

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

ROZHRANÍ PRO HRY S DATAPROJEKTOREM A LEAP MOTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JONÁŠ HOLCNER

BRNO 2015



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

ROZHRANÍ PRO HRY S DATAPROJEKTOREM A LEAP MOTION

GAME INTERFACE WITH DATA PROJECTOR AND LEAP MOTION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JONÁŠ HOLCNER

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ ZAHRÁDKA

BRNO 2015

Abstrakt

Cílem této práce je navrhnout herní uživatelské rozhraní a hru využívající potenciálu zařízení Leap Motion v kombinaci s promítaným obrazem dataprojektoru. Součástí práce je kalibrace projektoru, za pomoci které je možné převádět body mezi souřadnými prostory zařízení Leap Motion a projektoru. V kombinaci s daty získanými z Leap Motion toto umožňuje uživateli ovládat aplikace pohyby rukou přímo v obraze projektoru. Vytvořené rozhraní obsahuje projektor promítající na stůl a Leap Motion sledující pohyb rukou nad tímto obrazem. Pro demonstraci připraveného rozhraní byla vytvořena hra v enginu Unity 3D.

Abstract

The goal of this paper is to develop a gaming user interface and a game itself using potential of the Leap Motion Controller in combination with projected image of a projector. This interface requires a projector calibration for user to be able to interact with the image. The calibration enables translation of points between coordinate spaces of Leap Motion and projector. Combined with data from Leap Motion, this allows user to interact with applications and games only by hands within the projected image. The solution contains a projector and a Leap Motion located over a table looking at the table desk. Leap Motion is used to track hand motions above the table. The capabilities of the proposed interface are demonstrated on a computer game, which was developed in Unity 3D engine.

Klíčová slova

Leap Motion, kalibrace kamery, kalibrace projektoru, rozhraní Leap Motion projektor, Unity 3D, ovládání gesty, ovládání pohybem rukou

Keywords

Leap Motion, camera calibration, projector calibration, camera resectioning, Leap Motion projector interface, Unity 3D, gesture control, hand motion control

Citace

Jonáš Holcner: Rozhraní pro hry s dataprojektorem a Leap Motion, bakalářská práce, Brno, FIT VUT v Brně, 2015

Rozhraní pro hry s dataprojektorem a Leap Motion

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jiřího Zahrádky

.....

Jonáš Holcner
19. května 2015

Poděkování

Tímto bych rád poděkoval Ing. Jiřímu Zahrádkovi za vedení mé práce a za jeho pomoc a užitečné rady.

© Jonáš Holcner, 2015.

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

1 Úvod	3
2 Pohybové senzory	4
2.1 PlayStation Move	4
2.2 Wii	4
2.3 Myo	5
2.4 Kinect	6
2.5 Leap Motion	6
2.6 Srovnání	8
3 Herní engine	9
3.1 Unreal Engine	9
3.2 CryEngine	10
3.3 Unity	10
4 Leap API	12
4.1 Controller	12
4.2 Frame	12
4.3 Třídy reprezentující prvky reálného světa	12
5 Kalibrace	14
5.1 Kamera	14
5.2 Projektor	15
6 Návrh rozhraní a demonstrační hry	17
6.1 Sestava	17
6.2 Kalibrace projektoru	17
6.3 Demonstrační hra	20
7 Implementace	25
7.1 Kalibrační program	25
7.2 Demonstrační hra	25
8 Experimenty	28
8.1 Vyhodnocení přesnosti kalibrace	28
8.2 Uživatelská přívětivost rozhraní a hry	29
9 Závěr	33

A Obsah CD	36
B Kalibrační aplikace	37
B.1 Potřebný software	37
B.2 Ovládání programu	37
B.3 Výstupní formát	38
C Manuál demonstrační hry	39
C.1 Potřebný software	39

Kapitola 1

Úvod

V poslední době se přichází s mnoha novými způsoby ovládní počítačů a interakce s aplikacemi. Ne všechny tyto způsoby se však ujmou a naplní svůj potenciál. Jedním z nich je i ovladač Leap Motion, který umožňuje sledovat a převádět do počítače pohyby a gesta rukou. Tato práce se snaží najít další možnosti využití tohoto ovladače a to jeho použitím v kombinaci s dataprojektorem.

Hlavním cílem práce je vytvořit herní uživatelské rozhraní s využitím sestavy projektor - Leap Motion za pomoci kalibrace projektoru, která nám umožní určit polohu uživatelových rukou vůči obrazu projektoru. Druhým stěžejním cílem práce je návrh a vytvoření hry využívající možnosti tohoto rozhraní, tedy hry, kterou bude možné ovládat pomocí pohybu a gest rukou v promítaném obraze projektoru.

V úvodní části práce (kapitola 2) se pojednává o různých zařízeních určených pro ovládní aplikací za pomoci pohybu rukou nebo celého těla. Následuje kapitola 3, obsahující srovnání několika herních enginů, což je software určený pro vývoj her. Dále je podrobněji rozebrána knihovna Leap API (kapitola 4) a princip kalibrace projektoru a kamery (kapitola 5). Poté se práce zaměřuje na návrh sestavy rozhraní, návrh postupu kalibrace projektoru a návrh demonstrační hry (kapitola 6). Předposlední kapitola 7 se zabývá implementací navržené kalibrační aplikace a demonstrační hry. V závěru práce (kapitola 8) je sepsáno, jakým způsobem probíhaly a dopadly experimenty zkoumající přesnost navržené kalibrace a dále vyhodnocení dotazníku, který byl vyplněn sadou uživatelů, kteří otestovali uživatelskou přívětivost rozhraní a demonstrační hry.

Kapitola 2

Pohybové senzory

Tato kapitola slouží jako přehled a srovnání různých druhů senzorů pohybu. Pracují na různých principech a to buď rozpoznávání v obraze, sledování pohybu pomocí akcelerometrů a nebo jako kombinace obojího. Větší pozornost je věnována senzorům, které snímají tělo nebo jeho části v obraze.

2.1 PlayStation Move

PlayStation Move (na obrázku 2.1) je ovladač od firmy Sony uvedený na trh v roce 2010 určený k použití s herní konzolí PlayStation. Jeho hlavní částí je hůlka zakončená svítící koulí, kterou uživatel drží v ruce. Tato hůlka má v sobě snímače, které snímají její pohyb (gyroskop a akcelerometr). V kombinaci s kamerou, která sleduje svítící část, PlayStation Move určuje pozici a naklonění ovladače ¹.



Obrázek 2.1: PlayStation Move

2.2 Wii

Konzole Wii (obrázek 2.2) od společnosti Nintendo přišla na trh v roce 2006. Hlavním ovladačem této konzole je Wii Remote. Ten pomocí zabudovaných akcelerometrů sleduje pohyb ve třech osách. Dále má na sobě kameru, která snímá záření infračervených diod

¹<http://blog.us.playstation.com/2010/09/07/playstation-move-the-ultimate-faq/>

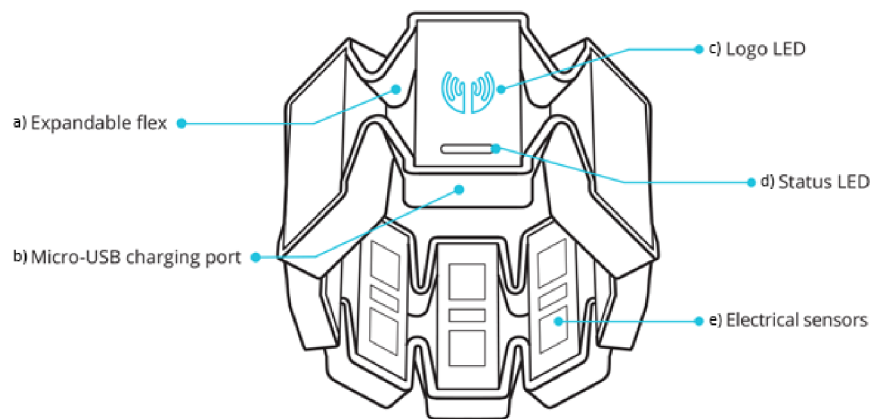
umístěných na konzoli. Tím určuje svoji polohu vůči konzoli. Tato kombinace umožňuje uživateli držícímu Remote v ruce ovládat hry pomocí pohybů [12].



Obrázek 2.2: Wii Remote od společnosti Nintendo [12]

2.3 Myo

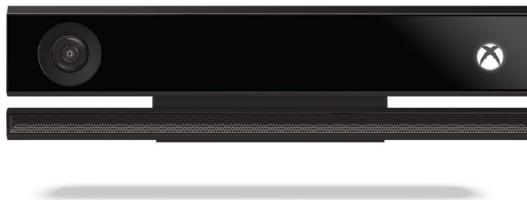
Myo [1] od firmy Thalmic Labs (obrázek 2.3) je zařízení ve tvaru náramku rozpoznávající gesta jako jsou stisk nebo roztažení ruky. Myo obsahuje gyroskop, akcelerometr, magnetometr a především EMG senzory sledující signály ze svalů uživatele. Proto je jeho ideální umístění na předloktí, kde jsou signály ze svalů nejsilnější. Zařízení je kompatibilní se systémy Windows, Mac, iOS i Android a komunikuje pomocí technologie Bluetooth.



Obrázek 2.3: ovladač Myo [1] - a) roztažitelný náramek, b) micro-USB nabíjení, c) LEDkové logo, d) LEDkový ukazatel statusu, e) elektrické senzory

2.4 Kinect

Kinect (obrázek 2.4) je pohybový senzor od firmy Microsoft. Původní verze, vydaná v roce 2010, byla určena pro spolupráci s herní konzolí Xbox 360. Druhá generace (obrázek 2.4), vydaná v roce 2014, je dostupná k novější verzi konzole Xbox One. Kinect má dostupné SDK pro Windows, které umožňuje vývojářům dělat aplikace v C++, C# nebo Visual Basicu. Neoficiální zdroje poskytují možnost vývoje i na jiných platformách².



Obrázek 2.4: Druhá generace Kinect dodávaná se zařízením Xbox One [7]

Kinect 2. generace disponuje RGB kamerou s Full HD rozlišením (1920 x 1080) s 30 snímky za sekundu. Pro snímání prostoru využívá hloubkový senzor s rozlišením 512 x 424 také s 30 snímky za sekundu. Dále má k dispozici infračervený vysílač (512 x 424, 30 snímků za sekundu), díky kterému dokáže fungovat při libovolném osvětlení. Kinect je určený pro sledování celého těla (sleduje 25 různých bodů v postavě člověka) a dokáže sledovat až šest lidí najednou [7]. Kromě pohybu těla dokáže Kinect rozpoznávat také uživatelův výraz, jestli má nasazené brýle, otevřená ústa nebo jak rychlý má uživatel srdeční tep. Za pomoci pole čtyř mikrofónů Kinect rozpoznává hlasové příkazy a směr, ze kterého byly vydány.

2.5 Leap Motion

Leap Motion (obrázek 2.5) je snímač navržený pro bezkontaktní rozpoznávání rukou, jednotlivých prstů a gest. Je zamýšlený jako ovladač umožňující nový druh interakce s počítačem. Kromě rukou rozpoznává také nástroje jako je tužka nebo jiná věc podobného tvaru a délky. Zařízení na trh uvedla firma Leap Motion, Inc. v roce 2013.

Leap Motion snímá 0,2 krychlových metrů prostoru (obrázek 2.6) v okolí zařízení, formujících tvar obrácené pyramidy. Rozsah snímání je přibližně od 25 mm do 600 mm před zařízením. Leap Motion obsahuje tři infračervené diody dodávající osvětlení. Vzdálenost dosahu snímače je limitována šířením světla z těchto diod skrz prostor - od určité vzdálenosti na ruce není vidět. Intenzita záření je omezena maximálním proudem, který lze čerpat z připojení přes USB. Diody generují 3D vzor teček infračerveného světla³, který snímají dvě černobílé infračervené kamery se zorným polem 150°

Tato data pak zpracovává software dodávaný se zařízením, který z nich získá polohu rukou vůči Leap Motion Controlleru. Leap Motion negeneruje hloubkovou mapu, ale aplikuje blíže nespecifikované pokročilé algoritmy na snímaná data. Data se získávají maximálně v rychlosti 290 snímků za sekundu přes rozhraní USB.

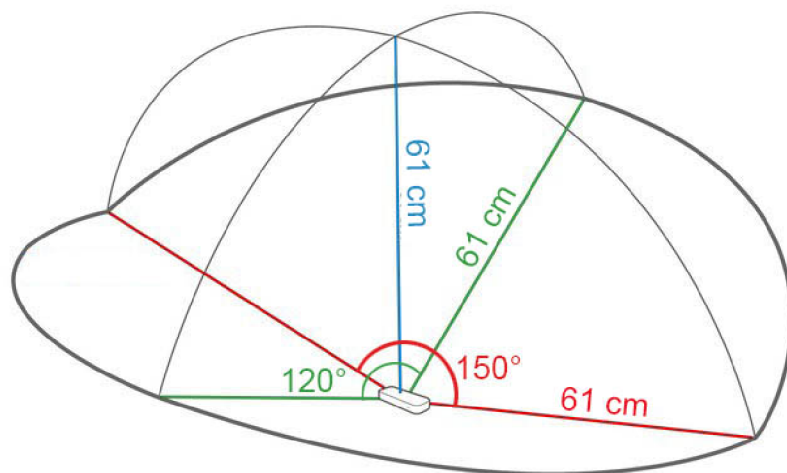
²http://openkinect.org/wiki/Main_Page

³https://youtu.be/UI5EBzU_QqM



Obrázek 2.5: Leap Motion [2] snímá ruce uživatele. Ty se pak dají názorně zobrazit na obrazovce.

Výrobce dále udává maximální přesnost 0,01 mm. Podle studie [14] však této přesnosti nelze dosáhnout. Studie, ve které pro větší přesnost při testování použili průmyslového robota, uvádí celkový průměr přesnosti Leap Motion Controlleru pouze 0,7 mm. Jiná studie [10] uvádí, že přesnost zařízení zásadně klesá, když se sledovaný objekt vzdálí nad 250 mm nad senzorem. Pro použití v této práci je tato přesnost však stále postačující.

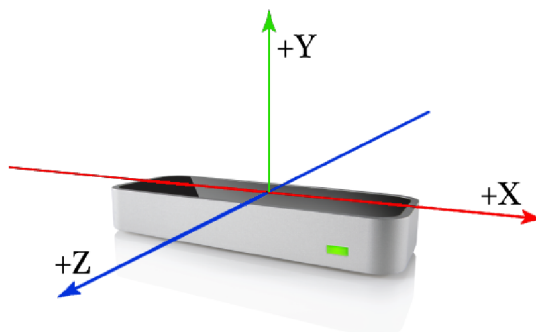


Obrázek 2.6: Reakční oblast [2] - 0,6 metrů nad ovladačem, 0,6 metrů široká na každé straně (150°), 0,6 metrů hluboká na každé straně (120°)

Leap Motion používá pravoruký souřadný systém (obrázek 2.7). Na toto je důležité dát pozor při využívání dat z Leapu v aplikacích. Počátek systému se nachází v prostředku vrcholu zařízení. Bod $[0, 0, 0]$ by se získal při položení prstu doprostřed zařízení v případě, že by zařízení umožňovalo snímat v nulové vzdálenosti. Leap Motion však potřebuje, aby ruka byla aspoň trochu vzdálená od povrchu.

Souřadnice jsou udávány v milimetrech. Úhly jsou poskytovány v radiánech. Ovladače pro Leap Motion jsou dostupné pro Windows, Mac i Linux.

Distribuce aplikací pro Leap Motion je řešena přes webový obchod LEAP Motion App Store⁴.



Obrázek 2.7: Leap Motion používá pravoruký souřadný systém zařízení [2]

Informace v této sekci vycházejí z [2, 5].

2.6 Srovnání

Pro tuto práci jsou vhodnější zařízení, která poskytují informace o pohybu rukou, abychom mohli rozpoznávat různá uživatelská gesta. Tím se stávají nevhodnými Wii a PlayStation Move. Myo sice poskytuje tyto informace, ale nemá možnost určit polohu ruky v prostoru. Tato vlastnost je zapotřebí pro vytvoření rozhraní spolupracujícího s projektorem. Zbývá nám Kinect a Leap. Oba tyto senzory jsou vhodnými kandidáty. Pro rozlišování jednotlivých prstů je však lepší a přímo určený Leap Motion a proto je i použit v této práci.

⁴<https://apps.leapmotion.com/>

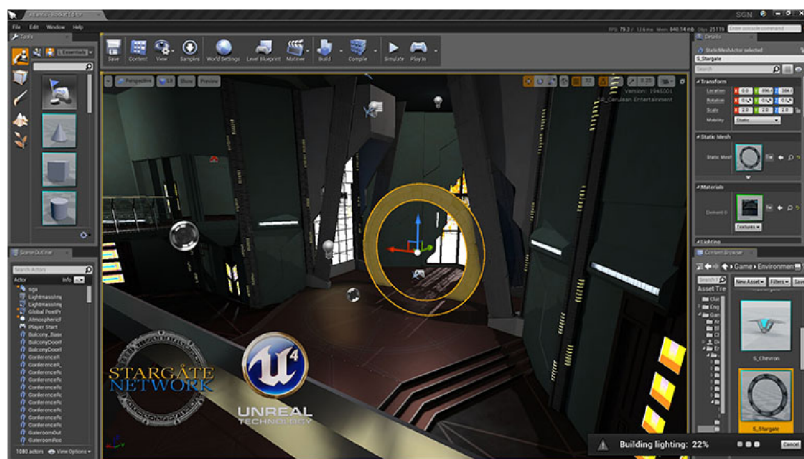
Kapitola 3

Herní engine

Herní engine [15] je softwarový framework uzpůsobený pro vývoj video her. Zpravidla umožňuje vykreslovat a upravovat herní modely, počítat fyzikální děje a pomocí skriptů vytvářet herní mechanismy. Některé enginey jsou dostupné zdarma, poté se platí za nadstandardní služby či procenta výtěžku ze hry, a jiné jsou dostupné pouze za peníze. Tuto kapitolu zde uvádím, protože jsem se rozhodl pro vývoj demonstrační hry použít herní engine, který tento vývoj výrazným způsobem usnadní. Kapitola tedy poskytuje přehled a možnosti různých engineů.

3.1 Unreal Engine

Unreal Engine¹, herní engine od firmy Epic Games, původně vznikl v roce 1998. Základní verze Unreal Engine, včetně zdrojových kódů, je od března tohoto roku poskytována zdarma a peníze firma vybírá od vývojářů až od určité výše zisků. Nedávno byl také uveden plugin pro Unreal Engine s podporou zařízení Leap Motion.



Obrázek 3.1: Unreal Engine 4

Engine je zaměřený na vývoj takzvaných FPS² her. První a nejnámější hrou vyvinutou

¹<https://www.unrealengine.com/what-is-unreal-engine-4>

²First-person shooter neboli střílečka z pohledu první osoby

v tomto programu je Unreal³, podle které se program také jmenuje.

Hry se v tomto enginu vyvíjejí v jazyce C++ a dají se použít na různých platformách: Windows, Xbox, Linux, PlayStation, iOS, Android a ve webových prohlížečích. Pro sdílení obsahu mezi vývojaři je dostupný Unreal Engine Marketplace⁴, ve kterém lze zakoupit předpřipravené hry nebo herní modely a prostředí.

3.2 CryEngine

Firma Crytek začala s vývojem CryEngine⁵ v letech 2001-2004 pro svoji první hru Far Cry. V základu je CryEngine dostupný komerčně a to formou měsíčního předplatného. Zato se však neplatí poplatky ze zisků ze hry. Pro nekomerční použití je dostupná starší verze CryEngine, pro kterou nejsou vydávány aktualizace. Slouží tedy spíše pro ukázkou možností CryEngine.



Obrázek 3.2: CryEngine

CryEngine si zakládá na vysokých detailech prostředí a precizním a realistickým fyzikálním modelu. Jeho nevýhodou je větší čas nezbytný k pochopení základů, což je způsobeno výraznější uživatelskou náročností.

V CryEngine se vyvíjí v jazyce C++. Umožňuje vývoj her pro tyto platformy: Windows, Linux, PlayStation, Wii U, Xbox, iOS a Android.

3.3 Unity

Unity⁶ je herní engine od firmy Unity Technologies vydaný v roce 2005. Unity je dodáváno ve dvou verzích - Unity Personal, základní verze zdarma a Unity Pro, které za měsíční poplatek umožňuje pokročilejší funkce.

Skripty je v Unity možné vytvářet ve třech různých jazycích - JavaScript, C# a Boo. Unity je považováno za nejvhodnější herní engine pro začátečníky v oblasti 3D her a to nejen díky svému přehlednému prostředí, ale také protože je pro něj připravené široké spektrum názorných návodů. Unity Asset store⁷ poskytuje možnost prodeje a distribuce

³<http://en.wikipedia.org/wiki/Unreal>

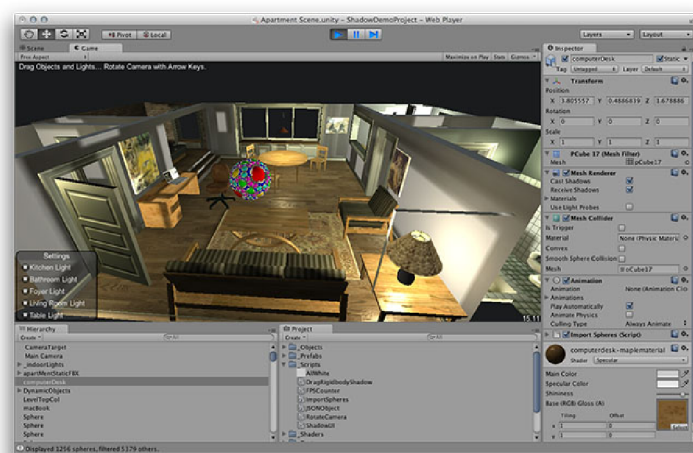
⁴<https://www.unrealengine.com/marketplace>

⁵<http://cryengine.com/>

⁶<http://unity3d.com/>

⁷<https://www.assetstore.unity3d.com/en/>

balíčků s připravenými skripty, texturami a jinými doplňky. Pro Unity je dostupná podpora knihovny OpenCV a Leap Motion API.



Obrázek 3.3: Unity 3D

Unity klade důraz na přenositelnost her. Jednou vytvořená hra se dá snadno sestavit pro různé platformy. Oproti běžně podporovaným platformám jako jsou Windows, Linux, PlayStation, Xbox a Wii podporuje také BlackBerry a Windows Phone.

Kapitola 4

Leap API

K Leap Motion Controlleru je pro vývojáře ze stránek výrobce¹ dostupné Leap Motion Developer SDK poskytující Leap API². Toto rozhraní umožňuje vývojářům pohodlný přístup k datům získávaných ze senzoru. API je dostupné pro řadu různých jazyků. Součástí Leap Motion SDK je dokumentace pro Javascript, Unity, C#, C++, Javu, Python, Objective-C a nově také pro Unreal Engine. Je tedy vidět, že Leap Motion Controller má širokou podporu. API umožňuje přístup k třídám popsaných v dokumentaci [3]. Následuje popis nejdůležitějších tříd tohoto API.

4.1 Controller

Třída `Controller` reprezentuje fyzické zařízení. Obsahuje informace související s konkrétním zařízením jako je jeho `id` nebo orientace a stav připojení. Přístup k ostatním datům a třídám je možný pouze skrze tuto třídu.

4.2 Frame

Leap Motion snímaná data poskytuje po snímcích - záznamech z určitého času - nazývaných se `Frame`. Tato třída poskytuje přístup k počátku datového modelu, který je uložen ve snímku. Každý `Frame` obsahuje data jako jsou poloha a stav snímaných rukou. Třída uchovává v paměti historii 60 snímků nazpět. API poskytuje dva způsoby, jakým `Frame` získat:

- Polling (dotazování) - v případě potřeby (na dotázání), `Controller` poskytne `Frame`. Tento způsob je vhodný pokud aplikace přirozeně běží v cyklu a na jeho začátku si pokaždé vyžádá aktuální data.
- Callbacks (zpětná volání) - tento způsob vyžaduje použít `Listener` (posluchač). Tento `Listener` obdrží nový `Frame` kdykoliv je k dispozici.

4.3 Třídy reprezentující prvky reálného světa

Jednotlivé objekty mají při výskytu ve scéně přiřazený unikátní identifikátor. Tento identifikátor jim zůstává napříč snímky. V případě, že se objekt ze scény ztratí a později se zase

¹<https://developer.leapmotion.com/downloads>

²Application Programming Interface

objeví, je možné že dostane přiřazený jiný identifikátor, protože bude rozpoznán jako nový objekt. Toto je častý problém Leap Motion Controlleru - ruka, kterou zrovna Leap Motion sleduje, se v jednu chvíli rozpozná jako nová ruka a dostane přiřazené nové `id`.

Hand - Třída **Hand** uchovává a poskytuje informace o vlastnostech rukou, které zrovna detekuje Leap Motion. Data obsahují mimo jiné polohu a směr natočení dlaně a intenzitu sevření dlaně. Skrze tuto třídu je možné získat **Pointables** patřící k této ruce.

Pointable - Tato třída reprezentuje prsty a nástroje, jako je tužka, které Leap Motion také snímá. Poskytuje informace o poloze a natočení pro každý z těchto objektů. Dělí se na **Finger** a **Tool**. Pro prsty je možné zjistit polohu pro každý z jejich kloubů.

Gesture - Třída **Gesture** reprezentuje pohyby uživateli ruky, které software ovladače Leap Motion rozpoznal jako jeden z přednastavených pohybů. Tato gesta jsou pozorována zvlášť pro každý prst nebo nástroj a mohou být použita jako způsob ovládní nebo příkazu. **Gesture** se dělí na

- **Circle** – Prst pohybující se ve směru kruhu.
- **Swipe** – Dlouhý, přímočarý pohyb ruky a jejích prstů.
- **KeyTap** – Pohyb prstu jako při klikání na klávesnici.
- **ScreenTap** – Pohyb prstu jako při klikání na svislou obrazovku.

Kapitola 5

Kalibrace

Vytvářím uživatelské rozhraní, ovladatelné pohyby a gesty rukou přímo v obraze projektoru. Takové rozhraní vyžaduje správně nakalibrované jednotlivé komponenty. V mém případě je to Leap Motion a projektor. Tato kapitola popisuje, jakým způsobem funguje kalibrace kamery a systému kamera - projektor, ze které vychází i použitá kalibrace systému Leap Motion - projektor.

5.1 Kamera

Camera resectioning¹ neboli kalibrace kamery je způsob, jakým se získávají externí a interní parametry kamery. Jak se můžeme dočíst v práci [16]: „Správná kalibrace kamery dává možnost vypočítat prostorovou pozici pozorovaného objektu známé velikosti vzhledem ke kameře. Kamera provádí mapování mezi 3D světovým metrickým systémem a 2D souřadným systémem obrazu v pixelech. Aby bylo možné provést inverzní proces, tedy spočítat 3D metrické souřadnice z 2D pixelů, je nutné znát prvky interní a externí orientace kamery. Tyto parametry jsou výsledkem kalibračního procesu.“

Tsai, R. Y [13] popisuje tyto interní parametry:

- f_x, f_y - ohnisková vzdálenost
- k_1 - koeficient zkreslení čočky
- S_x - koeficient faktoru nejistoty daný chybou časování kamery
- c_x, c_y - poloha hlavního snímkového bodu

V základním modelu bez koeficientu zkreslení čočky pak matici interních parametrů znázorňuje matice 5.1.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Externí parametry (matice 5.2) slouží pro transformaci z 3D světového souřadného systému snímaného objektu do 3D souřadného systému kamery [13].

¹http://en.wikipedia.org/w/index.php?title=Camera_resectioning&oldid=639926037

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (5.2)$$

- r_i - určují rotaci
- t_i - určují posun

Výsledná rovnice pro převod bodů mezi 2D a 3D souřadným systémem - matice 5.3.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5.3)$$

- u, v - souřadnice ve 2D souřadném systému
- X, Y, Z - souřadnice ve 3D souřadném systému

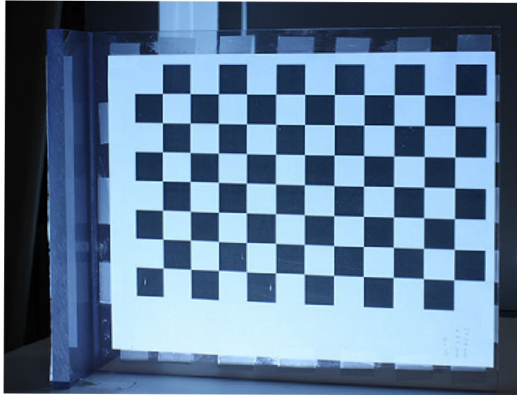
Jednou z nejznámějších metod kalibrace prezentuje Zhang [17]. Jeho technika vyžaduje rovinnou šachovnicovou mřížku se známou velikostí zachycenou v nejméně dvou orientacích (v různém natočení). V jednotlivých orientacích tohoto vzoru se vyhledají body rohů. Z těchto bodů se následně vypočítá projektivní transformace mezi n různými nasnímanými obrazy.

Pro výpočet kalibrace je tedy potřeba získat množinu dvojic 2D obrazové roviny v pixelech a 3D bodů souřadného systému kalibračního vzoru.

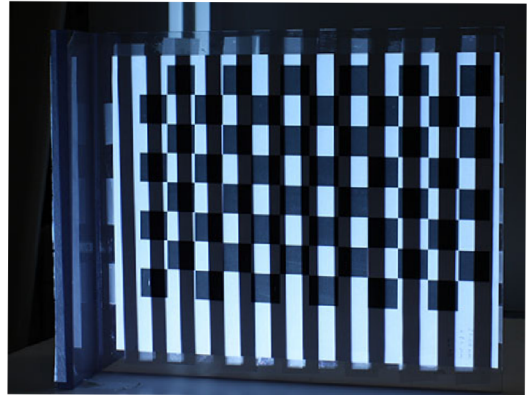
5.2 Projektor

K projektoru při kalibraci můžeme přistupovat jako k inverzní kameře - promítá obraz, místo aby ho snímala [8]. Způsob kalibrace je tedy stejný jako kalibrace kamery, jenom je potřeba k určení korespondujících 2D a 3D bodů v obraze projektoru použít kameru, která bude body snímat místo projektoru. Kalibrací projektoru získáme jeho polohu vůči kameře.

V práci [11] postupují tak, že při snímání snímků pro kalibraci kamery promítají projektoem do šachovnicového vzoru sekvence šedých pruhů (obrázek 5.1). Pro každou orientaci vzoru pak snímají několik snímků místo jednoho. Každý pixel z těchto snímků přidruží s řádkem a sloupcem projektoru (obrázek 5.2). Poté určí lokální homografii pro každý roh šachovnicového vzoru v obraze. Nakonec každý z těchto rohů konvertují ze souřadnic kamery do souřadnic projektoru podle nalezené homografie.

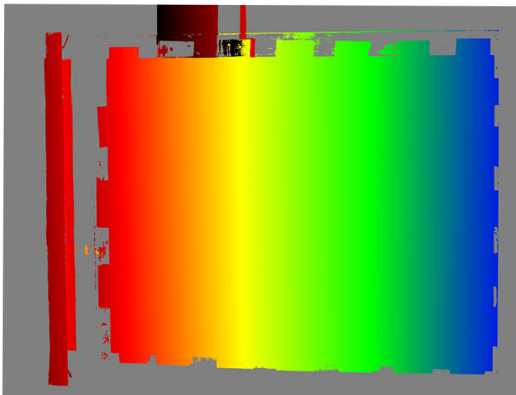


(a) Projektor promítá pouze bílé světlo

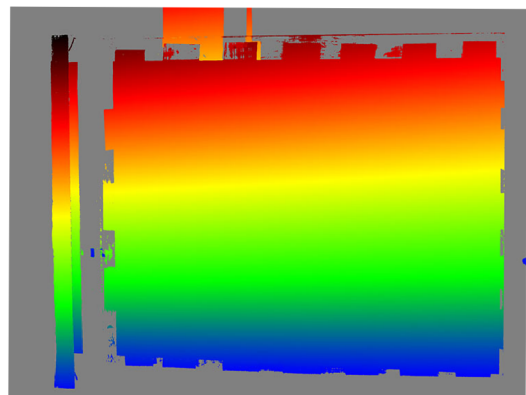


(b) Projektor promítá sekvenci šedých pruhů

Obrázek 5.1: Projekce sekvence šedých pruhů do šachovnicového vzoru. Pomocí této sekvence se určí lokální homografie rohů šachovnicového vzoru. [11]



(a) Přiřazení ke sloupcům



(b) Přiřazení k řádkům

Obrázek 5.2: Pixely stejné barvy korespondují se stejným řádkem nebo sloupcem projektoru. [11]

Kapitola 6

Návrh rozhraní a demonstrační hry

V této kapitole je popsán návrh všech důležitých součástí této práce. Nejprve je zde popsána sestava, ze které se rozhraní skládá (sekce 6.1). Následuje návrh průběhu kalibrace a jejího testování (sekce 6.2). Poslední sekce této kapitoly se zabývá návrhem demonstrační hry (sekce 6.3).

6.1 Sestava

Zde jsou popsány jednotlivé prvky ze kterých se navržené rozhraní skládá (obrázek 6.1).

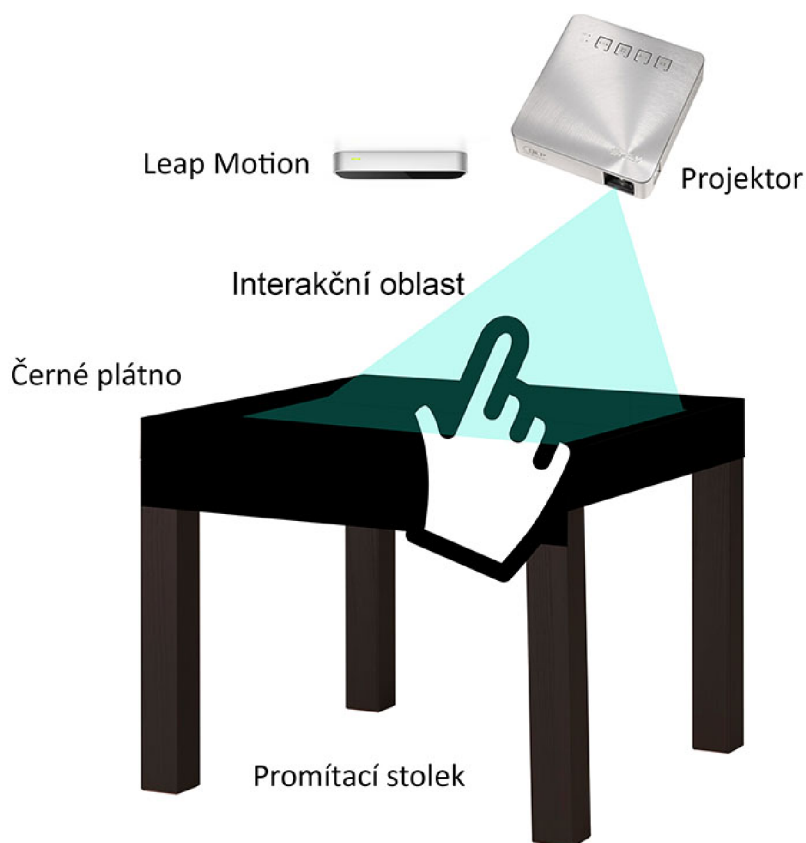
- Leap Motion, který snímá ruce uživatele.
- Projektor, který promítá upravovaný obraz. Konkrétní projektor, který bude použitý při testování, je Asus S1 s rozlišením 800x600.
- Stativ či jiné upevnění pro projektor a Leap Motion.
- Pracovní stůl, na který se promítá obraz, překrytý černým plátnem.

Leap Motion je primárně určen k použití v položené poloze, kdy v prázdném prostoru nad ním sleduje pohyb a gesta rukou. Při tomto používání vykazuje nejlepší výsledky. Problém je, chceme-li senzor používat proti nějakému povrchu, typicky stolu. Ovladač nemá problém s prací vzhůru nohama, ale při použití proti většině povrchů má problém s určováním rukou. Tento problém vzniká, protože Leap Motion využívá ke snímání infračervené kamery a jejich světlo se od povrchu odráží a senzor pak ruce v obraze správně nerozpozná. Proto je důležitou součástí sestavy černé plátno, které pohlcuje přebytečné světlo vydávané senzorem umožňuje mu tak korektně fungovat. Nevýhodou tohoto řešení je, že černé plátno pohlcuje také světlo promítané projektorem, což způsobuje horší kvalitu vzhledu projekce.

Leap Motion i projektor musí být v nehybném stavu. Pokud se pohne s jedním či druhým zařízením po úspěšné kalibraci, přestanou zjištěné hodnoty odpovídat jejich vzájemné pozici a tím pádem přestanou odpovídat i hodnoty transformací pozice rukou ze souřadného systému Leap Motion do souřadného systému obrazu projektoru.

6.2 Kalibrace projektoru

Jak již bylo řečeno v kapitole 5, pro kalibraci projektoru je nutno použít jiné zařízení, obvykle kameru, která snímá body v obraze projektoru a je s její pomocí možné určit odpovídající 3D souřadnice bodů ve scéně.



Obrázek 6.1: Navržené rozhraní - Leap Motion i projektor směřují na stůl. Po úspěšné kalibraci bude moci uživatel ovládat aplikace pomocí rukou v prostoru mezi stolem a ovladačem Leap Motion.

V našem případě máme k dispozici Leap Motion, který je ze svého principu dodáván jako nakalibrované zařízení. Udává 3D polohu rukou ve svém souřadném systému (figuruje jako počátek systému). Při kalibraci systému Leap Motion - projektor zastupuje pozici kamery, která běžně provádí mapování mezi 3D světovým metrickým systémem a 2D souřadným systémem obrazu v pixelech.

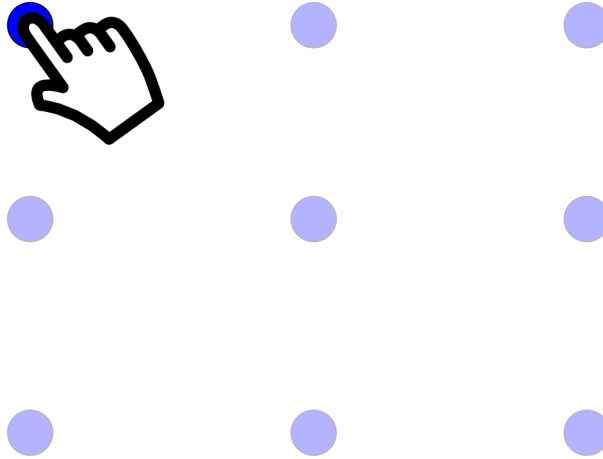
Kalibrací projektoru získáme jeho polohu vůči ovladači Leap Motion. Toto je zapotřebí, abychom mohli upravovat obraz projektoru na základě pohybu uživatelových rukou. Správná kalibrace by měla dávat možnost pracovat ve veškerém prostoru mezi obrazem promítaným na stůl a Leapem. Je tedy nutné kalibraci navrhnout tak, aby fungovala i v prostoru nad stolem a ne pouze v jeho rovině.

6.2.1 Získání bodů a výpočet kalibrace

Vstupem kalibrace je dvojice souřadnic korespondujících 2D bodů v pixelech obrazu projektoru a 3D bodů v souřadném systému Leap Motion. Je nutné si dopředu zvolit souřadnice bodů v obraze projektoru a k nim získat jejich korespondující souřadnice v prostoru s pomocí Leap Motion. Proces získávání souřadnic bude následující:

Zvolíme si souřadnice 2D bodů jako matici velikosti 3x3, kterou budeme postupně promítat do obrazu projektoru. Ke každému z bodů uživatel přiloží prst, tím zjistíme kore-

spondující souřadnice bodu v prostoru. Po vteřině, kdy uživatel nepohybuje s prstem se souřadnice uloží, bod zmizí a rozsvítí se následující bod. Aby kalibrace fungovala správně nejen v rovině obrazu projektoru, ale i nad ním, je nutné prsty k bodu přikládat v různých výškách nad projekční plochou. Proces je znázorněn na obrázku 6.2. Jeden takovýto průběh nazvěme *kalibrační běh*. Počet kalibračních běhů je závislý na požadované přesnosti kalibrace.



Obrázek 6.2: Návrh kalibračního procesu - uživatel přikládá prst k postupně promítaným bodům dopředu zvolených souřadnic, aby bylo možné k nim získat korespondující bod v prostoru. Ze získaných korespondencí se dá následně provést kalibrace.

Poté se provede výpočet kalibračních matic. Při výpočtu se kromě vstupních bodů použije také odhadnutá matice interních parametrů. Pro projektor Asus S1 je to matice 6.1.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 962 & 0 & 400 \\ 0 & 962 & 300 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.1)$$

- c_x - polovina šířky obrazu projektoru v pixelech
- c_y - polovina výšky obrazu projektoru v pixelech
- f_x, f_y - ohnisková vzdálenost vyjádřená v pixelech určená pro projektor Asus S1 použitý v rozhraní při testování

Podrobnější informace o matici interních parametrů již byly zmíněny v kapitole 5.3.

6.2.2 Testovací promítání

Tato fáze bude sloužit jako rychlý a neexaktní způsob ověření přesnosti kalibrace. Na souřadnice v obraze projektoru, která se určí jako transformovaná pozice uživatelova prstu, se bude promítat bod. V případě, že kalibrace proběhla úspěšně, se tento bod vykreslí přímo na prst. Přepočítání souřadnic ze souřadného systému Leap Motion do obrazu projektoru bude probíhat pomocí získaných externích a interních parametrů projektoru.

6.2.3 Exaktní vyhodnocení přesnosti

Pro exaktní vyhodnocení přesnosti bude použita chyba reprojekce [9]. Aby se pro vyhodnocení použily jiné body než které byly použity pro kalibraci, nasnímá se nových 5*5 dvojic bodů stejným způsobem, jako byl popsán v sekci 6.2.1. Pro tyto body se prst bude přikládat pouze v rovině projekce. Pro každý 3D bod ze souřadného systému Leap Motion se vypočítá, kam by se promítl s použitím získaných kalibračních parametrů, a zjistí se rozdíl mezi touto hodnotou a hodnotou, kam se promítnout měl, tedy 2D souřadnicí v obraze projektoru pro kterou se tento její korespondující 3D bod naměřil. Výsledek udává celkovou průměrnou chybu reprojekce přes všechny měřené body (rovnice 6.2).

$$reprojError = \sqrt{\frac{\sum d(x_i, \hat{x}_i)^2}{n}} \quad (6.2)$$

6.3 Demonstrační hra

Tato sekce se zabývá návrhem hry, na které chceme ukázat možnosti použití navrženého rozhraní mezi projektorem a Leap Motion.

6.3.1 Princip hry

Hra bude ve stylu jednoduché vesmírné arkády. Uživatel ovládá vesmírnou loď, proti které létají různé překážky. Těmto překážkám se dá buďto vyhnout, nebo je zničit pomocí lodního děla. Do děla je nutné sbírat náboje. Hra probíhá v kolech - v každém dalším kole se objeví více a navíc rychleji se pohybujících překážek a nepřátel. Za každé úspěšně dokončené kolo a sestřelené předměty se získávají body.

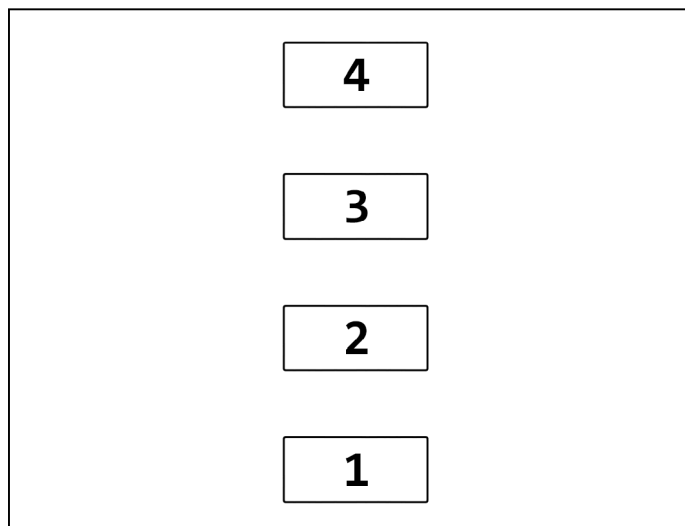
Protože cílem hry je názorně ukázat možnosti připraveného rozhraní, hra bude primárně ovládaná pomocí pohybu a gest rukou. Bude však možné ji ovládat i pomocí klávesnice a myši. Hlavním ovládacím prvkem bude uživatelský ukazováček (v základním nastavení pravé ruky). V dalším textu už pouze jako *prst*.

6.3.2 Menu

V návrhu menu je nejzásadnější rozložení a velikost jednotlivých tlačítek a v našem případě také vizuální odezva [4]. U každého menu je zásadní, aby uživatel nevybíral nechtěně jiné položky, než měl na mysli. Ve hře ovládané prstem je však toto obzvláště nebezpečné. Jednotlivá tlačítka musí být dostatečně velká a oddělená od ostatních, aby k těmto náhodným výběrům nedocházelo.

V běžných hrách bývají položky v menu seřazeny podle používanosti od shora dolů. To odpovídá zvyklostem a také způsobu, jakým se čte. V našem případě jsem se ale rozhodl řadit položky v menu odspodu nahoru. Uživatel bude pravděpodobně sedět nebo stát před stolem, na který se promítá obraz a k bližším tlačítkům se dostane snáze. Návrh rozložení tlačítek je vidět na obrázku 6.3.

Protože rozhraní Leap Motion - projektor je unikátní ve svém použití, nabízí se otázka jaký způsob interakce s menu je nejvhodnější. Proto hra bude obsahovat dva možné způsoby, aby se následně dalo ověřit, který z nich je více uživatelsky příjemnější. Způsob ovládání si uživatel zvolí před začátkem hry a bude možné ho v průběhu změnit.



Obrázek 6.3: Návrh rozložení tlačítek menu - jednotlivá tlačítka jsou velká, jednoduchého tvaru s mezery mezi ostatními, aby nedocházelo k nechtěnému výběru. Aby měl uživatel snadnější přístup k častějším volbám, jsou řazeny odspodu nahoru.

Ukazování

Díky kalibraci je možné využít veškerý volný prostor nad obrazem projektoru. Výběr se provede najetím prstu do oblasti tlačítka v libovolné výšce nad obrazem. Následně se spustí časovač znázorněný vyplňujícím se kolečkem. Po jeho doběhnutí se potvrdí výběr tlačítka a provede se jeho přiřazená akce. V případě, že uživatel přeruší výběr přesunutím prstu, časovač se zastaví a výběr se zruší. Po aktivování tlačítka bude následovat interval jedné vteřiny, po kterou nebude možné vybrat další tlačítka, aby nedocházelo k nechtěnému zahájení výběru. Princip je znázorněn obrázkem 6.4.



Obrázek 6.4: Selektce tlačítka v menu ukazovacím způsobem - při najetí prstem do prostoru tlačítka se spustí časovač znázorněný vyplňujícím se kolečkem. Po jeho doběhnutí se potvrdí výběr.

Klikání

Tento způsob ovládání je běžný pro moderní zařízení jako jsou chytré telefony a tablety. Uživatelé jsou na tento přístup již zvyklí. Výběr se provede kliknutím na plochu tlačítka menu v rovině obraze projektoru, tedy kliknutím na stůl. Proto bude potřeba ještě před prvotním zobrazením menu tuto rovinu získat. Pro určení ovládací roviny bude muset uživatel před začátkem hry *ukázat* na tři různé promítané body, ze kterých se dá rovina vypočítat. Protože určená rovina nemusí přesně odpovídat rovině stolu, bude se jako klik brát výskyt prstu do určité vzdálenosti k naměřené rovině podle vzorce 6.3.

$$|X\rho| < 12 \quad (6.3)$$

- X je pozice konečku ovládacího *prstu* v souřadnicích ovladače Leap Motion
- ρ je naměřená projekční rovina

Položky menu

Zde je popsána struktura a možnosti hlavního menu a menu dostupného v průběhu hry. Není zde řešen způsob výběru jednotlivých prvků a jako start hry se zde počítá chvíle, kdy se zobrazí hlavní menu. Důvod opačného řazení položek je k nalezení na začátku sekce 6.3.2.

Po startu hry se zobrazí hlavní menu, které má následující strukturu a chování:

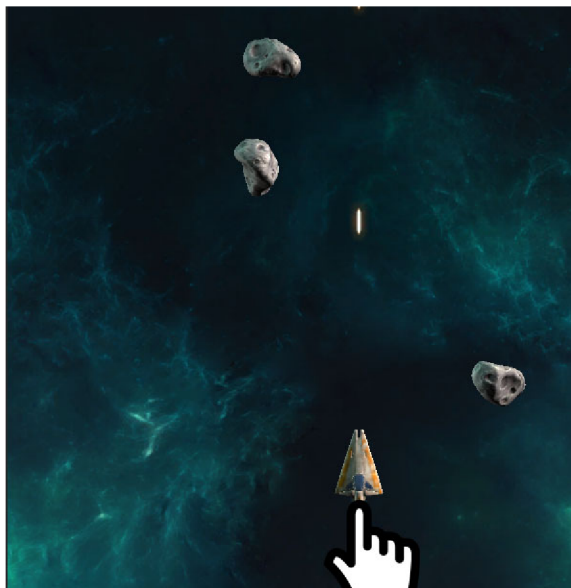
- Konec - Ukončení aplikace.
- Nastavení - Zde si uživatel bude moci zvolit ovládací ruku. V základu se bude hra ovládat pravou rukou. Dále zde bude na výběr způsob ovládání (ukazování/klikání/klávesnice+myš) a možnost znovu naměřit rovinu pro *klikání*.
- Výsledky - Zobrazí se nejlepší dosažený bodový výsledek a možnost ho vynulovat.
- Start - Tato volba spustí hru.

Po spuštění hry bude mít uživatel možnost dostat se do herního menu. To se otevře v případě pozastavení hry. Hra se pozastaví buď když ovládací ruka opustí prostor snímaný Leap Motion, nebo pokud uživatel sevře ruku v pěst. Herní menu má tuto strukturu:

- Menu - Ukončí se rozehraná hra a zobrazí se hlavní menu.
- Restart - Ukončí se rozehraná hra a pustí se nová od znova.
- Pokračovat - Spustí se přerušovaná hra.

6.3.3 Vesmírná loď

Vesmírná loď je hlavní uživatelem ovládaný prvek. Loď se bude držet před uživatelským prstem, jak je naznačeno obrázkem 6.5, a pohybuje se tedy naráz s uživatelskou rukou. Její natočení odpovídá natočení prstu vůči obrazu. V případě kontaktu lodi s ostatními objekty se loď zničí a hra končí. Výjimkou jsou bedny s náboji.



Obrázek 6.5: Návrh principu ovládání vesmírné lodi. Loď je kus před uživatelským prstem a pohybuje se naráz s ním.

Střelba

Loď, v případě že má náboje, může střílet. Střela zničí všechny ostatní objekty vyskytující se ve hře. Za všechny zničené objekty, kromě beden s náboji, se získávají body. Způsob, jakým uživatel může vystřelit, závisí na zvolené metodě ovládání menu (sekce 6.3.2).

V případě že zvolený způsob ovládání je *ukazování*, střelba se provede gestem přiložení palce k ukazováčku (obrázek 6.6). Střelba nastane ve chvíli, kdy platí vzorec 6.4, a provádí se do chvíle, kdy buďto dojdou náboje nebo uživatel gesto přeruší. Mezi jednotlivými střelami bude interval 0,4 vteřiny.

$$\angle \vec{u}_1 \vec{u}_2 < 15^\circ \quad (6.4)$$

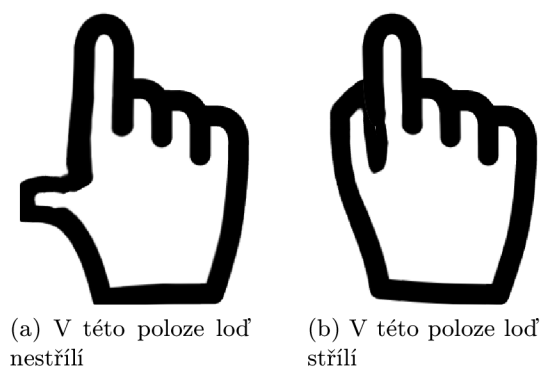
- \vec{u}_1 je směrový vektor konečku palce ovládací ruky
- \vec{u}_2 je směrový vektor konečku ukazováčku ovládací ruky

Při zvoleném módu *klikání* se střelba provede kliknutím kdekoli v obraze. *Klikání* funguje stejným způsobem, jaký již byl popsán v sekci 6.3.2. V tomto případě nebude interval mezi střelami - vystřelí se při každém kliknutí.

Náboje

Náboje se ve hře budou vyskytovat ve formě beden, které uživatel skrze vesmírnou loď sbírá. Za každou bednu obdrží 10 nábojů. Maximální množství nábojů v zásobě bude 30. Stav nábojů se bude zobrazovat

1. V levém horním rohu obrazu v číselné podobě.



Obrázek 6.6: Aktivace střelby v režimu ukazování. Podle úhlu mezi ukazováčkem a palcem uživatelské ovládací ruky se loď snaží střílet nebo ne.

2. Jako proužky promítané na uživatelský ovládací prst. Za každých 10 nábojů bude jeden proužek. Proužky budou měnit poměrně velikost a barvu podle úbytku nábojů.

Dotazníkem se vyhodnotí, který z těchto dvou způsobů je pro uživatele příjemnější. Zjistíme tím, jak na uživatele působí informativní prvky hry promítané přímo na ruku.

6.3.4 Tutoriál

Hra bude poskytovat úvodní tutoriál, ve kterém se hráč postupně seznámí ze způsoby ovládání a herními mechanismy. V průběhu tutoriálu se také nasnímají body potřebné pro určení roviny pro klikání. V případě, že se uživatel tutoriál rozhodne přeskočit, proběhne toto snímání před otevřením menu.

Kapitola 7

Implementace

V této kapitole je popis způsobu implementace kalibračního programu a demonstrační hry. Fotografie výsledné sestavy je ukázána na obrázku 7.2.

7.1 Kalibrační program

Kalibrační program byl implementován v jazyce Python¹ verze 2.7. Pro vykreslování grafického rozhraní je použita knihovna PyQt². Výpočty související s kalibrací jsou prováděny pomocí knihovny OpenCV³. Samotná kalibrace je uskutečněna metodou OpenCV *CalibrateCamera()*. Přepočítání mezi souřadnicemi Leap Motion do souřadnic obrazu projektoru zajišťuje metoda *ProjectPoints()*. Hlavní okno aplikace poskytuje možnost výběru rozlišení projektoru, počet požadovaných kalibračních cyklů a vzdálenost vykreslovaných bodů od okrajů obrazu. Podle návrhu v sekci 6.2.1 program umožňuje provést a testovat kalibraci projektoru. Kalibraci program exportuje do souboru ve formátu JSON, aby bylo možné ji použít v jiných programech. Podrobnější informace o použití kalibrační aplikace a výstupním formátu jsou k nalezení v příloze B. Na obrázku 7.1 jsou promítnuty polohy a vzájemné natočení bodů použitých při kalibraci, Leapu a projektoru určené z externích parametrů získaných kalibrací.

Třída *MainWindow* zobrazuje hlavní okno aplikace s nastavením a spuštěním kalibrace. Třída *CalibWindow* vytváří okno, ve kterém probíhá kalibrace a vykreslují se jednotlivé kalibrační body. Třída *Calibration* řídí průběh a výpočet kalibrace. Třída *MyLeap* zajišťuje komunikaci se zařízením Leap Motion a poskytuje data z něho získaná.

7.2 Demonstrační hra

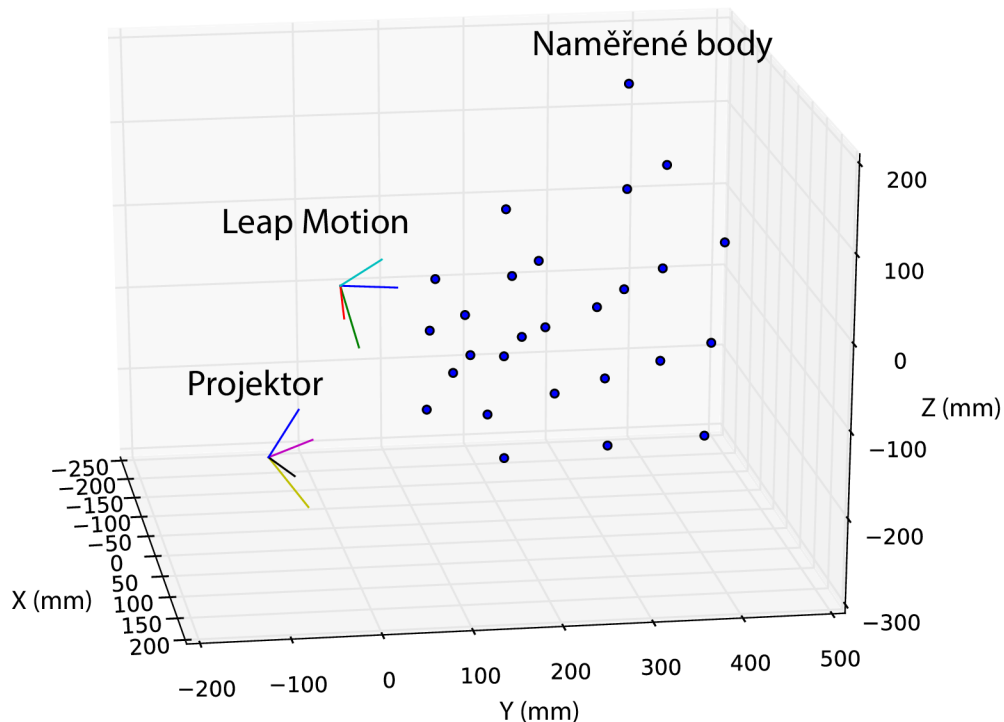
V této sekci jsou popsány podstatné části demonstrační hry. Z dostupných herních enginů jsem pro tuto práci zvolil Unity⁴ a to z několika důvodů. Unity má nejlépe dostupné návody a herní modely, dobře zpracovanou dokumentaci, pro účely práce stačí jeho bezplatná verze a také má dostupnou podporu knihovny OpenCV a Leap Motion API. Dalším vhodným kandidátem by byl také Unreal Engine, ale ten v době počátku práce nebyl ještě distribuován zdarma. CryEngine by pro projekt tohoto rozsahu byl zbytečný.

¹<https://www.python.org/>

²<http://www.riverbankcomputing.co.uk/software/pyqt/intro>

³http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

⁴<http://unity3d.com/>



Obrázek 7.1: Znázornění vzájemné polohy projektoru, Leap Motion a kalibračních bodů určené po úspěšné kalibraci. Vše je promítnuto v souřadném systému Leapu.

Hra vychází z projektu [6], ze kterého jsou použity všechny potřebné zvuky a modely. Unity dává na výběr z několika různých programovacích jazyků pro používání ve skriptech. Z nich jsem si zvolil jazyk C#. Pro výpočty transformací bodů mezi souřadnými soustavami je použita knihovna OpenCVsharp⁵. Načítání kalibračních matic ze souboru je vyřešeno pomocí knihovny SimpleJSON⁶.

Řídícím skriptem celé aplikace je *GameController*. *GameController* sleduje a přepíná stav a kola hry, generuje nepřátelské objekty a sleduje stav skóre a nábojů hráče.

Hráč (loď) je řízen skriptem *PlayerController*. Skript určuje pohyb lodi, řídí střelbu a volá pozastavení hry podle vybraného způsobu ovládání a přítomnosti rukou v obraze. Přepočítává také polohu prstu v obraze do světových souřadnic aplikace. Pohyb lodi je omezen okraji obrazu. Vyjede-li uživatel rukou ven z obrazu, ale ruka je pořád snímána zařízením Leap Motion, loď zůstane v místě obrazu nejbližší k ruce.

Data z Leap Motion získává a zpracovává skript *LeapController*. Data jsou získávána periodicky v metodě *Update*. Tato metoda je v Unity volána při každém vykreslování snímku. *LeapController* přepočítává souřadnice získávané Leap Motion do souřadnic v obraze a poskytuje tato data ostatním skriptům.

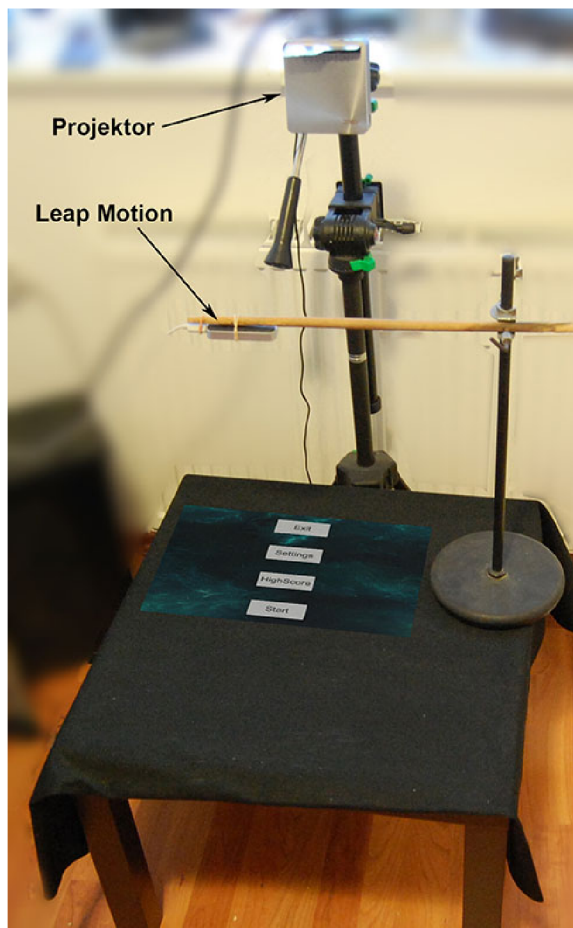
Menu, respektive jeho tlačítka, je vytvořeno UI prvky Unity - Button. Reakce menu

⁵<https://github.com/shimat/opencvsharp>

⁶<https://github.com/simplejson/simplejson>

zajišťuje skript *MenuController*. Aby tlačítka bylo možné ovládat pomocí prstů a nejen pomocí myši a klávesnice, každé tlačítko je ovládáno skriptem *LeapButtonController*. Tento skript podle zvoleného ovládání zjišťuje, zda-li na tlačítko bylo kliknuto nebo jestli na něj uživatel ukazuje a měl by se spustit časovač.

Tutoriál je řízen skriptem se shodným jménem *Tutorial*, který přepíná jednotlivé naučné scény podle uživatelových reakcí.



Obrázek 7.2: Výsledná sestava použitá při experimentování. Projektor je připevněný na stativu. Leap Motion je upevněný pomocí gumičky na dřevěné tyčce držící na stojanu od lampičky. Černé plátno umožňuje korektní rozpoznávání rukou. Prostor, ve kterém se dá se hrou interagovat, je mezi stolem a Leapem. Do stran je omezený viditelností Leap Motion.

Kapitola 8

Experimenty

V této kapitole je nejprve popsán průběh a vyhodnocení experimentů prováděných za účelem zjištění přesnosti kalibrace. V druhé sekci je vyhodnocení dotazníku pokládaného uživatelům, kteří rozhraní a hru vyzkoušeli.

8.1 Vyhodnocení přesnosti kalibrace

V první sadě experimentů jsem provedl kalibraci šestkrát pro různé počty kalibračních bodů (9, 27, 45). 3D vstupní body kalibrace jsem určoval jako průměr všech souřadnic získaných z Leap Motion po dobu jedné vteřiny, kdy se sledovaný prst nepohne v žádné ose o víc jak 7 mm. Pro každou kalibraci jsem následně provedl dvakrát měření chyby reprojekce postupem, který je popsán v sekci 6.2.3. Výsledky jsou v tabulce 8.1.

Počet bodů	9	27	45
Chyba reprojekce (mm)	16,20	10,59	13,66

Tabulka 8.1: Výsledky první sady měření - průměrná hodnota chyby reprojekce podle množství bodů, ze kterých se kalibrace počítala. 3D vstupní body kalibrace byly určeny jako *průměr* všech souřadnic získaných z Leap Motion po dobu jedné vteřiny, kdy se sledovaný prst nepohnul v žádné ose o víc jak 7 mm.

Výsledná chyba se ukázala být dosti vysoká a výsledky neodpovídají předpokladu, že čím více se použije měřených bodů, tím nižší by chyba měla být. Kalibrace je velice citlivá na uživatelskou schopnost přesně ukazovat na jednotlivé promítané body. To znamená, že nikdy nedosáhneme dokonalé přesnosti. Jako hlavní problém se však ukázalo omezené fungování Leapu proti stolu, kdy při snímání prstu v okrajích obrazu projektoru vracel zkreslené hodnoty, což způsobilo výraznou chybu.

Provedl jsem proto několik úprav. Promítané kalibrační body jsem posunul více do středu obrazu, aby nedocházelo k získávání nepřesných dat. Dále jsem zmenšil hodnotu, která resetuje časovač jedné vteřiny, po kterou se snímají body z Leapu v případě pohybu prstu ze 7 mm, na hodnotu 3 mm. Pro nižší hodnoty jsem již nebyl schopen ukazovat na promítané body s dostatečně malým pohybem prstu. Poté jsem provedl další dvě sady experimentů po 5 měřeních kalibrací pro 9, 27 a 45 bodů a ke každé kalibraci jsem znovu dvakrát měřil její chybu reprojekce. Pro první sadu se použitý bod z nasnímaných souřadnic získal znovu průměrem, pro druhou sadu se určil jako medián ze získaných souřadnic.

Výsledky jsou v tabulce 8.2. Hodnoty *reproj* 1 a 2 vyjadřují první a druhou naměřenou chybu reprojekce a jsou udávány v milimetrech.

Snímání	Počet bodů					
	9		27		45	
	reproj 1	reproj 2	reproj 1	reproj 2	reproj 1	reproj 2
1.	5,74	6,33	3,28	2,97	2,69	2,73
2.	5,23	4,78	4,47	4,15	3,82	3,31
3.	4,73	5,19	6,93	6,94	4,15	3,72
4.	6,30	6,25	6,16	4,53	4,86	2,99
5.	8,53	7,67	7,63	8,02	8,53	5,48
Průměrná chyba (mm)	6,08		5,51		4,23	

(a) Průměr

Snímání	Počet bodů					
	9		27		45	
	reproj 1	reproj 2	reproj 1	reproj 2	reproj 1	reproj 2
1.	4,67	4,74	2,75	2,92	3,05	3,44
2.	4,33	4,08	2,91	2,83	2,84	3,29
3.	10,10	9,96	3,44	3,43	3,01	2,57
4.	6,35	6,25	3,46	4,53	3,91	2,65
5.	7,14	6,17	4,63	4,02	3,34	3,07
Průměrná chyba (mm)	6,38		3,49		3,12	

(b) Medián

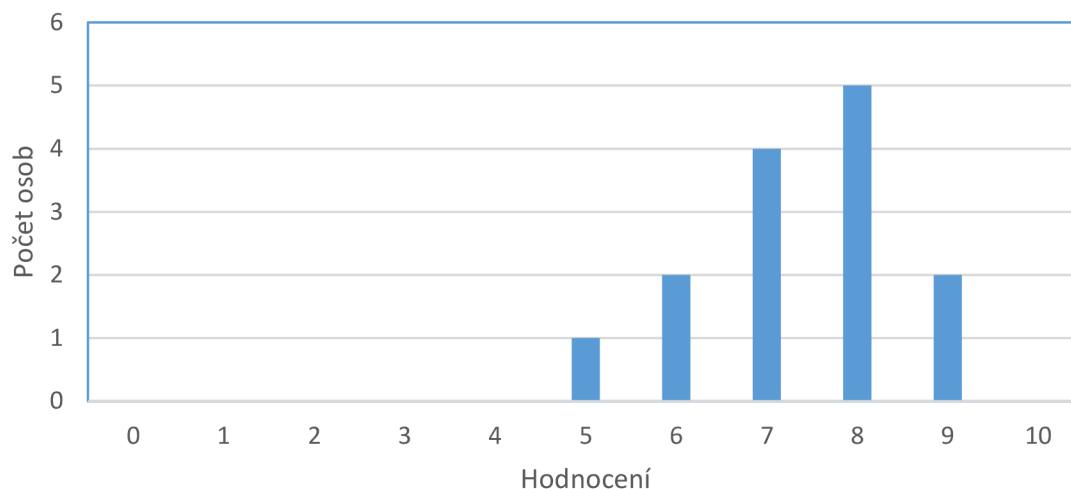
Tabulka 8.2: Výsledky druhé sady měření - průměrná hodnota chyby reprojekce podle množství bodů, ze kterých se kalibrace počítala. 3D vstupní body kalibrace byly určeny v jednom případě jako *průměr*, v druhém jako *medián* všech souřadnic získaných z Leap Motion po dobu jedné vteřiny, kdy se sledovaný prst nepohnul v žádné ose o víc jak 3 mm.

Tyto výsledky již reflektují předpoklad, že čím víc se použije vstupních bodů, tím přesnější kalibrace je. O něco lépe vyšla kalibrace při použití mediánu. To bych však nepovažoval za průkazné, protože tento druh kalibrace nepodává konzistentní výsledky a je náchylný k častým a velkým rozdílům ve výsledcích přesnosti.

8.2 Uživatelská přívětivost rozhraní a hry

Protože hra není koncipovaná tak, aby se dal změřit čas provedení nějaké činnosti, zaměřil jsem se při testování hlavně na subjektivní pocit použitelnosti rozhraní a dále na porovnání různých způsobů ovládání (klikání a ukazování viz sekce 6.3.2). Testování se zúčastnilo 14 lidí. Všichni byli seznámeni s principem a fungování zařízení Leap Motion a prošli úvodním tutorialem hry, ve kterém je vysvětleno, jak se hra ovládá. Každý z nich následně vyzkoušel ovládání hry i menu v obou možných způsobech. Poté každý vyplnil dotazník, jehož závěry zde prezentuji.

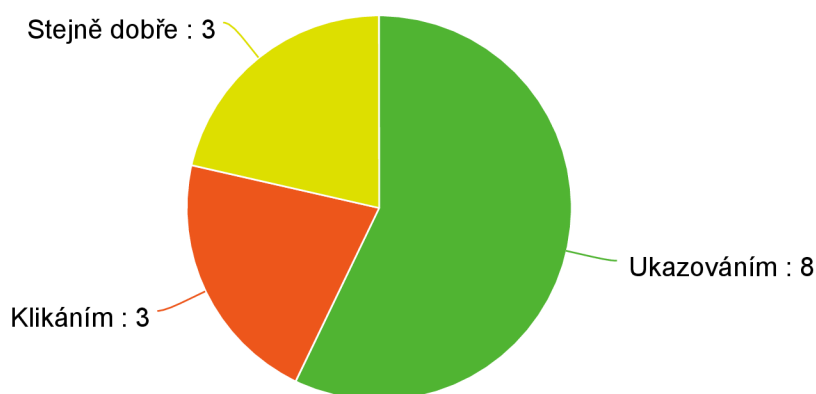
Celkový dojem ze hry a rozhraní



Obrázek 8.1: Graf s hodnocením respondentů

Celkový dojem (graf 8.1) byl hodnocen v rozmezí 0-10 bodů, kde 0 bodů je nejhorší možné hodnocení a 10 bodů je nejlepší možné hodnocení. Všichni dotazovaní hodnotili jejich celkový dojem hodnotou 5 a výš. Nejčastější hodnotou je 8 bodů. Z toho jsem usoudil, že představené rozhraní má potenciál být používáno běžným uživatelem. Hodnoty možná nedosahují vyšších čísel, protože je zapotřebí, aby uživatel chvíli trénoval, jak rozhraní reaguje a jakým způsobem se ovládá a to může kazit prvotní dojem uživatele.

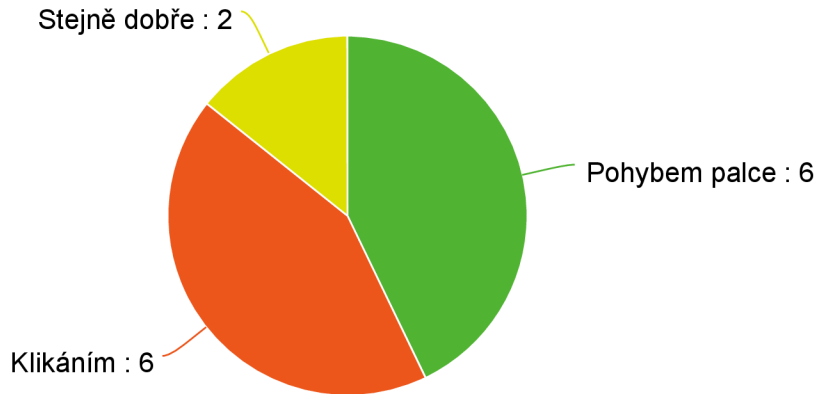
Kterým způsobem se vám lépe ovládalo menu?



Obrázek 8.2: Graf s hodnocením respondentů

V hodnocení, kterým ovládacím způsobem se lépe ovládá menu (graf 8.2), vyšel výrazně lépe ukazovací způsob a to i přestože tento způsob ovládnutí je neobvyklý a o něco pomalejší než klikání. Hlavním problémem pro respondenty bylo, že při kliknutí častěji nedocházelo k požadovanému výběru.

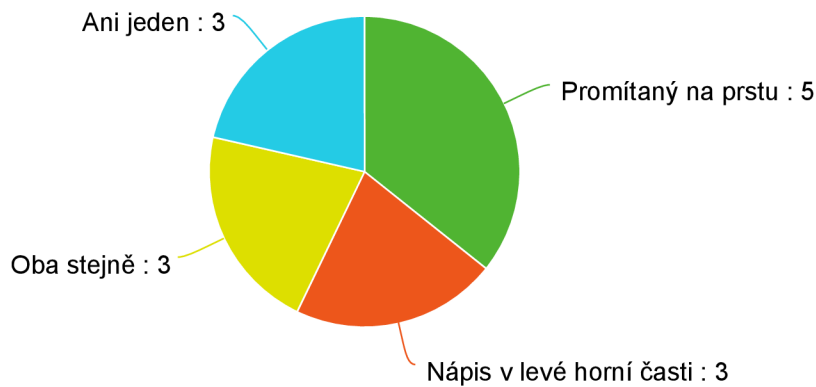
Kterým způsobem se vám lépe střílelo?



Obrázek 8.3: Graf s hodnocením respondentů

Jak je vidět v grafu 8.3, hodnocení způsobů střelby dopadlo nerozhodně. V případě střelby prováděné pohybem palce k ukazováčku docházelo k nechtěnému provedení střelby z důvodů nezvyku uživatelů na tento pohyb a také kvůli nezanedbatelné fyzické náročnosti tohoto gesta. Nechtěná střelba nastávala také v případech, kdy uživatel natočil ruku a Leap Motion vyhodnotil špatným způsobem polohu prstů. Při klikání byl naopak problém, že střelba se občas neprovedla, i když měla.

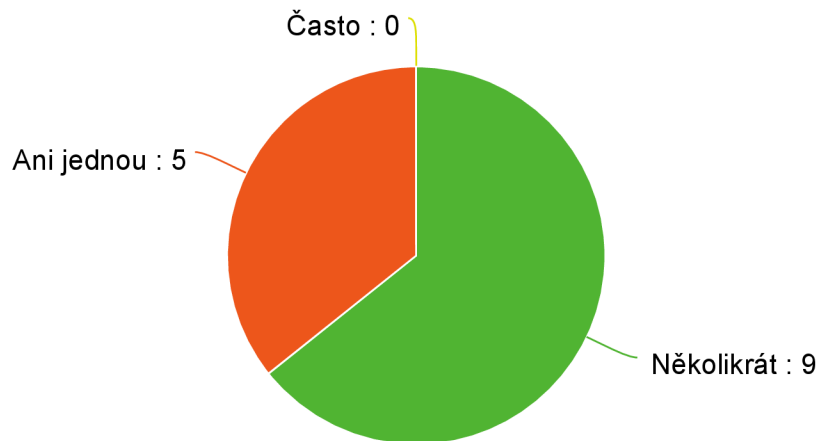
Který zásobník nábojů jste více sledoval?



Obrázek 8.4: Graf s hodnocením respondentů

Graf 8.4 ukazuje výsledky hodnocení, které bylo prováděno za účelem zjištění, jak na uživatele působí informativní prvky hry promítané přímo na jeho ruku. Výsledky jsou značně nerozhodné, s malou převahou uživatelé sledovali více zásobník promítaný na prst místo klasické varianty.

Docházelo k neúmyslnému pozastavování hry?



Obrázek 8.5: Graf s hodnocením respondentů

Značné části uživatelů (graf 8.5) se stalo, že se hra necíleně přerušila. Nejčastěji toto bylo způsobené tím, že nevěděli, jak velký je prostor, kde Leap Motion ještě snímá ruku, a tak se s rukou dostali mimo tento prostor.

8.2.1 Závěry testování

Z testování a hodnocení vyplývá, že k bezproblémovému používání rozhraní je důležité se s ním důkladně seznámit a vyzkoušet si jeho reakce. Hlavně ze začátku měli uživatelé problém s jeho ovládáním. V případě, že by se povedlo zlepšit reakce rozhraní na kliknutí, patrně by tento způsob byl pro uživatele příjemnější. Promítání informativních herních prvků na ruku se ukázalo jako použitelné, z výsledků však není jasné, zda by tento způsob mohl překonat obvyklejší nápisy a jiné prvky promítané do obrazu staticky.

Kapitola 9

Závěr

Hlavním cílem této práce bylo navrhnout a implementovat rozhraní pro hry s dataprojektorem a Leap Motion a pro toto rozhraní dále vytvořit hru, která bude demonstrovat možnosti uvedeného rozhraní. Úspěšně jsem vytvořil a v práci popsal experimentální sestavu rozhraní, způsob její kalibrace a demonstrační hru.

Pro lepší pochopení toho, jak se dá pracovat se zařízením Leap Motion, je v práci poskytnut přehled informací o Leap Motion a o API, které je pro něj dostupné. Dále se práce zabývá srovnáním dostupných herních engineů a kalibrací kamery a projektoru. Kalibrace je v práci popsána, protože je to metoda, kterou jsem použil pro uvedené rozhraní.

V návrhu kalibrace je vysvětleno, jakým způsobem se provádí - ukazováním na body promítané v obraze projektoru - a jakým způsobem se vyhodnocuje její přesnost. Přesnost navržené kalibrace je daleko nižší než například přesnost kalibračních metod typických pro sestavy kamera - projektor. To je způsobené tím, že je velice závislá na uživatelské schopnosti přesně ukazovat na promítané kalibrační body. Dalším problémem jsou souřadnice získané ze zařízení Leap Motion, které při pohybu ruky v okrajových oblastech projekce vrací zkreslené údaje o poloze prstu. Nejmenší dosažená chyba reprojekce se pohybuje kolem **3 mm**.

Kapitola o návrhu demonstrační hry se zabývá hlavně vzhledem a rozložením ovládacího menu a dvěma různými způsoby ovládání.

Uvedené rozhraní bylo testováno na demonstrační hře za účasti 14 různých uživatelů. Z vyhodnocení dotazníků vyplývá, že uživatelé mají z rozhraní a hry převážně dobrý dojem (**7,35 bodů z 10**), a že každý z uvedených způsobů ovládání hry má svoje klady i zápory.

Zajímavým rozšířením práce by bylo zmenšení chyby reprojekce a to například použitím robotického ramena nebo jiného způsobu, který by byl přesnější než kalibrace pomocí rukou.

Literatura

- [1] Introducing Myo Gesture Control Armband [online]. 2013 [cit. 2015-04-07].
URL <https://www.thalmic.com/en/myo/>
- [2] Leap Motion product [online]. 2014-03-09 [cit. 2015-01-09].
URL <https://www.leapmotion.com/product>
- [3] Leap Motion SDK and Plugin Documentation [online]. 2014-03-09 [cit. 2015-01-09].
URL <https://developer.leapmotion.com/documentation/>
- [4] Menu Design Guidelines [online]. 2014-03-09 [cit. 2015-04-02].
URL https://developer.leapmotion.com/documentation/csharp/practices/Leap_Menu_Design_Guidelines.htm
- [5] How Does the Leap Motion Controller Work? [online]. 2014-08-09 [cit. 2015-03-29].
URL <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>
- [6] Project: Space Shooter [online]. 2015-27-4 [cit. 2015-04-07].
URL <http://unity3d.com/learn/tutorials/projects/space-shooter>
- [7] Kinect for Windows features [online]. 2015 [cit. 2015-04-07].
URL <http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx>
- [8] Gabriel Falcao, J. M., Natalia Hurtos: Plane-based calibration of a projector-camera system. 2008.
URL https://procamcalib.googlecode.com/files/ProCam_Calib_v2.pdf
- [9] Hartley, R.; Zisserman, A.: *Multiple View Geometry in Computer Vision*. New York, NY, USA: Cambridge University Press, druhé vydání, 2003, ISBN 0521540518.
- [10] Jože Guna, M. P. S. T., Grega Jakus; Sodnik, J.: An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors*, ročník 14, 2014, ISSN 1424-8220.
URL <http://www.mdpi.com/1424-8220/14/2/3702>
- [11] Moreno, D.; Taubin, G.: Simple, Accurate, and Robust Projector-Camera Calibration. In *Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 3DIMPVT '12*, Washington, DC, USA: IEEE Computer Society, 2012, ISBN 978-0-7695-4873-9, s. 464–471, doi:10.1109/3DIMPVT.2012.77.
URL <http://dx.doi.org/10.1109/3DIMPVT.2012.77>

- [12] Nintendo: What is Wii? [online]. 2007 [cit. 2015-04-07], [Online; accessed 22-April-2015].
URL <http://web.archive.org/web/20080212080618/http://wii.nintendo.com/controller.jsp>
- [13] Tsai, R. Y.: Radiometry. kapitola A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses, USA: Jones and Bartlett Publishers, Inc., 1992, ISBN 0-86720-294-7, s. 221–244.
URL <http://dl.acm.org/citation.cfm?id=136913.136938>
- [14] Weichert, F.; Bachmann, D.; Rudak, B.; aj.: Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors*, ročník 13, č. 5, 2013: s. 6380–6393, ISSN 1424-8220, doi:10.3390/s130506380.
URL <http://www.mdpi.com/1424-8220/13/5/6380>
- [15] Wikipedia: Game engine — Wikipedia, The Free Encyclopedia [online]. 2015 [cit. 2015-04-020].
URL http://en.wikipedia.org/w/index.php?title=Game_engine&oldid=655158910
- [16] Zahrádka, J.: ROZŠÍŘENÉ UŽIVATELSKÉ ROZHRAŇÍ. 2011.
URL <http://www.fit.vutbr.cz/study/DP/DP.php?id=11321&file=t>
- [17] Zhang, Z.: A Flexible New Technique for Camera Calibration. 1998.
URL <http://research.microsoft.com/en-us/um/people/zhang/Papers/TR98-71.pdf>

Příloha A

Obsah CD

- text práce ve formátu PDF a zdrojové soubory pro prostředí L^AT_EX
- zdrojové soubory kalibrační aplikace a projekt hry pro prostředí Unity
- kalibrační aplikace a hra sestavené pro platformu Windows
- video ukázka průběhu kalibrace a předvedení demonstrační hry
- ilustrativní plakát reprezentující výsledky zadání

Příloha B

Kalibrační aplikace

Zde je popsáno, co vše je potřeba pro spuštění kalibračního programu, jakým způsobem se ovládá a dále je uvedena tabulka experimentů měření přesnosti kalibrace.

B.1 Potřebný software

Pro spuštění programu je potřeba:

- Python 2
- knihovna numpy ¹
- knihovna PyQt4
- knihovna OpenCV (cv2)
- knihovna pro Leap přiložená s programem

B.2 Ovládání programu

Po spuštění programu se spustí jeho hlavní okno ve kterém jsou všechny volby aplikace.

Popis modifikovatelných parametrů:

- Width - šířka promítaného obrazu projektoru v pixelech
- Height - výška promítaného obrazu projektoru v pixelech
- Cycles - počet požadovaných kalibračních běhů
- Shift - posun vykreslovaných bodů od okrajů okna v pixelech
- Head mount optmization - nastavuje zda se Leap Motion používá vzhůru nohama nebo ne

Popis funkcí

- Calibrate - vyvolá výběr souboru do kterého se kalibrace uloží. Popis formátu souboru je popsán v sekci **B.3**. Následně se otevře samotné kalibrační okno. Po dokončení kalibrace se spustí fáze testování kalibrace.

¹<http://www.numpy.org/>

- Load - vyvolá výběr souboru s uloženou kalibrací. Pro tuto kalibraci se spustí fáze testování kalibrace.
- Reprojection - vyvolá výběr souboru kalibrace pro které chceme spočítat chybu reprojekce. Po té proběhne snímání bodů a výpočet chyby.
- ReprojLoad - umožní zobrazení výpočtu dříve spočítané reprojekční chyby.

B.3 Výstupní formát

Výstup je ukládán ve formátu JSON. Obsahuje parametry:

- worldPoints - poloha nasnímaných bodů ve 3D souřadném systému Leap Motion
- imagePoints - poloha nasnímaných v pixelech obrazu projektoru
- tvecs - vektor posunu
- rvecs - vektor rotací
- dist - vektor koeficientů zkreslení
- mtx - camera matrix neboli matice vnitřních parametrů
- height - výška obrazu projektoru v pixelech
- width - šířka obrazu projektoru v pixelech

Příloha C

Manuál demonstrační hry

Zde je popsáno co vše je potřeba pro překlad a spuštění demonstrační hry.

C.1 Potřebný software

Pro překlad je potřebný engine Unity verze 4.6 nebo vyš. Pro správnou funkčnost hry v editoru Unity je nutné zkopírovat soubor získaný z kalibrační aplikace do složky Assets. Pro správný běh samotné hry je nutné tento soubor zkopírovat do složky přeložené hry *zvolenyNazevHry_Data*. Dále je potřeba mít v systémové proměnné PATH dll soubory knihovny OpenCV nebo tyto soubory zkopírovat ze zdrojové složky hry do složky sestavené hry. Nakonec je potřeba do složky sestavené hry nakopírovat také Leap.dll a LeapCSSharp.dll.