

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Programové rozhraní Google Earth



2012

Vít Schnaubelt

Anotace

Práce obsahuje popis možností uživatelského rozhraní pluginu Google Earth, stručnou dokumentaci programového rozhraní, odkaz na webovou aplikaci demonstrující toto rozhraní umístěnou na internetu a návod pro práci s touto aplikací. Dokumentace rozhraní obsahuje popis a návod na vytvoření základních objektů rozhraní.

Děkuji vedoucímu práce doc. RNDr. Michalu Krupkovi, Ph.D., za vedení a veškerou pomoc při řešení problémů, při zpracování této bakalářské práce. Dále společnosti Google za vytvoření přívětivého rozhraní k pluginu Google Earth a jejich dokumentaci.

Obsah

| | |
|--|-----------|
| 1. Úvod | 7 |
| 2. Plugin Google Earth | 8 |
| 2.1. Instalace | 8 |
| 2.2. Kompatibilita | 9 |
| 3. KML | 9 |
| 4. Možnosti pluginu | 10 |
| 4.1. Pohledy a pozice | 10 |
| 4.2. Značky a bubliny | 10 |
| 4.3. Linie a polygony | 11 |
| 4.4. KML: | 11 |
| 4.5. Obloha, Mars, Měsíc | 11 |
| 4.6. Nastavení pluginu | 12 |
| 4.7. Nastavení vrstev | 12 |
| 5. Uživatelská příručka | 13 |
| 5.1. Pohledy a pozice | 13 |
| 5.2. Značky a bubliny | 13 |
| 5.3. Linie a polygony | 14 |
| 5.4. KML | 15 |
| 5.5. Obloha, Měsíc, Mars | 15 |
| 5.6. Nastavení pluginu | 16 |
| 5.7. Info | 16 |
| 6. Programové rozhraní pluginu Google Earth | 17 |
| 6.1. Vložení pluginu na stránku | 17 |
| 6.2. Ovládání kamery | 18 |
| 6.3. Značky (Placemarks) | 18 |
| 6.3.1. Vytvoření pomocí funkcí: | 18 |
| 6.3.2. Vytvoření pomocí KML | 20 |
| 6.4. Bubliny (Balloons) | 20 |
| 6.4.1. Základní bublina | 21 |
| 6.4.2. HTML string bublina | 21 |
| 6.4.3. HTML DIV bublina | 21 |
| 6.5. Linie a polygony | 22 |
| 6.5.1. Vytvoření linie | 22 |
| 6.5.2. Vytvoření polygonu | 23 |
| 6.5.3. Styly | 23 |
| 6.6. Vlastní textury (Overlays) | 24 |
| 6.7. KML | 24 |

| | |
|--|-----------|
| 6.8. Obloha, Mars, Měsíc (Sky, Mars, Moon) | 25 |
| 6.9. Nastavení pluginu (Options) | 25 |
| 6.10. Nastavení vrstev (layers) | 26 |
| 6.11. Události (Events) | 26 |
| 7. Dokumentace tříd | 27 |
| 7.1. KmlObject | 27 |
| 7.2. KmlFeature | 28 |
| 7.3. KmlLookAt | 29 |
| 7.4. KmlAltitudeModeEnum | 30 |
| 7.5. KmlCoordArray | 31 |
| 7.6. KmlGeometry | 32 |
| 7.7. GESchemaObjectContainer | 32 |
| 7.8. KmlPoint | 33 |
| 7.9. KmlLineString | 34 |
| 7.10. KmlPolygon | 34 |
| 7.11. KmlPlacemark | 35 |
| 7.12. GEAbstractBalloon | 35 |
| 7.13. GEHtmlBalloon | 36 |
| Závěr | 38 |
| Conclusions | 39 |
| Reference | 40 |
| A. Obsah příloženého CD | 41 |

Seznam obrázků

| | | |
|-----|-------------------------------------|----|
| 1. | Plugin Google Earth | 8 |
| 2. | Ukázka značek a bublin | 10 |
| 3. | Ukázka polygonů a linií | 11 |
| 4. | Země, Měsíc, Mars, obloha | 12 |
| 5. | Ukázka vrstev | 12 |
| 6. | Menu pohledů a pozic | 13 |
| 7. | Menu značek | 14 |
| 8. | Menu linií a polygonů | 14 |
| 9. | Menu KML | 15 |
| 10. | Menu planet | 15 |
| 11. | Menu nastavení | 16 |
| 12. | Menu nápovědy a mazání | 16 |

1. Úvod

Práce obsahuje stručnou dokumentaci popisující uživatelské rozhraní pluginu Google Earth a webovou aplikaci demonstrující funkčnost tohoto rozhraní. Aplikace umí základní funkčnost tohoto rozhraní jako přidávání značek, objektů, změnu pozice, zobrazení a export do jazyka KML a další. Práce je rozdělena do třech hlavních částí. V první části jsou popisovány možnosti rozhraní, pochopitelné bez znalostí programování. Následuje uživatelská příručka popisující práci s aplikací. A třetí, neméně důležitá část, je popis programového rozhraní, který obsahuje popis funkcí a ukázky kódu v JavaScriptu a HTML. Další součásti práce jsou popis pluginu Google Earth, návod na instalaci a jeho kompatibilita, Stručný popis jazyka KML a dokumentace nejdůležitějších tříd.

2. Plugin Google Earth



Obrázek 1. Plugin Google Earth

Google Earth, dříve zvaný Earth Viewer, byl vytvořen firmou Keyhole a roku 2004 odkoupen společností Google. Google Earth je program sloužící pro pozorování 3D modelu planety s různými možnostmi. Od dubna 2010 existuje Google Earth plugin, který umožňuje vložit plnohodnotný model planety do webového prohlížeče. Plugin umožňuje přidávat objekty jako značky, polygony, měnit pozici a natočení kamery, zobrazovat různá vesmírná tělesa (Země, Měsíc, Mars) a měnit nastavení zobrazení předmětu jako budovy, stromy, cesty, hranice ... Plugin také umožňuje práci s geodatovým formátem [KML](#). Vedle Google Earth společnost Google ještě pracuje na projektu Google Maps. Z počátku byly tyto dva projekty rozdílné, Google Maps byla webová aplikace a Google Earth desktopová. Od vzniku Google Earth pluginu a vložení dokonalých družicových snímků do Google Maps jsou si tyto programy až na některé funkcionality (Street view u Google Maps a různé planety u Google Earth) téměř totožné. Dalé existuje placená verze Google Earth, která obsahuje navíc nástroje pro měření vzdálenosti, export do obrázků s vysokým rozlišením, přístup k většímu množství dat, lepší podporu a další. Poslední verzi pluginu je nyní Google Earth plugin 6.2.2.6613, u desktopové verze Google Earth 6.2.

2.1. Instalace

Plugin lze stáhnout na stránkách [3]. Po instalaci stačí restartovat prohlížeč a plugin je funkční.

2.2. Kompatibilita

Google Earth plugin je nyní podporován pouze ve vybraných prohlížečích a operačních systémech.

Microsoft Windows(XP, Vista)

- Google Chrome 5.0+
- Internet Explorer 7.0+
- Firefox 3.0+
- Flock 1.0+

Apple Mac OS X 10.5 and higher (Intel)

- Google Chrome 5.0+
- Firefox 3.0+
- Safari 3.1+

Plugin může také fungovat i v prohlížečích podporujících NPAPI s npruntime rozšířením (npruntime extension). Operační systém Windows 7 není přímo podporován ale plugin pod tímto systémem funguje.

3. KML

KML neboli Keyhole Markup Language je aplikací jazyka XML a slouží pro uchování, publikaci a distribuci geografických dat. Byl vyvinut firmou Keyhole pro projekt Earth Viewer a roku 2004 společně s jmenovaným projektem prodán společnosti Google Earth. KML původně sloužil pouze pro Google Earth a Google Maps, ale od roku 2008, kdy se jeho verze KML 2.2 stala standardem OGC (Open Geospatial Consortium), se začíná používat i v jiných GIS (Geografický informační systém). Jazyk umožňuje reprezenovat body, linie, plochy, pohledy a jiné. Google Earth plugin umožňuje aplikovat všechny KML 2.2 tagy a dokáže vracet KML kód některých svých objektů. Podrobná dokumentace jazyka KML k nalezení zde [\[1\]](#).

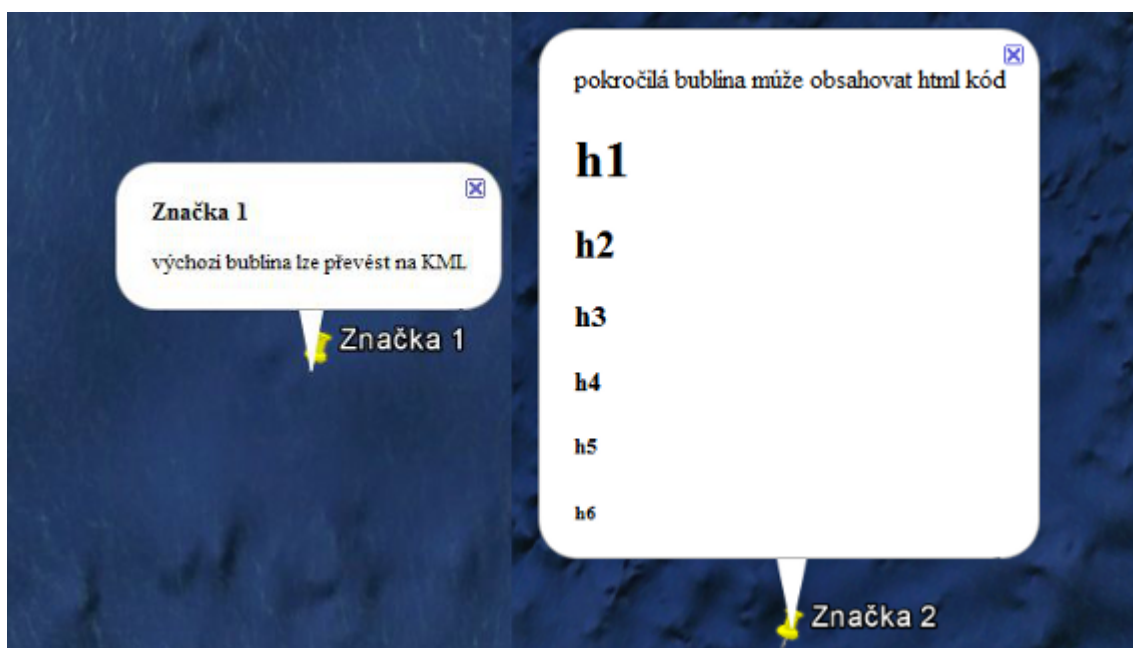
4. Možnosti pluginu

Tato kapitola obsahuje popis všech možností které Google Earth plugin umožňuje pomocí programového rozhraní. Pro stanovení pozice se v pluginu používá souřadnicovým systémem WGC84 (zeměpisná délka +90 až -90, zeměpisná šířka +180 až -180). Výšku může plugin reprezentovat různými způsoby jako nadmořská, nadzemská, výška nad mořským dnem. Další možností je, že objekty budou upoutány na zemi nebo mořské dno.

4.1. Pohledy a pozice

Plugin umožňuje měnit pozici, natočení a nadmořskou výšku pozorovatele. Pohled je tvořen pozicí prostředního bodu reprezentovaného pomocí souřadnicového systému popsáno v předešlé kapitole. Výšky, která je dána velikostí v metrech a druhem reprezentace výšky pozorovatele (popsané v předešlé kapitole) do maximální velikosti 63781,370km. Dále je možné měnit natočení kamery až o 90 stupňů od osy planety s pozorovatelem, a kolem této osy.

4.2. Značky a bubliny



Obrázek 2. Ukázka značek a bublin

Pomocí rozhraní je možné na glóbus přidávat značkám které po rozkliknutí zobrazí bublinu s daným textem. Tyto značky mohou být libovolně pojmenovány,

mohou být různě velké, a navíc mohou změnit svoji velikost po najetí kurzorem. Po kliknutí na značku se zobrazí bublina s nastaveným textem.

4.3. Linie a polygony



Obrázek 3. Ukázka polygonů a linií

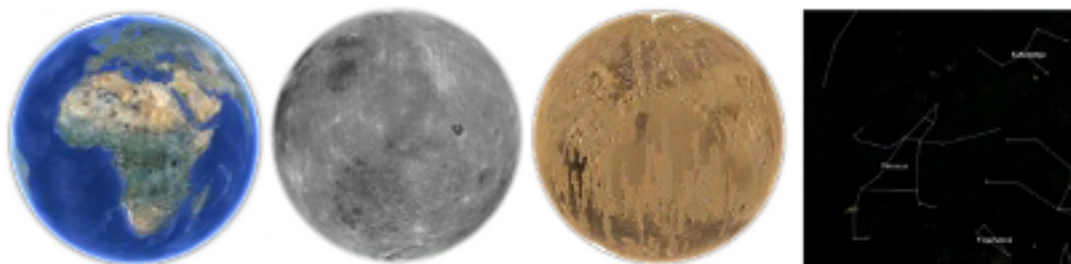
Na planetu je možné také přidávat různé útvary jako linie a polygony. U těchto těles je možné měnit šířku čáry a barvu čáry. Tělesa se mohou nacházet v různých výškách v závislosti na druhu počítání výšky (nadmořská, nadzemská ...). Linie i polygony jsou reprezentovány množinou bodů ve formátu WGS84. Linie je zobrazena jako propojení zadaných bodů v daném pořadí. Polygon je tělo podobné linii, ale navíc je zde propojen první a poslední bod, a takto vytyčená plocha je vybarvena šedou barvou. Linie i polygony mohou být doplněny po zemi, čímž mohou vzniknout různá 3D tělesa.

4.4. KML:

Plugin umožňuje zobrazovat na glóbu útvary definované v jazyce [KML](#) a převádět objekty jako značky, linie, polygony na KML kód.

4.5. Obloha, Mars, Měsíc

Země není jediným objektem který dokáže plugin zobrazit. Plugin dále umožňuje, mimo zemi, zobrazit planetu Mars, Měsíc a hvězdnou oblohu země. Se všemi planetami i s oblohou je možné pracovat stejně jako se Zemí.

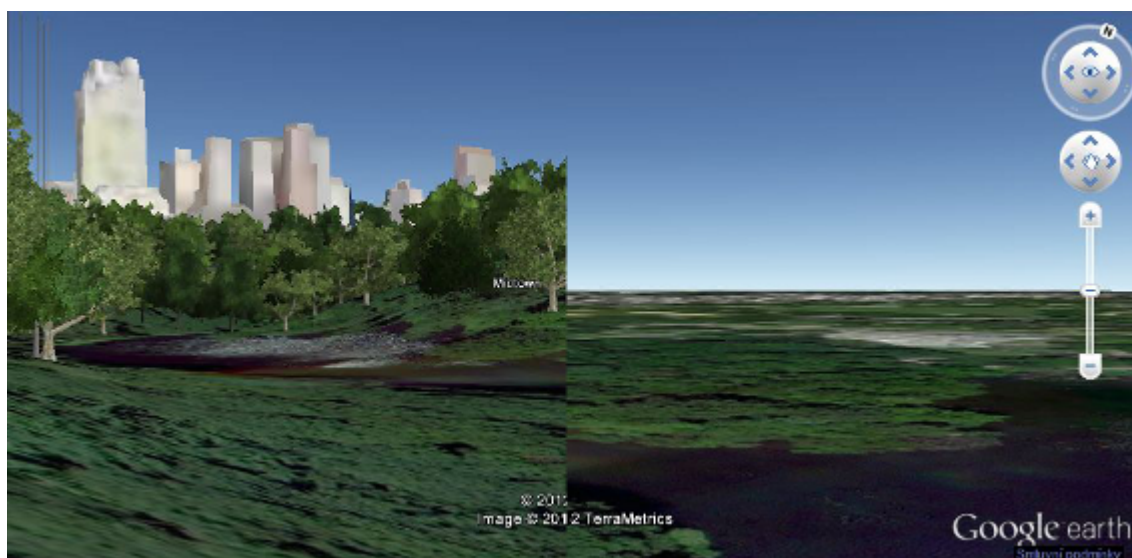


Obrázek 4. Země, Měsíc, Mars, obloha

4.6. Nastavení pluginu

Plugin umožňuje nastavovat další prvky zpříjemňující uživatelskou práci. Umožňuje zobrazit stavový řádek, ve kterém jsou zobrazeny souřadnice kurzoru myši, výška země na těchto souřadnicích a výška pozorovatele. Dále je možné měnit viditelnost ovládacího prvku, na pravé straně pluginu, ovladače času (slouží pro změnu stáří pořízených snímků), dále zobrazení mřížky, měřítka, minimapky, atmosféry a možnost zakázat ovládání myši.

4.7. Nastavení vrstev



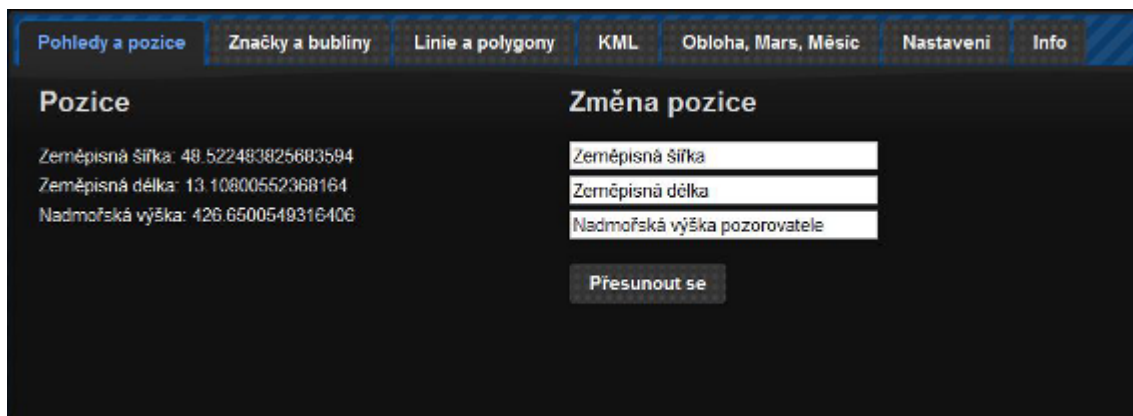
Obrázek 5. Ukázka vrstev

Vedle obvyčejného nastavení pluginu se zde nachází ještě nastavení vrstev, které umožňuje zobrazovat objekty jako státní hranice, budovy, cesty, stromy, slunce a 3D terén. Budovy a domy jsou dostupné pouze na některých místech planety Země.

5. Uživatelská příručka

Webová aplikace se skládá z HTML souboru ve standardu HTML 4.01, několika javascriptových souborů, obsahujících funkcionalitu a CSS souboru, obsahujícího styly. Stránky s aplikací jsou dostupné na adrese www.bcprojekt.wz.cz. Na horní straně stránky se nachází samotný plugin, pod kterým se vyskytuje menu. Menu sloužící pro změnu spodní části stránky, kde se nachází komponenty dané části menu. Aplikace je vytvořena pro demonstrování základních částí uživatelského rozhraní s ukázkou funkcionality rozhraní. Pro správný chod webové aplikace musí počítač splňovat podmínky [kompatibility](#) a musí mít povolen JavaScript ve webovém prohlížeči. Tato kapitola obsahuje popis a pokyny k ovládní aplikace.

5.1. Pohledy a pozice

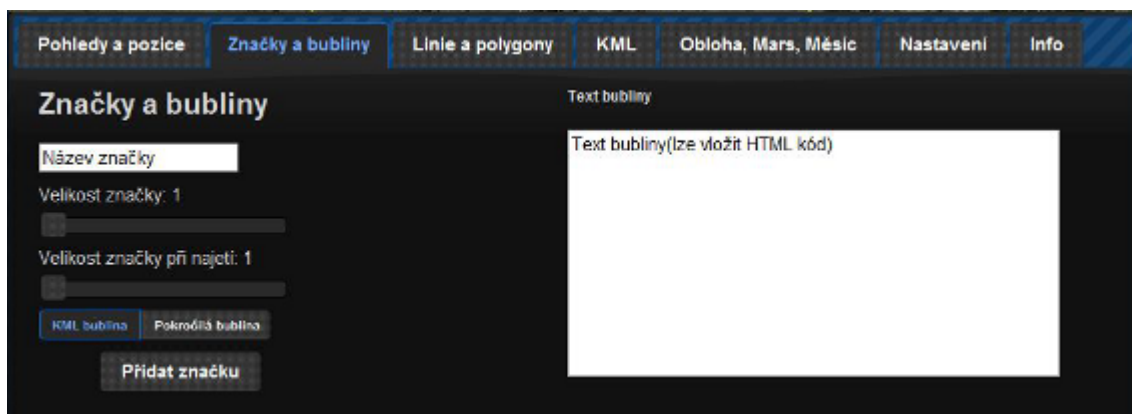


Obrázek 6. Menu pohledů a pozic

Část Pohledy a pozice se zabývá zobrazením pozice kurzoru myši a možností změnit pozici dle zadání uživatele. V levé části je zobrazena zeměpisná šířka, délka a nadmořská výška kurzoru myši. Na pravé straně se nachází textové pole pro zadání vlastních souřadnic pro změnu pozice. Do textových polí je možné zadávat hodnoty s desetinou tečkou (i čárkou), pokud budou hodnoty mimo správný rozsah (zeměpisná šířka +90 až -90, zeměpisná výška +180 až -180 a nadmořská výška > 0) aplikace nahlasí chybu špatně zadaných hodnot. Při správně zadaných hodnotách se planeta natočí na daný bod.

5.2. Značky a bubliny

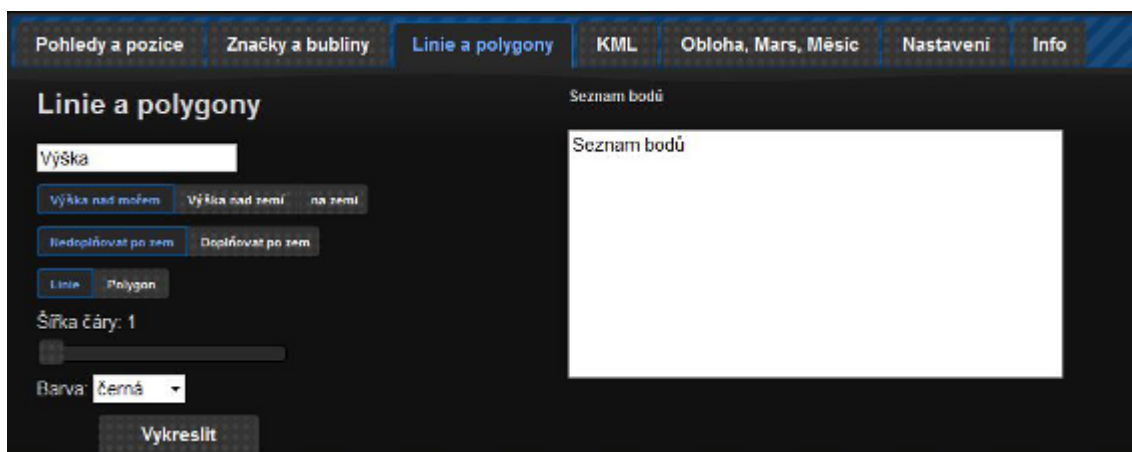
Podmenu Značky a bubliny umožňuje přidávat na planetu značky které po rozkliknutí zobrazí text bubliny. Značky mohou mít různou velikost vzhledem k najetí kurzoru myši. Pomocí komponent je možné nastavit jméno značky, velikost, typ bubliny a samotný text bubliny. Rozdíl v typech bublin je, že [KML](#)



Obrázek 7. Menu značek

(základní) bublina lze převést na KML, oproti ní pokročilá bublina nelze převést na KML, ale může obsahovat HTML kód který se zobrazí.

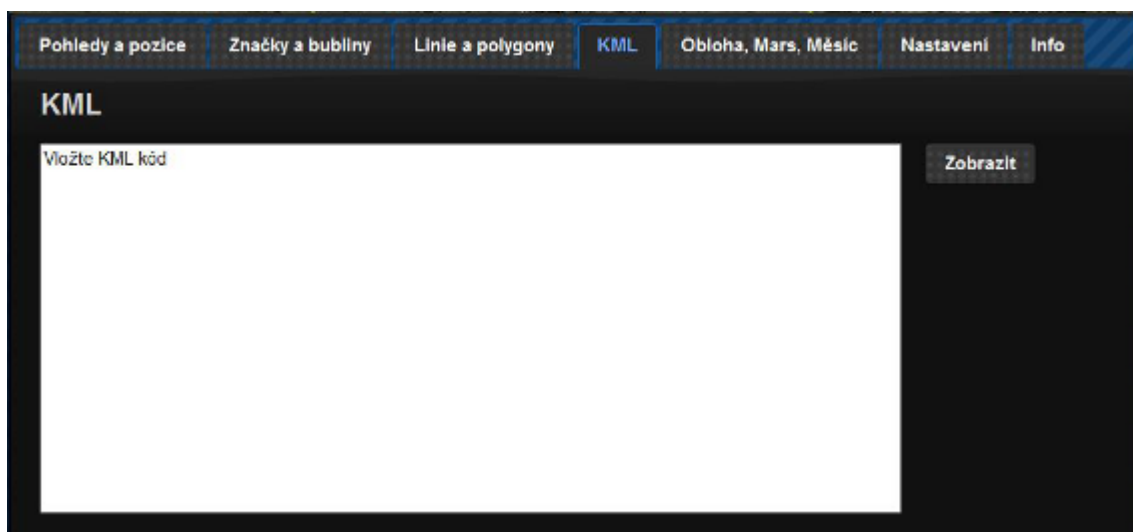
5.3. Linie a polygony



Obrázek 8. Menu linií a polygonů

Část Linie a polygony umožňuje pomocí rozhraní přidávat na planetu různé geometrické obrazce. V levé části obrazovky se nachází komponenty pro nastavení výšky, nastavení typu počítání výšky, nastavení doplnění po zemi (*extrude*), volbu mezi linií a polygonem (při tvorbě polygonu navíc volbu mezi vnitřním a venkovním bodem), nastavení šířky čáry 1–10 a volbu barvy čáry. Na pravé straně se nachází seznam bodů, které se přidávají kliknutím na glóbus. Seznam se automaticky vymaže při přidání bodu, nebo změně vykreslovaných objektů z linie na polygon a obráceně. Po stisknutí tlačítka vykreslit se daný obrazec zobrazí na glóbu.

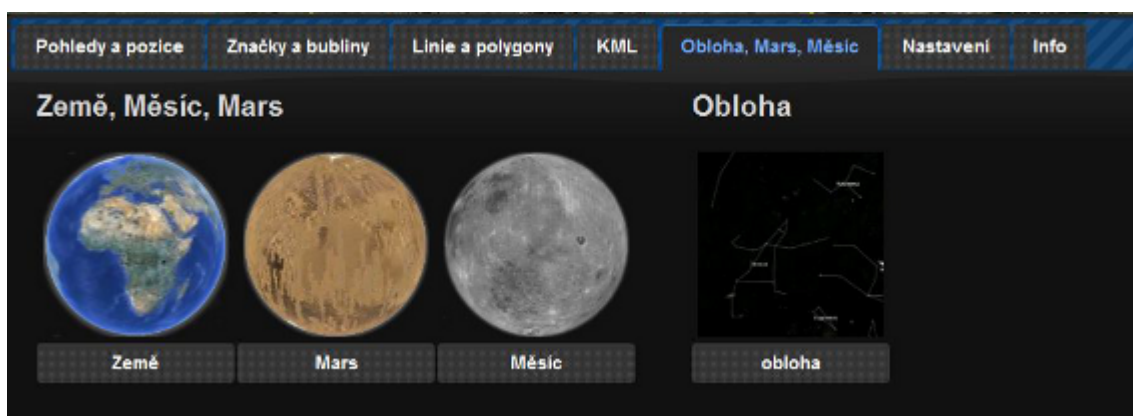
5.4. KML



Obrázek 9. Menu KML

V KML části se na levé straně nachází velké textové pole do kterého se zobrazuje KML přidávaných objektů a do kterého je možné přidat vlastní KML řetězec a zobrazit ho pomocí tlačítka "Zobrazit".

5.5. Obloha, Měsíc, Mars



Obrázek 10. Menu planet

V této sekci se nachází pouze 4 tlačítka s obrázky, které změny zobrazovanou planetu na uvedenou. Změnou planety se smažou všechny objekty(značky, linie ...), které se nacházely na předchozí planetě.

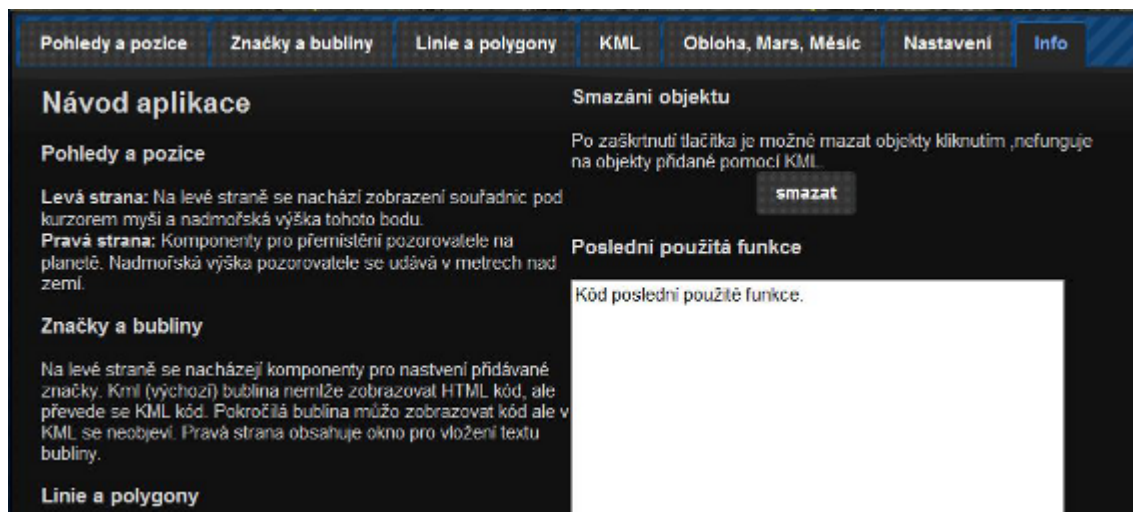
5.6. Nastavení pluginu



Obrázek 11. Menu nastavení

V levé části nastavení se nachází nastavení pluginu umožňující zobrazovat a skrývat ovládací prvky pluginu popsané v [uživatelské příručce](#) a [dokumentaci](#). Na pravé straně se nachází nastavení vrstev také popsané v [uživatelské příručce](#) a [dokumentaci](#). Z možností zobrazení, má mimo Zemi smysl používat pouze volbu terénu a Slunce.

5.7. Info



Obrázek 12. Menu nápovědy a mazání

Tato část obsahuje na levé straně velmi stručný návod pro práci s pluginem a na straně pravé část pro mazání objektů a část zobrazující kód. V Části pro

mazání objektů se nachází zaškrtnutí (checkbox), při jeho zaškrtnutí je možné mazat objekty na planetě kliknutím myši. Druhá část na pravé straně zobrazuje kód naposled použité funkce.

6. Programové rozhraní pluginu Google Earth

Tato kapitola stručně popisuje základní funkcionalitu z hlediska programátora. Kapitola obsahuje ukázky v HTML a JavaScriptu. Pro správné pochopení této kapitoly se předpokládá alespoň základní znalost HTML, JavaScriptu, a anglického jazyka. Podrobná dokumentace rozhraní k nalezení zde [2].

6.1. Vložení pluginu na stránku

Pro vložení pluginu na internetovou stránku je nutné načíst plugin v hlavičce pomocí scriptu `<script type="text/javascript" src="https://www.google.com/jsapi"> </script>` a vytvořit pro plugin element DIV s jednoznačným identifikátorem ID a libovolnou velikostí (doporučuji 200x200 pixelů a větší), například `<div id="map3d" style="height: 400px; width: 600px;"></div>`. Dále je nutné po načtení stránky (zajišťuje funkce `google.setOnLoadCallback(init)`) zavolat funkci `google.earth.createInstance('map3d', initCB, failureCB)` kde 'map3d' reprezentuje jednoznačný identifikátor ID. Tato funkce zobrazí plugin do patřičného DIVu a pokud tato akce proběhne úspěšně, zavolá funkci `initCB`. V této funkci se do proměnné `ge` uloží instance pluginu a zviditelní plugin v DIVu. Viz. následující kód.

```
<html>
<head>
  <title>Sample</title>
  <script type="text/javascript" src="https://www.google.com/jsapi">
  </script>
  <script type="text/javascript">
    var ge;
    google.load("earth", "1");
    function init() {
      google.earth.createInstance('map3d', initCB, failureCB);
    }
    function initCB(instance) {
      ge = instance;
      ge.getWindow().setVisibility(true);
    }
    function failureCB(errorCode) {
```

```

    }
    google.setOnLoadCallback(init);
</script>

</head>
<body>
  <div id="map3d" style="height: 400px; width: 600px;"></div>
</body>
</html>

```

6.2. Ovládání kamery

K ovládání kamery se využívá třída `KmlLookAt`. Získání stávajícího pohledu se provádí pomocí funkce `ge.getView().copyAsLookAt(kmlAltitudeMode)`. Nastavení nového pohledu provedeme funkcí `ge.getView().setAbstractView(lookAt)`. Pomocí funkcí popsaných v `KmlLookAt` je možné vytvořit téměř libovolný pohled na danou planetu. Jedinné omezení nasává při pozorování oblohy, kde není možné nastavit hodnotu natočení s osou země (tilt).

```

// uložení stávajícího pohledu
var lookAt = ge.getView().copyAsLookAt(ge.ALTITUDE_RELATIVE_TO_GROUND);

// natočení kolem obou os a oddálení pohledu
lookAt.setHeading(lookAt.getHeading() + 45.0);
lookAt.setTilt(lookAt.getTilt() + 45.0);
lookAt.setRange(lookAt.getRange() * 2.0);

// aplikování pohledu
ge.getView().setAbstractView(lookAt);

```

Ukázka změny stávajícího pohledu

6.3. Značky (Placemarks)

Značka (anglicky Placemark) je ikona s textem určená bodem. Značka je reprezentována třídou `KmlPlacemark`. Značku je možné vytvořit dvěma způsoby a to pomocí funkcí, poskytovaných rozhraním, nebo vytvořením KML řetězce a jeho přeložením.

6.3.1. Vytvoření pomocí funkcí:

Značka se vytváří pomocí funkce `ge.createPlacemark(string id)` a na glóbus se přidává pomocí funkcí `ge.getFeatures.appendChild(placemark)`. Pomocí

funkcí třídy `KmlFeautre` můžeme značce nastavit parametry zobrazení. Značka se zobrazí jako ikona se jménem (parametr `name` třídy `KmlFeature`) a po kliknutí se zobrazí bublina s textem (parametr `description` třídy `KmlFeature`). Značce lze také změnit ikonu, její velikost a tyto vlastnosti při události přejetí myši, viz následující kód.

```
// vytvoření značky
var placemark = ge.createPlacemark('');
placemark.setName("placemark");
placemark.setDescription("placemark");

// nastavení pozice
var point = ge.createPoint('');
point.setLatitude(12.345);
point.setLongitude(54.321);
placemark.setGeometry(point);

// vytvoření stylů
var styleMap = ge.createStyleMap('');

// vytvoření normálního stylu
var normalStyle = ge.createStyle('');
var normalIcon = ge.createIcon('');
normalIcon.setHref(
  'http://maps.google.com/mapfiles/kml/paddle/red-circle.png');
normalStyle.getIconStyle().setIcon(normalIcon);

// vytvoření stylu pro událost přejetí myši
var highlightStyle = ge.createStyle('');
var highlightIcon = ge.createIcon('');
highlightIcon.setHref(
  'http://google-maps-icons.googlecode.com/files/girlfriend.png');
highlightStyle.getIconStyle().setIcon(highlightIcon);
highlightStyle.getIconStyle().setScale(5.0);

styleMap.setNormalStyle(normalStyle);
styleMap.setHighlightStyle(highlightStyle);

// aplikování stylů
placemark.setStyleSelector(styleMap);

//přidání značky na planetu
ge.getFeatures().appendChild(placemark);
```

6.3.2. Vytvoření pomocí KML

Pro vytvoření značky za pomoci KML nám stačí pouze vytvořit textový řetězec obsahující informace o dané značce a použít na tento řetězec funkci `ge.getFeatures().appendChild(string)`. Následující řetězec reprezentuje stejnou značku, jako značka z [předešlého odstavce](#)

```
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml"
<Placemark>
<name>placemark</name>
<description>placemark</description>
<StyleMap>
<Pair>
<key>normal</key>
<Style>
<IconStyle>
<Icon>
<href>http://maps.google.com/mapfiles/kml/paddle/red-circle.png</href>
</Icon>
</IconStyle>
</Style>
</Pair>
<Pair>
<key>highlight</key>
<Style>
<IconStyle>
<scale>5</scale>
<Icon>
<href>http://google-maps-icons.googlecode.com/files/girlfriend.png</href>
</Icon>
</IconStyle>
</Style>
</Pair>
</StyleMap>
<Point>
<coordinates>54.32100000000001,12.345,0</coordinates>
</Point>
</Placemark>
</kml>
```

6.4. Bubliny (Balloons)

Bubliny slouží jako okno s doplňujícími informacemi objektů značek. Bublínám lze nastavit různé parametry popsané v kapitole [GEAbstractBalloon](#) a

[GEHtmlBalloon](#). Plugin google earth obsahuje tři druhy bublin popsané dále v kapitole.

6.4.1. Základní bublina

Základní bublina nemusí být nijak vytvořena, pokud značce přidáme funkcí `setDescription(string)` popis. Tak právě tento popis se automaticky zobrazí v okně po kliknutí na značku. Pokud značka nemá popis bude zobrazeno její jméno (pokud neobsahuje jméno nebude zobrazovat nic). Základní bublina nemůže obsahovat HTML a CSS. Značka vytvořená v kapitole [značky](#) obsahuje právě tuto bublinu.

6.4.2. HTML string bublina

HTML string bublina je pokročilou bublinou. Může obsahovat HTML, CSS a JavaScript. Bublinu lze vytvořit pomocí funkce `createHtmlStringBalloon()`. Více možností nastavení v kapitole [GEAbstractBalloon](#) a [GEHtmlBalloon](#).

```
google.earth.addListener(placemark, 'click', function (event) {
    event.preventDefault();

    var balloon = ge.createHtmlStringBalloon('');
    balloon.setFeature(placemark);
    balloon.setContentString(text);
    ge.setBalloon(balloon);
});
```

Ukázka přidání bubliny značce placemark.

6.4.3. HTML DIV bublina

HTML DIV bublina má stejné možnosti jako [předešlá](#) s tím rozšířením, že na objekty v DIV bublině se vztahuje i globální CSS soubor a je k těmto objektům možné přistupovat z jiné části aplikace a stránky. Vytváří se pomocí funkce `createHtmlDivBalloon()`. Více možností nastavení v kapitole [GEAbstractBalloon](#) a [GEHtmlBalloon](#).

```
google.earth.addListener(placemark, 'click', function (event) {
    event.preventDefault();

    var balloon = ge.createHtmlDivBalloon('');
    balloon.setFeature(placemark);
    balloon.setContentString(text);
    ge.setBalloon(balloon);
});
```

Ukázka přidání bubliny znače placemark.

6.5. Linie a polygony

Pomocí API lze na planetu vykreslit libovolné linie a polygony různých barev a šířek. Tyto objekty se vytváří podobně jako značky, tedy vytvořením objektu `KmlPlacemark`, ale místo jednoho bodu obsahují seznam tvořený třídou `KmlLineString` nebo `KmlPolygon`. Těmto třídám je možné nastavit parametry doplnění po zem(extrude) a kopírování povrchu(tessellate). Dále můžeme těmto objektům změnit barvu, šířku a průhlednost. Stejně jako značky můžou i tyto objekty vlastnit `bubliny` a mohou být vytvořeny pomocí KML.

6.5.1. Vytvoření linie

Pro vytvoření přímky vyrobíme objekt `značky KmlPlacemark`. Dalším krokem je vytvoření seznamu bodů, které bude reprezentovat třída `KmlLineString` a přidáme je znače funkcí `setGeometry(lineString)`. Takto vytvořená linie již může být přidána na planetu funkcí `ge.getFeatures().appendChild(placemark)` viz. příklad.

```
// Vytvoření značky
var lineStringPlacemark = ge.createPlacemark('');

// Vytvoření seznamu bodů
var lineString = ge.createLineString('');
// Přidání bodů znače
lineStringPlacemark.setGeometry(lineString);

// Přidání jednotlivých bodů
lineString.getCoordinates().pushLatLngAlt(48.754, -121.835, 0);
lineString.getCoordinates().pushLatLngAlt(48.764, -121.828, 0);
lineString.getCoordinates().pushLatLngAlt(48.776, -121.818, 0);
lineString.getCoordinates().pushLatLngAlt(48.787, -121.794, 0);
lineString.getCoordinates().pushLatLngAlt(48.781, -121.778, 0);
lineString.getCoordinates().pushLatLngAlt(48.771, -121.766, 0);
lineString.getCoordinates().pushLatLngAlt(48.757, -121.768, 0);
lineString.getCoordinates().pushLatLngAlt(48.747, -121.773, 0);

// Přidání značky na zemi
ge.getFeatures().appendChild(lineStringPlacemark);
```

Ukázka vytvoření linie

6.5.2. Vytvoření polygonu

Vytvoření polygonů probíhá stejně jako vytvoření linií z předešlé kapitoly, s jedinou výjimkou, a to, že seznam bodů je reprezentovaný třídou `KmlPolygon`. Takto reprezentovaný polygon může obsahovat vnitřní výřez.

```
// Vytvoření polygonu
var polygon = ge.createPolygon('');
// Přidání bodů značce
polygonPlacemark.setGeometry(polygon);

// Přidání venkovních bodů
var outer = ge.createLinearRing('');
outer.getCoordinates().pushLatLngAlt(48.80, -121.80, 700);
outer.getCoordinates().pushLatLngAlt(48.80, -121.90, 700);
outer.getCoordinates().pushLatLngAlt(48.70, -121.90, 700);
outer.getCoordinates().pushLatLngAlt(48.70, -121.80, 700);
polygon.setOuterBoundary(outer);

// Přidání vnitřních bodů
var inner = ge.createLinearRing('');
inner.getCoordinates().pushLatLngAlt(48.77, -121.83, 700);
inner.getCoordinates().pushLatLngAlt(48.77, -121.87, 700);
inner.getCoordinates().pushLatLngAlt(48.73, -121.87, 700);
inner.getCoordinates().pushLatLngAlt(48.73, -121.83, 700);
polygon.getInnerBoundaries().appendChild(inner);
```

Vytvoření třídy `KmlPolygon` a její přidání značce

6.5.3. Styly

Liniím i polygonům lze měnit šířku, barvu a průhlednost čáry pomocí funkcí `setWidth(int)` a `getColor().set('OORRGGBB')` (OO reprezentuje průhlednost). Tyto funkce se musí aplikovat na styl značky získaný funkcí `getStyleSelector().getLineStyle()`.

```
// Vytvoření stylu, přidání značce a nastavení šířky a barvy.
polygonPlacemark.setStyleSelector(ge.createStyle(''));
var lineStyle = polygonPlacemark.getStyleSelector().getLineStyle();
lineStyle.setWidth(5);
lineStyle.getColor().set('9900ffff');
```

Ukázka vytvoření a přidání stylu

6.6. Vlastní textury (Overlays)

Prostřednictvím rozhraní je také možné překrývat plochu zeměkoule vlastními obrázky. Takto vložené obrázky budou kopírovat plochu terénu a budou překrývat originální texturu. Texturu lze vložit pomocí pár příkazů, viz. následující příklad.

```
// Vytvoření textury překryvu
var groundOverlay = ge.createGroundOverlay('');

// vytvoření obrázku a jeho přidání textuře
var icon = ge.createIcon('');
icon.setHref("http://www.google.com/logos/earthday08.gif");
groundOverlay.setIcon(icon);

// vytvoření pozice pro obrázek (šířka, šířka, délka délka, natočení)
var latLonBox = ge.createLatLonBox('');
latLonBox.setBox(48.80, 48.75, -121.77, -121.85, 0);
groundOverlay.setLatLonBox(latLonBox);

// přidání textury na planetu
ge.getFeatures().appendChild(groundOverlay);
```

6.7. KML

Pro aplikování KML řetězce na planetu stačí převést řetězec na objekt typu `KmlObject` pomocí funkce `ge.parseKml(kmlString)` a poté přidat tento objekt na planetu funkcí `ge.getFeatures().appendChild(kmlObject)`. V praxi se používá ještě funkce `kmlObject.getAbstractView()` pro zjištění, zda obsahuje KML objekt pohled, a případně může následovat kód pro přesunutí na tuto pozici. Plugin také umí převést daný objekt na KML řetězec pomocí funkce `kmlObject.getKml()`. Více v následujících ukázkách.

```
var kmlString = kmltext;
var kmlObject = ge.parseKml(kmlString);
ge.getFeatures().appendChild(kmlObject);
if (kmlObject.getAbstractView()) ge.getView().setAbstractView(kmlObject.getAbs
```

Ukázka aplikování KML

```
var placemark = ge.createPlacemark('');
var point = ge.createPoint('');
point.setLatitude(12.345);
point.setLongitude(54.321);
placemark.setGeometry(point);
```



```
ge.getFeatures().appendChild(placemark);
var kml = placemark.getKml();
```

Ukázka získání KML řetězce objektu značky

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/km
<Placemark>
<Point>
<coordinates>54.32100000000001,12.345,0</coordinates>
</Point>
</Placemark>
</kml>
```

Ukázka získaného KML

6.8. Obloha, Mars, Měsíc (Sky, Mars, Moon)

Přepnutí na oblohu se provádí pomocí jedné funkce a to `ge.getOptions().setMapType(ge.MAP_TYPE_SKY)`. Volání je možné provést z kterékoliv zobrazené planety. Před zobrazením Země, Marsu nebo Měsíce je nutné nejdříve vymazat obsah DIVu, kde se nachází zobrazení pluginu a poté přidat do DIVu vybraný objekt podobně jako při [vlození pluginu na stránku](#). Při přepnutí na planetu se vytvoří její nová instance, takže nebude obsahovat žádné prvky, které se nacházely na předchozí planetě.

```
document.getElementById('map3d').innerHTML = '';
google.earth.createInstance('map3d', initCB, failureCB, {
database: 'http://khmdb.google.com/?db=mars'
});
```

Ukázka zobrazení planety Mars

6.9. Nastavení pluginu (Options)

Nastavení pluginu probíhá nastavováním parametrů třídy `GeOption`. Třída obsahuje pravdivostní hodnoty o zobrazení prvků (status bar, panel ovládání kamery, mřížka, minimapka, měřítko a atmosféra). Pro nastavení se používají funkce jako `option.setStatusBarVisibility(bool)`. Další funkce jsou v následujícím příkladu. Další možnosti nastavení jsou zobrazit panel času, který umožňuje měnit stáří obrázků, ze kterých je planeta poskládána. Pro některá místa planety existuje pouze omezené množství časů pořízení obrázků. Posledním možností je zakázat výchozí ovládání pohybu planety myší. Všechny zmíněné možnosti se nacházejí v následujícím příkladu.

```

var option = ge.getOptions();

option.setStatusBarVisibility(true);
option.setGridVisibility(true);
option.setOverviewMapVisibility(false);
option.setScaleLegendVisibility(false);
option.setAtmosphereVisibility(true);
option.setMouseNavigationEnabled(true);

ge.getNavigationControl().setVisibility(true);
ge.getTime().setHistoricalImageryEnabled(false);

```

Ukázka nastavení pluginu

6.10. Nastavení vrstev (layers)

Nastavení zobrazených vrstev jako stromy, budovy, lesy, atmosféra, terén a cesty probíhá pomocí funkce `ge.getLayerRoot().enableLayerById(ge.LAYER_BORDERS, bool)`, kde hodnota `ge.LAYER_BORDERS` může být nahrazena jinými, uvedenými v následujícím příkladu. Zobrazení Slunce probíhá funkcí `ge.getSun().setVisibility(bool)`. V následujícím příkladu jsou uvedeny všechny možné vrstvy.

```

ge.getLayerRoot().enableLayerById(ge.LAYER_BORDERS, true);
ge.getLayerRoot().enableLayerById(ge.LAYER_BUILDING, );
ge.getLayerRoot().enableLayerById(ge.LAYER_BUILDINGS_LOW_RESOLUTION, );
ge.getLayerRoot().enableLayerById(ge.LAYER_ROADS, );
ge.getLayerRoot().enableLayerById(ge.LAYER_TERRAIN, );
ge.getLayerRoot().enableLayerById(ge.LAYER_TREES, );

ge.getSun().setVisibility($('#radiol71').is(':checked'));

```

Ukázka nastavení vrstev

6.11. Události (Events)

Rozhraní Google Earth umožňuje používat události jako `click`, `dblclick`, `mouseover`, `mousedown`, `mouseup`, `mouseout` a `mousemove`. Dále události pohledu `viewchangebegin`, `viewchangeend` a `viewchange`. Události mohou být přidávány většině geometrických objektů, pohledu, nebo glóbu. Veškeré přidané události lze také zrušit. Pomocí těchto základních událostí je možné vytvořit si i vlastní pokročilé, jako `mousedrag` a další.

```

var eventHandler = function(){ ... };

// Přidání události
google.earth.addEventListener(ge.getGlobe(), 'mousemove', eventHandler);
// Odebrání události
google.earth.removeEventListener(ge.getGlobe(), 'mousemove', eventHandler);

```

Ukázka přidání a odebrání události, u odebírání je nutné zachovat stejné parametry jako při přidávání

Rozhraní také umožňuje přidat události i objektům mimo Google Earth plugin. V následujícím příkladu je ukázka takového kódu, ovšem není možné zaručit funkčnost u všech prohlížečů.

```

function addDomListener(element, eventName, listener) {
  if (element.addEventListener) // většina prohlížečů
    element.addEventListener(eventName, listener, false);
  else if (element.attachEvent) // IE
    element.attachEvent('on' + eventName, listener);
}

```

Vzor pro přidávání událostí

7. Dokumentace tříd

Tato kapitola popisuje základní třídy aplikace Google Earth, jejich parametry a jejich funkce. Mezi základní třídy patří bod, čára, polygon, výčtový typ pro zadávání výšky a další. Tyto třídy není možné přímo zobrazit ale jsou nepostradatelné pro funkčnost celého rozhraní.

7.1. KmlObject

Třída `KmlObject` je jednou z hlavních tříd celého rozhraní. Z této abstraktní třídy dědí všechny ostatní třídy, mimo třídy pro nastavení okna. Třída obsahuje jednoznačný identifikátor ID typu string.

parametry

- **string ID** – jednoznačný identifikátor objektu

funkce

- **string getType()** – vrací typ objektu
- **bool equals(KmlObject compareTo)** – vrací pravdivostní hodnotu porovnání dvou objektů
- **string getId()** – vrací ID objektu
- **KmlObject KmlObject.getParentNode()** – vrací rodiče objektu
- **void KmlObject.release()** – smaže objekt a uvolní jeho ID

7.2. KmlFeature

Abstraktní Třída KmlFeature dědí z třídy KmlObject a obsahuje základní vlastnosti prvku.

parametry

- **string name** – uživatelem definované jméno objektu
- **bool visibility** – udává viditelnost objektu
- **string address** – nestrukturovaná adresa (ulice, město, stát, směrovací číslo)
- **string snippet** – text určený pro stručný popis vlastností
- **string description** – text určený pro popis objektu
- **KmlAbstractView abstractView** – proměnná pro uchování třídy pohledu
- **KmlStyleSelector styleSelector** – udává styl objektu KmlPlacemark
- **float opacity** – udává průhlednost objektu (0 = průhledný, 1 = plně viditelný)

funkce

- **void setName(string name)** – nastaví jméno objektu
- **string getName()** – vrací jméno objektu
- **void setVisibility(bool visibility)** – nastaví viditelnost objektu
- **bool getVisibility()** – vrací viditelnost objektu

- **void setSnippet(string snippet)** – nastaví hodnotu snippet
- **string getSnippet()** – vrací hodnotu snippet
- **void setDescription(string description)** – nastaví popis objektu
- **string getDescription()** – vrací popis objektu
- **void setAbstractView(KmlAbstractView abstractView)** – nastaví proměnnou abstractView
- **KmlAbstractView getAbstractView()** – vrací objekt abstractView
- **void setStyleSelector(KmlStyleSelector styleSelector)** – nastaví styl objektu
- **KmlStyleSelector getStyleSelector()** – vrací styl objektu
- **void setOpacity(float opacity)** – nastaví hodnotu opacity
- **float getOpacity()** – vrací hodnotu opacity
- **string getKml()** – vrací Kml řetězec objektu

7.3. KmlLookAt

KmlLookAt je třída reprezentující pohled. Obsahuje parametry pro nastavení

Parametry

- **double latitude** – zeměpisná šířka středu pohledu
- **double longitude** – zeměpisná délka středu pohledu
- **double heading** – směr natočení ve stupních kolem osy země (výchozí hodnota 0 sever), nabývá 0-360
- **double tilt** – směr natočení ve stupních s osou země (0 pohled dolů, 90 pohled na horizont)
- **double range** – vzdálenost pohledu od bodu
- **double altitude** – vzdálenost pohledu od země
- **KmlAltitudeModeEnum altitudeMode** – výčtový typ výšky ([KmlAltitudeModeEnum](#))

funkce

- **void set(double latitude, double longitude, double altitude, KmlAltitudeModeEnum altitudeMode, double heading, double tilt, double range)** – nastaví všechny parametry pohledu
- **double getLatitude()** – vrátí zeměpisnou délku
- **void setLatitude(double latitude)** – nastaví zeměpisnou délku
- **double getLongitude()** – vrátí zeměpisnou šířku
- **void setLongitude(double longitude)** – nastaví zeměpisnou šířku
- **double getRange()** – vrátí vzdálenost od bodu
- **void setRange(double range)** – nastaví vzdálenost od bodu
- **double getTilt()** – vrátí natočení kolem osy
- **void setTilt(double tilt)** – nastaví natočení kolem osy
- **double getHeading()** – vrátí natočení s osou
- **void setHeading(double heading)** – nastaví natočení s osou
- **double getAltitude()** – vrátí výšku od země
- **void setAltitude(double altitude)** – nastaví výšku od země
- **KmlAltitudeModeEnum getAltitudeMode()** – vrátí způsob vnímání výšky
- **void setAltitudeMode(KmlAltitudeModeEnum altitudeMode)** – nastaví způsob vnímání výšky

7.4. KmlAltitudeModeEnum

`KmlAltitudeModeEnum` je výčetový typ nabývající různých hodnot, podle kterých se odvíjí výška objektu.

- **ALTITUDE_CLAMP_TO_GROUND** – objekt se budou nacházet přímo na zemi (nezáleží na výšce), pokud se v daném místě nenachází země nad hladinou, objekt bude na hladině
- **ALTITUDE_RELATIVE_TO_GROUND** – objekt se bude nacházet danou výšku nad/pod zemí v daném bodě

- **ALTITUDE_ABSOLUTE** – objekt bude v nadmořské výšce
- **ALTITUDE_CLAMP_TO_SEA_FLOOR** – podobně jako u **ALTITUDE_CLAMP_TO_GROUND**, objekt bude na mořském dně (pokud je pro danou pozici dostupné), pokud v daném bodě není moře chová se stejně jako **ALTITUDE_CLAMP_TO_GROUND**
- **ALTITUDE_RELATIVE_TO_SEA_FLOOR** – podobně jako u **ALTITUDE_RELATIVE_TO_GROUND**, ale počítá se od mořského dna

7.5. KmlCoordArray

Třída reprezentující seznam souřadnic (tříd **KmlCoord**). Třída **KmlCoord** je stejná jako třída **KmlPoint** ale nedědí z třídy **KmGeometry**, tudíž ji není možné nastavit parametr **altitudeMode**.

parametry

- **KmlCoord coords** – seznam souřadnic
- **int length** – délka seznamu

funkce

- **KmlCoord get(int index)** – vrací daný prvek ze seznamu
- **void set(int index, KmlCoord coord)** – nastaví souřadnice na daném prvku
- **void setLatLngAlt(int index, double latitude, double longitude, double altitude)** – nastaví seznam na jeden prvek obsahující dané souřadnice
- **int pushLatLngAlt(double latitude, double longitude, double altitude)** – přidá na konec seznamu nový prvek a vrátí novou délku seznamu
- **KmlCoord pop()** – vrátí poslední prvek seznamu, smaže jej a zmenší délku seznamu
- **int unshiftLatLngAlt(double latitude, double longitude, double altitude)** – přidá na začátek seznamu nový prvek a vrátí novou délku seznamu
- **KmlCoord shift()** – vrátí první prvek seznamu, smaže jej a zmenší délku seznamu

- **void reverse()** – obrátí pořadí prvků v seznamu
- **void clear()** – smaže obsah seznamu
- **int getLength()** – vrátí délku seznamu

7.6. KmlGeometry

Třída `KmlGeometry` je abstraktní třída udržující informace o pozici objektu. Z této třídy dědí třídy `KmlLineString`, `KmlPoint` a `KmlPolygon`.

parametry

- **KmlAltitudeModeEnum altitudeMode** – výčtový typ výšky (`KmlAltitudeModeEnum`)
- **bool extrude** – pravdivostní hodnota zda má být od bodu protažena čára k zemi
- **bool tessellate** – pravdivostní hodnota zda má prvek kopírovat terén (např. zda se má čára chovat jako přímka mezi dvěma danými body, nebo kopírovat terén mezi těmito body)

funkce

- **void setAltitudeMode(KmlAltitudeModeEnum altitudeMode)** – nastaví způsob vnímání výšky
- **KmlAltitudeModeEnum getAltitudeMode()** – získání způsobu vnímání nadmořské výšky
- **void setExtrude(bool extrude)** – nastavení hodnoty extrude
- **bool getExtrude()** – získání hodnoty extrude
- **void setTessellate()** – nastavení hodnoty tessellate
- **bool getTessellate()** – získání hodnoty tessellate

7.7. GESchemaObjectContainer

Abstraktní třída reprezentující seznam tříd `KmlObject`.

parametry

- **KmlObject objects** – seznam prvků `KmlObject`

funkce

- **KmlObject appendChild(KmlObject object)** – přidá prvek na konec seznamu a vrátí přidávaný prvek
- **KmlObject removeChild(KmlObject oldChild)** – odebere prvek ze seznamu a vrátí odebíraný prvek
- **KmlObject insertBefore(KmlObject newChild, KmlObject refChild)** – přidá prvek do seznamu před daný prvek, vrátí přidávaný prvek
- **KmlObject replaceChild(KmlObject newChild, KmlObject oldChild)** – přidá prvek do seznamu místo daného prvku, vrátí starý prvek
- **bool hasChildNodes()** – vrací pravdu pokud seznam není prázdný
- **KmlObject getFirstChild()** – vrací první prvek
- **KmlObject getLastChild()** – vrací poslední prvek

7.8. KmlPoint

KmlPoint je třída reprezentující bod a dědí z třídy **KmlGeometry**. Vytváří se pomocí funkce `ge.createPoint(string id)`. Pomocí `id` lze k bodu přistupovat.

parametry

- **double latitude** – zeměpisná šířka ve stupních
- **double longitude** – zeměpisná délka ve stupních
- **double altitude** – nadzemská/nadmořská (dle proměnné `KmlAltitudeModeEnum`) výška v metrech

funkce

- **void setLatLng(double latitude, double longitude)** – nastaví zeměpisnou šířku a zeměpisnou výšku
- **void setLatLngAlt (double latitude, double longitude, double altitude)** – nastaví zeměpisnou šířku, zeměpisnou výšku, a nadzemskou/nadmořskou výšku
- **void setLatitude(double latitude)** – nastavení zeměpisné šířky
- **double getLatitude()** – získání zeměpisné šířky

- **void setLongitude(double longitude)** – nastavení zeměpisné výšky
- **double getLongitude()** – získání zeměpisné výšky
- **void setAltitude(double altitude)** – nastavení výšky
- **double getAltitude()** – získání výšky

7.9. KmlLineStyle

Třída `KmlLineStyle` reprezentuje čáru a dědí z třídy `KmlGeometry`. Vytváří se pomocí funkce `ge.createLineStyle(string id)`. Pomocí `id` lze k bodu přistupovat.

parametry

- **KmlCoordArray coordinates** – seznam souřadnic reprezentovaný pomocí `KmlCoordArray`
- **double AltitudeOffset** – hodnota o kterou budou navýšeny všechny souřadnice v seznamu

funkce

- **KmlCoordArray KmlLineStyle.getCoordinates()** – vrací ukazatel na seznam souřadnic
- **void KmlLineStyle.setAltitudeOffset(double altitudeOffset)** – nastaví hodnotu `AltitudeOffset`
- **double getAltitudeOffset()** – vrátí hodnotu `AltitudeOffset`

7.10. KmlPolygon

Třída `KmlPolygon` reprezentuje polygon a dědí z třídy `KmlGeometry`. Vytváří se pomocí funkce `ge.createPolygon(string id)`. Pomocí `id` lze k bodu přistupovat.

parametry

- **KmlLinearRing outerBoundary** – venkovní souřadnice polygonu. Třída `KmlLinearRing` dědí z třídy `KmlLineStyle`.
- **GELinearRingContainer innerBoundaries** – seznam objektů třídy `KmlLinearRing` (třída `GELinearRingContainer` dědí z třídy `GESchemaObjectContainer`)

funkce

- **void setOuterBoundary(KmlLinearRing outerBoundary)** – nastaví proměnnou `outerBoundary`
- **KmlLinearRing getOuterBoundary()** – vrátí hodnotu `outerBoundary`
- **GELinearRingContainer getInnerBoundaries()** – vrátí hodnotu `innerBoundaries`

7.11. KmlPlacemark

Třída `KmlPlacemark` obsahuje pouze jednu proměnnou typu `KmlGeometry` a funkce pro její nastavení. Třída `KmlPlacemark` dědí z třídy `KmlFuture`, která obsahuje další parametry jako jméno, popis, styl,...

parametry

- **KmlGeometry geometry** – obsahuje souřadnice bodu

funkce

- **KmlPlacemark createPlacemark(string id)** – vytvoří objekt třídy `KmlPlacemark`
- **void setGeometry(KmlGeometry geometry)** – nastaví objekt `geometry`
- **KmlGeometry getGeometry()** – vrátí objekt `geometry`

7.12. GEAbstractBalloon

`GEAbstractBalloon` reprezentuje abstraktní třídu, ze které dědí všechny třídy `bublin`, zobrazitelné v pluginu. Třída obsahuje základní vlastnosti a parametry `bubliny`.

parametry

- **string ID** – jednoznačný identifikátor objektu
- **int minWidth** – minimální šířka `bubliny`
- **int minHeight** – minimální výška `bubliny`
- **int maxWidth** – maximální šířka `bubliny`
- **int maxHeight** – maximální výška `bubliny`
- **bool closeButtonEnabled** – zapnutí/vypnutí zavíracího tlačítka

funkce

- **string getId ()** – vrací ID bubliny
- **void setId (string id)** – nastaví ID bublině
- **KmlFeature getFeature ()** – vrací ukazatel na prvek který obsahuje bublinu
- **void setFeature (KmlFeature feature)** – nastaví prvek bublině
- **int getMinWidth ()** – vrací minimální šířku
- **void setMinWidth (int minWidth)** – nastaví minimální šířku
- **int getMinHeight ()** – vrací minimální výšku
- **void setMinHeight (int minHeight)** – nastaví minimální výšku
- **int getMaxWidth ()** – vrací maximální šířku
- **void setMaxWidth (int maxWidth)** – nastaví maximální šířku
- **int getMaxHeight ()** – vrací maximální výšku
- **void setMaxHeight (int maxHeight)** – nastaví maximální výšku
- **bool getCloseButtonEnabled ()** – vrací hodnotu closeButtonEnabled
- **void setCloseButtonEnabled (bool closeButtonEnabled)** – nastavení hodnoty closeButtonEnabled

7.13. GEHtmlBalloon

Abstraktní třída `GEHtmlBalloon` dědí z třídy `GEAbstractBalloon` a obsahuje další vlastnosti týkající se obsahu bublin. Z této třídy dědí třídy `GEHtmlDivBalloon`(`getContentDiv`, `setContentDiv`) a `GEHtmlStringBalloon`(`getContentString`, `setContentString`). Funkce uvedené v závorkách slouží pro nastavení a získání řetězce bubliny. Typy bublin jsou popsány v kapitole [bubliny](#).

parametry

- **string foregroundColor** – barva popředí (formát `#RRGGBB`)
- **string backgroundColor** – barva pozadí (formát `#RRGGBB`)

funkce:

- **string getForegroundColor ()** – vrátí barvu popředí
- **void setForegroundColor (string foregroundColor)** – nastaví barvu popředí
- **string getBackgroundColor ()** – vrátí barvu pozadí
- **void setBackgroundColor (string backgroundColor)** – nastaví barvu pozadí

Závěr

Cílem práce bylo vytvoření webové aplikace demonstrující uživatelské rozhraní pluginu Google Earth a její umístění na internet společně s uživatelskou příručkou, což bylo splněno. Dále bylo úkolem vytvoření stručné dokumentace programového rozhraní aplikace Google Earth plugin, kterou práce obsahuje. Do budoucna by bylo možné rozšířit aplikaci i dokumentaci o další méně důležité funkce a třídy.

Conclusions

The goal was to create a web application which demonstrates user interface of the Google Earth plugin and its placement on the internet along with the users' guide, which was fulfilled. The next task was to create a concise documentation of the Google Earth plugin user interface which is in the project. In the future it's possible to extend the application and the documentation with other less important functions and classes.

Reference

- [1] <https://developers.google.com/kml/documentation/>
- [2] <https://developers.google.com/earth/documentation/>
- [3] <http://www.google.com/earth/explore/products/plugin.html>

A. Obsah přiloženého CD

Část práce popisující soubory na přiloženém CD.

`bin/`

Ve složce `bin/` se nachází ZIP archiv obsahující dvě složky. Ve složce "komplet" se nachází kompletní webová aplikace funkční na iOS pokud se nachází na stránkách www.bcprojekt.wz.cz, na Windows není doména www.bcprojekt.wz.cz podmínkou (způsobeno poslední verzí pluginu). Ve složce "webzdarma" se nachází HTML soubor, který se nachází na stránkách www.bcprojekt.wz.cz a na těchto stránkách funguje pro všechny OS. Tento HTML soubor si zbytek souborů načítá ze stránek <http://phoenix.inf.upol.cz/schnaubv/>, kde se rovněž nachází kompletní aplikace (z důvodu poslední verze pluginu funkční pouze na Windows).

`doc/`

Složka obsahuje ZIP archiv s PDF textem bakalářské práce a všechny potřebné soubory pro jeho vytvoření

`src/`

Ve složce `src/` se nachází stejný soubor jako ve složce `bin/`

`readme.txt`

Instrukce pro práci se soubory stránky.