Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Editor 3D scén pro simulaci eroze

Plzeň 2015 Martin Zadražil

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 25. června 2015

Martin Zadražil

Poděkování

Chtěl bych poděkovat své vedoucí bakalářské práce Ing. Věře Skorkovské za odborné vedení, za pomoc a rady při zpracování této práce.

Abstract

This report aims to study of all existing SW application for 3D modeling to develop our own user friendly application for modeling 3D scenes for simulating an erosion. The main requirements for the application is to be able to edit the terrain model easily, to add their own metadata to the objects in the scene which are used as parameters for the erosion-simulating program (eg. coefficient of erosion) and to be able to save the metadata in a format understandable for the erosion-simulating program. After the study, the web application was created with the aim to label and shift more peaks easily in an area that is able to work with the necessary metadata and export or import a scene in agreed formats. The 3D model of the landscape around the village of Racice in the middle reaches of the Berounka river was created in this application to verify its ability to create large and complex models of contours which can be used in the erosion-simulating program afterwards.

Abstrakt

Cílem práce je prostudovat stávající software pro 3D modelování a navrhnout vlastní, uživatelsky přívětivou, aplikaci sloužící k modelování 3D scén pro simulaci eroze. Hlavní požadavky na aplikaci jsou: možnost jednoduše editovat model terénu, možnost přidat objektům ve scéně vlastní metadata, jež slouží jako parametry pro program simulující erozi (např. koeficient vymílání) a následně mít možnost scénu s těmito metadaty uložit ve formátu srozumitelném pro zmíněný program. Po prostudování dostupných programů pro 3D modelování byla vytvořena webová aplikace s důrazem na snadné označování a posun většího množství vrcholů v ploše, která umí pracovat s potřebnými metadaty a exportovat, či importovat scénu v dohodnutých formátech. V této aplikaci byl vytvořen 3D model krajiny v okolí obce Račice na středním toku řeky Berounky k ověření schopnosti programu vytvářet i velké a složité modely krajinných reliéfů, které lze následně použít v programu pro simulaci eroze.

Obsah

1 Úvod

Cílem této práce je seznámit se s existujícím programovým vybavením pro 3D modelování a následně vytvořit vlastní uživatelsky přívětivou aplikaci pro práci s 3D objekty. Aplikace bude umožňovat návrh a sestavení scény pro simulaci eroze a uložení této scény do souboru dohodnutého formátu. [Skorkovská(2012)]

V první části budu popisovat několik existujících aplikací pro modelování ve 3D běžících v prostředí GNU/Linux. U těchto programů budu posuzovat ovládání scény a transformace objektů, které je důležité pro modelování krajinného reliéfu. Naopak nebudu posuzovat práci se světlem, zrcadlením, texturami či animací neboť tyto funkce nejsou pro účel práce stěžejní. Budu tedy popisovat základní rozložení ovládacích prvků každého programu, manipulaci s celou scénou, vytváření, označování a odstraňování objektů, translaci, rotaci a škálování objektů a změnu geometrie. Cílem této části je prozkoumat různé přístupy k modelování 3D objektů.

V druhé části popíšu samotný postup vytváření aplikace, zdůvodnění, proč jsem aplikaci realizoval jako webovou, a některé implementační detaily jako nastínění struktury aplikace, popis nejdůležitějších tříd a popis použitých algoritmů.

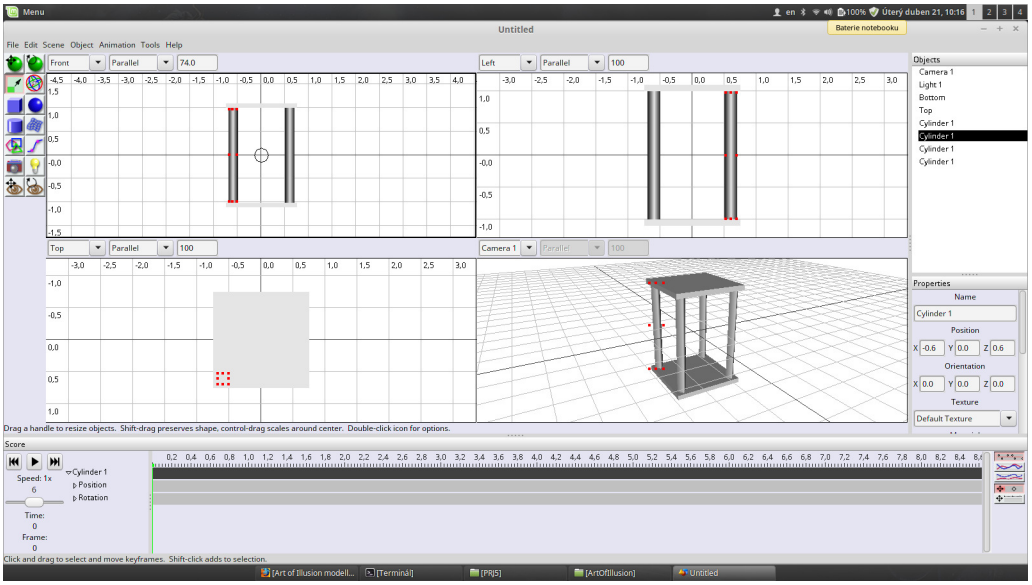
Po přečtení práce získáte základní představu možnostech modelování krajinného reliéfu v dostupných programech pro 3D modelování, a dozvíte se, jak vymodelovat a uložit krajinný reliéf v mnou vytvořené aplikaci. Zároveň získáte také představu o struktuře, možnostech a některých implementačních detailech této aplikace.

2 Existující software pro 3D modelování

2.1 Art of Illusion

Art of Illusion je softwarový balík pro 3D modelování, texturování, raytracing a další typy počítačového vykreslování napsaný převážně v jazyce *Java*. První veřejná verze vyšla 29. září 1999. Program byl vytvořen Peterem Eastmanem [aoi(a)] a je licencován jako open-source pod licencí GPL. [aoi(b)] Program je multiplatformní.

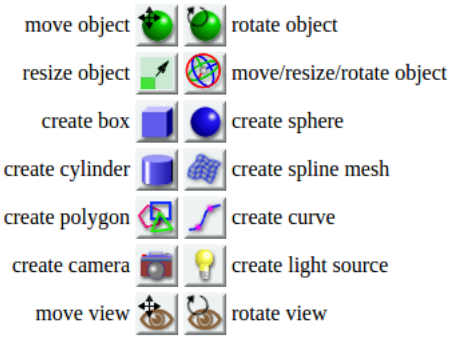
2.1.1 Pracovní plocha



Obrázek 2.1: Pracovní prostředí *Art of Illusion*

Pracovní plocha *Art of Illusion* (obr. 2.1) je rozdělená do čtyř panelů ve kterých jsou čtyři pohledy na modelovaný objekt. Tři pohledy jsou ortogonální projekcí modelovaného objektu, každý pohled ve směru jedné osy. Čtvrtý pohled je perspektivní projekcí modelovaného objektu. Vlevo se na-

chází nástrojový panel (obr. 2.2), vpravo se nachází seznam objektů na scéně a pod ním panel s vlastnostmi objektu, se kterým se právě pracuje. Ve spodní části okna se pak nachází nástroj pro vytváření animací.



Obrázek 2.2: Panel nástrojů *Art of Illusion*

2.1.2 Manipulace se scénou

Pro manipulaci se scénou slouží nástroje *Translate View* a *Rotate View*. Po kliknutí na nástroj *Translate View* lze celou scénou pohybovat podržením levého či pravého tlačítka myši a následným tažením. Otáčením kolečka myši lze přibližovat či oddalovat pohled a při stisknutí kolečka myši lze scénou rotovat. Po kliknutí na nástroj *Rotate View* lze scénou rotovat podržením levého tlačítka myši a následným tažením, posouvat podržením pravého tlačítka myši a následným tažením. Podržení kolečka myši funguje v obou módech.

Pokud je při posunu či otáčení scény držena klávesa SHIFT, probíhá úkon pouze podle jedné osy. Totéž platí i pro rotaci a translaci objektů.

2.1.3 Vložení, označení a odstranění objektu

Objekt se vkládá vybráním typu objektu z panelu nástrojů a následným tažením myši ve scéně, čímž se přibližně určí velikost budoucího objektu. Na výběr je kvádr, koule, válec, plocha definovaná křivkou, polygon a křivka. Z plochých objektů je možno vytvořit rotační tělesa.

Pokud je zapnutý některý z nástrojů pro vkládání objektů, je označený objekt vždy ten poslední vložený do scény. Pokud je používán některý z

nástrojů pro posun, rotaci nebo škálování objektu, vybere se ten objekt, na který uživatel kliknul. Objekt lze rovněž vybrat kliknutím na jeho název v seznamu objektů po pravé straně.

Objekt lze odstranit po vyvolání nabídky kliknutím pravého tlačítka myši přímo na objekt ve scéně či na jeho název v seznamu objektů a vybráním položky *Clear*.

2.1.4 Translace, rotace a škálování objektu

Translaci lze provést nástrojem
emphthree.jsMove Object podržením tlačítka myši na objektu a následným posouváním, či pouze vybráním objektu a poté pomocí šipek na klávesnici.

Pro rotaci objektu slouží nástroj
emphthree.jsRotate Object a rovněž se zde používá podržení myši nad objektem a následné tažení. Bohužel zde nefunguje trik s klávesou SHIFT a tak je rotace tímto způsobem dosti zmatečná.

Škálování objektu lze provést vybráním nástroje *Resize Object*, podržením myši na úchopovém bodu a tažením. Zde bohužel nejde škálovat všechny hrany objektu stejnoměrně.

Všechny výše zmíněné transformace lze přesně zadat v panelu *Properties* na pravé straně okna.

2.1.5 Změna geometrie objektu

Před změnou geometrie je potřeba nad objektem vyvolat vyskakovací nabídku a zde vybrat převedení na trojúhelníkovou síť a poté opět ve vyskakovací nabídce nad objektem vybrat editaci objektu. Zobrazí se okno s objektem ve kterém jsou opět čtyři pohledy ve stejném rozložení jako na hlavní ploše. Zde si lze vybrat mezi bodem, hranou a plochou ve spodní levé části okna a různými transformacemi v horní levé části okna. Transformaci lze provést podržením tlačítka myši nad bodem či ikonou operace a následným tažením či pomocí šipek na klávesnici. Po kliknutí na tlačítko *OK* se okno zavře a na hlavní ploše je již vidět objekt se změněnou geometrií. Bohužel u manipulace s body, hranami a plochami se mi nikde nepodařilo nalézt přesnou číselnou

reprezentaci.

2.1.6 Další vlastnosti

V panelu *Properties* lze též změnit název objektu.

2.1.7 Zhodnocení

Art of Illusion není pro tvorbu krajiny ve 3D příliš uživatelsky přívětivý, přesto by v něm šlo s trochou úsilí krajinný reliéf vymodelovat.

2.2 FreeCAD

FreeCAD je svobodný *CAD* (*computer aided desing*) systém zejména pro parametrické 3D modelování se zaměřením na strojírenství a produktový design. Je napsaný v kombinaci jazyků *C++* a *Python*. [fcc()] První verze vyšla 29. září 2002 a za autory jsou považováni Juergen Riegel, Werner Mayer a Yorik van Havre. Program je vyvíjen pod licencí LGPL v2[fcl(2014)] a je multiplatformní.

2.2.1 Pracovní plocha

V horní části plochy je hlavní nabídka a klasický panel nástrojů. Zde je důležitý zejména výběr tzv. *workbench* - sady nástrojů:

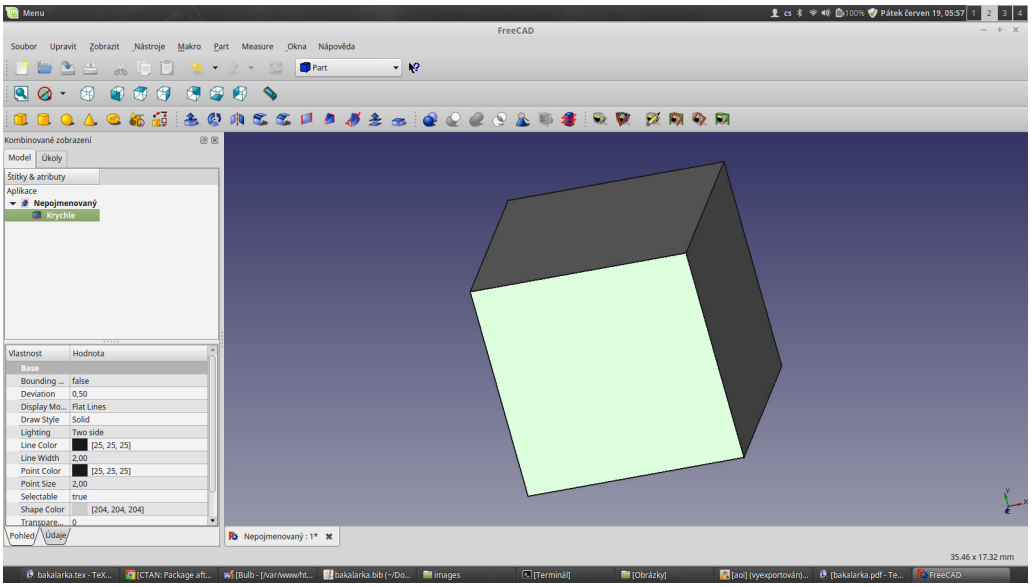
Arch - nástroje pro modelování staveb a stavebních prvků.

Draft - nástroje pro vytváření 2D obrazců.

Drawing - nástroje pro vkládání popisků do technického výkresu.

Fem - nástroje pro vytváření mechanicky pohyblivých spojů.

Image - dva nástroje pro vkládání obrázků do plochy.



Obrázek 2.3: Pracovní prostředí *FreeCAD*

Inspection - pouze prohlížení výkresu.

Mesh design - nástroje pro nahrávání trojúhelníkových sítí.

Part - nejvíce využívaný *workbench*, slouží k vytváření technických dílů.

Part design - rozšíření předchozí *workbench*.

Plot - nástroje pro vytváření grafů.

Points - nástroje pro práci s množinou bodů.

Raytracing - nástroje pro práci s osvětlením scény.

Robot - nástroje pro vytváření trajektorie pohybu robotů a robotických ramen.

Ship - nástroje pro návrh lodí.

Spreadsheet - nástroje pro vytváření titulní strany technického výkresu.

Start - úvodní stránka.

Test Framework - spouštění automatických testů programu *FreeCAD*.

Web - jednoduchý webový prohlížeč.

V levé části okna je v horním panelu seznam objektů na scéně, ve spodním panelu přehled vlastností právě označeného objektu. Zbytek okna zaujímá scéna.

2.2.2 Manipulace se scénou

Ve *FreeCAD*u existují čtyři styly manipulace se scénou. Lze je přepínat vyvoláním vyskakovací nabídky nad pracovní plochou a poté vybráním položky Styly Navigace.

Inventor Navigace

Přiblížení a oddálení scény lze provést otáčením kolečka myši. Posun scény se provádí podržením stisknutého kolečka myši a následným tažením. Rotace se provádí podržením stisknutého levého tlačítka myši a následným tažením. Po puštění tlačítka myši objekt zůstane rotovat dokud není do scény znovu kliknuto. Čím prudší je pohyb myší, tím rychlejší je následná rotace.

CAD Navigace

Přiblížení a oddálení scény lze provést otáčením kolečka myši. Posun scény se provádí podržením stisknutého kolečka myši a následným tažením nebo podržením stisknutého pravého tlačítka myši za současného držení klávesy Ctrl. Do třetice lze použít šipky na klávesnici. Scénu lze také uvést do rotace jako v případě *Inventor Navigace* stisknutím kolečka myši, následně levého či pravého tlačítka myši a tažením.

Blender Navigace

Pokud je vybrán tento typ navigace, je manipulace se scénou stejná jako v programu Blender, který je popsán výše.

Touchpad Navigace

Přiblížení a oddálení scény lze provést otáčením kolečka myši, klávesami Page Up a Page Down nebo podržením kláves Shift + Ctrl a pohybem myši. Scénu lze posouvat tahem myši za současně stisknuté klávesy Shift. Rotaci scény lze provést podržením klávesy Alt a pohybem myši nebo stisknutím kláves Shift + Ctrl, podržením stiknutého pravého tlačítka myši a tažením.

2.2.3 Vložení, označení a odstranění objektu

Vložení objektu se provádí ve workbenchi Part. Na výběr je krychle, válec, koule kužel, torus a nástroje na vytváření vlastních tvarů z primitiv. Po kliknutí na příslušné tlačítko v liště nástrojů se objekt rovnou vloží do scény.

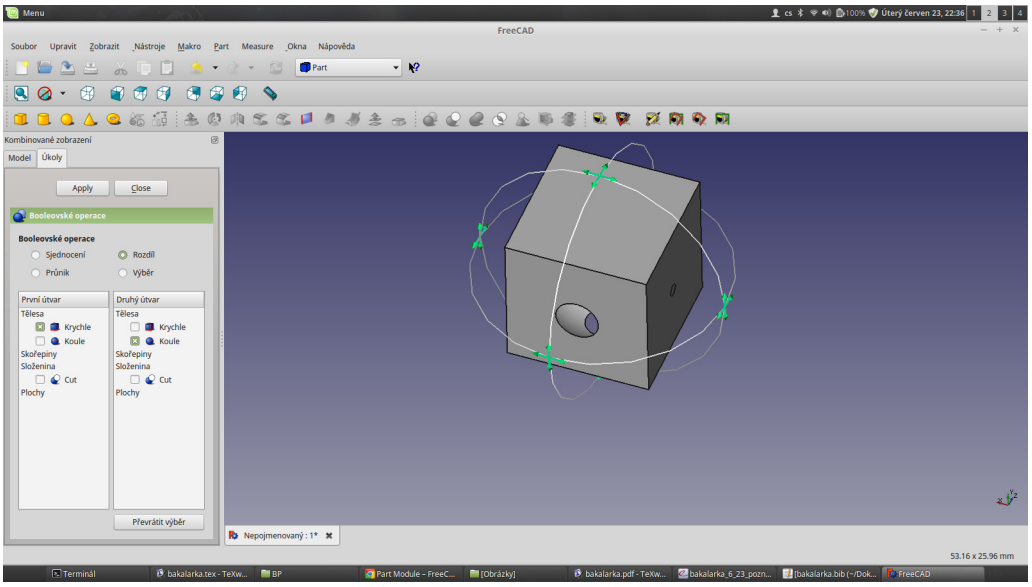
Způsob označení objektu je závislý na použitém stylu navigace. V Inventor Navigaci lze označit kliknutím levým tlačítkem na objekt za současného podržení stisknuté klávesy Ctrl. Tímto způsobem lze označit více objektů najednou. V CAD Navigaci, Blender Navigaci a Touchpad Navigaci lze označit objekt prostým kliknutím levého tlačítka myši. Pokud je stisknuta klávesa Ctrl, lze takto označit více objektů najednou. Ve *FreeCAD*u není při označování rozdíl mezi objektem, stěnou, hranou nebo vrcholem. Za označený se považuje ten objekt, který má označenou alespoň jednu stěnu, hranu či vrchol. Jakýkoliv objekt lze rovněž označit v seznamu objektů v levém horním panelu.

Je-li objekt označený, lze ho smazat buď po vyvolání vyskakovací nabídky po kliknutí pravým tlačítkem myši a zde vybrat Odstranit nebo stisknutím klávesy Delete.

2.2.4 Translace, rotace a škálování objektu

Je-li objekt označený, na panelu vlevo dole v záložce Údaje lze měnit v textových polích jeho vlastnosti.

2.2.5 Změna geometrie objektu



Obrázek 2.4: Odečtení koule z vnitřku krychle v prostředí *FreeCAD*

FreeCAD pracuje s objekty jinak než ostatní zmíněné programy pro práce ve 3D. Nepracuje pouze s povrchem tělesa, ale s celým jeho objemem. Výsledné těleso se proto vytváří sčítáním či odčítáním různých tvarů, jak je zřejmé z obr. 2.4.

2.2.6 Zhodnocení

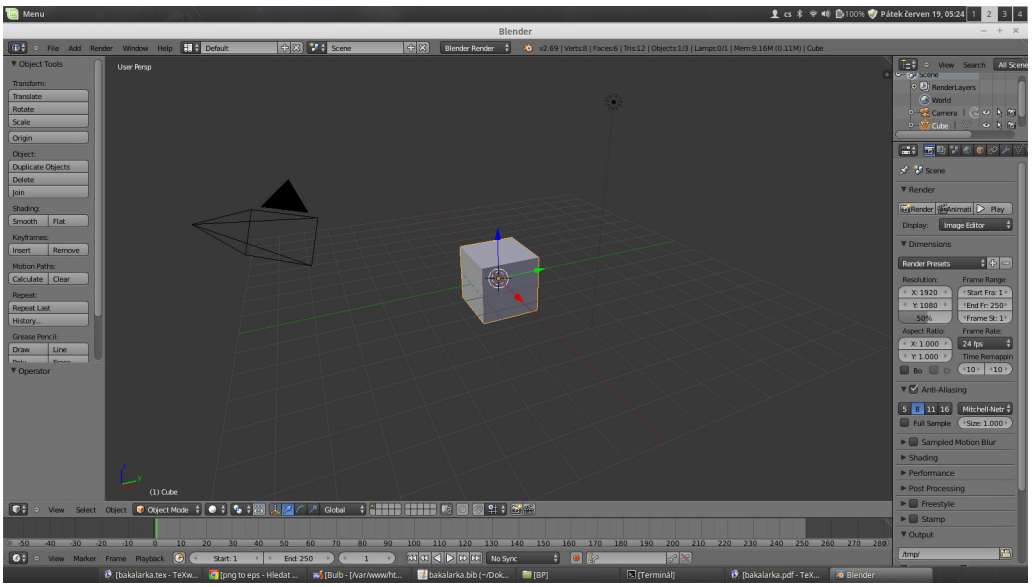
FreeCAD je pro tvorbu 3D krajinné scenérie naprosto nevhodný. Neumí pracovat s nepravidelnými objekty, zato má spoustu vlastností, které v tomto případě nevyužijeme.

2.3 Blender

Blender je svobodný software pro modelování a vykreslování 3D počítačové grafiky a animací s využitím různých technik (např. sledování paprsku, radiosita, scanline rendering, GI). Vlastní interface je vykreslován pomocí

knihovny *OpenGL*. *OpenGL* umožňuje nejen hardwarovou akceleraci vykreslování 2D a 3D objektů, ale především snadnou přenositelnost na všechny podporované platformy. Je napsaný v *C/C++* a některé pluginy jsou napsány v *Python*u, který blender využívá jako skriptovací jazyk. Za hlavního autora je považován Ton Roosendaal a první verze vyšla v roce 1995, od roku 2002 je *Blender* licencován licencí GNU GPL v2. [ble(2014b)] V současné době jeho další vývoj zaštiťuje nadace *Blender Foundation*. [ble(2008)] Program je multiplatformní. [ble(2014a)]

2.3.1 Pracovní plocha



Obrázek 2.5: Pracovní prostředí *Blenderu*

Blender má jednu pracovní plochu s perspektivní projekcí objektu, jak lze vidět na obrázku 2.5. V horní části okna se nachází hlavní menu, v pravé části okna se nachází zejména stromový seznam objektů scénou počínaje. Pod ním se nachází tlačítkové menu pro přepínání kontextů a panel aktuálního kontextu. Kontexty jsou následující:

Render - vše, co souvisí s vykreslováním (objekty, rozměry, anti-aliasing, výkon atd.).

Scene - gravitace ve scéně, jednotky a další obecné informace.

World - osvětlení, obloha, mlha, hvězdy atd.

Object - transformace a nastavení objektu

Constraints - nastavení omezení a hranic při transformaci objektů

Modifiers - nedestruktivní operace, které mají vliv na zobrazení objektů, např. vyhlazování či zrcadlení.

Object Data - obsahuje všechna data specifická pro objekt, zejména skupiny vertexů.

Materials - nastavení materiálů (barva, průhlednost, odrazivost).

Textures - specifikace povrchu objektu společně s materiálem.

Particles - velké množství malých objektů (např. hvězdy), kterými lze manipulovat pomocí silového pole.

Physics - nastavení fyzikálních zákonů platících ve scéně.

Ve spodní části okna je nástroj pro animaci a zejména potom nástroj pro výběr módů, který určuje, zda operace probíhají nad celým objektem nebo pouze nad jeho částmi jako jsou vrcholy, hrany a stěny. V levé části okna jsou ve standardním zobrazení dva panely. Horní panel s výběrem transformace (např. posun, otáčení a škálování) a dolní panel s detaily transformace (např. výběr osy podle které se transformace provádí a skalární hodnotu této transformace).

Blender je velmi rozsáhlý program a proto výčet nástrojů, tlačítek a přepínačů zdaleka není kompletní.

2.3.2 Manipulace se scénou

Scénu lze přiblížit otáčením kolečka myši nebo podržením stisknutého kolečka myši a a následným tažením se stisknutou klávesou Ctrl. Podržením kolečka myši a jejím tažením lze scénou rotovat a stejným postupem se stisknutou klávesou Shift lze scénou posouvat.

2.3.3 Vložení, označení a odstranění objektu

Objekt lze vybrat z nabídky *Add* v hlavním menu. Po kliknutí se objekt vloží do scény v místě 3D kurzoru.

Výběr objektu lze provést kliknutím pravého tlačítka myši na objekt ve scéně a nebo kliknutím na název objektu v seznamu objektů po pravé straně. Pokud je stisknuta klávesa SHIFT, lze vybrat více objektů najednou.
Objekt lze odstranit ze scény kliknutím na tlačítko *Delete* v levém panelu *Object Tools* nebo stisknutím klávesy DELETE či klávesy X na klávesnici. Poté se zobrazí vyskakovací nabídka, ve které se potvrdí odstranění objektu ze scény.

2.3.4 Translace, rotace a škálování objektu

Objekt lze posouvat po kliknutí na tlačítko *Translate* v levém horním panelu. Po kliknutí se objekt rovnou začne pohybovat za pohybem kurzoru. Po druhém kliknutí se objekt zastaví a pak lze ve spodním panelu zadat hodnotu posunu v rámci osy číselně. Další možností je použít klávesu G, která spouští nástroj pro posun *Grab*. Poté se objekt pohybuje opět volně za kurzorem, proto lze klávesami X,Y a Z určit osu, ve které bude posun probíhat. K jemnému doladění pozice lze použít šipkové klávesy. Nástroj *Grab* je možné zapnout také tlačítkem s ikonou šipky ve spodní liště.

Pro otáčení objektu platí to samé co pro posun. Vyvolává se buď kliknutím na tlačítko *Rotate* v levém horním panelu nebo stiskem klávesy R popřípadě kliknutím na tlačítko s ikonou oblouku ve spodní liště.

Škálování lze spustit opět buď kliknutím na tlačítko *Scale* v levém horním panelu, klávesou S nebo kliknutím na tlačítko s ikonou paličky ve spodní liště. Vedle tohoto tlačítka se také nachází výběr souřadnicového systému.

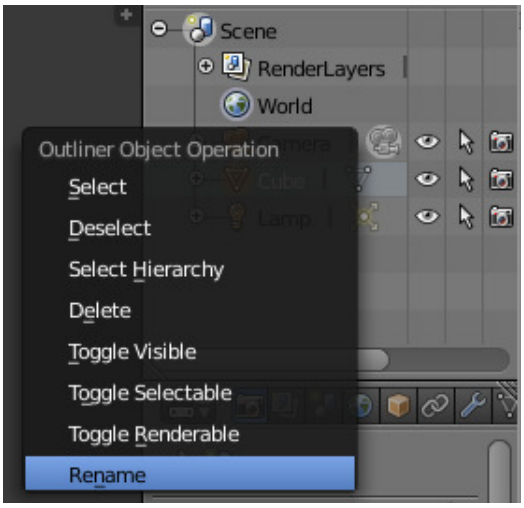
2.3.5 Změna geometrie objektu

Před samotnou změnou geometrie je potřeba se přepnout z objektového do editačního módu buď výběrem ve spodní liště, nebo stisknutím klávesy TAB. V editačním módu lze vybírat vrcholy kliknutím pravým tlačítkem myši stejně jako objekty v objektovém módu. Při podržení stisknuté klávesy CTRL

lze vybrat více vrcholů najednou.
S označenými vrcholy lze provádět stejné transformace jako s celými objekty.
Osy procházejí středem označené oblasti.

2.3.6 Další vlastnosti

Objekt lze přejmenovat vyvoláním vyskakovací nabídky kliknutím pravým tlačítkem na název objektu v seznamu objektů na panelu vpravo nahoře, vybráním volby *Rename* a následnou editací textového pole, jak lze vidět na obrázku 2.6.



Obrázek 2.6: Přejmenování objektu v *Blender*u

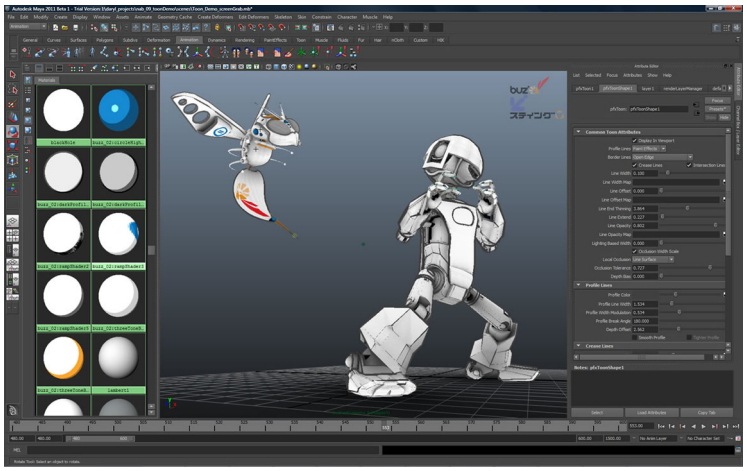
2.3.7 Zhodnocení

Ze tří testovaných programů je *Blender* pro modelování krajinné scenérie ve 3D nejvhodnější.

2.4 Proprietární software

Proprietární software jsem zejména z technických důvodů netestoval, přesto je nezbytné zmínit alespoň jeho dva nejznámější zástupce.

2.4.1 Autodesk Maya



Obrázek 2.7: Pracovní prostředí *Autodesk Maya*

Autodesk Maya (obr. 2.7) je proprietární software pro 3D modelování, 3D animaci, 3D zobrazování a vytváření dynamických efektů. [may(a)] Název je odvozen ze sanskrtského slova *Maya*, hinduistického konceptu iluze. [Johari()] Program je multiplatformní [may(c)], napsaný převážně v *C++* a *Python*u. [may(b)]

2.4.2 Autodesk 3ds Max

Obrázek 2.8: Pracovní prostředí *Autodesk 3ds Max*

Autodesk 3ds Max (obr. 2.8) je proprietární profesionální program pro 3D počítačovou grafiku vyvíjený společností *Autodesk Media and Entertainment* [max(a)]. Umožňuje 3D modelování, 3D animaci, práci s texturami, 3D vykreslování a vytváření dynamických efektů. [max(b)] Program je dostupný pro operační systém Microsoft Windows 7/8/8.1 64-bit. [max(c)]

3 Realizace

Rozhodl jsem se pro napsání editoru jako webové aplikace. To sebou přináší řadu výhod:

- Známý, dobře zdokumentovaný a všude dostupný ekosystém.
- Multiplatformnost.
- Možnost rychlého vydávání nových verzí a tím zrychlení vývojového cyklu.

A samozřejmě také některé nevýhody:

- Omezení způsobená prohlížečem, například poněkud nepohodlný export hotových scén.
- Omezení vycházející z nekompatibility jednotlivých prohlížečů a jejich verzí.

Realizovaný program se jmenuje **Bulb**. Je napsán v jazycích *CoffeeScript* a *JavaScript* (*ECMAScript 5*), pro vytvoření indexového souboru je použit značkovací jazyk *HTML* a pro kaskádové styly jazyk *Stylus*. *CoffeeScript* je jazyk, který se překládá do *JavaScript*u (tzv. *transpiler*). Jeho syntaxe je inspirovaná jazyky *Python* a *Ruby* a implementuje mnoho vlastností z těchto dvou jazyků. [Ashkenas et al.(2012)] Stylus patří do rodiny dynamických CSS jazyků, jež jsou také často nazývány preprocesory a které rozšiřují CSS 3 o dynamické věci jako proměnné, mixiny, operace a funkce. [Nampoothiri(2012)] Grafická primitiva jsou za pomoci knihovny *three.js*. Tato knihovna nabízí následující nástroje [three.js authors(2012)]:

Vykreslovače: *WebGL* (hardwarově akcelerovaný), *Canvas*, *SVG* (softwarové)

Scény: přidávání a odebírání objektů v reálném čase, mlha.

Kamery: perspektivní a ortografická.

Ovládací prvky: *Trackball Controls* (ovládání celé scény), *Transform Controls* (posun, otáčení a škálování objektů)

Světla: rozptýlené, reflektor, bodové, polokulové.

Materiály: Solidní materiály, odrazivé materiály, normálové materiály atd.

Objekty: Tělesa, částice, čáry.

Geometrie: plochy, kostky, koule, válce, torus atd.

Export/Import: nástroje pro export/import z/do *JSON* a *OBJ*.

Matematické nástroje: matice, vektory atd.

Program je komponentově orientovaný. Každá komponenta je jednoznačně určena svým ID a vykresluje se do kontejneru, což je *HTML tag* (nejčastěji div nebo span), který má toto ID ve svém atributu id. Mimo to obsahuje ještě mnoho nevizuálních tříd. Třídy z objektu Bulb, tj. začínající Bulb. a třídy z frameworku *ComponentJS*, tj. začínající CJS. jsou původní, ostatní třídy jsou převzaté.

Komponenty tvoří komponentový strom odpovídající *DOM*u (*Document Object Model*). Pokud se překreslí komponenta blíže ke kořeni stromu, vyvolá překreslení i na svých podřízených komponentách.

Hlavní komponentou je třída Bulb.Document, která je v komponentovém stromu postavena nejvýše, tj. je kořenem. Ta jednak zachytává a následně deleguje události vyvolané na podřízených komponentách (tzv. *event delegation*) a jednak funguje jako spojovací můstek mezi komponentami, které o sobě navzájem neví.

Samotné vykreslování má na starosti třída Bulb.Canvas. Ta obsahuje scénu (*three.js*), kameru (*three.js*), vykreslovač (*three.js*), THREE.TrackballControls, THREE.TransformControls, Bulb.SelectControl a kolekci objektů.
THREE.TrackballControl je nástroj knihovny THREE.JS, který umožňuje pohyb se scénou. THREE.TransformControls je opět nástroj knihovny *three.js*, který umožnuje transformace (posun, otáčení, škálování) pomocí myši a směrové růžice.

Bulb.Select Control je třída na označování těles a bodů. Obsahuje mimo jiné instanci třídy Bulb.VertexControl, která umožňuje pohyb jednotlivými vrcholy ať už za pomoci myši či za pomoci klávesnice.

3.1 Popis nejdůležitějších tříd

Všechny zde popisované třídy jsou původní a mnou napsané.

3.1.1 CJS.Component

Předek všech komponent, tj. objektů na stránce, které se vykreslují a reagují na události. Komponenty mohou být uspořádány do stromu. Jsou jednoznačně definovány svým ID, které odpovídá atributu id HTML elementu, do kterého se daná komponenta vykresluje.

Přehled nejdůležitějších metod

CJS.Component(id, parent) Konstruktor, parametrem je id - textový identifikátor a parent - instance nadřazené komponenty.

getId() Vrátí identifikátor komponenty.

setParent(parent) Nastaví nadřazenou komponentu v komponentovém stromu.

setChild(child) Nastaví podřazenou komponentu v komponentovém stromu.

getChildById(id) Vrátí přímo podřazenou komponentu s daným id.

findChildById Najde a vrátí podřazenou komponentu v jakékoliv podřazené úrovni komponentového stromu.

getEvent(name) Vrátí instanci třídy CJS.Event.

setChild(child) Nastaví podřazenou komponentu v komponentovém stromu.

open(title, x, y) Zobrazí komponentu jako dialog s titulkem title na pozici x, y

close() Je-li komponenta dialogem, zavře ji.

getHtml() Vrátí *HTML* kód komponenty.

render() Vykreslí komponentu a všechny jí podřazené komponenty.

renderFinish() Na komponentě a jí podřazených komponentách provede dodatečné akce po vykreslení.

restoreView() Na komponentě a jí podřazených komponentách provede akce pro aktualizaci obsahu.

3.1.2 CJS.Document

Dědí od CJS.Component. CJS.Document je nejvyšší komponentou, která obsahuje všechny ostatní. Zachytává události a deleguje je zpátky na jednotlivé komponenty.

Přehled nejdůležitějších metod

bindEvents() Zaregistruje posluchače událostí.

3.1.3 CJS.Event

Uživatelem definovaná událost.

Přehled nejdůležitějších metod

subscribe(obj, func) Zaregistruje posluchače události. func je callback, obj je instance objektu, který callback obsahuje.

fire() Vyvolá událost. Může mít libovolný počet libovolných argumentů.

3.1.4 Bulb.Document

Hlavní třída celého programu. Dědí od CJS.Document. Slouží jako most mezi ostatními komponentami - zachytává jejich události (*event delegation*) a na základě nich rozhoduje o chování jiných komponent. Zároveň funguje jako továrna pro své podřízené komponenty: Bulb.Canvas, Bulb.Toolbar, CJS.TabMenu,

Bulb.SaveDialog, Bulb.SettingsDialog, Bulb.HelpDialog a některé další pomocné třídy (Bulb.Exporter).

Přehled nejdůležitějších metod

clear() Vymaže historii a znovu načte stránku. Historie je uložená v *local storage*, což je persistentní datové úložiště dostupné v prohlížeči. [Foundation(2015)]

getCanvas() Vrátí instanci kreslícího plátna (Bulb.Canvas).

getToolbar() Vrátí instanci pravého panelu (Bulb.Toolbar).

getObjectList() Vrátí instanci seznamu objektů v pravém panelu.

getProperties() Vrátí instanci panelu se záložkami *Mesh* a *Vertices* v pravém panelu.

getSaveDialog() Vrátí instanci dialogového okénka pro ukládání.

getSettingsDialog() Vrátí instanci dialogového okénka s nastaveními.

getExporter() Vrátí instanci exportéru (Bulb.Exporter).

export(saveTypeId, filename) Exportuje scénu do souboru filename a vyvolá stažení souboru. Pro export používá třídu Bulb.Exporter. Stahování vyvolává vložením odkazu do *DOM*u a následným spuštěním události kliknutí nad tímto odkazem.

undo() Krok zpět v historii. Historie je objekt, který obsahuje pole s jednotlivými stavy scény uloženými ve formátu *JSON* a pozici v tomto poli. *Redo* pak jednoduše posune tento ukazatel pozice o 1 dolů a načte stav scény z pole stavů.

redo() Krok vpřed v historii. Posune ukazatel pozice o 1 nahoru pokud takto zvýšený ukazatel neukazuje mimo rozsah pole se stavy.

load() Vyvolá upload souboru a rozhoduje o zpracování podle typu souboru.

parseFile(text, ext) Zpracuje text parserem (uživatelem definovaný importní skript) pro příponu ext.

addObjectToCanvas(data) Vloží na plátno objekt načtený ze souboru (data).

saveStatus() Uloží aktuální stav scény do historie, tj. zvýší ukazatel na pozici v poli stavů o 1 a přidá stav scény ve formátu *JSON* do pole stavů na pozici, na kterou ukazuje ukazatel.

bindEvents() Přidá další posluchače událostí, zejména pro stisky kláves a pro opuštění stránky. V tu chvíli se totiž ukládá historie, která byla dosud držena pouze v paměti, do *local storage*.

3.1.5 Bulb.Canvas

Kreslicí plátno se scénou. Dědí od CJS.Component. Funguje též jako továrna pro podřízené komponenty: THREE.PersperctiveCamera, THREE.Scene, THREE.WebGLRenderer, THREE.TrackballControls, THREE.TransformControls, Bulb.ObjectCollection, Bulb.SelectControl, Bulb.SelectHelper a THREE.GridHelper.

Přehled nejdůležitějších metod

getObjectCollection() Vytvoří (pokud není) a vrátí kolekci s objekty typu THREE.Mesh. Kolekce je důležitá pro seznam objektů, jsou v ní totiž pouze objekty typu THREE.Mesh, kdežto ve scéně jsou i různé helpery atd.

getRenderer() Vytvoří (pokud není) a vrátí instanci vykreslovače, konkrétně THREE.WebGLRenderer, který používá hardwarovou akceleraci.

getScene() Vytvoří (pokud není) a vrátí instanci scény (THREE.Scene).

getCamera() Vytvoří (pokud není) a vrátí instanci kamery.

getTrackballControls() Vytvoří (pokud není) a vrátí instanci třídy pro pohyb se scénou pomocí myši.

getTransformControls() Vytvoří (pokud není) a vrátí instanci třídy pro transformace objektu pomocí myši.

getSelectControl() Vytvoří (pokud není) a vrátí instanci Bulb.SelectControls, což je třída pro označování vrcholů pomocí myši.

getSelectHelper() Vytvoří (pokud není) a vrátí instanci Bulb.SelectHelper, což je třída, která zvýrazňuje označené vrcholy. Pokud jsou označeny

nesousedící body, jsou označeny oranžovým prostorovým křížkem, pokud jsou sousedící, je oranžově vyznačena hrana mezi nimi.

setMode(mode) Nastaví mód mode (Bulb.MODE_MESH nebo Bulb.MODE_VERTICES). V módu Bulb.MODE_MESH uživatel pracuje s celými objekty, v módu Bulb.MODE_VERTICES s jednotlivými vrcholy.

getMode() Vrátí aktuálně používaný mód.

moveSelectedObject(step, axis) Pohne s označeným objektem ve směru osy axis o vzdálenost step.

setTransformMode(transofrmMode) Nastaví typ transformace (posun, otáčení, škálování)

getTransformMode() Vrátí nastavený typ transformace.

setTransformSpace(space) Nastaví vztažnou soustavu souřadnic pro transformaci (globální nebo vázanou k objektu)

getWireframeHelper(key) Vrátí instanci třídy Bulb.WireframeHelper. Klíč key udává, je-li tato třída asociovaná s označeným objektem nebo se zobrazuje se při najetí na objekt kurzorem myši.

getSelectedObject() Vrátí označený objekt na scéně. Objekt na scéně je instance třídy THREE.Mesh.

addLight() Přidá na scénu bodový zdroj bílého světla (THREE.PointLight).

getMaterial() Vytvoří (pokud není) a vrátí instanci materiálu.

getWidth() Vrátí šířku plátna.

getHeight() Vrátí výšku plátna.

get*JSON*() Vrátí stav scény a označený objektu ve formát *JSON*.

set*JSON*(*JSON*) Nastaví scénu podle stavu uloženého ve formátu *JSON*.

initScene() Inicializuje scénu, tj. vloží do ní THRE.GridHelper (mřížka v rovině os x a z) a THREE.AxisHelper (osová růžice).

addLoadedObject(object) Vloží na scénu objekt, který byl předtím načten ze souboru. Objekt načtený ze souboru se definován geometrií typu THREE.BufferedGeometry, která je sice paměťově méně náročná než THREE.Geometry, protože je tvořená pouze maticí vrcholů, ale nelze ji nijak měnit. Je tedy nutné tuto geometrii přepočítat na THREE.Geometry až následně vložit do scény.

addObject(geometry, name, position) Vloží do scény objekt, definovaný geometrií geometry se jménem name na pozici position.

addCircle(name) Vloží do scény kruh se jménem name.

addPlane(name) Vloží do scény čtvercovou plochu se jménem name.

addSphere(name) Vloží do scény kouli se jménem name.

addCube(name) Vloží do scény krychli se jménem name.

addCylinder(name) Vloží do scény válec se jménem name.

addTorus(name) Vloží do scény torus se jménem name.

addVector(name) Vloží do scény bod se jménem name.

addWireframeHelper(key, object) Asociuje k objektu object instanci třídy THREE.WireframeHelper pod klíčem key.

removeWireframeHelper(key, object) Odstraní asociaci instance třídy THREE.WireframeHelper k objektu object pod klíčem key.

remove(objectId) Odstraní objekt s id objectId ze scény.

renameObject(id, name) Nastaví objektu s ID id jméno name.

replaceObject(params) Nahradí objekt po změně geometrie.

selectObject(objectId, fireEvents) Označí objekt s ID objectId. Příznak fireEvents udává, zdali se mají spustit události spojené s označením objektu.

mouseOverMesh(event) Spouští akce, které se dějí při přechodu kurzoru myši nad objektem.

getIntersect(event, objects) Vrátí průniky objekty objects paprskem vyslaným od kamery ke kurzoru myši.

resize() Přepočítá a překreslí scénu po změně velikosti okna.

restoreView() Překreslí scénu.

3.1.6 Bulb.SelectControl

Dědí od Bulb.SelectHelper. Stará se o označování jednotlivých vrcholů. Pohyb s vrcholy deleguje na podřízený objekt třídy Bulb.VertexControl. Při označování z THREE.Raycaster získám bod průniku označujícího paprsku a příslušnou plochu v objektu. V této ploše naleznu nejmenší vzdálenost od bodu průniku k některému vrcholu. Tento vrchol poté prohlásím za označený. Bulb.SelectControl také počítá pozici směrové růžice. Ta je v těžišti označených vrcholů. Spočítá se vektorovým součtem vrcholů a následným skalárním dělením jejich počtem.

Přehled nejdůležitějších metod

Bulb.SelectControl(camera, scene, domElement) Konstruktor přijímá instanci kamery, scény a *DOM* element ve kterém je scéna vykreslena.

setSelectedObject(object) Nastaví označený objekt. Označený objekt obsahuje pole selecteds, ve kterém jsou indexy jednotlivých označených vrcholů.

deactivate() Deaktivuje a odstraní Bulb.SelectControl ze scény.

getVertexControl Vytvoří (pokud není) a vrátí objekt třídy Bulb.VertexControl. Tato třída slouží k pohybování s vrcholy.

getRaycaster() Vytvoří (pokud není) a vrátí instanci THREE.Raycaster. THREE.Raycaster dostane seznam objektů a vysílá myšlený paprsek a vrací objekt, ve kterém jsou kolekce objektů protnutých tímto paprskem, protnuté plochy těchto objektů a body protnutí.

getIntersect(event, objects) Vrátí průniky objekty objects paprskem vyslaným od kamery ke kurzoru.

highlightVector(vector) Nastaví vrchol vector jako aktuální (je nad ním kurzor myši)

unhighlightVector() Nastaví vrchol vector jako neaktuální (není nad ním kurzor myši, který tam dříve byl)

toggleSelectMode() Přepne mód (označování nebo pohyb s vrcholy)

selectVector(add) Označí vrcholy podle přepínače add. (0 - označí právě jeden vrchol, 1 - přidá vrchol do seznamu označených, 2 - označí všechny vrcholy na nejkratší cestě mezi posledním vrcholem v seznamu označených vrcholů a aktuálně označeným vrcholem, 4 - označí všechny vrcholy uvnitř uzavřené oblasti ohraničené jinými označenými vrcholy)

selectShortestPath(start, end) Označí všechny vrcholy na nejkratší cestě ze start do end. Používá se zde Dijkstrův algoritmus popsaný níže.

getMin(lengths, nonVisited) Vrátí nejbližší dosud nenavštívený bod.

getNeighbors() Vrátí pole, jehož indexem je index vrcholu a hodnotou pole sousedních vrcholů.

floodFillSelect(vertex) Označí všechny vrcholy uvnitř uzavřené oblasti ohraničené jinými označenými vrcholy. Označování začíná od vrcholu vertex. Používá se zde algoritmus semínkového vyplňování popsaný níže.

3.1.7 Bulb.VertexControl

Třída slouží k posunu jednotlivých vrcholů. Vykreslí do scény směrovou růžici složenou z šipek (instancí třídy THREE.ArrowHelper). Pod každou šipkou je vykreslena (avšak nezobrazena) pomocná plocha. Vektorový rozdíl mezi středem směrové růžice a bodem průniku myšleného paprsku vyslaného insntancí třídy THREE.Raycaster určuje velikost posunu. Tento přístup lze použít při pohybu v osách X, Y a Z. Při pohybu po normále lze použít šipky na klávesnici. Normála se počítá Jako průměr z normál vrcholů pro jednotlivé plochy na nichž daný vrchol participuje.

3.2 Popis použitých algoritmů

3.2.1 Dijkstrův algoritmus pro nalezení nejkratší cesty mezi dvěma uzly v grafu

V programu je možné označit vrcholy na nejkratší spojnici dvou vrcholů. Při takovém označování vrcholů je nutné získat nejkratší cestu mezi posledním označeným vrcholem a právě označovaným vrcholem a všechny vrcholy na této cestě označit. K získání nejkratší cesty byl použit Dijkstrův algoritmus: [Dijkstra(1959)]

Jedná se o problém nalezení nejkratší možné délky mezi dvěma uzly (P,Q) v grafu. Uzly rozdělíme na tři skupiny:

- A. Uzly, k nimž známe minimální vzdálenost z P (již navštívené).
- B. Uzly, které bezprostředně sousedí s již navštívenými uzly.
- C. Ostatní dosud nenavštívené uzly.

Větve rozdělíme také na tři skupiny:

- I. Větve, které jsou součástí nejkratších cest k již navštíveným vrcholům.
- II. Větve, ze kterých bude vybrána další větev, jež má být umístěna do množiny I. Právě 1 větev z této množiny povede do každého uzlu z množiny B.
- III. Ostatní větve (zamítnuté nebo dosud nenavštívené větve).

Na začátku jsou všechny uzly v množině C a všechny větve v množině III. Nyní převedeme uzel P do množiny A a od té doby budeme opakovaně provádět následující kroky:

- Krok 1. Zvažme všechny větve r spojující uzel právě převedený do množiny A s uzly R z množiny B nebo C. Jestliže uzel R patří do množiny B, zjišťujeme, zda použitím větve r získáme kratší cestu z P do R než

je známá cesta, která používá odpovídající větev z množiny II. Pokud tomu tak není, větev r zamítneme.Pokud však použití větve r má za následek kratší spojení z P do R než dosud získané, nahradí odpovídající větev v množině II a ta druhá je zamítnuta. Jestliže uzel R patří k k množině C, přidá se do množiny B a větev r se přidá do množiny II.

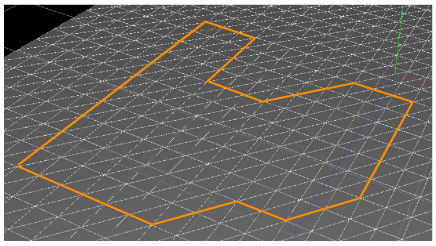
- Krok 2. Každý uzel z množiny B může být spojen s uzlem P pouze jedním způsobem pokud se omezíme na větve z množiny I a jednu větev z množiny II. V tomto významu má každý uzel v množině B vzdálenost z uzlu P: uzel s minimální vzdáleností z P je převeden do z množiny B do množiny A a odpovídající větev je převedena z množiny II do množiny I. Poté se vrátíme ke kroku 1 a opakujeme proces dokud uzel Q není převeden do množiny A. V tu chvíli je nalezeno řešení.

Pseudokód s Dijkstrovým algoritmem

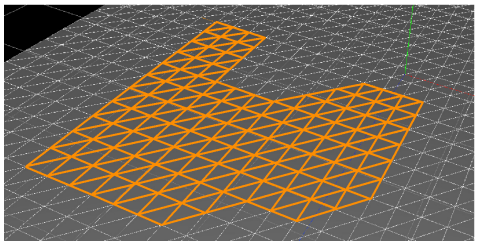
```
function dijkstra(start, end) {
    lengths = [] //Pole delek jednotlivych cest, na zacatku prazdne
    path = [] //Pole jednotlivych cest, na zacatku prazdne

    nonVisited = [pole dosud nenavstivenych vrcholu]

    // Priprava pole se vzdalenostmi mezi vrcholy. Vzdalenost k vrcholu
    // start bude 0 (prave se v nem nachazime) a ke vsem ostatnim
    // vrcholum nekonecno.
    for(i in nonVisited) {
        if(i is start) {
            lengths[i] = 0
        } else {
            lengths[i] = Infinity
        }
        path[i] = undefined
    }

    // Prochazime, dokud zbyvaji nejake nenavstivene vrcholy
    while(not empty nonVisited) {

        // Ziskame nejblizsi dosud nenavstiveny vrchol
        vertex = getMin(lengths, nonVisited)

        // Pokud je nejblizsi dosud nenavstiveny vrchol konecnym vrcholem,
        // ukoncime prohledavani.
         if(vertex is end) break

        // Vyjmeme tento vrchol z mnoziny dosud nenavstivenych vrcholu.
        nonVisited.splice(vertex)


        for(neighbour of vertex) { //Projdeme sousedy vrcholu

            // Do pole se vzdalenostmi pridame nejkratsi cestu k sousedum.
            alt = lengths[vertex]+1;
            if(alt < lengths[neighbour]) {
```

```
                lengths[neighbour] = alt

                // Pamatujeme si cestu
                path[neighbour] = vertex
            }
        }
    }
    shortestWay = []
    vertex = end

    // Zjistime nejkratsi cestu tak, ze projdeme nalezene vrcholy
    // a tyto vkladame na zacatek pole s nejkratsi cestou.
    while(not empty(path[vertex])) {
        shortestWay.unshift(vertex)
        vertex = path[vertex]
    }
    return(shortestWay)
}
```

3.2.2 Algoritmus semínkového vyplňování (flood fill)



Obrázek 3.1: Ohraničená oblast



Obrázek 3.2: Vyplněná oblast

Pokud je na tělese vytvořena uzavřená oblast ohraničená označenými vrcholy a hranami mezi nimi (obr. 3.1), je možné následně označit všechny vrcholy uvnitř takto uzavřené oblasti (obr. 3.2). Pro označení vrcholů uvnitř uzavřené oblasti jsem použil algoritmus semínkového vyplňování. Vyplňovací algoritmy tohoto typu se často používají v rastrových zařízeních a v interaktivní grafice. Pro vyplnění se předpokládá, že je známý alespoň jeden vnitřní bod vyplňované oblasti (tzv. semínko). Nejčastěji ho definuje sám uživatel ukázáním dovnitř oblasti. Nejzákladnější algoritmus tohoto typu má rekurzivní charakter. Pracuje se přímo s body rastru a testují se vždy sousední body semínka. Při vyšetřování se zjišťuje, zda je sousední bod hraniční nebo ne a zda je již vybarvený (v tomto případě označený). Pokud bod není hraniční ani vybarvený, vybarví se a v nové úrovni rekurze se použije jako semínko. [Sobota(1995)]

Pseudokód s algoritmem semínkového vyplňování (flood fill)

```
function floodFillSelect(vertex) {

    // Pridame vrchol do seznamu oznacenych
    select(vertex)

    // Prochazime sousedy vrcholu a pokud neni soused oznaceny,
    // predavame ho jako parametr do volani floodFillSelect
    for(neighbour of vertex when neighbour not in selectedVertices) {
        floodFillSelect(neighbour) //rekurze
    }
}
```

3.3 Technologická omezení

Vzhledem k tomu, že se jedná o aplikaci běžící uvnitř prohlížeče, jsou zde jistá technologická omezení. Jedním z nich je i nemožnost vyvolání standardního dialogového okna *Uložit jako...* Místo toho je exportovaným souborům přistupováno jako ke stahovaným.

4 Vytvoření scény

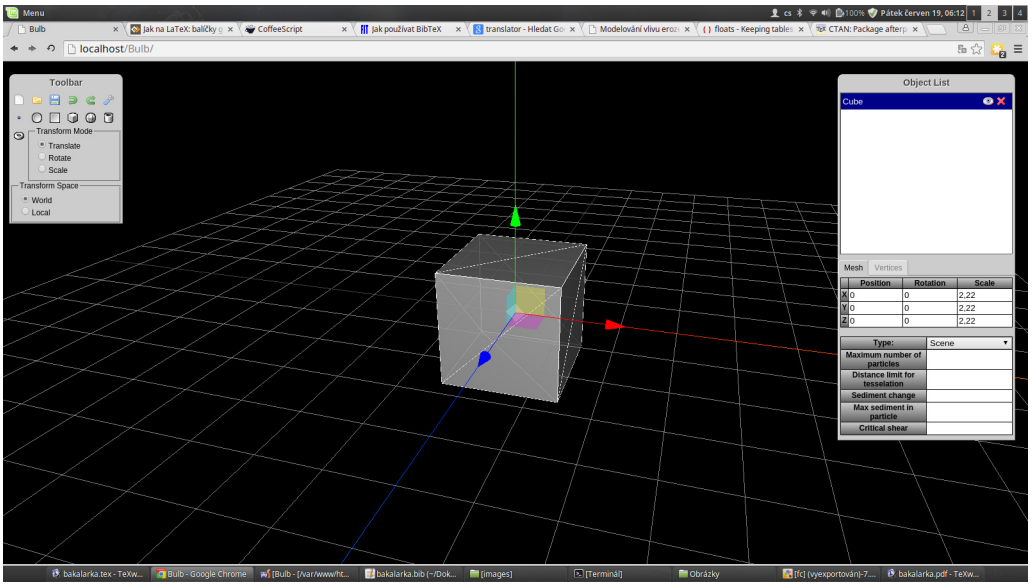
4.1 Překlad, instalace a spuštění programu

Program lze spustit z přiloženého CD otevřením souboru `index.html` v internetovém prohlížeči nebo přímo na adrese `http://bulb.freetech.cz`. Pokud chcete program přeložit, například po změně ve zdrojových kódech, je potřeba mít nainstalovaný *transpiler JavaScript*u *CoffeeScript* a *CSS preprocesor Stylus*. Překlad se provádí v konzoli spuštěním sekvence příkazů v kořenovém adresáři projektu:

```
$ coffee -c /coffee
$ stylus stylus/
```

4.2 Ovládání programu

4.2.1 Pracovní plocha



Obrázek 4.1: Pracovní prostředí Bulb

Pracovní plocha (obr. 4.1) je tvořena samotnou kreslicí plochou, nad kterou se nachází dvě okna. V levém okně se nacházejí panely nástrojů a přepínače transformací a souřadného systému. Horní panel nástrojů je obecný a nalezneme zde tlačítka:

New - vytvoření nové scény.

Import - import objektu nebo scény ze souboru.

Export - export objektu nebo scény do souboru.

Undo - vrátí situaci na scéně o akci zpět.

Redo - posune vrácenou scénu o akci vpřed.

Settings - otevře panel s nastavením.

Help - otevře panel s výpisem klávesových zkratek.

Spodní panel nástrojů obsahuje tlačítka pro vložení objektů do scény a ta jsou:

Vector - vloží do scény bod.

Circle - vloží do scény kružnici.

Plane - vloží do scény plochu.

Cube - vloží do scény krychli.

Sphere - vloží do scény kouli.

Cylinder - vloží do scény válec.

Torus - vloží do scény torus.

Pod panely se nachází přepínač transformací. Je-li na něm vybráno Translate, s objektem se posouvá, je-li vybráno Rotate, s objektem se rotuje a je-li vybráno Scale, lze měnit objektu velikost. Poslední přepínač určuje použitou vztažnou soustavu: buď globální pro celou scénu nebo lokální, svázanou s vybraným objektem. V pravém okně se nachází seznam objektů, ve kterém lze objekty označovat, přejmenovávat a mazat. Pod seznamem objektů je panel se záložkami. Vybraná záložka určuje mód aplikace. Záložky jsou:

Mesh - manipulace s celým objektem a jeho daty. V tomto módu lze provádět transformace (posun, otočení, změnu velikosti) objektu a vybírat celé objekty. Také zde lze editovat uživatelem definovaná metadata na objektu.

Vertices - manipulace s konkrétní geometrií (např. u plochy lze vybrat z kolika trojúhelníků bude složená, u koule poloměry a vyhlazení atd.) nebo s jednotlivými vrcholy obecné geometrie. V tomto módu lze vybírat jednotlivé vrcholy a posouvat jimi. Posunutí vrcholu u objektů s konkrétní geometrií (krychle, koule, válec atd.) způsobí její zobecnění.

4.2.2 Manipulace se scénou

Scénu lze posouvat držením pravého tlačítka a současným pohybem myši. Rotovat scénou lze podržením levého tlačítka a současným tažením myši a přiblížení lze provést kolečkem myši. Tyto úkony nelze provádět, je-li aplikace v módu *Vertices* a současně se kurzor myši nachází nad vybraným objektem. Tato funkcionalita usnadňuje výběr vrcholu ve scéně.

4.2.3 Vložení, označení a odstranění objektu

Objekt je možné do scény vložit tlačítky z panelu nástrojů popsaných výše. Označení lze provést buď kliknutím na objekt ve scéně nebo jeho vybráním v seznamu objektů v pravém okně. Objekt lze odstranit kliknutím na křížek v seznamu objektů nebo stisknutím klávesy Delete, pokud je objekt označen.

4.2.4 Translace, rotace a škálování objektu

Po označení objektu se nad tímto objektem objeví směrová růžice (*gizmo*). Kliknutím na šipku nebo plochu v této růžici lze objekt posouvat po ose symbolizované šipkou či po vybrané ploše. Hodnotu posunu lze také zadat číselně do tabulky na záložce *Mesh*. Otáčet objektem lze po přepnutí na *Rotation* v přepínači v levém okně. Tvar směrové růžice se změní na soustavu kružnic a následně kliknutím a tažením kurzoru podél jedné z těchto kružnic lze objektem rotovat podle osy symbolizované kružnicí. Hodnotu otáčení lze také zadat číselně ve stupních na záložce *Mesh*. Po vybrání *Scale* v přepínači

v levém okně lze měnit velikost objektu, opět pomocí směrové růžice. Lze měnit velikost objektu v jednotlivých osách a také celkovou velikost objektu. Velikost objektu lze také vložit číselně do tabulky na záložce *Mesh*. Typ transformace lze vybrat také s pomocí klávesnice. Stisk klávesy Tpřepne mód transformace na translaci, stisk klávesy R přepne mód transformace na rotaci a stisk klávesy S přepne mód na změnu velikosti.

4.2.5 Změna geometrie objektu

Změna geometrie se provádí nad označeným objektem v módu Vertices. Geometrie může být konkrétní (kruh, čtverec, krychle, koule, válec, torus) nebo obecná, specifikovaná pouze vrcholy a hranami mezi nimi. Po změně pozice vrcholu se z konkrétní geometrie stává obecná geometrie.

Konkrétní geometrie

Kruh (Circle) Pravidelný n-úhelník.

Radius - poloměr kruhu, číselná hodnota, v základu 1.

Segments - počet segmentů ze kterých je kruh poskládán, celočíselná hodnota, v základu 8. Čím více segmentů, tím je kruh přesnější.

Theta Start - úhel, na kterém bude začínat vykreslování kruhu. Číselná hodnota v radiánech, v základu 0.

Theta Length - úhel kruhové výseče. Číselná hodnota v radiánech, v základu $2 * \pi$

Čtverec (Plane) Obdélník složený z trojúhelníkových ploch, v základním nastavení se stejně dlouhými stranami.

Width - šířka, číselná hodnota, v základu 1.

Height - výška, číselná hodnota, v základu 1.

Width Segments - počet segmentů na šířku, celočíselná hodnota, v základu 1

Height Segments - počet segmentů na výšku, celočíselná hodnota, v základu 1

Krychle (Cube) Kvádr složený z trojúhelníkových ploch.

Width - šířka, číselná hodnota, v základu 1.

Height - délka, číselná hodnota, v základu 1.

Depth - hloubka, číselná hodnota, v základu 1.

Width Segments - počet segmentů na šířku, celočíselná hodnota, v základu 1

Height Segments - počet segmentů na výšku, celočíselná hodnota, v základu 1

Depth Segments - počet segmentů v hloubce, celočíselná hodnota, v základu 1

Koule (Sphere)

Radius - poloměr koule, číselná hodnota, v základu 1.

Width Segments - počet segmentů na šířku, celočíselná hodnota, v základu 1. Čím je počet segmentů vyšší, tím je koule přesnější.

Height Segments - počet segmentů na výšku, celočíselná hodnota, v základu 1. Čím je počet segmentů vyšší, tím je koule přesnější.

Phi Start - úhel, na kterém začíná kulová výseč ve výšce koule. Číselná hodnota v radiánech, v základu 0.

Phi Length - úhel kulové výseče ve výšce koule. Číselná hodnota v radiánech, v základu $2 * \pi$

Theta Start - úhel, na kterém začíná kulová výseč v šířce koule. Číselná hodnota v radiánech, v základu 0.

Theta Length - úhel kulové výseče v šířce koule. Číselná hodnota v radiánech, v základu $2 * \pi$

Válec (Cylinder)

Radius Top - poloměr horní podstavy, číselná hodnota, v základu 1.

Radius Bottom - poloměr dolní podstavy, číselná hodnota, v základu 1.

Height - výška, číselná hodnota, v základu 1.

Radial Segments - počet segmentů na v podstavách, celočíselná hodnota, v základu 8. Čím je počet segmentů vyšší, tím je válec přesnější.

Height Segments - počet segmentů na výšku, celočíselná hodnota, v základu 1.

Open Ended - příznak, jestli je válec otevřený nebo má podstavy. Logická hodnota, v základu logická 0 (*false*).

Theta Start - úhel, na kterém začíná válcová. Číselná hodnota v radiánech, v základu 0.

Theta Length - úhel válcové výseče. Číselná hodnota v radiánech, v základu $2 * \pi$

Torus

Radius - vnitřní poloměr toru, číselná hodnota, v základu 1.

Tube - poloměr kružnice, jejíž rotací torus vzniká, číselná hodnota, v základu 0.5.

Radial Segments - počet segmentů v kružnici, jejíž rotací torus vzniká. Celočíselná hodnota, v základu 32. Čím je počet segmentů vyšší, tím je torus přesnější.

Tubular Segments - počet rotačních segmentů toru. Celočíselná hodnota, v základu 32. Čím je počet segmentů vyšší, tím je torus přesnější.

Arc - část kruhu, kterou torus zaujímá. Číselná hodnota v radiánech, v základu $2 * \pi$

Obecná geometrie

Obecnou geometrii lze měnit pohybem jednotlivých vrcholů. Vrcholy lze označovat kliknutím myši na okolí vrcholu a samotné označování má několik módů:

Obyčejné kliknutí. Označí neoznačený bod, případně odznačí označený bod. Vždy označí pouze jeden bod.

Kliknutí s držením klávesy Ctrl. Přidá bod do seznamu označených bodů, pokud je neoznačený, v opačném případě jej odebere ze seznamu označených bodů.

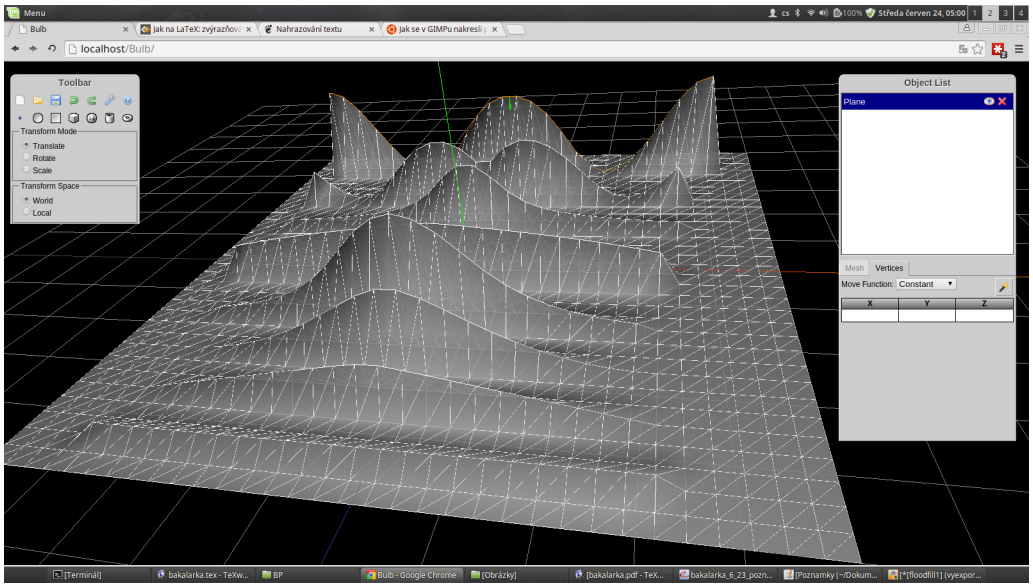
Kliknutí s držením klávesy Shift. Označí všechny body na nejkratší spojnici mezi posledním bodem v seznamu označených bodů a právě označovaným bodem. Tato funkce umožňuje označovat uzavřené oblasti.

Kliknutí na vrchol po stisknutí klávesy F. Označí všechny body uvnitř uzavřené oblasti. Uzavřenou oblastí se rozumí oblast, jejíž hranici tvoří množina označených vrcholů a hran mezi nimi.

Po označení lze body posouvat několika různými způsoby:

- Myší za pomocí směrové růžice.
- Myší nebo šipkami na klávesnici po vybrání osy. Výběr osy se provádí stiskem klávesy X, Y nebo Z podle požadované osy.
- Šipkami na klávesnici ve směru normály. Tato funkce je dostupná po stisknutí klávesy N. Polohu bodů lze rovněž definovat přímým zapsáním do tabulky v pravém panelu.Pokud jsou označeny více jak tři body, tabulka je společná pro všechny označené body, tj. všem označeným bodům se nastaví zapsaná hodnota.

Vrcholy lze pohybovat po různých křivkách určených matematickými funkcemi, ve kterých je parametrem vzdálenost od těžiště označených vrcholů. Výsledek lze vidět na obr. 4.2.

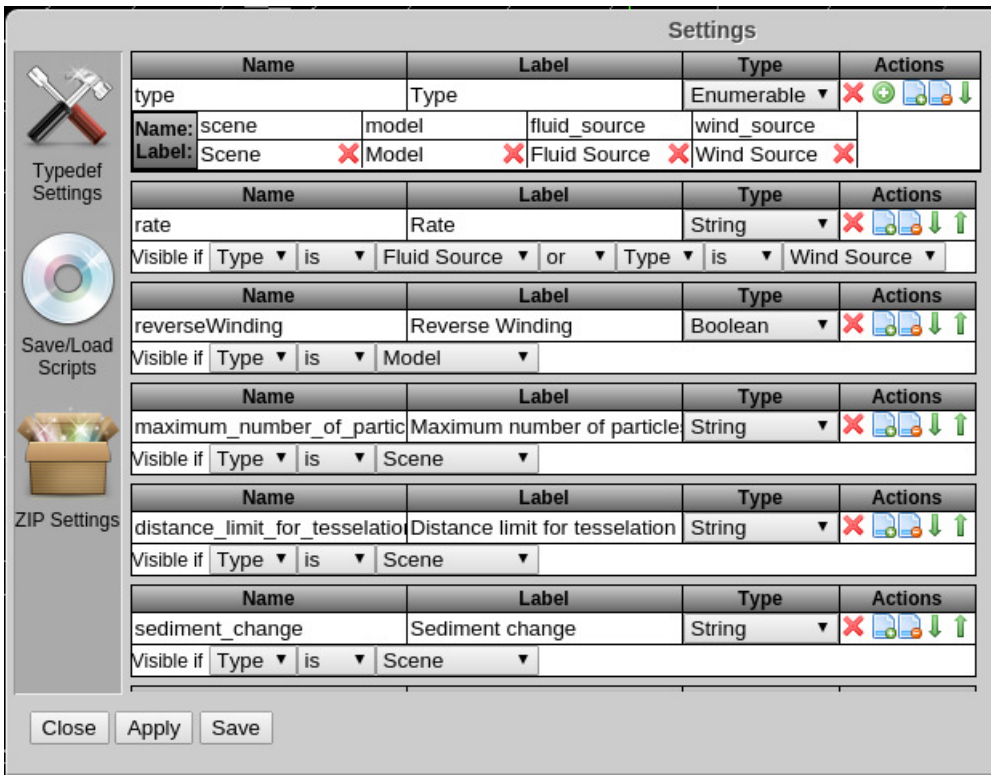


Obrázek 4.2: Zepředu: konstanta, lineární funkce, kvadratická funkce, exponenciální funkce, přirozený logaritmus, hyperbola, sinus (posunutý konstantní funkcí nahoru aby byl lépe vidět), kosinus (posunutý konstantní funkcí nahoru aby byl lépe vidět)

4.2.6 Další vlastnosti

Uživatelské vlastnosti objektů

Objektu lze přiřadit uživatelem definovaná data. Hodnotu těchto dat lze editovat přímo na označeném objektu v módu *Mesh*. Názvy, typy a vlastnosti těchto dat se definují v nabídce *Settings* na panelu nástrojů v levém panelu. Novou vlastnost přidáme kliknutím na tlačítko *Add variable*. Objeví se nová položka obsahující položky *Name*, *Label*, *Type* a ikonky akcí. *Name* (název) smí obsahovat pouze alfanumerické znaky bez mezer a je identifikátorem ve skriptech, které tuto proměnou mohou používat. *Label* (návěstí) smí obsahovat jakékoliv znaky a je popiskem proměnné. Tato hodnota se zobrazuje jako popisek v záložce *Mesh*. Type (typ) definuje typ proměnné. Na výběr je *String*, *Boolean* a *Enumerable*. *String* (řetězec) znamená, že proměnná může obsahovat libovolnou posloupnost znaků. *Boolean* znamená, že proměnná může obsahovat logickou 0 (*false*) nebo logickou 1 (*true*). *Enumerable* znamená, že hodnota proměnné může nabývat pouze hodnot ze seznamu. Tlačítka akcí:



Obrázek 4.3: Dialogové okno Settings

Delete - smazání položky.

Add Option - pouze je-li vybrán typ *Enumerable*. Přidá další položku do seznamu možných položek typu *Enumerable*. Proměnná typu *Enumerable* se v záložce *Mesh* jeví jako vysouvací seznam. Položky tohoto seznamu jsou vypsané v tabulce pod proměnou. První je vždy název, tj. identifikátor pro skripty a platí pro něj to samé, co pro název proměnné. Druhou položkou je návěstí, tj. popisek, pod kterým bude položka ve vysouvacím seznamu. Následuje křížek na smazání této položky ze seznamu.

Add Rule - přidá zobrazovací pravidlo. Zobrazovací pravidlo definuje, kdy je proměnná v záložce *Mesh* viditelná. Pravidla se odvíjí z ostatních proměnných, které jsou typu *Enumerable*, a dají se skládat jako věta.

Move Down - pouze není-li položka poslední. Posune položku níže.

Move Up - pouze není-li položka první. Posune položku výše.

Vedle tlačítka *Add variable* se nalézá ještě tlačítko *Restore Defaults*, které obnoví základní nastavení uživatelských vlastností.

Uživatelské skripty pro export a import scény

Scénu lze importovat a exportovat v *ZIP* archivu. Archiv se skládá z *OBJ* souborů, ve kterých jsou uložené objekty ve scéně a dalšího souboru, který definuje dodatečné vlastnosti scény. Co přesně bude v *ZIP* archivu lze nadefinovat v panelu *Settings* pod ikonou *ZIP settings*, přičemž se vychází z uživatelských proměnných na objektech, které jsou typu *Enumerable*. Podmínka pro vložení objektu ve formě *OBJ* souboru do *ZIP* archivu se potom sestavuje stejně jako podmínka pro zobrazení proměnné jak bylo popsáno výše.

Soubory s popisem scény jsou v archivu vždy všechny, tak jak jsou definovány uživatelem (tj. je-li uživatelem řečeno, že se popis scény bude generovat ve formátu TXT a *XML*, budou v *ZIP* archivu oba soubory, jak `*.txt`, tak `*.xml`).

Skript na sestavování či parsování souboru s popisem scény lze vytvořit a editovat v panelu *Settings* pod ikonou *Save/Load Scripts*. V prvním rozbalovacím seznamu jsou již vytvořené skripty. Pokud chceme vytvořit nový skript, necháme tento rozbalovací seznam prázdný. Druhý rozbalovací seznam s titulkem Type určuje, zda-li skript slouží pro export nebo import. V poli *Label* se nachází krátký popisek skriptu. Ten bude také následně vidět v prvním rozbalovacím seznamu. V posledním poli *Extension* se nachází přípona, kterou bude mít soubor vygenerovaný tímto skriptem. Tlačítkem *Add* následně přidáme skript do seznamu.

Samotný skript lze psát v jednoduchém editoru níže. Tento editor umí číslování řádků, obarvování syntaxe a jednoduché našeptávání. Skripty se zapisují v jazyce *JavaScript* (*ECMAScript*). Jazyk *JavaScript* byl zvolen pro jeho relativní jednoduchost a všeobecnou dostupnost.

Exportní skript Základem každého exportního skriptu je funkce přijímající jako jediný parametr objekt `scene`. Výstupem této funkce je textový řetězec, který bude následně uložen do souboru. Nejjednodušším předpisem této funkce bude tedy:

```
function(scene) {
    return '';
}
```

Přehled použitelných objektů a jejich vlastností:

scene.children - pole s objekty na scéně.

scene.userData - objekt s uživatelem definovaným vlastnostmi.

scene.getObjectByName(name) - vrátí objekt, který se jmenuje name.

Přehled vlastností objektu získaného iterací nad scene.children:

object.position.x - pozice objektu na ose X.

object.position.y - pozice objektu na ose Y.

object.position.z - pozice objektu na ose Z.

object.rotation.x - rotace objektu kolem osy X.

object.rotation.x - rotace objektu kolem osy Y.

object.rotation.x - rotace objektu kolem osy Z.

object.scale.x - škálování objektu na ose X.

object.scale.y - škálování objektu na ose Y.

object.scale.z - škálování objektu na ose Z.

object.userData - objekt s uživatelem definovaným vlastnostmi.

Importní skript Základem importního skriptu je funkce přijímající parametr data, což je řetězec s obsahem importovaného souboru a objekt canvas, což je objekt kreslicí plochy.

Přehled metod a vlastností objektu canvas:

canvas.getScene() - vrátí objekt scény.

canvas.addVector(name) - vloží do scény bod pojmenovaný name.

canvas.addCircle(name) - vloží do scény kruh pojmenovaný name.

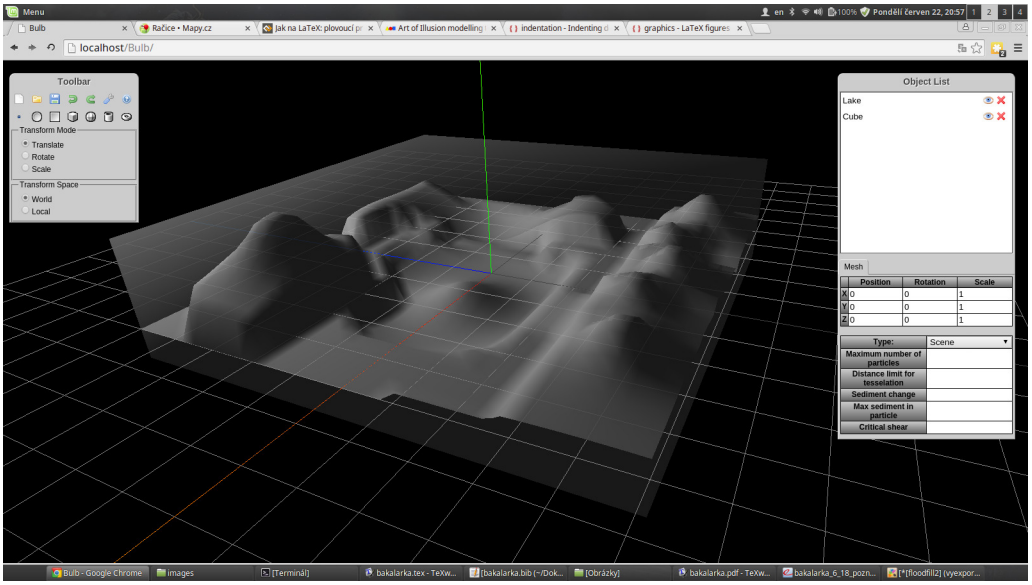
canvas.addPlane(name) - vloží do scény čtverec pojmenovaný name.

canvas.addSphere(name) - vloží do scény kouli pojmenovanou name.

canvas.addCube(name) - vloží do scény krychli pojmenovanou name.

canvas.addCylinder(name) - vloží do scény válec pojmenovaný name.

canvas.addTorus(name) - vloží do scény torus pojmenovaný name.



Obrázek 4.4: Scéna s jezerem

V programu je již implementován výchozí exportní a importní skript pro textový formát. Soubor s daty o scéně má příponu `*.settings` a jeho obsah vygenerovaný pro scénu na obrázku 4.4 vypadá takto:

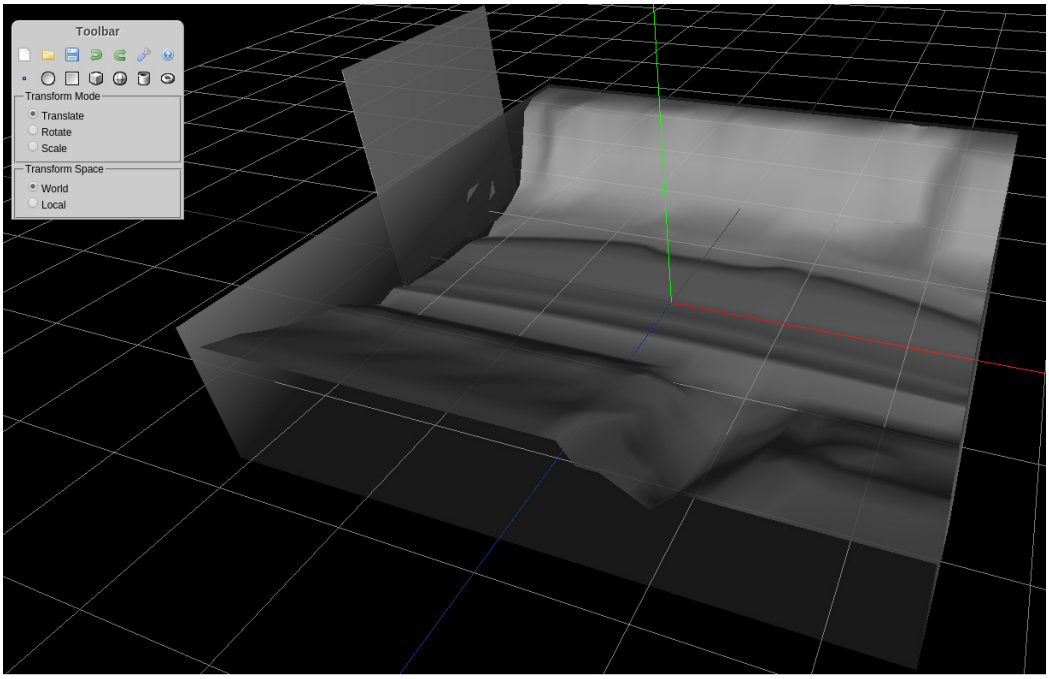
```
# Scene settings
----------------
# models (filename reverseWinding scaleFactor translation)
m cube.obj 0 100/100/100 0/0/0
m lake.obj 1 10/10/10 0/0/0

# fluid source (rate position size)
f 10 -20/-10/40 10/10/10

# wind source (rate position size)
w 10 -20/-25/-10 40/1/39

# maximum number of particles
p 20000
```

```
# distance limit for tesselation
d 100

# material (sediment_change max_sediment_in_particle critical_shear)
s 0.02 1.0 0.25
```

Také je již implementován výchozí exportní skript pro formát *XML*. Soubor s daty má příponu `*.xml` a jeho obsah vypadá takto:

```
<scene maxNumbersOfParticles="20000" distanceLimitForTesselation="100" >
  <model file="cube.obj" reverseWinding="0">
   <position x="0" y="0" z="0" />
   <scale x="100" y="100" z="100" />
   <material sedimentChange="0.02" maxSedimentInParticle="1.0" criticalShear="0.25" />
  </model>
  <model file="lake.obj" reverseWinding="1">
    <position x="0" y="0" z="0" />
    <scale x="10" y="10" z="10" />
    <material sedimentChange="0.02" maxSedimentInParticle="1.0" criticalShear="0.25" />
  </model>
  <fluidSource rate="10">
    <position x="-20" y="-10" z="40" />
    <scale x="10" y="10" z="10" />
  </fluidSource>
  <windSource rate="10">
    <position x="-20" y="-25" z="-10" />
    <scale x="40" y="1" z="39" />
  </windSource>
</scene>
```

4.3 Ukázka složitější scény



Obrázek 4.5: Model krajiny v bezprostředním okolí obce Račice

Pro ukázku vytvoření scény na obrázku 4.5 jsem si vybral bezprostřední okolí obce Račice, okres Rakovník, na středním toku řeky Berounky. Postupné modelování této scény si můžete prohlédnout na obrázkovém tutoriálu v příloze A.

5 Závěr

Po zhodnocení stávajícího, zejména svobodného softwaru pro modelování ve 3D byla vytvořena webová aplikace se zaměřením na modelování krajinného reliéfu, s možností ukládat scénu ve formátu, který zachovává dodatečné parametry scény a objektů v ní, jež jsou důležité pro správnou funkci softwaru pro simulaci vlivu eroze na terén. [Skorkovská(2012)] Z charakteristických funkcí, kvůli kterým byl program vytvářen bych rád zmínil systém označování a manipulace vrcholů na velkých plochách, možnost volby funkce posunu (konstantní, lineární, kvadratická, exponenciální, logaritmická, hyperbolická, sinus a kosinus), možnost přidat objektům ve scéně dodatečná metadata ohledně erozivního působení (např. koeficient vymílání apod.) a možnost editovat strukturu a typ těchto metadat.

Z použité technologie vyplývají některá omezení, například nemožnost vyvolat dialog "Uložit jako..."a ke každému souboru se tudíž přistupuje, jako by byl stažen z internetu.

Během práce jsem musel řešit hlavně problém s výkonem zvoleného řešení, zejména při manipulaci s jednotlivými vrcholy. Poprvé tento problém nastal kvůli vypisování seznamu vrcholů do záložky *Vertices*. Další zrychlení nastalo po přepsání třídy THREE.GridHelper (zvýraznění trojúhelníkové sítě). Původní THREE.GridHelper z knihovny *three.js* totiž neuměl měnit svou geometrii a proto bylo nutné vždy vytvořit novou instanci. Poslední úprava spočívala ve zvýraznění označených vrcholů. Původně byla do každého vrcholu umístěna malá krychle. Toto řešení opět při větším množství označených vrcholů neúměrně zatěžovalo systém a proto bylo nahrazeno současným řešením spočívajícím ve vyznačení hran mezi sousedními označenými vrcholy.

Vytvořením modelu krajinného reliéfu okolí obce Račice, okres Rakovník, na středním toku řeky Berounky jsem si otestoval použitelnost a uživatelskou přívětivost aplikace při vytváření modelů krajinných reliéfů ve 3D.

Literatura

[aoi(a)] Contact - Art of Illusion. http://www.artofillusion.org/contact, a. online, citováno 21. 6. 2015.

[aoi(b)] License - Art of Illusion. http://www.artofillusion.org/license, b. online, citováno 21. 6. 2015.

[ble(2014a)] Download - blender.org. https://www.blender.org/download/, 2014a. online, citováno 21. 6. 2015.

[ble(2008)] History - blender.org. https://www.blender.org/foundation/history/, 2008. online, citováno 21. 6. 2015.

[ble(2014b)] License - blender.org. https://www.blender.org/about/license/, 2014b. online, citováno 21. 6. 2015.

[fcc()] FreeCAD/FreeCAD_sf_master. https://github.com/FreeCAD/FreeCAD_sf_master. online, citováno 21. 6. 2015.

[fcl(2014)] License - FreeCAD. http://www.freecadweb.org/wiki/index.php?title=Licence, 2014. online, citováno 21. 6. 2015.

[max(a)] Autodesk | Design, Engineering, & Entertainment Software. http://www.autodesk.com/, a. online, citováno 21. 6. 2015.

[max(b)] 3D modeling and animation tools. http://www.autodesk.com/products/3ds-max/features/all/gallery-view, b. online, citováno 21. 6. 2015.

[max(c)] System requirements for Autodesk 3ds Max 2016. http://knowledge.autodesk.com/support/3ds-max/troubleshooting/caas/sfdcarticles/sfdcarticles/System-requirements-for-Autodesk-3ds-Max-2016.html, c. online, citováno 21. 6. 2015.

[may(a)] Toolsets for character creation and animation. http://www.autodesk.com/products/maya/features/all/gallery-view, a. online, citováno 21. 6. 2015.

[may(b)] Python/Maya: Introductory tutorial. http://cgkit.sourceforge.net/maya_tutorials/intro/, b. online, citováno 21. 6. 2015.

[may(c)] System requirements for Autodesk Maya 2016. http://knowledge.autodesk.com/support/maya/troubleshooting/caas/sfdcarticles/sfdcarticles/System-requirements-for-Autodesk-Maya-2016.html, c. online, citováno 21. 6. 2015.

[Ashkenas et al.(2012)] ASHKENAS, J. et al. CoffeeScript, 2012. online, citováno 21. 6. 2015.

[Dijkstra(1959)] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische mathematik*. 1959, 1, 1, s. 269–271.

[Foundation(2015)] FOUNDATION, M. DOM Storage guide. https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Storage, 2015. online, citováno 21. 6. 2015.

[Johari()] JOHARI, S. Hindu Goddesses : Maya - Hindu Goddess. http://www.sanatansociety.org/hindu_gods_and_goddesses/maya.htm. online, citováno 21. 6. 2015.

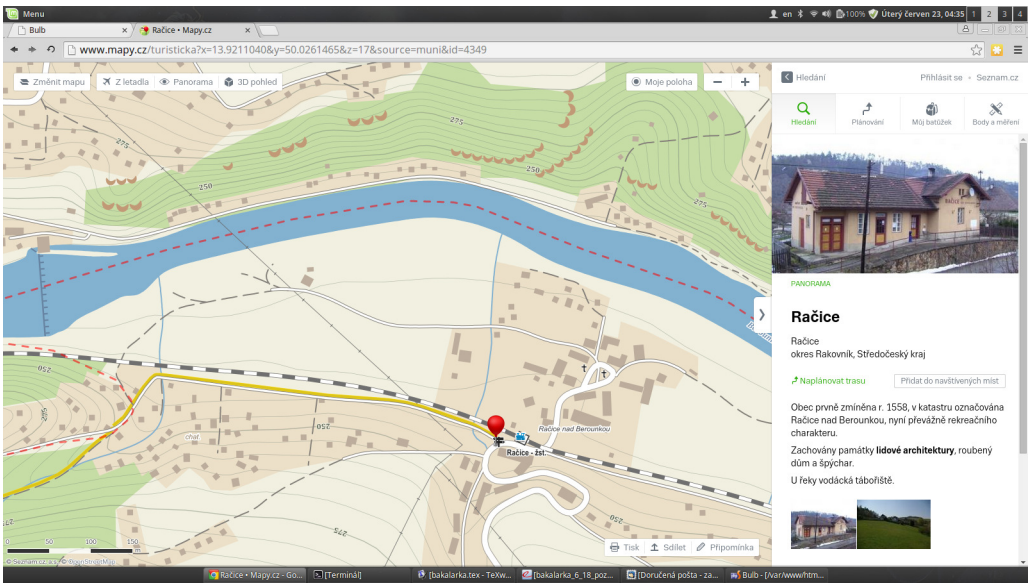
[Nampoothiri(2012)] NAMPOOTHIRI, H. N. Let's do LESS. 2012.

[Skorkovská(2012)] SKORKOVSKá, V. Modeling of Erosion Impacts on the Terrain. Master's thesis, University of West Bohemia, Pilsen, Czech Republic, 2012.

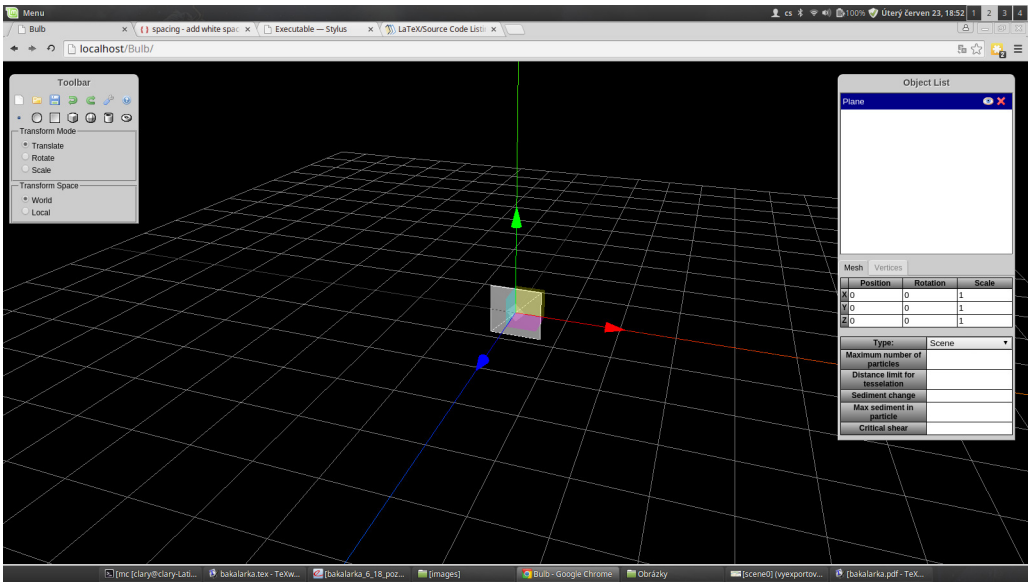
[Sobota(1995)] SOBOTA, B. *Počítačová grafika a jazyk C*. České Budějovice : Kopp, 1995.

[three.js authors(2012)] AUTHORS. three.js Features. https://github.com/mrdoob/three.js/wiki/Features, 2012. online, citováno 21. 6. 2015.

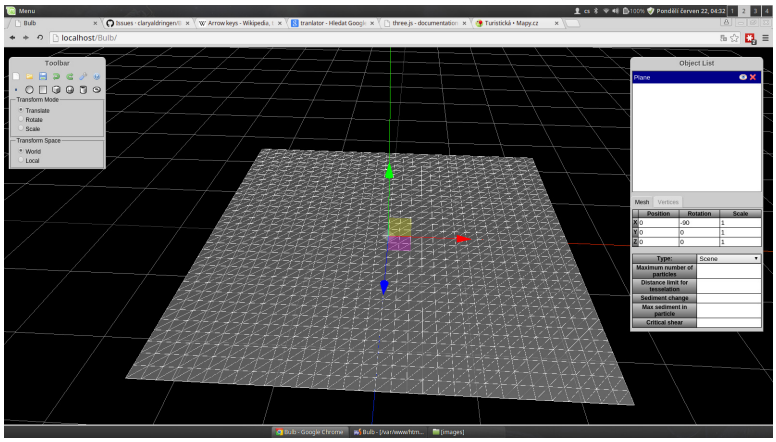
A Tvorba scény krok po kroku



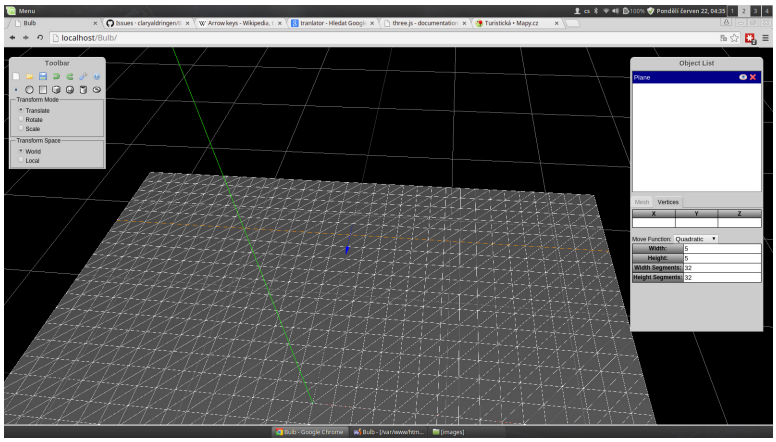
Obrázek A.1: Mapa obce Račice a blízkého okolí.



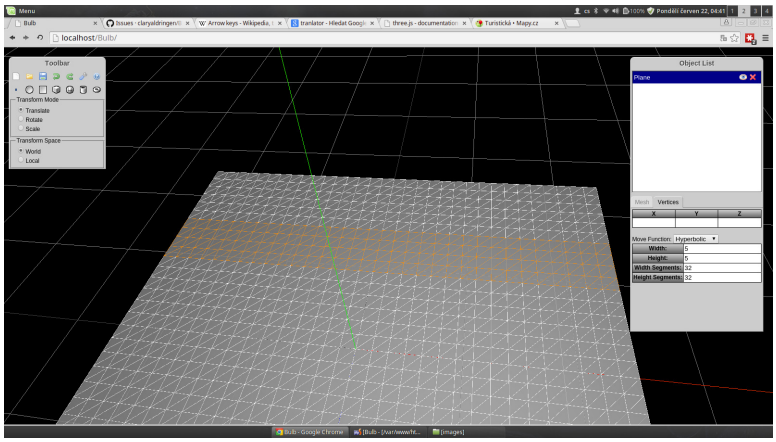
Obrázek A.2: Nástrojem Plane vytvoříme plochu.



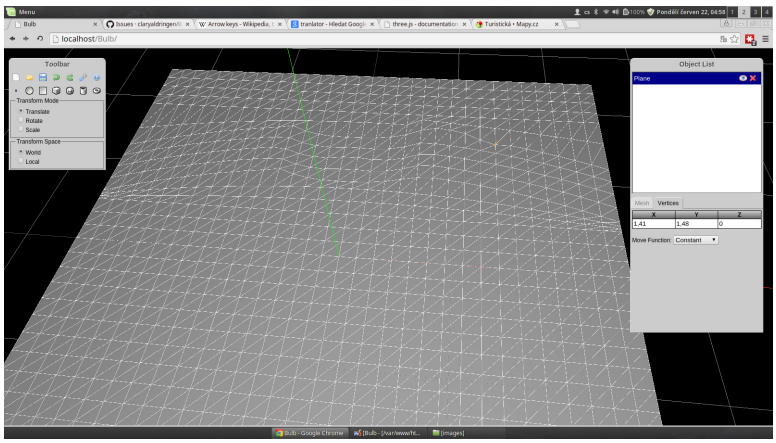
Obrázek A.3: Vytvořenou plochu v záložce Vertices zvětšíme na velikost 5x5 a rozdělíme na 32x32 plošek.



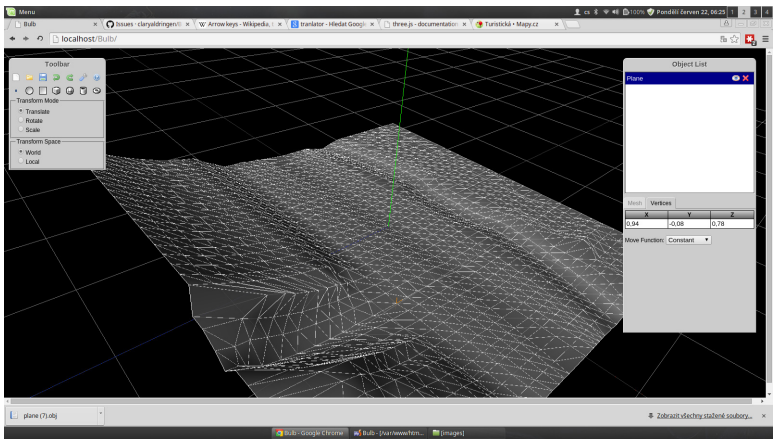
Obrázek A.4: Na ploše označíme krajní body se stisknutou klávesou Shift a označí se všechny body na nejkratší spojnici mezi nimi.



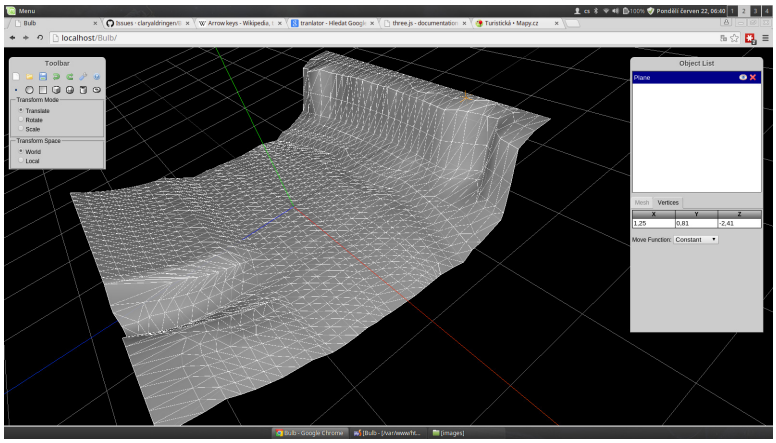
Obrázek A.5: Postupně označíme uzavřenou plochu a poté označíme za pomocí klávesy F vnitřek plochy.



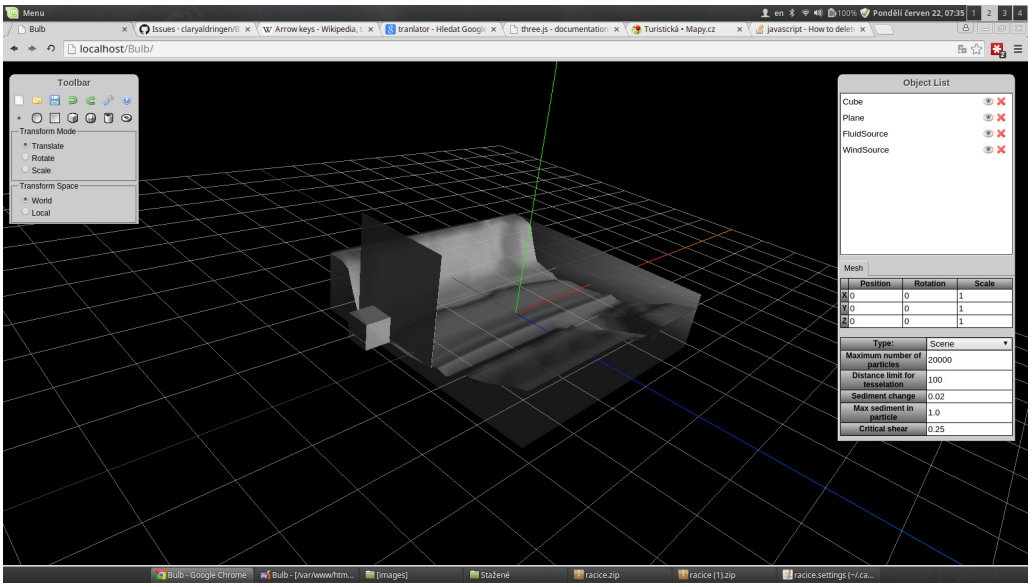
Obrázek A.6: Posunem podle osy z za použití různých funkcí vymodelujeme zakřivení koryta řeky.



Obrázek A.7: Vymodelované koryto řeky s přítokem Račického potoka.



Obrázek A.8: Vymodelována skalní stěna naproti obci Račice.



Obrázek A.9: Scéna doplněna o zdroj vody a větru.

Hotovou scénu uložíme do počítače tlačítkem *Export...* na panelu nástrojů. V zobrazeném dialogovém okně vybereme možnost *Scene and terrain settings (*.zip)*. Po kliknutí na tlačítko *Download* bude scéna ve formátu *ZIP* stažena do počítače.

Seznam obrázků