



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**EFEKTIVNÍ VYUŽITÍ OMEZENÉHO
HERNÍHO PROSTORU**

EFFECTIVE USE OF LIMITED GAME SPACE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ BRESTIČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL NAJMAN

BRNO 2018

Abstrakt

Zařízení pro virtuální realitu (VR), která jsou již dnes dostupná běžným spotřebitelům, umožňují pohyb uživatele po virtuální scéně na základě sledování jeho fyzického pohybu ve skutečném světě. S použitím takové metody jako hlavního či jediného způsobu pohybu hráče ve VR se však poji určitá úskalí, a naprostá většina VR aplikací je proto stále založena na jiných metodách, které ale dostupný prostor ani uvedené vlastnosti zařízení příliš nevyužívají. Hlavním účelem této práce je navrhnout a předvést možný způsob implementace vedoucí k využití dostupného fyzického prostoru, a pokusit se následně vyhodnotit, zda aplikace přímo založená na popsaném způsobu pohybu může uživatele lépe zaujmout a nabídnout jim lepší zážitek, a zda tedy má význam takovou aplikaci vyvíjet i přes existenci zmíněných nástrah.

Abstract

Current generation of consumer electronics products for virtual reality (VR) allows to track user motion in physical play area which can be converted into a corresponding change of their position in a virtual environment. There are, however, some potential pitfalls with using this approach as the main method of user motion in the virtual space. Therefore, vast majority of contemporary applications is based on different methods that do not take advantage of the available play area and the possibilities of the input devices in their entirety. The main goal of this thesis is to propose and demonstrate a feasible way of implementation allowing the use of available physical space and to try to evaluate whether basing the user motion in an application on the described approach can result in an increase of user interest, attract more users and offer them better experience, and whether it therefore can be profitable in spite of all the related disadvantages.

Klíčová slova

HCI, uživatelské rozhraní, virtuální realita, room-scale VR, techniky pohybu

Keywords

HCI, user interface, virtual reality, room-scale VR, travel techniques

Citace

BRESTIČ, Tomáš. *Efektivní využití omezeného herního prostoru*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Pavel Najman.

Efektivní využití omezeného herního prostoru

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Pavla Najmana.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Brestič
23. května 2018

Poděkování

Děkuji panu Ing. Pavlu Najmanovi za trpělivé vedení, odborné rady, cenné zkušenosti a věcné připomínky, které mi poskytl při tvorbě této práce.

© Tomáš Brestič, 2018

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	2
2 Stávající situace.....	3
2.1 Koncept virtuální reality.....	3
2.2 Zařízení pro virtuální realitu.....	4
2.3 Pohyb ve virtuálním prostředí.....	10
3 Analýza a obecný návrh.....	14
3.1 Konceptuální návrh metody pohybu.....	14
3.2 Návrh testovacího prostředí.....	18
4 Implementace.....	21
4.1 Použité nástroje.....	21
4.2 Systém překrývajících se místností.....	24
4.3 Implementované testovací prostředí.....	36
5 Testování.....	41
5.1 Průběh testování.....	41
5.2 Vyhodnocení testování.....	43
5.3 Možnosti navazujícího testování.....	46
6 Závěr.....	48
Literatura.....	49

1 Úvod

Mezi hlavní výhody a nejčastěji vyzdvihované aspekty virtuální reality (*virtual reality*¹, zkr. VR; viz kapitola 2.1) patří schopnost pohltit uživatele – vytvoření silné iluze vlastní přítomnosti v neexistujícím prostředí. Současná komerčně nabízená zařízení (viz kapitola 2.2) již často umožňují také přímou interakci s objekty přítomnými ve virtuální scéně pomocí ovladačů – uchopování a přesné natáčení i umístování předmětů v třidimenzionálním prostoru přímo navázané na pohyby rukou uživatele, přičemž takové prvky ovládání bývají označovány jako přirozené a intuitivní, neboť odpovídají způsobu, jakým lidé běžně interagují s objekty ve skutečném světě (viz kapitola 2.3).

Taková zařízení též zpřístupňují široké veřejnosti možnost sledování pozice uživatele v rámci reálné místnosti – tzv. *room-scale VR*. Sledovaný fyzický pohyb osoby pak může být převeden na pohyb ve virtuální scéně. Popsaná technika pohybu pak je jen dalším krokem ve směru zvyšování intuitivnosti a jistě musí být taktéž velmi přirozená, musí pomáhat vtahovat uživatele do fiktivního světa a prohlubovat jeho zážitek. Proč tedy naprostá většina ze vznikajících aplikací pro VR nevyužívá plně potenciál moderních zařízení a nezakládá přímo pohyb v rámci svého prostředí na této metodě? Proč naopak spoléhají na poměrně umělé a nerealistické techniky, které relativně narušují nastolenou iluzi „alternativní reality“?

Uvedené použití nejspíše nebude tak přímočaře jednoduché a bezproblémové, jak by se na první pohled mohlo zdát. V rámci této práce budou popsána nejen jeho možná úskalí a nástrahy, ale i předpokládané výhody (viz kapitola 3). Dále bude navržen způsob a podmínky, za jakých lze využít sledovaný fyzický pohyb ve skutečné místnosti jako hlavní způsob pohybu ve virtuálním prostoru, a s využitím vhodných nástrojů (viz kapitola 4.1) bude vytvořena ukázková scéna, v níž bude tento přístup aplikován (viz kapitola 4.2). Následně bude testován a porovnán se současnou nejrozšířenější technikou pohybu (viz kapitola 5). Hlavními otázkami, které si přitom práce klade, je to, zda je tento princip skutečně použitelný, oč je jeho reálná implementace náročnější oproti běžným řešením, jak citelný je skutečný dopad souvisejících omezení a zda VR aplikace rozsáhle využívající fyzický pohyb či dokonce na něm založená může být uživateli lépe přijímána a hodnocena jako subjektivně „lepší“ oproti obdobné aplikaci, která jej nevyužívá. To vše dohromady by mělo pomoci rozhodnout, zda v případě, kdy si návrhář či vývojář aplikace může mezi uvedenými přístupy vybrat, pro něj může být výhodné opustit tradiční umělé prostředky a svázat pohyb ve scéně s fyzickým pohybem uživatele; potažmo má-li vůbec *room-scale VR* v současné podobě nějaký praktický význam (viz kapitola 6).

Některé části textu – zejména kapitoly 2 a 3 popisující stávající situaci v oblasti VR, koncepci uvedené techniky pohybu a obecnou podobu zamýšleného testování, stejně jako část kapitoly 4, která představuje knihovnu VRTK – většinou přejímají skutečnosti zjištěné a popsané v rámci semestrálního projektu souvisejícího s touto prací.

¹ Kurzívou jsou v textu označeny zavedené anglické termíny sloužící pro snazší vyhledávání a orientaci čtenáře v související cizojazyčné literatuře či dokumentaci.

2 Stávající situace

Následující část textu vymezuje oblast virtuální reality a pokračuje popisem existujících principů využívaných pro ovládání a zprostředkování pohybu uživatele po virtuální scéně, a to jak z hlediska hardwarových zařízení, tak z pohledu konceptů, metafor a technik uživatelského rozhraní; včetně konkrétních příkladů jejich použití v současné praxi.

2.1 Koncept virtuální reality

Snaha o vytvoření iluze pobytu v nereálném prostředí, která by cele pohlcovala uživatele a co nejdokonaleji klamala jeho smysly, je dosti stará a lze ji v jistém smyslu stopovat do historie až k panoramatickým nástěnným malbám či ještě dále, vždy s využitím technologie dostupné v dané době [1]. Snaha o funkční elektronické řešení probíhá v určitých vlnách již zhruba od poloviny minulého století, ale teprve v několika posledních letech začíná být technologie umožňující uspokojivou úroveň věrnosti a interaktivity takové iluze reálně dostupná běžným spotřebitelům [2]. Následující část práce má za úkol stručně shrnout problematiku virtuální reality, formulovat její vlastnosti a představit některé z možných oblastí, v nichž se v současné době dá uplatnit.

2.1.1 Definice pojmu virtuální reality

V odborné literatuře se vyskytuje řada různých definic pojmu „virtuální realita“. Některé publikace tento pojem vymezují pomocí technologií, které jsou k její realizaci použity, jiné se snaží obecněji pojmenovat vlastnosti virtuální reality, vycházejíce z jejích funkcí a cílů. Jednotlivé definice se v jádru obvykle významově překrývají, často se však liší v ohraničení tohoto pojmu – v určení mezi toho, co ještě virtuální realita je, a co už není. V nejširším smyslu mají aplikace virtuální reality společné tyto body [3]:

- řízené chování prostředí
- přítomnost organismu (uživatele)
- umělá stimulace jeho smyslů
- vyvolání pocitu fyzické přítomnosti ve virtuálním světě

Řízeným chováním prostředí je myšleno to, že se toto chová předem navrženým způsobem, jenž je v moci jeho návrháře/vývojáře. Uživatelem nemusí být obecně člověk, systémy virtuální reality byly aplikovány u několika výzkumných experimentů i na hlodavce, hmyz, opice. Působením na jeden či více smyslových orgánů jsou reálné vjemy nahrazeny umělými. Skrze toto „ošálení smyslů“ by si uživatel ideálně neměl být vědom toho, že se nachází v neexistujícím či pozměněném prostředí, měl by být do něj zcela vtažen. Do výše uvedeného vymezení v jistém smyslu pasuje např. i sledování filmu v kině, často se tedy jako další aspekt uvádí možnost přímého ovlivňování prostředí uživatelem – interaktivita [2]; případně se podle této charakteristiky dělí VR systémy na tzv. „s otevřenou smyčkou“ a „s uzavřenou smyčkou“ (*open-loop* a *closed-loop*) [3].

Někdy se virtuální realita vymezuje vůči rozšířené realitě (*augmented reality*, zkr. AR), přičemž rozlišovací vlastností je to, že snahou VR je vytvářet kompletní iluzi umělého, neskutečného prostředí a zcela nahradit podněty z reálného prostředí umělými, zatímco AR používá vjemy

skutečného světa doplněné o virtuální prvky. Oblastí smíšené reality (*mixed reality*, zkr. MR) je pak celé spojité spektrum typů zprostředkovaných prostředí od čistě virtuálních po zcela skutečné. [2]

2.1.2 Použití virtuální reality

Patrně nejvíce exponovaným aktuálním případem užití virtuální reality jsou její aplikace v zábavním průmyslu, tedy speciální filmy a zejména hry. VR však lze v praxi využít i v řadě jiných oblastí. Mezi takové příklady patří simulace tréninkových podmínek specifických schopností v prostředí, jejichž ne-virtuální provedení by bylo velmi nákladné, nebezpečné či jinak problematické, např. bojový vojenský výcvik či cvičné provádění lékařských operací. Podobná řízená simulace prostředí může být výhodná i pro provádění kognitivních experimentů nebo terapií psychologických potíží. Existují též pokusy o použití VR headsetů k nápravě či prevenci zrakových vad, jako je tupozrakost. VR je možné použít i pro interaktivní vizualizaci rozličných produktů, čehož lze využít při jejich návrhu nebo třeba k jejich prezentaci klientům před vlastní realizací, např. v podobě virtuálních prohlídek budov. [2][3]

2.2 Zařízení pro virtuální realitu

Vstupní a výstupní hardwarová zařízení jsou nedílnou součástí každé interakce člověka s počítačem (*human-computer interaction*, zkr. HCI), jako je ta nutná pro VR. Jejich úkolem je převádět či překládat uživatelské akce do formy zpracovatelné strojem (vstupní zařízení) a naopak transformovat informace o vnitřním stavu aplikace do podoby srozumitelné člověku, působit na jeho smysly (výstupní zařízení). [2]

2.2.1 Vstupní zařízení

Prostřednictvím vstupních zařízení může uživatel ovlivňovat virtuální prostředí nebo jeho zobrazení. Na možnostech uplatněných vstupních zařízení pak do jisté míry závisí vhodnost použití konkrétních interakčních technik, aplikace by tedy měla být navrhována s jejich znalostí a s ohledem na ně.

Ve VR aplikacích je možno použít „tradiční“ vstupní zařízení, jako je klávesnice či herní ovladač (*gamepad*). Kromě toho ale může být vstupem vlastní pohyb uživatele. Někjaká forma takového sledování může být dokonce nutná ke správnému zobrazení scény v závislosti na použitém zobrazovacím zařízení (viz níže), např. u VR headsetů. Přesnost takového sledování společně s dostatečně nízkým celkovým zpožděním (*latency*) systému (tj. doba od vykonání akce k jejímu projevu v uživateli vnímané simulaci) je pak zásadní pro zamezení nepříjemných pocitů uživatele. V případě, kdy dochází k citelným rozporům v informacích poskytovaných zrakovým a rovnovážným ústrojím uživatele, u něj totiž často dochází k vyvolání pocitů nevolnosti (*VR sickness*), citlivost na tento jev je však značně subjektivní [4]. Typickým parametrem zařízení používaných pro spojitě sledování pohybu je počet stupňů volnosti (*degrees of freedom*, zkr. DOF), který v tomto případě označuje, kolik nezávislých směrů pohybu a otáčení je rozlišováno. Např. kolečko běžné počítačové myši má obvykle 1 DOF (rotace kolem jedné osy), myš samotná 2 DOF (posun ve dvou osách), zatímco u zařízení pro sledování objektu ve 3D prostoru (*3D tracker*; viz níže) je obvyklých 6 DOF (posun i rotace ve třech osách). Další důležitou charakteristikou je rozsah snímání, jenž omezuje velikost sledované oblasti.

Sledování ve 3D prostoru (*3D spatial tracking*) může být založeno na mnoha různých fyzikálních principech, jako jsou:

- mechanické snímání
- magnetické snímání
- akustické (zejména ultrazvukové) snímání
- radarové snímání
- bioelektrické snímání
- inerciální snímání
- optické snímání
- hybridní snímání

Dnešní spotřebitelská řešení nejčastěji používají hybridní přístup primárně založený na optickém snímání, neboť zařízení tohoto typu jsou relativně lehká a cenově dostupná (na rozdíl od těch využívajících mechanické snímání), dostatečně přesná v požadovaném prostorovém rozsahu (na rozdíl od akustických, která mají typicky příliš malý rozsah, a klasických radarových, která naopak nemají dostatečnou přesnost), snadno použitelná (na rozdíl od bioelektrických) a dostatečně odolná proti rušení (na rozdíl od magnetického snímání). Rozlišujeme přitom přístupy nazývané „*outside-in*“, kdy senzory jsou rozestaveny na pevných místech fyzického prostředí, a „*inside-out*“, které využívá senzory umístěné na sledovaném objektu. Další možné dělení je na přístupy využívající speciální značky (*marker-based*) a nevyužívající značky (*markerless*). *Marker-based* techniky počítají s přítomností předem definovaného vzoru (*marker, pattern*), ať už jde o vytištěný obrázek nebo třeba vyzařování světla v určitém tvaru. *Markerless* využívají detekci a sledování význačných bodů v obraze. Optické snímání bývá za účelem zvýšení robustnosti doplněno o snímání inerciální založené na gyroskopech a akcelerometrech, které dává rychlejší (byť obvykle méně přesné) odhady změny polohy a pokrývá jeho krátkodobé výpadky způsobené přerušением linie přímé viditelnosti (*line of sight*). Samostatné inerciální snímání není vhodné, neboť výstup čidel je pro získání výsledné pozice nutné integrovat, což znamená možnost pouze relativního měření a náchylnost k rychlé kumulaci chyby s časem. Konkrétní příklady existujících zařízení pracujících na uvedeném principu jsou popsány v kapitole 2.2.3.

Sledovány jsou obvykle minimálně pozice a natočení hlavy (kvůli zobrazování s headsetem – zmíněno výše), případně rukou (neboť ty jsou přirozeným základním prostředkem pro interakci s objekty scény). V závislosti na povaze aplikace může být výhodné sledovat i trup, nohy nebo jednotlivé prsty, přičemž existují konkrétní technická řešení (nejčastěji mechanická a optická), která to umožňují. Samostatnou problematiku pak tvoří sledování směru pohledu. Kromě uvedených přístupů nebo v kombinaci s nimi lze použít jako vstup aplikace též ne-prostorové principy, jako např. hlasové ovládání či snímání mozkové aktivity. [2]

2.2.2 Výstupní zařízení

Jak bylo zmíněno v kapitole 2.1.1, základní vlastností virtuální reality je schopnost navozovat uživateli pocit přítomnosti v jiném prostředí, než v jakém se reálně nachází, prostřednictvím působení na jeho smyslová ústrojí. Právě k tomu slouží výstupní zařízení. Nejčastěji jde o řízené vyvolávání zrakových, sluchových nebo hmatových vjemů, existují však i méně rozšířená řešení poskytující stimuly čichové a chuťové [2], případně přímo ovlivňující rovnovážně ústrojí člověka [3].

Zobrazovací zařízení

Zobrazovací zařízení zprostředkovávají vizuální vjemy virtuální scény a jsou tedy v naprosté většině případů převážným zdrojem informací pro uživatele, proto budou jejich hlavní parametry popsány poněkud detailněji. V současnosti nejpopulárnější podobou zobrazovacího zařízení pro VR je tzv. *VR headset* – „brýle“ tvořené displejem připevněným na hlavě uživatele (*head-mounted display*, zkr. HMD, případně *head-worn display*, zkr. HWD) v kombinaci s vhodně zvolenými čočkami. Výhodou tohoto přístupu je zejména fakt, že uživatel má displej stále před sebou a zobrazení není příliš ovlivněno externími vlivy, jako je vnější osvětlení. Existují však i jiné přístupy k zobrazování virtuální scény, např. s využitím standardních monitorů či televizí rozmístěných kolem uživatele (*surround-screen displays*), přední i zadní projekce na rovné i zakřivené promítací plochy a podobně.

Mezi charakteristiky zobrazovacích zařízení, které mají zásadní vliv na kvalitu vnímání virtuálního prostředí, patří [2]:

- obnovovací frekvence (*refresh rate*)
- zorné pole (*field of view*)
- rozlišení (*resolution*)
- ergonomie (*ergonomics*)
- mechanismus vytvoření hloubkového vjemu (*depth cues*)

Zatímco zbytek uvedených parametrů typicky ovlivňuje jen kvalitu a věrnost zážitku, případně pohodlí při použití zařízení, zajištění vhodných hodnot první z nich je nutné pro zabránění fyzické nevolnosti uživatele. Dostatečná **obnovovací frekvence** (v kombinaci s vykreslovací frekvencí, která je dána grafickou náročností aplikace a výkonem počítače) podmiňuje plynulost vykreslení scény, která je nutná k zamezení nepříjemných pocitů při vnímání, pokud se obraz pohybuje, zejména pak při jeho pohybu v reakci na pohyb uživatele. Čas mezi obnoveními se totiž podílí na celkovém zpoždění zobrazení. Obnovovací frekvence se obvykle uvádí v hertzech (zkr. Hz). Latence přenosu a vlastního zobrazení ve výstupním zařízení typicky netvoří problém, může se však teoreticky projevat např. u bezdrátových řešení.

Zorné pole udává prostorový úhel, který displej zabírá v pohledu uživatele. Čím větší je tento úhel, tím větší využití zorného pole člověka (to se udává jako zhruba 180°) dovoluje, což podporuje vyšší úroveň vtážení uživatele do virtuální scény (imerzi) a umožňuje uplatnění větší části jeho periferního vidění pro získávání informací, čímž je zvýšen jeho celkový přehled o dění ve virtuálním prostoru, který jej obklopuje.

Prostorové **rozlišení** je dáno prostorovou frekvencí elementů obrazu a tedy úzce souvisí s velikostí pixelů. Taková veličina se nejčastěji udává v jednotkách dpi (*dots per inch*). Vnímané rozlišení pak (podobně jako zorný úhel) závisí navíc na vzdálenosti pozorovatele od displeje. Častěji se však jako hodnota rozlišení uvádí počet pixelů, v případě typického pravidelného obdélníkového displeje typicky ve formě počtu sloupců a počtu řádků, z čehož lze při shodné velikosti všech stran pixelu snadno vyjádřit poměr stran. Vyšší vnímané rozlišení obecně podporuje lepší přesvědčivost zobrazení virtuální scény.

Pojem **ergonomie** se pojí s pohodlím uživatele při používání zařízení a její vliv je zásadní zejména v případech, jako je VR headset, kdy uživatel nosí zařízení na hlavě. Takové zařízení by mělo být např. co nejlehčí a jeho hmotnost by měla být ideálně rovnoměrně distribuována, aby se předešlo vyvolání bolestí krčního svalstva při dlouhodobém používání. Na druhou stranu, ergonomii by bylo možné zvýšit také odstraněním kabelu vedoucího do headsetu. Bezdrátové řešení však

vyžaduje umístění vysílače, přijímače a zejména napájení v podobě akumulátoru, což zase zvyšuje celkovou hmotnost zařízení, které by uživatel nosil.

Vjem hloubky je u současných 3D zobrazovacích zařízení nejčastěji vytvořen primárně (kromě monoskopických efektů, jako je pohybová paralaxa, perspektiva a překrývání; zejména pro blízké objekty) vzájemným posunem obrazu objektu pro pravé a levé oko v závislosti na jeho vzdálenosti od kamery ve scéně. Směr pohledu obou očí je fixován na objekt, jejich sbíhavost je pak větší pro bližší objekty a menší pro vzdálené. Nicméně čočky očí jsou stále akomodovány na vzdálenost displeje, aby obraz viděly ostře, u VR headsetů tedy vždy na blízko a v případě projekce spíše do dálky. Vzniká tím určitý rozpor v postavení zrakového aparátu a tedy informací, z nichž mozek usuzuje znalosti o prostorovém uspořádání sledované scény (*accomodation-vergence mismatch*). Citlivost na tento jev je patrně dosti subjektivní, čímž lze vysvětlit to, že některým osobám je použití takovýchto zařízení nepříjemné, a také to, že po delším pobytu ve VR někteří uživatelé hlásí únavu očí a dočasně „divné“, změněné vidění [5]. Je možné odstranit jedinečným použitím volumetrického, tzv. „skutečného 3D zobrazení“ (*true 3D display*); zařízení, která by jej poskytovala, jsou však v současnosti stále v rané fázi vývoje.

2.2.3 Současné reálné příklady

V posledních letech se objevuje velké množství různých technických řešení hardwaru pro VR. Pro cíle této práce jsou však relevantní zejména ty, které jsou rozšířené mezi spotřebiteli (tedy běžně k dostání na trhu a relativně cenově dostupné) a umožňují *room-scale VR*. Této oblasti v době psaní tohoto textu jasně dominují zařízení HTC Vive a Oculus Rift (s ovladači Touch). Níže jsou popsány jejich základní parametry.

HTC Vive

Headset HTC Vive (viz Obrázek 1) obsahuje displej (parametry viz Tabulka 1), čočky s nastavitelnou roztečí, mikrofon, konektor pro stereo sluchátka (ta jsou součástí balení) a kameru, která uživateli umožňuje zobrazení jeho reálného okolí bez nutnosti sejmutí headsetu z hlavy, což je užitečné zejména k prevenci či řešení zamotání se do kabelu.

Typ displeje	OLED
Rozlišení	2160×1200 px
Obnovovací frekvence	90 Hz
Zorné pole	110°

Tabulka 1: Parametry displeje pro HTC Vive

Každý ze dvou identických bezdrátových ovladačů disponuje tlačítkem sevření dlaně (*grip*), tlačítkem pod ukazováčkem (*trigger*), dotykovou plochou (*touchpad*) a dvěma tlačítky typicky používanými pro vstup do menu. Ovladače navíc nabízí možnost hmatové odezvy v podobě vibrací. Jsou napájeny z vestavěného akumulátoru.

Systém sledování je nazván Lighthouse (česky „maják“) a funguje na principu pravidelného skenování prostoru paprskem ze dvou synchronizovaných vysílačů, přičemž tento je detekován fotocitlivými body na povrchu headsetu i ovladačů a z časové souslednosti takových detekcí lze určit polohu i natočení jednotlivých zařízení [2]. Pokud jsou vysílače umístěny naproti sobě, synchronizují

se automaticky, v opačném případě je nutné je za tím účelem propojit kabelem. Systém se vyrovnává s krátkodobými výpadky sledování (způsobenými např. dočasným přerušením přímé viditelnosti mezi vysílačem a zařízením) použitím informací z akcelerometrického a gyroskopického snímání. [6]



Obrázek 1: HTC Vive²

HTC představilo na konferenci CES v lednu 2018 nový headset Vive Pro s vylepšeným displejem (AMOLED, 2880×1600 px), dvěma kamerami v přední části a upraveným pevnějším systémem popruhů s integrovanými sluchátky, podobným do té doby dostupnému Deluxe Audio Strap³ [7]. Související inovací prošly i ovladače a zejména systém sledování – nově lze využívat prostoru velikosti až 10×10 metrů. Zařízení je komerčně dostupné od dubna 2018.

Oculus Rift s ovladači Touch

Poslední verze headsetu Rift určená pro spotřebitelský trh je označena jako CV1. Je k dostání společně s ovladači Touch a dvěma senzory. Pro tzv. *360 setup* (tj. hráč stojí na jednom místě, ale může se plně otáčet) a *room-scale VR* jsou explicitně doporučeny tři kamerové senzory, což znamená pořízení jednoho samostatného senzoru navíc; lze je však zprovoznit i se dvěma senzory, byť takové uspořádání Oculus označuje jako „experimentální“ a upozorňuje na možnou nižší kvalitu sledování⁴. Je tedy patrné, že současná verze Riftu se na *room-scale VR* (na rozdíl od Vive) primárně spíše nezaměřuje. Tato tendence je často reflektována i v ovládání aplikací, které jsou pro něj určeny – scény jsou mnohdy postaveny jedním hlavním směrem⁵, případně umožňují hráči otáčet se pomocí

² zdroj obrázku: <https://www.flickr.com/photos/92587836@N04/24177102722>

³ viz <https://www.vive.com/eu/vive-deluxe-audio-strap/>

⁴ viz <https://support.oculus.com/188772188235494/>

⁵ např. demonstrační aplikace First Contact

stisku tlačítek (obvykle skokově o osminu či čtvrtinu plné otáčky)⁶, zatímco u her vyvíjených pro Vive nejsou uvedené vlastnosti typické.

Základní technické parametry displeje Riftu jsou v podstatě shodné s těmi, které poskytuje Vive (viz Tabulka 2). Headset taktéž obsahuje mikrofon a má nastavitelnou rozteč čoček. Liší se však způsob sledování, který zde funguje primárně na principu snímání infračervených světelných bodů, jež jsou aktivně vyzářovány diodami v headsetu i ovladačích, pomocí kamer.

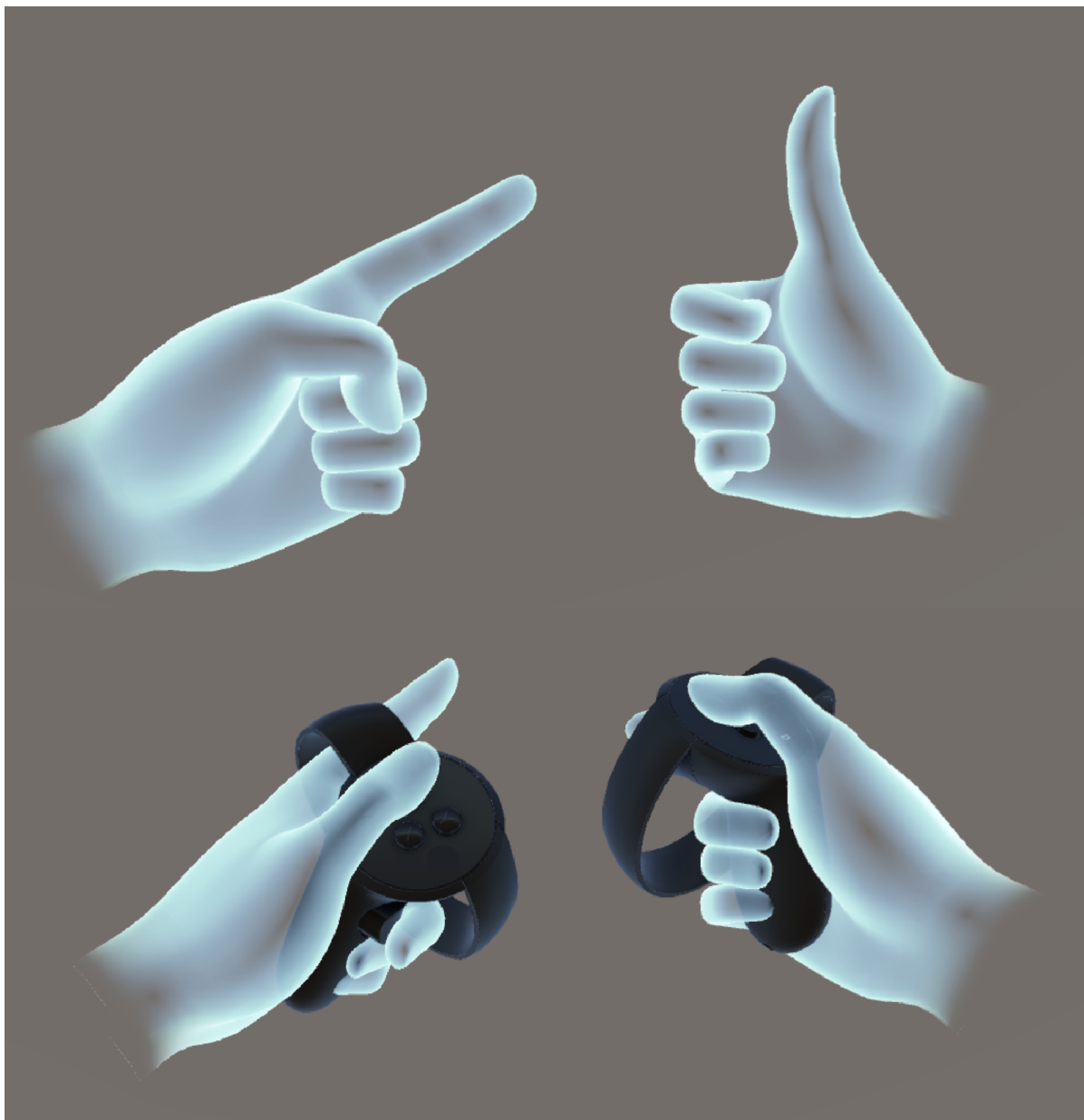
Typ displeje	OLED
Rozlišení	2160×1200 px
Obnovovací frekvence	90 Hz
Zorné pole	110°

Tabulka 2: Parametry displeje pro Oculus Rift

Bezdrátové ovladače Touch nejsou identické pro obě ruce, ale naopak je jeden pevně určen pro pravou a druhý pro levou ruku. Namísto *touchpadu* je přítomen *joystick* a uvnitř jsou oproti ovladačům Vive navíc umístěny dvouhodnotové senzory blízkosti umožňující odhadnout postavení prstů ruky a tuto zobrazit s využitím Oculus Avatar SDK (viz Obrázek 2). Tím je uživateli dovoleno provádět gesta sloužící pro komunikaci s ostatními hráči nebo funkční odlišování sémantiky interakcí s objekty. Rozmístění i počet tlačítek je obdobné jako má Vive a ovladače taktéž dokáží vibrovat, jsou však napájeny výměnnými bateriemi. [8]

Na rozdíl od Vive jsou dále u Riftu stereo sluchátka přímo konstrukční součástí headsetu, lze je ovšem demontovat. Rift naopak nemá kameru umožňující uživateli zobrazit jeho reálné okolí. Datové kabely jsou zároveň využity k napájení, čímž je kabeláž a její zapojení jednodušší než v případě Vive, kabel propojující headset s počítačem je však kratší, což samo o sobě omezuje velikost herní oblasti. Vzhledem k jinému způsobu sledování je nutné počítačem zpracovat i data ze senzorů, které jsou pevně umístěny ve fyzickém prostoru, a systém tak pro popsání způsobu provozu vyžaduje alespoň dva porty typu USB 3.0 a jeden USB 2.0, má tedy v tomto ohledu vyšší nároky na počet dostupných konektorů i využití sběrnice.

6 např. hry Echo Arena nebo Robo Recall



Obrázek 2: Zobrazení postavení ruky bez/s modely ovladačů Touch

2.3 Pohyb ve virtuálním prostředí

V mnoha případech je žádoucí poskytnout uživateli možnost aktivně měnit svou pozici ve virtuálním prostředí (*travel*). Obecně je vhodné, aby byla použita technika, která takový pohyb zprostředkovává, co nejpřirozenější a nejintuitivnější, aby ji mohl uživatel provádět automaticky a mohl se soustředit na záměr svých akcí, nikoliv na způsob jejich provedení. Zároveň je nezbytné, aby nedocházelo k vyvolávání nevolnosti způsobené nesouladem informací vnímaných vizuálně s těmi z rovnovážného aparátu. Přitom je obvykle nutné, aby se v prostředí dobře orientoval, věděl, kde je on sám i místo,

kam se chce dostat (*wayfinding*). Vhodnost techniky pohybu závisí na konkrétní úloze a podobě scény, kterou má plnit, a je tedy přinejmenším aplikačně specifická. Lišit se mohou požadavky na překonávanou vzdálenost, přesnost pohybu, nutnost kontroly nad jeho rychlostí a podobně. Jednotlivé techniky pohybu mohou být klasifikovány na základě různých vlastností. Možná dělení zahrnují [2]:

- aktivní a pasivní
- fyzické a virtuální
- pomocí dekompozice úlohy
- podle použité metafory

Aktivní a pasivní

U aktivních technik je pohyb řízen uživatelem, zatímco v případě pasivních je prováděn podle logiky aplikace a uživatel nad ním nemá kontrolu. Existují i kombinace obou přístupů, např. naplánování trasy uživatelem a následné automatické provedení pohybu.

Fyzické a virtuální

Při použití fyzických technik je využit skutečný pohyb těla uživatele, který je převeden na změnu pozice či rotace ve virtuálním prostředí. Na provedení pohybu virtuální technikou nemá pohyb těla uživatele vliv. Např. chůze v rámci klasických desktopových her z pohledu první osoby je technikou čistě virtuální. Ve VR aplikacích jsou obě kategorie často kombinovány – rozhlížení může být dáno fyzickým otáčením hlavy, zatímco vlastní posun po scéně je virtuální.

Dekompozice úlohy

Pohyb jakožto úlohu lze dekomponovat např. na výběr směru/cíle, určení zrychlení/rychlosti a definici stavu vstupů, které způsobují vyvolání, pokračování nebo zastavení pohybu. Specifikací těchto podúloh lze navrhnout celou škálu různých technik pohybu. Např. pohyb může být prováděn konstantní rychlostí po dobu držení tlačítka směrem, kterým se uživatel dívá.

Dle metafory

Techniky pohybu lze klasifikovat také podle interakční metafory, kterou používají. Takové dělení je abstraktnější, má menší návaznost na jejich implementaci. Základní typy dělení technik pohybu podle metafory jsou:

- založené na chůzi
- zatačení, resp. výběr směru
- výběr cesty/cíle
- pohyb skrze manipulaci s objektem

Techniky **založené na chůzi** zahrnují samozřejmě přímou reálnou chůzi, která je však omezena velikostí sledované plochy. K odstranění tohoto omezení lze použít speciální zařízení na principu běžeckého pásu nebo podložky s nízkým třením, která udržují uživatele na stálé pozici ve fyzickém prostoru. Alternativou je tzv. „přesměrovaná“ chůze (*redirected walking*), spočívající v zajištění rozdílu mezi trajektorií uživatele ve skutečném prostoru a jejím obrazem v prostoru virtuálním. Uživatel pak tento rozdíl automaticky kompenzuje zvýšením nebo snížením rozsahu vlastního pohybu. Výsledek může vypadat tak, že uživateli je při chůzi po křivce (v kruhu) vyvoláván vjem chůze přímé. Další možností je změna měřítka pohybu, tedy převod mírného reálného pohybu do

rozsáhlejšího virtuálního (*scaled walking*). Lze využít i částečné chůze, např. chůze na místě (detekovatelné třeba z pohybu rukou; *walking in place*), případně techniku „lidského joysticku“ (*human joystick*), při které uživatel ovládá pohyb změnou pozice na ovládací podložce.

Směr pohybu může být udáván např. směrem pohledu nebo ukazováním rukama (ovladači), může při tom být využita jedna nebo obě ruce, pohyb může probíhat za ukazovaným směrem nebo proti němu. K získání směru může být použito též sledování orientace těla nebo jeho náklonu, případně specializovaná vstupní zařízení.

Jiným přístupem je nenutit uživatele přímo vykonávat požadovaný pohyb, ať už je fyzický či virtuální, ale nechat ho **specifikovat cíl** a poté provést pohyb pasivně, bez jeho účasti. Takový způsob není příliš přirozený, ale je poměrně jednoduše pochopitelný i proveditelný. Do této kategorie spadá teleportování. To může proběhnout buď jako skoková změna pozice nebo jako plynulý „přelet“ do pozice nové. Skoková změna může vést ke ztrátě orientace uživatele, plynulý přechod zase vzhledem k tomu, že jde o pohyb, který není průběžně řízen uživatelem, může způsobovat nevolnost a je tedy nutné vhodně nastavit jeho rychlost. Cíl může být zadán např. určením bodu virtuálního prostředí pomocí ukazatele, zadáním souřadnic, výběrem z možných předem definovaných destinací nebo s pomocí mapy či zmenšeniny scény, případně může uživatel podobným způsobem nakreslit trajektorii kýženého pohybu nebo vytyčit body, kterými má procházet.

Pohyb lze založit také na **manipulaci s objektem**. Lze např. pohybovat kamerou prostřednictvím snímání pohybu ruky – virtuální scéna je vykreslena dle pozice a natočení sledované ruky (ovladače) v ní (*camera-in-hand*). Uživatel též může pohybovat virtuální reprezentací sebe sama ve zmenšené verzi scény (*avatar manipulation*), případně může nastavovat svou pozici relativně k nějakému statickému objektu scény (např. v podobě šplhání po horolezecké zdi; *fixed-object manipulation*), nebo naopak pohybovat virtuální scénou relativně k vlastní pozici (*world manipulation, scene-in-hand*). [2]

Praktické zastoupení

Tabulka 3 zobrazuje techniky pohybu použité v 15 VR aplikacích vydaných v obchodě Steam s největším počtem aktivních majitelů podle serveru SteamSpy⁷. Počet hráčů je zaokrouhlen na celé tisíce. Uvedené údaje o technikách pohybu jsou většinou odhadnuty z dostupných propagačních materiálů a specifikovaných požadavků hry na herní oblast, mohou se tedy lišit od skutečnosti. V případě možnosti „žádná“ může hráč stále chodit v rámci své herní plochy, nicméně hra s touto možností nepočítá jako s nutnou součástí – taková aplikace je určena též pro mód vsedě (*seated*) nebo vestoje (velikost plochy 1 × 1 m, resp. kruh o průměru 1 m; *standing-only*). Možnost „chůze v rámci malého prostoru“ znamená, že hra využívá *room-scale VR*, nicméně neklade požadavky na větší prostor, než je minimální 2 × 1,5 m, ani nepoužívá žádnou techniku umístění rozsáhlejšího virtuálního prostoru do této plochy.

Hra *Climbey* využívá zajímavé techniky skákání v duchu výše popsané chůze na místě – střídavé máchání rukama znamená chůzi, současné vymrštění obou rukou je skok. Jak je však z tabulky patrné, u uvedených her, které se neodehrávají jen na jednom místě virtuálního prostředí, převažuje teleportace (technika založená na výběru cíle), ať již volná nebo v podobě výběru z předem definovaných cílových míst.

⁷ zdroj dat: <http://steamspy.com/tag/VR+Only> (leden 2018)

Název	Cena	Počet hráčů	Technika pohybu
The Lab	zdarma	567 000	teleport
Rec Room	zdarma	244 000	teleport
Waltz of the Wizard	zdarma	172 000	žádná
Virtual Desktop	14,99 \$	156 000	žádná
Job Simulator	19,99 \$	153 000	žádná/chůze v rámci malého prostoru
Space Pirate Trainer	14,99 \$	127 000	žádná
Audioshield	19,99 \$	114 000	žádná
Arizona Sunshine	39,99 \$	103 000	teleport
SUPERHOT VR	24,99 \$	95 000	žádná
The Gallery – Episode 1: Call of the Starseed	19,99 \$	81 000	teleport
The Brookhaven Experiment	19,99 \$	61 000	žádná
Rick and Morty: Virtual Rick-ality	29,99 \$	60 000	teleport
Whirligig VR Media Player	3,99 \$	35 000	žádná
Climbey	9,99 \$	34 000	chůze na místě, šplhání
Richie's Plank Experience	14,99 \$	34 000	chůze v rámci malého prostoru, létání po směru ovladačů, pasivní pohyb

Tabulka 3: Porovnání technik pohybu nejrozšířenějších VR aplikací⁷

3 Analýza a obecný návrh

Kapitola 3.1 definuje obecné vlastnosti uvažovaného způsobu pohybu a zařazuje jej v kontextu klasifikace pohybových technik, přičemž vychází ze skutečností popsaných v kapitole předchozí. Stručně představuje dva základní principy, které by bylo možné použít při implementaci aplikace, jež by takovou techniku pohybu používala. Jde spíše o povšechný úvod s cílem vytvořit hrubou představu o podobě takového řešení, možnostech jeho realizace, souvisejících vlastnostech a omezeních, které přináší, a hlediscích, které je třeba při jeho tvorbě adresovat; to vše bez ohledu na přístup zvolený k jeho případné implementaci. Detailní popis konkrétního řešení realizovaného v rámci praktické části této práce je potom uveden v kapitole 4.2.

Kapitola 3.2 specifikuje základní parametry navrženého testovacího prostředí tak, aby posloužilo k porovnání předmětného přístupu s metodami v současné době obvyklými. Detailní rozbor skutečně vytvořených testovacích scén, které z této specifikace vycházejí, je předmětem kapitoly 4.3.

3.1 Konceptuální návrh metody pohybu

Cílem práce je navrhnout a vytvořit způsob využití dostupného herního prostoru. Základní ideou je přitom „jednoduše“ založit pohyb ve virtuálním prostředí pouze na přímém přenosu fyzického pohybu uživatele. V rámci klasifikace nastíněné v předchozí kapitole jde tedy o techniku aktivní, fyzickou a založenou na chůzi. Problém tvoří omezení herní plochy, která je pro sledování pohybu k dispozici – její rozsah přitom není limitován primárně technickými omezeními současných přístrojů, ale spíše skutečnou velikostí dostupného prostoru v místnostech bytů potenciálních uživatelů (tj. jde-li o aplikaci nabízenou přímo jednotlivým spotřebitelům; v případě VR heren a podobných zařízení bývá typicky dostupný prostor větší). Jde tedy o parametr, u kterého nelze očekávat žádné podstatné vylepšení s postupujícím časem. Vytvoření zajímavé aplikace nabízející dostatek herního obsahu a uspokojivě rozmanité prostředí na ploše v řádu jednotek čtverečních metrů (viz níže) může být velice často náročné či nemožné. Rozsah sledované chůze, kterou může uživatel provádět, je však pochopitelně omezen dostupnou herní plochou. Otázkou tedy je, jak dosáhnout umístění dostatečně rozsáhlého virtuálního prostředí do omezeného herního prostoru tak, aby bylo možné využití přirozené chůze jako jediného prostředku pohybu. Navržené řešení spočívá v rozdělení virtuálního prostoru na jednotlivé místnosti (či jiné dílčí kusy) a „recyklaci“ dostupného fyzického prostoru tím, že se virtuální místnosti budou překrývat a fyzický prostor částečně sdílet, viz Obrázek 3. Nad tímto konceptem je snad možné uvažovat jako nad projekcí 4D prostoru do 3D světa, kdy se přechodem mezi místnostmi posune hráč na jinou úroveň, vrstvu. To ve svém důsledku umožňuje umístění většího virtuálního prostoru do menšího prostoru fyzického.

Jedná se o techniku, již lze označit jako bezprostřední, nevyužívající žádné umělé metafory a kterou je z toho důvodu možné považovat za vysoce přirozenou [2], není však tak univerzální jako např. teleportace. Jedinou „nepřirozeností“, kterou zavádí, je fakt, že různá místa ve scéně mohou být umístěna do stejných pozic skutečného prostoru; hráč může dojít na totéž místo jinou cestou a být přitom v jiné části virtuálního prostředí – to může potenciálně snižovat přehlednost takového virtuálního prostředí a ztěžovat orientaci uživatele v něm. Nevýhodou z pohledu případné monetizace

aplikace je implicitní omezení šíře potenciální uživatelské základny, neboť vyžaduje poměrně velkou herní oblast, přičemž s rostoucími nároky na obsah této plochy se snižuje počet uživatelů, kteří ji mají k dispozici (viz Tabulka 4). Ostatní praktické dopady použití popsané techniky zahrnují rozměrovou omezenost místností, z nichž se virtuální prostředí skládá, přičemž pobyt v takových prostorách může být některým uživatelům nepříjemný nebo může působit nepřírozně, zejména pokud podoba takového prostředí nebude rezonovat se zaměřením aplikace, tedy např. odpovídat tématu a atmosféře hry. Dále je možné, že takový způsob pohybu bude narušovat uživatelskou orientaci ve virtuálním prostředí jako celku. Při setrvalé navigaci uživatele jedním směrem podél herního prostoru (např. chodba zatáčející stále vpravo umístěná po obvodu herní plochy) může postup působit monotónně a navíc se zvyšuje riziko zakopnutí o kabel, který vede k headsetu, je tedy vhodné tvořit místnosti rozmanitějšího tvaru a směr průchodu střídat. Nejen proto vyžaduje použití této techniky obezřetný návrh jednotlivých místností.

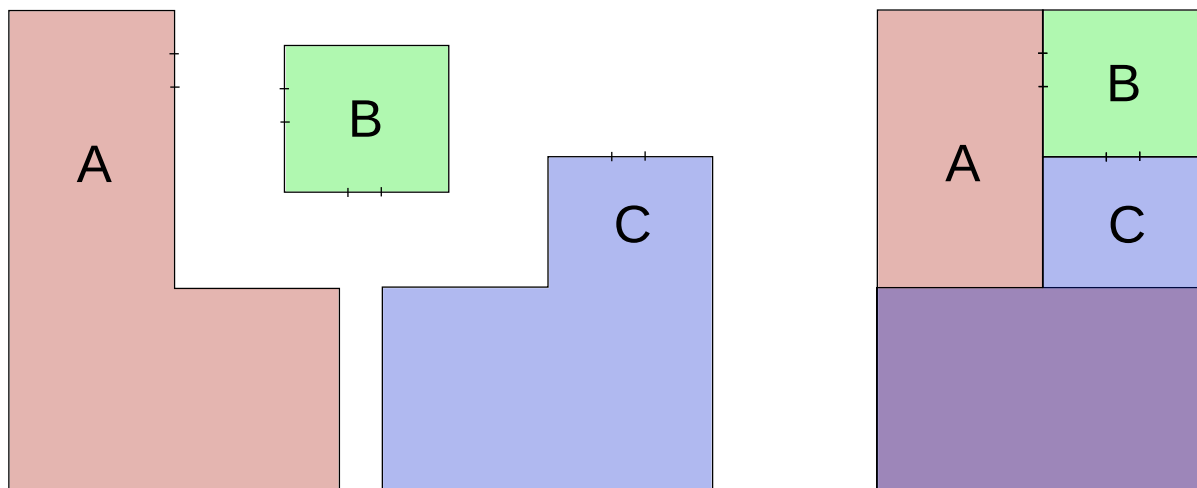
Tabulka 4 uvádí, kolik procent aktivních uživatelů platformy Steam, kteří mají kompatibilní headset, disponuje herní oblastí větší, než je ta daná rozměry odpovídajícími hodnotám příslušného řádku a sloupce. Jde o statistiku použití SteamVR za říjen/listopad 2017. Údaj zveřejnil vývojář společnosti Valve na diskuzním fóru Steamu [9].

	1,5 m	2,0 m	2,5 m	3,0 m	3,5 m	4,0 m
2,0 m	69,0%	52,0%				
2,5 m	45,0%	39,0%	20,0%			
3,0 m	21,0%	19,0%	12,0%	5,0%		
3,5 m	8,0%	8,0%	6,0%	3,0%	1,0%	
4,0 m	3,0%	3,0%	2,0%	1,0%	0,7%	0,3%

Tabulka 4: Srovnání procentuálního zastoupení velikostí herního prostoru [9]

V práci, jejímiž autory jsou Suma et al., je vyhodnocen potenciál použití techniky pohybu po virtuálním prostředí založené na uvedených principech, přičemž její závěr konstatuje, že jde o slibný způsob umožnění pohybu v aplikaci prostřednictvím provádění fyzické chůze, zejména z hlediska jeho celkové přirozenosti a přijetí uživateli [10]. Práce však využívá pouze velmi elementární implementaci takového prostoru, s překryvem dvou místností nepřevyšujícím 75 % plochy, navíc je virtuální scéna zasazena do herní oblasti velikosti 9 × 9 metrů, přičemž takové řešení není vhodné k vydání pro spotřebitelský trh, neboť prostorem uvedené velikosti koncoví uživatelé typicky nedisponují. Systém navržený a implementovaný v této práci bude překrývání virtuálních místností využívat podstatně intenzivněji (ideálně škálovatelně a nejlépe potenciálně neomezeně) na významně menším fyzickém prostoru. Výše uvedená práce se však snažila o vytvoření takového prostoru, jehož překryvu si uživatel nemá být vědom, zatímco systém, který bude výstupem této práce nepředpokládá, že by si při jeho užívání překryvu bylo možné nevšimnout (ani se o to nesnaží). To by však nemělo samo o sobě činit pobyt v něm nijak nepříjemným a tedy omezovat možnosti jeho použití.

Obrázek 3 ilustruje konkrétní elementární příklad půdorysu takové scény tvořené třemi místnostmi označenými A, B a C s vyznačením dveří. V levé části obrázku jsou tyto místnosti zobrazeny odděleně, v pravé jsou složeny dohromady. Jak je z obrázku patrné, z místnosti A je možno dveřmi projít do místnosti B, a z té poté do místnosti C, případně opačným směrem. Fialová oblast představuje tu část fyzického prostoru, která je místnostmi A a C sdílena.



Obrázek 3: Ilustrační příklad místností

K zajištění přístupnosti celého virtuálního prostoru v rámci herní plochy nejmenší podporované velikosti je třeba udržovat pevnou, konzistentní korelaci pozice v herní ploše s pozicí v 3D scéně. Z toho důvodu nelze při použití virtuálního prostoru s překrývajícími se místnostmi, v němž má být hlavní technikou pohybu fyzická chůze, jednoduše tuto techniku kombinovat s většinou jiných aktivních technik – hráč by se tak mohl posunout ve scéně, aniž by změnil svou pozici ve skutečném prostoru, přičemž při pozdější nutnosti dojít na konkrétní místo scény pomocí fyzické chůze by mu mohl právě rozdíl v prostoru, který takovým pohybem vznikl, chybět.

Obecné implementační problémy

Kromě dosti evidentního problému spočívajícího v zajištění korektního vykreslení aktuálně platné části překrývajících se místností (nejlépe včetně správného osvětlení a stínů) je třeba při implementaci popsané metody vyřešit i další, na první pohled ne nutně zřejmé zvláštnosti. Zásadní je zajištění korektního chování fyziky, zejména v případě dynamických (tj. pohyblivých, přenosných) objektů: je nutné, aby objekty interagovaly se statickými částmi příslušné místnosti, v níž se nacházejí, stejně jako s dalšími dynamickými objekty v téže místnosti, ale naopak vůbec nerefletovaly objekty příslušející do místností jiných, a to i v případě, že jim odpovídá blízká či stejná pozice ve skutečném prostoru herní plochy – to vše nejlépe nezávisle na tom, zda dané objekty jsou nebo nejsou v daném okamžiku viditelné. To se týká i interakcí s hráčem, resp. jeho reprezentací ve scéně – je nutné umožnit uživateli např. zvednout objekt, který je v téže části virtuálního prostoru, ale zabránit zvedání objektů, které jsou virtuálně jinde. Dále je vhodné vyřešit specifické chování jevů souvisejících s prostorovou vzdáleností – komplikace v tomto případě vyplývá z faktu, že vzdálenost mezi dvěma body v předmětné virtuální scéně nemusí být rovna Euklidovské vzdálenosti bodů reálného prostoru, které jim odpovídají (virtuální vzdálenost může být větší). Příkladem může být útlum hlasitosti zvuků a jiné efekty související s jeho šířením, které zde obecně není validní modelovat jako pouhou přímou propagaci od zdroje do bodu prostoru, v němž se nachází posluchač.

Zcela obecné řešení zmíněných (a případných dalších) problémů může představovat značnou výzvu, nicméně realizace takového komplexního řešení nemusí být nezbytně nutná – míra složitosti vykreslovacího algoritmu např. závisí zejména na maximálním počtu překrývajících se místností

zobrazovaných současně a jejich přípustném tvaru, což bude v praxi dáno konkrétní podobou hry, podmíněno návrhem úrovní a tedy ovlivněno např. zvolenými herními mechanikami, dějovou linií a podobně. Níže jsou navrženy koncepty dvou širších přístupů, z nichž by mohla implementace vycházet, aby bylo dosaženo výše popsaného efektu, jistě však existují i některé další.

Koncept přepínání místností

Patrně nejjednodušší možnou variantou je úplně se vyhnout současnému zobrazování místností (nebo obecně částí prostoru), jejichž části se mohou překrývat. Jednou z možností jak zaručit, že nevznikne nutnost takového vykreslování, je vložit mezi překrývající se místnosti jakousi přechodovou komoru, která se ani s jednou z nich nepřekrývá, a zajistit, že v každém okamžiku mohou být otevřeny maximálně jedny z dveří vedoucích do místností, které se překrývají; to znamená vynutit při přechodu zavření vstupních dveří do přechodové komory a teprve poté umožnit otevření dveří do překrývající se místnosti. Výchozí místnost je „vypnuta“ a cílová poté „zapnuta“ podle výběru dveří. V kontextu scény znázorněné na pravé straně Obrázku 3 zastává funkci přechodové komory místnost B. Postup průchodu z A do C by byl následující:

1. Hráč se nachází v místnosti A. Všechny dveře jsou zavřeny. Dveře mezi A a B jsou aktivní (lze je otevřít), dveře mezi B a C nikoliv. Místnost C není vykreslována.
2. Hráč otevře dveře mezi místnostmi A a B a projde jimi do místnosti B.
3. Hráč za sebou zavře dveře vedoucí zpět do místnosti A, případně jsou po jeho průchodu zavřeny automaticky.
4. Místnost A přestane být vykreslována a dveře mezi B a C jsou aktivovány.
5. Hráč začne interagovat se dveřmi vedoucími do místnosti C.
6. Místnost C začne být vykreslována, dveře do místnosti A jsou deaktivovány.
7. Hráč přejde do místnosti C.

Spouštěcí událostí v bodě 5 může být například detekce „doteku“ ovladače s klikou dveří, přičemž se lze takto dotýkat pouze jedné z klik (jako platná může být považována poslední nastavší událost tohoto typu) a pouze klika, které se hráč dotýká, může být použita k otevření dveří.

Výhodou popsané verze je kromě implementační jednoduchosti též relativně vysoká efektivita z pohledu vykreslování – části scény, které nemohou být vidět, vůbec nejsou předávány k zobrazení. Nevýhodou je nezbytnost častého nucení hráče do uzavřených prostor sestávajících pouze z malé části herní oblasti, která již sama o sobě obvykle není nikterak rozsáhlá, což pravděpodobně povede ke zhoršení herního zážitku. Průchod několika bezprostředně po sobě následujícími místnostmi tvořenými alespoň polovinou herní oblasti nelze obecně výše popsanou variantou realizovat, v závislosti na tvaru místností (případně po jejich obezřetném návrhu za tímto účelem) by však bylo možné rozdělit je na několik menších částí – „podmístností“ a tyto přepínat v reakci na průchod uživatele definovanými body či úseky. Takové rozdělení by však nejspíše bylo nutné provádět poměrně pracně „ručně“ už jako součást návrhu scény; jeho automatizace nad obecnou scénou totiž představuje samostatný netriviální problém. Tento přístup je dále rozvinut a popsán v kapitole 4.2.

Koncept šablonování průhledů

Možnou alternativou je vykreslení ve více průchodech s použitím *stencil bufferu*, umožňujícího vykreslení obsahu jednotlivých místností pouze na ta místa obrazu, v nichž mají být viditelné. Všechny objekty (resp. jejich materiály) by měly nastavené unikátní číslo místnosti, ve které se nacházejí. Zárubně dveří by byly vyplněny plochou, která by se nezobrazovala (nezapisovala do

frame bufferu), pouze by ovlivňovala hodnoty ve *stencil bufferu*. Před vykreslením každého snímku by byl *stencil buffer* nastaven na hodnotu odpovídající místnosti, v níž se hráč (kamera) nachází, poté by byly zapsány na místa odpovídající ve *stencil bufferu* odpovídající průhledům do jiných místností (tj. skrz dveře) hodnoty příslušející souvisejícím místnostem. Následně by byly vykresleny všechny objekty všech místností se *stencil testem* na rovnost hodnoty místnosti nastavené v jejich materiálu.

Objekty nacházející se mezi hráčem a dveřmi, ale patřící do jiné místnosti, která však sdílí část fyzického prostoru s místností, ve které se právě nachází hráč, by se zobrazovaly do průmětu dveří, které do této místnosti vedou, což je nežádoucí jev. Je tedy nutné buď návrhem místností zajistit, že k takové konstelaci nemůže dojít, případně místnosti rozdělit na menší celky (tj. masky umístit nejen do dveří) nebo vhodně použít doplňkovou hloubkovou informaci (tj. zobrazovat jen ty předměty z druhé místnosti, které jsou dále než dveře). Naivní použití této metody vede k zahazování velkého množství zpracovaných fragmentů při *stencil testu*, tj. až po dokončení průchodu objektů jiných místností celým vykreslovacím řetězcem. Pro praktické použití by tedy bylo vhodné přidat nějaký speciální *occlusion culling* zohledňující překrývání místností a průhledy mezi nimi, případně jiné omezení počtu vykreslovaných objektů. Dalším problémem jsou stíny zasahující do více místností – pro každý průhled („masku dveří“) by bylo nutné např. počítat vlastní *shadow mapu*.

3.2 Návrh testovacího prostředí

Podoba testovacího prostředí vyplývá z položené otázky, k jejímuž zodpovězení má testování posloužit. Obě tato témata jsou detailněji popsána v následujících odstavcích.

3.2.1 Záměr testování

Cílem práce je pokusit se odpovědět na otázku, zda v případě, kdy se návrhář může rozhodnout založit pohyb ve své hře na fyzickém pohybu hráče (viz kapitola 3.1) namísto využití obvyklých technik (viz kapitola 2.3), může obecně být takové rozhodnutí výhodné. To znamená, zda může výsledná hra lépe zaujmout hráče (třeba už jen tím, že se tímto liší od jinak podobných her), příjemněji se ovládat, vyvolat silnější atmosféru a zážitek nebo se jinak pozitivně odlišovat od své konkurence a vystupovat z ní. Takováto technika pohybu totiž implicitně představuje nároky na větší herní oblast, čímž je omezována množina potenciálních uživatelů na ty, kteří mají k dispozici dostatečný prostor, navíc je typicky návrhově i implementačně komplikovanější (viz kapitola 3.1). Pokud její případná aplikace nevede ke zřetelně lepšímu vnímání uživatelů, je tedy obecně spíše nevýhodné ji zvolit.

Sledované uživatelské hodnocení je tedy veskrze subjektivního rázu. Pro popsané rozhodnutí není ani příliš relevantní konkrétní důvod, který vedl uživatele k lepšímu hodnocení jedné techniky pohybu než jiné, jakkoliv tento důvod může být zajímavý pro další analýzu. Za tím účelem je vhodné formulovat otázku co nejobecněji, např. ptát se, která varianta se uživateli „více líbila“, která byla „lepší“ nebo kterou „preferoval“. Naopak snaha o objektivní vyhodnocení vede v triviálních případech (např. samostatné měření času potřebného k vykonání určitého úkonu) k předem zřejmým nebo naopak zcela nahodilým a o ničem nevypovídajícím závěrům v závislosti na podobě scén, na kterých takové vyhodnocení probíhá. Komplexní objektivní vyhodnocení by mohlo být žádoucí pro porovnání konkrétních případů (tj. za určitým účelem nebo ve specifickém virtuálním prostředí)

použití srovnávaných technik pohybu; při jeho provedení v rozsahu odpovídajícím diplomové práci by však bylo obtížné prokázat všeobecnou platnost tvrzení z něj vyvozených.

3.2.2 Podoba testovacího prostředí

Testovací prostředí pro porovnání technik pohybu ve virtuálním prostředí se bude skládat z několika dvojic scén, v každé dvojici bude jedna scéna využívat pohyb teleportací (reprezentující obvyklé techniky pohybu jakožto nejčastější z nich) a druhá výše navrženou technikou založenou pouze na sledování fyzického pohybu uživatele. Vzhledem ke skutečnosti, že tyto metody pohybu jsou velmi odlišné, musí jim být podoba příslušné scény přizpůsobena. Nelze validně porovnávat totožné scény lišící se pouze použitou technikou pohybu, protože podoba scén samotných obecně favorizuje určitou techniku – její použití v dané scéně je a priori vhodnější. Protože obě uvedené techniky jsou porovnávány nepřímou (uživatelé vzájemně hodnotí pouze pár scén, v nichž jsou použity, nikoliv přímo techniky jako takové), musí být za účelem dosažení vytyčených cílů testování tyto scény zpracovány „co nejlépe je možné“ s ohledem na ten daný způsob pohybu, který je v nich použit. To znamená, že se může velikost, tvar i grafická podoba obou scén v testovacím páru do jisté míry lišit. Přesto je však nutné zajistit určité společné vlastnosti, aby byly scény rozumně porovnatelné. Touto vlastností bude zejména zadaný úkol, který má ve scénách hráč splnit, např. najít konkrétní předmět a přenést jej na určité místo. To odpovídá záměru testování v tom smyslu, že při volbě techniky pohybu pro aplikaci už je obvykle vytyčeno širší zaměření hry a typ úkonů, které v ní má uživatel vykonávat. Srovnatelná bude i vizuální stylizace a grafická kvalita zpracování obou scén v jednom testovacím páru.

Výhodou jedné z technik pohybu může být např. i snazší orientace v prostoru. Je proto žádoucí, aby součástí testování byly i rozsáhlejší scény, v nichž je zachování orientace uživatele nutné. Měly by také být zastoupeny co možná nejrozličnější typy úkonů a interakčních technik, které lze ve hrách využívajících uvažované techniky pohybu uplatnit, např. chytání, přenášení a umísťování objektů; odstraňování překážek; manipulace s objekty s omezeným pohybem (dveře apod.).

Může se stát, že jedna scéna z testovacího páru bude svým zpracováním nějak převyšovat scénu druhou, což by samozřejmě nežádoucím způsobem ovlivnilo celkové vyhodnocení porovnání obou technik. Vliv této eventuality může být minimalizován zvýšením počtu testovacích párů. Zároveň bude třeba co největšího počtu uživatelských hodnocení, s ohledem na cíle práce ideálně od hráčů, kteří již mají s VR hlubší zkušenosti, popř. vlastní VR headset, neboť tito tvoří reálnou zákaznickou základnu, na kterou vydávané aplikace míří. V prostorách laboratoře FIT VUT lze prakticky provést jen značně omezený počet testování, a to s osobami, které ve většině případů mají jen velmi malé nebo žádné zkušenosti s VR. Proto by bylo optimální vytvořit aplikaci, která by přidělovala testovací páry pseudonáhodně a tuto zveřejnit ke stažení zdarma např. přes obchod Steam, na němž jsou vydávány jak aplikace pro Vive, tak pro Rift. Uživatelé by mohli být k provedení testování motivováni odemčením achievementu a ostatních testovacích scén k vyzkoušení, které by proběhlo po dokončení testování, a jejich jednotlivá hodnocení by byla odesílána a ukládána v databázi. Testování na FIT VUT by pak mohlo probíhat v menším rozsahu (s menším počtem účastníků), ale s detailnějšími otázkami či neformálním rozhovorem po jeho skončení. „Kvalitativní“ testování by tak umožňovalo interpretovat získaná data, „kvantitativní“ by ověřovalo obecnost jejich platnosti.

Závěrem následného vyhodnocení nemusí nezbytně být jen jasná odpověď ve prospěch té či oné techniky pohybu, ale je možné, že se ukáže, že problém není takto obecně jednoznačně rozhodnutelný (tedy že je preference specifická pro určité typy scén) nebo že scény byly navrženy nevhodně – to by se mohlo projevit např. výrazně vyšší preferencí jedné ze scén u převážné většiny testovacích párů. V každém případě bude nutné interpretovat výsledky opatrně a nejlépe získané závěry dále ověřit, přičemž by se taková navazující práce mohla podrobněji zabývat otázkou v oblasti podstatně specifictější.

4 Implementace

V úvodu kapitoly jsou představeny dostupné nástroje, které byly použity při implementaci praktické části práce a lze je s výhodou využít i při tvorbě jiných VR aplikací. Následuje detailní popis principu vytvořeného řešení dovolující chůzi ve virtuálním prostoru s překrývajícími se místnostmi včetně zmínky možnosti dalších rozšíření a vylepšení představené implementace, které by bylo vhodné doplnit před použitím tohoto systému v komerční aplikaci. Závěrečná podkapitola přibližuje konkrétní podobu a účel ostatních částí testovacího prostředí spolu s vybranými komponentami, které byly při jeho tvorbě implementovány.

4.1 Použité nástroje⁸

Scény, z nichž se skládá testovací prostředí, jsou vytvořeny za použití herního engine Unity a jazyka C#. Podporují zařízení HTC Vive a Oculus Rift (s ovladači Touch), neboť jde o v současnosti nejrozšířenější VR systémy s podporou *room-scale VR*. Unity nabízí implicitní podporu požadovaných VR zařízení, nicméně na Unity Asset Store jsou zdarma k dispozici též balíčky *SteamVR Plugin* a *Oculus Integration* vydané pod hlavičkou výrobců příslušných zařízení a obsahující mj. aktuální verze příslušných SDK společně s několika příklady, které znázorňují způsob práce s nimi a některé jejich možnosti. Základní popis Unity a jeho použití pro vývoj VR aplikace lze najít např. v předchozích diplomových pracích z FIT VUT, jako je práce Ing. Novotného [4].

4.1.1 VRTK

Dalším užitečným balíčkem je *VRTK – Virtual Reality Toolkit*. Jeho použití přidává úroveň abstrakce (zvyšuje nezávislost na konkrétním zařízení), usnadňuje implementaci interakcí s objekty a nabízí kolekci řešení typických dílčích problémů, které je často nutné adresovat při tvorbě VR aplikace. Tyto mohou posloužit čistě jako inspirace, základ vlastní implementace nebo mohou být přímo použity v projektu, přičemž takové použití je často velmi jednoduché – stačí umístit správný skript do relevantního *GameObjectu* v Unity Editoru a případně nastavit reference na další související komponenty. VRTK je dostupný zdarma (MIT licence), obsahuje ukázkové scény zaměřené na jednotlivé dílčí součásti⁹, je poměrně kvalitně dokumentován¹⁰ a k dispozici jsou též názorná videa ilustrující nejzásadnější možnosti toolkitu a znázorňující postup, jakým je lze využít¹¹. V provozu je též komunitní chat s množstvím aktivních uživatelů¹².

Podpora několika SDK, události

Komponenta `VRTK_SDKManager` zajišťuje správu a přepínání několika SDK v rámci aplikace a VRTK tak tvoří určitou abstraktní vrstvu nad podporovanými SDK/zařízeními (včetně obou výše

⁸ Kapitola částečně přejímá informace zjištěné autorem při práci na projektu do předmětu GZN [11].

⁹ obsah popsán na <https://vrtoolkit.readme.io/docs/examples>

¹⁰ viz <https://vrtoolkit.readme.io/docs>

¹¹ viz tutorials.vrtk.io nebo <https://www.youtube.com/channel/UCWRk-LEMUNoZxUmY1wO7DBQ>

¹² viz chat.vrtk.io nebo <https://vrtoolkit.slack.com/messages>, přihlášení do diskuze skrz invite.vrtk.io

zmíněných), což umožňuje mj. jednotnou, obecnou obsluhu vstupních událostí bez ohledu na konkrétní zařízení, se kterým je aplikace spuštěna. K jejich předávání jsou využity standardní Unity/C# události (*events*), ty jsou ovladačem vyvolávány prostřednictvím skriptu `VRTK_ControllerEvents`. Další události lze využít např. také k detekci přepnutí používaného SDK.

Interakce s objekty

Při implementaci interaktivních předmětů s vlastním chováním je výhodné dědit ze třídy `VRTK_InteractiveObject`, přičemž programování chování takového objektu je obdobné jako v případě odvození ze standardní třídy Unity `MonoBehaviour` (včetně přístupnosti obvyklých metod – `Awake`, `Start`, `Update` atd.), navíc jsou však k dispozici (k překrytí, *overriding*) metody vyvolávané v reakci na vstupní událost související s daným objektem (tj. obvykle při současné přítomnosti ovladače uvnitř daného objektu), např. `StartTouching`, `Grabbed`, `Ungrabbed` a podobně.

Ukazatele

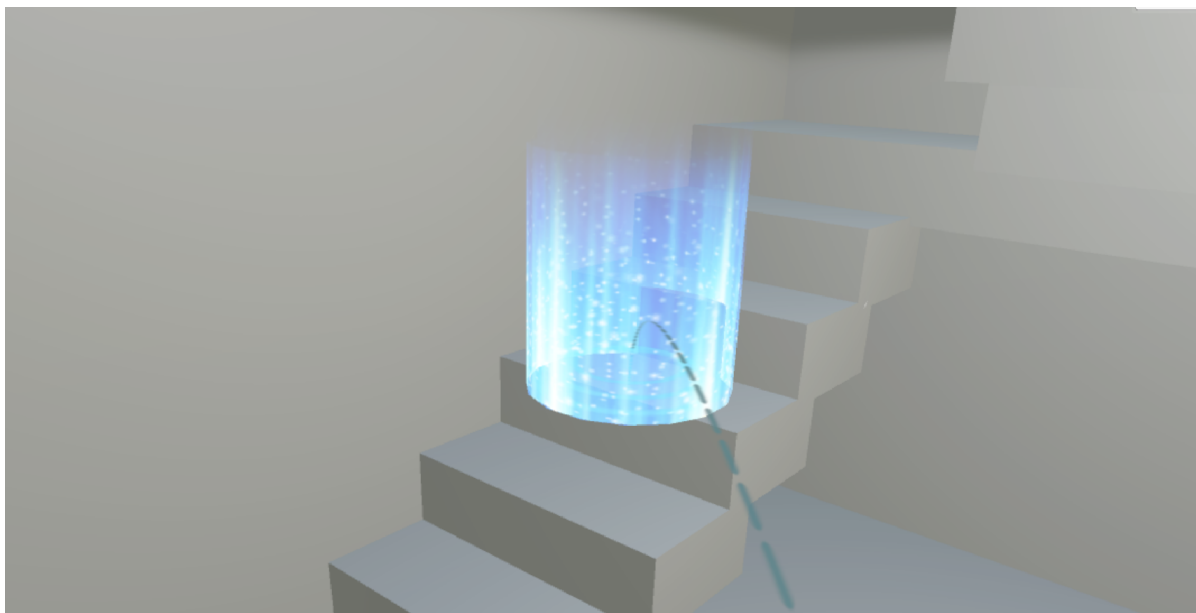
K použití jsou přichystané např. ukazatele (*pointers*), ať už přímý a křivkový pro ovladače (viz Obrázek 4), nebo ukazatel po směru pohledu (*gaze pointer*) pro headset. Ty lze využívat pro označování míst ve scéně (např. za účelem pohybu – *target-based travel*) nebo interakce se standardními komponentami Unity 2D UI (což je vhodné pro ovládání herního menu; pro zajištění kompatibility ukazatelů s Unity UI slouží skript `VRTK_UIPointer`).

Techniky pohybu

Součástí je též implementace několika různých technik pohybu vhodných pro použití ve VR:

- *teleporting* – skoková změna pozice s volitelným zčernáním obrazovky
- *dash teleport* – plynulý „přelet“ do nové pozice
- *touchpad movement* – pohyb ovládaný touchpadem nebo joystickem na ovladači
- *move-in-place* – pohyb založený na máchání rukama, resp. pochodování na místě
- *climbing* – šplhání, přitahování se k uchopeným objektům
- *room extender* – fyzický pohyb je převeden na rozsáhlejší pohyb ve virtuální scéně

Z hlediska tříd technik pohybu jsou tedy zastoupeny techniky jak fyzické, tak virtuální aktivní, místy kombinované i s technikami pasivními. V případě *target-based* technik lze definovat plochy výlučně povolené nebo naopak zakázané (skrze *policy list* nebo *destination area*), upravovat vertikální pozici hráče podle výšky cíle (*height adjust teleport*) a podobně.



Obrázek 4: Ukazatel pro teleport s přizpůsobením výšce cíle

Fyzika těla hráče

Soubor komponent obstarávající vzájemné působení hráče a scény lze označit termíny „fyzika těla“ či „přítomnost těla“ (*body physics, body presence*). Jde především o ošetření důsledků fyzického pohybu hráče, jehož rozsah není pod obecnou kontrolou vývojáře aplikace, což samozřejmě vyvolává celou řadu specifických problémů, které se u běžné desktopové aplikace, počítačové či konzolové hry neobjevují.

Patří sem detekce kolize hlavy hráče, resp. headsetu s elementy scény, např. při vkročení do zdí. V reakci na ni je možné vyvolat částečné či úplné zčernání obrazu, což je poměrně účelný způsob zamezení pohledu hráče skrz neprůhledné překážky a často efektivně zajišťuje automatický, instinktivní ústup hráče zpět do volného prostoru – oblasti validní pro jeho pohyb. Ve VRTK lze využít komponenty `VRTK_HeadsetCollision`, `VRTK_HeadsetFade` a `VRTK_HeadsetCollisionFade` společně s `VRTK_PolicyList`, který specifikuje podmínky pro objekty, které způsobí zčernání obrazu.

Zaznamenávání kolizí headsetu nebo ovladačů je ve spojení s detekcí překážek mezi headsetem a ovladačem výhodná též pro zamezení teleportace skrz překážky. K tomuto účelu slouží skripty `VRTK_TeleportDisableOnHeadsetCollision`, `VRTK_HeadsetControllerAware` a `VRTK_TeleportDisableOnControllerObscured`.

Další aspekt vychází ze skutečnosti, že herní oblast je obvykle tvořena rovným povrchem a sledovaný pohyb hráče se tedy typicky odehrává v rovině, zatímco podklad scény může být vertikálně členitý (terén, schody, ...). Často je vhodné, aby na hráče ve virtuálním prostředí (resp. jeho reprezentaci v podobě kamery ve virtuální scéně) působila gravitace a zároveň aby virtuální tělo hráče kolidovalo s povrchem scény, tedy aby byla výška hlavy nastavována správně relativně k úrovni podlahy. Kombinace těchto působení umožňuje uživateli pohybovat se ve virtuálním prostředí do kopce a z kopce, padat při vykročení za určitou hranu a podobně, zatímco vykonává pouhou chůzi po rovině v rámci dané herní plochy. Vzhledem k tomu, že spojený pád může ve VR působit na uživatele

nepříjemně, může být vhodné od určité hodnoty výškového rozdílu použít podobný mechanismus, jako v případě teleportu – skokovou změnu polohy volitelně oddělenou zatemněním obrazu. To vše zajišťuje `VRTK_BodyPhysics`.

Vzhledem k tomu, že u uvažovaných zařízení je sledována pouze hlava a nikoliv trup či zbytek těla, vyvolává řešení popsané v předchozím odstavci pád ve virtuální scéně jak v případě vykročení za danou hranu, tak v případě pouhého náklonu přes ni. Jedním ze způsobů umožnění pohledu přes hranu bez pádu, resp. odlišení tohoto případu od skutečného překročení hrany, které by mělo vyústit v pád, je odvození pravděpodobné polohy těla z pozice rukou, jež je známá díky sledování ovladačů. Ve VRTK k tomu slouží nastavení *fall restriction* například tak, že k pádu nedojde, dokud je alespoň jeden z ovladačů nad plochou, na které hráč původně stál. K pádu je tedy třeba, aby oba ovladače i headset opustily danou plošinu, což dobře koreluje s reálným stavem, neboť člověk obvykle kompenzuje náklon vysunutím těla (včetně rukou) opačným směrem, zatímco při chůzi má ruce spíše před sebou nebo (v průmětu do horizontální roviny ve směru pohybu) v úrovni hlavy, a v praxi tedy toto řešení obvykle vede k vyvolání právě požadované akce.

Simulator

Při nutnosti rychlého otestování ovládání bez přístupu k VR zařízení je možné použít *simulator*, který lze zapnout namísto použití skutečného SDK. Aplikace je pak ovládána pomocí běžných vstupních zařízení (klávesnice a myš) podobně jako při použití standardního Unity *FPS controlleru* s tím rozdílem, že je navíc možné ovládat i polohu a natočení „rukou“. Vzhledem k tomu, že je v podstatě emulována funkce tří 6-DOF zařízení pomocí 2-DOF a kombinace kláves, není práce se *simulátorem* příliš snadná, přesná, pohodlná ani rychlá a komplexnější simultánní pohyby tak lze provádět špatně či vůbec, nicméně v některých situacích (např. ladění interakcí s GUI) si i s tímto omezeným nástrojem může vývojář vystačit.

4.2 Systém překrývajících se místností

Implementovaný systém zobrazování a řešení fyzikálních interakcí v prostoru s překrývajících se místnostmi pro virtuální realitu demonstruje použití navržené techniky pohybu v praxi a tvoří stěžejní součást praktické části práce. V základu vychází z konceptu přepínání místností popsaného v kapitole 3.1, ovšem podstatně rozvinutého (zejména ve smyslu rozdělení prostoru a oddělení fyziky od vykreslování) tak, aby výsledek umožňoval plynulé procházení otevřeným spojitým překrývajícím se prostorem bez narušení v podobě „přechodových komor“ a související nutnosti uzavření hráče do pevných oddělovacích překážek.

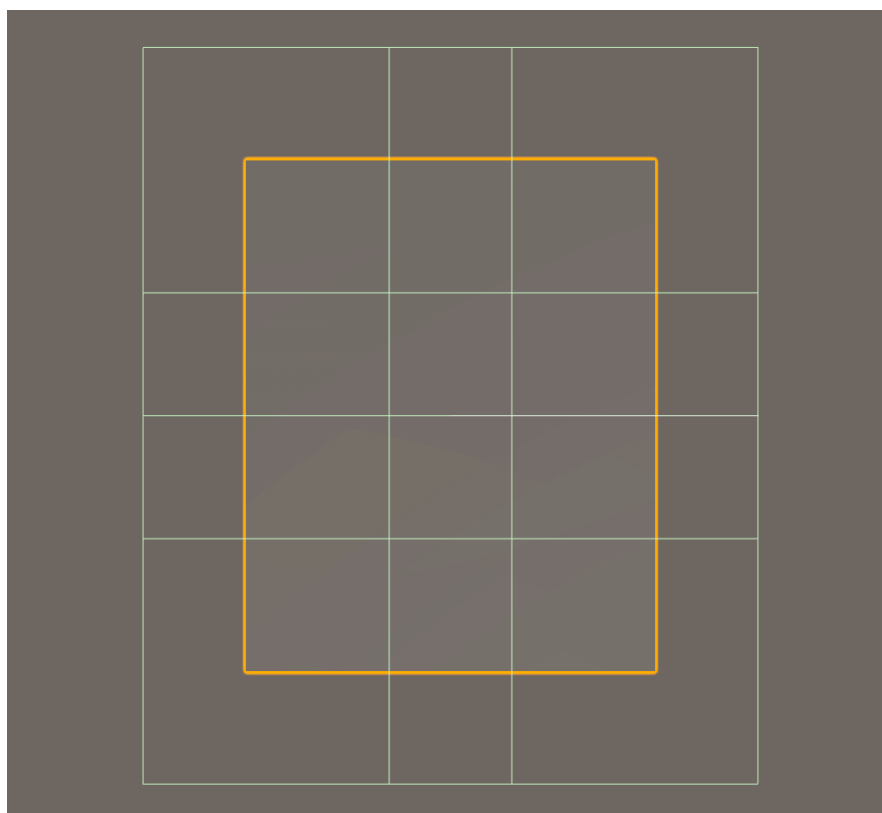
Následující odstavce popisují přístup zvolený k realizaci demonstračního řešení a mohou tvořit výchozí bod či inspiraci při případném budoucím vytváření aplikace s pohybem založeným na obdobných principech.

4.2.1 Rozdělení prostoru

Pro korektní zobrazování i chování fyzikálních kolizí je využito rozdělení prostoru do mřížky menších celků, které lze označit jako „Pole“. Rozsah potřebného fyzického prostoru byl zvolen na $2 \times 2,5$ m, jelikož tvoří vhodný kompromis mezi vlastní dostupností (takový prostor má k dispozici

téměř 40 % uživatelů – viz Tabulka 4 – což tvoří rozumně velký segment trhu pro případné reálné komerční nasazení, jehož výhodnost je předmětem vyhodnocení testování a tedy integrální součástí této práce) a možností umístění virtuálních chodeb dostatečné šířky (pojme 3 chodby široké 60 cm vedle sebe v menším rozměru a 4 ve větším).

Vzdálenost hranic vnitřních Polí mřížky byla po experimentech zvolena na 60 cm, neboť odpovídá minimální šířce chodby pro komfortní průchod (vliv velikosti Pole popsán níže). S tímto rozestupem je pravidelně rozdělena vnitřní část mřížky, vnější Pole pokrývají teoreticky celý zbytek 3D prostoru, viz Obrázek 5 (oranžově je vyznačen předpokládaný herní prostor $2 \times 2,5$ m). Horizontálními řezy není 3D prostor členěn a hranice jsou čistě vertikální, neboť vytvořená scéna nebude obsahovat žádné šikmé (jiné než svislé) zdi a hráč se pohybuje výlučně ve vodorovné rovině (může se pochopitelně skrčit, ale scéna neobsahuje žádné rampy, schody, výtahy apod., které by potenciálně vynucovaly složitější řešení viditelnosti). Celkový počet Polí pokrývajících celý vytyčený prostor je tedy 12.

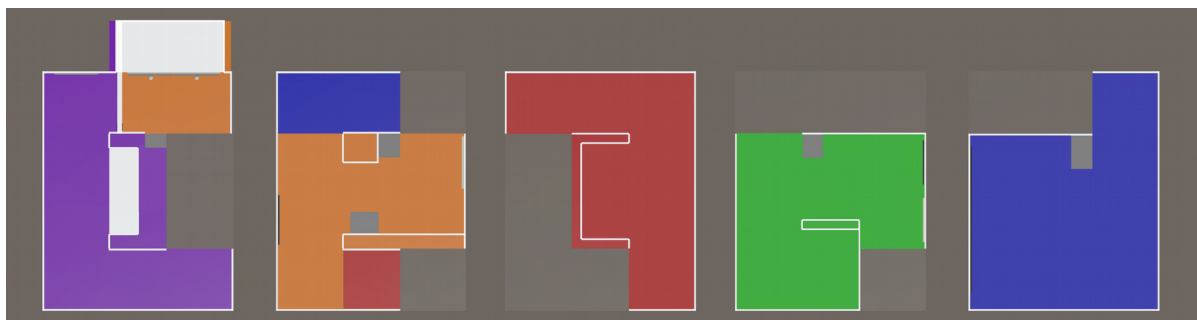


Obrázek 5: Rozdělení prostoru do mřížky Polí

Jedna verze obsahu jednoho Pole je označena jako „Buňka“ (v kódu používán angl. výraz *Cells*). Jednotlivé „verze“ prostoru, které se mohou vzájemně překrývat – sady Buněk, které na sebe navzájem navazují – jsou nazvány „Řezy“ (angl. *Sections*). Označení vychází z možného pohledu na tyto verze 3D prostoru jako na řezy 4D prostoru (případně vrstvy 4D prostoru s diskretní 4. dimenzí; slovo „Vrstva“/*Layer* je však již v Unity používáno v jiném kontextu a mohlo by tedy dojít ke zmatení, byť oba termíny spolu ve vytvořeném řešení dosti úzce souvisí – viz dále).

Hierarchie je tedy následující: scéna je rozdělena na Řezy, které definují obsah prostoru (rozděleného do Polí) pomocí 12 Buněk, které jsou (z hlediska půdorysu prostoru) rozděleny do

mřížky 4×3 , přičemž obsah prostoru definovaný jedním Řezem dohromady tvoří souvislou dílčí část překrývajících se virtuálního prostředí. Buňky odpovídající vnitřním Polím mají pravidelný rozměr 60×60 cm v půdorysu, vnější Buňky mohou být i větší, splňuje-li jejich tvar ostatní omezení – viz 4.2.2. Ne všechny Pole musí mít v rámci jednoho Řezu přiřazenou Buňku, tedy být v daném Řezu obsazené – takové neúplné Řezy jsou vhodné zejména pro zajištění přechodového prostoru pro dynamické objekty, viz 4.2.3. Samostatné číslo Buňky v Rámci Řezu tedy označuje Pole, neboli určitou oblast v 3D prostoru, která je pevně svázána s částí herní plochy a na níž může být ve scéně umístěn různý obsah v závislosti na Řezu, který je v daném Poli aktivní. Dvojice identifikující Pole a Řez označuje konkrétní Buňku, tedy část virtuálního prostoru, konkrétní obsah, který má na dané 3D oblasti (Poli odpovídající oné Buňce) být přiřazen. Obrázek 6 předkládá konkrétní příklad možné podoby Řezů virtuálního prostoru, které jsou pro přehlednost zobrazeny vedle sebe, při použití jsou však ve scéně umístěny všechny na tomtéž místě (přes sebe). Bílé čáry jsou stěny jednotlivých místností.



Obrázek 6: Půdorysy jednotlivých Řezů virtuálního prostoru demonstrační scény

4.2.2 Statické části scény

Prvky scény/místnosti, kterou jsou nehybné, resp. mají vhodně omezený pohyb alespoň v tom smyslu, že nemohou opustit prostor příslušející dané Buňce, jsou nazývány „Statické“ (např. patřičně umístěné otevíratelné dveře či šuplíky jsou v tomto smyslu také brány jako Statické). Tyto jsou na základě jejich pozice v rámci návrhu scény přiřazeny jako potomci *GameObjectu* příslušejícího některé konkrétní Buňce, která je součástí konkrétního Řezu.

Zobrazování

Statické části jsou zobrazovány a skrývány vždy po celých Buňkách. Každá existující instance Buňky spadající do konkrétního Řezu (a tedy odpovídající určité oblasti ve virtuálním prostředí) má přiřazenu alespoň jednu sadu pravidel (data nese *CellSectionsSet*), která definuje, které ostatní Buňky jsou z dané Buňky viditelné, resp. z kterého Řezu se má vzít obsah každého jednotlivého Pole prostoru, aby byla celá scéna zobrazena korektně z pohledu objektu nacházejícího se v dané Buňce v daném Řezu. Takové sady pravidel mohou být v současné implementaci až dvě v jedné Buňce, každá se vztahuje k určité části Buňky oddělené vertikální rovinou (přepínání pravidel zajišťuje *CellRulesSwitcher*); sada viditelných Buněk se tedy může lišit i mezi různými pozicemi v rámci jedné Buňky, nicméně viditelnost každé jednotlivé Buňky může být v každém okamžiku/každé pozici ve virtuálním prostoru jediné buď zapnutá, nebo vypnutá. Je tedy třeba zajistit

v rámci návrhu virtuálního prostředí to, aby z každého jeho místa bylo možné vidět maximálně jednu verzi prostoru (odpovídající jedné Buňce) v každém jednom Poli. V případě, že dojde k tomu, že z určitého místa by měl být vidět v rámci stejného Pole obsah více různých Řezů, a není únosné upravit tvar a rozložení místností tak, aby se tomuto předešlo, je možné umístit do scény neprůhledný objekt vysoký od podlahy až ke stropu, který bude bránit průhledu na všechny části, které by měly obsahovat vzájemně různé Řezy, kromě jedné. Takovým objektem může často být např. regál s neprůhlednými bočnicemi nebo skříňka, čímž je zabraný prostor stále využitelný pro manipulaci s objekty, nicméně jeho umístění snižuje pocitovou otevřenost prostoru a typicky zužuje prostor pro průchod hráče, a proto je třeba snažit se takových zásahů vyvarovat.

Buňka (resp. komponenta `CellScript`) si již v Editoru (při volání metody `Reset` – při změně obsahu Buňky je tedy nutno tuto resetovat) nalezne všechny *Renderery* ležící v hierarchii níže (v potomcích). Volání metod `Show` a `Hide` pak vyvolá zapnutí, resp. vypnutí vykreslování všech těchto objektů.

Z výše uvedeného vyplývá, že veškerá statická geometrie scény by měla být přísně rozdělena do výše popsaných Buněk, mělo by být zajištěno, že žádná část příslušející do jednoho Pole, se nemůže dostat do Pole jiného, tedy například větší skříň, která zasahuje do dvou různých Polí, aniž by se hýbala, by měla být „rozříznuta“ na dva modely podél dané hranice Polí, a tyto modely by měly být následně přiřazeny daným (různým) Buňkám, jak demonstruje Obrázek 7. Nejde o podmínku nezbytně nutnou pro korektní zobrazení scény (závisí na konkrétním tvaru prostoru a případně mohou takové situace být přizpůsobeny zmíněné sady pravidel některých Buněk), nicméně její splnění značně usnadňuje návrh pravidel pro určování viditelné množiny Buněk a snižuje riziko „problíkávaní“ objektů, které by neměly být v dané části virtuálního prostoru vidět. Eventuálně lze s takovým rozsáhlejším statickým objektem pracovat jako s dynamickým, pokud splňuje ostatní relevantní podmínky (viz 4.2.3).



Obrázek 7: Příklad rozdělení geometrie Statického objektu (vlevo) mezi dvě Buňky

Fyzika

Důležitou součástí fyzikálního systému Unity jsou tzv. Vrstvy (angl. *layers*). Objektu s kolizní geometrií je přiřazena konkrétní Vrstva, která zajišťuje, že tento fyzikálně interaguje právě s vhodnou množinou ostatních objektů podle nastavení tzv. kolizní matice (*layer collision matrix*) – ta určuje, které páry Vrstev spolu kolidují. Právě tento koncept je využit k zajištění korektního chování fyzikálních interakcí mezi objekty v implementovaném systému: dané Buňce a všem jejím potomkům je nastavena Vrstva příslušející Řezu, v němž se nachází, přičemž objekty ve vrstvách různých Řezů spolu nekolidují. Tím způsobem je zajištěno, že se Statickými částmi kolidují jen ty Dynamické objekty, které se nacházejí v relevantních částech virtuálního prostoru (viz 4.2.3).

4.2.3 Dynamické objekty

Na rozdíl od Statických částí scény, „Dynamické“ objekty se mohou po virtuálním prostoru volně pohybovat, např. mohou být přenášeny. Z toho důvodu nepřísluší pevně žádné Buňce ani Řezu. To, kde ve virtuálním prostoru se v každém okamžiku nacházejí (tj. určení příslušných Buněk), je však zásadní pro zajištění jejich korektního zobrazování i fyzikálních interakcí (včetně interakcí s objekty hráče, např. umožnění jejich sebrání hráčem v odpovídajících postaveních, a naopak zabránění takovému uchopení v případě vzájemných konfigurací, kdy hráč/ovladač a Dynamický objekt jsou v jiných částech virtuálního prostředí – viz 4.2.4). Každému Dynamickému objektu je nastaven pouze Řez, ve kterém se nachází při spuštění scény, a o další sledování jeho přítomnosti ve virtuálním prostředí se stará třída `CellPresence` v kombinaci se soustavou *trigger colliderů* (komponenty umožňující detekovat kolize nikoliv za účelem provádění fyzikálních simulací, ale pouze k vyvolání události upozorňující na vstup/výstup objektu do/z určitého prostoru) odpovídajících mřížce Polí (viz Obrázek 5). Při vstupu do dalšího Pole je odpovídající Řez odvozen z Řezů Buněk, v nichž se Dynamický objekt již nachází (případně z přiřazeného počátečního Řezu, nenachází-li se Dynamický objekt ještě v žádné Buňce). Dynamický objekt si tedy de facto udržuje vlastní kontext prostoru – informaci o tom, kde ve virtuální scéně se nachází – z něho může zjistit, „jak vypadá“ související okolí, do něhož má eventuálně vstoupit. Jelikož je k těmto detekcím využívána kolizní geometrie objektu a zároveň je na nich založeno určování množiny Buněk, do kterých objekt zasahuje, a tato informace je důležitá pro korektní zobrazení objektu (viz níže), je zásadní, aby *collider* objektu dobře odpovídal jeho zobrazovanému tvaru.

Kvůli zjednodušení algoritmu zajišťujícího validní zobrazování i fyziku je předpokládána maximální velikost jednoho Dynamického objektu nižší než nejmenší velikost Pole, tj. 60 cm. Tím je zajištěno to, že jeden Dynamický objekt může v každém okamžiku zasahovat nejvýše do 4 různých Buněk, které se nacházejí až ve dvou řádcích a až ve dvou sloupcích mřížky Polí.

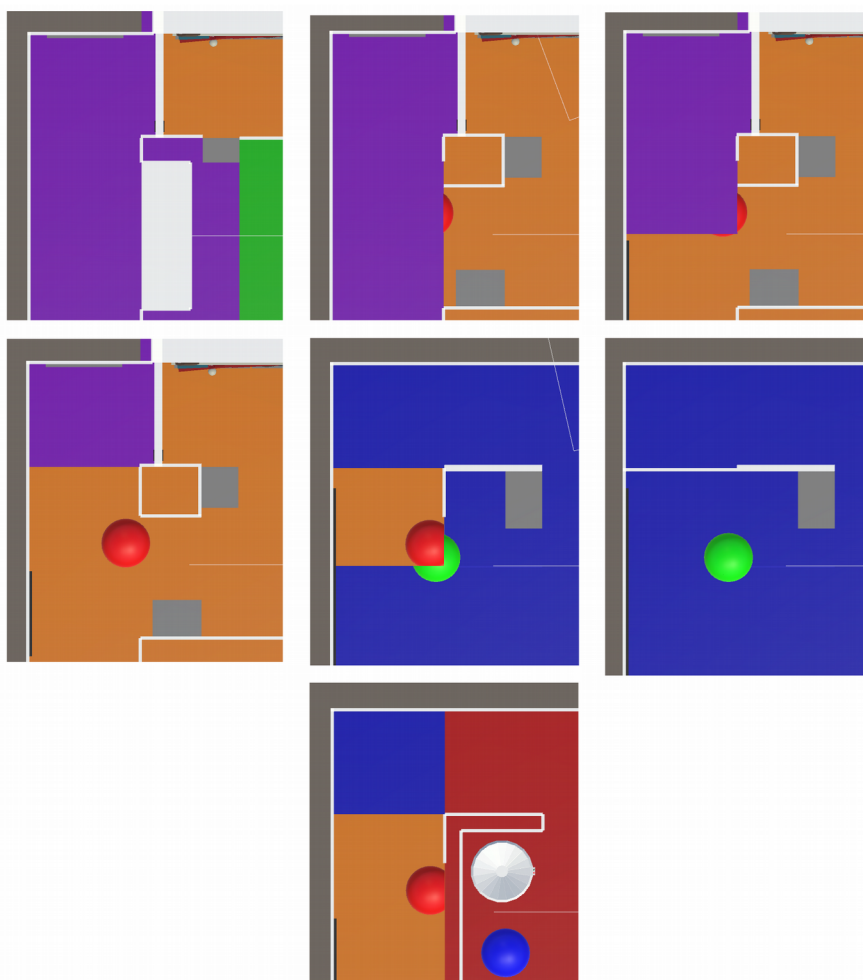
Zobrazování

Jak bylo popsáno výše, Dynamický objekt se může nacházet současně ve více Buňkách, přičemž jejich viditelnost se může vzájemně lišit. Pokud jsou všechny tyto Buňky aktivní (viditelné), pak je zobrazován také celý objekt. Pokud jsou všechny Buňky z hlediska vykreslování deaktivované (skryté), pak není objekt vůbec vykreslován. Pokud však jsou některé z nich zobrazované, zatímco jiné nikoliv, pak je třeba oříznout objekt rovinou či rovinami odpovídající hranici mezi danými Poli/Buňkami, aby bylo zaručeno to, že jeho část nemůže být viděna v místě, jehož aktivní obsah

odpovídá jinému Řezu, než v jakém se objekt v daném Poli nachází. Z údajů o Buňkách, do nichž objekt aktuálně zasahuje, v kombinaci s informacemi o aktivních Buňkách, je tedy v takovém případě potřeba odvodit rovinu a směr ořezu. Vzhledem k tomu, že rozdělení do Polí je pouze v půdorysu a mřížka je zarovnaná s osami, je i ořezová rovina vždy vertikální a zarovnaná s jednou z horizontálních os.

Dynamický objekt může zasahovat až do dvou sloupců a dvou řádků mřížky dělicí dostupný 3D prostor, což znamená, že je současně ořezáván až dvěma rovinami – k tomu dochází v případě, kdy objekt zasahuje do čtyř různých Buněk a právě jedna z nich se svou viditelností liší od ostatních (k situaci, kdy je třeba objekt oříznout dvěma rovinami a zobrazit jej ve dvou Buňkách, které odpovídají Polím umístěným v mřížce diagonálně, by v „rozumném“, dobře navrženém prostoru nemělo docházet a tento případ proto nebyl implementován). Při ořezu dvěma rovinami je tedy navíc ještě nutné určit, zda se má zahodit pouze průnik ořezu získaných rovin, nebo celé sjednocení.

Informace o viditelnosti objektu a případně o způsobu ořezu je aktualizována při každé změně množiny Buněk, které objekt obsahuje (tj. při vstupu nebo opuštění Pole) a při změně množiny aktivních (zobrazovaných) Buněk. První z těchto impulzů detekuje DynamicObject sám, druhý je vyvolán pomocí události, kterou spouští RoomCellContentManager a k naslouchání na niž se každý Dynamický objekt přihlašuje.



Obrázek 8: Příklady různých situací při ořezu Dynamických objektů

Obrázek 8 ukazuje pohled rovnoběžnou projekcí shora na tutéž část scény při sedmi různých pozicích hráče ve virtuálním prostoru. Červená a zelená koule jsou Dynamické objekty a jejich pozice ve virtuálním prostoru je ve všech zobrazených situacích totožná – červená koule se nachází v místnosti s oranžovou podlahou a zelená koule je v modré místnosti, přičemž jejich pozici ve virtuálním překrývajícím se prostředí odpovídá podobná pozice v 3D prostoru scény a tedy i v herním prostoru, tudíž se v těchto prostorech překrývají. Číslujeme-li jednotlivé situace zleva doprava a shora dolů, pak pro červenou kouli platí, že:

- v 1. a 6. je její vykreslování zcela vypnuto,
- v 2. a 7. je ořezávána toutéž rovinou, ale pokaždé v opačném směru,
- ve 3. je ořezávána dvěma rovinami, přičemž je zahozen pouze průnik obou ořezů,
- v 5. je ořezávána stejnými rovinami jako ve 3., ale v opačném směru a je zahozeno celé sjednocení obou ořezů,
- ve 4. není ořezávána.

O vlastní ořez se stará speciální *surface shader* `DynamicObjectShader` využívající vestavěnou funkci `clip`.

Fyzika

Podobně jako v případě zobrazování, i u zajišťování validních fyzikálních interakcí Dynamických objektů je situace složitější než u Statických, byť opět vychází z totožného základu.

Nacházejí-li se všechny Buňky, do nichž Dynamický objekt zasahuje, v tomtéž Řezu, je tomuto objektu přiřazena scéna příslušející danému Řezu obdobně jako v případě objektů Statických. Pokud se však naopak Řezy jednotlivých Buněk navzájem liší, je třeba zajistit, aby objekt korektně kolidoval s objekty ve všech zastoupených Řezech, v návaznosti na tvar vytvořené scény s překrývajícím se místnostmi však nebylo při implementaci nutné uvažovat příslušnost objektu do více než 2 různých Řezů současně. Je nicméně nutné vytvořit a přiřazovat „přechodové“ Vrstvy pro všechny dvojice Řezů, které spolu „sousedí“ (mezi nimiž lze přecházet), přičemž tyto vrstvy mají nastaveny aktivní kolize jak s objekty ve vrstvách obou Řezů, tak v oné přechodové vrstvě samé.

Pokud by ovšem byly Řezy úplné (pro všechna Pole by byla nastavena příslušná Buňka), znamenal by vstup do nového Řezu vstup do nového, dosud objektem neobsazeného Pole, přičemž tomuto by odpovídala jiná Buňka původního Řezu, do níž neměl v logice virtuálního prostoru objekt vstoupit (jestliže virtuální prostor vedl do nového Řezu, nemohl zároveň vést do téhož Pole původního Řezu). Z pohledu vykreslování toto netvoří problém – v rámci `CellPresence` daného objektu by byla díky pravidlům Buněk a souvislosti virtuálního prostoru pro dané nové Pole přiřazena Buňka z nového Řezu a tudíž by byl na daném Poli objekt viditelný právě tehdy, když by byla viditelná nová Buňka. Jenže odpovídající Buňka původního Řezu může obsahovat objekty (ať již Statické či Dynamické) s kolizní geometrií, s nimiž by logicky předmětný objekt vůbec neměl fyzikálně interagovat (neboť v takové buňce vůbec nemá být), zatímco objekt obecně může být na příslušném „kolizním“ místě ve 3D prostoru (nachází se v rámci téhož Pole) a zároveň má nastavenou přechodovou Vrstvu, která s takovými objekty koliduje (pro umožnění kolizí s objekty v původní Buňce, z níž vyšel). Proto je třeba používat neúplné Řezy, kdy právě Pole umožňující přechod mezi dvěma Řezy musí mít „svou“ Buňku jedině v jednom z nich. Z obdobného důvodu je žádoucí, aby byly Řezy tvořeny takovou sadou Buněk, která reprezentuje souvislý virtuální prostor, resp. aby Řez tvořil dílčí konzistentní celek virtuálního prostoru, tj. aby například nebyla virtuální chodba sahající

přes 3 Pole v jednom Řezu přerušena Buňkou jiné místnosti, zatímco prostřední blok chodby se nacházel v řezu zcela jiném.

4.2.4 Objekty hráče

Standardní funkci VR zařízení (převod sledovaného pohybu do scény, interakce s objekty pomocí ovladačů, zobrazování na HMD, ...) zajišťuje VRTK v kombinaci s příslušným SDK. Pro zajištění použitelnosti systému překrývajícího se prostoru je však třeba navíc ošetřit další specifické chování spojené s přítomností hráče ve virtuální scéně.

Ovladače

Funkční interakce (tj. v případě vytvořené demonstrační scény prakticky výlučně uchopování a přenášení) s virtuálními objekty obstarává z hlediska ovladače soubor komponent `VRTK_ControllerEvents`, `VRTK_InteractTouch`, `VRTK_InteractGrab` a `VRTK_InteractUse`, na straně daného objektu potom `VRTK_InteractableObject`. Jelikož se však virtuální prostor v rámci herní plochy překrývá, je nutné zajistit, aby hráč mohl interagovat pouze s těmi objekty, které odpovídají jeho pozici ve virtuálním prostředí, nikoliv se všemi, které jsou v určité vzdálenosti od ovladačů v kontextu vymezeného 3D hracího prostoru; resp. je třeba vyloučit možnost uchopování (příp. další akcí) objektů, které neodpovídají pozici reprezentace hráčových rukou ve virtuálním prostoru. Nelze k tomu ale již využít pouhé přiřazení určité Vrstvy, jako tomu bylo u Dynamických objektů, neboť pohyb ovladačů, resp. rukou hráče po scéně nelze omezit virtuální zdí, která by je zastavovala, bránila v jejich průchodu do jiných částí scény mimo „povolené“ souvislé kusy virtuálního prostoru. Uvažovaná zařízení neposkytují vývojáři mechanismus, kterým by mohl zastavit provádění uživatelova fyzického pohybu, a prosté neprovedení tohoto pohybu ve scéně vytváří snad více problémů, než kolik řeší – narušuje pohlcení uživatele do virtuálního prostředí, vede k potenciálnímu „zapomenutí“ virtuálních rukou v jiné části scény (pokud by byl jejich pohyb blokován, aniž by si toho uživatel všiml) atd. Zároveň není ideální zabránit veškerým interakcím při každém kontaktu s takovou překážkou – takové řešení by znamenalo, že např. sebrání malého objektu ležícího na podlaze v rohu místnosti by bylo dosti náročné a pravděpodobně frustrující, jelikož *collider* ovladače by v takovém postavení často zasahoval do některé ze zdí. Je tedy nutné počítat s tím, že ovladače mohou být prostrčeny skrz zeď a nacházet se tak v různých počtech a kombinacích Řezů, které by byly (v případě validní konfigurace scény) u Dynamických objektů nepřipustné. Není praktické tvořit Vrstvy reflektující tyto kombinace jen k ošetření tohoto stavu. Naopak jsou vytvořeny dvě Vrstvy speciálně pro ovladače (jedna pro každý z nich) a kolizní matice je dynamicky upravována v závislosti na přítomnosti ovladačů v jednotlivých řezech. Pro obecně korektní řešení je nutné zapnout kolize nejen s Vrstvami příslušejícími všem obsazeným Řezům, ale i se všemi „přechodovými“ Vrstvami, které by měly s těmito Řezy kolidovat.

K získání informace o tom, do kterého Pole ovladač vstoupil nebo které opustil, je použita sada *trigger colliderů*, shodně jako v případě Dynamických objektů. Příslušné Řezy v těchto Polích ale nejsou určovány z vlastního kontextu prostoru (Buněk, v nichž se ovladače již nacházejí), ale pouze na základě aktivních Buněk, tedy podle kontextu z pohledu headsetu (viz níže). Jde teoreticky o kompromisní řešení, neboť v případě držení interaktivního objektu a určité sekvence pohybů rukou po scéně se může stát, že se „ruka“ a objekt dostanou do různých Řezů. Nicméně v naprosté většině

reálných případů chce uživatel interagovat právě s tím, co vidí, přičemž typicky zvedá předměty, které jsou relativně blízko – maximálně v sousedním Poli – a stav tohoto Pole v podstatě musí být stejný z kontextu headsetu i případného kontextu ovladače. Při přenášení zase hráč typicky pohybuje hlavou i rukama relativně shodně a relativně blízko u sebe. To znamená, že praktický dopad omezení vyvolaných popsáním řešením je minimální. Alternativní řešení založené čistě na vlastním kontextu prostoru (tj. určování Řezů nově navštívených Polí jako u Dynamických objektů) by bylo mnohem méně vhodné, neboť hráč by pak mohl např. rukou obkroužit stěnu a vrátit se do původní pozice v rámci herní plochy ve fyzickém prostoru, zatímco virtuální reprezentace ruky by se nacházela ve zcela odlišné části virtuálního prostředí – každá z rukou by tak mohla být virtuálně úplně jinde, přestože by měl hráč ruce u sebe, což je pravděpodobně značně matoucí stav, do kterého by se hráč zřejmě nejčastěji dostal omylem a bylo by potenciálně dosti náročné se z něj zotavit (tj. dopravit obě „ruce“ opět do téhož virtuálního prostoru). S využitím tohoto přístupu by navíc muselo být zvlášť ošetřeno strkání ovladačů do virtuálních zdí, neboť v takovém případě by mohla být pravidla jednotlivých obsazených Buněk (kvůli pohybu mimo spojitý virtuální prostor) vzájemně nekonzistentní (mohla by si protřečít) a určení platného kontextu prostoru by tedy rozhodně nebylo přímočaré.

Headset

GameObject reprezentující headset ve scéně slouží pro účely popisovaného systému zejména k vyvolání aktualizace množiny aktivních (viditelných) Buněk (tj. obsah kterého Řezu má být zobrazen v každém Poli), přičemž je využívána kombinace komponent již uvedených v předchozích kapitolách o rozdělení prostoru a Statických a Dynamických objektech. Nejzásadnější komponentou pro zajištění této funkcionality je *RoomCellContentManager*, který právě informace o aktivních Buněkách udržuje a spravuje, včetně specifikace částí prostoru, kterou jsou aktivní při prvotním spuštění scény, což ovlivňuje, kde ve virtuálním prostoru hra začne. Pole, v němž se hráč nachází, je tentokrát (na rozdíl od Dynamických objektů) určeno pouze výpočtem, jehož vstupem je pozice headsetu ve scéně, nikoliv soustavou *trigger colliderů* – z hlediska zobrazení není podstatný seznam všech Polí/Buněk, do nichž headset zasahuje, neboť může být v každém případě zobrazena pouze jedna verze scény, a reprezentace headsetu jediným bodem je tedy dostačující. Při přejití do jiného Pole je získána jemu příslušející Buňka z množiny aktivních, a následně je soubor nových aktivních Buněk nastaven dle seznamu Buněk odpovídající její platné sadě pravidel (*CellSectionsSet* – viz 4.2.2). Předpokládá se přitom samozřejmě spojitost virtuálního prostoru a korektní nastavení sady pravidel všech Buněk – pokud by totiž hráč vstoupil do Buňky, která na předchozí zpětně nenavazuje (tj. v její sadě pravidel je pro předchozí Pole jiný Řez než ten, z něhož hráč přišel), hráč se opačným pohybem nedostane do té části virtuálního prostoru, z níž původně přišel.

I v případě korektně specifikovaných pravidel všech Buněk však může k popsané situaci dojít, pokud hráč projde místem, které není určeno k průchodu (typicky zdi), přičemž v takové hranici Polí není virtuální prostor souvislý a takový přechod v něm není validní. K prevenci těchto situací je třeba detekovat vstup headsetu do zdi (nebo podobné překážky – Statického objektu), která odděluje takové nespojitě části virtuálního prostoru, a při jeho zjištění vypnout veškeré změny množiny aktivních Buněk. Za tím účelem je headsetu přidán *trigger collider*. Předpokládá se, že kolizní geometrie relevantních Statických objektů je správně nastavena nezávisle na tomto problému – její správný tvar je nutný už k zabránění obdobného průchodu Dynamických objektů. Je nutné zároveň zajistit, aby headset tímto způsobem nereagoval na vstup do překážek, které nejsou součástí aktivních Buněk

(průchod mezi dvěma sousedními Poli může být validní v jednom Řezu a nevalidní v jiném – to, zda má být povolen, závisí na Řezu, v němž se headset nachází, tedy právě na množině aktivních Buněk, neboť ta je tvořena právě Buňkami, které jsou aktuálně relevantní z pohledu headsetu). Princip, kterým je toto zajištěno, je totožný jako ten, který se používá pro zajištění, že ovladače interagují právě s relevantními objekty – headset má vytvořenou vlastní Vrstvu, přičemž její kolizní maska („řádek“ v kolizní matici) je nastavena podle aktivních Řezů Polí, do nichž headset zasahuje.

Průhledy skrz všechny neprůhledné objekty jsou obvykle nežádoucí a měly by být ve VR aplikacích nějakým způsobem ošetřeny – fyzický pohyb uživatele nelze omezit, ale zároveň je často nepřijatelné, aby mu prostrčení hlavy umožnilo vidět např. skrz zamčené dveře. Jelikož jde o obecný problém, týkající se mnohých VR aplikací, VRTK nabízí jeho řešení v podobě kombinace skriptů `VRTK_HeadsetCollision`, `VRTK_HeadsetFade` a `VRTK_HeadsetCollisionFade`, sloužící k detekci kolizí s headsetem a ztmavení obrazu v reakci na ně. Pokud však v implementovaném systému vejde hráč do zdi, je obecně potřebné, aby se vrátil do toho Pole, z něhož do zdi vešel, předtím, než je možné pokračovat (znovu zapnout přepínání prostoru). Je vhodné na tuto skutečnost hráče upozornit a pomoci mu najít příslušné místo, zejména pokud „uvnitř zdi“ učinil rozsáhlejší pohyb a mohl by mít potíže najít místo, z něhož vyšel. K tomu byl vytvořen vlastní `HeadsetCollisionScript` a `HeadsetPositionRestrictionManager` s několika pomocnými skripty, které zabrání zobrazení obsahu scény (překryjí jej neprůhlednou plochou), vykreslí kruh označující pozici Pole, na které je třeba dojít, a umístí před headset šipku směřující na tento kruh (tato je viditelná i v případě, že je hráč otočen na opačnou stranu). Z místa, na které se má hráč vrátit, je navíc vyslán zvuk, který by měl hráči ještě více usnadnit orientaci. Při skrývání scény se navíc rozlišuje, zda má hráč hlavu pouze v objektu, nebo v hranici části virtuálního prostoru. Tyto případy jsou odlišeny *tagem* příslušného *GameObjectu*. Pokud je hlava hráče pouze v „obyčejném“ objektu, není nutné zakazovat úpravy stavu aktivních Buněk a nutit hráče k návratu do specifického pole – hráč může nadále procházet prostorem bez přidání omezení, dochází pouze ke ztmavení/překrytí obrazu.

4.2.5 Omezení vytvořeného řešení a možnosti jeho vylepšení

Vytvořené řešení úspěšně zprostředkovává uživateli vjem pohybu po virtuálním prostoru, který se v rámci 3D prostoru, do něhož je umístěn, překrývá, což bylo hlavním cílem stanoveným pro praktickou část této práce. Přesto takové řešení není zcela dokonalé – skrývá v sobě jistá omezení, kterých by si měl být vývojář při jeho použití vědom, a nabízí prostor pro další rozšíření vlastní funkcionality, které by bylo vhodné provést před jeho uplatněním v komerčním produktu.

Velikost Pole/Buňky

Použití popsaného řešení přináší důsledky pro návrh virtuálního prostředí. Základním parametrem, který je nutné vhodně zvolit a následně brát v úvahu během dalšího návrhu prostředí, je velikost Polí, resp. vzdálenost hranic Polí ve vnitřní části mřížky. Pro snazší návrh podoby scény/tvaru místností by teoreticky bylo výhodné mít menší Pole/Buňky, neboť řešení zobrazování by mohlo být modulárnější – vykreslování scény by bylo zapínáno a vypínáno po menších celcích. Na druhou stranu by však menší Pole/Buňky znamenaly nutnost rozdělení geometrie Statických objektů na menší části a pravděpodobně také pracnější specifikaci přepínacích pravidel. Z tohoto pohledu se tedy jako zajímavé jeví nějaké adaptivní podrozdělování prostoru jen tam, kde je to potřeba. Vzhledem k úzké

návaznosti zobrazování i nastavení fyziky Dynamických objektů na velikosti Polí/Buněk by však bylo nutné podstatně změnit i jejich implementaci, která doposud předpokládá tuto velikost jako neměnnou a uniformní.

Využití Vrstev

Jelikož je správné fungování fyzikálních simulací a detekce vstupu/setrvání/opuštění objektu v části virtuálního prostoru v rámci implementovaného řešení založeno na výpočtu kolizí fyzikálním systémem Unity při současném intenzivním využívání jeho Vrstev, je třeba reflektovat fakt, že těchto Vrstev je k dispozici omezené množství. Celkový počet Vrstev, s nimiž fyzika Unity pracuje, je 32, k podpoře virtuálního prostoru s překrývajícími se místnostmi však nelze použít všechny – 8 z nich je rezervováno pro vestavěné Vrstvy Unity, dále je třeba počítat s Vrstvami pro každý z ovladačů a pro headset, případně další Vrstvy pak mohou být nutné pro zajištění správného chování určitých herních objektů. Počet dostupných Vrstev má potenciální omezující dopad na rozsah virtuálního prostoru, neboť jedna Vrstva je nutná pro každý Řez a každý možný přechod mezi dvěma Řezy (přechod možný v kontextu virtuálního prostoru, nikoliv nutně pro každou dvojici Řezů). Počet potřebných Vrstev je tedy dán počtem Řezů a celkovou kardinalitou jejich sousednosti. Pro lepší představu je možné uvést, že k realizaci vytvořené demonstrační scény, která se skládá z 5 Řezů s možností cyklického průchodu místnostmi a jednou „sleponou“ místností, je využito celkem 10 Vrstev pro jednotlivé Řezy a přechody mezi nimi.

Pokud by bylo třeba vytvořit rozsáhlejší prostor, přičemž by už nebylo možné využít další Vrstvy, existuje řešení v podobě „recyklace“ Vrstev stávajících – lze-li část virtuálního prostoru uzavřít, oddělit od prostoru ostatního, pak je možné všechny *GameObjecty*, které se v něm nacházejí (včetně Dynamických objektů), zcela vypnout a Vrstvy příslušející daným Řezům a přechodům mezi nimi použít pro Řezy jiné. Je tím pochopitelně narušena kontinuita fyzikálních simulací v rámci tohoto prostoru, dopad této skutečnosti by však v praxi neměl být nijak kritický; eventuálně by bylo možné pokračovat v průběhu fyziky přesunutím celého tohoto podprostoru na jiné místo ve 3D prostoru scény. Příkladem vhodného zaručení oddělení daného prostoru od zbytku může být nástup uživatele do uzavřené kabiny virtuálního výtahu a jeho odjezd do jiného patra virtuálního prostoru.

Zvýraznění oblasti

Současný systém zvýraznění oblasti, do níž má hráč dojít, je tvořen pouze kruhem umístěným na příslušné Ploce. Uživatel může vstoupit do dané plochy takovým způsobem, že jeho headset je uvnitř nějakého objektu, pročež nedojde k obnovení standardního zobrazení scény, což může být pro hráče matoucí. Bylo by tedy vhodné navíc zobrazit půdorys či obrysy alespoň Statických částí scény, které se nacházejí v bezprostředním okolí tohoto bodu, což by mohlo dát uživateli lepší informaci o skutečné pozici, do které je třeba se dostat.

Související překryv scény, který zabraňuje jejímu zobrazení, navíc aktuálně překrývá i modely ovladačů při použití SteamVR (při použití OVR jsou avatary rukou zobrazeny korektně). Není-li scéna překryta při spuštění hry, kdy často uživatel teprve hledá ovladače v reálném prostoru s využitím pohledu „skrze“ headset, není dopad tohoto problému nijak zásadní, nicméně pro budoucí obecné použití by bylo žádoucí jej ošetřit.

Odrazy světla

Pro zobrazení okolního prostředí na povrchu lesklých objektů je v Unity možné použít vestavěný systém *reflection probe*, které vytvářejí příslušné *cubemap*. Ve scéně s překrývajícím se prostorem by teoreticky mělo být možné je téměř přímo použít, pravděpodobně bude nutné jen vytvořit jednu *reflection probe* pro každou Buňku (popř. skupinu Buněk, které jsou přepínány společně) a Dynamickým objektům je přidělovat podle aktuální pozice těchto objektů ve virtuálním prostředí (nikoliv pouze dle jejich 3D souřadnice). Bylo by však vhodné implementovat automatický mechanismus spouštění výpočtu jednotlivých *cubemap* tak, aby byl brán v úvahu obsah virtuálního prostoru relevantní pro danou Buňku (podobně jako jsou samostatně načítány všechny *renderery* Statických objektů dané Buňky). Experimenty s takovým řešením nebyly z časových důvodů v rámci této práce provedeny.

Stíny

Vykreslování jednotlivých objektů je zapínáno a vypínáno podle toho, zda oblast virtuálního prostoru, v níž se nacházejí, je potenciálně viditelná. Jejich stíny však často mohou sahat i mimo oblast, na základě které je toto rozhodování prováděno, a v takovém případě by potenciálně docházelo při změně stavu jejich vykreslování ke skokovému přeblikávání těchto jejich stínů. Vzhledem ke způsobu fungování vrhání a zachytávání stínů v Unity nelze ani definovat omezenou skupinu objektů, jejichž stíny by neměly vytvářet popsany problém (např. ovladače), a této skupině stíny zapnout, zatímco ostatním objektům by byly vypnuty, neboť každý objekt může pouze „připisovat“ stín, a tedy dopadající stín je vrhán i za objekt (jakoby skrz něj), který sám stíny nevrhá, bez ohledu na to, zda tento objekt stíny zachytává či nikoliv. Vrhání stínů v oblasti využívající překrývajícím se virtuální prostor tedy tvoří samostatný netriviální problém, který vyžaduje komplexní, celkové řešení. To bude pravděpodobně zahrnovat výpočet vlastních *shadow map* pro každou Buňku virtuálního prostoru. Ve vytvořené demonstrační scéně jsou jednoduše všechny stíny vypnuty.

Prostorový zvuk

Jak již bylo naznačeno v kapitole 3.1, v popsáném překrývajícím se prostoru je vzdálenost dvou bodů potenciálně větší, než je Euklidovská vzdálenost bodů ve 3D, které jim přísluší. Prostorový zvuk je zároveň jedním z cenných vodítek, které mohou uživateli pomoci orientovat se ve virtuálním prostředí a usnadňovat jeho zaměření na konkrétní herní objekt. Proto by bylo vhodné zajistit korektní šíření zvuku v takovém virtuálním prostředí. Tato problematika je sama o sobě potenciálně značně komplexní a nebyla v rámci praktické části práce vůbec řešena. Případnou implementaci takového systému by mohlo být (zejména pro zajištění kompatibility se standardními audio komponentami Unity) vhodné založit na *Audio Spatializer SDK*.

Nový vykreslovací řetězec Unity

V průběhu vytváření praktické části této práce byla vydána nová verze Unity (verze 2018.1 a 2018.2), která představila nový, výrazně přepracovaný způsob práce s vykreslovacím řetězcem – *Scriptable Render Pipeline*. Některé z možností, jež nabízí, by se mohly ukázat jako užitečné pro zobrazování překrývajícím se prostoru nebo optimalizaci jeho vykreslování. Bylo by tedy vhodné prozkoumat tyto možnosti a případně upravit stávající implementaci.

4.3 Implementované testovací prostředí

V souladu s původním konceptuálním návrhem (kapitola 3.2.2) je testovací prostředí rozděleno na páry, kde jeden testovací pár je složen ze dvou podobných scén. Obě tyto scény předkládají uživateli tentýž úkol, ale mohou být mírně přizpůsobeny dané technice pohybu, přičemž v jedné verzi je k pohybu hráče po virtuálním prostoru použita umělá technika pohybu založená na výběru cíle – teleport – a v druhé fyzická chůze. Celkem byly vytvořeny 3 takovéto páry, každý z nich je zaměřen na porovnání jiného aspektu pohybu – první pár porovnává dopady přítomnosti a pohybu v různých typech prostoru (překrývající se a klasický 3D prostor), další dva páry představují aktivity, které by bylo možné do takových prostorů zasadit, a srovnávají přívětivost jejich vykonávání s použitím jednotlivých technik pohybu, tedy vlastně slouží ke zjištění, zda pohyb pomocí chůze, který by byl umožněn sestavením překrývajícího se prostoru, nabízí v daných případech nějaké výhody oproti teleportu. Záměrně přitom nejde o „náhodný výběr“ aktivit, jež lze provozovat ve virtuálním prostředí, ale spíše o snahu pokrýt škálu různých typů pohybů, které by případně bylo nutné vykonávat v aplikacích, u nichž by bylo možné uvažovat o implementaci využívající fyzickou chůzi.

Uživatelské preference v rámci jednotlivých testovacích párů jsou sledovány nezávisle. Podoba a účel těchto testovacích párů jsou představeny v následujících odstavcích, páry jsou přitom nazývány podle úkolu, který je v nich uživateli zadán.

Hledání koulí

Uživatel má ve scéně složené z pěti barevně odlišených místností najít tři barevné koule a umístit každou z nich na podstavec v místnosti, která odpovídá barvě dané koule. Verze využívající chůzi používá implementovaný systém překrývající se místností. Verze s teleportem je naopak tvořena obvyklým, nijak se nepřekrývajícím prostorem. Místnosti v této verzi jsou rovněž větší (blíže velikosti běžných obytných místností), neboť to použitý způsob pohybu dovoluje, zatímco v případě překrývajícího se prostoru je nutné vměstnat všechny místnosti na malý 3D prostor a jejich rozměr je tudíž omezený. Sousednost i přibližný tvar místností jsou však mezi verzemi shodné. Ekvivalentní je též vybavení místností, přítomné dekorace a úroveň grafického zpracování.

Scéna s chozením demonstruje možnost realizace výše uvedeného konceptu překrývajícího se prostoru – skládá se z pěti místností (viz Obrázek 6), přičemž celková virtuální plocha dostupná pro chození tvoří přibližně 350 % specifikované rozlohy herní oblasti. Překrývání prostoru je tedy využito poměrně intenzivně. Účelem tohoto testovacího páru je zejména zjistit, zda se v takovém prostoru může uživatel pohodlně pohybovat, zda se v něm oproti standardnímu prostoru dobře orientuje a zda mu není nepříjemná větší stísněnost takového prostoru, která je použitému přístupu do jisté míry vlastní. Prostor, do něhož jsou scény situovány/stylizovány, bylo zvoleno se snahou o co největší neutralnost – jde v podstatě o místnosti (poněkud nedovybaveného a neurčitého) bytu, přičemž takový druh prostoru je implicitně omezen, ale zase ne dostatečně na to, aby sám o sobě plně ospravedlňoval či vysvětloval drobnost pokojů ve scéně využívající chůzi.

Ve scéně s překrývajícími se místnostmi je uplatněn pokus o využití případného prostoru dostupného nad rámeček minimálního požadovaného rozsahu (tj. nad $2 \times 2,5$ m). Ten spočívá v umístění regálu tak, aby byl ve fyzickém prostoru mapován těsně za hranici minimální plochy, přičemž tento je zaplněn interaktivními Dynamickými objekty, které však hráč nijak nepotřebuje ke splnění úkolu, v případě vytvořené scény jsou těmito „výplňkovými“ objekty knihy. To znamená, že

má-li uživatel k dispozici větší než minimální prostor, může jej využít – je schopen z regálu vyjmout další objekty, které může zkoumat. Pokud takovým prostorem nedisponuje, uživatel tím z pohledu možnosti dokončení hry není nijak znevýhodněn a virtuální prostor alespoň vypadá větší. Je ovšem zásadní, aby byl tento prostor zpočátku dostatečně vyplněn nedůležitými objekty podobné velikosti, jako jsou ty důležité k dokončení mise – tak si hráč nemůže omylem umístit důležitý objekt mimo svůj dosah, neboť pokud byl schopen odebrat původní objekt, který prostor mimo požadovanou plochu zabíral, aby tam následně mohl umístit klíčový objekt, pak je zákonitě též schopen dostat tento obdobným způsobem opět ven. Dalším aplikovaným mechanismem využívajícím prostor mimo minimální specifikovaný rozsah jsou zásuvky („šuplíky“) – při jejich zasunutí mohou zabírat ony samy i jejich obsah prostor ležící potenciálně vně hrací plochy, přičemž v rámci vytyčeného prostoru stačí mít jejich rukojeti, čímž je garantována možnost přitažení tohoto obsahu do dosahu hráče, aniž by v zavřeném stavu zabíraly užitečný herní prostor. Podoba obou těchto prostředků v demonstrační scéně ilustruje Obrázek 9.



Obrázek 9: Příklad využití prostoru mimo minimální specifikovanou hrací oblast

Mačkání tlačítek

Uprostřed velké místnosti se nachází kvádr, na jehož povrchu jsou rozmístěna tlačítka. Úkolem je pohybovat se kolem tohoto kvádru a zamáčknout všechna tlačítka. Obě scény jsou shodné, jen v případě verze s teleportem je hráč při spuštění umístěn dále od kvádru a *camera rig* (tj. *GameObject* obsahující reprezentaci hráče a nakalibrované herní plochy) je do scény umístěn záměrně pootočeně kolem vertikální osy (v půdorysu úhlopříčně oproti kvádru), což znamená, že k obejití kvádru by byla potřeba podstatně větší herní plocha – to dohromady vyvolává nutnost hráče teleportovat se a tedy využívat tuto techniku pohybu namísto chození. Pokud by nebylo přikročeno k těmto opatřením, hráč by mohl kvádr obejít stejně jako v případě druhé scény z testovacího páru, neboť má k dispozici tentýž prostor a jeho pohyb nelze nijak omezit. Alternativou by bylo zvětšení kvádru, to však poněkud narušuje vzájemnou podobnost a tedy i porovnatelnost obou scén.

Tato dvojice scén testuje zejména uživatelskou schopnost dopravit se s využitím příslušné techniky pohybu na určité místo, které mu umožní stisk dosud nevyřízených tlačítek. V případě teleportu musí hráč buď věnovat větší úsilí namíření do správné pozice, která mu dovolí dosáhnout na velké množství tlačítek, nebo se musí častěji teleportovat po menších kouscích, přičemž buď používá ruku střídavě k mačkání a směřování teleportu, nebo provádí každou rukou jinou činnost; zároveň se však může teleportem posunout o větší kus za kratší dobu.

Chytání ovoce

Kolem hráče jsou rozmístěna čtyři děla, která směrem dovnitř scény střílejí jablka. Jeho úkolem je ovoce chytit nejlépe hlavou, případně rukou a následně si je přemístit do hlavy („sníst“). V obou variantách jablka dopadají do prostoru scény odpovídajícího definované herní ploše, jediným rozdílem je to, že v případě chůze je střelba jablek navíc nastavena tak, že každý ovocný projektil překoná polovinu scény – tím by mělo být zaručeno, že jablko je v rámci definované herní plochy v „chytatelné“ výšce, zatímco při teleportu je pro chycení některých jablek nutné dostat se mimo prostor odpovídající herní ploše a tedy použít teleport. To ovšem znamená, že pro chytání jablek v teleportovací verzi je prakticky nutné vykonat rozsáhlejší pohyb, byť oblast dopadu jablek je ekvivalentní. To teoreticky znamená, že verze s teleportem je náročnější, což by narušovalo porovnatelnost obou scén; porovnání by však stále mělo být validní za předpokladu přijetí domněnky, že používá-li hráč teleport, musí se z místa chytání jednoho jablka každopádně teleportovat do místa chytání dalšího, přičemž náročnost provedení takového posunu je buď nezávislá na vzdálenosti, která má být překonána, nebo je dokonce snazší pro vzdálenosti střední (vyšší desítky centimetrů až jednotky metrů) než pro vyloženě malé (jednotky až nízké desítky centimetrů) – tuto domněnku by však bylo vhodné ověřit dalším testováním.

K usnadnění udržování uživatelského přehledu o okolních jablkách, jejich výšce nad podložkou, vzdálenosti od hráče a směru jejich letu je v obou verzích použita kombinace audiovizuálních efektů: jablko za letu vydává zvuk, který je zpracován jako prostorový (k tomu je využit dostupný *Audio Spatializer*); za letícím jablkem je kreslena stopa (využívá *trail renderer*); na zemi je zobrazen pro každé letící jablko kruh označující místo jeho dopadu, přičemž tento se postupně zmenšuje a zneprůhledňuje s klesající výškou jablka nad zemí.

Předmětem testování je v tomto případě schopnost uživatele dosáhnout rychle konkrétního místa virtuálního prostoru, aniž by tento cíl závisel na jeho vůli, zatímco je třeba si udržovat dobré povědomí o okolním prostoru a objektech, které se v něm pohybují. Předpokládanou nevýhodou

teleportu je nespojitý pohyb, který může narušovat orientaci uživatele, zejména v případě, kdy se tento teleportuje o kousek blíže či dále, než původně zamýšlel. Zároveň je při použití teleportu nutno sledovat specifikovaný cíl pohybu (dole) a zároveň padající jablka (nahore). Při chytání jablek rukama je navíc nutné střídat tuto akci se zaměřováním teleportu, případně používat každou ruku k jiné akci. Výhodou teleportu může být opět možnost rozsáhlejšího pohybu za kratší čas.

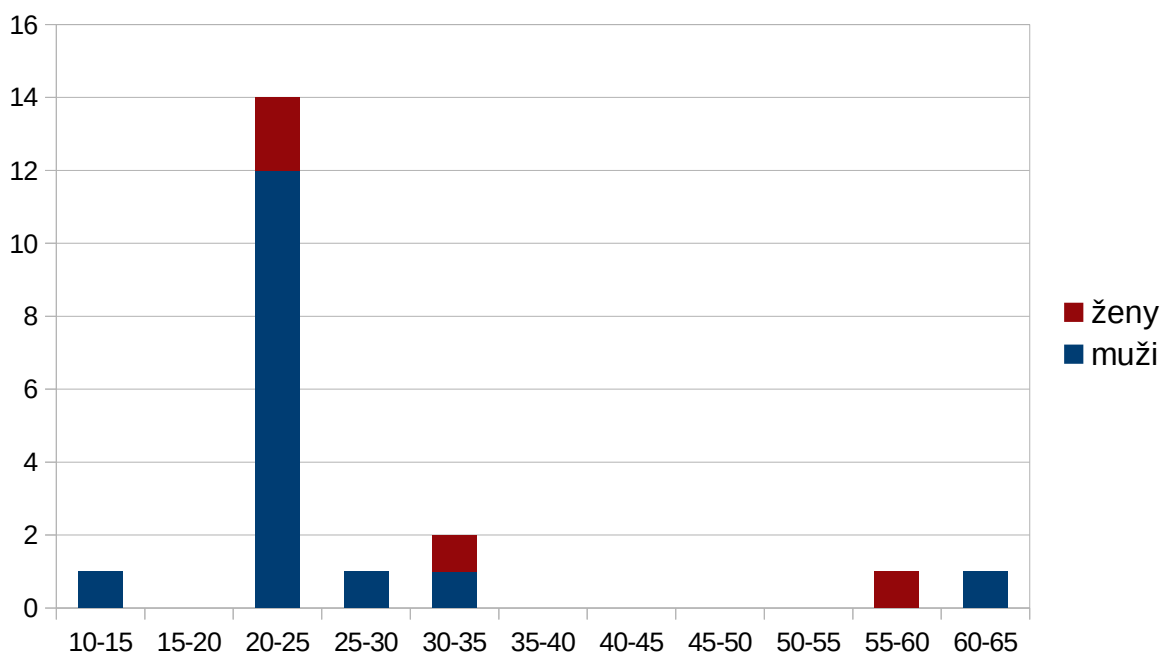
5 Testování

Za účelem porovnání praktických vlastností vytvořeného řešení vůči obvyklému pohybu založeném na teleportu bylo provedeno testování vycházející z návrhu popsaného v kapitole 3.2 Hlavním sledovaným parametrem byla preference jedné z verzí scén respondenty.

5.1 Průběh testování

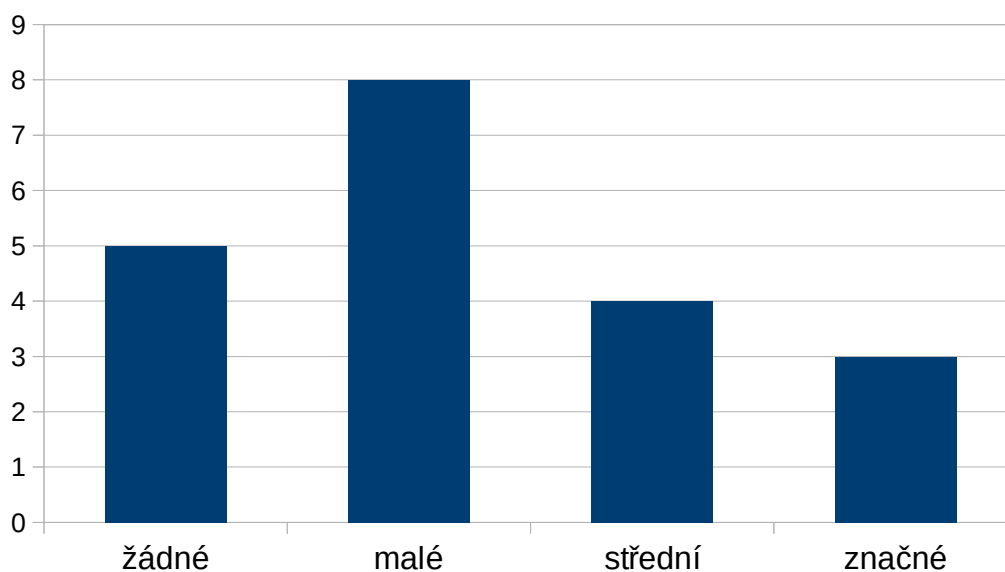
Testování probíhalo převážně (13 vzorků) v laboratoři O105 v Centru výpočetní techniky (CVT) Fakulty informačních technologií Vysokého učení technického v Brně na zařízení HTC Vive, kde velikost vytyčeného herního prostoru byla téměř $3 \times 2,5$ metru, nicméně prostor umožňoval hráči vstoupit i mimo vyznačenou plochu, aniž by mu hrozil náraz do nějakých fyzických překážek. Zbylá část testování (7 vzorků) byla provedena v domácnosti autora práce na zařízení Oculus Rift s ovladači Touch, přičemž dostupný prostor těsně splňoval požadovaných $2,5 \times 2$ metrů a bezprostředně za jeho hranicemi se nacházely fyzické objekty. V obou případech bylo testovací prostředí spouštěno na počítači s dostatečným výpočetním/grafickým výkonem a všechny testovací scény tak byly správně zobrazovány při stabilních 90 snímcích za sekundu (FPS).

Testování se zúčastnilo celkem 20 osob (2 však z vlastních časových důvodů nedokončily všechny testovací páry), naprostou většinu z nich tvořili studenti FIT, jednalo se tedy převážně o muže ve věku kolem 25 let, jak je patrné z Grafu 1.



Graf 1: Zastoupení pohlaví a věkových kategorií účastníků testování

Svoje předchozí zkušenosti s virtuální realitou označila více než polovina účastníků jako žádné či malé, přičemž nejčastěji se dostali do kontaktu s VR v podobě mobilních headsetů využívajících „chytré“ telefony, menší část z nich navštívila VR hernu. Rozložení skupiny testovacích uživatelů z hlediska aktivních zkušeností s použitím VR ukazuje Graf 2 – „malá“ úroveň zkušeností znamená, že si uživatel někdy před testováním pobyt ve virtuální realitě pouze vyzkoušel v délce trvání do cca 2 hodin; „střední“ znamená několik jednotek hodin strávených ve VR, např. během návštěv VR herny; „značné“ zkušenosti má uživatel dlouhodobě opakovaně používající VR headset. Jak je z Grafu 2 patrné, z hlediska zkušeností s aktivním používáním VR není složení skupiny testovacích uživatelů vzhledem k záměru testování zcela ideální, neboť základní otázkou, která má být dle návrhu (kapitola 3.2.1) testováním zodpovězena, je to, zda použití popsané techniky pohybu a řešení prostoru vede k lepšímu subjektivně hodnocenému uživatelskému zážitku oproti běžným VR aplikacím a zda tedy může tvořit konkurenční výhodu při případné nabídce takové aplikace uživatelům virtuální reality. Z toho pohledu by bylo ideální provádět testování na osobách tvořících potenciální zákaznickou základnu takové případné aplikace, tedy zejména na majitelích relevantních zařízení pro VR, jejichž zkušenosti s aktivním používáním virtuální reality vesměs spadají do kategorie „značné“. Na druhou stranu lze díky této skladbě uživatelů vzájemně porovnat preference skupin s různou úrovní zkušeností.



Graf 2: Zkušenosti účastníků testování s aktivním používáním VR

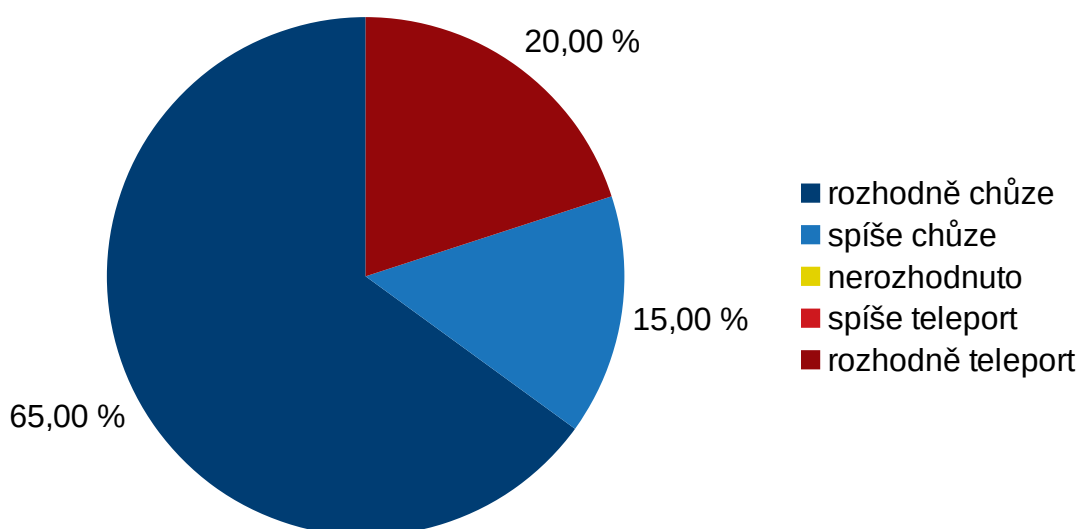
Účastníci testování byli nejprve seznámeni se způsobem ovládání a poté jim byly postupně spouštěny jednotlivé testovací páry, přičemž pořadí scén v rámci páru bylo určeno pseudonáhodně, aby se zamezilo vlivu tohoto pořadí v agregovaném vyhodnocení. Hráč si mohl nechat spustit danou scénu znovu, pokud o to měl zájem. Plnění úkolů probíhalo pod dohledem autora práce, který na začátku každé scény vysvětlil předložený úkol a případně slovně pomáhal uživateli řešit problémy, do nichž se při průchodu scénou dostal, přičemž tyto byly zřejmě způsobovány především absencí zkušeností s používáním VR aplikací (typicky se jednalo o zmatení v reakci na vstup do virtuální překážky), neboť zkušenější uživatelé se do takových problémů téměř nedostávali. Po úspěšném splnění úkolu v obou scénách daného páru proběhl s každým účastníkem neformální rozhovor, jehož základním účelem bylo zjistit verzi, kterou uživatel upřednostňoval, a důvody této jeho preference.

5.2 Vyhodnocení testování

Jednotlivé testování páry byly vyhodnoceny samostatně, kromě toho byl vyhodnocen i vliv pořadí spuštění scén v rámci páru a předchozích zkušeností uživatelů s virtuální realitou na jejich preference.

Hledání koulí

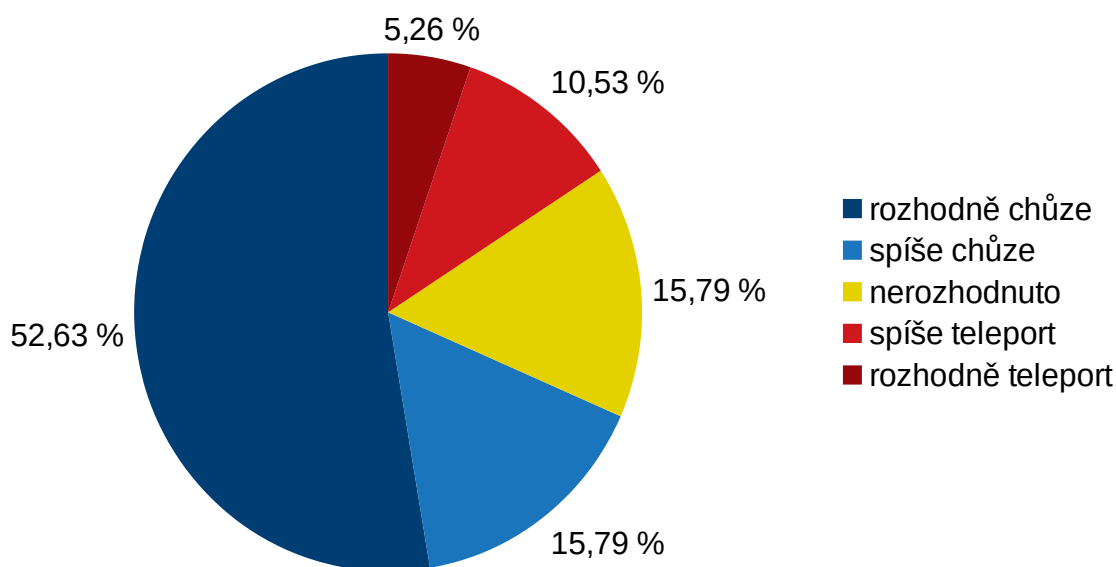
Graf 3 zobrazuje celkové preference verzi scény s hledáním koulí. Lze nahlédnout, že převážná část uživatelů při testu shledala chození příjemnějším, zajímavějším či zábavnějším způsobem pohybu. Mezi nejpozoruhodnější z výpovědí účastníků testování patří názor, že chůze vede k intenzivnějšímu zážitku a silnějšímu vnímání okolního virtuálního prostředí (2 uživatelé konstatují, že při chození věnovali více pozornosti obrazům, které jsou v prostoru rozmístěny, a lépe si je prohlédli, což oba hodnotili kladně). Dalším postřehem je to, že chůze nezaměstnává ruce a ty tak mohou být soustředěně používány na provádění jiných akcí (není potřeba např. držet předmět a zároveň směřovat teleport). V odůvodnění kladného hodnocení chůze se nejčastěji objevovala originalita řešení pomocí prostoru s překrývajícími se místnostmi, vyšší míra vtažení do virtuálního prostředí a přirozenost takového ovládání. Někteří uživatelé dokonce sami navrhovali možné případy, kdy by podobný způsob pohybu a práce s prostorem mohl tvořit vlastní zábavnou součást hry, přičemž taková mechanika by s pomocí teleportu nebyla realizovatelná – např. chůze po úzké římsě/lávce/okraji srázu nebo nutnost podlezení překážky. Jako důvod preference teleportu je naopak jmenována nepřirozenost překrývajícího se prostoru vedoucí ke ztrátě orientace hráče a riziko zamotání se do kabelů při chůzi.



Graf 3: Preference – Hledání koulí

Mačkání tlačítek

Jak je patrné z dat Grafu 4, v případě mačkání tlačítek hovoří nasbírané hodnocení opět ve prospěch chůze, rozdíl však zde není tak značný a objevuje se též relativně početná skupina nerozhodnutých uživatelů. To mohlo být do jisté míry způsobeno tím, že v laboratoři, v níž probíhalo testování, je dostupná herní plocha natolik velká, že při úvodním teleportu do vhodné polohy mohl být kvádr s tlačítky v podstatě celý obejit navzdory uplatněným opatřením pro zajištění nutnosti používání teleportu popsaného v kapitole 4.3. Uváděné argumenty ve prospěch chůze se týkaly zejména jednoduchosti, přirozenosti a přesnosti tohoto přístupu, proti teleportu pak zazněla nepříjemnost přesné specifikace cíle teleportu při pohybu na krátké vzdálenosti. Ti, kterým lépe vyhovoval pohyb teleportem, jej naopak označovali jako rychlejší a zábavnější, případně si pochvalovali fakt, že jim při pohybu tolik nepřekáží kabeláž.

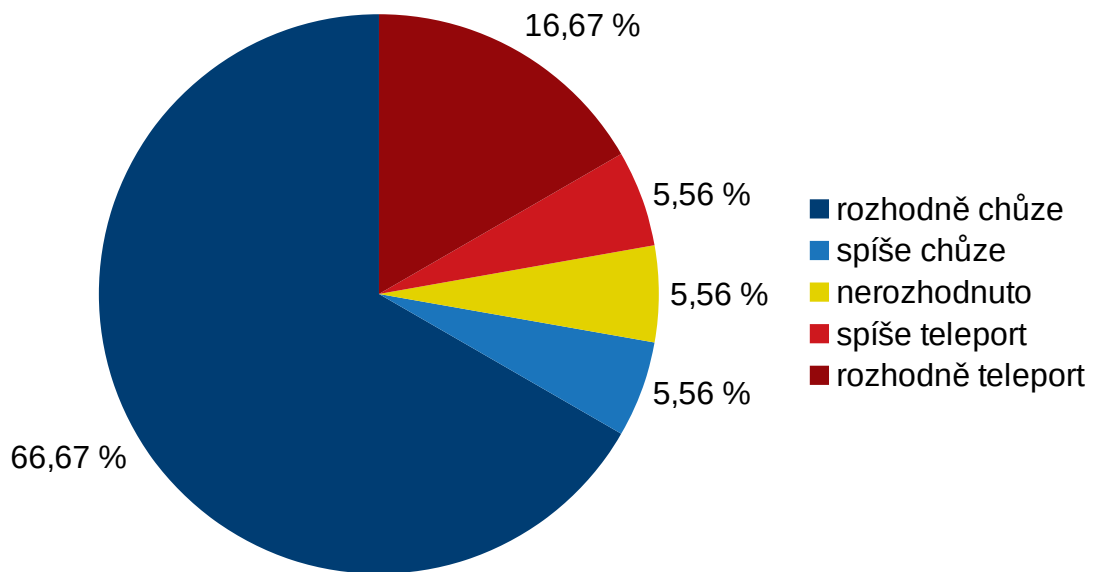


Graf 4: Preference – Mačkání tlačítek

Chytání ovoce

U chytání ovoce je poměr velikosti skupiny preferující chození k té, která favorizuje teleport, velmi podobný jako u mačkání tlačítek, názory jsou však obecně polarizovanější – počet nerozhodnutých je nižší a větší část lidí má silný názor pro jednu či druhou z technik pohybu, což je znázorněno na Grafu 5. U tohoto úkolu bylo pozoruhodné, že někteří ze zúčastněných jej shledávali velmi zábavným, zatímco jiným byl přímo nepříjemný, a přitom se tyto jedinci v rámci těchto skupin neshodují vůbec, mírou zkušeností ani výslednou preferencí konkrétní techniky pohybu. Nejčastějším argumentem pro použití chůze byla ztráta orientace a náročnost nutného současného sledování různých prvků scény (tj. padajících jablek v horní části a zároveň zaměřování cíle teleportu dole) v případě teleportu. Příznivci teleportu poukazovali opět na jeho rychlost a zábavnost, přičemž argumentovali tím, že si lze na takový způsob ovládnání rychle zvyknout, zautomatizovat jej a poté již k žádné znatelné ztrátě orientace podle nich nedochází. Jeden z hráčů se při chození bál, že narazí do

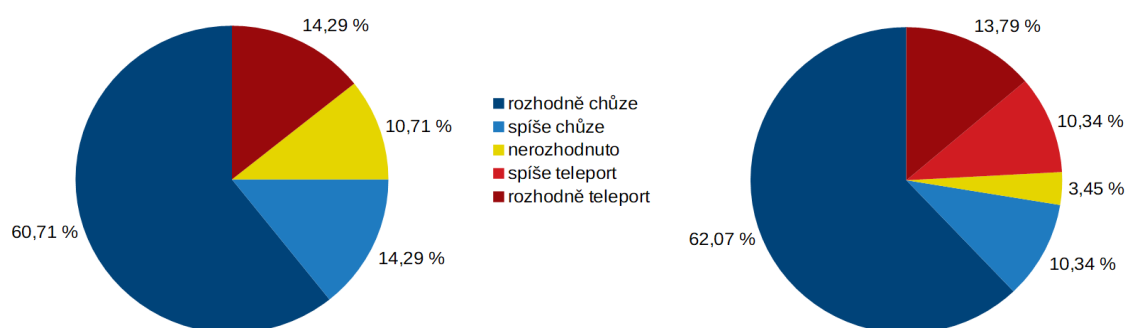
nějakého fyzického objektu. Jiný zase vypověděl, že při chůzi a současném rozhlížení se po letícím ovoci ztrácí rovnováhu, a proto se mu teleport zamlouval více.



Graf 5: Preference – Chytání ovoce

Vliv pořadí testování

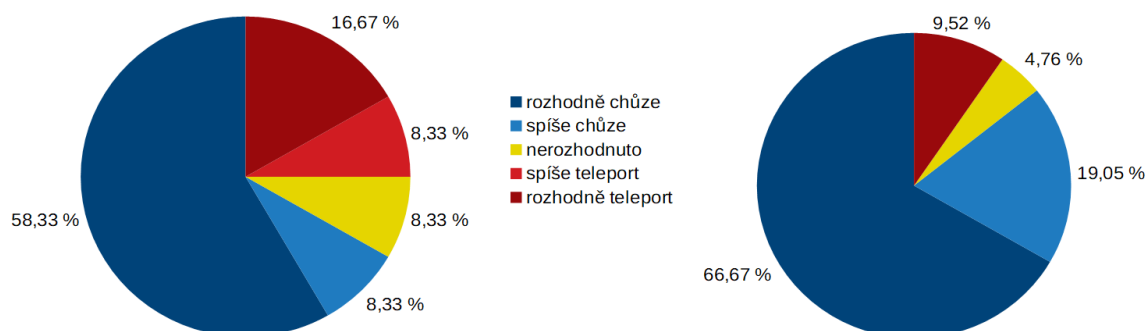
Pořadí scén v rámci testovacích párů bylo jednotlivým účastníkům testování přiřazováno nahodile. Graf 6 porovnává hodnocení získaná tehdy, byla-li nejprve spuštěna scéna využívající teleport (vlevo), s případy, kdy hráč vyzkoušel nejprve verzi s chůzí (vpravo). Data ukazují, že vyjádřené preference se v závislosti na pořadí spuštění nijak podstatně neliší a uživatelé jím tak pravděpodobně nebyli ovlivněni.



Graf 6: Vliv pořadí testovacích scén na hodnocení

Vliv zkušeností respondentů

Vzhledem k rozmanitosti úrovně zkušeností respondentů s používání virtuální reality může být zajímavé porovnat celkové preference méně zkušených (žádné či malé zkušenosti) proti zkušenějším (střední či značné zkušenosti). Taková data prezentuje Graf 7. Z těchto dat vyplývá, že zkušenější uživatelé (na Grafu 7 vpravo) častěji preferovali chůzi. Tento závěr může být překvapivý, neboť nejčastěji skloňovanou výhodou přímého použití fyzické chůze je intuitivita jejího používání k pohybu po virtuální scéně – pro začátečníky by tak měla na rozdíl od teleportu představovat způsob pohybu, který již důvěrně znají, pročez by se jim hypoteticky měla vykonávat snáze. Možným důvodem vyšší oblíbenosti chůze u zkušených hráčů však zase může být to, že jsou si vědomi značného rozšíření teleportu ve stávajících aplikacích, a proto více oceňují originalitu alternativního řešení.



Graf 7: Vliv zkušeností uživatelů na hodnocení

Interpretace výsledků testování

V souhrnu lze konstatovat, že získaná data jednoznačně nasvědčují tomu, že v případě aktivit podobných těm, které jsou prováděny ve scénách testovacího prostředí, by mohlo být výhodné nahradit umělou techniku pohybu (teleport) přirozenou chůzí, přičemž použití prostoru s překrývajícími se místnostmi může být vhodným způsobem jak toho dosáhnout. Pravděpodobně tedy existují aplikace, u kterých by založení pohybu na principu chůze v překrývajícím se virtuálním prostoru mohlo vést k zážitku, který je uživateli hodnocen jako lepší než v případě řešení alternativních, což by mohlo tvořit konkurenční výhodu oproti podobným dostupným aplikacím.

Výsledky testování samozřejmě nelze validně interpretovat tím způsobem, že by snad chůze systémem překrývajících se místností byla univerzálně použitelná a za všech okolností lepší, než je pohyb pomocí teleportu. Zároveň je třeba brát v úvahu omezenou velikost skupiny účastníků testování a jejich možné zaujetí mj. kvůli časté osobní známosti s autorem práce. Z toho důvodu by bylo velmi vhodné ověřit obecnou platnost získaných dat dalším, rozsáhlejším testováním.

5.3 Možnosti navazujícího testování

Výstupem výše uvedeného testování je tedy kromě poměru preferencí jednotlivých scén v rámci testovacích párů i seznam aspektů, které se uživatelům líbily nebo jim naopak vadily; je tak do jisté míry možno interpretovat, proč uživatelé dávají přednost té či oné technice pohybu. Podle původního návrhu měla na tuto výše popsanou „kvalitativní“ část testování navazovat část „kvantitativní“, tj.

sběr podstatně většího množství „neinterpretovaných“ dat – pouhých hodnocení jednotlivých scén, ať již ve formě binárního vyjádření preference, určení bodu na stupnici udávajícího míru preference či podobně, avšak bez následného hloubkového rozhovoru a tedy zjišťování důvodů, které za touto volbou stojí. Tato část testování měla být provedena na vzorku uživatelů skutečně odpovídajících záměru testování, tj. na pravidelně aktivních uživatelích VR. Tím by bylo možné ověřit obecnější platnost vyvozených závěrů, případně to, zda jsou validní i pro takovouto cílovou skupinu. Testování mělo být provedeno vydáním aplikace obsahující tytéž testovací páry v obchodě Steam, kde by byla dostupná uživatelům zdarma.

Vinou zdržení při implementaci, délce trvání schvalovacího řízení při zakládání vývojářského účtu a vydání hry a nutnosti implementovat další specifické komponenty, které nejsou příliš relevantní pro praktickou část této práce – jako je uživatelské rozhraní k popisu ovládání a úkolů, případně k zadávání hodnocení, dále vlastní sběr dat, *achievements* sloužící k motivaci uživatelů k dokončení úkolů a zadání hodnocení, formulace souhlasu se zpracováním dat apod. – nebyla aplikace vydána včas, aby v termínu stanoveném pro odevzdání práce byla k dispozici takto nasbíraná uživatelská data. Provádění tohoto výzkumu je však plánováno i po odevzdání práce.

6 Závěr

Cílem práce bylo navrhnout, implementovat a vyhodnotit způsob, jakým by bylo možné lépe využít herní prostor, který je k dispozici při použití současných zařízení pro virtuální realitu. Za tím účelem byly prozkoumány obecné principy fungování i reálné příklady takových vstupně-výstupních zařízení s důrazem na ty, které umožňují uživateli provádět sledovaný pohyb v rámci fyzického herního prostoru. Různé přístupy k řešení pohybu uživatele v rámci virtuálního prostředí byly rovněž prostudovány, na jejich základě byl poté navržen konkrétní techniky využívající sledování reálného pohybu uživatele jako jediný způsob změny jeho pozice ve virtuální scéně. Následně proběhlo hlubší seznámení s nástroji, které je možné použít k jeho realizaci. S jejich pomocí byl navržený způsob využití prostoru implementován, přičemž bylo vytvořeno testovací prostředí sloužící k porovnání vlastností zvoleného přístupu s aktuálně nejrozšířenější alternativní technikou pohybu.

Vytvořené řešení nabízí praktickou možnost uplatnění fyzické chůze v rámci omezeného herního prostoru k pohybu po relativně rozsáhlé virtuální scéně. Právě díky využívání takové bezprostřední a přirozené techniky pohybu by mělo být možné zajistit intuitivnější ovládání, hlubší ponoření uživatele do virtuálního světa i zaujetí případných zákazníků originalitou takového přístupu. Tomu odpovídá výsledek uživatelského hodnocení, kdy tento koncept úspěšně obstál oproti referenční technice pohybu. Představuje tedy obecně potenciálně atraktivnější alternativu k obvyklým přístupům k tvorbě VR aplikací. Existují nicméně specifické aspekty, které nebyly v rámci praktické části práce adresovány – tyto tvoří možnosti dalšího rozšíření předložené implementace, přičemž nejzásadnější z nich byly analyzovány a popsány v rámci samostatné části textu.

Literatura

- [1] How did virtual reality begin? *Virtual Reality Society* [online]. [cit. 2018-01-16]. Dostupné z: <https://www.vrs.org.uk/virtual-reality/beginning.html>
- [2] LAVIOLA, JR., Joseph J., Ernst KRUIFF, Ryan P. MCMAHAN, Doug A. BOWMAN a Ivan POUPYREV. *3D User Interfaces: Theory and Practice*. 2nd edition. Hoboken, NJ: Pearson Education, 2017. ISBN 978-0-13-403432-4.
- [3] LAVALLE, Steven M. *Virtual Reality* [online]. University of Illinois: Cambridge University Press, 2017 [cit. 2018-01-16]. Dostupné z: <http://vr.cs.uiuc.edu/vrbook.pdf>
- [4] NOVOTNÝ, Miroslav. *VR Interaktivní aplikace*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Milet.
- [5] HOFFMAN, David M., Ahna R. GIRSHICK, Kurt AKELEY a Martin S. BANKS. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*. 2008, **8**(3), 33. DOI: 10.1167/8.3.33. ISSN 1534-7362. Dostupné také z: <http://jov.arvojournals.org/article.aspx?doi=10.1167/8.3.33>
- [6] *VIVE™ | VIVE Virtual Reality System* [online]. [cit. 2018-01-21]. Dostupné z: <https://www.vive.com/us/product/vive-virtual-reality-system/>
- [7] *VIVE Pro | The professional-grade VR headset* [online]. [cit. 2018-01-21]. Dostupné z: <https://www.vive.com/us/product/vive-pro/>
- [8] Oculus Rift. *Oculus* [online]. [cit. 2018-01-21]. Dostupné z: <https://www.oculus.com/rift/>
- [9] SteamVR Play Area Size Stats. *SteamVR Developer Hardware General Discussion* [online]. [cit. 2018-01-16]. Dostupné z: <https://steamcommunity.com/app/358720/discussions/0/350532536103514259/?ctp=3#c1482109512310264401>
- [10] SUMA, E. A., Z. LIPPS, S. FINKELSTEIN, D. M. KRUM a M. BOLAS. Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture. *IEEE Transactions on Visualization and Computer Graphics* [online]. 2012, **18**(4), 555-564 [cit. 2018-02-13]. DOI: 10.1109/TVCG.2012.47. ISSN 1077-2626. Dostupné z: <http://ieeexplore.ieee.org/document/6165136/>
- [11] BRESTIČ, Tomáš. *Nástroje pro tvorbu VR aplikací*. Brno, 2017. Projekt do předmětu Grafická a zvuková rozhraní a normy. Vysoké učení technické v Brně, Fakulta informačních technologií.

Seznam příloh

Příloha 1: DVD obsahující zdrojový projekt v Unity, spustitelnou verzi testovacích scén a video zachycující průchod demonstrační scénou