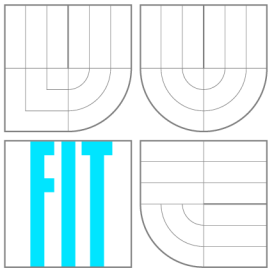


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# UŽIVATELSKÉ ROZHŘANÍ PRO PRÁCI S POČÍTAČEM VE VIRTUÁLNÍ REALITĚ

USER INTERFACE FOR WORK WITH COMPUTER IN VIRTUAL REALITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL PAZDERA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2016

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

## Zadání diplomové práce

Řešitel: **Pazdera Michal, Bc.**

Obor: Počítačové sítě a komunikace

Téma: **Uživatelské rozhraní pro práci s počítačem ve virtuální realitě**  
**User Interface for Work with Computer in Virtual Reality**

Kategorie: Uživatelská rozhraní

### Pokyny:

1. Seznamte se s problematikou pohybu a navigace v počítačovém virtuálním světě (VR), možnostmi zařízení pro snímání akcí uživatele a testování uživatelských rozhraní (UI).
2. Navrhněte UI komponenty pro práci s počítačem s využitím VR a vybraných ovládacích zařízení. Zaměřte se na základní úlohy uživatele s počítačem.
3. Vyberte vhodné technologie a dostupné knihovny a navržené komponenty implementujte.
4. Proveďte testy na uživateli, vyhodnoťte a aktualizujte navržené rozhraní komponent.
5. Realizujte demonstrační aplikaci.
6. Vytvořte plakát a krátké demonstrační video reprezentující Vaše řešení.

### Literatura:

- B. Shneiderman, C. Plaisant. Designing the user interface: strategies for effective human-computer interaction, Addison-Wesley Computing, ISBN-10: 0-321-53735-1, 2009
- R. Hartson, P. Pyla. The UX Book: Process and Guidelines for Ensuring a Quality User Experience, Morgan Kaufmann, ISBN-10: 0123852412, 2012
- Dále dle pokynu vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 částečně bod 3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

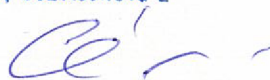
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
612 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato práce se zabývá možnostmi ovládání počítače ve virtuální realitě. Jejím cílem je vytvořit uživatelské rozhraní, které v ní bude umožňovat pracovat s počítačem. Nejprve studuje možnosti zařízení pro snímání akcí uživatele a jejich využití pro ovládání a způsoby interakce ve virtuální realitě. Na základě získaných informací se zaměřuje na způsoby interakce, které využívají pro ovládání snímání rukou uživatele. Práce řeší problém výběru a manipulace s virtuálními objekty. Představuje tři způsoby interakce, které testuje v rámci experimentu a následně hodnotí.

## Abstract

This thesis explores various ways of controlling computer in virtual reality. The aim of this thesis is to create a user interface which would allow the user to control computer in virtual reality. First it explores the possibilities of detecting users actions using sensors and its usability for controls and various interaction techniques in virtual reality. Based on the information gathered about the described topics it focuses on various ways of interactions using hands as controllers. This thesis tackles the issue of selecting and manipulating virtual objects. It introduces the design of three interaction techniques, which are then tested and evaluated.

## Klíčová slova

3D uživatelské rozhraní, ovládání počítače, virtuální realita, interakce, virtuální objekty, Oculus Rift, LeapMotion

## Keywords

3D user interface, computer interaction, virtual reality, interaction, virtual objects, Oculus Rift, LeapMotion

## Citace

PAZDERA, Michal. *Uživatelské rozhraní pro práci s počítačem ve virtuální realitě*. Brno, 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Beran Vítězslav.

# Uživatelské rozhraní pro práci s počítačem ve virtuální realitě

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Michal Pazdera  
24.května 2016

## Poděkování

Rád bych na tomto místě poděkoval vedoucímu mé diplomové práce panu Ing. Vítězslavovu Beranovi, Ph.D., za poskytnutí vybavení, metodické vedení, osobní přístup a mnoho cenných rad, kterých jsem při realizaci této práce využil.

© Michal Pazdera, 2016.

*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

|  |           |
|--|-----------|
| <b>1 Úvod</b>  | <b>2</b>  |
| <b>2 Hardware a technologie umožňující virtuální realitu</b>       | <b>4</b>  |
| 2.1 Přehled zařízení vytvářejících virtuální realitu . . . . .     | 5         |
| 2.2 Zařízení pro snímání vstupních akcí uživatele . . . . .        | 7         |
| <b>3 Ovládání počítače a uživatelská rozhraní</b>                  | <b>10</b> |
| 3.1 3D uživatelská rozhraní . . . . .                              | 11        |
| 3.2 Způsoby interakce ve virtuální realitě . . . . .               | 14        |
| <b>4 Testování uživatelských rozhraní</b>                          | <b>19</b> |
| 4.1 Lidské faktory ovlivňující práci člověka s počítačem . . . . . | 19        |
| 4.2 Způsob provádění vědeckého výzkumu . . . . .                   | 22        |
| 4.3 Jak připravit a provádět HCI experiment . . . . .              | 23        |
| <b>5 Návrh řešení</b>  | <b>26</b> |
| 5.1 Definice cíle a představení problému . . . . .                 | 26        |
| 5.2 Výběr zařízení pro ovládání rozhraní . . . . .                 | 27        |
| 5.3 Analýza problému a možnosti jeho řešení . . . . .              | 28        |
| 5.4 Koncept systému . . . . .                                      | 30        |
| 5.5 Analýza a způsob řešení jednotlivých částí problému . . . . .  | 31        |
| <b>6 Realizace</b>   | <b>35</b> |
| 6.1 Zvolené nástroje pro implementaci a jejich integrace . . . . . | 35        |
| 6.2 Implementace navrženého rozhraní . . . . .                     | 36        |
| 6.3 Návrh a příprava experimentu . . . . .                         | 40        |
| 6.4 Experiment . . . . .   | 43        |
| 6.5 Výsledky experimentu a vyhodnocení rozhraní . . . . .          | 44        |
| 6.6 Aktualizace rozhraní na základě výsledků experimentu . . . . . | 51        |
| 6.7 Demonstrační aplikace . . . . .                                | 51        |
| <b>7 Závěr</b>   | <b>52</b> |
| <b>Literatura</b>  | <b>53</b> |
| <b>Přílohy</b>   | <b>55</b> |
| Seznam příloh . . . . .  | 56        |
| <b>A Obsah CD</b>  | <b>57</b> |

# Kapitola 1

## Úvod

Virtuální realita zažívá v posledních několika letech nevídaný rozkvět. I když se o ní začalo poprvé mluvit již v roce 1957, vždy byla spíše záležitostí daleké budoucnosti a zůstávala odsouzena pouze k laboratornímu použití v rukou několika nadšených výzkumníků. Zájem o virtuální realitu se historicky vyvíjel ve vlnách. Pokaždé, když byla objevena nějaká nová významná technologie, tak mezi lidmi pohybujícími se v této oblasti vznikla naděje, že právě díky ní a současnému stavu výpočetní síly počítače a grafických karet, který je několikanásobně vyšší, než tomu bylo dříve, nastane v daném roce průlom a virtuální revoluce. Bohužel se vždy nakonec ukázalo, že technologie nejsou dostatečně vyspělé a výpočetní výkon není natolik vysoký, aby se dala virtuální realita prakticky používat. A pokud to vypadlo, že by i přes všechny její technické nedostatky té doby mohla uspět, tak narazila na problém finanční dostupnosti. To se však změnilo před pár lety. Díky tlaku mobilního průmyslu na nestále lepší displeje pro chytré telefony, pokročila technologie natolik, že virtuální realita je na dosah široké veřejnosti.

V době psaní tohoto textu se nacházíme na hranici počátku využívání virtuální reality širokou veřejností. Tento rok vydali dva přední výrobci své dlouho očekávané konzumní verze brýlí pro virtuální realitu (Oculus Rift a HTC Vive). Pokrok v této oblasti byl natolik rapidní, že výzkum mnoha souvisejících odvětví za ním zaostal. I když je virtuální realita předmětem výzkumu mnoho let, tak zatím nebyly stanoveny žádné standardní postupy a celosvětově uznávané procesy, které by se staly pevným základem při práci s virtuální realitou. To obzvláště platí pro uživatelská rozhraní ve virtuální realitě a způsoby jeho ovládání.

Cílem této práce je navrhnout a vytvořit uživatelské rozhraní, které bude umožňovat pracovat s počítačem ve virtuální realitě. Zaměřuje se na základní způsoby interakce ve virtuální realitě, které lze využít pro ovládání počítače se zvolenými ovládacími zařízeními. Na základě studia ostatních prací se soustředí zejména na ovládání pomocí rukou uživatele, pro které navrhuje různé způsoby interakce. Při návrhu se snaží vycházet z poznatků ostatních výzkumníků, ale současně se snaží o inovativní přístup. Práce řeší problémy výběru a manipulace s virtuálními objekty a interakci pomocí dotyku.

Logická stavba práce je rozdělena do sedmi po sobě navazujících kapitol včetně úvodu a závěru. Vlastní text práce začíná druhou kapitolou, která je věnována zařízením pro virtuální realitu a snímání akcí uživatele. Další dvě kapitoly popisují teorii uživatelských rozhraní a jeho testování. Páta kapitola se zabývá návrhem řešení. Konkrétně definuje cíle a představuje problémy v závislosti na zvolených ovládacích zařízeních, které je nutné k dosažení cílů vyřešit. Problémy analyzuje a představuje možný způsob řešení. K jeho lepšímu porozumění vytváří koncept systému a podrobný popis jednotlivých částí řešení. V šesté kapitole

je popsán způsob realizace návrhu, otestování vytvořeného řešení a vyhodnocení výsledků testů. Před vlastní implementací je uveden proces volby nástrojů pro implementaci a jejich integrace. Následně je zdokumentován průběh vývoje a konkrétní způsoby řešení. Pro otestování vytvořeného rozhraní komponent je navržen experiment, který se soustředí na jeho výkonnost a použitelnost. Průběh experimentu je zdokumentován a následně jsou diskutovány jeho výsledky, které poslouží pro srovnání vytvořených metod interakce. Nakonec je popsána aktualizace rozhraní komponent na základě poznatku z experimentu a představena demonstrační aplikace.

Diplomová práce opustila cíle stanovené v rámci semestrálního projektu. Ze semestrálního projektu využívá pouze informace o zařízeních v druhé kapitole a první části kapitoly třetí o uživatelských rozhraních.

## Kapitola 2

# Hardware a technologie umožňující virtuální realitu

Virtuální realita, dále jen VR, je oblast, která je zkoumána již od 60. let 20. století [9]. Cruz-Neira ji definoval jako prostředí generované počítačem, které je pohlcující, interaktivní, třídimenzionální, zaměřené na uživatele a využívající více senzorů s kombinací technologií potřebných pro vytvoření tohoto prostředí. První zmínka o VR se váže na pány Mortona Heliga a Ivana Sutherlanda.

Helig sestrojil v roce 1957 zřízení Sensorama viz obrázek 2.1. Jednalo se o mechanický stroj pro promítání snímků, který se snažil zapojit všechny smysly člověka. Například do každého oka promítal mírně odlišný obraz, přehrával zvuk a dokonce foukal vzduch a vibroval se sedadlem. Sensorama je považován za první zařízení snažící se vytvořit virtuální realitu.



Obrázek 2.1: Heligovo zařízení Sensorama a první HMD profesora Sutherlanda

V roce 1966 sestrojil profesor Sutherland na harvardské univerzitě první tzv. Head Mounted Display (HMD), který umožňoval snímání pohybu hlavy a promítání obrazu. Zařízení bylo natolik těžké, že muselo být zavěšeno na stropě. Dobový snímek zařízení je uveden na obrázku 2.1. Největšího zájmu a pozornost veřejnosti získala VR na začátku 90. let. Avšak jako v předcházejících letech narazila na technologická omezení, která byla jedním z hlavních aspektů bránících ve vývoji a rozšíření VR. V posledních letech však technologie postoupily natolik, že VR je téměř na dosah široké veřejnosti. A to především díky trhu s chytrými telefony, jenž posunul kvalitu rozlišení a odezvy displejů natolik, že

je možné virtuální obsah věrohodně zobrazovat. Spolu se zobrazením jsou samozřejmě také kladeny nároky na stále výkonnější grafické karty a výpočetní výkon počítačů obecně.

V této kapitole se věnuji základním principům, které se používají v souvislosti s vytvářením VR, a také popisují několik nejnovějších zařízení již dostupných na trhu nebo jenž jsou ve vývoji. V první řadě se jedná o zařízení určená pro zobrazení VR a následně pro snímání akcí uživatele, které umožňují pohyb a interakci ve virtuálním světě.

## 2.1 Přehled zařízení vytvářejících virtuální realitu

Základním zařízením pro vytvoření virtuální reality jsou virtuální brýle, někdy nazývané také jako VR headset nebo Head mounted display (HMD). Brýle se skládají z displeje, může jich být i více, a speciálních optických čoček upravených tak, aby vytvářely stereoskopický obraz (viz obrázek 2.2). Toho se dosahuje tím, že se vykreslují pro každé oko mírně odlišné obrazy, které si lidský mozek spojí a vznikne stereoskopický obraz s informací o hloubce. Tento jev je nazýván binokulární disparita [11]. Můžeme ho snadno sledovat, pokud si před očima podržíme tužku a střídavě se na ní díváme jedním okem.

Čočky mapují displej na zorné pole, které je uživatel schopen vidět. Zorné pole člověka se pohybuje v rozmezí  $130^\circ$  až  $135^\circ$  ve vertikálním směru a až  $200^\circ$  ve směru horizontálním. Toto zorné pole však obsahuje i periferní vidění. Běžné binokulární zorné pole, oblast ostrého vidění a vnímání vzdálenosti objektů, je široké přibližně  $120^\circ$  [11].

Zatím se nepodařilo vynaleznout způsob, jak celé zorné pole člověka pro virtuální realitu vytvořit. Limit je stanoven především vlastnostmi čoček, které se na jeho vytvoření podílí nejvíce. Běžné zobrazované zorné pole je v současné době okolo  $100^\circ$  na  $100^\circ$ . Tedy blízké šířce binokulárního pole, bez oblastí periferního vidění.

Kvalita výsledného obrazu je kromě optiky závislá na displeji. Aby se obraz vyrovnal realitě, tak by musel mít displej rozlišení 8K pro každé oko a obnovovací frekvenci až 120Hz. Takovéto hodnoty jsou však se současnými technologiemi nereálné. Běžně se tak při používání brýlí setkáváme se zrnitým obrazem, kdy jsou pohledem viditelné jednotlivé pixely. Nízká obnovovací frekvence má za následek rozmazaný obraz při pohybu a může způsobovat nevolnost.



Obrázek 2.2: Části VR zařízení Oculus Rift DK2 a ukázka výstupu pro VR

Existují i některé výjimky, které uvádí hodnoty větší. Tato řešení se však většinou potýkají s nedokonalým překrytím obrazů, které nastává při zaostření oka na určitá místa scény. Zobrazovaný obraz je totiž jen iluze a pokud je dostatečně dobrá, tak jí mozek uvěří. Avšak jakákoliv nedokonalost může způsobit její přerušování, například výše popsané zdvojené

vidění, což často vede k nevolnostem, desorientaci a vytrhnutí z pocitu reálného vnímání okolí. Každé brýle musí obsahovat mechanismus pro snímání pozice a pohybu hlavy spolu s rozhraním pro vstup/výstup (např. HDMI, USB). Snímání pozice hlavy je nejčastěji řešeno pomocí emitorů infračerveného světla nebo laseru vhodně rozmístěných na povrchu brýlí a snímače umístěného v místnosti, který snímá vzdálenost těchto bodů. Dalším způsobem může být snímání pomocí kamer, které však bývá méně přesné. Pro pohyb hlavy se používá gyroskop, akcelerometr a magnetometr.



Obrázek 2.3: HMD zařízení pro virtuální realitu Oculus Rift DK2, HTC Vive, OSVR

Za vzkříšení VR má velkou zásluhu zakladatel společnosti Oculus VR Palmer Luckey, který v září roku 2012 spustil kampaň na serveru Kickstarter. Představil zde brýle s názvem Oculus Rift, které byly zpřístupněny vývojářům ve své první verzi Oculus DK1 (Development Kit). Následovala vylepšená verze DK2 (na obrázku 2.3), která přinesla vyšší rozlišení a menší latenci pomocí low-persistence displejů a dokonalejší snímání pohybů hlavy. Konsumní verze brýlí Oculus CV1 je od začátku tohoto roku dostupná k přeoobjednání a bude vydána v jeho první čtvrtině. Asi největším konkurentem je HTC Vive, který je vyvíjen pod záštitou společnosti Valve. Brýle si můžeme prohlédnout na obrázku 2.3. Datum vydání plné verze je naplánováno na první polovinu roku 2016. Současně jsou dostupné jen prototypy pro vybrané vývojáře. Zajímavostí těchto brýlí je, že pracují se dvěma senzory, které umožňují volný pohyb po místnosti.



Obrázek 2.4: HMD zařízení pro virtuální realitu StarVR, Fove, PlayStation VR

Ambiciózním projektem jsou brýle StarVR vyvíjené společností Starbreeze, které mají nabídnout zorné pole široké 210°, vysoké rozlišení až 5K a snímání pohybu očí. Prototyp, který si můžeme prohlédnout na obrázku 2.4, však zatím pro vývojáře není dostupný. Na snímání pohybu očí se také zaměřili vývojáři z Fove, které uvádí možnost sledovat na co se uživatel dívá jako hlavní přednost jejich produktu. Projekt začal také kampaní na serveru Kickstarter v květnu 2015. Vydání vývojářské verze (na obrázku 2.4) je naplánováno na období první čtvrtiny roku 2016. Díky snímání pohybu očí Fove uvádí, že je možné simulovat



zaostřování, navázat oční kontakt ve VR a možnost ovládání prvků uživatelských rozhraní pomocí pohledu. Zajímavým projektem je pak OSVR (viz 2.3). Jedná se o otevřenou komunitu, která se soustředí na VR. OSVR vytvořila prototyp, který lze získat v rámci Hacker Development Kit. Vznikají i zařízení pro jiné platformy, než počítač. Např. PlayStation VR pro konzoli zobrazené na obrázku 2.4, též známé pod názvem Sony Morpheus a několik VR brýlí využívající chytré telefony (Samsung Gear VR, Google Cartboard). Jedná se v podstatě o konstrukci s čočkami a jednoduchými ovládacími prvky pro možnost vstupu. O zobrazování a snímání pohyb hlavy se stará chytrý telefon.

V předchozím odstavci jsem již zmínil několik parametrů sledovaných u brýlích pro VR. Za důležité parametry se považuje: rozlišení displeje, odezvu displeje, obnovovací frekvence displeje, zorné pole, hmotnost, rozsah a přesnost snímání pohybu. Výše popsání zařízení a jejich parametry<sup>1</sup> jsou shrnuty v tabulce 2.1. Zařízení pro chytré telefony jsem neuváděl, protože se z velké části odvíjí od použitého telefonu.

| Název      | Rozlišení | Typ, poč. disp. | Frekvence | Zor. pole | Vydání    |
|------------|-----------|-----------------|-----------|-----------|-----------|
| Oculus DK1 | 1280x800  | 1xLCD           | 60Hz      | 100°      | 29.3.2013 |
| Oculus DK2 | 1920x1080 | 1xOLED          | 70Hz      | 110°      | 24.7.2014 |
| Oculus CV1 | 2160x1200 | 2x              | 90Hz      | –         | Q1 2016   |
| HTC Hive   | 2160x1200 | 2x              | 90Hz      | 110°      | lis. 2015 |
| Fove       | 2560x1440 | 2xWQHD          | 90Hz      | 100°      | kvě. 2016 |
| StarVR     | 5120x1440 | 2xLCD           | –         | 210°      | –         |
| OSVR DK1.2 | 1920x1080 | 1xOLED          | 60Hz      | 100°      | čer. 2015 |
| Morpheus   | 1920x1080 | 1xOLED          | 120Hz     | 100°      | Q1 2016   |

Tabulka 2.1: Přehled vlastností HMD zařízení pro virtuální realitu

## 2.2 Zařízení pro snímání vstupních akcí uživatele

V předchozí kapitole jsem představil několik zařízení pro zobrazování virtuální reality. Abychom měli možnost ovládání a interakce uvnitř virtuální reality je nutné od uživatele získávat vstupy. K tomuto slouží nejrůznější zařízení pro snímání akcí uživatele. Čím je snímání přesnější, tím je i vyšší věrohodnost výsledného splynutí s virtuálním světem. U vstupních zařízení lze sledovat různé parametry. Může to být vlastnost, kterou snímají, počet dimenzí, jestli jsou přímé nebo nepřímé a výkonnostní metriky (např. přesnost a chybovost). Vstupní zařízení lze rozlišovat podle parametru DOF (Degrees Of Freedom). Tento parametr udává počet dimenzí, které vstupní zařízení dokáže snímat. Pro virtuální realitu tedy interakci ve 3D jsou nejzajímavější vstupní zařízení zvládající 6DOF. Tedy snímání pozice a rotace v osách X, Y a Z. Patří sem zařízení pro snímání pohybu rukou, která umožňují zobrazit ruce uživatele ve virtuálním světě a manipulovat s objekty. Existují i řešení zahrnující snímání celého těla a možnosti přenesení pohybu do virtuálního světa.

Přenesení rukou uživatele do virtuálního světa výrazně zvyšuje pocit splynutí se simulovaným okolím. V současnosti existuje několik produktů, které se snímání rukou věnuje. Produkty lze rozdělit do dvou skupin. První skupina ruce rozpoznává z hloubkového obrazu senzoru. Druhá skupina řešení snímání pomocí přídavných zařízení, jako jsou rukavice či speciálních ovladače. Nejvýznamnějšími zástupci první skupiny jsou senzory LeapMotion (na obrázku 2.5) a Kinect 2. Kinect se více soustředí na rozpoznávání pohybu celé kostry

<sup>1</sup>Data čerpána z <http://www.uploadvr.com/vr-hmd-specs/>

uživatelé. Konkurentem LeapMotion byl projekt NimbleVR, který však byl po odkoupení společností Oculus pozastaven. Tyto řešení jsou pohodlnější, protože uživatel nemusí manipulovat s žádným zařízením. Avšak oproti rukavicím a speciálním zařízením do rukou bývají méně přesné a především méně spolehlivé s vyšší chybovostí. Často může docházet k přerušení snímání. Přesnost těchto systémů se odvíjí nejen od kvality použitého hardwaru a technologického zpracování, ale i od použitých rozpoznávacích algoritmů.



Obrázek 2.5: Zařízení pro snímání rukou LeapMotion, GloveOne, ManusVR

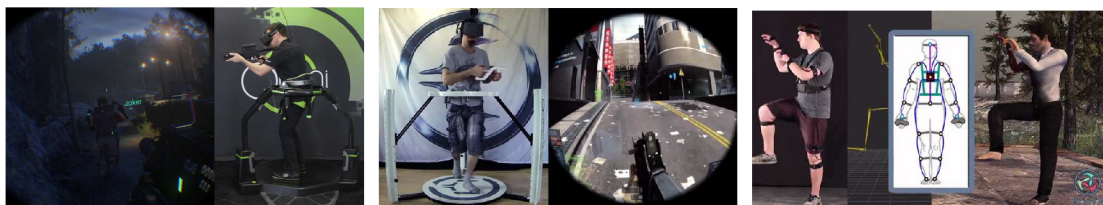
Z druhé skupiny jsou zajímavé rukavice ManusVR. První vývojářská verze se očekává v prvním čtvrtletí 2016. Rukavice jsou bezdrátové a umožňují snímání pohybu prstů a rukou. Podobným projektem jsou rukavice GloveOne, které by měly navíc přinést i hmatovou odezvu. První verze by měla být dostupná v druhém čtvrtletí tohoto roku. Obě rukavice si můžeme prohlédnout na obrázku 2.5. Co se týče představitelů ovladačů asi nejznámějším zařízením je STEM System od společnosti Sixsense (na obrázku 2.6), což je nástupce ovladačů Hydra. Obsahuje dva ovladače do rukou, které přenáší jejich pozici a pomocí tlačítek lze provádět libovolně na ně namapované akce např. uchopení předmětu. Zařízení pracují také jako ukazovátka, kterým lze zaměřovat na okolní objekty ve VR. Součástí mohou být senzory navíc, které lze připevnit kamkoliv a snímat jejich pozici. STEM zatím není dostupný a lze jej pouze přejednat. Další podobné ovladače jsou Oculus Touch a HTC Vive controllers (viz obrázek 2.6), které budou prodávány společně s jejich HMD. Zatím jsou však dostupné pouze vybraným vývojářům. Oculus Touch se nasazuje na ruce a dokáže snímat pohyb prstů a pozici rukou. Ovladače jsou navíc vybaveny tlačítky, na které lze mapovat akce. Je tak možné snímat uživateli ruce a kombinovat ho s ovládáním pomocí tlačítek. Toto řešení je přesnější a méně chybové oproti běžným snímačům.



Obrázek 2.6: Ovladače pro VR STEM Sixsense, HTC Vive controllers, Oculus Touch

Již jsem zmínil možnost snímání pohybu pomocí zařízení Kinect. Alternativami v této oblasti jsou systémy určené přímo pro umožnění pohybu ve VR tzv. treadmill. Jedná se o stanice, kde je uživatel zajištěn do konstrukce a může se na místě pohybovat do všech směrů. Jedním je zařízení pod názvem Virtualizer od společnosti Cyberith a dalším Virtuix omni. Zajímavým projektem je PrioVR, které se snaží pomocí senzoru přenést uživateli

tělo do virtuálního světa a vykreslovat jeho pohyby v reálném čase. Skládá se z několika bezdrátových senzorů a centrální jednotky, která komunikuje s počítačem. Zajímavé může být spojení PrioVR s některým z treadmill zařízení, které by umožnilo nejen se pohybovat, ale i přenést pohyby celého těla uživatele do VR. Tyto zařízení si můžeme prohlédnout na obrázku 2.7.



Obrázek 2.7: Systémy pro snímání těla a pohyb ve VR VirtuixOmni, Virtulizer, PrioVR

## Kapitola 3

# Ovládání počítače a uživatelská rozhraní

Ovládání počítače spadá do výzkumného oboru interakce mezi lidmi a počítačem (HCI – Human-Computer Interaction). Tento obor vznikl v 80. letech a staví na oborech, jako je psychologie a informatika. Zajímá se o způsob výměny informací mezi počítačem a jeho uživateli [11]. Je velmi blízký oboru zkoumání faktorů lidského chování, který se zabývá zejména o výkonnost, omezení a možnosti lidí při používání jakéhokoliv systému. Z pohledu návrhu systémů zkoumá jeho účinnost, bezpečnost, komfortnost a použitelnost. Z oboru psychologie se zabývá tématy jako je kognitivní a experimentální psychologie, které jsou užitečné při zkoumání, jak uživatelé interagují s různými typy počítačových systémů. HCI je obor s poměrně širokým rozsahem. Zahrnuje VR, uživatelská rozhraní (UI), grafická uživatelská rozhraní (GUI) a 3D uživatelská rozhraní (3DUI).

Člověk dnes běžně pracuje s počítačem pomocí GUI. GUI se skládá z ovládacích a zobrazovacích prvků tzv. widgetů. S jejich pomocí jsou prováděny akce vedoucí k cílům, kterých chce uživatel dosáhnout. Vytváří tak abstraktní pohled na ovládaný hardware. Pracuje se vstupy, kterými získává podněty od uživatele a výstupy, kterými poskytuje zpětnou vazbu [3]. První uživatelská rozhraní byla realizována pouze pomocí textového rozhraní a příkazové řádky. Od dob příkazové řádky se rozhraní postupně měnilo až do podoby, která je nám známa dnes. Jsou to rozhraní pracující s okny. Vstupními zařízeními pro ovládání jsou klávesnice a myš. Toto rozhraní v posledních letech nezaznamenalo příliš změn. Za zmínku stojí pokus společnosti Microsoft umožnit ovládání pomocí dotykových technologií ve verzi Windows 8. Systém vznikl v období, kdy se stávaly populárními chytré telefony s dotykovými displeji a Windows 8 měl přinést prvky z prostředí Windows Mobile do stolních počítačů a notebooků.

Změny uživatelského rozhraní jsou poháněny snahou poskytnout uživateli rozhraní, které mu umožní efektivně provádět požadované akce. Zde je na místě se pozastavit nad významem slova efektivně. Znamená to co nejrychleji? Nebo co nejsnadněji? Odpověď záleží na tom, který aspekt sledujeme a ideálně obojí. Velmi populárními požadavky na uživatelská rozhraní jsou intuitivnost, přehlednost a hezký vzhled. Tyto požadavky se dají shrnout pod pojmem uživatelská přívětivost. V posledních letech se uživatelská rozhraní mění především z pohledu stále lepšího grafického zpracování, což je umožněno výkonnějším hardwarem. Koncept plochy a oken je však v podstatě stále stejný už téměř 20 let. Dá se tedy tvrdit, že pro zobrazení na běžném monitoru je tento přístup tím nejlepším co může být.

Od momentu, kdy byly grafické karty dostatečně výkonné, začal také výzkum v ob-



lasti 3D grafických uživatelských rozhraní. Tyto rozhraní se často vyznačují využíváním speciálních vstupních zařízení a multimodalitou. Multimodálním uživatelským rozhraním nazýváme rozhraní, která umožňuje uživateli kombinovat více vstupních a výstupních zařízení pro interakci se systémem [2]. Například řeč, pohyb a nějaké vstupní zařízení. Vzniklo několik vědeckých publikací na toto téma a za nejvýznamnější osobu v této oblasti je považován D. Bowman, jehož kniha *3D User Interfaces: Theory and Practice* se stala referencí pro výzkum. Avšak 3D uživatelské rozhraní se nikdy nestaly natolik rozšířeny, aby mohly ohrozit klasické 2D GUI počítače a jsou spojovány se specializovanými aplikacemi [11].

S příchodem virtuální reality má využití uživatelských rozhraní v 3D daleko větší potenciál. Dá se říci, že se stane nutnou součástí uživatelských rozhraní ve virtuální realitě, která k němu přináší navíc další aspekty. Virtuální realita zásadně mění způsob interakce uživatele s počítačem. Přináší nové možnosti, jak v oblasti zobrazení, tak v oblasti ovládání. Pomocí zobrazení ve 3D a snímání uživatele lze vytvořit abstrakci bližší vnímání normálního světa [2]. Je otázkou, jestli virtuální realita zcela nahradí běžné periferie počítače nebo se budou oba přístupy doplňovat. Ale je jasné, že s rozvojem virtuální reality stoupá požadavek na nová uživatelská rozhraní a intenzivní výzkum v této oblasti.

### 3.1 3D uživatelská rozhraní

Evans [4] ve své práci kriticky popisuje historický vývoj 3D uživatelských rozhraní, jak v akademickém, tak v komerčním prostředí. Zdůrazňuje čtyři nejdůležitější doporučení týkající se interakce lidí s počítačem, na které by se při návrhu 3DUI mělo myslet.

#### Čtyři doporučení pro návrh 3DUI dle Envasa:

1. Zachovat podobnost – Snaha prezentovat data v podobném duchu, jak jsou uživatelé zvyklí. Avšak tak, aby byla poskytnuta nová informace.
2. Používat 3D za účelem poskytnutí dodatečné informace – používat účelně a ne s cílem vizuálně zkrášlit. Například metadata, která bývají často prezentována jako text lze v 3D zobrazovat přirozeným způsobem graficky (typ souboru, velikost, vlastnictví).
3. Rozhraní musí být rychlé a využívat co nejméně výpočetních zdrojů.
4. Vizually atraktivní rozhraní – v rozporu s 2, ale snaha zaujmout uživatele za účelem získání popularity.

První pokusy s 3DUI byly prováděny již v 90 letech spolu s představením dedikovaných grafických karet. Problém byl však stále slabý výkon a náročnost na výpočetní prostředky. Jednou z výrazných chyb tohoto období je považováno soustředění se na líbivé vizualizační efekty a opomenutí efektivity práce [4]. Na akademické půdě se v počátcích začalo výzkumu věnovat mnoho skupin, ale v roce 2001 poklesl o 3D rozhraní zájem po práci Cockbruna a McKenzieho, která demonstrovala, že 3D rozhraní nepřináší výhody pro určité úkoly správy souborů. Použily pro demonstraci program Data Mountain. Zájem opět narostl v letech 2005 a 2010, kdy docházelo k zlepšování hardwaru obecně a rozvoj v oblasti virtuální a rozšířené reality. V roce 2004 Bowman vydal první knihu věnovanou návrhu 3D uživatelských rozhraní s názvem *3D User Interfaces: Theory and Practice*, která se stala základem pro výzkum v tomto oboru. Ukázalo se, že je obtížné kriticky testovat 3D rozhraní, protože uživatelé mají 2D přístup natolik zažitý, že to staví 3DUI do nevýhody.

### 3.1.1 Techniky navigace ve virtuálním prostředí

Bowman [2] se již v této době věnuje virtuální realitě, avšak vlastnosti dříve dostupných zařízení se nedá srovnávat s dnešní situací. Popisuje možnosti technik navigace, které dělí na dvě komponenty – cestování (fyzický pohyb uživatele) a hledání cesty (komponenta soustředěná na kognitivní myšlení uživatele). Definuje pět skupin, do kterých se dá zařadit většina přístupů k cestování:

- Fyzický pohyb – využívá fyzický pohyb uživatele, který je snímán pomocí zařízení umožňující pohyb na místě (kola, chodící pásy).
- Manuální manipulace – techniky, které používají pro pohyb ruce uživatele. Příkladem může být uchopování vzduchu a přitahování virtuálního lana. Tento přístup je jednoduchý na naučení, ale bývá pro uživatele vyčerpávající.
- Řízení – přístup určování směru pohybu pohledem v kombinaci s jednoduchým ovládním pohybu. Podobné řízení auta, nebo letadla. Uživatel ovládá změny směru pohybem hlavy.
- Cestování založené na výběru cíle – uživatel zadá cíl, kam se chce přesunout a buď na toto místo může být teleportován, nebo automaticky postupně přesunut.
- Plánování cest – uživatel specifikuje cestu, kterou se chce pohybovat a následně je proveden přesun po ní.

Komponenta hledání cesty by měla podporovat vytvoření kognitivního modelu (mapy) prostředí, kde se uživatel nachází a usnadnit mu tak jeho orientaci. Podpora hledání cesty pomáhá zamezit desorientaci v případě, že by měl uživatel úplnou volnost pohybu. Využívá grafických indikátorů ale i zvuku.

### 3.1.2 Možnosti ovládání 3DUI

Jedním z problémů je způsob řešení ovládání systémů pomocí příkazů. Ve 2D rozhraních je tato oblast řešena pomocí různých ovládacích prvků, jako rolovací menu, tlačítka, textové vstupy, zaškrtačkové boxy atd. Bohužel tyto přístupy nejsou vždy použitelné ve virtuálním prostředí. Hlavním problémem je, že když se úloha prováděná v 2D převede do 3D, tak dochází ke snížení efektivity jejího provádění. Například práce s menu vznášejícím se v prostoru. Způsoby interakce v této oblasti jsou stále součástí výzkumu. Bowman rozděluje systémovou kontrolu do čtyř skupin:

- Grafická menu pro vizuální reprezentaci příkazů
- Hlasové příkazy – zadávání příkazu pomocí hlasu a jejich rozpoznání
- Interakce pomocí gest – mapování příkazů na specifické pohyby rukou a kontext
- Virtuální nástroje – objekty s definovanou funkcí (např virtuální štětec, ovladač.)

Výše popsané skupiny se často kombinují a je snahou o interakci bez módů, kde změna módu je pokud možno co nejpřirozenější. Dobrým přístupem je zobrazování prvků menu ve fixní vzdálenosti od uživatele a využití multimodálního rozhraní. Návrhář by měl veškeré akce doprovázet indikací o jejich provedení, aby byl uživatel informován o stavu systému.



Chybným předsudkem je dle Bowmana to, že protože se uživatel nachází ve virtuálním prostředí, tak by měl návrhář vždy použít interakce s 3D objekty. Naopak popisuje řadu výhod, které přináší 2D interakce oproti 3D pro určité úlohy. Jako příklad uvádí fyzickou odezvu, kterou poskytuje rozhraní tabletu pro psaní textu. Také většina akcí pro výběr jsou v podstatě 2D, i když často další manipulace může potřebovat interakci v prostoru. Navrhuje využít integraci obou technik interakce, které vedou k lehčímu používání a dělají rozhraní více intuitivní. Integrace vyžaduje pozornost z pohledu logického a fyzického. Je nežádoucí, aby byl uživatel nucen měnit fyzická zařízení při používání. A také aplikace by měla vědět, jakým způsobem má uživatel zařízení v daný okamžik používat. Příkladem je právě tablet, který uživatel s virtuálními brýlemi fyzicky nevidí a je potřeba jeho virtuální reprezentace. Ve výsledku vznikají fyzické pracovní stanice, které jsou mapovány do virtuálního světa.

### 3.1.3 Návrh interakce s 3DUI

Bowman představuje dvě filosofie návrhu interakce ve 3D – uměleckou a systematickou. Umělecká filozofie podporuje návrháře ve zkoušení nových přístupů, protože rozhraní v 3D ještě nejsou prozkoumány na takové úrovni, jak tomu je v případě 2D. Ještě pro 3DUI nebylo vytvořeno žádné paradigma jako je WIMP pro 2D, pro které existují příručky s metodologiemi a doporučeními například od Shneidermana. Úspěšný 3D design by měl stavět na existujícím výzkumu a měl by využívat interakce a nápady představené ostatními výzkumníky. Avšak návrhář by měl zapojit svoji kreativitu, aby řešení nějakým způsobem inovoval a vylepšil. Inspiraci lze čerpat ze dvou kategorií. První kategorie se inspiruje interakcí v reálném světě. Druhá naopak magickým přístupem, který může čerpat například ze sci-fi filmů. Umělecká filozofie je často vedena ad-hoc přístupem, který není vhodný pro výzkum, analýzu a vyhodnocování představených nápadů a použitých technik. Proto vznikla filozofie systematického přístupu.

Pro systematický přístup je charakteristické studium úkolů, které uživatelé provádějí, existujících technik a vlastností uživatele, prostředí a systému, které mohou ovlivnit jeho efektivitu. Základem je klasifikace, kategorizace a dekompozice úkolů a technik se snahou je více pochopit a vytvořit rámec pro návrh nových přístupů a vyhodnocování přístupů již existujících. Důraz je také kladen na vývoj v iteracích a jejich vyhodnocování, které vede ke změnám v návrhu a možnosti sledování celého procesu zpětně. Na závěr Bowman shrnuje několik obecných principů, pro návrh 3D rozhraní:

- Zvážit použití magického přístupu oproti přirozenému, když úloha klade požadavky na efektivitu a produktivitu
- Vybírat techniky pro interakci v závislosti na vyvíjené aplikaci
- Omezit stupeň volnosti vstupů, jak nejvíce je to možné a vytvářet uživateli fyzické a virtuální omezení pro podporu vedení při jejich provádění
- Využít výhod multimediálních rozhraní a vstupních zařízení, ale poskytnout uživateli přístup a podporu v jejich ovládání

### 3.1.4 Některé aplikace využívající 3DUI

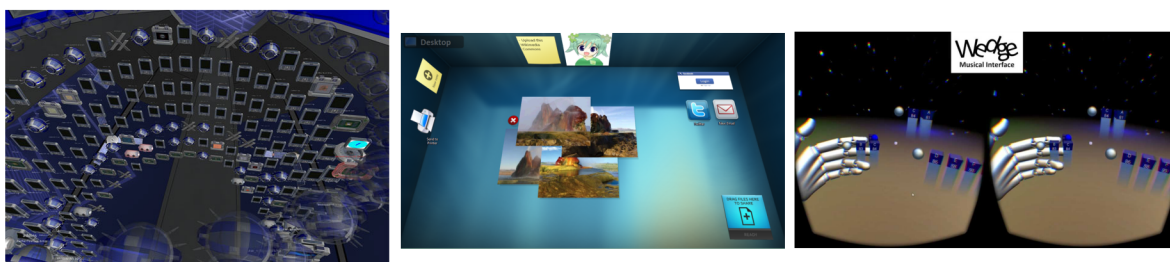
V počátcích byla snaha využít 3DUI v oblasti práce se souborovým systémem a procházení hierarchických. Hlavním argumentem byl nedostatek místa pro zobrazení ve 2D a nepřehlednost. Například aplikace Tactile 3D (viz 6.3) slouží jako správce souborů ve 3D. Vznikaly

i nové způsoby reprezentace hierarchických struktur, které ve své práci popsal Mikic [10]: Cone Tree, Hyperbolic Tree 3D, Botanical Tree a Burst Tree. Avšak trend poslední doby je využívat při správě souboru vyhledávání a tak vizualizace a práce se složitými strukturami zůstává spíše ve specializovaných oblastí vědeckých dat. Ukázky aplikací pracujících s hierarchickými strukturami jsou uvedeny na obrázku 3.1.



Obrázek 3.1: Ukázka vizualizace Cone Tree, Burst Tree a Hyperbolic Tree 3D

Dalším zajímavým projektem byl BumpTop. Šlo o aplikaci pracující s plochou počítače ve 3D pomocí dotykového displeje a stylusu. V roce 2010 ji odkoupil Google, ale o dva roky později oznámil, že nebude pokračovat v jejím vývoji. Každý rok se také pořádá konference IEEE 3DUI, jejichž součástí je soutěž aplikací využívající 3DUI. Představení aplikací v soutěžích od roku 2014 až do 2016 je možné shlédnout na kanálu IEEE 3DUI Contest<sup>1</sup> v platformě YouTube. Ukázka některých aplikací je na obrázku 3.2.



Obrázek 3.2: Ukázka Tactile 3D, BumpTop a Wedge: A Musical Interface z IEEE 3DUI Contest

## 3.2 Způsoby interakce ve virtuální realitě

Virtuální realita přímo vybízí k 3D interakci. Tedy interakci uživatele v kontextu trojrozměrného prostoru. Hlavním problémem, který 3D interakce řeší je manipulace s objektem nebo skupinou objektů. Manipulace rozdělujeme na podproblémy **výběru** a **transformace** (posuv a rotace) [11]. Ve snaze o zobecnění manipulačních metod Bowman, Foley a Poupyrev rozdělují proces na tzv *kanonické manipulační úkony*: selekce, pozicování a orientaci.

### Selekce

Je to proces získání požadovaného cíle z množiny možných cílů [11]. Umožňuje tedy uživateli vybrat objekt nebo skupinu objektů z nějaké jejich množiny. Selekcce poskytuje metaforu

<sup>1</sup>[https://www.youtube.com/channel/UC71Pn23ef\\_1q6XqXsTV0BWg](https://www.youtube.com/channel/UC71Pn23ef_1q6XqXsTV0BWg)

pro někoho, kdo vybírá objekt v reálném světě [3]. Při selekci pracujeme s následujícími informacemi [11]: vzdálenost cíle, směr k cíli, velikost cíle, počet selekcí a cíl.

### Pozicování

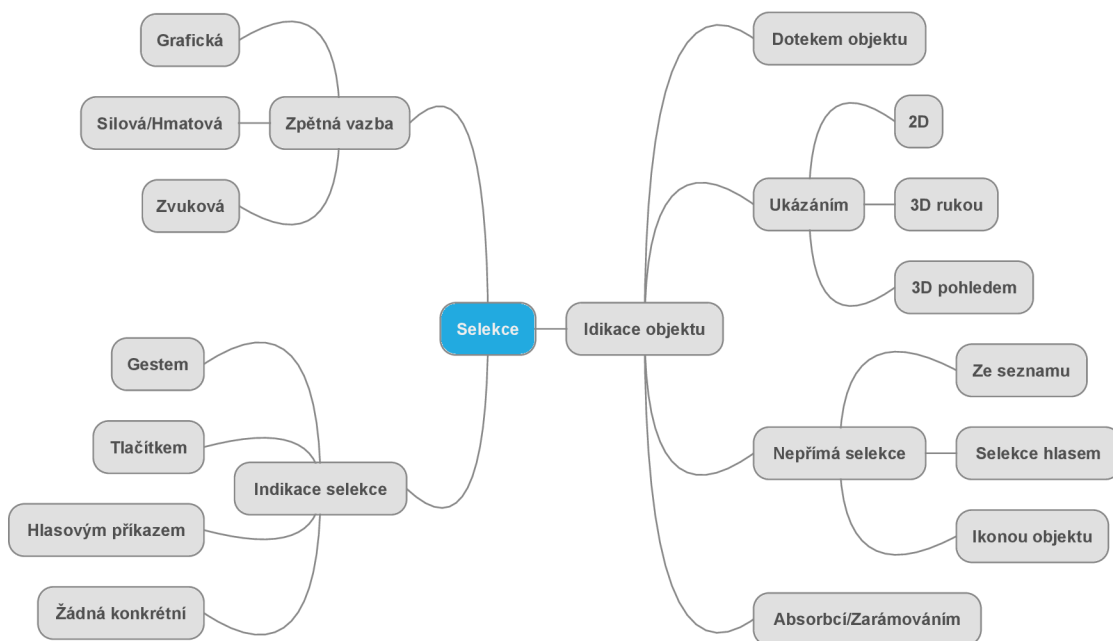
Pozicování umožňuje umístit cíl na různá místa. Tedy v 3D grafice transformace objektu nebo objektů v souřadném systému. Pracujeme s těmito informacemi [11]: vzdálenost k počáteční a cílové pozici, směr k počáteční a cílové pozici, vzdálenost posuvu cíle, přesnost, viditelnost, velikost aj. Pozicování poskytuje metaforu pro někoho, kdo posouvá objekt z bodu A do bodu B v reálném světě [3].

### Orientace

Orientace je proces umožňující uživateli otáčet cílovým objektem [11]. Při návrhu je potřeba myslet na to, že transformace obsahující rotaci může mít různé výsledky v závislosti na pořadí provedení transformací. Orientace pracuje s informacemi [11]: o kolik se bude rotovat, počáteční orientace, koncová orientace a požadovaná přesnost. Orientace poskytuje metaforu pro někoho, kdo otáčí objekt v reálném světě [3].

#### 3.2.1 Klasifikace technik pro 3D interakci

Za posledních třicet let výzkumu v oboru interakce ve virtuálním prostředí vznikly různé přístupy, jak tento problém řešit. Avšak nebyl kladen takový důraz na výzkum toho, jak tyto vznikající techniky interakce klasifikovat a vyhodnotit. První klasifikaci představil Bowman spolu se svými kolegy. Klasifikovali interakci pomocí dekompozice úkonů. Interakci dekomponují na nejvyšší úrovni do tří skupin: selekci, manipulaci a uvolnění [5]. Jednotlivé skupiny jsou znázorněny na obrázku 3.3.



Jeden z dalších způsobů klasifikace představil Poupyrev a jeho kolegové, kteří interakci klasifikovali na základě úrovně pohledu. Rozlišují přístupy egocentrické a exocentrické [5].

Egocentrické techniky jsou nejběžnější ve virtuálním prostředí a staví na iluzi, že uživatel je přímo přítomen v prostředí, kde provádí interakci. Exocentrické techniky jsou naopak takové, kdy uživatel s virtuálním prostředím pracuje z vnějšku. Klasifikace je znázorněna na obrázku 3.4.

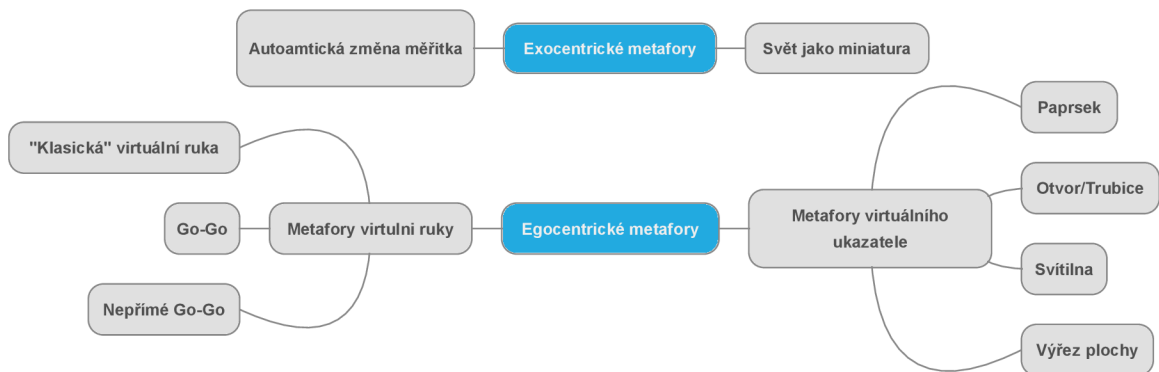


Obrázek 3.3: Klasifikace technik interakce dle Bowmana

Poslední klasifikaci představil na konferenci IEOM konané v roce 2014 Jung [5]. Vychází z klasifikací Bowmana a Poupyreva, které rozděluje do tří podkategorií v závislosti na způsobu interakce: přímá manipulace, fyzické ovládání a virtuální ovládání. Do kategorie přímé manipulace patří techniky, které pracují se snímáním pohybů rukou, jenž přenáší do virtuálního prostředí a umožňuje tak přirozenou manipulaci s objekty. Fyzické ovládání zahrnuje techniky využívající k interakci fyzická zařízení, která jsou přenášena do virtuálního světa. Například joystick a ovládací panely v kokpitu letadla. Virtuální ovládání naopak využívá k interakci virtuální objekty. Například virtuální ruka, či virtuální panely. Pro jejich ovládání se opět používá nějaký druh speciálních vstupní zařízení. Avšak v příkladu virtuální ruky se nesnímá pohyb ruky, ale pouze se virtuální ruka ovládá pomocí jiného vstupního zařízení. Tedy uživatel má například zařízení snímající 6DOF s tlačítkem. Ve virtuální realitě je zobrazena virtuální ruka a mění pozici v závislosti na snímání pohybu zařízení a pokud uživatel zmáčkne tlačítko může se provést nějaká akce, jako třeba uchopení objektu nebo zmáčknutí tlačítka na virtuálním panelu.



Pomocí výše popsaných způsobů klasifikace jsme schopni velmi přesně kategorizovat jakoukoliv interakci. Například si představme interakci, kdy uživatel používá virtuální brýle a má rukavice, které snímají pohyb jeho rukou. Nachází se ve virtuální místnosti a před sebou vidí virtuální objekty, které může ukázáním na ně vybrat a poté s nimi přirozeně manipulovat. Taková technika je tedy egocentrická a využívá přímou manipulaci. Indikaci objektu provede ukázáním rukou ve 3D. Tedy gestem a zpětná vazba je zajištěna graficky. Uchopení provede opět rukou pomocí gesta. Stejně tak pozice a orientace se odvíjí od pohybu ruky. Uvolnění opět provede gestem a objekt zůstane na aktuální pozici.



Obrázek 3.4: Klasifikace technik interakce dle Poupyřeva

### 3.2.2 Přímá manipulace s virtuálními objekty

S možností snímat uživateli ruce a přenést je do virtuální reality se přirozeně nabízí z egocentrických technik použít přímou interakci s virtuálními objekty. Zásadní výhodou používání přímé interakce pomocí rukou je přirozenost ovládnutí, které simuluje realitu. Pro uživatele je proto lehké se naučit rozhraní rychle ovládat. Přímá interakce však stále není zcela vyřešena a není zatím definován žádný standardní přístup.

Z výzkumných prací na toto téma lze rozeznat dva směry, kterými se udávají. Jedním je snaha dokonale simulovat ruce ve virtuálním prostředí. Takové pokusy pracují s různými specializovanými zařízeními a rukavicemi s haptickou odezvou. Například Jun-Sik a Jung-Min ve své práci popisují přístup, jak interagovat s virtuálními objekty na základě fyzické simulace [6]. Detekují kolizi virtuálních modelů a simulují deformaci modelu ruky při kontaktu s pevným objektem a třením v místě kontaktu. Dalším směrem je interakce založená na detekci gest ruky, na něž lze reagovat akcemi. Interakci založenou na gestech použili ve své práci Lu a Shark [7], kteří použili pro manipulaci s virtuálními objekty snímání pomocí rukavice. Tyto přístupy jsou oproti simulacím méně náročné a mají větší volnost a možnosti při ovládnutí. Pomocí gest lze používat magické přístupy, o kterých se zmiňuje ve své práci Bowman [3].

Mapování akcí na definovaná gesta ruky si lze představit jako klávesové zkratky, ale lze pomocí gest řešit i přímou manipulaci s virtuálními objekty. Význam gest se navíc může v různém kontextu měnit. Počet statických gest rukou je totiž omezený, protože je nutné, aby byla snadno rozpoznatelná a rozlišitelná. Proto vznikla dynamická gesta, která umožňují sekvence gest spolu s trajektorií ruky. Příkladem dynamického gesta je např. nakreslení kruhu ve vzduchu, nebo pohyb ruky do boku. Využití gest pro manipulaci s virtuálními objekty lze řešit například detekováním gesta sevření pěsti v blízkosti pozice objektu, se kterým uživatel chce interagovat. Na základě detekce gesta lze následně provést uchopení

objektu. Podobné může být použití gesta sevření ruky pro uchopení objektu, na který se uživatel právě dívá. Pomocí gest se dá řešit například i navigace ve VR nebo natáčení objektů. Používání gest má však také svá úskalí v nutnosti uživatelů učit se jejich význam a způsob použití. Rozhraní z rozsáhlým počtem gest může být poměrně náročné na zapamatování. Protiargumentem může být, že klávesové zkratky pro ovládání je nutné se také naučit. Avšak je nutné brát v úvahu, že operace prováděné pomocí klávesové zkratky lze provést i bez ní, například přes interakci s menu. Je tedy dobré uživateli poskytnout jednotný a jednoduchý způsob ovládání pomocí kterého může provádět všechny podporované akce spolu s gesty.

### 3.2.3 Ovládání řešené pomocí gest rukou

Příklad použití dynamický gest a grafické odezvy pro interakci s 3D uživatelským rozhraní ve svém článku popsali Monica Bordegoni a Matthias Hemmje [1], kde představují tzv. Gesture Language (jazyku gest). Navrhli systém pro rozpoznání gest, který použili pro komunikaci s uživatelským rozhraním aplikace. Pro snímání vstupu použili rukavici VPL DataGlove, klávesnici a myš. Použili sekvence gest, například ukázání prstem a natáčení ruky pro navigaci, gesto pistole pro výběr objektu, gesto ukázáním dvou dvěma prsty pro seskupování objektů, ukázání prstu a jeho zahnutí pro otevření obsahu objektu, přiblížení a oddálení pomocí pohybu dlaně dopředu a dozadu, uchopení a přesun objektu pomocí pěsti.

První rozsáhlý jazyk gest byl navržen a použit ve filmu Minority report Johnem Underkofflerem. Avšak bohužel se mi nepodařilo najít celý originální text návrhu. Underkoffler se interakcí pomocí gest stále zabývá a na toto téma vedl několik přednášek na různých konferencích. V roce 2010 předvedl na konferenci TED<sup>2</sup> uživatelské rozhraní pracující s gesty, na jehož vývoji se podílí.

### 3.2.4 Hybridní manipulační techniky s virtuálními objekty

Do teď jsem popisoval pouze přímou manipulaci s objekty, kdy se manipuluje s objektem v dosahu uživatelových rukou. Avšak uživatelé často potřebují manipulovat i s objekty mimo jejich dosah. Proto se řeší způsoby výběru objektu. Nejčastěji se pro výběr používá paprsek, který je vyslán směrem na cílový objekt [11]. Jedná se v podstatě o způsob laserového ukazovátka. Paprsek se tedy připevňuje na zařízení, které umožňují 6DOF. Může to být například na konci ukazovačku ruky v případě snímání rukou nebo na vhodnou část fyzického zařízení jako je STEM System viz kapitola 2.2. Je také možné paprsek upevnit na HMD a využívat tak pro výběr pohyby hlavy a směr pohledu uživatele. Pokud je dostupná technologie snímání pohledu, lze využít vysílání paprsku ze zornic uživatele.

Kombinací selekce pomocí paprsku a přímé manipulace s vybraným objektem vznikají tzv. hybridní nebo kombinované manipulační techniky [11]. Jednu takovou techniku představil Bowman ve své knize pod názvem HOMER (Hand-centered Object Manipulation Extending Ray-casting) [3]. Tato technika využívá k výběru právě paprsek a poté připevní ruku uživatele k vybranému objektu a umožňuje s ním manipulovat stejně, jako by byl v dosahu uživatele.

---

<sup>2</sup><https://www.youtube.com/watch?v=b6YTQJVzw1I>



## Kapitola 4

# Testování uživatelských rozhraní

V této kapitole se zaměřuji na techniky a postupy používané při testování a vyhodnocování uživatelských rozhraní. Vycházím z knihy Scotta I. MacKenzieho [8] věnované empirickému testování uživatelských rozhraní, která mi sloužila jako reference pro studium v této oblasti. Kniha obsahuje kapitolu na téma lidských faktorů, důležitých pro testování a vyhodnocování v oblasti HCI spolu s používanými metrikami. Dále v ní lze nalézt základní principy pro vedení vědeckého výzkumu a návod, jak správně navrhnout a provést HCI experiment nebo uživatelskou studii.

Nejdříve, než se pustím hlouběji do teorie testování UI, bych se chtěl zamyslet nad tím, proč vlastně provádíme testování? Testujeme pro to, abychom si mohli odpovědět na otázky, které si klademe. Je tento přístup dobrý? Jak moc je dobrý? Vylepšuje nějak moje řešení aktuální praxi?. Tyto otázky však mohou být velmi obecné a jejich zodpovězení náročné nebo subjektivní. Proto musí být otázky dobře formulované a zodpovězené pouze na základě pozorování a měření. Toho docílíme tak, že se budeme držet postupů aparátu vědeckého zkoumání. Nyní se naskýtá vhodná chvíle vysvětlit si několik pojmů. Slovo empirie znamená zkušenost získanou pozorováním, případně pokusem. Používá se, když chceme zdůraznit cílený a řízený způsob získávání zkušeností. Empirické vědy jsou tedy vědy založené na opakovatelných a ověřitelných experimentech. Uživatelská studie je proces složený z dílčích experimentů prováděných na uživateli, kdy jsou jim představeny dva nebo více druhů rozhraní, které jsou následně empiricky a kvantitativně srovnávány [8].

### 4.1 Lidské faktory ovlivňující práci člověka s počítačem

Lidské faktory jsou největší výzvou v HCI [8]. Člověk je ve srovnání s počítačem velmi komplikovaný a to především kvůli variabilitě. Lidé jsou různí v mnoha pohledech. Ať je to věk, pohlaví, zkušenost, výška, síla, rychlost, vzdělání, postižení a tak bych mohl pokračovat do nekonečna. Tato variabilita má za následek, že práce v oboru HCI nebude nikdy precizní, nýbrž pouhou aproximací. Návrh rozhraní, které bude vyhovovat všem bez rozdílu a bude pro všechny nejlepší možné je nereálný. Vždy nalezneme skupinu lidí, pro které nebude vhodné nebo bude zcela nepoužitelné. Například jakékoliv grafické rozhraní pro nevidomé. Proto je důležité dobře znát cílového uživatele. Ve snaze pochopit lidské chování a kategorizovat jeho aspekty, které jsou důležité pro HCI, vznikly deskriptivní modely člověka.

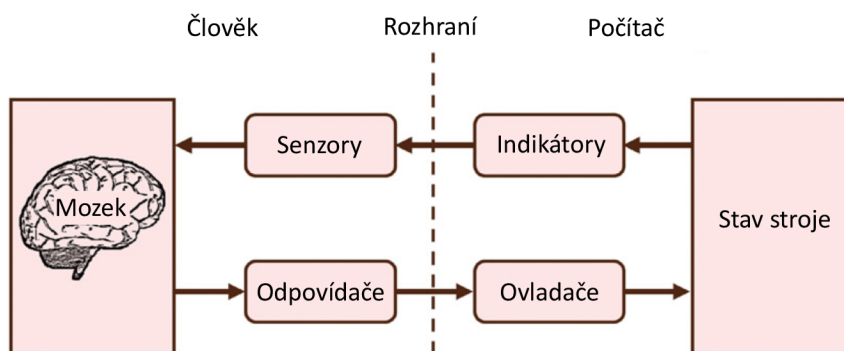
Jedním takovým modelem je *Newellova časová stupnice akcí člověka*, která rozděluje různé typy akcí člověka do časových pásem podle toho, jak dlouho daná akce trvá. Model

používá pásma: biologická, kognitivní, racionální a sociální. Nejběžnější závislou proměnou experimentálního výzkumu v HCI je čas. Newellův model je proto pro HCI relevantní a při výzkumu lze vždy zkoumaný úkon zařadit do nějakého z uvedených pásem. Například úkon selekce při interakci bude spadat do kognitivního pásma, které se pohybuje v rozmezí od stovek milisekund do několika desítek sekund. Od délky trvání úkonů se odvíjí způsob jejich zkoumání. Platí, že pro krátko trvající úkony bývají vhodné experimentální metody. Naopak pro dlouho trvající úkony bývají vhodné metody kvantitativní nebo metoda pozorování.

### Model uživatele pracujícího s počítačem

Pro výzkum se často používá model uživatele ovládajícího počítač, který je znázorněn na obrázku 4.1. Uživatel sleduje stav počítače na jeho **indikátorech** (výstupy) přes své **senzory** (vstupy) a ovládá stav počítače pomocí jeho **ovladačů** (vstupy) svými **odpověďmi** (výstupy). Čárkovaně je znázorněna oblast, kde dochází k interakci mezi člověkem a počítačem pomocí rozhraní. Toto je oblast, kterou při testování uživatelských rozhraní sledujeme a měříme metriky úkonů, ze kterých se skládá interakce.

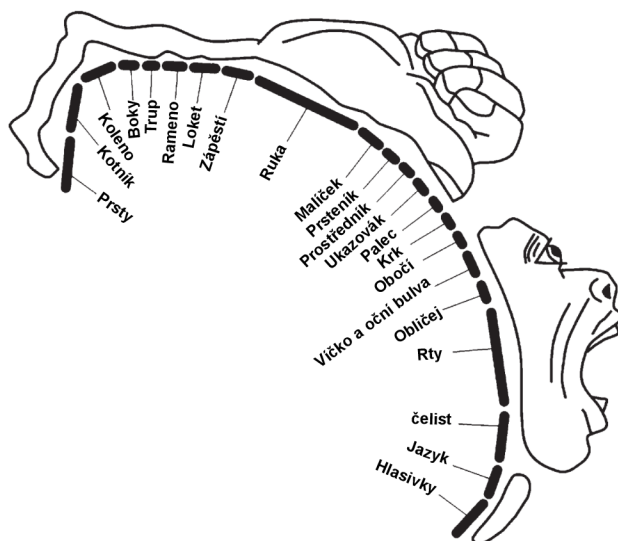
Senzory respektive lidské vstupy jsou: zrak, sluch, hmat, čich a chuť. Mimo klasické smysli rozlišujeme i speciální smysli jako: pocit naléhavosti, orientační smysl, intuice a morální smysl. Tyto speciální smysly určují, jak se uživatel cítí při interakci s počítačem.



Obrázek 4.1: Schéma modelující uživatele pracujícího s počítačem

Odpovědače, respektive lidské výstupy pomocí kterých uživatelé ovládají počítač jsou znázorněny na obrázku 4.2. Nejvíce se používají pro interakci končetiny (především ruce), pak hlas a nakonec oči. Ruce používáme například při interakci s myší a klávesnicí nebo dotykovým displejem. Nohy můžeme používat například k ovládání pedálů. Hlas používáme k zadávání hlasových příkazů, které jsou počítačem rozpoznávány. Méně tradiční jsou pak oči, které mohou být použity jako vstup například u paralyzovaných lidí.

Řídící jednotkou člověka je mozek. První část, která je aktivní při zpracovávání vnějších vstupních podnětů je percepce. Centrum percepce obsahuje nedokonalosti vnímání člověka, které se často zneužívají. Například neschopnost vnímat některé frekvence zvuku či velmi rychle blikající obraz. Další vlastností vnímání je dvojznačnost, která má za následek různou reprezentaci sledovaného objektu, známou v optických klamech. Mimo vnímání mozek plní mnoho dalších funkcí, které mají za následek možnost přemýšlet, uvažovat, rozhodovat se atd. Tyto aspekty řadíme do kognitivního myšlení, které se snažíme také měřit. Avšak měření kognitivní zátěže je velmi náročné a je jednou z výzev v oblasti HCI. Poslední částí mozku, která nás zajímá je paměť. Rozlišuje se na dlouhodobou a krátkodobou paměť.



Obrázek 4.2: Grafické znázornění částí lidského těla používaných jako výstupy

#### 4.1.1 Měření výkonnosti člověka

Pokud člověk provádí jakýkoliv úkon, lze měřit jeho *výkonnost* při provádění. Nejlepší výkonnost je většinou asociována s rychlejším nebo více přesným prováděním. Pokud sledujeme tyto metriky, tak se dostáváme k základnímu kompromisu u výkonnosti. Rychlost za cenu menší přesnosti a naopak. Čím rychleji provádíme nějaký úkon, tím je vyšší šance, že uděláme chybu. Naopak pokud zpomalíme, zvyšuje se přesnost. Cílem je dosáhnout výsledku, který je pro uživatele komfortní a konzistentní při používání.

Jednou z nejzákladnějších metrik je *reakční doba*. Definujeme ji jako dobu mezi vznikem nějakého vnějšího stimulu a započítáním reakce na něj. Zajímavým faktem o reakční době je, že se mění v závislosti na smyslu, který je stimulován. Pokud seřadíme smysly dle nejnižší reakční doby, pak na prvním místě je zvuk následován zrakem, čichem a na poslední příčce se umístí hmat. Obdobou reakční doby je metrika nazývaná *vizuální vyhledávání*. Sleduje dobu, kterou trvá uživateli naleznou hledanou položku v množině  $n$  položek. Tento údaj nás může zajímat například při rozhodování, jaké rozložení prvků v rozhraní použít. Další zajímavou metrikou je *pozornost*. Často se sleduje kolik úkonů je uživatel schopen provádět zároveň. Kolik pozornosti vyžaduje konkrétní úkon. Jak je pozorností ovlivněna výkonnost atd. Asi nejvíce sledovanou metrikou je *chyba*. Je nejdůležitější především při testování nových rozhraní a interakčních metod. Chybu definujeme jako diskretní událost, která nastane při provádění nějakého úkolu a má za následek odchýlení od správného a požadovaného výsledku. Typicky se chybovost udává jako počet chybných pokusů vůči všem prováděným pokusům. Při zaznamenání chyby je snahou naleznout, proč k ní došlo a pokud možno zdroje chyb odstranit. Hledání příčin chyb často bývá náročné, protože ji může způsobovat téměř cokoliv. Níže uvádím několik metrik používaných při testování interakce [11].

### Výkonnostní metriky pro úkony interakce:

- Čas provedení: Čas měřený od započetí do ukončení provádění zkoumaného úkonu.
- Přesnost: Aktuální pozice oproti požadované pozici.
- Chybovost: Kolikrát uživatel selhal úspěšně dokončit úkon?
- Jednoduchost používání: Kognitivní zátěž uživatele.
- Jednoduchost naučení: Zlepšení uživatele v rámci sledované doby.
- Pocit přítomnosti: Slinutí s okolím a míra prostorového vnímání.

## 4.2 Způsob provádění vědeckého výzkumu

Nejdříve si pojďme uvést definici toho co vlastně chápeme pod slovem výzkum a co by měl splňovat. Výzkum můžeme chápat jako proces shromažďování informací týkajících se nějakého konkrétního tématu za účelem zkoumání a experimentování s cílem objevu nových faktů nebo souvislostí [8]. Provádění experimentů je centrální složkou mnoha výzkumů v HCI. Experiment se v HCI někdy nazývá také uživatelská studie. Výzkum je prováděn na základě metodologie, která může být formální nebo vytvořená jen pro konkrétní zkoumaný případ. Přednost je dáván metodologiím, které jsou formální a standardizované. Zaručují totiž konzistenci a formát výzkumných prací, které lze pak snadněji srovnávat. Pojďme se vrátit k uvedené definici, která mluví o faktech. Fakta dokazují, že výsledky naší práce jsou validní a nezpochybnitelné. Důležité je, že výzkumná práce musí být reprodukovatelná. Pokud tomu tak není, je zbytečná. Proto je kladen důraz na používání standardizovaných metodologií. Metodologie zajišťují, že výzkumná práce bude dostatečně detailní na to, aby byla reprodukovatelná a jiný vědec si mohl ověřit její poznatky nebo využít představené postupy ve své práci. Příkladem je metoda SUXES, která slouží k vyhodnocování interakčních technik [12]. Metoda pracuje s očekáváními a zážitky uživatelů.

### 4.2.1 Výzkumné metody

Pro výzkum v HCI se používají tři základní metody, jak vést výzkum: *sledovací*, *experimentální* a *korelační* [8]. Všechny tyto metody jsou empirické. Tedy jsou založeny na sledování a zkušenostech. Avšak metody se odlišují v cílech, odbornosti a stylu výzkumu.

#### Metoda založená na sledování

Tato metoda se věnuje pozorování chování lidí při ovládání počítače. Sledování probíhá v přirozeném prostředí mimo laboratoř. Přístup metody bývá spíše kvalitativní, než kvantitativní a soustředí se na relevanci na úkor menší přesnosti. Používá se pro objevování a objasňování důvodů základního lidského chování. Odpovídá na otázky typu *jak* a *proč*. Zaměřuje se na lidské vlastnosti, které se špatně měří jako myšlenky, pocity, chování, emoce, požitky atd. Při pozorování se nic neměří, ale píšou se poznámky, fotí, natáčí video či zvukový záznam. Tedy metody založené na sledování se přiklání ke zkoumání a znamenávání kvality interakce, než k měření výkonnosti. Používáme techniky dotazování v rozhovoru nebo dotazníku, vyprávění příběhu, či provedení zážitkem a skupinové diskuze (focus group).



## Experimentální metoda

Metoda získává znalosti pomocí kontrolovaných experimentů v laboratoři. Někdy se jí proto také říká vědecká metoda. Získávání znalostí se provádí za účelem nových poznatků nebo ke studiu stávajících poznatků a jejich ověření či vyvrácení, ale také korekci, integraci a rozšíření. Je kladen důraz na přesnost. Tato metoda odstraňuje či potlačuje vnější rušivé faktory reálného světa, které vnáší diversitu a chaos. Kontrolovaný experiment se musí skládat alespoň ze dvou proměnných: *nezávislé* a *závislé*.

Nezávislá proměnná je v HCI většinou nějaká vlastnost rozhraní nebo interakce, která mění jeho konfiguraci. Rozhraní pak je předkládáno uživatelům v odlišných konfiguracích, které jsou otestovány a následně vzájemně porovnány. Pro kontrolovaný experiment je tedy nutné předložit alespoň dvě konfigurace. V HCI se často u nových systémů setkáváme s vyhodnocením použitelnosti nebo uživatelským testováním. Tyto postupy se většinou neřídí experimentální metodou a důvod je jednoduchý: neobsahují nezávislou proměnnou. Vyhodnocení použitelnosti je v těchto případech myšleno jako vyhodnocení jednoho rozhraní z pohledu jeho silných a slabých stránek. Tento postup lze kvalifikovat jako výzkum, ale už ne jako experimentální výzkum. Závislá proměnná je nějaká vlastnost lidského chování, kterou lze sledovat, kvantifikovat a měřit. Nejčastěji je to čas v souvislosti délky trvání dokončení definovaného úkolu. Dobře navržený a provedený experiment umožňuje z naměřených dat odvodit jimi podložené závěry. Vztah nezávislé a závislé proměnné je popisován jako příčina a důsledek. Říkáme tedy, že nastavení rozhraní nebo interakční metody zapříčinilo změny ve sledované závislé proměnné. Tento závěr nelze vyvodit pomocí sledovací nebo korelační metody.

## Korelační metoda

Korelační metoda spočívá ve hledání vztahů mezi proměnnými. Využívá kvantifikace proměnných jako např. věk a plat. Pro proměnné, které nelze kvantifikovat, se vytvoří skupiny např. typ osobnosti a pohlaví. Data se získávají různými způsoby jako pozorování, rozhovory, dotazníky a měření. Korelační metody často doprovází metody experimentální.

## 4.3 Jak připravit a provádět HCI experiment

MacKenzie [8] ve své knize přirovnává návrh experimentu k oddělení signálu od šumu. Signál reprezentuje proměnnou, která nás zajímá (např. interakční technika). Šum je vše ostatní. Tedy náhodné ovlivňující faktory. Sem spadá například vlastnosti okolí jako: teplota, osvětlení, zvuk v pozadí, rozkývaná židle nebo odlesky na obrazovce. Ovlivňujícím faktorem jsou i samotní účastníci, kteří také nesou variabilní složku. Například jejich nálada.

Pokud chceme provádět experiment je nutné vytvořit metodu. Metoda popisuje, jak je experiment navržen a jak se má provést. Zahrnuje rozhodnutí o tom: jak budou zvoleni účastníci, jaký bude použit hardware a software, jaké úkoly budou účastníci provádět a v jakém pořadí, jakým způsobem budou účastníci obeznámeni a připraveni, jaké budou proměnné, jaká data se budou měřit a analyzovat atd. Mít dobře vytvořenou metodu je zásadní. Metoda by se měla skládat z dílčích experimentů, při kterých jsou účastníci vystaveni testovaným podmínkám a přitom se sleduje a měří jejich chování.

### 4.3.1 Návrh experimentu

Návrh experimentu zahrnuje zvolení a definování proměnných, jaké úkoly a procedury budou použity, kolik bude účastníků a jak je získáme. Nejdříve je dobré určit, jaké budou experimentované proměnné. Přemýšlení nad proměnnými nás nutí si klást specifické otázky, které jsou vhodné pro testování. Nejdůležitějšími proměnnými jsou *nezávislé* a *závislé*. Krátce jsem se o nich již zmiňoval v souvislosti experimentální metodou. Když přemýšlíme nad proměnnými, tak výrazy jako *rychleji* nebo *méně kroků* vedou k závislým proměnným, které reprezentují lidské chování jsou měřitelné. Naopak výrazy typu *mé nové rozhraní oproti existujícímu* vedou k nezávislým proměnným. Doporučuje se používat jednu nebo dvě nezávislé proměnné, maximálně tři.

Měření závislých proměnných je doporučováno provádět softwarově na základě časových razítek, stisku tlačítka nebo jiné události zaznamenané pomocí softwaru. Naměřená data by se měly ukládat tak, aby byla připravená pro následnou analýzu. Měly by být označeny jednoznačným kódem, který určuje jakého účastníka a úkolu se týká.

Existují další tři proměnné, které se většinou nezmiňují v experimentu, ale při návrhu je dobré o nich vědět. Jsou to *kontrolní*, *náhodné* a *matoucí*. Kontrolní proměnné jsou takové, které by mohly ovlivnit závislé proměnné, ale nejsou součástí experimentu. Patří sem například osvětlení místnosti, teplota, zvuk v pozadí atd. Je snahou nastavit, aby byli kontrolní proměnné ve všech případech pokud možno neměnné a nemohly tak ovlivnit měřené výsledky. Náhodné proměnné jsou kontrolní proměnné u kterých je povoleno, aby byly náhodné. Vnáší do měření větší variabilitu, ale výhodou je generalizace výsledků. Typicky to jsou vlastnosti účastníků jako: výška, váha, velikost rukou, síla stisku, nálada, nervozita, pohlaví, IQ atd. Matoucí proměnné jsou takové, které ovlivňují výsledky s nezávislou proměnnou. Není pak jasné, jestli naměřené závislé proměnné jsou ovlivněny nezávislou proměnnou nebo matoucí. Příkladem může být experiment za účelem rozhodnout, zda vzdálenost kamery od uživatele ovlivňuje ovládání pohledem. Nezávislou proměnnou experimentu je vzdálenost kamery. V prvním případě je na brýlích účastníka umístěna malá kamera *A* a v dalším je použita speciální snímávací kamera *B* na monitoru. Kamera je matoucí proměnnou, protože pokud jeden z přístupů má výrazný efekt na ovládání může to, že byl způsoben vlastnostmi kamery.

Stěžejní částí experimentu je definování úkolů, které budou účastníci provádět. Při provádění experimentu se úkol nemění, pouze jeho volné proměnné. Úkol by měl reprezentovat aktivitu, kterou uživatelé s rozhraním budou provádět. Měl by být takový, aby při změně nezávislé proměnné umožnil zaznamenat reakce na její změnu, které odhalují přínosy rozhraní nebo jeho potencionální problémy. Často je volba úkolu zřejmá. Pokud se experiment zabývá přínosy zadávání vstupu funkce graficky oproti tradičnímu textovému, tak úkolem bude zadání vstupu funkce oběma způsoby.

Jakmile máme vyřešeny proměnné a úkoly, je čas zaměřit se na účastníky experimentu. Při výběru účastníků se bere v potaz cílová skupina uživatelů. Pokud experimentujeme s rozhraním ovládání přístroje pro operaci mozku, je nežádoucí, aby účastníky byli lidé, kteří nikdy neoperovali. Pokud ovšem není cílem zkoumání vhodnost pro neznalé uživatele. Stejně tak rozhraní mající ulehčit práci s počítačem seniorům, nebudeme testovat na studentech informačních technologií. Dalším parametrem, který musíme zvážit je počet účastníků. Čím větší počet je zvolen, tím více jsou výsledky statisticky významnější. Avšak pokud je sledovaný rozdíl velmi malý, velký počet účastníků jej může učinit hůře odhalitelný. Naopak pokud použijeme účastníků málo výsledky mohou mít slabou statistickou hodnotu a rozdíl se nemusí vůbec objevit. Pokud existuje výzkum na podobné téma, je dobrým pravidlem



použít stejný počet účastníků, jako byl použit v onom výzkumu. V HCI se často provádějí vyhodnocení použitelnosti, u kterých je známo, že malý počet účastníků (většinou 5 expertů) je dostatečný pro odhalení 80% problémů užívání [8]. Ideálně by měli být účastníci vybráni zcela náhodně z populace. Pokud je kladen požadavek na nějakou konkrétní vlastnost (např. 5 let praxe operování), měl by být uveden v popisu experimentu.

### 4.3.2 Návrh dotazníku

Součástí téměř každého experimentu v HCI je dotazník. Slouží ke dvěma účelům. První je získat informace o demografii (věk, pohlaví atd.) a míru zkušenosti s použitou technologií. Dalším je pak získat názory účastníků na použité zařízení nebo interakci, kterou testovali. Dotazníky doprovázející HCI experiment bývají stručné a jejich vyplnění zabere jen pár minut. Otázky v dotazníku mohou být kladeny několika způsoby v závislosti na tom, co nás zajímá. Vyhovující jsou uzavřené otázky, protože omezují účastníkovu odpověď do malé množiny možností. Uzavřené otázky ulehčují následující analýzu. Otevřené otázky naopak umožňují účastníkům lépe vyjádřit svůj názor. Oba přestupy lze kombinovat uvedením otevřené možnosti *ostatní*. Dále se používají intervaly například při dotazu na věk. Často nás zajímá názor nebo pocity uživatele. K zjištění těchto informací se používá Likertova škála. Kdy se dotazujeme na ohodnocení požitku na stupnici od 1 do 7.

### 4.3.3 Provádění experimentu

Jakmile je experiment navržen, vytvořen a otestován, účastníci vybráni a je stanoven termín můžeme začít s testováním. Avšak, než tak učiníme, je vhodné provést pilotní testování s jedním nebo dvěma účastníky, abychom si ověřili, že vše proběhne bez problémů a měli představu o tom jak dlouho testování zabere. Pokud jsme si jisti, že vše je v pořádku můžeme provést ostré testování. Průběh experimentu je následující. Uvítáme účastníka a seznámíme jej s experimentem. Následně jej často necháme vyplnit krátký dotazník na demografické informace a jeho zkušenosti s použitými technologiemi. Představíme účastníkovi testovací stanoviště a vysvětlíme a demonstrujeme mu úkol, který bude provádět. Pokud je to vhodné, můžeme provést pokus na nečisto.

Instrukce poskytnuté účastníkovi hrají důležitou roli a závisí na podstatě úkolu. Instrukce musí být opatrně zváženy a pro každého účastníka stejné. Pokud se účastník dotáže na objasnění problému, musíme být obezřetní, protože jakákoliv další informace může mít za následek odlišné chování účastníka od ostatních. Dohlížející na experiment hraje speciální roli veřejné osobnosti experimentu a měl by vystupovat neutrálně. Účastníci by neměli mít pocit, že je na ně činěn nátlak, aby vytvořili specifický výsledek. Také by neměl být příliš zaujatý, aby nevzbudil v účastnících nervozitu. Naopak by zase neměl být lhostejný a podněcovat tak k lhostejnosti účastníka, která může mít za následek špatnou výkonnost.

## Kapitola 5

# Návrh řešení

V rámci této kapitoly je čtenáři představen problém a způsob, jakým jej ve své práci řeším. Nejdříve definuji cíle, které si tato práce klade a představuji problémy, které je nutno vyřešit. Poté se krátce zastavím nad výběrem zařízení, pro které budu rozhraní navrhovat. Následně vymezuji modelové situace příkladů použití a popisuji problémy interakce, jenž v obecném návrhu řeším a analyzuji. Při analýze představuji dva návrhy způsobu interakce, které nazývám přímá a nepřímá manipulace. Následně uvádím celkový koncept mého řešení a jednotlivých představených problémů interakce. Svá stanoviska stavím na poznacích, které jsem uvedl v předchozích kapitolách.

V kapitole 2 jsem zmapoval dostupná zařízení pro virtuální realitu. Tento přehled jsem vytvořil proto, abych získal představu o tom, jaké jsou v současné době možnosti v oblasti virtuální reality a zvolil zařízení, na které budu rozhraní cílit. Vlastnosti hardwaru jsou totiž z pohledu návrhu uživatelského rozhraní velmi důležité. Definuji možnosti a omezení, se kterými je při návrhu nutné pracovat.

Kapitola 3 pojednává o ovládání počítače a uživatelských rozhraních. Popisuji zde 3DUI, která jsou pro uživatelská rozhraní ve virtuální realitě nutností. Mimo jiné také zmiňuji doporučení, kterými jsem se snažil při návrhu řídit, abych se vyvaroval běžným chybám a navázal na zkušenosti ostatních. Zaměřil jsem se především na způsoby ovládání a interakce, které bych mohl využít ve svém řešení.

### 5.1 Definice cíle a představení problému

Cílem práce je vytvořit rozhraní komponent umožňující ovládat počítač ve virtuální realitě. Zaměřím se na základní způsoby interakce uživatele s virtuálním prostředím, které umožní vytvářet složitější rozhraní pro ovládání dle potřebné aplikace.

Interakce člověka ve virtuální realitě přináší několik problémů, které je potřeba vyřešit. Situace je následující. Uživatel v realitě sedí na židli a má nasazeny brýle pro virtuální realitu. Do brýlí je možné vykreslit libovolnou 3D scénu a umístit jej tak do jakéhokoliv virtuálního prostředí, kterému se také říká virtuální svět. Uživatel se v prostředí může díky snímání pohybu hlavy brýlemi libovolně rozhlížet s volností 6DOF stejně jako v realitě. Brýle tedy zastupují roli **indikátoru** zpětné vazby počítače v modelu interakce člověka s počítačem viz kapitola 4.1 a slouží jako výstupní zařízení. Současně jsou díky snímání pohybu hlavy uživatele v roli **ovladače** pro zadávání příkazů k ovládání počítače, tedy vstupním zařízením. Prvním problémem je způsob volby, jakým bude uživatel interakci **ovládat**. Tedy zadávat své vstupy. K zadávání vstupů můžeme volit z celé řady **odpoví-**

dačů tvořených částmi lidského těla. V kapitole 2.2 jsem popsal zařízení, která se nejčastěji pro ovládání ve virtuální realitě používají a jejich vlastnosti.

Jakmile máme zvolený způsob ovládání, můžeme dále analyzovat možnosti, které nám zvolený způsob ovládání přináší pro interakci s virtuálním světem. V podkapitole 3.2 jsem se věnoval způsobům interakce ve virtuální realitě. Interakci považuji za hlavní komponentu rozhraní. Jelikož se uživatel nachází ve 3D scéně virtuálního světa, tak se dle mého názoru neze vyhnout 3D interakci a 3DUI viz 3.1. Uživatel ve virtuálním prostředí potřebuje způsob, jakým interagovat s objekty, které se v něm nachází. Hlavním problémem interakce uživatele s okolními virtuálními objekty je **manipulace**.

Manipulace se rozděluje na podproblémy **výběru** objektu a jeho **transformace** viz 3.2. Výběr slouží k určení toho, s jakým objektem nebo objekty chce uživatel manipulovat. Transformace pak řeší samotný přesun objektu ve scéně. Tedy **uchopení** objektu, **změnu** jeho **pozice** a **orientace**, a nakonec **uvolnění**. Dalším aspektem, který často používáme při interakci pomocí rukou je **dotyk** prsty. Dotyk si můžeme představit jako formu výběru při manipulaci. Pomocí dotyku lze vytvářet interakce zmáčknutí ovládacího prvku a také oživit virtuální svět a umožnit aby reagoval na dotyk uživatele.

Všechny akce by měly být doprovázeny zpětnou vazbou tak, aby uživateli pomáhala a vedla jej v provádění interakce. Zpětnou vazbou také vytváříme uživateli omezující podmínky prostředí, které mu pomáhají se lépe orientovat. Zpětná vazba a její vizualizace tvoří další problém, který je nutno vyřešit.

Pokud vyřeším všechny tyto problémy, tak umožním uživateli manipulovat s virtuálními a dotýkat se těch v jeho okolí. Pomocí tohoto rozhraní pak lze vytvářet libovolnou abstrakci pro ovládání aplikace v závislosti na tom, co má aplikace umožňovat.

## 5.2 Výběr zařízení pro ovládání rozhraní

Než začneme s návrhem je nutné zvolit zařízení, pro která budu své rozhraní navrhovat. V podkapitole 2.1 jsem se zaměřil na technické parametry jednotlivých virtuálních brýlí. Všichni zástupci jsou až na drobné odchylky velmi vyrovnaní. Za důležité parametry z pohledu navrhovaného uživatelského rozhraní považuji rozlišení a zorné pole. Rozlišení definuje kvalitu zobrazení a odvíjí se od něj například čitelnost textu či přesnost, se kterou je uživatel schopen rozpoznávat své okolí. Zorné pole zase určuje pracovní plochu. Dále jsem sledoval specifické vlastnosti, které jsou zajímavé pro ovládání. Jako například snímání pohybu očí a možnosti snímání pohybu hlavy. Většina zařízení podporuje snímání pohybu hlavy s volností 6DOF. Naopak snímání pohledu umožňují pouze brýle *Fove*. Je dobré volit zařízení, která jsou populární a rozšířená mezi uživateli. Proto se zaměřím na brýle jako *Oculus Rift DK2* nebo *HTC Vive*. Konkrétně využiji brýle *Oculus Rift DK2*, které jsou na trhu nejdéle.

Brýle samotné neumožňují dostatečný způsob ovládání. Zkoumal jsem se proto také dodatečná zařízení pro snímání pohybu uživatele 2.2. Za nejzajímavější považuji snímání pohybu rukou, které přináší zajímavé možnosti z pohledu interakce s uživatelským rozhraním. Ruce jsou nejpřirozenější způsob, jakým člověk interaguje se svým okolím a používají se jako hlavní způsob ovládání většiny běžných uživatelských rozhraní. Všechny zařízení pro snímání rukou podporují volnost 6DOF a liší se v přesnosti a latenci snímání. Předpokládám, že rukavice a speciální zařízení do rukou budou mít lepší přesnost, avšak mohou být pro uživatele méně pohodlné. Nejrozšířenějším zařízením je senzor *LeapMotion*, který je určen přímo pro snímání pohybu rukou a prstů. Senzor má také širokou podporu pro vývojáře a je cenově dostupný.

## 5.3 Analýza problému a možnosti jeho řešení

Uživatel se tedy nachází ve 3D scéně, kde je mu známa poloha uživatelské hlavy a její orientace spolu s polohou rukou a prstů. Díky snímání rukou můžeme promítnout do scény jejich virtuální reprezentaci. Virtuální ruce bude uživatel používat k interakci se svým okolím stejně jako v realitě tak, aby je mohl použít pro ovládání různých virtuálních prvků v jejich dosahu i mimo něj.

### 5.3.1 Vymezení pracovního prostředí

Uživatel bude sedět na místě, bez možnosti se v realitě přesouvat. Kolem sebe bude mít volný prostor, tak aby se v něm mohl naklánět a volně pohybovat rukama v celém jejich dosahu. Tento prostor mu bude sloužit jako pracovní prostředí ve virtuální realitě. Celková velikost pracovního prostoru je ve virtuální realitě omezena rozsahem snímání použitých zařízení. Konkrétně pohybem hlavy a zorným polem, které jsou brýle schopny zobrazit v kombinaci s prostorem snímání rukou. Rozsah snímání zařízením LeapMotion a zorné pole Oculus Rift DK2 je znázorněno na obrázku 5.1. Snahou je, aby snímání prostoru v ideálním případě pokryl celé okolí uživatele v dosahu jeho rukou. Toho můžeme s pomocí zvoleným zařízením dosáhnout díky možnosti natáčení hlavy a upevnění senzoru LeapMotion na přední straně brýlí (v režimu HMD).



Obrázek 5.1: Rozsah snímání rukou LeapMotion a zorné pole brýlí Oculus Rift DK2

### Možnosti interakce

Okolí uživatele vizualizované ve virtuální realitě může být jakékoliv a z pohledu mé práce jej nepovažuji za důležité. Důležitý je způsob, jakým bude uživatel se svým okolím interagovat. Virtuální svět se stejně jako v realitě skládá z virtuálních objektů, se kterými chce uživatel interagovat pomocí rukou. Prvním problémem je manipulace s těmito virtuálními objekty. Dále uživatel požaduje, aby virtuální objekty byly živé a reagovali na dotyk jeho virtuální ruky. Proto je nutné detekovat kolize a simulovat pohyb objektů na základě fyzikálních zákonů známých z reálného světa.

Při manipulaci musíme vyřešit problém **výběru** objektů a jeho následné **uchopení**, **přesun**, změnu **orientace** a **uvolnění**. Pro objekty, které jsou v dosahu uživatele se nabízí přístup **přímé manipulace** viz 3.2.2. Tedy jistá simulace interakce pomocí rukou v reálném



světě. Zde existují dva směry, které se často používají: abstraktní přístup pomocí gest či fyzikální simulace.

Fyzikální simulace pracuje na principu kolizí, deformací a tření. Tento přístup je zajímavý, avšak z praktického hlediska je oproti abstraktním přístupům velmi náročný na implementaci a především také klade vysoké požadavky na přesnost snímání rukou. Pokud při manipulaci s uchopeným objektem dojde k nepřesnosti snímání polohy prstu, která zapříčiní, že prst se přestane dotýkat objektu, s velkou pravděpodobností dojde k jeho upuštění. Protože jsem se rozhodl použít sensorické snímání rukou infračerveným světlem, které rozpoznává ruce a prsty z hloubkového obrazu a je méně přesné, nepovažuji tento přístup za vhodný.

Můžeme tedy využít přístup abstraktní, který výběr a uchopení zjednodušuje například pomocí **gest**. Výběr lze provést na základě blízkosti ruky a objektu. Samotné uchopení se pak provede definovaným gestem. Objekt se po uchopení připevní k nějakému bodu ruky a provádí se jeho posun a změna orientace v závislosti na pohybu ruky. Uvolnění objektu se provede při detekci přerušování provádění gesta uchopení.

Přímá manipulace má však zásadní nevýhodu. Nelze interagovat s objekty mimo dosah rukou uživatele. Možným řešením je umožnit uživateli se ve scéně přesouvat. Avšak uživatel v realitě většinou sedí na nějakém místě a jeho přesunutí by bylo nepřirozené i když existují způsoby, které tento problém řeší. Větším problémem je však neefektivita provádění takovéto manipulace. Uživatel by se musel neustále přesouvat za cílovými objekty. Existují proto tzv. hybridní manipulační techniky pro **nepřímou manipulaci** viz 3.2.4, které používají k výběru vzdálených objektů *paprsek*. Paprsek lze využít pro různé magické způsoby interakce ve virtuální realitě a vytvářet tak zajímavé způsoby ovládnutí, které jsou efektivní nebo nějakým způsobem usnadňují provádění úkolů. Použitím paprsku efektivně rozšiřuje pracovní prostor uživatele za hranice dosahu jeho rukou.

Jednou z možností je využít pro práci s paprskem snímání pohybu hlavy a připevnit paprsek mezi oči uživatele. Uživatel by pak prováděl výběr pomocí pohledu na daný objekt. Další možností je využít pro výběr ruce. Například vysílat paprsek z ukazováku. S vybraným objektem pak uživatel může manipulovat stejně, jako by jej přímo uchopil. Avšak tento přístup by mu dovolal s objektem pohybovat pouze v jeho lokálním okolí, kde se nachází. To je dosti omezující. Většinou uživatel potřebuje objekt přesouvat z pozice, kde se právě sám nachází. Je proto potřeba navrhnout způsob, jak tento problém řešit. Řešení by se mělo snažit vycházet z reality tak, aby bylo pokud možno co nejvíce přirozené. Řešením je objekt připevnit na pomyslnou přímkou od uživateli ruky. Uživatel by s objektem pak mohl manipulovat, jako by držel tuto přímkou v ruce.

Posledním představeným problémem je možnost dotyku okolních objektů. Tento problém lze řešit pomocí fyzických kolizí modelů. Dotyk lze také využít jako způsob interakce pro ovládnutí libovolných prvků. Příkladem mohou být klasická tlačítka, která lze použít pro ovládnutí aplikace pomocí menu.

### 5.3.2 Návrh způsobů interakce s virtuálními objekty

Na základě představených možností interakce jsem navrhl tři metody interakce s virtuálními objekty, které umožňují objekty přesouvat ve scéně. Dále jsem navrhl interakci pomocí dotyku virtuálního objektu, kterou lze použít pro vytváření virtuálních ovládacích prvků. Například tlačítka pro tvorbu menu. Navržené metody popisují pomocí modelových situací vytvoření věže z virtuálních kostek a jehlanu níže.

### **Metoda přímé manipulace**

Uživatel se nachází ve virtuální místnosti a před sebou má na stole dvě kostky a jehlan, ze kterých chce poskládat věž. Protože má objekty v dosahu rukou použije metodu přímé manipulace. Ve scéně vidí virtuální reprezentaci rukou, které kopírují pohyb jeho rukou v realitě na základě dat ze senzoru. Uživatel přesune ruku k objektu, který chce uchopit a v moment, kdy je možné jej uchopit se objekt zvýrazní. Poté provede gesto uchopení objektu, které vybraný objekt přichytí k ruce, jenž gesto provedla. Uchopení je opět vizualizováno odlišným zvýrazněním uchopeného objektu. Poté se objekt pohybuje s rukou dokud uživatel nepřeruší gesto pro jeho uchopení a objekt je upuštěn. Tímto způsobem přesune první kostku. Potom levou rukou uchopí další kostku a chce si ji předat do pravé ruky, aby mohl levou rukou uchopit jehlan. Provede gesto pravou rukou v blízkosti uchopené kostky, aby ji uchopil oběma rukama a poté kostku z levé ruky uvolní. Nakonec volnou rukou uchopí jehlan a dokončí stavbu věže.

### **Metoda nepřímé manipulace s výběrem pomocí pohledu**

Uživatel plní stejný úkol jako v případě přímé manipulace s tím rozdílem, že kostky jsou mimo jeho dosah. Pro výběr použije zaměřovač, který má ve středu svého pohledu. Pomocí zaměřovače vybere kostku, která se zvýrazní. Zvýrazněnou kostku uchopí gestem a pracuje s ní, jako by ji držel upevněnou na tyči v ruce, kterou provedl gesto. Tímto způsobem položí kostky na sebe. Nakonec vybere jehlan, který je převrácený a musí ho otočit. Uchopí jej oběma rukama a jako by hranol držel na tyči mezi rukama jej natočí do požadované pozice a umístí na vrchol věže.

### **Metoda nepřímé manipulace s výběrem pomocí ruky**

Úkol zůstává stejný jako při nepřímé manipulaci popsané výše, ale místo výběru pohledem použije uživatel k výběru ruku. Na jeho dlaních jsou zvýrazněny body, ze kterých při jejich propnutí začne vycházet paprsek. Pomocí paprsku vybere cílový objekt. Dále již pokračuje stejným způsobem.

### **Interakce pomocí dotyku**

Pro postavení věže uživateli chybí jehlan. Ve scéně je objekt tlačítka, který vytváří jehlany. Uživatel na tlačítko klikne pravým ukazovákem a vytvoří nový objekt jehlanu. Po sestavení věže ji zboří pěstí.

## **5.4 Koncept systému**

Pro návrh systému jsem se rozhodl použít návrhový vzor model MVC (Model View Controller), který slouží k návrhu uživatelských rozhraní. Rozděluje vytvářené softwarové řešení na tři části, které spolu komunikují tak, že odděluje interní reprezentaci dat a způsoby získávání a zobrazování informace. Grafická reprezentace návrhu pomocí MVC je uvedena na obrázcích 5.2.

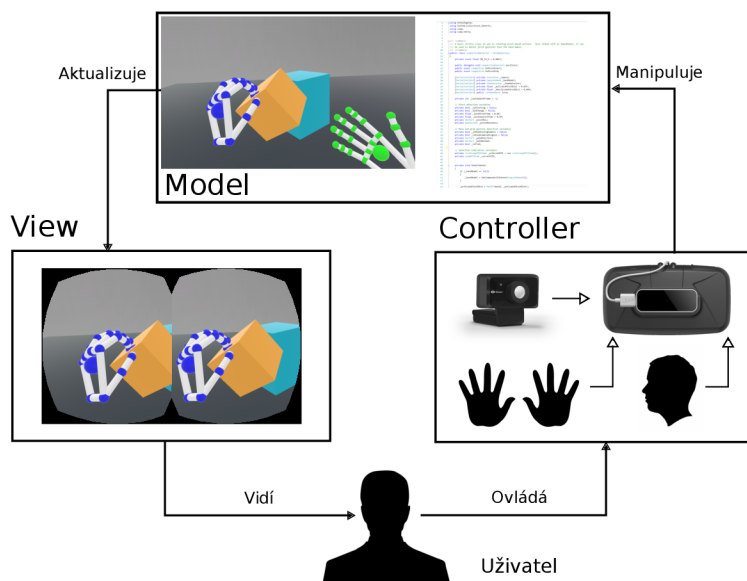
Model pro mne bude reprezentovat 3D scénu virtuálního světa a logiku pro jeho upravení. Bude uchovávat informace o objektech ve scéně jako je jejich stav, pozice, rotace, 3D model, materiál textury, logiku chování objektu atd. Dále se bude skládat z logického



modelu rozhraní, který bude ovládat změnu modelu scény a implementovat navržené způsoby interakce. Model získává vstupy od Controlleru a poskytuje informace do View pro jejich zobrazení uživateli.

Controller se stará o získávání vstupních dat od uživatele a jejich zpracování pro Model. Pro mne bude tvořit rozhraní pro získávání a zpracovávání dat z modelu ruky senzoru LeapMotion a modelu hlavy z brýlí Oculus Rift DK2. Bude se starat o detekci gest a zpracovaná data o ruce a hlavě bude poskytovat modelu.

View je část aplikace, která slouží k vizualizaci zpětné vazby akcí, jež uživatel provádí s rozhraním. View se bude starat o vykreslování 3D scény virtuálního světa do brýlí.



Obrázek 5.2: Koncept systému pro ovládání počítače pomocí MVC

## 5.5 Analýza a způsob řešení jednotlivých částí problému

V této kapitole jsou podrobně popsány způsoby řešení všech dílčích částí navržených metod interakce.

### 5.5.1 Detekce gest ruky

Na základě dat o ruce získaných ze senzoru LeapMotion budu kontrolovat pózu ruky a v případě, že detekuji gesto definované gesto, vyvolám událost o jeho detekování, kterou rozšířím mezi všechny zájemce o ni. Pro uchopení objektu použiji gesto štípnutí. Gesto budu detekovat na základě vzdálenosti mezi konci prstů ukazováku a palce. Práce s rukou je součástí Controlleru.

### 5.5.2 Výběr objektu při přímé manipulaci

Při manipulaci budu pro uchopení objektu používat gesto štípnutí ukazovákem a palcem. K výběru objektu se proto nabízí využít bodu mezi těmito prsty. Objekt bude vybrán v případě, že se ho tento bod dotýká. Indikaci vybraného objektu uživateli provedeme jeho grafickým zvýrazněním. Pozici bodu pro výběr lze určit z pozic konců prstů. S těmito

informacemi se pracuje při kontrole stavu ruky. Při kontrole se tedy může tento bod určit a sledovat, zda se dotýká nějakého objektu či ne. Aby bylo možné zrušit výběr objektu musíme si pamatovat objekt, který byl vybrán jako poslední. V případě, že se bod objektu přestane dotýkat, zrušíme označení výběru zapamatovaného objektu a indikujeme změnu odebráním jeho zvýraznění.

### 5.5.3 Výběr vzdáleného objektu

Jedná se o způsob výběru vzdáleného objektu pro nepřímou manipulaci. Navrhl jsem dvě varianty výběru: **pohledem** a **rukou**. Oba přístupy pracují s paprskem. Pokud paprsek trefí nějaký manipulovatelný objekt, tak jej označíme jako vybraný a provedeme zpětnou vazbu uživateli jeho zvýrazněním. Rozhraní bude umožňovat dva režimy výběru. Buď může být vybrán pouze jeden objekt nebo více objektů současně. V režimu jednoho objektu se vždy vybere pouze poslední objekt trefený paprskem. Tedy pokud je vybrán nějaký jiný objekt, tak musí být jeho výběr zrušen. Stejně tak při trefení již vybraného objektu se jeho výběr zruší. V režimu více objektu může uživatel paprskem vybrat libovolný počet objektů a opětovným trefením již vybraného, jeho výběr opět zrušit a odstranit ho tak z množiny vybraných objektů.

Výběr pohledem lze použít díky tomu, že máme k dispozici informace o změnách pozice hlavy. Můžeme tedy na pozici mezi oči uživatele umístit bod, ze kterého vyšleme paprsek do scény a provádět jím výše popsany výběr. Přítomnost paprsku by měla být uživateli nějakým způsobem indikována. Indikaci provedeme pomocí vykreslení zaměřovače, který mu také usnadní míření na cílové objekty.

Druhá varianta výběru pomocí rukou využije data o pozici kostí ruky. V první verzi návrhu jsem umístil paprsek na konec ukazováku ruky tak, aby směřoval ve směru pokračování prstu. Zamýšlel jsem, že by uživatel mohl provádět ovládání paprsku gestem pistole. Paprsek by začal vycházet z prstu v případě, že je gesto detekováno. Aby uživatel mohl mířit, tak by byl paprsek na rozdíl od výběru pohledem vykreslován do scény. Možnost výběru paprskem by byla indikována uživateli rozsvíceným koncem prstu. Při implementaci toho návrhu jsem však narazil na problém, že směr paprsku byl z důvodu nepřesnosti nestabilní a mírně se neustále měnil. Mohlo za to nepřesné snímání prstu, i když se prst v realitě nepohyboval. Vzniklá chyba se navíc se zvyšující vzdáleností cíle zvyšovala a nestabilita se tak projevovala více. Rozhodl jsem se proto místo prstu použít normálu vycházející ze středu dlaně. Snímání dlaně je již oproti směru prstu daleko přesnější. Spuštění paprsku uživatel provede propnutím dlaně a indikace je opět řešena rozsvícením bodu, odkud je paprsek vysílán spolu s jeho vykreslením.

### 5.5.4 Uchopení a uvolnění objektu

Uchopení a uvolnění objektu bude zadáváno pomocí gesta štípnutí palcem a ukazovákem, jak jsem již dříve zmiňoval. O detekci započatí a přerušení gesta se bude starat detektor gest, který umožňuje o detekci akce informovat zájemce pomocí události. Každý manipulovatelný objekt se tedy přihlásí k odběru těchto událostí. Logika uchopení objektu bude následující. V případě přímé manipulace se uchopí pouze objekt, který je vybrán a nachází se v blízkosti bodu mezi palcem a ukazovákem ruky, která právě provedla gesto. Naopak, když je detekováno přerušení gesta, tak bude uchopený objekt uvolněn. Objekt může být současně uchopen oběma rukama, proto je nutné uchovávat informaci o uchopení pro každou ruku. Uchopení objektu znamená jeho připevnění k bodu štípnutí (mezi ukazovákem a palcem). K uvolnění objektu dojde, když není objekt uchopen ani jednou rukou. Uživatel

tedy může každou rukou uchopit jiný objekt nebo uchopit objekt stejný a předávat si jej mezi rukama. Podobná logika se použije i pro nepřímou manipulaci. Avšak objekt nemusí být v blízkosti ruky a při detekci gesta jsou k bodu štípnutí připevněny všechny vybrané objekty. Při nepřímé manipulaci tedy nelze uchopit každou rukou jiné objekty. To je žádoucí z důvodu zamýšlené interakce se vzdálenými objekty uchopenými oběma rukama zároveň.

### 5.5.5 Transformace objektu při manipulaci

Pod pojmem transformace objektu chápu změnu jeho pozice a orientace. Pozice a orientace objektu se vztahuje vůči trojrozměrnému souřadnému systému scény. Pozice bývá reprezentována vektorem souřadnic. Orientace bývá reprezentována vektorem eulerovských úhlu reprezentující změnu natočení v každé rovině nebo tzv. kvaternionem. V počítačové grafice se transformace provádí pomocí transformačních matic a řeší ji herní engine. K transformaci dochází poté co je objekt uchopen.

Každému manipulovatelnému objektu přiřadíme tzv. kotvu. Kotvu si lze představit jako neviditelný objekt, který má také nějakou svoji pozici a orientaci v souřadném systému. Při uchopení objektu gestem štípnutí připevníme kotvu k místu dotyku prstů a objekt, který k ní patří poté přesouváme relativně vůči ní. Tedy pokud se změní pozice kotvy, změní se o stejný úsek pozice objektu. Stejně tomu je i v případě změny orientace. Posuv kotvy a změnu její orientace provedeme v závislosti na posunu bodu dotyku ruky a jejím natočením. V případě přímé manipulace se ve výsledku objekt pohybuje jako by byl v daném bodě uchopen.

Při nepřímé manipulaci s objektem se objekt pohybuje, jako by byl připevněn na drátu, který uživatel drží mezi prsty. Výše popsany postup zachycuje případ, kdy je objekt uchopen pouze jednou rukou. Pohyb objektem oběma rukama se liší pro přímou a nepřímou manipulaci. U přímé manipulace je kotva umístěná na středu přímky mezi místy uchopení. Posuv se opět provádí o změnu vůči poslední pozici tohoto bodu. Rotace se provede pomocí výpočtu interpolace mezi rotacemi obou ruk v polovině intervalu a kotva se poté zarovná s touto rotací. Nepřímá manipulace se liší v tom, že i objekt se přesune na střed přímky mezi místy uchopení. Následně provedeme posun s rotací a pak je objekt vrácen zpět na původní pozici. Pohyb s objektem se v tomto případě chová, jako by byl připevněn na drátu mezi rukama a přitom se nachází jinde. Díky tomu je možné rotovat a pohybovat se vzdáleným objektem namíste.

### 5.5.6 Řešení interakce pomocí dotyku a návrh menu

Interakci pomocí dotyku lze řešit pomocí kolizí objektů. Kostí ruky budou reprezentovány kolizními objekty. Díky kolizím lze simulovat kolize ruky s prostředím jako v reálném světě.

Pro ovládání pomocí dotyku použijí pouze poslední kost ukazováku. Ovládací prvky budou reagovat pouze na dotyk s pravým ukazovákem pomocí detekce kolizí s jeho kostí. Pokud bude detekován začátek kolize, tak se nastaví stav dotyku objektu. Naopak při přerušování kolize se změní stav dotyku zpět. Na tyto události lze reagovat v závislosti na požadované funkčnosti. Výše popsany způsob interakce použijí pro mačkání tlačítek v menu.

Menu ve virtuální realitě implementujeme pomocí 3D objektů ve scéně, které někam umístíme. Možným řešením je vytvořit virtuální panel s ovládacími prvky napevno umístěný ve scéně nebo vytvořit ovladač, se kterým může uživatel volně manipulovat a umístit jej někam do scény. Tyto přístupy kopírují realitu. Jejich nevýhodou je nutnost soustředit se na umístění menu ve scéně. Dobrou praktikou je menu umísťovat tak, aby k němu měl uživatel neustálý a snadný přístup. Proto je vhodné využít nějaký magický přístup. Já

použiji menu, které bude připevněné k levé ruce a bude se zobrazovat pouze pokud dlaň ruky směřuje směrem k pohledu uživatele. Menu bude jednoduché, složené pouze z tlačítek. Bude podporovat tlačítka typu: zapnuto-vypnuto, přepínání mezi více stavy, výběr jednoho z více tlačítek (jako u ovládání přehrávání kazet na rádiu). Stavy tlačítka budou vizualizovat změnou jeho podkladové barvy (zapnuté, vypnuté, zamáčknuté, vymáčknuté, zmáčknuté). Tímto způsobem lze vytvořit libovolné ovládací prvky dle požadavků aplikace. Například jezdec či interaktivně měnící se menu.

# Kapitola 6

## Realizace

V následujícím textu dokumentuji proces práce při vývoji navržených komponent rozhraní a následný postup jeho vyhodnocování. Kapitola obsahuje informace o implementaci, experimentech, získaných poznatcích při testování a jejich využití pro aktualizaci rozhraní.

Nejprve popisují volbu nástrojů použitých pro implementaci dle zvolených zařízení. Uvádím informace o jejich integraci a způsobu práce s nimi. Potom se již věnuji implementaci jednotlivých částí návrhu z předchozí kapitoly, aby čtenář získal představu o tom, jakým způsobem jsou problémy vyřešeny a jak spolu jednotlivé části pracují. Dále se věnuji návrhu experimentu pro otestování vytvořených komponent a popisují způsob jeho provedení. Čtenář se dozví informace o testovaných úkolech, účastnících, prostředí, měřených hodnotách, dotazovaných otázkách a v poslední řadě o tom, jak budou komponenty vyhodnocovány. Naměřené hodnoty využiji ke srovnání způsobů interakce z pohledu jejich výkonnosti a získaná zpětná vazba od uživatelů poslouží k vylepšení rozhraní a vyhodnocení použitelnosti interakčních technik z pohledu uživatele. V závěrečné části kapitoly popisují aktualizaci rozhraní na základě poznatků získaných při experimentu a krátce popisují aplikace, které demonstrují využití navrženého rozhraní.

### 6.1 Zvolené nástroje pro implementaci a jejich integrace

Výrobci zařízení poskytují vývojářům tzv. **SDK** (software development kit). SDK obsahuje balíčky nástrojů pro implementaci aplikací využívající dané zařízení. SDK umožňuje se zařízením komunikovat pomocí API. Což je kód implementující rozhraní, který většinou bývá poskytován ve formě knihovny. Rozhraní většinou bývá udržováno ve více programovacích jazycích pomocí tzv. wrapperů, aby měl vývojář volnost výběru. Nejčastěji jsou podporovány programovací jazyky C/C++, Objective-C, C# a Java.

Vývoj aplikace pro virtuální realitu znamená práci s **3D grafikou**. Pro vývoj 3D aplikací existují tzv. **herní engine**, které poskytují kompletní framework pro usnadnění práce. Implementují všechny potřebné multimediální části jako například: vykreslování, zvuk, logiku, fyziku, síťovou komunikaci a umělou inteligenci. Vše je obsaženo v rámci editoru, který poskytuje funkčnost formou nástrojů a umožňuje vyvíjet a vydávat aplikace pro různé platformy.

Virtuální brýle Oculus Rift DK2 jsou podporovány většinou předních herních engineů jako Unity 5, Unreal Engine 4 a CryEngine 3. Vývojáři obou stran spolu spolupracují na integraci SDK do svých herních engineů a je pak snadné zmíněná zařízení v aplikaci zprovoznit. Často se vše dá provést pomocí nastavení uvnitř editoru. Vývojář má pak přístup

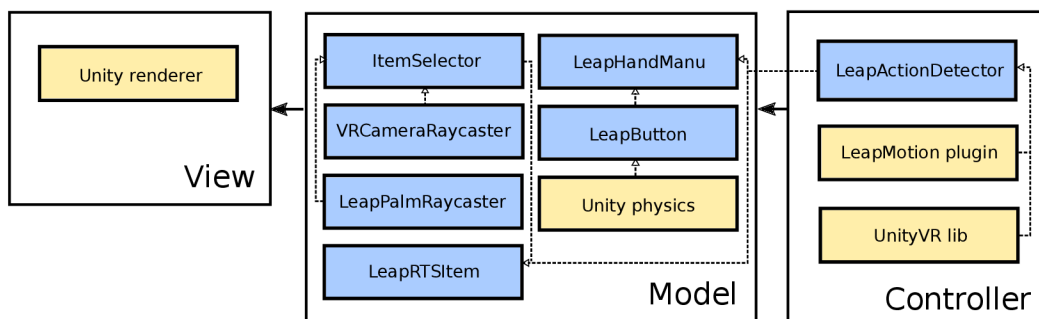


k rozhraní SDK přes knihovny v rámci jmenného prostoru engine. Rozhraní zařízení Oculus Rift DK2 je implementováno v jazyce C/C++. Podobná situace je i pro senzor LeapMotion, který má své rozhraní implementované také v jazyce C/C++. LeapMotion lze integrovat do engineů Unity 5 a Unreal Engine 4. Unity 5 jej integruje pomocí oficiálního balíku, který se nahraje do projektu. Balík používá wrapper jazyka C# a obsahuje i tzv. prefabs pro reprezentaci rukou ve scéně. Integrace do Unreal Engine je řešena pomocí původně komunitního pluginu leap-ue4, který se stal od verze engineu 4.11 oficiálním a je jeho součástí. Plugin je napsán v jazyce C++ a stačí jej aktivovat v editoru.

Integrace zařízení v herních enginech umožňuje rychlé započatí vývoje. Unreal Engine 4 a Unity 5 jsou pro nekomerční účely volně dostupné a díky integraci obou vybraných zařízení jsou dobrými kandidáty pro použití. Oba zástupci jsou na podobné úrovni ve většině aspektech a proto volba jednoho z nich závisí především na zkušenostech a preferencích vývojáře. Ani s jedním engineem jsem neměl dřívější zkušenosti. Nejdříve jsem se rozhodl pracovat s Unreal Engine 4, protože používá jazyk C++. Avšak narazil jsem na problém se špatnou přesností snímání rukou. Krátce poté byla vydána nová beta verze SDK pod názvem Orion, která výrazně zlepšuje přesnost a robustnost snímání senzoru LeapMotion. Nové SDK však bylo v té době integrované pouze pro engine **Unity 5**. Z tohoto důvodu jsem se rozhodl přejít na tento engine.

## 6.2 Implementace navrženého rozhraní

K implementaci jsem použil herní engine Unity 5 spolu s brýlemi Oculus Rift DK2 a senzorem LeapMotion. Způsob jejich integrace je popsán v podkapitole 6.1. Vyvíjel jsem na operačním systému Windows 7 v jazyce C#, kde je k používání zařízení potřeba nainstalovat Oculus Runtime s SDK (verze 0.8) a LeapMotion Orion SDK (verze 3.1.2).



Obrázek 6.1: Struktura komponent aplikace ovládání počítače

Rozhraní se skládá z komponent znázorněných na obrázku 6.1. Žlutou barvou jsou označeny knihovny a systémy, které využívám. Modrou barvou jsou zvýrazněny komponenty, které vytváří jádro aplikace. *LeapActionDetector* se stará o rozpoznávání akcí spojených s rukou a informuje o nich událostmi. *LeapRTSItem* řeší manipulaci s objektem pomocí navržených technik interakce. Pokud chceme, aby bylo možné s objektem manipulovat, přiřadíme mu tuto komponentu. Komponenta *ItemSelector* řídí výběr objektů jednou z představených metod. Spolupracuje s *VRCameraRaycaster* a *LeapPalmRaycaster*, které implementují způsob výběru objektu paprskem. *LeapHandMenu* implementuje menu zobrazované při pohledu na dlaň levé ruky. Skládá se z tlačítek *LeapButton*, která reagují na dotyk praveho ukazovaku.

V následujících podkapitolách se blíže věnuji způsobům implementace řešení jednotlivých problémů. Tyto podkapitoly obsahují podrobnější informace komponentách, které uvádí do souvislosti a čtenáři tak pomohou lépe pochopit princip, jakým spolu spolupracují.

### 6.2.1 Detekce gesta štípnutí

O detekci gesta štípnutí se stejně jako o všechny ostatní akce spojené s rukou uživatele stará třída *LeapActionDetector*. V aplikaci musí být dvě instance této třídy. Každá pro jednu ruku.

Senzor *LeapMotion* poskytuje data o rukách po snímcích ve třídě *Frame*. Každý snímek obsahuje aktuální reprezentace obou ruk. Ruku tvoří třída *Hand*, která obsahuje všechny informace o ruce ze senzoru. Reprezentace ruky se skládá z paže (třída *Arm*), pěti prstů (seznam *Fingers* objektů třídy *Finger*) a vlastností jako: směr (*Direction*), pozice dlaně (*PalmPosition*) a orientace (*Basis*). Každý prst se skládá ze čtyř kostí reprezentovaných třídou *Bone*. Prst navíc uchovává pozici špičky prstu a směr. Kost nese informace o pozicích kloubů z obou stran kosti, pozici středu kosti, délce, orientaci a typu kosti.

Detektor je navržen tak, aby komunikoval s třídou *CapsuleHand*, která je součástí balíku *LeapMotion* pro Unity 5 a obstarává vykreslování ruky do scény. Při každé změně snímku vykresluje ruku pomocí primitiv v Unity 5. O získávání aktuálních dat ze senzoru se stará třída *LeapServiceProvider*. Mimo to slouží i jako rozhraní pro přístup k datům o snímané ruce ve *Hand*. Detekce gesta se provádí po snímcích. Při každé změně snímku se sleduje vzdálenost mezi konci ukazováku a palce. Vzdálenost lze získat přímo z rozhraní. Pokud tato vzdálenost překročí některou z nastavených hranic pro aktivaci či deaktivaci gesta (3cm), tak dojde k zaznamenání změny stavu a je vyvolána odpovídající událost. Událost obsahuje informaci o pozici detekovaného štípnutí ve formě parametru.

K šíření zpráv o detekci gesta používám systém událostí jazyka C#. Kdokoliv je přihlášen k odběru této události je o ni informován. Informace o stavu gesta jsou také zpřístupněné přes metody detektoru.

### 6.2.2 Výběr objektu

Navrhl jsem tři způsoby výběru objektu. Prvním je výběr bodem mezi palcem a ukazovákem při přímé manipulaci. Další dva způsoby používají k výběru paprsek. Jeden paprsek se ovládá pomocí pohybů hlavy a slouží k výběru pohledem. Další vystřeluje z dlaní uživatele.

Objekty při přímé manipulaci vybírá *LeapActionDetector*. Při změně snímku vytvoří na pozici mezi palcem a ukazovákem kouli a detekuje, zda nedochází ke kolizím. Využívám metody *OverlapSphere* Unity 5, která vrací objekty v poli. K uchování informace o naposledy vybraných objektech se používá seznam. Původní vybrané objekty jsou odznačeny a odstraněny ze seznamu. Naopak nově nalezené objekty označí jako vybrané a přidá je do seznamu.

Výběr vzdálených objektů řeší komponenta *ItemSelector*, která nese informace o používaném způsobu interakce a režimu výběru. Podle nastavené interakce zapíná příslušné způsoby výběru a režim určuje, zda lze vybrat pouze jeden objekt nebo více současně. Paprsek a jeho ovládání implementují komponenty *VRCameraRaycaster* a *LeapPalmRaycaster*. Obě komponenty využívají pro paprsek fyzikálního systému Unity 5. Paprsek implementuje třída *Ray*. Je definován bodem, ze kterého má vycházet a vektorem určující jeho směr. Paprsek lze do scény vyslat metodou *Raycast* do definované vzdálenosti. Pokud je zasažen nějaký ob-

jekt, tak je nastaven do proměnné typu *RayCastHit*. Komponenty vrací informaci o zásahu třídě *ItemSelector*, která následně provádí logiku výběru.

V režimu výběru pouze jednoho objektu si pamatuje poslední vybraný a zasažený objekt. Pokud byl zasažen nový objekt, tak se zruší výběr posledního vybraného. Nově zasažený objekt se vybere a uloží jako poslední vybraný objekt. Pokud je zasaženým objektem současně poslední vybraný objekt, tak se invertuje jeho hodnota výběru. Když není zasažen žádný objekt, tak se informace o posledním zasaženém objektu vyprázdní. Díky tomu lze vybírat jeden objekt ve scéně a opětovným zasažením výběr zrušit.

Režim výběru více objektů pracuje podobně. Jen si vybrané objekty ukládá do slovníku podle jejich jednoznačného ID. A při trefení jakéhokoliv nového objektu invertuje hodnotu výběru a v případě, že ve výsledku má být objekt vybrán, tak jej vloží do slovníku. Naopak pokud má být odebrán, tak jej ze slovníku odstraní.

Komponenta *VRCameraRaycaster* používá výše popsany způsob vysílání paprsku do scény pro odhalení vzdálených objektů. Jako zdroj paprsku používám střed kamery a jeho orientaci nastavuji dle vektoru směřujícího dopředu z ní. Paprsek indikuji zaměřovačem formou bodu. O vykreslení bodu zaměřovače se stará třída *Reticle*. Bod je vykreslován na pozici, kam se trefil paprsek. Bod je nutné vykreslovat do scény na místo jeho dopadu, jinak by docházelo k jeho zdvojení způsobeného binokulární disparitou viz 2.1.

*LeapPalmRaycaster* lze použít na každou ruku. Musí mít přístup ke třídě *LeapActionDetector*, ze které získává informace a bodu ve středu dlaně a vektoru normály dlaně. Místo klasického paprsku vysílám do scény válec. Důvodem jsou lepší detekce výběru. Při použití paprsku docházelo kvůli jeho mírnému kolísání k několikanásobnému detekování výběru jednoho objektu, když jej paprsek trefoval na hranu. Díky použití válce je toto kolísání zmírněno. Tento jev se ještě víc objevoval při použití paprsku z prstu, od kterého jsem nakonec upustil. Paprsek se ovládá propnutím dlaně, které sleduje *LeapActionDetector*. Zpětnou vazbu uživateli řeším vykreslením paprsku do scény přímkou mezi zdrojem a jeho cílem. Indikaci možnosti použití paprsku vizualizuji rozsvícením bodu ve středu dlaní.

### 6.2.3 Manipulace s objektem

Manipulaci řeší třída *LeapRTSItem*, která nese informace o tom, zda je objekt vybrán nebo uchopen a jakou rukou. Podle těchto stavů se nastavuje barva jeho materiálu, což slouží k vizualizování zpětné vazby uživateli. Její instance se při vytvoření registruje k odběru událostí o gestu štípnutí od obou detektorů *LeapActionDetector*. Rozlišují se dvě události: započítání gesta nebo ukončení gesta. Na tyto události třída reaguje změnou stavu o uchopení objektu každou z ruk. Při uchopení objektu dojde k přerušení působení gravitace na něj a je změněna barva materiálu objektu tak, aby bylo uživateli indikováno, že je uchopen. Při uvolnění objektu se kontroluje, zda není zároveň uchopen druhou rukou. Pokud ne, tak je obnoveno působení gravitace, což má za následek upuštění objektu. Uchopit lze pouze objekt, který je vybraný.

Pro transformaci objektu se využívá principu kotvy dle návrhu. Kotva je reprezentována neviditelným objektem. Pro pohybování s objekty v Unity 5 se používá třída *Transform*, která je členem každého 3D objektu ve scéně. Objekty lze řadit do stromové struktury. Při pohybu s rodičovským objektem se pohybují i jeho potomci. A to tak, aby byla zachována jejich pozice vůči rodičovskému objektu. Objekt kotvy se vytváří současně s instancí *LeapRTSItem*. Po vytvoření se nastaví jako rodič skutečného, objektu se kterým chceme pohybovat a odkaz na jeho *Transform* zaznamená. Při uchopení objektu se vztah mezi kotvou a objektem dočasně přeruší proto, aby se kotva nastavila na pozici štípnutí a natočila se dle

orientace ruky nebo rukou v případě manipulace oběma rukama. Po nastavení kotvy se opět vztah obnoví. Uchopený objekt se následně při každém novém snímku přesouvá posunem kotvy na aktuální pozici štípnutí a rotuje jejím natočením dle nové orientace ruky.

Způsob výpočtu nové pozice a orientace kotvy se liší podle toho, jestli je objekt uchopen pouze jednou rukou nebo oběma. V případě uchopení jednou rukou se použije výše popsáný způsob z dát ruky, která drží. Pokud je objekt uchopen oběma rukama použije se bod na přímce mezi body štípnutí rukou. Získáme jej součtem vektorů pozic, který vydělíme dvěma. Pro rotování objektů se v Unity 5 používají kvaterniony. Výslednou rotaci kotvy vypočteme interpolací mezi rotacemi rukou v bodě 0,5. Interpolace se používá na plynule natáčení objektu z počáteční rotace do cílové. Bod určuje orientaci přesně v polovině natáčení. Kotvu zarovnáme s rotací tak, že z ní získáme vektor jedné z lokálních os rotace. Kotvu nerotujeme tak, aby se její odpovídající lokální osa shodovala se získaným vektorem a další osa směřovala k jednomu z bodů štípnutí. Tento přístup rotace se používá při přímé manipulaci a má za následek, že se objekt chová, jako by jej uživatel držel rukama v bodech štípnutí. Nepřímá manipulace mírně modifikuje postup rotování. Při rotování je skutečný objekt přesunut na pozici kotvy a po dokončení rotace se vrátí na původní pozici. Tímto se objekt při nepřímé manipulaci chová, jako by se nacházel přesně na středu přímkou mezi body štípnutí rukou. Způsob manipulace s objektem určuje komponenta *ItemSelector* pomocí hodnoty výčtu, který reprezentuje používanou metodu.

#### 6.2.4 Implementace menu a interakce dotykem

Menu implementuje třída *LeapHandMenu*, která řeší jeho zobrazení a skrytí v závislosti na tom, zda dlaň ruky směřuje směrem k pohledu uživatele. Tuto informaci poskytuje komponenta *LeapActionDetector*. Vyhodnocení toho, zda dlaň směřuje k pohledu uživatele lze provést pomocí skalárního součinu vektorů. Vezmeme vektor normály dlaně a vektor směru kamery. Pokud jsou tyto vektory normalizované, tak výsledek skalárního součinu leží vždy v intervalu  $[-1, 1]$ . Hodnota skalárního součinu pak udává míru podobnosti směrů vektorů. Opačné vektory mají skalární součin roven -1, kolmé 0 a shodné 1. V našem případě nás zajímá situace, kdy vektory směřují proti sobě. Jinými slovy, kdy jsou opačné. Pohybujeme se tedy v záporné části intervalu. Menu se zobrazuje, pokud skalární součin vektorů leží v intervalu  $[-1, -0.75]$ .

Menu se skládá z tlačítek *LeapButton*, která reagují na dotyk ukazováku. Menu slouží pouze jako způsob umístění tlačítek. V editoru do něj lze umístit libovolný počet tlačítek. Logiku menu je pak třeba naprogramovat zvlášť. K detekci stlačení tlačítka pomocí ukazováku používám kolize v Unity 5. Při detekci kolize je nastaveno tlačítko do stavu stlačeno a při přerušení kolize se stav stlačení zruší. Tlačítko má další dva stavy, které lze využít k vytváření interakcí s menu: zapnutí a aktivace. Zapnutí se nastavuje podle typu tlačítka. Podporuji typ, kdy zmačknutím se mění stav zapnuto/vypnuto. Další typ ponechává po stlačení tlačítka vždy zapnuté a poslední jej zase vždy vypne. Aktivace tlačítka zakázání a povolení možnosti s ním interagovat. Každý stav mění barvu materiálu tlačítka, aby měl uživatel zpětnou vazbu v jakém stavu se nachází. Text tlačítka je možné modifikovat přes vlastnost *Text*.

## 6.3 Návrh a příprava experimentu

Cílem experimentu je srovnat vytvořené způsoby interakce s virtuálními objekty ve virtuální realitě. Porovnávání budu provádět z pohledu výkonnosti a použitelnosti. K otestování výkonnosti použiji měření. Pro analyzování použitelnosti metod interakce využiji vyhodnocovací metodu SUXES. Po dokončení experimentu bude uživatel dotázán na doplňující otázky a jeho pocity při experimentu, které poslouží k vyhodnocení rozhraní a získání zpětné vazby pro možné vylepšení. Pocity budou dotazovány pomocí tzv. Likertovy škály. Uživatel odpovídá na otázku číslem na škále 1-7, kdy je vždy uveden význam krajních hodnot. Z odpovědí lze pak pomocí střední hodnoty a rozptylu zjistit převažující názor uživatelů na dotazovanou otázku.

Metoda SUXES pracuje s očekáváními a zážitky uživatele. Nejdříve jsou uživatelům představeny jednotlivé způsoby interakce. Metoda poté spočívá v tom, že uživatel je před prováděním úkolů požádán, aby vyznačil svá očekávání o představených způsobech interakce do dotazníku. V dotazníku jsou připraveny otázky na vlastnosti rozhraní jako: rychlost, přesnost, složitost atd. U každé otázky je stupnice od 1 do 7 a pod čísly jsou prázdné čtverčky. Po stranách stupnice jsou napsány výrazy vysvětlující škálování stupnice. V případě rychlosti například 1 pro pomalé a 7 rychlé. Jeho úkolem je vyznačit hranice akceptovatelné a požadované úrovně. Vzniklý interval se nazývá zóna zájmu. Až uživatel dokončí všechny úkoly experimentu, je znovu požádán, aby křížkem označil, jakou měl při užívání zkušenost. Díky tomu můžeme rozhodnout, zda byly zkušenosti s rozhraním v zóně zájmu. Pokud uživatel označí zkušenost mimo zónu zájmu pod hranici přijatelnosti, předpokládáme, že měl negativní uživatelský zážitek a interakce nesplnila jeho očekávání. Naopak pokud označí nad požadovanou hranicí, znamená to, že interakce předčila jeho očekávání a měl vynikající uživatelský zážitek. Vše v zóně zájmu je považováno za uspokojivý zážitek.

### 6.3.1 Definice volných a vázaných proměnných

Vázané proměnné jsou ty, které při testování sledujeme. Cílem experimentu je srovnat výkonnost různých způsobů interakce. Výkonnost lze vyhodnocovat kvantitativně. Bude mě zajímat rychlost a přesnost při provádění úkolů.

Volné proměnné jsou takové, které můžeme v rámci úkolu měnit a zajímá nás, jak a čím se projeví. Volnou proměnnou je způsob interakce. Tedy přímá, nepřímá s výběrem pohledem a nepřímá s výběrem dlaní. Zajímat nás bude změna vázaných proměnných výkonnosti při plnění shodného úkolu jinou interakční metodou.

### 6.3.2 Testované úkoly

Experiment bude složen ze tří úkolů. Úlohy jsou navrženy tak, aby otestovaly různé způsoby interakce s virtuálními. První úkol si klade za cíl získat představu o intuitivnosti každého přístupu, druhý a třetí úkol se soustředí na otestování výkonnosti jednotlivých metod.

#### 1. úkol

Uživatel má za úkol přesunout kostku na vyznačenou pozici. Před jeho provedením je seznámen se způsoby interakce pouze omezeně. Tedy, že existují tři způsoby, jak lze s objekty pomocí rukou manipulovat. Jeden z nich umožňuje manipulovat jen s objekty v dosahu rukou a další dva i se vzdálenými. Nakonec mu jsou vysvětleny kroky, ze kterých se interakce obecně skládá. Tedy výběr, uchopení, přesun a uvolnění.



Způsob otestování intuitivnosti spočívá v tom, že uživatelé budou rozděleni na tři skupiny. Každá skupina bude mít nastaven jiný způsob interakce. Budeme měřit čas, za jak dlouho uživatel úkol splní. To jak dlouho trvalo uživatelům, než přišli na způsob ovládání poslouží jako ukazatel intuitivnosti. Kostka bude mít vždy shodnou počáteční a cílovou pozici.

## 2. úkol

Úkolem je přesunout tři kostky na jejich vyznačené pozice. Budeme měřit délku plnění úkolu a přesnost, s jakou uživatelé kostky na dané pozice umístili. Stejný úkol budou provádět pro každý způsob interakce ve dvou kolech. V prvním běhu jim bude řečeno, aby kostku po umístění na cílovou pozici již neupravovali a zdůrazníme, že je měřena doba provádění. V druhém běhu jim naopak řekneme, aby kostky umístili co nejpřesněji, a že jejich výslednou pozici mohou libovolně upravovat. Rozmístění kostek i jejich cílových pozic bude vždy stejné.

## 3. úkol

Cílem úkolu je otestovat způsoby výběru vzdálených objektů. Zajímá nás opět čas a přesnost. Před uživatelem bude ve vzdálenosti 1,5 metrů umístěno devět kostek. Úkol se bude skládat ze dvou částí. V první části se bude postupně zvýrazňovat jeden z objektů a cílem je zvýrazněný objekt vybrat. V druhé části bude zvýrazněna množina objektů, které musí uživatel vybrat. Stejný úkol uživatel splní pro oba způsoby výběru. Zvýrazňovány budou vždy stejné objekty, aby zůstaly zachovány shodné podmínky pro všechny uživatele a to i v rámci testování způsobu výběru. Měřit budu opět čas splnění úkolu a přesnost výběru.

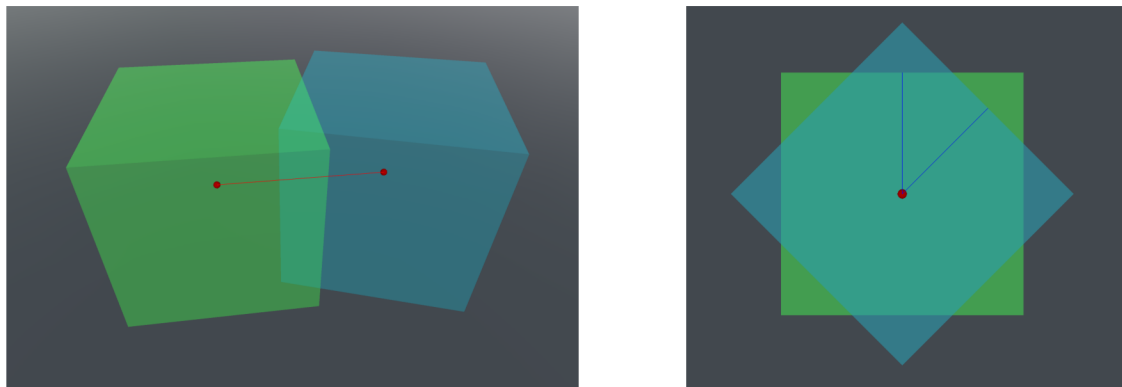
### 6.3.3 Způsob měření a tvorba aplikace pro experiment

Pro provedení experimentu jsem vytvořil aplikaci, která obsahuje výše popsané úkoly experimentu. Využívá navržené komponenty rozhraní. Experiment ovládá uživatel pomocí menu uvnitř virtuální reality a jeho provádění je poloautomatické.

V aplikaci jde přepínat mezi úkoly experimentu pomocí tlačítek menu. Po jejím spuštění je vždy aktivní první úkol. Ve scéně je zobrazen virtuální stůl, který slouží jako pracoviště a oblast pro zobrazování textu instrukcí k provádění úkolu. Text je zobrazován mírně nad úrovní stolu ve vzdálenosti 1 metru od uživatele a je natočen tak, aby směřoval proti pohledu uživatele. Započítí provádění úkolu se ovládá pomocí tlačítka **Start** a ukončení tlačítkem **Stop**. Informační text k úkolu je po započatí jeho provádění skryt. Po ukončení se automaticky nastaví další úkol v pořadí a instrukce k němu. Úkol lze opakovat přepnutím na něj pomocí tlačítka.

Během provádění úkolů se měří sledované vázané proměnné. Měření délky provádění úkolu řeším pomocí časových razítek při stisknutí tlačítek **Start** a **Stop**. Přesnost výsledné pozice kostky po jejím přesunutím měřím jako vzdálenost od středu a úhel vychýlení od cílové pozice. Výpočet vzdálenosti se provede z vektorů pozic středů objektů metodou *Distance*. Výpočet vychýlení je mírně náročnější. Problém si můžeme představit pomocí dvou kostek, které jsou umístěny přes sebe a nějaké jejich části se překrývají. Natočení kostek se liší pouze na ose  $y$ . Pokud jednou z nich začneme natáčet okolo os  $y$ , tak se mění část jejich překryvu následovně. Kostky jsou dokonale překryty, pokud je jejich úhel natočení v ose  $y$  stejný nebo jsou navzájem kolmé či rovnoběžné. Tedy po krocích  $90^\circ$ . V intervalu  $(0^\circ, 45^\circ)$  postupně míra překrytí klesá až do  $45^\circ$ , kdy je míra překrytí nejmenší. Dochází

tedy k maximální nepřesnosti mezi kostkami. Pohybem kostkou dále v intervalu  $(45^\circ, 90^\circ)$  míra překrytí postupně narůstá zpět do úplného překryvu kostek při úhlu  $90^\circ$ . Vychýlení vypočítáme jako rozdíl úhlů v ose  $y$ , který upravíme tak, aby spadl do  $90^\circ$ . Následně určíme výsledné vychýlení v intervalu  $(0^\circ, 45^\circ)$ . Pokud je větší než  $45^\circ$  vychýlení odpovídá doplňku do  $90^\circ$ . Přesnost výběru cílů budu udávat v procentech jako počet obrácenou hodnotu počtu všech zásahů vůči počtu správně zasažených cílů.



Obrázek 6.2: Výpočet nepřesnosti umístění kostky

Uživatelé jsou při provádění experimentu označeni unikátním ID, který slouží pro jeho označení při ukládání hodnot do souborů. ID se mění při dokončení experimentu. Výsledky měření každého úkolu se ukládají do vlastního souboru označeného ID a názvem úkolu. Před ukončením experimentu může uživatel opakovat libovolný úkol a data v odpovídajících souborech se aktualizují.

### 6.3.4 Návrh dotazníku

Dotazník se skládá ze tří částí: obecná, očekávání a zážitky SUXES, pocity.

V obecné části se dotazují na věk uživatel vyznačením do jaké věkové skupiny patří, jeho odbornost v oblasti IT výběrem jedné z možností a na jeho zkušenosti s virtuální realitou a zařízeními pro snímání akcí uživatele. Otázky na zkušenost jsou koncipovány výběrem možnosti zda někdy používal podobné zařízení a závislosti na odpovědi dodatečnou otázkou. Pokud ano, tak v jaké souvislosti. Jestli ne, tak zda o ví, že taková zařízení existují.

Druhá část se dotazuje na očekávání účastníka před tím, než začne provádět zadané úkoly a po jejich dokončení slouží k vyznačení zážitků. Jsou zde tři oblasti, každá pro jeden způsob interakce. Každá oblast se skládá ze stupnice 1 až 7 a následně tvrzení o vlastnostech interakce se čtvercovými poli pro každou číslici stupnice. Krajní hodnoty stupnice jsou pojmenovány. Dotazují se na tyto vlastnosti: rychlost, přesnost, chybovost, únava, přirozenost a složitost. Uživatel má vyznačit interval zájmu obdélníkem, kde krajní hranice značí akceptovatelnou a požadovanou úroveň.

Poslední část obsahuje otázku na pocity a preference uživatele. Pro vyznačení odpovědi na otázku o pocitech je použita Likertova škála. Dotazují se i na to, který způsob interakce preferuje a proč, pocity únavy rukou a očí, míru soustředění a jak se účastníkovi líbila interakce s menu. Dotazník je ukončen vybídnutím k využití volného místa ke sdělení připomínek a názorů účastníka. Co se mu líbilo nebo nelíbilo? Co by změnil? Co mu chybělo?

## 6.4 Experiment

Experiment probíhal na půdě fakulty informačních technologií v koridoru L211. Stanoviště pro testování zde bylo připraveno od 9:00 do 15:00, kdy účastníci přicházeli průběžně na předem smluvený čas. Interval vyhrazený na provedení experimentu byl stanoven na 10 minut.

Před ostrým experimentem jsem provedl pilotní test, abych vyzkoušel zda je implementovaná aplikace připravená k použití a průběh experimentu je bezproblémový. Během pilotního testu jsem neodhalil žádné problémy.

### 6.4.1 Účastníci

Experimentu se účastnilo 30 osob (26 mužů a 4 ženy) ve věku 18-26 let. Byli vybráni pomocí inzerce o možnosti účastnit se experimentu s virtuální realitou přes studentské skupiny všech ročníků studentů fakulty informačních technologií VUT na sociální síti Facebook.

Většina účastníků proto byla tvořena studenty fakulty, jejichž znalosti v oblasti informačních technologií jsou v rámci populace nadprůměrné. Dva účastníci byli studenty jiné fakulty a zařadili se do skupiny běžných uživatelů. Všichni účastníci měli povědomí o zařízeních pro snímání pohybu rukou. Z toho 4 účastníci někdy používali zařízení Kinect a 1 zařízení Leap Motion pro ovládání hry rukou. Až na jednoho z účastníků věděli o brýlích pro virtuální realitu všichni a 6 z nich nějaké brýle již dříve vyzkoušelo. Žádný z účastníků neměl zkušenost s oběma zařízeními zároveň. Čtyři účastníci měli dioptrické brýle a pro testování si je museli sundat.

### 6.4.2 Testovací prostředí

Stanoviště pro testování se skládalo ze stolu, židle, stolního počítače a zařízení Oculus Rift DK2 a LeapMotion. Byl použit výkonný počítač, který splňuje minimální hardwarové požadavky pro virtuální brýle. Senzor pro snímání pohybu hlavy byl umístěn na horní hraně monitoru v úrovni očí. Senzor LeapMotion byl upevněn na přední část brýlí, tak aby snímal oblast před uživatelem ve směru jeho pohledu. Tedy v režimu HMD.



Obrázek 6.3: Ukázka pracoviště při provádění experimentu

Na počítači byl nainstalován operační systém Windows 7 a všechny potřebné ovladače pro ostatní zařízení. K testování a měření byla připravena vlastní aplikace, která využívá komponenty navrženého rozhraní a vede účastníky při provádění úkolů experimentu.

### 6.4.3 Průběh experimentu

Účastník je po uvítání krátce seznámen s procesem průběhu experimentu, který testuje tři způsoby interakce s virtuálními objekty a je požádán, aby vyplnil první část dotazníku. Až tak učiní je požádán, aby se přesunul na testovací stanoviště, které dozor mu představí. Vysvětlí mu, jak se má usadit a jaká zařízení ho budou snímat. Poté mu je představena aplikace pro provádění experimentu a problémem interakce s virtuálním objektem ve virtuální realitě. Dozor mu vysvětlí způsob ovládání menu a logiku interakce s objektem složenou z výběru, uchopení, přesunu a uvolnění objektu. Následně je požádán aby provedl první úkol, ve kterém má za úkol přesunout kostku na vyznačenou pozici a je upozorněn, že je nastaven jeden ze způsobů interakce. Cílem je zjistit, jak dlouho mu bude trvat, než odhalí způsob ovládání a provede úkol. Čas se začíná měřit po stisku tlačítka **Start** do stisku **Stop** a slouží jako ukazatel intuitivnosti způsobu interakce.

Po dokončení prvního úkolu jsou účastníkovi podrobně vysvětleny všechny způsoby interakce. Uchopení a uvolnění pomocí gesta štípnutí, výběr a přímá manipulace rukama, způsob nepřímé manipulace a výběr pohledem nebo paprskem z ruky. Následně je seznámen s druhou částí dotazníků o očekáváních a zážitcích SUXES a způsobem jeho vyplnění. Poté je požádán o vyplnění očekávání a až tak učiní je seznámen s druhým úkolem, kde je cílem přesunout tři kostky na vyznačená místa. Nejdříve, aniž by upravoval výslednou pozici a je upozorněn, že je měřen čas provádění. Tento úkol provede pro každý způsob interakce. V dalším kole je mu povoleno libovolně upravovat výslednou pozici a je upozorněn že je měřena přesnost umístění. Opět provede přesun pro všechny způsoby interakce. Během přesunu je měřen čas po stisknutí tlačítka **Start** do stisku **Stop**, kdy je vypočtena nepřesnost umístění každé z kostek. Cílem je změřit rychlost a přesnost každého způsobu interakce v prvním případě bez úpravy výsledné pozice a v druhém s libovolnou úpravou.

Posledním úkolem je otestování způsobu výběru vzdálených objektů, který probíhá ve dvou kolech. Před uživatelem je zobrazeno devět objektů ve vzdálenosti 1,5 metru a jeho úkolem je vybrat pouze zvýrazněný objekt. V prvním kole postupně vybírá šest objektů, kdy pouze jeden z devíti objektů je zvýrazněn a po jeho výběru se zvýrazní další. V druhém kole je současně zvýrazněno pět objektů a cílem je vybrat všechny zvýrazněné objekty. Uživatel provede každé kolo oběma způsoby výběru. Tedy pohledem a paprskem z ruky. Během provádění měříme čas v úsek od stisku tlačítka **Start** do stisku **Stop**. Navíc měříme i nepřesnost výběru, tedy počet správných a špatných výběrů objektů.

Když účastník dokončí všechny úkoly, je požádán o vyplnění zážitků do druhé části dotazníku a všech ostatních otázek v části třetí.

## 6.5 Výsledky experimentu a vyhodnocení rozhraní

Během experimentu jsem testoval rozhraní komponent z pohledu výkonnosti a použitelnosti. Všechny naměřené hodnoty jsou dostupné na CD, které je součástí této práce. V následující kapitole bude uvedena pouze část naměřených dat a především závěry, které jsem z nich vyvodil.

Nejdříve se podíváme na výsledky týkající se použitelnosti, které byly získány pomocí dotazníku a sledováním uživatelů při provádění úloh experimentu. Vyplněné dotazníky jsou v naskenované podobě také součástí CD. Stěžejní částí je vyhodnocení metodou SUXES, která odhalí, jak si vedly testované způsoby interakce. Pro každý způsob interakce je prezentována grafická reprezentace výsledků SUXES znázorňující intervaly zájmu ke všem dotazovaným vlastnostem a výsledné zážitky uživatelů. Uvedené hodnoty jsou mediánem

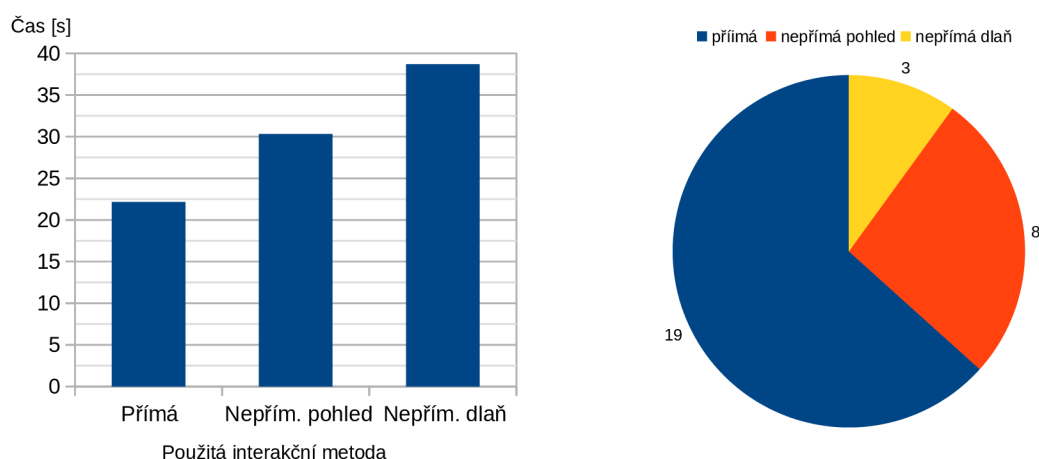
odpovědí od všech uživatelů. Následně jsou metody interakce celkově vyhodnoceny pomocí mediánu všech vlastností a je tak získán obecný pohled na uživatelský zážitek s interakční metodou. Zjištěný výsledek je srovnán s preferencemi uživatelů. Jakou metodu uživatele preferovali a proč? Jako ukazatel použitelnosti slouží i intuitivnost, která byla testována v rámci prvního úkolu. Na závěr jsou poskytnuty informace o pocitech uživatelů. Jaká byla únava jejich rukou a očí? Jak moc se museli při používání metod soustředit? Jaké byly nejčastější momenty frustrace? Jaký měli názor na ovládání pomocí menu? Vyhodnocení těchto otázek je provedeno pomocí průměrné hodnoty odpovědí na Likertově škále. Průměrná hodnota nám říká výsledné umístění na škále. Míra shody ostatních odpovědí je znázorněna rozptylem. Pro doplnění je uveden také graf zastoupení odpovědí.

V druhé části kapitoly čtenář nalezne srovnání výkonnosti jednotlivých metod interakce pro představené úkoly. Srovnání jsem rozšířil o výsledky měření výkonnosti zkušeného uživatele, které lépe vypovídají o možnostech interakčních metod. Provedl jsem měření, při kterém jsem plnil úkoly experimentu celkem v dvaceti opakováních. Srovnání se zaměřuje na rychlost plnění úkolů a přesnost umístění objektů po jejich přesunu. Pro zajímavost jsem do srovnání uvedl i rychlost provádění shodného úkolu v realitě. Čtenář se dozví odpověď na otázku: Který přístup je nejlepší a proč? Srovnání je založeno na průměrných hodnotách získaných měřeními.

Kapitola je ukončena zhodnocením vytvořeného rozhraní komponent. Hodnotím zde představené interakční metody, jejich vlastnosti a vhodnost pro určité typy úloh. Zamýšlím se nad možnostmi, které přináší pro tvorbu uživatelských rozhraní a jejich ovládání. Na závěr shrnuji názory uživatelů a zkušenosti, které jsem získal jejich pozorováním při provádění experimentu. Identifikuji problémová místa a možná vylepšení rozhraní.

### 6.5.1 Použitelnost interakčních metod

V rámci prvního a druhého úkolu uživatelé testovali tři způsoby interakce: přímá manipulace, nepřímá manipulace s výběrem pohledem a nepřímá manipulace s výběrem paprskem z dlaně. Při prvním úkolu byly uživatelé rozděleni na třetiny a každá skupina pracovala s jinou metodou a měli za úkol přesunout kostku. Cílem bylo zjistit intuitivnost interakce. Jako ukazatel byl použit čas dokončení úlohy.



Obrázek 6.4: Doba splnění úlohy pro zjištění intuitivnosti a preference uživatelů

Nejintuitivnější se ukázala přímá manipulace (obrázek 6.4), která dosáhla průměrné



doby splnění 22,15 sekund. Což odpovídá také tomu, že většina uživatelů se nejprve kostku pokusila uchopit nebo vzít do dlaní. Na druhé pozici se umístil způsob s výběrem pomocí pohledu s časem 30,33 sekund následovaný výběrem paprskem z dlaně 38,7 sekund. Uživatelé si většinou všimli ukazatele ve výhledu a nebo objekt vybrali náhodou při pohledu tím, že se na něj podívali. Největší problém měli uživatelé s výběrem objektu pomocí paprsku z dlaně, který je na rozdíl paprsku pro výběr pohled nutno spustit gestem. Většina z nich nevěnovala pozornost zářícím bodům na dlaních nebo jim trvalo dlouho, než si jich všimli, případně na ně byli upozorněni. Poté se jim již záhy podařilo vyslat paprsek při propnutí dlaně. Nejčastěji, když rukou zkoušeli zamávat. Tyto výsledky potvrdily moje očekávání.

Gesto uchopení štípnutím většinou nedělalo uživatelům problém jej odhalit. Gesto je detekováno na základě blízkosti prstů a má poměrně velkou toleranci, proto umožňuje uchopení i v některých pozicích sevření ruky v pěst a štípnutím všemi prsty, což část uživatelů využila. Tyto způsoby uchopení však přinášely neočekávané potíže při přesouvání objektu, způsobené špatným snímáním rukou v těchto polohách. Přesun při přímé manipulaci byl bezproblémový, protože kopíruje realitu. Naopak přesouvání při nepřímé manipulaci se překvapivě ukázalo jako problémové, i když využívá přirozených pohybů rukama a napodobuje manipulaci s objektem na tyči. Uživatelé po uchopení často nevěděli, co mají s objektem dělat, aby jej posunuli blíže. Nakonec velká část z nich objekt postupným posouváním jednou rukou přesunula. Překvapivě téměř nikdo nepoužil obě ruce. Zajímavostí byl jeden uživatel, který kostku přesunul pouze pomocí kolizí s fyzickým modelem rukou.

## Vyhodnocení metodou SUXES

K otestování způsobů interakce pro vyhodnocení pomocí metody SUXES sloužily uživatelům úkoly 2. a 3. Uživatelé hodnotili následující vlastnosti: rychlost, přesnost, chybovost, únavnost, přirozenost a složitost. V tuto chvíli již byli plně seznámeni se všemi způsoby interakce, avšak neměli možnost si je vyzkoušet, aby nebyli ovlivněni při vyplňování požadavků. Uživatelé na metody kladli vysoké požadavky, jak můžeme vidět pomocí šedě znázorněných oblastí zájmu na obrázku 6.5, které se pohybují v horní polovině. Na grafickém znázornění můžeme také sledovat, že oblasti zájmu jsou pro všechny způsoby podobné. U přímé manipulace byly kladeny oproti ostatním vyšší nároky na přesnost a chybovost. Naopak uživatelé akceptují menší rychlost a mají nižší požadavky na únavnost. Nároky na rychlost metod nepřímé manipulace jsou shodné a při výběru paprskem z dlaně akceptují nižší přesnost a požadují nižší složitost, ale mají vyšší nároky na to, aby byl přístup méně unavující.

|             | Přímá manipulace |   |   |   |   |   |   | Nepřímá s výběrem pohledem |   |   |   |   |   |   | Nepřímá s výběrem dlaní |   |   |   |   |   |   |           |
|-------------|------------------|---|---|---|---|---|---|----------------------------|---|---|---|---|---|---|-------------------------|---|---|---|---|---|---|-----------|
|             | 1                | 2 | 3 | 4 | 5 | 6 | 7 | 1                          | 2 | 3 | 4 | 5 | 6 | 7 | 1                       | 2 | 3 | 4 | 5 | 6 | 7 |           |
| pomalé      |                  |   |   |   | x |   |   |                            |   |   |   |   | x |   |                         |   |   |   | x |   |   | rychlé    |
| nepřesné    |                  |   |   |   | x |   |   |                            |   |   |   |   | x |   |                         |   |   |   | x |   |   | přesné    |
| chybové     |                  |   |   |   | x |   |   |                            |   |   |   | x |   |   |                         |   |   |   | x |   |   | bez chyb  |
| unavující   |                  |   |   |   | x |   |   |                            |   |   |   | x |   |   |                         |   |   |   | x |   |   | bez únavy |
| nepřirozené |                  |   |   |   |   | x |   |                            |   |   |   |   | x |   |                         |   |   |   | x |   |   | přirozené |
| složitě     |                  |   |   |   |   | x |   |                            |   |   |   |   | x |   |                         |   |   |   | x |   |   | snadné    |
| špatné      |                  |   |   |   | x |   |   |                            |   |   |   |   | x |   |                         |   |   |   | x |   |   | dobré     |

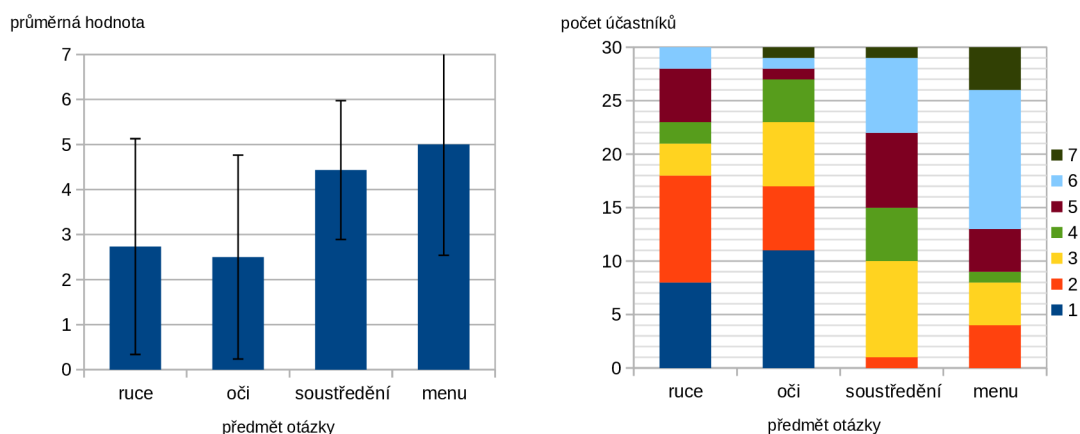
Obrázek 6.5: Grafická reprezentace výsledků odpovědí uživatelů metody SUXES

Nejlépe byl hodnocen přístup nepřímé manipulace s výběrem pohledem, jak můžeme vidět na souhrnném vyhodnocení ve spodní části obrázku 6.5. Ostatní způsoby předčil ve

všech aspektech. Přímá manipulace byla zase až na únavnost ve všech vlastnostech lepší, než nepřímá s výběrem dlaní. Všechny výsledky spadají do intervalu zájmu a můžeme tedy tvrdit, že uživatelé měli uspokojivý zážitek. I když nejlépe hodnoceným způsobem byla nepřímá interakce s výběrem pomocí pohledu, tak uživatelé preferovali přímou interakci viz obrázek 6.4. Důvodem byla přirozenost a jednoduchost někdy i za cenu přesnosti. Naopak uživatelé, kteří preferovali nepřímou s výběrem pohledem zmiňovali menší námahu a nutnost natahovat se k objektům. Také jim přišla přesnější.

## Uživatelská zkušenost

V poslední části dotazníku jsem se uživatelů ptal na otázky ohledně jejich pocitů. Výsledky jsou znázorněny v grafech na obrázku 6.6. Můžeme vidět, pocit únavy očí a rukou hodnotili spíše jako mírný. Většina do hranice 3. Z rozptylu i rozložení však vidíme, že jejich názor nebyl jednoznačný. S únavou očí měli problém především lidé, kteří nosili brýle do dálky. Z organizačních důvodů nebylo možné upravovat nastavení brýlí a jejich kalibraci. Což nejvíce ovlivnilo právě uživatele s poruchou zraku a ty s nejvíce odlišnou vzdáleností očí. Špatná kalibrace měla nežádoucí vliv na kvalitu zobrazení, které se projevuje špatným vnímáním vzdálenosti v prostoru nebo rozdvojeným viděním. Uživatelé pak měli potíže s plněním úkolů vyžadující přesnost a také se jim špatně ovládalo menu. Hodnocení interakce s menu je znázorněno na obrázku 6.6 a bylo převážně kladné. Míru soustředění při používání rozhraní vnímali uživatelé různě, ale spíše se přikláněli k vyšší náročnosti na soustředění. Ptal jsem se také, zda uživatelé pociťovali frustraci a pokud ano, tak kdy. Celkem 11 z nich uvedlo, že ano a nejčastějším momentem byla nepřesnost při uvolnění objektu, která má za následek jeho vychýlení od požadované pozice a přetrvání výběru objektu i po jeho uvolnění, což způsobovalo nechtěné uchopení objektu.



Obrázek 6.6: Průměrné hodnoty pocitů uživatelů a rozložení odpovědí na Likerově škále

### 6.5.2 Výkonnost interakčních metod

Pro zjištění výkonnosti navrženého rozhraní a jednotlivých způsobů interakce jsem se rozhodl analyzovat čas potřebný pro dokončení úkolu a přesnost výsledného umístění kostek v případě druhého úkolu nebo přesnost výběru v třetím. Tyto proměnné jsem měřil v průběhu provádění celého experimentu a k srovnání použiji výsledky z úkolu 2. a 3. Pro

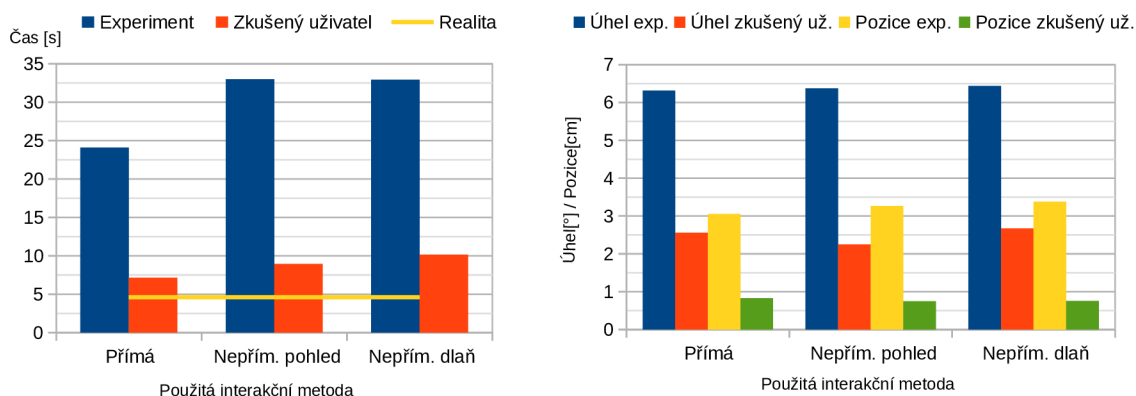
účely srovnání jsem navíc naměřil výkonnost práce zkušeného uživatele, jak sem již zmínil v úvodní části. Nyní se pojdme podívat na souhrnné výsledky pro jednotlivé úkoly.

## Úkol 2. přesun kostek

Tento úkol se opakoval ve dvou běžích. V prvním jsem se soustředil na rychlost provádění interakce. Pro srovnání metod s realitou jsem uvedl hodnoty naměřené při provádění shodného úkolu v realitě, kdy jsem provedl měření v deseti opakováních. Výsledky jsou vyneseny v grafech na obrázku 6.7 spolu s odpovídající dosaženou přesností.

Na grafu délky provádění úkolu lze pozorovat, že účastníkům experimentu trvalo přesunutí kostek v průměru o 20 sekund déle, než v případě zkušeného uživatele. Bylo to způsobené tím, že uživatelé používali rozhraní poprvé a navíc se jich většina se zařízeními pro virtuální realitu a snímání pohybu rukou dříve nesetkala. Největším problémem měli se snímáním rukou, kdy zapomínali, že ruce musí mít v oblasti snímání senzoru a při nepřímé manipulaci si musí dávat pozor na pózu ruky. Často se pak stávalo, že nechtěně uchopovali objekt a v případě, že byla ruka navíc na hranici snímání docházelo k jeho přesunu na nečekané pozice.

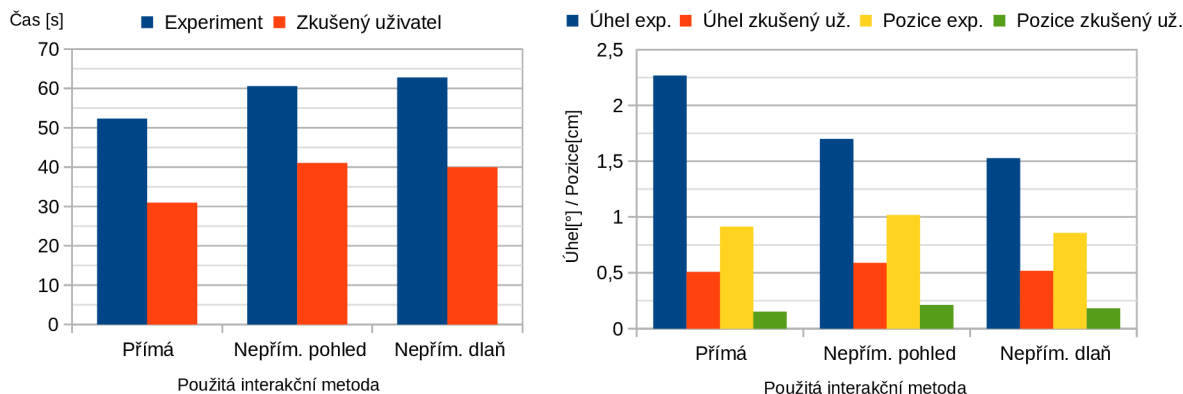
Dále můžeme vidět, že nejrychlejším způsobem byla přímá manipulace, následovaná nepřímou manipulací s výběrem pohledem a poslední se umístila nepřímá s výběrem paprskem z dlaně. Tento trend platil pro všechny úlohy. V případě zkušeného uživatele se dosažená rychlost provádění blíží realitě. V případě přímé manipulace se liší zhruba o 2 sekundy.



Obrázek 6.7: Srovnání délky provádění a přesnosti prvního běhu 2.úkolu

Dosažená přesnost všemi metodami byla shodná a liší se pouze mezi účastníky experimentu a zkušeným uživatelem, který dosahoval lepších výsledků. Výsledný rozdíl úhlu a pozice se liší v průměru o  $4^\circ$  a 2 cm. Důvod je shodný jako v při délce provádění.

V druhém běhu úkolu jsem se zaměřil na přesnost dosaženou jednotlivými metodami interakce. Naměřené výsledky jsou vyneseny v grafech na obrázku 6.8. Graf přesnosti nám opět ukazuje rozdíl v přesnosti účastníky experimentu vůči zkušenému uživateli. Rozdíl již není tak velký a pohybuje se okolo  $1^\circ$  až  $2^\circ$  v úhlu a 7 mm v pozici. Přesnost je opět téměř stejná pro všechny metody. Mírné rozdíly v přesnosti účastníky experimentu s největší pravděpodobností ukazuje zlepšení uživatelů učením. Nepřímá manipulace je použita ve dvou metodách, a proto se v ní uživatelé více zlepšují. U zkušeného uživatele tento jev nevidíme. Zlepšení vůči prvnímu běhu, kdy uživatelé neupravovali výslednou pozici je vidět na obrázcích 6.7 a 6.8. Zlepšila se průměrně o  $4^\circ$  a 2 cm u účastníků experimentu a  $2^\circ$  a 0,5 mm u zkušeného uživatele.

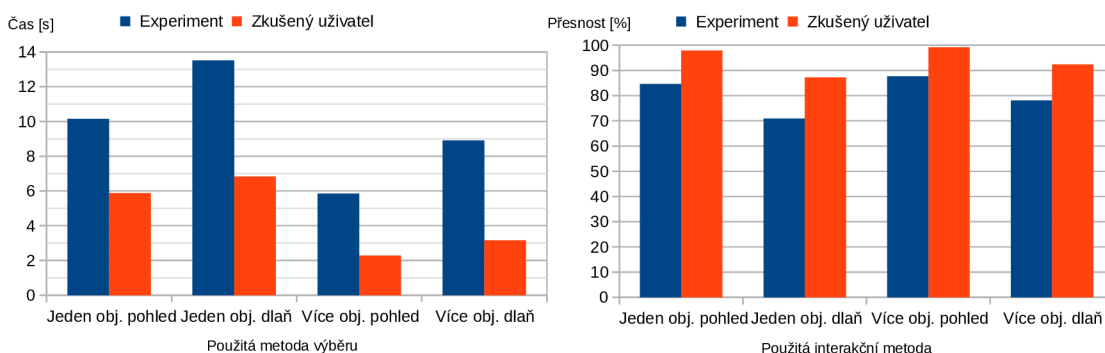


Obrázek 6.8: Srovnání délky provádění a přesnosti druhého běhu 2. úkolu

Pokud se zaměříme na čas provádění na obrázku 6.8, tak se opět liší pouze mezi účastníky zkušeným uživatelem a to zhruba o 20 sekund, jinak se trend zachovává a přímá metoda je nejrychlejší. Doba provádění se oproti prvnímu kolu (viz obrázek 6.7) v případě účastníků na úkor přesnosti zvedla přibližně dvojnásobně a u zkušeného uživatele až čtyřnásobně. V případě zkušeného uživatele je potřebná doba využita pouze na úpravu výsledné pozice na rozdíl od účastníků, kteří velkou část trávili přesouváním kostky a řešením chyb.

### Úkol 3. výběr objektů

Úkol se skládal ze dvou částí. V první se vybíral pouze jeden objekt a v druhé celá množina objektů. Z grafů na obrázku 6.9 můžeme vyčíst, že výběr pohledem byl rychlejší a přesnější, než výběr paprskem z dlaně. Rozdíl je zhruba 3 sekundy u účastníků experimentu a 1 sekundu u zkušeného uživatele. Rozdíly mezi částmi úkolu jsou způsobeny povahou úkolu. Přesnost byla nižší pro první část úkolu, což bylo způsobené jeho vyšší náročností. Rozdíl mezi způsoby je zhruba 10% pro všechny případy.



Obrázek 6.9: Srovnání délky provádění a přesnosti výběru objektů 3. úkolu

Výsledky měření na obrázcích 6.7 a 6.8 ukázaly, že nejlepší metodou z pohledu výkonnosti je přímá manipulace. Dosahuje nejvyšší rychlosti a po naučení ovládání stejné přesnosti, jako ostatní metody. Pro nové uživatele je však méně přesná než metody nepřímé manipulace. Ty jsou z pohledu výkonnosti velmi blízké, ale na obrázku 6.9 můžeme vidět, že výběr pohledem je rychlejší a více přesný, než výběr paprskem z dlaně.

### 6.5.3 Zhodnocení

V této práci bylo navrženo a realizováno rozhraní komponent, které umožňují provádět základní akce ovládání počítače ve virtuální realitě. Zaměřil jsem se na interakci uživatele se svým okolím ve 3D. Představil jsem tři metody interakce, které umožňují uživateli dotýkat se, vybírat, uchopovat a přesouvat virtuální objekty pomocí rukou. Všechny metody využívají přirozené interakce z reálného světa a jsou díky tomu intuitivní a snadné k naučení, i když pro interakci se vzdálenými objekty a menu jsem použil magické přístupy. Při využití magických přístupů stavím na zkušenostech získaných výzkumníky v této oblasti. Používám výběr objektu pomocí paprsku a představuji vlastní způsoby manipulace s vybranými objekty.

Toto rozhraní bylo otestováno na 30 uživateli a ukázalo se, že je použitelné pro plnění definovaných problémů a splňuje stanovené cíle. Výsledky experimentu ukázaly, že nejpoužitelnější metodou je nepřímá manipulace s objekty s výběrem pomocí pohledu, ale uživatelé nejvíce preferují přímou manipulaci, která je nejvíce intuitivní a přirozená. Také je z pohledu výkonnosti nejrychlejší a dosahuje srovnatelné přesnosti jako metody ostatní. Přímá manipulace je proto nejvhodnější způsob pro interakci a ovládání rozhraní. Je dobrá pro ovládání prvků v dosahu uživatele pomocí rukou a kombinací s interakcí pomocí dotyku umožňuje vytvářet libovolné složitější interakce pro plnění konkrétních úkolů v závislosti na aplikaci, které má rozhraní řešit. Nedostatkem je omezení na práci pouze s prvky v dosahu uživatele. Tento problém odstraňuje nepřímá manipulace a možnost výběru vzdálených objektů. Nejlepším přístupem je výběr pomocí pohledu. Nabízí se proto použít jejich kombinaci a získat tak výhody obou metod současně. Výsledné rozhraní je potom možné ovládat rukama i pohledem uživatele a umožňuje interakci s libovolným virtuálním objektem v dohledu uživatele.

Při experimentování jsem se dotazoval na cennou zpětnou vazbu od uživatelů a pozoroval jsem jejich chování při používání rozhraní. Odhalil jsem následující nedostatky rozhraní a náměty pro vylepšení. Velkým nedostatkem se ukázala nedostatečná vizualizace při nepřímé manipulaci, která způsobovala uživatelům potíže orientovat se při přesunu objektu. Dále uživatelé zmiňovali požadavek na odznačení vybraného objektu po jeho uvolnění. Důvodem bylo jeho opětované nechtěné uchopení po umístění na požadovanou pozici. Požadavku se dá vyhovět, ale je otázkou jaký dopad bude mít změna na metody nepřímé interakce. Pokud by chtěl uživatel po umístění objektu s ním znovu pracovat, tak by jej musel nejdříve vybrat. Avšak objekt se většinou předává mezi rukama a uvolňuje se až, když s ním uživatel nechce nadále pracovat.

Během experimentu jsem odhalil nedostatek interakce s mačkáním tlačítek menu. Docházelo k nechtěnému zdvojenému stisknutí tlačítka na místo stisku jednoho. To bylo způsobeno tím, že detekuji pouze kolizi kosti na špičce prstu. Uživatelé, kteří tlačítko zmáčkli příliš hluboko až za hranici koncové kosti prstu, způsobili detekování kolize dvakrát. Celá kost prošla tlačítkem a zmáčkla jej jednou směrem zmáčknutí a podruhé při návratu zpět. Uživatelé si také stěžovali na velkou toleranci detekce gesta štípnutí, která způsobovala pohyb s objektem i při jeho upouštění a měla za následek nepřesnost ve výsledné pozici. Řešení tohoto problému není triviální, protože snížením tolerance dochází k zvýšení chybovosti způsobující nechtěné upuštění objektu. To je způsobeno nepřesností snímání rukou.

Obecně jsem si při sledování uživatelů všiml, že byli z počátku při interakci velmi bojácní a nejistí toho co si mohou dovolit. Měli problémy se snímáním rukou, protože na něj nebyli zvyklí a dříve se s ním nesešli. Velké problémy se snímáním byly pro mě překvapující, protože jsem se v průběhu vývoje se snímáním natolik sžil, že jej téměř nevnímám. Auto-



matically mám ruce v prostoru snímání senzoru a pohyb rukama uzpůsobuji tak, aby byl co nejlépe snímatelný senzorem. Například mám ruce vždy rozevřené v neutrální poloze, gesto štípnutí provádím zřetelně a tak, aby zbytek prstu ruky byl snímán, což zlepšuje přesnost snímání změny orientace. Uživatelé také často prováděli gesto štípnutí za dlaní ruky, kterou stínili senzoru. Všechny tyto problémy souvisí s nezkušeností uživatelů, kteří často senzor používali poprvé v životě.

## 6.6 Aktualizace rozhraní na základě výsledků experimentu

V kapitole zhodnocení jsem uvedl možná vylepšení a problémy, které jsem odhalil experimentováním. Na základě těchto poznatků jsem aktualizoval rozhraní.

Vylepšil jsem vizualizaci nepřímé manipulace. Nyní je indikováno uchopení na pomyslné přímce jejím zobrazením do scény. Přímka je v případě uchopení jednou rukou vedena z bodu kotvy, tedy pozice štípnutí, do bodu středu uchopeného objektu. Při uchopení oběma rukama je manipulace doprovázena indikací uchopení přímkou mezi body štípnutí každé ruky a další přímkou spojující její střed a uchopený objekt. Přímka mezi body slouží k vizualizaci změny úhlu objektu a přímka směřující k objektu jeho ukotvení.

Upravil jsem také způsob interakce se vzdálenými objekty. Nyní je výběr objektu po upuštění zrušen. Opravil jsem i chybu interakce s tlačítkem dotykem tak, že se detekují kolize se všemi kostmi prstu. V poslední řadě jsem se pokoušel odstranit nepřesnost vznikající při uvolnění objektů tolerancí gesta štípnutí. Vyzkoušel jsem posun objektu podle bodů stabilnějšího bodu středu dlaně namísto bodu štípnutí. Ve výsledku však tato změna neměla na přesnost velký přínos. Manipulace s objektem byla kvůli provedené změně méně přirozená. Ponechal jsem tedy tento problém jako námět pro další práci.

## 6.7 Demonstrační aplikace

Pro demonstraci jsem vytvořil aplikaci hřiště, která obsahuje všechny způsoby interakce. Tedy přímou interakci s blízkými objekty a nepřímou interakci se vzdálenými objekty s možnostmi výběru jednoho nebo více objektů pohledem či paprskem z ruky. Uživatel může pomocí menu vytvářet nové objekty a libovolně přepínat mezi způsoby interakce. Navíc může měnit měřítko objektů. Dále objekty prostředí reagují na kolize s rukama dle fyzikálních zákonů z reálného světa.

Jako demonstraci může uživatel využít aplikaci experimentu a vyzkoušet si splnit všechny představené objekty. Pro ovládání experimentu se používá menu. V průběhu experimentu jsou uživatelům zobrazovány instrukce popisující zadání úkolů.

# Kapitola 7

## Závěr

Cílem této práce bylo navrhnout a vytvořit uživatelské rozhraní, které bude umožňovat uživateli pracovat s počítačem ve virtuální realitě pro zvolená ovládací zařízení.

Začal jsem studiem zařízení pro virtuální realitu a snímání akcí uživatele. Díky tomu jsem si udělal přehled o možnostech, které zařízení přináší pro ovládání. Následně jsem se věnoval studiu 3D uživatelských rozhraní. Zaměřil jsem se na doporučení pro návrh a možnosti jejich ovládání. Získané informace mě vedly k podrobnějšímu studiu způsobů interakce ve virtuální realitě. Rozhodl jsem se pro ovládání využít ruce uživatele a snímání pohybu hlavy. Navrhl jsem tři způsoby interakce, které řeší problém výběru a manipulace s virtuálními objekty a interakci založenou na dotyku.

Pro realizaci návrhu jsem zvolil zařízení LeapMotion a Oculus Rift DK2. Implementaci jsem provedl s pomocí herního enginu Unity 5. Vytvořené rozhraní umožňuje uživateli manipulovat s objekty v dosahu jeho rukou metodou interakce, kterou jsem nazval přímá manipulace. Pro manipulaci s objekty mimo jeho dosah může využít jednu z metod nepřímé manipulace, které umožňují vybírat vzdálené objekt pohledem nebo paprskem z dlaně. Navíc může s objekty manipulovat pomocí dotyku, jehož využití jsem prezentoval interakcí s tlačítky menu. S využitím představených metod interakce lze vytvářet libovolná rozhraní na vyšší úrovni abstrakce v závislosti na úkolu, který má plnit.

Vyhodnocení rozhraní jsem provedl pomocí experimentu, kterého se zúčastnilo 30 uživatelů. Zaměřil jsem se na jeho použitelnost a výkonnost vytvořených způsobů interakce s virtuálními objekty, které vzájemně srovnávám. Jako nejlepší způsob se ukázala být přímá manipulace. Byla nejrychlejší ze všech metod a dosahovala stejné přesnosti, jako metody ostatní. Uživatelé ji také volili, jako preferovaný způsob interakce. Na druhé pozici se umístila nepřímá manipulace s výběrem pomocí pohledu. Tato metoda byla uživateli vyhodnocena, jako nejlépe použitelná.

Existuje několik oblastí, kterými by se mohla práce dále ubírat. V první řadě je prostor pro vylepšení přesnosti představených způsobů interakce a jejich opětovné vyhodnocení. Možné vylepšení se nabízí i ve zlepšení způsobů vizualizace. Další možnou oblastí je rozšíření rozhraní o množinu ovládacích prvků, které by poskytovalo. Vývojáři, kteří používají rozhraní ve své aplikaci, by mohli tyto ovládací prvky ihned použít a usnadnit si tak tvorbu rozhraní. Posledním směrem, kterým by se mohla práce ubírat, je její využití pro konkrétní způsob ovládání nějaké aplikace ve virtuální realitě a specializování rozhraní pro ni. Příkladem může být využití pro ovládání robotického ramene k přesouvání reálných objektů pomocí interakce s jejich modelem ve virtuální realitě.

# Literatura

- [1] Bordegoni, M.; Hemmje, M.: A Dynamic Gesture Language and Graphical Feedback for Interaction in a 3D User Interface. *Computer Graphics Forum*, ročník 12, č. 3, 1993: s. 1–11, ISSN 1467-8659, doi:10.1111/1467-8659.1230001.
- [2] Bowman, D. A.; Kruijff, E.; LaViola, J. J.; aj.: An Introduction to 3D User Interface Design. *Presence: Teleoper. Virtual Environ.*, ročník 10, č. 1, Únor 2001: s. 96–108, ISSN 1054-7460.
- [3] Bowman, D. A.; Kruijff, E.; LaViola, J. J.; aj.: *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004, ISBN 0201758679.
- [4] Evans, A.; Agenjo, J.; Gonzales, E.; aj.: Thinking About 3D-design Constraints for Effective 3D Interfaces. In *Proceedings of the 2010 Conference on Visual Media Production, CVMP '10*, Washington, DC, USA: IEEE Computer Society, 2010, ISBN 978-0-7695-4268-3, s. 131–140.
- [5] Jaemoon Jung, D. H., Hanjun Park; other: A Review on Interaction Techniques in Virtual Environments. In *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management*, Curran Associates, Inc., 2014, ISBN 9781632668462, s. 1582–1590.
- [6] Kim, J.-S.; Park, J.-M.: Physics-based hand interaction with virtual objects. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, s. 3814–3819.
- [7] Lu, G.; Shark, L.-K.; Hall, G.; aj.: Immersive manipulation of virtual objects through glove-based hand gesture interaction. *Virtual Reality*, ročník 16, č. 3, 2012: s. 243–252, ISSN 1434-9957, doi:10.1007/s10055-011-0195-9.
- [8] MacKenzie, S. I.: *Human-Computer Interaction: An Empirical Research Perspective*. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann, 2013, ISBN 978-0-12-405865-1.
- [9] Mazuryk, T.; Gervautz, M.: Virtual Reality History, Applications, Technology and Future. Technická Zpráva TR-186-2-96-06, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, Únor 1996.
- [10] Mikic, D.; Skorin-Kapov, L.; Dobrijevic, O.; aj.: 3D visualization of the geographical organizational structure of the grid. In *Electrotechnical Conference, 2004*.

*MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, ročník 2, Květen 2004, s. 689–692 Vol.2.

- [11] Ortega, F. R.; Abyarjoo, F.; Barreto, A.; aj.: *Interaction Design for 3D User Interfaces*. Broken Sound Parkway, NW, USA: A K Peters/CRC Press, 2016, ISBN 1482216949.
- [12] Turunen, M.; Hakulinen, J.; Melto, A.; aj.: SUXES - user experience evaluation method for spoken and multimodal interaction. In *Annual Conference of the International Speech Communication Association*, 2009, s. 2567–2570.

# Přílohy



## Seznam příloh

**A Obsah CD**

**57**

# Příloha A

## Obsah CD

1. /experiment/data/ – data naměřená při experimentu
2. /experiment/graphs/ – grafy ze souhrnů experimentu
3. /experiment/reports/ – souhrny dat experimentu
4. /experiment/scans/ – naskenované formuláře experimentu
5. /experiment/tools/ – nástroje pro zpracování dat exp.
6. /sources/ – zdrojové kódy
7. /tex/ – zdrojové kódy textu práce
8. /xpazde12-dp.pdf – pdf verze technické zprávy
9. /xpazde12-dp-printable.pdf – pdf verze pro tisk
10. /README – popis obsahu adresářů