Implementace 3D interaktivní animace do webového portálu pro prezentaci produktů ve vinařství

Bakalářská práce

Vedoucí práce: Ing. Jaromír Landa Ph.D.

Aleš Brunner

Brno 2015

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Implementace 3D interaktivní animace do webo**vého portálu pro prezentaci produktů ve vinařství vypracoval/a samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 5. června 2015

Abstract

BRUNNER, A. Implementation of 3D interactive animation to a web portal for presentation of products in the winery. Brno: Mendel University, 2015.

This Bachelor thesis deals with creation of 3D animated models that are possible to view and interact with in web browser.

This Bachelor thesis deals with creation 3D animated models of machines for the processing of wine, which can be viewed and interact with them in a web browser. Given assignment is solved by using 3D Studio Max and for additional scripts is used X3DOM framework based on WebGL technology. Purpose of this bachelor thesis is to facilitate the presentation of machines to customers through the web.

Keywords

3D modeling, animation, web

Abstrakt

BRUNNER, A. Implementace 3D interaktivní animace do webového portálu pro prezentaci produktů ve vinařství. Brno: Mendelova univerzita v Brně, 2015.

Bakalářská práce se zabývá vytvořením 3D animovaných modelů strojů pro zpracování vína, které je možno prohlížet, i s nimi interagovat ve webovém prohlížeči. Zvolené zadání je vyřešeno pomocí programu 3D Studio Max a pro přidání skriptů je použito X3DOM frameworku využívající WebGL technologii. Účelem této práce je usnadnění prezentace strojů zákazníkům skrze web.

Klíčová slova

3D modelování, animace, web

Obsah

	1.1	Úvo	d	9
	1.2	Cíl p	oráce	9
	1.3	Met	odika	9
2	Те	oret	ický přehled	11
	2.1	3D g	grafika	
	2.2	Scéi	na	
	2.3	Мос	lelovací techniky	
	2.3	.1	Primitivní modelování	
	2.3	.2	Polygonální modelování	
	2.3	.3	Modelování pomocí křivek	
	2.4	Tex	turování	14
	2.5	Svě	tla	
	2.6	Ani	mace	
	2.7	Kan	nery	
	2.8	Ren	derování	
3	An	alýz	a	19
	3.1	Mož	źnosti 3D programů	
	3.1	.1	3ds Max	
	3.1	.2	Cinema 4D	
	3.1	.3	Blender	
	3.1	.4	Rhinoceros	
	3.1	.5	Výběr 3D programu	
	3.2	Ana	lýza zobrazení modelů pomocí webového prohlížeče	
	3.2	.1	WebGL	
	3.2	.2	Formát JSON	
	3.2	.3	Formát X3D	23
	3.2	.4	Výběr formátu a vývojové platformy	25

4 P	raktio	ká část	26
4.1	Мос	lelování produktů	26
4	.1.1	Pracovní prostředí 3ds Max	26
4	.1.2	Tvorba modelu odstopkovače	
4	.1.3	Tvorba modelu hydrolisu	
4	.1.4	Texturování	35
4	.1.5	Osvětlení	
4	.1.6	Animace	
4	.1.7	Export modelů	
4.2	Imp	lementace scény do webového prostředí	
4	.2.1	Popis struktury X3DOM dokumentu	
4	.2.2	Tvorba základních ovládacích prvků scény	
4	.2.3	Přímá interakce s částmi modelů	
4	.2.4	Funkce bublinového popisku	40
4.3	Tvo	rba stránek a implementace scény	
5 Z	ávěr a	u diskuze	42
5.1	Vyu	žitelnost a přínos práce	
5.2	Moż	źnosti vylepšení	
6 L	iterat	ura	43
A S	nímky	ze scény uvnitř webového portálu	46
B C	D		48

Seznam obrázků

0br. 1	Ukázka metody "box modeling" (Unit 66 - 3D Modelling, 2013)	12
0br. 2	Interpolační a aproximační křivka (Keremcaliskan, 2010)	13
0br. 3	2D Bézierova křivka a 3D NURBS plocha (Web3D, 2014)	13
Obr. 4	Ukázka kombinace textur (Treehouse, 2014)	15
0br. 5	Typy světelných zdrojů v 3D grafice (Okino, 2013)	16
0br. 6	Náhled na dialogové okno ThreeJSExporter	23
Obr. 7	Náhled na dialogové okno pluginu InstantExport	24
0br. 8	Uživatelské prostředí programu 3ds Max	26
Obr. 9	Fotka stroje pro odstopkování hroznů, neboli Odstopkovače	28
0br. 10	Vytváření vnitřních stěn pomocí <i>Bridge</i> u horního krytu	29
0br. 11	Horní kryt (se zrcadlenou druhou polovinou)	29
0br. 12	Tvarování vnitřní a vnější stěny (zadní pohled)	30
Obr. 13 spin	Finální vnitřní a přední stěna (upravená až podle velikosti rál a síta)	30
0br. 14	Vytváření spirály a využití <i>Loft</i>	30
0br. 15	Použití <i>Chamfer</i> pro vytvoření kruhu z bodu	31
Obr. 16 přid	Kopírování hotového pásu vedle sebe, za pomocí chytávání	31
Obr. 17 dvo	Tvorba madla a umístění pomocného bodu do středu pro jité zrcadlení	32
0br. 18	Hotové válcové síto s drážkami, čelním krytem a madlem	32
Obr. 19 lopa	Tvar poloviny lopatky a práce s <i>Array</i> pro nakopírování atek podél tyče	32
0br. 20	Vsunutí polygonů pomocí jejich lokálních normál	33

0br. 21	Fotka stroje hydrolisu zleva	33
0br. 22	Fotka stroje hydrolisu zepředu bez koše	33
Obr. 23 matic	Modely z primitivních tvarů: středová tyč, horní příruba, ze víka, koš, pryžový vak	34
0br. 24	Sběrná nádoba a víko vytvořené pomocí rotace křivky	34
0br. 25	Tvary, jejichž zaoblení je tvořené pomocí křivek	35
Obr. 26 pryžo	Ukázka Material Editoru a vytvoření matného materiálu na ový vak	36
0br. 27	Hydrolis náhled 1	46
Obr. 28 pryžo	Hydrolis náhled 2 (průhledný a vysunutý obal, nafouklý ový vak)	46
0br. 29	Odstopkovač náhled 1 (průhledný obal)	47
Obr. 30	Odstopkovač náhled 2 (skrytý obal)	47

Úvod a cíl práce

1.1 Úvod

V dnešní době mají téměř všechny firmy vlastní internetové stránky, které slouží pro zprostředkování informací o službách a produktech, jenž firmy nabízejí. Fyzické produkty se na webových stránkách prezentují převážně pouze skrze obrázek (i když lze využít i jednoduché GIF animace, nebo videa).

Vše je v pořádku pro jednoduché věci, kde není zapotřebí popisovat jejich funkcionality. Tohoto se využívá například v katalogovém zobrazení. Avšak pokud se jedná o komplexnější, technicky zaměřené předměty, vyžadující celkovou nebo detailnější představu, jako například stroje, je obrázková ukázka značně limitující. Situace je většinou řešena více obrázky a delším textovým popisem. V horších případech je zákazník ochuzen o informace, kterých si možná žádá, protože je v oboru třeba nováček a nemá o tom žádnou představu. Ne jinak je tomu i například ve vinařském odvětví, kde se pro zpracování vína používají různé stroje, lišící se funkcemi a tudíž i cenou.

Zákazník by měl mít při výběru dostatek informací, aby se mohl správně rozhodnout pro stroj, který odpovídá jeho preferencím. K tomu by posloužila kvalitní demonstrace stroje na webové stránce, umožňující prohlédnutí stroje a popis jeho funkcí, aby se zákazníkovi dostalo patřičných informací, jako by tomu bylo při osobní návštěvě prodejny. Proto bude tato práce zaměřena na vylepšení prezentace vinařských strojů při online prodeji.

1.2 Cíl práce

Cílem práce je vytvořit 3D modely strojů používaných při zpracování vína i jejich animaci. Dále vytvořit rozhraní umožňující zobrazení těchto modelů ve webovém prohlížeči a přímou interakci s nimi. Navrhnout webový portál ukázkové vinařské firmy a implementovat rozhraní se scénou.

Výsledná práce bude sloužit pro zlepšení prezentace strojů, díky poskytování lepší představy zákazníkům.

1.3 Metodika

K práci bude využit program Autodesk 3Ds Max, v němž se vytvoří věrné modely strojů a jejich animace, k čemuž bude zapotřebí určitá znalost programu a modelovacích technik. Jako předloha pro modelování se použijí ručně pořízené fotografie získané po domluvě a návštěvě firmy zabývající se prodejem vinařských produktů. Případné textury se vytvoří v programu Adobe Photoshop a na modely aplikují v již zmíněném 3Ds Maxu.

Pro následnou práci s modely bude zapotřebí jejich export do vhodného formátu, se kterým si rozumí současné webové technologie. Webový portál příkladné firmy bude vytvořen pomocí HTML jazyka a JavaScriptu. Bude obsahovat základní informace o firmě, ale především 3D scénu pro ukázku produktů, jež bude obsahovat rozhraní zajišťující prohlížení modelů a funkce pro interakci s nimi, to jest různé náhledy kamer a manipulace s jednotlivými částmi, za účelem ukázky funkčnosti zařízení.

2 Teoretický přehled

Tato kapitola slouží pro všeobecné seznámení s 3D grafikou. Budou zde vysvětleny pojmy jako scéna, její nasvícení pomocí světel, zachycení obsahu scény pomocí kamer a render scény. Dále budou rozebrány a vysvětleny různé přístupy k modelování objektů a jejich následné texturování.

2.1 3D grafika

Pojmem trojdimenzionální neboli trojrozměrná grafika se informatice rozumí práce s objekty, které mají šířku, délku, výšku a tudíž pro jejich zobrazení je zapotřebí trojrozměrného souřadnicového systému s osami X, Y, Z. Tyto objekty jsou definovány podobně, jak je tomu u vektorové grafiky a to:

- Plochami, neboli "*polygon"* základní zobrazitelný prvek
- Hranami, neboli "*edge"* přímka spojující vrcholy
- Vrcholy, neboli "*vertex"* vrcholový bod polygonu

V současné době je nedílnou součástí počítačového odvětví, protože umožňuje v informatickém světě napodobovat vnější reálný svět, který je také trojrozměrný. Pomineme-li herní a filmařský průmysl, 3D grafika se využívá v mnoha důležitých odvětví, jako je architektura, konstrukční průmysl, věda atd. (Žára, 2004)

2.2 Scéna

Tímto pojmem se označuje prostor, ve kterém se nachází 3D objekty a ostatní pomocné prvky. Je to jakýsi virtuální 3D svět, který má téměř nekonečný rozměr a může obsahovat téměř nekonečný počet objektů. Pro snazší práci a orientaci ve scéně se používají většinou 4 základní pohledy (shora, zepředu, zleva a prostorový). Podobně jako u 2D grafiky je možno objekty ve scéně pojmenovat, skrýt, zmrazit (uzamknout), roztřídit do skupin nebo vrstev. To vše za účelem lepší manipulace a přehlednosti.

2.3 Modelovací techniky

Modelování značí vytváření 3D objektů a jejich tvarování, aby se dosáhlo věrné či požadované podoby. Pro tuto práci je možno využít více způsobů, které se nazývají modelovací techniky.

2.3.1 Primitivní modelování

Jedná se o nejjednodušší způsob vytváření 3D objektů, někdy také nazýváno jako parametrické modelování, protože se nastavují jen parametry. V této technice se využívá takzvaných geometrických primitiv, jako je kvádr, koule, válec, kužel. A za pomocí booleanovských operací, jako je spojení, průnik, rozdíl jsou vytvořeny komplexní modely. Primitivní modelování se používá hlavně v CAD systémech nebo pak jako základ pro detailnější polygonální modelování. (Žára, 2004)

2.3.2 Polygonální modelování

Jedná se o nejvíce rozšířený způsob modelování, protože je jednoduchý na pochopení. Tento způsob poskytuje největší svobodu v modelování a umožňuje pracovat na nejnižší úrovni detailů při práci se samotnými body.

Jednou z nejvíce oblíbených a populárních metod polygonálního modelování je technika zvaná **box modeling**. Kdy se použije jako základní objekt kvádr, jehož povrch se rozdělí na více dílčích segmentů, které se vysunou do prostoru a dále se s nimi pracuje opakováním stejné techniky, tj. rozdělení podle potřeb na dílčí segmenty, vysouvání ploch a upravování jejich pozic dokud se nedosáhne požadovaného tvaru a požadovaných detailů. Výhodou je, že si sám tvůrce určuje detailnost a jemnost modelu. Používá se například při vytváření postav, ale samozřejmě i jakýchkoliv jiných objektů. (Animation Arena, 2012)

Na box modeling navazuje téma **low poly modeling**, což znamená modelování s co nejnižším nutným počtem polygonů. Low poly modeling se využívá všude tam, kde se modely zobrazují ve skutečném čase (real time rendering), jako jsou hry. Znamená to, že se klade důraz na plynulost vykreslování a úroveň detailů je až druhořadá. Oproti tomu modely s vysokým počtem polygonů, které se využívají ve filmech, jsou náročné na výpočetní techniku a jejich vykreslování trvá značně déle.



Obr. 1 Ukázka metody "box modeling" (Unit 66 - 3D Modelling, 2013)

2.3.3 Modelování pomocí křivek

Jedná se o takzvaný **spline modeling**, jehož základem je křivka tvořená minimálně dvěma řídícími body. Tyto řídící body udávají tvar křivky a to buď metodou využívající interpolaci, nebo aproximaci. Rozdíl mezi nimi je, že interpolační křivka musí procházet zadanými body, naproti tomu aproximační křivka jimi procházet nemusí.



Obr. 2 Interpolační a aproximační křivka (Keremcaliskan, 2010)

Mezi nejběžnější křivky v 3D grafice patří **Bézierova křivka** a z ní vycházející **NURBS** (Non-uniform rational basis spline). Jedná se o aproximační metodu geometricky popisující tvar plochy. Je ideální pro modelování složitých tvarů, které lze snadno ručně měnit pomocí pár bodů.

Neuniformní racionální spline označuje křivku s řídícími body, kde má každý svoji vlastní váhu, která vypovídá o síle vlivu daného řídicího bodu na tvar výsledné křivky vůči síle vlivu ostatních řídicích bodů.



Obr. 3 2D Bézierova křivka a 3D NURBS plocha (Web3D, 2014)

Díky vysoké úrovni matematické přesnosti zaručuje modelování pomocí křivek hladký přechod povrchu plochy. Používá se například ve strojírenství při vytváření karoserií aut, trupu letadel a lodí.

Kromě tvarování plochy, jsou křivky využívány i pro další způsoby modelování. Jedním ze základních je tažením po křivce. Například, když dráhu křivky kopíruje kružnice, tak výsledkem bude roura. Ovšem tvar, který bude kopírovat křivku, může být libovolný, tudíž jakýkoliv jiný 2D obrys. Další možností je rotace křivky a tím vytvoření 3D objektu. Toho se využívá při modelování všech rotačních těles a souměrných předmětů, hlavně ve strojírenství, ale i například při vytváření běžných věcí jako váz, talířů atd. (PUMA, 2015)

2.4 Texturování

Texturování je technika, při níž se vkládá materiál definující různé vizuální vlastnosti na dosud bezbarvý 3D objekt. Jednoduše řečeno se tedy těleso obalí do obrázku. Tato metoda využívá předem připravený rastrový obrázek a využívá se nejčastěji. Texturu lze generovat matematickými funkcemi, pak se jedná o procedurální textury. Jejich výhodou je ostrost renderu z dálky i blízky, protože se textura přizpůsobí (Root.cz, 2008).

Textura se může promítat na obou stranách povrchu tělesa, například pokud se jedná jen o plošný objekt, ale u objemových těles je to zbytečné. V případě velkých ploch s opakujícím se vzorem, není potřeba rozsáhlých obrázků, které by zabíraly zbytečnou velikost na disku a paměti. V takových situacích lze využít malý vzorový obraz, u něhož se nastaví, aby se neustále kopíroval. Jedná se o takzvaný **tiling**, v překladu pak dláždění.

Materiál může obsahovat více typů textur, které se vzájemně doplňují a vytváří tak dokonalejší optickou představu o povrchu materiálu, jako je průhlednost, lesklost, hloubka. Mezi nejvíce používané textury se řadí:

- Diffuse barevná mapa, jedná se o základní typ textury popisující barvu povrchu při perfektně rozptýleném světle, kde není nic lesklé.
- Bump výšková mapa, která vytváří iluzi o výškových změnách povrchu. Využívá poznatku o dopadu a odrazu světla, díky kterému se vytváří stíny a lidský mozek si vytvoří představu o 3D prostoru. Jelikož se aplikuje na rovný povrch, nejlepší výsledky podává iluze, z kolmého pohledu. Mapa je uložena ve stupni šedi, kde černá značí nulovou výšku a bílá maximální výšku.
- Normal novější typ výškové mapy, který může podávat lepší výsledky, avšak zobrazuje občas nežádoucí lesk. Pracuje s normálovým vektorem jednotlivých pixelů, kde v kanály RGB značí XYZ hodnoty vektoru.
- Specular odlesková mapa, určující intenzitu odlesku pixelu po nasvícení. Je velice užitečná, jelikož obraz může obsahovat různé druhy materiálů, jako např. kov, dřevo, bláto a každý materiál různě odráží světlo a lesk.
- Reflection odrazová mapa, která umožňuje zrcadlit na povrchu tělesa okolní prostředí. Používá se při velice lesklých povrchu jako je zrcadlo, sklo, kov či voda.
- Opacity / Transparency průhledová mapa, určuje míru průhlednosti v daných místech.
- Refraction / Distortion pokřivující mapa, jenž se využívá pro materiály, které lámou světlo, jako je sklo, voda.
- Displacement povrchová mapa, která přímo ovlivňuje tvar objektu, jehož povrch se vyvýší podle stupni šedé v mapě. Nevýhodou je, že je zapotřebí velkého počtu polygonů, avšak výhodou je lehké modelování složitých detailů pro high poly modely (z kterých lze následně vytvořit normal mapa pro low poly modely). Další příklady displacement mapy může být snadné vytvoření hor či vodní hladiny.

(Žára, 2004).



Obr. 4 Ukázka kombinace textur (Treehouse, 2014)

Texturování je stěžejní technika při realizaci modelu, který má vypadat věrohodně a reprezentovat objekt ze skutečného světa.

Způsob, jakým bude textura na podkladové těleso umístěna, se nazývá mapování. Určuje, kde a jak má být textura deformována, aby byl obraz rovnoměrně rozprostřen po tělese. Většinou lze použít z nabídky základních geometrických primitiv, jakými jsou rovina, koule nebo válec. Pokud je objekt složitých tvarů, je zapotřebí použít náročnější techniku zvanou UVW Unwrap, kdy se celý povrch rozřízne a rozprostře na 2D plátno, jako při skládání papírových modelů (3ds Max pro všechny, 2014).

2.5 Světla

Bez světla by nebylo vidět vytvořené modely, ani jejich texturovaný povrch. Navíc je velice důležitým prvkem pro dodání reálnosti zobrazovaným objektům ve scéně. Světelný paprsek ve scéně je naprogramován, aby se choval tak, jako se chová ve skutečném světě. Paprsek se šíří přímočaře, odráží od povrchu, lomí při průchodu povrchem. Toto veškeré chování dokáže počítač simulovat, avšak je zapotřebí vysoká výpočetní síla. Algoritmus, který se tímto zabývá, se nazývá **Raytracing**, v překladu sledování paprsku. Jelikož je nereálné sledovat obrovské množství paprsků šířící se všemi směry od zdroje světla, metoda pracuje pouze s paprsky, které skončí u pozorovatele. Pro zjednodušení se postupuje od konce. Svazek paprsků se vyšle od pozorovatele (kamery) na objekty do scény, paprsky se odrazí od povrchu předmětů různými směry, ale ty které skončí ve světelném zdroji, oznamují, že tato část povrchu má být osvětlena. (3d Grafika, 2000)

Kvůli rychlejšímu vykreslování, využitelnému při real-time renderu, byl vytvořen Phongův osvětlovací model, který sice není fyzikálně zcela korektní, avšak výsledky jsou velmi uspokojivé. Jeho princip odrazu světla spočívá v kombinaci tří světelných složek:

- 1. Okolní (ambient) světlo, je všesměrové a konstantní. Napodobuje sekundární odražené a rozptýlené světlo vznikající mnohonásobnými odrazy od ostatních těles a zajišťuje, aby povrch odvracený od zdroje nebyl zcela černý.
- Rozptýlené (diffuse) světlo, určuje intenzitu té části světla, jenž se od matného povrchu objektu odráží rovnoměrně do všech stran a jeho použití vytváří trojrozměrný vzhled ve scéně.
- 3. Lesklé (specular) světlo, určuje intenzitu té části světla, jenž se od povrchu odráží hlavně v jednom směru podle zákona odrazu. Jinými slovy, vytváří na povrchu odlesk.

Světelný zdroj je definován celou řadou nastavitelných parametrů. Mezi základní, které lze světlu nastavit je intenzita, barva (teplota). V počítačové grafice existuje několik typů světel, které se liší určitými vlastnosti (Grafika, 2010).

- Bodové světlo (point light) nejobyčejnější typ světla. Je definováno jako bod, ze kterého je vyzařováno světlo rovnoměrně na všechny strany, podobně jako žárovka.
- Plošné světlo (area light) světelný zdroj tohoto typu emituje paprsky rovnoměrně z celé své plochy (jejíž velikost lze definovat). Je fyzikálně korektní, protože tvrdost a intenzita stínů předmětů klesá s rostoucí vzdáleností od světla. Je však výpočetně nejnáročnější.
- Kuželové světlo (spot light) vyzařuje rozbíhající se svazek paprsků do kuželového prostoru. Chování světla lze ovládat pomocí změny průměry podstavy kužele nebo změnou jeho výšky. Příkladem můžou být reflektory u auta.
- Směrové světlo (directional light) vrhá svazek rovnoběžných paprsků. Předstírá vzdálený zdroj světla, například slunce, které září jedním směrem. Jeho pozice ve scéně nehraje při osvětlení žádnou roli, jelikož se chová jako nekonečně velká a vzdálená plocha rovnoměrně vrhající paprsky.
- Okolní světlo (ambient light) simuluje přirozené denní světlo dopadající na všechny objekty ze všech stran. Ve scéně si ho lze představit jako bodové světlo vyzařující paprsky na všechny strany, navíc v kombinaci nekonečně velkou koulí obalující všechny objekty, z jejíž vnitřní stěny je emitováno světlo.



Obr. 5 Typy světelných zdrojů v 3D grafice (Okino, 2013)

2.6 Animace

Pojem animace představuje iluzi pohybu. Ve skutečnosti je animace tvořena jednotlivými po sobě jdoucími statickými snímky, které se od sebe maličko liší. Využívá se nedokonalosti lidského oka, jež je schopno zaznamenávat pouze určitou rychlostí, která stačila člověku pro přežití v přírodě. Tudíž pokud se obraz mění dostatečnou rychlostí / frekvencí, pro oko vzniká iluze plynulého přechodu. Tato frekvence se značí FPS a udává počet zobrazených snímku za jednu sekundu. Například při zobrazování filmů se používá 24-30 fps, avšak pro uspokojivě plynulý zážitek z hraní počítačových her je zapotřebí 60 a více snímků za sekundu.

Animace v 3D prostředí je vytvářena stejně jako 2D animace a to díky metodě klíčových snímků, či jinak zvané keyframing. Kde se všechny, animátorem nastavené atributy uloží do snímku na časové ose. Takto vytvořené snímky jsou klíčové a definují důležité změny. Pro zbývající prázdné snímky počítač následně automaticky dopočítá plynulý přechod. V 3D grafice lze do animace promítnout změna nejen pozice, rotace, velikost modelů, ale veškerá úprava atributů všech prvků, jako jsou světla a kamery. Animování je složitá činnost, poskytující maximální svobodu animačním prostředím a kreativitou animátora.

Pro usnadnění při vytváření pohybu s jednotlivými částmi objektu, se objektu přidává virtuální kostra, jež ovlivňuje tvar polygonů. Kostra je tvořená jako strom, obsahující uzly rodiče a děti. Pohybem, rotací rodičů jsou ovlivňování děti, poloha dětí však zpětně rodiče neovlivňuje. Ukázka může být lidská ruka, kde pohyb ramenem ovlivní polohu lokte, zápěstí a prstů v prostoru, ale pohyb jednoho prstu nemá vliv na nadřazené klouby a kosti. Odtud se tato metoda nazývá **skeletální** (hierarchická) animace. Kromě lidského těla se aplikuje kostra na všechny živočichy, ramena strojů, rostliny atd (Kříž, 2010; Žára, 2004).

Jelikož vytvoření věrohodných pohybů je velice náročné a delší animace zaberou obrovské množství času, animátor může sáhnout po technologii **motion capture**, která mu značně ulehčí práci. Jedná se o přímé zachycení lidských pohybů (nejen končetin, ale i obličejové mimiky) pomocí kamer, přenesení informací o pohybu do počítače a jejich aplikace na model. Nevýhodou je nutnost dražšího vybavení a využití většinou pouze na humanoidní model (Glosy o animovaném filmu, 2015).

Další způsob, jak lze animování objektů a jejich chování v prostředí usnadnit je využití simulátorů fyzických jevů, jako je gravitace, kolize, chování kapalin atd.

2.7 Kamery

Kamery pomáhají lépe zachytit objekty a dění ve scéně. Lze jich vytvořit velké množství a rozmístit ve scéně tak, aby poskytovali divákovi dobrý přehled. Následným přepínáním mezi nimi se dosáhne atraktivního filmařského střihu. Kamery můžou být staticky umístěny v jedné poloze, nebo se jimi může volně pohybovat po scéně, či se dokonce dají přichytit k nějakému objektu tak, že ho neustále doprovází a snímají. Kamery mají navíc specifické nastavení umožňující různé efekty, pročež se je také vyplatí preferovat před standardním pohledem. Mezi ně patří orientace pohledu, nastavení velikosti záběru změnou zorného úhlu, nastavení ohniska pro definování blízkosti a vzdálenosti snímání ve scéně, ostření na objekty v určité vzdálenosti a rozmazávání okolí, či rozmazávání objektů v závislosti na jejich rychlosti. (3ds Max pro všechny, 2014)

2.8 Renderování

Anglické slovo render se většinou nepřekládá, avšak dalo by se vyjádřit slovem vykreslování. Renderování se používá v 2D a 3D grafice pro výpočet závěrečného obrazu. V 3D grafice se do konečného vzhledu scény reálně promítá hlavně kombinace působení světel na materiály a objekty. Přestože se s objekty manipuluje v trojdimenzionálním prostoru, renderovacím výstupem je 2D záznam. Tím může být buď snímek, nebo video (3ds Max pro všechny, 2014).

Typů vykreslovacích modelů je velká řada. Mezi nejznámější a i nejlepší patří V-Ray, Mental Ray a iRay. Všechny typy obsahují různá nastavení, jimiž lze ovlivnit finální výsledek. Mezi základní patří:

- Určení, počtu renderovaných snímků
- Velikost renderované části
- Výstupní velikost/rozlišení
- Nastavení renderovatelné části
- Zapínání pokročilého osvětlení
- Výstupní cesta renderu a výstupní formát

Renderování je časově velmi náročný proces. Vše záleží na úrovni detailů scény, především počtu polygonů, výpočtu reálného chování světel a vzhledů stínů. Proto je zapotřebí výkonného hardware. Pro náročnější scény, které by se renderovaly týdny, se vytváří renderovací farmy. Jsou to vzájemně propojené počítače s vysokým výkonem, které společně vykreslují celkovou scénu, respektive animaci.

3 Analýza

Dříve než se začne s vytvářením modelů a samotné práce, je vždy potřeba zjistit dostupné možnosti pro realizaci cíle. Jejich následné prozkoumání a zvolení té nejlepší a nejefektivnější varianty. Toho se dosáhne v následujících analýzách. Nejdříve je potřeba vybrat kvalitní modelovací nástroj, ve kterém půjde dobře vytvořit modely strojů. A následně zvolit metodu publikování těchto modelů na a jejich realtimové zobrazení v internetovém prohlížeči.

3.1 Možnosti 3D programů

Počítačová 3D grafika je na světě už celkem delší dobu, a v dnešních dobách je hojně využívána. Proto již bylo vytvořeno hodně grafických nástrojů, které umožňují realizaci 3D modelů a vše s nimi spojené. V této analýze je vybráno pouze pár nejznámějších a hojně používaných programů. V závěru kapitoly pak bude definován nejvhodnější software.

3.1.1 3ds Max

Program Autodesk 3ds Max, je profesionální program a využívá se jako nejrozšířenější robustní nástroj pro práci v 3D odvětví. Má dalekou historii sahající až k prvním počátkům operačních systémů. Nyní je vydávána každý rok nová verze programu, v současné době to je verze 2015. Program je orientován pro velmi širokou škálu 3D tvůrčích činností, protože je velice univerzální a komplexní. Díky kvalitně propracovaným funkcím umožňující modelování, texturování, animování, vizuální efekty, simulování, se používá k tvorbě počítačových her, filmů, efektů, i v architektuře a designerství.

Při vytváření modelů lze použít primitivní, polygonální i NURBS modelování. Nejvíce se však orientuje na polygonální modelování. Podporuje v úměrné míře i NURBS modelování. Součástí programu je komplexní editor materiálů, kde lze vytvářet a pracovat s materiály a texturami. Samozřejmostí je vytváření a práce se světly, kamerami. Pro renderování scén lze vybrat z modulů, jako jsou Scanline a Mental Ray.

Je dobře přizpůsobitelný, díky možnosti vytvoření vlastních programů pomocí skriptovacího jazyka MAXScript, a rozšířitelný, díky již vytvořeným modulům (pluginy), kterých je k dispozici rozsáhlá knihovna od spousty vývojářů. Výhodou je kompatibilita s dalšími produkty od firmy Autodesk, ale samozřejmě zvládá import a export nejrůznějších formátů. Nevýhodou může být, že program je dostpuný pouze v angličtině, zato největší plus je kvalitní dokumentace a nespočetné množství návodů, článků a tutoriálů. Cena za takto špičkový produkt se pohybuje kolem 120 000 Kč, naštěstí však nabízí 30denní zkušební trial verzi, bez žádných omezení (Autodesk, 2013).

3.1.2 Cinema 4D

Název Cinema 4D může být pro 3D aplikaci lehce zavádějící, nicméně se jedná o profesionální multiplatformní program od německé firmy Maxon. Kvalitou a funkcemi se řadí hned vedle 3ds Max, protože se od něj až tak moc neliší. Také totiž propojuje všechny důležité prvky, jako je polygonové modelování, texturování, animace, nasvícení, rendering. Taktéž podporuje primitivní a NURBS modelování, a úpravu modelů pomocí modifikátorů. Pracovní prostředí lze lehce přizpůsobit dle potřeby pojmenováním a přemístěním nabídek.

V programu jsou implementovány moduly, zjednodušující práci grafika. Za zmínku stojí moduly jako *BodyPaint 3D*, pro jednodušší texturování, díky malování přímo na 3D objekt. *Dynamics* modul pro simulaci fyziky a kolizí. Modul *Hair* pro tvorbu vlasů, srsti, trávy. *MOCCA* zase slouží pro animaci pohybu postav, a jejich oděvů. *PyroCluster* umožňuje tvorbu ohně či kouře. V neposlední řadě stojí za zmínku modul *Thinking Particles* pro pokročilou tvorbu částicových systémů.

Jedna z výhod může být, že program je lokalizován do českého jazyka, ale nevýhodou by mohlo být pravděpodobně menší množství podkladů a rad, než má 3ds Max. Za kompletní balíček nejnovější verze R16 je třeba zaplatit 100 000 Kč, ovšem také je možnost zdarma využít plnou zkušební verzi po dobu 42 dní (Maxon, 2015).

3.1.3 Blender

Blender je 3D aplikace, která je proslavená především tím, že je to multiplatformní open source projekt. Znamená to tedy, že jej je možno využívat na různých operačních systémech a k dostání je zdarma, i pro komerční využití. V současné době je vyvíjen skupinou Blender Fundation, která vznikla roku 2002, uveřejněním zdrojového kódu programu. Díky GNU GPL licenci můžou externí programátoři ovládající jazyk Python, modifikovat kód, psát různé nástavby a přispívat k vývoji. Program je poměrně mladý a nejnovější verze je Blender 2.74, přesto však program dosahuje vysoké úrovni, rovnající se drahým komerčním programům. Zvládá totiž kromě polygonálního modelování i texturování, animování, osvětlování, rendering, vše na vysoce uspokojující úrovni.

Prostředí Blenderu je velice intuitivní, nastavitelné a navíc je přeloženo do spousty jazyků včetně češtiny. K většině akcí je přístup urychlen pomocí klávesových zkratek. Obsahuje dokonce video editor a vlastní Game Engine, díky čemuž může produkovat výsledky podobě prezentací, filmů nebo her (Blender3dcz, 2005).

3.1.4 Rhinoceros

Rhinoceros (zkráceně Rhino) od firmy McNeel & Associates, je 3D grafický program. Je proslaven svým zaměřením především na NURBS modelování. Díky používáním NURBS křivek lze snadno s matematickou přesností vytvořit složité tvary, jako jsou hladce zakřivené povrchy např. karoserie automobilů, trupy lodí a letadel. Proto se používá především v architektuře, designerství, strojírenském průmyslu, a tedy i místo skupiny programů CAD/CAM.

Program má jednoduché rozhraní, v němž se každý lehce naučí pracovat. Proto je také často využíván jako učební software ve školách nebo pro seznámení s 3D grafikou. Základem je vytváření primitivních těles a následnými booleanovskými operacemi mezi nimi. Na určité úrovni podporuje také práci s polygony, ale většinou jen při exportu. Rychlou práci zajistí šikovný příkazový řádek, našeptávající příkazy pro provádění operací, namísto ježdění a klikání kursorem. Součástí programu je také skriptovací jazyk. Rhinoceros je především využíván pro modelování, ale co se týče texturování, animace a renderingu, je na tom hůře. Tento nedostatek lze řešit pomocí zásuvných modulů a pluginů, které je však potřeba dokoupit zvlášť. Například pro renderování je pak možno využít moduly jako jsou Maxwell, Brazil, Flamingo a V-ray.

Mezi výhody patří český překlad programu, import a export mnoho formátů a nízké ceně. V současné době je nejnovější verze Rhinoceros 5 stojí zhruba 30 000 Kč, a jak bývá zvykem, je tu samozřejmě i možnost vyzkoušet plnou verzi po dobu 90 dní zdarma (R. McNeel Associates, 2015).

3.1.5 Výběr 3D programu

Ve všech programech, které byly zmíněny, lze vytvořit kvalitní výsledky, protože se jedná o výběr z těch nejlepších a světoznámých nástrojů, jež využívají profesionálové. Taktéž mají většinu rysů, funkcí a metodik, jak dosáhnout 3D grafických cílů, identických, díky čemuž je i lehčí mezi nimi přecházet. Jelikož je potřeba se pro nějaký rozhodnout, je potřeba se podívat na minoritní rozdíly a započíst klady a zápory.

Rhinoceros je vhodný pro začátečníky a pochopení grafiky, česky lokalizován a levný, avšak mezi jeho nevýhody patří horší renderovací kvality, slabší možnosti v texturování a absence tvorby animací. Silná orientace na modelování využívající NURBS křivky, se může taktéž zdát nevýhodou pro vytváření technických prvků s ostrými hranami. Z výše uvedených důvodů mi program nepřišel jako ideální volba.

Cinema 4D, 3ds Max a Blender jsou velice podobné nástroje orientovány na populární polygonové modelování. Na první pohled se dá říct, že se liší jen vzhledem prostředí a intuitivností. Samozřejmě jsou mezi nimi vetší rozdíly, ovšem ty je možno rozlišit až při zkoumání na hlubších úrovních, nebo až po používání jednotlivých aplikací.

Blender je skvělý a oblíbený, především pro "free" licenci, kteráž se vyplatí při stálém využívání. Tento fakt je však zanedbatelný při vývoji krátkodobého projektu, protože lze využít trialových verzí ostatních placených programů. Přestože dosahuje kvalit drahého software, jako velký nedostatek působí nedokonalá dokumentace, a také menší množství návodů a video tutoriálů, které tvoří převážně jen fanoušci a veřejnost. Je to zapříčiněno mladostí softwaru, ale naštěstí je toto mínus postupem času stále odstraňováno. To je jediný důvod, proč jsem se nevydal touto cestou. Je důležité mít dostatek materiálů pro studium, aby práce šla od ruky a nebylo potřeba neustálo pátrat po řešení, či se zdlouhavě dotazovat na fórech.

3ds Max a Cinema 4D se řadí mezi giganty ve stejné cenové kategorii a profesionálnosti. Ovšem i zde platí názor, že 3ds Max stál při vzniku 3D grafiky a modelování a má za sebou dlouhou historii a tudíž i obrovskou základnu a publicitu ze strany veřejnosti. Obsahuje nespočet literatury, video návodů a detailní dokumentaci. Tento argument mi přišel jako stěžejní, vedle faktu, že se jedná o profesionální nástroj, jenž má již v základu implantované všechny důležité funkce jako animování, texturování, renderování a tak dále. Proto právě 3ds Max bude využit jako prostředek pro splnění jedné z části této práce, tj. tvorba modelů vinařských strojů a jejich animace.

3.2 Analýza zobrazení modelů pomocí webového prohlížeče

Internetový prohlížeč je software, který má každý, a je využíván k stále většímu počtu věcí. Zvládne číst různá data a zobrazovat je jako dokumenty, video, zvuk či jiný obsah. Většinu toho dokáže díky zásuvným modulům. Na základě toho vznikla i myšlenka využit internetový prohlížeč pro zobrazení 3D modelů. Tento nápad se podařilo plně realizovat, a to jak externími zásuvnými moduly, tak také přímou implementací pomocí JavaScriptu.

3.2.1 WebGL

Zkratka WebGL v překladu znamená grafická knihovna pro web. Jde o aplikační rozhraní napsané v JavaScriptu, které umožňuje interaktivně zobrazovat 3D obsah, bez nutnosti instalace přídavných modulů do prohlížeče. Podpora této knihovny byla vložena do všech prohlížečů, a nyní se široce využívá, ať se jedná o hry, produktový marketing či zobrazování nejrůznějších dat. Zobrazení je umožněno díky HTML5 elementu zvaném canvas. Jde o nový bohatý způsob zobrazování dat, jako bývaly dříve jen obrázky, audio, či video. Díky své interaktivitě, pomalu vytlačuje Adobe Flash, a získává podporu od všech obřích společností jako Google, Apple, Mozilla, Opera, a v poslední době i Microsoft. WebGL programy jsou tvořeny obslužním JavaScriptovým kódem a kódem shaderu¹. Pro jejich zobrazení stačí mít pouze grafickou kartu, která zvládá shader rendering.

WebGL se začalo vyvíjet v roce 2007. Od roku 2009 je vyvíjeno neziskovou organizací Khronos Group, která vydala specifikace pro verzi WebGL 1.0 na začátku roku 2011. Technologie je založena na OpenGL ES 2.0², jenž je využitelné na mnoha platformách a pracuje přímo s grafickým procesorem. V současnosti se chystá představení WebGL 2.0 založené na OpenGL ES 3.0, které přináší hodně změn a rozšíření možností (Khronos, 2014).

¹ Algoritmus určující způsob, jakým se 3D povrchové vlastnosti objektů budou renderovat

² Část OpenGL pro vykreslování 2D a 3D grafiky

3.2.2 Formát JSON

JSON neboli JavaScript Object Notation (JavaScriptový objektový zápis) označuje, jak je již z významu zkratky patrné, datový formát přizpůsobený pro přenos datových objektů mezi aplikacemi. Jedná se o textový zápis párových dat (tj. popis–hodnota), které jsou lehce čitelné i člověkem. Slouží jako odlehčená alternativa k XML, jehož struktura zápisu je zbytečně obsáhlá. Tento formát objektového zápisu dat je také využíván pro zobrazení 3D grafiky na webu.

Three.js je API³ postavené na WebGL, které umožňuje vytvořit scénu ve webovém prohlížeči bez nutnosti přídavných "pluginů", a které umožní načíst 3D model v JSON formátu. První verze byla publikována v roce 2010. Samotné vývojové prostředí je realizováno jedním JavaScriptovým souborem, který se načte uvnitř souboru webové stránky. V tomto souboru se pak ručním psaním funkcí vytváří a manipuluje se všemi ostatními prvky jako je scéna, světla, kamery, vykreslováním, interakcemi. Modely lze exportovat z 3ds Maxu do formátu JSON pomocí skriptového souboru ThreeJSExporter (TreeJS,2015; Treehouse. 2013).



Obr. 6 Náhled na dialogové okno ThreeJSExporter

3.2.3 Formát X3D

Tento formát vychází ze staršího VRML (Virtual Reality Modeling Language), který byl v roce 1997 prohlášen za standardní formát pro reprezentaci trojdimenzionálních těles ve webovém prostředí. Díky standardizaci je podporován velkou řadou nástrojů, a slouží jako výměnný formát. Formát X3D jakožto následovník VRML má lepší funkcionality jako například lepší komprese souboru a i rychlejší vykreslování. Objekt je vyjádřen pomocí XML zápisu, což přináší řadu výhod, mezi hlavní pak

³ Rozhraní pro snazší programování a vývoj aplikací, je to sbírka tříd a funkcí

snadné začlenění do webu. Součástí objektu je popis jeho geometrie, materiálu, osvětlení a animace (Web3D, 2015)

X3D je spjato s X3DOM, kde X3D charakterizuje model, či scénu, která nelze jen tak přímo zobrazit ve webovém prohlížeči. X3DOM pak slouží tedy jako opensource framework, který umožňuje zobrazení a interakci s objekty v HTML, XHTML, a XML dokumentech. Toto API je složeno z takzvaných uzlů, které jsou do sebe hierarchicky zanořovány. X3DOM je postaveno na WebGL, takže platí vše, co bylo výše popsáno, jako třeba přednost, že tato scéna s elementy je renderována bez nutnosti žádných zásuvných modulů.

Díky tomuto frameworku je možné vytvářet entity přímo v kódu, jako např. kamery, světla, či základní tvary. Ale lepší a přívětivější varianta je exportovat scénu z 3D modelovacího nástroje. K tomuto účelu slouží plugin dostupný zdarma pro 3ds Max s názvem InstantExport. V tomto exportéru je možno zvolit výstupní formát souborů, a to jednak samotný model klasickým VRML (přípona WRL) nebo XML (přípona X3D) zápisem, či model včetně celkové obslužné webové scény, tedy X3DOM ve formátu HTML, či XHTML. Objekty ve všech formátech podporují export i jejich animací, tedy to co je zaznamenáno na časové ose. Ta lze rozdělit na více samostatných animací, vymezením jejich délek a přiřazení jejich názvu, které pak můžou být jednotlivě přehrávány a volány v kódu. K exportovanému souboru se překopírují i používané textury (X3DOM, 2015).

InstantExport v0.5 Beta							
Animation Export Animation TAC Optimize Name Start End	Texture O Use Full Path O Copy Texture Ove Path Texture/	eride 🗸					
Name 0 100 Add Update Delete	Export Bones Coord Int.	Encoding Classic XML HTML XHTML					
Special Geometry BinaryGeometry Common Surface Shader Use Common Surface Shader Export Tanger							
PML Generate PML Generate Main TAC Path The folderpath of the TAC-Fries Definition of the trace MIR. Fin		Browse					
About	Cancel	Generate Export					

Obr. 7 Náhled na dialogové okno pluginu InstantExport

3.2.4 Výběr formátu a vývojové platformy

Three.js je velmi populární, ale na první pohled se jevil jako složitý způsob. Je totiž třeba si většinu naprogramovat ručně vlastní silou. Jedná se sice jen o počáteční vytvoření proměnných z již definovaných vzorů, jako je scéna, kamery, import modelu, vykreslení a podobných obslužných věcí. Není potřeba navrhovat nic nového, jen využívat jejich metody a vše co k tomu patří. Ale zaučení se s těmito věci pracovat zabere určitý čas. Což působilo jako nevýhoda. Navíc exportér nemá moc nastavení a u animace lze nastavit jen koncentrace rychlost přehrávání snímků (fps).

Formát X3D má za sebou dlouhou historii, díky navázání na VRML, který je dokonce standardem pro reprezentaci dat trojdimenzionálních objektů na webu. XML reprezentace dat je velmi rozšířená a rozumí si dobře s webovým prostředím. Export nabízí více možností, jako vymezení času animace, ale hlavní výhoda je výstup přímo do HTML souboru. Tato přímo spustitelná webová stránka, obsahuje scénu, světla, model a textury. Prostředí také obsahuje funkce pro obsluhu elementů (scéna, model, kamera atd.). Díky pohybovým senzorům, které se navážou na objekty lze s nimi zajistit interakci.

Oba formáty a způsoby jsou velmi používány a lze díky nim vytvořit obsáhle scény ve webovém prostředí, které nepotřebují žádné zásuvné pluginy, protože jsou založené na WebGL. Avšak hlubší průzkum obou prostředí by byl časově náročnější, proto jsem rozhodl pro X3DOM, které fungovalo bezproblémově a je vhodné pro začátečníky, díky dobrým návodům.

4 Praktická část

V této části bude vylíčen samotný postup při realizaci práce. Nejdříve však bude popsáno samotné prostředí a funkce vybraného nástroje 3ds Max, ve kterém budou modely vytvářeny. Byl zvolen z důvodů, uvedených v předchozí kapitole, jako hlavní bylo zdůrazněno, že se jedná o špičku mezi modelovacími nástroji, k jehož širokým možnostem exportu přispívá spousta pluginů od externích tvůrců. Pro krátké projekty stačí nabízená 30 denní verze zdarma. Dále budou popsány a zobrazeny kroky sledující tvorbu modelů vinařských strojů, včetně aplikace textur a nasvícení scény.

Následující část bude zaměřena na práci v X3DOM a tvorbu interaktivního prostředí a funkcí zajišť ující manipulaci s těmito exportovanými modely.

V poslední části se bude nacházet popis vytvoření webových stránek příkladné firmy a začlenění prvku se scénou na daný web.

4.1 Modelování produktů

4.1.1 Pracovní prostředí 3ds Max

Pracovní prostředí 3ds Maxu je po mnoho verzí stále stejné, což umožňuje bezproblémový přechod na novější verze. Popis tohoto prostředí bude omezen jen pro základní charakteristiku, protože pro detailní popis tu není prostor a k tomu poslouží manuály. Zaměřeno bude tedy především na prvky, kterých bylo používáno v práci.



Obr. 8 Uživatelské prostředí programu 3ds Max

Na první pohled je vidět, že okno je rozděleno čtyřmi náhledy z různých stran, toto lze změnit, stejně jako ostatní další úpravy uživatelského prostředí, mezi které patří zobrazovaní pomocných panelů a jejich přemisťování. Jako každý klasický program má Panel hlavních nabídek, přes který se dají vyvolat snad všechny funkce programu. Ty nejdůležitější mají své zástupce v podobě tlačítek pod ním. Mezi ně patří hlavně práce s objekty, a to jejich označení, posun, rotace či transformace velikosti. Užitečná a používána byla i další, jako tlačítko pro zapnutí skokové rotace po stupních, tlačítko pro zarovnání pozic dvou objektu k sobě, přístup do okna s materiály, či render scény.

Napravo se nachází nepostradatelný panel, který je tvořen šesti záložkami *Create, Modify, Hierarchy, Motion, Display, Utilites.*

Záložka *Create* pak má dalších sedm podzáložek, pro tvorbu 3D primitivních objektů, 2D tvarů a křivek, světel, kamer, pomocných objektů a dalších prvků.

Záložka *Modify* slouží pro upravování vlastností aktuálně označených objektů a přidávání modifikátorů, jenž je také mění (modifikují). Modifikátory, kterých bylo využito, jsou *Mirror* pro vytvoření zrcadlové poloviny modelu, *Bend* pro ohnutí plochy, *UVW Map* pro umístění textury. V této záložce se také nejčastěji pracuje, protože právě zde jsou funkce umožňující polygonové modelování. Ty jsou definovány tlačítky, některé lze vidět také na předešlém obrázku. Nejpoužívanější jsou *Extrude* pro vytažení polygonu do prostoru, *Inset* vloží nový polygon dovnitř starého kopírující jeho tvar, *Weld* pro sloučení bodů dle nastavené vzdálenosti, *Cut* ruční vytváření hran, propojování bodů, rozdělování polygonů, *Connect* pro propojení dvou bodu a vytvoření nové hrany, *Cap* vyplní díru, chápanou jako chybějící polygon, *Chamfer* zkosí hranu, v praxi to znamená rozdělení hrany na dvě, dá se aplikovat i na rozdělení bodů, *Bridge* spojí dvě hrany polygonem, *Attach* sloučí cizí elementy do jednoho objektu, *Detach* odpojí skupinu polygonů a vytvoří samostatný podobjekt nebo objekt.

V záložce *Hierarchy* mění poloha pivotu (středu) objektu, což je velmi často potřeba pro další práci s objektem.

Záložka *Motion* se používá pro definování pohybu objektu a jeho chování. To značí posun, rotaci a zvětšení.

Spodní část okna je vyhrazena pro časovou osu, potřebnou pro animaci a práci se snímky. Pravá dolní část poskytuje tlačítka pro snazší orientaci ve scéně.

4.1.2 Tvorba modelu odstopkovače

Odstopkovač se ve vinařství používá jako zařízení, které automaticky odděluje stopky od bobulí hroznů. Skládá se ze dvou či tří pater, záleží na typu stroje, jenž se liší různými detaily, ale funkcionalita zůstává stejná. Horní patro tvoří šnekovitá spirála, která posouvá vhozené trsy hroznů a dávkuje je do otvoru, kterým propadají do druhého zpracovávajícího patra. Toto patro obstarává samotný proces odstopkování. Je realizováno sítem, což je vodorovný nerezový válec s otvory v jeho stěnách. A hřídelí s lopatkami vedenou jeho středem, která díky rychlé rotaci naráží lopatkami do trsu a rozbíjí vazbu mezi hroznem a stopkou. Uvolněné hrozny propadávají otvory ve stěnách, zatímco se stopka, díky natočeným lopatkám, posouvá k východu a vypadává koncovým otvorem ven. Těmito údery jsou hrozny značně rozmačkány, a vzniklá kaše se v nejnižším patře tlačí další šnekovitou spirálou do otvoru s čerpadlem, které zpracovanou hmotu žene hadicí ven ze stroje.



Obr. 9 Fotka stroje pro odstopkování hroznů, neboli Odstopkovače

Ochranný kryt

Tento stroj je tvořen jednotlivými částmi, které jsou až na pár výjimek symetrické, takže stačí vymodelovat vždy jen jednu stranu a tu později zrcadlit. Model jsem začal vytvářet od horního patra, a to nerezovým obalem, jehož základ tvořil kvádr rozdělený třemi segmenty po délce a dvěma segmenty po šířce. Po převedení na *Editable Poly* jsem označil a přesunul dělící hrany těsně ke stranám, aby vznikly kraje a uprostřed zbyl velký polygon, který jsem smazal z horní i spodní strany, taktéž jsem smazal všechny polygony tvořící pravou stranu, kvůli možností zrca-

dlení. Dovnitř čnějící horní a dolní polygony jsem spojil přes funkci *Bridge* a vytvořil vnitřní stěnu.



Obr. 10 Vytváření vnitřních stěn pomocí Bridge u horního krytu

Spodní stranu jsem dále vysunul za užití *Extrude* a přemístil, aby levá stěna byla zkosená. Pro celkový náhled použil modifikátor *Mirror*.



Obr. 11 Horní kryt (se zrcadlenou druhou polovinou)

Od tohoto momentu jsem smazal společný spodní polygon a vysouval a tvaroval zvlášť vnitřní či vnější stěnu. Dále jsem takto vysouval dolní hranu a upravoval její pozici, abych dosáhl požadovaného tvaru. Zatímco vnější stěna je až dolů rovná, vnitřní stěna je zkosena směrem ke středu. Vytvořil jsem malé zalomení, jež bude sloužit jako jezdící drážka pro výsuvné síto. Poté jsem stěnu vysouval a tvaroval do oválného tvaru, a vytvářel tak prostor pro síto válcového tvaru, které představuje největší část vnitřku stroje. Níže jsem nechal prostor pro dolní patro, kde se bude nacházet šnekovitá spirála. Přední stěnu jsem tvořil stejnou technikou, ale až poté co jsem měl vytvořené a umístěné válcové síto s drážkami. Jehož tvar jsem napodoboval a obepínal (obrázek 13).



Obr. 12 Tvarování vnitřní a vnější stěny (zadní pohled)



Šnekovitá spirála

Spirálu jsem vytvořil pomocí základního 2D tvaru *Helix,* nastavil vhodný poloměr, délku a hustotu otáček. Poté jsem vytvořil tvar obdélník, na který jsem aplikoval modifikátor Edit Poly a lehce upravil jeho tvar. 3D objekt vzniknul až kombinací těchto dvou 2D objektů, a to za pomocí funkce *Loft,* která se nachází v *Compound Objects.* Tato funkce umožní táhnout 2D tvar po křivce, takže stačilo označit spirálu jako dráhu, která získá obdélník jako svůj tvar. Pro správný tvar výsledné šnekovité spirály bylo nutno odkliknout ve vzniklém *"Loft objektu"* v podmenu *Skin para-meters* položku *Banking.* Kvůli ušetření polygonů jsem smazal vnitřní polygon a nastavil *Shape Steps* na 0 a *Path Steps* na rozumnou hodnotu jako je 5. Tento objekt jsem převedl na *Editable Poly* a pro dosažení hladkosti povrchu označil všechny polygony kromě horních a zařadit do stejné *Smoothing Group*.

Nyní už stačilo, aby vedla středem tyč, tu jsem jednoduše vytvořil z primitivního objektu *Cylinder*, který měl malý poloměr a délku odpovídající celému stroji. Tyto dva objekty jsem spojil pomocí *Attach* do jednoho objektu.

Objekt jsem zkopíroval, jednou do horního patra, pro posouvání trsů a jednou do dolního patra pro posouvání rozmačkaných hroznů do čerpadla. Tyto spirály jsou identické kromě faktu, že každá se otáčí opačným směrem kvůli mechanizmu ozubených kol. Proto stačilo tu dolní otočit o 180°.



Obr. 14 Vytváření spirály a využití Loft

Válcové síto

První pokus, kdy jsem vytvořil pouze válec s texturou průhledných děr, vypadal dost obyčejně a nerealisticky, jelikož díry mají vystouplé okraje. Ani použití bump a normal map nepřineslo lepší výsledek. Rozhodl jsem se tedy, že nejvěrohodnější bude výstupky skutečně vymodelovat. Jedním ze zádrhelů však bylo, že díry nejsou umístěny v mřížkové struktuře (pravoúhle vedle sebe a nad sebou), nýbrž v šachovnicové, tj. šikmo nad sebou po 45 stupních. Po různých pokusech vytvoření kruhových polygonů a jejich vysunutí přímo na válci, kde bylo potřeba označovat spoustu polygonů ručně a pracovat s jejich lokálními normálovými osami, jsem dospěl k následujícímu postupu. Vytvořil jsem 2D plochu Plane, která měla jednu stranu dlouho rovno obvodu válce, který jsem vypočítal a druhou stranu dvacetkrát menší než je délka celého síta. Tuto plochu jsem rozdělil na tři podélné segmenty a úměrným počtem vodorovných segmentů, tak aby vznikly čtvercové polygony. Po převedení na Editable Poly a označení všech vrcholů, jsem pomocí funkce Tessellate a zaškrtnutí možnosti Face dosáhl protnutí všech vrcholů také šikmo. V každém řádku jsem označil na přeskáčku vždy dva body, tvořené pěti nebo osmi přímkami tak, aby vznikl šachovnicový vzor. Poté jsem pomocí Chamfer rozvětvil tyto body a vytvořil osmiúhelníky, které zastupují kruh při nízké úrovni detailu (obrázek 15). Tyto kruhové polygony jsem pomocí Inset zkopíroval dovnitř a vnitřně vzniklou část smazal, aby zde byly díry, či půl díry. Všechny hrany děr jsem označil a posunul nahoru, čímž vznikly vystouplé okraje. Aktivoval jsem přichytávání k vrcholům a zkopíroval tento pás dvacetkrát za sebe (obrázek 16). Dílčí pásy sjednotil pomocí Attach do jednoho objektu a aplikoval modifikátor Bend, díky čemuž jsem ohnul tento pás do neúplného válce, kterému jsem pomocí obyčejných ploch dodělával rovný strop a šikmé plochy pro jako jezdící drážky.



Obr. 15 Použití Chamfer pro vytvoření kruhu z bodu

Obr. 16 Kopírování hotového pásu vedle sebe, za pomocí přichytávání

U válce jsem potom vepředu vytvořil čelo s otvorem, kudy vypadávají stopky. Čelo tvoří plocha, u které jsem vysunul jeden polygon a upravil tvar rozšířením základny. Pro úchopné madlo na tomto čelním krytu, jsem použil základní tvar kvádr, ze kterého jsem po převedení na *Editable Poly* vytvořil čtvrtinu madla, kterou jsem zrcadlil podle osy X a osy Y. Před aplikací modifikátoru *Mirror* se však musel přemístit Pivot neboli střed objektu na jeho kraj. K tomu pomohlo vytvořit pomocný bod z nabídky *Helpers*, přichytit ho na vrchol a poté přichytit Pivot na tento pomocný bod. Tuto techniku jsem používal často pro přemístění Pivotu u dalších objektů, u kterých jsem zrcadlil jejich druhou polovinu.



Obr. 17 Tvorba madla a umístění pomocného bodu do středu pro dvojité zrcadlení

Obr. 18 Hotové válcové síto s drážkami, čelním krytem a madlem

Odstopkovací hřídel s lopatkami

Pro tyčovou hřídel jsem opět použil základní válec s malým průměrem. Lopatku jsem vytvořil, pomocí 2D tvaru *Rectangle*, přičemž jsem opět modeloval jen polovinu. Tento tvar jsem konvertoval do *Editable Spline*, kde jsem označil rohový bod a pomocí *Fillet* vytvořil zaoblení. Poté tvar vytáhl do 3D prostoru pomocí *Extrude* a zrcadlil pomocí *Mirror*. Tuto lopatku jsem umístil na tyč a přes menu *Tools/Array* jsem vytvořil 9 duplikátů, kde jsem mezi nimi nastavil rovnoměrný rozestup a vzrůstající otáčení kolem tyče, kvůli čemuž jsem musel pivot lopatky přesunout do středu tyčové hřídele a pozici těchto dvou pivotů sjednotit za použití *Align*.



Obr. 19 Tvar poloviny lopatky a práce s Array pro nakopírování lopatek podél tyče

Motor

Na tomto motoru jsou charakteristickým prvkem chladící žebra. Pro jejich vymodelování jsem vycházel ze základního tvaru *Cylinder*, jehož hrany jsem rozdvojil kousek od sebe pomocí *Chamfer*, poté označil zbylé větší polygony, a pomocí *Extrude* podle lokálních normál jednotlivých polygonů, je vsunul dovnitř.



Obr. 20 Vsunutí polygonů pomocí jejich lokálních normál

4.1.3 Tvorba modelu hydrolisu

Hydrolis je zařízení, díky němuž se tlakem vylisuje šťáva obsažená v dužnině hroznů, či jiného ovoce. "Hydro" znamená, že pro přenášení tlaku se používá voda, namísto vzduchu či jiného média. Stroj je tvořen železným válcovitým košem, v jehož stěnách jsou malé otvory sloužící jako síto. Ve středu koše je pryžový, který se roztahuje, když se do něj zespod tlakuje voda. Dužnina v koši je tlačena proti kovovým stěnám, šťáva je vytlačována a protéká otvory ven do sběrné nádoby dole.



- Obr. 21 Fotka stroje hydrolisu zleva
- Obr. 22 Fotka stroje hydrolisu zepředu bez koše

Modelování tohoto stroje bylo značně jednodušší, takže i popis tvorby bude krátký. Většinu objektů jsem vytvořil z primitivních těles Válce nebo Kvádru, jejichž polygony stačilo rozdělit na několik dílčích pomocí nástroje *Connect* a poté tvarovat.



Obr. 23 Modely z primitivních tvarů: středová tyč, horní příruba, matice víka, koš, pryžový vak

Sběrná nádoba a víko

Profil těchto dvou objektů jsem obkreslil z fotografií pomocí křivky, díky čemuž bylo snadné docílit plynulého zaoblení. Poté jsem aplikoval modifikátor *Lathe*, za pomocí kterého jsem tyto křivky rotoval o 360° dle zvolené osy a krajního bodu, a následkem čehož vznikly tenkostěnné 3D modely. Pro získání druhé stěny jsem tuto stěnu zkopíroval a zmenšil přes klávesu *Shift* a změnu velikosti *Resize*. Vzniklou mezeru mezi stěnami jsem uzavřel a přemostil pomocí *Bridge*. U sběrné nádoby jsem ještě extrudoval řadu polygonů ze dna, která odděluje odtokový žlábek od prostoru pro koš s lisovanou dužninou. A také připojil odtokovou rouru, avšak pro ni jsem musel nejprve vyříznout díru, tím že jsem tento tvar odebral ze stěny nádoby pomocí funkce *Boolean* nacházející se v nabídce *Compound Objects*.



Obr. 24 Sběrná nádoba a víko vytvořené pomocí rotace křivky

Základna a potrubí

Pro tvorbu trojúhelníkových noh jsem zvolil základní 2D tvar *NGon* z nabídky *Shapes/Splines* a navolil 3 strany, čímž jsem vytvořil trojúhelník. Po převedení na *Editable Spline* jsem pomocí *Fillet* zaoblil rohy. Poté převedl do 3D objektů zaškrt-

nutím *Enable In Render* a *Enable In Viewport.* Stejnou metodou jsem udělal i ostatní zaoblené válcovité tvary, jako jsou úchopy, potrubí a ventily. Napouštějící ventil na konci potrubí jsem oddělil jako element, který jsem zkopíroval a otočil, aby čněl ze strany, jakožto výpustný ventil.



Obr. 25 Tvary, jejichž zaoblení je tvořené pomocí křivek

4.1.4 Texturování

Převážná část dílů tvořící stroje jsou vyrobeny z nerezavějící oceli. Tohoto materiálu jsem snažil docílit ve výchozím "Material editoru" zvolením světle šedé barvy a definováním vhodné úrovně lesklosti (Glossiness) a odlesku (Specular Level), aby se vzhled podobal kovu na fotografiích. Pro větší diverzitu textur jsem vytvořil další dvě až tři alternativy tohoto materiálu s jemně odlišnými barvami a úrovní lesku, které jsem aplikoval na dílčí části, jako jsou ventily u hydrolisu nebo šnekovité spirály, lopatky a čelní kryt u odstopkovače. Tyto prvky jsou pak lépe viditelné a nesplývají s celkem. Pro další části jsem také vytvořil samostatný materiál a použil odpovídající barvu, jako oranžovou pro pryžový vak, červenou pro kohoutky u hydrolisu, či u odstopkovače černou pro kolečka a motor, žlutou a červenou pro tlačítko motoru. Tyto materiály jsou převážně matné, proto jsem razantně snížil lesklost a pro ještě lepší matný efekt nastavil namísto bílé odleskové barvy, zbarvení podobné materiálu (obrázek 26).

Pro koš u modelu hydrolisu jsem ve v programu Photoshop vytvořil fragment textury s ladící šedou barvou a oválnými průhlednými dírami, díky čemuž bylo zapotřebí zvolit PNG formát souboru, který disponuje kromě RGB barevnými kanály navíc tzv. alfa kanálem uchovávající informaci o absenci barvy neboli průhlednosti. Takto hotový fragment jsem poté po povrchu válcovitého koše díky volbě "tiling" opakoval vertikálním i horizontálním směrem tak, aby na sebe jednotlivé kousky plynule navazovaly. Na středovou tyč jsem použil texturu závitů staženou z internetu⁴, ze které jsem ve Photoshopu díky NVIDIA pluginu vytvořil i normálovou mapu, která vyobrazuje hloubku mezi závity.

U odstopkovače jsem vytvořil texturu mřížky k motoru, již bylo potřeba umístit za pomocí modifikátoru *UVW Mapping*, který skýtá lepší nastavení, než se nachází u textury v Material Editoru.



Obr. 26 Ukázka Material Editoru a vytvoření matného materiálu na pryžový vak

4.1.5 Osvětlení

Přestože 3ds Max i X3DOM přidává do scény globální osvětlení automaticky, aby bylo možno standardně prohlížet objekty ze všech stran, toto osvětlení působí obyčejně a přidáním vlastních světel se dosáhne značně lepších výsledků. Z tohoto důvodu jsem přidal vlastní osvětlení i já. Nevýhodou je však, že jakmile se světlo přidá do scény, zruší se efekt globálního osvětlení a vznikne osvětlená a neosvětlená strana. Proto, aby bylo možno opět prohlížet model ze všech stran, jsem do scény přidal a rovnoměrně rozmístil více světel. Pro pěkné osvětlení hydrolisu jsem nad něj kolem dokola rozmístil tři bodová (žárovková) světla po 120°, která ho přirozeně ozařují ze všech stran. A zespod ještě umístil jedno směrové světlo vrhající paprsky na rovné dno nádoby.

Odstopkovač je větší stroj obdélníkového tvaru, proto jsem zde použil sedm bodových světel, abych zajistil plynulé rozptýlení paprsků, osvětlující všechny jeho části. Navíc je rozdíl, jak vypadá světelná scéna uvnitř 3ds Maxu, jenž využívá kva-

⁴Dostupná zdarma na https://www.flickr.com/photos/gritzi/1069643021/

litní pomalý render. A ve webovém prostředí, jež musí vykreslovat scénu v reálném čase, takže kvalita světel je výrazně slabá.

4.1.6 Animace

Jelikož bude později pro modely vytvořená přímá interakce ve webovém rozhraní, kde bude uživatel sám obsluhovat modely a kamerové náhledy. Není třeba vyvářet ukázkové video, tedy filmové animace využívající kamery.

Funkcionalitou hydrolisu je pouhé nafouknutí pryžového vaku, díky vhánění vody potrubím pod velkým tlakem. Kvůli zvětšení vaku nemá cenu vytvářet animační vrstvu, jelikož je zajímavější dosáhnout tohoto uvnitř X3DOM scény za pomocí uživatelovi interakce.

Naproti tomu, odstopkovač je složitější stroj s více funkcionalitami, poháněný elektrickým motorem. Proto jsem provoz tohoto stroje znázornil samočinnou animací, která je uložena v animační vrstvě a bude ji možno také spustit ve webové scéně. Funkcionalita odstopkovače spočívá v rotaci šnekovitých spirál pro dávkování trsů a posouvání kaše, rotaci hřídele s lopatkami pro odstopkování, rotaci lopatek čerpadla a navíc rotaci mechanizmů, které zajištují pohyb těchto dílů, jako je hřídel vedená z motoru a ozubená kola. Součásti, které se točí stejným směrem a nachází se na stejné úrovni, tudíž se otáčí kolem stejné osy, jsem pomocí Select and Link spojil k jednomu mateřskému prvku. Z tohoto prvku se stal rodič, u nějž jsem nastavil otáčení, a k němuž byly napojeny děti, kopírující jeho chování. Rodičovské objekty byly celkem čtyři a to hřídel motoru a tři tyčové hřídele v každém patře stroje, kde ke každé byly napojeny vždy 3 dětské objekty, otáčející se společně s hřídelí na téže úrovni, mezi kterými byly objekty jako šnekovité spirály, soubor lopatek, ozubená kola atd. Těmto rodičovským objektům, točících se velkou rychlostí nekonečně dlouho, jsem přidal samostatný atribut, který zajišťuje nekonečné otáčení, a skrze který lze nastavit i rychlost onoho otáčení. Pro vytvoření takového regulátoru jsem po označení hřídele vybral položku Parameter Editor nacházející se v menu Animation, kde lze vytvořit atribut s vlastním jménem, datovým typem a rozsahem. Aby tento regulátor zajištoval rotaci objektu je potřeba přiřadit mu danou činnost, kterou má zastávat, proto jsem v záložce Motion zvolil Y Rotation a poté tlačítkem Assign Controller, které vyvolalo okno vyžadující další editace (vytvoření proměnné a matematického výrazu), vybral výše vytvořený regulátor. Tento postup jsem opakoval u ostatních hřídelí a poté každé definoval její vlastní rychlost otáčení.

4.1.7 Export modelů

Export modelů se provede zvolením této položky v nabídce vyvolané přes horní logo programu. Zde se jako typ zvolí doinstalovaný plugin InstantExport, jehož nastavení se objeví až po zadání cesty a názvu souboru. V tomto dialogovém okně (obrázek 7) bylo nutné zvolit formát XHTML, protože ve formátu HTML se nezobrazovali externě dodávané textury. Dále odškrtnout *Common Surface Shader*, aby fungovala průhlednost u PNG textury. A u odstopkovače ještě zaškrtnout animaci.

4.2 Implementace scény do webového prostředí

4.2.1 Popis struktury X3DOM dokumentu

Poté co jsou modely vyexportovány je práce v 3d modelovacím nástroji u konce. S modely a jejich díly se bude pracovat už jen v textové podobě, při tvorbě webové scény. Výhodou X3DOM exportéru InstantExport bylo, že automaticky vytvořil HTML stránku s X3DOM scénou, ve které je vložen daný model.

Tento dokument je klasický HTML soubor, tvořený hlavičkou a tělem. Kde jsou v hlavičce načítány deklarační soubory x3dom.css a x3dom.js popisující strukturu scény a obsahující definice obslužných funkcí. V těle je pak tato scéna tvořena prvky, které jsou do sebe hierarchicky zanořovány stromovou strukturou. Tyto X3D elementy jsou tvořeny párovými značkami vymezující začátek a konec, stejně jak tomu je u HTML značek, a také mají vlastní atributy, specifikující jejich vlastnosti a obsahující hodnoty, ke kterým lze přistupovat a měnit. Těmto prvkům se také říká uzly.

Prvním uzlem, jenž označuje začátek a později konec je <X3D>, obsahuje důležité atributy výšku a šířku scény v pixelech. Uvnitř něj se nachází uzly jako <Viewpoint>, jenž definuje pohled na scénu, pomocí atributů pozice a orientace této kamery. Takových kamer lze vytvořit více a později mezi nimi jen přepínat, <Background> definuje barvu pozadí scény. Na stejné úrovni se nachází i světla jako <PointLight> či <DirectionalLight>, která byla importována společně s modelem, ale lze je vytvořit i ručně.

Samotný model je rozdělen do objektů, podle toho jak byl tvořen. Tyto objekty se nachází v uzlech <Transform>, který uchovává jejich pozice v prostoru, a dále obsahuje vnořené poduzly <Shapes>, které zastupují podobjekty, či elementy ze kterých je objekt složen. V každém uzlu <Shape> se nachází uzel <Appearance>, který definuje vzhled povrchu pomocí dílčích uzlů <Material>, <ImageTexture> a <TextureTransform>.

K všem uzlům a prvkům se může přistupovat klasicky přes funkci document.getElementById. A s jejich atributy se pracuje stejně jako s ostatními HTML elementy pomocí JavaScriptových funkcí.

4.2.2 Tvorba základních ovládacích prvků scény

Jako základní ovládání jsem pomocí HTML vytvořil pás tlačítek, který se nachází v horní části scény. Jejich vzhled jsem nadefinoval pomocí CSS stylů. Na pravé straně jsem umístil tlačítka pro přepínání kamerových pohledů. To je zajištěno pomocí jejich atributu *onclick*, jenž odkazuje na funkci, která aktivuje jednotlivé pohledy pomocí změny atributu *set_bind* na *true*, dle následující ukázky funkce sloužící k přístupu k prvku a změny jeho hodnoty v atributu:

document.getElementById('prostorovy').setAttribute('set_bind','true');

Jednotlivé kamerové uzly *<Viewpoint>* lze snadno získat při prohlížení objektu, kdy se klávesou D vyvolá debugovací konsole, kam poté klávesa V vytiskne souřadnice

aktuálního pohledu na scénu. Takto jsem si zkopíroval a uložil dané pohledy do kódu.

Jelikož modely mají šedý povrch, umístil jsem na levou stranu tlačítko "Změnit pozadí", jenž mění barvu pozadí scény z černé na bílou a naopak, aby se mohl uživatel rozhodnout, na kterém podkladu jsou pro něj modely lépe viditelné. Obslužná funkce přistoupí k uzlu Background pomocí vlastně nadefinovaného ID a otestuje, jaká barva se nachází v jeho atributu skyColor a podle toho mu nastaví bílou nebo černou barvu.

Pro další tlačítko "Zprůhlednit" jsem napsal funkci, která zprůhledňuje materiál vnějších dílů stroje tak, aby bylo možno vidět vnitřní díly stroje. Daným zprůhledňovaným objektům bylo potřeba vytvořit duplikátní materiál, jenž se opět pojmenuje pomocí identifikátoru, který zajistí, že se bude pracovat pouze s těmito částmi. Tlačítko je trojfázové, po prvním kliknutí se nastaví průhlednost na 70 % a při druhém stisku tlačítka se tyto objekty úplně skryjí přiřazením hodnoty *false* pro atributu *render*. Třetím stiskem se hodnoty opět vrací na původní plnou barvu a průhlednost se nastaví na 0,001 %, protože nulová hodnota nefungovala, a atribut *render* je třeba změnit zpět na *true*. Toto vše je řešeno v dané funkci pomocí iterujícího se počítadla stisku a podmínky s třemi možnostmi.

4.2.3 Přímá interakce s částmi modelů

Uvnitř webu lze s modelem běžně manipulovat jako v 3D programu, to znamená otáčet, přibližovat, a posouvat, avšak jako přímá interakce je považována možnost pohybovat přímo s jednotlivými částmi modelu, aniž by byla zapotřebí obslužná tlačítka. U hydrolisu je možnost vysunout model víka a koše pomocí tažení myší. Taktéž u odstopkovače lze vysunout válcové síto a navíc spustit animaci stroje po kliknutí na model tlačítka nad motorem.

V X3D scéně lze manipulovat s modelem či jeho částmi pomocí takzvaných sensorů. Jedná se o neviditelné objekty, které sledují svůj prostor, a po kliknutí do něj se aktivuje jejich funkčnost. Tento tvar většinou přijmou pomocí již vytvořeného modelu, tím že se s ním svážou pomocí skupiny. V praxi to znamená, že se kód definující sensor zapíše nad či pod kód 3D objektu, jehož začátek a konec je definován uzlem *«Transform»*, a tyto zápisy se obalí značkou *«group»* na začátku a *«/group»* na konci. V X3D jsou čtyři typy sensorů: Touch, Plane, Cylinder a Sphere. Pro vysunutí dílů u hydrolisu a odstopkovače jsem použil PlaneSensor, který je právě určen pro posouvání. Sensorům lze také vymezit rozsah pohybu pomocí jejich atributů.

Stejně tak i tlačítko motoru, které je složené z více částí jsem obalil do uzlu *<group>*, který tyto tvary spojuje a zajistí, že po kliknutí kamkoli na tyto tvary se volá jeho událost *onclick*. Ta odkazuje na funkci *prehrat*, jež spouští animaci v nekonečné smyčce. Pro jednoduchost lze však spustit tuto funkci i z lišty pomocí tlačítka.

Pro scénu s hydrolisem jsem umístil do lišty tlačítek navíc posuvník, skrze který lze nafukovat pryžový vak určený pro lisování. Realizace je pomocí HTML prvku *input type range*, který při změně hodnoty předává atributem *oninput* danou hodnotu funkci *nafoukni*. V této funkci se přistupuje k 3D objektu a jeho atributu *scale,* u nějž se vkládá hodnota na X a Y souřadnice, tudíž se válec roztahuje do šířky rovnoměrně a výška zůstává stejná.

```
function nafoukni(hodnota) {
    document.getElementById("guma").setAttribute('scale',hodnota+'
        1.708293 '+hodnota).innerHTML = hodnota;
}
<input class="scene-slider" type="range" id="posuvnik"
oninput="nafoukni(this.value)" min="1" max="1.8" step="0.1" va-
lue="1"/>
```

4.2.4 Funkce bublinového popisku

Funkce bublinového popisku způsobuje, že po přejetí myší nad modelem, se zobrazuje vedle kursoru informační rámeček "popup" s názvem jednotlivých částí strojů. Celá funkcionalita je tvořena třemi funkcemi "listenerů", které jsou přiřazeny jednotlivým objektům modelu.

- showPopup stará se o zobrazení popup divu. A podle cíle v události rozlišuje, o jaký element modelu se jedná
- hidePopup skrývá popup div
- movePopup stará se o pozicování popup divu podle souřadnic kurzoru myši

```
var showPopup = function(e) {
   $('div#pop-up').show()
   .css('top', e.layerY + moveDown)
   .css('left', e.layerX + moveLeft)
   .appendTo('body');
   var popup = document.getElementById('pop-up');
   if (e.target == document.getElementById('odvzdusnovaci ventil')) {
     popup.textContent = 'Odvzdušňovací ventil';
    } else if (e.target == document.getElementById('sroub vaku a vi-
   ka')) {
     popup.textContent = 'Šroub vaku a víka';
    } else if (e.target == document.getElementById('montazni mati-
    ce')){
     popup.textContent = 'Montážní matice';
var hidePopup = function(e) {
    $('div#pop-up').hide();
}
var movePopup = function(e) {
    $('div#pop-up').css('top', e.layerY
                                            +
                                                 moveDown).css('left',
    e.layerX + moveLeft);
}
```

O realizaci se stará funkce *addEventListener*, která přidá prvkům na stránce zachytávání události pro přejetí myší (mouseover), opuštění myší (mouseout) a pohyb myši po prvku (mousemove). Na události jsou přichyceny funkce pro zobrazení rámečku (showPopup), skrytí rámečku (hidePopup) a pohyb rámečku zároveň s myší (movePopup). Tyto události mají všechny části, u kterých je v modelu povolena bublinková nápověda. Všechny části modelu musejí mít každý unikátní identifikátor, aby se k danému dílu stroje přiřadil konkrétní popisek s jeho názvem.

```
document.getElementById('odvzdusnovaci ventil').addEventListener(
'mouseover', showPopup);
document.getElementById('odvzdusnovaci ventil').addEventListener(
'mouseout', hidePopup);
document.getElementById('odvzdusnovaci vetiltil').addEventListener(
'mousemove', movePopup);
...
```

4.3 Tvorba stránek a implementace scény

V současnosti je moderní, aby se obrázky na webu zobrazovaly pomocí popup galerie, která umožní jednoduché procházení obsahu či jeho zavření. Toto elegantní řešení jsem použil pro zobrazení scény, takže po kliknutí na náhled produktu se vyvolá samostatný rámeček se scénou, který překryje stránku. Takováto webová galerie je tvořena souborem JavaScriptových funkcí a CSS stylů. Existuje velká řada těchto galerií, které mají své deklarační soubory volně dostupné ke stažení. Stejně tak tomu je i u **prettyPhoto** galerie, kterou jsem zvolil. V šabloně jsem smazal nepotřebné prvky, jako byly tlačítka odkazující na sociální sítě atd. Dále stačilo zapsat odkaz na soubor s modelem a odkaz na obrázek náhledu, definovat velikost a rozlišení rámečku. Nakonec jsem zadal popis v záhlaví, kam jsem vložil název daného stroje, a zápatí, kam jsem vepsal informace potřebné k obsluze scény.

Tento náhledový rámeček a jeho vnitřní scénu jsem nejdříve odlaďoval na jednoduché prázdné stránce, a po dosažení funkčnosti všech prvků jsem se soustředil na tvorbu webu. Pro tvorbu webových stránek lze také využít zdarma dostupných šablon s již přednastavenými prvky a CSS styly. Takový postup je rychlejší a šetří tvůrci čas. Nejrozšířenější nese název **Bootstrap**. Jedná se o kompletní kolekci HTML nástrojů, CSS stylů a JavaScriptové komponenty. Tento vývojářský framework obsahuje sladěné, moderní formátování prvků jako jsou text, tabulka, tlačítka, formuláře atd. Mezi volitelné JavaScriptové rozšiřující složky patří pak rozbalovací nabídky, bublinková okna, výstražné boxy, našeptávač či různé efekty přechodů. Dalším velkým přínosem je, že šablona je optimalizovaná pro různá rozlišení včetně mobilního, a zaručuje kompatibilitu ve všech prohlížečích. Po stažení a zorientování se ve vývojové předloze jsem nejdříve smazal nepotřebné elementy. A poté zasadil vlastní obsah. Obrázky a text jsem si se souhlasem propůjčil ze stránek firmy a po zarovnání jsem vložil náhledy modelů strojů a sloučil kódy pro vyvolání scény v náhledovém rámečku galerie.

5 Závěr a diskuze

Cílem práce bylo vytvořit webový portál obsahující scénu zobrazující modely vinařských strojů, které budou interaktivně zanimovány. Tohoto cíle jsem dosáhl využitím webového rozhraní X3D, do kterého jsem importoval dva modely strojů, jenž jsem zhotovil pomocí programu 3Ds Max. Ve zmíněném rozhraní jsem importovaným modelům následně pomocí JavaScriptu naprogramoval funkce, které ovládají jejich jednotlivé části a zajišťují s nimi přímou interaktivní manipulaci, a taktéž manipulaci se scénou.

Grafická stránka webového portálu působí příjemným dojmem, taktéž i interaktivní scéna, dle mého názoru, obsahuje zdařilé modely vinařských strojů, které jsou tvořeny na pomezí mezi dobrou úrovní detailů a nízkou výpočetní náročností.

Záměr práce a požadovanou funkcionalitu tedy považuji za splněné.

5.1 Využitelnost a přínos práce

Práce byla vytvářena ve spolupráci s firmou BS Vinařské potřeby s.r.o., která nápad uvítala. Přínos bakalářské práce spočívá ve snazší prezentaci prodávaných strojů skrze web. Pomocí 3D scény si lze kompletně prohlédnout zařízení, jeho funkčnost, či názvy jednotlivých částí. Mohlo by se tak jednat o vylepšené katalogové zobrazení.

Firma byla s výsledkem spokojená, a co se týče praktické využitelnosti, rádi práci v budoucnu použijí a umístí na jejich skutečný web. V současné době je k dispozici její náhled na adrese www.brunner.8u.cz.

5.2 Možnosti vylepšení

U modelů zobrazovaných v reálném čase je vždy důležité optimalizovat počet polygonů na minimum. I když jsem se o to snažil, jistě by to šlo ještě lépe, a proto to také považuji za vhodné vylepšení této práce. Scéna by se poté načítala o trochu rychleji a zvýšila by se plynulost prohlížení na slabých zařízeních, jako jsou staré počítače s nevýkonnou grafickou kartou, či v současné době spíše chytré telefony a tablety.

Modely by mohly mít také lepší kvalitu materiálu, který by vypadal více realisticky. To souvisí také s problematikou kvality osvětlení a stínů, jež dodávají nejvíce reálnosti scéně. Stejnou věc poznamenala i firma. Toto je asi nejdůležitější směr, kterým by se vylepšování mělo ubírat, ale z mého pohledu také náročné, protože se musí z daných možností vymáčknout maximum, aby výsledek vypadal znatelně lépe. Bohužel uvnitř webu je kvalita těchto věci limitována vykreslovacím jádrem, které se stěží vyrovná tomu uvnitř 3D modelovacím programu.

Jako další doplnění by mohlo být rozšířit portál vetší řadou modelů reprezentující další nabízené produkty, a přidat propracovanější interakci.

6 Literatura

- Animation Arena. 2012. *Introduction to 3D Modeling* [online]. [cit. 2015-05-14]. Dostupné z: http://www.animationarena.com/introduction-to-3dmodeling.html
- AUTODESK. 2015. Autodesk: *3ds Max* [online]. [cit. 2015-05-13]. Dostupné z: http://usa.autodesk.com/3ds-max/
- Blender3dcz. 2005. *Charakteristika programu Blender* [online]. [cit. 2015-05-12]. Dostupné z: http://www.blender3d.cz/drupal/?q=charakteristika
- Grafika. 2010. Autodesk Maya 2011: Budiž světlo! [online]. [cit. 2015-05-14]. Dostupné z: http://www.grafika.cz/rubriky/3d-grafika/autodesk-maya-2011budiz-svetlo-1--138109cz
- Glosy o animovaném filmu. 2015. *Motion Capture* [online]. [cit. 2015-05-14]. Dostupné z: http://glosy.animace.com/motion-capture/
- Keremcaliskan. 2010. Tutoral on Computer Animation [online]. [cit. 2015-05-14]. Dostupné z: http://www.keremcaliskan.com/a-tutorial-on-computeranimation-ii-2/
- Khronos. 2014. *WebGL Specifications* [online]. [cit. 2015-05-14]. Dostupné z: https://www.khronos.org/registry/webgl/specs/latest/
- Kříž, Jan. 2010. *Mistrovství v 3ds Max.* Brno: Computer Press, 1151 s. ISBN 978-80-251-2464-2.
- Maxon. Cinema 4D Studio. 2015 *Everything you need for high-end 3D*. [online]. [cit. 2015-5-14]. Dostupné z: http://www.maxon.net/products/cinema-4d-studio/whoshould-use-it.html
- Okino. 2013. *Lights* [online]. [cit. 2015-05-14]. Dostupné z: http://www.okino.com/new/toolkit/image22.gif
- PUMA. 2015. *Aproximační NURBS křivky* [online]. [cit. 2015-05-14]. Dostupné z: http://puma.feld.cvut.cz/cs/maple/16/Approx%20NURBS%20krivka2.html
- R. McNeel Associates. 2015. *Rhinoceros Learn.* [online]. [cit. 2015-5-10]. Dostupné z: http://www.rhino3d.com/learn
- Root.cz. 2008. *Pseudonáhodné procedurální textury* [online]. [cit. 2015-05-22]. Dostupné z: http://www.root.cz/clanky/pseudonahodne-proceduralni-textury/
- ThreeJS. 2013. *Docs* [online]. [cit. 2015-05-14]. Dostupné z: http://threejs.org/docs/
- Treehouse. 2014. *Real-Time 3D on the Web* [online]. [cit. 2015-05-14]. Dostupné z: http://blog.teamtreehouse.com/real-time-3d-web
- Treehouse. 2013. *The Beginner's Guide to three.js* [online]. [cit. 2015-05-14]. Dostupné z: http://blog.teamtreehouse.com/the-beginners-guide-to-three-js
- Unit 66 3D Modelling. 2013. *Mesh Construction* [online]. [cit. 2015-05-14]. Dostupné z: http://unit66lf.blogspot.cz/2013/06/unit-66-3d-modelling.html

- X3DOM. 2015. *About* [online]. [cit. 2015-05-14]. Dostupné z: http://www.x3dom.org/?page_id=2
- Web3D. 2014. NURBS Component [online]. [cit. 2015-05-14]. Dostupné z: www.web3d.org/documents/specifications/19775-1/V3.3/Part01/components/nurbs.html
- Web3D. 2015. *Why Use X3D* [online]. [cit. 2015-05-14]. Dostupné z: http://www.web3d.org/x3d/why-use-x3d
- Žára, J. a kol. 2004. *Moderní počítačová grafika. 2. vyd.* Brno: Computer Press,
- ISBN 80-251-0454-0.
- 3D grafika. 2000. *Seriál o 3D světě Světlo a stín, díl druhý* [online]. [cit. 2015-05-22]. Dostupné z: http://www.3dgrafika.cz/print.php?&id=320
- 3ds Max pro všechny. 2014. *Standardní kamery* [online]. [cit. 2015-05-14]. Dostupné z: http://3dsmax.4fan.cz/kamery.html
- 3ds Max pro všechny. 2014. *Texturování* [online]. [cit. 2015-05-14]. Dostupné z: http://3dsmax.4fan.cz/materialy.html
- 3ds Max pro všechny. 2014. *Renderování* [online]. [cit. 2015-05-14]. Dostupné z: http://3dsmax.4fan.cz/rendery.html

Přílohy

A Snímky ze scény uvnitř webového portálu



Obr. 27 Hydrolis náhled 1



Obr. 28 Hydrolis náhled 2 (průhledný a vysunutý obal, nafouklý pryžový vak)



Obr. 29 Odstopkovač náhled 1 (průhledný obal)



Obr. 30 Odstopkovač náhled 2 (skrytý obal)

B CD

K výtisku bakalářské práce je přiložené CD obsahující:

- modely a textury
- HTML soubory portálu včetně scény a vyexportovanými modely
- práci v elektronické podobě