

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Janda**

Studijní program: Softwarové technologie a management  
Obor: Web a multimedia

Název tématu: **Knihovna pro interakci v imersivních VR zařízeních**

Pokyny pro vypracování:

Na základě svého předchozího projektu navrhnete a implementujete knihovnu, která umožní ovládání aplikace pomocí sady gest. Knihovna bude využívat hloubkový senzor a vůči aplikaci se bude chovat jako vstupní zařízení umožňující rotaci (objektu nebo scény), změnu měřítko a posun (sliding). Implementujte alespoň dvě demonstrační aplikace pro prostředí systému CAVE v jazyce C++. Navrhnete způsob testování a výsledky vhodně okomentujte.

Seznam odborné literatury:

Jiří Žára, Bedřich Beneš, Jiří Sochor, Petr Felkel: Moderní počítačová grafika (2. vydání), Computer Press, 2005, ISBN 80-251-0454-0

Vedoucí: Ing. Roman Berka, Ph.D.

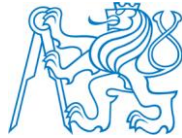
Platnost zadání: do konce letního semestru 2015/2016

  
prof. Ing. Jiří Žára, CSc.  
vedoucí katedry



  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 18. 4. 2015



**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

---

**Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce**

**Knihovna pro interakci v imersivních VR zařízeních**

**An Interaction Tool for Immersive VR Devices**

Bakalářská práce

Studijní program: Softwarové technologie a management  
Studijní obor: Web a multimédia

Vedoucí práce: Ing. Roman Berka, Ph.D.

**Martin Janda**

---

**Praha 2015**



## **Abstrakt**

Tato bakalářská práce se věnuje návrhu a implementaci knihovny pro ovládání imersivních VR zařízení pomocí gest. V práci je nejdříve uveden přehled vstupních zařízení, které se pro interakci v prostředí VR používají, dále je navržena sada gest umožňující rotaci, posun a změnu měřítka a poté je popsán návrh a implementace vlastní knihovny. Pro demonstraci funkcí knihovny jsou implementovány tři vzorové aplikace, pomocí kterých je v závěru práce knihovna otestována testy použitelnosti.

## **Abstract**

This bachelor thesis deals with design and implementation of a gesture interaction tool for immersive VR devices. The first part of the thesis describes input devices that are used for interaction in VR environment. Subsequently set of gestures is designed to allow rotation, moving and scaling, and then the thesis describes design and implementation of the library. In the conclusion, the library is tested by usability tests using sample applications.



# Prohlášení

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

V Praze dne: .....

.....

(podpis autora)



# Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Romanu Berkovi, Ph.D. za odborné vedení a cenné rady. Dále bych chtěl poděkovat účastníkům uživatelských testů, bez kterých by nemohly být výsledky práce ověřeny.





# Obsah

1	Úvod .....	11
2	Vstupní zařízení používaná pro interakci v prostředí VR .....	12
2.1	Standardní vstupní zařízení .....	12
2.2	Tracking .....	12
2.3	Typy trackerů .....	13
2.3.1	Mechanické trackery .....	13
2.3.2	Magnetické trackery .....	13
2.3.3	Ultrazvukové trackery .....	14
2.3.4	Optické trackery .....	15
2.3.5	Hloubkový senzor .....	15
3	Interaktivní techniky .....	17
3.1	Navigace .....	17
3.2	Výběr a manipulace .....	18
3.3	Ovládání systému .....	18
4	Cíle řešení .....	19
5	Návrh řešení .....	20
5.1	Požadavky .....	20
5.2	Návrh gest .....	20
5.3	Sada gest .....	21
5.3.1	Změna měřítka .....	21
5.3.2	Rotace .....	21
5.3.3	Změna pozice .....	22
5.3.4	Přechod mezi gesty .....	22
5.4	Navrhovaná struktura knihovny .....	23
5.4.1	Zpracování gesta .....	24
6	Zvolená technologie .....	25
6.1	PrimeSense Carmine 1.09 .....	25
7	Implementace .....	27
7.1	Použité technologie a knihovny .....	27
7.2	Osy ovládání .....	27
7.3	Tracker .....	27

7.4	GestureInput .....	28
7.5	Třídy Gesture .....	28
7.5.1	RecognisingGesture .....	29
7.5.2	ScaleGesture .....	29
7.5.3	RotateYGesture .....	29
7.5.4	RotateXGesture .....	29
7.5.5	TranslateXZGesture .....	30
7.5.6	TranslateYGesture .....	30
8	Vzorové aplikace.....	31
8.1	ManipulationDemo .....	31
8.2	NavigationDemo.....	32
8.3	GalleryDemo.....	33
9	Testování .....	34
9.1	Testovací metoda .....	34
9.2	Cílová skupina.....	34
9.3	Průběh testu .....	34
9.4	Analýza testu .....	35
9.5	Nalezené problémy .....	38
9.6	Zhodnocení.....	39
10	Závěr .....	40
11	Literatura.....	41

## Seznam příloh

A	Obsah přiloženého CD	43
B	Seznam úkolů pro test použitelnosti	44
C	Uživatelská příručka k aplikacím	46
D	Návod na instalaci	49
E	Návod na vytvoření aplikace s použitím knihovny	50

# 1 Úvod

V současné době existují různá zařízení, která dokážou uživateli zprostředkovat vizuální vjem virtuální reality. Jsou to například velkoplošné projekce, stereoskopické projekce nebo speciální zařízení jako CAVE<sup>1</sup>. Pokud ale chceme uživateli umožnit interagovat se zařízením, běžné ovládací prvky nejsou příliš vhodné.

Běžné vstupní zařízení jako klávesnice a myš není vhodné používat pro ovládání virtuální reality už proto, že uživatel většinou nesedí za stolem, ale pohybuje se v různém prostředí. Také se pomocí těchto nástrojů těžko realizuje práce s 3D prostorem tak, aby byla intuitivní. Zásadní problém ale je, že tyto prvky uživatele od virtuální reality odpoutávají. Proto je zajímavé zabývat se jinými možnostmi, jak umožnit uživateli interagovat s prostředím virtuální reality tak, aby to pro něj bylo co nejpřirozenější a nerušilo to jeho vjem virtuální reality.

Návrh systému, který by umožnil přirozenou interakci uživatele s VR<sup>2</sup> zařízeními, znamená v první řadě zvolit vhodnou technologii snímání vstupu. Vstupních zařízení používaných ve virtuální realitě pro tyto účely dnes existuje mnoho a není žádný standard, který by byl všeobecně uznáván jako optimální řešení. Dále je třeba navrhnout vlastní úkony (např. gesta) které musí uživatel vykonat pro zadání příkazu. Tyto úkony by měly být co nejpřirozenější, aby uživateli nerušily vjem virtuální reality, ale zároveň by měly poskytovat co nejefektivnější ovládání. Tyto dva požadavky někdy nelze plně skloubit a je třeba hledat kompromisy. Nakonec je třeba poskytnout pro tento systém jednoduché rozhraní, prostřednictvím kterého bude využitelný při tvorbě koncových aplikací pro VR zařízení.

Tato práce si klade za cíl vhodný systém interakce s VR zařízeními navrhnout a implementovat knihovnu, která umožní jeho využití v aplikacích pro VR.

---

<sup>1</sup> CAVE – Cave Automatic Virtual Environment

<sup>2</sup> VR – virtuální realita

## 2 Vstupní zařízení používaná pro interakci v prostředí VR

Tato kapitola nabízí přehled zařízení používaných v současné době pro interakci v prostředí virtuální reality, jejich výhody a nevýhody. Z tohoto přehledu a z přehledu interaktivních technik (kapitola 3) vychází návrh systému.

### 2.1 Standardní vstupní zařízení

Pro interakci uživatele s prostředím virtuální reality lze použít standardní vstupní zařízení jako je klávesnice, myš nebo joystick. Tato zařízení však nejsou vhodná pro některé aplikace například proto, že neumožňují uživateli volný pohyb, jejich ovládání není přirozené a ruší uživateli vjem virtuální reality.

Proto se pro tyto účely často používá technologie zvaná tracking, pomocí které lze zařízení ovládat pohyby těla<sup>[1]</sup>.

### 2.2 Tracking

Pomocí různých technik se snímá pozice částí těla uživatele v prostoru, data jsou posílána do počítače a převedena do virtuálního prostředí. Prostředí VR lze potom ovládat přirozenými pohyby a gesty. Speciální hardware, který toto umožňuje, se nazývá tracker.

Důležité parametry trackerů<sup>[2]</sup>:

- Přesnost (accuracy)
- Zpoždění (latency)
- Chvění (jitter)
- Posun (drift)

## 2.3 Typy trackerů

### 2.3.1 Mechanické trackery

Mechanické trackery jsou složeny z částí, které jsou vzájemně propojeny pohyblivými spoji, na nichž jsou senzory<sup>[2]</sup>.

V počítači je vytvořen kinematický model podle rozměrů jednotlivých částí trackeru. Tento model umožňuje určit polohu a orientaci jednoho konce trackeru relativně vůči druhému pomocí dat ze senzorů na spojích trackeru. Tato data jsou vyčítána v reálném čase. Zafixováním jednoho konce trackeru a připojením druhého na pohyblivý objekt lze sledovat pozici tohoto objektu v prostoru.

Výhody mechanických trackerů

- Jsou jednoduché a snadno se ovládají
- Přesnost je konstantní a je závislá na rozlišení spojových senzorů
- Na rozdíl od magnetických jsou imunní na magnetické pole v blízkosti měření
- Mají velmi malé chvění a nejmenší zpoždění ze všech typů trackerů

Nevýhody

- Omezený dosah závislý na rozměrech mechanického ramene
- Pokud jsou jednotlivé spoje dlouhé, roste jejich váha, setrvačnost a systém je náchylnější k nechtěným oscilacím
- Omezení pohybu uživatele
- Obleky s mechanickými trackery jsou těžké a ruší uživatele od vnímání virtuální reality

### 2.3.2 Magnetické trackery

Bezkontaktní trackery začaly nahrazovat dřívější mechanické, protože neruší a neomezují uživatele.

Magnetický tracker je bezkontaktní zařízení pro měření polohy, které využívá magnetické pole generované nepohyblivým vysílačem k určení aktuální polohy pohyblivého přijímače<sup>[2]</sup>.

Vysílač se skládá ze tří antén zformovaných do vzájemně kolmých cívek navinutých na feromagnetických kostkách. Tyto antény generují vzájemně ortogonální magnetické pole. Při průchodu pole přijímačem vzniká napětí, které se skládá ze tří voltů pro každé pole. Napětí z přijímačů je vzorkováno a pomocí kalibračního algoritmu je vypočítána pozice a orientace přijímače vůči vysílači. Poté jsou datové pakety poslány do počítače přes komunikační linky. Pokud je přijímač připevněný na pohyblivý objekt, může počítač sledovat pohyb tohoto objektu vůči zafixovanému vysílači.

### **AC magnetické trackery**

Vysílač magnetického pole vyvolává vířivé proudy v okolí kovů - velké kovové objekty v okolí zhoršují přesnost.

### **DC magnetické trackery**

Snaha odstranit vířivé proudy

U DC magnetických trackerů přijímač obsahuje tři magnetometry.

Výhody magnetických trackerů

- Nemusí být volný prostor mezi vysílačem a přijímačem
- Bezkontaktní

Nevýhody

- Mohou být rušeny magnetickým polem, kovové objekty v blízkosti zhoršují přesnost

## **2.3.3 Ultrazvukové trackery**

Ultrazvukový tracker je bezkontaktní zařízení pro měření polohy, které pro určení polohy pohyblivého přijímače využívá ultrazvuk produkovaný statickým vysílačem<sup>[2]</sup>.

Ultrazvukový tracker se skládá z vysílače, přijímače a elektronické řídicí jednotky. Vysílač tvoří tři ultrazvukové reproduktory namontované 30 cm od sebe na pevném trojúhelníkovém rámu. Přijímač tvoří tři mikrofony namontované na menším trojúhelníkovém rámu. Tento rám je připevněn na objekt, který chceme sledovat.

Ultrazvukové trackery jsou jednodušší a představují levnější alternativu k magnetickým.

Měření polohy ultrazvukovými trackery je založeno na rychlosti šíření zvuku. Reproktory na vysílači jsou postupně ve smyčce aktivovány a pro každý reproduktor je změřena vzdálenost ke každému ze tří mikrofonů na přijímači. Data jsou v řídicí jednotce přepočítána pomocí kalibračních konstant na pozici a orientaci a jsou poslána do počítače.

Obnovovací frekvence ultrazvukových trackerů je přibližně 50krát za vteřinu, což je méně než polovina obnovovací frekvence magnetických trackerů<sup>[2]</sup>. V případě, že je potřeba sledovat více přijímačů (například více částí těla) je možné použít časový multiplexing a sledovat až čtyři přijímače pro jeden vysílač. Nevýhoda je, že se tím sníží obnovovací frekvence pouze na 12krát za vteřinu.

Výhody

- Jednoduché, levnější než magnetické
- Bezkontaktní

#### Nevýhody

- Malá obnovovací frekvence
- Mezi vysílačem a přijímačem nesmí být žádná překážka (na rozdíl od magnetických)
- Mohou být rušeny okolními zvuky

### 2.3.4 Optické trackery

Optický tracker je bezkontaktní zařízení, které pro měření aktuální polohy objektu využívá optický senzor<sup>[2]</sup>.

#### Outside-looking-in

Sledovací zařízení (CCD kamera nebo jiný foto senzor) je zafixován a signalizační světla jsou umístěna na sledovaném objektu. Pozice je přímo změřena a orientace je dopočítána z pozičních dat. Citlivost měření se snižuje s klesající vzdáleností mezi signalizačními světly na objektu a s rostoucí vzdáleností mezi objektem a kamerou.

#### Inside-looking-out

Kamera je připevněná na sledovaném objektu. Maximální citlivost je pro změnu orientace a pracovní prostor je teoreticky neomezený.

Tradiční optické trackery jsou typu Outside-looking-in a používají se převážně v zařízeních motion capture pro vytváření animací pro počítačové hry a filmy.

#### Výhody

- Bezkontaktní
- Obnovovací frekvence je vyšší než u ultrazvukových a zpoždění je nižší
- Větší pracovní prostor

#### Nevýhody

- Mezi vysílačem a přijímačem nesmí být žádná překážka (na rozdíl od magnetických)

### 2.3.5 Hloubkový senzor

Hloubkový senzor je zařízení, které pro určení polohy sledovaného objektu využívá hloubkový obraz z kamery.

Hloubkový senzor se skládá z vysílače, který vysílá infračervený laserový paprsek a optického snímače v infračerveném spektru. Hloubkový obraz scény je získán technologií zvanou Light Coding. Při této technologii je do scény promítnut vzor světelných značek a snímán snímačem. Obraz, který se vrátí do snímače, je deformován podle polohy objektů ve scéně a jeho porovnáním s původním vzorem lze polohu objektů dopočítat. Vznikne hloubková mapa scény, ze které lze pomocí různých algoritmů rozlišit objekty a sledovat jejich pozice<sup>[4][5][6]</sup>.



## *Kapitola 2: Vstupní zařízení používaná pro interakci v prostředí VR*

### Výhody

- Bezkontaktní
- Jeden kompaktní senzor
- Není potřeba speciální oblek ani žádné další zařízení

### Nevýhody

- Omezený pracovní prostor (dosah senzoru)
- Postava je snímána jen zepředu – zakrytí některých částí těla

## 3 Interaktivní techniky

Uživatel vykonává v prostředí VR různé úlohy, které vyžadují odlišný způsob interakce. Proto rozlišujeme různé interaktivní techniky, členěné podle základních typů úloh, kterých s nimi lze dosáhnout.

Základní interaktivní techniky v prostředí virtuální reality jsou navigace v prostoru, výběr a manipulace s objekty a ovládání systému<sup>[1]</sup>.

### 3.1 Navigace

V aplikacích s rozsáhlým 3D prostředím je navigace nejběžnější úloha. Je třeba zajistit, aby se uživatel mohl snadno a efektivně pohybovat mezi lokacemi a aby ho samotná navigace nezatěžovala při plnění jiných úkolů.

Navigační úlohy lze rozdělit do tří kategorií. První úlohou je průzkum, kdy uživatel nemá žádný jasný cíl, pouze prozkoumává prostředí. Dále hledání, což je navigace, při které se uživatel snaží dosáhnout dané cílové lokace. Poslední úlohou je manévrování, které je charakteristické přesnými pohyby na krátkou vzdálenost a slouží k přemístění pohledu do pozice, která je výhodnější pro plnění jiného úkolu<sup>[3]</sup>.

Navigační úlohy lze realizovat následujícími způsoby:

1. **Fyzický pohyb:** Pro pohyb ve virtuálním prostředí je použita fyzická chůze uživatele. Tuto techniku lze realizovat například motion trackingem, chůzí na místě nebo pomocí zařízení jako běžecký pás. Tento způsob je vhodný, pokud je vyžadováno aby interakce působila co nejreálnějším dojmem.
2. **Ruční manipulace s pohledem:** Uživatel pomocí svých rukou efektivně ovládá pozici pohledu. Může se například posouvat a přitahovat k okolnímu prostředí. Tato technika je efektivní a snadno naučitelná, ale někdy může být pro uživatele fyzicky náročná.
3. **Řízení:** Řízení je kontinuální specifikace směru pohybu. Směr může být určen například otočením hlavy nebo ruky. Tato technika je běžná a efektivní.
4. **Zaměření na cíl:** Uživatel zvolí cíl, kam se chce přesunout a systém provede buďto okamžitý přesun nebo postupný přechod mezi počáteční a cílovou lokací.
5. **Plánování cesty:** Uživatel specifikuje cestu, například nakreslením na mapu prostředí, a poté systém provede po této cestě přesun.

## 3.2 Výběr a manipulace

Uživatel musí být schopen manipulovat s virtuálními objekty. Manipulační úlohy zahrnují výběr, změnu pozice a orientace objektů<sup>[3]</sup>. Ruční manipulace s objekty je hlavní složkou interakce nejen ve virtuálním ale i v reálném prostředí, proto má její implementace zásadní vliv na kvalitu celého uživatelského rozhraní.

Klasický způsob realizace manipulační techniky je manipulace pomocí 3D kurzoru, který reprezentuje virtuální ruku, jejíž pohyby korespondují s pohyby ruky uživatele. Manipulovat s objekty lze potom stejnými pohyby jako v reálném světě. Tento způsob je pro uživatele intuitivní, ale lze manipulovat jen s objekty, které jsou v dosahu uživatele. Tento problém řeší například Go-Go technologie<sup>[7]</sup> nelineárním mapováním pohybu ruky, nebo technologie využívající ray casting<sup>[7]</sup>. Při ray castingu je z virtuální ruky vržen paprsek, pomocí kterého uživatel vybírá objekty, se kterými chce manipulovat.

## 3.3 Ovládání systému

Ovládání systému je úloha, při které uživatel dává systému příkazy pro aktivování nějaké funkcionality nebo změnu režimu<sup>[3]</sup>.

Ve virtuálním prostředí nejsou použitelné všechny techniky interakce, které se běžně používají v desktopových aplikacích. Hlavním důvodem je 3D prostor, ve kterém jsou běžné úlohy z 2D prostředí neefektivní. Další problém je absence fyzické odezvy a omezení pohybu v určitém směru, jako například při pohybu myši po desce stolu<sup>[3] [8]</sup>.

Techniky ovládání systému pro imersivní virtuální prostředí lze rozdělit do následujících kategorií:

1. Grafická menu – vizuální reprezentace příkazů
2. Hlasové příkazy – nabídky přístupné prostřednictvím hlasových povelů
3. Gesta – příkazy pomocí sady gest
4. Nástroje – virtuální objekty se specifickou funkcí

## 4 Cíle řešení

Práce si klade následující cíle:

1. Navrhnout způsob interakce v imersivních VR zařízeních tak, aby byl pro uživatele pohodlnější a přirozenější než ovládání pomocí běžných vstupních zařízení jako je klávesnice a myš. Vjem virtuální reality by neměl být narušen vstupním zařízením a způsobem ovládání.
2. Ovládání navrhnout tak, aby bylo dostatečně efektivní, pro uživatele lehce zapamatovatelné a aby jeho použití nebylo fyzicky náročné.
3. Implementovat knihovnu, která bude obecně použitelná v různých aplikacích pro virtuální realitu a vnější aplikaci bude jednoduše poskytovat výstup hodnot ovládání jako ze vstupního zařízení.
4. Na základě testování ověřit funkčnost a použitelnost navrženého systému.

## 5 Návrh řešení

Tato kapitola nejdříve stanovuje požadavky pro sadu gest, knihovnu a celý systém, poté se věnuje problematice návrhu gest a jejich popisu, dále návrhu knihovny, její struktury a zpracování gest.

### 5.1 Požadavky

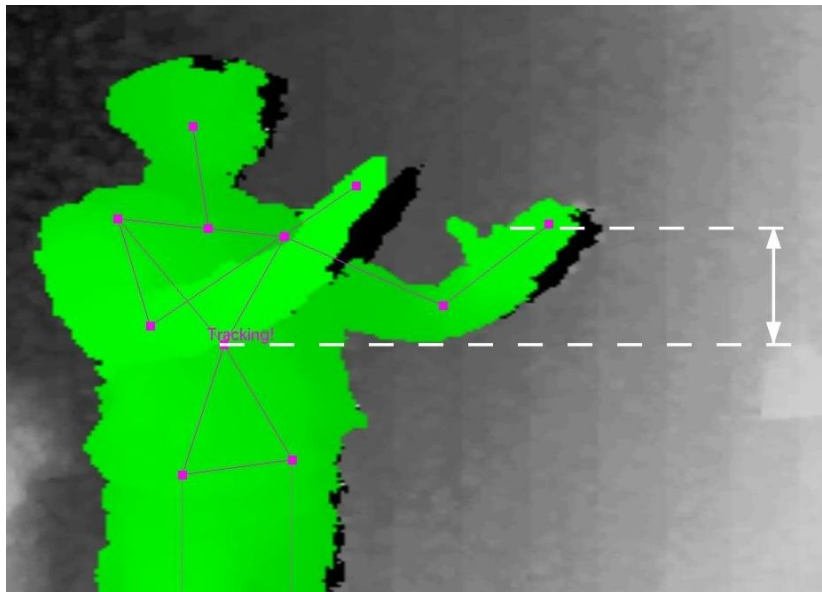
Systém bude umožňovat zadávání povelů jednoduchými gesty, která budou pro uživatele snadno vykonatelná, přirozená a spojí si je s požadovanou akcí. Příkazy budou obecně použitelné v různých aplikacích a zařízeních pro virtuální realitu zejména v aplikacích pro CAVE. Systém bude umožňovat uživateli zadávat příkaz pro rotaci objektu v trojrozměrném prostoru, posun objektu a změnu měřítko. Výstup ze zadaného povelu bude číselný údaj ze škály hodnot a systém bude umožňovat nastavit minimální a maximální hranici škály hodnot u každého povelu samostatně.

### 5.2 Návrh gest

U interakce pomocí gest je důležité zajistit, aby samotné ovládání nebylo pro uživatele příliš fyzicky náročné, aby nemusel držet ruce v nepřírodných polohách a aby systém bezpečně poznal, kdy uživatel vykonává gesto a kdy ne<sup>[9]</sup>. Z těchto důvodů byly zvoleny dvě základní pozice – aktivní a pasivní. Při pasivní pozici má uživatel ruce volně spuštěné podél těla a systém nerozpoznává žádné gesto, pouze hlídá přechod do aktivní pozice. Při aktivní pozici má uživatel alespoň jednu ruku zdvíženou před sebou. V tomto stavu uživatel vykonává gesto, systém jej rozpoznává a zpracovává. Každé gesto je ukončeno přechodem do pasivní pozice.

Pro přechod mezi aktivní a pasivní pozicí je klíčová právě poloha rukou. Pro člověka je přirozené při gestikulaci držet ruce před sebou přibližně ve výšce středu trupu nebo nad ní. Pro rozpoznání přechodu do aktivní pozice by tedy mohla být použita absolutní pozice výšky rukou nebo vzdálenost od těla. Tyto hodnoty jsou však ovlivněny fyzickými proporci konkrétního člověka, na kterých by systém měl být nezávislý. Další možnost by byla určit pozici podle rotace kloubů ramen a loktů, tyto rotace však také nejsou jednoznačně určeny. Pro některé uživatele je přirozenější mít při ovládání ruky téměř natažené, pro některé víc ohnuté a tím je rotace kloubů ovlivněna. Z těchto důvodů je pro určení přechodu do aktivní pozice použito srovnání

výšky dlaní s výškou středu trupu (viz ilustrace 1). Pozice je pak bezpečně rozpoznána nezávisle na fyzických proporcích uživatele nebo na natažení rukou.



Ilustrace 1: Aktivní pozice

## 5.3 Sada gest

Sada gest byla zvolena tak, aby odpovídala nejběžnějším operacím, byla intuitivní a obecně použitelná.

### 5.3.1 Změna měřítka

Gesto pro změnu měřítka je určeno pro zmenšování a zvětšování, případně oddalování a přibližování jakýchkoli 3D nebo 2D objektů, kamery nebo celé scény. Je realizováno přibližováním a oddalováním dlaní od sebe, což je intuitivní způsob podobný gestům používaným například na dotykových displejích.

Při vykonávání gesta uživatel začíná v pasivní pozici, kdy má obě ruce spuštěné podél těla. Poté zvedne obě ruce tak, že jsou dlaní nad horizontální linií středu trupu a začne dlaně oddalovat nebo přibližovat k sobě. Tím se aktivuje gesto změna měřítka. Pro ukončení gesta spustí uživatel obě ruce pod linii středu trupu a tím se dostane do pasivní pozice.

### 5.3.2 Rotace

Rotaci lze provádět kolem osy X nebo Y. Toto gesto je určeno především pro změnu orientace objektů nebo otočení kamery, ale jelikož je rotace kolem osy Y vykonána prostým posunem ruky do strany, lze využít i pro jiné účely, jako například listování nebo posun.

Pro vykonávání gesta uživatel zvedne pravou ruku tak, aby dlaň byla nad horizontální linií středu trupu, a pohne s ní do strany. Tím se aktivuje rotace podle osy Y a uživatel pak pohybem ruky stranou rotuje s objektem. Pro rotaci podle osy X uživatel zvedne i levou ruku a pohybem pravé ruky nahoru nebo dolů ovládá rotaci podle osy X. Gesto je ukončeno, když spustí uživatel obě ruce podél těla.

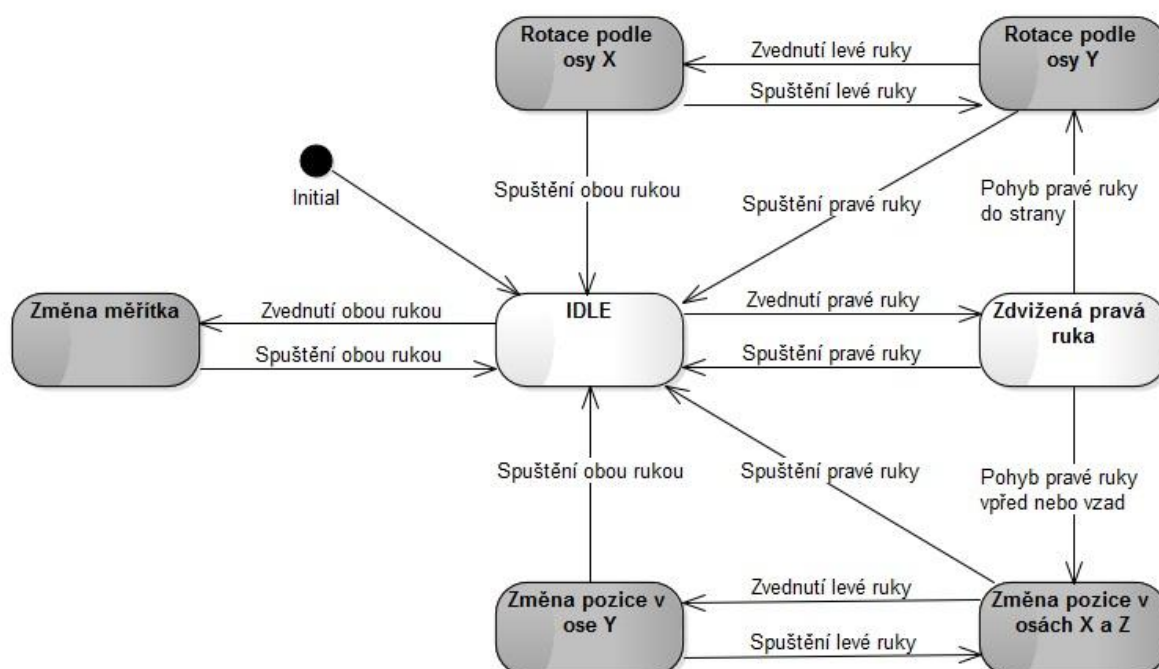
### 5.3.3 Změna pozice

Změna pozice je gesto určené pro posun objektu v 3D prostoru. Posun se provádí odděleně v rovině XZ a ve směru osy Y.

Pro aktivování gesta uživatel zvedne pravou ruku tak, aby dlaň byla nad horizontální linií středu trupu, a pohne s ní vpřed nebo vzad. Poté pozicí ruky ovládá pozici objektu v rovině XZ. Pro posun v ose Y uživatel zvedne i levou ruku a výškou pravé pak ovládá pozici objektu v ose X. Gesto je ukončeno spuštěním obou rukou podél těla.

### 5.3.4 Přejít mezi gesty

Přejít mezi jednotlivými gesty je naznačen v následujícím stavovém diagramu (ilustrace 2). Stav, ve kterých dochází k vykonávání gesta, jsou zvýrazněny šedou barvou, přechodové stavy jsou bílé.



Ilustrace 2: Stavový diagram přechodů mezi gesty

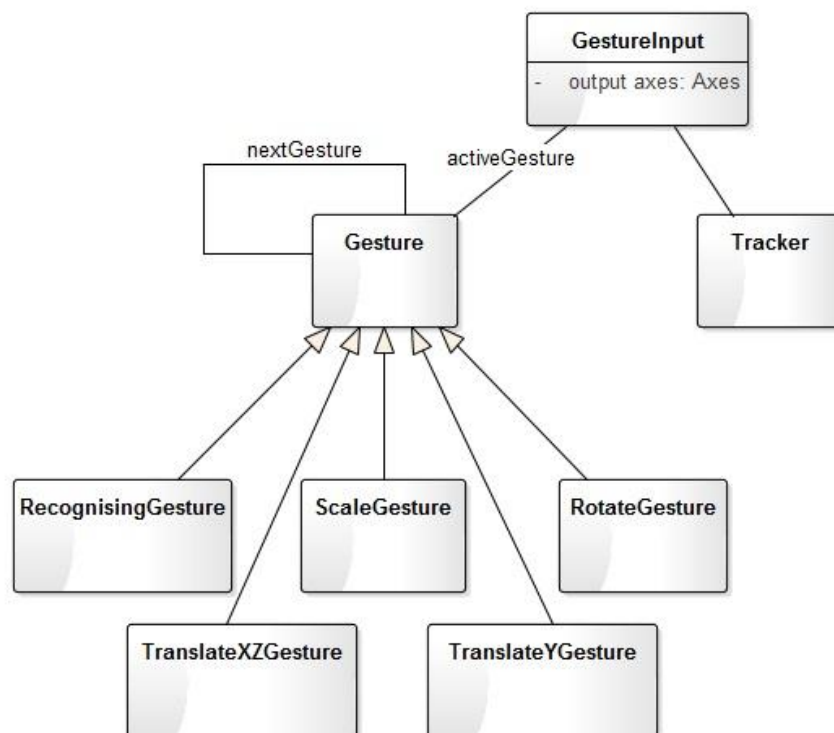
## 5.4 Navrhovaná struktura knihovny

Navrhovaná knihovna je rozdělena do tří hlavních částí. V jedné části dochází ke zpracování dat z hloubkového senzoru, v další k vyhodnocení gest a poslední část propojuje předchozí dvě části a slouží ke komunikaci s vnější aplikací.

Zpracování dat z hloubkového senzoru probíhá ve třídě `Tracker`. Při spuštění aplikace nejdřív tato třída rozpozná postavu uživatele a poté poskytuje skeleton postavy třídám zpracovávajícím rozpoznání gest.

Rozpoznání a zpracování jednotlivých gest zajišťuje třída `Gesture` a třídy, které ji rozšiřují. Pro každý typ gesta je jedna rozšiřující třída, která zpracuje data ze skeletonu a poskytne hodnotu výstupu ovládání a typ gesta, které má následovat.

Ke komunikaci s vnější aplikací slouží třída `GestureInput`, která také uchovává informace o aktivních povelích a hodnotách výstupu ovládání. Z vnější aplikace lze na této třídě zavolat metodu, kterou dojde k aktualizaci skeletonu a zpracování vstupu třídou pro příslušné gesto.



Ilustrace 3: Struktura knihovny

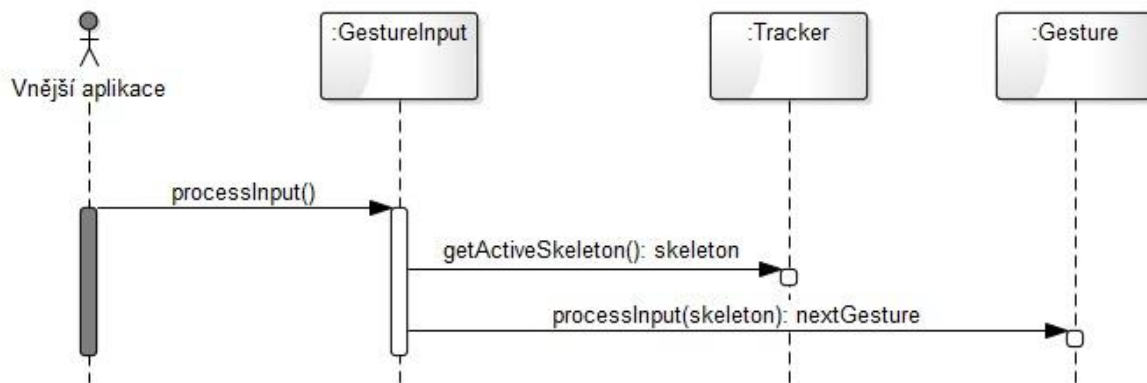


### 5.4.1 Zpracování gesta

Zpracování gesta probíhá vždy po zavolání metody `processInput` na instanci třídy `GestureInput`. Tato metoda by měla být volána vnější aplikací v rámci renderovací smyčky.

Při zpracování gesta je nejdřív zavolána metoda `getActiveSkeleton` na instanci třídy `Tracker`, kterou třída `GestureInput` uchovává. Tato metoda načte aktuální snímek z hloubkového senzoru, aktualizuje skeleton postavy a vrátí ho. Poté je volána metoda `processInput` na instanci aktivního gesta a jako parametr je předán skeleton. V této metodě dojde ke zpracování dat ze skeletonu, vyhodnocení gesta a příslušné změně výstupních hodnot ovládání.

Přečtení výstupních hodnot vnější aplikací probíhá prostřednictvím `get` metody na instanci třídy `GestureInput`.



Ilustrace 4: Diagram zpracování gesta

## 6 Zvolená technologie

V zařízeních pro virtuální realitu, například CAVE, nebo při projekci není nutné poskytnout uživateli velký prostor pro pohyb a nemusí se ani počítat s překážkami mezi uživatelem a snímacím zařízením<sup>[11]</sup>. Je ale vhodné, když se uživatel nemusí oblékat do speciálních obleků, připínat si senzory a ideálně ani nemusí držet žádné ovládací zařízení. Proto byl na základě přehledu používaných technologií (kapitola 2) jako vhodné zařízení pro snímání pohybu postavy zvolen hloubkový senzor.

Knihovna byla navržena konkrétně pro hloubkový senzor Carmine 1.09 od firmy PrimeSense. Firma PrimeSense se zabývá 3D senzory a přirozenou interakcí člověka s digitálními zařízeními od roku 2005<sup>[12]</sup> a jejich technologie je využívána v komerčních zařízeních jako například Microsoft Kinect<sup>[13]</sup>. Pro senzory PrimeSense je k dispozici open source SDK<sup>3</sup> OpenNI<sup>[14]</sup> a nadstavba NiTE<sup>[15]</sup>, která umožňuje rozpoznání postav a tracking skeletonu z dat z hloubkového senzoru.

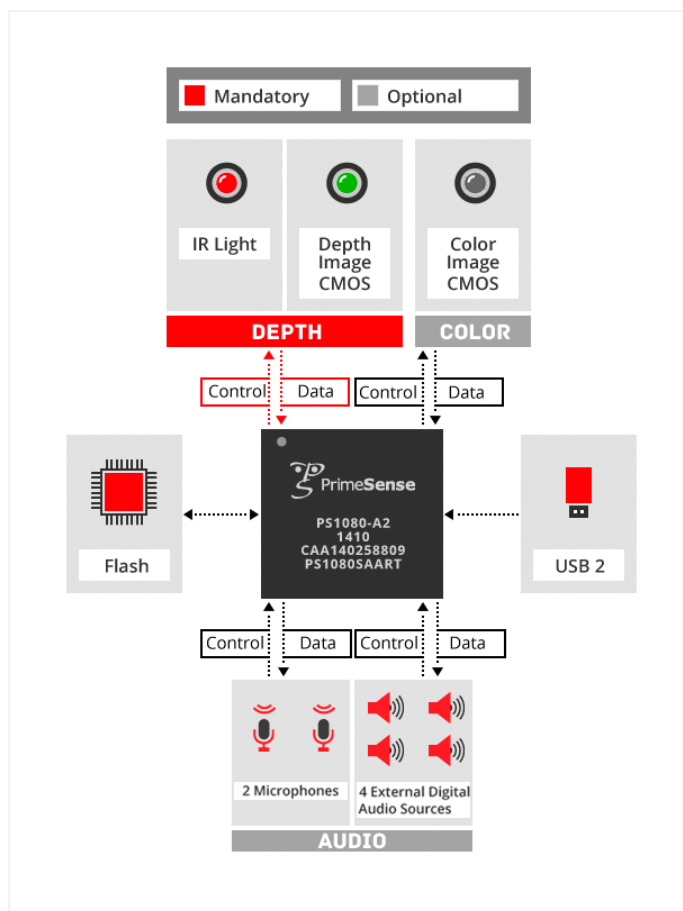
### 6.1 PrimeSense Carmine 1.09

Carmine 1.09 je hloubkový senzor pracující na principu vysílání a snímání infračerveného světla<sup>[16]</sup>, tato technologie je popsána v kapitole 2.3.5 Hloubkový senzor.

Zařízení má kompaktní rozměry – 18 x 2,5 x 3,5 cm a obsahuje vysílač infračerveného světla, senzor v infračerveném spektru, CMOS senzor pro barevný obraz a dva mikrofony. Výsledný hloubkový obraz má rozlišení 640 x 480 pixelů a přenos dat mezi senzorem a počítačem probíhá přes USB rozhraní. Prostřednictvím USB je senzor i napájen<sup>[16]</sup>.

---

<sup>3</sup> SDK – (Software development kit) soubor nástrojů pro vývoj software



Ilustrace 5: Schéma senzoru PrimeSense Carmine<sup>[19]</sup>

Parametry senzoru PrimeSense Carmine 1.09 jsou uvedeny v následující tabulce<sup>[17]</sup>:

Carmine 1.09	
Operační dosah	0,35 – 1,4 m
Zorné pole (horizontálně x vertikálně)	57,5 x 45 °
Velikost hloubkového obrazu	640 x 480 px
Prostorové rozlišení ve směru x/y při 0,5 m	0,9 mm
Hloubkové rozlišení při 0,5 m	1 mm
Maximální počet snímků za sekundu	60
Velikost barevného obrazu při 30 FPS	640 x 480 px
Rozhraní pro přenos dat	USB 2.0
Vestavěné mikrofony	2
Externí digitální audio vstupy	4
Rozměry (šířka x výška x hloubka)	18 x 2,5 x 3,5 cm
Napájení	USB
Maximální spotřeba energie	2,25 W

Tabulka 1: Parametry Carmine 1.09

## 7 Implementace

V této kapitole jsou popsány technologie a knihovny, které byly pro implementaci použity a poté samotná implementace navržené knihovny.

### 7.1 Použité technologie a knihovny

Aplikace je implementována v jazyce C++ a využívá standardní knihovny tohoto jazyka.

Dále byly pro implementaci použity tyto externí knihovny:

#### OpenNI

OpenNI (Open Natural Interaction) je volně dostupné SDK pro vývoj aplikací pracujících s 3D snímáním<sup>[14]</sup>. Pro implementaci knihovny byla použita verze 2.2.

#### NiTE

NiTE je nadstavbová knihovna OpenNI, která z dat z hloubkového senzoru umožňuje rozpoznání a tracking skeletonu postav<sup>[15]</sup>. Pro implementaci knihovny byla použita verze 2.0.

### 7.2 Osy ovládání

Před popisem vlastní implementace je nutné zmínit osy ovládání.

Kvůli jednoduchému univerzálnímu použití knihovny jako vstupní zařízení byla zavedena struktura `os`, které reprezentují výstupy ovládání. Hodnoty na jednotlivých osách jsou nastaveny příkazy pomocí gest a jsou přístupné pro čtení vnější aplikací. Z vnější aplikace je také možno nastavit minimální a maximální mez hodnot na každé ose.

Knihovna poskytuje tyto osy: *Scale*, *PositionX*, *PositionY*, *PositionZ*, *RotationX*, *RotationY*.

### 7.3 Tracker

Třída `Tracker` poskytuje skeleton postavy třídám `Gesture` pro následující zpracování a zajišťuje inicializaci knihovny NiTE. Obsahuje inicializační metodu `initialize`, ve které

## Kapitola 7: Implementace

proběhne inicializace knihovny NiTE a vytvoří se nový `UserTracker`. Pokud vytvoření `UserTrackeru` proběhne v pořádku, metoda vrátí `true`. V případě, že se vytvoření nepodaří, například proto, že není připojené správné zařízení, metoda vrátí `false`.

Hlavní metoda této třídy je `getActiveSkeleton`. Tato metoda je volána při čtení vstupu a vrací skeleton postavy uživatele z aktuálního snímku hloubkového senzoru. Na začátku metody dojde k ověření, zda byla úspěšně provedena inicializace. Pokud ne, je vyhozena výjimka `NotInitialized`. Dále dojde k načtení snímku a v případě, že neproběhne úspěšně, k vyhození výjimky `ReadFrameException`. Nakonec dojde k načtení rozpoznaných osob ze snímku a určení uživatele. Pokud je uživatel nový, je zahájen tracking, pokud už je uživatel sledován, metoda končí vrácením skeletonu uživatele. Jestliže na snímku nejsou rozpoznány žádné osoby, je vyhozena výjimka `NoTrackedUser`.

### 7.4 GestureInput

Třída `GestureInput` slouží ke komunikaci s vnější aplikací. Uchovává aktuální gesto, předává mu skeleton uživatele a vnější aplikaci poskytuje výstup os ovládání.

Konstruktor `GestureInput` musí být volán s jedním parametrem, kterým je odkaz na objekt třídy `Tracker`. Odkaz na tracker je v konstruktoru uložen do atributu třídy a je použit k získávání aktuálního skeletonu při zpracování vstupu. V konstruktoru jsou dále nastaveny defaultní hodnoty pro každou osu ovládání a její minimální a maximální krajní meze.

Hlavní metodou této třídy je metoda `processInput`, která je volána vnější aplikací v renderovací smyčce a provádí zpracování vstupu. V této metodě dojde k pokusu o získání skeletonu zavoláním `getActiveSkeleton` na instanci objektu `Tracker`. Pokud při pokusu dojde k výjimce, je metoda ukončena s návratovou hodnotou `false`. Pokud je získání skeletonu úspěšné, je zavoláno zpracování vstupu na instanci aktuálního gesta a jako parametr je předán odkaz na skeleton. Poté je jako aktuální gesto nastaveno gesto, které má následovat. V případě úspěšného zpracování vrací metoda hodnotu `true`.

Dále má třída `GestureInput` pro každou osu ovládání metodu `get`, pomocí které je získána hodnota osy, a metodu `setRange` s parametry `min` a `max`, pomocí které lze nastavit krajní meze dané osy.

### 7.5 Třídy Gesture

Pro implementaci logiky zpracování jednotlivých gest jsou určeny třídy, které jsou potomky třídy `Gesture`. Tyto třídy musí implementovat její virtuální metodu `processInput`, která zajišťuje vlastní zpracování gesta a nastavuje hodnoty výstupních os.

Rodičovská třída `Gesture` a tedy i všichni její potomci uchovávají atributy:

**axes** – data výstupních os,

**nextGesture** – odkaz na gesto, které má následovat,

**gestureType** – typ gesta z výčtu `GestureType`, typy gest jsou: `RECOGNISING`, `SCALE`, `ROTATE_X`, `ROTATE_Y`, `TRANSLATE_XZ`, `TRANSLATE_Y`, `NONE`,

**dropReset** – nabývá hodnoty *true* nebo *false*, pokud je *true*, osy jsou po každém ukončení gesta vynulovány, pokud je *false*, hodnoty os se mezi jednotlivými gesty sčítají.

Atributy `axes`, `nextGesture` a `gestureType` mají veřejné metody `get`, atribut `dropReset` má veřejnou metodu `set`.

### 7.5.1 RecognisingGesture

`RecognisingGesture` je po inicializaci nastaveno jako první gesto a v metodě `processInput` rozlišuje, jaké gesto má následovat. Pokud jsou obě ruce zdviženy, je jako následující gesto nastaveno `ScaleGesture`. Pokud je zdvižena pouze pravá ruka a následuje pohyb pravé ruky do strany, následující gesto je `RotateYGesture`, pokud následuje pohyb pravé ruky vpřed nebo vzad, následující gesto je `TranslateXZGesture`. Pokud nenastane ani jedna z těchto situací, je jako následující gesto vráceno opět `RecognisingGesture`. Přejechy mezi jednotlivými gesty jsou znázorněny v diagramu Ilustrace 2 v kapitole 5.3.4.

### 7.5.2 ScaleGesture

`ScaleGesture` obdrží v konstruktoru vzdálenost dlaní od sebe při vzniku tohoto gesta a v metodě `processInput` porovnáním této vzdálenosti s aktuální vzdáleností dlaní určí hodnotu měřítka a nastaví ji na osu `scale`. Pokud dojde k poklesu obou dlaní pod linii středu trupu, je gesto ukončeno a jako následující je vráceno `RecognisingGesture`.

### 7.5.3 RotateYGesture

`RotateYGesture` porovnáním pozice pravé ruky při vzniku tohoto gesta s aktuální pozicí určí hodnotu rotace a nastaví ji na osu `rotateY`. Pokud dojde ke zdvižení levé ruky, je jako následující gesto vráceno `RotateXGesture`, a pokud dojde ke spuštění obou rukou, je gesto ukončeno.

### 7.5.4 RotateXGesture

`RotateXGesture` v metodě `processInput` nastaví hodnotu na ose `rotateX` podle změny pozice pravé ruky v ose `y`. Gesto je ukončeno v případě, že dojde ke spuštění obou rukou. V případě, že dojde ke spuštění pouze levé ruky, následuje `RotateYGesture`.

### **7.5.5 TranslateXZGesture**

`TranslateXZGesture` nastaví hodnoty os `positionX` a `positionZ` podle změny pozice pravé ruky v osách `x` a `z`. Pokud dojde ke zdvižení levé ruky, následuje `TranslateYGesture`. Gesto je ukončeno v případě, že dojde ke spuštění pravé ruky.

### **7.5.6 TranslateYGesture**

Poslední implementované gesto je `TranslateYGesture`. U tohoto gesta je v metodě `processInput` nastavena hodnota osy `positionY` podle změny pozice pravé ruky v ose `y`. Pokud dojde ke spuštění obou rukou, je gesto ukončeno. V případě, že dojde ke spuštění pouze levé ruky, následuje `TranslateXZGesture`.

## 8 Vzorové aplikace

Pro demonstraci a testování funkcí knihovny byly implementovány tři vzorové aplikace, na kterých jsou ukázány základní techniky interakce, které lze s knihovnou realizovat. První aplikace demonstruje manipulaci s 3D objektem, druhá navigaci v 3D prostoru a třetí ovládání dvourozměrných prvků.

Aplikace jsou implementovány v jazyce C++ a k vykreslení grafiky je použito rozhraní OpenGL. Použitá verze OpenGL je 4.4.

Pro implementaci vzorových aplikací byly použity tyto knihovny:

- GLUT 3.7 – podpora pro práci s okny
- GLM (OpenGL Mathematics) – knihovna pro matematické výpočty při programování OpenGL
- Assimp 3.0 (Open Asset Import Library) – knihovna pro načtení různých formátů 3D modelů
- DevIL 1.7.8 (Developer's Image Library) – knihovna pro práci s obrázky

Všechny tyto knihovny jsou poskytovány pod licencí open source.

### 8.1 ManipulationDemo

Aplikace ManipulationDemo demonstruje interaktivní techniku manipulace s 3D objektem. Aplikace umožňuje načíst model ve formátu obj, vykreslit ho a v reálném čase na něm provádět transformace jako posunutí, rotaci a změnu měřítka. Tyto transformace jsou uživatelem zadávány prostřednictvím gest.





Ilustrace 6: Aplikace ManipulationDemo

Aplikace se spustí souborem ManipulationDemo.exe a jako parametr lze zadat název souboru s modelem, který má být načten. Pokud není zadán žádný parametr, je načten vzorový model umístěný ve složce data.

## 8.2 NavigationDemo

Aplikaci NavigationDemo je demonstrována navigace v 3D prostoru. Po spuštění je vykreslena scéna města, uživatel pomocí gest ovládá kameru a pohybuje se tak po ulicích mezi budovami. Otočení kamery je ovládáno pomocí gesta rotace podle osy Y a rychlost pohybu vpřed nebo vzad je ovládána gestem změna pozice v ose X. Pro testovací účely je možné aplikaci ovládat i klávesnicí a myší.

Aplikace obsahuje čtyři mody, které uživatele navigují pomocí červených šipek po určité trase (viz ilustrace 8).



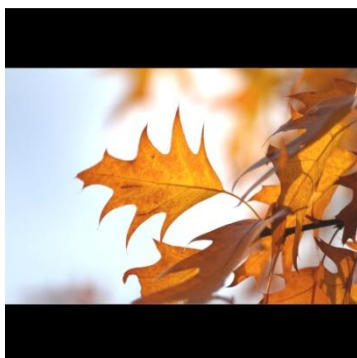
Ilustrace 7: Aplikace NavigationDemo



Ilustrace 8: Navigace pomocí šipek

## 8.3 GalleryDemo

Aplikace GalleryDemo je obrázková galerie, která demonstruje využití knihovny pro ovládání 2D aplikace. Aplikace umožňuje prohlížení sady obrázků, které je možné zvětšovat pomocí gesta změna měřítka a přecházet mezi jednotlivými obrázky pomocí gesta rotace podle osy Y.



*Ilustrace 9: Přechod snímků 1*



*Ilustrace 10: Přechod snímků 2*



*Ilustrace 11: Přechod snímků 3*

Na obrázcích 9, 10 a 11 je znázorněn přechod mezi dvěma fotografiemi v galerii.

## 9 Testování

Za účelem ověření funkčnosti a použitelnosti navrženého systému byly provedeny testy použitelnosti.

### 9.1 Testovací metoda

Jako testovací metoda byl zvolen kvalitativní test použitelnosti s uživateli. Při této metodě je vybráno několik uživatelů – participantů, kteří patří do cílové skupiny, a ti jsou pozváni k testu. Test probíhá tak, že každý participant samostatně plní předem připravené úkoly v testovaném systému. Celý průběh plnění úkolů je monitorován a je zaznamenáváno, kde dochází k případným problémům. Následně jsou na základě záznamů z testů popsány problémy systému a je vyhodnocen výsledek testu. Touto metodou lze důkladně prověřit použitelnost daného systému a lze objevit a popsat problémy, které nebyly zatím zjištěny.

### 9.2 Cílová skupina

Zařízení pro virtuální realitu se dnes kromě zábavního a uměleckého průmyslu rozšířila i do oblasti výzkumu, vzdělávání a tréninku, ale například i lékařství. Cílová skupina uživatelů aplikací pro VR zařízení je tedy široká.

K testu byli vybráni uživatelé, kteří mají alespoň základní znalosti práce s počítačem a jsou ve věku mezi osmnácti a čtyřiceti lety.

### 9.3 Průběh testu

K testování byli pozváni 4 participanti, z toho 2 muži a 2 ženy ve věku 23, 26, 18 a 39 let. Testování probíhalo ve třech vzorových aplikacích, které demonstrují funkce navrženého systému. Vzorové aplikace jsou popsány v kapitole 8. Pro tyto aplikace byl sestaven seznam úkolů, který byl zvolen tak, aby ověřil funkčnost a použitelnost systému. Úkoly jsou seřazeny od jednoduchých po obtížnější. V první části je testováno použití základních příkazů na 2D aplikaci, poté technika manipulace s objektem, což je primární určení gest, a nakonec je na poslední aplikaci otestována schopnost uživatele použít příkazy pod mírným časovým tlakem.

Seznam úkolů ve stejné formě, jako byl předložen participantům, je v příloze B. Před začátkem testu byli participanté seznámeni s průběhem testu a s testovanými aplikacemi. Po skončení testu byli dotázáni na fyzickou náročnost, subjektivní pocit z ovládnání a na preferovanou aplikaci. Délka testu trvala od čtyřiceti pěti do sedmdesáti minut.

## 9.4 Analýza testu

### Úkol č. 1: Gesta - listování

Pokuste se listovat fotografiemi. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

#### Očekávané výsledky:

Participantům nebude dělat listování problém. Gesto pro listování je intuitivní.

#### Výsledky:

Jeden participant se nejdřív snažil listovat oběma rukama, později provedl správné gesto. Participant číslo 3 se snažil listovat levou rukou.

#### Poznámky:

Listování levou rukou u participanta číslo 3 je způsobené tím, že je levák a je pro něj tedy přirozenější pro ovládnání používat levou ruku.

### Úkol č. 2: Gesta - zvětšování

Pokuste se fotografii zvětšit a poté vrátit do původní velikosti. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

#### Očekávané výsledky:

Zvětšování a zmenšování nebude dělat participantům problém. Gesto pro zvětšování je intuitivní.

#### Výsledky:

Zvětšování nedělalo participantům problém, občas nevěděli, jak ukončit gesto.

### Úkol č. 3: Ovládnání galerie

Projděte si celý seznam fotografií, vyzkoušejte listování na obě strany, některé fotografie si zvětšete.

#### Očekávané výsledky:

Participanté nebudou mít s procházením galerie problém.

#### Výsledky:

Po seznámení se s gesty byli všichni participanté schopni bez problému procházet galerií.

### Úkol č. 4: Gesta – manipulace

Pokuste se zvětšit a zmenšit auto, otáčet s ním a pohybovat po prostoru. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

#### Očekávané výsledky:

Zvětšování a zmenšování nebude dělat participantům problém, na gesta použitá pro otáčení a pohyb po prostoru nemusejí všichni přijít.

#### Výsledky:

Většina participantů se snažila posun realizovat oběma rukama. Někteří nepřišli na gesto rotace. Zvětšování a zmenšování se dařilo všem participantům bez problému.

### Úkol č. 5: Volná manipulace

Vyzkoušejte si libovolně manipulovat s autem.

#### Očekávané výsledky:

Po předvedení gest nebude dělat participantům manipulace problém, gesta si zapamatují a budou je umět použít.

#### Výsledky:

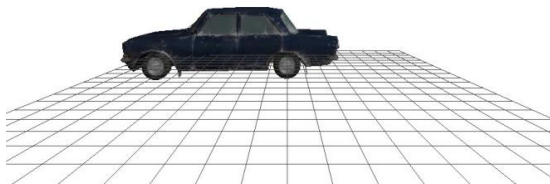
Některým participantům trvalo déle, než se v gestech zorientovali a uměli je bez problému použít. Většinou dělala největší problém rotace podle osy X. Po chvíli zkoušení byli všichni participanté schopni s autem manipulovat.

### Úkol č. 6: Přesná manipulace

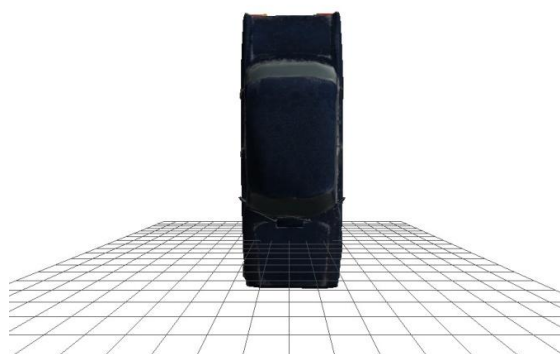
Nastavte auto co nejpřesněji do pozic uvedených na následujících obrázcích. Pozice procházejte postupně od 1. do 4.



Ilustrace 12: Pozice 1



Ilustrace 13: Pozice 2



Ilustrace 14: Pozice 3



Ilustrace 15: Pozice 4

**Očekávané výsledky:**

Pozice 1, 2 a 4 nebude dělat participantům problém. Na pozici 3 se někteří mohou zdržet, ale měli by ji zvládnout.

**Výsledky:**

Nastavení modelu do všech uvedených pozic zvládli všichni participanti bez větších problémů. Někteří občas spustili příkaz změny pozice, když chtěli spustit rotaci.

## Úkol č. 7: Gesta – navigace

Pokuste se pohybovat s kamerou po scéně. Možné povely jsou pohyb vpřed, vzad a rotace kamery do stran. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

**Očekávané výsledky:**

Participantům nebude dělat problém pohybovat s kamerou vpřed a vzad ani rotovat do stran.

**Výsledky:**

Většina participantů se snažila při aktivním gestu změny pozice zároveň zatáčet, jinak jim gesta pro navigaci nedělala problém.

## Úkol č. 8: Volná navigace

Vyzkoušejte si volný pohyb po scéně. Snažte se nevyjet ze scény a vyhýbat se kolizím s budovami.

**Očekávané výsledky:**

Někteří participanti se mohou pohybovat nejistě a nezvládat složitější manévry, všichni si ale zapamatují použitá gesta a budou je umět použít. Pro dokonalejší ovládnutí navigace v prostoru je třeba určitý trénink.

**Výsledky:**

Participantů jsou schopni se po scéně pohybovat. Některým dělá stále problém oddělit od sebe gesto pro ovládnutí rychlosti a gesto pro zatáčení.

### Úkol č. 9: Sledování trasy

Projděte trasu, která je vyznačena červenými šipkami. Na konci trasy je červený kruh, na kterém se pokuste co nejpřesněji zastavit. Při průchodu trasy se snažte pohybovat po silnici a vyhýbat se kolizím s budovami.

**Očekávané výsledky:**

Při pomalé rychlosti budou participanti schopni trasu projít.

**Výsledky:**

Některým participantům se občas spustí změna pozice, když chtějí provést rotaci.

### Úkol č. 10: Přesná navigace – zastavení

Rozjedte se na plnou rychlost vpřed a zastavte přesně v červeném kruhu. Poté se otočte o 180°, opět plnou rychlostí se vraťte a zastavte přesně v zeleném kruhu.

**Očekávané výsledky:**

Zastavení na přesné pozici se některým participantům povede až na několikátý pokus.

**Výsledky:**

Zastavit na dané pozici se dařilo participantům bez větších problémů.

### Úkol č. 11: Přesná navigace – zatáčení

Projeďte co nejrychleji trasu vyznačenou šipkami. Na konci trasy zastavte v červeném kruhu.

**Očekávané výsledky:**

Zatáčení v rychlosti může dělat některým participantům problém, všichni by ale měli být schopni se po případném vyjetí z trasy vrátit a úkol dokončit.

**Výsledky:**

Při snaze rychlého ovládní participanti často spustí změnu pozice, když chtějí provést rotaci. Participant číslo 1 často vyjížděl z trasy a úkol nedokončil.

**Poznámky:**

Časté špatné vyhodnocení gesta u tohoto úkolu je způsobené tím, že participanti jsou pod určitým tlakem a ve snaze rychle provést určitý příkaz, často provedou gesto špatně.

## 9.5 Nalezené problémy

### Nález č. 1

Pro leváky by bylo přirozenější vykonávat některá gesta druhou rukou. Gesta jsou navržena tak, že převažuje ovládní pravou rukou, což je výhodnější pro praváky.

Tento problém by bylo možné odstranit úpravou gest tak, aby gesta vykonávaná pravou rukou bylo možné vykonat i levou rukou.

## Nález č. 2

Gesta pro posun v ose Y a rotaci kolem osy X nebyla pro všechny participanty zcela intuitivní. Posun by většina participantů realizovala uchopením a posunutím oběma rukama.

Tato gesta by bylo možné nahradit intuitivnějšími, ale ovládání by potom mohlo být méně efektivní.

## Nález č. 3

Participanti občas aktivovali jiné gesto, než chtěli. Tento problém nastával hlavně při plnění úkolu č. 11, kdy se participanti snažili jednat rychle a gesta často vykonávali špatně.

## 9.6 Zhodnocení

Test použitelnosti ukázal, že uživatelé jsou schopni navržený systém interakce pomocí gest používat ve všech třech vzorových aplikacích. Uživatelsky nejpřívětivější se ukázala aplikace ManipulationDemo, která demonstuje gesta na základních povelích pro manipulaci s objektem.

Některá gesta nejsou natolik intuitivní, aby je uživatelé použili bez jejich předchozí znalosti. Uživatelé jsou ale schopni se je rychle naučit, pochopit jejich význam a zapamatovat si jaký příkaz reprezentují.

V situacích, kdy jsou uživatelé pod časovým tlakem a snaží se aplikaci ovládat co nejrychleji, dochází častěji k jinému vyhodnocení gesta, než jaké uživatel zamýšlel. Je proto vhodnější navržený systém používat v aplikacích, kde uživatel nebude vystaven časovému tlaku. Pro schopnost používat ovládání gesty spolehlivě i pod časovým tlakem je zapotřebí určitý trénink uživatele.

Ovládání aplikací pomocí navržených gest se po 45 až 70 minut dlouhém používání neprojevovalo jako fyzicky náročné.



## 10 Závěr

Pomocí navržené sady gest a implementované knihovny lze realizovat interakci v imersivních VR zařízeních bez použití běžných vstupních zařízení. Interakce probíhá přes hloubkový senzor, takže uživatel nepotřebuje žádný ovladač ani speciální oblek a může se volně pohybovat v dosahu senzoru. Uživatel ovládá aplikaci prostřednictvím gestikulace rukou, což je pro člověka přirozené, a na základě výsledků testů použitelnosti lze konstatovat, že je ovládání efektivní, pro uživatele lehce zapamatovatelné a fyzicky nenáročné. Otestováním použitelnosti třech vzorových aplikací, kde každá aplikace využívá knihovnu pro realizaci jiné interaktivní techniky, byla také ověřena univerzálnost knihovny a možnost jejího použití v různých aplikacích pro virtuální realitu.

Možnost dalšího rozšíření práce je implementovat systém tak, aby se choval jako vstupní zařízení standardu HID<sup>4</sup>. Navržené ovládání gesty by potom bylo možné použít jednoduše jako běžné ovládací zařízení (typu gamepad) bez nutnosti používat speciální knihovnu.

---

<sup>4</sup> HID – Human Interface Device

# 11 Literatura

1. HANNEMA, Don. Interaction in Virtual Reality. *Interaction in Virtual Reality*.
2. BURDEA, Grigore a Philippe COIFFET. *Virtual reality technology*. 2nd ed. Hoboken, N.J.: J. Wiley-Interscience, c2003, xvi, 444 p. ISBN 04-713-6089-9.
3. BOWMAN, Doug A., Ernst KRUIJFF, Joseph J. LAVIOLA a Ivan POUPYREV. An Introduction to 3-D User Interface Design. Presence: Teleoperators and Virtual Environments. 2001, vol. 10, issue 1, s. 96-108. DOI: 10.1162/105474601750182342. Dostupné z: <http://www.mitpressjournals.org/doi/abs/10.1162/105474601750182342>
4. Kinect for Windows Sensor Components and Specifications. [online]. [cit. 2015-01-08]. Dostupné z: <http://msdn.microsoft.com/en-us/library/jj131033.aspx>
5. MACCORMICK, John. How does the Kinect work. [online]. [cit. 2015-01-08]. Dostupné z: <http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>
6. How Kinect depth sensor works – stereo triangulation? *Mirror Image* [online]. [cit. 2015-05-21]. Dostupné z: <https://mirror2image.wordpress.com/2010/11/30/how-kinect-works-stereo-triangulation/>
7. BOWMAN, Doug A. a Larry F. HODGES. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *Proceedings of the 1997 symposium on Interactive 3D graphics - SI3D '97*. New York, New York, USA: ACM Press, 1997, 35-ff. DOI: 10.1145/253284.253301. Dostupné z: <http://portal.acm.org/citation.cfm?doid=253284.253301>
8. ROBLES-DE-LA-TORRE, G. The Importance of the Sense of Touch in Virtual and Real Environments. *IEEE Multimedia*. 2006, **13**(3): 24-30. DOI: 10.1109/mmul.2006.69.
9. ALCOVERRO, Marcel, Xavier SUAU, Josep R. MORROS, Adolfo LÓPEZ-MÉNDEZ, Albert GIL, Javier RUIZ-HIDALGO a Josep R. CASAS. Gesture control interface for immersive panoramic displays. *Multimedia Tools and Applications*. 2014, **73**(1): 491-517. DOI: 10.1007/s11042-013-1605-7. ISSN 1380-7501. Dostupné také z: <http://link.springer.com/10.1007/s11042-013-1605-7>
10. NASSIRI, Nasser, Norman POWELL a David MOORE. Evaluation of human interactions in virtual reality environments. *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry - VRCAI '08*.

## Kapitola 11: Literatura

New York, New York, USA: ACM Press, 2008, s. 1-. DOI: 10.1145/1477862.1477918.  
Dostupné z: <http://portal.acm.org/citation.cfm?doid=1477862.1477918>

11. *VisCube C1, C2 CAVE Immersive 3D Display* [online]. [cit. 2015-05-21]. Dostupné z: <http://www.visbox.com/VisCube-models.pdf>
12. *About 3D Sensors Technology - PrimeSense* [online]. [cit. 2015-05-21]. Dostupné z: <http://web.archive.org/web/20131125074040/http://www.primesense.com/about/>
13. MILLER, Michael J. *PrimeSense: Motion Control Beyond the Kinect*. Dostupné také z: <http://forwardthinking.pcmag.com/gadgets/282321-primesense-motion-control-beyond-the-kinect>
14. *About OpenNI - 3D sensing Technology for depth sensors* [online]. [cit. 2015-05-21]. Dostupné z: <http://www.openni.ru/about/index.html>
15. *NiTE 2.2.0.11* [online]. [cit. 2015-05-21]. Dostupné z: <http://www.openni.ru/files/nite/index.html>
16. *Technology - PrimeSense* [online]. [cit. 2015-05-21]. Dostupné z: <http://web.archive.org/web/20131102094504/http://www.primesense.com/solutions/technology/>
17. *PrimeSense 3D Sensors* [online]. [cit. 2015-05-21]. Dostupné z: <http://www.i3du.gr/pdf/primesense.pdf>
18. MCKESSON, Jason L. *Learning Modern 3D Graphics Programming*. [online]. [cit. 2015-01-11]. Dostupné z: <http://www.arcsynthesis.org/gltut/>
19. Zdroj obrázku:  
[http://web.archive.org/web/20131106083243im\\_/http://www.primesense.com/wp-content/uploads/2012/10/Technical\\_Overview.png](http://web.archive.org/web/20131106083243im_/http://www.primesense.com/wp-content/uploads/2012/10/Technical_Overview.png)

V ukázkových aplikacích jsou použity 3D modely volně dostupné na: <http://tf3dm.com/>

# Příloha A – obsah CD

Adresáře CD:

- **GestureInteractionTool**
  - **Documentation** – programátorská dokumentace
  - **Include** – hlavičkové soubory knihovny
  - **Lib** – soubor knihovny GestureInteractionTool.lib
  - **Samples**
    - **Bin** – spustitelné binární soubory ukázkových aplikací
    - Zdrojové soubory ukázkových aplikací
  - **Source** – zdrojové soubory knihovny
- *BP\_Janda\_Martin\_2015.pdf* – elektronická verze práce
- *BP\_Janda\_Martin\_2015.docx* – elektronická verze práce ve zdrojovém formátu

# Příloha B – Seznam úkolů pro test použitelnosti

Aplikace: GalleryDemo

## 1) Gesta – listování

Pokuste se listovat fotografiemi. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

## 2) Gesta – zvětšování

Pokuste se fotografii zvětšit a poté vrátit do původní velikosti. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

## 3) Ovládání galerie

Projděte si celý seznam fotografií, vyzkoušejte listování na obě strany, některé fotografie si zvětšete.

Aplikace: ManipulationDemo

## 4) Gesta – manipulace

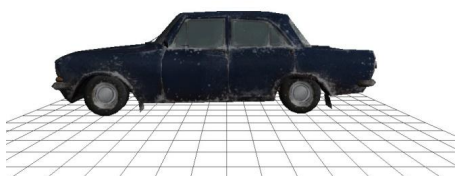
Pokuste se zvětšit a zmenšit auto, otáčet s ním a pohybovat po prostoru. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

## 5) Volná manipulace

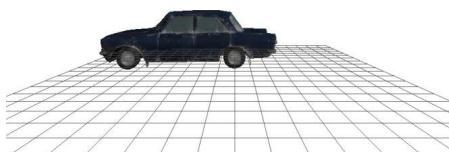
Vyzkoušejte si libovolně manipulovat s autem.

## 6) Přesná manipulace

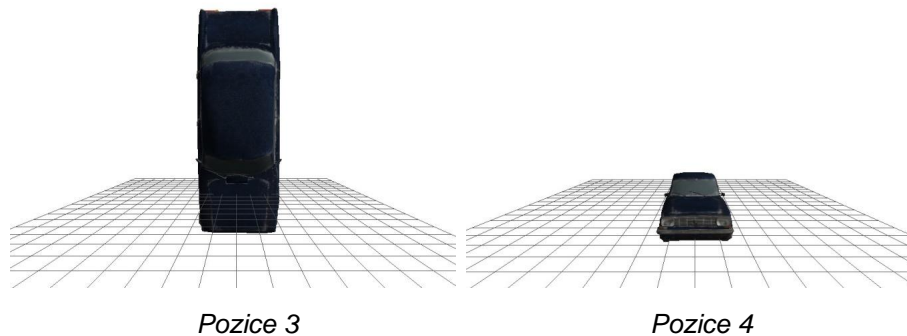
Nastavte auto co nejpřesněji do pozic, uvedených na následujících obrázcích. Pozice procházejte postupně od 1. do 4.



*Pozice 1*



*Pozice 2*



Aplikace: NavigationDemo

### **7) Gesta – navigace**

Pokuste se pohybovat s kamerou po scéně. Možné povely jsou pohyb vpřed, vzad a rotace kamery do stran. Použijte k tomu gestikulaci rukou, jaká je podle vás pro daný příkaz přirozená.

### **8) Volná navigace**

Vyzkoušejte si volný pohyb po scéně. Snažte se nevyjet ze scény a vyhýbat se kolizím s budovami.

### **9) Sledování trasy**

Projděte trasu, která je vyznačena červenými šipkami. Na konci trasy je červený kruh, na kterém se pokuste co nejpřesněji zastavit. Při průchodu trasy se snažte pohybovat po silnici a vyhýbat se kolizím s budovami.

### **10) Přesná navigace – zastavení**

Rozjed'te se na plnou rychlost vpřed a zastavte přesně v červeném kruhu. Poté se otočte o 180°, opět plnou rychlostí se vra'tte a zastavte přesně v zeleném kruhu.

### **11) Přesná navigace – zatáčení**

Projed'te co nejrychleji trasu vyznačenou šipkami. Na konci trasy zastavte v červeném kruhu.

# Příloha C – Uživatelská příručka k aplikacím

## Uživatelská příručka – ManipulationDemo

### Spuštění aplikace

Aplikace ManipulationDemo se spustí souborem ManipulationDemo.exe.

Zadáním parametru s cestou a názvem souboru, lze načíst vlastní 3D model ve formátu .obj. Pokud není zadán žádný parametr, je načten ukázkový model car.obj umístěný ve složce data v kořenovém adresáři aplikace.

Po spuštění aplikace je otevřeno okno, ve kterém je vykreslen daný model a čtvercová síť.



Pokud je připojen hloubkový senzor, lze ovládat manipulaci s modelem prostřednictvím gest.

### Ovládání aplikace

Příkaz	Klávesnice	Gesta
Zvětšení / zmenšení modelu		Změna měřítka
Otočení modelu podle osy X		Rotace X
Otočení modelu podle osy Y		Rotace Y
Posun ve směru X nebo Z		Posun XZ
Posun ve směru Y		Posun Y
Reset pozice auta	R	
Restartovat tracking	T	

### Ukončení aplikace

Aplikace se ukončí zavřením hlavního okna.

## Uživatelská příručka – NavigationDemo

### Spuštění aplikace

Aplikace NavigationDemo se spustí souborem NavigationDemo.exe bez zadávání parametrů.

Po spuštění aplikace je otevřeno okno, ve kterém je vykreslen model města.



Pokud je připojen hloubkový senzor, lze pomocí gest ovládat pozici kamery.

Mezerníkem lze přepínat módy aplikace.

### Ovládání aplikace

Příkaz	Klávesnice, myš	Gesta
Pohyb kamery vpřed / vzad	Šipka nahoru / dolů	Posun X
Otočení kamery	Myš doprava / doleva	Rotace Y
Zapnout / vypnout otáčení myší	C	
Změna modu	mezerník	
Restartovat tracking	T	

### Ukončení aplikace

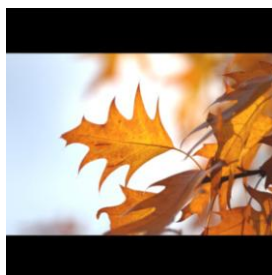
Aplikace se ukončí zavřením hlavního okna.

## Uživatelská příručka – GalleryDemo

### Spuštění aplikace

Aplikace GalleryDemo se spustí souborem GalleryDemo.exe bez zadávání parametrů.

Po spuštění aplikace je otevřeno okno, ve kterém je vykreslen první obrázek z galerie.



Pokud je připojen hloubkový senzor, lze pomocí gest listovat mezi obrázky a zvětšovat je.



### Ovládání aplikace

Příkaz	Klávesnice	Gesta
Zvětšení / zmenšení obrázku		Změna měřítka
Listování vpřed		Rotace Y +
Listování vzad		Rotace Y -
Restartovat tracking	T	

### Ukončení aplikace

Aplikace se ukončí zavřením hlavního okna.

## Příloha D – Návod na instalaci knihovny

Knihovna vyžaduje OpenNI a NiTE

Instalace OpenNI: <http://www.openni.ru/openni-sdk/index.html>

Instalace NiTE: <http://www.openni.ru/files/nite/index.html>

Pro použití knihovny stačí připojit hlavičkové soubory a soubor *GestureInteractionTool.lib* k projektu C++.

Hlavičkové soubory se nachází ve složce *GestureInteractionTool/Include*, soubor *GestureInteractionTool.lib* se nachází se složce *GestureInteractionTool/Lib*.

# Příloha E – Návod na vytvoření aplikace s použitím knihovny

1. Po instalaci knihovny (viz příloha D) je třeba do zdrojového kódu aplikace vložit následující hlavičkové soubory:

```
#include "gint/gint.h"  
#include "gint/Tracker.h"  
#include "gint/GestureInput.h"
```

2. Na začátku aplikace je třeba vytvořit objekt třídy Tracker a GestureInput a poté tracker inicializovat.

```
gint::Tracker* tracker = new gint::Tracker();  
gint::GestureInput* input = new gint::GestureInput(tracker);  
tracker->initialize();
```

3. Voláním metody processInput na instanci třídy GestureInput v rámci renderovací smyčky aplikace bude docházet ke zpracování vstupu z hloubkového senzoru.

```
input->processInput();
```

4. Výstupní data ovládání je možno číst metodou get s názvem příslušné osy na instanci třídy GestureInput.

```
input->getScale();  
input->getPosX();  
input->getPosY();  
input->getPosZ();  
input->getRotX();  
input->getRotY();
```