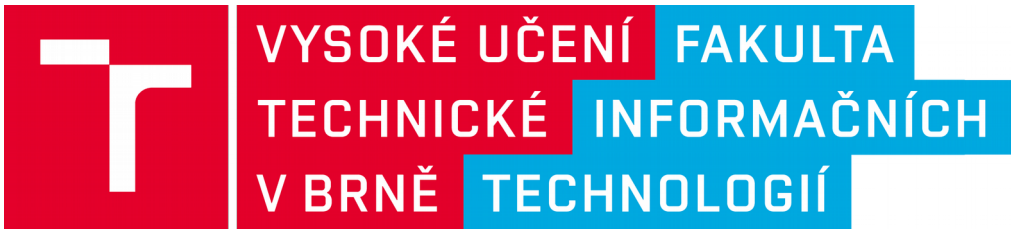
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZOBRAZENÍ 3D SCÉNY VE WEBOVÉM PROHLÍŽEČI

3D SCENE VISUALIZATION IN A WEB BROWSER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ MAHR

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL ŠPANĚL Ph.D.

BRNO 2017

Abstrakt

Tato práce se zabývá návrhem a implementací interaktivního prohlížeče 3D modelů ve webovém prohlížeči. Cílem práce je věrné představení průmyslových a komerčních výrobků potenciálním zákazníkům daného produktu. Výsledná aplikace umožňuje rozklad objektu na jednotlivé díly a speciální poloprůhledné zobrazení se zvýrazněním hran. Pro implementaci je využita webová technologie WebGL a knihovna Three.js.

Abstract

This thesis deals with design and implementation of an interactive 3D model viewer in a web browser. Aim of the thesis is to faithfully present industrial and commercial products to potential customers of the product. The final application allows to decompose the object into individual parts and a special semi-transparent edge-highlighted display. The WebGL web technology and the Three.js library are used for implementation.

Klíčová slova

WebGL, 3D, GUI, JavaScript, HTML5, shader, prohlížeč 3D modelů

Keywords

WebGL, 3D, GUI, JavaScript, HTML5, shader, 3D model viewer

Citace

MAHR, Lukáš. *Zobrazení 3D scény ve webovém prohlížeči*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Španěl Michal.

Zobrazení 3D scény ve webovém prohlížeči.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Michala Španěla Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Mahr
17. května 2017

Poděkování

Děkuji vedoucímu mé práce Ing. Michalu Španělovi, Ph.D., za veškeré rady, připomínky a odbornou pomoc při tvorbě této bakalářské práce.

Obsah

Kapitola 1

Úvod

Na počátku mnoha komerčních a průmyslových výrobků je digitální 3D prototyp nebo kompletní model produktu. Tvorba modelů je výhodná díky možnosti otestovat konečný produkt před započetím výroby a ušetřit tak náklady za materiál a výrobní proces. Modernizace technologií také umožňuje vytváření 3D modelů skenováním reálných objektů nebo převod z 2D materiálů do více rozměrného prostoru. Modely lze snadno a rychle revidovat a aktualizovat.

Právě trojrozměrné zobrazení modelu výrobku dává zákazníkovi nejvěrnější představu o výsledném produktu. Výhodou tohoto znázornění oproti dvourozměrnému zobrazení pomocí fotografií a jiných materiálů, je možnost „osahání“ si produktu díky pohledu ze všech stran, pohledu dovnitř modelu nebo rozložení modelu na jednotlivé díly a součástky. Atraktivní je také možnost změny barvy či textury, což je vhodné například pro vybrání finální podoby produktu před objednáním. Takovým případem může být například výběr barvy vozidla, provedení interiéru, změna typu kol a jiných položek, kdy zákazník před samotnou koupí vidí výslednou podobu vozidla ze všech stran a pohledů pouhým pohybem myši a v pohodlí webového prohlížeče.

Trojrozměrná vizualizace produktů je zejména vhodná u složitých propracovaných modelů, jako jsou různé stroje skládající se z mnoha součástí, technicky složité díly a součástky, designově výjimečné objekty nebo objekty složené z mnoha do sebe zapadajících dílů. Cílem je předat zákazníkovi co nejvíce informací o daném objektu a navodit pocit skutečné hmatatelné věci. To vede ke zvýšení zájmu o produkt a větší pravděpodobnosti prodeje nebo splnění jiného cíle. Vždy záleží za jakým účelem data vizualizujeme.

V rámci této práce vznikl interaktivní prohlížeč 3D modelů postavený na knihovně Three.js. Kromě běžných funkcí, které poskytují podobné aplikace tohoto typu, jako je například otáčení a přiblížení modelu, má prohlížeč několik funkcí navíc. Jednou z nich je animované rozložení modelu na jednotlivé díly, což dává uživateli pocit skutečného objektu a lepší představu o jeho složení. Další je možnost zobrazení poloprůhledného modelu se zvýrazněnými hranami, simulující rentgenové záření (X-Ray).

Na začátku této práce srovnám dostupná a nejvíce používaná řešení pro zobrazení modelů ve webovém prohlížeči (Kapitola 2).Ve třetí kapitole (3) popíši dostupné a využívané technologie pro implementaci prohlížeče 3D modelů do webového prohlížeče. Požadavky a nároky na aplikaci budou popsány v kapitole 4 a následně popíši finální implementaci webového prohlížeče (kapitola 5). Výsledky implementace jsou ukázány v kapitole 6. Nakonec předvedu aplikaci potenciálním zákazníkům (kapitola 7) a na úplný závěr shrnu celou práci v kapitole 8.

Kapitola 2

Existující řešení pro zobrazení 3D modelu ve webovém prohlížeči

Díky velké podpoře WebGL mezi zařízeními a prohlížeči (Obrázek 3.1), stále rychlejší výpočetní technice a růstu povědomí o možnostech 3D vizualizací a virtuální realitě existuje mnoho hotových řešení, které umožňují zobrazení 3D obsahu ve webovém prohlížeči a přibývají stále nová. Z pohledu dostupnosti a použití můžeme tyto aplikace rozdělit do dvou skupin: cloudová řešení a pluginy (rozšíření) pro zobrazení 3D modelů ve webovém prohlížeči.

2.1 Definice cloudu

Cloud je obecný pojem pro cokoliv, kde jde o poskytování pronajímaných služeb přes Internet. Slovo samotné znamená anglicky mrak a pochází ze zvyku kreslit ve schematických obrázcích komunikaci přes Internet jako obrázek mraku [1]. Cloud dělíme do tří modelů služeb: SaaS, PaaS a laaS. V rámci této práce si popíšeme pouze první z uvedených modelů SaaS (Software-as-a-Service). Tento model poskytuje nezávislost na koncovém zařízení. Jeho služby mohou být použity z libovolného počítače nebo chytrého zařízení připojeného k Internetu, například osobního počítače, notebooku, smartphonu, tabletu, chytrého televizoru, čtečky elektronických knih a dalších zařízení. Příkladem cloudových aplikací jsou sociální sítě (Facebook, Google+), knihovny obrázků (Picasa, Flicker) nebo knihovny videí (Youtube, Vimeo).

Obrázek 2.1: Schéma cloudu typu Saas.

2.2 Cloudová řešení

Tyto aplikace jsou založené na cloudu (2.1), který běží ve webovém prohlížeči. Zpravidla umožňují nejen zobrazení modelu ve webovém prohlížeči 3D modelů na vlastních stránkách, ale také vložení okna s 3D scénou do vlastní webové prezentace nebo sdílení scény na různé sociální sítě a portály. Modely lze snadno nahrávat na vzdálený server přes webové rozhraní.

Výhodou těchto aplikací je snadná obsluha. K nahrání a zobrazení modelů není potřebná instalace žádného softwaru ani rozšíření. Stejně tak není nutná znalost WebGL a dalších webových technologií. Aplikace disponují kvalitní podporou a širokou komunitou uživatelů. Jsou tedy velice vhodné pro získávání zpětné vazby co se týče tvorby 3D modelů. Za těmito aplikacemi stojí poměrně velké společnosti a početné týmy vývojářů. Díky tomu jsou poskytované služby velice kvalitní a stále se vyvíjí.

Nevýhodami pak mohou být omezené služby při registraci zdarma. U všech níže zmíněných aplikací jsou základní funkce zdarma. Plné využití aplikací je pak zpoplatněno různými měsíčními platebními plány. Ne každému také vyhovuje nahrávání svých dat a modelů na cizí servery. V závislosti na zakoupených plánech jednotlivých aplikací pak také mohou být modely veřejně dostupné jak k prohlížení, tak ke stažení. To je v mnoha případech nežádoucí stejně jako loga aplikací u vkládaných 3D scén do vlastních stránek.

Příklady cloudových WebGL aplikací

Sketchfab.com

Největší světová platforma pro publikování, sdílení a procházení 3D obsahu on-line. Disponuje také možností přepnout procházení modelů a scén do režimu Virtuální reality. Komunita skýtá téměř tři čtvrtě milionu uživatelů a přes milion modelů. Sketchfab je oficiálním partnerem pro sdílení 3D obsahu on-line některých známých, jako je například Adobe Photoshop, sociální síť Facebook nebo Twitter. Aplikace vnikla v roce 2012 v Paříži.

Služby s registrací zdarma	Dostupné služby po rozšíření účtu
Neomezený počet upload	Neomezený počet upload
50MB na soubor	Až 500MB na soubor
5 vysvětlivek ve scéně	Až 50 vysvětlivek ve scénu
Emailová podpora do 48 hodin	Emailová podpora do 12 hodin
	Přizpůsobení vestavěného prohlížeče
	Soukromé scény a scény chráněné heslem
	Nahrávání vlastních pozadí
	Nahrávání vlastního HDR prostředí
	Možnost zařazení do PRO adresáře v galerii modelů
	Získání „PRO“ statusu

Tabulka 2.1: Porovnání služeb Sketchfab.com.

- **Funkce 3D prohlížeče sketchfab:**
  - otáčení kamery kolem objektu a přiblížení
  - přemístění objektu ve scéně
  - sdílení scény pomocí vkládání HTML tagu <iframe>
  - režim virtuální reality
  - ovládání více prsty na dotykových zařízeních

- režim celé obrazovky a režim „Theater“
- možnost zobrazení wireframe (drátěný model)
- změna režimu renderování materiálu
- více režimů kamery
- výběr textury modelu

Obrázek 2.2: Snímek obrazovky 3D prohlížeče aplikace sketchfab.com. Zdroj: https://sketchfab.com.

Clara.io

Clara.io je plnohodnotný 3D modelovací, animační a renderovací software vyvinutý kanadskou společností Exocortex[1]. Clara.io tedy na rozdíl od předchozích aplikací, kromě vykreslování modelů, umožňuje také modelování a editaci modelů. Tato aplikace je dobrým nástrojem pro vytváření a ladění obsahu JavaScriptových WebGL knihoven Three.js a babylon.js.

- **Funkce:**
  - polygonální modelování
  - vektorové modelování geometrických objektů (CSG)
  - key-frame animace a skeletální animace
  - hierarchický graf scény
  - mapování textur
  - fotorealistické vykreslování
  - import a export FBX, Collada, OBJ, STL a Three.js
  - editace scény více spolupracujících uživatelů v reálném čase
  - řízení revizí (verze a historie)
  - skriptování, Pluginy & REST API
  - Knihovna 3D modelů

1 http://exocortex.com

Služby s registrací zdarma	Dostupné služby po rozšíření účtu
5 GB úložiště v cloudu	Až 100 GB úložiště v cloudu
Neomezený počet veřejných scén	Neomezený počet veřejných scén
10 soukromých scén	Neomezený počet veřejných scén
1 hodina osobního vykreslování	Neomezený čas osobního vykreslování
Neomezený počet vykreslovacích úloh	Neomezený počet vykreslovacích úloh
2 souběžné práce	Až 100 souběžných prací
2 úlohy importu a exportu	8 úlohy importu a exportu
WebGL vkládání scény	WebGL vkládání scény s kustomizací
V-Ray vkládání scény	V-Ray vkládání scény s kustomizací
Podpora prostřednictvím veřejných fór	Podpora prostřednictvím emailu nebo skype.
	Prioritní vykreslení
	Bezriziková 30denní zkušební verze

Tabulka 2.2: Porovnání služeb Clara.io

- **Funkce 3D prohlížeče modelů clara.io**
  - pohyb kamery kolem objektu a přiblížení/oddálení scény
  - resetováni pozice kamery
  - sdílení scény pomocí vkládání HTML tagu <iframe>
  - snímek scény
  - režim celé obrazovky
  - zobrazení jako obrázek neboWebGL vykreslení

Obrázek 2.3: Snímem obbrazovky 3D prohlížeče Clara.io. Zdroj: https://clara.io/library.

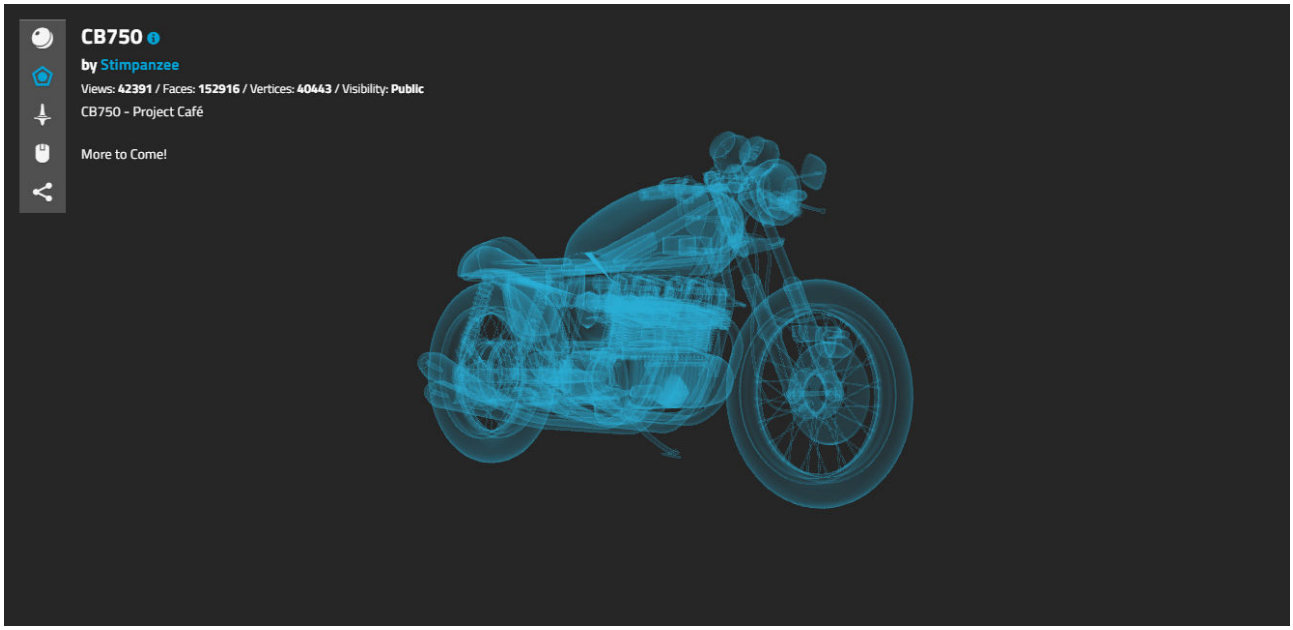
p3d.in

Další velice populární aplikace pro snadné sdílení a prohlížení 3D obsahu on-line. Standardně jsou modely nahrávány do veřejné galerie, kde jsou modely dostupné pro všechny návštěvníky webu. S rozšířeným účtem lze nahrávat soukromé modely, viditelné pouze pro povolené uživatele.

Základní přizpůsobení vložené scény WebGL	Pokročilé přizpůsobení vložené scény
Omezené rozlišení obrázků	Velké rozlišení obrázků
Prostor 100MB	Až 500GB prostoru
Neomezeny počet veřejných modelů	Neomezený počet veřejných modelů
Počet modelů neuvedených v seznamu: 5	Neomezený počet modelů neuvedených v seznamu
10 privátních modelů	Neomezený počet privátních modelů

Tabulka 2.3: Porovnání služeb p3d.in

- **Funkce 3D prohlížeče modelů p3d.in**
  - změna materiálu modelu(wireframe, wireframe s texturou, bez stínů, X-ray a další)
  - automatická rotace modelu
  - sdílení scény pomocí vkládání HTML tagu <iframe>
  - změna stylu navigace
  - funkce zapnutí subdivision surface



Obrázek 2.4: Snímek obrazovky 3D prohlížeče p3d.in. Zdroj: p3d.in

Souhrn

Možností sdílení a vkládaní 3D obsahu na cloudová úložiště je velké množství. Jsou vhodné spíše pro sdílení modelů za účelem vývoje a ladění, než k prezentaci produktů pro prodej a podobné účely, i když tuto funkci umí plnohodnotně zastat.

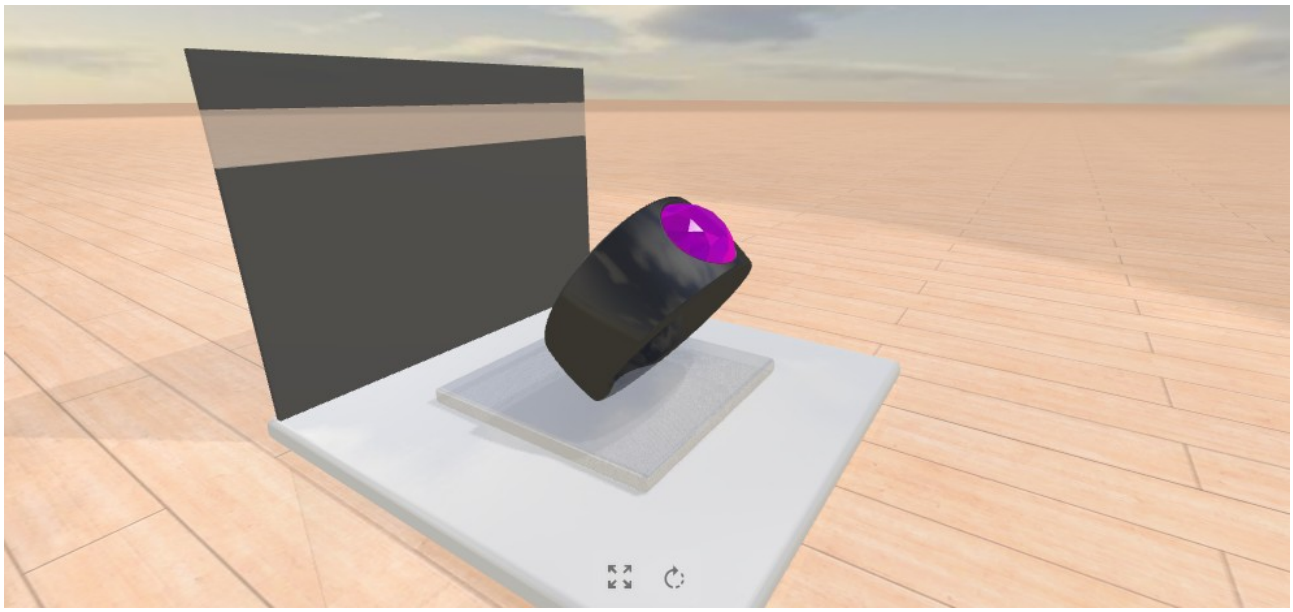
2.3 Pluginy (rozšíření) pro zobrazení 3D modelů ve webovém prohlížeči.

Pokud nechceme svá trojrozměrná data a modely sdílet s dalšími uživateli ale chce mít svá data pod kontrolou a zobrazovat produkty výhradně koncovým zákazníkům, je výhodnější využít řešení pomocí samostatného rozšíření (pluginu) pro webové stránky. Takovému řešení se také věnuje tato práce.

Příklady rozšíření pro zobrazení 3D modelů

Canvasio3D

Jde o plugin od společnosti Canvasio3D pro redakční systém Wordpress. Rozšíření přidává do redakčního systému možnost nahrávání 3D modelů a jejich zobrazení v kdekoliv ve webové prezentaci. Plugin je dostupný v neplacené a placené verzi. Podporované formáty jsou .stl, .dae, .obj & .mtl. Toto řešení je omezené pouze pro systém Wordpress. Neplacená verze poskytuje pouze jednoduchý 3D prohlížeč modelů, více přizpůsobení modelů a jejich zobrazení najdeme v „PRO“ verzi tohoto pluginu.



Obrázek 2.5: Snímek obrazovky rozšíření Canvasio3D .

Zdroj:https://www.canvasio3d.com/canvasio3d-example4-connect-it/?caObjID=20

- **Hlavní funkce rozšíření:**
  - zobrazení více modelů do jedné scény
  - automatické načítání materiálů a textu r(.mtl) u formátu .obj
  - automatické přehrávání animace u formátu .dae (collada)
  - nastavení scény (barva modelu, velikost modelu, odraz světla, lesk, průhlednost, obrázek na pozadí prostředí, model na pozadí, stín objektu, podlaha scény světelné podmínky, zářivost hlavního zdroje světla, pozice a rotace objektu)
  -
- **Fakta:**
  - vznik: 2015
  - poslední aktualizace: Duben 2017
  - postaveno na základe knihovny Three.js
  - velikost JavaScriptu (verze zdarma): 905kb

Příklad použití nalezneme na https://swoppy-shop.com/konfigurator/.

Kento 3D Model Viewer
Další plugin pro zobrazení 3D modelů na systému WordPress.

- **Funkce:**
  - zobrazení 3D modelu
  - podporované formáty: .obj, STL, Autodesk 3DS, OpenCTM
  - rotace o 360 stupňů a možnost přiblížit nebo oddálit pohled

- **Fakta:**
  - vznik: 2015
  - velikost JavaScriptu 146kb
  - postaveno na jsc3d.js
  - poslední aktualizace: 2015

- **Nedostatky:**
  - Pouze základní pohyb s kamerou. Chybí více funkcí.
  - Zastaralý

Obrázek 2.6: Kento 3D Model Viewer. Zdroj:
https://cs.wordpress.org/plugins/kento-3d-model-viewer/.

3D Model Viewer
Poslední z pluginů pro Wordpress.

- **Funcke:**
  - zobrazení 3D modelu
  - podporované formáty: .dae (Collada) a .obj
  - definice pozadí

    - nastavení prostředí
    - nastavení světelných podmínek
    - nastavení směrového světla
    - nastavení pozice kamery a modelů
    - nastavení velikosti modelu

- **Fakta:**
    - Poslední aktualizace: 2016
    - Postaveno na knihovně Three.js

- **Nedostatky:**
    - Chybí další funkce pro zobrazení modelu

Souhrn

Možností nahrání 3D modelů na své webové stránky za účelem propagace výrobků zákazníkům nebo podporu prodeje bez využití cloudových aplikací je velmi málo. Existuje několik modulů pro systém Wordpress ale samostatně stojící modul nebo rozšíření implementující prohlížeč 3D modelů nebyl nalezen. To je jedním z hlavních důvodů vzniku této práce. Chci umožnit vývojářům a administrátorům webových prezentací vkládání 3D obsahu na webové stránky a z druhé strany poskytnou zákazníkům věrohodné a kvalitní informace o daném objektu. Moje řešení půjde použít na jakékoliv webové stránce a jeho instalace bude možná bez hlubokých znalostí webových technologií.

Kapitola 3

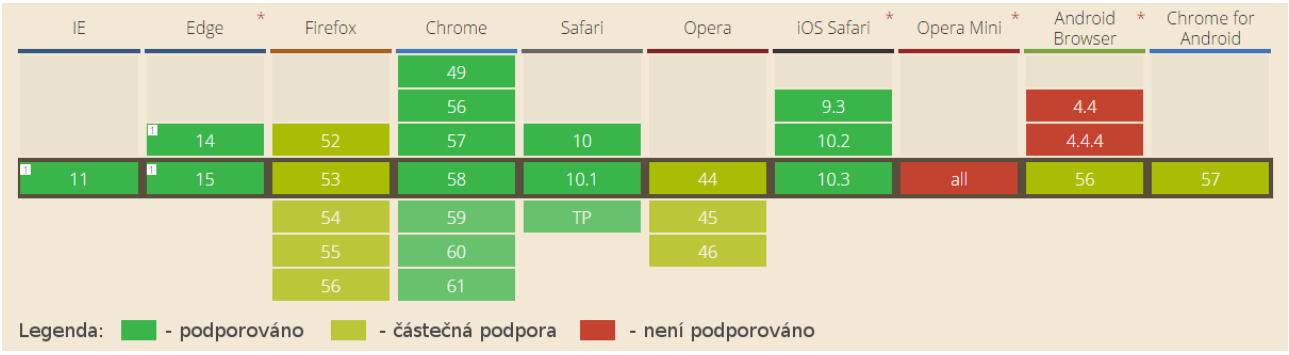
Nástroje pro zobrazení 3D scény ve webovém prohlížeči

3D grafika se objevila s počátky počítačových systémů, které se datují až do šedesátých let minulého století. Je významnou součástí aplikací v oblasti inženýrství, architektury, her a zábavy, výzkumu, průmyslu, prodeje a marketingu a také vzdělávání a školení. Dříve byla 3D grafika a 3D aplikace výsadou špičkových a nákladných počítačových systémů. Dnes je hardware pro zpracování 3D grafiky součástí téměř každého počítače, mobilního a chytrých zařízení. Do webových prohlížečů bylo WebGL přidalo v roce

3.1 Co je WebGL

Web Graphic Library, nebo zkráceně WebGL, je JavaScriptové API, Které umožňuje kompatibilním webovým prohlížečům nativní vykreslování 2D a 3D grafiku bez nutnosti použití zasůvných modulů(pluginů). WebGL programy se skládají ze dvou částí a to obslužného kódu napsaného v JavaScriptu a kódu shaderu, který pprovádí GPU(grafická karta počítače). O vývoj a udržování WebGL se stará nezisková organizace Khronos group[2].

Samotné API WebGL je velice silný nástroj, je ale také poměrné složité. Z důvodu složitosti vzniklo několik knihoven, které práci s API značně ulehčují a zvyšují produktivitu při práci s WebGL 3.2.



Obrázek 3.1: Podpora WebGL ve webových prohlížečích. Zdroj: http://caniuse.com/#feat=webgl.

2 https://www.khronos.org

3.2 Porovnání WebGL knihoven

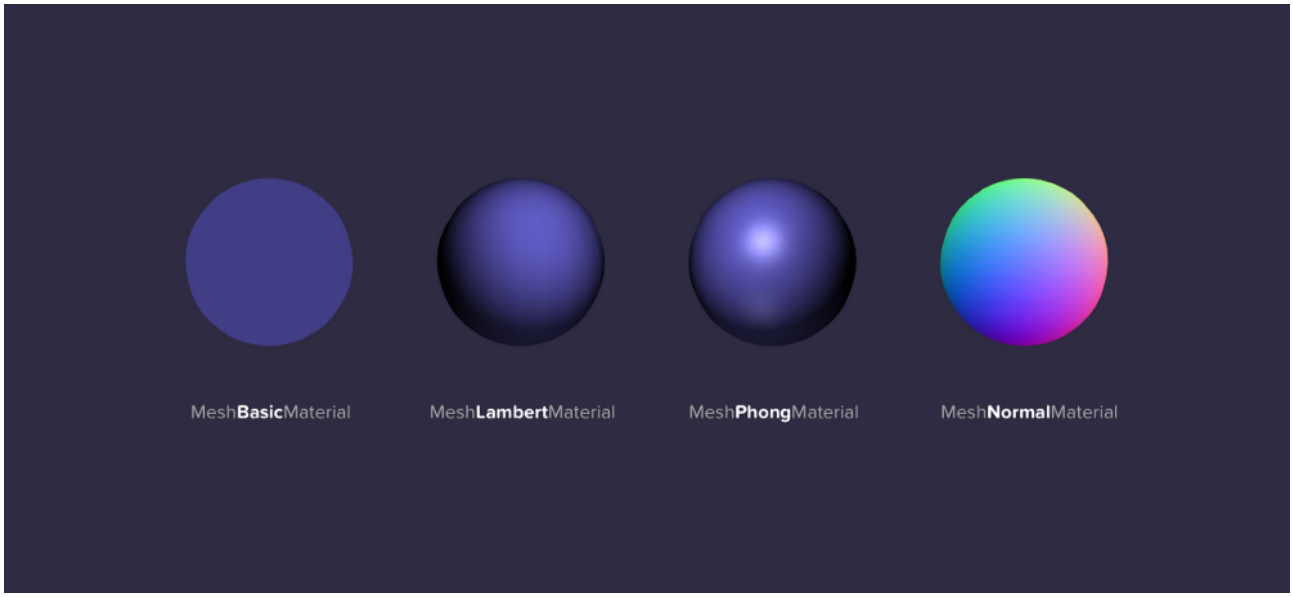
Představení nejznámější a nejpoužívanější knihovny pro práci s WebGL. Tyto knihovny poskytují vývojářům abstraktní základnu pro tvorbu WebGL aplikací od animovaných log až po interaktivní 3D hry. V rámci mé práce jsem se zaměřil na knihovny s open source licencí.

Three.js

Knihovna Three.js byla poprvé zveřejněna Ricardem Cabellem v dubnu roku 2010 a byla původně napsána v jazyce ActionScript. Následně byla přeložena do jazyka JavaScript. Knihovnu vytvořila stejnojmenná společnost Three.js. Hodí se spíše pro menší iteraktviní aplikace. Proto je právě tato knihovna vybrána jako základ pro implementaci prohlížeče 3D modelů v této práci. Three.js je vyvíjena pod MIT licencí a je k nalezení v repozitáři na GitHub.com[3].

Hlavní funkce knihovny:

- přidávání a odebírání objektů do scény za běhu
- přidání mlhy do scény
- antialiasing
- perspektivní a ortografické kamery
- animace
- světla – globální zdroj světla, směrová, bodová, bodové kruhové
- stíny – vržený stín, dopad stínu
- materiály – lambert, Phong, hladké stínování
- sprites a 2D vrstvy
- objekty – drátěný model, částice, linky, kostra a další
- export/import nástroje pro vytváření souborů JSON kompatibilních s knihovnami Blender, CTM, FBX, 3DS MAS a OBJ



Obrázek 3.2: Základní materiály Three.js. Zdroj: https://medium.com/@necsoft/three-js-101-hello-world-part-1-443207b1ebe1.

3 https://github.com/mrdoob/three.js/

Babylon.js

Knihovna Babylon.js, je novější, na scéně se objevila v létě 2013. Byla představen společně s první oficiální podporou WebGL API pro Internet Explorer 11. Navzdory původu od společnostui Redmond's labs, Babylon.js (stejně jako Three.js) udržuje licenci open source a to konkrétně Apache Licence 2.0[4]. Chcete-li vytvářet hry, je potřebný také herní engine, který obsahuje další funkce, jako je například detekce kolize, částice a speciální efekty. Pro tyto účely je vhodná knihovna Babylon.js.

- **Hlavní funkce knihovny:**
  - perspektivní a ortografické kamery
  - engine pro simulaci kolizí
  - antialiasing
  - animace
  - částice
  - sprites a 2D vrstvy
  - optimalizační algorytmy
  - mlha
  - stínové mapy
  - alpha mapy
  - video textury
  - dynamické mřížky
  - výškové mapy
  - scénu babylon lze převést z .OBJ, .FBX a MXB.
  - exporter pro Blender

Hlavním rozdílem mezi těmito dvěma knihovnami je jejich zamýšlené použití. V obou těchto knihovnách lze implementovat stejnou 3D animaci, každá ale byla vytvořena za jiným účelem. Je tedy důležité si dopředu uvědomit co chceme vytvořit a se kterou z těchto knihoven tohoto cíle dosáhneme.

Three.js byl vytvořen s jedním cílem: využít webové vykreslování k vytváření 3D grafického obsahu a animací. Tento flexibilní design činí ze společnosti Three.js skvělý nástroj pro webové animace pro všeobecné účely, jako jsou loga nebo aplikace pro modelování.

Tam, kde se Three.js pokouší přenést do tabulky WebGL širokou škálu animačních vlastností, Babylon.js zaujímá cílenější přístup. Původně navržený jako herní engine Silverlight, Babylon.js si zachovává svou náklonnost k vývoji webových her s funkcemi jako detekce kolizí a antialiasing. Jak již bylo řečeno, Babylon.js je stále plně schopen generovat webovou grafiku a animace.

JSC3D

JSC3D je nástrojová sada založená na jazycích HTML5 a Javascriptu. Poskytuje prohlížeč 3D modelů. Sada vznikla za účelem prezentace 3D modelů produktů a malých scén na webové stránce. ´Projekt byl naposledy aktualizován v roce 2014. Projekt je pod MIT licencí[5].

Funkce a možnosti toolkitu:

- ortografické zobrazení CAD modelů
- pevné směrové osvětlení (světlomet)
- vykreslování pomocí bodů

4 https://en.wikipedia.org/wiki/Apache_License

5 https://opensource.org/licenses/mit-license.php

- vykreslování jako drátěný modelovací
- plošné stínování
- hladké stínování s úhlem natočení
- texturace
- mip-mapping
- mapování prostředí
- picking
- vestavěné načítání pro soubory Wavefront obj
- vestavěné načítání pro soubory STL
- možnost načítat soubory Autodesk 3DS
- možnost načítat soubory OpenCTM
- ovládání pomocí dotyku více prsty na mobilních zařízení

3.3 Technologie spojené s vývojem aplikace ve WebGL

HTML5

Technologie WebGL je úzce svázána se značkovacím jazykem HTML verze 5, který byl standardizován v roce 2012. Od této verze je nově v jazyce mnoho vylepšení a novinek. Mezi nejdůležitější patří:

- **Nové elementy v HTML5:**
  - sémantické prvky: <header>, <footer>, <article>, <section>
  - nové atributy elemetu input: datum, čas, číslo, kalendář a rozsah
  - nové grafické elementy: <svg> a <canvas>
  - nové multimediální elementy: <audio> a <video>

- **Nové API rozhraní v HTML5:**
  - Geolokace
  - Frag & Drop
  - Local storage
  - Cash aplikací
  - Web Worker
  - SSE

Element <canvas>
Pro grafické programování v prohlížeči je nejdůležitější rozdíl mezi HTML 4 a 5 Doplnění prvku <canvas>. Tento element umožňuje dynamické vykreslování 2D a 3D grafiky ve webovém prohlížeči. Pro manipulaci oblasti uvnitř plátna může být použit jazyk JavaScript.

CSS

Kaskádové styly (CSS) poskytují jednoduchý způsob jak stylovat obsah na našem webu. První verze CSS neexistovala, dokud se neobjevil jazyk HTML a nestal se v roce 1996 oficiálním standardem. Stejně jako HTML5 a jeho vztah k dřívějším verzím HTML, CSS3 je přirozené rozšíření předcházejících verzí CSS. CSS3 je výkonnější než dřívější verze a zavádí četné vizuální efekty, jako jsou stíny, stínování textu, zaoblené rohy a přechody [2].

JavaScript

JavaScript (JS) je lehký interpretovaný nebo JIT-kompilovaný programovací jazyk s prvotřídními funkcemi. Zatímco je nejvíce známý jako skriptovací jazyk pro webové stránky, používá se i pro mnoho jiných prostředí, které nepracují ve webovém prostředí, například node.js a Apache CouchDB. JavaScript je prototypový, multiparadigmatický, dynamický jazyk podporující objektově orientované, imperativní a deklarativní (např. Funkční programování) styly [3].

Export a formát modelů

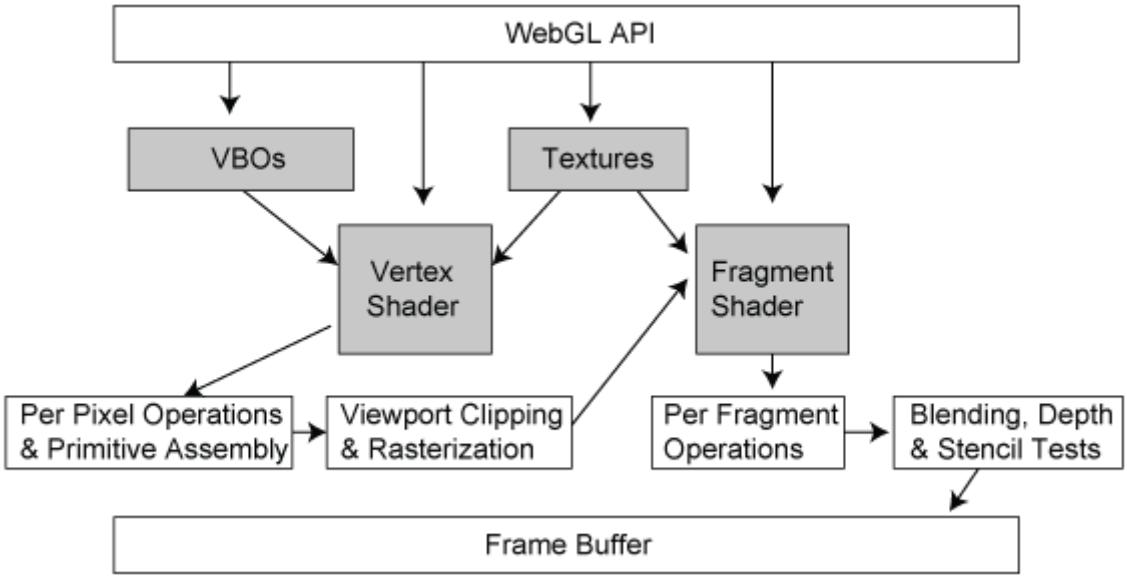
Technologií WebGL je možno zobrazit několik formátů definujících model. Nejznámějším formátem je formát OBJ. OBJ formát však nepodporuje animace modelu. V naší práci jsme proto použili formát Collada, který lze exportovat včetně animace. Pro vytváření, editaci a export modelů existuje mnoho profesionálních programů. Například Blender, 3ds Max, Maya, Lightwave a Modo. V kapitole 2 jsme si také ukázali, že existují on-line aplikace pro tvorbu a úpravu modelů, jako například Clara.io.

Materiály a shadery

Definic speciálních materiálů využívá grafické pipelineWebGL. Pipeline se skládá z několika kroků, které prochází obraz od počáteční definice až po konečné vykreslování obrazovky. Kroky jsou prováděny v předem definovaném pořadí. Komponenty pipline mohou být buď funkční, nebo programovatelné. Mají pevná funkčnost nebo programovatelné shadery [4].

Tradiční pipeline mají pevnou implementaci. Počáteční definice obrazu je sada bodů vrcholu a informací spojených s těmito body, jako je barva, normální vektor a souřadnice textury. S pevnou funkčností se operace provádějí v nastaveném pořadí. Některé prvky, jako je například osvětlení nebo aplikaci textur, můžeme zakázat, ale nelze modifikovat způsob, jakým jsou prováděny základní výpočty osvětlení nebo textur. Grafický pipeline OpenGL před verzí 2.0 používal pouze fixní funkce. Pevně definovaná funkčnost. umožňuje rychlejší a snadnější generování obrázků, protože do systému jsou již zabudovány vzorce osvětlení a stínování. Nicméně je tento způsob omezený, protože nemůžeme tyto nastavení přepsat. OpenGL pevné funkce měly samostatné kroky pipeline pro transformaci vertexu a osvětlení. To se nyní provádí v rámci shaderu vertexu (VS) a fragmentového shaderu (FS), stejně jako všechny texturní aplikace(souhrn barev, mlha a alfa testy). Na obrázku 3.2 je znázorněno jak rozhraní API WebGL interaguje programovatelné a neprogramovatelné součásti pipeline.

Programovatelné pipeline může zobrazovat větší rozsah efektů, protože můžeme definovat části pipeline (ne všechny) a přepisovat výpočty použité pro výpočet barev, polohy texturních souřadnic nebo modelu osvětlení. Programovatelné komponenty pipeline používají program vertexu a fragmentový program, které jsou společně označovány jako shadery. Tyto shadery jsou provozovány na výkonných grafických procesorových jednotkách (GPU), které se nacházejí v moderních počítačích. OpenGL verze 2.0 až 3.0 umožňují použití buď fixní funkce, nebo shaderů. Rozhraní API aplikací OpenGL ES a WebGL, které jsou omezené, podporuje pouze shadery a ne pevnou funkcionalitu.



Obrázek 3.3: Zjednodušené schéma WebGL programovatelných pipeline. Převzato z [4].

GL Shading Language (GLSL)

GLSL je založen na jazyce C ++ a jde vlastně o dva samostatné, ale úzce příbuzné jazyky pro vertexové a fragmentové procesory. Zkompilovaný zdroj na každém procesoru je označován jako VS nebo FS. VS a FS jsou propojeny Společně jeden program běží na GPU. VS praceuje na každém vertexu a je odpovědný za konečné nastavení umístění vertexu. FS pracuje na části rastrového obrazu(působí na každý pixel) a nastavuje konečné barvy. Nelze změnit pozici Fragment nebo pohled na sousední data fragmentu. VS může odesílat data do FS. Konečným cílem Program Shader je aktualizovat vyrovnávací paměť rámců (výkres) nebo vyrovnávací paměť textur [4].

WebGL používá skriptovací jazyk JavaScript, který definované shadery naváže do programových aplikací GLSL. Shadery lze definovat dvěma způsoby:

- Vložením zdrojů VS a FS do stejného zdrojového souboru v html tagu <script> "X-shader / x-vertex" nebo "x-shader / x-fragment"
- Umístění VS a FS do externích souborů a jejich načítání pomocí Ajaxu

Uniforms

Uniforms jsou globální proměnné GLSL nastavené pro celé primitivum, tedy ne pro jednotlivé vrcholy. Nelze použít například mezi glBegin a glEnd. Slouží k nastavení nezměněných hodnot v rámci objektu. Proměnná typu uniform může být pouze čtená a to jak ve shaderu, tak ve fragmentu shaderu.

Formáty pro ukládání 3D modelů

Collada

COLLAborative Design Activity (zkráceně Collada) představuje formát pro ukládání 3D objektů a animací. Je založena na otevřeném XML schématu, tj. můžeme ji snadno přečíst, vytvářet a editovat v libovolném textovém editoru. Standardní koncovkou je .dae. Jde o univerzální formát, ve kterém ke možné definovat jak malé objekty, tak celé scény s dalšími objekty, jako jsou kamery a osvětlení scény.

OBJ a MTL

Formát OBJ je standardní formát 3D obrazu, který definuje geometrii objektu. Byl vytvořen společnost Wavefront Technologies[6]. Tento formát je otevřený a je dnes podporován většinou grafických aplikací a software. Obsahuje geometrii trojrozměrného objektu včetně 3D souřadnic, polohy každého vrcholu, souřadnice textur, souřadnice vrcholů polygonů a další informace. Tneto formát na rozdíl od Collada neukládá animace. Soubory objektů mohou být ve formátu ASCII (.obj) nebo v binárním formátu (.mod).

STL

Souborový formát STL, jako zkratka vycházející z technologie 3D tisku zvané stereolitografie, byl vyvinut firmou 3D Systems. Soubor popisuje třírozměrnou povrchovou geometrii modelu a je nejčastěji používán pro export dat do 3D tiskáren z CAD softwaru nebo softwarových 3D modelářů.

6 https://en.wikipedia.org/wiki/Wavefront_Technologies

Kapitola 4

Návrh prohlížeče 3D modelů do webového prohlížeče

Jak jsem již zmínil v úvodu, moje práce se zabývá vývojem prohlížeče 3D modelů do webového prohlížeče. Před návrhem aplikace jsem provedl výzkum podobných řešení. Po vyhodnocení kladů a záporů konkurenčních řešení (kapitola 2) a vnesení vlastní invence, vznikly následující požadavky pro vznik aplikace.

4.1 Analýza a specifikace požadavků pro vývoj prohlížeče 3D modelů

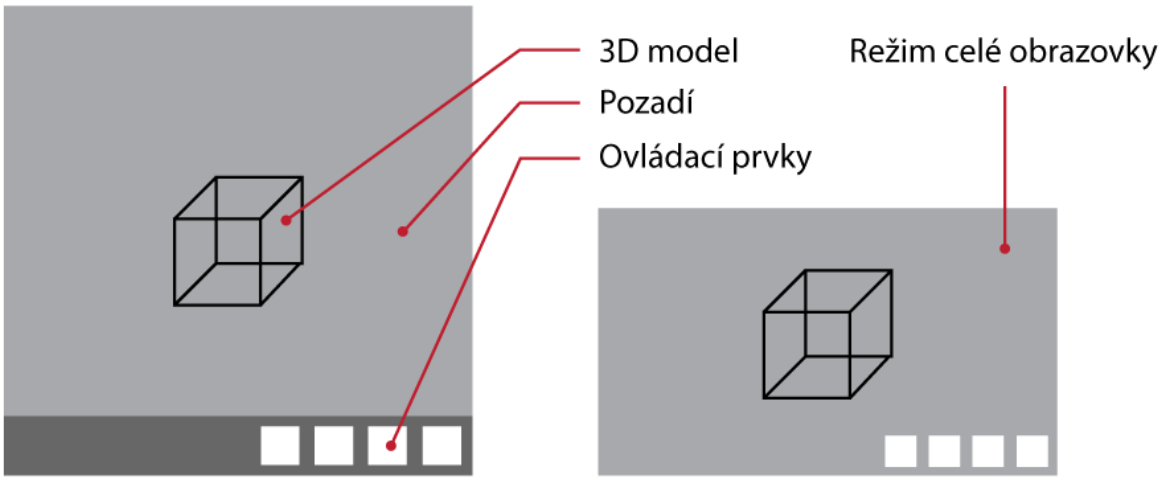
Prohlížeč bude možné použít v jakékoliv webové prezentaci. Nebude tedy závislý na operačním systému nebo platformě, na které je webová prezentace postavená. Aplikace bude optimalizovaná pro všechny prohlížeče s podporou WebGL. Primárně bude aplikace sloužit k zobrazování technicky složitějších produktů, návodům k sestavení výrobků a prezentaci produktů. Typ zobrazovaného objektu není závazný, aplikace umožní zobrazit libovolný model podporovaného formátu. Prohlížeč bude obsahovat jednoduché GUI pro rychlé a intuitivní ovládání.

Hlavní požadované funkce aplikace:

- chtěl bych podporovat co nejvíce dostupných formátů, minimálně však: OBJ & MTL a DAE (Collada)
- s modely bude možno otáčet ve všech osách o 360 stupňů
- přiblížení a oddálení pohledu
- nahlédnuti dovnitř objektu pomocí řezu plochy kamerou
- řez modelem pomocí ořezových ploch a nastavení pozice těchto ploch
- implementace speciálního poloprůhledného materiálu simulujícího rentgenové vidění (X-ray efekt)
- spuštění animace modelu a spuštění animace opačným směrem (rozložit objekt na části a poté zase složit)
- volba z přednastavených materiálů a textur
- animace rotace modelu podél osy
- režim celé obrazovky
- restart zobrazení do původního po načtení aplikace
- zobrazení modelu jako wereframe (drátěný model)

4.2 Návrh uživatelského rozhraní

Hlavním cílem je zobrazení produktu. Většinu prostoru tedy vyplňuje scéna s modelem. Na spodní části aplikace jsou umístěny ovládací prvky. Byly zvoleny ilustrativní ikony znázorňující danou akci. Pro lepší orientaci uživatelů se po najetí na ikonu zobrazuje „tooltip“ popisující dané tlačítko.



Obrázek 4.1: Návrh uživatelského rozhraní prohlížeče 3D modelů

4.3 Analýza a návrh zobrazovaných dat

Při navrhování aplikace jsem zkoumal jaká data a v jakém formátu bude prohlížeč modelů zobrazovat. Jako Vybral jsem formát Collada (COLLAborative Design Activity). Tento formát umožňuje na rozdíl od jiných ukládat model také s animací. Právě animace modelu chci využívat v mém prohlížeči pro rozklad objektu. V modelu by tedy byla uložena animace, kterou aplikace bude umět společně s modelem načíst a na vyžádání animaci spustit. Pro úpravu modelů a export dat jsem zvolil grafický editor Blender a to ze dvou důvodů. Prvním je open-source licence tohoto 3D modelovacího softwaru a druhým jeho časté používání s knihovnou Three.js

4.4 Návrh poloprůhledného zobrazení (rentgen)

Pro aplikaci chci navrhnou materiál nebo texturu, která bude simulovat rentgenové vidění. V praxi funguje rentgen tak, že zdroj vysílá záření směrem na objekt a paprsky procházející objektem při dopadu na povrch fluoreskují. Při průchodu objektem se paprsky v různé míře pohlcují. Čím tlustší vrstvou materiálu tedy paprsky prochází, tím více se pohltí. Tento efekt se budu snažit nasimulovat pomocí GLSL shaderů popsaných v výše. Shadery mi umožní provést transformace materiálu podobné rentgenu.

4.5 Návrh ořezových ploch modelu

Ořezávání v počítačové grafice znamená selektivně povolit nebo zakázat operaci vykreslování pro konkrétní body nebo oblast ve scéně. V této práci chci dosáhnou stavu, kdy budou vykreslovány pouze pixely jedním směrem od průsečíku ořezové plochy s modelem. Pixely nacházející se opačným směrem od průsečíku jsou zahozeny a nebudou vykresleny.

4.6 Další možná rozšíření aplikace

V budoucnu bych chtěl implementovat tyto funkce:

- administrační rozhraní pro nahrávání a správu souborů s modely
- administrační rozhraní nastavení aplikace
- podpora více formátů a jejich automatické rozpoznávání
- možnost kombinace více modelů v jedné scéně
- konfigurace pozadí a okolí objektu
- možnost změny bitmapové textury
- popisky jednotlivých částí modelu
- podpora více animací v modelu – krokování
- nastavení vlastního pozadí
- natavení světelných podmínek prostředí
- interaktivní ovládání pozice ořezových ploch
- možnost drag&drop jednotlivých částí modelu přímo v okně aplikace
- režim virtuální reality
- sdílení na sociální sítě

Kapitola 5

Implementace

V této kapitole se budu věnovat konkrétní realizaci prohlížeče 3D modelů. Jednoduché zobrazení 3D objektu ve webovém prohlížeči není příliš složité. Stačí znát základy HTML a JavaScriptu. K tomu také přispívá kvalitní dokumentace JavasScriptových knihoven pro WebGL. Tvorba komplexního prohlížeče 3D objektů s mnoha užitečnými funkcemi je mnohem náročnější úkol. Za potření je podrobná znalost JavaScriptu a pokročilá znalost CSS a HTML. Důležitá je také znalost formátů, ve kterých lze 3D modely publikovat. Při vývoji a testování jsem narazili na nutnost různých manipulací a úprav objektů, je tedy také zapotřebí umět zacházet s modely a editačním nebo modelovacím software.

5.1 Postup implementace a použité moduly knihovny Three.js

Téměř celá aplikace, tedy jádro aplikace a veškerá funkcionalita, je napsána v jazyce JavaScript. Pouze ovládací prvky (GUI) nad jádrem aplikace jsou definovány jazykem HTML5 a vzhled je popsán pomocí kaskádových stylů CSS3.Prohlížeč aplikaci je možné přepínat mezi různými materiály objektu a zapínat dodatečně funkce jako jsou animace. Pro některé materiály jsou využívány shadery, které jsou psány v programovacím jazyce pro psaní shaderů GLSL. Základem pro celou aplikaci je technologie WebGL (3.1). Pro pracování s WebGL jsem zvolil knihovnu Three,js. Tato knihovna je ze zkoumaných (3.2) nejvhodnější pro tvorbu aplikace, kterou se zabývá tato práce.

Součástí knihovny je přídavný modul detektor podpory WebGL v aktuálním prohlížeči, který se spouští jako první při inicializaci aplikace. Tento detektor vytvoří HTML5 element canvas a pokusí se v něm vykreslit WebGL obsah. Pokud se mu to nepodaří, informuje uživatele o skutečnosti, že daný prohlížeč nepodporuje WebGL.

Pokud je daný prohlížeč podporován, aplikace deklaruje a inicializuje potřebné proměnné pro běh a natavení celého programu. Inicializuje se například velikost plátna, úhel pohledu kamery a barva okolního světla. Poté je volána funkce pro inicializaci programu init(). Zde se nejdříve uloží do proměnne DOM element s konkrétním ID #threejs_canvas a dále s tímto element pracuji. Poté se vytvoří a do uloženého elementu threejs_canvas vloží grafické ovládací prvky aplikace. Po vytvoření UI prvků vytvořím objekt „renderer“, který se stará o vykresleni dat do scény. Aby bylo možné na scénu nahlížet, je vytvořen objekt kamery a je mu nastavena pozice v prostředí. Hlavním objektem, do kterého budou vkládány všechny potřebné součásti a objekty vykreslené scény, je objekt s názvem scene. Po vložení kamery do scény na určité souřadnice, inicializuji a přidám do scény osvětlení. Používám tři druhy světel: AmbientLight(okolní světlo), DirectionalLight(směrové světlo) a PointLight(bodobé světlo). Směrových světel je ve scéně použito několik, aby byly modely nasvícené rovnoměrně a dobře viditelné ze všech úhlů. Pro správu textur je vytvořen „LoadingManager“, který podává informace o průběhu načítání souborů textur. V mém prohlížeči 3D objektů používám nyní modely ve formátu .dae(Collada). Pro jejich načítání využiji „ColladaLoader“ dostupný v balíčcích Three.js. Pomocí loaderu je načten ukázkový model a spolu s jeho načtením jsou z

modelu uloženy do proměnných potřebná data animace pro budoucí použití. Po načtení modelu definuji vlastní materiály, model vystředím ve scéně a nastavím měřítko. Poté vložím daný model do scény. Nyní dříve vytvořenému objektu pro renderování nastavím rozměry plátna, barvu pozadí a poměr stran a vložím do DOM elementu pro vykreslení scény threejs_canvas.

Pohyb ve scéně je zajištěn JavaScriptovým nasloucháním eventů o pohybu myši v plátně a připraveného modulu z knihovny Three.js pro ovládání kamer TrackballControls. Tímto je v aplikace zajištěný pohyb kamery kolem objektu ve všech směrech a možnost přibližovat a oddalovat pohled.

Načítání modelů

Jak jsem již zmínili výše, knihovna Three.js obsahuje připravené balíčky hotových modulů. V aplikací používáme formát Collada. Pro tento formát je připraven ColladaLoader. Načítání modelů je asynchronní. Průběh načítání dat sleduje objekt LoadingManager. Ověřuje také úspěšnost této operace. Výstup manageru je v naší aplikaci směřován pouze do JavaScriptové konzole. Průběh načítání modelu není graficky zpracován a nejsou hlídané chybové stavy.

Animace scény

Pro animaci je použita funkce requestAnimationFrame. Tato novější metoda prohlížečů window.requestAnimationFrame je lepší než starší metody window.setTimeout(volání funkce po fixním zpoždění) a window.setInterval(opakované volání funkce po fixním zpoždění mezi voláními). Tyto funkce mohou být použity pro úpravu snímků za sekundu renderování. Důvodem, proč je nová metoda window.requestAnimationFrame lepší než starší metody je, že je přesnější a nebude vykreslovat scénu pokud jste na jiné kartě(záložce) v prohlížeči. Druhou výhodou je že použití requestAnimationFrame pomáhá prevenci vybíjení baterie na mobilních zařízeních [4]. Podpora této funkce já závislá na prohlížeči.

Pro nastavení snímku za sekundu implementujeme funkci animate(). Tato funkce nastavuje čas příštího snímku a pomocí metody requestAnimationFrame volá rekurzivně sama sebe. V každém běhu této funkce se volá pro vykreslení scény render().

Animace uložená v modelu

Pro animaci uloženou v exportovaném modelu je implementována funkce start(). Funkce start prochází jednotlivé dílčí části modelu a jejich animace. Pro každý snímek nastaví pozice všech mřížek modelu a aktualizuje hodnoty matice scény pro příští vykreslení. V naší aplikace probíhá animace pouze jednou po jejím zavolání. Je možné nastavení nekonečného opakování animace parametrem loop.

Pohyb modelu(rotace)

Pohyb lze zařídit dvěma způsoby. Změnou pozice modelu ve scéně a změnou pozice kamery(pohledu na scénu). V mé aplikaci používám změny pozice modelu. Při zapnutí funkce přičítám k proměnné která určuje natočení modelu podél osy Y krok 0,015. Tento krok se přičítá při každém volání funkce render, tedy při každém snímku. Krok byl zvolen takový, aby animace působila plynule a přirozeně.

```
model.rotation.y += 0.015;
```

Ukázka 1: Nastavení proměnné pro rotaci modelu.

Materiály

V aplikaci používáme několik vlastních materiálů. Některé využívají možnosti nastavení vnitřních materiálů knihovny Three.js. Jiné jsme vytvořili pomocí definic vlastních shaderů. V knihovně Three.js existuje typ materiálu ShaderMateriál, kterému nastavíme zdrojovy vertex shader a fragment shader. Shadery jsme definovali ve zdrojovém html souboru aplikace.

Materiál X-Ray

```
xRayMaterial = new THREE.ShaderMaterial({
      uniforms: {
            p: {
                  type: "f",
                  value: 2
            },
            glowColor: {
                  type: "c",
                  value: new THREE.Color(0x84ccff)
            },
      },
      vertexShader: document.getElementById('vertexShader').textContent,
      fragmentShader:
      document.getElementById('fragmentShader').textContent,
      side: THREE.DoubleSide,
      blending: THREE.AdditiveBlending,
      transparent: true,
      depthWrite: false
});
```

Ukázka 2: Příklad vytvoření vlastního materiálu použitím shaderů.

pro materiál simulující rentgen jsme použili použili Threej.js ShaderMaterial, kterému jsme nastavili uniforms s těmito parametry: „p“ (typ f: -single float) pro intesitu záře a „glowColor“ (typ c - single color) pro barvu záře. Dále jsme použili vertex shader pro definování polohy vrcholů a fragment shader pro nastavení barvy každého pixelu. Shadery násobením intensity překrývajících se ploch a zvýraznění hran simulují efekt X-Ray(rentgen).

```
<script id="vertexShader" type="x-shader/x-vertex">
      uniform float p;
      varying float intensity;
      void main() {
            vec3 vNormal = normalize( normalMatrix * normal );
            intensity = pow(1.0 - abs(dot(vNormal, vec3(0, 0, 1))), p);
            gl_Position = projectionMatrix * modelViewMatrix * vec4(
position, 1.0 ); }
 </script>
<script id="fragmentShader" type="x-shader/x-vertex">
      uniform vec3 glowColor;
      varying float intensity;
      void main() {
            vec3 glow = glowColor * intensity; gl_FragColor = vec4( glow, 1.0
);    }
 </script>
```

Ukázka 3: Příklad definice shaderů v jazycce GLSL.

- `abs(dot(vNormal, vec3(0, 0, 1)))`
  - zvýraznění ploch směrujících ke nebo od kamery

- `1.0 - abs(dot(vNormal, vec3(0, 0, 1)))`
  - otočením získáme zvýraznění hran(kolmé k pohledu kamery)
  - pow() tohoto výrazu odnotou p nám pak dá výsledný efekt X-Ray

Ořezové masky

Na začátku vývoje naší aplikace jsme se pokusili implementovat ořezovou plochu pomocí shaderů. Myšlenou bylo vytvořit materiál, který ve scéně vykresluje pouze body jedním směrem od ořezové plochy. Body nacházející se opačným směrem by shader nevykreslil.

Z důvody velké poptávky vývojářů během vývoje započala také implementace ořezových ploch přímo do knihovny Three.js. Naše řešení jsme tedy nahradili nově přidanou funkcí do jádra knihovny.

Řez objektu objektive kamery je implementován přímo v objektu kamery knihovny Three.js. Do objektu proměnné můžeme nastavit proměnné „near“ a „far“. Tyto hodnoty určují prostor, ve kterém bude objekt vykreslen.

5.2 Grafické rozhraní (Front-end)

Již při volání inicializace jsme do vykreslovacího plátna vložili ovládací prvky. Tyto ovládací prvky po kliknutí vnitřní funkce knihovny Three.js, nebo vlastní funkce této aplikace. UI aplikace je definováno jazykem HTML. Kód je vkládán pomocí JavaScriptu do hlavního DOM elemetu aplikace. Pro zobrazení ikon byl vybrán ikonický font Font Awesome(1).

Funkce grafického rozhraní aplikace

Pro běh aplikace a funkce uživatelského rozhraní jsme implementovali několik funkcí:

- isFullScreen – funkce testující režim zobrazení aplikace
- enterFS – funkce pro přepnutí aplikace do režimu celé obrazovky
- exitFS - funkce pro vypnutí režimu celé obrazovky
- toggleFS – funkce pro přepnutí z jednoho režimu na druhý

5.3 Export modelů pro 3D prohlížeč

Pro implementaci prohlížeče jsem vytvořil vlastní model formátu Collada. Pro testování aplikace jsem pak využil různých modelů z dostupných veřejných galerií na internetu. Jak už bylo dřív zmíněno, pro práci s modely jsem zvolil program Blender, který umožňuje export do formátu Collada bez dodatečných pluginů. Blender nabízí několik módů editace. Model je potřebné vyvíjet při interním módu s názvem „Blender render“. Při použití jiného módu módu není exportovaný soubor kompatibilní s knihovnou Three.js.

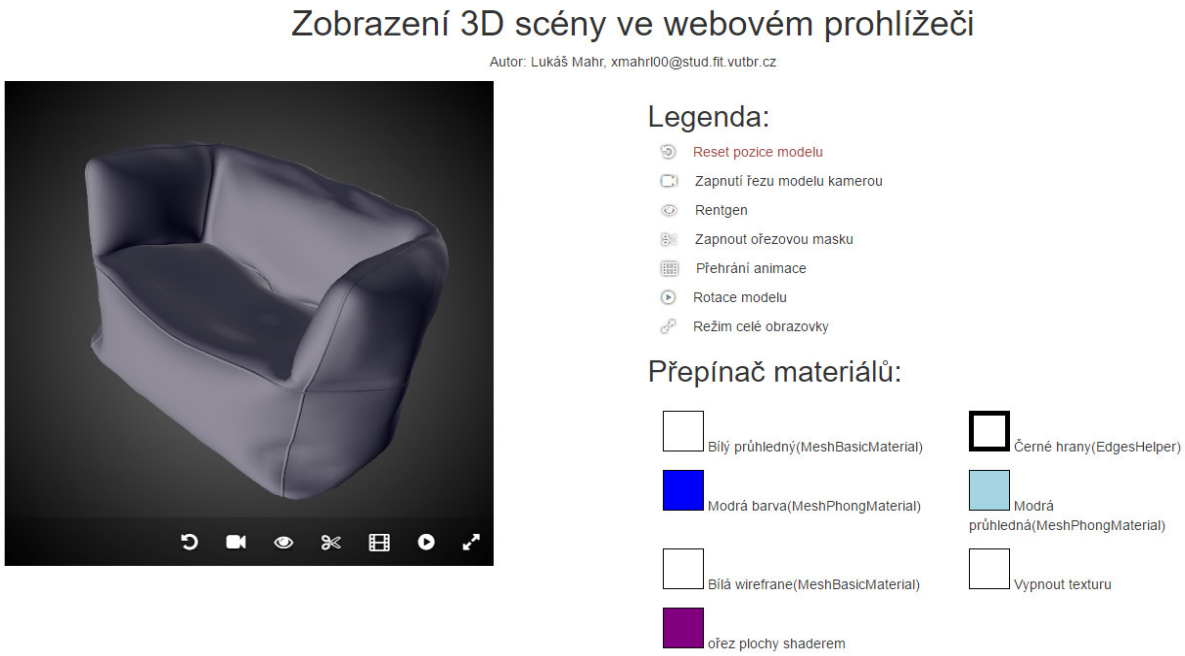
Při exportu z programu Blender je důležité mít správné mapování textur na objekt, jinak by se objekty s animací nevykreslovali správně, respektive objekt by byl bez textur nebo by byly textury chybně vykreslené. Při exportu do formátu Collada je nutné mít u zdrojového souboru modelu přiložené použité textury při vytváření model, textury tedy nejsou součástí modelu.

Pro animaci modelu, kterou používám pro rozpad modelu na části, je zapotřebí danou animaci vytvořit v 3D grafickém programu. V této práci jsem vytvořil model dřevěného stolu. V softwaru Blender jsem pomocí časové osy nastavil každé části modelu pozici v určitý čas animace (key-frames). Pohyb z jednoho bodu do druhého dopočítává program automaticky. Při exportu je potřeba zatrhnou volbu „export key-frames“. Model můžeme vyexportovat jak s animací, tak bez.

Kapitola 6

Výsledky

Výsledkem mé bakalářské práce je interaktivní prohlížeč 3D modelů. Prohlížeč umožňuje zobrazení 3D modelů formátu Collada ve webovém prohlížeči a poskytuje vhodné funkce pro manipulaci s objektem a jeho důkladnému prohlédnutí. Tyto funkce jsou demonstrovány níže.



Obrázek 6.1: Současný stav pohlížeče 3D modelů

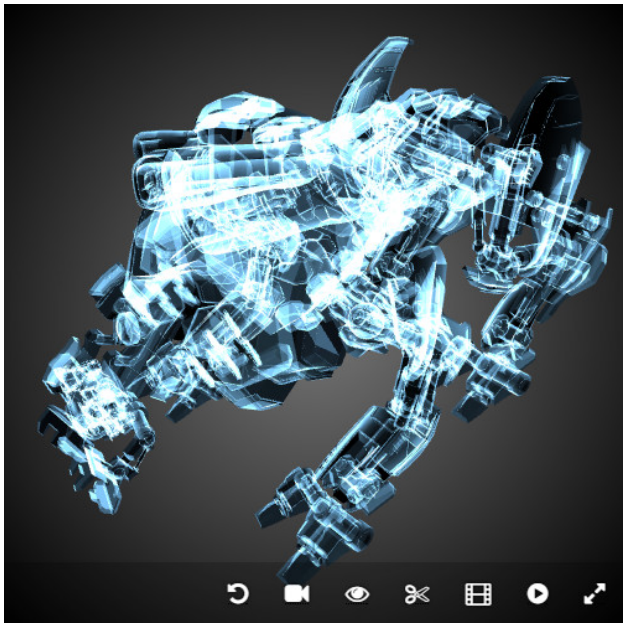
Ovládací prvky

Aplikace obsahuje tyto ovládací prvky:

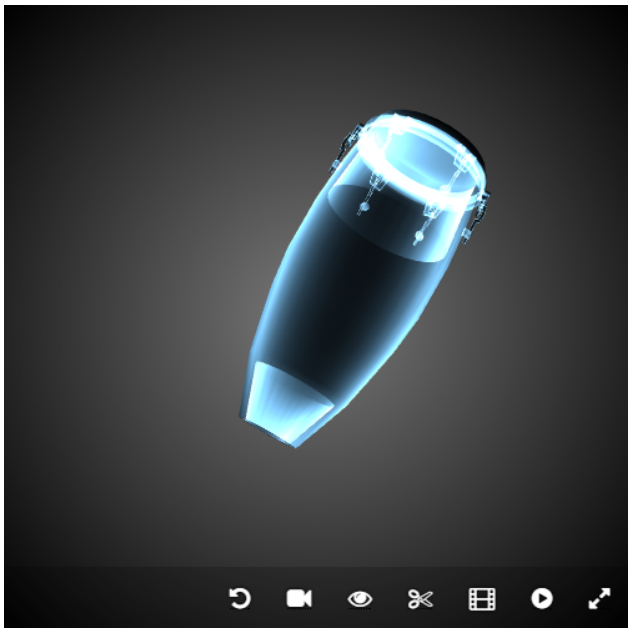
- restart pozice modelu na původní po spuštění aplikace
- zapnutí řezu scény objektivem kamery
- zapnutí rentgen pohledu (X-Ray)
- spuštění animace modelu
- kontinuální rotace modelu kolem osy Y
- zapnutí a vypnutí režimu celé obrazovky

6.1 Ukázka hlavních funkcí aplikace

- **Simulace rentgenu (X-Ray)**

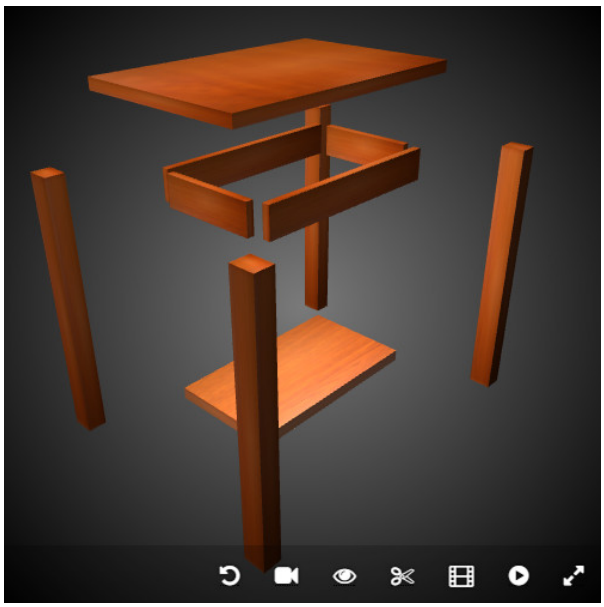


Obrázek 6.2: Ukázka rentgen pohledu - robot.

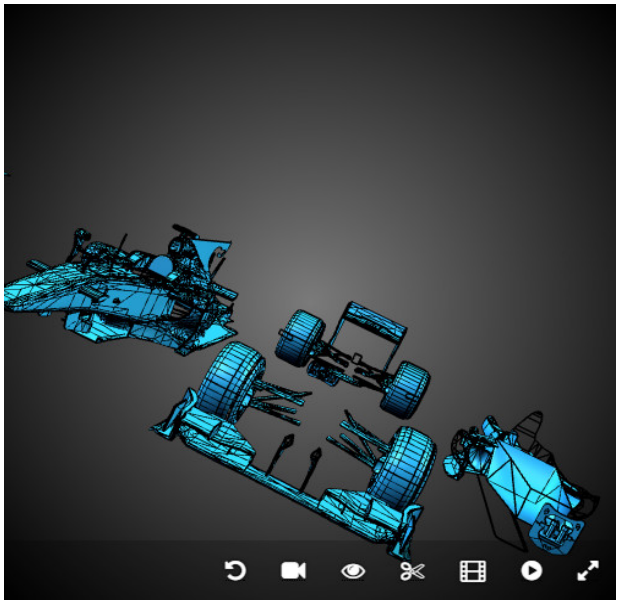


Obrázek 6.3: Ukázka rentgen pohledu – buben.

- **Animace objektu**

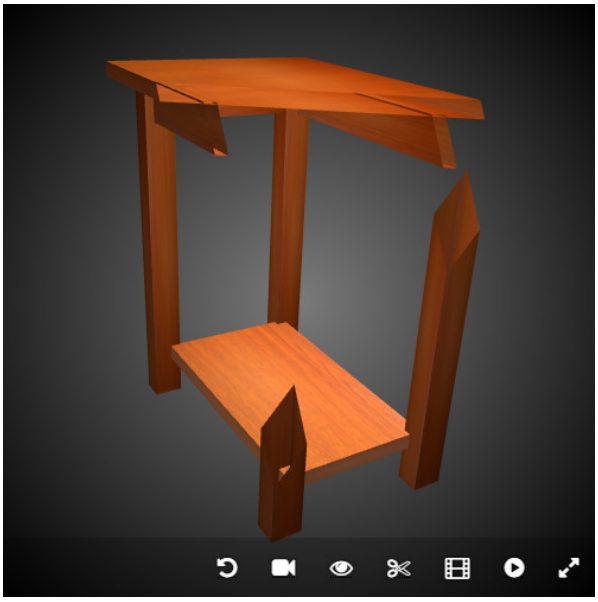


Obrázek 6.4: Ukázka rozložení modelu na části - stůl.

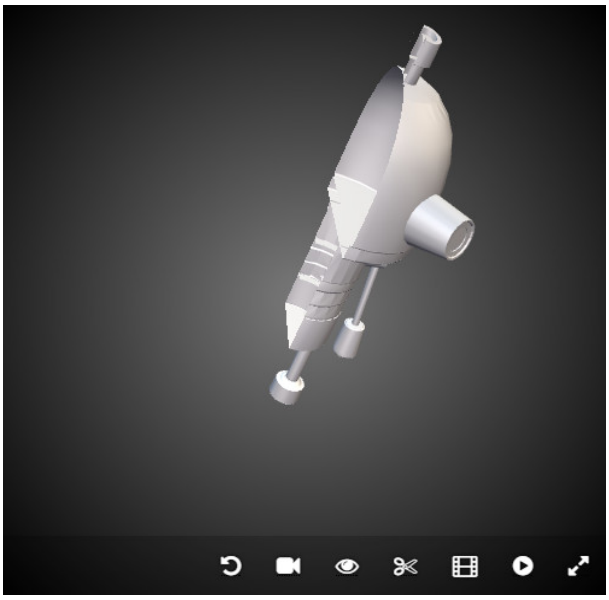


Obrázek 6.5: Ukáka rozložení modelu na části - formule.

- **Řez modelu plochou kamery a ořezovou plochou**



Obrázek 6.6: Ukázka řezu modelu objektivem kamery.



Obrázek 6.7: Ukázka řezu modelu ořezovou plochou.

6.2 Běh aplikace a testování

Aplikace byla vyvíjena a testována v prohlížeči Google Chrome na operačním systému Windows a MAC. V ostatních prohlížečích nyní není garantována správná funkčnost aplikace. Pro běh aplikace je zapotřebí HTTP server. Důvodem jsou požadavky (requesty) na načítaní modelů a textur.

Nedostatky aplikace

Aplikace trpí na drobné nedostatky a chyby, které jsem v rámci této práce z důvodu časové kapacity neopravil. Nedostatkem je například umístění přepínání materiálů mimo hlavní rozhraní aplikace. Nekorektní chování aplikace je například absence možnosti vrátit se na původní pohled při změně materiálu nebo vypnutí ořezové plochy po jejím zapnutí. Celkově by měli ovládací prvky aplikace fungovat jako přepínače (vypnout / zapnout funkci).Oprava chyb, další funkce a vylepšení jsou naplánované pro budoucí vývoj aplikace.

Kapitola 7

Hodnocení uživatelů

Aplikaci jsem předvedl vzorku osmi uživatelů. Z toho dva vedoucí marketingových agentur, zabývající se také 3D grafikou a modelováním. Dále aplikaci vyzkoušeli dva běžní uživatelé internetu a dva spolužáci z Fakulty informační technologií. Nakonec jsem prohlížeč přestavil dvěma profesionálním programátorům. Cílem bylo získat zpětnou vazbu o vzhledu aplikace, ale hlavně ovládaní a funkčnosti aplikace a pochopení jejího účelu a smyslu.

Testování probíhalo ústním nebo elektronický předáním odkazu na 3D prohlížeč. Uživatelé byli krátce seznámeni s účelem této aplikace a jaké je její možné budoucí použití. Uživatelé měli libovolný časový rozsah na vyzkoušení aplikace a také libovolnou formu zpětné vazby.

Uživatelé ocenili tyto vlastnosti:

- vzhled aplikace a práci se světlem
- modely jsou dobře nasvícené a mohou poskytovat věrohodnou představu o produktu
- rotaci modelu podél osy
- animaci s rozkladem, ale nutno přidat restart animace nebo možnost animovat pozpátku

Uživatelé narazili tyto nedostatky:

- ovládací prvky umístěně vně aplikace (přepínaní materiálů) jsou matoucí a není jasný jejich význam a jaké mají funkce. Tyto prvky by měli být také začleněny do UI aplikace
- tlačítko pro restart scény obnovuje pouze pohled. Mělo by obnovit celou scénu do původního zobrazení
- animace by mohla jít spustit zpět opačným směrem
- některé funkce jsou aplikovány najednou nejdou vypnout
- bylo by dobré zařídit testování na více než jedno modelu

Tvůrci 3D modelů, společnosti zabývající se 3d technologiemi a vývojáři poptávají administrační rozhraní pro správu nahrávání modelů a nastavení aplikace. Dále by ocenily dokumentaci k integraci prohlížeče 3D modelů do webové prezentace a možnost vyzkoušet a otestovat plnou verzi aplikace.

Kapitola 8

Závěr

Cílem této bakalářské práce s názvem zobrazení 3D scény ve webovém prohlížeči byl návrh a implementace prohlížeče 3D modelů ve webovém prohlížeči za účelem propagace produktů na internetu. Za tímto cílem vznikla aplikace postavená na WebGL a knihovně pro práci s touho technologií Three.js. Výslednou aplikaci můžeme vidět na obrázku 6.1. Aplikace umí zobrazovat modely ve formátu .dae(Collada), který umožňuje ukládání a přehrávání animací. Objekty lze pozorovat ze všech úhlů pomocí manipulace s kamerou. Na modely jsou aplikované textury dodané k modelu. Na model lze aplikovat vlastní materiály definované v aplikaci. Pro některé speciální materiály byly navrženy a použity shadery. Aplikace umožňuje pohled dovnitř produktů díky ořezovým plochám kamery nebo v prostoru.

Oproti konkurenčním zásuvným modelům a rozšířením podobného zaměření má tato aplikace užitečné funkce navíc. Například poloprůhledné zobrazení se zvýrazněným hran nebo řez objektem. Jen málo podobných aplikací podporuje animace a ještě méně je umožňuje ovládat. Některé funkce, které jsme nalezli u konkurence naopak chybí. Důvodem je časová kapacita na vytvoření projektu a některé funkcionality nejsou vzhledem k zaměření této aplikace vhodné.

Hodnocení uživatelů poskytlo zpětnou vazbu k současnému stavu aplikace a pomohlo k navržení budoucího vývoje aplikace. Tím je odstraněni hlavních nedostatků aplikace a začít vývoj administračního rozhraní. Celkově se prohlížeč 3D modelů dotazovaným uživatelům líbil. 3D Prohlížeč by mohl být v budoucnu reálně používán marketingovými společnostmi a vývojáři webových prezentací. Aby byla aplikace schopná reálného provozu, vyžaduje ještě další vývoj a úpravy. Do budoucna Je v plánu na aplikaci dále pracovat a rozvíjet ji až do finálního produktu.

Literatura

[1]: *CESNET – MetaCentrum: Definice cloudu* [online]. 2016 [cit. 2017-05-17]. Dostupné z: https://wiki.metacentrum.cz/wiki/Definice_cloudu

[2]: CASTRO, Elizabeth. a Bruce. HYSLOP. *HTML5 and CSS3: visual quickstart guide*. 7th ed. Berkeley, CA: Peachpit Press, c2012. ISBN 978-0-321-71961-4.

[3]: *Mozilla Developer Network: JavaScript* [online]. [cit. 2017-05-17]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScrip

[4]: DANCHILLA, Brian. *Beginning WebGL for HTML5*. Berkeley, Calif.: Apress, 2012. ISBN 9781430239970.

Seznam příloh

Příloha A

Obsah CD

Na přiloženém disku se nachází následující soubory a složky:

- /src/ - složka se zdrojovými soubory aplikace
- /doc/ - složka se zdrojovými soubory technické zprávy
- /poster_data/ - složka se zdrojovým souborem plakátu
- /poster.png – demonstrační plakát aplikace
- /xmahrl00.pdf – elektronická verze této technické zprávy

Dodatek B

Plakát