



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra počítačové grafiky a interakce**

Bakalářská práce

VR vizualizace černé díry s využitím sférické kamery

Jakub Peleška

Otevřená informatika (BS) - Počítačové hry a grafika

Květen 2021

https://gitlab.fel.cvut.cz/pelesjak/blackholevr_bp

Vedoucí práce: Ing. David Sedláček, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Peleška** Jméno: **Jakub** Osobní číslo: **483727**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Otevřená informatika**
Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

VR vizualizace černé díry s využitím sférické kamery

Název bakalářské práce anglicky:

Black Hole visualization in virtual reality with 360° camera

Pokyny pro vypracování:

Seznamte se přístupy pro vizualizaci černých děr, zařízeními pro virtuální realitu (VR) a kamerami, které zachycují obraz ve všech směrech (tzv. sférické kamery).

Navrhněte a implementujte VR zážitek pro dva uživatele přibližující chování černé díry v reálném prostředí (místnost). Pro realizaci použijte obraz pořízený sférickou kamerou (360°x180°) v reálném čase, který bude zobrazen jako tzv. skybox ve virtuální scéně. Tento obraz a i další objekty scény, deformujte pomocí 'černé díry' uprostřed virtuální místnosti (černá díra bude v pozici, kde se nachází reálná všesměrová kamera). Návštěvníci budou v této místnosti, na opačných stranách od černé díry, tím pádem bude docházet k deformaci vizualizace návštěvníků v pohledu toho druhého v závislosti na vzdálenosti od černé díry.

Definujte limity navrženého systému a otestujte jej jak z pohledu výkonnosti, tak z pohledu kvality vizualizace. Vizualizaci otestujte také s minimálně třemi páry uživatelů.

Teorii z oblasti černých děr a jejich vizualizace konzultujte s Mgr. Michalem Bursou, Ph.D. z Astronomického Ústavu Akademie Věd ČR.

Seznam doporučené literatury:

- 1] Dieter Schmalstieg, and Tobias Hollerer. Augmented Reality: Principles and Practice (Usability), Addison Wesley 2016
- 2] Jason Jerald. 2015. The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA.
- 3] Ivan Platonov, Interaktivní vizualizace černých děr, BP FEL, 2016. <https://dspace.cvut.cz/handle/10467/64702>
- 4] Otto Seiskari, Ray-traced simulation of a black hole, 2015. <https://github.com/oseiskar/black-hole>
- 5] <https://sixemic.github.io/Interstellar/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. David Sedláček, Ph.D., katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **18.03.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **19.02.2023**

Ing. David Sedláček, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Chtěl bych poděkovat Ing. Davidu Sedláčkovi, Ph.D. za pomoc a rady spojené s vývojem pro virtuální realitu. Také bych chtěl poděkovat Mgr. Michalovi Bursovi, Ph.D. za informace a konzultace vázané k vlastnostem černých děr.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

.....

V Praze dne 21. 5. 2021

Jakub Peleška

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

.....

In Prague 21. 5. 2021

Jakub Peleška

Abstrakt / Abstract

Předmětem této bakalářské práce je proces vizualizace černých děr. Pro zvýšení prožitku uživatele je použita VR technologie se sférickým videem. Důraz je kladen na nízkou výpočetní náročnost za účelem kvalitního zážitku ve virtuálním světě.

Klíčová slova: černé díry; vizualizace; interaktivní; VR; Unity; HDRP; sférická kamera; sférické video.

This bachelor thesis shows a process of visualization of black holes. To maximize the level of user immersion virtual reality and 360 video background was used. Low computation complexity is the main priority to present a high-quality user experience in the VR space.

Keywords: black holes; visualization; interactive; virtual reality; Unity; HDRP; 360° camera; 360° video.

Title translation: Black Hole visualization in virtual reality with 360° camera

Obsah /

1 Úvod	1
2 Analýza	2
2.1 Obecná relativita	2
2.2 Vlastnosti černých děr	2
2.3 Virtuální realita	2
2.4 Analýza problematiky	3
2.5 Sférická kamera a video	4
2.6 Možnosti vizualizace	5
2.6.1 Ray tracing	5
2.6.2 Mapovací data	5
2.6.3 Aproximační data	5
2.7 Výchozí stav	7
2.8 Unity	7
2.9 Rešerše a jiné projekty	8
3 Návrh řešení	9
3.1 Požadavky	9
3.2 Platforma	9
3.3 Příprava dat	9
3.4 Struktura scény	11
3.4.1 Vizualizace černé díry ...	11
3.4.2 Sférické video	11
3.4.3 Reprezentace uživatele...	12
3.4.4 Systém pro více uživa- telů	12
4 Implementace	13
4.1 Generování mapovacích dat ...	13
4.2 Unity scéna	14
4.2.1 Získání sférického po- hledu kamery	15
4.3 Vykreslování černé díry	15
4.3.1 Logika shaderu pro model černé díry	16
4.3.2 Logika shaderu pro aproximaci černé díry	17
4.4 Sférické video	17
4.5 Síťové spojení a reprezentace uživatele	17
4.6 Ovládání scény	18
5 Testování	19
5.1 Preference ovládání pohybu ...	19
5.2 Přehled výsledného vzhledu aplikace	20
5.3 Test použitelnosti	22
5.3.1 Testovací postup	22
5.3.2 Výstup testování	23
5.4 Limity aplikace	23
5.4.1 Výkon aplikace	23
5.4.2 Srovnání vizualizací	24
6 Závěr	26
6.1 Další vývoj	26
Literatura	27
A Struktura přiložených souborů ...	29
B Uživatelský manuál	30
B.1 Spuštění programu pro ge- nerování dat	30
B.2 Spuštění projektu v Unity 3D .	30
C Odkazy	32
C.1 Gitlab repozitář projektu	32
C.2 Ukázkové video	32
C.3 3D Modely	32
C.4 Unity packages	32

Tabulky / Obrázky

5.1. Testované možnosti ovládní pohybu.	19
5.2. Výkon aplikace v závislosti na sférickém pohledu.	23
1.1. První fotka skutečné černé díry.	1
2.1. Acer Windows Mixed Reality Headset AH101.	3
2.2. Sférická kamera Samsung Gear 360.	4
2.3. Sférický pohled pořízený pomocí Samsung Gear 360.	5
2.4. Ilustrace aproximačního výpočtu distorze černé díry.	6
2.5. Efekt černé díry z výchozího programu.	7
2.6. Diagram průběhu Unity HDRP řetězce.	8
3.1. Ilustrace generování dat v programu BH Warp.	10
3.2. Ilustrace upraveného způsobu generování dat.	11
4.1. Zpracovaná textura mapovacích dat.	14
4.2. Sférický záběr z testovací scény.	15
4.3. Řetězec souřadnicových transformací.	16
4.4. Model astronauta použitého pro reprezentaci uživatele.	18
5.1. Záběr z hlavního menu.	20
5.2. Vizualizace černé díry pomocí vygenerovaných dat.	20
5.3. Záběr z pause menu v hlavní scéně.	21
5.4. Vizualizace černé díry pomocí aproximační metody.	21
5.5. Pohled uživatel zády k černé díře.	21
5.6. Testovací aranžmá.	22
5.7. Vizualizace černé díry pomocí vygenerovaných dat a aproximační funkce.	24

Kapitola 1

Úvod

Černá díra je extrémní objekt, který vzniká na konci života masivních hvězd, kdy v jejich jádru dojde palivo pro termojadernou fúzi. Následkem toho začne převládat gravitační síla nad tlakem záření a hvězda se začne hroutit. V důsledku hroucení jsou odhozeny vnější vrstvy hvězdy. Pokud je i po této fázi hmotnost zbytkové hmoty přibližně 5 krát větší než je hmotnost Slunce, vzniká těleso, jehož gravitační pole je natolik silné, že od určitého místa z něj neunikne ani světlo. Toto těleso nazýváme černá díra a hranici, z níž neunikne ani světlo, horizont událostí.

Silné gravitační pole černé díry „ohýbá“ prostor i nad horizontem událostí a díky tomu dochází v jejím okolí k efektu gravitační čočky. Tato vlastnost je velmi zajímavá pro vizualizace, avšak její realistické znázornění je výpočetně náročné, proto pro znázornění ve virtuální realitě budou použita předpočítaná mapovací data založená na matematické knihovně SIM5[1].

Virtuální realita přináší uživateli vysokou míru imerze, pomocí které se prohlubuje zážitek z prostředí, kterému je vystaven. Takto mohou být uživateli představeny efekty gravitační čočky černé díry způsobem, kterým získá intuici pro chování prostoru v její blízkosti.



Obrázek 1.1. První fotka skutečné černé díry.

Kapitola 2

Analýza

2.1 Obecná relativita

Obecná teorie relativity poskytuje rámec pro modely struktury vesmíru, jeho vývoje a konečného osudu. Předpovídá, že vesmír se musí buď rozpínat nebo smršťovat. Jejím vytvořením se prostor a čas staly dynamickou entitou, jenž je ovlivněna přítomností hmoty a energie. Právě tento vztah způsobuje zakřivení časoprostoru, které je nejdůležitějším konceptem teorie relativity. Všechny hmotné objekty zakřivují prostor, avšak znatelný efekt je možné pozorovat až u velmi hmotných a zároveň velmi kompaktních těles jako jsou právě černé díry. Deformace časoprostoru, pak ovlivňuje trajektorie pohybu všech částic (fotonů nevyjímaje) skrze okolí těchto těles. Z pohledu pozorovatele se může zdát, že částice urazí delší vzdálenost než by musela, avšak z pohledu částice jde o přímou trasu, kterou nazýváme geodetika. Více se můžete dozvědět například v [2].

2.2 Vlastnosti černých děr

Černé díry mohou být definovány pomocí pouze třech čísel a to: hmotnosti, momentu hybnosti a elektrického náboje. Pro její vizualizaci je však důležité znát pouze hmotnost a rotaci.

Efekt gravitační čočky je fenomén, kdy fotony prochází skrz silné gravitační pole a jejich trajektorie jsou tak značně zakřiveny. Fotony z jednoho objektu se tak mohou pohybovat po dvou či více různých cestách a dochází tak k vícenásobnému zobrazení objektu pozorovateli na druhé straně pole. Deformace obrazu připomíná pak efekt optické čočky, odtud název gravitační čočka.

Červený a modrý posun je druhý efekt, který je vhodné zmínit, avšak zatím není cílem ho implementovat do vizualizace. Pokud se zdroj záření vzhledem k pozorovateli pohybuje, dochází k frekvenčnímu posunu spektra. Pokud se zdroj a pozorovatel oddalují dochází k červenému posunu, v opačném případě jde o modrý posun. Při průletu fotonu kolem černé díry dochází k posunu i kvůli zpomalení času silnou gravitací.

2.3 Virtuální realita

Virtuální realita nebo zkráceně VR je technologie, která zprostředkovává přenesení uživatele do umělého prostředí, se kterým může interagovat. Pro účely zvýšení míry imerze uživatele do virtuálního světa jsou využívány různé prostředky, které nahrazují lidské vjemy a umocňují tak přirozenost daného prostředí.

Pro upřesnění pojmu imerze, bych uvedl její definici z knihy Jasona Jeralda[3], která se věnuje kompletnímu rozboru návrhu, vývoje a testování VR aplikací:

*Imerze je objektivní úroveň míry, do které je virtuálně realitní aplikace schopná předávat podněty smyslovým receptorům uživatele způsobem, který je extezivní, odpovídá ostatním smyslům, je živý, interaktivní a dodává uživateli informace.*¹

Podobnou technologií k VR je smíšená realita či „mixed reality“, liší se od ní především tím, že některé vjemy, které jsou uživateli předávány, jsou založené na skutečném světě. Podle poměru reálného a virtuálního světa, dělíme smíšenou realitu na rozšířenou realitu a rozšířenou virtualitu. Rozlišuje se také způsob, jak je smíšená realita vytvořena – přímá a nepřímá metoda. Přímá metoda využívá nejčastěji brýlí s poloprůhledným displejem, který předává uživateli informace o rozpoznávaných prvcích v reálném světě. Nepřímá metoda využívá záznamového zařízení, skrze které se uživatel na svět dívá v jeho upravené podobě. Podrobnější informace lze nalézt v knize Dietera Schmalstiega a Tobiase Hollera[4].

Náhlavní displeje, které zprostředkovávají imerzi uživatele do virtuální či smíšené reality, pak dělíme do třech kategorií podle výpočetního prostředku, který je využit pro vytvoření scény. Jsou to takzvané „tethered“ displeje, které jsou pomocí HDMI či USB kabelu spojené s externím počítačem, který zajišťuje výpočet a vykreslení scény. Dále jsou to „standalone“ či „all-in-one“ displeje, které, jak název napovídá, nejsou vázané na žádné externí zařízení a veškerý výpočetní výkon je zajištěn samotným displejem. Posledním typem jsou mobilní VR sety, které se dají pořídit jako rozšíření k chytrému telefonu. Jde o nejlevnější možnost, jak si zajistit zážitek z virtuální reality, avšak možnosti jsou silně omezené výkonem a kvalitou displeje mobilního telefonu.



Obrázek 2.1. Acer Windows Mixed Reality Headset AH101.

2.4 Analýza problematiky

Předmětem této práce bude vytvoření virtuální scény s černou dírou pro dva uživatele s využitím dat sférického videa. Pro černou díru budeme uvažovat její nejjednodušší model, tedy Schwarzschildovu černou díru, která nemá elektrický náboj ani moment hybnosti. Tento fakt značně zjednoduší paměťovou i výpočetní náročnost aplikace, jelikož černá díra včetně prostoru okolo ní budou sféricky symetrické.

Stěžejním problémem této práce bude vhodný způsob získávání dat pro deformaci obrazu a vytváření efektu černé díry. Pro virtuálně realitní aplikace je zásadní dosažení co největší plynulosti a rychlosti zobrazování. Nižší výpočetní náročnost scény umožňuje potenciální využití standalone VR headsetu. Plynulostí zobrazování se předchází také

¹ Originál definice: *Immersion is the objective degree to which a VR system and application projects stimuli onto the sensory receptors of users in a way that is extensive, matching, surrounding, vivid, interactive, and plot informing.*

kinetóze, která by v opačném případě mohla u uživatele nastat. Kinetóza je obecně stavem, při kterém vjem pohybu získaný vestibulárním systémem nesouhlasí s vizuálními vjemy. Výsledkem toho může být uživateli nevolno či ho může začít bolet hlava¹.

Pro vykreslování černé díry bude vhodné implementovat řešení založené na dříve vypočítaných datech, která budou ve scéně interpretována postupem, kterým vznikne plynulý a realistický model černé díry. Další možností je pak využití aproximační rovnice, která vytváří data, která se blíží skutečnému efektu gravitační čočky černé díry.

Pro použití sférického videa ve scéně bude nutné vytvořit systém, který bude schopný vykreslovat sférické video jako pozadí virtuálního prostředí. Samotné zdrojové video kamery pak bude tímto systémem zpracováno a dále vykreslováno do pozadí scény.

Dalším problémem je napojení a synchronizace více uživatelů v jedné scéně. Za tímto účelem bude nutné zvolit formu síťové komunikace mezi uživateli a vytvořit systém, který bude synchronizovat a zobrazovat jejich pohyby ve scéně. Důležité bude také sjednotit pozici uživatelů ve virtuální realitě vůči reálnému světu. Toho může být docíleno například díky fixnímu známému bodu, na které se umístí VR headsety a nastaví se jejich virtuální pozice na tento známý bod, například střed místnosti, vůči tomuto bodu následně posuneme virtuální scénu.

2.5 Sférická kamera a video

Sférická kamera či 360° kamera je speciální typ záznamového zařízení, které umožňuje pořizování panoramatických videí ze všech směrů vycházejících z pozice kamery. Tato vlastnost je vhodná především pro vykreslování skybox textury, která slouží jako pozadí virtuálního světa.

Zdaleka nejčastěji používanou projekcí pro sférické video je ekvidistantní válcové zobrazení, které zobrazuje sférické souřadnice do plochy, kde na ose x je souřadnice zeměpisné délky a na ose y je souřadnice zeměpisné šířky. Dalšími formáty jsou pak cube map videa, pyramidová projekce vyvinutá Facebookem² pro optimalizaci velikosti video souboru či rovnoramenná cube map projekce, kterou vyvinul Google³ za účelem rovnoměrné distribuce pixelů v obraze nezávislé na pohledu uživatele. Samotný video záznam pak může být pořízen monoskopicky (jediný záznam pro obě oči) či stereoskopicky, kdy je ukládán pohled z každého oka zvlášť, čímž je vytvářen prostorový efekt.



Obrázek 2.2. Sférická kamera Samsung Gear 360.

¹ <https://virtualspeech.com/blog/motion-sickness-vr>

² <https://engineering.fb.com/2016/01/21/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/>

³ <https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/>



Obrázek 2.3. Sférický pohled pořízený pomocí Samsung Gear 360.

2.6 Možnosti vizualizace

2.6.1 Ray tracing

Ray tracing je vysoce náročný proces, pomocí kterého je možné přesně spočítat pohyb paprsků světla skrze daný prostor. Tato metoda může být implementována různými způsoby za účelem vizualizace černé díry. Jednou možností je vytvoření geodézického ray-tracingového algoritmu, který sleduje trajektorie paprsků světla v okolí černé díry v Schwarzschildově prostoru. Světelné paprsky jsou určeny bodem z něhož vycházejí a 4-dimenzionálním vektorem hybnosti, kde čtvrtou složku tvoří čas. Tento postup byl použit v práci Ivana Platonova[5].

Hlavním problémem pro použití ray tracingu, je nutnost pro jeho specifickou implementaci při vizualizaci černé díry, která by ve spojení s virtuální realitou, představovala mnoho implementačních neduhů.

2.6.2 Mapovací data

Problém náročné implementace ray tracingu pro VR může být vyřešen pomocí předpočítání dat mimo VR. Data jsou následně použita standardními způsoby pro deformaci obrazu v post-processingovém shaderu. Tímto postupem však není možné dosáhnout absolutně realistického výsledku, jelikož pro to by bylo nutné vytvořit mapovací data pro všechny možné vzdálenosti a inklinace kamery od černé díry. Vhodným způsobem interpolace se sníží množství dat nutných pro vytvoření realistického zážitku ve VR, čímž se sníží i paměťová náročnost programu. Podobný postup popsali a využili Anne-mieke Verbraeck a Elmar Eisemann[6].

2.6.3 Aproximační data

Poslední možností je pak využití aproximační funkce, která má výrazně nižší výpočetní náročnost oproti ray tracing metodě a je proto možné ji použít ve VR scéně. U aproximace, jak je již z významu slova patrné, se bude jednat o odhad chování paprsků v okolí černé díry. Přiblížení skutečné deformaci, kterou aproximační funkce zprostředkovávají má své limity, které jsou dané především intervalem vstupních hodnot, pro které daná

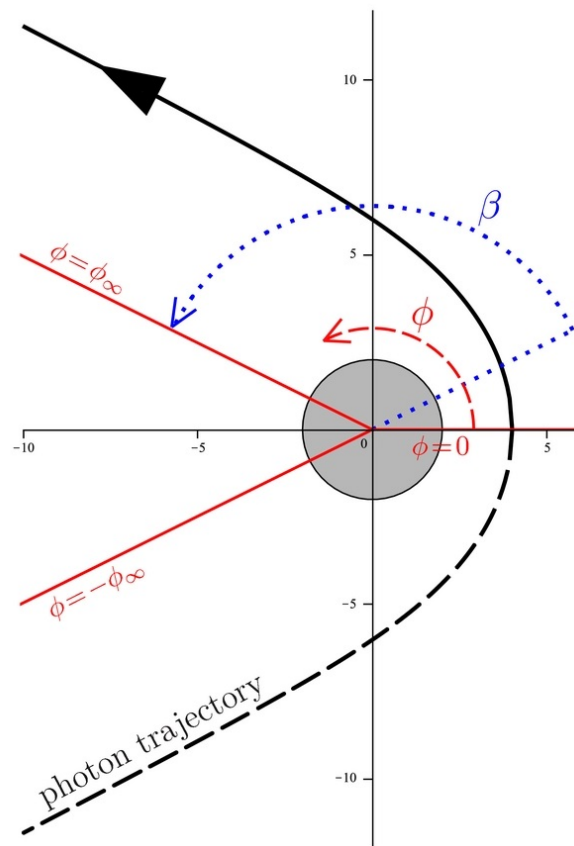
funkce funguje nejlépe. Aproximační funkce a jejich vlastnosti popsal ve článku Oldřich Semerák [7]. Pro účely této práce bude pak důležitá rovnice:

$$\cos\phi = \frac{r_0}{r} - \frac{M}{r_0 - \alpha M} \frac{(r - r_0)(2r + r_0)}{(r - \omega M)^2}, \quad (1)$$

kde pro M , jenž je hmotností černé díry, je uvažováno 1, čímž se získá jednodušší rovnice:

$$\cos\phi = \frac{r_0}{r} - \frac{1}{r_0 - \alpha} \frac{(r - r_0)(2r + r_0)}{(r - \omega)^2} \quad (2)$$

Výsledkem aproximace je cosinus úhlu ϕ , pro určení distorze obrazu je potřebný úhel β , viz. ilustrace 2.4. Efekt této rovnice porovnám s předvypočtenými daty z ray tracingové metody. Zbylé proměnné rovnice popíší níže.



Obrázek 2.4. Ilustrace aproximačního výpočtu distorze černé díry. Úhel β určuje míru distorze dráhy fotonu. β může být určena rovnicí $\beta = 2\phi - \pi$, což platí díky symetrii dráhy letu fotonu. Ilustrace převzata z [7].

Pro správné fungování aproximace je nutné korektně určit všechny vstupní parametry. Těmi jsou r , které značí vzdálenost pozorovatele od černé díry, a r_0 , které je vzdáleností v bodu obratu radiálního pohybu. Dále jsou to pak konstanty α a ω , jenž dále upravují vlastnosti aproximační funkce. Hodnoty pomocí, kterých se aproximace blíží nejvíce skutečné deformaci způsobené černou dírou, byly určeny jako 1,77 pro α a 1,45 pro ω . r_0 lze určit ze směrového vektoru paprsku a rovnice,

$$r_0 = b\sqrt{1 - \frac{2}{r_0}}, \quad (3)$$

kde

$$b = \frac{|L|}{E} = \sin \alpha \frac{r}{\sqrt{1 - \frac{2}{r}}} \quad (4)$$

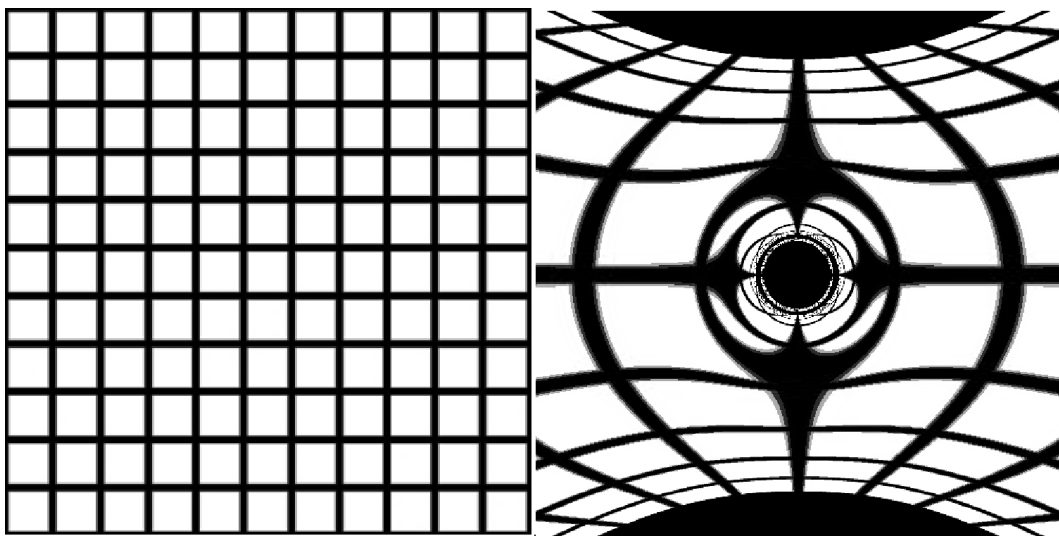
Kde L je moment hybnosti fotonu, E je energie fotonu a samotné b je naměřený parametr trajektorie fotonu. α je úhel svíraný směrem letu fotonu v počáteční pozici a černou dírou. Dalšími úpravami získáme rovnici:

$$r_0 = \frac{2b}{(3\sqrt{3})^{1/3}} \cos\left(\frac{\arccos\left(\frac{-3\sqrt{3}}{b}\right)}{3}\right) \quad (5)$$

Výslednou rovnici je již možné použít pro výpočet aproximace.

2.7 Výchozí stav

Bakalářská práce staví na knihovně SIM5[1] a programu BH Warp, který umožňuje výpočet tras fotonů v Kerrově prostoru okolí černé díry. Pozice fotonu po průchodu prostorem je zachycena na virtuální ploše a uložena do transformační matice. Data pak mohou být použita k deformaci 2D obrazu, přesunem pixelů obrazu na pozice, na které by se pixel (foton) dostal při průletu skrze zakřivený prostor silného gravitačního pole.



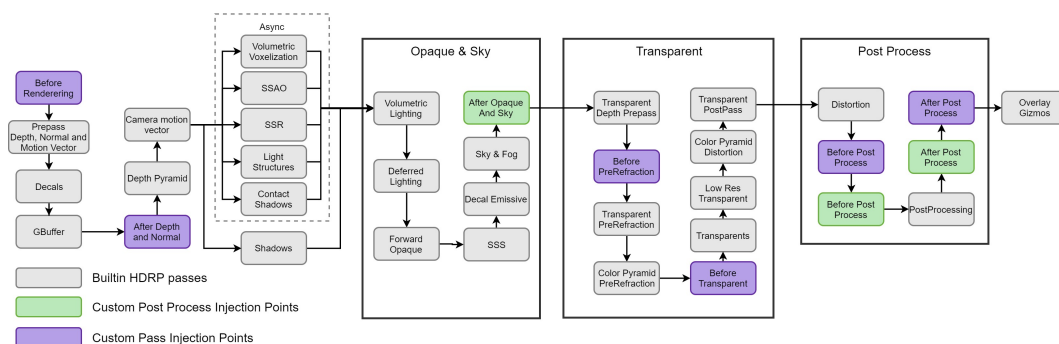
Obrázek 2.5. Vlevo se nachází nedeformovaná testovací mřížka. Vpravo je na testovací mřížku aplikována deformace obrazu programem BH Warp.

2.8 Unity

Unity je herní engine, který umožňuje vývoj na různých platformách. Slouží k vytváření her, simulací a dalších typů projektů ve 2D, 3D, virtuální i rozšířené realitě. Mimo podporu hlavních platforem a typů projektů, zprostředkovává Unity i volbu vykreslovacího řetězce podle grafických požadavků vývojáře. Vykreslovací řetězec či render pipeline provádí sérii operací, při kterých bere obsah zvolené scény a určitým způsobem je vykresluje na obrazovku cílového zařízení. Základními operacemi zobrazování scény jsou ořezávání (culling), vykreslování (rendering) a post processing.

Unity nabízí 4 typy vykreslovacích řetězců:

- Základní build-in řetězec, který slouží k obecnému využití, avšak umožňuje pouze omezené možnosti úprav a rozšíření do vykreslovacího cyklu.
- Scriptable render pipeline, která umožňuje vytvoření zcela vlastního řetězce pro speciální potřeby projektu. Dovoluje tak dále optimalizovat výkonnost a způsob vykreslování.
- Universal render pipeline je předpřipravená verze scriptable řetězce, která zprostředkovává jednoduchý a rychlý způsob pro přizpůsobování vykreslovacího řetězce bez nutných úprav pro jednotlivé platformy.
- High Definition render pipeline je také předpřipravená verze scriptable řetězce, která se soustředí na vysoké rozlišení a nejnovější technologie pro platformy s vyšším výpočetním výkonem.



Obrázek 2.6. Diagram průběhu Unity HDRP řetězce.

2.9 Rešerše a jiné projekty

Projektů, jejichž cílem bylo vytvořit interaktivní model černé díry, lze nalézt poměrně mnoho. Například již zmíněná práce Ivana Platonova[5], která používá real-time ray-tracing pro simulaci libovolného typu černé díry. Projekt Otta Seiskara[8] je také real-time simulací černé díry s použitím ray-traycingu, avšak pouze jejího nejjednoduššího modelu - Schwarzschildovy černé díry. Díky tomuto zjednodušení je možné realizovat výpočet ve webovém prohlížeči¹. Řešení problému použité Riccardem Antonellim[9] je velmi podobné předchozímu projektu, avšak v tomto případě jsou data pro deformaci obrazu předem vygenerována. Projekt je možné také vyzkoušet online². Posledním projektem, který zmíním, je práce Annemieke Verbraeck a Elmar Eisemanna[6], kteří pro svou vizualizaci využívají předem vygenerovaná data, která simulují efekt libovolné černé díry. Data ukládají do adaptivní sférické textury, čímž získávají možnost generovat informace v různé hustotě dle pozice vůči černé díře.

Stejně tak je možné nalézt množství vizualizací pro virtuální realitu, například od Jordyho Davelaara[10]. Avšak kombinaci těchto dvou aspektů se mi nepodařilo v jiném veřejném projektu nalézt, tudíž je možné, že tato práce je v tomto ohledu skutečně jedinečná.

Projekt, který by využíval pro vizualizaci černé díry aproximační funkci, se mi nepodařilo nalézt.

¹ <https://oseiskar.github.io/black-hole/>

² <http://spiro.fisica.unipd.it/~antonell/schwarzschild/live/>

Kapitola 3

Návrh řešení

V této části se zaměříme na aplikační design, požadavky a způsoby, jakými je vyřešit v rámci samostatného projektu.

3.1 Požadavky

Finální verze aplikace má být použitelná pro vzdělávací a prezentační účely Astronomického ústavu Akademie věd ČR. Proto by měla být výsledná aplikace uživatelsky přívětivá a interaktivní.

Parametry, které musí aplikace obsahovat:

- 360-stupňové video zobrazované jako skybox
- Efekt černé díry korektně zobrazený ve VR
- Vykreslování efektu černé díry podle pohledu uživatele
- Podpora pro 2 či více uživatelů ve scéně
- Nízká výpočetní náročnost aplikace

3.2 Platforma

Pro vývoj jsem zvolil herní engine Unity pro Windows. Důvodem k tomu je především dřívější zkušenost s tímto vývojovým prostředím a také široká podpora různých zařízení pro VR. Výsledný projekt tudíž bude možné spustit, jak na Windows Mixed Reality headsetu, tak i na HTC Vive či Oculus Quest bez nutnosti provádění změn ve zdrojovém kódu. Volbou herního enginu byly zvoleny i programovací jazyky. Unity využívá především C# s výjimkou shaderů, které používají HLSL.

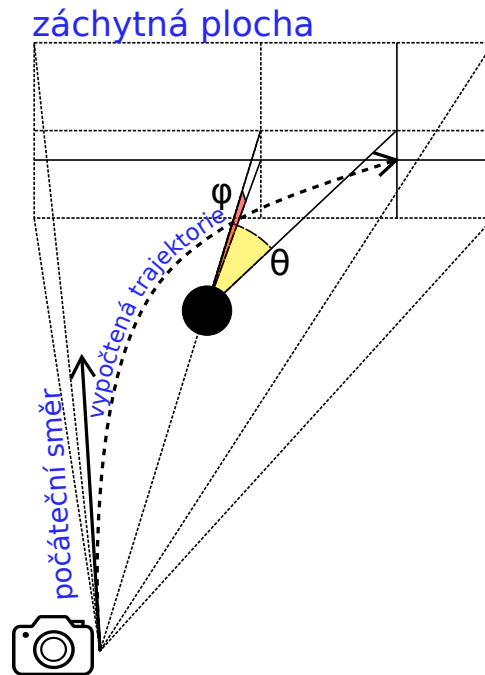
Pro vykreslovací řetězec v rámci Unity jsem zvolil High Definition render pipeline, která umožňuje jednoduché vytvoření post processingového shaderu s knihovnamí potřebnými pro zobrazení výsledného efektu černé díry.

Pro testování korektního fungování aplikace bude použit Acer Windows Mixed Reality Headset AH101, obrázek 2.1, který mi byl zapůjčen katedrou počítačové grafiky a interakce. Vytváření dat pro efekt černé díry bude v programu napsaném v C s knihovnamí pro Linux, jelikož je základ tohoto programu již připraven a jeho převádění do Windows a Unity by bylo zbytečně náročné.

3.3 Příprava dat

Základem projektu je vytvořená Linux aplikace BHWarp pro generování mapovacích dat pro efekt černé díry. Data je možné generovat podle několika základních parametrů, a to vzdálenosti a výšky kamery vůči černé díře, rotace černé díry a velikosti výsledné transformační mapy. Aplikace pak iteruje přes jednotlivé paprsky s různými počátečními směrovými vektory a vypočte jejich trasu skrze okolí černé díry. Pokud daný foton není

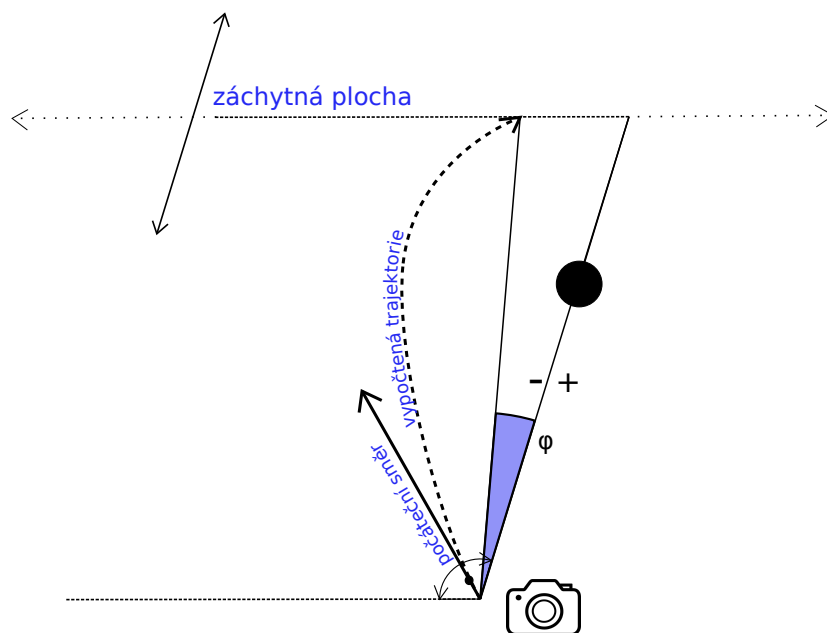
pohlčen černou dírou, je v efektivním nekonečnu uložena jeho pozice vůči černé díře ve sférických souřadnicích.



Obrázek 3.1. Ilustrace generování dat v BHWarp. φ a θ jsou úhly, které byly ukládány původním programem.

Pro vytvoření dat pro VR scénu bude potřeba udělat v programu několik změn. Jelikož uvažujeme model Schwarzschildovy černé díry je možné data značně redukovat. Sférická symetrie zajišťuje, že stačí vypočítat trasy paprsků, které mají nulovou vertikální hodnotu směrového vektoru a horizontální hodnotu směrového vektoru v intervalu $\langle 0; \frac{\pi}{2} \rangle$ tedy $\langle 0^\circ; 90^\circ \rangle$. Data následně mohou být rotována okolo spojnice kamery a černé díry, čímž se získá deformace všech paprsků. Touto úpravou se sníží nutné množství dat na 1-dimenzionální pole, kde dříve bylo nutné pole 2-dimenzionální.

Další úpravou je pak přidání parametru pro pozici plochy, na níž se bude zastavovat výpočet trasy a ukládat pozice fotonu. Tato pozice se bude ukládat vůči kameře, nikoli vůči černé díře, jak tomu bylo v původním programu.



Obrázek 3.2. Ilustrace upraveného způsobu generování dat. Přes počáteční směr fotonu je iterováno od směru kolmého k černé díře až po směr, který míří přímo do černé díry. φ je ukládaným úhlem, kdy φ je negativní, pokud dopadne vlevo od černé díry, a pozitivní, pokud dopadne vpravo.

3.4 Struktura scény

Aplikace se bude skládat ze dvou scén. První scéna bude obsahovat hlavní menu s jednoduchým nastavením, informacemi o aplikaci a jejím ovládání a možností aplikaci ukončit či vstoupit do hlavní scény s vizualizací černé díry.

Druhá a hlavní scéna bude obsahovat vizualizaci černé díry, spolu s nastavením vizualizace.

3.4.1 Vizualizace černé díry

Data pro vykreslování černé díry budou v Unity dále zpracována do formátu vhodného pro shader. Upravená data budou načtena skriptem pro řízení zobrazování efektu černé díry. Tato data jsou pak předána post-processingovému shaderu spolu s informacemi o velikosti dat, pozici uživatele a dalšími parametry. V samotném shaderu bude pak obraz deformován podle dostupných dat pro vytvoření efektu černé díry.

Aproximační vizualizace bude taktéž realizována pomocí post-processingového shaderu, jemuž budou předávány parametry řídicím skriptem.

3.4.2 Sférické video

Pro zobrazování sférického videa do skyboxu bude vytvořen shader, který vykreslí 2D texturu 360 stupňového záběru v ekvidistantní projekci do pozadí scény. Samotné video bude pak vykreslováno do textury, která bude zmíněným způsobem zobrazena do skyboxu.

■ 3.4.3 Repräsentace uživatele

Uživatel bude ve scéně reprezentován modelem skafandru, do kterého bude virtuálně přemístěn. Pozice modelu bude vázaná na pohyby náhlavního displeje. Dále pak budou vázány pozice levého a pravého ovladače na pohyby rukou. Realistické pohyby budou vytvořeny pomocí inverzní kinematiky. Inverzní kinematika je metoda, při které jsou vypočteny rotace jednotlivých kloubů kostry nutných pro dosažení dané pozice koncové části modelu. V tomto případě je to pozice ovladače, na kterou je vázaná koncová pozice ruky. Více informací je možné nalézt například zde [11].

Pro maximalizaci dynamičnosti uživatelského skafandru bude také přidána statická animace nohou modelu.

Objekt uživatele bude obsahovat hlavní kameru, která se bude pohybovat podle pohybu VR headsetu používaného uživatelem. Hlavní kamera bude používat post-processingový shader, který bude vykreslovat černou díru do scény.

■ 3.4.4 Systém pro více uživatelů

Aplikace bude podporovat 2 uživatele v hlavní scéně. Při připojení uživatele do scény bude zjištěno, zda existuje volné místo v nějaké již existující síťové místnosti. Pokud tomu tak nebude, bude vytvořena nová místnost. Skrze tu bude vytvořena a sdílena instance modelu, který bude reprezentovat uživatele. Instance bude pak zobrazována všem uživatelům ve scéně, ale pouze uživatel, který instanci vytvořil bude moci měnit její pozici. Tímto způsobem budou sdíleny pohyby jednotlivých uživatelů.

Kapitola 4

Implementace

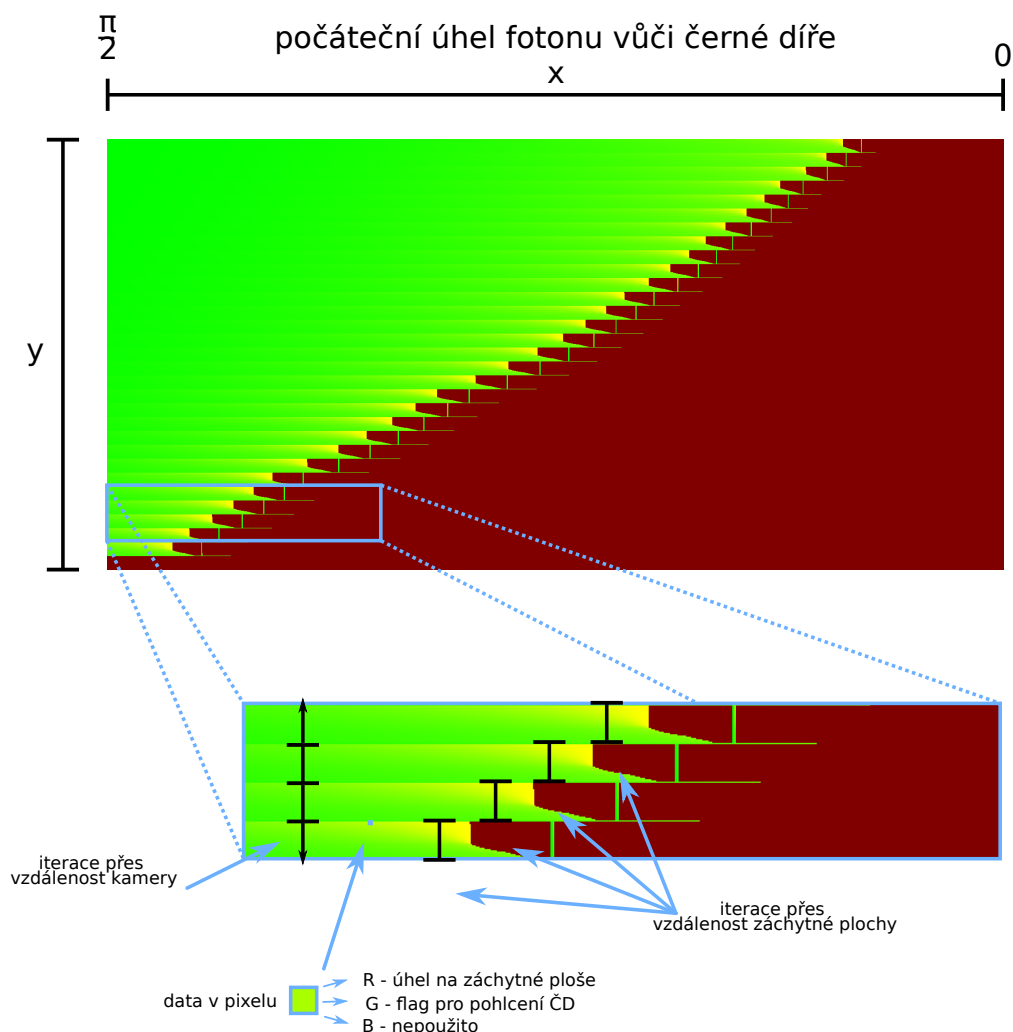
V této kapitole popíšeme způsob, jakým byl návrh řešení aplikace uskutečněn. Také se zaměřím na její vnitřní fungování a uživatelské rozhraní.

4.1 Generování mapovacích dat

Program BH Warp, pro generování dat za účelem vytváření efektu černé díry, ukládal informace o výsledné pozici fotonu ve sférických souřadnicích. Tyto souřadnice pak byly převedeny do UV souřadnic v shaderu, který vykresloval černou díru pomocí gnómonické projekce. Tato struktura dat není pro použití v Unity vhodná především ze dvou důvodů. Prvním je obtížné předání velkého množství dat s pohyblivou řádovou čárkou, které jsou mimo interval $\langle 0; 1 \rangle$. Druhým důvodem je ztráta přesnosti během výpočtu, který v původním programu využívá datového typu double, který má dvojnásobnou přesnost oproti typu float, který je nutný pro předání dat shaderu v jazyce HLSL. Tyto problémy bylo možné eliminovat použitím nové struktury, kdy je ukládán sinus úhlu od spojnice pozorovatele a černé díry k výsledné pozici fotonu.

Po vložení dat do Unity projektu jsou vytvořená data před použitím ve scéně konvertována skriptem BytesConverter z binární reprezentace do EXR formátu, který již představuje standardní formát pro textury v grafice. Pro převedení je nutné provést konverzi z intervalu $\langle -1; 1 \rangle$, jenž je oborem hodnot funkce sinus, na interval $\langle 0; 1 \rangle$, jehož hodnoty je možné uložit do EXR formátu. Důležitou vlastností EXR formátu je vysoký dynamický rozsah barev, což znamená, že je možné ukládat jednotlivé kanály barev až jako 32-bitová čísla, což je dvakrát více než limit pro PNG formát, který umožňuje maximálně 16 bitů pro barevný kanál.

Pro vytvoření efektu černé díry byla vygenerována mapa dat s různými vzdálenostmi pozorovatele a zachytné plochy od černé díry. Posun mezi jednotlivými vzdálenostmi použitými pro generování dat nebyl lineární, jelikož změna, ke které dochází, se zvyšuje s blízkostí k černé díře. Data pro vzdálenosti blízké černé díře mají tak vyšší hustotu než ta, která se blíží efektivnímu nekonečnu. Díky úpravě struktury generovaných dat, umožněné uvažováním pouze Schwarzschildovy černé díry, jsou data pro dané parametry redukována pouze na 1-dimenzionální pole, kde v původním programu bylo nutné pole 2-dimenzionální. Důsledkem toho bylo možné efektivně a kompaktně uložit veškerá data do jediné textury.



Obrázek 4.1. Zpracovaná textura mapovacích dat.

4.2 Unity scéna

V první fázi práce byl projekt implementován v Unity s Universal render pipeline. Avšak nebylo možné u této implementace zůstat, jelikož podpora post-processing VR shaderů nebyla dostačující. Hlavním limitem bylo špatné navázání do vykreslovacího řetězce a absence separátních transformačních matic pro levé a pravé oko. U nových verzí¹ URP dokonce není implementována žádná podpora vlastních post-processing efektů.

V důsledku toho byl skrze Unity Hub vytvořen nový projekt s High Definition render pipeline ve verzi 10.3.2. Verze Unity zvolená pro projekt je 2020.3.0f1. Do projektu byly přidány balíčky:

- XR Plugin Management, který se stará o správu VR a AR subsystémů.
- Windows XR Plugin, který zajišťuje podporu pro Windows Mixed Reality zařízení, které bylo zvoleno pro testování.
- Photon Unity Networking 2, který zprostředkovává jednoduché vytvoření síťového spojení více uživatelů.

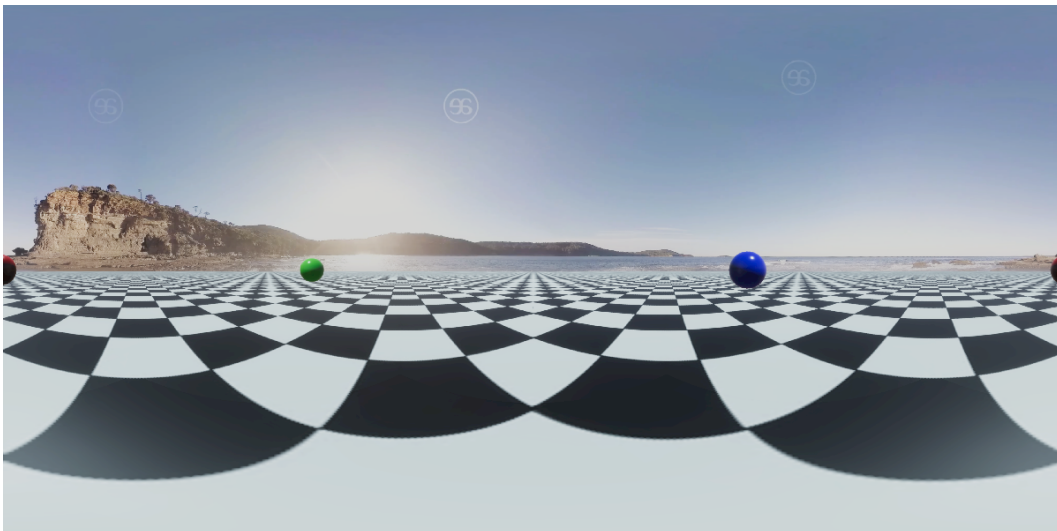
¹ Vztahuje se na verze 8.0. až 12.0.

- XR Interaction Toolkit, který usnadňuje implementaci ovládaní pohybu a reprezentace uživatele ve VR
- uEcho - malý balíček usnadňující ladění síťového spojení.

Projekt se skládá ze dvou scén, kde první scéna obsahuje hlavní menu s ovládáním, informacemi o projektu a možnostmi pro její ukončení a vstup do hlavní scény. Hlavní scéna obsahuje vizualizaci černé díry s nastavením, které umožňuje vybrat způsob vizualizace. Do scény byl přidán objekt reprezentující pozici černé díry a XR Rig reprezentující uživatele a objekty pro řízení zobrazování černé díry.

4.2.1 Získání sférického pohledu kamery

Pro potřeby vizualizace je nutné získat i pohled mimo základní pohled uživatele. Z toho důvodu byla do scény přidána druhá kamera, která je vázána na pozici hlavní kamery, jenž pořizuje pohled ze všech směrů. Sférický pohled je vykreslován do cube map textury, která je následně transformována do textury v ekvidistantní válcové projekci. Tato textura slouží pak jako jeden z parametrů pro vykreslování černé díry.



Obrázek 4.2. Sférický záběr z testovací scény.

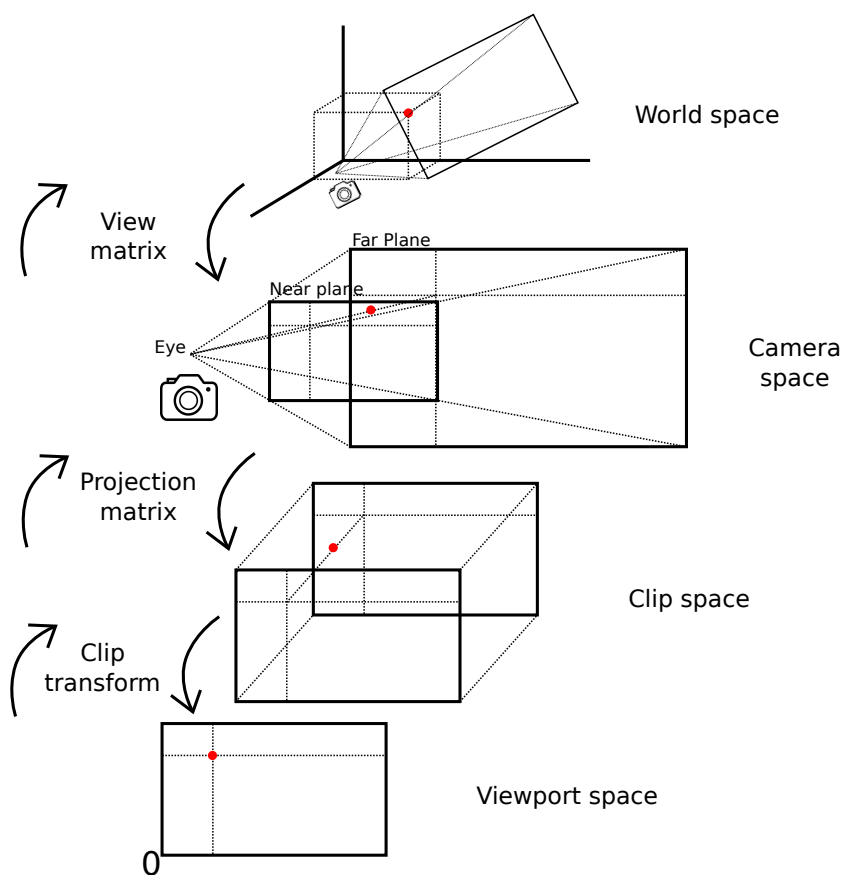
4.3 Vykreslování černé díry

Použitím interního systému v High Definition render pipeline pro vkládání vlastních shaderů do vykreslovacího řetězce byly vytvořeny dva post-processingové shadery. BHEffect shader vytváří vizualizaci černé díry z vygenerovaných dat, BHApprox shader pomocí aproximační funkce. Pro řízení vykreslování byl vytvořen skript BHManager. Tento načítá mapovací data a určuje, která se mají použít. Dále upravuje parametry shaderů dle nastavení a aktuální pozice uživatele.

Do každého post processingového shaderu je zasílán obraz, jenž by byl vykreslen, pokud by shader nebyl vsazen do vykreslovacího řetězce. Účelem shaderu je pak tento obraz vhodným způsobem modifikovat. Samotný soubor shaderu se dělí na vertex a fragment shader. Pro vykreslování černé díry je důležitý pouze fragment shader, který bude volán zvlášť pro každý pixel obrazovky cílového zařízení.

Při zavolání fragment shaderu obou vizualizací černé díry dojde nejdříve ke kontrole, zda je daný fragment/pixel ve výřezu, kde by měl být efekt vykreslován. Toho

je docíleno převodem souřadnic viewportu do světových souřadnic. To je možné pomocí inverzních matic perspektivní projekce a transformace do prostoru kamery, viz ilustrace 4.3. Ze souřadnic světového prostoru lze určit úhel, který daný pixel svírá s černou dírou, pomocí skalárního součinu. Ze získaného úhlu je možné již jednoduše určit, zda se má efekt použít či nikoliv, tj. pokud je úhel v absolutní hodnotě menší než 90 stupňů je možné efekt použít. Tento limit je určený strukturou generovaných dat, tak i výstupními daty aproximace. Z bodu pixelu ve světových souřadnicích je také možné určit osu využitím vektorového součinu. Okolo této osy by měl být bod rotován deformací prostoru černou dírou. Do vektorového součinu je vložen bod pixelu a bod černé díry, jež jsou posunuty o pozici kamery, čímž se získají vektory, k nimž se hledá kolmý vektor osy rotace.



Obrázek 4.3. Řetězec souřadnicových transformací.

■ 4.3.1 Logika shaderu pro model černé díry

Využitím vypočteného úhlu svíraného vykreslovaným pixelem a černou dírou spolu s vzdáleností uživatele od černé díry lze hrubě určit data pro deformaci obrazu. Úhel určuje horizontální souřadnici a vzdálenost vertikální souřadnici v textuře, viz ilustrace 4.1. Pro ideální fungování vizualizace by bylo nutné určit i vzdálenost od černé díry, ve které se má nacházet bod po deformaci obrazu, to však není možné určit v post processingovém shaderu. Tento problém by bylo možné v budoucnu překlenout použitím určité formy ray tracingu. Důsledkem tohoto problému je, že pohled uživatele musí být deformován daty, která předpokládají, že vše mimo černou díru se nachází v efektivním nekonečnu od uživatele. Aby byl problém kompenzován, je efekt vypnutý pro objekty nacházející se v bezprostřední blízkosti od uživatele.

Pomocí zmíněných parametrů jsou určeny dva nejbližší datové body, mezi nimiž je pak interpolováno. Ze získaných mapovacích dat jsou odděleny jednotlivé složky barvy. R udává nový úhel, který má bod svírat s uživatelem a černou dírou, G složka pak říká, zda trajektorie fotonu končí na horizontu událostí či nikoliv. B složka není využita. Pokud je G složka rovna nule, byl foton pohlcen a je tudíž vykreslen jako černý. V opačném případě je pomocí získaného úhlu rotován bod ve světových souřadnicích okolo dříve vypočtené osy. Tento nový bod je následně opět převeden buď do souřadnic viewportu, kde je získána nová barva pro vykreslení, nebo, pokud bod leží mimo původní pohled uživatele, je transformován do sférických souřadnic, které slouží k získání barvy ve sférickém pohledu kamery. Na závěr je původní a nová barva smíchána v poměru dle vzdálenosti od středu černé díry, čímž je vytvořen hladký okraj kolem vykreslovaného efektu.

4.3.2 Logika shaderu pro aproximaci černé díry

Pro použití aproximační funkce je potřeba určit vzdálenost periastra trajektorie r_0 , k tomu poslouží úhel pixelu svíraný s černou dírou a vzdálenost od pozorovatele od černé díry. Tyto hodnoty se dosadí do rovnic popsanych v analýze aproximačních dat 2.6.3. Výstupem aproximační funkce je pak cosinus úhlu ϕ , z něhož se získá deformační úhel jako $\beta = 2\phi - \pi$, kde β je hledaný úhel. Výsledný pohled deformovaný aproximací efektu gravitační čočky černé díry je získán stejným způsobem jako u předvypočtených dat, tedy je získána barva buď z původního pohledu či ze sférického pohledu.

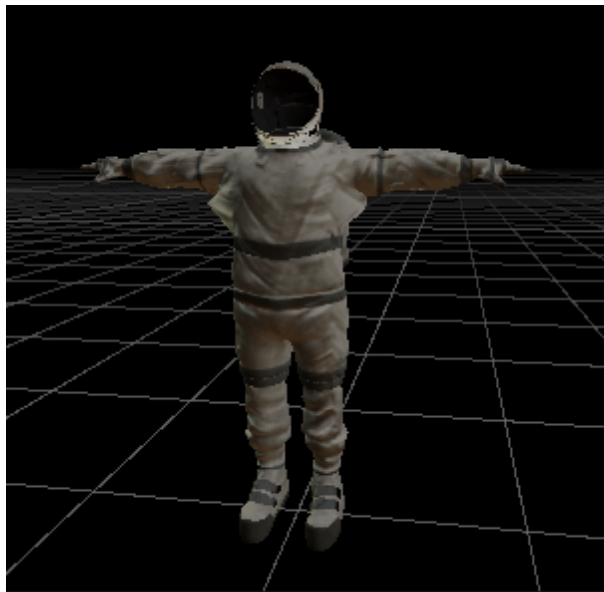
4.4 Sférické video

Proces vykreslování videa do skybox textury byl zajištěn následujícím postupem. Nejdříve byla vytvořena nová vykreslovací textura s velikostmi stran odpovídajícími rozměrům testovacího videa. Do této textury pak vykresluje obraz herní objekt s komponentou video přehrávače, v němž je vložené sférické video. Dále byl vytvořen sky shader, který přijímá tuto texturu jako parametr. Shader pak převodem bodu pozadí na sférické souřadnice určí barvu z textury videa v ekvidistantní válcové projekci. Barva je pak upravena parametrem expozice, čímž je možné modifikovat jas videa.

4.5 Síťové spojení a reprezentace uživatele

Pro zajištění síťového spojení a synchronizace scény byl použit balíček Photon Unity Networking, zkráceně PUN. PUN umožňuje síťovou synchronizaci scén na bázi místností. Při připojení uživatele do scény je zjištěno, zda existuje volné místo v nějaké již existující síťové místnosti, pokud tomu tak není je vytvořena nová místnost. Prostřednictvím místnosti je sdílena pozice náhlavních displejů a ovladačů, které jsou následně použity pro animaci pohybů uživatelských reprezentací.

Uživatel je ve scéně reprezentován modelem astronauta, jehož pohyby jsou vázány na pohyby náhlavního displeje a ovladačů. Pro animaci modelu byla vytvořena komponenta Animator Controller, která zprostředkovává řízení animací daného objektu. Skrze tuto komponentu je možné implementovat inverzní kinematiku pro pohyb rukou humanoidní kostry. Inverzní kinematika docílí toho, že pohyby ovladačů jsou mapované na pohyby rukou uživatele skafandru. Dále byla přidána animace nohou, která se opakuje ve smyčce. Cílem této animace bylo zvýšení dynamičnosti modelu. V neposlední řadě pozice modelu ve scéně je napojena na pozici náhlavního displeje ve scéně. Tedy kamera uživatele pohledu je umístěna do helmy skafandru.



Obrázek 4.4. Model astronauta použitého pro reprezentaci uživatele.

4.6 Ovládání scény

Pohyb ve scéně je primárně zajištěn přímo pohybem uživatele v reálném světě, kdy jsou použita data z náhlavního displeje. Pohyb uživatele ve skutečném světě může být limitován různými faktory. Těmi mohou být velikost místnosti, ve které se uživatel nachází, či délkou kabelu, který propojuje náhlavní displej s počítačem. Z těchto důvodů je přidána do aplikace i sekundární možnost pohybu pomocí ovladačů. Vhodný způsob nastavení ovládání bude zvolen ve fázi testování.

Kapitola 5

Testování

V této kapitole zhodnotím výsledky implementace aplikace spolu s jejími limity. Stejně tak zvolím vhodné nastavení ovládání pro pohyb ve scéně. Také přidám test použitelnosti s názory uživatelů na vizualizaci.

5.1 Preference ovládání pohybu

Pohyb a změna pohledu ve scéně jsou primárně vázány na pohyby uživatele v reálném světě. Je zde ale i sekundární ovládání pomocí ovladačů, které je neméně důležité, proto je nutné zvolit intuitivní nastavení. Za tímto účelem byli zvoleni dva dobrovolníci s minimálními zkušenostmi ve virtuální realitě, kteří vyzkouší několik možností způsobu pohybu ve scéně.

volba	možnost A	možnost B
způsob otáčení kamery	plynulý	skokový
rychlost otáčení	$60 \frac{\circ}{s}$	$30 \frac{\circ}{s}$
rozložení ovládání pohybu a otáčení	symetrické	asymetrické

Tabulka 5.1. Testované možnosti ovládání pohybu.

Dobrovolníkům byly postupně prezentovány 3 volby nastavení parametrů ovládání, pro které bylo vždy na výběr ze dvou možností. Tyto možnosti byly zvoleny na základě osobního testování ovladatelnosti.

■ Způsob otáčení kamery

Dobrovolníky byl zvolen plynulý způsob otáčení. Hlavním důvodem byla ztráta imerze u skokového otáčení. Plynulé otáčení se jevilo jako přirozenější a podobnější skutečnému pohybu.

■ Rychlost otáčení

Pro určení parametru rychlosti otáčení byla opakována první volba. Jak v případě plynulého i skokového otáčení byla zvolena nižší hodnota, tedy $30 \frac{\circ}{s}$, respektive skok o 30° . Důvodem byl opět pocit vyšší imerze.

■ Rozložení ovládání pohybu a otáčení

Poslední volbou bylo rozložení ovládání. První možností bylo symetrické ovládání, kdy jsou na obou ovladačích stejné možnosti ovládání. Joystick ovládá otáčení a dotykový sensor pohyb. Druhou možností bylo asymetrické ovládání, kdy se ovládá joystickem pravého ovladače pohyb a joystickem levého ovladače otáčení.

V této volbě bylo zvoleno asymetrické ovládání. Jako hlavní důvod byla uvedena neintuitivita dotykového sensoru.

Výstupem testu je volba plynulého otáčení kamery při rychlosti $30 \frac{\circ}{s}$, kdy jsou ovládací prvky rozloženy dle asymetrického ovládání.

5.2 Přehled výsledného vzhledu aplikace

Při spuštění aplikace se uživatel dostane do hlavního menu, kde dozví možnosti ovládání spolu se základními informacemi o aplikaci. Po zvolení tlačítka **PLAY** je přenesen do hlavní scény.



Obrázek 5.1. Záběr z hlavního menu.

Po přesunu do hlavní scény je pohled uživatele namířen přímo k černé díře. Jako výchozí je zvolena vizualizace pomocí vygenerovaných dat.



Obrázek 5.2. Pohled uživatele po přesunu do hlavní scény.

Stalčením tlačítka pro menu na ovladači se otevře pause menu, kde může uživatel změnit způsob vizualizace černé díry na aproximační metodu či ji úplně vypnout. Skrze toto menu je také možné se vrátit zpět do úvodní scény.



Obrázek 5.3. Záběr z pause menu v hlavní scéně. Uživatel vybírá možnost vizualizace pomocí aproximační metody.

Pokud uživatel zvolí v menu aproximační metodu, tak je nastavena jako způsob vizualizace černé díry.



Obrázek 5.4. Pohled uživatele po zvolení aproximační metody pro vizualizaci černé díry.



Obrázek 5.5. Pohled uživatele zády k černé díře na kosmickou loď.

5.3 Test použitelnosti

Cílem této aplikace je její použití za účelem vzdělávání a prezentace vlastností černých děr širší veřejnosti. Z tohoto důvodu je důležité zjistit všechny nedostatky současné verze, aby bylo možné je opravit před představením programu veřejnosti. Zde popíši postup zvolený pro testování vlastností a použitelnosti aplikace.

Aplikace byla zkoušena pouze s jedním uživatelem v hlavní scéně. Testování bylo uskutečněno v mé soukromé pracovně, kdy jsem byl v místnosti já a testovací uživatel. Celkem se zúčastnili 4 dobrovolníci ve věku od 20 do 25 let. Účastníci měli minimální či nulové zkušenosti s virtuální realitou. Znalosti o černých dírách se různily, avšak všichni účastníci měli alespoň základní povědomí o jejich vlastnostech.



Obrázek 5.6. Testovací aranžmá.

5.3.1 Testovací postup

Na začátku testu byl dobrovolník dotázán na následující otázky:

- Jaký je váš věk?
- Mohl(a) byste popsat, co je to černá díra?
- Máte nějaké zkušenosti s použitím VR technologie?

Následně byl předveden testovací VR headset s ovladači a limity pohybu v něm. Dále byla také účastníkovi stručně popsána aplikace, kterou bude testovat.

Po úvodu začala testovací sekvence, kdy měl dobrovolník postupně splnit tyto body:

- Nasadte si náhlavní displej.
- Přejděte do hlavní scéně.
- Přiblížte se k černé díře pomocí ovladačů.
- Otočte se pomocí ovladačů.
- Změňte způsob vizualizace.
- Opusťte hlavní scéně.

Po dokončení zadaných úkolů byl účastník tázán na následující otázky:

- Jak by jste hodnotil(a) vizualizaci černé díry?

- Přišlo vám ovládání intuitivní?
- Dělal se vám nevolno při používání aplikace?
- Co byste změnil(a)?

5.3.2 Výstup testování

Opakujícím se problémem u testování bylo otvírání pause menu v hlavní scéně. Dobrovolníci měli problém s nalezením menu tlačítka, které je poměrně malé. Řešením tohoto problému by mohlo být přesunutí pause menu pod tlačítko grip, které je snadno určitelné.

Druhým problémem byla absence zábran ve scéně. To zapříčinilo, že se účastníci pohybovali někdy i daleko od pozice černé díry, kde již vizualizace nefunguje korektně. Problém je možné vyřešit přidáním zábran, které budou limitovat pohyb uživatele ve scéně.

Ostatní ovládací prvky se zdály být intuitivní. Žádnému dobrovolníkovi se v aplikaci neudělalo špatně. Vizualizace černé díry byla hodnocena pozitivně. Jeden z dobrovolníků navrhl přidání více předmětů do scény, za účelem lepší orientace v prostoru.

Před použitím pro širší veřejnost by měl být proveden test zaměřený pouze na kinezózu. Způsob a postup provedení takového testu je popsán v knize Jasona Jeralda[3].

5.4 Limity aplikace

V této sekci shrnu výpočetní náročnost aplikace, spolu s nedostatky vizualizace černé díry a limitujícími faktory.

5.4.1 Výkon aplikace

Průběžným testováním aplikace bylo zjištěno, že nejvyšší výpočetní zátěž způsobuje získávání sférického obrazu z pohledu uživatele. Důvodem zátěže je fakt, že pro získání dat je nutné vykreslit 6 různých pohledů kamery a vykreslené textury dále zpracovat do jediné textury v ekvidistantní válcové projekci. Tato data jsou nutná pro běh vizualizace černé díry.

Výkon aplikace byl testován na zařízení s následujícími parametry:

- CPU: AMD™ Ryzen 5 3600
- RAM: 2x Crucial Ballistix 8GB DDR4 3200 MHz
- GPU: NVIDIA GeForce GTX 1080
- HMD: Acer Windows Mixed Reality Headset AH101

Jelikož vykreslování sférického pohledu je nejnáročnějším procesem v aplikaci, zvolil jsem jej jako parametr výkonu.

sférický pohled	FPS
vypnutý	90
obnova jednou za 6 snímků	64
obnova při každém snímku	23,5

Tabulka 5.2. Výkon aplikace v závislosti na sférickém pohledu. Maximální hodnota FPS je 90. Tento limit je způsobený testovacím náhlavním displejem.

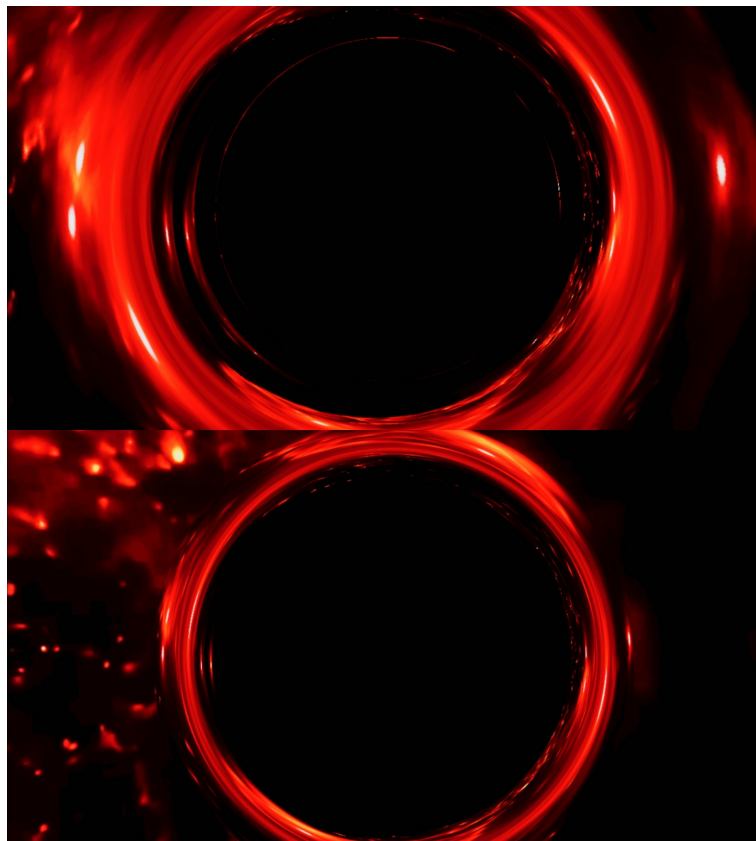
Obnova sférického pohledu při každém snímku není přípustná, jelikož pro kvalitní zážitek ve VR je třeba alespoň 60 FPS. Při vypnutém sférickém pohledu chybí ve

vizualizaci část potřebných dat a velmi se tak snižuje imerze. Při obnově pohledu jednou za 6 snímků, tj. při každém snímku je obnovena jedna část sférického pohledu, vykazuje aplikace dostatečný výkon a nižší obnovovací frekvence sférického pohledu není příliš znatelná.

Pro použití obnovy při každém snímku by bylo třeba výrazně výkonější výpočetní zařízení či optimalizace získávání sférického pohledu.

■ 5.4.2 Srovnání vizualizací

Oba přístupy, pro vykreslování černé díry, mají své nedostatky. Tyto limity by bylo možné minimalizovat spojením postupů do jediného, který by používal oba přístupy. To však v této práci nebylo realizováno. Pojďme tedy určit, jaké jsou nedostatky a vlastnosti použitých postupů.



Obrázek 5.7. Nahoře vizualizace pomocí vygenerovaných dat. Dole vizualizace pomocí aproximační funkce.

U vygenerovaných dat je možné deformovat veškerý obraz, který svírá s černou dírou a uživatelem úhel menší než 90 stupňů. Deformace se mění v závislosti na vzdálenosti uživatele od objektu, kdy maximální vzdálenost je 25 jednotek a minimální 0.

Hlavním limitem současné verze vykreslování je absence změny deformace obrazu v závislosti na vzdálenosti pozorovaného objektu od černé díry. Data potřebná pro tuto funkcionalitu jsou vygenerována, avšak způsob, kterým je vykreslován objekt černé díry, neumožňuje určit, zda se ve předem vypočtené trase fotonu nachází nějaký objekt či nikoliv. Z tohoto důvodu je pro veškerý deformovaný obraz použita maximální vzdálenost od černé díry.

Druhým nedostatkem vizualizace je nutnost interpolace mezi daty. Pokud se přechází z interpolace dat pro dvě vzdálenosti na data pro jiné dvě vzdálenosti, může dojít k malému skoku, který narušuje imerzi uživatele. Tento problém je však pouze okrajový a bylo by možné jej vyřešit větším množstvím dat.

U aproximační vizualizace je hlavním nedostatkem vysoká nepřesnost odhadu deformace, pokud je vzdálenost fotonu v periastru, tedy parametr r_0 , nižší než 4 jednotky. Z tohoto také plyne limit, pro jaké vzdálenosti uživatele od černé díry je možné efekt použít. Není zde žádný horní limit, ale pouze spodní, kterým jsou již zmíněné 4 jednotky. Tento fakt je poměrně limitující, jelikož se uživatel nemůže plně přiblížit k záhadnému tělesu, jímž černá díra je. Vizualizace pomocí aproximační funkce je oproti předchozímu způsobu spojitá. Deformace se tudíž mění plynule při pohybu uživatele.

Podobně jako u vygenerovaných dat, není možná deformace obrazu v závislosti na vzdálenosti pozorovaného objektu od černé díry.

Kapitola 6

Závěr

Cílem bakalářské práce bylo vytvořit aplikaci ve virtuální realitě, ve které budou moci dva uživatelé pozorovat vizualizaci černé díry a zkoumat její efekt gravitační čočky na deformaci obrazu. Druhým úkolem bylo vytvořit systém pro vykreslování sférického videa do pozadí scény. Vizualizace černé díry byla uskutečněna dvěma způsoby. V první řadě je to postup založený na základě předem vygenerovaných dat, která vzešla úpravou výchozího programu, poskytnutého Michalem Bursou, na výpočet drah fotonů v okolí černé díry. Druhým přístupem je pak aproximační funkce, jenž se blíže přibližuje skutečným vlastnostem černé díry.

Použití herního enginu Unity zajišťuje stabilní a plynulý běh aplikace s podporou virtuální reality pro různé typy náhlavních displejů. Zajištěn byl i systém, který podporuje připojení dvou, případně i více uživatelů do jediné scény, kteří tak mohou společně pozorovat efekty černé díry. V neposlední řadě bylo realizováno i vykreslování sférického videa do pozadí virtuálního prostředí. Tímto způsobem byly realizovány všechny body popsané v 3.1, avšak práce má i své limity 5.4, které by mohly být do budoucna odstraněny. I přesto jsem s výsledky své práce spokojen.

6.1 Další vývoj

Hlavním bodem pro další vývoj aplikace je doplnění vykreslování deformace obrazu i v závislosti na vzdálenosti pozorovaného objektu od černé díry. Tato funkcionality by mohla být implementována pomocí vlastní implementace jednoduchého ray-tracing shaderu.

V druhé řadě je to pak spojení vizualizace pomocí vygenerovaných dat a aproximační metody, za účelem minimalizace jejich negativních vlastností.

Dále je to také napojení pohledu ze sférické kamery do textury, která je vykreslována do pozadí scény. Za tímto účelem bude muset být vytvořen datový tok, který propojí data z kamery s aplikací.

Nakonec je to také další rozšíření generování dat. Kdy bude určena dynamicky maximální velikost úhlu, pod kterým bude paprsek ještě ovlivněn černou dírou. Pro data generovaná z velké vzdálenosti bude tento úhel malý. Naopak pro data v blízkosti černé díry bude úhel větší než 90° , což znamená, že pohled uživatele bude deformován i při pohledu od černé díry.

Doufám, že v budoucnu budu mít možnost tyto body uskutečnit a posunout tak tento projekt dál.



Literatura

- [1] Michal Bursa. *SIM5: Library for ray-tracing and radiation transport in general relativity*. 2018.
- [2] Pavel Příhoda, Petr Kulhánek, David Břeň a Ivan Havlíč. *Vesmír*. Dorling Kindersley Limited, 2005.
- [3] Jason Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. New York, NY, USA: Association for Computing Machinery and Morgan Claypool, 2015.
- [4] Dieter Schmalstieg a Tobias Hollerer. *Augmented Reality: Principles and Practice (Usability)*. Redwood City, CA, USA: Addison Wesley, 2016.
- [5] Ivan Platonov. *Black Holes Interactive Visualization*. 2016.
<https://dspace.cvut.cz/handle/10467/64702>.
- [6] Annemieke Verbraeck a Elmar Eisemann. Interactive Black-Hole Visualization. *IEEE Transactions on Visualization and Computer Graphics*. 2021, 27 (2), 796-805. DOI 10.1109/TVCG.2020.3030452.
- [7] O. Semerák. APPROXIMATING LIGHT RAYS IN THE SCHWARZSCHILD FIELD. *The Astrophysical Journal*. 2015, 800 (1), 77. DOI 10.1088/0004-637x/800/1/77.
- [8] Otto Seiskari. *Ray-traced simulation of a black hole*. 2015.
<https://github.com/oseiskar/black-hole>. original-date: 2015-12-06T16:46:22Z.
- [9] Riccardo Antonelli. *How to draw a Black Hole*.
<http://rantonels.github.io/starless/>.
- [10] Jordy Davelaar. *Black holes in virtual reality*.
<https://jordydavelaar.com/?p=51>.
- [11] MathWorks. *Inverse Kinematics*.
<https://www.mathworks.com/discovery/inverse-kinematics.html>.

Příloha A

Struktura přiložených souborů

- Linux projekt pro generování dat
 - Postup pro spuštění naleznete v příloze B - Uživatelský manuál.
 - *bh-data-gen.zip*
- VR projekt v herním enginu Unity 3D
 - Níže vypsání soubory tvoří projekt v Unity verze 2020.3.0f1. Postup pro spuštění naleznete v příloze B - Uživatelský manuál.
 - *UnityProject.zip*
 - *Video.zip*
- Ilustrační materiály
 - *showcase-video.mp4*
 - *images.zip*
- Textový soubor se základními informacemi o projektu.
 - *README.txt*

Příloha B

Uživatelský manuál

B.1 Spuštění programu pro generování dat

- Požadavky
 - Zařízení s operačním systémem Linux
 - Knihovna SDL2, instalace → `sudo apt-get install libsdl2-dev`
 - Knihovna SDL2_image, instalace → `sudo apt-get install libsdl-image1.2-dev`
- Spuštění
 - Stáhněte soubor `bh-data-gen.zip` a rozbalte ho
 - Otevřete rozbalený adresář v konzoli
 - Pro vytvoření spustitelného programu použijte → `make`
 - Následně pro generování nových dat použijte → `./bh-data-gen profile-name data-radius-in-pixels save`
 - Pro načtení existujících dat použijte → `./bh-data-gen name-of-file-with-generated-data`
- Ovládání jednoduché vizualizace
 - Šipka vlevo → nižší FOV
 - Šipka vpravo → vyšší FOV
 - Šipka nahoru → vyšší R0
 - Šipka dolů → nižší R0
 - Page up → vyšší R_MAX
 - Page down → nižší R_MAX

B.2 Spuštění projektu v Unity 3D

- Požadavky pro spuštění bez úprav v projektu
 - Zařízení podporující Windows Mixed Reality
 - Windows Mixed Reality headset s ovladači
 - Unity Hub s instalací Unity verze 2020.3.0f1
- Spuštění
 - Stáhněte soubor `UnityProject.zip` a rozbalte ho
 - V rozbaleném adresáři otevřete složku `Assets`
 - V této složce rozbalte soubor → `Video.zip`
 - Nyní přidejte projekt do Unity Hubu a spusťte jej. Bude to několik minut trvat.

- Po načtení projektu otevřete složku *Scenes* a dvojitým kliknutím otevřete scénu *MenuScene*.
 - Pokud budete dotázáni na stažení souborů balíčku Text Mesh Pro, stáhněte pouze nutné soubory.
 - Tlačítkem PLAY v horní části spustíte aplikaci.
- Ovládání
- Joystickem levého ovladače se otáčí kamera
 - Joystickem pravého ovladače se posouváte
 - Tlačítkem Grip na pravém ovladači otevřete menu
 - Tlačítko Select na obou ovladačích slouží k výběru položek v menu

Příloha C

Odkazy

C.1 Gitlab repozitář projektu

- https://gitlab.fel.cvut.cz/pelesjak/blackholevr_bp

C.2 Ukázkové video

- <https://www.youtube.com/watch?v=cVHN3FFI-xc>

C.3 3D Modely

- <https://free3d.com/3d-model/intergalactic-spaceship-in-blender-28-eevee-394046.html>

C.4 Unity packages

- Photon Unity Networking - <https://doc.photonengine.com/en-us/pun/current/getting-started/pun-intro/>
- Starfield Skybox - <https://assetstore.unity.com/packages/2d/textures-materials/sky/starfield-skybox-92717>