



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV MANAGEMENTU

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUT OF MANAGEMENT

ELEKTRONICKÝ OBCHOD VE 3D ROZMĚRU

E-SHOP IN 3D DIMENSION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN SPOUSTA

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. JIŘÍ DVOŘÁK, DrSc.

BRNO 2011

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Martin Spousta

Řízení a ekonomika podniku (6208T097)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

Elektronický obchod ve 3D rozměru

v anglickém jazyce:

E-shop in 3D Dimension

Pokyny pro vypracování:

Úvod
Vymezení problému a cíle práce
Teoretická východiska práce
Analýza problému a současné situace
Vlastní návrhy řešení, přínos návrhů řešení
Závěr
Seznam použité literatury
Přílohy


Seznam odborné literatury:


- FRIMMEL, M. Elektronický obchod/právní úprava. 1.vyd. Praha: Prospektum, 2002, 321 s., ISBN 80-7175-114-6
POKORNÝ, P. Naučte se 3D grafiku. 1.vyd. Praha: BEN, 2006, 241 s. ISBN 80-7300-203-5
POUR, J. Informační systémy a elektronické podnikání. 1.vyd. Praha: VŠE Praha, 2003, 136 s., ISBN 80-245-0227-5
RYBKA, M., MALÝ, O. Jak komunikovat elektronicky. 1.vyd. Praha: Grada, 2002, 119 s., ISBN 80-247-0208-8
SMEJKAL, V. Právo informačních a telekomunikačních systémů. 1.vyd. Praha: C. H. Beck, 2004, 770 s., ISBN 80-7179-765-0

Vedoucí diplomové práce: prof. Ing. Jiří Dvořák, DrSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2010/11.




PhDr. Martina Rašticová, Ph.D.
Ředitel ústavu


doc. RNDr. Anna Putnová, Ph.D., MBA
Děkanka

V Brně, dne 25.3.2011

Abstrakce

Diplomová práce pojednává o problematice tvorby 3D elektronických obchodů v obecné rovině praktického využití. Obsahuje popis současného stavu elektronických obchodů, teoretické řešení problému přechodu na 3D s popisem výhod, nevýhod a zdůvodnění nového řešení. V neposlední řadě obsahuje i jeho praktické provedení.

Abstraction

This Diploma thesis deals with problems of making 3D e-shops for their widely usage. It contains the description of current level of e-shop, theoretical solution of the problem of transition to 3D with a description of advantages, disadvantages and warrant of new solution. It contains the practical solution as well.

Klíčová slova

3D, vertex, Blender, extrudování, modelace, rendering, animace, WebGL

Key words

3D, vertex, Blender, extrude, modelling, rendering, animation, WebGL

Bibliografická citace práce:

SPOUSTA, M. *Elektronický obchod ve 3D rozměru*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2011. 87 s. Vedoucí diplomové práce prof. Ing. Jiří Dvořák, DrSc.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 25. května 2011

Podpis

Poděkování

Tímto bych rád poděkoval všem, kteří se svými znalostmi a konzultacemi podíleli na vzniku této bakalářské práce, zapůjčili mi literaturu a poskytli cenné informace. Především potom prof. Ing. Jiřímu Dvořákovi. Děkuji za Váš čas i trpělivost.

Obsah

Úvod	10
1 Cíle práce.....	11
1.2 Informační zdroje.....	11
1.2.1 Klasické zdroje - knihy	11
1.2.2 Virtuální zdroje.....	11
1.2.3 Zdroje vysokých škol	11
2 Analýza současného stavu.....	12
2.1 Historie elektronického obchodu	12
2.2 Elektronický obchod v ČR – současnost	13
2.3 Právní překážky elektronického obchodování	14
2.4 Cenová politika prodejců	17
2.5 Rozdělení elektronického obchodu	18
2.6 Elektronický podpis	18
3 Teoretická východiska řešení	20
3.1 Proč vytvářet 3D elektronický obchod.....	20
3.2 Způsoby převodu na 3D.....	20
3.2.1 Fotostudio	20
3.2.2 3D pdf.....	23
3.2.3 3D grafika	25
3.2.4 3D fotografie	26
3.3 3D zobrazovací techniky	30

3.3.1 Aktivní 3D projekce	30
3.3.2 Pasivní 3D projekce.....	31
3.3.3 Autostereoskopická 3D projekce.....	32
3.3.4 Očekávaný vývoj prodejů 3D televizí	34
3.3.5 Srovnání 3D zobrazovacích metod.....	34
3.4 3D kino	35
3.4.1 Dolby 3D	36
3.4.2 Master Image	36
3.4.3 Real-D pro Sony CineAlta 4K.....	37
3.4.4 XpanD.....	38
3.5 Zdravotní aspekty 3D zobrazování.....	39
3.5.1 3D TV - co se děje při sledování?	39
3.5.2 3D TV není pro každého	41
3.5.3 Veletrh Opta 2011	42
3.6 WebGL	42
3.6.1 Historie	43
3.6.2 Použití.....	43
3.7 Kinect	45
4 Návrh řešení.....	49
4.1 Blender	49
4.2 Tvorba 3D modelu v Blenderu.....	50
4.2.1 Blueprint	51
4.2.2 Model místnosti	51

4.2.3 Materiálové nastavení.....	53
4.2.4 Model výstavní plochy	59
4.2.5 Modely sklenic	65
4.2.6 Celkový model 3D obchodu.....	69
4.3 Ovládání 3D elektronického obchodu	71
5 Závěr	73
6 Seznam použité literatury	74
Přílohy.....	77

Úvod

Všechny mocné a silné civilizace ve světové historii byly mistry v obchodování a jejich ekonomika byla na směně zboží značně závislá. Osvojení schopnosti dobře prodávat své výrobky bylo klíčové pro silnou ekonomiku.

V dnešním světě je schopnost dobře prodávat své výrobky ještě důležitější, protože tvrdá konkurence nedovoluje dělat chyby a na trhu je vždy dost jiných schopných firem, které by rády prodaly své vlastní zboží.

Velmi důležitým prodejním místem je dnes internet. Konkurence zde se nejvíce blíží dokonalé konkurenci a pro uživatele je velmi snadné si výrobky různých firem najít a porovnat, zjistit přesnou cenu a porovnat ji s konkurencí.

Nejmodernější zobrazovací technologie dovoluje tomu, kdo je ochotný změnit zavedené trendy, naprosto změnit způsob nabízení zboží. Zákazníkovi je dnes možné ukázat obrázek či model zboží, který pomocí pokroku v technologii 3D doslova vystupuje ze zobrazovací jednotky a to jak z televize, tak z monitoru počítače.

Právě těmito výhodami a možnostmi zobrazení se bude tato diplomová práce zabývat, uvede příklady řešení přechodu na 3D modely a v neposlední řadě bude obsahovat i praktickou ukázkou modelace 3D obchodu s inovovaným řešením pohybu v katalogu (nabídce) obchodu.

Do značné míry bude tato práce poukazovat na nutnost inovace a evoluce prodejního kanálu, který za vývojem 3D zobrazování zbytečně zaostává.

1 Cíle práce

Na základě analýzy informačních zdrojů vytvořte model:

- Elektronických 2D obchodů
- Přechodu na 3D elektronický obchod
- 3D katalogu

1.2 Informační zdroje

1.2.1 Klasické zdroje - knihy

Knihy o 3D grafice nejsou příliš rozšířené už kvůli podstatě této činnosti, která se celá odehrává na počítači, proto jsou pro tuto práci využity především virtuální zdroje.

1.2.2 Virtuální zdroje

Nejvíce využívanými zdroji pro tuto práci byly virtuální zdroje. Zde jsem našel nejvíce informací o dané problematice, vizí respektovaných odborníků a nepřeberné množství návodů, rad, a v neposlední řadě jsou zde stovky webů věnovaných publikacím prací grafiků.

1.2.3 Zdroje vysokých škol

Mezi zdroje z vysokých škol patří především moje vlastní bakalářská práce, která se podobné problematice věnovala již v roce 2009.

Tato diplomová práce obsahuje poznatky z práce bakalářské a rozšiřuje, doplňuje a koriguje skutečnosti na základě velmi rychlého vývoje odvětví 3D zobrazování.

2 Analýza současného stavu

2.1 Historie elektronického obchodu

Začátky internetového obchodování se datují do roku 1992, kdy se v USA začaly jako první komodita prodávat hudební nahrávky na CD, poté následovaly knížky a dárkové předměty. Elektronika se, zvláště kvůli své vyšší ceně, začala prodávat až v pozdějších letech. V České republice se elektronický obchod uskutečňuje asi od roku 1996.

Rychlý rozmach zaznamenaly internetové obchody v letech 1994 a 1995 s nástupem protokolů http a www, kdy se jejich vzhled začal podobat těm, které známe dnes. Největšími hráči tohoto nového trhu se rychle staly velké společnosti, které nabízely stejné zboží, jako ve svých kamenných obchodech, ale s nižší cenou. Vývoj trhu ale nebyl všude ve světě stejný – hlavní rozdíly se začaly projevovat mezi Evropou a USA hned ze začátku ve způsobech platby, Evropané nedůvěřovali on-line platbám, zato Američané tento snadný způsob naopak uvítali. Dalším důvodem bylo rozšíření platebních karet, které bylo v USA masovější, a Američané se neobávali zboží tímto způsobem platit, navíc si doručený balík nechali přistavit na svá zápraží, kde si ho po příchodu domů jednoduše vyzvedli. Jak uvádí Jiří Matoušek z datart.cz *„Z našeho pohledu to působí neuvěřitelně. Opravdu tam už tenkrát byla obrovská důvěra v elektronické platby a také v to, že doručený výrobek u dveří zůstane až do našeho příchodu domů.“* Jednou ze zvláštností českého trhu je dobírka, kterou ještě v roce 2006 preferovalo 60 % klientů. Dobírku v této podobě naopak v USA vůbec neznají. *„Jedním z důvodů tohoto stavu je především nedůvěra a historicky špatná zkušenost českých zákazníků. Navíc pro platby na internetu jsou mnozí nuceni požádat u své banky o aktivaci této služby.“* vysvětluje Jiří Navrátil z on-line obchodu specialisty na elektroniku, společnosti Datart. (15)

2.2 Elektronický obchod v ČR – současnost

Teprve na začátku třetího tisíciletí začínají čeští zákazníci vnímat nákup přes internet jako relativně bezpečný. Důvodem je především profesionálnější přístup některých on-line prodejců. Obecně se začíná zkracovat doba dodání zboží zákazníkům a silnější elektronické obchody začínají fungovat na smluvní bázi nad velkoobchody. Fakticky to znamená, že mají relativně přesné informace o stavu prodávaného zboží ve velkoskladu, od kterého toto zboží odebírají – toto zboží začínají navíc uvádět jako skladové. Rozšiřuje se také využívání on-line plateb debetními i kreditními kartami.

Zákazníci však nyní čekají více než jen dobře odladěnou prodejní www aplikaci s pěknými obrázky a objednáním dopravy. Samozřejmostí by měla být perfektní logistika a profesionální poprodejní služby. Zákazníky dále zajímá možnost snadné reklamace, možnost vrácení zboží, servisu a dalších služeb, které poskytují kamenné prodejny. Chtějí mít jistotu, že zakoupením zboží se o ně prodejce nepřestane starat.

Spojení elektronického obchodu s kamenným řetězcem existuje již poměrně dlouhou dobu v USA i v Evropě. Příkladem jsou Target.com, Walmart.com, Bestbuy.com, Comet.co.uk. Zatímco dříve bylo některé zboží prodáváno jen na webu a jiné zase pouze v síti kamenných prodejen, dnes se mohou zákazníci těšit ze spojení výhod u obou způsobů prodeje. Část sortimentu některé řetězce prodávají ovšem pouze přes web a naopak.

„Dostupnost a pohodlí nákupu přes internet je propojeno se zázemím obchodního řetězce,“ shrnuje Jiří Navrátil, ředitel internetového obchodu společnosti Datart, která spustila svůj e-shop jako první obchodní řetězec v České republice v prosinci roku 2005.(15)

Dle nejnovějšího průzkumu Českého statistického úřadu z roku 2010 vyplývá, že alespoň jednou nakoupilo přes internet 25 % všech jednotlivců ve věku 16 a více let (2,2 mil. jednotlivců). Významným zjištěním také je, že za posledních 12 měsíců využilo elektronický nákup 1,9 mil. jednotlivců ve věku nad 16 let, z čehož můžeme vyčíst spokojenost zákazníků s tímto druhem obchodování.

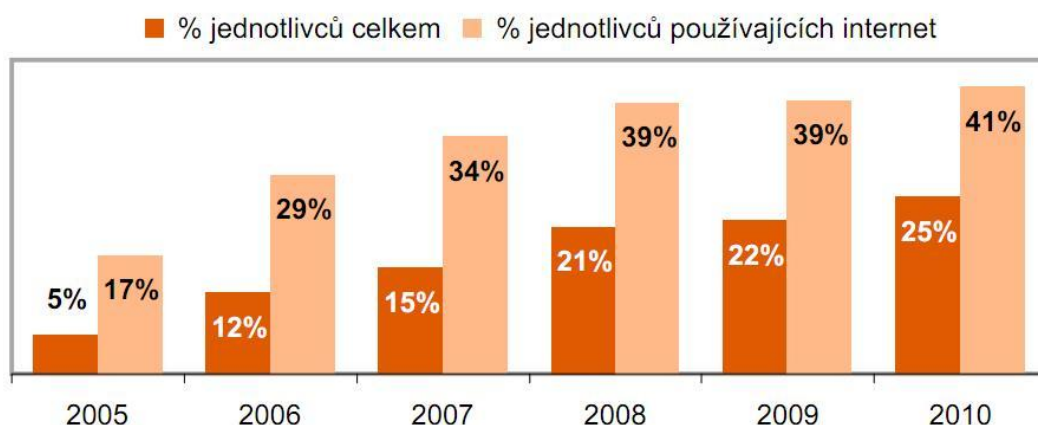
Graf č. 1 jasně vypovídá o rostoucím zájmu o e-nákup mezi lety 2005 a 2010, kdy se počet nakupujících zvýšil o 500 %. Mění se také struktura on-line nakupujících – v roce 2004 tvořili muži 67 % všech nakupujících na internetu, v roce 2010 to je 55 %.

Nejpopulárnější je on-line nakupování mezi jednotlivci s vysokoškolským vzděláním (v posledním roce nakoupilo přes internet 44% všech jednotlivců s vysokoškolským vzděláním), dále mezi studenty (40 % z nich v posledním roce nakoupilo přes internet), obyvateli Prahy (30 %) a jednotlivci ve věku 25 až 34 let (44 %).

33% všech on-line nakupujících jsou jednotlivci ve věku 25–34 let. Aktivní v nakupování přes internet jsou i jednotlivci ve věku 16–24 let, kteří tvoří 23% všech on-line nakupujících. Jednotlivci ve věku 35–44 let tvoří pětinu všech nakupujících na internetu.

Nejčastěji nakupovaným zbožím (resp. službou) jsou – vstupenky, oblečení/obuv, knihy, časopisy a učebnice, elektronika, kosmetika, sportovní potřeby a bílá technika.

Muži ve srovnání se ženami nakupují výrazně více elektroniku, mobilní telefony, fotoaparáty, filmy a hudbu, počítačový software a hardware. Ženy ve srovnání s muži nakupují více kosmetiku, oblečení a obuv, hračky, jiné vybavení bytu/domácnosti.(8)



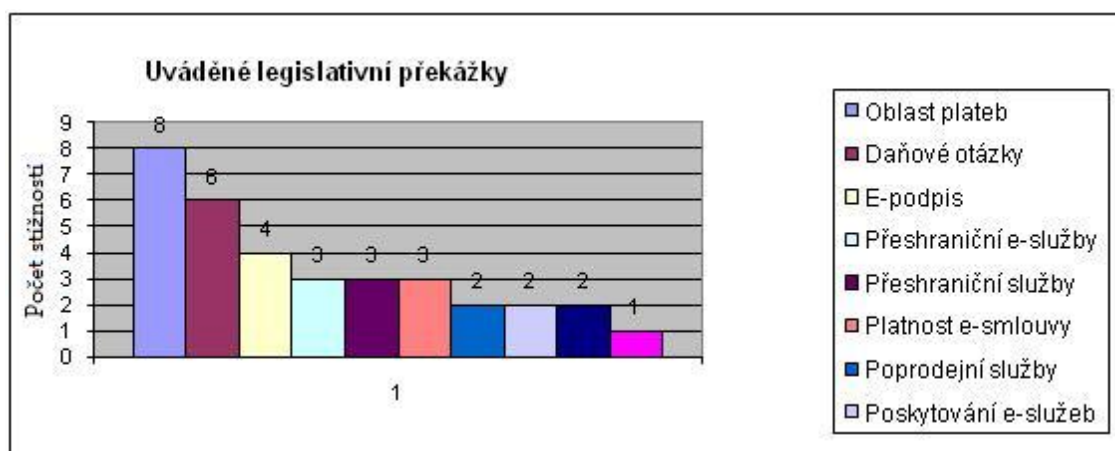
Graf 1: Procento jednotlivců, kteří na internetu nakoupili alespoň jednou za daný rok (8)

2.3 Právní překážky elektronického obchodování

Evropská komise si je plně vědoma, že pro další rozvoj elektronického obchodu je důležité odstranit všechny potenciální právní překážky ve všech členských i kandidátských zemích. Pro tento účel připravila v roce 2005 speciální hodnotící dotazník, jehož cílem bylo zmapovat názory firem, které s elektronickým obchodem

mají praktické zkušenosti. V České republice propagovalo tento průzkum Centrum pro elektronický obchod (www.CentrumEO.cz) nejen aktivním doporučením vyplnit přímo dotazník Evropské komise na oficiálních WWW stránkách, ale též připravilo vlastní českou verzi, která byla zaslána nejen hlavním hráčům na IT trhu, ale díky partnerské spolupráci s firmou Zoner i zhruba 800 provozovatelům elektronických prodejen (e-shopů na platformě Zoner inShop).

Z 30 částečně či kompletně vyplněných dotazníků vyplývají tyto konkrétní oblasti, kde společnosti vnímají legislativní překážky elektronickému podnikání:



Graf 2: Uváděné legislativní překážky (21)

Kromě výše uvedených důvodů uváděli zástupci firem ještě tyto důvody: potřeba bankovní licence pro provozování mikroplatebního systému, problematické uznávání elektronicky podepsaných dokumentů státní správou (finančními úřady), příliš komplikované nároky na účetní operace, a řešení otázek v souvislosti s ochranou soukromí, podmínkami prodeje a reklamačním řádem. Mezi dotázanými společnostmi převažuje ochota podělit se s Komisí o svoje zkušenosti.

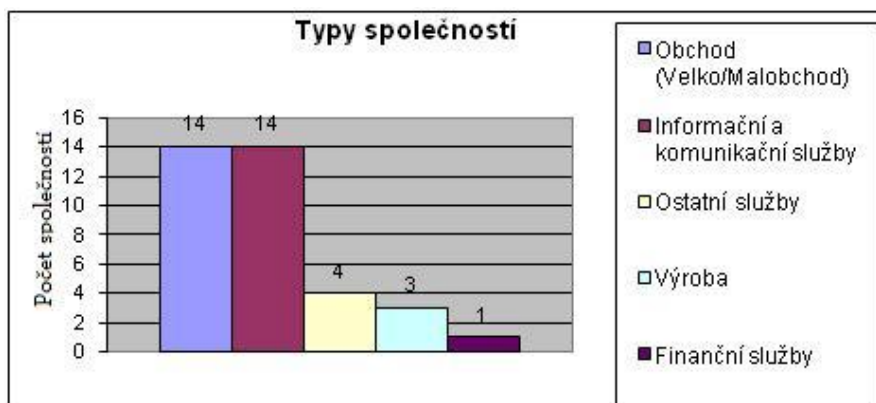
V 7 případech byl protistranou v problému spotřebitel či spotřebitelská asociace, v 5 případech veřejná správa a ve 2 případech společnost či oborová asociace. Valná většina stížností byla lokálního charakteru (9 v rámci ČR), pouze 3 případy narazily na obtíže v přeshraničním elektronickém podnikání (konkrétně Čína, Německo, Slovensko, Itálie, Maďarsko, Polsko a Nizozemí).

Alarmujícím faktem je, že v 5 případech způsobila daný legislativní problém nedostatečná transparentnost existující legislativy a nedostatečné znalosti o existující

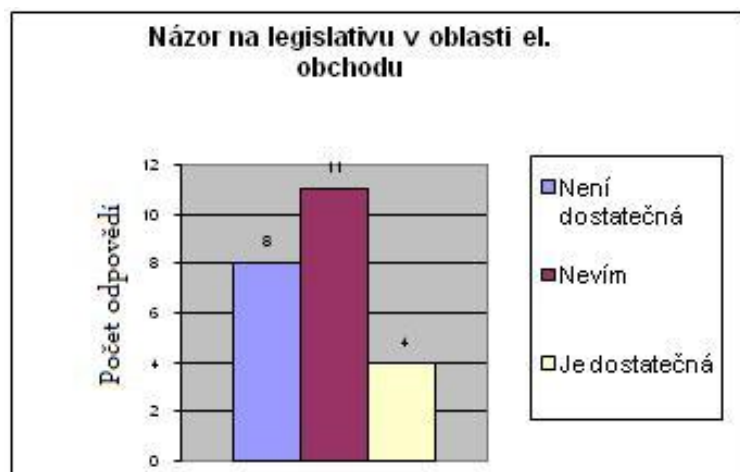
legislativě, s odstupem se jedná o rozdílnou legislativu platnou v jiných zemích (2x), absence právní regulace dané otázky (1x) a porušení platné legislativy protistranou, včetně nekalé konkurence (též 1x).

„Zajímavým výsledkem průzkumu legislativních překážek je kromě konkrétních identifikovaných problematických otázek ještě zjištění, že ani společnosti, které se elektronickým podnikáním přímo živí, nejsou příliš dobře informovány o platné legislativě a pokud jsou, tak se spíše domnívají, že je nedostatečná“ uvádí Michal Zálešák - předseda, Centrum pro elektronický obchod.(21)

Centrum pro elektronický obchod je odbornou sekci České společnosti pro systémovou integraci, která sdružuje firmy a jednotlivce s profesním zájem o problematiku elektronického obchodu a e-governmentu. Na jeho podporu realizuje odborné semináře, průzkumy a jiné osvětové aktivity, podílí se na připomínkování nově připravované legislativy, spolupracuje s klíčovými institucemi a ostatními asociacemi a snaží se aktivně napomáhat své členské základně v rozvoji elektronického podnikání.(21)



Graf 3: Typy společností (21)



Graf 4: Názor na legislativu v oblasti el. obchodu (21)

2.4 Cenová politika prodejců

Ve světě se cenová politika u různých obchodních řetězců liší. Někde nabízejí na webu výrobky za ceny shodné s těmi ve své kamenné síti, jinde jsou ceny diferencované a nejčastěji jde o kombinaci obojího.

Obecně lze říci, že rozdíly v cenách na internetu a v prodejnách jsou menší v USA než v Evropě. Zajímavým faktem je, že pokud nakupujete v USA na internetu a jste z jiného státu než samotný prodejce, neplatíte státní daň. Třeba u elektroniky ušetříte mezi 7 - 10 % z ceny zboží.

„Neplatí rozšířené klišé, že ceny na internetu jsou obecně mnohem nižší než v kamenných prodejnách. Snažíme se porovnávat ceny co největšího množství našich výrobků u e-konkurence a ceny těch výrobků, které jsou u ní levnější, dorovnáваме.“ vysvětluje cenovou politiku českého průkopníka v propojení obchodního řetězce s e-shopem, společnosti Datart, Jiří Matoušek.(15)

„Podíl výrobků, který je na Datart.cz nyní levnější než v našich prodejnách činí zhruba jednu polovinu. Druhou polovinu tedy tvoří výrobky s cenami shodnými nebo naopak nižšími na našich prodejnách. Přesto konkurujeme na internetu také cenou. V budoucnu se rozdílné ceny internetu oproti kamenným prodejnám přiblíží ještě více.“ předpovídá Jiří Navrátil.(15)

2.5 Rozdělení elektronického obchodu

B2B – Business to Business

Je druhem elektronického obchodování, které využívají distribuční a prodejní sítě, ve kterých mohou mezi sebou komunikovat výrobci, pobočky, distributoři, velkoobchody, dealeři nebo obchodní zástupci. Rozdílem mezi obchody B2B a B2C (viz dále) je to, že prodávající strana (distributor, výrobce) zná předem nakupujícího. Většinou se jedná o partnera, který má předem stanoveny obchodní podmínky, za kterých může nakupovat. Klasickým příkladem elektronického obchodu B2B jsou elektronická tržiště, na která mají přístup pouze registrovaní účastníci. Někteří velcí odběratelé organizují taková tržiště formou dražby, kdy za minimálních nákladů se během relativně krátkého času shromáždí velké množství nabídek.(21)

B2C – Business to Consumer

Tento typ elektronického obchodu je zaměřen na prodej koncovým zákazníkům – spotřebitelům, jedná se tedy o jakousi obdobu kamenného obchodu na internetu. Oproti klasickému obchodu má ale internetový obchod po nákupu kompletní informace o kupujícím, může je tedy využít pro své strategie prodeje a marketingu, individuálnímu přístupu, případně může zákazníkovi nabízet nové zboží pomocí e-mailové komunikace.(21)

2.6 Elektronický podpis

Elektronickým podpisem jsou údaje v elektronické podobě, které jsou připojené nebo logicky spojené s datovou zprávou a které jsou použity ke zjištění totožnosti oprávněné osoby ve vztahu k datové zprávě. Pro praxi je významné, aby tento podpis byl tzv. bezpečným či zaručeným. Tuto náležitost splňuje elektronický podpis vydaný tzv. certifikační autoritou tj. subjektem, který má k vydávání el. podpisu - klíče - odpovídající povolení.

Certifikační autority lze snadno nalézt na internetu. Zaručené elektronické podpisy tedy splňují následující požadavky:

- a) jsou jednoznačně spojené s podepisující osobou
- b) umožňují identifikaci podepisující osoby ve vztahu k datové zprávě
- c) byly vytvořeny a připojeny k datové zprávě pomocí prostředků, které podepisující osoba může udržet pod svou výhradní kontrolou
- d) jsou k datové zprávě, ke které se vztahuje, připojeny takovým způsobem, že je možno zjistit jakoukoliv následnou změnu dat

3 Teoretická východiska řešení

V dnešní době se ke zlepšení ekonomických výsledků každé společnosti používá mnoho různých nástrojů a marketingových strategií. V této kapitole se pokusíme nastínit hrubý obraz dopadu investice společnosti do realizace 3D katalogu, který by měl poskytnout společnosti výhodu nad konkurencí v oblasti, která je dnes ještě nepříliš prozkoumanou a neužívanou i přes svůj nesporný potenciál.

Definice: 3D či 3-D je zkratka výrazu „trojdimenzionální“, „trojrozměrný“ a označuje svět, který je možné popsat třemi rozměry (viz kartézská soustava souřadnic), předměty ve trojrozměrném světě mají objem, často také označuje techniky používané pro zobrazení či prohlížení zdánlivě trojrozměrných objektů na plochém (dvojrzměrném, 2D) médiu (na papíře, filmovém plátnu, počítačové obrazovce apod.).(32)

3.1 Proč vytvářet 3D elektronický obchod

- Vylepšená možnost prezentace nabízeného produktu zákazníkovi
- Získání náskoku před konkurencí
- Inovativní přístup s ohledem na nové možnosti techniky

3.2 Způsoby převodu na 3D

3.2.1 Fotostudio

Patrně nejsnazším a nejrychlejším způsobem převodu objektu do podoby trojrozměrného zobrazení je použití tzv. fotostudia.

Principem metody je nafocení předmětu z několika stran a poté spojení těchto fotografií za sebe do sekvence, ve které se následně můžeme pohybovat a tím si daný předmět prohlížet. Nejběžněji jde jen o rotaci okolo osy y - otáčení. Na celý kruh (360°) je použito 12 fotografických snímků, vždy po 30°. Pokud se přidá i možnost rotace okolo osy x- nahléd, vzroste několikanásobně výsledný objem dat dle toho, kolik dalších

úhlu pohledu přidáme. Na konkrétním příkladu klíče ze stránek www.innovate.cz můžeme klíč na ose x otočit o 90° po 4 krocích, celkem jde tedy o $12 \times 4 = 48$ fotografií.

Fotografie se pořizují ve fotoateliéru společnosti, její zástupci si osobně převezmou výrobek, nafotí a vrátí ho zpět. Na velikosti objektu nezáleží, je možné fotit maličkosti (šperky, zapalovače, sponky) i věci větších rozměrů (motorka, křeslo atd.).

Kompletní sada snímků jednoho výrobku je pořizena do 2 minut, zpracování pro web všech snímků je pak do 2 hodin. V případě předmětů s většími rozměry není problém dovézt fotografické studio k zákazníkovi a nafocení provést v jeho prostorách.

Výsledné 3D obrázky je možné ihned umístit na webové stránky. Není potřeba speciální software a obsluha prohlížeče je snadná a intuitivní. Do současné webové stránky stačí vložit jednoduchý kód, který je součástí dodávky 3D fotografie. Jednotlivé foto záběry je poté možné použít např. i pro tiskoviny.(10)

Cena

- Mobilní fotostudio

Pronájem závisí na lokalitě a požadovaném rozsahu. Orientační cena je 25 000 Kč za den, podrobnější a přesnější kalkulaci provede poskytovatel individuálně.(17)

- 3D fotostudio

Zajištění 3D fotografie se vždy odvíjí od počtu výrobků. Cena za jednotlivé kusy je uvedena v tabulce níže. Při větších objemech připravujeme individuální kalkulace.(10)

Služba	Cena do 3 ks	Cena při 4 - 10 ks
3D foto - otáčení 2D	2 800 Kč/výrobek	1 700 Kč/výrobek
3D foto - otáčení 3D	3 500 Kč/výrobek	1 900 Kč/výrobek
3D foto - člověk	4 500 Kč/snímek	3 500 Kč/snímek
3D katalog	6 900 Kč/katalog	

Tabulka 1: Orientační cena za 3D foto

- Hosting

Každý nafocený výrobek, osobu nebo 3D katalog je nutné umístit na server podporující 3D prohlížeč. Cena za hosting každých 100MB dat je 200 Kč/měsíc. Při větším objemu dat nabízí poskytovatel výrazné slevy.

Ceny u 3D FOTO a FOTO ČLOVĚK jsou vždy uvedeny za jeden snímek i s přípravou pro umístění na web. Cena za množstevní objednávky nad 10 ks je předmětem individuálních kalkulací. Cena je bez dopravy, programování v rámci zákaznickových webových stránek, případné náklady spojené s přípravou focení (800 Kč/hod), pronájmem prostor a DPH. Konečná cena je vždy specifikovaná po upřesnění objemu a náročnosti zakázky.

Klady

- Rychlost vytvoření modelu
- Cena za vyhotovení
- Fotografická přesnost obrazu
- Při použití dobré komprese nízká velikost výsledného modelu

Zápory

- Natáčení po „skocích“
- Při zoomu ztráta kvality obrazu
- Pro možnost změny barvy nebo rotace části modelu je nutné nafotit další sérii fotografií – vyšší cena
- Nemožnost pozdějších úprav
- Jen pro menší objekty

Tato metoda realizace je vhodná pro tvarově složité modely, které je možné nafotit, tzn. jsou společností fyzicky dostupné a současně by neměly být větší než motocykl nebo křeslo. S úspěchem se touto metodou vytváří elektronické brožury, které je možné prohlížet na webových stránkách hypermarketů.

3.2.2 3D pdf

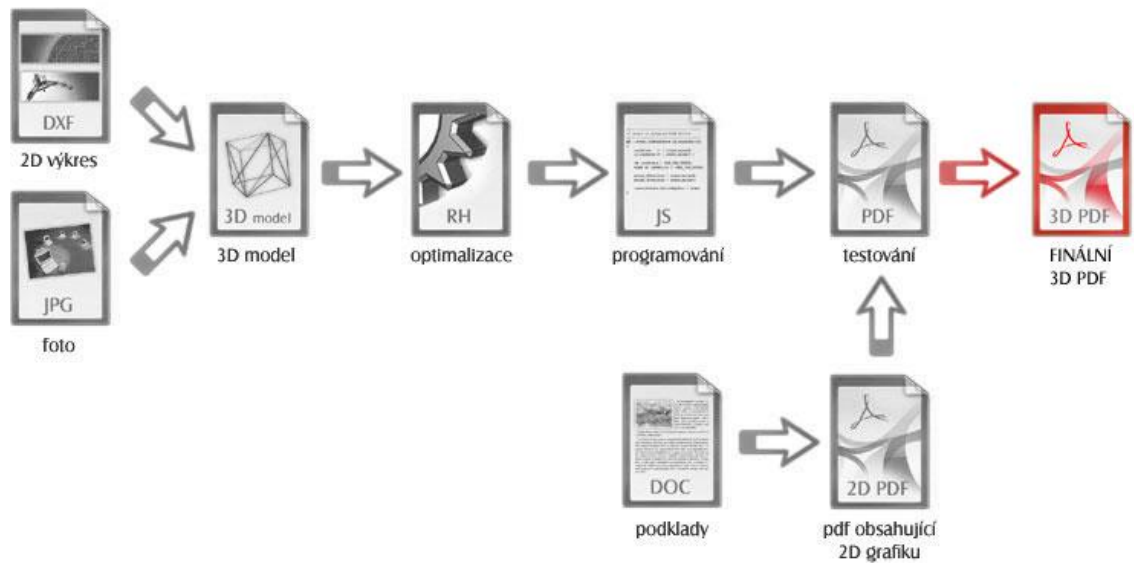
Možnost využití 3D interaktivní vizualizace při prostorovém zobrazení výrobků v propagačních pdf 3D dokumentech je díky programovatelnému rozhraní Adobe Acrobatu pdf 3D takřikajíc neomezená. Tato interaktivita umožní zákazníkům získat veškeré potřebné informace o výrobku. Vytvořením názorných animací, změnou materiálů i barev, posunem objektů, znázorněním montážních a demontážních postupů, může být snadno prezentován výrobek „na dálku“ v plné funkčnosti.

Přípravné práce jsou závislé na dodaných podkladech a od těchto se poté odvíjí výrazným způsobem i cena výsledného projektu. Většinou objednavatel nevlastní již předpřipravený 3D model, proto vše začíná převodem dvojrozměrných konstrukčních plánů do trojrozměrného modelu. U jednodušších tvarů může postačovat fotografický materiál s měřítkem.

Dalším krokem je příprava 3D modelu pro implementování do pdf - vhodné nastavení textur, barev, materiálů, osvětlení, přípravu animací a to vše s ohledem na použití v pdf dokumentu.

Velkou výhodou je programování speciálních funkcí, které nejdou připravit jako standardní animace – posuny objektů, akce spouštěné při označení jednotlivých částí modelu, změny textur, materiálů a barev, pohyb kamery, apod.

Posledním krokem je napojení naprogramovaných akcí mezi pdf dokumentem a 3D objektem, pokud zákazník dokument v pdf formátu nemá, nebo má zájem o novou verzi optimalizovanou pro 3D model, většinou je grafické studio schopno zajistit i tuto realizaci dle přesného zadání zákazníka.(1,13)



Obrázek 1: Realizace 3D pdf (1)

Cena

Celková cena za vytvoření 3d pdf je součtem dílčích úkonů zobrazených v předešlém schématu. Celkovou cenu proto zásadně ovlivňují dodané podklady a požadavky na funkčnost. Z tohoto důvodu nelze cenu vměstnat do tabulky. Pro každou zakázku je třeba připravit kalkulaci (většinou bezplatně) na základě poskytnutých požadavků - specifikací vlastností, které jsou od 3D pdf očekávány a možnostech dodání vlastních podkladů. Pro základní představu o ceně za interaktivní 3D PDF lze uvést, že spodní hranice ceny v případě dodání 3D elektronických dat začíná od 4 000,- Kč a může se vyšplhat až k desítkám tisíc korun dle typu 3D objektu, požadavků a náročnosti zhotovení. Zákazník tedy musí zvážit tyto základní otázky:

1) Jaké vlastní podklady je schopen dodat

3D model v elektronické podobě

2D výkres v elektronické podobě

2D výkres v papírové podobě

Fotografie výrobku

PDF - obsahující grafiku stránky, na které bude 3d model zobrazen

2) Požadavky na funkčnost

Zde je třeba co nejpřesněji popsat požadavky a funkce, které jsou od 3D modelu očekávány. Malým průvodcem interaktivitu modelu mohou být internetové galerie nebo ukázky realizovaných prací reklamní společnosti.

3) Dokumenty pdf

Bude nový model zařazen do stávajícího katalogu, nebo bude třeba vytvořit pro model katalog nový?

Klady

- Možnost vytvoření modelu zatím neexistujícího předmětu/stroje
- Dokonalá možnost prohlédnutí a změny modelu – rotace, zoom, změna barevného schématu, rozklad na součásti
- Neexistují žádná omezení velikosti předlohy

Zápory

- Vysoká cena projektu
- Časová náročnost zhotovení modelu
- Objemnost dat
- Vysoká hardwarová náročnost
- Omezená věrohodnost – není to realita

3.2.3 3D grafika

Je způsob velmi podobný 3D pdf, ale pro převádění modelu do trojrozměrného počítačového prostředí můžeme využít širší paletu nástrojů a programů, kdy jsou pro internetové prezentace využity nejmodernější technologie, jako jsou například X3D, VRML, OpenGL, Flash, Java a JavaScript, Adobe Acrobat, 3D PDF, CAD, DWG, DXF, DGN, CAE, CAM, ActiveX, Direct3D, PHP, PAL, C++, DDS, DWF, FBX, Gbel, KML, DWF, HTML, RGS, 3DS Max, stereo vize, virtuální realita a mnohé další.

Tyto programy se používají pro složitější modely, jakými jsou například nemovitosti, automobily a jiné velké objekty, za jejichž perfektní prezentaci neváhají zadavatelé utratit nemalé prostředky. Velmi drahé jsou totiž jak programy, ve kterých animátoři pracují, tak jejich samotná práce a když vezmeme v potaz počet lidí, kteří na celém výsledném projektu podílejí, a čas, který na modelaci složitých modelů stráví, je jasné, že cena modelů může dosáhnout stovek tisíc korun.(6)

Klady

- Libovolný pohyb okolo/v modelu
- Možnost měnit součásti – libovolné nastavení uživatelem
- Pořizování renderů z 3D animace
- Neomezené možnosti

Zápory

- Dlouhá doba realizace
- Cena výsledného projektu
- HW náročnost

3.2.4 3D fotografie

3.2.4.1 Anaglyfy

Anaglyf je druhem 3D fotografie, kterou zná snad každý – jedná se o dvě běžné fotografie, každá z mírně jiného úhlu (obraz pro levé a pro pravé oko), které jsou pomocí speciálního programu proloženy, čímž vznikne jedna nová fotografie s 3D efektem.

Na prohlédnutí anaglyfu je nutné mít speciální brýle – známé s modrým a červeným sklíčkem, což jsou vlastně filtry, které umožní každému oku vidět mírně jiný obrázek a tím vytvořit iluzi 3D fotografie s použitím obyčejného fotoaparátu.

Určitým nepříjemným omezením tohoto způsobu trojrozměrného obrazu je značně špatné podání červené a modré barvy na fotografii kvůli výše zmíněným barevným filtrům.



Obrázek 2: Anaglyf (27)

Jak je vidět na obrázku 2, anaglyf vznikne překrytím dvou fotografií a to tak, že objekty, které jsou blízko pozorovateli, jsou více „rozostřené“ a ty, které jsou vzdálené, jsou téměř beze změny.

Pokud se například na výše uvedený obrázek podíváte se speciálními brýlemi, levé oko (s červeným filtrem) neuvidí červené obrysy, ale pouze modře posunuté obrysy fotografie a pravé oko (s modrým filtrem) zase neuvidí modré obrysy, ale jen ty červené. Kvůli samotnému principu vytváření efektu u anaglyfu je tedy podání červené a modré barvy velmi slabé.

Pro komerční účely je tato technologie nevhodná a její využití – kvůli snadné realizaci a velice levným brýlím – se omezuje téměř výlučně na časopisy pro děti, případně jako zpestření pro domácí fotografie z dovolené.



Obrázek 3: Anaglyfické brýle(28)

Klady

- Snadné na vytvoření
- Postačí běžný fotoaparát
- Velice levné 3D brýle

Zápory

- Špatné podání modré a červené barvy
- Nutnost vždy vyfotit alespoň 2 fotografie, těžké zachytit pohybující se objekt

3.2.4.2 3D fotoaparáty

Fujifilm jako první výrobce na světě ve 3. čtvrtletí roku 2009 začal prodávat 3D fotoaparát.

Trojrozměrného efektu fotografie je dosahováno tím, že je fotoaparát vybaven dvěma objektivy, které snímají fotografované objekty z mírně jiného úhlu a pomocí speciálního programu a s využitím Real Foto Procesoru 3D jsou tyto fotografie proloženy a vytvoří jednu novou, trojrozměrnou fotografii. Navíc je možné s tímto druhem fotoaparátu natáčet i trojrozměrná videa.(25)

Součástí „balíčku“ firmy Fujifilm je i fotorámeček a 3D fototiskárna, která, při použití speciálního papíru, umožňuje vidět trojrozměrnou fotografii i bez použití speciálních 3D brýlí.(9)



Obrázek 4: Fujifilm FinePix REAL 3D W1 (9)

Pro prohlížení zachycených fotografií a videí slouží zadní display, který díky speciální technologii umožňuje prohlížení záznamu trojrozměrně a to i bez použití 3D brýlí.(9)



Obrázek 5: Dvě fotografie z různých úhlů a její zpracování do 3D fotografie (25)

3.3 3D zobrazovací techniky

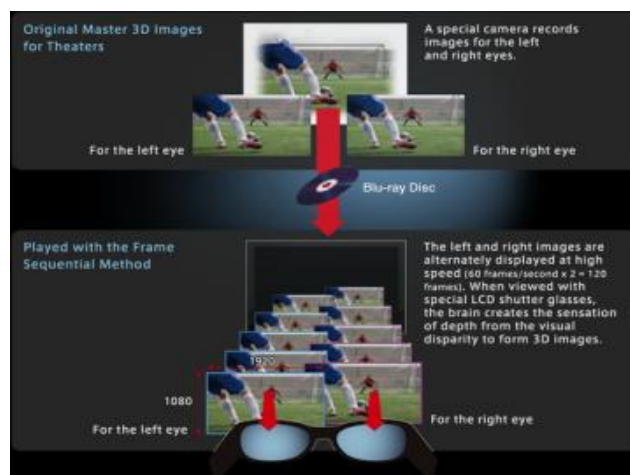
3D zobrazování je v dnešní době velmi slibnou technologií, která je prozatím jen okrajovou funkcí moderních televizí.

Nejvíce potenciálních zákazníků se s 3D obrazem setkala v kinech. Průkopníky v této oblasti byla kina společnosti IMAX, v současné době jsou 3D filmy promítány i v multikinech společnosti Palace Cinemas.(12)

3.3.1 Aktivní 3D projekce

Diváci sledují obraz, který se promítá na plátno (nebo i monitor či televizor) s dvojnásobnou snímkovou frekvencí, přičemž na filmovém pásu jsou střídavě proložené obrazy pro levé a pravé oko. Elektronické brýle diváka se dálkově (většinou s pomocí IrDA paprsku, nebo kablíkem) synchronizují se zdrojem vysílání a střídavě zatmívají levé nebo pravé oko.

Výsledkem je, že každý lichý snímek vidí návštěvník kinosálu jedním okem a každý sudý okem druhým. Tímto systémem se sice sníží frekvence promítaných obrazů na každé oko na polovinu, nicméně každé oko návštěvníka kina dostává pouze správně předurčený obraz. Z dvojice oddělených snímků mozek následně skládá skutečnou trojrozměrnou scénu. Takovýto systém využívá například i 3D kino IMAX SOLIDO, nebo dostupné domácí stereosestavy. Výhodou je, že k této projekci není potřeba žádného speciálního projekčního plátna nebo monitoru.(3)



Obrázek 6: Aktivní 3D projekce (25)

Monitor nebo projektor s pomocí zdroje signálu generuje při vysoké frekvenci (100 – 120 Hz) dva různé obrazy. Uživatel má brýle synchronizované se zobrazovacím zařízením (většinou s pomocí IrDA paprsku). Brýle střídavě díky LCD panelům v očních zatměňují levé a pravé oko, tak aby došlo ke správnému rozdělení obrazové informace.(3)

Výhody:

Kvalitní plnobarevné zobrazení. Pro projekci stačí normální projekční plátno. Metoda také funguje velmi dobře i se standardními CRT monitory.

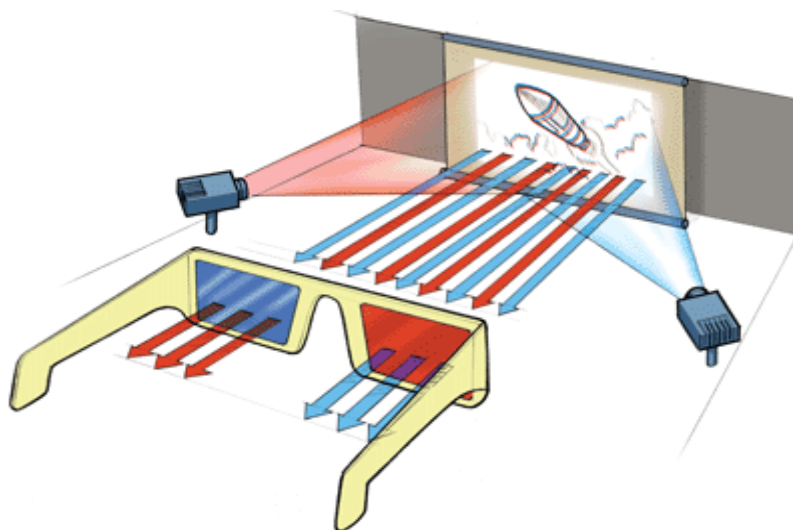
Nevýhody:

Každý uživatel musí mít poměrně komplikované elektronické brýle – není vhodné pro velký počet lidí. Projekce vyžaduje specializovaný nestandardní projektor

3.3.2 Pasivní 3D projekce

Pasivní 3D projekce je založena na brýlích, které mají v očních polarizační filtry. Jedna očnice má polarizační filtr orientovaný tak, že propouští pouze světlo kmitající v horizontální rovině. Druhá očnice obsahuje stejný o devadesát stupňů otočený filtr. Tedy takový, že propouští pouze světlo kmitající ve vertikální rovině.

Dva obrazy se promítají na jednu projekční plochu, přičemž před každým projektorem je upevněn taktéž polarizační filtr. Nastavení filtrů na projektoru koresponduje s nastavením filtrů na brýlích. Dvojice obrazů (pro pravé a levé oko) se následně promítá na jednu projekční plochu, která je vyrobena ze speciálního materiálu a opatřena povrchem, který zachová polarizaci dopadajícího světla. Odražené obrazy od projekční plochy se dostávají k divákovi, nicméně do každého oka pronikne (díky polarizačním filtrům v očních) pouze příslušný obraz.



Obrázek 7: Pasivní 3D projekce (18)

Na projekční plátno se promítá ze dvou projektorů. Každý z projektorů má filtr, který propouští světlo kmitající jen v určité rovině. Uživatel má brýle s optickými polarizačními filtry.(18)

Výhody:

Kvalitní obraz ve vysokém rozlišení. Vhodné pro větší počet lidí (metodu používají standardní kina IMAX 3D). Velmi stabilní obraz.

Nevýhody:

Jsou zapotřebí 2 projektory. Je nutné speciální projekční plátno s nedepolarizujícím povrchem. Pro pasivní projekci nelze použít monitor.

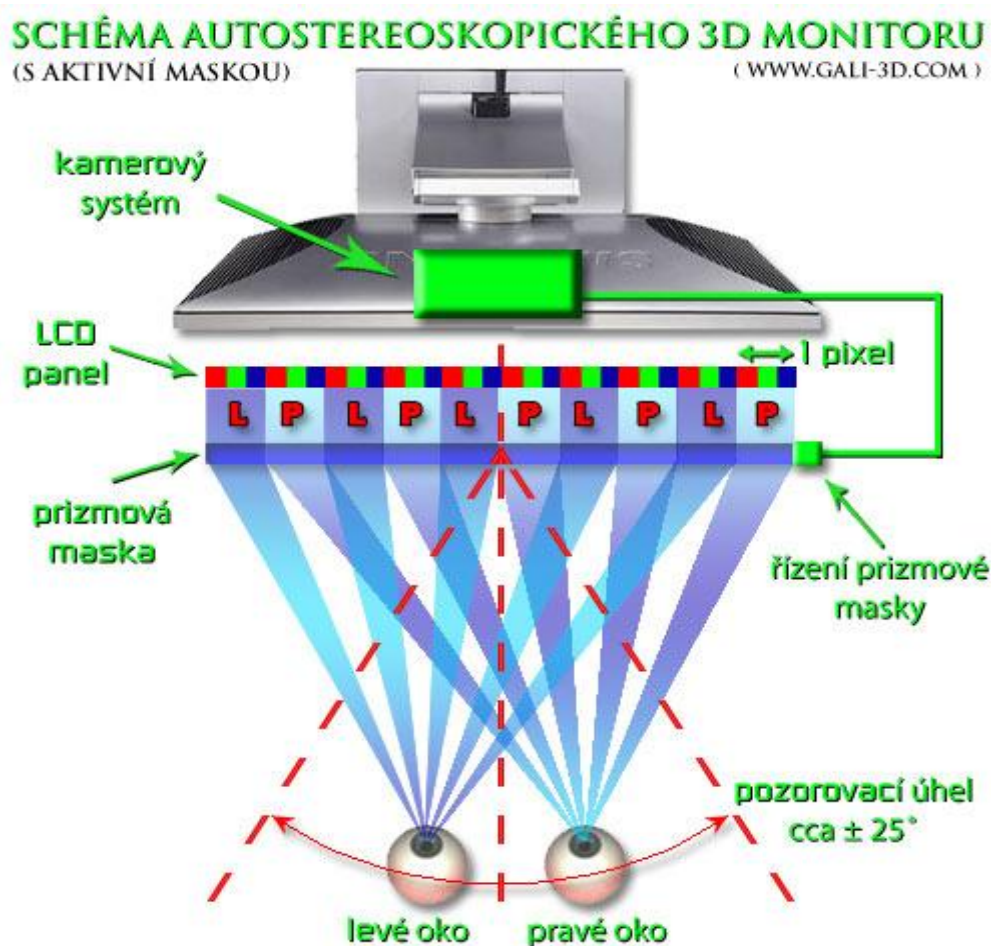
3.3.3 Autostereoskopická 3D projekce

Monitory založené právě na tomto principu mají velkou výhodu, že není potřeba sledovat scénu s brýlemi, ať již pasivními, či aktivními. Před (většinou LCD) monitorem je umístěna speciální fólie (někdy nazývána jako prizmová maska), jejímž úkolem je lámat různé svislé pixelové sloupce vedle sebe vždy trochu jiným směrem.

Některé monitory nabízejí dva směry. Tudíž každý lichý pixelový sloupec je zlomen směrem jedním a každý sudý svislý sloupec pixelů směrem druhým. V tomto případě dostává divák do každého oka svůj oddělený obraz a mozek skládá 3D scénu

správně. U pasivních autostereoskopických monitorů je nucen uživatel sedět v předem dané pozici před obrazovkou, při vychýlení z tzv. "sweet spotu" dojde ke ztrátě 3D vjemu. Tuto situaci výrazně řeší aktivní stereoskopický monitor, který s pomocí kamerového systému sleduje oči uživatele (v jiné variantě čelenku na hlavě uživatele) a upravuje prizmovou masku před monitorem tak, aby i při výrazném pohybu hlavy nedocházelo k porušení 3D obrazu.

Vyjma těchto dvou druhů monitorů, existuje také další zajímavá verze, hodí se např. pro reklamní účely a spočívá v tom, že je lom světla veden až do celkem devíti směrů, tudíž se na monitor může současně dívat větší množství diváků a přitom je velkou výhodou, že scénu mohou dokonce i obcházet - devět obrazů je totiž generováno synchronně. Při pohybu před monitorem tak diváci pozorují scénu s více úhlů, což je velmi poutavé.(4)



Obrázek 8: Autostereoskopická projekce(4)

Výhody:

Plnobarevný obraz. Nejsou třeba žádné brýle. Malé nároky na prostor a pohodlí bez brýlí při sledování 3D.

Nevýhody:

Autostereoskopický monitor je drahé zařízení. Zařízení není vhodné pro velký počet pozorovatelů najednou.

3.3.4 Očekávaný vývoj prodeje 3D televizí

Dle odhadů výrobců podle vývoje odbytu se v roce 2011 zvýší prodej televizí s trojrozměrným obrazem více než pětkrát. V tržbách za ploché televize se tak jejich podíl zvýší na 11 procent, uvedla to průzkumová agentura IHS iSuppli. Prodej 3D televizorů roste díky vyšší výrobě a klesajícím cenám.

iSuppli předpovídá, že výrobci letos dodají na světový trh 23,4 miliónu 3D televizorů proti 4,2 miliónu za minulý rok. Do roku 2015 by se roční prodej mohl zvýšit na 159 miliónů 3D přístrojů. V té době budou podle iSuppli 3D přístroje představovat více než polovinu prodeje plochých televizí.

Výrobci v čele s korejskými koncerny Samsung Electronics a LG Electronics a japonskou Sony se spoléhají na to, že jim trojrozměrný obraz a další dražší přístroje pomůžou vymanit se z nízkých ziskových marží. Rentabilitu odvětví napínají nízkonákladoví výrobci.

Jiná průzkumová agentura DisplaySearch minulý týden předpověděla, že na trhu s 3D televizory vzrostou tržby o 50 procent a odvětví do roku 2014 dosáhne úrovně 100 miliónů přístrojů, napsala agentura Reuters.(7)

3.3.5 Srovnání 3D zobrazovacích metod

V tabulce je vidět, že některé stereoskopické metody jsou vhodnější na různých místech. Například pasivní metoda se nejvíce hodí pro projekce s vyšším počtem diváků - např. 3D kino, větší firemní prezentace. Aktivní metoda je vhodnější pro menší skupinu, nebo pro domácí využití např. pro hraní 3D her na PC a autostereoskopický monitor obdobně. U něj je dokonce dobré využití v reklamě a při výstavách

a exhibicích. Anaglyph, který je finančně nejdostupnější je velmi špatná metoda a hodí se pouze pro specifické, víceméně jen reklamní využití.

Metoda	Barevná informace	Rozlišení obrazu	Vhodné pro projekci	Zobrazení na monitoru	Počet diváků	Náklady
Anaglyph	kompletní ztráta	střední	ano	ano	vysoký	velmi nízké
Aktivní zobrazení	plná	vysoké	ano	ano (ne LCD)	omezený	vyšší
Pasivní zobrazení	plná	vysoké	ano	ne	vysoký	střední
Autostereoskopický monitor	plná	střední	ne	ano	velmi omezený	vyšší

Tabulka 2: Srovnání 3D zobrazovacích technologií (3)

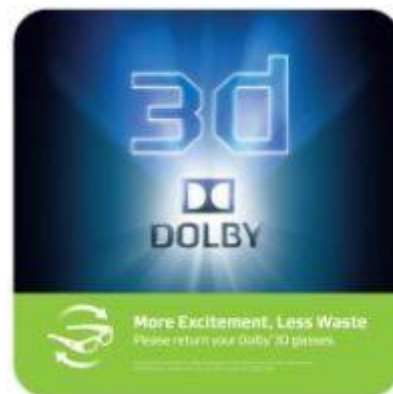
3.4 3D kino

V roce 1986 bylo u příležitosti Expo 86' ve Vancouveru otevřeno první stálé 3D kino společnosti IMAX. Trojrozměrné filmy používaly (a dodnes převážně používají) 70 mm kinofilm, který nabízí ekvivalent rozlišení 4096x2160 obrazových bodů. Tuto technologii dnes využívá asi 280 kin IMAX po celém světě.

Pro snazší a levnější distribuci je většina dnešních 3D kin vybavena digitálními přehrávači záznamu, který ale bohužel ztrácí digitalizací na kvalitě. Tuto technologii používají „běžná“ 3D kina společnosti Palace Cinemas a asi 150 kin společnosti IMAX.(2)

3.4.1 Dolby 3D

Začneme podle abecedy, tedy prvním systémem je systém nabízený renomovanou společností Dolby. Ten využívá k oddělení obrazu pro levé a pravé oko princip oddělení vlnových délek ve třech barevných složkách. Systém požaduje instalaci rotačního dvousložkového filtru do samotného projektoru, který je ovládán ze speciálního modulu propojeného s kinoserverem.



Pro diváky je systém dle mého názoru pohodlný. Brýle mají i díky vyšší výrobní ceně filtrů menší skla, takže v kombinaci s dioptrickými brýlemi u krátkozrakých diváků se mohou zdát subjektivně malá. Zatím jsem ale nezaznamenal problém s výhledem na celé plátno i u panoramatických sálů.

Pro provozovatele kin má Dolby 3-D jednu obrovskou výhodu, a sice kompatibilitu s klasickým bílým plátnem. To umožňuje zachování i 2-D projekce jak z digitálního, tak i klasického 35 mm filmového projektoru. Jedná se tedy o systém, který bych označil za vhodný zejména pro jednosálová kina. Cena brýlí se pohybuje v řádech stokorun za jeden kus a v případě poškození lze v brýlích měnit jen jednotlivé filtry. Vyšší provozní náklady pak ale zaznamenáme na spotřebě energie a životnosti projekční lampy, neboť systém má vyšší ztrátu světla a pro kvalitní zobrazení je nutné zvýšit příkon na usměřovači.(2, obrázek tamtéž)

3.4.2 Master Image

Korejský systém MasterImage je levnější variantou využívající systém kruhové polarizace obrazu. Jedná se o zařízení, které se na vozíku umísťuje co nejbližší před objektiv projektoru. Ve starší verzi ve vlastním zařízení rotují dva kotouče se dvěma sadami polarizačních filtrů proti sobě. U nové verze MI-2100 jde



již o kotouč jediný. Zařízení je samozřejmě propojeno s kinoserverem pro správnou synchronizaci obrazu.

Z pohledu diváka bych měl k vlastnímu zobrazení 3-D menší výhrady u starší verze, u které když si provozovatel kina nedá pozor a objedná DCP digitální kopii bez kompenzace duchů, takzvanou „Non Ghost Busted“, se může stát, že se občas obraz rozdvojí a stereoskopický vjem se ztratí. U MI-2100 jsem osobně zatím tento problém nezaznamenal. I pro diváka je novější systém MI-2100 relativně příjemnější zvláště díky novému typu jednorázových brýlí, jež jsou lehčí a pohodlnější, jak původně dodávané.

Pro provozovatele kin systém přináší jednoznačné úspory jak na straně brýlí, tak i úspory na provoz výbojky, která nepotřebuje tak vysoký výkon, neboť ztrátovost světla na filtrech není tak markantní. Nevýhodou je nutnost investice do kvalitního postříbřeného plátna typu SilverScreen, která se může pohybovat v řádu až stovek tisíců korun.

To v případě, že bychom potřebovali na daném sále promítat i klasická 2-D představení, která nemůžeme promítat na plátno postříbřené kvůli takzvaným „hot spots“. Ty vznikají v místech nejjasnějšího obrazu a laicky bych je přirovnal kupříkladu k rozzářeným dopravním značkám s reflexní vrstvou, když je v noci osvětlí světlomet automobilu. Pro tyto účely je nutné v kinosále instalovat stahovací stříbrné plátno před klasické matové. Systém je tak výhodný zejména pro multiplexy, které mohou vyčlenit jeden sál pouze na digitální projekce ve 3-D stereoskopickém zobrazení.(2, obrázek tamtéž)

3.4.3 Real-D pro Sony CineAlta 4K

Systém Real-D vychází ze stejného principu, jako MasterImage, tedy z oddělení obrazů pro levé a pravé oko pomocí kruhové polarizace obrazu. Obraz ale nevzniká střídavým promítáním

obrazu pro levé a pravé oko, ale samostatnou projekcí dvou obrazů o rozlišení 2K na SXRD čípech s rozlišením 4K. Divák má k dispozici ještě pohodlnější brýle, jak



v případě systému MasterImage MI-2100. Na druhou stranu systém opět pochopitelně vyžaduje stříbrné plátno typu SilverScreen, tudíž v jednosálových kinech je nutné opět kombinovat bílé a stahovací stříbrné, aby byla zajištěna zpětná kompatibilita s 2-D projekcí. Pro diváka má systém nevýhodu i v deformaci formátu 1,85:1, který nepříjemně ořezává do formátu blízkého spíše Cinemascope.

Pro provozovatele kin je kromě vyšších nákladů na stříbrné plátno nepříjemná i skutečnost licenčních poplatků za užívání systému, která jej může znatelněji prodražit a ubírá ziskům vzniklým potřebou méně výkonné lampy, jelikož stejně, jako u MasterImage, má nižší ztrátovost světla.(2, obrázek tamtéž)

3.4.4 XpanD

System XpanD pracuje na principu aktivních brýlí, kdy se výsledný stereoskopický obraz vytváří až díky sklům v brýlích z tekutých krystalů, která střídavě zakrývají divákům oči tak, aby vždy viděli ten správný obraz. S kinoserverem a projektorem jsou brýle synchronizovány přes infračervené rozhraní. Pro diváky systém přináší velmi kvalitní výsledný obraz, kde nejsou problémy se zobrazením objektů při vyšší konvergenci. Pouze starší model 3-D brýlí, která využívají ještě některá kina, může při delším sledování začít tlačit díky větší hmotnosti do nosu.



Pro provozovatele kin systém XpanD na jedné straně přináší výhodu využití klasického bílého plátna, na stranu druhou nevýhodu v podobě vysoké ceny aktivních brýlí. Brýle jsou dosti náchylné na poškození, zejména aktivních filtrů, které jsou dosti křehké, a neinformovaný divák je může zničit obyčejným leštěním třeba o tričko.

Pořizovací cena za jedny brýle se pohybuje v řádu až několika tisíců korun a tudíž se vyplatí, v případě výběrového řízení, zadat větší množství brýlí za sníženou cenu, aby bylo k dispozici dostatek rezervních. XpanD pomáhá šetřit i elektrickou energií neboť ztrátovost světla je jen nepatrně horší, jak v případě dvou výše zmíněných polarizačních metod.(2, obrázek tamtéž)

3.5 Zdravotní aspekty 3D zobrazování

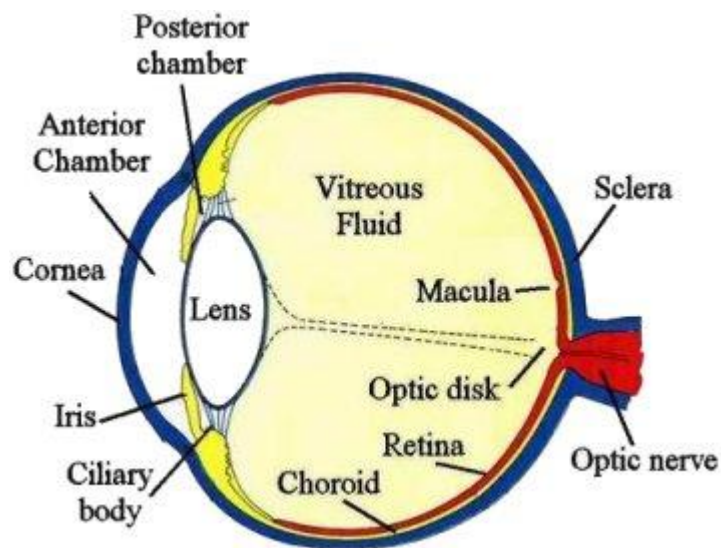
3D technologie totiž ještě nezaznamenala dostatečně široké rozšíření, aby bylo možné laicky posoudit, co se bude dít, pokud s ní budeme v každodenním styku. V tomto článku bychom proto chtěli naše čtenáře seznámit se závěry, ke kterým dospěly vědecké výzkumy, které se na tuto problematiku speciálně zaměřily.

O zdravotních rizicích spojených se sledováním 3D televize se začalo rozsáhle diskutovat v dubnu tohoto roku, kdy vydala společnost Samsung varování před možnými negativními reakcemi organismu. Hovořilo se v ní o tom, že se mohou vyskytnout změny ve zrakovém vnímání, mdloby, závratě, oční či svalové tiky, nutkání ke zvracení, záchvaty, křeče, potíže s orientací nebo dokonce ztráty vědomí. Ve zprávě bylo uvedeno, že 3D televizi nemají sledovat děti, těhotné ženy, staří lidé a také ti, kdo jsou pod vlivem alkoholu.

Tyto informace sice mohou znít trochu hroživě, je si však třeba uvědomit, že varování představovalo především pojistku proti soudním sporům, které by se případně mohly v budoucnosti objevit. V současnosti byla zveřejněna nová zpráva, která je již ve svých formulacích o poznání mírnější. (1)

3.5.1 3D TV - co se děje při sledování?

Při sledování 3D obrazu nastává nesoulad mezi skutečností a virtuálním světem. Každé z očí dostává mírně odlišné obrázky určité scény. Mozek poté tyto dva obrazy spojuje, a tak dochází k vytvoření iluze trojrozměrného prostoru. Je sice pravda, že skutečný svět je trojrozměrný – to, co nám nabízejí 3D televize, však není trojrozměrný svět, ale pouze jeho iluze. Steven Nussinowitz z Kalifornské univerzity v Los Angeles v této souvislosti zmiňuje dva příklady toho, jak může dojít k dezorientaci diváků.



Obrázek 9: Popis lidského oka (22)

Pokud se k nám ve skutečném světě přibližuje nějaký objekt, dochází k tomu, že se sbíhají optické osy obou očí (k tzv. konvergenci), a zároveň probíhá akomodace čoček, která umožňuje udržet objekt v zorném poli. Když se díváme na 3D filmy, jsme vystaveni klamnému pocitu, že se k nám objekt přibližuje. Dochází k "senzorickému konfliktu" – optické osy očí se začnou sbíhat, ale čočky jsou při tom zaměřeny na obrazovku či promítací plátno, které zůstává stále ve stejné vzdálenosti.

Druhým příkladem je situace, kdy se nám při sledování určité scény zdá, že se pohybujeme. Vestibulární systém, který je zodpovědný za vnímání polohy a pohybu, však při tom vydává mozku informaci, že zůstáváme stát na místě.

Z těchto důvodů může snadno docházet k pocitům nevolnosti. Mozek dostává určité signály o hloubce daného objektu, zároveň mu ale jiné potřebné signály scházejí. Když si poté po skončení filmu sundáme brýle, mozek bude mít pravděpodobně nadále problémy s vnímáním hloubky. U některých lidí může být doba potřebná pro návrat k normálnímu vnímání delší – vážná rizika to může představovat např. pro řidiče.

Pokud sledujeme 3D filmy jednou za měsíc v kině, nemělo by s největší pravděpodobností dojít k žádnému závažnému poškození nervové soustavy. Jiná situace ale může nastat tehdy, jestliže se 3D televize stane součástí našeho každodenního života.

Rizikovou skupinu zde představují především děti, které tráví několik hodin denně sledováním televize a hraním počítačových her. 3D televize v takovém případě může vážně narušit jejich schopnost prostorového vnímání a způsobit nevratné změny v jejich nervové soustavě.(22)

3.5.2 3D TV není pro každého

Dominick Maino z Americké optometrické asociace je v hodnocení rizik 3D televize o trochu zdrženlivější. I on však hovoří o tom, že může vyvolávat oční napětí, dvojitě vidění, bolesti hlavy či dokonce nutkání ke zvracení. Zdůrazňuje, že především u dětí jsou zapotřebí pravidelné oční prohlídky, které by zjišťovaly, zda nedochází k poškození binokulárního vidění. Rodiče by se měli zároveň snažit výrazně omezit čas, který děti tráví u 3D televize a hraním 3D počítačových her.

Konečně nesmíme zapomenout na to, že poměrně rozsáhlá skupina lidí si nebude moci zážitek z 3D televize vychutnat vůbec. Jde především o ty, kdo již v současnosti trpí nějakou oční chorobou. Ve Velké Británii byla podniknuta studie, do které byli zapojeni lidé trpící různými očními poruchami, jako jsou oční glaukom, degenerativní porucha žluté skvrny, albinismus nebo retinitis pigmentosa. Bylo sledováno, jakým způsobem vnímají 3D pořady ve srovnání s klasickým 2D, a také jak snášejí brýle pro sledování 3D obrazu ve spojení se svými běžnými dioptrickými pomůckami.

Bylo zjištěno, že s postupujícím časem stoupá procento těch, kteří nejsou schopni 3D obraz vnímat. Zatímco ráno se jednalo o 7 % lidí ze sledovaného vzorku, tak večer to bylo už 12 %. Dalších 20 % lidí potom uvádělo, že v nich sledování pohybových scén vyvolává pocity nevolnosti.

Někteří odborníci se proto domnívají, že technologie, kterou využívají současné 3D televize, nemá šanci dosáhnout skutečně masivního rozšíření. John Honeycutt z televizní společnosti Discovery Communications uvedl, že až u 20 % populace vyvolává sledování 3D televize negativní reakce, které souvisejí s funkcí vnitřního ucha. Jedná se především o lidi, kteří obtížněji snášejí cestování ve vlaku proti směru jízdy nebo kteří mají problémy se čtením knih v dopravních prostředcích. Pravidelné sledování současné technologie 3D televize by však mohlo mít negativní vliv na značně širší okruh diváků.(22)

3.5.3 Veletrh Opta 2011

Jedním z témat již 17. ročníku veletrhu Opta, který letos navštívilo přes 5 000 lidí, bylo i riziko sledování 3D filmů. „*Máte doma 3D televizi? Pokud ne, tak vám gratuluji, protože to je krok úplně špatným směrem,*“ uvedl na tiskové konferenci viceprezident Společenstva českých optiků a optometristů Ivan Vymyslický. Trojdimenzionální obraz sice představuje lákavou vidinu, jeho dlouhodobé sledování však dle slov Vymyslického při současné úrovni technologie nelze doporučit.

„*Film je vždy nafilmován pod určitou vzdáleností mezi středy zornic, která je u každého jiná. Sledování filmu proto nutí diváka šilhat dovnitř nebo ven, což může vést ke spoustě různých potíží,*“ varuje. K těm nejčastějším patří únava očí, bolest hlavy či malátnost. Za jednu z nejčastěji uváděných překážek bránící masovému rozmachu 3D obrazu je považována nutnost mít nasazené speciální brýle. Ty dle slov optiků není možné individuálně přizpůsobit, takže divákům zkrátka nesedí.

V současnosti používaný princip fungování 3D televizorů s rychlým střídáním obrazů pro levé a pravé oko je obzvlášť namáhavý pro lidi s neurologickými problémy, epileptiky apod. Dlouhodobé sledování 3D obrazu navíc negativně ovlivňuje odhad vzdálenosti. Největší nebezpečí přitom při pravidelném sledování 3D filmů hrozí malým dětem, jejichž zrak ještě není zcela vyvinut.

Před sledováním 3D obrazu již v minulosti varovala například společnost Sony a rozšířila podmínky užití svých zařízení o bod, který má uživatele před nežádoucími účinky varovat.(22)

3.6 WebGL



Web Graphics Library je technologie, která umožňuje implementovat hardwarově akcelerovanou 3D grafiku do libovolného kompatibilního prohlížeče a to bez nutnosti instalace dalšího software.(30, obrázek tamtéž)

3.6.1 Historie

Duchovním otcem implementace hardwarově akcelerované 3D grafiky je Vladimír Vukićević ze společnosti Mozilla, který v roce 2006 poprvé představil předchůdce WebGL – systém Canvas 3D.

Již v roce 2007 představila Mozilla i Opera vlastní implementace Canvas 3D do svých prohlížečů.

Rok 2009 přinesl spojení společností Apple, Google, Mozilla, Opera a neziskové společnosti Khronos, které společně, pod vedením Vladimíra Vukićeviče, vytvořily WebGL Working Group.

Výsledkem spolupráce těchto společností bylo v březnu 2011 vydání WebGL verze 1.0 a téměř okamžitá implementace této technologie do webových prohlížečů Google Chrome, Mozilla Firefox 4, Safari a Opera.

WebGL je po update firmware dostupné i pro telefony s mobilními Windows a s Google Android 2.2.(30)

3.6.2 Použití

Technologie WebGL je naprosto klíčová pro přechod elektronických obchodů z 2D do trojrozměrného zobrazení.

3D obchod s kvalitní grafikou vyžaduje vysoký výpočetní výkon počítače, který by žádný dnešní procesor nebyl schopen zvládnout bez kooperace s procesorem na grafické kartě, který je pro vykreslování vhodnější. Zde vstupuje WebGL, který pomocí OpenGL ES 2.0 deleguje vykreslování grafiky na grafický procesor. Díky tomuto řešení se pro 3D obchody stává omezujícím faktorem rychlost připojení k internetu. Pokud by rychlost přenosu dat byla nízká a načítání stránek trojrozměrného obchodu by zabíralo neúměrně dlouhou dobu, nedalo by se počítat s vysokou oblibou u uživatelů – potenciálních zákazníků.

Teoreticky je možné při využití dnešních nejnovějších, ale již poměrně běžně dostupných, technologií dosáhnout obrazové kvality elektronického obchodu na úrovni 3D her. Navíc pro WebGL není problémem ani použití aktivní nebo pasivní technologie

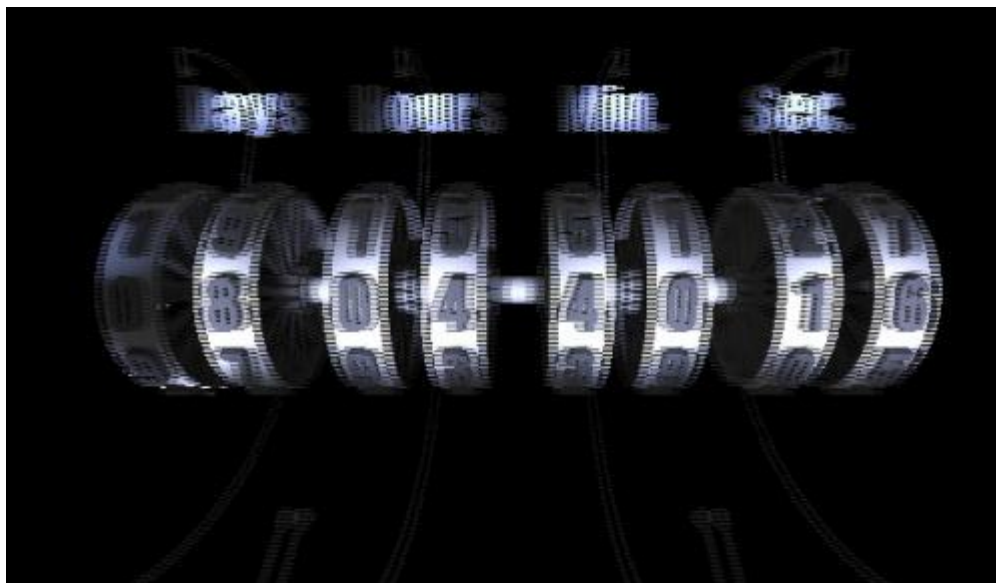
3D zobrazování, což ukazuje na orientaci této technologie právě pro využití v kombinaci s trojrozměrnými brýlemi.(30)

Na obrázku číslo 10 je screenshot ze hry vyvinuté společností Mozilla, založené na WebGL, běžící přímo v internetovém prohlížeči.

Obrázek 11 ukazuje trojrozměrné hodiny, které jsou s použitím pasivních brýlí krásně plastické.



Obrázek 10: Screenshot z web hry založené na WebGL (30)



Obrázek 11: 3D hodiny (30)

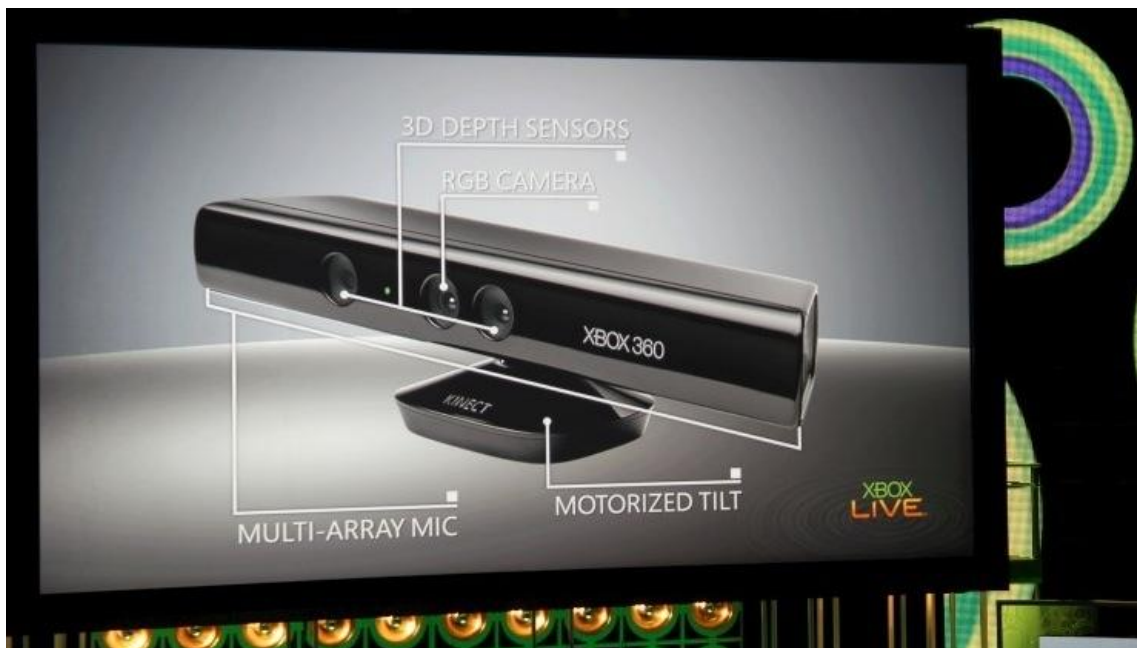
3.7 Kinect

4. listopadu 2011 společnost Microsoft vydala dlouho očekávanou novinku pro svou herní konzoli.

Jedná se o herní zařízení, které hráči umožňuje ovládání her přímo svým tělem a nepotřebuje tím pádem žádné přídavné ovládací zařízení.

Dosud používaná konkurenční zařízení byla bezdrátová, ale uživatel musel v ruce držet poměrně velký ovladač velikosti dálkového ovládání k televizi. Zařízení Kinect tuto nutnost zcela odbourává a nabízí uživateli nebývalou volnost pohybu v prostoru.

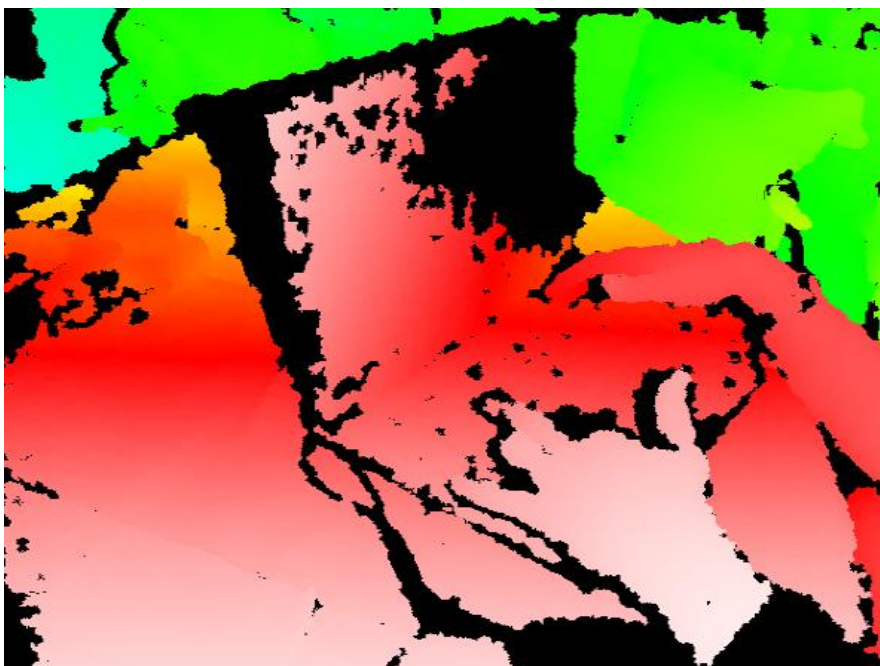
Principem funkce zařízení je snímání scény několika kamerami. Zařízení obsahuje kameru pro měření vzdálenosti Izraelské společnosti PrimeSense, která v reálném čase zaznamenává infračervené odrazy od objektů před kamerou. Tyto odrazy jsou v kombinaci s RGB kamerou a speciálním prostorovým mikrofonem rekonstruovány tak, že přístroj dokonale zaznamenává pohyby těla, jedná se o tzv. 3D motion capture. Dále systém umí zpracovávat obraz tváře a reagovat na různé grimasy.



Obrázek 12: Zařízení Kinect (16)



Obrázek 13: Infračervený obrázek snímané scény (16)



Obrázek 14: Hloubka obrazu je reprezentována barvami – bílá nejbližší, modrá nejvzálenější objekt (16)

Zařízení vysílá neustále infračervené světlo, které je zachycováno černobílým CMOS senzorem. Tento senzor zaznamenává scénu trojrozměrně za jakýchkoliv světelných podmínek.

Dle informací od prodejců je Kinect schopen současně zaznamenávat až 6 lidí v záběru kamery včetně dvou hráčů, u kterých zkoumá pohyby 20 kloubů. PrimeSense dokonce uvádí, že skutečný počet sledovaných lidí je omezen pouze šířkou záběru kamery.

Pracovní dosah zařízení je od 1,2 m do 3,5 m, což při šířce záběru vytváří herní plochu o velikosti 6 m².

Ihned po vydání se objevily společnosti, které nabízely finanční odměnu programátorovi, který jako první vytvoří tzv. opensource ovladač, aby pro Kineck mohly vznikat amatérské hry a zařízení mohlo být využíváno i na PC.

Spojením firem Asus a PrimeSense vznikl pracovní tým, který se zavázal v druhé polovině roku 2011 vydat zařízení podobné Kinectu, ale výhradně pro potřeby uživatelů PC.(16)

Společnost Microsoft se zařízením Kinect vytvořila světový rekord, který je zapsaný v Guinnessově knize rekordů, v prodeji jakéhokoli elektronického zařízení, když za prvních 60 dní prodeje prodala 8 000 000 kusů tohoto zařízení.(26)

Recenze Kinectu vesměs chválí použitou technologii jako průkopnickou, často ale dodávají, že nastávají potíže se zpožděním čtení pohybů, což by ale mohlo vyřešit zlepšení softwarového vybavení zařízení.(5)

4 Návrh řešení

4.1 Blender

Blender je software pro 3D modelování, animaci, tvorbu her, rendering a přehrávání. Je to program, který je založený na grafické knihovně OpenGL. Díky tomu je dostupný pro velké množství operačních systémů, jako je Windows, Linux, Irix nebo Mac OS X. Další výhodou Blenderu je, že je k dispozici zcela zdarma, a to včetně zdrojových kódů.

Existuje samozřejmě i spousta jiných programů, které mají stejné či podobné zaměření jako Blender a disponují i lepšími nástroji. Jmenujme například 3D Studio Max, Maya, Lightwave, Cinema 4D nebo Rhinoceros. Tyto programy se řadí mezi profesionální a lze v nich provádět pro laika až neuvěřitelné věci, které pak lze vidět v multimediálních prezentacích, počítačových hrách či filmových scénách. Již několik let se produkují filmy, které byly kompletně vytvořeny na počítačích s využitím výše jmenovaných programů. Takto vytvořených filmů stále více narůstá a i v běžně vytvářených filmech se scény a efekty vytvořené těmito programy používají čím dál tím častěji. Důvodem nejsou jen možnosti současných počítačů, ale i ekonomické hledisko. Dá se tak ušetřit nemálo finančních prostředků.

Není možné přesně říci, který ze jmenovaných programů je nejlepší. Každý z nich má nějaké charakteristické vlastnosti – přednosti a nedostatky. Obecně se dá říci, že je celkem jedno, který program 3D grafik používá, důležité je naučit se zvolený program dokonale používat a postupem času případně přejít na ten, který se „zdá lepší“.

Důvodů, proč je Blender použit pro realizaci 3D v této práci je několik. Prvním a hlavním je jistě to, že je zdarma. Jedna licence výše zmíněných komerčních programů obvykle stojí desítky až stovky tisíc korun (*3D Studio Max 9* vč. DPH 112 tis. Kč). Blender oproti tomu otevírá oblast 3D grafiky i pro začátečníky, aniž by museli investovat finanční prostředky.

Dalším důvodem je jeho velikost a hardwarová nenáročnost. Velikost instalačního souboru aktuální verze 2.48a pro Windows je 9,1 MB. Vzhledem

k možnostem, kterými tento program disponuje se to zdá neuvěřitelné. Jeho rychlost a výkonnost přirozeně závisí na konfiguraci počítače. Blender nicméně funguje i na staříčké 3Dfx Banshee ve Windows 98.(19)

Stejně jako u jiných programů, je i Blender neustále vyvíjen a pravidelně vznikají nové verze, ve kterých jsou implementované nové funkce a rozšířené možnosti. Díky tomu a více než desetileté historii vývoje jeho popularita neustále narůstá a pracuje s ním obrovské množství počítačových uživatelů. To dokládají i údaje od tvůrců tohoto programu, kteří na vývojářských www stránkách uvedli, že za prvních 8 měsíců roku 2005 měli více než 2 miliony downloadů.(6)

Je třeba poznamenat, že Blender má i stinné stránky, mezi které jistě patří uživatelské prostředí. Zvláště uživatelé Windows jsou hned na začátku nepříjemně překvapeni, když se hledaná položka Help neskrývá pod notoricky známým a vžitým tlačítkem F1 – tím se otevírá nový projekt.

4.2 Tvorba 3D modelu v Blenderu

V této kapitole bude představen postup vytvoření modelu elektronického obchodu.

Cílem 3D obchodu je přinést zákazníkovi iluzi nakupování tak, jak ho dobře zná a vyvolat v něm dojem, že je fyzicky se zbožím v kontaktu.

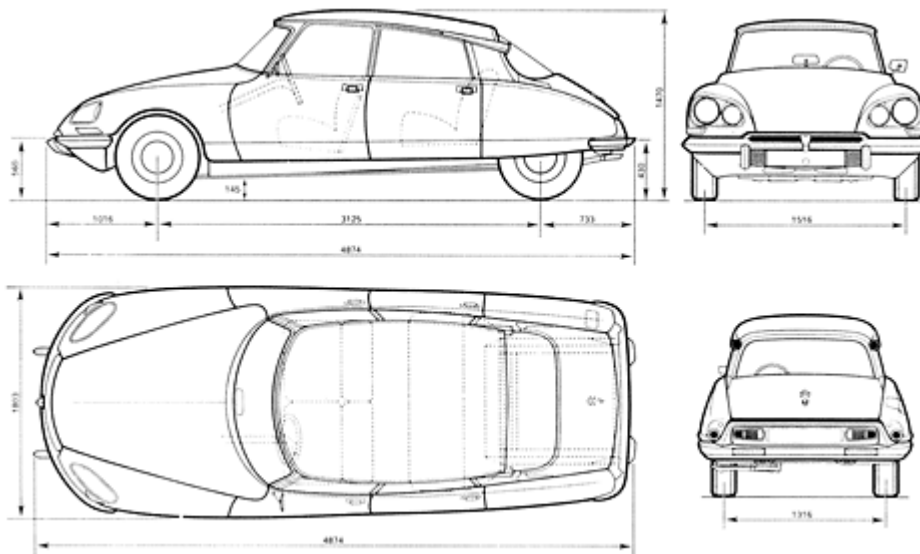
Je proto nutné nasimulovat prostředí obchodu realisticky, věrně do něj umístit zboží, informace o zboží a mnohé další objekty.

Většina částí modelu obchodu se skládá ze čtverců, které jsou potaženy texturami, naopak modely zboží jsou tvořeny především křivkami.

4.2.1 Blueprint

Stejně jako při malování portrétu malířem i 3D grafik potřebuje na vytvoření dokonalého modelu předlohu požadovaného vzhledu. K tomu slouží takzvané blueprints – jde o monochromatickou dokumentaci, technický výkres či ukázkou designu. Jeho významnost spočívá v tom, že se skládá ze čtyř pohledů na tu samou věc, po rozdělení na jednotlivé části a vsazení do animačního programu nám tedy umožňuje přesně tvarovat jednotlivé části objektu. Důležitou vlastností blueprintu je také to, že jsou v něm vyznačeny rozměry objektu, například v milimetrech.

Častým problémem (až ve 40 % případů) je nesrovnalost v rozměrech jednotlivých pohledů, takže je třeba dobře vybírat zdroj a před použitím jednotlivé proporce porovnat.



Obrázek 15: Blueprint Citroen (29)

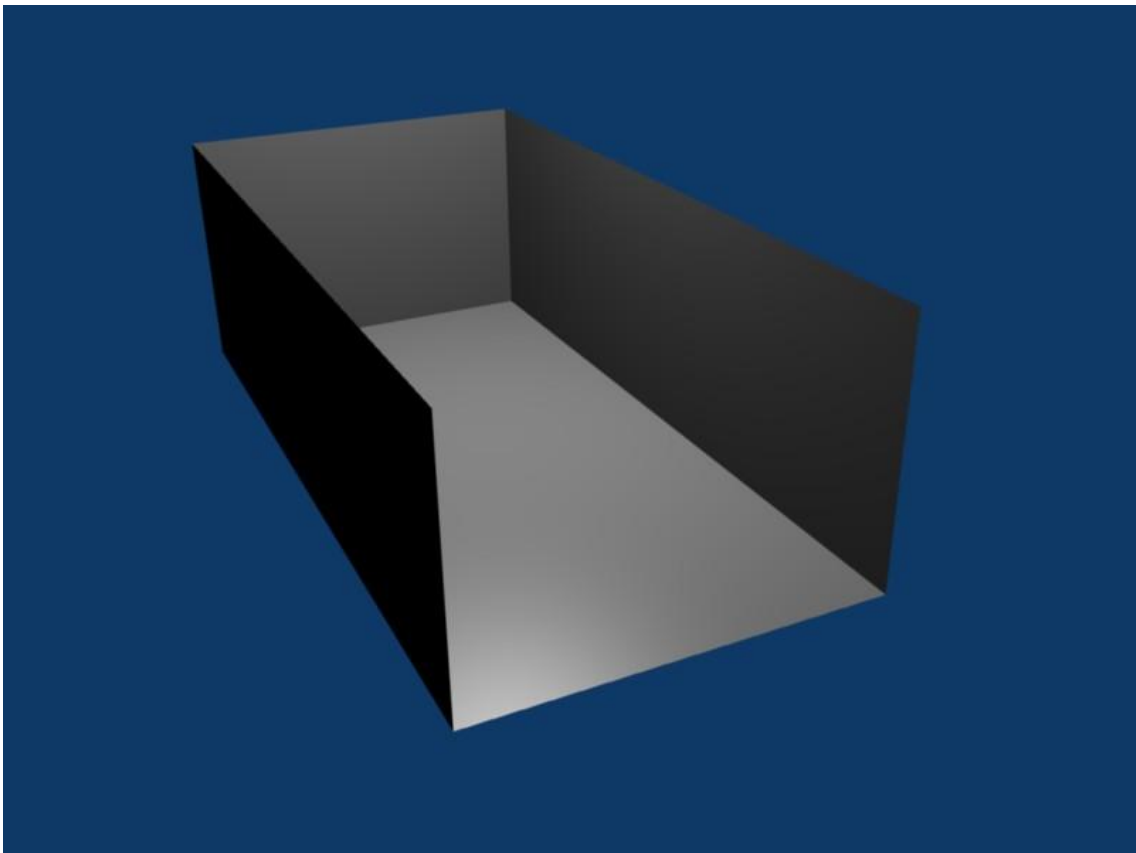
4.2.2 Model místnosti

Běžný tvar místnosti je kvádr. V podstatě to tedy znamená, že se skládá ze čtyř stěn, podlahy a stropu. Převáděno do jazyka grafiky to znamená, že místnost je

definována osmi body, tzv. vertexy, dvanácti hranami, šesti plochami – mash objekty, které tvoří viditelné stěny místnosti.

Pro účely názorné ukázky a lepšího přehledu v modelu budou odstraněny plochy stropu a jedné ze stěn tak, aby bylo možné pořídít screenshoty z větší vzdálenosti a lepšího úhlu.

Základní nastavení mash objektů nedefinuje žádné jejich vlastnosti, kromě barvy a průhlednosti. Veškeré další vlastnosti budou mash objektům přidány později tak, jak budeme postupovat v modelaci 3D obchodu.



Obrázek 16: Model místnosti 3D elektronického obchodu

4.2.3 Materiálové nastavení

Materiálové nastavení v Blenderu má obrovské množství možností a ovládnutí práce s tímto prvkem programu umožňuje dělat opravdová kouzla.

Každý model, který je v Blenderu vytvořen, je šedý, nereflexní a neprůhledný. Aby bylo dosaženo reálného vzhledu objektů, je nutno přiřadit každému vlastnosti, které odpovídají jeho reálnému vzhledu. Není například možné, aby sklo nebylo průhledné, aby se v ocelových částech neodráželo okolí nebo aby objekty nevrhaly stíny. Všechny tyto detaily velice ovlivňují výsledný efekt animace a jejich nastavování se věnují celé týmy odborníků, kteří získání potřebných textur věnují velice dlouhou dobu a veliké úsilí.

Jednou z možností, jak získat textury na kvalitní potažení objektů, je jejich koupě od specializovaných firem na internetu. Například server www.cgtextures.com nabízí desítky tisíc fotografií, které jsou svou kvalitou vhodné k použití v 3D grafice. Na tomto serveru je možné najít snad všechny fotografie, respektive fotografie všeho, co by grafik mohl potřebovat.

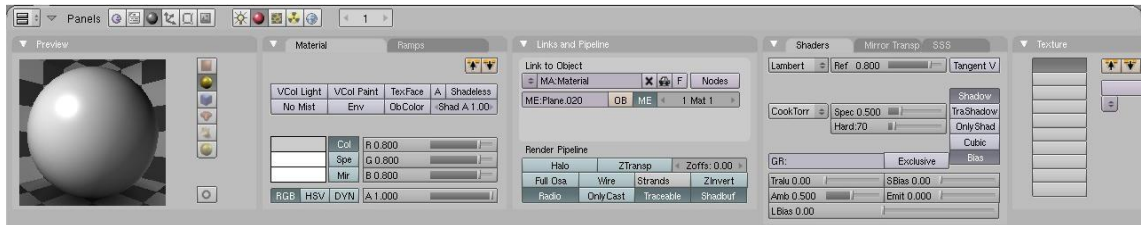
Každá fotogalerie má ještě několik podkategorií, jako je tomu například u kategorie „Cihla“, kde bychom našli podkategorie „Podlaha“, „Zdivo malé“, „Zdivo velké“, „Zdivo oblé“, „Zdivo moderní“. Obdobné rozvětvení je u všech kategorií obrázků, hledání vhodné textury je proto velmi snadné a přehledné.

Pokud grafik potřebuje pro svou práci texturu, kterou na podobných serverech nenajde, jako například konkrétní obraz budovy, přírody, předmětu nebo člověka, musí takovouto fotografii pořídit sám, nebo si ji vytvořit v grafickém programu, jako je například Photoshop nebo Gimp. Tyto programy jsou grafiky využívány v podstatě vždy, jak bude názorně ukázáno dále.

Na obrázku 17 je zobrazen panel pro nastavení materiálu a textur tak, jak se používá v Blenderu.

V levé části se nachází okno s objektem, který názorně ukazuje, jak bude nanášená textura a materiál vypadat při renderingu objektu. Toto okno neukazuje animovaný objekt, ale jen určité tvary, na které jsou textury pro představu nanášeny. Jedná se o čtverec – plochu, kouli, kvádr, hlavu opice Suzanne (jedná se o jakéhosi maskota Blenderu), vlasy či trávu.

Aby si grafik mohl udělat představu o tom, jak bude výsledná textura na objektu vypadat, musí si vybrat tvar, který je modelovanému objektu nejpodobnější. I když se tento způsob může zdát nepřehledný, je poměrně dostatečný a hlavně zbytečně nezatěžuje procesor zbytečným nanášením textury na složitý modelovaný objekt.



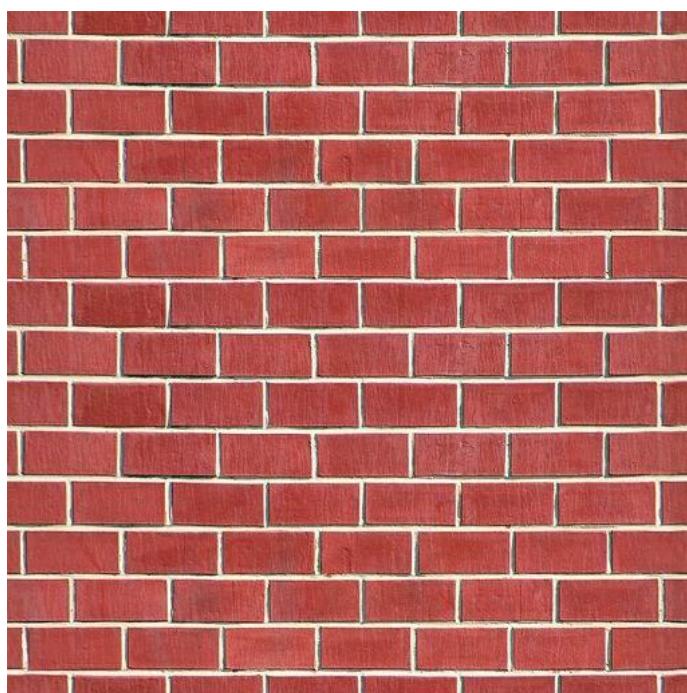
Obrázek 17: Panel pro nastavení materiálu a textur

Vpravo od náhledového okna v panelu „Material“ je panel „Col“ pro nastavení barvy objektu, kterou je možno zadávat jak z palety, tak třemi posuvníky R, G, B. Panely „Spe“ a „Mir“ jsou určeny pro nastavení barvy odlesků okolí v modelovaném objektu.

Významnými položkami jsou „Shaders“ (stínování) a „Mirror Transp“ (zrcadlení, odrazy), které umožňují přidávat objektům reálné světelné vlastnosti a jsou proto důležité pro výsledný dobrý efekt celé scény.

Nyní nastavíme textury pro podlahu a pro zdi obchodu. Jako první je samozřejmě nutné najít si požadovaný obrázek, který má být jako textura použitý.

Pro tento 3D obchod byla záměrně vybrána textura kamenné zdi, proto, abychom na ní mohli demonstrovat plastičnost výsledného efektu.

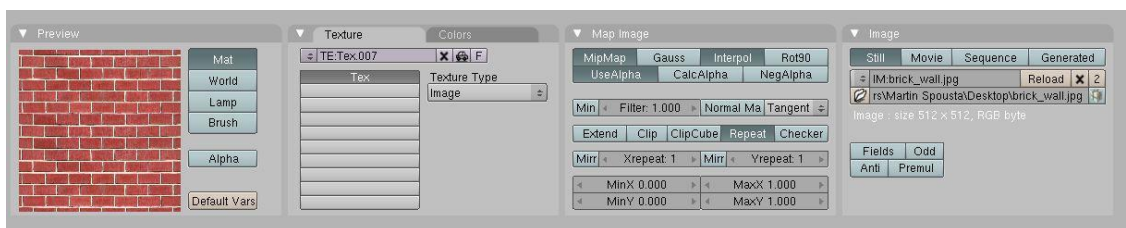


Obrázek 18: Textura použitá na stěny (31)



Obrázek 19: Textura použitá pro podlahu (31)

Nastavení textur v Blenderu je velmi snadné, zvláště pokud se jedná o nanášení textury na plochu a ne na trojrozměrný model.



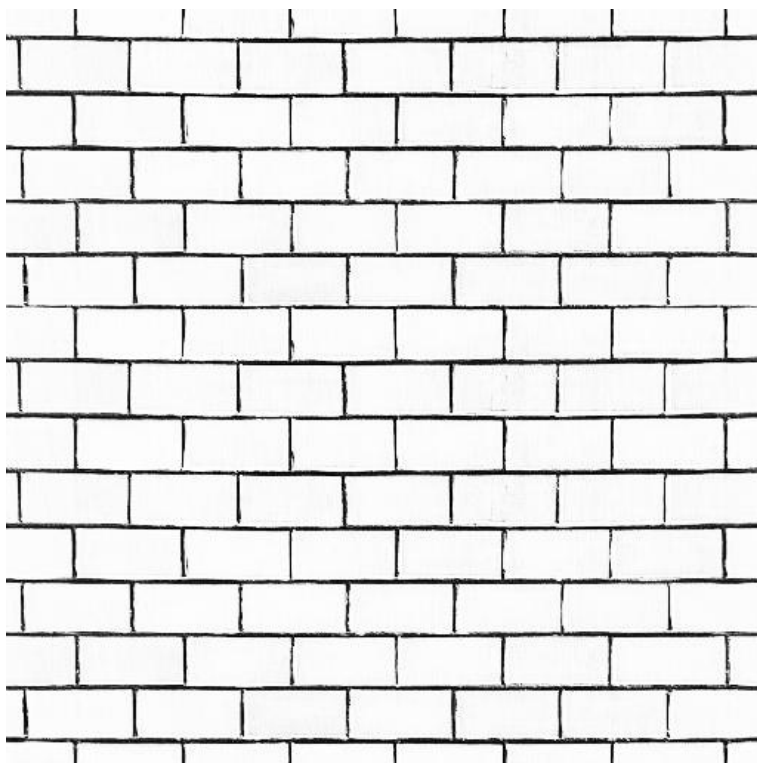
Obrázek 20: Přiřazení textury zdi

Na obrázku 20 je ukázka nastavení textur. Za povšimnutí stojí levá část, kde je preview textury, která je zvolená a pravá část, ve které se nachází pole pro samotný výběr obrázku, který bude jako textura sloužit.

Blender má do značné míry automatizované činnosti nanášení textur, takže grafik není nucen měnit velikost obrázku textury tak, aby se přesně shodovala s velikostí objektu, na který je textura nanášena. Pokud se totiž rozměry neshodují, Blender obrázek upraví sám a to buď tak, že ho roztáhne nebo naopak smrskne na rozměry objektu.

Aby bylo u zdí dosaženo plastičnosti, používá Blender zajímavý systém, který je poměrně jednoduchý, ale velice efektivní. Jedná se o takzvané bump mapy.

Vycházet budeme z obrázku, který je použitý pro texturu, ale v Photoshopu odstraníme barvy, čímž dostaneme černobílý obrázek. Upravíme kontrast, jas a zvýrazníme okraje tak, aby rozdíly mezi bílou černou, která reprezentuje místa s větší hloubkou (malta mezi cihlami), a černou bílou, kterou mají místa vystouplá, byly co nejostřejší.



Obrázek 21: Upravená textura pro plastickou zed'

Tento upravený obrázek je následně použitý jako druhá textura na stejném objektu. Vypnutím barev této textury získáme požadovaný „tahák“ pro plastičnost, kterou navíc ještě znásobíme zapnutím funkce Nor.

Výsledkem tohoto dvojího nanesení textur je reálnější vzhled cihlové zdi s malou výpočetní náročností pro počítače a je tím velice vhodné pro použití v internetovém 3D obchodě.

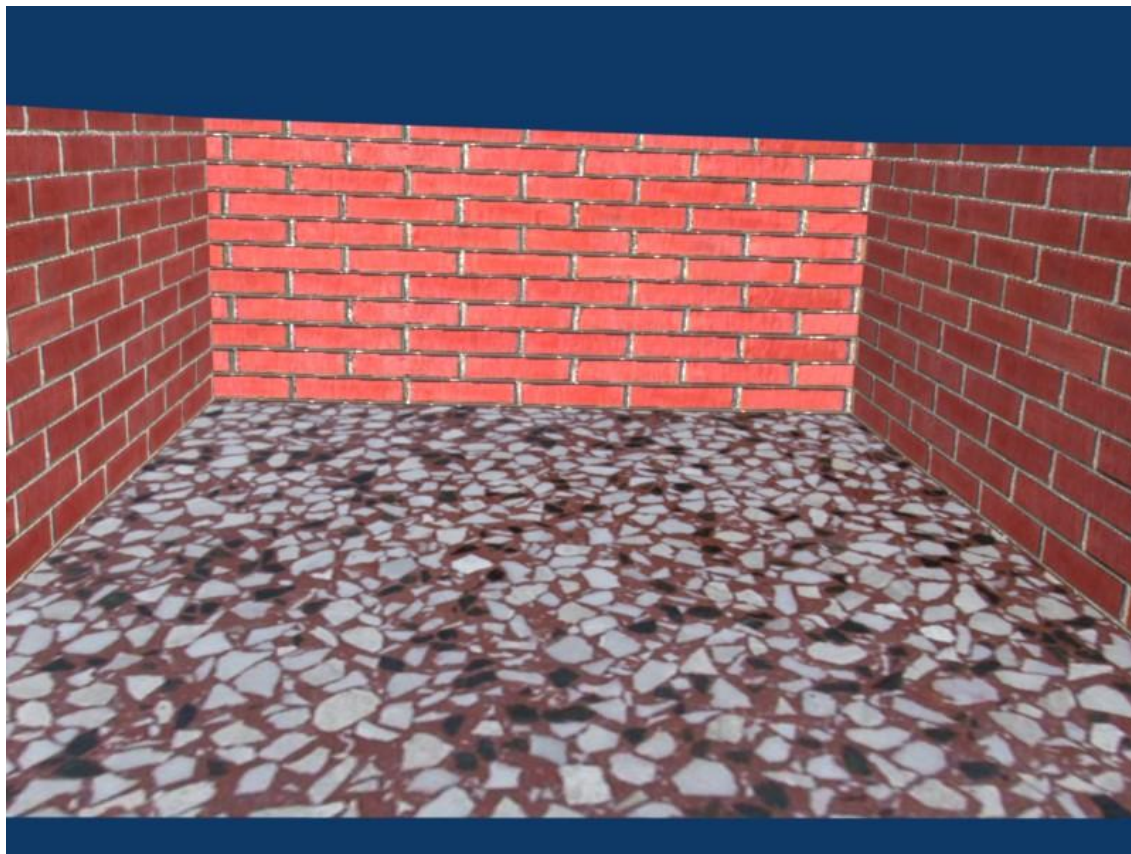


Obrázek 22: Cihlová zeď s použitím dvou textur



Obrázek 23: Původní cihlová zeď

Po nanesení textur na stěny a na podlahu máme základní kostru internetového 3D obchodu.



Obrázek 24: Základní model 3D internetového obchodu

4.2.4 Model výstavní plochy

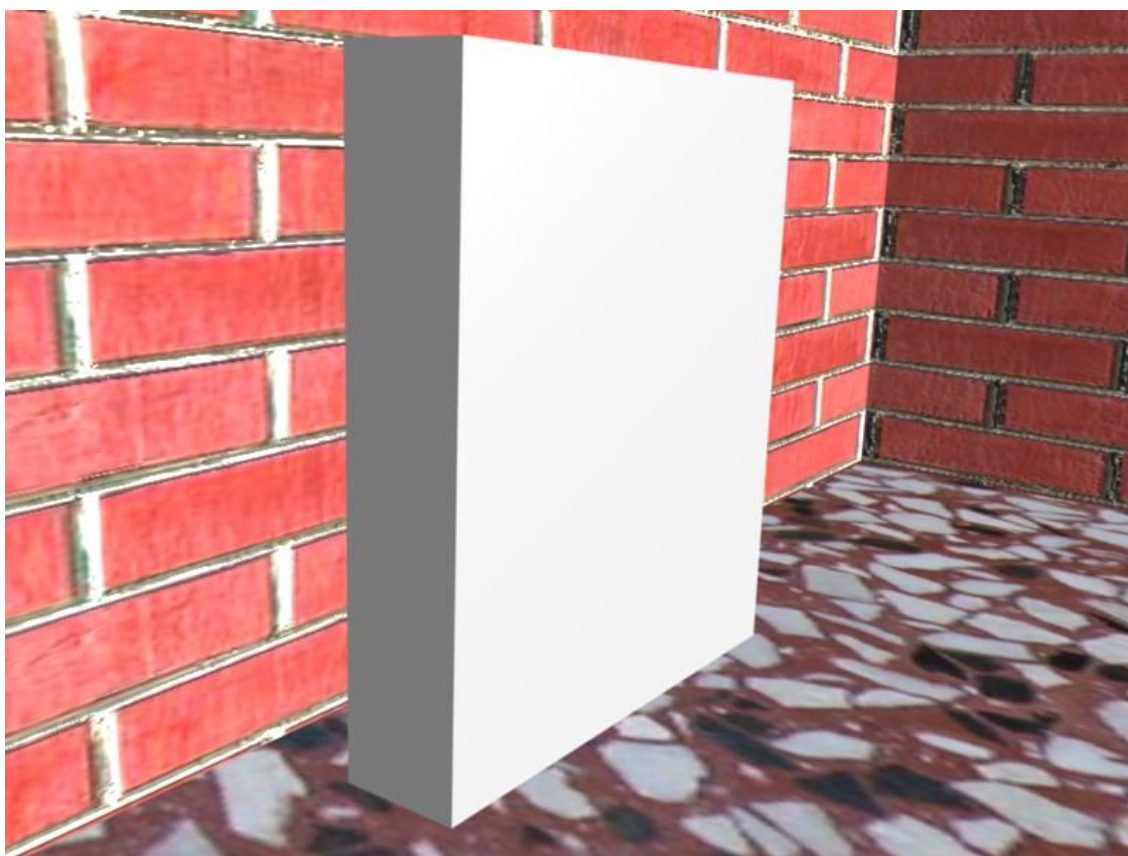
Pro vystavování nabízeného zboží je samozřejmě nutné mít adekvátní místo, kam bychom předmět prodeje mohli umístit. Musíme proto vytvořit model regálu či police, kam produkt umístíme.

Jako zboží do stojanu byly vybrány sklenice. Důvodem výběru pro demonstraci 3D obchodu se sklenicemi jsou vazby na společnost, která sklenice prodává, a její obchodní výsledky jsou v poslední době neuspokojivé. Cílem je tedy přinést touto

cestou zajímavé oživení stránek společnosti, výměnou za přístup k několika fotografiím sklenic, které budou v další části sloužit jako blueprints pro jejich modelaci v Blenderu.

Stojan na prezentaci skleniček by měl být designově zajímavý, ale nemusí odpovídat žádné reálné předloze, proto vytvoříme libovolný takový, který by splnil požadovanou funkci. Řekněme, že moderní trendy se dnes zaměřují na kombinaci skla a kovu, proto bude stojan pro naše potřeby tvořen právě těmito dvěma materiály. Ve skutečném světě by takovýto kovový stojan těžko mohl existovat, protože jeho hmotnost by určitě převyšovala únosnou mez. Pro účely grafiky a elektronického obchodu je ale tato skutečnost irelevantní.

Základním objektem, který pro stojan použijeme, bude kostka, protože je požadovanému tvaru kvádrů ze základních nabízených objektů Blenderu nejpodobnější. V editačním módu, kde vidíme objekt tak, jak ho tvoří jednotlivé vertexy, hrany a plochy, označíme vertexy jedné strany a jejich posunutím po ose x, vertexy horní strany roztáhneme po ose y nahoru, čímž dosáhneme tvaru kvádrů.

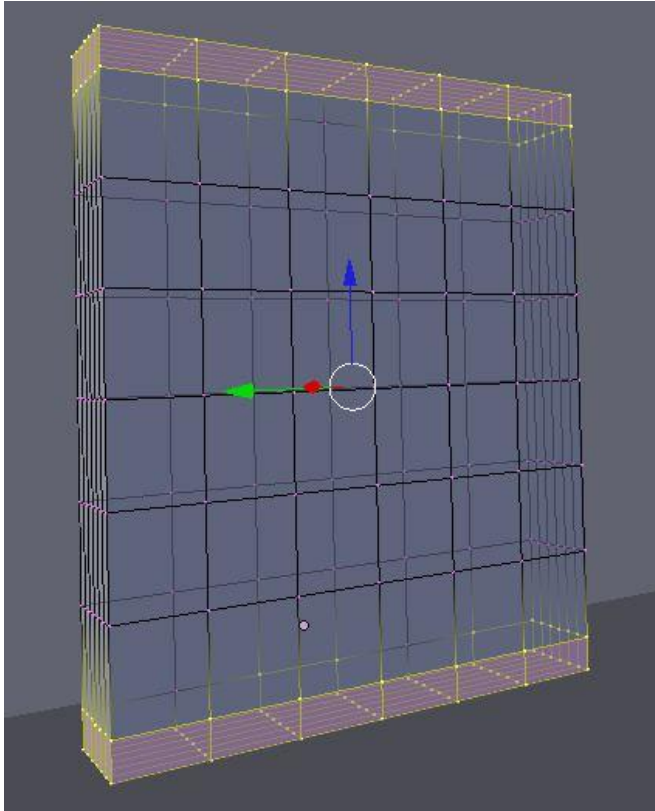


Obrázek 25: Základní tvar stojanu

Zatím je tento bez přiděleného materiálu a barvy, takže jeho vlastnosti jsou jen neprůhlednost a šedá barva, jak je vidět na obrázku 25.

Dalším krokem v modelaci je rozšíření počtu vertexů. To je nutné proto, abychom dosáhli více bodů, které vytvoří plochy, se kterými můžeme následně pracovat a tím modelovat stojan podle našich představ. Pro zvýšení počtu vertexů se používá funkce *subdivide*, která každou hranu rozdělí na několik částí tak, jako jsou na pravítku označeny centimetry a zároveň protilehlé body spojí do čtverců. Čtverec (nebo trojúhelník) je základní plochou v Blenderu a používání mnohoúhelníků se nedoporučuje, jelikož při renderaci často způsobují nemalé problémy a jejich odstranění v pozdějších fázích modelace je velice obtížné. Funkcí *subdivide* tedy rozdělíme model na celkem 256 vertexů. Větší počet vertexů znamená více informací, se kterými musí procesor pracovat a také větší velikost výsledného modelu, proto je dobré držet jejich počet na nejnižší možné úrovni.

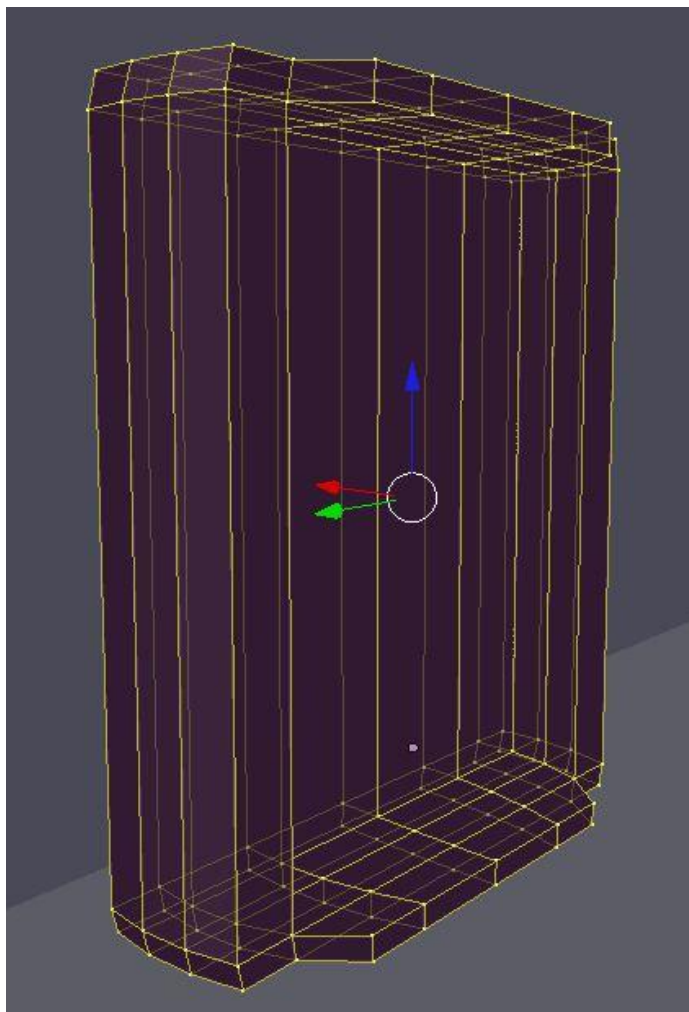
Nyní funkcí *extrude* zvýšíme počet vertexů na horní i spodní straně stojanu. Tato funkce v podstatě zkopíruje body na jejich stejné pozice, ale pokud je posuneme po ose z nahoře, respektive dolů, získáme nové plochy, se kterými můžeme manipulovat a měnit tvar modelu.



Obrázek 26: Stojan v editačním módu

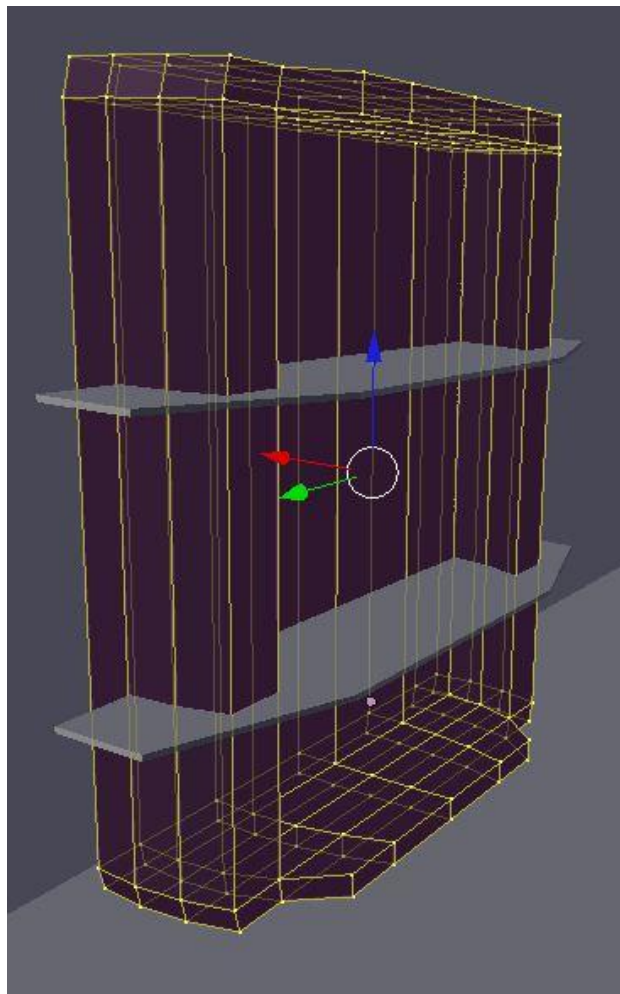
Na obrázku 26 je zachycený stojan v editačním módu Blenderu. Tento mód nám umožňuje vidět to, z čeho se objekt skládá. Vertexy jsou na vrcholu každé hrany a v každém křížení hran, plochy se nachází mezi hranami.

Nyní proběhne samotné modelování stojanu posuny vertexů. Vzhledem k tomu, že tato činnost zahrnuje doslova slovky posunů vertexů, extrudování, změny velikostí a mnoho dalších činností, nebude zde popsána detailně.



Obrázek 27: Upravený stojan v editačním módu

V editačním módu ještě stojanu přidáme plochy (police), na které se budou sklenice stavět. Základní tvar bude čtverec, který extrudujeme, a po ose z posuneme vertexy o několik milimetrů, čímž dostaneme ze čtverce prostorové těleso - kvádr. Kvádr musíme roztáhnout po ose x tak, aby byl minimálně tak široký, jako samotný stojan. Funkcí *subdivide* rozdělíme kvádr na tři díly a dva krajní díly zúžíme tak, aby police ve stojanu působila designově pěkně.

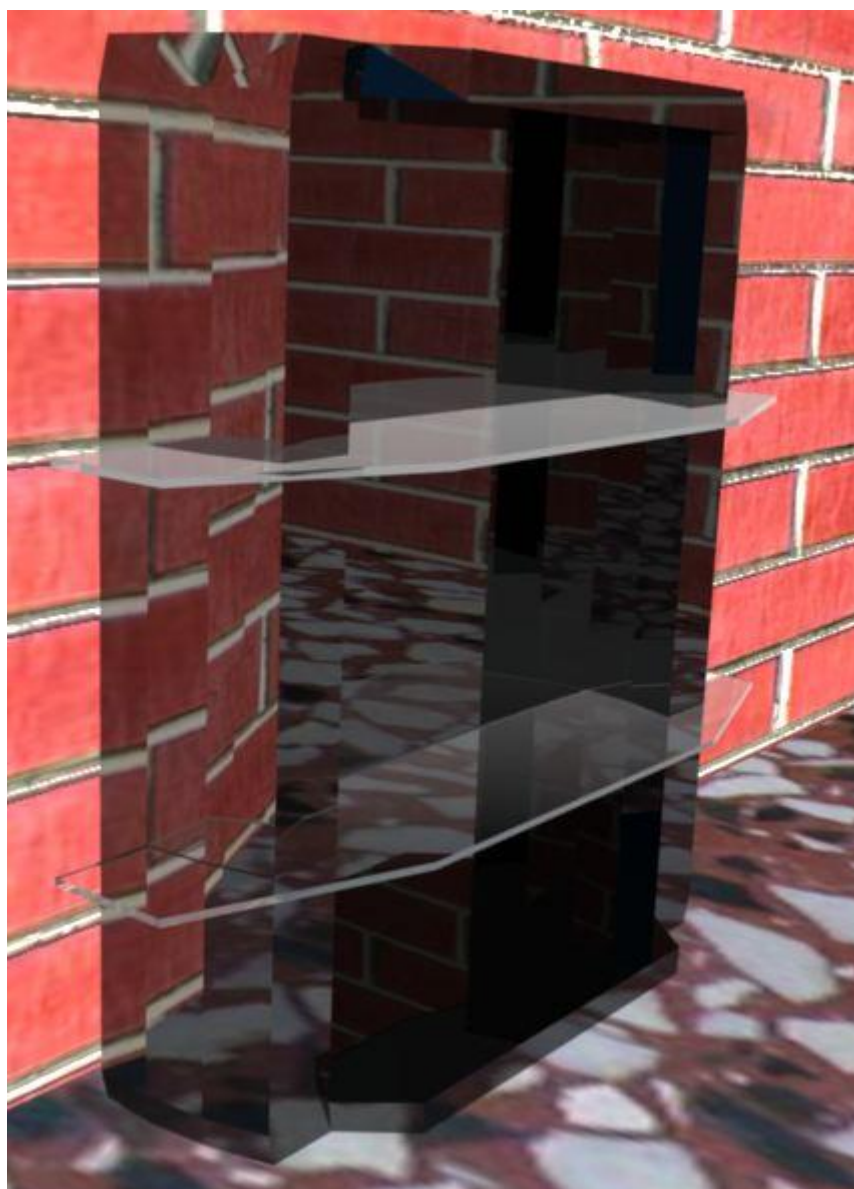


Obrázek 28: Kompletní stojan v editačním módu

Stojanu je potřeba zadat jeho vlastnosti, jako je barva a míra reflexnosti materiálu. Dle výše zmíněného bude povrch simulovat odrazy tak, jak by se odrážely od leštěného kovového povrchu.

Toto nastavení se provádí v záložce Mirror Transp aktivací tlačítka Ray Mirror a samozřejmě ještě množstvím jiných drobných nastavení, která upravují míru odrazů, jejich barvy, hrubost odrazů, kvalitu při renderaci a další.

Skleněné police jsou nastavovány podobně, oproti kovu mají ještě aktivovány vlastnost průhlednosti Ray Transp – opět doplněnou o mnohá drobná nastavení.



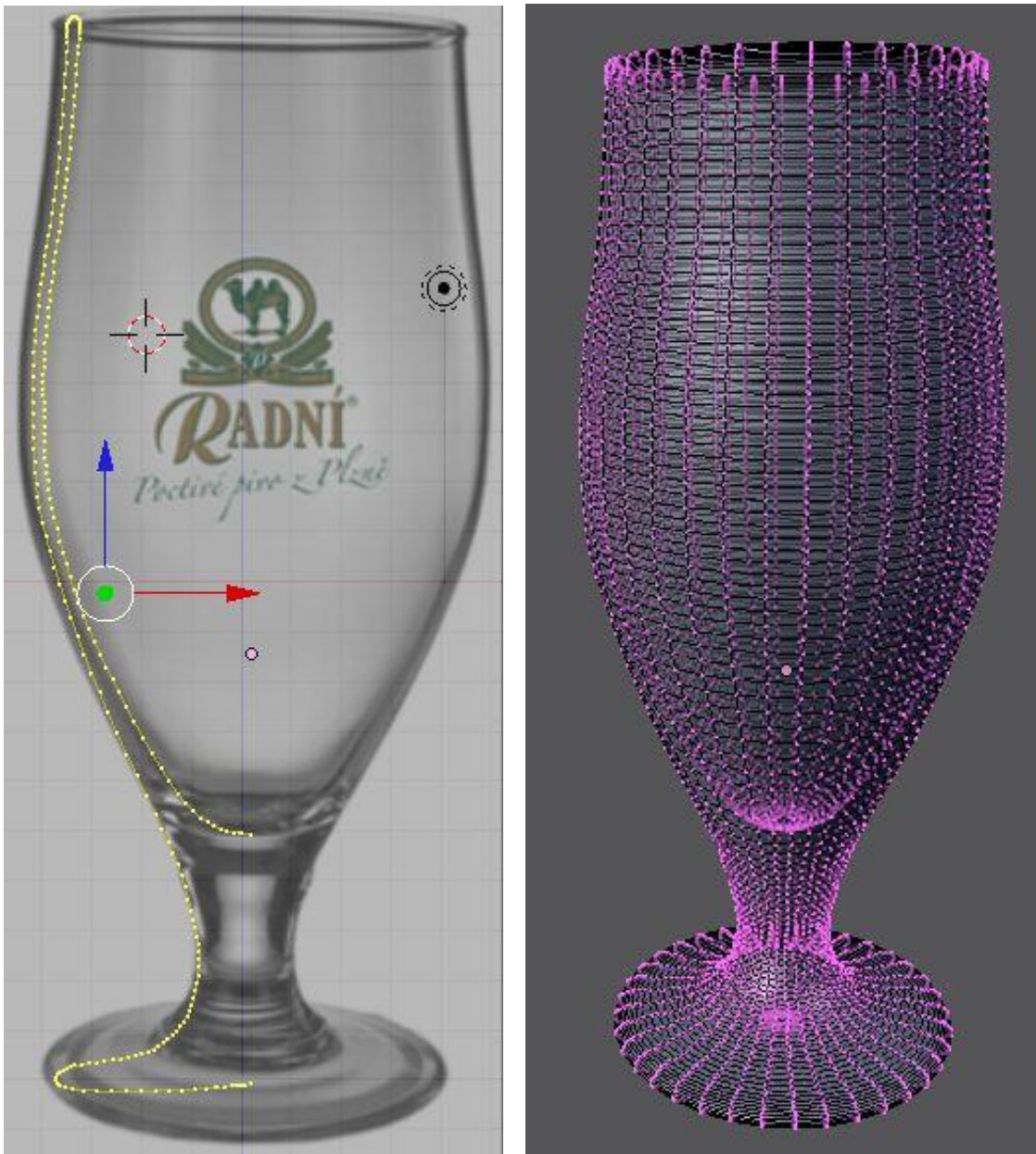
Obrázek 29: Kompletní renderovaný stojan

4.2.5 Modely sklenic

V nabídce firem, které obchodují se sklenicemi, jsou určitě stovky skleniček, nebo alespoň desítky. Pro demonstraci jejich modelování v Blenderu bude stačit jen několik kusů. Průřez nabídky mohou prezentovat tři skleničky, každá jiného typu –

tlustostěnná sklenice například na pivo, tenkostěnná sklenice na víno a tenká vysoká sklenice na šampaňské.

Začněme s pивní sklenicí. Do Blenderu si nahrajeme obrázek sklenice, výše zmíněný blueprint. Zde platí jedno omezení, respektive doporučení a to takové, že je zcela nezbytné blueprint sklenice umístit tak, aby osa z procházela přesným středem sklenice na obrázku.



Obrázek 30: Základní umístění vertexů a vertexy po použití funkce spin

Když je obrázek sklenice perfektně umístěn, začneme nanášet vertexy tak, aby jejich pozice odpovídala hranám skla. V podstatě tak jde o to, abychom vertexy vymezili prostor, kde má být pevný materiál – sklo.

Musíme dbát na to, aby poslední vertexy (nejblíže osy z), byly přesně na této ose. Toho lze dosáhnout tím, že jejich pozici vepíšeme přímo do pole, které udává pozici vertexu ve všech třech rozměrech.

Pro tvorbu modelu sklenice nyní použijeme funkci spin s hodnotou 360°. Aby se sklenice vytvořila správně, musíme se před použitím funkce spin dívat na připravovaný model shora, tzn. rovnoběžně s osou z.

Po namodelování tvaru sklenice je opět nutné nastavit její materiál tak, aby odpovídal reálnému sklu, můžeme například využít materiál, který jsme přiřazovali skleněným policím při modelování stojanu.



Obrázek 31: Srovnání předlohy pивní sklenice a jejího modelu v Blenderu

Stejným způsobem jsou modelovány zbylé dvě sklenice na šampaňské a na víno.



Obrázek 32: Srovnání předlohy sklenice na šampaňské a jejího modelu v Blenderu

Grafické podání sklenic je na vysoké úrovni a od reality téměř neodlišitelné, navíc má zákazník možnost si se sklenicí libovolně manipulovat, může ji otáčet, zvětšovat, v případě potisku si tento může na virtuální sklenici prohlédnout, pokud samozřejmě bude model doplněn o další modul, který dovolí nanášet na plochu modelu

obrázky nebo nápisy, které si zákazník přeje. I pokud by tento modul v elektronickém obchodu nefiguroval, není pro grafika problém jakýkoliv obrázek na sklenici věrně nanést během několika minut a zákazníkovi poskytnout nabídku.



Obrázek 33: Srovnání předlohy sklenice na víno a jejího modelu v Blenderu

4.2.6 Celkový model 3D obchodu

Spojením zvlášť namodelovaných částí do jediného modelu získáme požadovaný 3D elektronický obchod. Tato jedna namodelovaná místnost samozřejmě

tvoří jen část celého obchodu, ale pro demonstraci vzhledu, postupů a práce s takovýmto obchodem je dostačující.

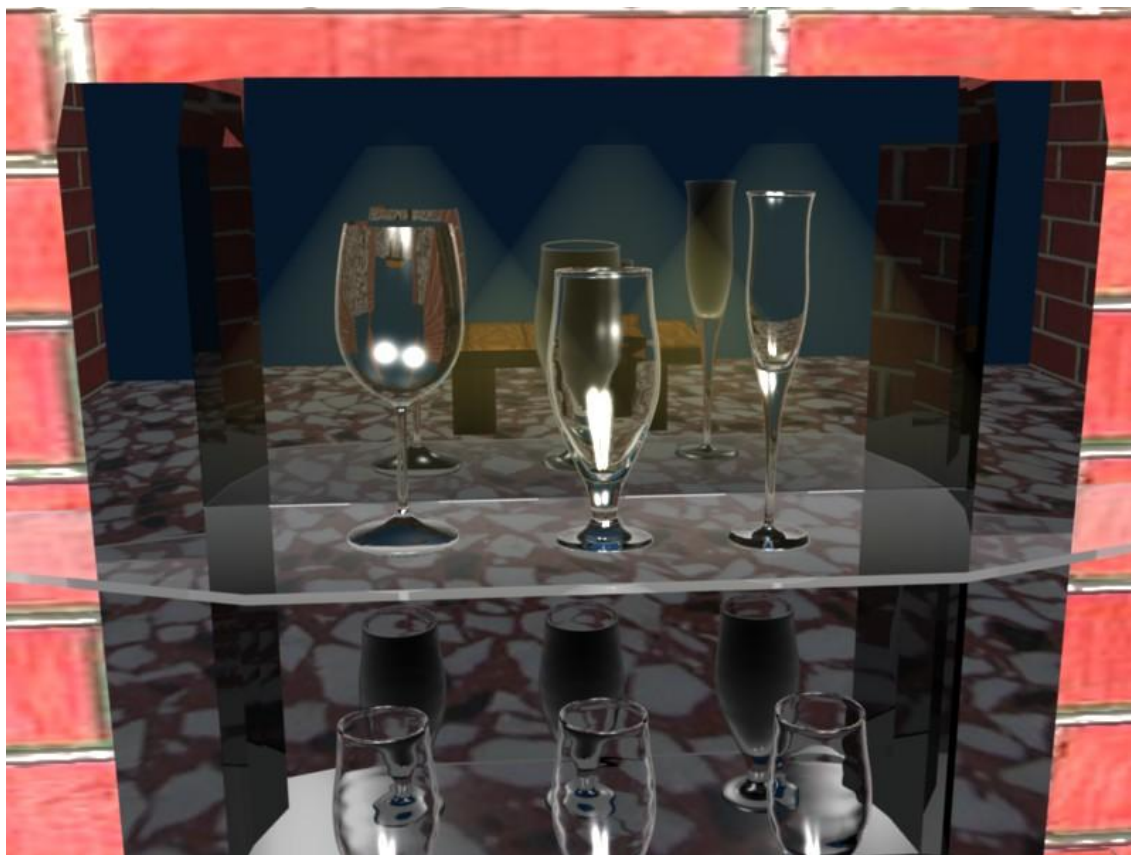


Obrázek 34: 3D elektronický obchod

Na obrázku 34 je vidět kompletní 3D elektronický obchod tak, jak by se dal využít pro komerční účely. V kovovém stojanu jsou vidět reálné odlesky okolní cihlové zdi a i odraz podlahy. Odrazy ve stojanu se samozřejmě mění podle toho, jak je natočena kamera, která scénu snímá. Ve sklenicích se odráží dvojice světla, které scénu osvětlují.

Na obrázku číslo 35 na následující straně je detailní záběr na horní řadu sklenic, která je nasvětlena žlutým světlem. Smyslem tohoto světla je napodobovat nasvícení bodovými světly, jako je například halogenové světlo, oblíbené pro osvětlování výrobků v prodejních regálech.

Při bližším pohledu je v odrazu na stojanu vidět dřevěný stůl, který se nachází za kamerou. Takový by například mohl sloužit na prezentaci vybraného kusu zboží.



Obrázek 35: Detail na stojan se sklenicemi

4.3 Ovládání 3D elektronického obchodu

Inovativní přístup k zobrazení nabídky může díky moderní technice podpořen i zcela originálním ovládním celého obchodu.

Společnost Microsoft na konci roku 2010 vydala novinku Kinect, přídatné zařízení k Xbox 360, která umožňuje uživatelům ovládat hry vlastním tělem. Pomocí kombinace pohybů a mluveného slova je možné ovládat i samotné zařízení a spouštět programy.

Tato technologie ze světa sci-fi je začátkem revoluce v ovládní počítačů. Dlouze používaná myš, poměrně nové dotykové obrazovky, to jsou technologie, které s trojrozměrným zobrazením moderních monitorů a televizí nemohou spolupracovat. Současné ovládací metody se spoléhají jen na dvojrozměrný pohyb.

Trojrozměrného obrazu, který je jen iluzí, se uživatel fyzicky dotknout nemůže, ale díky zařízení pro snímání pohybu těla může počítač propočtem zjistit, čeho se uživatel „dotýká“ a reagovat tak na jeho podměty. Není tedy problém si virtuální zboží uchopit, otáčet přibližovat či oddalovat pomocí pohybu rukou, zákazník by se mohl doslova procházet obchodem.

5 Závěr

Cílem práce bylo názorně ukázat, jak by mohl vypadat 3D elektronický obchod budoucnosti při využití technologií, které jsou dnes běžně dostupné, nebo se tyto technologie velice rychlým tempem rozšiřují. Důvodem, proč by firmy měly investovat do 3D elektronických obchodů, je posílení konkurenceschopnosti na vysoce konkurenčním poli internetového prodeje.

3D modely dnes využívají především automobilky a developéři, kteří jsou díky nim schopni zákazníkovi ukázat budoucí automobil nebo dům a to i s výhledem, jaký jeho nemovitost bude mít. Zákazník o zboží získá více informací a prodej se tím do značné míry zjednoduší.

Z informací, které jsou v této práci obsaženy, je jasně patrné, že 3D technologie nezadržitelně proniká do obyčejného života a očekávání zákazníků se s tímto trendem značně mění. Zákazníci, kteří nakupují na internetu, si zvykli na technologické novinky a společnost, která dokáže jejich očekávání naplnit a nakupování inovovat, má velký potenciál převzít podstatnou část trhu.

Důsledkem globalizace jsou nadnárodní internetové obchody, které působí v desítkách států po celém světě a své zboží nabízejí stovkám milionů potenciálních zákazníků ve stejném elektronickém obchodě, jen v upravené jazykové mutaci. Pro takto silné společnosti není problém přejít na odlišný způsob prezentace zboží a začít 3D elektronické obchody naplno využívat.

V budoucnu lze očekávat masivní rozšíření této technologie včetně variací na ni, jako jsou například zrcadla, která budou ve spolupráci s počítačem a kamerami schopny ukázat zákazníkovi, jak požadovaný kus oblečení vypadá oblečený přímo na něm, odpadne nutnost nosit speciální 3D brýle a cena 3D zobrazovací techniky klesne tak, jako klesly ceny plasmových a LCD televizí. Poté se 3D technologie stane naprosto běžnou součástí našeho života.

6 Seznam použité literatury

- 1) *3D pdf*. [online]. 2009 [cit. 2010 – 11 - 10]. Dostupné z: <www.pdf3d.cz>.
- 2) *3D systémy v českých kinech*. [online]. 2010 [cit. 2011 - 04 - 03]. Dostupné z: <<http://www.3dkina.cz/3d-systemy-v-ceskych-kinech>>.
- 3) *Aktivní 3D projekce*. [online]. 2009 [cit. 2011 - 04 - 15]. Dostupné z: <<http://nardic.cz/prehled-3d-technologie/aktivni-3d-projekce>>.
- 4) *Autostereoskopické monitory*. [online]. 2009 [cit. 2011 - 04 - 15]. Dostupné z: <<http://www.gali-3d.com/cz/techno-a-stereo-m/techno-a-stereo-m.php>>.
- 5) BAKALAR, J. *Xbox 360 Kinect*. [online]. 2010 [cit. 2010 - 04 - 15]. Dostupné z: <http://reviews.cnet.com/xbox-360-accessories/xbox-360-kinect/4505-9994_7-34182443.html?tag=cnetRiver#reviewPage1>. Poslední aktualizace 11.4.2010.
- 6) *Blender*. [online]. 2010 [cit. 2010 - 12 - 20]. Dostupné z: <<http://www.grafika.cz/blender>>. Poslední úprava 14.3.2011.
- 7) *Ceny 3D televizorů klesnou, jejich prodej letos vzroste více než pětkrát*. [online]. 2011 [cit. 2011 - 05 - 12]. Dostupné z: <<http://www.novinky.cz/internet-a-pc/232836-ceny-3d-televizoru-klesnou-jejich-prodej-letos-vzroste-vice-nez-petkrat.html?ref=boxE>>. Poslední aktualizace 10.5.2011.
- 8) *Co na internetu nakupujeme*. [online]. 2011 [cit. 2011 - 04 - 25]. Dostupné z: <http://www.czso.cz/csu/redakce.nsf/i/co_na_internetu_nakupujeme>. Poslední aktualizace 21.4.2011.
- 9) *FinePix REAL 3D W1*. [online]. 2009 [cit. 2011 - 03 - 20]. Dostupné z: <http://www.fujifilm.com/products/3d/camera/finepix_real3dw1/>.
- 10) *Fotostudio*. [online]. 2008 [cit. 2010 - 12 - 03]. Dostupné z: <<http://www.3dphotovision.com>>.
- 11) FRIMMEL, M. *Elektronický obchod/právní úprava*. 1. vyd. Praha: Prospektum, 2002, 321 s., ISBN 80-7175-114-6.
- 12) *Imax rok a půl po začátku digitální éry*. [online]. 2009 [cit. 2011 - 04 - 10]. Dostupné z: <<http://www.3dkina.cz/imax-rok-a-pul-po-zacatku-digitalni-ery>>.

- 13) *Interaktivní pdf*. [online]. 2009 [cit. 2011 - 02 - 27]. Dostupné z:
<<http://www.pdf3d.cz>>. Poslední úprava 30.9.2009.
- 14) *Internet drtí kamenné obchody*. [online]. 2010 [cit. 2011 - 02 - 22]. Dostupné z:
<<http://managerweb.ihned.cz>>. Poslední úprava 20.3.2010.
- 15) *Internetová reklama*. [online]. 2009 [cit. 2011 - 02 - 24]. Dostupné z:
<http://www.marketingovenoviny.cz/internetova_reklama%ref> . Poslední úprava 12.5.2009.
- 16) *Kinect*. [online]. 2011 [cit. 2011 - 04 - 14]. Dostupné z:
<<http://en.wikipedia.org/wiki/Kinect>>. Poslední aktualizace 17.3.2011.
- 17) *Mobilní fotostudio*. [online]. 2009 [cit. 2010 - 10 - 15]. Dostupné z:
<<http://www.innovate.cz>>.
- 18) *Pasivní 3D projekce*. [online]. 2009 [cit. 2011 - 04 - 15]. Dostupné z:
<<http://nardic.cz/prehled-3d-technologie/pasivni-3d-projekce> >.
- 19) POKORNÝ, P. *Blender – Naučte se 3D grafiku*. 1. vyd. Praha: BEN, 2006, 241 s., ISBN 80-7300-203-5.
- 20) POUR, J. a kol. *Informační systémy a elektronické podnikání*. 1. vyd. Praha: VŠE Praha, 2003, 136 s., ISBN 80-245-0227-5
- 21) *Překážky v zahraničním obchodu*. [online]. 2010 [cit. 2010 - 12 - 05]. Dostupné z: <<http://www.businessinfo.cz/cz/clanek/orientace-v-pravnich-ukonech/elektronicky-obchod-opu/1000818/7013/>>. Poslední aktualizace 7.6.2010.
- 22) RIPPL, M. *Optici varují před dlouhodobým sledováním 3D obrazu*. [online]. 2011 [cit. 2011 - 04 - 11]. Dostupné z: <<http://pcworld.cz/novinky/optici-varuji-pred-dlouhodobym-sledovanim-3d-obrazu-17166>>. Poslední aktualizace 27.2.2011.
- 23) RYBKA, M., MALÝ, O. *Jak komunikovat elektronicky*. 1. vyd. Praha: Grada, 2002, 119 s., ISBN 80-247-0208-8.
- 24) SMEJKAL, V. *Právo informačních a telekomunikačních systémů*. 1. vyd. Praha: C. H. Beck, 2004, 770 s., ISBN 80-7179-765-0.

- 25) SpeedyXx. *Recenze FinePix Real 3D W1*. [online]. 2010 [cit. 2011 - 04 - 10].
Dostupné z: <<http://www.recenze.okamzite.eu/articles/finepix-real-3d-w1-zachytte-sve-foto-3d/>>. Poslední aktualizace 15.3.2010.
- 26) STEVENS, T. *Microsoft sells 10 million Kinects, 10 Million Kinect games*. [online]. 2011 [cit. 2011 - 04 - 14]. Dostupné z:
<<http://www.engadget.com/2011/03/09/microsoft-sells-10-million-kinects-10-million-kinect-games/>>. Poslední aktualizace 9.3.2011.
- 27) *Obrázek anaglyf*. [online]. 2009 [cit. 2011 - 04 - 12]. Dostupné z:
<http://anaglyfy.xf.cz/photo/pictures/Letecky%20den%20NATO/snimek%2040_a.jpg>.
- 28) *Obrázek brýle*. [online]. 2008 [cit. 2011 - 04 - 23]. Dostupné z:
<<http://3dfoto.wz.cz/bryle.htm#bryle>>.
- 29) *Obrázek Citroen blueprint*. [online]. 2010 [cit. 2011 - 05 - 02]. Dostupné z:
<<http://nekonecnapohoda.cz/blog/9-designcidizajn/1090-blueprinty-aut-pro-pripravu-grafiky>>.
- 30) *WebGL*. [online]. 2011 [cit. 2011 - 05 - 05]. Dostupné z
<<http://en.wikipedia.org/wiki/WebGL>>. Poslední aktualizace 17.3.2011.
- 31) *Web blueprints*. [online]. 2011 [cit. 2011 - 02 - 12]. Dostupné z:
<<http://www.cgtextures.com>>.
- 32) *Wiki 3D*. [online]. 2011 [cit. 2011 - 03 - 19]. Dostupné z:
<<http://cs.wikipedia.org/wiki/3D>>. Poslední aktualizace 26.3.2011.

Přílohy

Seznam obrázků

<i>Obrázek 1: Realizace 3D pdf (1)</i>	24
<i>Obrázek 2: Anaglyf (27)</i>	27
<i>Obrázek 3: Anaglyfické brýle(28)</i>	28
<i>Obrázek 4: Fujifilm FinePix REAL 3D W1 (9)</i>	29
<i>Obrázek 5: Dvě fotografie z různých úhlů a její zpracování do 3D fotografie (25)</i>	29
<i>Obrázek 6: Aktivní 3D projekce (25)</i>	30
<i>Obrázek 7: Pasivní 3D projekce (18)</i>	32
<i>Obrázek 8: Autostereoskopická projekce(4)</i>	33
<i>Obrázek 9: Popis lidského oka (22)</i>	40
<i>Obrázek 10: Screenshot z web hry založené na WebGL (30)</i>	44
<i>Obrázek 11: 3D hodiny (30)</i>	45
<i>Obrázek 12: Zařízení Kinect (16)</i>	46
<i>Obrázek 13: Infračervený obrázek snímané scény (16)</i>	46
<i>Obrázek 14: Hloubka obrazu je reprezentována barvami – bílá nejbližší, modrá nejvzálenější objekt (16)</i>	47
<i>Obrázek 15: Blueprint Citroen (29)</i>	51
<i>Obrázek 16: Model místnosti 3D elektronického obchodu</i>	52
<i>Obrázek 17: Panel pro nastavení materiálu a textur</i>	54
<i>Obrázek 18: Textura použitá na stěny (31)</i>	55

<i>Obrázek 19: Textura použitá pro podlahu (31)</i>	55
<i>Obrázek 20: Přiřazení textury zdi</i>	56
<i>Obrázek 21: Upravená textura pro plastickou zed'</i>	57
<i>Obrázek 22: Cihlová zed' s použitím dvou textur</i>	58
<i>Obrázek 23: Původní cihlová zed'</i>	58
<i>Obrázek 24: Základní model 3D internetového obchodu</i>	59
<i>Obrázek 25: Základní tvar stojanu</i>	60
<i>Obrázek 26: Stojan v editačním módu</i>	62
<i>Obrázek 27: Upravený stojan v editačním módu</i>	63
<i>Obrázek 28: Kompletní stojan v editačním módu</i>	64
<i>Obrázek 29: Kompletní renderovaný stojan</i>	65
<i>Obrázek 30: Základní umístění vertexů a vertexy po použití funkce spin</i>	66
<i>Obrázek 31: Srovnání předlohy pивní sklenice a jejího modelu v Blenderu</i>	67
<i>Obrázek 32: Srovnání předlohy sklenice na šampaňské a jejího modelu v Blenderu</i>	68
<i>Obrázek 33: Srovnání předlohy sklenice na víno a jejího modelu v Blenderu</i>	69
<i>Obrázek 34: 3D elektronický obchod</i>	70
<i>Obrázek 35: Detail na stojan se sklenicemi</i>	71

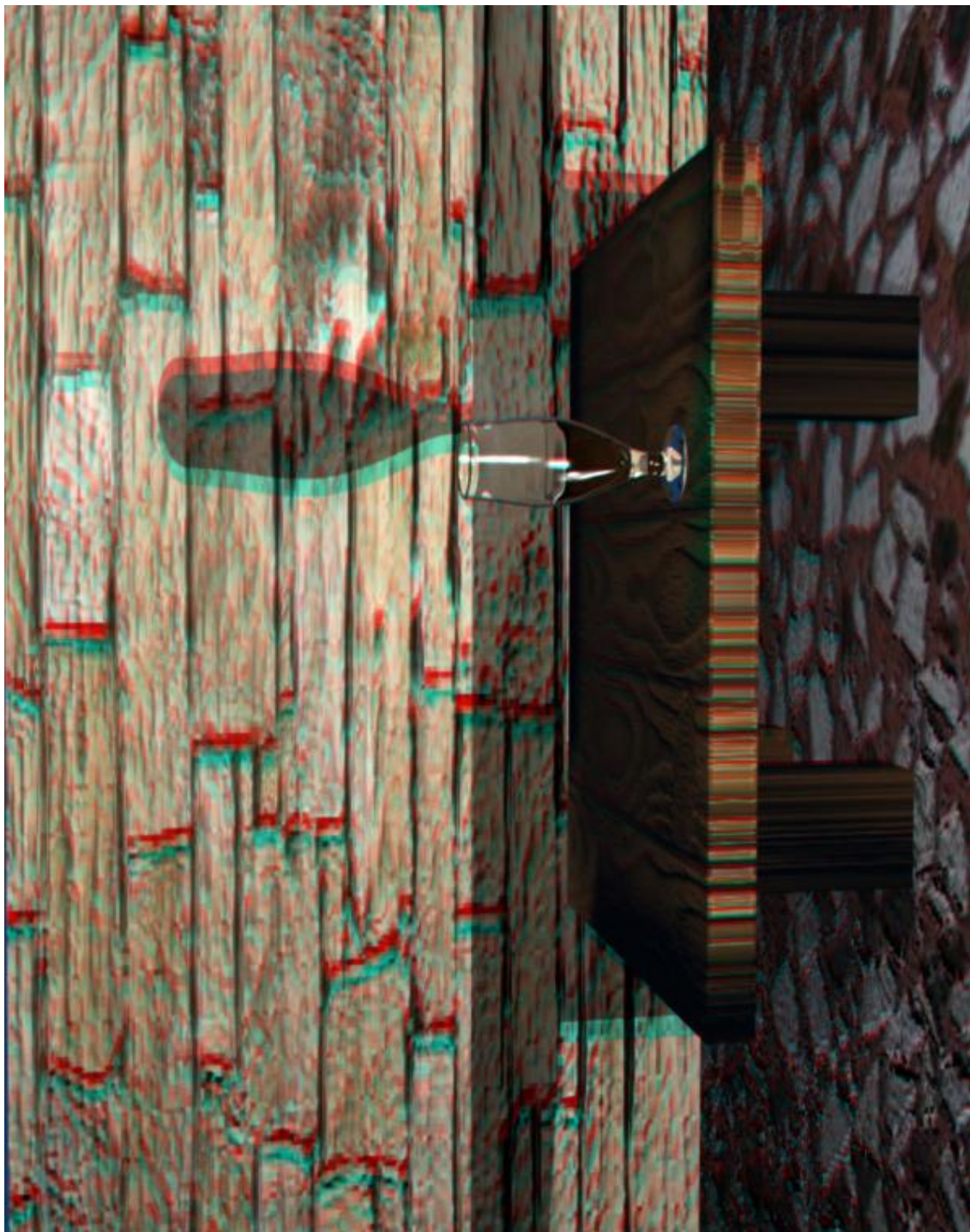
Seznam tabulek

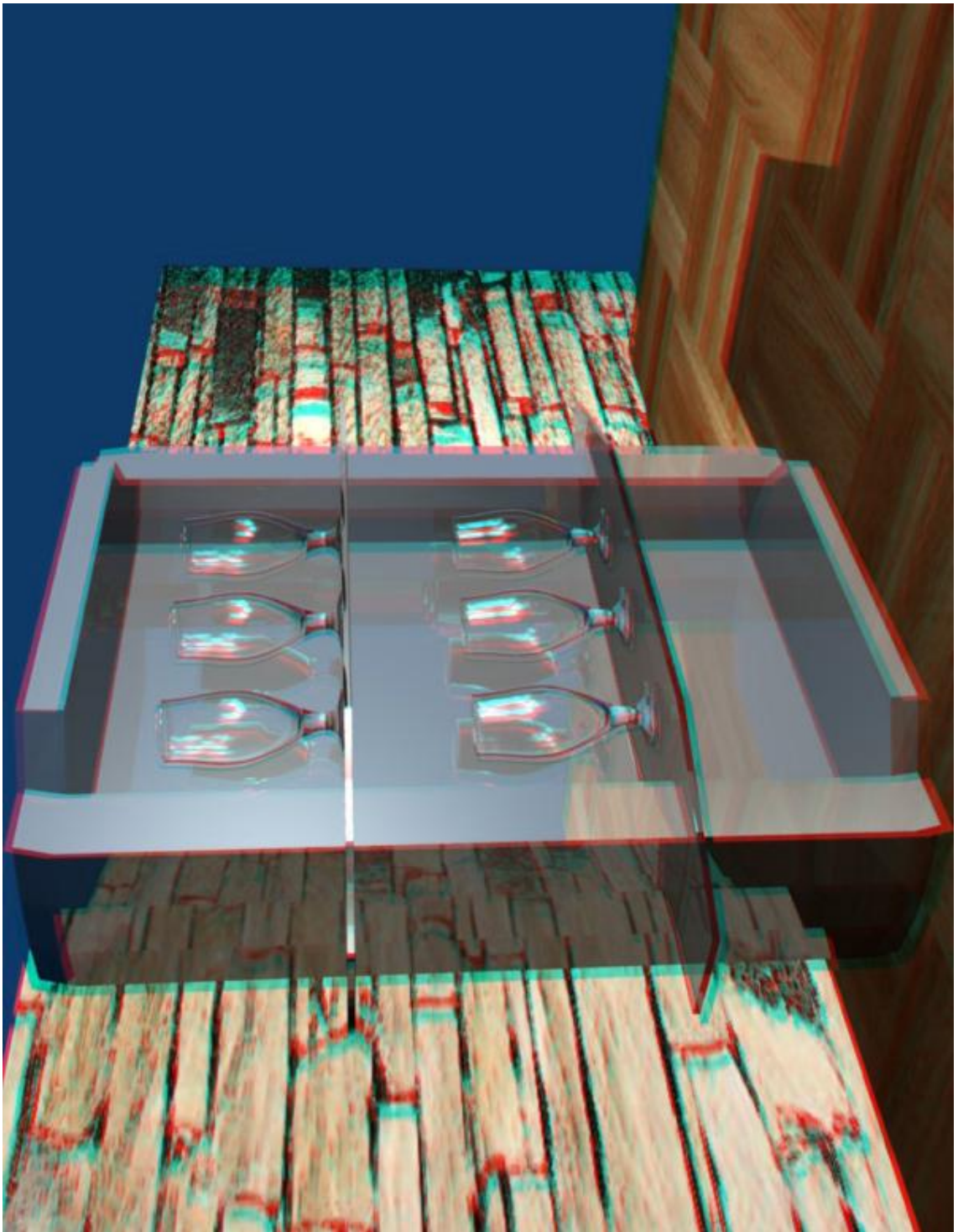
<i>Tabulka 1: Orientační cena za 3D foto</i>	21
<i>Tabulka 2: Srovnání 3D zobrazovacích technologií</i>	35

Seznam grafů

Graf 1: Procento jednotlivců, kteří na internetu nakoupili alespoň jednou za daný rok.....	16
Graf 2: Uváděné legislativní překážky.....	17
Graf 3: Typy společností.....	18
Graf 4: Názor na legislativu v oblasti el. Obchodu.....	19

Příloha 1 - alternativní 3D elektronický obchod (anaglyf)





Příloha 2 - WebGL Quit reference card

WebGL 1.0 API Quick Reference Card - Page 1

WebGL® is a software interface for accessing graphics hardware from within a web browser. Based on OpenGL ES 2.0, WebGL allows a programmer to specify the objects and operations involved in producing high-quality graphical images, specifically color images of 3D objects.

- [n.n.] refers to sections in the WebGL 1.0 specification, available at www.khronos.org/webgl
- Content marked in purple does not have a corresponding function in OpenGL ES. The OpenGL ES 2.0 specification is available at www.khronos.org/registry/gles

WebGL function calls behave identically to their OpenGL ES counterparts unless otherwise noted.

Interfaces

Interfaces are optional requests and may be ignored by an implementation. See `getContextAttributes` for actual values.

WebGLContextAttributes [5.2]

This interface contains requested drawing surface attributes and is passed as the second parameter to `getContext`.

Attributes:

alpha	Default: true
If true, requests a drawing buffer with an alpha channel for the purposes of performing OpenGL destination alpha operations and compositing with the page.	
depth	Default: true
If true, requests drawing buffer with a depth buffer of at least 16 bits.	
stencil	Default: false
If true, requests a stencil buffer of at least 8 bits.	
antialias	Default: true
If true, requests drawing buffer with antialiasing using its choice of technique (multisample/supersample) and quality.	
premultipliedAlpha	Default: true
If true, requests drawing buffer which contains colors with premultiplied alpha. (Ignored if Alpha is false.)	
preserveDrawingBuffer	Default: false
If true, requests that contents of the drawing buffer remain in between frames, at potential performance cost.	

Per-Fragment Operations [5.13.3]

void **blendColor**(float red, float green, float blue, float alpha)

void **blendEquation**(enum mode)

mode: See `modeRGB` for `blendEquationSeparate`

void **blendEquationSeparate**(enum modeRGB, enum modeAlpha)

modeRGB, modeAlpha: FUNC_ADD, FUNC_SUBTRACT, FUNC_REVERSE, SUBTRACT

void **blendFunc**(enum sfactor, enum dfactor)

sfactor: Same as for `dfactor`, plus SRC_ALPHA, SATURATE

dfactor: ZERO, ONE, [ONE_MINUS_]SRC_COLOR, [ONE_MINUS_]DST_COLOR, [ONE_MINUS_]SRC_ALPHA, [ONE_MINUS_]DST_ALPHA, [ONE_MINUS_]CONSTANT_COLOR, [ONE_MINUS_]CONSTANT_ALPHA

Note: Src and dst factors may not both reference constant color

void **blendFuncSeparate**(enum srcRGB, enum dstRGB, enum srcAlpha, enum dstAlpha)

srcRGB, srcAlpha: See `sfactor` for `blendFunc`

dstRGB, dstAlpha: See `dfactor` for `blendFunc`

Note: Src and dst factors may not both reference constant color

void **depthFunc**(enum func)

func: NEVER, ALWAYS, LESS, EQUAL, LEQUAL, GREATER, GEQUAL, NOTEQUAL

void **sampleCoverage**(float value, bool invert)

void **stencilFunc**(enum func, int ref, uint mask)

func: NEVER, ALWAYS, LESS, LEQUAL, [NOTEQUAL, GREATER, GEQUAL]

void **stencilFuncSeparate**(enum face, enum func, int ref, uint mask)

face: FRONT, BACK, FRONT_AND_BACK

func: NEVER, ALWAYS, LESS, LEQUAL, [NOTEQUAL, GREATER, GEQUAL]

void **stencilOp**(enum fail, enum zfail, enum zpass)

fail, zfail, and zpass: KEEP, ZERO, REPLACE, INCR, DECR, INVERT, INCR_WRAP, DECR_WRAP

void **stencilOpSeparate**(enum face, enum fail, enum zfail, enum zpass)

face: FRONT, BACK, FRONT_AND_BACK

fail, zfail, and zpass: See fail, zfail, and zpass for `stencilOp`

Detect and Enable Extensions [5.13.14]

string[] `getSupportedExtensions()`

object `getExtension(string name)`

The WebGL Context and getContext() [2.5]

This object manages OpenGL state and renders to the a drawing buffer, which must also be created at the same time of as the context creation. Create the `WebGLRenderingContext` object and drawing buffer by calling the `getContext` method of a given `HTMLCanvasElement` object with the exact string 'webgl'. The drawing buffer is also created by `getContext`.

For example:

```
<!DOCTYPE html>
<html><body>
  <canvas id="c"></canvas>
  <script type="text/javascript">
    var canvas = document.getElementById("c");
    var gl = canvas.getContext("webgl");
    gl.clearColor(1.0, 0.0, 0.0, 1.0);
    gl.clear(gl.COLOR_BUFFER_BIT);
  </script>
</body></html>
```

WebGLObject [5.3]

This is the parent interface for all WebGL resource objects.

Resource interface objects:

WebGLBuffer [5.4]	OpenGL Buffer Object.
WebGLProgram [5.6]	OpenGL Program Object.
WebGLRenderbuffer [5.7]	OpenGL Renderbuffer Object.
WebGLShader [5.8]	OpenGL Shader Object.
WebGLTexture [5.9]	OpenGL Texture Object.
WebGLUniformLocation [5.10]	Location of a uniform variable in a shader program.
WebGLActiveInfo [5.11]	Information returned from calls to <code>getActiveAttrib</code> and <code>getActiveUniform</code> . Has the following read-only properties: name, location, size, type.

WebGLRenderingContext [5.13]

This is the principal interface in WebGL. The functions listed on this reference card are available within this interface.

Attributes:

canvas	Type: HTMLCanvasElement
A reference to the canvas element which created this context.	
drawingBufferWidth	Type: GLsizei
The actual width of the drawing buffer, which may differ from the width attribute of the HTMLCanvasElement if the implementation is unable to satisfy the requested width or height.	
drawingBufferHeight	Type: GLsizei
The actual height of the drawing buffer, which may differ from the height attribute of the HTMLCanvasElement if the implementation is unable to satisfy the requested width or height	

ArrayBuffer and Typed Arrays [5.12]

Data is transferred to WebGL using `ArrayBuffer` and views. Buffers represent unstructured binary data, which can be modified using one or more typed array views.

Buffers

ArrayBuffer(ulong byteLength)

ulong byteLength: read-only, length of view in bytes.

Creates a new buffer. To modify the data, create one or more views referencing it.

Views

In the following, `ViewType` may be `Int8Array`, `Int16Array`, `Int32Array`, `Uint8Array`, `Uint16Array`, `Uint32Array`, `Float32Array`.

ViewType(ulong length)

Creates a view and a new underlying buffer.

ulong length: Read-only, number of elements in this view.

ViewType(ViewType other)

Creates new underlying buffer and copies 'other' array.

ViewType(type[] other)

Creates new underlying buffer and copies 'other' array.

ViewType(ArrayBuffer buffer, [optional] ulong byteOffset, [optional] ulong length)

Create a new view of given buffer, starting at optional byte offset, extending for optional length elements.

ArrayBuffer *buffer*: Read-only, buffer backing this view

ulong *byteOffset*: Read-only, byte offset of view start in buffer

ulong *length*: Read-only, number of elements in this view

Other Properties

ulong *byteLength*: Read-only, length of view in bytes.

const ulong *BYTES_PER_ELEMENT*: element size in bytes.

Methods

view[] = get/set element i

set(ViewType other, [optional] ulong offset)

set(type[] other, [optional] ulong offset)

Replace elements in this view with those from other, starting at optional offset.

ViewType *subset*(long begin, [optional] long end)

Return a subset of this view, referencing the same underlying buffer.

Whole Framebuffer Operations [5.13.3]

void **clear**(ulong mask) [5.13.11]

mask: Bitwise OR of (COLOR, DEPTH, STENCIL)_BUFFER_BIT

void **clearColor**(float red, float green, float blue, float alpha)

void **clearDepth**(float depth)

depth: Clamped to the range 0 to 1.

void **clearStencil**(int s)

void **colorMask**(bool red, bool green, bool blue, bool alpha)

void **depthMask**(bool flag)

void **stencilMask**(uint mask)

void **stencilMaskSeparate**(enum face, uint mask)

face: FRONT, BACK, FRONT_AND_BACK

Buffer Objects [5.13.5]

Once bound, buffers may not be rebound with a different Target.

void **bindBuffer**(enum target, Object buffer)

target: ARRAY_BUFFER, ELEMENT_ARRAY_BUFFER

void **bufferData**(enum target, long size, enum usage)

target: ARRAY_BUFFER, ELEMENT_ARRAY_BUFFER

usage: STATIC_DRAW, STREAM_DRAW, DYNAMIC_DRAW

void **bufferData**(enum target, Object data, enum usage)

target and usage: Same as for `bufferData` above

void **bufferSubData**(enum target, long offset, Object data)

target: ARRAY_BUFFER, ELEMENT_ARRAY_BUFFER

Object **createBuffer()**

Note: Corresponding OpenGL ES function is `GenBuffers`

void **deleteBuffer**(Object buffer)

any **getBufferParameter**(enum target, enum pname)

target: ARRAY_BUFFER, ELEMENT_ARRAY_BUFFER

pname: BUFFER_SIZE, BUFFER_USAGE

bool **isBuffer**(Object buffer)

View and Clip [5.13.3 - 5.13.4]

The viewport specifies the affine transformation of x and y from normalized device coordinates to window coordinates. Drawing buffer size is determined by the `HTMLCanvasElement`.

void **depthRange**(float zNear, float zFar)

zNear: Clamped to the range 0 to 1. Must be <= zFar

zFar: Clamped to the range 0 to 1.

void **scissor**(int x, int y, long width, long height)

void **viewport**(int x, int y, long width, long height)

Rasterization [5.13.3]

void **cullFace**(enum mode)

mode: BACK, FRONT_AND_BACK, FRONT

void **frontFace**(enum mode)

mode: CCW, CW

void **lineWidth**(float width)

void **polygonOffset**(float factor, float units)

Detect context lost events [5.13.13]

bool **isContextLost()**

WebGL 1.0 API Quick Reference Card - Page 2

Programs and Shaders [5.13.9]

Rendering with OpenGL ES 2.0 requires the use of shaders. Shaders must be loaded with a source string (`shaderSource`), compiled (`compileShader`), and attached to a program (`attachShader`) which must be linked (`linkProgram`) and then used (`useProgram`).

```
void attachShader(Object program, Object shader)
void bindAttribLocation(Object program, uint index,
string name)
void compileShader(Object shader)
Object createProgram()
Object createShader(enum type)
type: VERTEX_SHADER, FRAGMENT_SHADER
void deleteProgram(Object program)
void deleteShader(Object shader)
void detachShader(Object program, Object shader)
Object[] getAttachedShaders(Object program)
any getProgramParameter(Object program, enum pname)
Note: Corresponding OpenGL ES function is GetProgramiv
pname: DELETE_STATUS, LINK_STATUS, VALIDATE_STATUS,
ATTACHED_SHADERS, ACTIVE_ATTRIBUTES, UNIFORMS
string getProgramInfoLog(Object program)
any getShaderParameter(Object shader, enum pname)
Note: Corresponding OpenGL ES function is GetShaderiv
pname: SHADER_TYPE, DELETE_STATUS, COMPILER_STATUS
string getShaderInfoLog(Object shader)
string getShaderSource(Object shader)
bool isProgram(Object program)
bool isShader(Object shader)
void linkProgram(Object program)
void shaderSource(Object shader, string source)
void useProgram(Object program)
void validateProgram(Object program)
```

Uniforms and Attributes [5.13.10]

Values used by the shaders are as uniform of vertex attributes.

```
void disableVertexAttribArray(uint index)
index: [0, MAX_VERTEX_ATTRIBS - 1]
void enableVertexAttribArray(uint index)
index: [0, MAX_VERTEX_ATTRIBS - 1]
Object getActiveAttrib(Object program, uint index)
Object getActiveUniform(Object program, uint index)
along getAttribLocation(Object program, string name)
any getUniform(Object program, uint location)
uint getUniformLocation(Object program, string name)
any getVertexAttrib(uint index, enum pname)
pname: CURRENT_VERTEX_ATTRIB, VERTEX_ATTRIB_ARRAY,
[BUFFER_BINDING, ENABLED, SIZE, STRIDE, TYPE, NORMALIZED]
long getVertexAttribOffset(uint index, enum pname)
Note: Corresponding OpenGL ES function is GetVertexAttribPointerv
pname: VERTEX_ATTRIB_ARRAY_POINTER
void uniform[1234][f](uint location, ...)
void uniform[1234][fv](uint location, Array value)
void uniformMatrix[234]fv(uint location, bool transpose, Array)
transpose: FALSE
void vertexAttrib[1234]f(uint index, ...)
void vertexAttrib[1234]fv(uint index, Array value)
void vertexAttribPointer(uint index, int size, enum type,
bool normalized, long stride, long offset)
type: BYTE, SHORT, UNSIGNED_BYTE, SHORT, FIXED, FLOAT
index: [0, MAX_VERTEX_ATTRIBS - 1]
stride: [0, 255]
offset, stride: must be a multiple of the type size in WebGL
```

Framebuffer Objects [5.13.6]

Framebuffer objects provide an alternative rendering target to the drawing buffer.

```
void bindFramebuffer(enum target, Object framebuffer)
target: FRAMEBUFFER
enum checkFramebufferStatus(enum target)
target: FRAMEBUFFER
Returns: FRAMEBUFFER_COMPLETE, UNSUPPORTED,
FRAMEBUFFER_INCOMPLETE_ATTACHMENT, DIMENSIONS,
MISSING_ATTACHMENT
```

Texture Objects [5.13.8]

Texture objects provide storage and state for texturing operations. WebGL adds an error for operations relating to the currently bound texture if no texture is bound.

```
void activeTexture(enum texture) [5.13.3]
texture: TEXTURED_TEXTURE[i] where i =
MAX_COMBINED_TEXTURE_IMAGE_UNITS - 1
void bindTexture(enum target, Object texture)
target: TEXTURE_2D, TEXTURE_CUBE_MAP
void copyTexImage2D(enum target, int level,
enum internalformat, long width, long height,
long height, int border)
target: TEXTURE_2D, TEXTURE_CUBE_MAP_POSITIVE_X, Y, Z,
TEXTURE_CUBE_MAP_NEGATIVE_X, Y, Z
internalformat: ALPHA, LUMINANCE, LUMINANCE_ALPHA, RGB[A]
void copyTexSubImage2D(enum target, int level,
int xoffset, int yoffset, int x, int y, long width,
long height)
target: See target for copyTexImage2D
Object createTexture()
Note: Corresponding OpenGL ES function is GenTextures
void deleteTexture(Object texture)
void generateMipmap(enum target)
target: TEXTURE_2D, TEXTURE_CUBE_MAP
any getTexParameter(enum target, enum pname)
target: TEXTURE_2D, TEXTURE_CUBE_MAP
pname: TEXTURE_WRAP_S, T, TEXTURE_MIN, MAG, FILTER
bool isTexture(Object texture)
void texImage2D(enum target, int level,
enum internalformat, long width, long height,
int border, enum format, enum type, Object pixels)
void texImage2D(enum target, int level,
enum internalformat, enum format, enum type,
Object object)
Note: The following values apply to all variations of texImage2D.
target: See target for copyTexImage2D
internalformat: See internalformat for copyTexImage2D
format: ALPHA, RGB, RGBA, LUMINANCE, LUMINANCE_ALPHA
type: UNSIGNED_BYTE, UNSIGNED_SHORT_5_6_5,
UNSIGNED_SHORT_4_4_4_4, UNSIGNED_SHORT_5_5_5_1
object: pixels of type ImageData, image of type HTMLImageElement,
canvas of type HTMLCanvasElement,
video of type HTMLVideoElement
void texParameterf(enum target, enum pname, float param)
target: TEXTURE_2D, TEXTURE_CUBE_MAP
pname: TEXTURE_WRAP_S, T, TEXTURE_MIN, MAG, FILTER
void texParameteri(enum target, enum pname, int param)
target: TEXTURE_2D, TEXTURE_CUBE_MAP
pname: TEXTURE_WRAP_S, T, TEXTURE_MIN, MAG, FILTER
void texSubImage2D(enum target, int level, int xoffset,
int yoffset, long width, long height, enum format,
enum type, Object pixels)
void texSubImage2D(enum target, int level, int xoffset,
int yoffset, enum format, enum type, Object object)
Note: Following values apply to all variations of
texSubImage2D.
target: TEXTURE_CUBE_MAP_POSITIVE_X, Y, Z,
TEXTURE_CUBE_MAP_NEGATIVE_X, Y, Z
format and type: See format and type for texImage2D
object: Same as for texImage2D
```

Writing to the Draw Buffer [5.13.11]

When rendering is directed to drawing buffer, OpenGL ES 2.0 rendering calls cause the drawing buffer to be presented to the HTML page compositor at start of next compositing operation.

```
void drawArrays(enum mode, int first, long count)
mode: POINTS, LINE_STRIP, LINE_LOOP, LINES, TRIANGLE_STRIP,
TRIANGLE_FAN, TRIANGLES
first: May not be a negative value.
void drawElements(enum mode, long count, enum type,
long offset)
mode: POINTS, LINE_STRIP, LINE_LOOP, LINES, TRIANGLE_STRIP,
TRIANGLE_FAN, TRIANGLES
type: UNSIGNED_BYTE, UNSIGNED_SHORT
```

Special Functions [5.13.7]

```
contextStruct getContextAttributes() [5.13.2]
void disable(enum cap)
cap: BLEND, CULL_FACE, DEPTH_TEST, DITHER,
POLYGON_OFFSET_FILL, SAMPLE_ALPHA_TO_COVERAGE,
SAMPLE_COVERAGE, SCISSOR_TEST, STENCIL_TEST
void enable(enum cap)
cap: See cap for disable
void finish() [5.13.11]
void flush() [5.13.11]
enum getError()
Returns: OUT_OF_MEMORY, INVALID_ENUM, OPERATION,
FRAMEBUFFER_OPERATION, VALUE, NO_ERROR,
CONTEXT_LOST_WEBGL
any getParameter(enum pname)
pname: ALPHA, RED, GREEN, BLUE, SUBPIXEL_BITS,
ACTIVE_TEXTURE, ALIASED_LINE_WIDTH_POINT_SIZE_RANGE,
ARRAY_BUFFER_BINDING, BLEND_DST_ALPHA, RGB,
BLEND_EQUATION_ALPHA, RGB, BLEND_SRC_ALPHA, RGB,
BLEND_COLOR, COLOR_CLEAR_VALUE, WRITEMASK,
[NUM_COMPRESSED_TEXTURE_FORMATS, CULL_FACE_MODE],
CURRENT_PROGRAM, DEPTH_BITS, CLEAR_VALUE, FUNC,
RANGE, TEST, WRITEMASK, ELEMENT_ARRAY_BUFFER_BINDING,
DITHER, FRAMEBUFFER_BINDING, FRONT_FACE,
GENERATE_MIPMAP_HINT, LINE_WIDTH,
MAX_COMBINED_TEXTURE_IMAGE_UNITS,
MAX_CUBE_MAP_TEXTURE_RENDERBUFFER_TEXTURE_SIZE,
MAX_VARYING_VECTORS, MAX_VERTEX_ATTRIBS,
TEXTURE_IMAGE_UNITS, UNIFORM_VECTORS,
MAX_VIEWPORT_DIMS, PACK_ALIGNMENT,
POLYGON_OFFSET_FACTOR, FILL, UNITS,
RENDERBUFFER_BINDING, RENDERER, SAMPLE_BUFFERS,
SAMPLE_COVERAGE_INVERT, VALUE, SAMPLES,
SCISSOR_BOX, TEST, SHADING_LANGUAGE_VERSION,
STENCIL_BITS, CLEAR_VALUE, TEST, STENCIL_BACK_FAIL,
FUNC, REVALUE_MASK, WRITEMASK,
STENCIL_BACK_PASS_DEPTH_FAIL, PASS,
TEXTURE_BINDING_2D, CUBE_MAP, UNPACK_ALIGNMENT,
UNPACK_COLORSPACE_CONVERSION_WEBGL_FLIP_Y_WEBGL,
PREMULTIPLY_ALPHA_WEBGL, VENDOR, VERSION, VIEWPORT
void hint(enum target, enum mode)
target: GENERATE_MIPMAP_HINT
hint: FASTEST, NICEST, DONT_CARE
bool isEnabled(enum cap)
cap: See cap for disable
void pixelStore(enum pname, int param)
pname: UNPACK_ALIGNMENT, PACK_ALIGNMENT,
UNPACK_FLIP_Y_WEBGL, PREMULTIPLY_ALPHA_WEBGL,
UNPACK_COLORSPACE_CONVERSION_WEBGL
```

Renderbuffer Objects [5.13.7]

Renderbuffer objects are used to provide storage for the individual buffers used in a framebuffer object.

```
void bindRenderbuffer(enum target, Object renderbuffer)
target: RENDERBUFFER
Object createRenderbuffer()
Note: Corresponding OpenGL ES function is GenRenderbuffers
void deleteRenderbuffer(Object renderbuffer)
any getRenderbufferParameter(enum target, enum pname)
target: RENDERBUFFER
pname: RENDERBUFFER_WIDTH, HEIGHT, INTERNAL_FORMAT,
RENDERBUFFER_RED, GREEN, BLUE, ALPHA, DEPTH, STENCIL_SIZE
bool isRenderbuffer(Object renderbuffer)
void renderbufferStorage(enum target,
enum internalformat, long width, long height)
target: RENDERBUFFER
internalformat: DEPTH_COMPONENT16, RGBA4, RGB5_A1,
RGB565, STENCIL_INDEX8
```

Read Back Pixels [5.13.12]

Pixels in the current framebuffer can be read back into an `ArrayBufferView` object.

```
void readPixels(int x, int y, long width, long height,
enum format, enum type, Object pixels)
format: RGBA
type: UNSIGNED_BYTE
```

Object createFramebuffer()

Note: Corresponding OpenGL ES function is `GenFramebuffers`

```
void deleteFramebuffer(Object buffer)
void framebufferRenderbuffer(enum target, enum attachment,
enum renderbuffertarget, Object renderbuffer)
target: FRAMEBUFFER
attachment: COLOR_ATTACHMENT0, [DEPTH, STENCIL] ATTACHMENT
renderbuffertarget: RENDERBUFFER
bool isFramebuffer(Object framebuffer)
```

void framebufferTexture2D(enum target, enum attachment,

```
enum textarget, Object texture, int level)
target and attachment: Same as for framebufferRenderbuffer
textarget: TEXTURE_2D, TEXTURE_CUBE_MAP_POSITIVE_X, Y, Z,
TEXTURE_CUBE_MAP_NEGATIVE_X, Y, Z
any getFramebufferAttachmentParameter(enum target,
enum attachment, enum pname)
target and attachment: Same as for framebufferRenderbuffer
pname: FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE, NAME,
FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL,
FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE
```


OpenGL ES Shading Language 1.0 Quick Reference Card - Page 3

The OpenGL® ES Shading Language is two closely-related languages which are used to create shaders for the vertex and fragment processors contained in the OpenGL ES processing pipeline.

[n.n.n] refers to sections in the OpenGL ES Shading Language 1.0 specification at www.khronos.org/registry/gles

Types [4.1]

A shader can aggregate these using arrays and structures to build more complex types. There are no pointer types.

Basic Types

void	no function return value or empty parameter list
bool	Boolean
int	signed integer
float	floating scalar
vec2, vec3, vec4	n-component floating point vector
bvec2, bvec3, bvec4	Boolean vector
ivec2, ivec3, ivec4	signed integer vector
mat2, mat3, mat4	2x2, 3x3, 4x4 float matrix
sampler2D	access a 2D texture
samplerCube	access cube mapped texture

Structures and Arrays [4.1.8, 4.1.9]

Structures	<pre>struct type-name { members } struct-name[]; // optional variable declaration, // optionally an array</pre>
Arrays	<pre>float foo[3]; * structures and blocks can be arrays * only 1-dimensional arrays supported * structure members can be arrays</pre>

Operators and Expressions

Operators [5.1] Numbered in order of precedence. The relational and equality operators > < <= >= == != evaluate to a Boolean. To compare vectors component-wise, use functions such as lessThan(), equal(), etc.

Operator	Description	Associativity
1. ()	parenthetical grouping	N/A
2. []	array subscript	L - R
3. ()	function call & constructor structure field or method selector, swizzler postfix increment and decrement	L - R
4. ++ --	prefix increment and decrement	R - L
5. * /	multiplicative	L - R
6. + -	additive	L - R
7. < > <= >=	relational	L - R
8. == !=	equality	L - R
9. &&	logical and	L - R
10. ^^	logical exclusive or	L - R
11.	logical inclusive or	L - R
12. ? :	selection (Selects one entire operand. Use mix() to select individual components of vectors.)	L - R
13. = += -= *= /=	assignment arithmetic assignments	L - R
14. ,	sequence	L - R

Vector Components [5.5]

In addition to array numeric subscript syntax, names of vector components are denoted by a single letter. Components can be swizzled and replicated, e.g.: pos.xx, pos.zy

{x, y, z, w}	Use when accessing vectors that represent points or normals
{r, g, b, a}	Use when accessing vectors that represent colors
{s, t, p, q}	Use when accessing vectors that represent texture coordinates

Preprocessor [3.4]

Preprocessor Directives

The number sign (#) can be immediately preceded or followed in its line by spaces or horizontal tabs.

#	#define	#undef	#if	#ifdef	#ifndef	#else
#elif	#endif	#error	#pragma	#extension	#version	#line

Examples of Preprocessor Directives

- "#version 100" in a shader program specifies that the program is written in GLSL ES version 1.00. It is optional. If used, it must occur before anything else in the program other than whitespace or comments.
- #extension extension_name : behavior, where behavior can be require, enable, warn, or disable; and where extension_name is the extension supported by the compiler

Predefined Macros

__LINE__	Decimal integer constant that is one more than the number of preceding new-lines in the current source string
__VERSION__	Decimal integer, e.g.: 100
GL_ES	Defined and set to integer 1 if running on an OpenGL-ES Shading Language.
GL_FRAGMENT_PRECISION_HIGH	1 if highp is supported in the fragment language, else undefined [4.5.4]

Qualifiers

Storage Qualifiers [4.3]

Variable declarations may be preceded by one storage qualifier.

none	(Default) local read/write memory, or input parameter
const	Compile-time constant, or read-only function parameter
attribute	Linkage between a vertex shader and OpenGL ES for per-vertex data
uniform	Value does not change across the primitive being processed, uniforms form the linkage between a shader, OpenGL ES, and the application
varying	Linkage between a vertex shader and fragment shader for interpolated data

Uniform [4.3.4]

Use to declare global variables whose values are the same across the entire primitive being processed. All uniform variables are read-only. Use uniform qualifiers with any basic data types, to declare a variable whose type is a structure, or an array of any of these. For example:

```
uniform vec4 lightPosition;
```

Varying [4.3.5]

The varying qualifier can be used only with the data types float, vec2, vec3, vec4, mat2, mat3, mat4, or arrays of these. Structures cannot be varying. Varying variables are required to have global scope. Declaration is as follows:

```
varying vec3 normal;
```

Parameter Qualifiers [4.4]

Input values are copied in at function call time, output values are copied out at function return time.

none	(Default) same as in
in	For function parameters passed into a function
out	For function parameters passed back out of a function, but not initialized for use when passed in
inout	For function parameters passed both into and out of a function

Precision and Precision Qualifiers [4.5]

Any floating point, integer, or sampler declaration can have the type preceded by one of these precision qualifiers:

highp	Satisfies minimum requirements for the vertex language. Optional in the fragment language.
mediump	Satisfies minimum requirements for the fragment language. Its range and precision is between that provided by lowp and highp.
lowp	Range and precision can be less than mediump, but still represents all color values for any color channel.

For example:

```
lowp float color;
varying mediump vec2 Coord;
lowp ivec2 fo(lowp mat3);
highp mat4 m;
```

Ranges & precisions for precision qualifiers (FP=floating point):

	FP Range	FP Magnitude Range	FP Precision	Integer Range
highp	(-2 ²⁴ , 2 ²⁴)	(2 ¹² , 2 ²⁰)	Relative 2 ⁻²⁴	(-2 ¹⁶ , 2 ¹⁶)
mediump	(-2 ¹⁴ , 2 ¹⁴)	(2 ¹¹ , 2 ¹⁴)	Relative 2 ⁻¹⁰	(-2 ¹⁰ , 2 ¹⁰)
lowp	(-2, 2)	(2 ⁴ , 2)	Absolute 2 ⁻⁸	(-2 ² , 2 ²)

A precision statement establishes a default precision qualifier for subsequent int, float, and sampler declarations, e.g.: precision highp int;

Invariant Qualifiers Examples [4.6]

#pragma STDGLO invariant(all)	Force all output variables to be invariant
invariant gl_Position;	Qualify a previously declared variable
invariant varying mediump vec3 Color;	Qualify as part of a variable declaration

Order of Qualification [4.7]

When multiple qualifications are present, they must follow a strict order. This order is as follows.

invariant, storage, precision storage, parameter, precision

Aggregate Operations and Constructors

Matrix Constructor Examples [5.4]

```
mat2(float) // init diagonal
mat2(vec2, vec2); // column-major order
mat2(float, float, float, float); // column-major order
```

Structure Constructor Example [5.4.3]

```
struct light {float intensity; vec3 pos; };
light lightVar = light(3.0, vec3(1.0, 2.0, 3.0));
```

Matrix Components [5.6]

Access components of a matrix with array subscripting syntax.

```
For example:
mat4 m; // m represents a matrix
m[1] = vec4(2.0); // sets second column to all 2.0
m[0][0] = 1.0; // sets upper left element to 1.0
m[2][3] = 2.0; // sets 4th element of 3rd column to 2.0
```

Examples of operations on matrices and vectors:

```
m = f * m; // scalar * matrix component-wise
v = f * v; // scalar * vector component-wise
v = v * v; // vector * vector component-wise
```

```
m = m +/ - m; // matrix component-wise addition/subtraction
m = m * m; // linear algebraic multiply
m = v * m; // row vector * matrix linear algebraic multiply
m = m * v; // matrix * column vector linear algebraic multiply
f = dot(v, v); // vector dot product
v = cross(v, v); // vector cross product
m = matrixCompMult(m, m); // component-wise multiply
```

Structure Operations [5.7]

Select structure fields using the period (.) operator. Other operators include:

.	field selector
== !=	equality
=	assignment

Array Operations [4.1.9]

Array elements are accessed using the array subscript operator "[]". For example:

```
diffuseColor += lightIntensity[3] * NdotL;
```

OpenGL ES Shading Language 1.0 Quick Reference Card - Page 4

Built-In Inputs, Outputs, and Constants [7]

Shader programs use Special Variables to communicate with fixed-function parts of the pipeline. Output Special Variables may be read back after writing. Input Special Variables are read-only. All Special Variables have global scope.

Vertex Shader Special Variables [7.1]

Outputs:

Variable	Description	Units or coordinate system
highp vec4 gl_Position;	transformed vertex position	clip coordinates
mediump float gl_PointSize;	transformed point size (point rasterization only)	pixels

Fragment Shader Special Variables [7.2]

Fragment shaders may write to gl_FragColor or to one or more elements of gl_FragData[], but not both. The size of the gl_FragData array is given by the built-in constant gl_MaxDrawBuffers.

Inputs:

Variable	Description	Units or coordinate system
mediump vec4 gl_FragCoord;	fragment position within frame buffer	window coordinates
bool gl_FrontFacing;	fragment belongs to a front-facing primitive	Boolean
mediump vec2 gl_PointCoord;	fragment position within a point (point rasterization only)	0.0 to 1.0 for each component

Outputs:

Variable	Description	Units or coordinate system
mediump vec4 gl_FragColor;	fragment color	RGBA color
mediump vec4 gl_FragData[n];	fragment color for color attachment n	RGBA color

Built-In Constants With Minimum Values [7.4]

Built-in Constant	Minimum value
const mediump int gl_MaxVertexAttribs	8
const mediump int gl_MaxVertexUniformVectors	128
const mediump int gl_MaxVaryingVectors	8
const mediump int gl_MaxVertexTextureImageUnits	0
const mediump int gl_MaxCombinedTextureImageUnits	8
const mediump int gl_MaxTextureImageUnits	8
const mediump int gl_MaxFragmentUniformVectors	16
const mediump int gl_MaxDrawBuffers	1

Built-In Uniform State [7.5]

Specifies depth range in window coordinates. If an implementation does not support highp precision in the fragment language, and state is listed as highp, then that state will only be available as mediump in the fragment language.

```
struct gl_DepthRangeParameters {
    highp float near; // n
    highp float far; // f
    highp float dif; // f - n
};
uniform gl_DepthRangeParameters gl_DepthRange;
```

Built-In Functions

Angle & Trigonometry Functions [8.1]

Component-wise operation. Parameters specified as angle are assumed to be in units of radians. T is float, vec2, vec3, vec4.

T radians(T degrees)	degrees to radians
T degrees(T radians)	radians to degrees
T sin(T angle)	sine
T cos(T angle)	cosine
T tan(T angle)	tangent
T asin(T x)	arc sine
T acos(T x)	arc cosine
T atan(T y, T x)	arc tangent
T atan(T y_over_x)	arc tangent

Exponential Functions [8.2]

Component-wise operation. T is float, vec2, vec3, vec4.

T pow(T x, T y)	x ^y
T exp(T x)	e ^x
T log(T x)	ln
T exp2(T x)	2 ^x
T log2(T x)	log ₂
T sqrt(T x)	square root
T inversesqrt(T x)	inverse square root

Common Functions [8.3]

Component-wise operation. T is float, vec2, vec3, vec4.

T abs(T x)	absolute value
T sign(T x)	returns -1.0, 0.0, or 1.0
T floor(T x)	nearest integer <= x
T ceil(T x)	nearest integer >= x
T fract(T x)	x - floor(x)
T mod(T x, T y)	modulus
T min(T x, float y)	minimum value
T min(T x, T y)	minimum value
T max(T x, float y)	maximum value
T max(T x, T y)	maximum value
T clamp(T x, T minVal, T maxVal)	min(max(x, minVal), maxVal)
T clamp(T x, float minVal, float maxVal)	min(max(x, minVal), maxVal)
T mix(T x, T y, T a)	linear blend of x and y
T mix(T x, T y, float a)	linear blend of x and y
T step(T edge, T x)	0.0 if x < edge, else 1.0
T step(float edge, T x)	0.0 if x < edge, else 1.0
T smoothstep(T edge0, T edge1, T x)	clip and smooth
T smoothstep(float edge0, float edge1, T x)	clip and smooth

Geometric Functions [8.4]

These functions operate on vectors as vectors, not component-wise. T is float, vec2, vec3, vec4.

float length(T x)	length of vector
float distance(T p0, T p1)	distance between points
float dot(T x, T y)	dot product
vec3 cross(vec3 x, vec3 y)	cross product
T normalize(T x)	normalize vector to length 1
T faceforward(T N, T I, T Nref)	returns N if dot(Nref, I) < 0, else -N
T reflect(T I, T N)	reflection direction I - 2 * dot(N,I) * N
T refract(T I, T N, float eta)	refraction vector

Matrix Functions [8.5]

Type mat is any matrix type.

mat matrixCompMult(mat x, mat y) multiply x by y component-wise

Vector Relational Functions [8.6]

Compare x and y component-wise. Sizes of input and return vectors for a particular call must match. Type bvec is bvecn; vec is vecn; ivec is ivec n (where n is 2, 3, or 4). T is the union of vec and ivec.

bvec lessThan(T x, T y)	x < y
bvec lessThanEqual(T x, T y)	x <= y
bvec greaterThan(T x, T y)	x > y
bvec greaterThanEqual(T x, T y)	x >= y
bvec equal(T x, T y)	x == y
bvec equal(bvec x, bvec y)	x == y
bvec notEqual(T x, T y)	x != y
bvec notEqual(bvec x, bvec y)	x != y
bool any(bvec x)	true if any component of x is true
bool all(bvec x)	true if all components of x are true
bvec not(bvec x)	logical complement of x

Texture Lookup Functions [8.7]

Available only in vertex shaders.

vec4 texture2DLod(sampler2D sampler, vec2 coord, float lod)
vec4 texture2DProjLod(sampler2D sampler, vec3 coord, float lod)
vec4 texture2DProjLod(sampler2D sampler, vec4 coord, float lod)
vec4 textureCubeLod(samplerCube sampler, vec3 coord, float lod)

Available only in fragment shaders.

vec4 texture2D(sampler2D sampler, vec2 coord, float bias)
vec4 texture2DProj(sampler2D sampler, vec3 coord, float bias)
vec4 texture2DProj(sampler2D sampler, vec4 coord, float bias)
vec4 textureCube(samplerCube sampler, vec3 coord, float bias)

Available in vertex and fragment shaders.

vec4 texture2D(sampler2D sampler, vec2 coord)
vec4 texture2DProj(sampler2D sampler, vec3 coord)
vec4 texture2DProj(sampler2D sampler, vec4 coord)
vec4 textureCube(samplerCube sampler, vec3 coord)

Statements and Structure

Iteration and Jumps [6]

Function Call	call by value-return
Iteration	for (;) { break, continue } while () { break, continue } do { break, continue } while ();
Selection	if () { } if () { } else { }
Jump	break, continue, return discard // Fragment shader only
Entry	void main()

Sample Program

A shader pair that applies diffuse and ambient lighting to a textured object.

```
Vertex Shader
uniform mat4 mvp_matrix; // model-view-projection matrix
uniform mat3 normal_matrix; // normal matrix
uniform vec3 ec_light_dir; // light direction in eye coords

attribute vec4 a_vertex; // vertex position
attribute vec3 a_normal; // vertex normal
attribute vec2 a_texcoord; // texture coordinates

varying float v_diffuse;
varying vec2 v_texcoord;

void main(void)
{
    // put vertex normal into eye coords
    vec3 ec_normal = normalize(normal_matrix * a_normal);

    // emit diffuse scale factor, texcoord, and position
    v_diffuse = max(dot(ec_light_dir, ec_normal), 0.0);
    v_texcoord = a_texcoord;
    gl_Position = mvp_matrix * a_vertex;
}

Fragment Shader
precision mediump float;
uniform sampler2D t_reflectance;
uniform vec4 i_ambient;

varying float v_diffuse;
varying vec2 v_texcoord;

void main(void)
{
    vec4 color = texture2D(t_reflectance, v_texcoord);
    gl_FragColor = color * (v_diffuse + i_ambient);
}
```



WebGL and OpenGL ES are registered trademarks of Khronos Group. The Khronos Group is an industry consortium creating open standards for the authoring and acceleration of parallel computing, graphics and dynamic media on a wide variety of platforms and devices. See www.khronos.org to learn about the Khronos Group. See www.khronos.org/webgl to learn about WebGL. See www.khronos.org/OpenGL to learn about OpenGL ES.

Příloha 3 - bezpečnostní informace pro sledování 3D TV



IMPORTANT HEALTH AND SAFETY INFORMATION FOR 3D TELEVISION VIEWING.

3D Viewing Conditions

- To watch in 3D mode, you need to put the 3D Active Glasses on and press the power button on top of the glasses. Fluorescent lighting may cause a flickering effect and direct sunlight may affect the operation of the 3D glasses. Turn off all fluorescent lighting and block sources of direct sunlight before watching in 3D mode.

Read and understand the following safety information before using the TV's 3D function.

- If you or any of your relatives have a history of seizures or epilepsy, please consult with a medical specialist before using the 3D function.
- Some viewers may experience discomfort while viewing 3D television such as dizziness, nausea and headaches. If you experience any such symptom, stop viewing 3D television, remove 3D Active Glasses and rest.
- Watching 3D images for an extended period of time may cause eye strain. If you feel eye strain, stop viewing 3D television, remove your 3D Active Glasses and rest.
- A responsible adult should frequently check on children who are using the 3D function. If there are any reports of tired eyes, headaches, dizziness, or nausea, have the child stop viewing 3D television and rest.
- Do not use the 3D function or 3D Active Glasses while walking or moving around. Using the 3D function or 3D Active Glasses while moving around may result in injuries due to running into objects, tripping, and/or falling.
- Do not use the 3D Active Glasses for any purpose (such as general eyeglasses, sunglasses, protective goggles, etc.) other than for which they were designed.