), ale s větší účinností a kontrolou ztráty informací. Stejně jako GeoJSON jsou soubory TopoJSON snadno modifikovány v textovém editoru a lze je komprimovat pomocí gzip.

Referenční implementace specifikace TopoJSON je k dispozici jako nástroj příkazového řádku pro zakódování TopoJSONu z GeoJSON (nebo ESRI Shapefiles) a knihovny JavaScript na straně klienta k dekodování TopoJSON zpět do GeoJSONu.

TopoJSON je také podporován populárním nástrojem OGR od verze 1.11 a PostGIS od verze 2.1.0.

```
{
  "type": "Topology",
  "objects": {
    "example": {
      "type": "GeometryCollection",
      "geometries": [
        {
          "type": "Point",
          "properties": {
            "prop0": "value0"
          },
          "coordinates": [102, 0.5]
        },
        {
          "type": "LineString",
          "properties": {
            "prop0": "value0",
            "prop1": 0
          },
          "arcs": [0]
        }
      ]
    }
  },
  "arcs": [
    [[102, 0], [103, 1], [104, 0], [105, 1]]
  ]
}
```

Obrázek 2.5: Příklad zápisu geometrické kolekce pomocí TopoJSON

2.2 Principy geografických projekcí

Při práci s geografickými a kartografickými data samozřejmě manipulujeme s reálnými světovými souřadnicemi. Aby ty souřadnice byly správně vykresleny na obrazovce, musíme je převést do vhodné podoby. A přesně to dělá geografická projekce. V této kapitole seznámíme se s několika známými geografickými projekcí a jejich principy.

2.2.1 Robinsonova projekce

Robinsonova projekce je mapová projekce světové mapy, která zobrazuje celý svět najednou. Byla speciálně vytvořena ve snaze najít dobrý způsob, jak snadno ukázat celou planetu jako plochý obraz. Projekce byla navržena Arthurem H. Robinsonem v roce 1963.

Dana projekce není ekvivalentní (zachovávají poměry ploch) ani konformní (zachovávají úhly). Arthur H. Robinson věřil, že tento způsob – více než

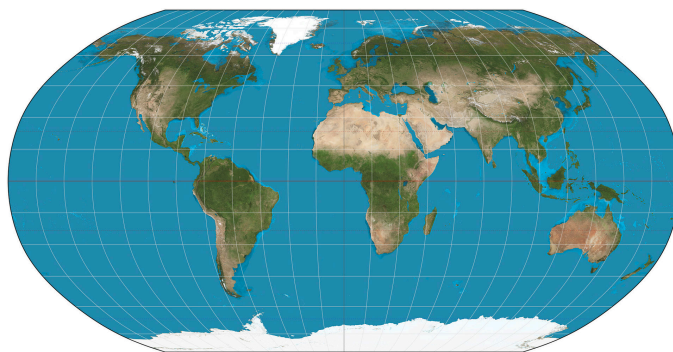
jakýkoli jiný - vedl k lepšímu celkovému pohledu na mapu. Poledníky se jemně zakřivují, vyhýbají se extrémům (krajům), ale tím se póly natáhnou na dlouhé čáry.

Zkreslení v okolí pólů je výrazné, ale směrem dál od pólů už přestává být zkreslení tak výrazné. Rovné rovnoběžky znamenají silné úhlové zkreslení ve vysokých zeměpisných šířkách směrem k vnějším okrajům mapy, což je chyba vlastní pseudocyklické projekci. Nicméně v době, kdy byla projekce vytvořena, efektivně splnila cíl vytvářet esteticky přitažlivé zobrazení celého světa.

| Šířka v st. | PLEN | PDFE |
|-------------|--------|--------|
| 00 | 1.0000 | 0.0000 |
| 05 | 0.9986 | 0.0620 |
| 10 | 0.9954 | 0.1240 |
| 15 | 0.9900 | 0.1860 |
| 20 | 0.9822 | 0.2480 |
| 25 | 0.9730 | 0.3100 |
| 30 | 0.9600 | 0.3720 |
| 35 | 0.9427 | 0.4340 |
| 40 | 0.9216 | 0.4958 |
| 45 | 0.8962 | 0.5571 |
| 50 | 0.8679 | 0.6176 |
| 55 | 0.8350 | 0.6769 |
| 60 | 0.7986 | 0.7346 |
| 65 | 0.7597 | 0.7903 |
| 70 | 0.7186 | 0.8435 |
| 75 | 0.6732 | 0.8936 |
| 80 | 0.6213 | 0.9394 |
| 85 | 0.5722 | 0.9761 |
| 90 | 0.5322 | 1.0000 |

Tabulka 2.2: Robinsonovy tabulkové souřadnice

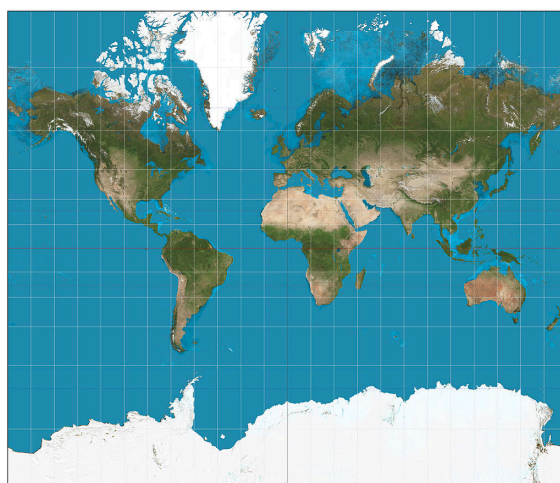
Kontinenty na této projekci mají poměrně správnou velikost a umístění. Proto se nazývá "orthophanic". Použitím empirického výzkumu s přístupem «trial-and-error» (přístup pokusu a omyl), Robinson sestavil tabulkové souřadnice, a takovým způsobem vlastně definoval projekce. Tabulka je indexována podle šířky po intervalech v 5 stupňů, mezilehlé hodnoty se vypočítají pomocí interpolace. Sloupec PLEN označuje délku rovnoběžky zeměpisné šířky a sloupec PDFE se vynásobí číslem 0.5072 pro získání vzdálenosti této rovnoběžky od rovníku. Poledníky jsou rovnoměrně rozmístěny na každé rovnoběžné šířce. Jak vypadají Robinsonovy souřadnice je vidět v tabulce 2.2.



Obrázek 2.6: Robinsonova projekce.

■ 2.2.2 Mercatorova projekce

Mercatorova projekce je válcovou mapovou projekcí, kterou představil flámský geograf a kartograf Gerardus Mercator v roce 1569. Stala se standardní mapovou projekcí pro námořní účely, protože byla schopna reprezentovat loxodromy (čára protínající poledníky pod stálým úhlem) jako přímky. Přestože lineární měřítko je stejné ve všech směrech kolem jakéhokoli bodu, čímž se zachovávají úhly a tvary malých objektů. Mercatorova projekce deformuje velikost objektů, jak se zeměpisná šířka zvyšuje od rovníku k pólům. Například kontinenty, jako je Grónsko a Antarktida, se zdají mnohem větší, než ve skutečnosti jsou v poměru ke kontinentům nacházejícím se blízko rovníku, jako je například střední Afrika. Ale nehledě na ten fakt, daná projekce je nejvíc používaná projekce, protože promítá obrázek nebo mapu na čtverečkovou plochu. S takovou plochou se mnohem lépe pracuje díky tomu, že manipulujeme dobře známými x a y souřadnicemi.

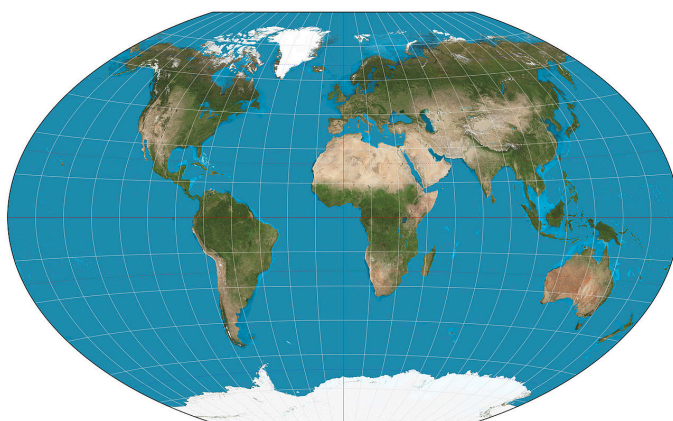


Obrázek 2.7: Mercatorova projekce.

Kvůli převedení obrazu do čtverečkové podoby mnoho známých online mapových služeb jako Mapy Bing, OpenStreetMap, Mapy Google, MapQuest, Mapy.cz Yahoo! a další používají Mercatorovou projekci pro své mapové podklady s názvem Web Mercator nebo Google Web Mercator.

2.2.3 Winkel Tripel

Winkel tripel projekce (dříve Winkel III) modifikovaná azimutální mapová projekce světa, je jednou z tří projekcí navržených německým kartografem Oswaldem Winkellem v roce 1921. Winkel Tripel je neobvyklý v tom, že je vytvořen průměrem souřadnic X a Y z dalších dvou zřídka používaných projekcí: Aitoff a Equirectangular. Výsledkem je projekce, která vypadá velmi podobně jako Robinsonova. Nicméně Winkel Tripel má několik jedinečných vlastností. Například rovnoběžky v Robinsonove projekci jsou rovnoběžné, zatímco ve Winkel Tripel se nacházejí lehce zakřivené nerovnoběžné linie. Název Tripel (německý výraz pro "trojitý") odkazuje na Winkelův cíl minimalizovat tři druhy zkreslení: oblast, směr a vzdálenost.



Obrázek 2.8: Winkel Tripel projekce.

2.2.4 Ortografická projekce

Použití ortografické projekce v kartografii se datuje do starověku. Stejně jako stereografická a gnómonická projekce je ortografická projekce perspektivní (nebo azimutální) projekcí, ve které je země koule promítnutá na tečnou rovinu nebo sečnou rovinu. Bod pohledu pro ortografické zobrazení je v nekonečné vzdálenosti. Zobrazuje polokouli zeměkoule, jak je vidět z vesmíru, kde horizont je velký kruh (linie dělící sféru na dvě polokoule). Tvary a plochy jsou zkreslené zejména v blízkosti okrajů.



Obrázek 2.9: Ortografická projekce

2.3 Webové technologie pro prezenaci kartografických dat

V dané kapitole jsou popsány tři webové technologie, které se používají pro renderování geografických a kartografických dat, a to jsou SVG, HTML Canvas a WebGL technologie.

2.3.1 SVG

SVG je vektorový grafický formát založený na XML. Obsah SVG může být statický, dynamický, interaktivní a animovaný. Tento formát je velmi flexibilní. U SVG lze určit chování objektů pomocí SVG DOM: například definovat chování objektu na klikání myši nebo změnu měřítka kolečkem myši.

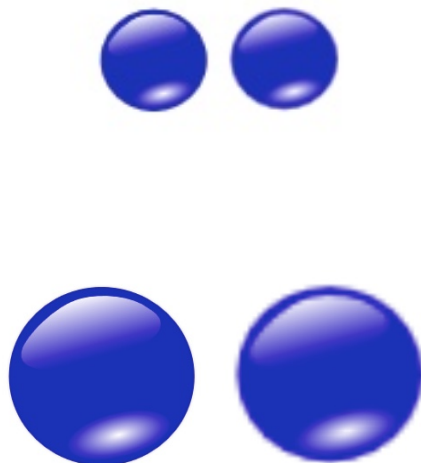
SVG technologie má hodně pozitivních vlastností a aspektů. Je podporována všemi populárními prohlížeči: Chrome, Explorer, Mozilla, Opera a Safari.

Další jednou z velkých výhod SVG je uchování kvality objektu při změně měřítka. Což nelze říct o HTML Canvas. Příklad chování SVG a Canvas při změně měřítka lze pozorovat na obrázku 2.10.

Také velkou výhodou je strukturovatelnost. Na základě přehledné struktury SVG je možné jednotlivé objekty vzájemně seskupovat, členit a dokonce jednoduše dodatečně upravovat. SVG pracuje v souřadnicovém systému, a proto provádět dodatečné změny lze pouze přepisem souřadnic jednotlivých objektů.

Dalším plusem SVG je to, že je možné kreslit SVG grafiku pomocí standardního vektorového editoru, jako je Adobe Illustrator nebo Inkscape anebo

používat čistý CSS pro ovládání jeho vzhledu a provádění atraktivní animace, na rozdíl od Canvas, s kterým se dá pracovat pouze prostřednictvím JavaScriptu.



Obrázek 2.10: Příklad zvětšení SVG obrázku (vlevo) a HTML Canvas obrázku (vpravo).

Ale SVG má i své nevýhody i když jich není moc. Jednou z takových nevýhod je zvětšení rozměru souboru pokud objekt obsahuje velké množství malých prvků. A to následně je důvodem delší doby vykreslování objektu.

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <svg height="100" width="100">
      <rect width="100" height="100" fill="red" />
    </svg>
  </body>
</html>
```

Obrázek 2.11: Příklad vytváření čtverce pomocí SVG

2.3.2 HTML Canvas

HTML Canvas je vlastně HTML prvek sloužící k scriptovatelnému vykreslování různých geometrií. Jinak můžeme říct, že Canvas je bitmapa, která má takovou vlastnost, že po zobrazení výsledné grafiky neudrží žádné informace o vykreslených objektech, ale veškeré vykreslené obrazce jsou uloženy

jako matice jednotlivých pixelů. V tom je největší výhoda Canvas, protože díky té vlastnosti je schopen moc rychle vykreslovat libovolný objekt, což je velmi užitečný při vytváření například real-time aplikací. Grafika je vytvářena i upravována výhradně pomocí skriptovacího jazyka, protože Canvas neobsahuje žádné elementy pro vytváření primitivních tvarů oproti SVG.

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <canvas height="100" width="100" id="myCanvas">
  </canvas>
  <script>
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
    ctx.fillStyle = "#FF0000";
    ctx.fillRect(0,0,100,100);
  </script>
  </body>
</html>
```

Obrázek 2.12: Příklad vytváření čtverce pomocí HTML Canvas

Nevýhodou HTML Canvas je ztráta kvality grafiky při změně rozlišení. Příklad toho chování Canvas je vidět na obrázku 2.10.

Co se týče podporovatelnosti, tak Canvas je podporován všemi populárními v dnešní době prohlížeči: Chrome, Explorer, Mozilla, Opera a Safari.

■ 2.3.3 WebGL

WebGL je JavaScript API pro renderování 2D a 3D interaktivní grafiky. Daná technologie je založena na OpenGL ES 2.0 (grafická knihovna) a poskytuje API pro 3D grafiku. Používá HTML5 prvek Canvas a DOM rozhraní. WebGL programy se skládají z obslužného kódu napsaného v JavaScriptu a kódu shaderu, který je vykonáván na grafické kartě počítače.

Pro reprezentaci grafiky pomocí WebGL je potřeba mít dva typy shaderů: vertex a fragment shader. Vertex shader je program který se provede na každém vrcholu neboli vertexu vstupní geometrie. Zde také se propisují i základní transformace jako translace (posun), rotace a scale (změna měřítko), a to se uskutečňuje násobením vrcholu konkrétní maticí.

Fragment shader je program, který se provádí na každém pixelu rasterizované geometrie, tedy pracuje s 2D obrazem geometrie. Tady se aplikují textury, přidává se barva pixelům.

V JavaScript souboru se shadery vytvářejí, kompiluje se a aplikují se na program, které je spojí dohromady. Pak se vytvářejí buffery, kde následně budou uloženy vrcholy geometrie a její indexy. WebGL umí vykreslovat jen trojúhelníky, to znamená, že každou geometrie rozdělí na trojúhelníky a do bufferu indexu uloží právě indexy vrcholů tvořících trojúhelníky.

```
// Create shaders
var vertexShader = gl.createShader(gl.VERTEX_SHADER);
var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);

gl.shaderSource(vertexShader, vertexShaderText);
gl.shaderSource(fragmentShader, fragmentShaderText);

gl.compileShader(vertexShader);
if (!gl.getShaderParameter(vertexShader, gl.COMPILE_STATUS)) {
    console.error('ERROR compiling vertex shader!', gl.getShaderInfoLog(vertexShader));
    return;
}

gl.compileShader(fragmentShader);
if (!gl.getShaderParameter(fragmentShader, gl.COMPILE_STATUS)) {
    console.error('ERROR compiling fragment shader!', gl.getShaderInfoLog(fragmentShader));
    return;
}

var program = gl.createProgram();

gl.attachShader(program, vertexShader);
gl.attachShader(program, fragmentShader);
gl.linkProgram(program);
```

Obrázek 2.13: Vytváření shaderů, jejich kompilace a aplikace na program ve WebGL.

Kapitola 3

Praktická část

Cílem praktické části dané bakalářské práce bylo vytvořit jednostránkovou aplikaci pro vizualizaci geografických a kartografických dat. Aplikace dovolí zobrazit geodata, která jsou zapsána ve formátu GeoJSON. Také aplikace bude naimplementována tak, aby geodata mohla být vykreslována třemi způsoby: SVG, HTML Canvas a WebGL technologie. Rovněž určitý webový nástroj bude mít podporu pro interaktivitu: ovládání posunu a měřítka pomocí myši.

Kapitola se skládá ze tří částí: implementace, kde je popsáno, jak se vytvářela aplikace a jaké knihovny byli při tom použity, struktura a funkce, popisující přesné fungování programu a jeho hlavní funkce a nakonec analýza, část obsahující porovnání třech použitých technologií.

3.1 Implementace

Daný webový nástroj pro zobrazení geografických a kartografických dat je napsán v programovacím jazyce JavaScript. Zároveň je předem stanoveno, že to je jednostránková aplikace. Pro implementaci bylo použito několik knihoven:

- knihovna d3.js;

d3.js je JavaScript knihovna, která pomáhá lehce manipulovat s daty a jejich vizualizací na webu za použití SVG, HTMLCanvas a CSS. V konkrétní práci jsem používala d3-geo, což je část d3 knihovny zodpovědná za práci s geodaty a obsahující velké množství funkcí různých geografických projekcí a funkcí pro manipulování jimi.

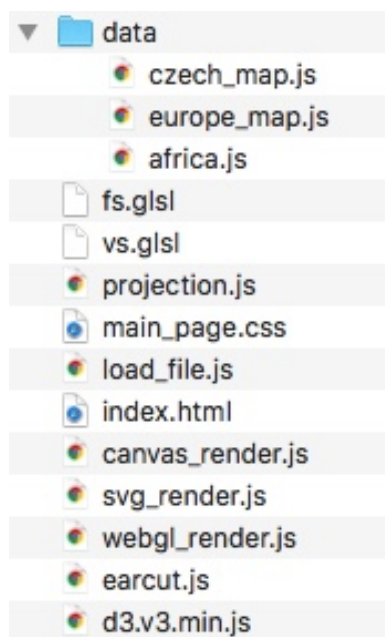
- earcut.js;

Earcut.js je malá a velmi rychlá JavaScript knihovna pro triangulaci polygonu. Tu jsem potřebovala při renderování dat pomocí WebGL, protože táto technologie pracuje s trojúhelníky.

Protože dána aplikace je jednostránková, obsahuje pouze jeden index.html soubor, který se spouští v prohlížeči pro zahájení využívání aplikace.

Aplikace má jeden .css soubor, který je zodpovědný za to, jak bude aplikace vypadat v prohlížeči.

Dále má naprogramovaný nástroj tři JavaScript soubory, nazývané se `svg_render.js`, `canvas_render.js` a `webgl_render.js`, které jsou zodpovědné za vykreslování geodat SVG, HTML Canvas a WebGL technologií.



Obrázek 3.1: Struktura naimplementované aplikace.

Soubor `projection.js` obsahuje nastavení Mercatorovy projekce, která pak se předává do vykreslujících funkcí jako parametr.

Další soubor `load_data.js` dělá to, že vybere GeoJSON soubor pro vizualizaci z lokální složky z počítače nebo nahraje GeoJSON soubor, který je definován předem.

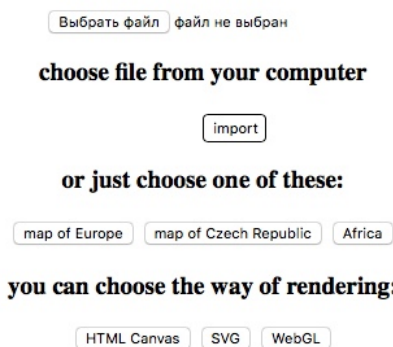
V programu jsou tři další JavaScript soubory. Nacházející ve vlastním adresáři `Data`, které vlastně reprezentují předem definovaná geodata, to jsou zapsané v GeoJSON formátu mapy České republiky, Evropy a Afriky.

Pro WebGL renderování aplikace obsahuje dva shadery, vertex a fragment shadery, s příponou `.gsl`.

3.2 Struktura a funkce

Připomenu, že naprogramovaný nástroj je jednostránkový. Při spuštění aplikací v prohlížeči je vidět, že obrazovka je rozdělená na dvě části: vlevo se nachází navigační oblast a vpravo oblast, kde se vykresluje GeoJSON soubor.

V levém menu (obrázek 3.2) je možné vybrat jeden z navržených GeoJSON souborů nebo je možnost vybrat lokální soubor z počítače. Také jak bylo řečeno v popisu kapitoly, v menu se dá vybrat způsob renderování objektů, což se dělá prostým klikáním na určité tlačítko. Mapy, které jsou uvedeny v aplikaci jako příklady, se defaultně vykreslují pomocí SVG technologie.



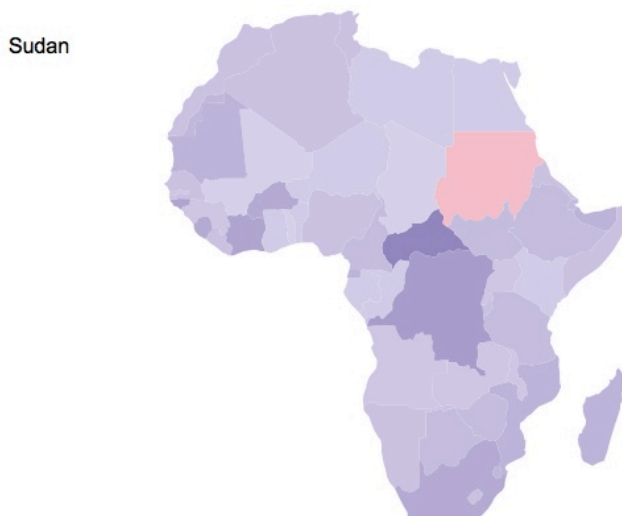
Obrázek 3.2: Menu aplikace.

Jak již bylo zmíněno výše, druhá část obrazovky je plochou pro vykreslování GeoJSON souboru. Na obrázku 3.3 lze pozorovat zobrazování mapy Afriky renderovanou pomocí SVG.



Obrázek 3.3: Příklad vykreslení mapy Afriky danou aplikací.

Při klikání na konkrétní země se mění barva prvku. Jak to vypadá na obrazovce je zobrazeno na obrázku 3.4.



Obrázek 3.4: Reakce aplikace na klikání.

Také v aplikaci je možné přibližovat kolečkem myši a posouvat obrázek klikáním a držením myši.

3.3 Analýza

Jedním z úkolů praktické části bakalářské práce je porovnat různé aspekty použitých webových technologií SVG, HTML Canvas a WebGL: komplexitu, výkon, náročnost na hardware, softwarovou podporu.

3.3.1 Komplexita

- Pokud hovoříme o komplexitě, tak nejsnadnější ze třech použitých webových technologií je SVG. Zahrnuje v sobě některé již předem definované tvary, jako kruh, čtverec, polygon atd. Tj. pro vykreslení nějaké jednoduché geometrie, jako kruh nebo čtverec ani není potřeba psát script v JavaScriptu. A zároveň SVG používá DOM, a ten obsahuje speciální funkce různých událostí, jako například ovládání myši. Vzor funkce ovládání pohybu myši je na obrázku 3.5.
- S HTML Canvas se věci trochu liší. Ve skutečnosti musíme Canvas vnímat jako plátno, na kterém potom budete zobrazovat nějakou geometrii. Takže zjevný velký rozdíl oproti SVG je přístup k tvarům pouze prostřednictvím javascriptového API. Data v Canvasu nemají žádnou strukturu, proto pokud budeme chtít v rámci události na Canvasu jako takovém analyzovat jeho obsah, musíme logiku převodu pozice myši na vektorová data implementovat ručně.

- A maximálně náročným v konkrétním případě bude kreslení pomocí WebGL technologie. Za prvé, WebGL vyžaduje dva zvláštní soubory: vertex a fragment shadery. Za druhé, je potřeba umět základy WebGL, abychom byli schopni propojit shadery, přečíst všechny vrcholy geometrie a jejich indexy a následně je vykreslit na obrazovku. Za třetí, složitost se spočívá v nutnosti triangulace vstupních dat. To je netriviální operace, která se neprovádí u ostatních technologií a přitom zabírá značné množství procesorového času.

```

// Highlight hovered object
function mouseover(d){
  d3.select(this).style('fill', 'pink');
  drawText(getName(d));
}

// Reset object color
function mouseout(){
  mapLayer.selectAll('path')
    .style('fill', function(d){return centered && d === centered ? '#D5708B' : fill(d)});
  textName.text('');
}

```

Obrázek 3.5: Příklad zápisu funkcí mouseover a mousout.

3.3.2 Výkon

Pro porovnání výkonu třech použitých technologií jsem vypočítala čas, potřebný na vykreslování několika objektů z různým počtem prvků všemi těmito technologiemi. Výsledkem je následující tabulka (Tab. 3.1).

| Počet features | Počet bodů | SVG(ms) | Canvas(ms) | WebGL(ms) |
|----------------|------------|---------|------------|-----------|
| 1 | 923 | 16.5 | 11.5 | 0.3 |
| 38 | 971 | 36.6 | 29.5 | 1 |
| 51 | 2141 | 40.3 | 24.9 | 2 |

Tabulka 3.1: Tabulka uvádějící přibližný čas potřebný na vykreslování konkrétního množství prvků objektu.

Podle tabulky vidíme, že se samozřejmě s rostoucím počtem prvků obsažených v geometrii zvyšuje čas renderování. A to pozorujeme u všech třech technologií. Také lze označit, že nejpomalejším způsobem vykreslování je SVG, dále stojí HTML Canvas a na prvním místě jako nejrychlejší se nachází WebGL.

■ 3.3.3 Náročnost na hardware

HTML Canvas běží dobře na jakémkoli moderním hardware. Pokud hardware dobře provozuje operační systém, pak i Canvas bude fungovat dobře. To se týče i SVG technologie. Ale náročnost na CPU v případě použití SVG je větší než při použití Canvas.

Při práci se složitým kontextem ve webových stránkách, například s 3D mapami, intenzivně roste náročnost na samotný prohlížeč a také na CPU (procesor počítače). Použití WebGL technologie je proto v tomto případě velmi vhodné, jelikož využívá pro práci grafickou kartu, čímž odstraňuje zátěž CPU, což nelze říct o SVG a HTML Canvas.

■ 3.3.4 Softwarová podpora

Co se týče softwarové podpory, i SVG i HTML Canvas jsou podporovány všemi moderními prohlížeči: Safari, Chrome, Opera, Firefox a Internet Explorer.

Pokud budeme mluvit o CSS Canvas kreslení, tedy funkci používající HTML Canvas jako obrázek pozadí, není podporována skoro žádným prohlížečem, pouze Safari.

Podporují je také i mobilní operační systémy a prohlížeče. Pro SVG existují speciální SVG Tiny a SVG Basic profily. Ale ne všechny mobilní prohlížeče podporují některé SVG funkce, jako například SVG fonty a transformace. K tím patří Opera Mini a UC prohlížeč na Android. Opera Mini navíc nepodporuje SVG animace.

Dostupnost WebGL hodně závisí na GPU (grafické kartě počítače), proto je podporován v dnešních populárních prohlížečích, ale ne v starších verzích. Ale to platí pokud hovoříme o verzi WebGL 1.0. Nové zveřejnění WebGL 2.0 zatím je podporováno pouze Google Chromem od verze 63 a Firefoxem 58 a novějším. V mobilních prohlížečích není podporován jenom Operou Mini.

■ 3.3.5 Výsledek analýzy

Jestli budeme chtít vykreslovat obsáhlá kartografická a geografická data, nejrychlejším způsobem jejich vizualizace bude pomocí WebGL technologie.

Ale pokud pro vizualizaci bude důležitější kvalita zobrazujících objektů, nikoliv rychlost, WebGL není tak vhodná varianta, jelikož WebGL stejně vykresluje ty objekty na Canvasu, a jak víme z kapitoly 2.3.2, jeho hlavní nevýhodou je ztráta kvality při změně měřítka. Takže v takové situaci je řešením použít SVG technologie.



Kapitola 4

Závěr

Cílem dané bakalářské práce bylo naimplementovat speciální nástroj pro vizualizaci geografických a kartografických dat, používající tři dobře známé webové technologie: SVG, HTML Canvas a WebGL.

Naimplementovaný program určitě může být rozšířen o další funkce. Například přidáním funkce ovládání dotykem pro použití aplikace na mobilních zařízeních. Také program může podporovat vizualizaci jiných datových formátů, které jsem zmiňovala v kapitole 2.1.

Práce na této aplikaci mi dovolila lépe se orientovat v programovacím jazyku JavaScript, s kterým jsem měla minimální zkušenosti, a také s webovými technologiemi SVG, HTML Canvas a WebGL. Rovněž jsem se seznámila se strukturou GeoJSON formátu pro ukládání geografických a kartografických dat.



Literatura

- [1] Scott Murray. *Interactive Data Visualization for the Web*. O'Reilly Media, 2013.
- [2] John P. Snyder. *Flattening the Earth*. The University of Chicago Press, 1993.
- [3] *Geografické mapové projekce*. 2018.
https://en.wikipedia.org/wiki/Map_projection
- [4] *WKT a WKB formáty*. 2014.
http://www.skylinesoft.com/SkylineGlobe/TerraExplorer/v6.5.1/APIReferenceGuide/Well_Known_Text_and_Well_Known_Binary_WKT_and_WKB.htm
- [5] *Extended geographic projections for d3-geo*. 2013. <https://github.com/d3/d3-geo-projection>
- [6] *Shapefile*. 2018.
<https://en.wikipedia.org/wiki/Shapefile>
- [7] *GeoJSON formát*. 2018.
<https://en.wikipedia.org/wiki/GeoJSON>
- [8] *TopoJSON formát*. 2012.
<https://github.com/topojson/topojson>
- [9] *GeoJSON mapy*. 2016.
<https://geojson-maps.ash.ms>
- [10] *Earcut knihovna*. 2016.
<https://github.com/mapbox/earcut>
- [11] *D3 geo video lessons*. 2012.
<https://www.dashingd3js.com/lessons/d3-geo-path>
- [12] *Softwarová podpora*. 2018.
<https://caniuse.com/#home>



Příloha A

Zkratky

| Zkratka | Význam |
|---------|--|
| WKT | Well-know Text |
| WKB | Well-known Binary |
| SQL | Structured Query Language |
| CLI | Call Level Interface |
| NDR | Network Data Representantion |
| XDR | eXtended Data Representantion |
| Esri | Environmental Systems Research Institute |
| GIS | Geographic Information System |
| JSON | JavaScript Object Notation |
| CSS | Cascading Style Sheets |
| XML | Extensible Markup Language |
| SVG | Scalable Vector Graphics |
| DOM | Document Object Model |
| API | Application Programming Interface |
| RAM | Random Access Memory |
| CPU | Central Processing Unit |
| GPU | Graphic Processing Unit |



Příloha B

Obsah přiloženého CD

- Text bakalářské práce (soubor BP-Siliv.pdf)
- Zdrojový kód naimplementované praktické části.