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

Diplomová práce

Vytváření experimentů pro hodnocení kvality renderovaných scén ve virtuální realitě

Plzeň 2018 Matěj Kareš

Místo této strany bude zadání práce.

Prohlášení

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

V Plzni dne 17. května 2018

Matěj Kareš

Abstract

The focus of this thesis is graphical effects and their subjective perception in virtual reality. In the theoretical part of the thesis, virtual reality, perception oriented experiments and graphical effects will be discussed.

The goal is to create an environment for conducting subjective experiments on the subject of visual quality perception. Based on the results of said experiments adjustments can be made to strike balance between visual quality and performance.

Abstrakt

Tato práce se zabývá analýzou virtuální reality, konkrétně grafickými efekty a jejich vnímáním. V teoretickém rozboru jsou probrány koncepty virtuální reality, grafické efekty a provádění subjektivních experimentů.

Cílem je vytvořit prostředí pro tvorbu experimentů, které slouží k získání statistických dat o kvalitě pozorovaných scén. Na základě výsledků experimentu lze upravovat parametry výpočetně náročných grafických efektů na takové úrovně, kdy je dosaženo nejvyššího výkonu s co nejmenším možným negativním dopadem na kvalitu pozorované scény.

Obsah

1 Úvod

Virtuální realita se v současné době těší velkému zájmu. S rostoucí popularitou samotného konceptu vzniká také množství zařízení a software, které tento zážitek zprostředkovávají. Na trhu se setkáváme převážně s *headsety* (zařízení umístěné na hlavu), které vykreslují scénu stereoskopicky, a tím vzniká iluze přítomnosti v jiném světě – virtuální realitě. Často jsou headsety vybaveny zvukotechnikou s podporou prostorového zvuku a k ovládání jsou nabízeny pohybové ovladače.

Stereoskopické vykreslování má velký dopad na výkon hardware. Přestože má virtuální realita své kořeny v polovině minulého století, až v dnešní době je hardware na takové úrovni, že dokáže vykreslovat dostatek snímků za sekundu v přijatelné kvalitě tak, aby se uživatelé cítili v simulovaném světě přirozeně a pohodlně.

Nelze si ale dovolit kvalitu, která je standardem moderních počítačových her a simulací, protože jsou zamýšleny pro jeden monitor, a kde nižší snímkovací frekvence (dále FPS) nevadí. Ve virtuální realitě jsou vyšší nároky na snímkovací frekvenci (obvykle 90 FPS) a především na jejich stabilitu, protože výkyvy jsou velmi nepříjemné a mohou způsobit nepohodlí až nevolnost. Navíc musí vývojáři počítat s tím, že jsou scény vykreslovány ze dvou kamer najednou. Z tohoto důvodu je potřeba častých optimalizací a kompromisů.

Grafické efekty přispívají k výslednému vzhledu renderované scény. Příkladem může být korekce barev, vyhlazení hran, částicové efekty, shadery vody nebo skla a mnoho dalších. Tyto efekty, pro svoji různorodost, mají rozdílný dopad na výkon aplikace. K podobně velkým rozdílům dochází i v pozorované kvalitě výsledné scény. Grafický efekt, který je velmi výpočetně náročný, může mít téměř neznatelný vizuální dopad a naopak.

Cílem této práce je umožnit provádění a vytváření experimentů zaměřených na pozorování a hodnocení kvality renderovaných scén ve virtuální realitě. Jednotliví testující budou moci vytvořit testovací scény ke zkoumání určitého grafického efektu a testovaný subjekt bude dotazován v jaké konfiguraci se mu testovaný efekt líbí nejvíce, zda-li je efekt jako takový příjemný na pozorování a nebo naopak, zda-li je testovaný efekt nepříjemný, vizuálně nepřitažlivý, případně zda je efekt vůbec pozorovatelný. Výsledky experimentu jsou určeny pro statistické zpracování, na jehož základě lze provádět optimalizace. Součástí experimentu je také pozorování, zda-li se grafický efekt nechová ve virtuální realitě jinak, než mimo ni.

2 Teoretický základ

2.1 Virtuální realita

Virtuální realita je počítačem simulovaná skutečnost, se kterou může uživatel interagovat. Jedná se o kombinaci technologií, které poskytují vjemy pro různé smysly uživatele (nejčastěji zrak a sluch). Definic existuje mnoho, ale liší se spíše rozsahem než fakticky [1–5].

2.1.1 Historie

Zařízení související s virtuální realitou mají kořeny v 50. letech 20. století, kdy Morton L. Heilig navrhl náhlavní obrazovku (Head Mounted Display dále jen HMD) nesoucí název *Telesphere Mask*, jejíž součástí byla projekce obrazu pro každé oko zvlášť, zvuk a vůně [6]. V roce 1962 Heilig sestrojil prototyp zvaný *Sensorama*, ve kterém promítal 5 krátkých filmů, do nichž pozorovatele zapojoval více smysly (zrak, sluch, čich a hmat) [7].

Roku 1969 Ivan Shuterland a jeho student Bob Sproull vytvořili první HMD, který byl vybaven snímáním pohybu a dokázal vykreslovat počítačem generovanou grafiku (jednalo se převážně o jednoduché geometrické obrazce). Pro svoji hmotnost musel být ukotven na stropě, a tak se headset vepsal do dějin pod názvem „*The Sword of Damocles*“ kvůli své podobnosti s mečem ze stejnojmenné řecké báje [3].

V 80. letech se virtuální realita stala vlastním oborem počítačové techniky. Výzkumníci se věnovali sběru informací o užitku a aplikacích virtuální reality. Roku 1987 Jaron Lanier poprvé definoval pojem „Virtuální realita“. V této době vzniklo mnoho zajímavých projektů a teorií, kam by se mohla virtuální realita posunout v budoucnu [2, 8]. Komerční vývoj byl utlumen a společnosti se zaměřovaly na využití virtuální reality ve zdravotnictví, armádě, vzdělávání a marketingu.

V roce 1995 vydala společnost *Nintendo* herní konzolu s názvem *Virtual Boy*. Nepraktická konstrukce, monochromatické zobrazování, nízká kvalita her a mnoho dalších problémů vedly k celkovému neúspěchu této konzole, a Virtual Boy se tak stal odstrašujícím příkladem toho, proč je vize virtuální reality v dané době nedosažitelná.

2.1.2 Virtuální realita od roku 2013

Společnost Valve v roce 2013 učinila průlom ve vývoji virtuální reality díky displejům s nízkou perzistivitou. Tato technologie byla také svižně přijata společností Oculus a je využívána dodnes. V roce 2015 společnosti HTC a Valve představily vlastní headset s pohybovými ovladači s názvem HTC Vive. Vive se vyznačoval *lighthouse* technologií (dva infračervené vysílače v opačných rozích místnosti, které poskytují synchronizační prostředí, ve kterém se orientuje headset a pohybové ovladače – paralela s tím, jak se orientují lodě podle majáků).

Vývoj headsetů lze charakterizovat podle možností interakce s virtuálním prostředím. První generace headsetů (např. *Oculus Rift DK1*) dokázala snímat rotaci headsetu, a uživatelé se tak mohli ve virtuálním prostředí rozhlížet. Ovládání bylo omezeno pouze na standardní periferie (myš, klávesnice, gamepad), ale i tak měla první generace úspěch.

Druhá generace představila snímač polohy headsetu, a tím bylo umožněno (do určité míry v závislosti na záběru snímače) pohybovat hlavou ve virtuálním prostředí (např. krčení nebo nahlížení za roh).

V aktuální generaci jsou k headsetům dodávány pohybové ovladače, které simulují uživatelovy ruce, a představují tak nový způsob interakce s virtuálním prostředím. V současné době se hledají bezdrátová řešení, aby se potlačily nechtěné kolize s kabely headsetů a výrobci se zaměřují na *plug and play* přístup (nejmenší možné úsilí ke zprovoznění virtuální reality).

2.1.3 Výhody virtuální reality

Virtuální realita nabízí řadu výhod oproti obvyklému používání počítače.

Stereoskopický obraz Oproti standardním monitorům, kde vnímáme jeden dvourozměrný obraz, stereoskopická projekce dovoluje pozorovatelům vnímat hloubku virtuálního prostředí. Uživatelé tak lépe vnímají velikosti a vzdálenosti objektů [9].

Teleprezence *Prezence* je definována jako pocit náležitosti do určitého prostředí. Pokud tento pocit získáme prostřednictvím komunikačního média, jedná se o *teleprezenci* [1]. Uživatelé se tak mohou nacházet ve vzdálených, nedosažitelných či počítačem generovaných prostředích a stále se v nich cítit přítomni. S teleprezencí je úzce spjata imerze.

Imerze Přes nasazený headset není uživatel rušen vizuálními vjemy z reálného okolí a vidí jen virtuální prostředí. Krom zraku existuje také možnost využít stereofonních systémů pro potlačení zvukových vjemů

z reality a nahradit je virtuálním ozvučením. Takto lze dosáhnout řádově vyšší imerze, než je tomu u běžných počítačových konfigurací.

2.1.4 Problémy virtuální reality

Kromě zmíněných výhod s sebou nese virtuální realita řadu nevýhod.

Prolínání realit Jelikož se uživatel s nasazeným headsetem pohybuje ve dvou prostředích zároveň, kdy jedno vnímá obrazově a druhé fyzicky, může dojít ke kolizi s předměty, které nejsou reprezentovány ve virtuálním prostředí. Příkladem mohou být stoly, židle nebo jen samotné kabely vedoucí mezi headsetem a počítačem. Vlivem těchto kolizí může dojít ke zranění nebo škodám na nábytku a vybavení. Opačným příkladem může být snaha o posazení se na židli ve virtuálním prostředí, která však neexistuje v reálném světě. Návrháři aplikací pro virtuální realitu by měli tato rizika uvažovat a přizpůsobovat jim návrh prostředí.

Screen door effect (fixed-pattern noise) U nevhodně konstruovaných obrazovek nebo při malém rozlišením je možné pozorovat mřížku mezi jednotlivými pixely. Obraz se pak jeví, jako bychom na scénu hleděli skrze síť proti hmyzu. Pro tuto analogii se jevu říká *screen door effect*.

Zkreslení obrazu Headsety obsahují optické čočky, které mapují obraz z diplayů na zorné pole lidského oka. Během tohoto převodu dochází k nevyhnutelnému zkreslení obrazu. Pro bezchybný převod by bylo potřeba mít 9 čoček na oko [5], což je příliš mnoho na konstrukci přijatelně velkých a těžkých headsetů. Existují však metody, kterými lze zkreslení zmírnit, nikoliv však eliminovat úplně. Jedná se především o správné umístění a tvar čoček. Na softwarové úrovni lze obraz upravit tak, aby dopad zkreslení byl co možná nejmenší [5]. Tvar čočky také limituje natočení očí, protože je ohnisko zaměřeno na čočku lidského oka v základní poloze. Nejsou-li oči v základní poloze, obraz se bude jevit rozostřeně a bude patrná *chromatická aberace*.

Omezené zorné pole Výše zmíněné technologické limitace zatím neumožňují věrnou reprezentaci periferního vidění, proto je zorné pole částečně omezeno. Periferní vidění je důležité pro správnou orientaci a vnímání okolí. Na toto téma bylo vedeno několik experimentů společností Valve a výsledkem je, že zorné pole by mělo být alespoň 80° [5]. Moderní headsety mají zorné pole 110° a postupně jsou představovány headsety se zorným polem až 200°.

Výpadek snímání K výpadku snímání může dojít, pokud je headset (případně i pohybové ovladače) mimo záběr pohybových snímačů. Tato situace může nastat, pokud hráč svou pozicí (popř. tělem) tvoří překážku mezi headsetem a snímačem. Důsledkem je desynchronizace virtuálního prostředí s pohyby uživatele, což vede k nepříjemným pocitům a nevolnosti.

Část problémů má dopad na zdraví člověka a je jim věnována část 2.1.5.

2.1.5 Vliv na zdraví

S příchodem technologie, která je navíc určená přímo k nošení na těle, je potřeba prozkoumat vliv na zdraví nositele. Níže uvedené problémy se mohou lišit v intenzitě u různých lidí [10].

Simulátorová nevolnost (Simulator Sickness) Jedná se o variantu pohybové nemoci (*kinetóza*), kdy dochází k rozdílu vnímání pohybu *vestibulárním systémem*[1] proti vizuálním vjemům. Pohybovou nemoc zažíváme například při cestování vozidlem. Ve virtuální realitě je pohybová nevolnost způsobena i obráceně, tj. kamera se nehýbe, přestože uživatel ano [11].

Dezorientace je pociťována uživateli, pokud je jejich *avatar*[2] přemisťován po virtuálním prostředí. V reálném světě je teleportace nepřirozená a lidská těla na ni nejsou zvyklá. Ve virtuálních prostředích je teleportace nejčastější způsob přemisťování právě proto, že částečně eliminuje vznik simulátorové nevolnosti. Problém s dezorientací je považován za neškodný, když iniciátorem teleportace je uživatel, a tedy změnu prostředí očekává. V opačném případě lze pozorovat ztrátu rovnováhy, krátkodobou zmatenost a někdy je potřeba pro vysoké nepohodlí headset sundat.

Nepohodlí headsetu Každé zařízení s sebou nese váhu použitých technologií. V historii byla některá zařízení tak těžká, že se musela připevňovat ke stropu. V roce 1997 průměrný headset vážil 2 kilogramy. Dnešní headsety se pohybují váhově okolo poloviny kilogramu. Problém s váhou je patrný, pokud není headset vhodně vyvážený. Uživatelé tak mohou cítit určitou míru nepohodlí a zaznamenány jsou i bolesti svalů v oblasti krční páteře.

[1] Vestibulární systém je smyslový orgán ve středním uchu, zodpovědný za vnímání rovnováhy.

[2] Avatar je ztělesnění uživatele ve virtuálním prostředí.

Hygiena Headsety se v provozu zahřívají a vzniklé teplo je emitováno po celém jeho povrchu. V kombinaci s fyzickou aktivitou, která je často vyžadována, dochází k pocení. Headset pak představuje vhodné prostředí pro přenos virů a bakterií. Výrobci moderních headsetů prodávají výměnné a omyvatelné vložky, aby bylo možné přenosu různých patogenů zabránit.

Zrak Vlivy monoskopických monitorů na zrak jsou již poměrně dobře zmapovány. Problematika stereoskopického zobrazování je dle Costella [10] komplexnější. Vzniká problém s přizpůsobením očí pro vnímání hloubky předmětů. Tento jev pramení z plochých obrazovek, ze kterých vnímáme virtuální prostředí a to je v rozporu s vnímáním reality. Naše oči zaostřují podle vzdálenosti předmětů. Ve virtuální realitě jediná vzdálenost, kterou skutečně vnímáme, je vzdálenost obrazovek od očí. Lidské oko je schopné se přizpůsobit, ale dlouhodobé užívání může vést k oční únavě (pálení, slzení, bolest hlavy) [11, 12].

2.1.6 Aplikace

Virtuální realita je využívána v mnoha oblastech právě pro svoji vysokou imerzi. Uvedeme si několik hlavních sektorů:

Počítačové hry Vyšší imerze a nový způsob interakce se hrou jsou hlavní síly použití virtuální reality v počítačových hrách. Mnoho z existujících titulů jsou stále jen projekty typu *proof of concept* a komunita stále objevuje, jak hry upravovat pro dosažení synergie s virtuální realitou.

Virtuální cestování Do této kategorie lze zařadit cestování prostřednictvím virtuální reality. Teleprezence umožňuje podívat se na různá místa po světě, aniž by bylo potřeba opustit domov. Stejně tak můžeme prohlížet fiktivní nebo nedosažitelná místa.

Kinematografie 3D filmy jsou poměrně známou oblastí kinematografie. Plně imerzivní virtuální realita umožňuje úplné obklopení dějem. Pro pořízení těchto filmů je potřeba speciálních technologií. Alternativou k natáčení obrazu z reálného světa může být umístění uživatele v renderované scéně např. u animovaných filmů.

Zdravotnictví Rehabilitace jsou jedním z odvětví medicíny, kde virtuální realita nabízí nespočet výhod. Lidé trpící na fóbie, nebo jiné psychologické poruchy, mohou být léčení ve specializovaných prostředích, kde je maximální kontrola nad tím, jaké vjemy pacient přijímá a jak na ně

reaguje [13]. Další oblastí je prohlížení 3D snímků přirozenějším způsobem než na obvyklých monitorech nebo trénink chirurgů a studentů, aniž by došlo k ohrožení lidských životů. Virtuální realita v medicíně je stále experimentálním odvětvím, ale postup je znatelný [14].

Armáda Simulace nebezpečných situací, letecké simulátory a výukové scénáře jsou jen některé známé aplikace virtuální reality v armádě. Velkou výhodou je výcvik personálu a vojáků, aniž by bylo riskováno drahé vybavení nebo lidské životy.

Marketing Například automobilky a stavební společnosti používají virtuální realitu pro předvedení produktů svým zákazníkům. U aut si může zákazník vytvořit vlastní konfiguraci vozidla a ve virtuální realitě si ji prohlédnout. Stejně tak si lze prohlédnout návrh kuchyně nebo kanceláře a učinit tak informovanější rozhodnutí.

2.2 Grafické efekty

Grafické (vizuální) efekty jsou procedury pro úpravu vzhledu výsledné scény. Cílem je dosažení realistické reprezentace zobrazované scény, nebo dosažení estetického efektu [15]. Časté uplatnění nacházejí grafické efekty především v počítačových hrách, filmech a simulacích, kde řeší například:

- světla a stíny,
- průhlednost a odraz,
- jas a kontrast,
- vyhlazení hran,
- deformaci obrazu,
- úpravu barev (color correction),
- hloubku ostrosti (depth of field)
- a mnoho dalších oblastí.

Jaké efekty lze vytvořit, je limitováno jen fantazií autora a dostupnými technologiemi. Tato volnost a diverzita má za následek velké rozdíly ve vizuálním dopadu na scénu a v potřebném výkonu jednotlivých efektů. Často je třeba zavádět kompromisy mezi vizuální kvalitou a výkonem, aby nedocházelo ke značné degradaci realtime průběhu aplikace.

Ve virtuální realitě jsou zvýšené požadavky na výkon, kvůli vyššímu rozlišení a snímkovací frekvenci, než je tomu u srovnatelných monoskopických obrazovek. U imerzivních virtuláních prostředí je navíc kladen důraz na realističnost a tedy na vysokou kvalitu vykreslované scény.

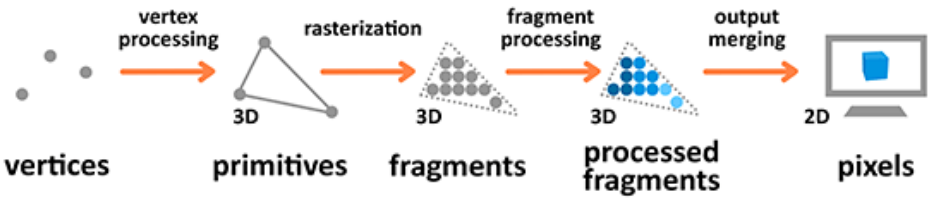
Pro realizaci grafických efektů se využívá *shaderů* právě díky možnosti paralelního vykonávání na grafických kartách (GPU) [16, 15].

2.2.1 Shader

Shadery jsou počítačové programy určené pro různé potřeby v oblasti počítačové grafiky. Shader pochází ze slovesa *to shade* (vystínovat – upravit nasvětlení). Krom stínování se shadery používají k realizaci grafických efektů (pre-processing i post-processing). Pro psaní shaderů byly vyvinuty speciální programovací jazyky např. *OpenGL Shading Language* (GLSL) a *High Level Shading Language* (HLSL) [16].

Shadery jsou spouštěny na GPU jako součást programovatelného grafického řetězce (*programmable graphics pipeline*), kdy je stejný program spuštěn na více jádrech paralelně.

Shadery se rozdělují na několik základních typů podle toho, pro kterou jednotku grafického řetězce jsou určeny. V současnosti patří mezi nejdůležitější *vertex*, *pixel* a *geometry shader*. Poslední generace grafických karet a nové verze grafických rozhraní (DirectX a OpenGL) nabízí shadery pro realizaci teselace [17]. Obecný průchod skrze graphics pipeline je ilustrován na obrázku 2.1.



Obrázek 2.1: Průchod skrze obecnou pipeline

Následující kapitoly jsou zaměřeny na několik známých (nebo často používaných) grafických efektů a jejich chování ve virtuální realitě. Probrat všechny grafické efekty je pro jejich různorodost nereálné, a tak jsem zvolil několik zástupců, ke kterým budu často referovat, konkrétně to jsou SSAO a Normal Mapping.

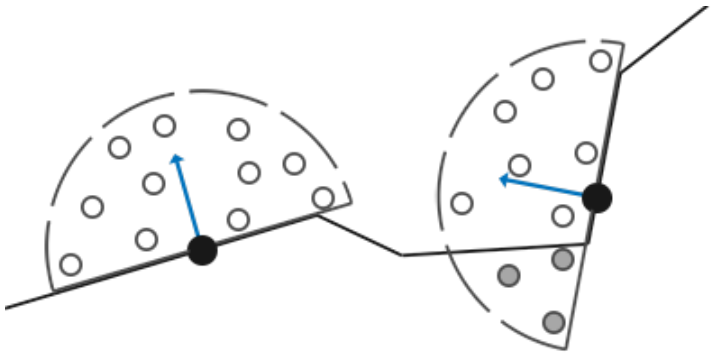
2.2.2 Ambient Occlusion

Ambient occlusion (česky známé taky jako *zastínění okolím*) je metoda, která pomáhá dodat dojem realistického, difúzního nasvícení tak, že zatemní regiony geometrie, které se nacházejí blízko jiné geometrie. Hypotéza je taková, že se světlo do těchto prostorů hůře dostane, a tedy se výsledný snímek bude jevit v těchto místech tmavší, jak by tomu bylo i v reálném světě [15, 18, 19].

Výpočet Ambient Occlusion

Algoritmů pro výpočet ambient occlusion (dále jen AO) existuje mnoho, avšak základní myšlenka je stejná. Z bodu, ve kterém chceme zjistit zastínění, je vztyčena hemisféra ve směru normály a dále je vypočteno, jak velká část hemisféry koliduje s okolní geometrií [19]. Výsledek výpočtu je často uváděn jako procento vzorků, kterým se podařilo opustit hemisféru. Konkrétní implementace je ale v rukou vývojářů.

Na obrázku 2.2 je jeden z možných přístupů k výpočtu AO. Bílé kuličky představují nezastíněné body a šedé kuličky jsou zastíněné body. U levé hemisféry je 11 nezastíněních bodů a žádný zastíněný. Výstupem je tedy hodnota 0, která značí, že tato část geometrie není zastíněna. U pravé hemisféry jsou 3 z 10 bodů zastíněné, proto je výstupem výpočtu hodnota 0.3.



Obrázek 2.2: Příklad výpočtu AO

Parametry výpočtu lze zobecnit na:

Vzorkovací frekvence algoritmu Udává pro kolik bodů geometrie bude AO vypočteno. Vysoké frekvence dávají přesnější výsledky, ale trvají déle. Nízké frekvence jsou podstatně rychlejší, ale může dojít k viditelnému šumu, který lze řešit dodatečným rozmazáním.

Počet bodů uvnitř hemisféry Parametr udává, kolik iterací je třeba provést pro výpočet konkrétního vzorku geometrie, a přímo tak negativně přispívá k celkové době výpočtu.

Rádius hemisféry Vymezuje prostor, ve kterém jsou voleny body (ať už náhodně či v definovaném vzoru) a hodnota tohoto parametru nemá vliv na rychlost výpočtu. Menší hemisféra lépe postihne detaily v okolí výpočtu než větší hemisféra při stejném počtu bodů. Větší rádius zase zohlední širší okolí.

Opět je třeba podotknout, že neexistuje jediný algoritmus a parametrizace je proto obtížná ne-li nemožná. Často je třeba s parametry experimentovat a vybrat takovou konfiguraci, která je pro zvolenou scénu (účel) nejvhodnější.

Occlusion Map

Jedním z přístupů realizace AO je aplikování tzv. okluzní mapy (*occlusion map*) do textury vykreslovaného objektu. V okluzních mapách je zastínění předpočítáno (obvykle už během vytváření modelu) a uloženo jako šedotónový obraz. Úroveň nasvětlení/zastínění části textury pak odpovídá hodnotě v okluzní mapě [20]. Výhodou tohoto přístupu je, že se mapa vypočte jednou a stane se součástí textury. Žádné další výpočty pak nejsou zapotřebí, což je vysoce efektivní přístup.

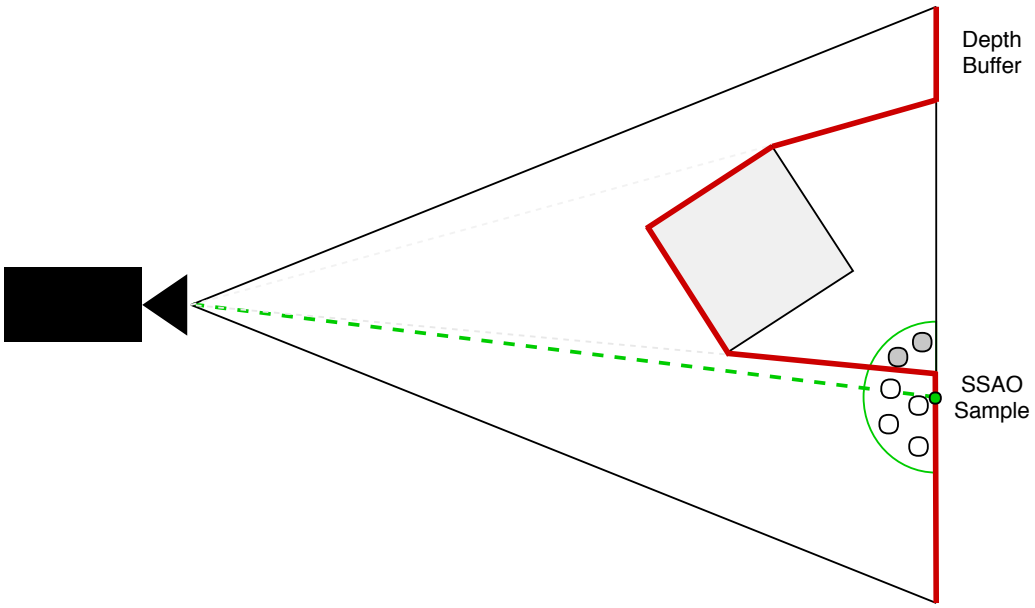
Okluzní mapy působí pouze na objekty, na kterých jsou umístěny a nijak nezasahují do okolí, což je nežádoucí. Samotný výpočet mapy je příliš náročný pro realtime aplikace, a proto je nevhodný pro scény s pohybujícími se objekty.

Screen Space Ambient Occlusion (SSAO)

Jiné řešení bylo zapotřebí, a tak v roce 2007 Vladimir Kajalin vyvinul techniku pro realtime výpočet AO, kterou nazval *Screen Space Ambient Occlusion* (SSAO).

SSAO je post-processing efekt pro výpočet zastínění pouze pro tu část scény, která je vykreslována na obrazovku (proto screen space) [19, 21]. Ve 2D obrazu ale nelze zjistit vzájemnou polohu objektů, a proto je do výpočtu zahrnut *depth buffer* (známé také jako *z-buffer*), který obsahuje informace o vzdálenosti pixelů od kamery. Jelikož algoritmus pracuje výhradně s depth bufferem, tak **nezáleží na komplexnosti scény**. Doba výpočtu AO pro každý snímek bude tedy konstantní (režijní náklady výpočtu jsou zanedbány).

Na obrázku 2.3 je ukázka využití depth bufferu pro výpočet SSAO. Počet vzorků ve standardním algoritmu je rovný počtu pixelů, které je potřeba vykreslit. Pro vyšší rozlišení může být samotný výpočet velmi náročný, a proto je využívána technika *downsampling* – výpočet probíhá na menším počtu vzorků, tedy je rychlejší, ale dochází ke ztrátě kvality vypočtené okluzní mapy. Doplňkem algoritmu také bývá rozostření vypočteného zastínění, díky kterému lze částečně zamaskovat některé nedokonalosti [21].



Obrázek 2.3: Výpočet SSAO s využitím depth bufferu

Barvy původního obrazu jsou pak vynásobeny hodnotami ve vypočtené okluzní mapě, a jas tak v zastíněných regionech klesne. Příklad použití SSAO je na obrázku 2.4 (vlevo je původní obraz, uprostřed je vypočtené zastínění a vpravo je kombinace obou předchozích snímků).

Obrázek 2.4: Použití SSAO

Mezi hlavní parametry SSAO patří:

- Intenzita
- Rádius
- Downsampling

SSAO ve virtuální realitě

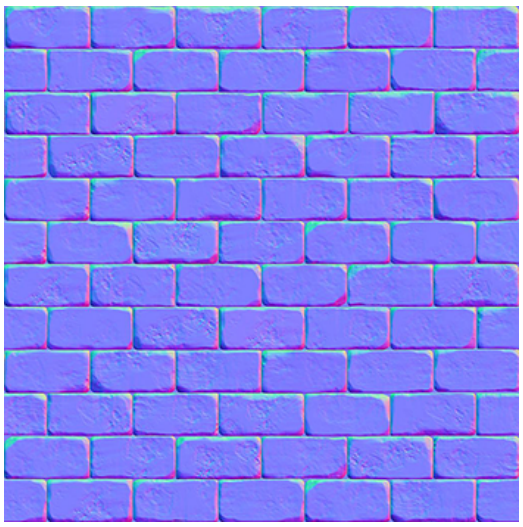
SSAO je poměrně náročné pro výpočet oproti ostatním běžným post-processing efektům. Jak bylo již řečeno, doba výpočtu závisí především na rozlišení, a to je u stereoskopických obrazovek vyšší než u srovnatelných monoskopických obrazovek. Z tohoto důvodu je často potřeba zavádět vysoký downsampling, který neodvratně vede ke snížení kvality AO.

Oproti monoskopickým monitorům vzniká nový problém. Vykreslovány jsou dva obrazy, které si musí lidský mozek spojit do jednoho. Obrazy se ve společných částech nesmí lišit jinak, než v úhlu pohledu. Právě úhel pohledu hraje v některých implementacích SSAO velkou roli. Stejná scéna může dát natolik odlišné okluzní mapy pro každé oko, že vznikne pro mozek těžko interpretovatelná situace a hrozí rizika popsaná v části 2.1.5, odstavec o zraku.

Tento problém se dá vyřešit modifikací algoritmu tak, aby se počítalo AO jen z jednoho úhlu pro obě oči, nebo zavedením společného depth bufferu pro oba snímky. Opět neexistuje jedno řešení, protože závažnost je úměrná použitému algoritmu a v některých případech se problém neprojevuje vůbec.

2.2.3 Normal Mapping

Normal mapping (NM) je v počítačové grafice technika pro vytvoření iluze prohlubní, vyboulenin a jiných detailů v povrchu, aniž by došlo ke změně jeho geometrie. Tato iluze vzniká tak, že je hlavní textura pokryta mapou, která obsahuje údaje o tom, jakým směrem je natočena normála. Opačným pohledem normálová mapa říká, z jakého směru je přijímáno světlo nehledě na skutečnou normálu geometrie. Směr je reprezentovaný jako trojrozměrný vektor, který je zakódovaný v barevných kanálech normálové mapy, kdy jednotlivé barevné složky jsou souřadnice směrového vektoru [22]. Příklad normálové mapy lze vidět na obrázku 2.5.

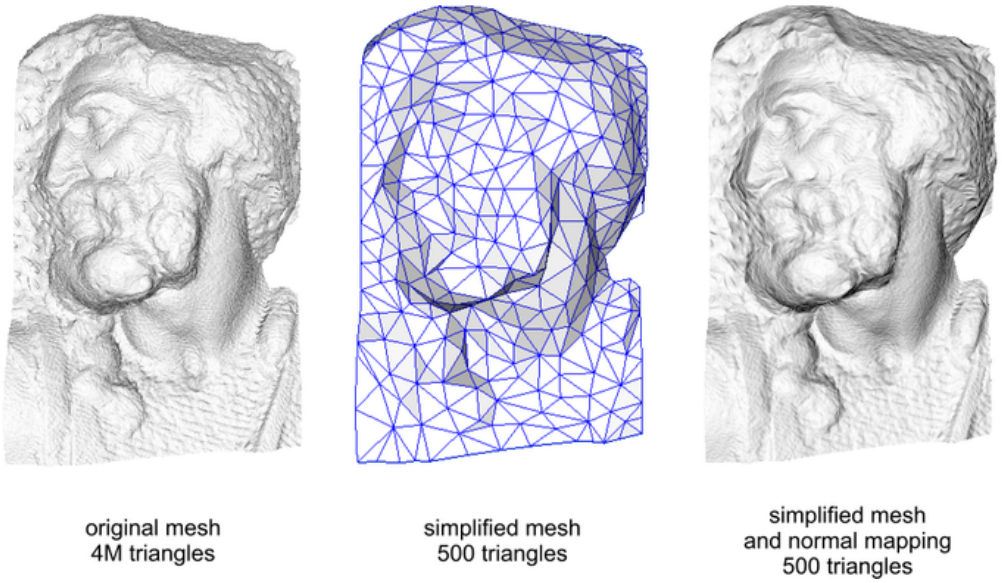


Obrázek 2.5: Ukázka normálové mapy

Normálové mapy jsou vytvářeny tak, že se z původního objektu vytvoří nějakou simplifikační metodou méně komplexní model a rozdíl těchto dvou objektů je pak uložen jako normálová mapa. Tomuto procesu se říká *normal map baking*. Ruční tvorba je poměrně složitá a praktikuje se jen zřídka.

Normálové mapování se hojně využívá u počítačových her, kde chceme používat co nejjednodušší geometrii, přesto však dostatek detailů. Stejně tak v 3D renderovaných filmech nebo scénách, jednodušší geometrie znamená kratší dobu renderingu. Svoji oblibu získaly zejména díky nízkým výpočetním nákladům a jednoduchosti použití. Na obrázku 2.6 je ilustrováno, jak lze pomocí normal mappingu snížit počet trojúhelníků v modelu ze 4 000 000 na 500 a přitom zachovat detaily. Nutno podotknout, že normálové mapování je vhodné pouze na malé detaily.

Tato technika bohužel není bezchybná. Vhledem k tomu, že nedochází k modifikaci geometrie, mohou některé pozorovací úhly iluzi degradovat



Obrázek 2.6: Využití normal mappingu k simplifikaci geometrie

až negovat. Ušetřené vrcholy geometrie jsou vyváženy nutností manipulace s další texturou a může být potřeba zavedení kompresních metod [23].

Normal Mapping ve virtuální realitě

Stereoskopie má za následek, že může být iluze degradována jednodušeji, ve virtuální realitě je patrnější, že se jedná o iluzi, právě díky lepšímu vnímání hloubky. Řešením je použití techniky *Parallax Mapping* nebo *Displacement Mapping*.

Parallax mapping (PM) využívá výškové mapy pro realističtější reprezentaci hloubky a nezávislost na různých úhlech pohledu [24]. Algoritmus také realizuje sebestínování pro posílení iluze. Parallax mapping je vhodná alternativa, není-li normal mapping dostatečně kvalitní.

Displacement mapping (DM) mění pozice vrcholů modelu pro vytvoření detailů – už se nejedná o iluzi. Z tohoto důvodu je potřeba, aby měl objekt dostatečně jemnou mřížku, a tím znovu vznikne problém, který má normal mapping řešit, a sice vysoký počet trojúhelníků na model [22]. Displacement mapping není vhodný pro virtuální realitu, pokud je cílem redukce výpočetních nákladů.

Mimo oslabení iluze nejeví normálové mapování žádné jiné problémy, než u monoskopických aplikací.

2.2.4 Problémy shaderů u stereoskopických zobrazení

U stereoskopických a asymetrických zobrazení existují specifické problémy pří výkonu kódu shaderů [25]:

Závislost na úhlu pohledu Tento problém se týká shaderů, které ve svých výpočtech zohledňují pozici kamery. Ve virtuální realitě nejsou oči nikdy na stejném místě, a proto se nelze vyhnout vykreslování ze dvou úhlů pohledu. Při tvorbě grafických efektů je potřeba toto omezení zhodnotit a provést příslušná opatření.

Návaznost ve spojích U víceobrazovkových virtuálních realit (např. CAVE) musí hrany (spoje) obrazovek navazovat. Je-li aplikován grafický efekt, který deformuje obraz (např. *Fisheye*, *Twirl*), dojde k porušení návaznosti a ztrátě imerze.

Zvýšenou pozornost je třeba věnovat při tvorbě efektů, které:

1. jsou výpočetně náročné,
2. jsou závislé na úhlu pohledu,
3. deformují obraz,
4. tvoří patrné rozdíly mezi pravým a levým okem,
5. blikají.

Některé efekty, jako např. korekce barev, motion blur, antialiasing, vykazují stejné chování ve virtuální realitě, jako mimo ni a dodatečné úpravy algoritmu nejsou potřeba [26].

V tabulce 2.1 lze vidět výsledky experimentů provedených společností MiddleVR [25], kde cílem bylo otestovat základní sadu post-processing efektů, která je součástí herního enginu *Unity3D Pro*. Tyto efekty nebyly nijak upravovány pro stereoskopické zobrazení a výsledky potvrzují výše uvedená kritéria.

Tabulka 2.1: Chování post-processing efektů ve virtuální realitě – *MiddleVR*

Efekt	Stereo OK	Návaznost ve spojích
Antialiasing (Post-processing)	OK	OK
Bloom	OK	Problémové
Bloom a Lens Flare	OK	Problémové
Blur	OK	Problémové
Motion Blur (camera)	OK	Problémové
Color Correction	OK	OK
Contrast Enhance	OK	Problémové
Contrast Stretch	OK	Problémové
Crease	OK	Problémové
Depth of Field	OK	Problémové
Depth of Field (Scatter)	OK	Problémové
Edge detect	Problémové	OK
Edge detect (normals)	OK	OK
Fisheye	Problémové	Problémové
Global Fog	OK	OK
Glow	OK	Problémové
Grayscale	OK	OK
Motion Blur	OK	OK
Noise / Grain	OK	Problémové
Noise	OK	Problémové
Screen Overlay	Problémové	Problémové
Sepia Tone	OK	OK
SSAO	OK	Problémové
Sun shafts	Problémové	Problémové
Tilt Shift	OK	Problémové
Tonemapping	OK	OK
Twirl	Problémové	Problémové
Vignette	OK	Problémové
Vortex	Problémové	Problémové

2.3 Lidské vnímání

Aby bylo možné efektivně a objektivně navrhovat experimenty, je potřeba porozumět lidskému vnímání vizuálních stimulů. Ačkoliv se jedná o významně subjektivní téma, lze ho do určité míry parametrizovat a modelovat. Jedním z modelů lidského vizuálního systému je HVS (*Human Visual System*) model.

2.3.1 HVS model

HVS model je používán pro práci s biologickými a psychologickými procesy lidského zraku. Tento model také slouží ke zjednodušení komplexního chování lidského zrakového systému, k potlačení nedokonalostí, nebo ke zdůraznění některých zrakových aspektů. Parametry týkající se HVS:

Contrast Sensitivity Function (CSF) Metrika pro zjištění schopnosti rozlišovat mezi různými úrovněmi kontrastu ve statickém obrazu. CSF udává, do kdy je pozorovatel schopen určit rozdíl mezi kontrastními stimuly se zvyšující se frekvencí střídání. Od určité hranice se různě kontrastní stimuly spojí v jeden. S narůstající znalosti HVS se postupně zvyšovala komplexnost CSF a vzniklo mnoho nových modelů.

Rozlišovací hranice se stala předmětem výzkumu nejen pro kontrastní stimuly, ale i pro chromatické. Opět je cílem nalézt hranici, kdy pozorovatel nerozliší dva podobné (přesto však rozdílné) stimuly. Toto je často využívaná metrika při provádění subjektivních testů týkajících se kvality obrazu. Ukázalo se však, že lidský zrak je více citlivý na změny jasu než barev [27].

Vizuální maskování Vnímání jednoho stimulu může být potlačeno přítomností jiného stimulu. Maskování je nejvíce patrné, když stimuly sdílí pozici, rotaci a frekvenci.

2.3.2 Pozorovatelé

Pozorovatele dělíme na experty a laiky v závislosti na jejich znalosti testované domény. Expert je takový pozorovatel, který má zkušenosti z oblasti obrazových artefaktů, které se mohou projevit během testu. Laik (nebo také „naivní“ pozorovatel) nemá žádné znalosti z domény obrazových artefaktů. Experty je vhodné využívat při konstrukci experimentů, avšak pro jejich provádění je výhodnější využít pozorovatelů, kteří neznají technické pozadí a detaily prováděných testů.

2.4 Provádění subjektivních testů

Subjektivní testování v oblasti počítačové grafiky je založeno na hodnoceních, která dávají pozorovatelé (subjekty) testovaným vizuálním stimulům. Experimenty jsou určeny pro hodnocení změn kvality, které vznikly aplikací modifikačních technik (komprese, post-processing, ...) .

Lidé jsou koncovým příjemcem obrazových a zvukových dat, proto je subjektivní testování nejpřesnějším a nejspolehlivějším prostředkem pro hodnocení kvality [28, 29]. Nicméně, oproti objektivnímu testování, je třeba čelit následujícím problémům:

- nejsou použitelné u realtime aplikací,
- výsledky jsou zatížené fyzickým a psychickým rozpoložením subjektu,
- výsledky jsou závislé na pozorovacích podmínkách,
- jsou časové náročné a drahé.

Provádění subjektivních experimentů (nejen v počítačové grafice) je poměrně dobře zmapovaná tématika. S rozvojem multimediálních technologií se začal objevovat problém s velikostí přenášených dat, a proto se začaly zavádět různé kompresní techniky. Ztrátová komprese s sebou nese cenu degradace kvality přenášeného média za účelem snížit jeho velikost (náklady na přenos). Prováděním experimentů lze získat potřebná data pro úpravu algoritmů na takovou úroveň, aby bylo možné co nejvíce omezit přenosové náklady a zároveň mít co nejmenší možný negativní dopad na kvalitu média [28, 30].

Mezinárodně uznávaná společnost **ITU** (International Telecommunication Union) se zabývá právě standardizací v oblasti telekomunikace. ITU proto vytvořila sadu metodik pro návrh a provádění subjektivních experimentů a hodnocení kvality vizuálních a zvukových stimulů [29]. Tyto metodiky dávají tzv *doporučení* jako např.:

- časové kvantum a místo provádění,
- výběr a počet subjektů,
- způsob prezentace stimulů,
- doby vystavení stimulu, hlasování a odpočinku
- absolutní nebo relativní hodnocení,
- rozsah hodnotící škály.

2.4.1 Druhy stimulů (signálů)

Během testu je pozorovatel postaven před sérii stimulů, které mohou být:

- Referenční (také originální)
- Modifikované (*impaired*, *processed*)

Subjekt může být s přítomností originálu seznámen. Na základě tohoto vědomí pak dělíme testy na do tří kategorií:

S úplnou referencí Originální stimul je v testovacích datech přítomen a subjekt po celou dobu experimentu ví, který to je.

Se skrytou referencí Originální stimul je opět součástí testovacích dat, avšak subjekt není s jeho přítomností obeznámen.

Bez reference Referenční stimul není v datech přítomen.

2.4.2 Metriky hodnocení stimulů

Subjekt během experimentu ohodnocuje prezentované stimuly podle hodnotící stupnice (obvykle v bodovém rozmezí 1 až 5 od nejhoršího po nejlepší). První metrikou je průměr z uvedených bodových ohodnocení – MOS (*Mean Opinion Score*).

$$\mathrm{MOS} = \frac{\sum_{n=1}^{n} R_n}{N} \tag{2.1}$$

Kde R je hodnocení stimulu n z množiny testovacích dat N. Pokud je v testovacích datech přítomen referenční stimul, je možné zavést další metriky, jako DMOS (*Difference MOS*) nebo CMOS (*Comparison MOS*). DMOS je rozdíl mezi referenčním a modifikovaným stimulem. Výpočet DMOS:

$$DMOS = MOS(I_{proc}) - MOS(I_{ref}) + max \tag{2.2}$$

kde I_{proc} je modifikovaný stimul, I_{ref} je referenční stimul a max je nejvyšší udělitelný počet bodů na hodnotící škále. U CMOS se hodnotí podobnost s referenčním stimulem. Pro porovnání jednotlivých hodnocení bylo zavedeno standardní skóre (*Z-score*). Z-score pro i-tého pozorovatele a j-tý stimul vypočteme podle vzorce:

$$z_{ij} = \frac{DMOS_{i,j} - \overline{d_l}}{\sigma_i} \tag{2.3}$$

kde $\overline{d_l}$ je střední hodnota DMOS a σ_i je standardní odchylka. Obě hodnoty jsou vypočteny přes stimuly i-tého pozorovatele.

2.4.3 Hodnotící škály

Hodnotící škály vymezují prostor odpovědí pro daný experiment. Mezi nejčastěji používané škály patří:

Grading scale Subjekt vybírá z několika diskrétních možností (nejčastěji z pěti). Příklady diskrétních škál viz tabulka 2.2.

Tabulka 2.2: Příklady možných diskrétních škál

Body	Kvalita	Pohodlí	Degradace obrazu
5	Výborná	Velmi pohodlné	Neznatelná
4	Dobrá	Pohodlné	Tolerovatelná
3	Dostačující	Normální	Trochu nepříjemná
2	Slabá	Nepohodlné	Nepříjemná
1	Špatná	Velmi nepoholné	Velmi nepříjemná

Continuos scale Subjekt vybírá odpověď na určeném intervalu (často posuvník). Odpovědi jsou pak namapovány na diskrétní množinu pro lepší statistické zpracování.

Comparison scale Porovnávací škála – subjekt porovná modifikovaný stimul s referenčním a jako hodnocení uvede rozdíl od referenčního (viz tabulka 2.3).

Tabulka 2.3: Příklad porovnávací škály

Hodnota	Slovně
-3	Mnohem horší
-2	Horší
-1	Nepatrně horší
0	Stejné
1	Nepatrně lepší
2	Lepší
3	Mnohem lepší

2.4.4 Metody provádění experimentů

Metody provádění subjektivních testů pro hodnocení kvality vizuálních stimulů dělíme na experimenty s jedním stimulem a dvěma stimuly. Následující příklady vychází z ITU-R BT.500-13 [31], ITU-T P.910 [32], ITU-R BT.1788 [33] a dalších doporučení.

Metody s jedním stimulem

ACR (Absolute Category Rating) Intuitivní přístup k provádění experimentu, kdy je pozorovateli postupně předloženo několik modifikovaných stimulů, a cílem je ohodnotit kvalitu na diskrétní škále – snadný výpočet MOS.

ACR-HR (Absolute Category Rating with Hidden Reference) Stejné jako předchozí metoda, jen je do testovacích dat přidán originální stimul, na který není pozorovatel nijak upozorněn (proto *hidden* - skrytý).

SSCQE (Single Stimulus Continuous Quality Evaluation) Tato metoda je vhodná pro hodnocení déle trvajících videí. Subjekt během vystavení stimulu hodnotí kvalitu pomocí posuvníku s definovanou škálou. Různé úseky média totiž mohou podléhat různým modifikacím, a tak je dynamické hodnocení přesnější. Častou kritikou této metody je absence reference, načež vznikla metoda DSCQS, popsaná níže.

SAMVIQ (Subjective Assessment of Multimedia Video Quality) Metoda popsána v ITU-R BT.1788 ITU [33] byla navržena specificky pro multimediální obsah. Pořadí stimulů, hodnocení a počet opakování je na uvážení pozorovatele. Testovací data musí obsahovat referenční stimul ať už explicitně, nebo skrytě.

Metody se dvěma a více stimuly

DSIS (Double Stimulus Impairment Scale) Pozorovateli je zobrazen originální stimul, poté modifikovaný stimul a následně je dotázán, jak moc nepříjemný vliv měla modifikace na kvalitu. O pořadí a typu stimulů je subjekt informován předem. Tato metoda je vhodná pro porovnání videí, kde je velký rozdíl v kvalitě (např. vysoká komprese).

DSCQS (Double Stimulus Continuous Quality Scale) Pozorovateli je představen originální stimul a modifikovaný stimul (v náhodném pořadí s možností jednoho opakování). Následně je vyzván k ohodnocení obou stimulů a vzniklý rozdíl je metrikou pro změnu kvality. Pozorovateli není řečeno, který stimul je referenční.

DCR (Degradation Category Rating) Pozorovateli je zobrazen referenční stimul a následně (po 2s pauze) modifikovaný stimul. Přítomnost reference je pozorovateli známá. Hodnotí se odlišnost od reference.

Binary Forced Choice Tato metoda využívá tří stimulů (jeden referenční a dva modifikované) a subjekt musí z modifikovaných vybrat ten, který je více podobný referenčnímu. Modifikované stimuly mohou být totožné, nebo dokonce obsahovat originál. Tato metoda je vhodná, pokud je cílem získat relativní kvalitu.

2.5 Vyhodnocování kvality stereoskopických scén

Stereoskopické zobrazovaní je vykreslování dvou obrazů stejné scény ze dvou horizontálně srovnaných úhlů pohledu. Zobrazované scény mohou být reálné (natočené speciálními 3D kamerami), nebo syntetické (generované počítačem, CGI). Úhel, který svírají přímky vedené ze dvou různých míst v prostoru k pozorovanému bodu se nazývá **paralaxa**. Vnímání hloubky závisí na velikosti paralaxy. Pokud je hodnota nízká, bude se vnímaná scéna jevit jako plochý obraz. Pokud je hodnota příliš vysoká, lidský mozek nebude schopný scénu vnímat [28]. V závislosti na velikosti paralaxy lze vnímat tři jevy:

Pozitivní paralaxa Objekty vnímáme za obrazovkou. Lze přirovnat k pozorování objektů za sklem.

Negativní paralaxa Objekty vnímáme před obrazovkou.

Nulová paralaxa Objekty vnímáme přesně na obrazovce.

Během provádění experimentů je vhodné testované objekty umisťovat do tzv. *komfortní zóny*. Komfortní zóna je mezi ± 0,2D (dioptrie) pro negativní paralaxu a ± 0,3D pro pozitivní paralaxu [34], což bylo stanoveno na základě hloubky ostrosti lidského oka. Objekty mimo komfortní zónu mohou být příčinou některých problémů popsaných v části 2.1.5, především se jedná o nevolnost a oční iritaci.

Další faktory, které jsou specifické pro stereoskopické zobrazovací systémy:

- rozlišení ve směru hloubky,
- pohyb ve směru hloubky,
- vnímané objekty jsou nepřirozeně velké/malé,
- vnímané objekty jsou nepřirozeně ploché (*cardboard effect*),

- nesouosost kamer,
- objekt vnímaný v negativní paralaxe je oříznut hranou obrazovky,
- prolínání obrazů - části jednoho kanálu jsou promítnuty v druhém kanálu.

Tyto faktory mají dopad na výslednou kvalitu stereoskopických materiálů. Dříve zmíněná společnost ITU vydala doporučení pro provádění subjektivních testů zaměřených na obrazovou kvalitu, hloubkovou kvalitu a vizuální pohodlí 3D projekcí [34]. Metody pro provádění experimentů byly převzaty z 2D (viz 2.4), a lze se tedy řídit doporučeními uvedenými výše.

Dobu testování je třeba upravit podle stupně nepohodlí testovaného materiálu např. obsahuje-li médium nepříjemné prvky (rychle se pohybující objekty, vysoká paralaxa, apod.). jako vhodná škála pro hodnocení může být škála pohodlí (viz tabulka 2.2) [28].

3 Cíle projektu

K úspěšnému splnění projektu byly stanoveny tyto cíle:

3.1 Prostředí pro tvorbu a provádění experimentů

Cílem je vytvořit prostředí pro provádění subjektivních experimentů pro hodnocení kvality renderovaných scén ve virtuální realitě. Doplňujícím požadavkem je možnost provádění podobných experimentů mimo virtuální realitu. Uživatel aplikace bude schopen testy nejen provádět, ale i tvořit nové.

Jako modelové experimenty bude připraveno testování efektů *normal mapping* (reprezentant experimentů s objektovými shadery) a *SSAO* (reprezentant post-processing efektů).

3.2 Pilotní studie

Po vyhotovení aplikace bude provedena pilotní studie na malém vzorku subjektů (laiků). Cílem je ověřit funkčnost testů a získat zpětnou vazbu ohledně provádění, na jejímž základě bude aplikace upravena.

3.3 Zhodnocení dosažených výsledků

Kriticky zhodnotit, zda je aplikace použitelná pro provádění subjektivních experimentů a zda jsou výsledky vhodné pro statistické zpracování. Součástí budou i doporučení pro vylepšení a další vývoj.

4 Požadavky na vypracování

Kromě cílů projektu byly vzneseny i konkrétní požadavky na funkcionalitu aplikace. Požadavky jsou rozděleny do kategorií podle důležitosti jejich naplnění.

Kritické požadavky:

- Možnost provádění subjektivních testů k hodnocení kvality renderovaných scén ve virtuální realitě.
- Úprava a tvoření zmíněných testů.
- Export získaných dat pro následné analytické zpracování.

Doplňující požadavky:

- Nahrávání, export a přehrávání průběhu experimentu.
- Možnost provádění experimentů i mimo virtuální realitu.
- Vstupní a výstupní soubory strukturované pro strojové zpracování.
- Vytvořit intuitivní ovládání (popř. návod), aby mohly být experimenty prováděny bez dozoru.
- Bude-li to možné, pokusit se aplikaci rozšířit na co nejvíce platforem. S tímto požadavkem souvisí i vhodné navržení uživatelského rozhraní a ovládání. Vzhledem k různorodosti hardware pro virtuální realitu budou interakce zjednodušeny, aby bylo možné pokrýt i platformy jako Gear VR.

Vzhledem k rozmanitosti grafických efektů jsem musel rozdělit provádění experimentů do dvou kategorií:

Objektové efekty Jedná se o grafické efekty, které jsou specifické pro objekt, na kterém jsou umístěny např. normal mapping, průhlednost, voda, částicové efekty, atd.

Post-processing efekty Tyto efekty jsou aplikovány na připravený snímek před jeho zobrazením např. antialiasing, screen space ambient occlusion, motion blur, bloom, atd.

5 Scénáře užití

V návrhu je potřeba stanovit jak a kým se bude aplikace používat. Následuje analýza uživatelských rolí a jejich scénářů užití. Ve vztahu k této aplikaci jsou definovány tři role: testovaný (také pozorovatel nebo subjekt), testující a analytik.

Případy užití – Testovaný

Provádění experimentu Testovaný prohlíží testované scény a odpovídá na kladené dotazy uvnitř aplikace.

Případy užití – Testující

Příprava experimentů Navrhuje a vytváří experimenty prostřednictvím konfiguračních souborů, popř. uvnitř editoru herního enginu.

Provádění experimentu Může být přítomen u vykonávání testu jako technická podpora, nebo pro sběr statistických dat o používání aplikace, neměl by se ale zapojovat do samotného průběhu, aby neovlivnil výsledky.

Prohlížení výsledků Testující také může prohlížet nahrávky experimentů a případně vyřadit ty, které nejsou, dle jeho soudu, dostatečně kvalitní pro zpracování analytikem.

Případy užití – Analytik

Prohlížení výsledků Může prohlížet výstupní data a záznamy (jsou-li dostupné)

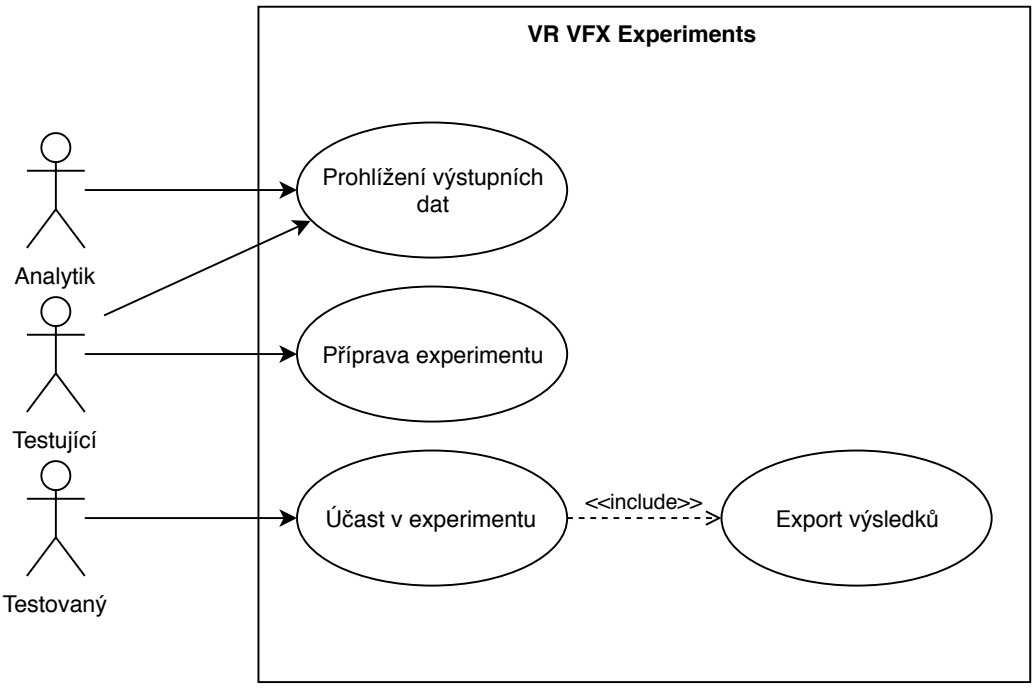
Vyvozování závěrů Z nasbíraných dat určí výsledek experimentu (stanovení hranic pro grafické efekty, vyřazení efektu pro nedostatečný vliv na kvalitu nebo pro příliš vysoké nároky na výkon)

5.1 Shrnutí

U testujících jejich používání aplikace spočívá především v manipulaci s daty doplněnou o možnost zaškolování subjektů (je-li to potřeba) a prohlížení nasbíraných dat. Testující a analytik mohou být jedna osoba, aby nedošlo k misinterpretaci účelu experimentů.

Testované (pokud možno) je třeba vybírat tak, aby neměli znalost testované domény, jinak budou výsledky experimentu zatížené určitou mírou předpojatosti (*bias*). Předpokládá se, že testovaný bude mít nějakou znalost virtuální reality (minimálně používání headsetu). V případě testovaných, kteří nemají žádné zkušenosti s virtuální realitou, bude experiment a proškolení probíhat s asistencí testujícího.

Na obrázku 5.1 je jednoduchý use case diagram navrhované aplikace.



Obrázek 5.1: Diagram případů užití

6 Návrh prostředí pro vytváření experimentů

Před vyhotovením aplikace je potřeba rozhodnout o implementaci následujících částí:

Hardwarová platforma Zařízení, která zprostředkovávají zážitek ve virtuální realitě.

Grafický engine Softwarové řešení pro vykreslování testovaných efektů, které umožní provádění experimentů ve virtuální realitě i mimo ni.

Styl provádění experimentu Jaké druhy testů provádět a jak. Vzhledem k tomu, že jsou grafické efekty velmi různorodé, je potřeba vytvořit obecnější prostředí, kde využijeme jejich společných vlastností k experimentům.

Tvorba a úprava experimentů Způsob, jakým lze vytvářet nové, či upravovat stávající experimenty.

Interakce s aplikací Možnosti ovládání pro různé hardwarové platformy. Zahrnuje možnosti pohybu v testovacích scénách i možnost zodpovězení otázek spojených s experimentem.

Formát vstupních dat Soubor formálně popisující detaily experimentu (testované efekty, druh experimentu, otázky na subjekt, atp.).

Formát výstupních dat Soubor odpovědí na otázky kladené během experimentu.

Formát záznamu experimentu Soubor obsahující nahrávku průběhu experimentu.

6.1 Hardwarová platforma

Virtuální realita stále nabírá na popularitě a na trhu se objevují nová zařízení. Jednotlivé společnosti naslouchají zpětné vazbě uživatelů, a stále tak přichází s novými modely a vylepšeními. Z tohoto důvodu existuje velký rozptyl na úrovni obrazové kvality i na úrovni interakce se systémem, a proto jsem se rozhodl udělat stručný výčet aktuálně používaných a populárních zařízení a jejich charakteristik. Na konci bude uvedena tabulka s číselným srovnáním některých parametrů jednotlivých zařízení.

Oculus Rift DK1 (PC) Jedná se o první headset společnosti Oculus vydaný v roce 2013. DK1 (*Development Kit 1*) označuje, že se jedná o vývojovou verzi. Vývojové verze představovaly zdroj důležité zpětné vazby a finančních prostředků pro další vývoj. Navzdory malému rozlišení a obnovovací frekvenci byl Rift přijat velmi dobře a společnost Oculus se mohla věnovat dalšímu vývoji.

Oculus Rift DK2 (PC) Vydán v roce 2014. Oproti předchozí verzi nabízí DK2 větší rozlišení a obnovovací frekvenci. Nově také přibyl infračervený senzor pro snímání pozice (dříve jen rotace). Přestože se stále jedná o vývojovou verzi, nabízí Rift DK2 plnohodnotný zážitek ve virtuální realitě. Ovládání je však stále omezeno na periferie, které lze používat i bez virtuální reality (klávesnice, myš, gamepad, ...).

Oculus Rift CV (PC) Verze CV (*Consumer version*) vyšla v roce 2016. Od předchozí verze Rift disponuje 90Hz obnovovací frekvencí, vyšším rozlišením, integrovanou zvukotechnikou. Pohybové ovladače Oculus Touch se staly součástí balení až o několik měsíců později. Ke snímání Rift používá dva prostorové snímače (případně tři, pokud je vyžadováno spolehlivé 360° snímání). V současnosti se jedná o nejpopulárnější hardware pro virtuální realitu na platformě Steam, kde tvoří přes 47 % z používaných systémů pro VR.

HTC Vive (PC) Vive je komerčně dostupný od roku 2016 a spolu s Oculus Rift CV má největší podíl na trhu s VR zařízeními. Od Riftu se liší především několikanásobně vyšší hrací plochou díky lighthouse technologii. Snímače pohybu jsou umístěny v opačných rozích místnosti a tak nemá Vive problém s 360° snímáním a tzv. *room-scale experience* (možnost pohybu v oblasti až 21m^2). Samotný headset je o 20 % těžší než Rift a pohybové ovladače jsou téměř dvakrát větší. Na platformě Steam zaujímá druhé místo se 45 % z používaných systémů.

Vive Pro (PC) *Pro* verze je dostupná od roku 2018. Jedná se pouze o headset (tedy bez snímačů a ovladačů), ale je zpětně kompatibilní s hardwarem HTC Vive. Primárně je určený pro marketingové účely (automobilky, realitní kanceláře) a pro nadšence do virtuální reality. Oproti svému předkovi nabízí větší rozlišení, AMOLED display a až 100m^2 snímané plochy (až s příchodem lighthouse 2 technologie). Častou kritikou je vysoká cena, která překračuje cenu HTC Vive, které je včetně snímačů a ovladačů.

Windows Mixed Reality (PC) Společnost Microsoft v roce 2017 uvedla jejich novou herní platformu – Windows Mixed Reality (WMR). Konkurovat se snaží především vysokým rozlišením na oko (1440×1440 px) a Plug and Play přístupem. Vyšší rozlišení je vyváženo použitím LCD, které jsou často odsuzovány pro jejich vysokou persistenci (zvláště při použití ve VR). Na rozdíl od předchozích systémů, kdy název popisoval jeden headset, jedná se zde o platformu, pro kterou existuje několik headsetů různých výrobců (Acer, Samsung, apod.).

Pozn.: WMR je poněkud zavádějící název, protože Mixed Reality znamená, že virtuální svět a reálný svět koexistují a i spolu interagují, nic z toho však není podporováno ani jedním z dostupných headsetů. Zařízení zprostředkovávají virtuální realitu a WMR je jen název platformy, kde se snad v budoucnu dočkáme i opravdových AR (Augmented Reality) a MR (Mixed Reality) zařízení.

Pimax 8K (PC) Headset od čínské společnosti, který jako první na trhu uvádí rozlišení 8k a pozorovací úhly až 200°. U standardní verze headsetu je uvedené 8k rozlišení dosaženo dvěma 3840×2160 obrazovkami, které jen zvětšují 2560×1440 renderovaný obsah, jedná se tedy spíše o marketingový tah, než skutečné hodnoty. Přesto je však screen door efekt značně potlačen a vysoké FOV (Field of view – úhel pohledu) dodává mnohem přirozenější periferní vidění. Pimax je kompatibilní s lighthouse technologií společnosti HTC Vive a i s jejich ovladači. Headset dostává mnoho kritiky kvůli nevhodně upraveným čočkám pro tak vysoké pozorovací úhly, a také pro použití CLPL (Customized low persistance liquid) displaye, jehož odezva je podstatně horší než u OLED a dochází k nepříjemnému rozmazávání.

Playstation VR (PS4) Headset vytvořený pro herní konzoli Playstation 4 je levnou alternativou k HTC Vive a Oculus Rift. Pohodlné upevnění na hlavu a odklápěcí display jsou hlavní přednosti headsetu. Snímání probíhá za pomoci světel umístěných na headsetu a může dojít

k výpadku při nevhodném natočení. Pro ovládání se používá *Sony DualShock 4* ovladač nebo *Playstation Move Motion* (vytvořený pro Playstation 3 v roce 2010). Ovladače jsou snímány stejně jako headset a při zastínění dochází k výpadkům snímání. Rozlišení headsetu 960×1080 na oko je dostatečné.

Samsung Gear VR (Android) Headset, vydaný roku 2015 a jeho následník z roku 2017 společností Samsung, představují platformu pro majitele vybraných chytrých telefonů stejné společnosti. Headset obsahuje posuvné čočky, gyroskop, touchpad a tlačítko zpět. Výkon, kvalita projekce a výdrž je závislá na použitém zařízení. Od headsetu nevedou žádné kabely, a je tak snadno přenositelný/skladovatelný. Důležité je podotknout, že pro svoji nízkou cenu je Gear VR velmi rozšířenou plaformou na poli virtuální reality, ale zároveň se jedná o vcelku uzavřenou kategorii, protože výkonový rozdíl oproti ostatním systémům je vysoký a aplikace jsou vyvíjeny cíleně.

Google Cardboard (mobilní telefony) Cardboard je z kartonu vyrobený držák na telefon, který má dvě čočky a je schopen vytvořit 3D zážitek na specificky upravených aplikacích. Cardboard je velmi levný způsob 3D projekce. Veškeré vlastnosti jsou závislé na typu použitého telefonu a aplikace.

V tabulce 6.1 je uvedeno zkrácené srovnání jednotlivých zařízení. Tabulka ani výčet nejsou kompletní, protože se trh s virtuální realitou se stále vyvíjí.

Tabulka 6.1: Srovnání vybraných headsetů

Zařízení	Display	Rozlišení (na oko)	Frekvence	FOV	Hmotnost
Oculus Rift CV	OLED	1080×1200	90 Hz	110°	470g
HTC Vive	OLED	1080×1200	90 Hz	110°	563g
Vive Pro	AMOLED	1400×1600	90 Hz	110°	510g
Windows MR	LCD	1440×1440	90 Hz	105°	450-650g
Pimax 8K	CLPL	3840×2160	80 Hz	200°	453-700g
Playstation VR	OLED	960×1080	120 Hz	100°	610g
Gear VR	– Závislé na použitém zařízení –				

Pro vývoj byly zvoleny platformy HTC Vive a Oculus Rift pro jejich popularitu, podíl v herním odvětví a dostupnost v univerzitním prostředí (více o podporovaných platformách viz 6.5).

6.2 Grafické a herní enginy

Grafické enginy jsou frameworky pro vykreslování informací (často 2D a 3D prvky) a následné zobrazení. Tyto enginy tvoří rozhraní pro nízkoúrovňová volání OpenGL a DirectX. Součástí může být i překlad a optimalizace ručně psaných shaderů.

Nadstavbou nad grafické enginy jsou **herní enginy**. Krom vykreslování poskytují herní enginy obsluhu vstupně výstupních zařízení, rutiny pro práci se zvukem, simulátor fyziky, abstrakci některý často užívaných herních prvků a síťové služby.

Použitím herního enginu je částečně určena architektura aplikace, nicméně cílem projektu je provádět experimenty pro nalezení hranic parametrů grafických efektů tak, aby se podle nich dal optimalizovat výkon zejména her a simulací. Z tohoto důvodu je použití herního enginu vysoce výhodné.

Následuje výčet možných herních enginů pro tuto práci. Ze seznamu jsem odstranil herní enginy, které nemají podporu virtuální reality, jelikož je to podmiňující požadavek. Dále jsem musel vynechat enginy, které jsou z nějakého důvodu nedostupné (např. proprietární enginy) nebo příliš drahé.

Unreal Engine 4 (UE4) je jedním z nejznámější herních enginů s velmi dobrou pověstí zejména proto, že disponuje prvotřídními grafickými možnostmi. Hlavním programovacím jazykem v rámci UE4 je C++ a pro neprogramátory existuje systém „blueprintů“, kde lze herní logiku popisovat graficky. Dříve byl prodáván do AAA společností za řádově miliony dolarů. Změna herního průmyslu však donutila vývojáře přehodnotit svůj obchodní model a UE4 je dnes zdarma s 5% *royalty* poplatky (část z výdělku hry). Vývojáři enginu aktivně podporují vývoj pro virtuální realitu, dokonce i společnost Oculus s UE4 uzavřela dohodu, že pokryje případné royalty poplatky.

CryEngine Společnost Crytek vytvořila CryEngine pro jejich první titul Far Cry. S grafickou kvalitou je na tom CryEngine srovnatelně jako Unreal Engine. Učící křivka je poněkud strmější než u konkurenčních enginů kvůli poměrně neintuitivnímu uživatelskému rozhraní. Dnes je dostupná verze *CryEngine V*, která má aplikovaný platební model *pay what you want*, kdy si uživatel vybírá, kolik je ochoten za používání platit. CryEngine nyní také poskytuje podporu pro virtuální realitu.

Godot 3 Herní engine Godot je zcela zdarma, od pořízení po distribuci her. Nechybí podpora 2D, 3D ani distribuce pro PC, Mac, Linux, Android, iOS, Windows Phone a Blackberry. Po grafické stránce je srovnatelný

s Unity (viz dále). Hlavním programovacím jazykem je *GDScript* (jazyk na bázi Pythonu), ale některé moduly lze psát v C++. Godot 3 podporuje populární zařízení pro virtuální realitu např. Oculus Rift, HTC Vive, Windows MR i mobilní varianty a další.

Unity 3D Tento herní engine nabízí širokou paletu funkcí pro libovolné typy her a vcelku intuitivní ovládání (oproti výše zmíněným). Díky aktivní a široké komunitě je i učící křivka přívětivější pro nové vývojáře. Unity staví především na podpoře a snadné distribuci pro mnoho platforem např. PC, Android, iOS, Playstation, Xbox, a dokonce i prohlížečové hry (Unity Web Player). Začínající vývojáři také ocení bezplatnou verzi a *asset store* (platforma pro nákup herních prvků). Unity poskytuje nativně podporu pro Oculus Rift, OpenVR, PlayStation VR, Google VR (Daydream i Cardboard), HoloLens a všechna Windows Mixed Reality zařízení.

Pro vývoj této aplikace jsem zvolil herní engine Unity 3D. Unreal Engine byl druhým nejlepším kandidátem právě díky mnohem lepším možnostem grafických efektů, nicméně popularita Unity, snadná distribuce na více platforem a částečně i fakt, že mám s Unity zkušenosti, převážily v jeho prospěch. Osobně si také myslím, že takto přinese aplikace užitek většímu počtu lidí.

6.3 Tvorba a úprava experimentů

Experimenty mohou být zaměřeny na různé grafické efekty a právě jejich různorodost vyžaduje scény specificky upravené pro pozorování daného efektu. V aplikaci je také umožněno používat vlastní efekty (shadery), a tedy je prakticky nemožné vytvořit univerzální sadu testovacích místností. Z tohoto důvodu je nutné umožnit tvůrcům experimentu vytvářet vlastní scény.

Stejné limitace se vztahují i na tvorbu jednotlivých otázek, které jsou během provádění experimentu kladeny subjektu. Lze vytvořit sadu obecných otázek, ve kterých se lze ptát na společné části grafických efektů (např. jeli pozorovaná scéna vizuálně přitažlivější s grafickým efektem či bez něj, nebo zda subjekt pozoroval nějaké nepřirozené vjemy), bohužel jsou tyto otázky příliš obecné a odpovědi na ně mohou být v některých případech bez hodnoty. Je potřeba vymyslet způsob, kterým tvůrce experimentu může formulovat otázky specifické pro testovaný efekt.

Následuje výčet možných řešení, jak uchopit tvorbu a úpravu experimentů:

Sada předdefinovaných experimentů Subjekt je postaven před jeden z množiny hotových experimentů. Výhodou je, že jsou výsledky snadno porovnatelné, protože vychází ze stejných testovacích dat. Toto řešení neposkytuje žádné přizpůsobení experimentů, proto není vhodné pro účely této práce.

Vlastní editor dává maximální kontrolu nad tím, jaké scény a otázky si mohou tvůrci experimentů vytvářet. K dispozici je sada předefinovaných místností, objektů i efektů, ale i možnost nahrát vlastní prvky. Bohužel je toto řešení velmi časově náročné a zároveň by se jednalo o duplikování funkcionality, kterou nabízí zvolený engine.

Konfigurační soubory Tyto soubory popisují jednotlivé experimenty. Lze je nahrát při spuštění aplikace či za běhu. Obsahují popis scény, efektu a otázek a je možné je přímo editovat. Nevýhodou je, že musíme používat jen prvky, které už se v aplikaci nachází, což značně omezí počet možných experimentů, které lze vytvořit.

Editor herního enginu Umožňuje absolutní kontrolu nad prováděnými experimenty. Součástí je 3D editor, kde si může tvůrce experimentu vytvářet vlastní testovací místnosti a scény. Odpadá závislost na předefinovaných scénách. Mimo to lze psát vlastní skripty pro ovládání experimentu.

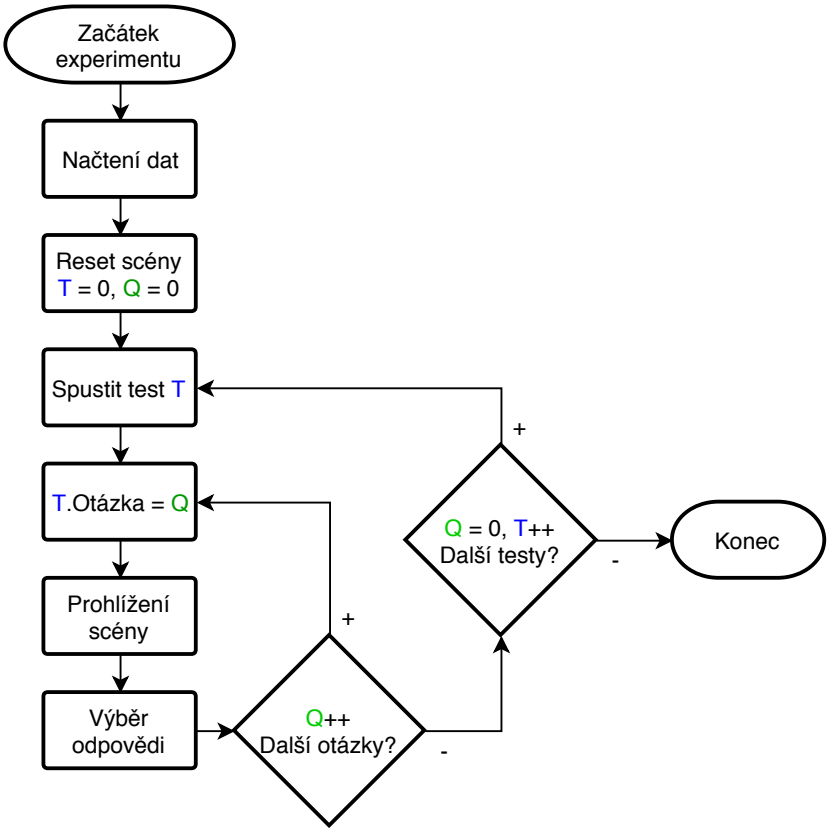
Pro potřeby této práce jsem zvolil kombinaci konfiguračních souborů a editor herního enginu. Vytvořit vlastní editor by bylo příliš časově náročné na rozsah této práce. Konfigurační soubory umožní snadné přenášení konfigurací experimentů mezi experimentátory. Stejně snadná je pak i dodatečná úprava parametrů experimentu (nastavení shaderů, otázky, odpovědi), která nevyžaduje novou kompilaci programu. Formátu těchto souborů je věnována část 6.7 na straně 46.

6.4 Styl provádění experimentu

Každý experiment je rozdělen na jednotlivé testy a každý test má několik otázek. Na obrázku 6.1 je graficky znázorněn průběh obecného experimentu. Po vyčerpání všech otázek u jednoho testu se přejde na další test a po vyčerpání testů experiment končí. Jak bylo již zmíněno v části 4, experimenty budou rozděleny do dvou konkrétních verzí: objektové shadery a post-processing efekty.

Objektové shadery obsahují dvakrát stejný objekt se stejným aplikovaným shaderem, ale rozdílnými parametry. Tento přístup je vhodný k určování hraničních hodnot (intenzita, počet iterací, apod.). Konkrétní otázky a přítomnost/absence reference je na uvážení tvůrce experimentu.

Post-processing efekty V tomto experimentu je vyloučena přítomnost dvou stimulů najednou, protože efekty zpravidla upravují celý obraz. Pro provádění těchto experimentů jsem navrhl systém místností (každý test bude jedna místnost) a testovaný bude moci přepínat mezi různými verzemi připravených post-processing efektů.



Obrázek 6.1: Obecný průběh experimentu

Na testujícím je zvolit vhodný typ experimentu, který nejlépe zkoumá konkrétní problematiku a případně vytvořit vhodné testovací místnosti, objekty respektive celé shadery.

6.5 Podporované platformy

Jedním z doplňujících požadavků je možnost distribuce aplikace na co nejširší možný počet platforem. Přestože je aplikace primárně vyvíjená pro platformu PC, díky použití herního enginu Unity lze bez problému (případně s minimálním úsilím) stejný kód distribuovat pro platformy jako např. Android, iOS, PS4, Xbox, Blackberry, atd. za předpokladu, že jsou k dispozici potřebné licence. Nejdůležitější ze zmíněných jsou PC, Android a PS4, protože jsou aktivní v oblasti virtuální reality. Bohužel během vývoje nebyla k dispozici licence pro PS4, a tak není aplikace na této platformě vyzkoušená.

6.6 Interakce s aplikací

Ovládání aplikace je potřeba uzpůsobit vyšším nárokům na množství podporovaných platforem. Někdy je k dispozici klávesnice, jindy jen jedno tlačítko. Pro povahu tohoto projektu je potřeba umožnit pohyb po scéně, prohlížení scény a volbu odpovědí na kladené dotazy. Následuje výčet možných periferií, které lze využít pro návrh interakce s aplikací:

Klávesnice a myš Standardní periferie počítačů. Klávesnicí je ovládána pozice kamery, myší rotace kamery. Jedná se o běžný, intuitivní přístup většiny počítačových her a simulací. Je velká pravděpodobnost, že uživatel tyto periferie vlastní. Bohužel ve virtuální realitě tato metoda ovládání působí simulátorovou nevolnost.

Gamepad Jedná se o obvyklou periferii u herních konzolí. Nejsou tak rozšířené a zdaleka nenabízí takové množství signálů jako klávesnice a myš, nicméně svoji oblibu získaly díky větší ergonomii a zapamatovatelnosti tlačítek. Intuitivní ovládání je také výhodné, je-li je zrak pozorovatele limitován nasazeným headsetem. Stejně jako u myši a klávesnice hrozí při pohybu ve virtuální realitě simulátorová nevolnost.

Pohybové ovladače Jedná se o sadu (obvykle dvou) polohově snímaných ovladačů, které jsou specializovány na interakce ve virtuální realitě. Moderní VR systémy, díky pohybovým ovladačům, simulují ruce uživatelů a interakce s virtuálním prostředí je mnohem přirozenější.

Leap Motion je senzor, který dvěma kamerami snímá pohyb a gesta rukou uživatele. Lze ho upevnit na headset a používat ve virtuální realitě. Přestože Leap nabízí atraktivní možnost ovládání, jeho rozšíření je poměrně vzácné.

Ovládání pohledem V potaz je třeba brát i sestavy, které nemají dostupné jiné periferie, než jen pohled samotný. V začátcích virtuální reality vniklo ovládání pohledem. Uživatel se natočí směrem k objektu, se kterým chce interagovat a po vypršení časového limitu, je-li pohled stále upřený na stejný objekt, se vyšle signál kliknutí. Jedná se o jednoduchou metodu interakce, která může být dostatečná, je-li tomu aplikace přizpůsobena.

V následujících částech budou diskutovány jednotlivé aspekty aplikace, možná řešení a konečné rozhodnutí.

6.6.1 Pohyb po scéně

Umožnění pohybu zvyšuje kvalitu získaných odpovědí, protože má subjekt větší prostor pro interakci se scénou a jeho rozhodnutí budou více informovaná. S ohledem na dříve uvedené periferie a primární zaměření na virtuální realitu jsou k dispozici následující možnosti pohybu:

Posuv Vhodná aproximace reálného pohybu. Při stisku příslušného tlačítka je avatar posouván ve virtuálním prostředí definovaným směrem. Posouvání se je velmi populární styl pohybu u her. Ve virtuální realitě ale hrozí nevolnost a ztráta rovnováhy, jelikož mozek vidí pohyb, ale nekoná (kinetóza).

Teleport Ačkoliv není teleportování přirozené, ve virtuální realitě je mnohem lépe snášeno, než prosté posouvání mezi body. Častým přístupem je ukazovátko, kterým je označen cíl, načež je uživatel přemístěn. Během teleportace je vhodné obraz ztmavit aby uživatel nevnímal samotné přemístění.

Fyzický pohyb U VR systémů, které nabízejí *room scale experience*, je možné chodit v reálném prostředí a stejný pohyb bude reprezentován ve virtuálním prostředí. Scény musí být specificky upravené pro tento druh pohybu, přesto se však jedná o velmi pohodlnou možnost.

Bez pohybu Je-li zážitek čistě pozorovací, není potřeba řešit pohyb ve virtuálním prostředí. Pohyb také může být vykonáván programově, což se ale kvůli zmíněným problémům nedoporučuje.

Armswinger Pohybová metoda specifická pro aplikace ve virtuální realitě, kdy uživatel napodobuje rukama gesto, které by jinak dělal, kdyby normálně chodil. Aplikace toto gesto rozpozná a posune uživatele ve

směru jeho pohledu. Pro tuto metodu je potřeba mít pohybové ovladače. Existují i variace, kdy uživatel dělá gesto, jako by se přitahoval za fiktivní lano. S tímto přístupem je zatím jen experimentováno, protože uživatelé dávají smíšenou zpětnou vazbu ohledně pohodlí této metody.

Jako pohybovou metodu jsem zvolil kombinaci všech zmíněných krom metody armswinger, která by pro malý užitek zabrala mnoho implementačního času. Ovládání pomocí klávesnice a myši bude realizováno pro testy mimo VR, jinak budou scény upravovány pro fyzický pohyb a teleportování. Pro mobilní aplikace bude dostatečné ovládání pohledem.

6.6.2 Uživatelské rozhraní

Podle Frageholta a Lorentzona [35] rozlišujeme 4 druhy rozhraní:

Diegetic rozhraní je viditelné (přítomné) ve virtuálním prostředí a herní charaktery ví o jeho přítomnosti (např. ovládání výtahu).

Non-diegetic rozhraní je přidáno na vykreslenou herní scénu a není součástí virtuálního světa, slouží pouze pro uživatele aplikace. Také známe jako HUD (*heads up display*).

Spatial rozhraní je součástí virtuálního prostředí, avšak herní charaktery o něm nevědí (např. obrysy postav).

Meta rozhraní mají základ ve virtuálním prostředí, ale nemusí v něm být reprezentovány prostorově (např. krvavé cákance na obrazovce indikující zranění herní postavy)

Pro účely této aplikace jsem zvolil *spatial* rozhraní, protože jsou ze zkušenosti vhodnou volbou pro použití ve virtuální realitě. Tato rozhraní budou umístěná na určených plochách uvnitř testovaných scén. Původním návrhem bylo umístit ovládací prvky na jeden z pohybových ovladačů, ale později jsem usoudil, že připevněním rozhraní k pohybovým ovladačům vznikne závislost na jejich existenci a tu nelze zaručit u všech zařízení. *Non-diegetic* rozhraní bude použito pro ovládání přehrávání záznamů.

6.6.3 Shrnutí

Interakce se systémem byly navrženy tak, aby bylo pokryto co největší množství platforem a aby vždy existoval způsob, jak experiment provést v co možná nejvyšší kvalitě.

6.7 Formát vstupních dat

V této a následujících dvou částech bude diskutován výběr vhodných formátů pro popis jednotlivých částí aplikace. Na výběr je z:

Textový soubor Specificky navržené textové soubory jsou vhodné pro snadnou úpravu lidmi díky jejich čitelnosti. Úprava je také snadná, protože textové editory bývají standardním vybavením operačních systémů.

Binární soubor Tyto soubory jsou oproti textovým hůře čitelné, zato jsou úspornější co se velikosti týče. Binární soubory jsou náchylné k chybám, kdy jeden špatný znak může učinit soubor nečitelným. Často je potřeba vyvíjet specializované editory těchto souborů, pokud jsou úpravy vyžadovány.

XML soubor Použití standardizovaných XML souborů nabízí řadu výhod. Mnoho textových editorů nabízí zvýrazňování syntaxe a obvykle podporují jazyk XML. Soubory ve formátu XML jsou odolnější vůči chybám než výše zmíněné přístupy. Vzhledem k tomu, že aplikace bude vyvíjena v Unity, které používá C# jako hlavní programovací jazyk, je možné využít nativní podpory serializace objektů do souborů formátu XML.

JSON soubor Kromě XML nabízí Unity serializaci do formátu JSON, jehož předností je menší velikosti, ovšem na úkor čitelnosti. Formát JSON je vhodný pro ukládání aplikačních nastavení nebo pro snazší interakce s webovými službami.

S ohledem na lepší čitelnost, která usnadňuje editování vstupních souborů, byl vybrán formát XML. Jedná se o čistě osobní preferenci a soubory formátu JSON jsou rovněž validní volbou. Binární soubory by byly těžko editovatelné a textové soubory příliš náchylné k chybám z pohledu složité a striktní syntaxe, nehledě na nutnost psaní vlastního parseru.

6.8 Formát výstupních dat

Výstupní data jsou určena pro strojové zpracování, nicméně lidská analýza nemůže být vyloučena, a navíc by musel být napsán specializovaný parser, proto i pro výstup volím soubory XML.

6.9 Formát záznamu experimentu

U nahrávek experimentu je potřeba zohlednit především velikost souborů, jelikož je ukládáno mnoho souřadnic za sekundu (v závislosti na rychlosti záznamu). Z tohoto důvodu jsem zvolil použití binárních souborů pro nahrávky experimentů.

7 Implementační detaily

Aplikace byla vytvořena v prostředí **Unity 3D 2017.3.1f1** (dále jen Unity), proto budou často zmiňovány konstrukce specifické pro tento herní engine.

7.1 Scény

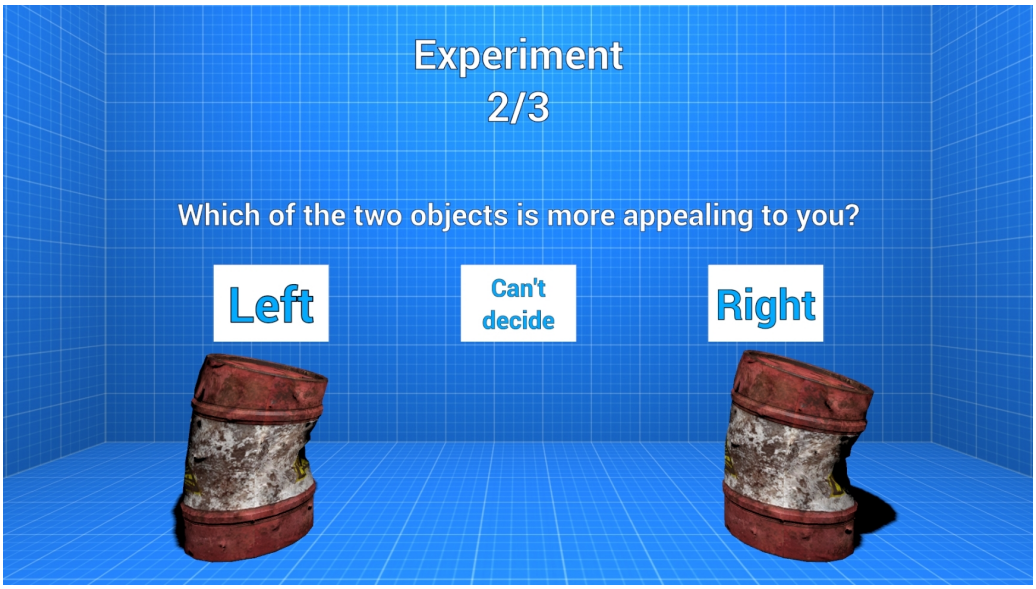
Scény jsou v terminologii Unity ucelené funkční bloky, mezi kterými lze přepínat. Tato aplikace se sestává ze tří scén:

Hlavní menu je úvodní scéna, která slouží jako rozcestník. Zde uživatel volí, který z dostupných experimentů chce (nebo má) provádět.

Post-processing Zde se provádí post-processing experimenty. Scéna se skládá z úvodní místnosti, kde je uživatel uvítán, je mu krátce vysvětlena náplň a průběh experimentu. Dále uživatel pokračuje do místností definovaných v souboru definice experimentu (viz obrázek 7.1).

Objektové shadery Tato scéna obsahuje větší místnost se dvěma fiktivními pódii, na kterých budou objekty k testování (viz obrázek 7.2).

Obrázek 7.1: Testovací místnost post-processing efektů



Obrázek 7.2: Testovací místnost objektových shaderů

Navíc je ještě přidána scéna `PrefabTestingScene` pro přípravu prefabů, což je ale možné i v ostatních scénách, a není proto zahrnuta do výsledného sestavení.

7.2 Persistence dat mezi scénami

Při přepínání scén nelze předávat parametry z jedné do druhé, a protože se soubory s experimentem (nebo nahrávkou) načítají již v hlavním menu, je potřeba navrhnout způsob předání dat do odpovídající scény. Tento problém řeší statická třída `ApplicationDataContainer`, která obsahuje načtený deserializovaný soubor a přepínač, který indikuje, zda se jedná o experiment nebo nahrávku. Přepnutím scény nedojde ke smazání uložených dat.

7.3 Formát vstupních a výstupních dat

Pro snadné používání jsem navrhl vstupní soubory ve formátu XML tak, aby se chovaly jako šablony pro vyplnění. Během experimentu jsou na definovaná místa do vstupního souboru ukládány odpovědi a po skončení testu je tento soubor uložen spolu s informacemi o systému, kde experiment proběhl.

Mezi hlavní výhody patří, že výstupní soubor stále obsahuje otázky a odpovědi, tedy je snazší se pak orientovat v tom, o jaký test se jednalo a

jaké otázky byly kladeny. Další výhodou je to, že výstupní soubor lze znovu použít jako vstupní, a tedy provést stejný experiment znovu, aniž by se musel dohledávat původní vstupní soubor. Příklad vstupního souboru je v ukázce 7.2 na straně 54.

7.4 Parametrizace grafických efektů

Ačkoliv lze najít společné prvky některých grafických efektů (např. antialiasing a SSAO mohou používat stejný algoritmus pro rozostření), toto není dostatečně dobrá abstrakce. Dva grafické efekty mohou být tak odlišné, že je potřeba s parametrizací jít až k základům (k shaderům). Samotné algoritmy už mají společných prvků více – počet iterací, velikost samplingu a to už jsou jednotlivé číselné hodnoty, které mají jasně definované datové typy, a proto jsou vhodné pro parametrizaci většiny efektů.

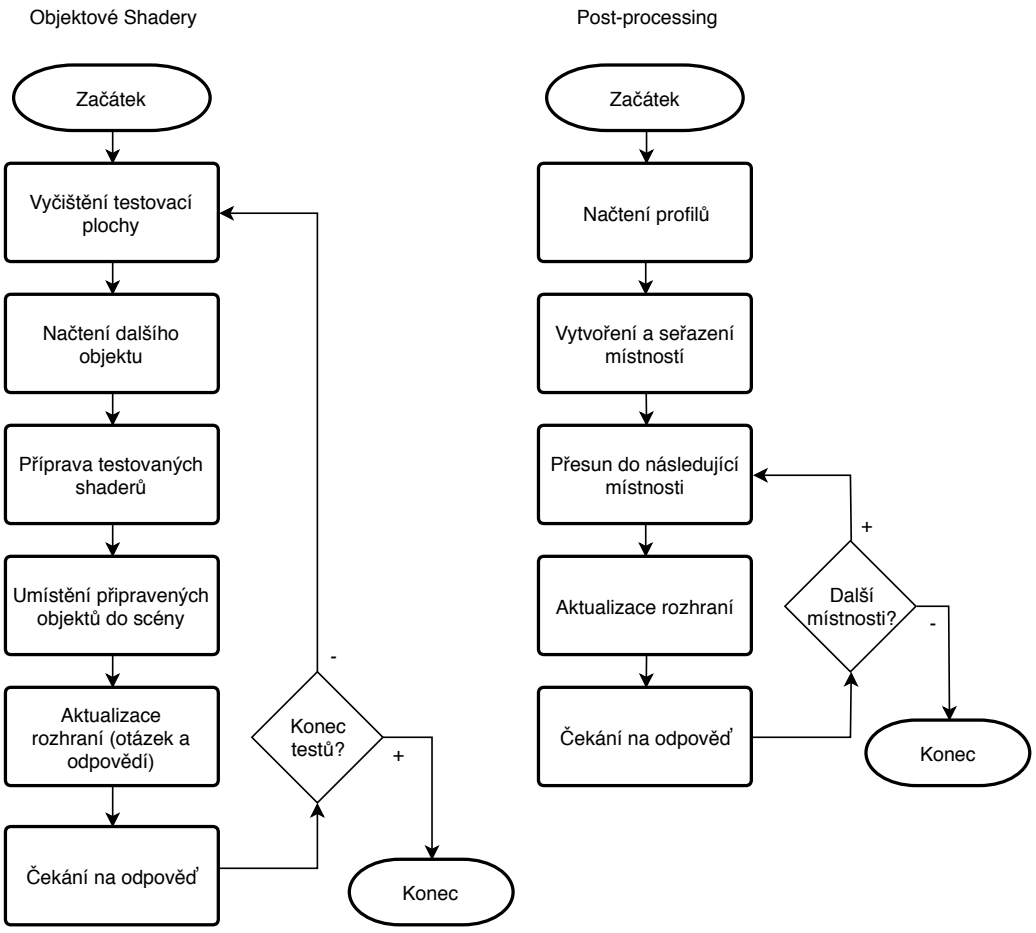
```
1  Shader "SampleShader"
2  {
3      Properties
4      {
5          // Parametry shaderu
6          _MyInt   ("integerName", Int) = 2
7          _MyFloat ("floatName", Float) = 1.5
8      }
9
10     SubShader
11     {
12         //Výkonný kód shaderu
13     }
14 }
```

Ukázka 7.1: Struktura shaderů v Unity

V Unity se při tvorbě shaderů parametry ukládají do `Properties` bloku a jsou pak přístupné programově (způsob reference ze vstupních souborů je v ukázce 7.1). Parametry lze reprezentovat trojicí: *název*, *datový typ*, *hodnota*. Ve třídě `EffectSettings` se řeší převod z textové podoby do příslušných datových typů a lze ji případně rozšířit o další, jsou-li podporované shaderem.

7.5 Řízení experimentů

Třídy `OOExperimentController` a `PPExperimentController` se starají o průběh experimentů (zjednodušená vizualizace průběhu viz obrázek 7.3). Prefixy `OO` (on object) a `PP` (post-processing) značí, o jaký druh experimentu se jedná.



Obrázek 7.3: Průběh řízení experimentu

Při spuštění scény je nejprve ověřeno, zda je načtený soubor nahrávka (podle příznaku v globálně přístupné statické třídě viz 7.2) a případně je kontrola nad průběhem předána nahrávacímu subsystému.

7.6 Nahrávací subsystém

Nahrávací subsystém má za úkol snímat a později přehrávat pozici a orientaci uživatele, aby bylo možné rozhodnout, zda se subjekt dostatečně pohyboval (rozhlížel) po testované scéně. Ve virtuální realitě to znamená snímání headsetu a pohybových ovladačů (jsou-li dostupné).

Nahrávání probíhá v rámci metody `FixedUpdate`, která je volána 50× za sekundu a slouží především pro fyzikální výpočty. Obvyklým přístupem je používání metody `Update`, která je volána s každým vykresleným snímkem, nicméně tato volání nejsou ekvidistantní kvůli různým vykreslovacím dobám, a také představují větší datový tok, který není pro přehrávání nutný.

Vzhledem k tomu, že výchozí třídy pro popis pozice a rotace v Unity nejsou serializovatelné, vytvořil jsem třídu `PointInTime`, která serializaci umožňuje.

Během přehrávání je přepnuto na ovladatelnou monoskopickou kameru, se kterou je pohybováno klávesnicí a myší. V uživatelském rozhraní tohoto režimu je tlačítko pro pozastavení a časová osa přehrávání s ovladatelným jezdcem.

7.7 Přepínač ovládání

Při vývoji aplikace bylo často potřeba přepínat mezi stereoskopickým a monoskopickým pohledem, a proto vznikla třída `PlayerControllerSwapper`. V konečné verzi programu měla být tato třída odstraněna, nicméně její užitek je značný, zejména při přehrávání záznamů na počítačích bez VR příslušenství. Ve finální verzi je tato funkcionalita deaktivována v průběhu experimentů.

7.8 Spouštění s/bez virtuální reality

Experimenty lze provádět i mimo prostředí virtuální reality. Cílem je získat srovnání výsledků obou platforem a najít klíčové odlišnosti. Přepínání režimů je implementováno pomocí výše zmíněné třídy `PlayerControllerSwapper`. V hlavním menu je možné přepnout režim provádění experimentů. Výstupní soubory obsahují informace o systému, kde byl experiment proveden a použitý režim.

7.9 Použité knihovny

Všechny knihovny, které jsem použil, jsou dostupné ve službě *Asset Store* (součást Unity) pod těmito názvy:

SteamVR Tato knihova od společnosti Valve poskytuje abstrakci pro některá VR zařízení (Oculus Rift, HTC Vive, Windows MR, aj.). V práci jsem ji využil hlavně pro hladký přechod mezi headsety. Steam VR rozpoznává připojená zařízení a poskytuje jednotné rozhraní pro jejich ovládání.

Virtual Reality Toolkit (VRTK) VRTK obaluje knihovnu SteamVR a poskytuje mnoho funkcí pro interakce s virtuálními prostředími. Mezi klíčové funkce patří teleport, uživatelská rozhraní specifická pro virtuální realitu, chytání a manipulace objektů, atd. V aplikaci používám VRTK k řešení většiny VR specifických problémů.

Post-processing stack Tento balíček umožňuje snadné používání post-processing efektů. Od Unity verze 2017 se stal oficiálním balíčkem a v dalších verzích se stane součástí Unity. Tento balíček jsem využil zejména pro možnost vytváření post-processing profilů, které lze za běhu přepínat a jsou kompatibilní s virtuální realitou. Vytvořené profily lze také snadno exportovat a distribuovat do dalších aplikací.

Text Mesh Pro (TMP) TMP je vylepšení stávajících fontů v Unity. Nepoužívá k sázení rastrový atlas, ale uchovává fonty ve vektorové podobě. Takto je dosaženo ostrého písma, což je důležité zejména ve virtuální realitě, kde bývá čtení obtížné. Mimo jiné také nabízí širokou paletu textových efektů, kterých bylo hojně využíváno.

7.10 Použité modely

Vzhledem k mým omezeným schopnostem v oblasti 3D modelování a z časových důvodů jsem se rozhodl použít hotové balíčky modelů, aby bylo v reálném čase možné aplikaci otestovat. Stejně jako použité knihovny i použité modely pochází z Asset store. V této práci jsem použil následující balíčky: BigFurniturePack, DesertEnvironment, RocksAndBoulders2, WesternProps.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Experiment
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ExperimentType>00</ExperimentType>
  <ExperimentStart>0001-01-01T00:00:00</ExperimentStart>
  <ExperimentEnd>0001-01-01T00:00:00</ExperimentEnd>
  <Tests>
    <Test00>
      <Questions>
        <Question>
          <Text>[This text can be whatever you want.]</Text>
          <Options>
            <Option>Answer 1</Option>
            <Option>Answer 2</Option>
          </Options>
          <Answer>0</Answer>
        </Question>
      </Questions>
      <ExperimentObjectName>WoodCrate</ExperimentObjectName>
      <ObjectOneSettings>
        <Settings>
          <PropertyName>_BumpMapIntensity</PropertyName>
          <PropertyType>float</PropertyType>
          <PropertyValue>1</PropertyValue>
        </Settings>
      </ObjectOneSettings>
      <ObjectTwoSettings>
        <Settings>
          <PropertyName>_BumpMapIntensity</PropertyName>
          <PropertyType>float</PropertyType>
          <PropertyValue>0</PropertyValue>
        </Settings>
      </ObjectTwoSettings>
    </Test00>
  </Tests>
</Experiment>
```

Ukázka 7.2: Ukázka vstupního souboru

8 Pilotní studie

K validaci funkčnosti byla 9. května 2018 provedena pilotní studie se třemi subjekty. Součástí pilotní studie bylo pozorování uživatelů a jejich interakce s aplikací za účelem zlepšení ovládání a nejasných částí experimentu.

8.1 Testovací podmínky a vybavení

Studie byla prováděna na zařízení Oculus Rift CV s pohybovými ovladači Touch. Aplikace byla spuštěna na počítači s těmito parametry:

- CPU: Intel i5-6500 (4 jádra, 3.20 GHz)
- GPU: NVIDIA GTX 1070 (8GB DDR5)
- RAM: 16GB DDR3
- OS: Microsoft Windows 10 (verze 1803)

Testovaní měli prostor přibližně 2×2 metry pro volný pohyb. Podpora z mé strany byla minimální.

8.2 Průběh experimentu

Experiment byl veden podle následujícího scénáře:

1. Krátký úvod k používání virtuální reality a seznámení subjektů s možnými riziky (nebezpečí kolizí, nevolnosti, atd.).
2. Dále byli testovaní navedeni k první části experimentu (objektové shadery).
3. Po dokončení stejným způsobem spouštěli druhou část experimentu (post-processing).
4. Po skončení druhého experimentu proběhla krátká diskuze za účelem získání zpětné vazby. Součástí byly i mé cílené dotazy na určité části aplikace (čitelnost písma, pohybové možnosti, apod.).

8.3 Změny na základě zpětné vazby

Následuje výčet zaznamenaných problémů a připomínek rozšířený o mé vlastní poznatky z pozorování subjektů (seřazeno od nejčastějších a nejzávažnějších po banální):

Čitelnost písma V závislosti na pozadí mohlo být písmo hůře čitelné. Písmo jsem upravil napříč celou aplikací, aby mělo světlé barvy a černý okraj. Takto upravené písmo je dobře čitelné nezávisle na podkladu.

Ovládání Teleport a výběr odpovědi byly na různých tlačítkách a testovaní často spoléhali na náhodu při jejich hledání. Tyto operace jsem spojil do jednoho tlačítka a upravil i některé další operace jako aktivování navigačního laseru.

Pozice odpovědí Původní myšlenka byla dát tlačítka s odpovědí co nejdále od sebe, aby nedocházelo k překlikům. Ukázalo se, že testovaní nemají problém s klikáním správných tlačítek a jejich rozptyl jen zpomaluje čtení, a tedy i celý experiment. Tlačítka jsou nyní více nashlukována uprostřed uživatelského rozhraní.

Celková doba trvání experimentu V nejdelším případě trval experiment 12 minut, což se setkalo s kritikou testovaných, kteří očekávali více. Toto je problém rozsahu experimentu, který byl pro nedostatek aktiv malý.

Malá diverzita testovaných místnosti a předmětů Tento problém pramení ze stejných důvodů jako problém předchozí. Pro vyhotovení aplikace byly použity bezplatné objekty z *asset store* a několik ručně vytvořených kulis, kterých v součtu není mnoho. Při vytváření dalších experimentů se předpokládá, že si tvůrce dodá vlastní objekty a připraví scény specificky pro řešený problém.

8.4 Závěr pilotní studie

Na třech testovacích subjektech bylo ověřeno, že je aplikace vhodná pro provádění expermientů zaměřených na hodnocení kvality renderovaných scén. Díky zpětné vazbě se podařilo eliminovat několik nedostatků.

9 Uživatelská příručka

V této kapitole budou popsány způsoby používání aplikace.

9.1 Spuštění

Spouštění aplikace je možné ve dvou režimech:

Standalone verze Prosté spuštění sestaveného projektu (`.exe`, `.apk`). Standardní přístup provádění experimentu. Stejný experiment může být rozdistribuován mezi testované.

V rámci Unity editoru Nahraný projekt lze spouštět v prostředí Unity, což je výhodné zejména pro testujícího, který může sledovat jednotlivé parametry experimentu. Tento přístup je také vhodný pro vývoj a návrh experimentů. Projekt je kompatibilní s Unity 3D 2017.3.1f1.

9.2 Vstupní a výstupní soubory

- Vstupní XML soubory musí být umístěny ve složce `Experiments` v kořenovém adresáři aplikace.
- Výstupní soubory jsou ukládány do složek `Results` respektive `Recordings` ve fromátech XML a REC.
- Výstupní XML soubory lze používat jako vstupní.

Prázdné šablony lze vygenerovat stiskem tlačítka `Create Templates` v menu s experimenty. Vytvořeny budou dva soubory, jeden pro objektové experimenty a druhý pro post-processing experimenty (odlišitelné podle názvu a podle tagu `<ExperimentType>`.

9.3 Provádění experimentu

Experimenty objektových shaderů Po načtení je uživatel umístěn do testovací místnosti. Před sebou na stěně bude mít uživatelské rozhraní s pokyny, co má dělat. Ovladače mají popisky, co které tlačítko dělá. Uživatel si podle pokynů prohlédne scénu a vybere jednu z nabízených odpovědí. Takto postupuje dokud nevyčerpá všechny otázky. Během

experimentu se smí uživatel pohybovat a teleportovat ve vymezené oblasti. Po dokončení je aplikace přepnuta zpět do hlavního menu.

Post-processing experimenty Po načtení je uživatel umístěn do pomocné místnosti, kde má před sebou na zdi umístěné uživatelské rozhraní, které obsahuje krátký návod a tlačítko ke spuštění. Po spuštění experimentu je přemístěn do první testovací místnosti, kde se opět řídí pokyny na zdi a vybírá odpovědi, které nejlépe odpovídají na kladené dotazy. Během těchto experimentů uživatel přepíná mezi jednotlivými verzemi připravených post-processing efektů. Pro zjednodušení ovládání je přepínání scén umístěno na pohybové ovladače i je součástí UI na zdi. Jsou-li zodpovězeny všechny otázky, pak je uživatel navrácen do pomocné místnosti, kde je mu poděkováno za jeho účast a tlačítkem *End* se může vrátit zpět do hlavního menu.

V hlavním menu lze aplikaci přepínat mezi režimem virtuální reality a bez pomocí tlačítka `Mode` nebo klávesou F2. Aplikace se po spuštění pokusí sama zjistit, je-li dostupný headset a režim automaticky přepne.

9.4 Sestavení aplikace

Sestavení probíhá v prostředí Unity 3D 2017.3.1f1, které řeší i veškeré závislosti spojené s překladem. Pro sestavení aplikace postupujte podle následujících bodů:

1. V hlavním menu: `File -> Build Settings`
2. V dialogovém okně vyberte cílovou platformu pro export (implicitně PC, Mac a Linux)
3. Stiskněte tlačítko `Build`

9.5 Vytváření vlastních experimentů

V následujících dvou částech budou uvedeny postupy pro tvorbu vlastních experimentů. Předpokládá se pokročilá znalost Unity 3D a psaní shaderů v HLSL.

9.5.1 Vytváření experimentů pro objektové shadery

Přidání nového testovaného objektu zahrnuje následující kroky:

1. Vytvoření modelu a shaderu (lze také exportovat z existujících Unity projektů).
2. Import do tohoto projektu.
3. Vytvoření *prefabu*[1], který je uložen do složky `Resources`

Nyní je tento objekt testovatelný a lze k němu referovat ve vstupních souborech jménem prefabu. Parametrizace se řídí náležitostmi, které byly popsány v částech 7.4 a 7.3.

9.5.2 Vytváření post-processing experimentů

Vytváření těchto experimentů se skládá ze dvou částí:

Tvorba scény Pokud není žádná z předpřipravených scén (místností) z nějakého důvodu vhodná, je potřeba vytvořit místnosti vlastní. Místnost může být libovolný prefab, který je označen tagem `ExperimentRoom`. Testovací místnosti musí obsahovat dva objekty (mohou být prázdné) s tagem `SpawnPoint` (pro umístění subjektu) a `CanvasAnchor` (pro umístění UI). Hotový prefab místnosti musí být umístěn ve složce `Resources/ExperimentRooms`.

Tvorba efektu Post-processing efekty jsou v Unity součástí struktury nazvané *Post-processing stack*. Tato struktura uchovává svoje konfigurace v profilech zvaných *Post processing profile*. Testované profily je potřeba uložit do složky `Resources/PostProProfiles` s názvem `ppp_` následováno pořadovým číslem (ukázky jsou součástí aplikace). Pořadové číslo určuje pořadí přepínání profilů pro testování. Počet profilů není omezený, ale doporučuji nepřekračovat 3. Podrobný návod k přidání post-processing efektů podle Hourdela viz [36].

Ve vstupním XML souboru se popisují pouze místnosti, které mají být pro experiment použity, otázky a možné odpovědi.

[1] Prefab je v Unity šablona udržující informace o objektu a jeho vlastnostech.

10 Závěr

Cílem projektu bylo vytvořit prostředí pro vytváření subjektivních experimentů pro hodnocení kvality renderovaných scén ve virtuální realitě za účelem hledání hraničních hodnot parametrů grafických efektů, na jejichž základě lze optimalizovat výkon her, simulací a dalších projektů ve virtuální realitě.

Výsledný software byl vytvořen v prostředí Unity 3D a splňuje zadání v celém rozsahu a to včetně některých doplňujících požadavků, které se objevily během vývoje. Krom dvou grafických efektů, na které byla práce zaměřena, se mi podařilo vytvořit vhodnou abstrakci většiny grafických efektů na úrovni jejich parametrů. Mimo hledání hraničních hodnot lze experimenty zaměřit na pozorování grafických artefaktů nebo na průzkum vlivu stereoskopického zobrazování na prezentaci grafických efektů. Na základě pilotní studie se podařilo validovat účel práce a opravit některé nedostatky. Aplikace je tedy vhodná pro provádění subjektivních experimentů a to nejen na posuzování kvality renderovaných scén.

Nad rámec zadání jsem implementoval nahrávání experimentů, které, přestože není dokonalé, postačí k analýze průběhu. Při návrhu byl kladen důraz na možnost distribuce pro další platformy než jen PC a zároveň na podporu více headsetů, proto jsou uživatelská rozhraní a ovládání uzpůsobeny tomuto požadavku. Aplikaci se podařilo bez problémů vyzkoušet na počítačích s poměrně vysokým rozpětím ve výkonu za použití zařízení Oculus Rift a HTC Vive.

Práce byla prezentována na studentské vědecké konferenci (SVK 2018) fakulty aplikovaných věd v Plzni.

10.1 Doporučení pro další vývoj

Zajímavým projektem by mohlo být vytvoření podobných experimentů v jiných herních enginech, zejména v Unreal Engine, který je známý pro své kvalitní zpracování grafických efektů.

Jako rozšíření stávajících experimentů doporučuji testování částicových efektů, které jsou velmi častým prvkem počítačových her. Částečně lze částicové efekty kombinovat s objektovými shadery a mohly by být pozorovány zajímavé interakce.

Přestože byl kladen důraz na široký záběr zařízení pro virtuální realitu, některé platformy se pro jejich nedostupnost nepodařilo otestovat jinak než

v simulátoru. Konkrétně Gear VR, Oculus Go, Daydream stojí za vyzkoušení. Aplikace byla navržena tak, aby snadný přechod umožňovala, nicméně není vyloučeno, že budou potřeba platformově specifické úpravy. Ovládání pohledem bylo vyzkoušeno a uživatelská rozhraní respektují limitace některých zařízení.

Prohlížení nahrávek je velmi jednoduché a nabízí se hned několik možností zlepšení. Přehrávání z pohledu subjektu může být jedno ze snazších rozšíření, jelikož to zaznamenaná data umožňují.

Seznam obrázků

Seznam tabulek

Literatura

[1] Steuer, J. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*. 1992, 42, 4, s. 73–93.

[2] McLellan, H. Virtual realities. *Handbook of research for educational communications and technology*. 1996, s. 457–487.

[3] Burdea Grigore, C. – Coiffet, P. *Virtual reality technology*. London: Wiley-Interscience, 1994.

[4] Boas, Y. Overview of virtual reality technologies. In *Interactive Multimedia Conference*, 2013, 2013.

[5] Abrash, M. What VR could, should, and almost certainly will be within two years. *Steam Dev Days, Seattle*. 2014.

[6] Heilig, M. L. Stereoscopic-television apparatus for individual use, October 4 1960. US Patent 2,955,156.

[7] Heilig, M. L. Sensorama simulator, August 28 1962. US Patent 3,050,870.

[8] Bricken, W. Virtual Reality: Directions of Growth Notes from the SIGGRAPH'90 Panel. *Virtual Reality: Directions of Growth*. 1990.

[9] Deering, M. High resolution virtual reality. *ACM SIGGRAPH Computer Graphics*. 1992, 26, 2, s. 195–202.

[10] Costello, P. J. *Health and safety issues associated with virtual reality: a review of current literature*. Citeseer, 1997.

[11] Hettinger, L. J. et al. Vection and simulator sickness. *Military Psychology*. 1990, 2, 3, s. 171.

[12] Shibata, T. et al. The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of vision*. 2011, 11, 8, s. 11–11.

[13] Schultheis, M. T. – Rizzo, A. A. The application of virtual reality technology in rehabilitation. *Rehabilitation psychology*. 2001, 46, 3, s. 296.

[14] Freeman, D. et al. The psychology of persecutory ideation II: a virtual reality experimental study. *The Journal of nervous and mental disease*. 2005, 193, 5, s. 309–315.

[15] Mitchell, J. – McTaggart, G. – Green, C. Shading in valve's source engine. In *Acm siggraph 2006 courses*, s. 129–142. ACM, 2006.

[16] De Moura, I. F. M. – Machado, L. S. Shader Integration in a Virtual Reality Framework. In *Virtual and Augmented Reality (SVR), 2013 XV Symposium on*, s. 276–279. IEEE, 2013.

[17] Rendering Pipeline Overview. *OpenGL Wiki* [online], 2017. Dostupné z: `http://www.khronos.org/opengl/wiki_opengl/index.php?title=Rendering_Pipeline_Overview&oldid=13843`. [cit. 21.4.2018].

[18] Zhukov, S. – Iones, A. – Kronin, G. An ambient light illumination model. In *Rendering Techniques' 98*. Springer, 1998. s. 45–55.

[19] Bavoil, L. – Sainz, M. Screen space ambient occlusion. *NVIDIA developer journal*. 2008, 6.

[20] Zhang, H. et al. Visibility culling using hierarchical occlusion maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, s. 77–88. ACM Press/Addison-Wesley Publishing Co., 1997.

[21] Ek, A. N.-K. – Johnson, O. – Assadi, R. Screen Space Ambient Occlusion. 2015.

[22] Chalk, A. – Fisher, B. Normal mapping solutions for Oculus Rift development, 2015.

[23] Munkberg, J. – Akenine-Möller, T. – Ström, J. High quality normal map compression. In *SIGGRAPH/EUROGRAPHICS Conference On Graphics Hardware: Proceedings of the 21 st ACM SIGGRAPH/Eurographics symposium on Graphics hardware: Vienna, Austria*, 3, s. 95–102, 2006.

[24] Kaneko, T. et al. Detailed shape representation with parallax mapping. In *Proceedings of ICAT*, 2001, s. 205–208, 2001.

[25] VR Compliant Post Processing Effects *MiddleVR* [online], 2016. Dostupné z: `https://support.middlevr.com/hc/en-us/articles/203900075-VR-compliant-post-processing-effects-in-Unity`. [cit. 29.4.2018].

[26] Goulekas, K. Visual Effects in A Digital World: A Comprehensive Glossary of over 7,000 Visual Effects Terms (The Morgan Kaufmann Series in Computer Graphics). 2001.

[27] Larabi, M.-c. – Brodbeck, V. – Fernandez, C. A novel approach for constructing an achromatic contrast sensitivity function by matching. In *Image Processing, 2006 IEEE International Conference on*, s. 441–444. IEEE, 2006.

[28] Opozda, S. – Sochan, A. The survey of subjective and objective methods for quality assessment of 2D and 3D images. *Theoretical and Applied Informatics*. 2014, 26, 1-2, s. 39–67.

[29] Pinson, M. H. – Wolf, S. Comparing subjective video quality testing methodologies. In *Visual Communications and Image Processing 2003*, 5150, s. 573–583. International Society for Optics and Photonics, 2003.

[30] Winkler, S. On the properties of subjective ratings in video quality experiments. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, s. 139–144. IEEE, 2009.

[31] BT, R. I.-R. Methodology for the subjective assessment of the quality of television pictures. 2002.

[32] ITU-T RECOMMENDATION, P. Subjective video quality assessment methods for multimedia applications. 1999.

[33] ITU, I. Methodology for the subjective assessment of video quality in multimedia applications. *Rapport technique, International Telecommunication Union*. 2007.

[34] Series, B. Subjective methods for the assessment of stereoscopic 3DTV systems. 2015.

[35] Fagerholt, E. – Lorentzon, M. Beyond the HUD-user interfaces for increased player immersion in FPS games. 2009.

[36] Hourdel, T. Writing Custom Effects [online]. `https://github.com/Unity-Technologies/PostProcessing/wiki/Writing-Custom-Effects`. [cit. 12.5.2018].