

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Vzdělávací hry ve virtuální realitě s vesmírnou tematikou

Bc. Ondřej Okluský

Program: Otevřená informatika
Obor: Počítačová grafika a interakce

Květen 2017

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

Poděkování / Prohlášení

Tímto bych rád poděkoval Ing. Davidovi Sedláčkovi Ph.D. za poskytnutí tohoto skvělého zadání, trpělivé vedení práce a odborné konzultace. Svě rodině, přítelkyni a přátelům za podporu v průběhu studia. Elišce Kuběnové za konzultace ohledně jazyka českého a korekturu textu. A všem, kdo si tuto hru v průběhu jejího vzniku zahráli a poskytlí mi otevřenou a hodnotnou kritiku.

I hereby declare that I have written the submitted thesis myself and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic theses.

.....
Ondřej Okluský

In Prague, May 25, 2017

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.

.....
Ondřej Okluský

V Praze, 25. května, 2017

Abstrakt / Abstract

V této práci jsem se zaměřil na tvorbu vzdělávací hry ve virtuální realitě. Navrhl jsem abstraktní interakční vrstvu, která umožňuje snadný vývoj napříč různými zařízeními pro virtuální realitu. Implementoval jsem různé navigační a interakční techniky. Výsledkem je vzdělávací hra s vesmírnou tematikou, řešení interakce, navigace a uživatelského rozhraní.

Klíčová slova: Virtuální realita; Vzdělávací hry; Vývoj her pro více platform.

In this thesis, I have focused on creating of an educational game in virtual reality. I've designed abstract interaction layer, which enables easy development on various devices for virtual reality. I've implemented some navigation and interaction techniques. The result of this thesis is educational game with space topic in virtual reality, solution of interaction, navigation and user interface.

Keywords: Virtual reality; Serious games; Multiplatform game development.

Title translation: Educational games with Space topic in virtual reality

/ Obsah

1 Úvod	1
2 Analýza problému	2
2.1 Rozbor zadání	2
2.1.1 Scéna 1 - Stavba rakety	2
2.1.2 Scéna 2 - Start rakety	2
2.1.3 Scéna 3 - Barycentrické míčky	2
2.1.4 Scéna 4 - Sběr vzorků měsíčních hornin	3
2.1.5 Scéna 5 - Vyšetřování vlastností odebraných hornin	3
2.2 Rešerše	3
2.2.1 Virtuální realita	3
2.2.2 Vzdělávání a virtuální realita	4
2.2.3 Kinetóza	5
2.2.4 Pohyb ve virtuální realitě	5
2.2.5 Aktivní stereoskopické zobrazení	6
2.2.6 Herní engine podporující virtuální realitu	8
2.2.7 Uživatelské rozhraní	10
2.2.8 Atlas textur	12
3 Návrh řešení	14
3.1 Multiplatformní systém interakce	14
3.2 Ovládání	16
3.2.1 PC	16
3.2.2 HTC Vive	17
3.2.3 Google Cardboard v2	20
3.3 Návrh scén	21
3.3.1 Výběr scén	22
3.3.2 Scéna 1 – Stavba rakety	22
3.3.3 Scéna 2 – Start rakety	24
3.3.4 Scéna 3 – Barycentrické míčky	25
3.3.5 Scéna 4 – Sběr vzorků měsíčních hornin	26
3.3.6 Scéna 5 – Vyšetřování vlastností odebraných hornin	27
3.3.7 Lokalizace	28
4 Implementace	29
4.1 Pohyb ve scéně pomocí HTC Vive ovladačů	29
5 Testování	33
5.1 Informace o zařízeních	33
5.2 Informace o testování	34
5.3 Výsledky získané z testování	34
5.3.1 Testování na Google Cardboard v2	36
6 Závěr	37
Literatura	39
A Dotazník	42
B Obrázky ze hry	51
C Zadání	57
D Obsah příloženého datového nosiče	61

Tabulky / Obrázky

5.1. Parametry počítače v laboratoři virtuální reality.....	33
5.2. Parametry telefonu Samsung Galaxy S6 Edge	33
2.1. Počet uživatelů virtuální reality	4
2.2. Top and bottom	7
2.3. Side by side.....	7
2.4. Vizuální srovnání Unreal engine 4 vs Unity 5	9
2.6. Rozdělení uživatelského rozhraní	11
2.7. Nediegetické uživatelské rozhraní	11
2.8. Diegetické uživatelské rozhraní	11
2.9. Meta uživatelské rozhraní.....	12
2.10. Prostorové uživatelské rozhraní	12
2.11. Atlas textur	12
3.1. Multiplatformní systém interakce	14
3.2. Diagram tříd pro multiplatformní systém interakce.....	15
3.3. Ovládání pro PC	17
3.4. Ovládání HTC Vive.....	18
3.5. Uživatelské rozhraní ovladačů HTC Vive	19
3.6. Ovládání Google Cardboard v2	20
3.7. Uživatelské rozhraní pro výběr nástroje v Google Cardboard v2	21
3.8. Menu.....	22
3.9. Uživatelské rozhraní pro část rakety	22
3.10. Pozicování uživatelského rozhraní pro část rakety.....	22
3.11. Systém pro stavbu rakety.....	23
3.12. Navigace ve scéně pomocí pojízdné plošiny	24
3.13. Přehrávač animace	25
3.14. Tvorba fyzikální soustavy.....	25
3.15. Dálkové ovládání vozítka	26
3.16. Ponos dálkového ovladače	27
3.17. Obrázky výbrusů měsíčních hornin.....	28
4.1. Imitace běhu pomocí ovladačů HTC Vive	29

4.2.	Kolizní objekt pro virtuální herní zóny	30
5.1.	Výsledky hodnocení scén podle VR zařízení	35
5.2.	Úprava uživatelského rozhraní .	36

Kapitola 1

Úvod

V této práci se budu zabývat tvorbou vzdělávací hry ve virtuální realitě Ze Země na Měsíc, která by měla sloužit jako doplněk projektového výukového dne na středních školách. V průběhu celodenního programu se studenti základních škol/středních škol seznámí s potřebnými teoretickými znalostmi spjatými s cestou Evropské kosmické agentury na Měsíc. Projektový den je rozdělen do postupně navazujících částí, kdy každá část je zakončena prožitkem ve virtuální realitě, kde si děti vyzkouší právě získané znalosti v „praxi“. Celý projekt vzniká ve spolupráci s ESERO [1] – Evropskou kanceláří pro vzdělávání o vesmíru. Posláním projektu ESERO je využití témat vesmíru ke zvýšení motivace žáků pro studium technických a přírodovědných předmětů.

Není tedy divu, že ESERO má zájem o vzdělávací hry ve virtuální realitě. V poslední době začíná být virtuální realita čím dál tím lákavější pro běžné uživatele a objevuje se stále více zařízení pro virtuální realitu, jmenovitě například PC, Google Cardboard, Gear VR, HTC Vive, Oculus Rift, Playstation VR, a souběžně s touto prací vznikají další, například Google Daydream a Microsoft VR. V této práci se budu zajímat o unifikaci vývoje speciálně pro zařízení Google Cardboard v2, HTC Vive a PC tak, aby bylo možné ušetřit čas a práci při vývoji her na tyto zařízení.

Google Cardboard v2, Gear VR, Google Daydream a další headsety, které pro virtuální realitu používají mobilní zařízení, jsou vhodné pro použití na středních školách právě proto, že používají mobilní zařízení. Vzhledem k omezeným rozpočtům základních a středních škol nelze spoléhat na to, že si každá škola bude moci dovolit pořídit zařízení HTC Vive nebo Oculus Rift, které slouží výhradně pro účely virtuální reality, v takovém množství, aby se mohli studenti plynule střídat. Naopak mobilní zařízení jsou výhodná především proto, že mohou naleznout i jiná využití. Pro vývoj hry použiji vhodný volně dostupný herní engine, který podporuje virtuální realitu a umožňuje export pro výše uvedené platformy.

Pro maximální usnadnění práce bude užitečné navrhnout jednotný systém manipulace a interakce ve virtuálním prostředí, který se postará o sjednocení napříč různými zařízeními. Dále je třeba prostudovat, jaké možnosti a omezení přináší vývoj her ve virtuální realitě. Je jasné, že některé současné techniky ovládání a i interakce používané v PC hrách nebude možné použít, protože vstupní zařízení se výrazně liší. Další problém, který je potřeba vyřešit, je navigace ve scéně, která může být omezená vzhledem k počtu vstupů a možnostem vstupního zařízení.

Na závěr provedu testování s cílovou skupinou, tedy s uživateli ve věkovém rozmezí 8 až 20 let. Ze získaných výsledků provedu vyhodnocení nalezených chyb a shrnu hodnocení uživatelů.

Kapitola 2

Analýza problému

2.1 Rozbor zadání

V této části práce zmíním zadání a tematiku jednotlivých her. U každé scény nastíním jaké problémy budu řešit, případně provedu nástin řešení, který mi pak pomůže při návrhu. Hra Ze Země na Měsíc bude obsahovat celkem pět scén. Na těchto scénách vyzkouším různé navigační a interakční techniky vhodné pro jednotlivá zařízení, provedu testování a diskusi o jejich vhodnosti.

2.1.1 Scéna 1 - Stavba rakety

První scéna se odehrává v hangáru pro stavbu vesmírných raket v Guyanském kosmickém centru ve Francouzské Guyaně. V hangáru jsou umístěny celkem tři různé typy raket, Ariane 5R, Ariane 5C a Ariane 5P, které jsou rozděleny na několik kusů. Cílem této hry je vyzkoušet si stavbu rakety typu Ariane 5 a informovat uživatele o tažné síle, hmotnostech a množství paliva, které raketa potřebuje pro dosažení kosmického prostoru.

Tato scéna vyžaduje manipulaci s částmi rakety a snadno pochopitelný systém pro spojování těchto částí. Dále je potřeba vyřešit pohyb po velké scéně, který navíc musí být realizován ve všech třech osách.

2.1.2 Scéna 2 - Start rakety

Druhá scéna se opět odehrává v Guyanském kosmickém centru ve Francouzské Guyaně. V této scéně se uživatel seznámí s procesem startu rakety typu Ariane 5P, který začíná přivezením rakety na odpalovací rampu. Následně jsou zažehnuty raketové motory první fáze, které jsou odhozeny po dosažení vesmírného prostoru, a nakonec dojde k zážehu motorů druhé fáze. Celý proces startu rakety bude možné libovolně posouvat, přehrávat a opakovat.

Stěžejní pro tuto scénu je vytvoření multimediálního přehrávače, který umožní uživateli různými způsoby ovlivňovat tok času ve scéně.

2.1.3 Scéna 3 - Barycentrické míčky

Třetí scéna se odehrává v modulu ISS - mezinárodní vesmírné stanice. Zde si uživatel bude moci poskládat soustavu objektů z míčů a tyček [2]. Na této soustavě bude simulováno chování soustavy těles v nulové gravitaci vzhledem k jejich barycentru.

V této scéně je třeba vytvořit systém pro spojování míčů a tyček tak, aby na nich korektně fungovala fyzikální simulace. Dále je potřeba vyřešit navigaci ve scéně, aby připomínala pohyb ve stavu beztíže.

■ 2.1.4 Scéna 4 - Sběr vzorků měsíčních hornin

Čtvrtá scéna se odehrává na povrchu Měsíce, kde plánuje Evropská kosmická agentura postavit vesmírnou základnu [19]. Cílem v této scéně je sběr vzorků měsíčních hornin, které následně budou využity k rozličným geologickým testům. Odběr vzorků hornin bude realizován pomocí dálkově ovládaného měsíčního vozítka.

Jelikož má být měsíční vozítko dálkově ovládané, v této scéně je třeba vytvořit přehledný dálkový ovladač. Tím bude moci uživatel navigovat vozítko k zónám vhodným pro sběr vzorků. Také bude moci sledovat, co vozítko právě dělá, pomocí displeje na dálkovém ovladači.

■ 2.1.5 Scéna 5 - Vyšetřování vlastností odebraných hornin

Pátá scéna se odehrává uvnitř měsíční základny. Na odebraných vzorcích hornin se budou provádět testy na půdní zrnitost a obsah vápníku. Test na půdní zrnitost se provádí pomocí soustavy sít s různou velikostí ok, v tomto případě dvou. Odebraný vzorek měsíční horniny se zvaží. Poté se síty rozdělí na 3 různě hrubé frakce – hrubou, středně hrubou a jemnou. Poměr váhy proseté frakce a celého vzorku určuje hmotnostní podíl této frakce. Test na přítomnost vápníků se provádí pomocí roztoku octa. Pokud vzorek horniny obsahuje vápník, pak v kombinaci s octem dojde k reakci, která se projevuje šuměním. Pro nejhrubší prosetou frakci je možné si zobrazit výbrus horniny pod mikroskopem, úkolem bude najít vzorek s vhodnou mikroskopickou strukturou.

V této scéně budou nádoby se získanými vzorky, ze kterých bude možné vzorek odebrat a podrobit ho testování. Každá nádoba pevně určuje vlastnosti vzorku, tedy hmotnostní podíly jednotlivých frakcí a to zda vzorek obsahuje vápník. Pomocí sít se ze vzorku budou oddělovat různé frakce, celkově tak vznikne 5 možností: hrubá frakce, střední frakce, jemná frakce, mix hrubé a střední frakce a mix střední a jemné frakce. Dále bude potřeba vytvořit váhu, na které bude možné vzorky a proseté frakce zvážit, tak aby mohl být určen hmotnostní podíl. Kromě nádob se vzorky zde bude umístěna i nádoba s octem, do které když se vzorek ponoří dá najevo vizuálním efektem, zda vzorek obsahuje vápník, či nikoliv. Poslední součástí této scény bude mikroskop, který pro nejhrubší frakci zobrazí výbrus horniny.

■ 2.2 Rešerše

V této části práce proberu některé základní informace týkající se problematiky. Provedu výběr vhodného herního enginu, ve kterém budu hru realizovat, a upozorním na některé aspekty vývoje her ve virtuální realitě.

■ 2.2.1 Virtuální realita

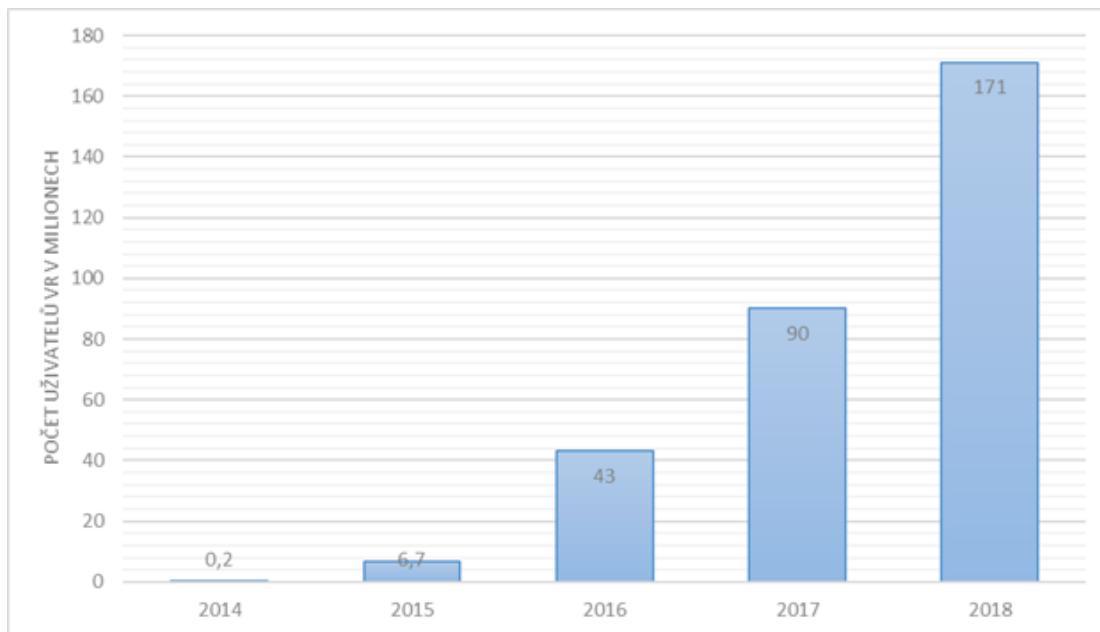
Cílem systému pro virtuální realitu[25] je poskytnout uživateli iluzi, že se nachází ve virtuálním prostředí. Dosažení této iluze se dociluje ovlivňováním lidských smyslů, především zraku, sluchu, hmatu, nebo také v některých simulátorech i vychýlením z rovnováhy. Svoji roli zde hraje i představivost a ochota uživatele přijmout iluzi virtuálního světa. Tato ochota nemusí být vždy nutně vědomá, protože smysly uživatelů se někdy podvědomě brání uvěřit prezentované iluzi. Hlavními znaky aplikací pro virtuální realitu jsou zobrazování a interakce v reálném čase, tedy obraz musí být uživateli vykreslován rychlostí nejméně 25 snímků za vteřinu. Jsou úzce spjaté s virtuálním prostředím, ve kterém je uživatel reprezentován avatarem. Avatar je zástupný objekt ovládaný uživatelem, který mu umožňuje se pohybovat ve virtuálním prostředí, rozhlížet se, interagovat a kolidovat s virtuálním prostředím. Často bývá používán zvuk ke zprostředkování

informace o objektech nebo prostředí, jako je například ozvěna ve velkém prázdném prostoru, nebo také na dokreslení atmosféry na pozadí.

Virtuální realitu můžeme dále dělit na pohlcující virtuální realitu a rozšířenou realitu.

Pohlcující virtuální realita je vždy úzce propojena se speciálními hardwarovými zařízeními, které mají uživatele co nejvíce oprostít od reálného světa a přenést ho, co možná nejvíce, do virtuálního prostředí. Mezi typická zařízení patří helma se stereoskopickými brýlemi, sluchátka a ovladače. Rozšířená realita je kombinace vjemů skutečného světa, obohacená o virtuální informace. Typicky jsou pohyb a rotace snímající kamery synchronizovány s uživatelem, a obraz snímáný touto kamerou je pak obohacen o další informace. Vhodným příkladem je tank, který snímá okolní prostor pouze kamerou, a do obrazu následně zvýrazňuje spojenecké a nepřátelské cíle.

V současné době zažívá virtuální realita obrovský rozmach a výrazně roste zájem uživatelů o používání virtuální reality viz 2.1. Díky snižující se ceně potřebných technologií pro výrobu zařízení pro virtuální realitu se stává dostupnou i běžným uživatelům. Jednodušší aplikace je tak možné provozovat v dobré kvalitě i na mobilních telefonech. Uplatnění virtuální reality je velmi široké a je možné ji využít v mnoha oblastech lidské činnosti. Ve sportu se využívá k dokonalému vytrénování pohybů. V medicíně je známo několik úspěšných léčebných procesů odstraňující fobie. Ve strojírenství a stavitelství je možné použít virtuální realitu například k prohlídkám ještě nepostavených budov, letadel nebo konstrukčních dílů. Ve vojenském průmyslu šetří peníze a životy při výcviku pilotů. A v neposlední řadě se stává čím dál větší součástí zábavního průmyslu.



Obrázek 2.1. Odhad počtu uživatelů VR [3] pro roky 2014-2018.

2.2.2 Vzdělávání a virtuální realita

Vzdělávací hry byly již dříve prokázány jako vhodná studijní pomůcka [22]. Mohou být využity ke vzdělávání dětí již od předškolního věku, během základního a středoškolského vzdělání, a dokonce i v dospělosti. Od jednoduchých her, procvičujících základy jazyka, matematiky, logiky a přírodních věd, je možné se dostat až k náročným simulátorům používaným k výuce specializovaných profesních oborů. Všechny tyto hry rozvíjejí koordinaci rukou a očí, motorické schopnosti a kognitivní schopnosti jako například paměť

a soustředění. Vzhledem k tomu, že dnešní děti tráví velké množství času hraním počítačových her, je přirozené uvažovat, že pokud by se věnovaly hraní vzdělávacích her, mohlo by pro ně být jednodušší se učit a získávat užitečné informace zábavnou formou. Vzdělávací hry mají nesporné množství výhod [24]:

- Žádná nebo jen malá rizika
- Experimenty jsou jednoduše opakovatelné, bez dlouhé přípravy a následného opětovného nastavování experimentu
- Umožňuje náhled z jiné perspektivy
- Umožňuje experimenty vyžadující speciální fyzikální podmínky nebo drahé propriety
- Realistické scénáře i scénáře speciálně upravené k danému tématu
- Mohou být prováděny na dálku, tím šetří čas i peníze
- Rozvíjí paměť a schopnost si vzpomenout
- Zjednodušuje složité problémy, těžko řešitelné situace a může uživatele připravit na obtíže postupně
- Předávají informace inovativní, nenucenou, zábavnou a interaktivní formou

Poslední položka jmenovaná v seznamu je obzvláště důležitá. Vzdělávání je jednodušší, pokud je podáno příjemnou formou, což vede k větší míře vcítění se do problému a snaze ho pochopit. Tento efekt by mohl být ještě navíc umocněn pomocí pohlcující virtuální reality, která umožní uživateli hlubší ponoření do virtuálního světa. Současně by mohla řešit i problém s tím, že uživatelé během některých vzdělávacích her ztrácejí zájem a pozornost.

■ 2.2.3 Kinetóza

Kinetóza, také známá jako cestovní nevolnost [20], je fyzický stav, ve kterém neodpovídá vjem pohybu ve vestibulárním ústrojí s vizuálním vjemem. Hlavními příznaky kinetózy jsou závrať, vyčerpání, zvedání žaludku či pocity nevolnosti. Může být způsobena dvěma různými příčinami.

Vjem ve vestibulárním ústrojí nás informuje o pohybu, zatímco vizuální vjem zůstává statický. Tato příčina je často spojena s cestou v nějakém dopravním prostředku, jakým je například auto, vlak nebo letadlo. Druhá příčina je přesným opakem té první. Vestibulární ústrojí nás informuje o tom, že se nehýbeme, ale velká část našeho vizuálního vjemu tvrdí, že ano. Druhá příčina může být způsobena například v myčce aut, kdy se velká část vizuálního vjemu pohybuje směrem k pozorovateli.

Kinetóza může být způsobena i oběma příčinami současně, a to za podmínek, že informace z vestibulárního ústrojí a vizuální vjem spolu nekorespondují. Často také bývá spojována s virtuální realitou a při neuvážené navigaci kamery po scéně může dojít k prvním příznakům kinetózy velmi rychle.

■ 2.2.4 Pohyb ve virtuální realitě

Jak již bylo řečeno, neuváženými pohyby kamery můžeme ve virtuální realitě u uživatele snadno vyvolat kinetózu. Z vlastní zkušenosti vím, že nejvíce k nevolnosti přispívají velké změny rychlosti, rychlá rotace při nízké snímkové frekvenci, rychlé přiblížení ke stěně, stropu nebo pád. Konstantní rychlost kamery většině uživatelů nezpůsobuje takové obtíže, jak je možné vidět v některých hrách pro GoogleVR, například přímo v ukázkách pro Google Cardboard, nebo ve hře VR Cave. Zážitky se však liší uživatel od uživatele, a proto se stále vyvíjejí nové techniky, které by měly být pro uživatele komfortnější.

Pohyb po scéně je nejčastěji realizován pomocí „teleportu“. Uživatel v takovém případě specifikuje místo, kam se chce ve scéně přenést, poté je uživateli na krátkou chvíli ztmaven obraz, kamera se přenesse na specifikované místo a obraz je následně opět zesvětlen. Tento způsob pohybu je velmi vhodný, protože vizuální systém uživatele není vystaven náhlým skokovým změnám.

V současné době se staly snadno dostupnými systémy HTC Vive, Oculus Rift a další, které snímají pohyb uživatele ve zkalibrovaném prostředí pomocí kamer a tento pohyb přenášejí do virtuálního prostředí. Pohyb ve virtuální realitě je tedy směsicí pohybu virtuálního a fyzického. Na krátké vzdálenosti je tedy možné se pohybovat touto technikou, která vzhledem k tomu, že vychází z fyzického pohybu nevyvolává kinetózu. V kombinaci s „teleportem“ je tedy možné se efektivně pohybovat po celé scéně bez vyvolání kinetózy.

Současné vznikající práce ukazují, že s trochou důvtipu je možné využít hardware pro virtuální realitu k získání plně virtuálního pohybu uživatele. S. Machkovech[21] navrhl zajímavou techniku, která využívá ovladače HTC Vive zasunutého za pásek u kalhot k získání orientace pánve uživatele. Pak z imitace běhu na místě, je možné za pomoci zmíněného ovladače získat akceleraci, kterou lze převést na pohyb ve scéně. Výhodou tohoto přístupu je, že během pohybu nemusíme používat ovladače v ruce k vyjádření pohybu. Pokud by bylo možné použít třetí ovladač, pak nepřijdeme o možnost mít v každé ruce jeden ovladač. Další techniku navrhl D. Dewhirst[18], který k vyjádření pohybu po scéně využívá obou ovladačů pro HTC Vive. Z ovladačů nebo z kamery je možné získat směrový vektor, kterým se má uživatel pohybovat. Pomocí imitace chůze, v tomto případě pohybem rukou, jsme opět schopni z ovladačů získat akceleraci, která je převedena na pohyb podél již získaného směrového vektoru. Jelikož oba principy využívají anatomické pohyby uživatele k vyjádření pohybu ve virtuální realitě, mají velký potenciál „ošálit“ vestibulární ústrojí, a nezpůsobovat tak kinetózu.

2.2.5 Aktivní stereoskopické zobrazení

Stereoskopické zobrazení [25] je technologie, která umožňuje zprostředkování 3D obrazu pozorovateli. Zornice lidských očí jsou od sebe vzdáleny přibližně 7 cm. Každé oko dodává do mozku mírně odlišný obraz. Výsledný obraz je složen z těchto dvou různých obrazů a z nepatrných rozdílů mezi nimi je mozek schopen odvodit prostorový vjem a vzdálenost od objektů. Pokud chceme uživateli zprostředkovat 3D vjem, musíme tedy vykreslovat zvlášť obraz, který má vidět pravé, a ten, který má vidět levé oko. Navíc je třeba zajistit, aby byl uživateli distribuován do konkrétního oka vždy obraz jemu příslušící.

Existuje samozřejmě více technologií pro stereoskopické zobrazování, jako například anaglyph, pasivní stereoskopická projekce a aktivní stereoskopická projekce. Jelikož je v zadání požadováno promítání na stereostěnu v laboratoři virtuální reality na Českém vysokém učení technickém v Praze, kde se používá aktivní stereoskopická projekce, zmíním se krátce pouze o jejím principu.

Problém promítnutí správného obrazu do příslušného oka je v aktivní stereoskopické projekci vyřešen pomocí synchronizace projektoru a zatmívacích brýlí. V okamžiku, kdy projektor promítá obraz pro pravé oko, mají brýle ztmavený levý průzor, a naopak, pokud projektor právě promítá obraz pro levé oko, mají brýle ztmavený pravý průzor. To zajišťuje, že se správný obraz dostane do příslušného oka.

Tento způsob projekce vyžaduje vykreslení dvakrát více snímků než u běžné projekce, protože potřebuje vykreslit snímky pro pravé i levé oko. Také proces vykreslování tomu musí být uzpůsobený. Většinou se používá kamerový systém, který má dvě kamery s



Obrázek 2.2. Příklad formátu Top and Bottom.

horizontální vzdáleností přibližně 7 cm. Existuje více způsobů, jakými lze obrazovou informaci projektoru předat. Většina stereoskopických projektorů podporuje formáty Top and Bottom, Side by Side a Frame Sequential.

Top and Bottom viz 2.2 je formát obrazu, který pomyslně rozděluje obraz vertikálně v polovině. Snímek pro levé oko je vykreslen do horní poloviny obrazu, snímek pro pravé oko je vykreslen do spodní poloviny obrazu. Nevýhodou tohoto formátu je, že vizuální informace pro levé i pravé oko je ve vertikálním směru vyjádřena pouze polovinou pixelů, ztratíme tedy část této informace, která musí být při projekci nazpět získána interpolací ze sousedních hodnot.



Obrázek 2.3. Příklad formátu Side by Side.

Side by Side viz 2.3 je podobný formát obrazu jako Top and Bottom. Odlišuje se tím, že obraz je pomyslně rozdělen v polovině horizontálně namísto vertikálně. Snímek pro levé oko je vykreslen v levé polovině obrazu a snímek pro pravé oko v pravé polovině obrazu. Jak již tomu bylo u formátu Top and Bottom, vlivem vykreslení obou snímků do jednoho obrazu dojde opět ke ztrátě obrazové informace, tentokrát však v horizontálním směru. Chybějící vizuální informace je, stejně jako v předchozím případě, při projekci získána nazpět interpolací ze sousedních hodnot.

Jelikož formáty Side by Side a Top and Bottom vykreslují snímek pro pravé a levé oko do jednoho obrazu přenášeného na projektor, redukuje se tímto způsobem počet potřebných datových přenosů na polovinu.

Frame Sequential je formát obrazu, který na rozdíl od formátů Side by Side a Top and Bottom zachovává obraz v plné kvalitě. Obrazy pro levé a pravé oko jsou zasílány projektoru střídavě za sebou. Jelikož jsou tyto obrazy posílány odděleně a v plné kvalitě, není již potřeba před projekcí obraz nijak upravovat.

■ 2.2.6 Herní engine podporující virtuální realitu

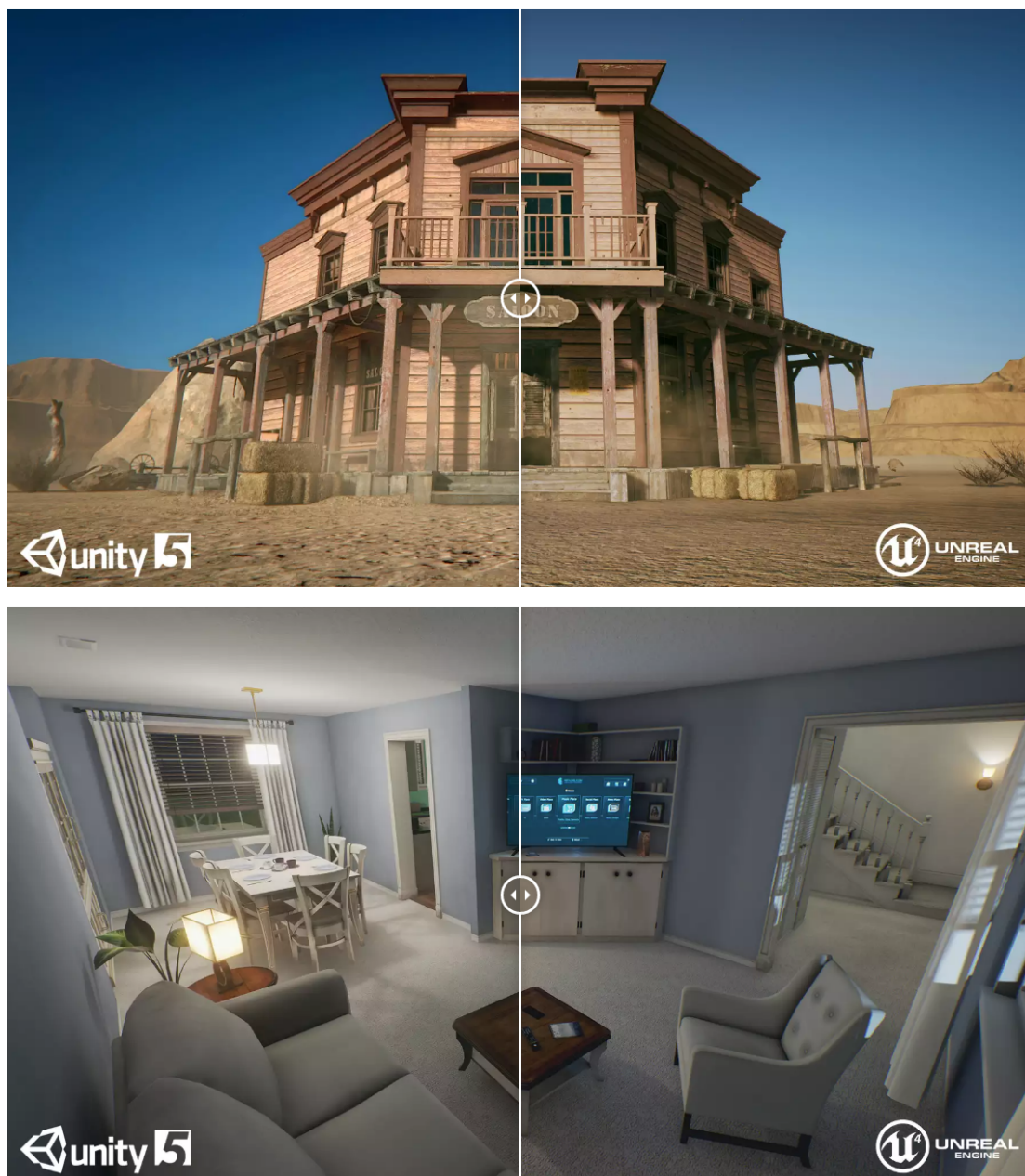
Je nástroj sloužící k vytváření multimediálního obsahu, počítačových her, filmů, předváděcích míst a virtuální reality. Herní engine je soubor funkcí umožňující práci s 2D a 3D grafikou, scénou, grafem scény, umělou inteligencí, vykreslováním, fyzikou, skripty, audiem, videm a dalšími. Vývoj vlastního herního enginu je náročná a dlouhotrvající záležitost, vyžadující tým zkušených profesionálů. Z toho důvodu je výhodné používat herní engine dostupné na trhu. Výhodou současných herních enginů je možnost multiplatformního exportu či dostupnost jejich dokumentace online, často navíc nabízí rozšiřitelnou funkčnost a uživatelské rozhraní. Jsou navrženy pro efektivní použití, uživatelské rozhraní je přehledné a v případě, že potřebujete pomoci, mají širokou uživatelskou základnu ochotnou vám poradit.

Nyní se blíže seznámíme s herními enginey, které jsem uvažoval pro využití v této práci. Hlavními kritérii pro výběr enginu jsou: podpora současných zařízení pro virtuální realitu, vzhledem k zadání export pro Android a PC, fyzikální simulace, jednoduché použití, dobrá dokumentace, schopnost importovat běžně používané formáty 3D modelů a možnost použití zdarma. Vybíral jsem mezi Unreal Engine 4 [4] a Unity 5 [5], které jsou hodnocené jako nejlepší herní enginey [23] podporující vývoj her pro virtuální realitu a splňují předchozí kritéria.

Unreal Engine 4 je herní engine, za kterým stojí společnost Epic Games, která má velké zkušenosti na poli AAA her. Jelikož tento engine vznikl z FPS hry, je vhodný pro vytváření FPS her. Na úkor toho vytváření her, které nejsou FPS, je o trochu složitější. V Unreal Enginu je implementováno mnoho užitečných nástrojů, které se hodí při vývoji her. Animace, post procesing, editor materiálů, generátor LOD, umělá inteligence pomocí behaviorálních stromů, blueprint (grafické programování), editor videa a šlo by pokračovat dále. Engine podporuje export pro hlavní herní platformy Windows PC, PlayStation 4, Xbox One, Mac OS X, iOS, Android, VR (SteamVR/HTC Vive, Oculus Rift, PlayStation VR, Google VR/Daydream, OSVR a Samsung Gear VR), Linux, SteamOS, a HTML5. Fyzika je řízena fyzikálním enginem PhysX 3.3. Použití není jednoduché a učící křivka je poměrně pomalá. Dokumentace je také poměrně problematická, co se týče přehlednosti a kompletnosti. Na internetu je mnoho tutoriálů pro grafiky, ale ani zdaleka tolik jich není pro programátory. Unreal engine podporuje import běžně používaných 3D formátů FBX a OBJ. Unreal engine je možné použít zdarma, pokud výdělek nepřesáhne \$3000 za kalendářní čtvrtletí, jinak si společnost Epic Games nárokuje 5 % z hrubého výdělku.

Unity 5 je herní engine, na kterém bylo v posledních letech odvedeno opravdu velké množství práce a díky tomu se může řadit mezi nejlepší volně dostupné herní enginey. Má velmi rychlou učící křivku, přehledné uživatelské rozhraní a opravdu excelentní dokumentaci s velkým množstvím návodů. Díky velké uživatelské základně je většina problémů s enginem rychle nalezena a řešena tvůrci enginu, nebo na internetu existuje postup, kterým tento problém řešil někdo jiný. Přestože Unity nepřichází s tak rozsáhlou nástrojovou sadou jako Unreal Engine, velké množství těchto nástrojů je na-

hraditelné nástroji, které je možné stáhnout zdarma nebo za malý poplatek z Unity Asset Store. Unity Engine podporuje export pro širokou škálu platforem Windows PC, Mac OS X, Linux, PlayStation 4, Xbox One, Wii U, Android, iOS, Windows Phone, Tizen, VR(SteamVR/HTC Vive, Oculus Rift, PlayStation VR, Google VR/Daydream a Samsung Gear VR, Microsoft Hololens), Android TV, SamsungSmart TV, tvOS, Nintendo Switch, WebGL. Fyzika je řízena fyzikálním enginem PhysX 3.3. Unity engine podporuje import běžně používaných 3D formátů FBX, OBJ a pokud je na počítači nainstalovaný některý z podporovaných modelářů, umí přímo pracovat i s jejich proprietárními formáty. Unity engine je možné použít zdarma, pokud výdělek nepřesáhne \$100 000 za poslední fiskální rok.



Obrázek 2.4. Vizuální srovnání Unreal engine 4 vs Unity 5. Převzato z [6].



Obrázek 2.5. Vizuální srovnání Unreal engine 4 vs Unity 5. Převzato z [6].

Unity 5 a Unreal Engine 4 používají rozdílné programovací jazyky. Unreal Engine 4 používá C++ s nadstavbou UScript. Unity 5 pro psaní skriptů používá C#. Pro spuštění hry se v Unreal Enginu musí celý kód zkompileovat, což trvá řádově několik sekund až pár minut, mezitím co v Unity je jakákoli změna téměř okamžitá, a proto umožňuje rychlejší prototypování. Nevýhoda Unity oproti Unreal Enginu pro virtuální realitu, která plyne z použití C# je, že správu paměti si neřídí programátor sám a běh Garbage Collectoru by mohl u některých aplikacích vyžadujících hodně paměti způsobovat potíže.

Dále je možné se zaměřit na vizuální stránku hry vytvořené v jednotlivých enginech. Jak se pokoušeli dokázat ve vizuálním porovnání [6] na blogu Not Lonely, mezi vizuální stránkou her v Unity a Unreal Enginu není velký rozdíl. Ve srovnání byly vytvořeny dvě identické scény v Unity a Unreal Engine, stejné textury, modely a nasvícení. Výsledky srovnání je možné vidět na obrázku 2.4.

Rozdíly mezi těmito enginey z hlediska virtuální reality, možností exportu a fyziky jsou nepatrné, oba podporují podobné množství frameworků pro virtuální realitu, dokonce z hlediska výsledného obrazu se moc neliší, jak je vidět na obrázku 2.4. Spíše než na podstatných rozdílech, záleží u těchto dvou engineů na osobních preferencích. Vzhledem k předchozím zkušenostem jsem se rozhodl používat herní engine Unity 5.

2.2.7 Uživatelské rozhraní

V klasických PC hrách překrývá často uživatelské rozhraní část obrazovky, aby hráče informovalo o jeho zdraví, skóre, zkušenostech a dalších informacích z virtuálního prostředí. Jedná se tedy o 2D reprezentaci informace v prostoru kamery, často je taky nazývána jako HUD(Heads Up Display). Tento přístup často ve virtuální realitě nefunguje, protože takovým uživatelským rozhraním uživateli stíníme jeho viditelný prostor. Dalším důvodem je, že vybraný herní engine Unity nepodporuje uživatelská rozhraní vykreslená v prostoru kamery[14] v zařízeních pro virtuální realitu.

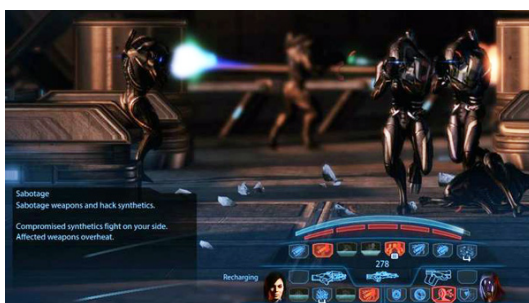
Z toho vyplývá, že pro reprezentaci uživatelského rozhraní nebude možné používat takové, které překrývá obrazovku, a tedy je vykreslené v prostoru kamery. Není ale nutné omezovat se pouze na 2D obraz v prostoru kamery. Uživatel se nachází ve virtu-

		Je reprezentace vizualizována ve 3D virtuálním světě?	
		Ne	Ano
Je uživatelské rozhraní součástí virtuálního světa?	Ne	nediegetické uživatelské rozhraní	prostorové uživatelské rozhraní
	Ano	meta uživatelské rozhraní	diegetická uživatelské rozhraní

Obrázek 2.6. Základní skupiny uživatelského rozhraní vzhledem k virtuálnímu světu [15].

álním prostředí, kde je kolem něj mnoho objektů virtuálního světa a je přímo žádoucí, aby s nimi bylo možné interagovat. Objekty virtuálního světa už samy o sobě nesou určitou míru informace a navíc s trochou kreativity je možné je použít k vyjádření uživatelského rozhraní.

Podle vztahu k virtuálnímu světu můžeme uživatelská rozhraní dělit do čtyř základních skupin [15] viz 2.6 a to diegetické uživatelské rozhraní, nediegetické uživatelské rozhraní, meta uživatelské rozhraní a prostorové uživatelské rozhraní.



Obrázek 2.7. Příklad nediegetického uživatelského rozhraní ze hry Mass Effect 3 [7].

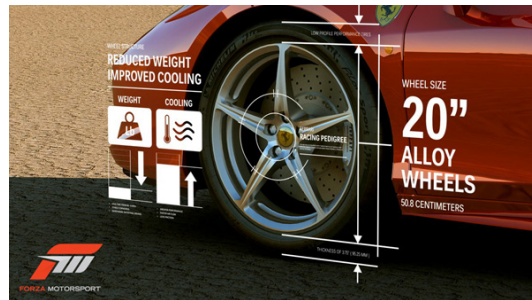


Obrázek 2.8. Příklad diegetického uživatelského rozhraní ze hry Syndicate vpravo [7].

Nediegetické uživatelské rozhraní viz 2.7 je takové, které není součástí fiktivního herního ani virtuálního světa. Jedná se o klasické 2D uživatelské rozhraní vykreslené mimo herní svět na část obrazovky informující o stavu hry. Obvyklými zástupci jsou různé ukazatele zdraví, zkušeností a peněz v počítačových hrách.



Obrázek 2.9. Příklad meta uživatelského rozhraní ze hry Call of duty: Modern warfare 2 [7].



Obrázek 2.10. Příklad prostorového uživatelského rozhraní ze hry Forza Motorsport [7].

Diegetické uživatelské rozhraní viz. 2.8 je rozhraní, které je součástí fiktivního herního světa a současně má svou 3D reprezentaci i ve virtuálním světě. Může se jednat například o mapu či kompas, který slouží uživateli k navigování virtuálním prostředím.

Meta uživatelské rozhraní viz 2.9 je součástí herního světa, ale nemá reprezentaci ve virtuálním světě. Jedná se o vizuální efekty spjaté s herním zážitkem, jako je například rozmazání obrazovky při únavě z dlouhého běhu, nebo třeba efekt krve na obrazovce při obdržení zranění.

Prostorové uživatelské rozhraní viz 2.10 je součástí virtuálního, ale již ne herního světa. Tímto způsobem mohou být vyjádřeny například textové informace navigující uživatele ve virtuálním světě. Pokud je text hodně dlouhý, je vhodné nepromítat tento text na rovinu, nýbrž na stěnu válce o vhodném poloměru. Tímto přístupem jsme schopni komfortněji zprostředkovat text uživateli a zvýšit tak jeho čitelnost v perspektivním promítání.

Z výše uvedených informací je možné usoudit, že pokud se budeme snažit vytvořit uživatelské rozhraní pro virtuální realitu, je dobré používat buď diegetické uživatelské rozhraní nebo prostorové uživatelské rozhraní.

2.2.8 Atlas textur



Obrázek 2.11. Atlas textur 16x16 ze hry Minecraft [8].

Je kolekce malých textur, příslušících 2D nebo 3D objektům, uložených ve velké textuře viz 2.11. Atlas textur se používá pro optimalizaci vykreslování, snižuje počet použitých texturovacích jednotek a přepínání stavů na grafické kartě. Dále umožňuje sloučit dohromady modely, které se mají poslat na grafickou kartu, a to za podmínky, že tyto modely sdílejí stejný materiál. Takto je možné snížit počet „draw calls“, neboli počet vykreslování na grafické kartě. To vede ke zkrácení doby potřebné pro vykreslení snímku.

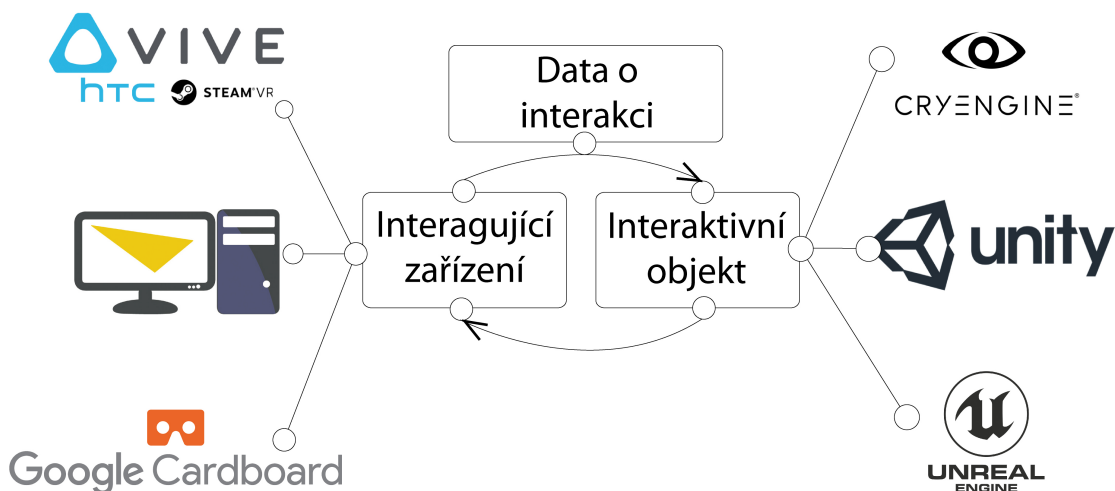
Atlas textur je možné vytvořit ručně vyrovnáním textur do atlasu, nebo je možné použít nástroj, který to udělá automaticky. Po sloučení textur do atlasu je nutné ještě upravit texturovací souřadnice modelů, aby byly správně naškálované a napozicované na příslušnou texturu v atlasu.

Kapitola 3

Návrh řešení

3.1 Multiplatformní systém interakce

V zadání jsou specifikována zařízení pro virtuální realitu, na kterých má být možné tuto hru hrát, a to Google Cardboard v2, PC a HTC Vive. Je rozumné požadovat určitý přesah. Pokud vyvíjíme hru v herním engineu jako je Unity, který podporuje export pro širokou škálu platform, byla by škoda přijmout řešení, které se striktně vymezuje pouze na tato tři zařízení. Kdybychom chtěli postupovat co nejjednodušší a přímou variantou, naprogramovali bychom hru pro každé z těchto zařízení v podstatě znova. Tento způsob provedení je však zbytečně pracný. Umožňuje nám však využít všechny možnosti, které vstupní zařízení a engine poskytují.

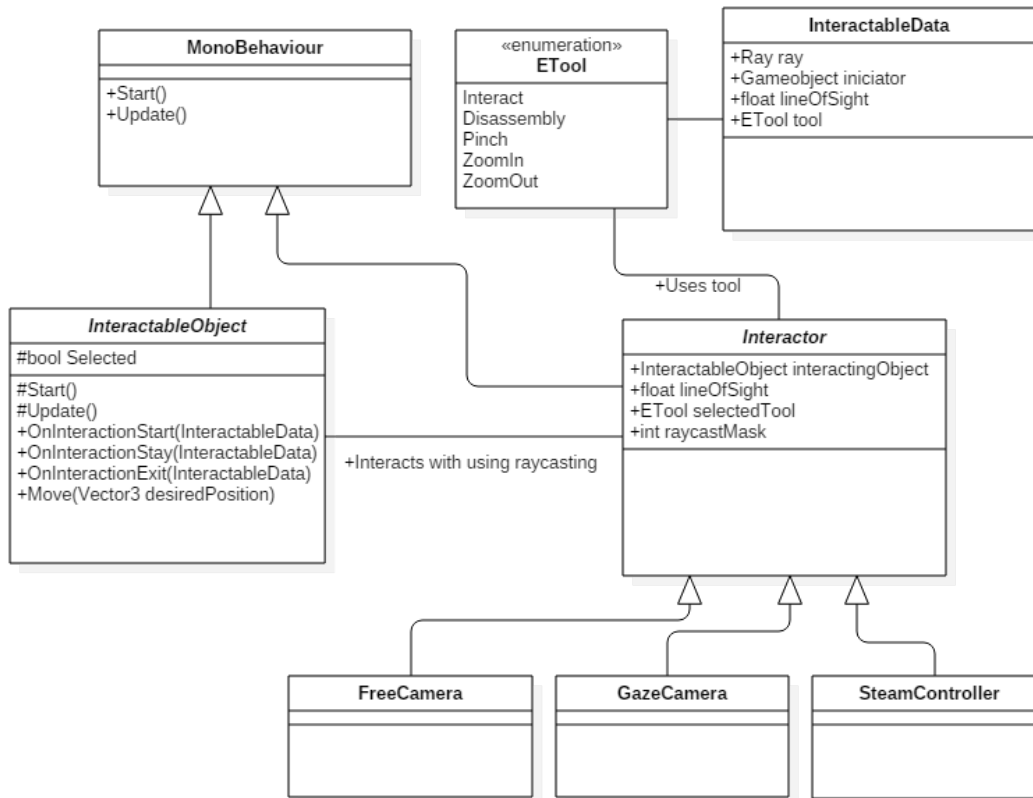


Obrázek 3.1. Návrh abstraktní vrstvy propojující vstupní zařízení a interaktivní objekty.

Já jsem se namísto toho zaměřil na návrh systému interakce, viz 3.1, který je schopný interakci napříč různými zařízeními pro virtuální realitu a pro jednotlivé objekty ve scéně sjednotit. Koncept, který jsem navrhl, je dostatečně abstraktní, aby mohl být jednoduše aplikovatelný napříč širokou škálou herních engineů. Libovolné vstupní zařízení může s objekty ve scéně interagovat. To je možné například pomocí výběru přes šablonu (stencil buffer), pomocí vrhání paprsku (jako v Unity 5) či pomocí kolizí. Při interakci s interaktivním objektem jsou na tomto objektu volány události informující o začátku interakce, setrvání v interakci a ukončení interakce. Během těchto událostí posílá interaktivní zařízení vybranému objektu data o interakci, obsahující například pozici, na kterou by měly být objekty přesunuty. Pomocí těchto událostí a obdržených dat lze jednoduše naprogramovat chování objektu. Navíc se může stát, že pomocí vybraného objektu ve scéně chceme upravit chování interagujícího zařízení, proto tento systém umožňuje volat události zpět na interagujícím zařízení. Hlavní výhodou tohoto

přístupu je, že na všech platformách a zařízeních je chování objektů totožné. A jelikož je chování objektů pro všechny vstupní zařízení implementováno pouze jednou, je mnohem snazší opravit chybu v programu.

Konkrétní návrh systému interakce je nutné upravit podle možností implementovaných v herním engine. V Unity 5 je možné vybírat objekty pomocí vrhání paprsku nebo pomocí kolizí. Pouze HTC Vive z výše uvedených zařízení umožňuje interakci oběma uvedenými způsoby – jak interakci pomocí vrhání paprsků, tak interakci pomocí kolizí, mezitím co PC a Google Cardboard v2 jsou vzhledem ke svým možnostem víceméně omezeny pouze na interakci pomocí vrhání paprsku. Vzhledem k tomuto faktu jsem se rozhodl sjednotit i tento přístup, a nadále používat pouze interakci pomocí vrhání paprsku a na všech zařízeních poskytnout podobný herní zážitek. V některých ze zadaných her by se bez použití tohoto přístupu nedalo obejít a jakákoli manipulace s objekty by byla velmi obtížná. V diagramu tříd viz 3.2 je možné vidět konkrétní návrh pro Unity 5.



Obrázek 3.2. Realizace multiplatformního systému interakce v Unity 5.

Třídy *Interactor* (reprezentuje interagující zařízení) a *InteractableObject* (reprezentuje interaktivní objekt) dědí z třídy *MonoBehaviour*, což je základní třída v engine Unity, která umožňuje hernímu objektu provést akci při vytvoření objektu a během každého snímku. *Interactor* během každého snímku vysílá do scény paprsky. *RaycastMask* umožňuje pomocí vrstev upravit, které objekty mají být paprskem zasaženy. Na jednotlivé vstupy vstupního zařízení mohou být namapovány různé akce. A zde se dostáváme k jednomu z problémů. Každé zařízení má různý počet vstupů, tlačítek a podobně. Google Cardboard v2 má pouze jedno tlačítko, HTC vive pět tlačítek a touchpad a na PC

můžeme využít téměř všechny klávesy a myš. Proto třída `Interactor` obsahuje proměnnou `SelectedTool`, která určuje, jaká akce se má při interakci provést. Interakce tedy může být efektivně namapována na jeden vstup a změnou `SelectedTool` docílíme odlišné funkčnosti. Proměnná `lineOfSight` je inicializována vždy na počátku interakce. Jedná se o vzdálenost mezi interagujícím zařízením (`Interactor`) a interaktivním objektem (`InteractableObject`). Tato vzdálenost slouží v průběhu interakce k výpočtu pozice, na kterou by se měl interagující objekt přemístit. Zmenšováním této vzdálenosti docílíme toho, že se interagující objekt bude přibližovat, zvětšováním naopak docílíme oddalování.

Potomci třídy `InteractableObject` mohou provádět nějakou akci v každém snímku nezávisle na vstupních zařízeních, nebo pomocí metod `OnInteractionEnter`, `OnInteractionStay` a `OnInteractionExit`, volaných při interakci z `Interactoru`, reagovat na vstupní zařízení. Jelikož `Interactor` nemusí být ve scéně pouze jeden, jako je tomu v případě ovladačů HTC Vive, obsahuje třída `InteractableObject` proměnnou, která umožňuje informovat nebo zamezit interakci dalšího `Interactoru`. Interakce pomocí vrhání paprsku nutně implikuje, že každý interaktivní objekt musí mít nějakou kolizní obálku. Pokud se má navíc tento objekt hýbat, je velmi vhodné, aby měl přiřazenou i komponentu `Rigidbody`[9], z důvodu optimalizace výpočtu fyziky. Dalším důvodem je, že pokud nastavujeme pozici objektu pomocí transformací namísto `Rigidbody`, nemůžeme kolize kontrolovat kontinuálně. Tedy pokud v prvním snímku nastavíme pozici objektu před zeď a v následujícím snímku za zeď, objekt se prostě a jednoduše objeví za zdí. Pokud ten samý pohyb správně vyjádříme pomocí komponenty `Rigidbody`, objekt se v prvním snímku objeví před zdí a v následujícím se o zeď zarazí. Použitím `Rigidbody` lze tedy dosáhnout realističtějšího pohybu objektu.

Při volání metod `OnInteractionEnter`, `OnInteractionStay` a `OnInteractionExit` jsou z `Interactoru` do `InteractableObjectu` předávána data o interakci. Vzhledem k tomu, že jsem se rozhodl používat interakci pomocí vrhání paprsků, obsahují tato data paprsek, kterým byl objekt zasažen, herní objekt reprezentující vstupní zařízení, vzdálenost, ve které byl objekt paprskem zasažen, a pozici, na kterou by se měl případně objekt přesunout, pokud tuto funkčnost implementuje. Tím, že je v datech o interakci předáván i herní objekt reprezentující vstupní zařízení, je zajištěno, že interagující objekt může reagovat zpět na vstupní zařízení.

3.2 Ovládání

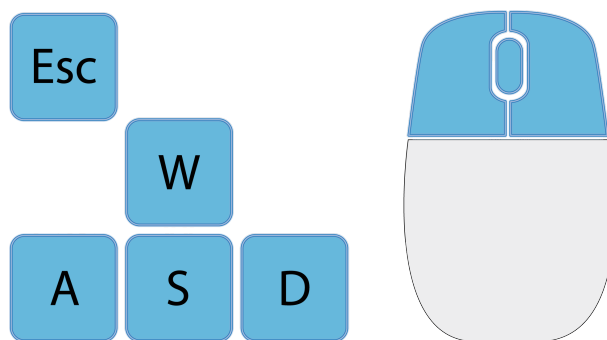
V této kapitole popíšu navržené ovládání a způsob, kterým jsem řešil navigaci a mapování vstupů na jednotlivých zařízeních. Jak jsem již popsal v předchozí kapitole 3.1, nedostatek vstupů některých zařízení je zajímavý a zásadní problém, a v této kapitole se mu tak budu dále věnovat.

3.2.1 PC

Při návrhu ovládání pro PC jsem se držel zaběhnutých a lety odzkoušených principů, které se běžně používají při tvorbě FPS her [10]. Pohyb avatara ve virtuálním prostředí je realizován pomocí kláves a rozhlížení pomocí myši, viz 3.3. Při návrhu ovládání pro jednotlivé hry jsem se snažil vytvořit velké množství funkčnosti a tuto funkčnost namapovat na co nejmenší počet vstupů. Jak jsem diskutoval v předchozí kapitole 3.1, pomocí nástrojů lze vyjádřit libovolnou funkčnost. V průběhu návrhu scén 3.3 se ukázalo, že maximální počet nástrojů, které jsem potřeboval použít, jsou dva. Z toho důvodu bylo možné namapovat nástroje na pravé a levé tlačítko myši. V případě, že by maximální počet nástrojů byl větší než dva, muselo by být pro výběr nástrojů

vytvořeno speciální uživatelské rozhraní, nebo by bylo třeba jejich mapování na některé z kláves. Pokud by se měl uživatel setkat se hrou pouze jednou v životě, bylo by použití uživatelského rozhraní pro výběr nástrojů jasná volba, protože by zvýšilo přehlednost a udělalo hru intuitivnější. Pokud by naopak byla hra určena pro opakované hraní, mapování na klávesy zvyšuje efektivitu uživatele.

Při interakci pomocí vrhání paprsku vychází paprsek z kamery a prochází středem obrazu vykresleného touto kamerou. V některých situacích by bylo těžké odhadnout z vykresleného obrazu, na který objekt ve scéně paprsek míří. Abych usnadnil interakci s objekty, přidal jsem bílý ukazatel do středu vykreslovaného obrazu jako součást uživatelského rozhraní pro PC.



Obrázek 3.3. Ovládání pomocí klávesnice a myši pro PC.

Ovládání klávesami

- Esc – Návrat do výběru scén 3.3.1
- W – Pohyb vpřed
- S – Pohyb vzad
- A – Pohyb doleva
- D – Pohyb doprava

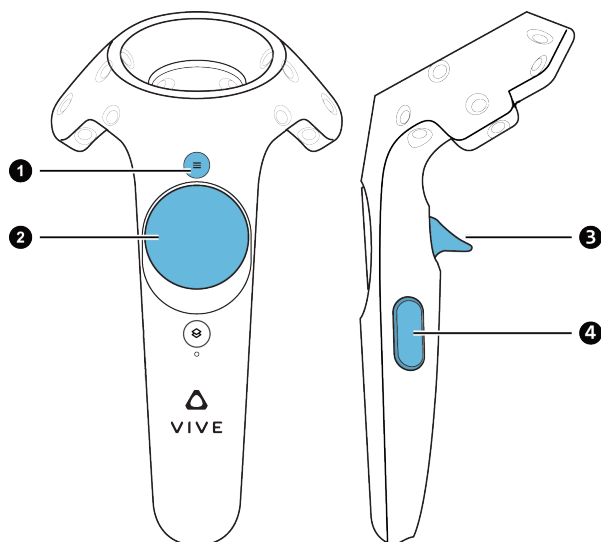
Ovládání myši

- Levé tlačítko myši – Nástroj Interakce
- Kolečko myši – Přiblížení nebo oddálení objektů při interakci
- Pravé tlačítko myši – Pro scénu specifický nástroj, nebo žádná akce

■ 3.2.2 HTC Vive

Jelikož HTC Vive je rok staré zařízení a obsahu pro virtuální realitu není mnoho, nedá se v tomto případě mluvit o zaběhnutém standardu, jako tomu je v případě ovládání pro PC 3.2.1. Při návrhu ovládání pro HTC Vive jsem se inspiroval v ostatních volně dostupných aplikacích na službě Steam [17], se kterou velmi úzce spolupracuje i HTC Vive. Ve výsledku jsem přišel s vlastním řešením. Snažil jsem se co možná nejvíce využít tvar ovladačů HTC Vive a pomocí toho docílit i intuitivního použití. Vstupy, které jsem použil pro realizaci ovládání, je možné vidět na obrázku 3.4, kde jsou vyznačeny modře.

Interakci nástrojem uloženým v SelectedTool jsem namapoval na spoušť (3). Díky vhodné konstrukci ovladače pokládá uživatel, který drží ovladač, intuitivně ukazováček na spoušť, což nápadně připomíná použití myši. Uživatel znalý práce s PC byl si měl být schopen jednoduše spojit kliknutí se stiskem spouště. Prostřednictvím touchpadu (2) jsem nahradil funkčnost kolečka myši. Touchpad se na ovladači ovládá pomocí palce a je snadné ho používat zároveň se spouští, což umožňuje jejich vzájemné kombinování.



Obrázek 3.4. Ovládání pomocí ovládačů HTC Vive.

Pomocí vertikální osy touchpadu (dotek v horní nebo dolní polovině touchpadu) je možné měnit hodnotu proměnné `lineOfSight`, která slouží ke změně vzdálenosti objektu při interakci. Protože ovladač HTC Vive disponuje pouze jednou spouští, je nutné nějakým způsobem umožnit uživateli vybírat nástroje. Každému ovladači HTC Vive jsem přiřadil cyklický seznam, specifický pro jednotlivé scény, ze kterého je možné vybírat dostupné nástroje. Kliknutím na touchpad v horizontální ose (v levé nebo pravé polovině touchpadu) je možné přepínat nástroje. Kliknutím vlevo se vybere nástroj uložený v cyklickém seznamu na indexu o jedna menším než je index aktuálně vybraného nástroje, kliknutím vpravo se vybere nástroj v cyklickém seznamu na indexu o jedna větším. Tlačítko (1), označené na ovladači ikonou pro menu, jsem se snažil využít opět intuitivním způsobem. Proto slouží k návratu do výběru scén 3.3.1. Model ovladače, který je reprezentován ve scéně, jsem doplnil o uživatelské rozhraní, jak je znázorněno na obrázku 3.5. Pomocí ikony umístěné nad touchpadem informuje uživatele o aktuálně vybraném nástroji. Dále jsou v uživatelském rozhraní ikony plus a mínus, které jsou umístěné na vertikální ose touchpadu, a na horizontální ose jsou obdobně umístěny šipky, aby tak tyto prvky korespondovaly s funkcí ovladače.

Navigaci ve scéně pro HTC Vive jsem realizoval kombinací následujících dvou přístupů. První přístup je směsicí fyzického a virtuálního pohybu a je přímo součástí implementované funkčnosti HTC Vive. HTC Vive přichází se dvěma kamerami, které snímají pozici helmy pro virtuální realitu (HMD - Head Mounted Display) a ovladačů v rozsahu prostoru viditelného těmito kamerami. To umožňuje přenášet pohyb z reálného světa do virtuálního. Pokud se tedy uživatel hýbe v prostoru snímaném kamerami, je HTC Vive schopný tento pohyb přenést do virtuálního světa. Maximální diagonální vzdálenost [16] kamer pro optimální herní zážitek je podle manuálu 5 metrů. Z toho vyplývá, že reálná herní zóna je prostor maximálně 3.5 x 3.5 metru při zachování rovnosti stran. Pomocí tohoto přístupu tedy není možné se pohybovat na velké vzdálenosti, a proto je tento přístup nutné kombinovat s jinou technikou navigace pro pohyb v rozsáhlých scénách.

HTC Vive je v Unity reprezentován ovladači, kamerou a kvádrem, který je o velikosti již zmíněné reálné herní zóny, a shodou okolností se také nazývá herní zóna. Pro odlišení ji dále budu nazývat virtuální herní zóna. Virtuální herní zóna je ohraničena mřížkou.



Obrázek 3.5. Uživatelské rozhraní ovladačů HTC Vive.

Pokud se uživatel přiblíží k okraji reálné herní zóny, tato mřížka se ve virtuálním prostoru zviditelní, aby uvědomila uživatele, že by mohl v reálném světě do něčeho narazit. Ovladače a kamera jsou v Unity potomky souřadného systému virtuální herní zóny. Pokud tedy přemístíme virtuální herní zónu, přemístí se s ní i ovladače a kamera. Využitím této vlastnosti můžeme zkonstruovat teleport, kterým je možné pohybovat virtuální herní zónou po scéně. Výběr místa, kam se má virtuální herní zóna přemístit, je prováděn pomocí ovladače. Lze k tomu využít buď metodu vrhání paprsku, pak se virtuální herní zóna přemístí do místa, kde paprsek narazil do virtuálního světa, nebo je možné jej realizovat vržením speciálního objektu, který když ve virtuálním světě dopadne, přenesse se virtuální herní zóna na místo dopadu tohoto objektu. V průběhu teleportace je vhodné uživateli postupně ztmavit viditelný obraz a po teleportování jej opět vrátit do normálního stavu. Rychlé přeblikávání a změna prostředí by mohly uživateli snadno způsobit kinetózu.

Pomocí teleportace je možné vyřešit pohyb ve scéně na velké vzdálenosti, což chybělo prvnímu přístupu. Namísto teleportace jsem se rozhodl použít přístup podle D. Dewhirsta [18] pomocí imitace běhu, jelikož je tento přístup mnohem reálnější a na kratší vzdálenosti i příjemnější. Princip tohoto přístupu jsem krátce popsal v kapitole Pohyb ve virtuální realitě (kapitola 2.2.4). Tlačítka (4) po stranách ovladače, též by je bylo možné nazvat jako úchop, jsou použita ke spuštění navigace. Směr a rychlost ovladačů jsou využity k výpočtu pohybu virtuální herní zóny. Implementační detaily a s tím související problémy jsou popsány v kapitole 4.1.

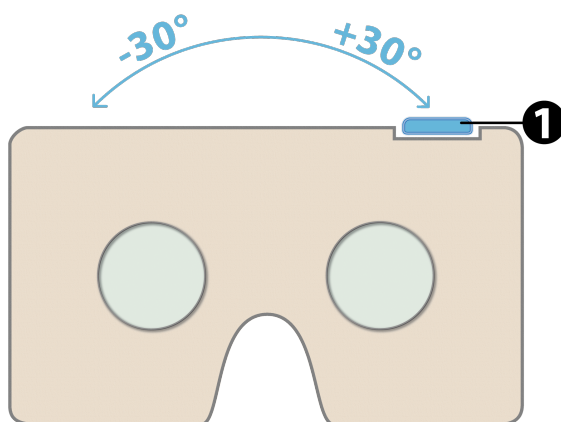
Podobně jako u PC i zde narážíme na problém, že nevíme, kam paprsek z ovladače míří. Proto další součástí uživatelského rozhraní ovladače je vizualizace paprsku, která umožňuje přesné zaměřování. Pokud paprsek míří mimo interaktivní objekt, je vizualizace paprsku modrá, v případě že míří na interaktivní objekt, je vizualizace paprsku zelená, a pokud je interaktivní objekt vybrán některým ovladačem, je vizualizace paprsku červená.

Ovládání HTC Vive

1. Tlačítko menu – Návrat do výběru scén 3.3.1
2. Touchpad
 - Dotyk na vertikální ose – Přiblížení nebo oddálení objektů při interakci.
 - Dotyk na horizontální ose – Změna nástroje
3. Spoušť – Použití nástroje uloženého v SelectedTool
4. Úchop – Navigace ve scéně

■ 3.2.3 Google Cardboard v2

Návrh ovládání pro Google Cardboard v2 byl opravdový oříšek. Interakci pomocí vrhání paprsku používá velmi podobně jako PC (popsáno v sekci 3.2.1). Ale vzhledem k tomu, že Google Cardboard v2 má pouze jedno tlačítko, viz (1) na obrázku 3.6, ovládání se komplikuje už při použití dvou nástrojů. Pokud navíc požadujeme stejné chování jako u ostatních zařízení, čili možnost si během interakce objekty přibližovat a oddalovat nebo umožnit návrat do menu, celý problém se komplikuje ještě více.



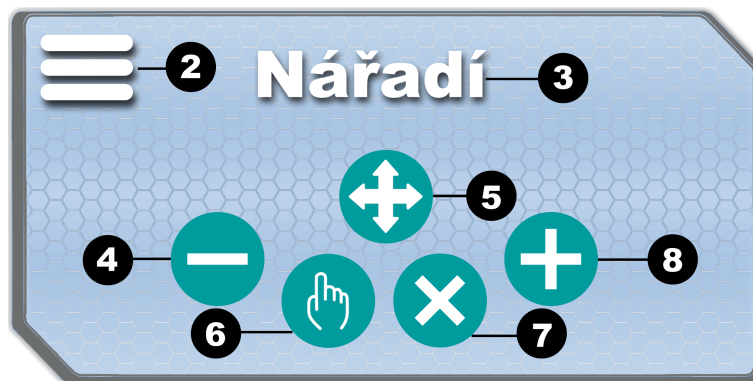
Obrázek 3.6. Ovládání pomocí Google Cardboard v2.

Abych nahradil nedostačující počet vstupů Google Cardboard v2, navrhl jsem uživatelské rozhraní, viz 3.7, které tyto nedostatky odstraňuje, jelikož umožňuje pohodlný výběr některého z nástrojů a návrat do menu. Nástroje a jejich počet, které je možné vybírat pomocí tohoto uživatelského rozhraní, je možné měnit tak, aby pokryly nároky konkrétní hry.

Ovládání Google Cardboard v2

1. Tlačítko aplikující vybraný nástroj 3.6
2. Tlačítko menu – Návrat do výběru scén 3.3.1
3. Tlačítko nástroje – Po stisknutí se uživatelské rozhraní přichytí k avataru, takže se pohybuje spolu s ním.

4. Nástroj pro přibližování – Shodná akce jako při výběru nástroje interakce, ale navíc ještě objekt po dobu interakce přibližuje.
5. Tlačítko pro pozicování menu
6. Nástroj interakce
7. Nástroj pro rozebírání
8. Nástroj pro oddalování – Shodná akce jako při výběru nástroje interakce, ale navíc ještě objekt po dobu interakce oddaluje.



Obrázek 3.7. Uživatelské rozhraní pro výběr nástroje v Google Cardboard v2.

Pro zvýšení komfortu jsem se rozhodl navrhnout druhou metodu pro přibližování a oddalování objektů a uživateli poskytnout obě možnosti. Pokud uživatel v průběhu interakce nakloní zařízení o 30° doleva, objekt se začne přibližovat, pokud zařízení nakloní o 30° doprava, objekt se začne oddalovat. Tato metoda se zdá být komfortnější než neustálé vyměňování nástrojů pro přibližování a oddalování, ale je nutné otestovat s uživateli, zda je to opravdu vhodnější varianta.

Navigace ve scéně je na Google Cardboard v2 realizována pomocí teleportu. Podlaha je speciálně ve verzi pro Google Cardboard v2 interaktivní a po interakci s ní umožňuje se do tohoto místa teleportovat, pokud není požadované místo vzdálené více než 10 metrů. Tato vzdálenost se mi zdála úměrná pro případ, že by se uživatel nechtěně teleportoval. Při vyšší vzdálenosti by mohlo dojít ke ztrátě kontextu ve scéně. Při teleportaci je uživateli zatemněn obraz a po dokončení teleportace opět navrácen do původního stavu.

Podobně jako na PC 3.2.1 je třeba informovat uživatele, kde je střed obrazu, kterým prochází vrhaný paprsek. K tomuto účelu jsem využil standardní ukazatel, který je součástí vývojových nástrojů pro Google Cardboard v2. Tento ukazatel je umístěn ve středu obrazu (každé oko má svůj vlastní ukazatel). Ukazatel je reprezentován kruhem, při interakci se zvětší, a změní se tak v kružnici o určité tloušťce.

3.3 Návrh scén

Scény byly podrobně specifikovány v kapitole 2.1 a krátce jsem popsal problémy, pro které jsem se snažil nalézt řešení. Přestože scény byly dobře specifikovány, zbylo zde hodně prostoru pro kreativní řešení a v této kapitole popíšu klíčové elementy při realizaci scén. Vzhledem k tomu, že někteří uživatelé z cílové skupiny už mohou mít zkušenosti s virtuální realitou a počítačovými hrami, předpokládám že nestráví velké množství času tím, že se budou učit systém ovládnutí, a proto začnou v poměrně krátké době objevovat připravený virtuální svět. Aby byla hra zábavná i po splnění zadaného úkolu, je možné interagovat i s objekty, které s úkolem a jeho plněním přímo nesouvisí.

3.3.1 Výběr scén

Úvodní scéna nebyla součástí zadání, přesto si myslím, že její použití v této hře je velmi vhodné. Jelikož je hra koncipovaná pro Google Cardboard v2 a HTC Vive, chtěl jsem dopřát uživateli možnost se po každé scéně zotavit, než se pustí do pokračování. V případě, že zotavení nepotřebuje, může se pustit bez prodlení do další scény. V opačném případě si může dopřát odpočinek bez ztráty kontextu. Další výhodou, která se týká i PC je, že ne každá střední škola, která by chtěla použít tuto hru v rámci výuky, bude mít dostatek zařízení pro každého studenta. Tímto způsobem je možné, aby se studenti prostrídali po jednotlivých scénách. Není třeba, aby jeden musel dokončit celou hru, než by byla spuštěna hra nová.

Pro úvodní scénu jsem zvolil tématické prostředí na oběžné dráze Země. Uživatelské rozhraní pro výběr scény je realizováno pěti zakřivenými plochami, viz 3.8, obklopujících uživatele tak, aby byla maximalizována jejich čitelnost. Při interakci s touto plochou se načte příslušná scéna.

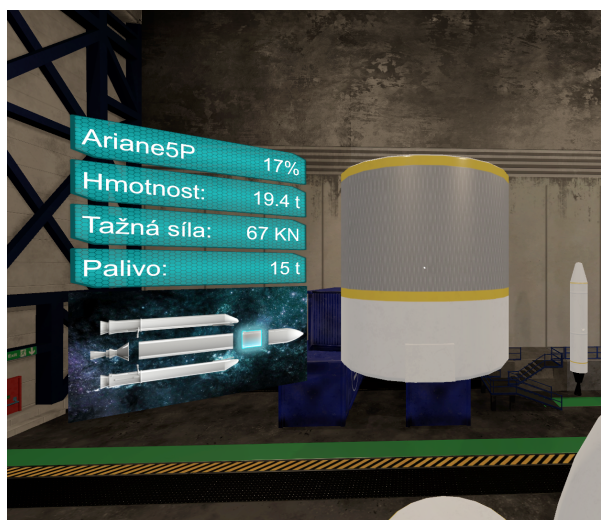


Obrázek 3.8. Uživatelské rozhraní pro výběr scén.

3.3.2 Scéna 1 – Stavba rakety



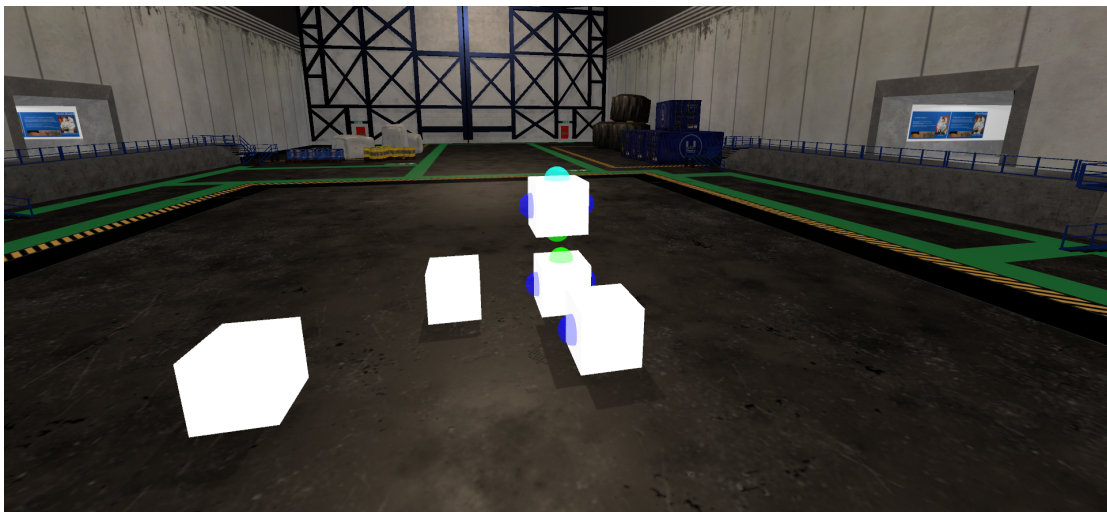
Obrázek 3.9. Uživatelské rozhraní zobrazující informace o části rakety.



Obrázek 3.10. Pozicování uživatelského rozhraní umístěného na válcové obálce objektu.

V této scéně jsem řešil problémy, jak uživateli prezentovat informace o částech rakety, se kterými interaguje, jak spojovat části rakety a navigaci ve scéně tak, aby stavba rakety byla efektivní a přehledná. Pro části rakety jsem navrhl informační panel, viz 3.9, který informuje o typu rakety, procentuální kompletnosti rakety, hmotnosti, tažné síle, a objemu paliva. Tento informační panel pro samostatnou část rakety zobrazí její vlastnosti, pro více spojených částí zobrazí součet jejich vlastností. Ve spodní části informačního panelu na obrázku 3.9 je pro samostatnou část rakety umístěn obrázek rakety, kterou uživatel staví, a je zvýrazněno, se kterým dílem interaguje. To usnadňuje vyhledávání dalších částí té samé rakety. Pro více spojených částí zobrazí informační panel v dolní části namísto obrázku rakety popisek stávající rakety.

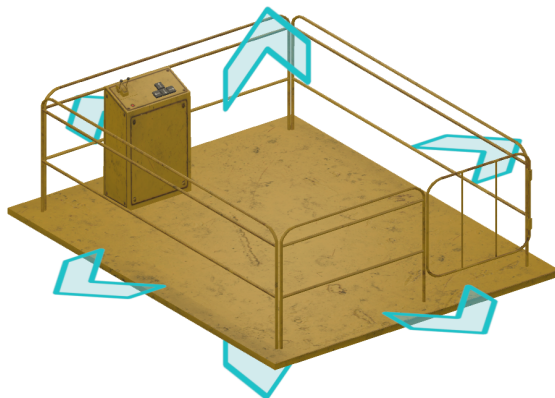
Důležitou součástí řešení problému je umístění informačního panelu k části rakety 3.10. Pro každou část rakety je na začátku hry vypočítán osově zarovnaný obalující kvádr, dále jen AABB (Axis-Aligned Bounding Box). Pro samostatnou část, nebo pro spojení částí, je pak vypočítán AABB obalující AABB jednotlivých částí. Nově vypočítaný AABB je rozšířen na obalující válec. Informační panel je do scény umístěn tak, že se pravým okrajem dotýká tohoto obalujícího válce, a pro pozorovatele se nalézá nalevo od objektu. Dopředný vektor informačního panelu leží v rovině XZ a míří směrem k pozorovateli.



Obrázek 3.11. Systém pro stavbu rakety.

Pro sestavení rakety jsem se snažil vymyslet co nejjednodušší systém, který by zároveň uživatele vizuálně naváděl, které díly je možné spojit. Spojování raket je realizované pomocí konektorů, viz 3.11. Konektory jsou barevné koule umístěné na místa, ve kterých by mělo dojít ke spojení objektů. Konektory jsou ve výchozím nastavení neviditelné. Při interakci se zviditelní nejen všechny konektory na části rakety, se kterou uživatel právě interaguje, ale také konektory jiných částí, které je možné s původní částí spojit. Součástí tohoto systému je i uživatelské rozhraní, které je reprezentované barvou konektorů. Konektory, které k sobě pasují, mají stejnou barvu. Když se dva konektory setkají, ověří se, zda k sobě pasují. Pokud ne, nedojde k žádné akci. Pokud k sobě konektory pasují, začnou se k sobě díly postupně přibližovat, dokud nedojde ke spojení středů konektorů, ve smyslu zástupných koulí. V tom okamžiku se části komponent RigidBody obou součástí rakety k sobě připojí prostřednictvím pevného spoje, který je reprezentován komponentou FixedJoint[11]. To umožňuje s oběma spojenými částmi hýbat jako s jedním celkem. Při spojení dvou částí rakety proběhne kontrola, zda raketa

již není kompletní. Pokud ano, bude moci uživatel scénu úspěšně opustit. Spojováním částí rakety pomocí komponent FixedJoint docílíme ještě jednoho užitečného efektu. Všechny propojené části spolu tvoří neorientovaný graf, který používám ke kontrole kompletnosti rakety. Pro účely této scény jsem zavedl speciální nástroj, který umožňuje uživateli rozdělit již spojené části rakety.



Obrázek 3.12. Navigace ve scéně pomocí pojízdné plošiny.

Jelikož výše zmíněný systém silně spoléhá na to, že na jednotlivé konektory uživatel uvidí, bylo potřeba vhodným způsobem umožnit uživateli pohybovat se ve scéně, a to nejlépe ve všech třech osách X, Y, Z. Z tohoto důvodu jsem použil pohyb ve scéně pomocí pojízdné plošiny, viz 3.12, která skvěle doplňuje industriální design scény. Plošina se může hýbat volně ve všech třech osách X, Y, Z, její ovládání je realizováno pomocí čtyř tlačítek, shodných se šipkami na klávesnici, a dvou pák, které slouží k pohybu plošiny nahoru a dolů.

■ 3.3.3 Scéna 2 – Start rakety

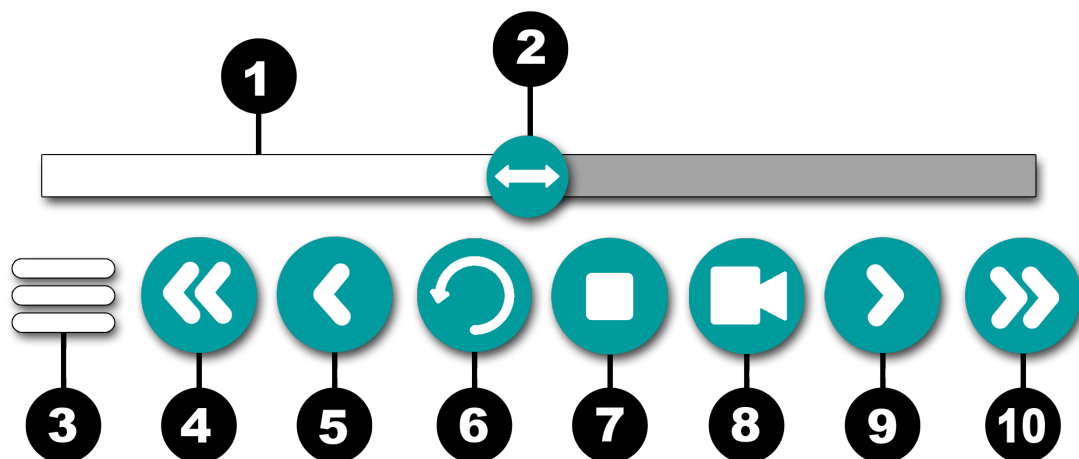
Ve scéně se startem rakety jsem řešil, jak zobrazit odlet rakety a přitom dát uživateli možnost ovlivňovat čas ve scéně. Pro převoz rakety, její start, odlet a doprovodné vizuální efekty jsem vytvořil jeden velký animační soubor, který toto vše řídí. Tento přístup mi umožnil nastavovat snímek, na který chci, aby se uživatel díval, rychlost přehrávání, zastavení animace a přehrání od začátku.

Abych zpřístupnil tyto funkce uživateli, navrhl jsem přehrávač 3.13, jehož uživatelské rozhraní je hodně podobné aktuálně používaným multimedialním přehrávačům. V tomto případě nemá smysl vymýšlet nový design uživatelského rozhraní, protože současně používaný design multimedialních přehrávačů je již mnohokrát otestovaný. Velkou výhodou tohoto designu je, že je notoricky známý a valná většina uživatelů s ním umí zacházet. A obdobně jsem se snažil používat i ikony, které jsou mezi uživateli známé, a snadno tak odvodí jejich funkčnost.

Popis tlačítek přehrávače 3.13

1. Časová osa
2. Posuvník
3. Menu – Návrat do výběru scén
4. Přehrávání zpět 3x
5. Přehrávání zpět 1x
6. Přehrát od začátku
7. Stop

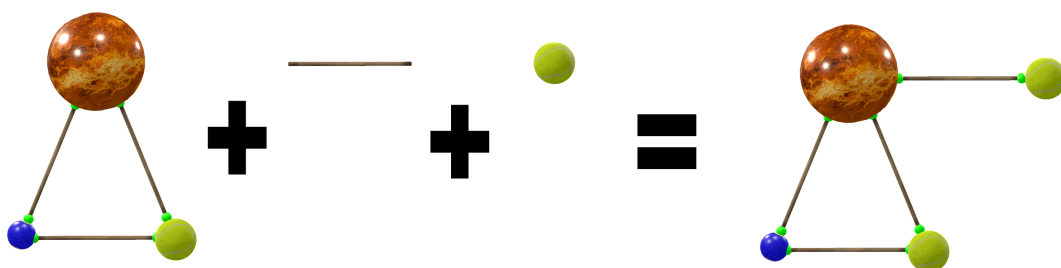
8. Změna pozorovatelný
9. Přehrávání dopředu 1x
10. Přehrávání dopředu 3x



Obrázek 3.13. Uživatelské rozhraní přehrávače pro řízení animace startu rakety.

■ 3.3.4 Scéna 3 – Barycentrické míčky

Ve scéně jsou umístěny čtyři šuplíky, které obsahují tyčku, bowlingovou kouli, tenisový míček a kulečnickovou kouli. Pokud ze šuplíku objekty vyjmete, zavře se a vygeneruje nový objekt tak, aby to uživatel nemohl vidět. Všechny čtyři objekty mají komponentu Rigidbody, takže je možné určovat jejich váhu a použít fyzikální systém v Unity, aby obstaral složité výpočty. Každá koule má rozdílnou váhu, aby bylo možné vytvořit i jednoduché nesymetrické soustavy. Pro stavbu soustavy objektů jsem použil podobné řešení jako při stavbě rakety 3.3.2.



Obrázek 3.14. Tvorba fyzikální soustavy.

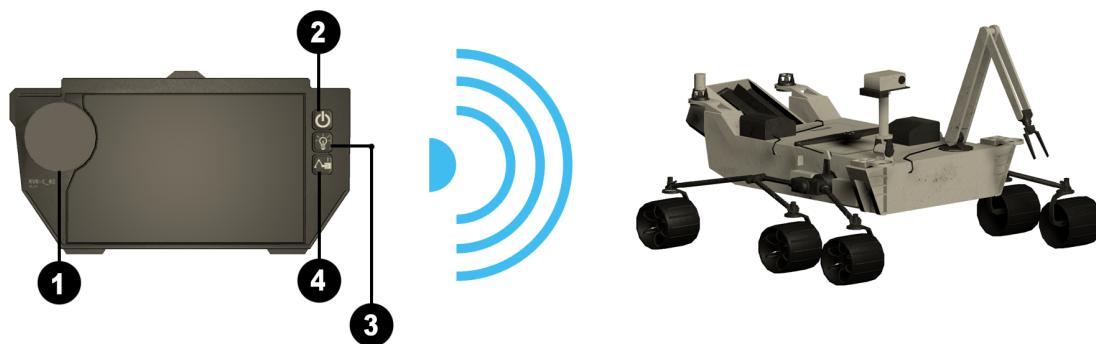
Koule jsou spojovány do soustavy tyčkami, viz 3.14, přičemž kouli je možné napojit dvěma způsoby. Pokud dojde ke spojení u konce tyčky, napojí se tyčka kolmo na kouli, v opačném případě dojde ke spojení bez jakékoli úpravy. Koule jsou připojovány k tyčkám opět pomocí pevného spoje FixedJoint, to ve výsledku umožňuje práci s celou soustavou a drží ji pohromadě. Po spojení dojde ke zvýraznění spojení pomocí malé zelené koule, která má pouze estetickou funkci. Pozice této spojovací zelené koule je vypočítána jako těžiště všech bodů, ve kterých tyčka koliduje s připojovanou koulí. Simulace fyziky na

takto definovaných objektech a soustavách je řízena Unity. Po uživateli je po sestavení soustavy již pouze vyžadován vstup ve formě aplikace síly na libovolné místo vytvořené soustavy. V této scéně je zaveden druhý speciální nástroj, kterým je akce strčení. Tento nástroj plně poskytuje možnost vstupu, který je od uživatele vyžadován. Účelem tohoto nástroje je aplikace síly působící ve směru paprsku vycházejícího z interagujícího zařízení v místě, kde paprsek narazí do soustavy.

Jelikož se tato scéna odehrává ve stavu beztíže a není možné, aby tento stav uživatel zažil během této hry, snažil jsem se alespoň vizuálně tento stav navodit. Pokusil jsem se tohoto docílit pomocí pomalé rotace modulu ISS, ve kterém se uživatel virtuálně nachází. Modul rotuje okolo osy Z v rozmezí -15° a 15° . Aby byla i navigace podobnější jako je při podmínkách ve stavu beztíže, uživatel se v této scéně nesmí pohybovat pomocí chůze. Namísto toho je nucen se přitahovat ke stěnám modulu. Přitahování na stěnu je realizováno pomocí interakce se stěnou, a tedy nevyžaduje žádný další speciální nástroj.

3.3.5 Scéna 4 – Sběr vzorků měsíčních hornin

Podle požadavků ze zadání jsem navrhl dálkové ovládání, které řídí měsíční vozítko, viz 3.15. Součástí dálkového ovladače je několik tlačítek, touchpad a obrazovka, na kterou se promítá obraz pořízený kamerou měsíčního vozítka tak, aby jej bylo možné ovládat i v případě, že na něj uživatel přímo nevidí. K ovládání pohybu vozítka slouží kruhový touchpad (1) umístěný v levé horní části ovladače. Vertikální osa touchpadu určuje směr chodu motoru – tedy zda vozítko jede vpřed nebo couvá. V horní i dolní polovině touchpadu je motor nastavený na maximální výkon nezávisle na vzdálenosti od pomyslné horizontální dělicí přímky. Horizontální osa určuje směr natočení předních kol vozítka, a tedy i jeho zatáčení. Kdyby nebyl v horní i dolní polovině touchpadu motor nastavený na maximální výkon, vozítko by se při maximálním natočení kol zastavilo.



Obrázek 3.15. Dálkové ovládání vozítka.

Tlačítko (2) slouží k zapnutí ovladače, tím se zahájí přenos obrazu mezi kamerou vozítka a ovladačem, zpřístupní se ostatní funkčnost a je možné pomocí touchpadu řídit vozítko. Tlačítko (3) umožňuje rozsvítit nebo zhasnout světlo v přední části vozítka. Tlačítko (4) slouží k odběru vzorků měsíční horniny robotickým ramenem. Tato funkčnost je zpřístupněna pouze v místě určeném pro odběr vzorků. Robotické rameno je řízeno pomocí inverzní kinematiky.

Aby se uživatelské rozhraní ovladače tematicky hodilo do scény s měsíční krajinou, je ovladač umístěn do rukou astronauta (viz obrázek 3.16) tak, aby to působilo, že jej uživatel drží sám. V tomto případě bude nutné použít dva různé přístupy k umístění

ovladače, jednu verzi pro HTC Vive a druhou pro Google Cardboard v2 a PC. V HTC Vive můžeme jednoduše umístit ovladač dopředu ve směru kamery a uživatel bude pomocí ovladačů HTC Vive schopen efektivně ovládat dálkový ovladač. Pokud bychom postupovali stejně na PC a Google Cardboard v2, vždy by bylo možné se dívat pouze do středu ovladače. Pro Google Cardboard v2 a PC se nastaví pozice ovladače teprve tehdy, pokud střed pohledu kamery je nalevo nebo napravo mimo ovladač.



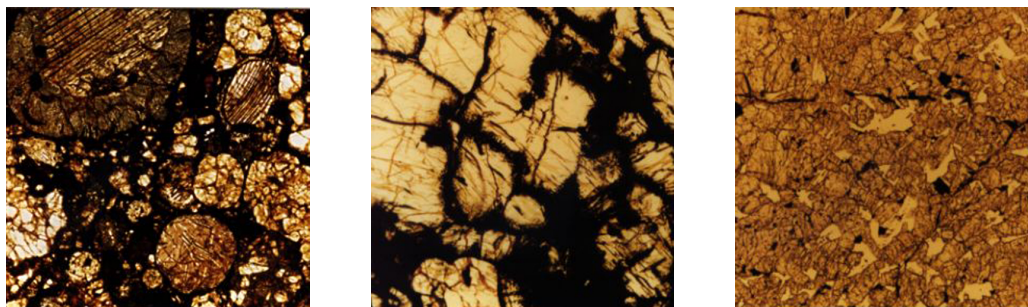
Obrázek 3.16. Ruce astronauta držící dálkový ovladač. Na obrazovce ovladače je možné vidět postavu astronauta. Zelené místo v dálce označené šipkou je cílové místo pro odběr vzorků. Oranžová hranice je maximální dojezdová vzdálenost vozítka.

Ruce astronauta jsou součástí uživatelského rozhraní, a proto jsou vidět z kamery avatara a nejsou vidět z kamery dálkově ovládaného vozítka. Naopak kamerou vozítka je vidět celá postava astronauta, která z kamery avatara vidět není. Toto rozhodnutí zjednodušuje práci s ovladačem a umožňuje udělat přirozenější animace pohybu pro astronauta. Dále je možné vidět na obrázku 3.16 místo pro odběr vzorků hornin, které je označené zelenou šipkou a zeleným prostorem. V tomto zeleném prostoru je zpřístupněna funkčnost pro odběr vzorků. Také je možné si povšimnout oranžové hranice, která specifikuje maximální dojezd vozítka.

Hra končí po odebrání vzorků ze tří různých stanovišť a návratu na základnu.

■ 3.3.6 Scéna 5 – Vyšetřování vlastností odebraných hornin

Tato scéna je velmi dobře specifikována zadáním a není zde moc prostor pro velkou kreativitu. Nejprve jsem se zabýval vytvořením nádob, ze kterých bude uživatel odebírat vzorky. Tyto nádoby jsou realizovány pomocí zásobníku, každá nádoba obsahuje 5 vzorků, při odebrání vzorku se změní množství horniny v nádobě. Obdobně tomu bude i při vrácení vzorku do nádoby. Vzhledem k tomu, že byly vzorky odebrány na třech různých stanovištích, jsou v této scéně umístěny tři nádoby obsahující jednotlivé vzorky. Pokud se uživatel pokusí smíchat vzorky, je notifikován o tom, že míchání vzorků je nepřijatelné. Vzorek je ale možné poté opět vyjmout a má původní vlastnosti. Klíčovými vlastnostmi vzorků pro tuto scénu je jejich mikroskopická struktura (na obrázku 3.17), obsah vápníku, hmotnost a podíl jednotlivých frakcí.



Obrázek 3.17. Obrázky výbrusů měsíčních hornin pod mikroskopem.

Dále jsou ve hře dostupná dvě síta, pomocí nichž je možné vzorek z nádoby rozdělít na různě hrubé frakce. Ze dvou sít vzniká kombinace pěti frakcí, kterými jsou: jemná, směs jemné a střední, střední, směs střední a hrubé a hrubá. Hmotnost každé frakce je určena nádobou a místem, odkud byly odebrány.

Vzorky a jednotlivé frakce je třeba zvážit, aby mohla být vypočítána jejich zrnitost. Proto je potřeba ve scéně realizovat váhu. Snažil jsem se v Unity vytvořit realistickou váhu z dostupných fyzikálních komponent. Tento přístup se nepodařilo realizovat, jelikož Unity neposkytuje nazpět mnoho informací z fyzikálního enginu, ze kterých by bylo možné váhu vypočítat. Namísto toho jsem použil jednoduchou aproximaci. Nad váhou je umístěn kolizní objekt, uvnitř kterého je počítána celková hmotnost, která na váze leží. Špatné na tomto přístupu je, že není realistický, a pokud by část objektu ležela na váze a část mimo, neposkytne správný výsledek. Váha je také snímána pouze v oblasti určené kolizním objektem. Vzhledem k tomu, že vzorky jsou oproti váze relativně malé objekty, je však tento problém zanedbatelný.

Dalším testem je možnost testovat vzorky na přítomnost vápníku. Ve scéně je umístěna nádoba s octem. Pro test v roztoku octa je opět vytvořen kolizní objekt. Pokud se vzorek nalézá uvnitř tohoto kolizního objektu, je zkontrolován na přítomnost vápníku. V případě, že vzorek vápník obsahuje, spustí se vizuální efekt šumění. V opačném případě se nic nestane.

Ve scéně je umístěn mikroskop s displejem, pod kterým je možné si prohlédnout zvětšené obrázky vzorků. V případě, že se jedná o nejhrubší frakci, je namísto zvětšeného obrazu na displeji zobrazena mikroskopická struktura vzorku.

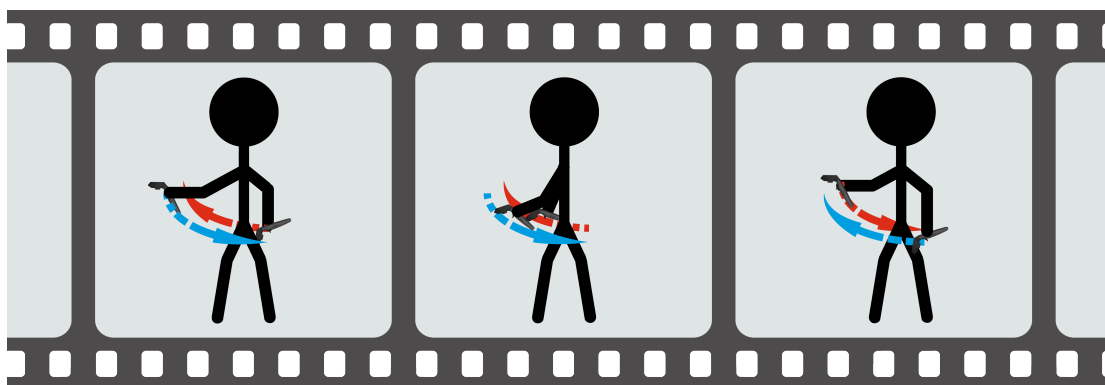
■ 3.3.7 Lokalizace

S přihlédnutím k tomu, že ESERO[1] je mezinárodní organizace, je zde předpoklad, že tato hra by mohla sloužit jako pomůcka výuky i mimo Českou republiku. Ve hře je celá řada textů – zadání na začátku scény, ukončení scény, nářadí pro Google Cardboard v2 a ještě například uživatelské rozhraní pro rakety, viz 3.9. Proto každé textové pole ve hře je přeložitelné a je označené speciální identifikátor (ID). Ve výběru scén 3.3.1 je na začátku hry možnost volby jazyka. Pomocí zmíněného ID a vybraného jazyka je textové pole schopné zjistit, jaký text má zobrazit. Lokalizace jsou uloženy v souborech ve formátu JSON, překlady z těchto souborů jsou načítány na začátku hry.

Kapitola 4

Implementace

4.1 Pohyb ve scéně pomocí HTC Vive ovladačů

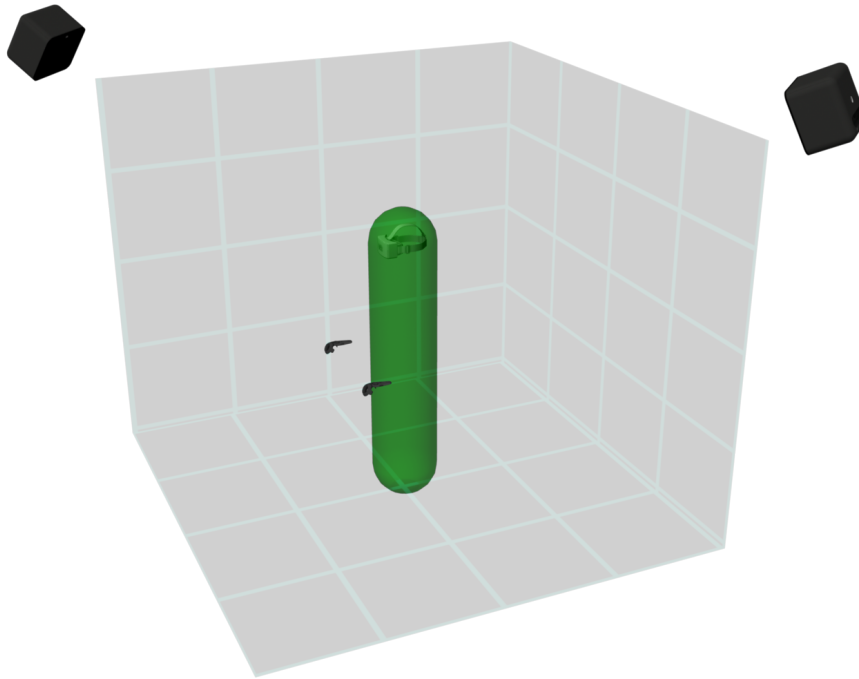


Obrázek 4.1. Imitace běhu pomocí ovladačů HTC Vive

V některých scénách je navigace řešena imitací běhu pomocí ovladačů HTC Vive. Imitací běhu je v tomto případě myšlen pouze pohyb rukou viz 4.1, který je převáděn na pohyb ve virtuálním prostoru. Jak jsem již zmínil v kapitole 3.2.2 zabývající se návrhem ovládání pro HTC Vive, v Unity je HTC Vive reprezentován virtuální herní zónou, ovladači a kamerou. Kamera a ovladače jsou potomky virtuální herní zóny. Pozice kamery i ovladačů jsou ve virtuální herní zóně automaticky nastavovány tak, aby korespondovaly s pozicí v reálné herní zóně.

V klasickém případě je na kameře umístěn nějaký kolizní objekt, který umožňuje detekovat kolize avatara, neboli virtuální reprezentaci uživatele, s prostředím. Pokud použijeme v Unity k vyjádření pohybu vhodné komponenty, kterými mohou být RigidBody nebo CharacterController[12], namísto prostého nastavování pozice, dosáhneme snadno realističtějšího chování. V souvislosti s tím však může nastat problém, který jsem také již dříve zmínil – pokud budeme nastavovat pozici objektu, nebo také kamery, pomocí transformací, může snadno dojít k nechtěnému procházení zdí nebo skrz jiné objekty. Pozice kamery ve virtuální herní zóně je nastavována automaticky, a to právě pomocí transformací. Pokud bychom přiřadili kolizní objekt k této kameře, příliš bychom tím nezískali. Naopak bychom způsobili hned několik problémů najednou. Pokud bychom se navíc snažili s kamerou hýbat, museli bychom se pak starat i o pohyb ovladačů. Dále bychom také byli ve sporu s implementací HTC Vive.

V tomto případě je potřeba kolizní objekt umístit v hierarchii výše a přiřadit ho přímo virtuální herní zóně viz 4.2. Pozice a výška kolizního objektu je v každém snímku přepočítána tak, aby byl zarovnán s pozicí kamery a také aby se adaptoval na výšku uživatele. To umožňuje ve hrách přímo využívat i výšku uživatele nebo uvažovat jeho skrčení. Vzhledem k tomu, že je kolizní objekt přiřazen virtuální herní zóně a pohyb



Obrázek 4.2. Kolizní objekt virtuální herní zóny.

je vyjádřen pomocí `CharacterControlleru`, je v případě, že dojde ke kolizi s obálkou, upravena trajektorie celé virtuální herní zóny, kamery i ovladačů.

Nyní, když je jasné, jakým způsobem bude v HTC Vive avatar kolidovat s prostředím, je čas popsat, jakým způsobem je realizována navigace ve scéně pomocí ovladačů. Stisknutím tlačítek po stranách ovladačů (4), též úchopy viz 3.4, se spustí režim navigace. Při imitaci běhu viz 4.1 jsou ovladače v pohybu. Jejich aktuální akcelerace je sečtena a exponenciálně vyhlazena s hodnotou z předchozího snímku tak, aby byl pohyb plynulejší. Výsledná akcelerace je použita pro výpočet velikosti vektoru rychlosti pohybu virtuální herní zóny. Směr rychlosti je vypočítán ze směru ovladačů. Dopředné vektory ovladačů se promítnou do roviny XZ, normalizují se a z normalizovaného vektoru jejich součtu je vypočítán výsledný směr rychlosti.

```

/*
 * This script is ment to be assigned to SteamVR [CameraRig]
 */
public class WalkingControllerScript : MonoBehaviour {

    public HandControllerScript left;
    public HandControllerScript right;
    /*
     * Character controller assigned to SteamVR [CameraRig]
     */
    private CharacterController controller;
    /*
     * Exponentially smoothed acceleration
     */
    private float smoothedAcceleration = 0.0f;
    /*

```

```

    * Exponential smoothing factor
    */
    private const float alpha = 0.7f;

// Use this for initialization
void Start () {
    if((controller = GetComponent<CharacterController>()) == null)
    {
        controller = gameObject.AddComponent<CharacterController>();
        controller.radius = 0.25f;
        controller.height = 2.0f;
        controller.center = new Vector3(0.0f, 1.0f, 0.0f);
    }
}

// Update is called once per frame
void Update () {
    Vector3 movement = Vector3.zero;
    Transform camera = Camera.main.transform;
    controller.center = new Vector3(
        camera.localPosition.x,
        camera.localPosition.y/2,
        camera.localPosition.z);
    controller.height = camera.localPosition.y;

    if (left == null)
    {
        left = transform.FindChild("Controller (left)")
            .GetComponent<HandControllerScript>();
    }
    if (right == null)
    {
        right = transform.FindChild("Controller (right)")
            .GetComponent<HandControllerScript>();
    }
    if (left != null && right != null
        && left.gripped && right.gripped)
    {
        //Get forward vectors
        Vector3 fw1 = left.transform.forward;
        Vector3 fw2 = right.transform.forward;
        //Project them to ground plane
        fw1.y = fw2.y = 0.0f;
        //Get average direction
        Vector3 direction =
            (fw1.normalized + fw2.normalized).normalized;

        //Exponentially smoothed acceleration
        float acceleration = smoothedAcceleration =
            ((left.Velocity.magnitude + right.Velocity.magnitude)
            * alpha / Time.deltaTime)
            + (1.0f-alpha) * smoothedAcceleration;
    }
}

```

```
        //Compute velocity in specified direction
        movement = direction * acceleration * Time.deltaTime;
    }
    controller.SimpleMove(movement);
}
}
```


Kapitola 5

Testování

Abych ověřil správnost implementace a vhodnost použitých prvků uživatelského rozhraní, uspořádal jsem uživatelské testování s několika participanty. Testování probíhalo v laboratoři virtuální reality na ČVUT a jeho předmětem byla vzniklá vzdělávací hra koncipovaná na tři různá zařízení pro virtuální realitu – HTC Vive, PC a Google Cardboard v2.

5.1 Informace o zařízeních

Testování verze pro HTC Vive a PC jsem provedl na počítači v laboratoři virtuální reality ČVUT na Karlově náměstí v Praze, parametry tohoto počítače je možné vidět v tabulce 5.1.

CPU	2x Intel Xeon E5-2630 v3
CPU počet jader	8
CPU počet vláken	16
CPU frekvence	2.4 GHz
CPU cache	20 MB
GPU	NVidia GeForce GTX TITAN Black
GPU paměť	6144 MB GDDR5
GPU počet jader	2880
GPU frekvence	967 MHz
RAM	64 GB
OS	Windows Server 2012 R2
SteamVR	Verze 1492726420

Tabulka 5.1. Parametry počítače v laboratoři virtuální reality.

Verzi pro Google Cardboard v2 jsem testoval na telefonu Samsung Galaxy S6 Edge, parametry tohoto telefonu je možné vidět v tabulce 5.2.

CPU	Exynos 7420 Octa-core	
CPU počet jader	4	4
CPU frekvence	2.1 GHz	1.5 GHz
GPU	Mali-T760MP8	
GPU počet jader	16	
GPU frekvence	700 MHz	
RAM	3 GB	
OS	Android 6.0	
Rozlišení	2560x1440	

Tabulka 5.2. Parametry telefonu Samsung Galaxy S6 Edge.

5.2 Informace o testování

Cílová skupina uživatelů pro testování byli studenti základních a středních škol, tedy lidé ve věku 8 až 20 let. Testování mělo tři poněkud odlišné cíle. Prvním z nich bylo prokázat bezchybnost programu. Druhým cílem bylo zjistit, zda navržená uživatelská rozhraní pro jednotlivá zařízení jsou intuitivní a dobře pochopitelná již po krátké instruktáži. A v neposlední řadě ověřit, jestli implementovaná navigace po scéně nezpůsobuje uživateli při hraní nevolnost.

Jak jsem již zmínil, testování proběhlo v laboratoři virtuální reality ČVUT na Karlově náměstí v Praze a zúčastnili se ho celkem čtyři participanti. Každý participant dostal za úkol postupně odehrát hru na všech třech zařízeních. Jelikož testování se zúčastnil dostatečný počet participantů, mohl každý začínat s jiným zařízením. Tudíž každé zařízení bylo otestováno alespoň jedním participantem, který hru nehrál ještě na žádném jiném zařízení.

S každým participantem jsem postupoval podle pevně stanoveného scénáře, který by měl odpovídat reálnému použití hry.

Scénář pro zařízení HTC Vive

- Úvod do tématu a děje hry
- Představení zařízení HTC Vive a implementovaného ovládání
- Zadání úkolu k první scéně
- Participant testuje hru
- Přestávka
- Zadání úkolu k druhé scéně
- Participant testuje hru
- Přestávka
- Zadání úkolu k třetí scéně
- Participant testuje hru
- Přestávka
- Zadání úkolu k čtvrté scéně
- Participant testuje hru
- Přestávka
- Zadání úkolu k páté scéně
- Participant testuje hru
- Přestávka
- Přesun na další zařízení

Každý participant po testování vyplnil dotazník, který je přiložen v příloze A. V první části dotazníku jsem se snažil zjistit, jaké klady a zápory pro ně hra měla a zda si participant odnesl nějaké nové informace. Ve druhé části dotazníku, která se týká přímo jednotlivých scén, jsem se snažil zjistit, zda je program bez chyb, jestli participant nepocítoval nevolnost a zda byli schopni pracovat s implementovaným uživatelským rozhraním. Dalším zajímavým aspektem je číselné hodnocení herních zážitků na jednotlivých zařízeních, to by mohlo v kombinaci s otevřenými otázkami určit zařízení, které má nejslabší herní zážitek, a proto je mu potřeba při dalším vývoji věnovat zvýšenou pozornost.

5.3 Výsledky získané z testování

Z odpovědí, které participanti uvedli do dotazníku A, vyplývá, že všichni participanti byli celkově s vytvořenou hrou spokojeni. V průběhu celého testování nebylo zjištěno,

že by některé součásti hry, ať už prvky uživatelského rozhraní nebo navigace ve scéně, způsobovaly nevolnost. Také během testování hry nebyla zjištěna žádná chyba, která by znemožnila hru plně hrát, dokončit úroveň nebo způsobila pád aplikace. Byly zjištěny dvě chyby, jedna spojená s uživatelským rozhraním, druhá s vizuální stránkou hry. Dále jsem také shrnul pozitiva a negativa, které participanti zmiňovali v dotazníku.

Popis nalezených chyb

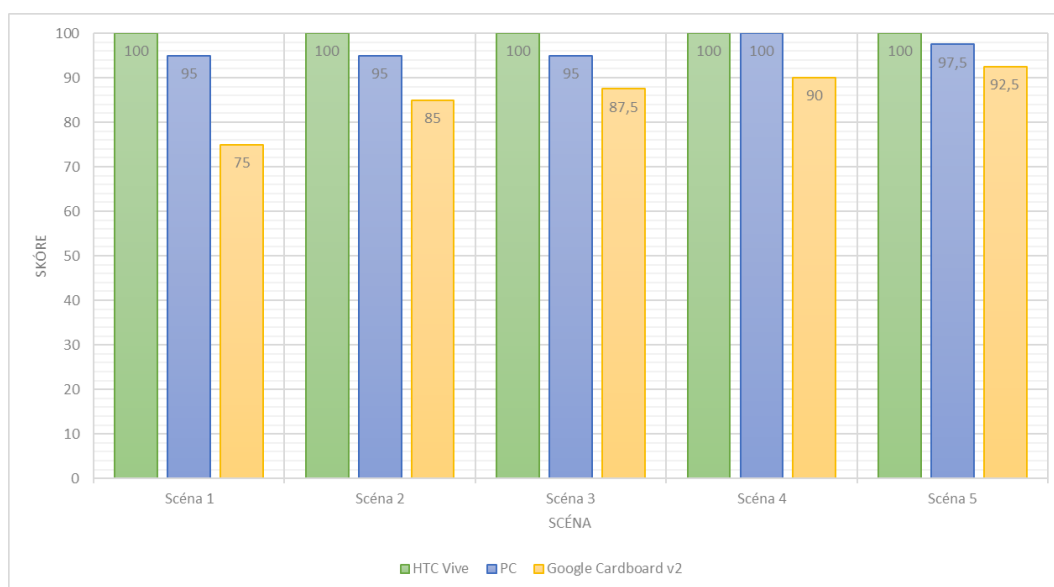
- Na zařízení Google Cardboard v2 je ve druhé a čtvrté scéně uživatelské rozhraní příliš blízko k uživateli a vzhledem k nastavené konvergenci nedojde ke spojení ukazatele, který slouží k zaměřování. Proto uživatel musí prvky zamířit do prostoru mezi čočky viděné v pravém a levém oku.
- Ve čtvrté scéně dochází k odlesku světla z reflektoru měsíčního vozítka uvnitř astronautovy helmy v případě, že je k vozítku otočen zády.

Pozitiva zjištěná při testování

- Možnost interakce s velkým počtem objektů, dokonce i s těmi, které nejsou potřeba ke splnění úkolu.
- Dobrá první zkušenost s virtuální realitou.
- Pěkné grafické zpracování.
- Zábavnost, promyšlenost a naučnost.
- Přirozená navigace po scéně.
- Elegantní řešení přibližování a oddalování objektů pro Google Cardboard v2.
- Kladné hodnocení diegetických uživatelských rozhraní.

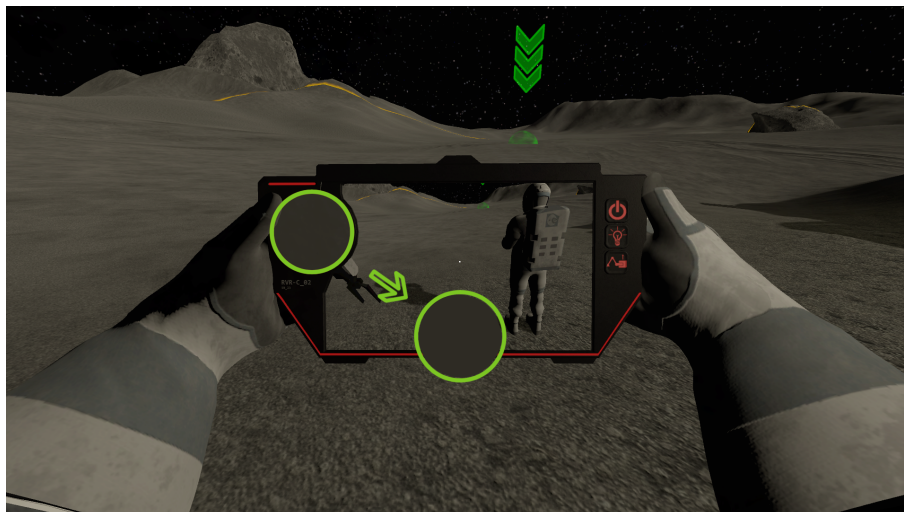
Negativa zjištěná při testování

- V pohledu pro Google Cardboard v2 se jeví uživatelské rozhraní ve druhé scéně příliš blízko.
- V pohledu pro Google Cardboard v2 participantům vadilo stále koukat nalevo, aby mohli ve čtvrté scéně ovládat měsíční vozítko.
- Google Cardboard v2 při dlouhém hraní přetěžuje deltový sval, jelikož nutí uživatele držet zařízení ve výšce očí, což nebylo participantům příjemné.
- Nižší frekvence snímků v zařízení Google Cardboard v2.



Obrázek 5.1. Výsledky hodnocení scén podle VR zařízení

Dále jsem vyhodnotil preference participantů podle jednotlivých zařízení. V grafu 5.1 je vidět hodnocení uživatelů pro jednotlivé scény. Nejlépe ze všech zařízení pro virtuální realitu dopadlo HTC Vive, následované PC a na posledním místě se umístil Google Cardboard v2. Vzhledem k tomu, že vlastně veškerá zjištěná negativa se týkala Google Cardboard v2, není překvapivé, že v hodnocení se umístil na posledním místě, a bylo by tedy vhodné se zaměřit na zkvalitnění verze pro toto zařízení. Zjištěné chyby by mohly být odstraněny další optimalizací, správným nastavením konvergence nebo úpravou uživatelského rozhraní. Ve čtvrté scéně by bylo možné uživatelské rozhraní opravit přesunutím ovládacího prvku z levé části ovladače na dolní okraj viz 5.2, což by umožnilo přímo při ovládání vidět nezkrácený obraz displeje.



Obrázek 5.2. Úprava uživatelského rozhraní ve čtvrté scéně na základě výsledků testování

■ 5.3.1 Testování na Google Cardboard v2

Oproti ostatním zařízením má pro účely testování aplikací ve virtuální realitě Google Cardboard v2 jednu velkou nevýhodu. V průběhu testování není možné sledovat, co uživatel právě dělá a s jakými problémy se v průběhu hry potýká. Na PC nebo HTC Vive je možné po celou dobu kontrolovat jednotlivé kroky uživatele, protože se zobrazují na displej počítače. To ale u Google Cardboard v2 není možné (výkonnostní problémy nebo technologické řešení), a tedy velmi záleží na komunikaci mezi participantem a moderátorem.

Kapitola 6

Závěr

Závěrem bych rád řekl, že bylo velice zajímavé seznámit se s tvorbou her pro virtuální realitu, stejně tak jako se dozvědět o plánovaném programu Evropské kosmické agentury. Ostatně i pro mě tato práce byla první zkušeností s HTC Vive. Přestože občas mělo Unity i HTC Vive své mouchy, vývoj hry byl zábava a podařilo se mi vyprodukovat funkční verzi hry pro všechna zadaná zařízení viz D. Obrázky ze hry jsou umístěny v příloze B. Navržený multiplatformní systém interakce mi umožnil snadno vyvíjet hru pro všechna tři zařízení a i oprava chyb byla o poznání jednodušší. Kvůli špatné dokumentaci a nedostatku návodů ze strany Unity se mi ze začátku práce z časových důvodů nepodařilo napojit tento interakční systém na standardní uživatelské rozhraní v Unity, a tak jsem musel některé části standardního uživatelského rozhraní v Unity přizpůsobit tím správným způsobem, aby do mého systému zapadly.

Stinnou stránkou vývoje pro více zařízení, která jsou výkonnostně hodně vzdáleny, je, že nejpomalejší zařízení omezuje ty rychlejší. Pokud nechceme dělat modely, skripty, fyziku a shadery speciálně pro pomalejší a rychlejší zařízení, musíme dospět k určitému kompromisu a celou dobu mít na paměti možnosti nejslabšího zařízení a šetřit s výpočetními zdroji. Ostatně běžná praxe ukazuje, že zdaleka ne všechny objekty je nutné mít ve scéně pro všechna zařízení, k čemuž jsem byl také nucen přistoupit. Některé náročnější skripty se mohou počítat namísto každého snímku pouze ob snímek. Snížením kvality vykreslování, antialiasingu a stínů je také možné ušetřit značné množství výpočetních prostředků.

Problém s hrami ve virtuální realitě je, dle mého názoru, v současné době především ten, že uživatelé nejsou zvyklí na standard nějakého ovládání, a tak je většina her pro virtuální realitu poměrně jednoduchá, přímočará, a tomuto problému se tak vyhýbají. V této práci jsem se snažil navrhnout co možná nejjednodušší a nejpřirozenější ovládání, abych tak předešel problémům s dlouhou učicí křivkou. Přesto ale od uživatele vyžadují více než jen pouhé zmáčknutí spouště a první okamžiky proto mohou být náročnější. To platí asi nejvíce pro zařízení Google Cardboard v2, kde bylo opravdu těžké držet návrh ovládání co nejjednodušší vzhledem k nedostatku vstupů.

Při návrhu jednotlivých uživatelských rozhraní jsem se snažil co nejvíce používat diegetické a prostové uživatelské rozhraní. Také jsem poskytl hodně odlišných zážitků při interakci a navigaci a celý návrh jsem podřizoval jednoduchosti a konzistenci napříč scénami, aby se tak uživatel v každé scéně nemusel učit vše od začátku. Nejvíce bych vyzdvihl navigaci pomocí ovladačů HTC Vive, která mi přijde skvělou metaforou a většina uživatelů na ni reagovala velmi kladně. Dále bych rád znovu zmínil scénu Barycentrické míčky 3.3.4, ve které podle mého názoru navigace vhodně dokreslila pohyb ve stavu beztíže.

Z testování vyplývá, že by bylo vhodné věnovat další čas speciálně zařízení Google Cardboard v2, protože se umístilo na posledním místě v hodnocení uživatelů a bylo zde nalezeno hned několik nedostatků. Dá se polemizovat, zda se do hodnocení uživateli 5.1 nepromítá právě dlouhá učicí křivka tohoto zařízení, protože každou další scénou se toto hodnocení zlepšovalo, mezitím co u ostatních zařízení byly výsledky poměrně vyvážené.

Kromě oprav nalezených nedostatků by tedy v navazující práci mohla proběhnout i další optimalizace pro zařízení Google Cardboard v2. Optimalizací modelů, fyziky a použitím atlasů textur 2.2.8, které by mohlo vést v Unity k dalším souvisejícím optimalizacím viz [13], by šlo jistě získat několik snímků za sekundu navíc.

Literatura

- [1] "ESERO".
"http://esero.sciencein.cz/esero" .
- [2] "Unity Fixed joint".
"https://docs.unity3d.com/Manual/class-FixedJoint.html" .
- [3] "Virtual reality users worldwide".
<https://www.statista.com/statistics/426469/active-virtual-reality-users-worldwide/>.
- [4] "Epic Games Unreal Engine 4".
"https://www.unrealengine.com" .
- [5] "Unity 5".
"https://unity3d.com" .
- [6] "Unreal Engine 4 and Unity 5 visual comparison".
"http://not-lonely.com/blog/making-of/unity-ue-comparison/" .
- [7] "User Interface design in video games".
"http://www.gamasutra.com/blogs/AnthonyStonehouse/20140227/211823/User_interface_design_in_video_games.php" .
- [8] "Minecraft texture atlas".
"http://www.9minecraft.net/albion-texture-pack/" .
- [9] "Unity Rigid body".
"https://docs.unity3d.com/Manual/class-Rigidbody.html" .
- [10] "First person shooter".
"https://en.wikipedia.org/wiki/First-person_shooter" .
- [11] "Barycentrické míčky ve vesmíru".
"https://www.youtube.com/watch?v=ZwBg4d7Wx1s" .
- [12] "Unity Character controller".
"https://docs.unity3d.com/Manual/class-CharacterController.html" .
- [13] "Dynamic batching".
"https://docs.unity3d.com/Manual/DrawCallBatching.html" .
- [14] "VR User interface".
"https://unity3d.com/learn/tutorials/topics/virtual-reality/user-interfaces-vr?playlist=22946" .
- [15] ANDREWS, M. "Gamasutra".
"http://www.gamasutra.com/view/feature/132674/game_ui_discoveries_what_players_.php" .
- [16] CORPORATION, HTC. "Manual HTC Vive".
"https://www.vive.com/eu/support/category_howto/tips-for-setting-up-the-base-stations.html" .

- [17] CORPORATION, Valve. "Steam".
"https://en.wikipedia.org/wiki/First-person_shooter" .
- [18] DEWHIRST, D. "Reddit".
"https://www.reddit.com/r/Vive/comments/4hafs1/vive_movement_system_based_off_ready_player_one/" .
- [19] EUROPEAN SPACE AGENCY. "ESA vesmírná základna".
"http://www.esa.int/Our_Activities/Space_Engineering_Technology/Building_a_lunar_base_with_3D_printing/" .
- [20] KAREL, L. a Ž. ALEŠ. "Chorobné znaky a príznaky".
- [21] MACHKOVECH, S. "arsTECHNICA".
<http://arstechnica.com/gaming/2016/04/for-vr-walking-forget-treadmills-just-stick-a-vive-wand-down-your-pants/>.
- [22] MAURICEHENDRIX, Per Backlund a. "Educational Games - Are They Worth the Effort?".
- [23] SLANT. "Engines for virtual reality".
"<https://www.slant.co/topics/2202/game-engines-for-virtual-reality-development>" .
- [24] VIRTUAL REALITY SOCIETY. "Virtual reality education".
"<https://www.vrs.org.uk/virtual-reality-education/advantages.html>" .
- [25] ŽÁRA, J., B. BENEŠ, J. SOCHOR a P. FELKEL. "Moderní počítačová grafika".

Příloha A

Dotazník

Vzdělávací hra s vesmírnou tematikou

V této sekci se prosím soustředte na obecné principy použité ve hře. Jako například ovládání, pohyb po scéně, způsob interakce, uživatelské rozhraní (texty umístěné v prostoru, přehrávač, ovladač ...)

*Povinné pole

1. Vaše pohlaví *

- Muž
 Žena

2. Váš věk *

- 10 až 15
 15 až 20
 20 a více

3. Vaše zkušenosti *

	Ano	Ne
Máte již nějaké zkušenosti s testováním?	<input type="radio"/>	<input type="radio"/>
Máte zkušenosti s počítačovými hrami?	<input type="radio"/>	<input type="radio"/>
Máte zkušenosti s virtuální realitou?	<input type="radio"/>	<input type="radio"/>

4. Líbila se Vám hra? *

	0	1	2	3	4	5	6	7	8	9	10	
Nelíbila	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Líbila

5. Napište, proč se Vám hra líbila či nelíbila *

6. Co konkrétně se Vám líbilo nejvíce?

7. Co konkrétně se Vám naopak nelíbilo, působilo na Vás nepříjemně, nebo byste změnili?

8. Dozvěděli jste se něco nového, zajímavého nebo užitečného?

Stavba rakety Ariane 5

9. Líbila se Vám hra? *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerná	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborná

10. Ohodnoťte herní zážitek na zařízení HTC Vive *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

11. Ohodnoťte herní zážitek na PC *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

12. Ohodnoťte herní zážitek na zařízení Google Cardboard v2 *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

13. Bylo uživatelské rozhraní přehledné a intuitivní? *

- Ano
 Ne

14. Byli jste schopni jednoduše splnit cíl hry? *

- Ano
 Ne

15. Pokud jste v některé z předchozích otázek odpověděli negativně (Ne, nebo skóre menší než 4), popište proč, případně co byste změnili.

16. Pokud jste v této hře pociťovali nevolnost, prosím jednoduše popište z čeho.

Start rakety

17. Líbila se Vám hra? *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerná	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborná

18. Ohodnotte herní zážitek na zařízení HTC Vive *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

19. Ohodnotte herní zážitek na PC *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

20. Ohodnoťte herní zážitek na zařízení Google Cardboard v2 *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

21. Bylo uživatelské rozhraní přehledné a intuitivní? *

Ano
 Ne

22. Byli jste schopni jednoduše splnit cíl hry? *

Ano
 Ne

23. Pokud jste v některé z předchozích otázek odpověděli negativně (Ne, nebo skóre menší než 4), popište proč, případně co byste změnili.

24. Pokud jste v této hře pociťovali nevolnost, prosím jednoduše popište z čeho.

Barycentrické míčky ve stavu beztlíže

25. Líbila se Vám hra? *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerná	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborná

26. Ohodnoťte herní zážitek na zařízení HTC Vive *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

27. Ohodnoťte herní zážitek na PC *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

28. Ohodnoťte herní zážitek na zařízení Google Cardboard v2 *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

29. Bylo uživatelské rozhraní přehledné a intuitivní? *

- Ano
 Ne

30. Byli jste schopni jednoduše splnit cíl hry? *

- Ano
 Ne

31. Pokud jste v některé z předchozích otázek odpověděli negativně (Ne, nebo skóre menší než 4), popište proč, případně co byste změnili.

32. Pokud jste v této hře pocítovali nevolnost, prosím jednoduše popište z čeho.

Sběr vzorků měsíčním vozítkem

33. Líbila se Vám hra? *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerná	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborná

34. Ohodnoťte herní zážitek na zařízení HTC Vive *

0 1 2 3 4 5 6 7 8 9 10

Příšerný Výborný

35. Ohodnoťte herní zážitek na PC *

0 1 2 3 4 5 6 7 8 9 10

Příšerný Výborný

36. Ohodnoťte herní zážitek na zařízení Google Cardboard v2 *

0 1 2 3 4 5 6 7 8 9 10

Příšerný Výborný

37. Bylo uživatelské rozhraní přehledné a intuitivní? *

Ano
 Ne

38. Byli jste schopni jednoduše splnit cíl hry? *

Ano
 Ne

39. Pokud jste v některé z předchozích otázek odpověděli negativně (Ne, nebo skóre menší než 4), popište proč, případně co byste změnili.

40. Pokud jste v této hře pociťovali nevolnost, prosím jednoduše popište z čeho.

Testování vzorků na měsíční základně

41. Líbila se Vám hra? *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerná	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborná

42. Ohodnoťte herní zážitek na zařízení HTC Vive *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

43. Ohodnoťte herní zážitek na PC *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

44. Ohodnoťte herní zážitek na zařízení Google Cardboard v2 *

	0	1	2	3	4	5	6	7	8	9	10	
Příšerný	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Výborný

45. Bylo uživatelské rozhraní přehledné a intuitivní? *

Ano

Ne

46. Byli jste schopni jednoduše splnit cíl hry? *

Ano

Ne

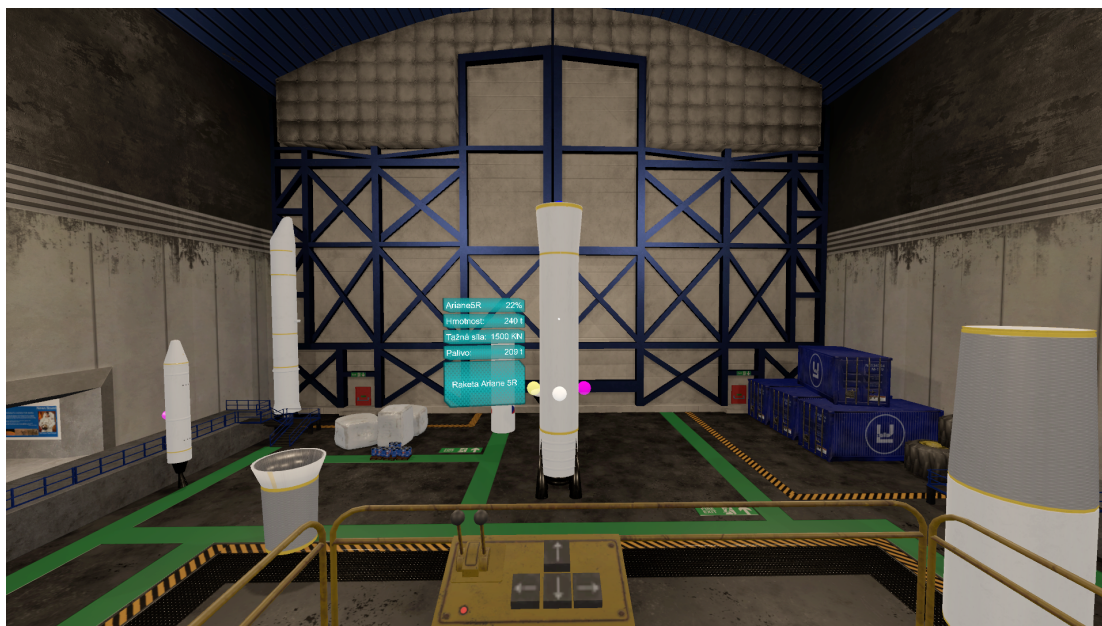
47. Pokud jste v některé z předchozích otázek odpověděli negativně(Ne, nebo skóre menší než 4), popište proč, případně co byste změnili.

48. Pokud jste v této hře pociťovali nevolnost, prosím jednoduše popište z čeho.

Powered by
 Google Forms

Příloha B

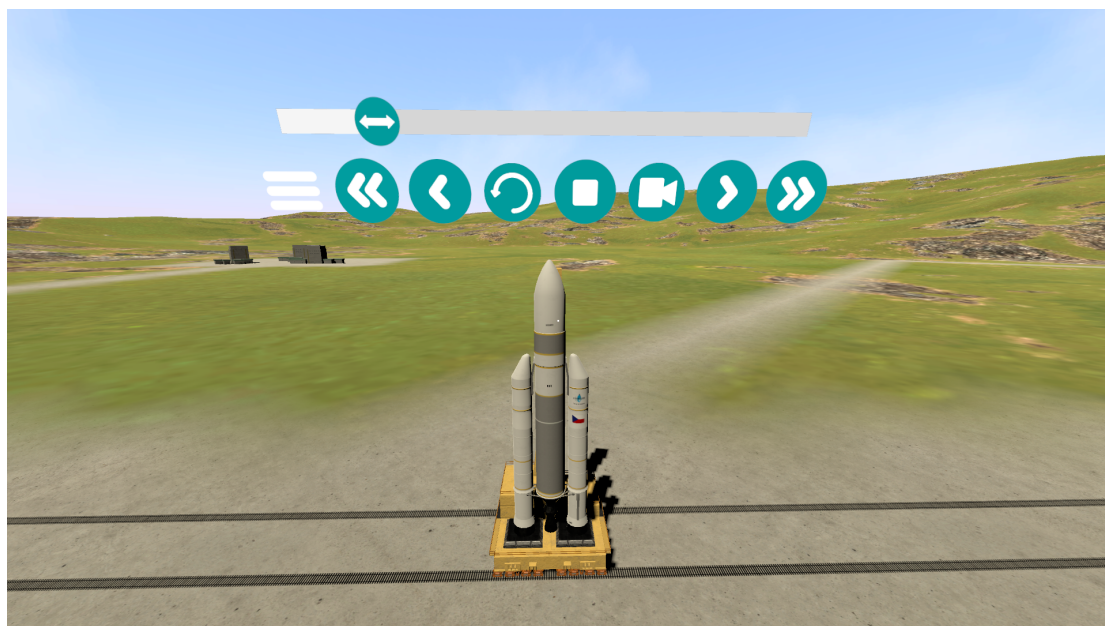
Obrázky ze hry



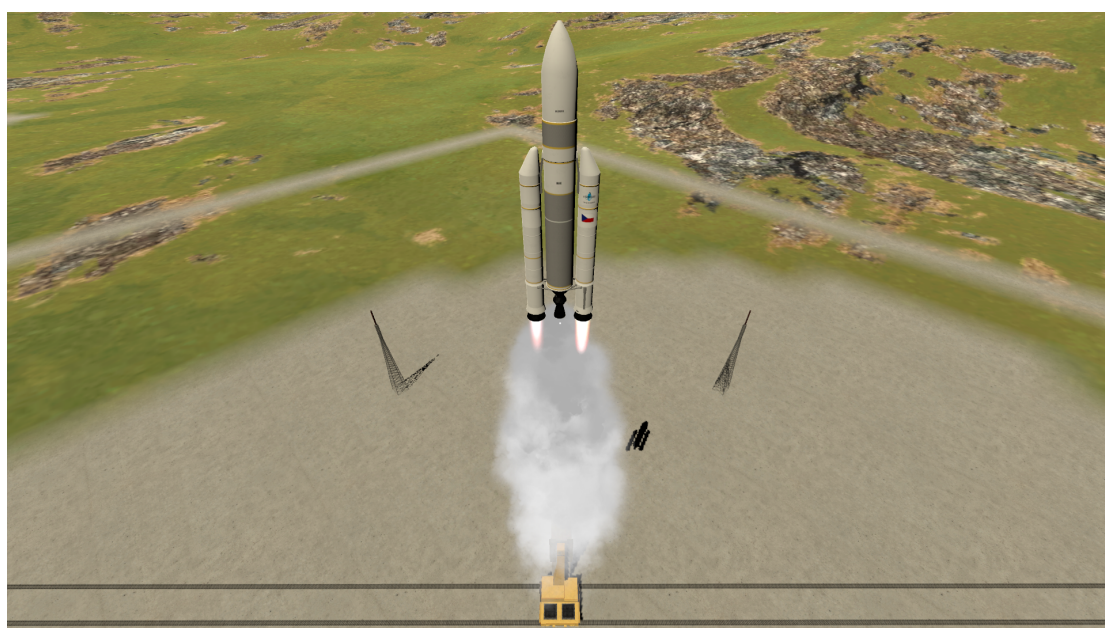
Obrázek B.1. Stavba rakety Ariane 5R.



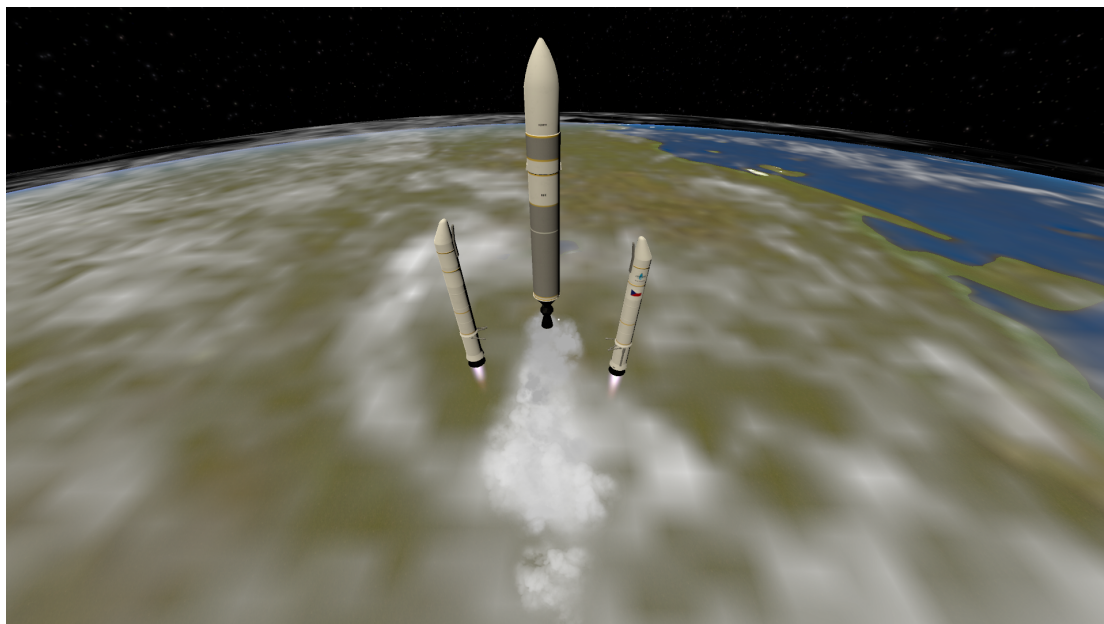
Obrázek B.2. Kompletní raketa Ariane 5P.



Obrázek B.3. Stěhování rakety Ariane 5P na startovací rampu.



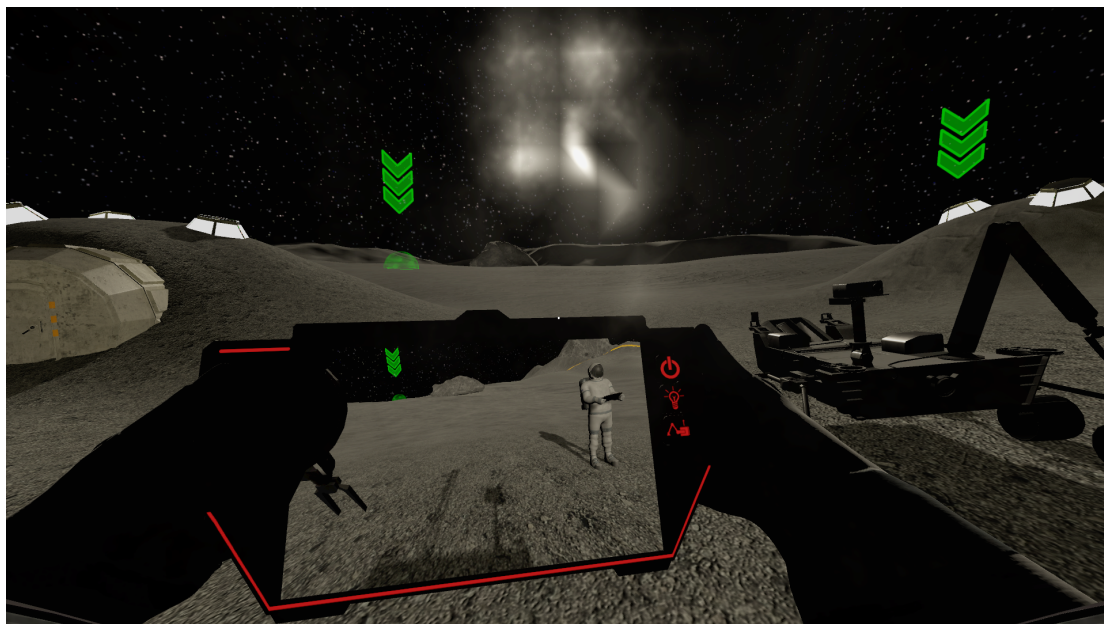
Obrázek B.4. Start rakety Ariane 5P.



Obrázek B.5. Raketa Ariane 5P odhazuje pomocné rakety.



Obrázek B.6. Simulovaný systém objektu v nulové gravitaci.



Obrázek B.7. Astronaut s dálkovým ovládáním na povrchu Měsíce. Odlesk na helmě způsobený sluníčkem.



Obrázek B.8. Test hornin na přítomnost vápníku.



Obrázek B.9. Test zrnitosti vzorků hornin odebraných na povrchu Měsíce.



Obrázek B.10. Zobrazení vzorku horniny pod mikroskopem.



Příloha C
Zadání

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Ondřej Okluský

Studijní program: Otevřená informatika
Obor: Počítačová grafika a interakce

Název tématu: Vzdělávací hry ve virtuální realitě s vesmírnou tematikou

Pokyny pro vypracování:

Navrhněte a implementujte vzdělávací hru s vesmírnou tematikou ve virtuální realitě. Hru koncipujte tak, aby ji bylo možné hrát na zařízeních PC, stereo stěně ve VRIlabu, Google Cardboard v2 a HTC Vive.

Hra bude obsahovat celkem pět scén, v rámci kterých navrhněte a implementujte různé navigační a interakční techniky vhodné pro specifická zařízení. Navržené techniky otestujte s reálnými uživateli a zhodnoťte jejich vhodnost pro dané zařízení. Tématickou náplň jednotlivých scén konzultujte s vedoucím práce. 3D modely dodá vedoucí práce.

Seznam odborné literatury:

- [1] The UX of VR: <http://www.uxofvr.com>
- [2] Unity tutorial - User Interfaces for VR: <https://unity3d.com/learn/tutorials/topics/virtual-reality/user-interfaces-vr>
- [3] The VR Book: Human-Centered Design for Virtual Reality. Jason Jerald. Morgan & Claypool Publishers (October 16, 2015). ISBN: 1970001127.

Vedoucí: Ing. David Sedláček, Ph.D.

Platnost zadání: do konce zimního semestru 2018/2019



V Praze dne 3.4.2017

Příloha D

Obsah přiloženého datového nosiče

Na přiloženém datovém nosiči je umístěno několik souborů, které vznikly v průběhu této práce. Spustitelné soubory pro zařízení specifikovaná v zadání, zdrojové kódy, obrázky a text této práce.

Přiložený datový nosič obsahuje tyto soubory a složky:

- **thesis.pdf** – text této práce
- **questionnaire.pdf** – dotazník k uživatelskému testování
- **pic** – obsahuje použité obrázky
- **exe** – obsahuje spustitelné soubory
- **src** – obsahuje Unity projekt, zdrojové kódy, textury a modely