

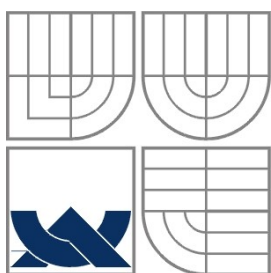
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií
Faculty of Information Technology

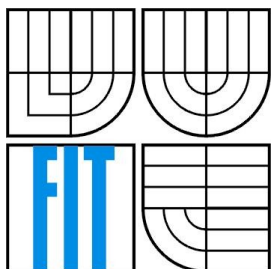
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

Brno, 2016

Bc. Vojtěch Kaisler



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

PROMÍTANÉ UŽIVATELSKÉ ROZHRANÍ – DESKOVÁ HRA

PROJECTED USER INREFACE – BOARD GAME

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Vojtěch Kaisler

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Najman

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání diplomové práce

Řešitel: **Kaisler Vojtěch, Bc.**

Obor: Počítačová grafika a multimédia

Téma: **Promítané uživatelské rozhraní - desková hra**
Projected User Interface - Board Game

Kategorie: Zpracování obrazu

Pokyny:

1. Seznamte se s metodami 2D/3D geometrie využívanými v oblasti počítačového vidění, zaměřte se na problematiku kalibrace projektorů a kamer.
2. Prostudujte metody a knihovny, které se v současné době používají k detekci rukou a prstů.
3. Navrhněte počítačovou hru (simulaci deskové hry), která bude využívat dataprojektor k promítání herního plánu na stůl. Hráči sedící okolo stolu budou moci interagovat s hrou pomocí rukou nebo přiložených nástrojů.
4. Implementujte navrženou hru.
5. Otestujte navrženou hru a vyhodnoťte její vlastnosti. Vytvořte ilustrativní plakát reprezentující výsledky zadání.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- První tři body zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

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

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

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

Vedoucí: **Najman Pavel, Ing., UPGM FIT VUT**

Datum zadání: 1. listopadu 2015

Datum odevzdání: 25. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, L.S. Technická 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá návrhem a realizací interaktivní stolní deskové hry. Tuto realizaci je možné rozdělit na detekci a projekci. Detekce prováděná infračervenou kamerou slouží ke snímání reakcí hráčů, kteří interagují se hrou pomocí jednoduchých gest rukou. Projekce promítá na stůl herní plán s figurkami pomocí dataprojektoru. Zvolenou promítanou hrou je stolní hra na hrdiny Dungeons & Dragons.

Abstract

The thesis brings a project and an implementation of an interactive board game. The implementation can be divided into a detection part and a projection part. Detection, realized by an infra-red camera, serves as a means for recording players' responses. The players interact with the game through simple hand gestures. Projection screens a game plan, which includes the players' figures, on the table with the help of a data projector. The game chosen for projection is a role-playing game Dungeons & Dragons.

Klíčová slova

Zpracování obrazu, počítačové vidění, OpenCV, Dungeons and Dragons, uživatelské rozhraní, infračervená kamera, detekce rukou, kalibrace

Keywords

Image segmentation, computer vision, OpenCV, Dungeons and Dragons, user interface, infrared camera, hand detection, calibration

Citace

Kaisler Vojtěch: Promítané uživatelské rozhraní – desková hra, diplomová práce, Brno, FIT VUT v Brně, 2016

Promítané uživatelské rozhraní – Desková hra

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pánů Ing. Jiřího Zahrádky a Ing. Pavla Najmana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Vojtěch Kaisler
25.5.2016

Poděkování

Chtěl bych poděkovat pánům Ing. Jiřímu Zahrádkovi a Ing. Pavlu Najmanovi za poskytnutí odborné pomoci při řešení této diplomové práce a tvorbě technické zprávy. Dále bych chtěl poděkovat panu prof. Pavlu Zemčíkovi za pomoc při zapůjčení infračervené kamery. Nakonec bych chtěl poděkovat panu Aleši Matějovi za nakreslení textur pro promítané postavy.

© Vojtěch Kaisler, 2016

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

Obsah

Obsah.....	1
1 Úvod.....	2
2 Existující interaktivní aplikace.....	3
2.1 Promítání na stěnu a zem.....	3
2.2 Interaktivní stůl.....	4
3 Kamera.....	5
3.1 Model kamery.....	5
3.2 Kalibrace kamery vůči dataprojektoru.....	6
3.3 Infračervená oblast spektra.....	7
4 Detekce ruky a prstů.....	9
4.1 Detekce.....	9
4.2 Segmentace.....	9
4.3 Klasifikace.....	10
4.4 Sledování.....	11
4.5 Knihovny a nástroje pro detekci rukou a prstů.....	11
5 Promítaná hra.....	14
5.1 Dungeons & Dragons.....	14
5.2 Specifikace.....	15
6 Návrh.....	16
6.1 Snímání a detekce gest rukou.....	16
6.2 Jádru aplikace.....	18
6.3 Zobrazované rozhraní aplikace.....	19
7 Implementace.....	21
7.1 Sestava.....	21
7.2 Detekce.....	22
7.3 Kontrolér.....	24
7.4 Projekce.....	27
7.5 Použité nástroje.....	30
7.6 Problémy při implementaci.....	31
8 Výsledky a testování.....	33
8.1 Detekované gesto.....	33
8.2 Testování.....	34
8.3 Uživatelské testování.....	35
8.4 Zhodnocení výsledků.....	38
9 Diskuze a budoucí práce.....	39
10 Závěr.....	40

1 Úvod

V dnešní době počítačových her je běžné, že se objevují i hry především pro mobilní zařízení, které mají předlohu v klasických deskových nebo karetních hrách. Je jednodušší mít větší množství her na přenosném zařízení, než uchovávat objemné krabice s plánky a figurkami. Přesto stále velké množství lidí vnímá deskové hry jako společenskou událost. Jenže co když právě není k dispozici hra, kterou by si chtěli zahrát všichni. Jednu z možných odpovědí je použití interaktivního stolu. Tato zařízení jsou ale většinou určena pro použití v kancelářích a tomu odpovídá i cena jejich pořízení. Jako další varianta interaktivního stolu může být pojata i aplikace prezentovaná v této diplomové práci.

V této práci je navržena a implementována aplikace, která umožňuje běžným projektorem promítat na stůl rozhraní deskové hry a zajišťuje interakci hráčů se hrou pomocí jednoduchých gest rukou, které jsou snímány infračervenou kamerou. V této práci je navrženo promítané rozhraní pro stolní hru na hrdiny *Dungeons & Dragons*.

Před návrhem a implementací samotného rozhraní je nutné projít existující aplikace, které zajišťují interaktivní projekci. Některé z těchto aplikací jsou popsány v kapitole 2. Pro zajištění interakce je třeba použít vhodnou detekci. Tato práce se zaměřuje na použití infračervené kamery, jejíž základní matematický model je popsán v kapitole 3. Ve snímcích z kamery je třeba detekovat gesta. Teoretický postup pro detekci gest je možné nalézt v kapitole 4. Po získání a klasifikaci gest je možné přejít k projekci zvolené hry. Rozboru hry na hrdiny *Dungeons & Dragons* je věnována kapitola 5. Následně je v kapitole 6 prezentován návrh rozhraní pro tuto hru. V kapitole 7 je poté popsána její implementace. Kapitola 8 popisuje testování navržené a implementované aplikace z pohledu klasifikace gest i uživatelské zkušenosti. Práci uzavírá kapitola 9 ve které jsou diskutovány výsledky testování, použitelnost implementované aplikace v praxi a navrhuje možná budoucí rozšíření.

Tato práce navazuje na semestrální projekt, jehož součástí byla teoretická příprava pro tuto diplomovou práci, tedy popis matematického modelu kamery, rozbor detekce klasifikace a sledování gest a dále návrh rozhraní pro vybranou hru.

2 Existující interaktivní aplikace

V současné době je trh s interaktivními technologiemi velmi otevřený a objevují se stále nové aplikace, které přinášejí uživatelům lepší zážitek při interakci s digitálním prostředím. Nejširší použití těchto technologií je především v zábavním průmyslu, kde nabízí velké množství interaktivních her, jednoduché reaktivní reklamy, či třeba rozšíření informací o právě konzumovaném nápoji. Při bližším pohledu na použité technologie je možné tyto rozdělit do dvou kategorií. První je promítání na stěnu či zem (kapitola 2.1), druhou je pak interaktivní stůl (kapitola 2.2).

Návrh aplikace popsaný v kapitole 6 řeší interaktivní projekci pomocí infračervené kamery. Toto řešení může být při zachování zpracování v reálném čase stejně efektivní jako níže uváděné aplikace. Díky využití kamery a dataprojektoru nebude navrhovaná aplikace vázána na dotykový panel nebo displej, ale přiblíží se otevřené projekci prvnímu typu aplikací při zachování přesnosti snímání gest druhého typu aplikací. Rozdílem oproti všem zmíněným provedením bude způsob interakce přes gesta prstů rukou.

2.1 Promítání na stěnu a zem

Do první kategorie patří aplikace, které zprostředkovávají promítání rozhraní na zem či stěnu a pokrývají tak relativně velký prostor. Tyto aplikace sledují pohyb člověka v promítaném prostoru a reagují na pohyb jeho končetin. Pohybem ruky je tam možné kupříkladu rozvlnit hladinu vody nebo nohou kopnout do míče, který se bude odrážet v projekčním prostoru. Do tohoto typu aplikací je možné zařadit například komerční řešení *GestureFX*[®] *GroundFX*[20] nebo *Po-motion*[21].

GestureFX[®] *GroundFX* patří do série interaktivních zařízení *GestureFX*. Kromě promítání na podlahu je v této sérii i promítání na zeď a interaktivní stůl. Promítání na zem i stěnu je v této aplikaci řešené pomocí dataprojektoru. Snímání pozice končetin provádí pomocí 2D nebo 3D kamer. Konkrétní použité technologie jsou patentovány.



Ilustrace 1: Příklad použití *GroundFX* aplikace, zdroj:[20]

Po-motion je dalším řešením interakce uživatelů při promítání na stěnu či zem. Jedná se čistě o softwarové řešení, přičemž je možné k projekci použít libovolný projektor či displej a libovolnou webkameru či snímáče *Kinect* pro detekci pohybu. Součástí základní verze je však použití infračervené kamery pro vytvoření hloubkové mapy, ve které se detekuje interakce s uživatelem[21]. Součástí tohoto řešení je i množství již vytvořených her a aplikací. Taktéž je zde možnost tvorby vlastních interaktivních aplikací. Konkrétní použité technologie a algoritmy nejsou zmíněny v běžně dostupné dokumentaci k tomuto nástroji.

2.2 Interaktivní stůl

Dalším typem použité technologie je interaktivní stůl. Z této kategorie je možné zmínit další komerční řešení, jako je *MagixTable*[22], *E-Table*[23] nebo *Vi-tech Multitouch Table*[24]. Tyto aplikace zobrazují projekci přímo na stůl a umožňují interakci většímu počtu lidí. Každá z těchto aplikací prezentuje jinou technologii promítání a detekce gest rukou.

MagixTable a *Vi-tech Multitouch Table* patří do skupiny řešení, která využívá velkého vícedotykového LCD displeje. Toto řešení nabízí okamžitou interakci s uživatelem. Způsob komunikace s uživatelem je stejný jako v případě běžných chytrých telefonů nebo tabletů. Vzhledem k tomu, že tato diplomová práce předpokládá použití libovolného stolu jako projekční plochy, jedná se v tomto případě o jiný typ technologie a tudíž nebude v této práci dále rozebírána.

E-Table je další aplikací technologie pro interaktivní stůl určená pro zlepšení zážitků z restaurací. Smyslem tohoto řešení je umožnit zákazníkům restaurací interaktivní způsob výběru pokrmů, pití či dalších služeb.



Ilustrace 2: Ukázka interaktivního stolu *E-Table*, zdroj:[27]

Interaktivita u této aplikace je řešena malým dotykovým panelem, který je možné vidět v pravé dolní části na ilustraci č. 2. Pomocí tohoto panelu je ovládáno jednoduché menu, skrz které je celá aplikace ovládána. Nejedná se tedy v tomto případě o plnohodnotný interaktivní stůl, ale spíše o rozhraní k promítanému obrazu.

3 Kamera

V této kapitole je popsán základní model kamery z teoretického a matematického pohledu. Dále je zde rozepsán teoretický přístup ke kalibraci kamery a sjednocení promítacího prostoru s dataprojektorem. Poslední část této kapitoly je věnována infračervené části spektra a možnosti jeho využití pro detekci rukou.

Základním principem kamery je mapování reálného 3D světa do plošného 2D snímku. Způsob tohoto mapování je vázán na použitý model kamery. Základem pro modelování kamery je tzv. *centrální projekce*. Tedy že paprsky ze snímané scény se protínají v jednom jediném bodě, v centru kamery. Rovina snímku je potom určena parametry kamery.

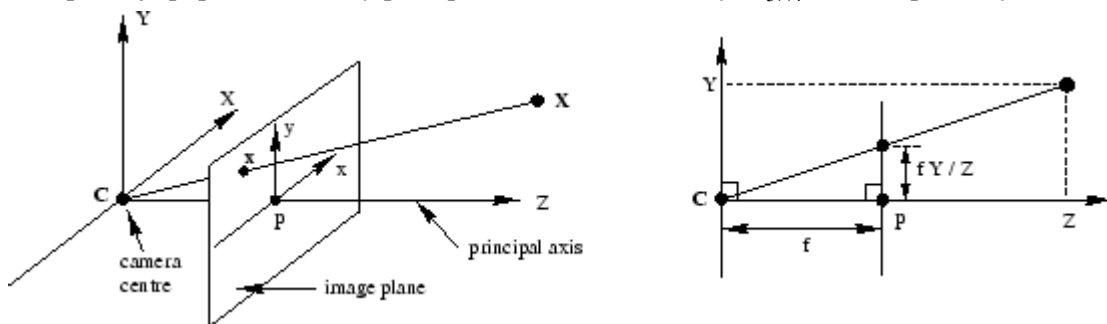
Za použití projektivní geometrie můžeme matematicky každý bod snímané scény zapsat v homogenních souřadnicích jako vektor $(X, Y, Z, T)^T$ a body na rovině snímku jako $(x, y, z)^T$. Vztah mezi těmito body určují parametry kamery a můžeme napsat:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = P_{3 \times 4} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix} \quad (1)$$

Kde $P_{3 \times 4}$ je označována jako matice kamery [1]. Následuje popis této matice, která je definována modelem kamery.

3.1 Model kamery

V této kapitole je popsán teoretický postup získání matice kamery $P_{3 \times 4}$. Tato kapitola vychází z [1]



Ilustrace 3: Model dírkové komory [1]

Vycházíme z modelu *camera obscura*, tedy dírková komora. Uvažujme, že střed projekce je zároveň počátkem eukleidovského prostoru. Dále uvažujme projekční rovinu na souřadnici $Z=f$. V tomto modelu je možné bod $\mathbf{x} = (X, Y, Z)^T$ mapovat na projekční rovinu

$(X, Y, Z)^T \Rightarrow (f X/Z, f Y/Z)^T$. Tím dosáhneme mapování 3D prostoru na 2D. Při popisu pomocí homogenních souřadnic dostáváme následující zápis:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} f X \\ f Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} f & & 0 \\ & f & 0 \\ & & f & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

Tento jednoduchý zápis předpokládá, že počátek souřadnic v projekční rovině je shodný s počátkem souřadnic celého snímaného prostoru. To lze předpokládat u dírkové komory, obecně to ale platit nemusí. Z toho důvodu je nutné výše uvedenou matici $P_{3 \times 4}$ upravit o posunutí počátečního bodu na souřadnicích $(p_x, p_y)^T$ na zápis:

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & f \end{bmatrix} \quad (3)$$

Tato matice společně s nulovým 3-rozměrným vektorem tvoří matici $P_{3 \times 4}$. Pro obecnější popis projekce bodu snímaného prostoru do projekční roviny musíme uvažovat, že souřadný systém kamery a snímaného prostoru mohou být obecně různé. Musíme tudíž počítat s posunutím a rotací při popisu vztahu mezi zmíněnými souřadnými systémy. Předpokládejme tedy, že bod X_{world} patří do snímaného prostoru a je zapsán 3-rozměrným nehomogenním vektorem a X_{cam} reprezentuje ten samý bod v souřadném systému kamery, pak můžeme zapsat $X_{cam} = R(X_{world} - C)$ kde C je reprezentace středu kamery v souřadnicích snímaného prostoru a $R_{3 \times 3}$ označuje matici vztahu mezi souřadnými systémy. V homogenních souřadnicích lze toto popsat jako:

$$X_{cam} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} X \quad (4)$$

Tím získáváme všechny parametry pro popis kamery a výslednou rovnici:

$$x = K R [I | -C] X \quad (5)$$

V této rovnici jsou popsány všechny parametry, které se týkají obecného popisu modelu dírkové komory. Parametry v matici K jsou tzv. *interní parametry kamery*. Matice R a C reprezentují orientaci a pozici kamery v souřadném systému snímaného prostoru, jedná se o tzv. *externí parametry kamery*. Externí parametry kamery budou pro tuto práci důležité, neboť jsou součástí kalibrace kamery vůči projektoru. V rámci níže popsané kalibrace se budeme zabývat především zjištěním pozice a natočením kamery vůči snímanému (projekčnímu) prostoru.

3.2 Kalibrace kamery vůči dataprojektoru

Z pohledu této práce se jedná o nalezení podobnosti mezi projekčním prostorem projektoru a snímaným prostorem kamery. Jak byl zmíněno výše, jedná se o nalezení středu kamery C a transformační matice R . Z pohledu kamery se jedná o nalezení homografie mezi bodem projekčního prostoru a bodem na snímku.

Nalezení homografie mezi těmito dvěma prostory je možné provést pomocí kalibrační roviny. Zde je možné využít skutečnosti, že projektor promítá na rovinu stolu. Díky projekci bodů, případně

čar na takovou plochu a následnému sledování projekce těchto čar je možné zjednodušit homografii do formy matice 3×3 , která je závislá na pozici kamery vůči projekční ploše. Homografie pak může být určena pomocí 4 a více bodů nebo čar[2]. Vzhledem k tomu, že je znám promítaný obraz, jedná se o nalezení podobnosti se snímaným obrazem.

Z matematického pohledu se jedná o nalezení perspektivní transformace H mezi významnými body v obou obrazech[1]

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = H_{3 \times 3} \begin{pmatrix} X \\ Y \\ T \end{pmatrix}. \quad (6)$$

Vzhledem k tomu, že transformační matice H je shodná s transformační maticí R a z pozice odpovídajících bodů jsme schopni určit pozici středu kamery C , je kalibrace kamery vůči projekčnímu prostoru dokončena.

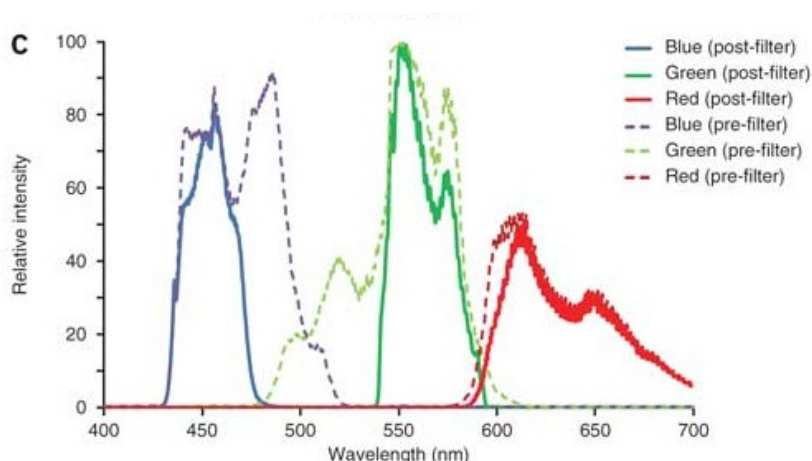
RANSAC

K nalezení transformační matice H lze použít algoritmus RANSAC. Plný název této metody zní RANdom Sample Consensus, česky Shoda náhodných vzorků, původně publikována pány Fischlerem a Bollesem v roce 1981.

Jedná se o iterativní metodu, která hledá v množině vstupních bodů model. Iterace se provádí dokud není dosaženo maximálního počtu iterací, nebo míra shody nalzeného řešení není dostatečná. Tento algoritmus je používán v mnoha odvětvích počítačového vidění při hledání geometrických objektů, homografie, případně korespondencí v obraze. [29]

3.3 Infračervená oblast spektra

Lidské oko vnímá záření o vlnových délkách mezi 380 nm a 780 nm [18]. U dataprojektorů lze předpokládat, že použité vlnové délky budou pokrývat tento rozsah. V závislosti na použité technologii bude rozdílné pokrytí. Ilustrace 4 ukazuje pokrytí vlnových délek pro LCD projektor.



Ilustrace 4: Spektrum pokrytí RGB LCD projektoru, zdroj:[17]

Pro snímání objektů v projekčním poli projektoru je proto vhodné zvolit jinou část spektra, než ve které projektor promítá. V případě použití běžné kamery by byla detekce rukou problematická, neboť by promítaný obraz výrazně ovlivňoval jak barvu kůže, tak i tvary detekovaných objektů.

Jednou z možností je proto zvolit infračervenou oblast v rozmezí vlnových délek mezi 800 nm a 1200 nm, tedy oblast blízkého infračerveného záření. V této frekvenční oblasti není možné sledovat teplotu lidského těla, tedy přímo detekovat končetiny. Při použití osvětlovacích LED s vhodnou frekvencí je ale možné tyto kamery použít pro detekci osvětlených objektů.

Infračervené kamery z této frekvenční oblasti se běžně používají jako bezpečnostní kamery, fotopasti nebo jako policejní radary. Použitím těchto kamer se zjednoduší samotná detekce rukou ve snímacím prostoru kamery, neboť promítaná data leží mimo rozlišovací schopnost a tudíž nebudou rušivým elementem ve zpracování obrazu.

4 Detekce ruky a prstů

Tato kapitola se zabývá identifikací rukou ve snímaném obraze. V první části je popsáno předzpracování obrazu, tedy nalezení oblasti, ve které je možné gesto určit. Další část se věnuje analýze nalezené oblasti a jejímu zpracování před klasifikací gesta, která je podrobně popsána v kapitole 4.3. V poslední podkapitole je uveden jednoduchý princip sledování pohybu ruky v sekvenci snímků.

Samostatná 4.5 podkapitola se poté věnuje knihovnám a nástrojům, které se běžně používají k detekci, segmentaci a klasifikaci gesta. Dále jsou zde uvedeny některé projekty, které tuto problematiku řeší.

4.1 Detekce

Detekci se v tomto případě rozumí rozlišení pozadí od detekovaného objektu, v tomto případě ruky. To je možné provést několika způsoby. Jednou z možností je využít *thresholding*, neboli detekci objektů podle intenzity jednotlivých pixelů. Největším problémem této metody je správné nastavení rozlišovací hodnoty. To je možné provést nastavením statické hodnoty, nebo použitím sofistikovanějších algoritmů pro určení vhodného prahu, např. Otsu.

Další možností je odečtení pozadí (*background subtraction*). Tuto metodu lze použít tehdy, když je změna v pozadí mnohem menší než změny v popředí. Jednou z metod, které provádí odečtení pozadí je *Gaussian mixture-based* pro segmentaci pozadí nebo popředí, představená v [3]. Taktéž je možné použít adaptivnější verzi této metody [4]. Další metodou je statistický odhad a *per-pixel* segmentace. Tato metoda hledá objekty, které leží s určitou pravděpodobností v popředí[5].

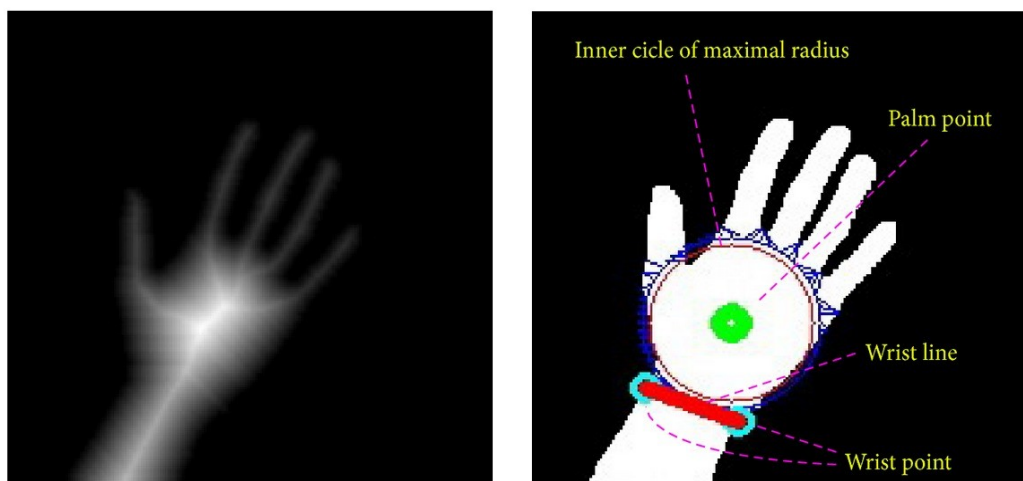
Detekci je možné provádět i podle barvy kůže. Ta se určí nejčastěji experimentálně. Metody založené na detekci podle barev se většinou vyhýbají RGB barevnému prostoru, protože ten je náchylný na změny intenzity a osvětlení. Z toho důvodu je vhodnější používat jiné barevné modely, například HSV [7]. Tato metoda je tedy vhodná pro použití v laboratorních podmínkách, kdy můžeme zajistit neměnnou hodnotu osvětlení a intenzity.

4.2 Segmentace

Samostatná detekce je pouze prvním krokem ke zjištění gesta, kterým se uživatel snaží s počítačem komunikovat. Pokud jsme úspěšně identifikovali část obrazu, která odpovídá ruce, je nyní nutné provést segmentaci pěsti a prstů. V této části budeme předpokládat, že se jedná o obecnou úlohu nalezení gesta ve snímku bez předchozí identifikace přesné polohy, ale již s odstraněným pozadím. Smyslem segmentace je určení pozice středu dlaně a identifikace jednotlivých prstů, které se účastní příslušného gesta. K tomuto účelu lze využít několik metod.

Jednou z nich je určení kontur ruky a nalezení konvexní obálky. Použitím hranového filtru jsou identifikovány body ležící na rozhraní mezi oblastmi s výplní. Tím získáme množinu bodů reprezentující kontury. Z těchto kontur je poté možné určit konvexní a konkávní body. Díky těmto bodům je možné určit pozici středu dlaně, případně identifikovat jednotlivé prsty[6]. Po identifikaci kontur je možné velikost dlaně normalizovat a jednotně otočit pro další klasifikaci.

Další možná metoda jak detekovat střed dlaně je použití vzdálenostní transformace, neboli vzdálenostní mapy[8]. Touto transformací získáme mapu ve které identifikujeme střed dlaně jako nejintenzivnější bod.

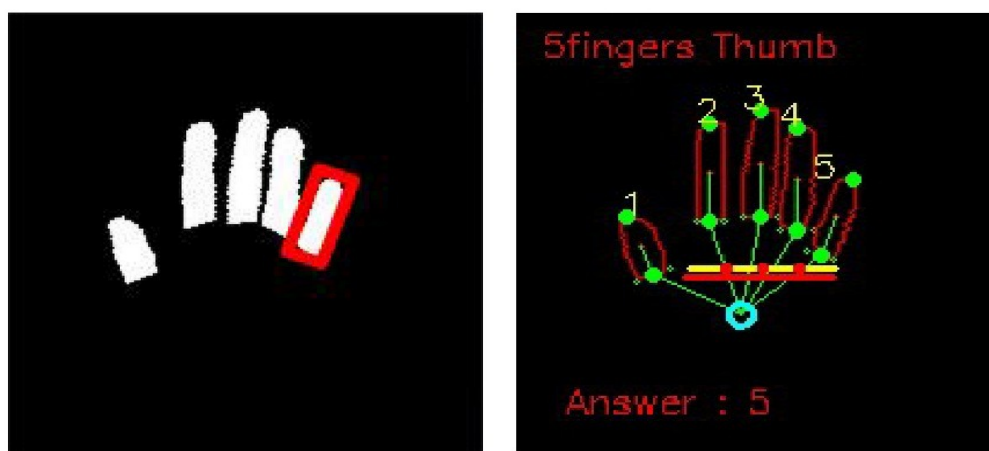


Ilustrace 5: Vzdálenostní transformace a nalezení středu dlaně [8]

Po nalezení středu dlaně je dále možné určit tvar dlaně bez prstů nalezením minimální kružnice se středem v nalezeném bodě, která je ve dlani vepsaná. Tím se identifikují body mezi prsty a zúžení na zápěstí. Díky známé pozici zápěstí a velikosti dlaně můžeme celý obraz normalizovat a otočit pro další klasifikaci.

4.3 Klasifikace

Klasifikací v rámci této práce rozumíme přiřazení detekované dlaně k příslušnému gestu. Vzhledem k tomu, že se v tomto případě nejedná o obecný případ určení typu objektu ve snímku, budou v této kapitole popsány pouze jednoduché metody pro rozpoznání třídy na základě jednoduchého příznakového vektoru. Ten získáme analýzou kontury nalezené ruky nebo, jak bude popsáno dále, analýzou snímku při zajištění normalizované velikosti a natočení[8].



Ilustrace 6: a) Oddělené prsty s obálkou, b) Klasifikace pozice prstů, zdroj: [8]

Uvažujme tedy snímek dlaně, který má jednotnou velikost pro všechny případy a přibližně stejné natočení. To můžeme předpokládat na základě použití poslední metody segmentace zmíněné v předchozí kapitole. Taktéž známe střed dlaně a poloměr v ní vepsané kružnice. Na základě těchto informací je možné získat samostatně oddělené prsty (viz. Ilustrace 6).

Z takto upraveného snímku můžeme pomocí minimálního obalu (*minimal bounding box*) získat informaci o pozici a šířce prstu, respektive jeho obálky. Nyní můžeme přistoupit k rozpoznání jednotlivých prstů. Ze znalosti středu dlaně a středů obálek prstů nejprve určíme, zda se v analyzovaném gestu nachází palec a to jednoduchým vytyčením úhlového prostoru, ve kterém se může nacházet. K tomuto účelu poslouží jednoduchý lineární klasifikátor. K určení příslušnosti ostatních prstů se provede vyznačení tzv. linie dlaně (*palm line*) (viz. Ilustrace 6), na základě které se provede označení které prsty jsou přítomny v gestu. Tímto způsobem získáme bitový vektor o 5ti prvních, kterým klasifikujeme analyzované gesto.

4.4 Sledování

Pro klasifikaci gest ve statickém obraze postačí využití metod pro detekci, segmentaci a klasifikaci. Vzhledem k tomu, že součástí této práce je zpracování videa v reálném čase, je nutné co nejlépe snížit náročnost samotné detekce a klasifikace. K tomuto účelu je možné použít sledování význačných prvků (*features*). Díky použití sledování není nutné procházet při detekci celou plochu snímku z kamery, ale postačí projít blízkou oblast v okolí význačných prvků, tedy pozice rukou.

Základem sledování je nalezení pozice stejného objektu v sekvenci snímků, respektive určení vzájemného přiřazení objektů v rámci sekvence snímků. Celý problém sledování je možné rozdělit do 3 částí: reprezentace obrazu, rozpoznávací model a pohybový model [13]. Vzhledem k tomu, že z předchozích kapitol je již známá jak reprezentace obrazu a rozpoznávací model, tak i pozice nalezených význačných prvků. Je tedy možné v rámci této práce řešit pouze pohybový model.

Vzhledem k tomu, že v této práci se nejedná o sledování objektů s více či méně předvídatelnou trajektorií pohybu, jako je tomu například u sledování chodců ([14], [15]), je možné zanedbat problém překrytí více sledovaných objektů a tyto považovat za neplatné.

Jednoduché a rychlé sledování již klasifikovaných objektů mezi snímky je možné dosáhnout použitím algoritmu KLT (*Kanade-Lucas-Thomasi*). Základem tohoto algoritmu je použití pyramidy snímků (vždy s nižším rozlišením ve vhodném přepočtu) pro aproximaci nové pozice a plynulé sledování význačných prvků pomocí výpočtu rozdílu pozic v sekvenci snímků [16]. Tento algoritmus umožňuje sledování více objektů v reálném čase i pro složitější operace [14],[15].

4.5 Knihovny a nástroje pro detekci rukou a prstů

V této kapitole jsou popsány některé knihovny a nástroje, které se v současnosti používají k detekci rukou a rozpoznávání gest.

V současné době je řešena především blízká interakce s počítačem za použití kamer snímajících hloubku (*depth camera*) bez používání značek (*markers*). Vytváření modelů takto snímaných rukou a následného ovládání programů. Mezi práce, které se věnují této problematice patří například [9], [10] a [11]. Existuje několik knihoven a nástrojů, které umožňují automatické

rozpoznání gest s využitím hloubkových snímků. Nejčastěji jsou tyto nástroje postaveny na senzorech Kinect. Do této kategorie patří nástroje Microsoft a otevřené projekty, například *kineticspace*[19].

Vzhledem k tomu, že tato práce se nebude zabývat použitím hloubkových kamer, bude hlavní součástí této kapitoly především popis nástrojů a metod zaměřených na rozpoznávání gest rukou z běžných snímků.

Nejprve bude ale stručně zmíněn Microsoft Kinect SDK, který umožňuje použití senzorů Kinect. Dále budou popsány knihovny Gesture Recognition Toolik a OpenCV, které umožňují zpracování dat snímaných kamerou bez explicitní znalosti hloubky.

Microsoft Kinect SDK

Tento nástroj umožňuje vytváření aplikací pro senzory Kinect firmy Microsoft. Součástí tohoto vývojového balíčku je rozpoznávání gest celého těla i mluvené řeči. Rozhraní knihovny je možné použít pro projekty psané v C++, C#, Visual Basic a dalších. V posledních verzích přibýly rozšířené nástroje pro detekci, sledování a rozpoznávání samostatných gest rukou.

Gesture Recognition Toolkit

Jedná se o multiplatformní, open-source, C++ knihovnu pod licencí MIT určenou pro rozpoznávání gest, jehož autorem je Nick Gillian[12]. Obsahuje řadu algoritmů z oblasti strojového učení, předzpracování i postzpracování snímků a získávání z zajímavých prvků (*features*) z dat. Celý problém rozpoznávání gest je v této knihovně zapouzdřen do objektu *GestureRecognitionPipeline*. V rámci tohoto objektu je možné nastavovat jednotlivé části zpracování dle vlastní volby. Po nastavení zpracování následuje nahrání testovacích dat a natrénování rozpoznávání. Poté je možné provádět rozpoznávání v reálném čase.

Tuto knihovnu je možné použít pro libovolný vstupní senzor, pokud ten může převést vstupní data do vektorového C++ *double* formátu. K této knihovně patří i jednoduché uživatelské rozhraní, které je součástí jejího instalačního balíčku.

OpenCV

Tato multiplatformní, *open-source* knihovna pod licencí BSD patří mezi nejčastěji používané nástroje spojené s počítačovým viděním. Umožňuje zpracování obrazu i videa, nelineární filtrace obrazu, odhadování pohybu, nalezení popředí i sledování objektů ve videu. Základní rozhraní knihovny je napsané pro C++, existuje však přepis do nejpoužívanějších programovacích jazyků. Velkou výhodou této knihovny je jak nízkoúrovňový přístup ke zpracovávaným datům, tak zapouzdření složitých algoritmů pro segmentaci a klasifikaci obrazu.

Z pohledu této práce je možné v OpenCV najít funkce pro analýzu obrazu, konkrétně vytváření histogramů, filtraci obrazu, geometrické transformace. Vzhledem k potřebě nalezení rukou v plynulém proudu snímků z kamery, je možné použít metody pro nalezení popředí/odstranění pozadí, analýzu pohybu a sledování objektů ve videu. Taktéž je možné v této knihovně najít metody pro jednoduchou i složitou klasifikaci.

Další knihovny a nástroje pro detekci rukou

V této kapitole je výčet několika dalších knihoven a nástrojů, které realizují detekci a klasifikaci gest rukou jak pro běžné kamery, tak pro specializované nástroje, jako je *kinect*.

První z těchto aplikací je *HandVu* [25]. Tato knihovna umožňuje detekovat běžné pozice rukou v reálném čase bez nutnosti kalibrace kamery. Vývoj na této knihovně zřejmě skončil v roce 2011.

Další zde uvedenou aplikací je *Xkin* [26]. Tento otevřený projekt je zaměřený na rozpoznávání gest rukou ze znakové řeči pro USA. K tomu používá hloubkovou kameru nástroje *kinect*.

5 Promítaná hra

V této kapitole bude popsána motivace pro tvorbu této hry a specifikace uživatelského rozhraní. Promítaná aplikace bude mít podobu stolního deskového rozhraní pro hru na hrdiny *Dungeons & Dragons*. Dále je popsán princip této hry s prvky a pomůckami, které je možné použít a zapracovat do grafického uživatelského rozhraní navrhované aplikace. Součástí návrhu aplikace je také přehled existujících interaktivních aplikací. Některé z nich budou popsány na konci této kapitoly.

5.1 Dungeons & Dragons

Dungeons & Dragons (DnD) je stolní hra na hrdiny. Originální verze pochází z roku 1974 a jejími autory jsou Gary Gygax a Dave Arneson. O současnou podobu této hry se stará studio *Wizards of the Coast*. Hru je možné hrát v minimálním počtu 2 hráčů, doporučený počet je 3-5 hráčů v jedné skupině. Každá skupina musí mít vypravěče (v pravidlech označován jako Pán hry). Ten vypráví a režíruje příběh, který hrají hráči. Hráči si vytváří postavy, se kterými poté hrají příběh vypravěče. Pravidla určují, co hráčská postava umí, co může dělat a co je naopak zakázané nebo mimo pravidla. V případě sporných pravidel má vypravěč hlavní slovo. Průběh hry je možné rozdělit na jednotlivá dobrodružství, kterými vypravěč hráče provází hra může skončit smrtí hráčských postav nebo zakončením příběhu. V obecném podání ale hra nemusí skončit nikdy. Pro účel této práce je uvažován koncept jednoduchého dobrodružství.



Ilustrace 7: Příklad herní plochy Dungeons & Dragons

Ke hře stačí pouze papír, tužka a kostky, přesto je vhodné mít několik dalších pomůcek a předpřipravených nástrojů. Mezi ty patří herní plán nebo mapa, na které se příběh odehrává, dále pak cínové nebo plastové postavičky, které slouží k přesnějšímu popisu především částí příběhu, ve kterém se odehrává střet nebo souboj. Především tyto pomůcky budou tvořit hlavní součást navrhované aplikace. Jako příklad herní plochy může posloužit ilustrace č. 7.

Každý hráč potřebuje osobní deník, na kterém jsou napsány vlastnosti, dovednosti a důležité aspekty jeho postavy. Do běžné výbavy každého hráče patří dále kostky a pár oblíbených postaviček. Všechny tyto prostředky je možné zahrnout do navrhované hry.

Vypravěč potřebuje mít připravené nehráčské postavy, které vystupují v jeho příběhu. Každá z takových postav může mít zjednodušený osobní deník a přiřazenou postavičku. Při soubojích a střetech je hlavní rolí vypravěče určovat pozice, dohled a další aspekty hráčských postav. Všechny tyto prvky je možné použít do navrhovaného rozhraní. Přenesením alespoň některých ze zmíněných prvků sníží nároky na vypravěče, zpřesní hru a umožní tak hráčům věnovat se jiným aspektům hry než pravidlům.

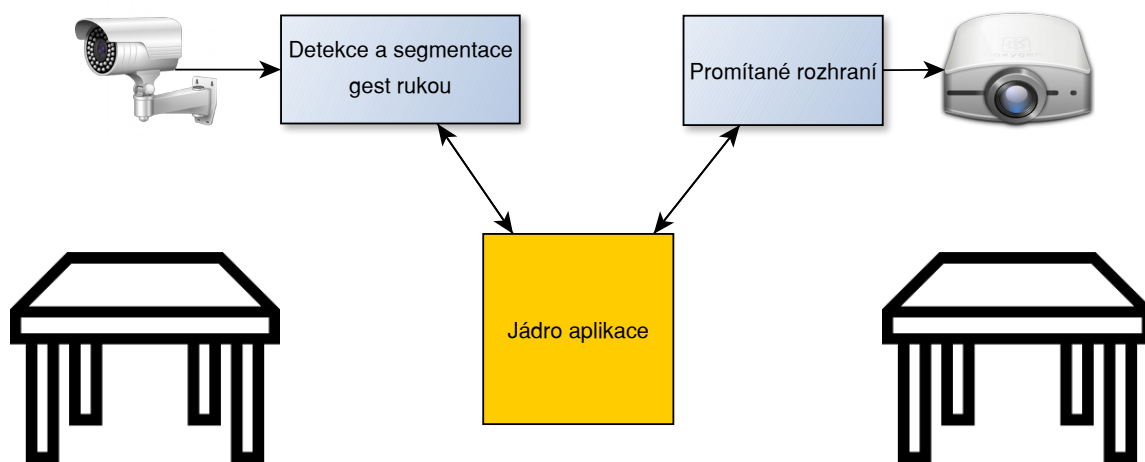
5.2 Specifikace

V předchozí kapitole byly nastíněny některé pomůcky a nástroje, které pomáhají vyprávěný příběh lépe vykreslit, umožňují hráčům rychlou orientaci na herním plánu, případně zjednodušují orientaci v pravidlech.

Několik takových prvků bude implementováno v navrhované aplikaci. Bude se jednat především o mapu, na kterou bude možné umístit virtuální figurky. Mapu bude možné ovládat pomocí gest rukou. Hráči budou mít možnost pracovat s figurkami na mapě, případně posouvat zobrazovaný výřez mapy. Mezi akce, které bude možné s figurkami provádět, je jejich umístění, změna pozice a zobrazení možného dohledu a dostřelu. V první verzi navrhované aplikace je počítáno pouze s 3 typy figurek (bojovník, lukostřelec, čaroděj), s možností tuto databázi posléze rozšířit. Vypravěč bude mít navíc ještě možnost měnit parametry mapy, jako je dohled figurek, osvětlení, případně prostředí a přidávání speciálních prvků do mapy. U speciálních prvků se počítá pouze s přidáním osvětlení jako jsou lucerny, pochodně a další. Taktéž bude možné tuto databázi posléze rozšířit.

6 Návrh

Celou aplikaci je možné rozdělit do několika částí (ilustrace č. 8). První částí je detekce rukou hráčů sedících okolo stolu. Tato část aplikace je popsána v kapitole 6.1. Druhou částí bude jádro aplikace. Třetí částí je promítané uživatelské rozhraní aplikace pomocí dataprojektoru. Návrh třetí aplikační části včetně návrhu ovládání bude popsán v kapitole 6.3. Dále je nutné zajistit komunikaci mezi detekcí rukou (první částí) a jádrem aplikace (druhou částí) a poté promítání uživatelského rozhraní aplikace (třetí část). Popis jádra aplikace včetně rozhraní pro pozici rukou a výstupu k vykreslení je popsán v kapitole 6.2.



Ilustrace 8: Rozdělení jednotlivých částí aplikace při návrhu

6.1 Snímání a detekce gest rukou

Návrh této části aplikace počítá s použitím infračervených kamer pro detekci rukou. Hlavním důvodem použití infra kamery je zmírnění, případně úplné odstranění obrazu promítaného projektorom a tím zjednodušení detekce rukou. Vzhledem k cenám a dosahu infračervených kamer, které by detekovaly teplotu lidského těla, počítá tento návrh s použitím kamery snímající pouze blízké infračervené záření. Taková kamera bude opatřena filtrem, který sníží propustnost viditelné části světla a propustí pouze blízké infračervené záření. Pro zlepšení viditelnosti rukou se počítá s případným nasvícením snímané plochy pomocí infra-LED (viz. kapitola 3.3).

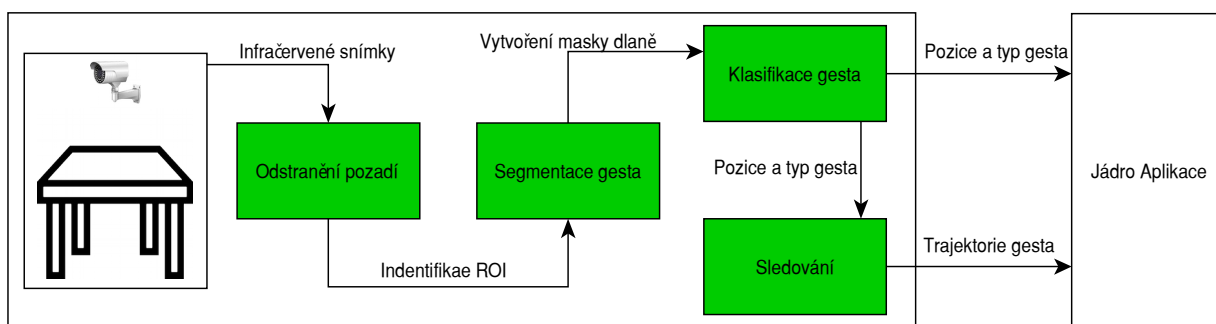
Snížení působení promítaného obrazu umožňuje zjednodušení vlastní detekce rukou. Při návrhu této části aplikace je počítáno s použitím metody odstranění pozadí (viz. kapitola 4.1) a následné detekce, segmentace a klasifikace gest rukou dle metod popsanych v kapitole 4.2, které vycházejí z práce [8]. V této práci je popsán postup detekce a segmentace pouze jedné ruky. V případě, že detekovaná ruka bude podstatně menší než samotný detekční prostor, je možné uvažovat detekci více rukou, v ideálním případě až šesti párů, což odpovídá maximálnímu počtu hráčů.

Samostatnou detekci rukou je v návrhu možné rozdělit do dvou epizod. V první epizodě půjde o vyhledání všech rukou ve snímku a určení jejich poloh. Pro každou nalezenou končetinu se určí

oblast zájmu (*ROI*) na pro další analýzu. Ve druhé epizodě bude každá nalezená oblast analyzována na výskyt gesta samostatně, bez nutnosti analýzy celého obrazu. To bude možné provést paralelně s využitím vícevláknového zpracování.

Mimo výše zmíněné epizody detekce rukou je nutné zachovat přehled o pozicích již nalezených končetin, jejichž pozice se změnila pouze nepatrně, nebo u kterých došlo ke změně typu gesta. K tomuto účelu bude použito sledování, jak je popsáno v kapitole 4.4. Sledování zjednoduší prohledávání celého snímku, neboť budeme znát oblast zájmu ve které se ruka nacházela v předchozích snímcích a tudíž je možné přejít rovnou k analýze gesta, případně upravení oblasti výskytu ruky. Pro každou sledovanou končetinu můžeme taktéž vytvořit jednoznačný identifikátor, který z krátkodobého hlediska umožní zjednoduší komunikaci s jádrem aplikace, která s těmito daty bude dále pracovat.

Druhé části aplikace (jádra) se bude tudíž předávat informace o identifikaci detekované ruky, její pozici a typu gesta. Celý postup zpracování obrazu, detekce gest a předávání informací další části aplikace je graficky znázorněn na diagramu č. 9.



Ilustrace 9: Schéma detekce a segmentace gesta

Gesta potřebná pro ovládání aplikace

Tato kapitola obsahuje výčet ideálního množství gest potřebných pro ovládání aplikace. Gesta by měla být jednoduchá a z uživatelského pohledu jednoznačná. Ze základu můžeme gesta rozdělit na statická a dynamická. Pro pohodlné ovládání by bylo třeba implementovat obě varianty. Při návrhu první verze uvažujeme pouze ta gesta, která jsou absolutně nezbytná k ovládání navrhované aplikace. Taktéž zde musí figurovat gesto, které nebude v aplikaci chápáno jako gesto, ale jako výchozí, klidový stav.

1. **Klidový stav** – Otevřená dlaň, všechny prsty jsou viditelné
2. **Výběr** – Ruka v pěst, žádný prst není vidět
3. **Začátek posunu** – Vztyčený ukazováček a prostředníček, dynamické gesto
4. **Zrušení výběru** – Ukazováček a malíček
5. **Výběr varianty 1-4** – Vztyčený palec, který označuje tuto skupinu + příslušný počet prstů od ukazováčku po prsteníček. Kupříkladu varianta 3. bude identifikována jak vztyčený palec, ukazováček a prostředníček.
6. **Rotace** – Roztažená dlaň, všech 5 prstů je vidět, dynamické gesto.

Těchto 6 gest je nutných pro plnou funkčnost navrhované aplikace. Gesta u kterých je zmíněna poznámka na dynamické gesto označují začátek tohoto gesta. Posunem ruky, popřípadě rotací dojde

k určení hodnoty dynamického gesta. V praktické části bude nutné implementovat minimálně první tři gesta, aby byla zachována minimální funkčnost navrhované aplikace.

6.2 Jádru aplikace

Hlavní část aplikace bude situována kolem mapy, která bude ovládána částí pro detekci rukou a jejíž výřez bude promítán do rozhraní. Tato mapa je popsána níže. Mimo mapu zde bude implementována logika ovládání figurek, uložení jejich pozice na mapě. Kromě této základní funkčnosti je možné přidat algoritmy určující dohled a dostřel figurek (s ohledem na překážky v mapě). Součástí jádra by měla být také databáze figurek, prostředí a případně speciálních prvků, které bude možné přidávat do mapy. Dále bude v jádru uložena informace o rozložení a pozicích důležitých prvků rozhraní.

Interaktivní mapa

Hlavním prvkem celé aplikace je mapa, jejíž výřez je součástí promítaného rozhraní a je označen jako interaktivní mapa (viz. Ilustrace 10). Na tuto mapu bude promítáno prostředí, které vytvoří vypravěč. V rámci první navrhované verze se předpokládá pouze jednoúrovňové prostředí. Součástí tohoto prostředí bude informace o povrchu (barva, textura). Kromě základních parametrů je možné přidat informace o typu a tvaru referenční mřížky přidané k mapě, pozice překážek, umístění osvětlení a poloha a pohyb přátelských i nepřátelských nehráčských postav. Tyto aspekty včetně víceúrovňových prostředí (např.: dvoupatrová budova se sklepením) mohou být součástí dalšího rozšíření. Pro první navrhovanou verzi je plánováno možné načtení libovolného snímku na kterém bude zobrazena příslušná herní mapa.

Do tohoto prostředí budou následně vloženy figurky hráčských postav. Navrhované prostředí i hráčské postavy budou mít 2D, v rámci rozšíření i 3D rysy. Základní informací u postav je jejich poloha na mapě, případně na ovládacích prvcích rozhraní. V rámci rozšíření je možné u postav počítat s dohledem, viditelností, případně dalšími vlastnostmi.

Změnu pozice figurek mohou provádět všichni účastníci hry. Umístění figurky by nemělo kolidovat s prvky prostředí. Informace o změně pozice figurky bude získána vhodným gestem na pozici zobrazované figurky, jak je popsáno v kapitole 6.1.

Součástí jádra budou také algoritmy pro zpracování dynamických gest, především jejich časování a ovládání.

Rozhraní aplikace – detekce rukou

Z části pro detekci rukou bude jádro přijímat informace o identifikaci a pozici ruky a typ gesta. Identifikátor ruky označuje konkrétní sledovanou končetinu. Tato informace je důležitá pro zpracování dynamických gest. V rámci statických gest bude přenášena pouze informace o jejich přechodu. Podle pozice gesta bude možné rozlišit gesta určená pro ovládání mapy a ovládacích prvků.

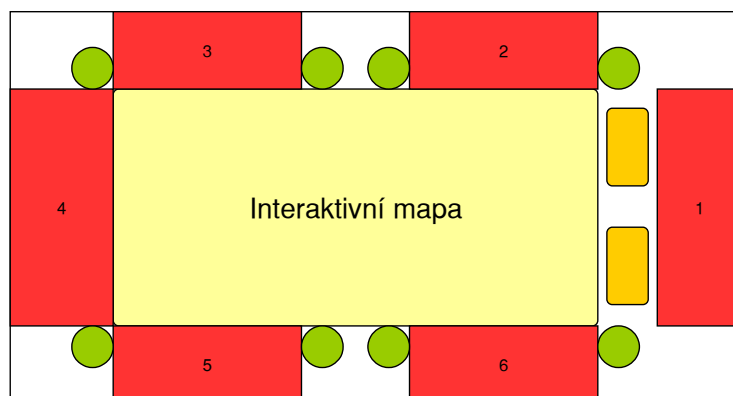
Pro zachování co nejmenší komunikace budou přenášeny pouze informace o změnách polohy, popřípadě gesta. Vzhledem k tomu, že bude detekováno větší množství končetin, bude nutné zajistit přenos pozice všech změněných informací. Díky identifikátoru ruky bude možné přenášet pouze informace o změně pozice, popřípadě gesta s příslušným identifikátorem. Na straně jádra se poté provede zapsání změny a případně změna stavu u dynamických gest.

Rozhraní aplikace – projekce

Z jádra aplikace bude výstupem výřez mapy. S výřezem mapy bude možné pohybovat pouze pokud to velikost mapy umožní. Další část rozhraní bude obstarávat nahrávání textur vykreslovaných figurek (pokud již nebudou načteny) a vlastnosti světla (pokud budou součástí mapy) do vykreslovací části.

6.3 Zobrazované rozhraní aplikace

V kapitole 5.1 jsou popsány prvky, které je možné implementovat do aplikace. Celý prostor stolu na kterém bude hra probíhat je nutné rozdělit na části, o jejichž ovládnutí se aplikace bude starat. Dále na části, ve kterých budou mít hráči osobní prostor, do kterého nebude mít aplikace přístup. Bez újmy na obecnosti můžeme předpokládat, že kolem herního stolu bude maximálně 5 hráčů a vypravěč. Dále můžeme předpokládat, že projekční prostor zasahuje do celého prostoru stolu. Součástí promítaného rozhraní tedy budou osobní prostory hráčů a vypravěče, dále interaktivní mapa (viz kapitola 6.2), která bude hlavním bodem celé aplikace a ovládací prvky. Ovládací prvky by měl mít na starosti vypravěč a proto jsou situovány blíže k čelu stolu. Rozhraní zobrazované aplikace je vidět na náčrtku č. 10.



Ilustrace 10: Návrh rozhraní aplikace

Na tomto náčrtku je možné rozlišit součásti rozhraní, které budou pro interakci s aplikací důležité. Červené oblasti 1-6 označují osobní prostor hráčů. Zelené oblasti jsou odkládací prostory pro uchování neaktivních postavíček a oranžové oblasti představují ovládací panel pro vypravěče.

Ovládací prvky

Ovládací prvky na straně vypravěče slouží k nastavení vlastností mapy. Jedná se o dvě „okna“ ve kterých bude mít pozice rukou speciální význam (oranžové oblasti na ilustraci 10). V prvním okně se gestem zvolí veličina, která se bude na mapě měnit (např.: viditelnost, osvětlení) a v druhém okně se dalším gestem určí změna velikosti.

Odkládací políčka (zelené oblasti na ilustraci 10) na straně hráčů budou obsahovat figurky, nad kterými mají hráči kontrolu. Figurky může ovládat kterýkoli z přisedících, z pohledu hráčů je ale vhodné mít vlastní figurky blízko a kdykoliv k dispozici, pokud nejsou zrovna součástí dění na mapě.

Za ovládací prvek je možné považovat i osobní prostor hráčů (červené oblasti na ilustraci 10), ve kterém jsou odloženy věci se hrou nesouvisející. Libovolné akce hráčů i vypravěče v těchto oblastech nesmí jakkoli ovlivnit chování zbytku herní plochy.

7 Implementace

V této kapitole je schématicky popsána implementace aplikace s předlohou v návrhu, viz. Kapitola 6. Aplikace je psána v jazyce *C++* s použitím knihoven *Qt* a *OpenCV*, pro perzistentní uložení dat je použita knihovna *RapidXml* (podrobnější popis v kapitole 7.5). Obdobně jako návrh, tak i technické řešení můžeme rozdělit na tři základní bloky, kterými jsou:

- *Detekce*
- *Kontrolér*
- *Projekce*

Kromě těchto hlavních částí je součástí implementace aplikační rozhraní, které uživateli umožňuje jednoduché přidávání charakterů a načítání mapy. Detekce vychází z algoritmu popsaného v kapitole 6.1. Jako kontrolér je dále označována část aplikace uvedená jako jádro v kapitole 6.2. Projekce je poté implementována na základě návrhu aplikace v kapitole 6.3.

Před popisem jednotlivých částí softwaru bude nejprve popsána samotná sestava. Poté následuje popis jednotlivých částí se strukturovaným popisem implementovaných algoritmů, případně jejich odchylek oproti předloženému navrhovanému postupu. Dále jsou v této kapitole zmíněné použité nástroje, které nebyli popsány již dříve. Na konci kapitoly je poté kapitola která popisuje problémy a úskalí při implementaci celé aplikace.

7.1 Sestava

V návrhu (kapitola 6) byl představen prvotní plán rozložení hardwarové sestavy. Ten se skládá z projektoru, infračervené kamery a herní plochy. Za herní plochu lze obecně považovat libovolný dostatečně veliký stůl. Projektor má být umístěn nad stolem se zajištěním projekce na stůl. Kamera by měla být umístěna v blízkosti projektoru tak aby bylo možné snímat projekční prostor. Pro kvalitní detekci by projekční prostor měl být plně obsažen v detekčním prostoru kamery.

Vzhledem k problémům při implementaci (viz. kapitola 7.6) bylo nutné tuto sestavu lehce upravit. Nebylo totiž možné zajistit dostatečně kvalitní odfiltrování promítaných dat od snímaných rukou. Proto byla sestava rozdělena na dvě části. První částí je projekční prostor, druhou částí je prostor detekční.

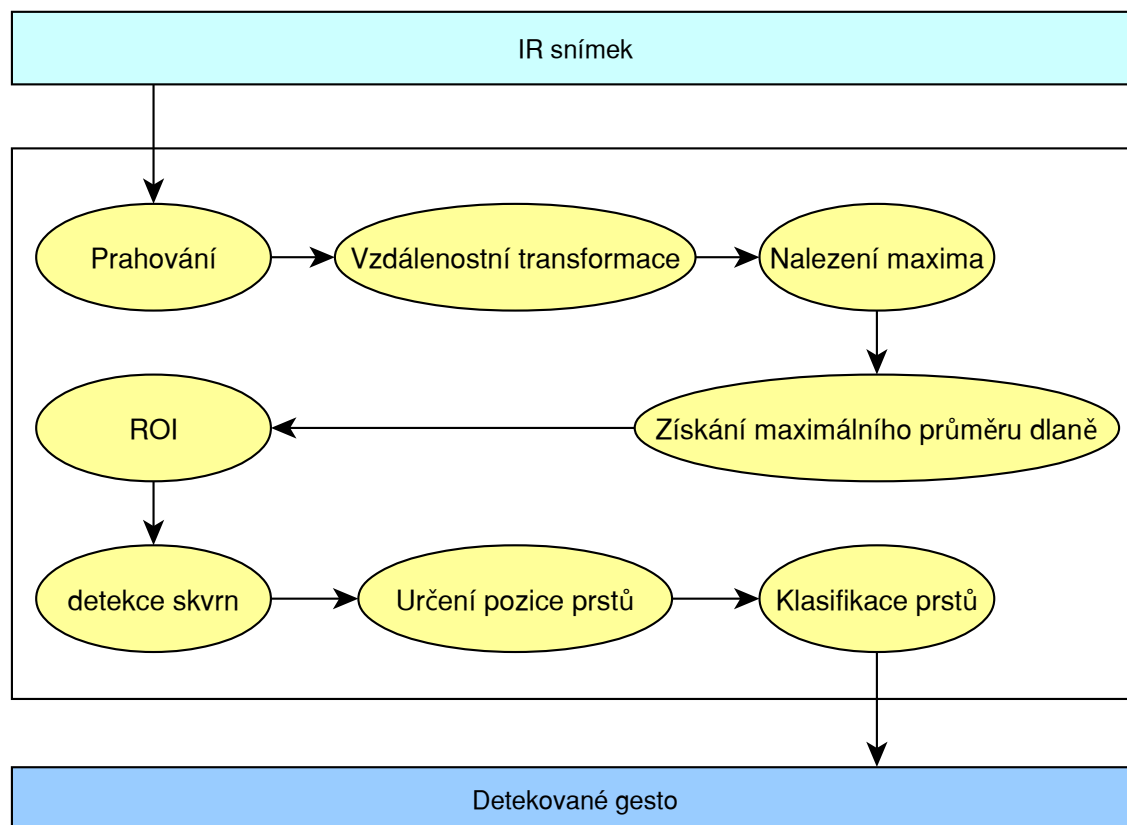
Na ilustraci 11. je vidět celá sestava. Projektor umístěný nad projekční plochou promítá uživatelské rozhraní. Před stolem je pak detekční prostor, přičemž pro snížení šumu z prostředí je použita speciální pokrývka, která minimalizuje záření v blízké infračervené oblasti světla. Kamera poté snímá tento prostor a reaguje na gesta uživatele.



Ilustrace 11: Sestava

7.2 Detekce

Princip určení pozice ruky v zorném poli kamery vychází z algoritmu, který byl popsán v kapitole 4. Vstupními daty pro zpracování detekce je sekvence snímků z infračervené kamery. Schematicky lze tento postup zobrazit následovně:



Ilustrace 12: Algoritmus detekce gesta

Vzhledem k tomu, že tato práce počítá s použitím infračervené kamery, je vstupní obraz pouze jednobarevný. Tím pádem je možné provést prvotní separaci šumu od dat pomocí prahování. Tím získáme černobílý obraz ve kterém hledáme gesto. Ve druhém kroku provedeme vzdálenostní transformaci (*distant transform*). V takto upravených datech hledáme nejintenzivnější body, kterými by mohli být středy dlaní. Tím získáme pozici gesta. Dalším krokem je určení orientace dlaně a nalezení masky dlaně. Masku dlaně nám umožňuje přesněji určit pozici palce a rozlišit ostatní prsty. Získáme ji snímáním vzdáleností v pravidelném úhlu v okolí již známého středu dlaně. Na této masce můžeme spolehlivě určit střed zápěstí. Rotace dlaně je pak určena středem dlaně a středem zápěstí. Tím získáme přibližnou oblast zájmu (*region of interest*). V této oblasti známe polohu středů dlaně i zápěstí a můžeme tedy odhadnout pozici jednotlivých prstů. Pro zjednodušení je použita detekce skvrn (*blob detection*) pro nalezení jednotlivých prstů. Dále jsou provedeny výřezy předpokládané pozice prstů. V případě že nalezená skvrna patří pouze do jednoho výřezu, pak je příslušný prst detekován. V případě že není některá ze skvrn identifikována, pak je provedena přesnější detekce pomocí zaplnění výřezu. K tomu dochází, pokud je detekovaná skvrna příliš velká. Tento parametr je velmi závislý na použité kamce a způsobu předchozího zpracování.

Speciálním případem rozpoznání prstu je detekce palce. Ten je totiž specifický polohou, lépe řečeno výraznějším úhlem mezi středem dlaně a zápěstí.

V posledním kroku je provedeno předání informace o pozici a aktivních prstech do kontroléru pro gesta, který takto získané gesto zaeviduje a vytvoří příslušný objekt, který se stará o jeho zprostředkování do kontroléru.

Struktura výše popsaného algoritmu je implementována následujícími třídami:

- `ImagePreprocessing`
- `HandDetection`
- `GestureProcessing`
- `Gesture`

Třída `ImagePreprocessing` zajišťuje tu část algoritmu, která je na digramu 12 ohraničena a to včetně určení příslušnosti prstu ke gestu. Z této třídy je informace o pozici a gestu předány na zpracování do třídy `GestureProcessing`, kde je gesto zpracováno do datové třídy `Gesture`. V této třídě se data uchovávají do té doby, dokud je gesto aktivní, čili dokud je sledovatelné a detekované ve vstupních snímcích. Sledování gesta je provedeno velmi jednoduchou kontrolou vzdálenosti mezi gesty detekovanými v sekvenci. Třída `GestureProcessing` slouží v podstatě jako generátor jednotlivých gest a používá se k jejich ukládání a manipulaci s nimi. Při změně pozice gesta dojde k přenosu signálu do třídy `HandDetection`, která jej přepošle ke zpracování do kontroléru. Tato třída zároveň zapouzdřuje a řídí činnost celé detekční části.

Zpracování gest

Z navrhovaných statických gest (kapitola 6.1) byli nakonec implementováni pouze tři a to:

- *Klidový stav* (otevřená dlaň)
- *Posun mapy* (vztyčený ukazováček a prostředníček)
- *Uchopení postavy* a její přesun (ruka v pěst, není vidět žádný prst)

To odpovídá přirozené práci s figurkami jako při použití reálných figurek a map. Detekce statických gest vychází z výše uvedeného algoritmu, jehož výstupem je již informace o pozici gesta a zúčastněných prstů. Z této informace je možné určit jedno z výše uvedených gest, případně žádné gesto pokud není rozpoznáno.

Scénář provádění dynamických gest počítá s tím, že uživatel přirozeným pohybem ruky v gestu otevřené dlaně položí ruku buďto pouze na mapu, nebo přímo na konkrétní zobrazenou figurku. Poté figurku uchopí stisknutím ruky v pěst obdobně, jak by to dělal pokud by v tom samém místě byla reálná figurka. Následným pohybem přemístí příslušnou figurku na jiné místo na mapě nebo na rozhraní. Toto dynamické gesto je ukončeno detekcí klidového stavu, tedy otevřením dlaně z pohledu uživatele.

Druhé implementované dynamické gesto slouží k posunu promítané mapy. Scénář tohoto gesta opět vychází z klidového stavu otevřené dlaně. Posun začne být aktivní v okamžiku změny tvaru detekované ruky do podoby vztyčeného ukazováčku a prostředníčku. V této podobě a pozici nad mapou je možné s mapou pohybovat stejným způsobem, jako při pohybu s postavou. Gesto je ukončeno opětovným přechodem do klidového stavu, nebo opuštěním detekčního prostoru.

7.3 Kontrolér

Hlavní funkcí kontroléru je zajištění ovládání a režie projekce. Součástí této části je tedy předávání dat mezi detekční a projekční částí, včetně zajištění kalibrace. Tato část implementace je vytvořena

pomocí 3 tříd. První z těchto tříd je `CodeData`. Tato třída zajišťuje propojení datového toku mezi detekcí, ze které přijímá signály s informací o změněném gestu, a projekcí, do které posílá informace o změně pozice zobrazovaných entit.

Kromě zprostředkování interakce mezi detekcí a projekcí kontrolér také zajišťuje reakce na uživatelské příkazy z aplikačního rozhraní, jako je načítání promítané mapy, lépe řečeno její přiřazení k projekci, přidání charakterů a přepínání režimů kalibrace a projekce.

Aby bylo možné provádět pohyb dynamických promítaných prvků, tedy mapy a charakterů, je nutné aby byla v kontroléru uložena pozice těchto entit.

Vzhledem k tomu, že výstupní projekce *OpenGL* pracuje v projekčním prostoru, který nabývá hodnot $-1 - +1$ pro osy X i Y s počátkem ve středu vykreslovaného okna a jejichž poměr je závislý na poměru stran vykreslovaného okna, je nutné tento prostor zobecnit tak aby bylo možné provést jednoduší zpracování vzhledem k detekční části, která pracuje s pozicí ve snímku v pixelech a s počátkem souřadného systému v levém horním rohu snímku. Z toho důvodu byla zvolena struktura `Point` pro uchování pozice dynamických entit ve speciálním souřadném systému. Počátkem tohoto systému je opět střed obrazovky aby byl zachován vztah k projekci. Rozměry tohoto prostoru by měli být v rozsahu známého rozlišení vykreslovaného okna v pixelech. Tím je vytvořena možnost pohodlné konverze mezi detekčním a speciálním souřadným systémem a následnou projekcí. Propojení mezi detekčním a speciálním souřadným systémem je proveden kalibrací.

Kromě kalibrace, která je implementována třídou `CalibrationModule` a která bude popsána dále je nutné v kontroléru zajistit uložení důležitých persistentních dat. Mezi tyto data patří především informace o pozici charakterů a mapy v projekčním prostoru a kalibrací získaná homomorfní matice. Ta se ukládá především z toho důvodu aby nebylo nutné provádět kalibraci při každém spuštění aplikace ve stejné hardwarové konfiguraci. Persistentní uložení dat je implementováno ve třídě `DataPersistences`. Data jsou ukládána do strukturovaného XML dokumentu jehož parsování zprostředkovává knihovna *RapidXml*.

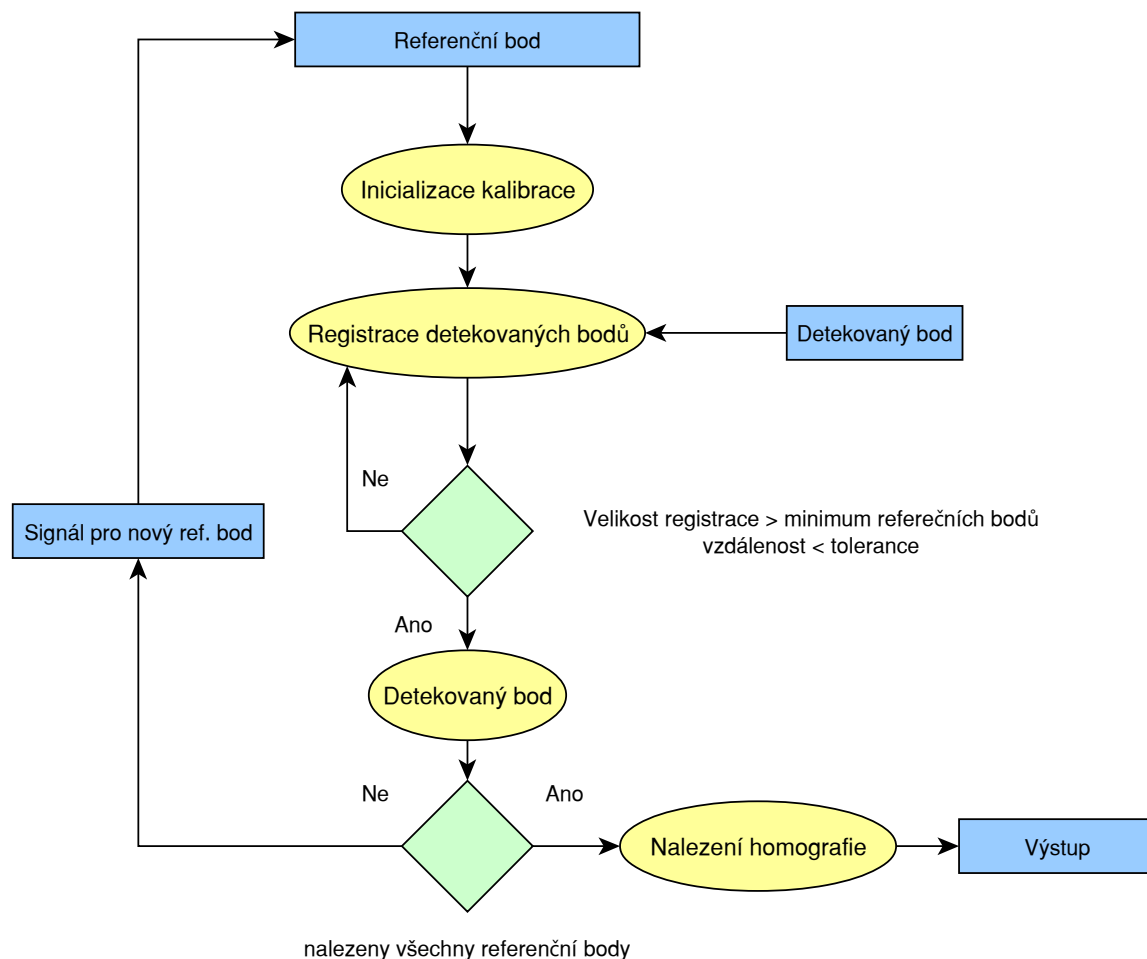
Kalibrace

Základní funkcí kalibrace je nalezení homomorfní matice mezi mezi známými body projekce a detekce. Tato funkčnost je implementována ve třídě `CalibrationModule`. Na diagramu 13 je vidět postup nalezení homomorfní matice.

Nejprve je vložen referenční bod, který bude promítán projekcí. U tohoto bodu je známa pozice ve speciálním souřadném systému kontroléru. Nyní je třeba nalézt odpovídající referenční bod v detekčním prostoru kamery. Třída `CoreData` zajistí propojení signálu nalezeného gesta z detekční části a předá je ke zpracování do třídy `CalibrationModule`. Samotná informace o pozici gesta ale nestačí, neboť není možné přesně určit referenční bod pouze z jednoho snímku, respektive jediného gesta. Místo toho je implementován zásobník, který bere v potaz posledních 10 snímků. To umožňuje určit detekovaný bod až po ustálení pohybu ruky v detekci, tedy poté co uživatel umístí ruku na příslušnou promítanou značku referenčního bodu. Takto nalezený referenční bod v detekčním prostoru je zaevidován a pokračuje se v kalibraci vložením dalšího referenčního bodu, respektive změnou pozice referenční značky při projekci.

Kalibrace jednotlivých bodů proběhne pro čtyři referenční body, což je minimální potřebný počet bodů pro určení homografie mezi vkládanými referenčními body a detekovanými body. K určení homografie je použita funkce knihovny *OpenCV* `findHomography()`. Tato metoda

používá metodu *RANSAC* jejíž podrobný popis je možné najít v kapitole 3.2. Vstupem metody je pole referenčních bodů a pole detekovaných bodů. Výstupem je poté příslušná homomorfní matice. Tím je kalibrace ukončena, objekt třídy *CalibrationModule* není již potřebný a režie třídy *CoreData* přepne aplikaci do projekčního režimu.



Ilustrace 13: Diagram průběhu kalibrace

Pro zvýšení přesnosti kalibrace je možné přidat větší počet referenčních bodů čímž je možné dosáhnout přesnější kalibrace za cenu jejího prodloužení.

Kalibraci by bylo možné provést pomocí referenční šachovnice. Zvolený postup je uvažován s ohledem na potencionální uživatele u kterých se nepředpokládá, že vlastní referenční šachovnici. V případě, že by projekční prostor byl v prostoru detekčním, je zvolený způsob kalibrace dostatečně přesný. Pokud by se ukázal opak, je možné zvýšit počet kalibračních bodů a tím zvýšit přesnost výpočtu homomorfní matice.

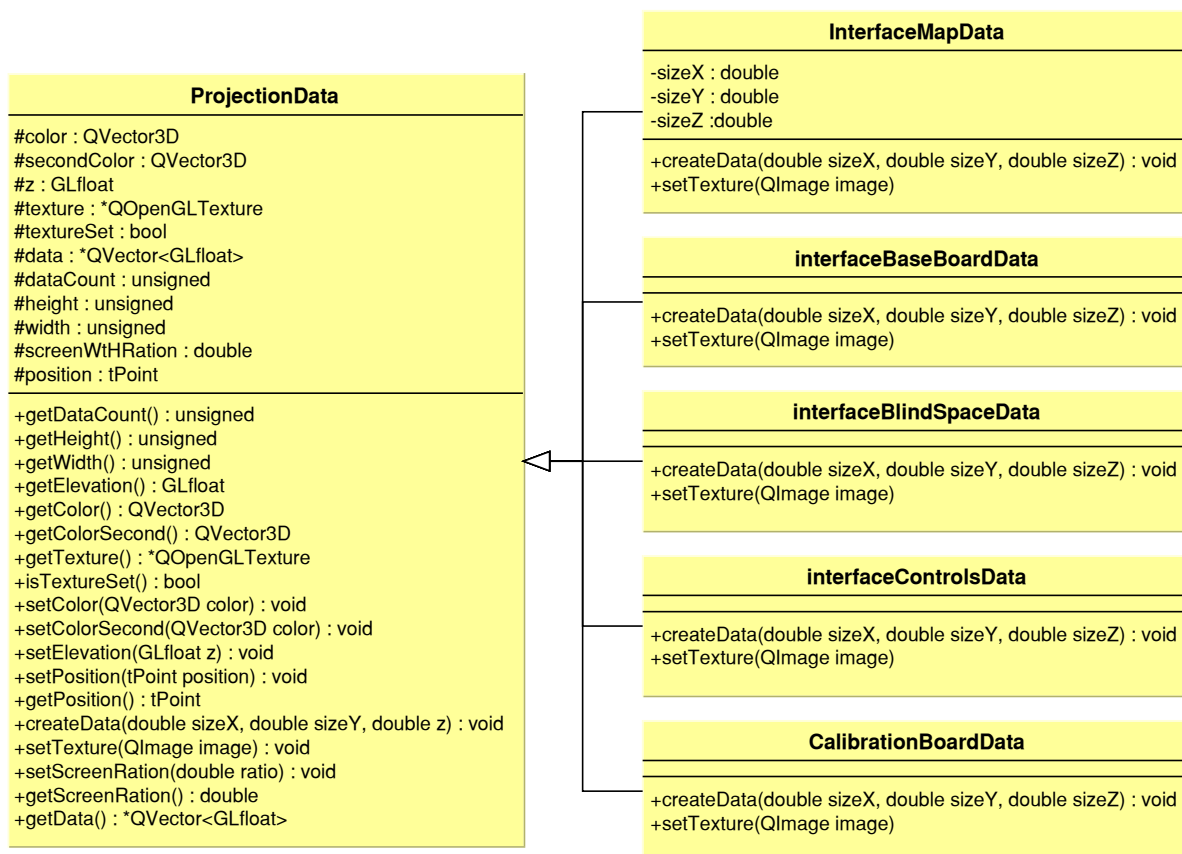
7.4 Projekce

Třetí a poslední významnou částí implementace je projekce rozhraní hry, mapy, postavíček a kalibračního prostředí. Data jsou vytvářena s ohledem na tvar a textury jednotlivých částí. V této části hraje významnou roli dědičnost tříd, neboť vykreslování je pro všechny objekty stejné, mění se pouze data.

Vykreslení je prováděno s ohledem na velikost promítaného prostoru a to tak, aby byl zachován poměr v promítání rozhraní i poměr stran promítaných tvarů a textur. Celá projekce je zamýšlena jako celoplošná. Hloubka v této projekci slouží pouze k určení pořadí a viditelnosti jednotlivých prvků.

Projekci jako takovou zapouzdřuje třída `ProjectionPlane`. Hlavní činností této třídy je zajištění vykreslování datových struktur rozhraní a charakterů a práce s nimi. `ProjectionPlane` jako taková je potomkem třídy `QOpenGLWidget`, která umožňuje použití *OpenGL* při programování vykreslování. Podtřídy třídy `QOpenGLWidget` přetěžují metodu `initializeGL()`, ve které je prováděna inicializace vykreslovacího řetězce *OpenGL* jako je vytvoření programu pro zpracování dat ve *vertex a fragment shaderu*. Dále pak `paintGL()` ve které se provádí vložení dat, úprava příslušných transformačních matic, načítání textur a další činnosti spojené s obsluhu vykreslování.

Metody použité pro vložení dat na grafickou kartu jsou `drawStaticEnviroment` a `drawDynamicElement` přičemž první metoda slouží k vykreslení statických prvků použitím jednotné transformační matice, kdežto metoda `drawDynamicElement` pracuje s transformačními maticemi pro jednotlivé vykreslované prvky.



Ilustrace 14: Digram tříd projekce

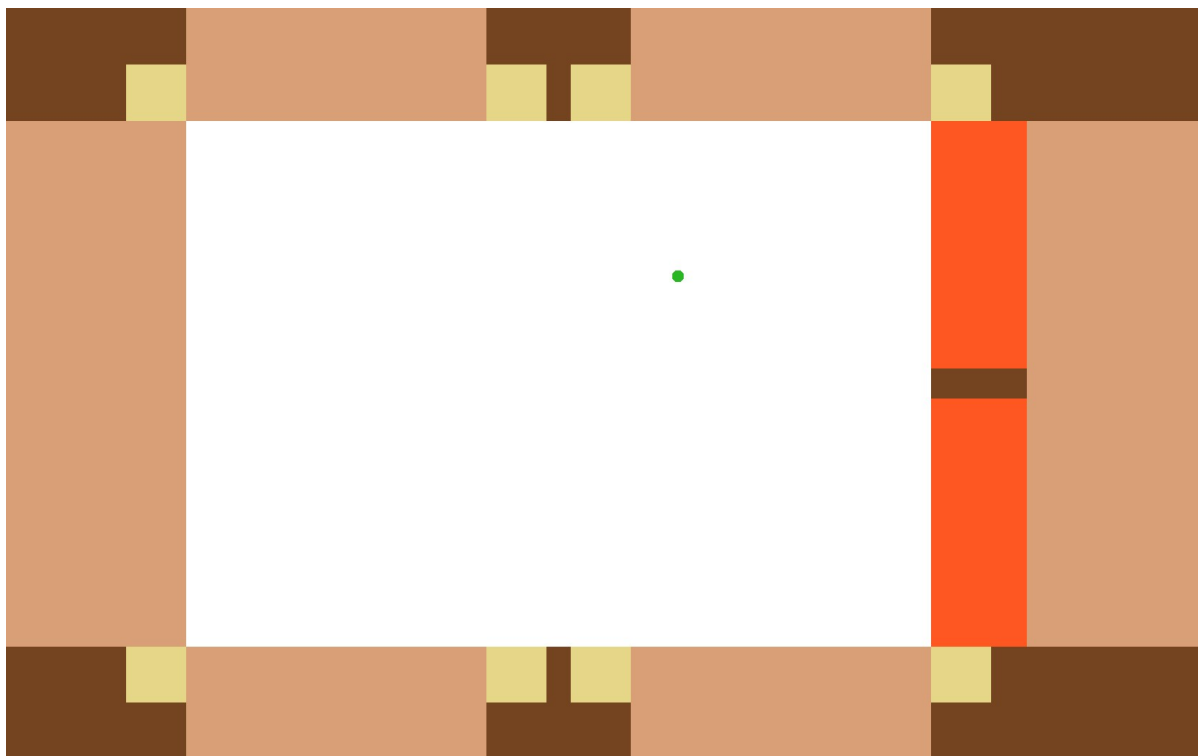
Rozhraní

Na diagramu 14 je vidět popis implementace jednotlivých tříd s ohledem na zachování jednotného rozhraní pro práci s libovolnými daty. Třída `ProjectionPlane` si uchovává informace o všech zobrazovaných objektech. V tomto případě se jedná o zobrazení statických dat, tedy uživatelského rozhraní. Jednotlivé implementační třídy odpovídají částem v promítaných datech.

Nejvýznamnějšími atributy jsou v rodičovské třídě `ProjectionPlane` především data, ve kterých jsou uloženy informace nutné k projekci. Tato data vytváří každý potomek dle své vlastní implementace metody `createData()`. V případě statického zobrazení rozhraní je dále důležitá informace o pozici v *z-bufferu*, který v této implementaci určuje viditelnost jednotlivých zobrazovaných prvků.

Promítané rozhraní poskytuje základní prvky pro ovládání hry a projekci mapy i charakterů. Z výše uvedeného diagramu tříd zpracovává veškeré prvky rozhraní jako statické prvky a z diagramu tříd charakterů promítá všechny třídy jako dynamické, pohyblivé prvky.

Jak je vidět na ilustraci č. 15, uživatelské rozhraní je implementováno dle návrhu (kapitola 6). Jedná se o jednoduchou implementaci projekce pomocí *OpenGL* ve třídě `ProjectionPlane`. Jednotlivé ovládací prvky jsou barevně rozlišeny. Každá barva odpovídá instanci jedné třídy z diagramu 14. Rozhraní ohraničuje bílý prostor do kterého je promítána mapa s charaktery. Charaktery je možné uchytit na mapě, nebo je odložit na rozhraní.

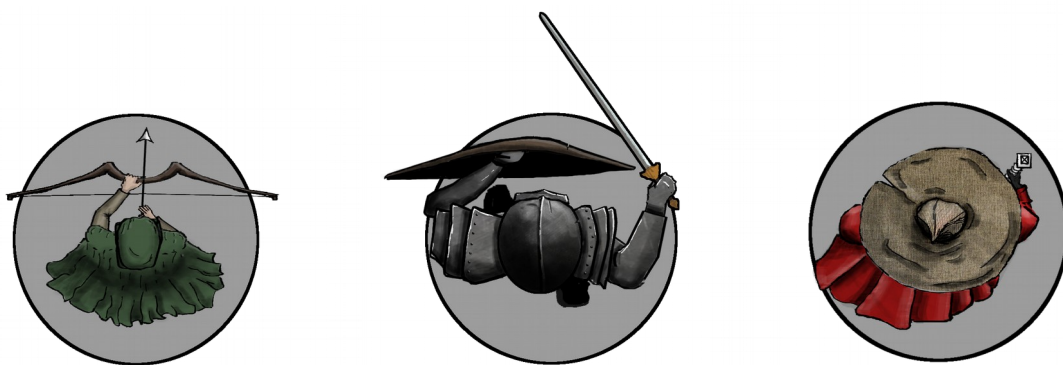


Ilustrace 15: Vykreslené rozhraní promítané hry

Charaktery jsou rozděleny do tří kategorií a to:

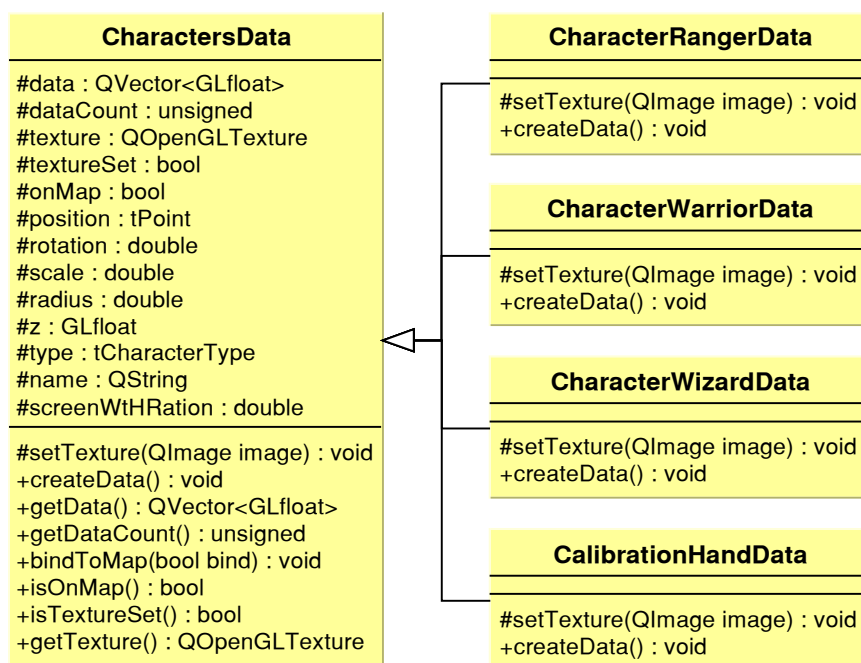
- Čaroděj
- Bojovník
- Lovec

Tyto tři postavy je možné zobrazit s vlastní texturou a tvarem, který je charakteristický a jedinečný. U každé z těchto postav je možné provádět změnu pozice a rotaci kolem středu. Rotace není v této práci propojena s příslušným gestem, tudíž je možné s postavou pouze pohybovat.



Ilustrace 16: Textury charakterů v této práci. Zleva: Lovec, bojovník a čaroděj

Textury postav jsou vidět na ilustraci č.: 16. Při pohledu na níže uvedený diagram tříd charakterů je vidět, že rozmanitost postav je možné rozšířit implementováním potomka třídy `CharactersData` u kterého je nutné pouze zajistit generování výstupních dat tvaru a textury.

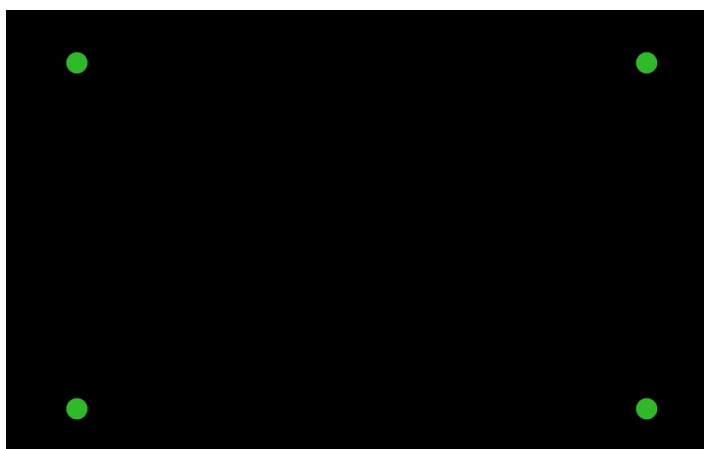


Ilustrace 17: Diagram tříd promítaných charakterů

V rodičovské třídě `CharactersData` je implementována základní projekční funkčnost pro zobrazování charakterů. Mezi nejdůležitější atributy této třídy patří opět `data`, které uchovávají informace nutné pro vykreslení dat pomocí *OpenGL* ve třídě `ProjectionPlane`. Dále pak pozice v *z-bufferu* `z`, která určuje viditelnost vůči ostatním zobrazovaným prvkům. Atribut `onMap` slouží k upevnění charakteru k mapě. Vykreslení uchycení k mapě je provedeno ve třídě `ProjectionPlane` s použitím projekční matice vykreslené mapy.

Kalibrace

Důležitou částí projekce je i kalibrační plocha. Původní návrh počítal se sjednoceným projekčním a detekčním prostorem a z toho důvodu byla kalibrace zamýšlena jako interaktivní projekce pro uživatele. Ta spočívá v projekci referenčních bodů. Referenční body jsou získány stejným postupem jako charakterly bez textury.



Ilustrace 18: Projekce kalibrace

Do každého z těchto bodů uživatel přiloží otevřenou dlaň a detekce určí její pozici. Po provedení několika iterací je možné získat homomorfni matici, která určuje sjednocení projekčního prostoru s detekčním. Podrobnější popis kalibrace je popsán v kapitole 7.3. Po provedení kalibrace je aplikace připravená k použití.

7.5 Použité nástroje

Celá aplikace je implementována v prostředí *Qt* verze 5.6.0. Využití frameworku *Qt* v této práci zjednodušuje projekci použitím *OpenGL widgetu*, který *Qt* zprostředkovává. Implementace také výrazně využívá mechanismu signálů a slotů pro komunikaci mezi objekty a jako reakci na zpracované detekované gesta.

V detekční části je jako hlavní nástroj použita knihovna *OpenCV*, jejíž popis potřebný pro tuto práci je v kapitole 4.5.

K perzistentnímu uložení dat je v této práci použita knihovna *RapidXml*, která zprostředkovává ukládání a načítání dat uložených ve strukturovaných souborech XML. Jedná se o rychlý přenositelný XML parser napsaný v jazyce C++.

7.6 Problémy při implementaci

Při implementaci aplikace vzniklo několik překážek, které odklonily vývoj od navrhovaného výsledku. Z převážné části se jedná o zpracování a detekci gest u které se předpokládalo použití infračervené kamery od firmy Cameo s minimálním rozlišením 1280x720p a frekvencí 50 snímků za vteřinu. Při předpokladu, že by kamera snímala celý herní stůl, lze uvažovat, že by jeden pixel odpovídal přibližně 2mm na stole. To by umožňovalo poměrně přesnou detekci při úvaze, že prostor detekované ruky je podstatně menší než celý detekční prostor. Detekční spektrum této kamery mělo snímat především blízké infračervené záření o vlnové délce v rozmezí 800 – 1200nm. To by umožnilo téměř úplně odstranit šum ve vstupních datech a to i při přímé projekci rozhraní do detekčního prostoru kamery.



Ilustrace 19: Problémová detekce při použití běžné webkamery

Především z důvodu pozdního dodání této kamery bylo nutné přehodnotit přístup k detekci a upravit příslušné části aplikace tak, aby ji bylo možné použít i s jiným, méně kvalitním vstupem. Dále jsou popsány změny v aplikaci a problémy, které použití méně kvalitní kamery přináší.

Základním důvodem použití infračervené kamery bylo odfiltrování viditelného světla z projekčního prostoru, což by výrazně usnadnilo detekci pozice a gesta ruky. Právě z důvodů velmi pozdního dodání infračervené kamery, kterou se nepodařilo v rozumné době před odevzdáním této práce zprovoznit bylo nutné použít méně kvalitní kameru. V případě této práce se jedná o běžnou webkameru s odstraněným infračerveným filtrem a rozlišením 640x480p a frekvenci 5 snímků za vteřinu. Rozlišení použité webkamery je čtvrtinové, což snižuje přesnost kamery. Takto upravená webkamera sice umožňuje snímání požadované části spektra, tedy přijímá i záření o vlnových délkách přesahující 800nm, ale neodfiltruje promítaná data a efektivně tak znemožňuje splnění prvotního návrhu na společný projekční a detekční prostor. Taktéž snížené rozlišení s přidaným šumem znemožňuje detekci původně zamýšleného prostoru a tudíž bylo nutné upravit detekci tak aby snímala gesta, která budou zabírat významnou část detekčního prostoru. Z toho důvodu bylo nutné návrh aplikace upravit tak aby se projekční a detekční prostor nepřekrývali. Touto úpravou je možné zachovat plnou funkčnost navrhované a implementované aplikace. Přesto tato úprava výrazně zasahuje do výsledné uživatelské zkušenosti při interakci s implementovanou aplikací.

Vzniklý detekční problém je vidět na ilustraci 19. První snímek zobrazuje snímaný prostor při projekci, druhý snímek pak ukazuje vstupní data pokud je detekce prováděna mimo promítaný prostor. Očekávaný vstup z původně uvažované Cameo kamery měl mít kvalitnější podobu druhého snímku.

I přes tyto problémy je výsledná implementovaná aplikace použitelná s nižší přesností a při rozdělení projekce a detekce.

8 Výsledky a testování

V této kapitole je popsáno testování aplikace a zhodnocení výsledků. Testování bylo prováděno za účelem určení přesnosti detekce a rozpoznání gesta. Dále byli provedeny uživatelské testy které měly za účel zjištění uživatelské přívětivosti implementované aplikace a to i s přihlédnutím na kvalitu detekce.

V první podkapitole je uveden postup určení gesta při testování, včetně rozboru možných problematických míst v algoritmu ve kterých by mohlo docházet k problémům při detekci. Dále jsou uvedeny výsledky testování detekce gesta, včetně zhodnocení. Následuje podkapitola uživatelského testování ve které je popsán scénář na kterém bylo testování prováděno. Výsledky tohoto testování jsou uvedeny v další podkapitole. Nakonec je uvedeno celkové shrnutí výsledků testování, určení problémových částí aplikace i postřehy hráčů kteří se podíleli na testování předložené aplikace.

8.1 Detekované gesto

Gesto je detekováno na základě nalezení vhodné oblasti zájmu, odstraněním dlaně a označením příslušných prstů. Algoritmus detekce byl podrobně popsán v kapitole 7.2. Střed dlaně je zde použit jako referenční bod pozice gesta. Na následujícím snímku je vidět postup detekce. Na prvním snímku je možné vidět čistá vstupní data z kamery. Druhý snímek pak reprezentuje oblast zájmu, tedy dlaně se zvýrazněným prostorem ve kterém jsou jednotlivé prsty očekávány. Na třetím snímku jsou zvýrazněny pole ve kterých je očekávána přítomnost prstu. V případě vhodného zaplnění příslušného



Ilustrace 20: Postup detekce a klasifikace gesta

pole je prst vyhodnocen jako přítomen. Určení těchto jednotlivých polí je jedním z možných problematických míst při detekci, neboť při špatně zvolené pozici může docházet k významným odchylkám při klasifikaci příslušného gesta. Detekce může být klasifikována chybně, pokud dojde k výraznějšímu zakřivení osy prostředníček – střed dlaně – střed zápěstí.

Testování bylo prováděno za různých světelných podmínek více lidmi s rozdílnou velikostí dlaní. Scénářem testování bylo položení dlaně s příslušným gestem do detekčního prostoru kamery. Následně se po půl vteřině zjišťovalo detekované gesto. Tento čas byl zvolen vzhledem k nižší frekvenci snímků přijímaných z webkamery a rychlosti zpracování gesta.

8.2 Testování

Testování bylo provedeno na stroji Acer E 15 s jádrem Intel i5-5200U, grafickou kartou NVIDIA GeForce 940M a webkamerou (parametry viz kapitola 7.6). Vzorek testovaných dat obsahuje 45 snímků na kterých jsou detekována gesta. Vzhledem k tomu, že bylo implementováno rozlišení pouze tří gest, připadá na jedno gesto vzorek o velikost 15 snímků. Úspěšnost této detekce je zobrazena na následující tabulce. Testování bylo provedeno s asistencí tří testerů, přičemž na každého testera připadá 5 snímků pro každé testované gesto.

Číslo testu	Klidový stav	Posun mapy	Uchopení postavy
1.	Klidový stav	Posun mapy	Uchopení postavy
2.	Klidový stav	Uchopení postavy	Uchopení postavy
3.	Nedefinováno	Posun mapy	Uchopení postavy
4.	Nedefinováno	Posun mapy	Uchopení postavy
5.	Klidový stav	Posun mapy	Uchopení postavy
6.	Klidový stav	Posun mapy	Uchopení postavy
7.	Posun mapy	Posun mapy	Uchopení postavy
8.	Klidový stav	Posun mapy	Klidový stav
9.	Klidový stav	Posun mapy	Posun mapy
10.	Klidový stav	Posun mapy	Uchopení postavy
11.	Klidový stav	Klidový stav	Uchopení postavy
12.	Klidový stav	Posun mapy	Posun mapy
13.	Klidový stav	Uchopení postavy	Uchopení postavy
14.	Nedefinováno	Posun mapy	Uchopení postavy
15.	Klidový stav	Uchopení postavy	Uchopení postavy

Tabulka 1: Výsledky testování detekce a klasifikace gest

Každý sloupec reprezentuje očekávané detekované gesto. Pro rekapitulaci klidový stav představuje otevřenou dlaň, posun mapy je určen vztyčeným ukazováčkem a prostředníčkem a nakonec uchopení postavy je gesto sevření ruky v pěst. V případě, že bylo detekováno chybné gesto je zapsáno gesto, které bylo detekováno na místo očekávaného gesta. Z těchto dat lze určit úspěšnost detekce jednotlivých gest a to:

- Klidový stav : 73%
- Posun mapy : 73%
- Uchopení postavy : 80%
- Celková úspěšnost detekce : 75.5%

Reakční čas mezi pohybem ruky a reakcí projekce na tuto změnu byl změřen na 200ms, což odpovídá frekvenci snímaných snímků pomocí webkamery. Aby bylo možné hovořit o zpracování v reálném čase musela by reakční doba být maximálně 100ms, v ideálním případě pak 50ms. Této odezvy by bylo teoreticky možné dosáhnout pomocí *Cameo* kamery, toto testování ale nebylo možné

provést a tudíž není možné reakční dobu považovat za spolehlivý parametr neboť nebylo provedeno testování limitů samotné aplikace.

8.3 Uživatelské testování

Smyslem uživatelského testování bylo ověření ovládání aplikace uživateli po zadání příslušného úkolu. Testovací scénáře měly za cíl ověřit smysluplnost použitých gest a změřit dobu během které se uživatelé seznámí se základním ovládáním aplikace.

Scénář č.1 – Ovládání hry

V rámci prvního scénáře byl uživatel posazen před detekční prostor, přičemž viděl na projekci. Následně měl gestem ruky uchopit figurku a posunout ji na jiné, předem určené místo v projekci. Dalším testovaným krokem bylo dalším gestem posunutí mapy. Testovanému uživateli byl dopředu předán seznam používaných gest a popsán způsob jejich použití. Výstupem tohoto testování je celková doba po kterou uživatel plnil zadané úkoly, dále pak časy provedení jednotlivých úkolů a nakonec zpětná reakce uživatele.

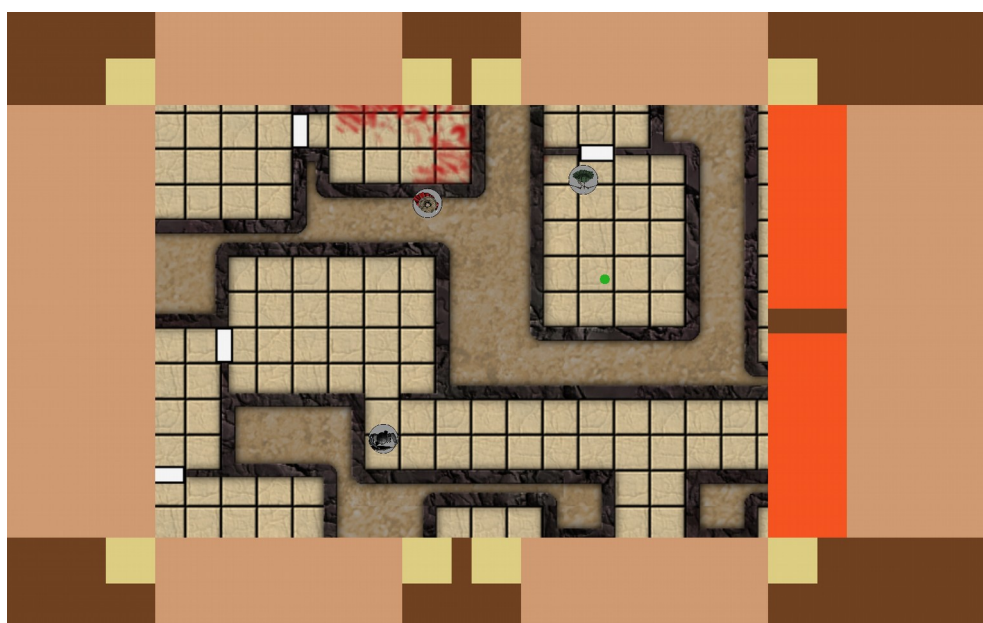
Na ilustracích 21 až 23 jsou znázorněny jednotlivé kroky, které měli být provedeny během tohoto scénáře. Sestava byla předem zkalibrována. Kalibrace jako taková je součástí druhého scénáře. Na ilustraci č. 21 je vidět počáteční uspořádání projekce před začátkem testování. Druhá ilustrace (č. 22) ukazuje možnou změnu pozice postavy, v tomto případě přemístění postavy lovce ze spodní poloviny projekční plochy do horní poloviny. Na posledním snímku (č. 23) je vidět posunutí mapy.



Ilustrace 21: Počáteční stav projekce před úkolem pohybu s postavou



Ilustrace 22: Projekce po posunutí figurky lovce



Ilustrace 23: Projekce po posunutí mapy

Výsledky testování tohoto scénáře jsou uvedeny v následující tabulce a to včetně zpětné vazby od uživatelů kteří prováděli testování. Testování se účastnili lidé ve věkovém rozmezí mezi 18 a 50 lety. Žádný z těchto účastníků předtím nepracoval s podobnou technologií bezdotykového ovládání.

Tester	Celk. Doba[s]	Úkol č. 1[s]	Úkol č. 2[s]	Zpětná vazba
1.	160	60	60	Občas neregistruje malou ruku, naučit se jak nastavit ruku chce praxi.
2.	80	35	35	Dobré, jen projekce se trochu seká.
3.	60	20	20	Detekce funguje skvěle.
4.	140	45	80	Funguje, ale nalézt okraje detekčního prostoru je problematické.
5.	100	38	45	Je složité soustředit se na projekci a ne na tvorbu gest.

Tabulka 2: Uživatelské testování scénáře č.1

Všem účastníkům testování trvalo delší dobu, než se zorientovali v ovládání promítaného obrazu. Většina měla velké problémy s prací s odděleným detekčním a projekčním prostorem a žádný z nich by neměl zájem s touto aplikací v současném stavu pracovat. Přesto mělo toto testování i pozitivní stránku a to tu, že všichni účastníci dříve nebo později požadovaný úkol bez další pomoci splnili.

Scénář č.2 – kalibrace

Cílem druhého scénáře bylo provedení kalibrace sestavy. Uživatel byl opět posazen před detekční prostor a viděl na projekci. Následně měl za cíl provést kalibraci dle předepsaného postupu. Následně po vložení figurky bylo úkolem provést přesun postavy na určené místo na mapě. Uživatel opět dostal připravený seznam gest s popisem jejich použití. Výstupem tohoto testování je opět doba během které se uživateli podařilo splnit zadaný úkol a jeho jednotlivé podčásti. Nakonec byla zapsána zpětná vazba z pohledu uživatele.

Uživatelé postupovali dle postupu kalibrace uvedeného v kapitole . Při pohybu s postavou byla práce obdobná jako na snímcích 21 a 22. Testování tohoto scénáře se zúčastnila jiná skupina lidí nežli prvního scénáře. Jednalo se tedy opět skupinu lidí, kteří s předloženou aplikací přišli do styku poprvé.

Tester	Celk. Doba[s]	Úkol č. 1[s]	Úkol č. 2[s]	Zpětná vazba
1.	/	300	/	Nereaguje pokud nedržím ruku vodorovně se zemí v poměrně nepohodlné pozici.
2.	160	120	20	Nejsou jasné hranice prostoru, který ještě zabírá kamera.
3.	280	240	35	Funkční, jen nefunguje pro levou ruku.
4.	300	255	35	Pokud omylem pohnu s podložkou, kalibrace se nepodaří, ale jinak vše funguje.
5.	250	200	35	OK.

Tabulka 3: Uživatelské testování scénáře č. 2

Všem účastníkům testování byl vysvětlen postup kalibrace a všem se nakonec podařilo provést kalibraci. Největším problémem při kalibraci byla nutnost opakovat kalibraci při pomalé reakci na změnu pozice právě kalibrovaného bodu. V případě dokončení kalibrace bez opakování pokynů bylo pro většinu účastníků již jednoduché přesunout promítanou figurku z jedné pozice na mapě do libovolné jiné.

8.4 Zhodnocení výsledků

V rámci testování byla otestována klasifikace detekovaných gest, přičemž úspěšnost detekce jednotlivých gest je 73% pro klidový stav a posun mapy, 80% pak pro detekci uchopení postavy. Vzniklá chyba klasifikace je způsobena problematickou detekcí jednotlivých prstů, které mohou být u každého člověka jinak rozloženy.

V rámci uživatelského testování byly předloženy dva scénáře každý o dvou úkolech, podle kterých bylo testování prováděno. V prvním z těchto scénářů se po uživatelích požadovalo uchopení postav, jejich přesunutí na mapě a následné položení zpět na mapu. Druhým úkolem prvního scénáře bylo posunutí mapy na které jsou postavy položeny. Druhým scénářem bylo provedení kalibrace sestavy a následně provedení prvního úkolu z předchozího scénáře.

Z výsledků obou scénářů je patrné, že aplikaci je možné reálně použít při skutečné hře, pokud bude možné sjednotit projekční prostor s detekčním, což ve většině případů způsobovalo značné problémy při plnění jednotlivých úkolů ve scénářích.

9 Diskuze a budoucí práce

Z výsledků testování vyplývá, že aplikaci by bylo možné použít po sjednocení projekčního a detekčního prostoru, tedy s použitím kvalitnější – skutečné infračervené kamery.

Celá aplikace je navržena a implementována tak, aby umožňovala úpravu a rozšíření ve všech hlavních aspektech. V případě detekce se jedná o přidání dalších gest, čehož lze dosáhnout jednoduchým přidáním položek do příslušných datových struktur a určením prstů které se příslušného gesta účastní. Taktéž je možné osamostatnit detekční modul a implementovat jej jako nástroj pro spolupráci s operačním systémem a ovládání pouze gesty.

V případě projekce je prosto pro rozšíření mnohem variabilnější. Pokud projekci rozdělíme na část s rozhraním a na část s mapou, pak je možné uvést několik příkladů jak vylepšit jedno či druhé. U rozhraní je možné vylepšit kvalitu promítaného rozhraní přidáním textur nebo animací, taktéž je možné důležité informace pro uživatele zobrazovat přímo na obrazovce (např.: návod na kalibraci). Všechny budoucí úpravy promítaného rozhraní by měli zlepšovat uživatelskou zkušenost při práci s implementovanou aplikací.

Dále je možné přidat interakci na mapě jako takové. Zde je velké množství možností jak zlepšit uživatelskou přívětivost, případně upravit rozhraní. Mezi ty by určitě měl patřit editor mapy, který na texturu mapy umístí přidané informaci o pozici stěn a následně zajistit projekci s ohledem na viditelnost prostředí a dohled charakterů, případně přidání pastí a dalších interaktivních prvků hry. Taktéž je možné přidat osvětlení ze svítidel které mohou mít charaktery ve vybavení, možnost přidání ovládání globálního osvětlení jak mapy tak i prostředí.

10 Závěr

Cílem této diplomové práce bylo navržení a implementace aplikace interaktivního uživatelského rozhraní pro zvolenou stolní hru. Interaktivita je v této práci zajištěna detekcí a klasifikací gest rukou, které snímá infračervená kamera. Jako prezentovaná hra byla zvolena stolní hra na hrdiny *Dungeons & Dragons*.

Před návrhem a implementací promítaného rozhraní bylo nutné zajistit průzkum podobných již existujících aplikací. Některé z nich jsou uvedeny a popsány v kapitole 2. Dále bylo třeba vytyčit matematický model kamery a její kalibrace vůči projektoru, což bylo popsáno v kapitole 3. Po umístění a kalibraci kamery bylo nutné zjistit teoretický postup pro detekci, klasifikaci a sledování gest rukou, kterými uživatelé hry ovládají. Tyto algoritmy a postupy byly popsány v kapitole 4. Po získání těchto teoretických znalostí bylo možné přistoupit k samotnému návrhu rozhraní pro zvolenou hru. Tímto návrhem se zabývala kapitola 6, ve které byl navržen styl projekce i způsob interakce uživatelů s promítanou hrou.

Se základem v návrhu byla vyvinuta aplikace interaktivní projekce. Nutným hardwarovým vybavením pro aplikaci je projektor a infračervená kamera, která snímá projekční prostor. Vytvořená aplikace umožňuje promítání mapy na kterou lze zobrazit herní figurky. Uživatelé, hráči, mohou s mapou i figurkami pohybovat jednoduchými gesty. Z technických důvodů nebylo možné dokončit práci v původním plánovaném rozložení, tedy se sjednoceným projekčním a detekčním prostorem. Aplikaci je nicméně možno použít při rozdělení těchto prostorů.

Na implementované aplikaci byly provedeny uživatelské testy, které poskytly cennou zpětnou vazbu pro budoucí rozšíření této práce. Kromě uživatelské přívětivosti promítané hry byla taktéž testována úspěšnost detekce vybraných gest. Popis a způsob testování byl prezentován v kapitole 8. Spolehlivost detekce a klasifikace byla určena na pouhých 75% při použití obyčejné webkamery, neboť kvalitní infračervená kamera nebyla k dispozici.

Jako budoucí vylepšení této práce by mohlo být použití kvalitní infračervené kamery a tudíž i skutečné sjednocení projekčního a detekčního prostoru, případně zlepšení spolehlivosti detekce a klasifikace gest. Při pohledu na projekční část je možné přidávat nové textury do rozhraní, zvětšení variability promítaných figurek. Zajímavým přínosem pro implementovanou hru by mohlo být vytvoření editoru map, ve kterém by byla přidávána informace o stěnách, osvětlení, případně další vlastnosti, které by se následně projevovali do projekce.

Zdroje a literatura

- [1] HARTLEY, Richard a Andrew ZISSERMAN. *Multiple view geometry*. 2. Cambridge: Cambridge University Press 2000, 2003. ISBN 978-0-521-54051-3.
- [2] ZHENGYOU ZHANG. Flexible camera calibration by viewing a plane from unknown orientations. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999, **1**: 666-673 vol.1. DOI: 10.1109/ICCV.1999.791289. ISBN 0-7695-0164-8. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=791289>
- [3] KAEWTRAKULPONG, Pakorn; BOWDEN, Richard. An improved adaptive background mixture model for real-time tracking with shadow detection. In: *Video-based surveillance systems*. Springer US, 2002. p. 135-144.
- [4] ZIVKOVIC, Zoran. Improved adaptive Gaussian mixture model for background subtraction. In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004. p. 28-31.
- [5] GODBEHERE, A. B., A. MATSUKAWA a K. GOLDBERG. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. *2012 American Control Conference (ACC)* [online]. IEEE, 2012, (1): 4305-4312 [cit. 2016-01-03]. DOI: 10.1109/ACC.2012.6315174. ISBN 978-1-4577-1096-4. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6315174>
- [6] DHAWAN, Amiraj; HONRAO, Vipul. Implementation of Hand Detection based Techniques for Human Computer Interaction. *arXiv preprint arXiv:1312.7560*, 2013.
- [7] KANG, Sung Kwan, Mi Young NAM a Phill Kyu RHEE. Color Based Hand and Finger Detection Technology for User Interaction: treaties and international agreements registered or filed and recorded with the Secretariat of the United Nations. *2008 International Conference on Convergence and Hybrid Information Technology*. IEEE, 2008, : 229-236. DOI: 10.1109/ICHIT.2008.292. ISBN 978-0-7695-3328-5. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4622829>
- [8] CHEN, Zhi-hua, Jung-Tae KIM, Jianning LIANG, Jing ZHANG a Yu-Bo YUAN. Real-Time Hand Gesture Recognition Using Finger Segmentation. *The Scientific World Journal*. 2014, **2014**. DOI: 10.1155/2014/267872. ISSN 2356-6140. Dostupné také z: <http://www.hindawi.com/journals/tswj/2014/267872/>
- [9] MA, Ziyang a Enhua WU. Real-time and robust hand tracking with a single depth camera. *The Visual Computer*. 2014, **30**(10): 1133-1144. DOI: 10.1007/s00371-013-0894-1. ISSN 0178-2789. Dostupné také z: <http://link.springer.com/10.1007/s00371-013-0894-1>

- [10] PARK, M. S., et al. Hand detection and tracking using depth and color information. In: *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'12)*. 2012. p. 779-785.
- [11] SRIDHAR, Srinath, et al. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015. p. 3213-3221.
- [12] GILLIAN a PARADISO. The Gesture Recognition Toolkit. *The Journal of Machine Learning Research*. 2014, **15**(1). ISSN 1532-4435.
- [13] BABENKO, Boris; YANG, Ming-Hsuan; BELONGIE, Serge. Visual tracking with online multiple instance learning. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009. p. 983-990.
- [14] BENFOLD, Ben; REID, Ian. Stable multi-target tracking in real-time surveillance video. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011. p. 3457-3464.
- [15] CHOI, Wongun; SAVARESE, Silvio. Multiple target tracking in world coordinate with single, minimally calibrated camera. In: *Computer Vision–ECCV 2010*. Springer Berlin Heidelberg, 2010. p. 553-567.
- [16] TOMASI, Carlo; KANADE, Takeo. *Detection and tracking of point features*. Pittsburgh: School of Computer Science, Carnegie Mellon Univ., 1991.
- [17] STIRMAN, Jeffrey N., et al. A multispectral optical illumination system with precise spatiotemporal control for the manipulation of optogenetic reagents. *Nature protocols*, 2012, 7.2: 207-220.
- [18] HUGHES, James. *Velká obrazová všeobecná encyklopedie*. Nové, aktualiz. vyd. Praha: Svojtka, 2007. ISBN 80-735-2823-1.
- [19] Kineticspace: Training, Analyzing and Recognizing 3D Gestures. *Google Project Hosting* [online]. 2012 [cit. 2016-01-11]. Dostupné z: <https://code.google.com/p/kineticspace/>
- [20] Interactive visual display system: GestureFX. GESTURETEK. *GestureTek* [online]. 2015 [cit. 2016-01-11]. Dostupné z: <http://www.gesturetek.com/gesturefx/introduction.php>
- [21] Interactive Floor and Wall Displays. PO-MOTION. *Po-motion* [online]. 2015 [cit. 2016-01-11]. Dostupné z: <http://www.po-motion.com/>
- [22] MagixTable. TOUCH MAGIX. *Touch Magix: Interactive Wall, Interactive Projector, Interactive Floor, Interactive Table* [online]. 2016 [cit. 2016-01-11]. Dostupné z: <http://www.touchmagix.com/products/magixtable>

- [23] E-Table full. E-TABLE TM. *E-Table* [online]. 2015 [cit. 2016-01-11]. Dostupné z: <http://www.e-table-interactive.com/solutions/etable.html>
- [24] ViTech Product & Service Limited. *ViTech Product* [online]. 2012 [cit. 2016-01-11]. Dostupné z: <http://www.vi-tech.com.hk/>
- [25] HandVu:Hand Gesture Recognition. *HandVu:Hand Gesture Recognition* [online]. 2011 [cit. 2016-01-11]. Dostupné z: <http://www.movesinstitute.org/~kolsch/HandVu/HandVu.html>
- [26] PEDERSOLI, Fabrizio, Sergio BENINI, Nicola ADAMI a Riccardo LEONARDI. XKin: an open source framework for hand pose and gesture recognition using kinect. *The Visual Computer*. 2014, **30**(10), 1107-1122. DOI: 10.1007/s00371-014-0921-x. ISSN 0178-2789. Dostupné také z: <http://link.springer.com/10.1007/s00371-014-0921-x>
- [27] Interactive Restaurant Tables. *TheCoolCollector* [online]. 2012 [cit. 2016-01-11]. Dostupné z: <http://thecoolcollector.net/diy/interactive-restaurant-tables/>
- [28] *RapidXml* [online]. 2009. Londýn: 2009 Marcin Kalicinski [cit. 2016-02-21]. Dostupné z: <http://rapidxml.sourceforge.net/index.htm>
- [29] CACEK, Pavel. 2015. *Tvorba panoramatických fotografií*. Brno. Diplomová práce. FIT VUT v Brně. Vedoucí práce Ing. Kamil Behůň.

Příloha A - Obsah přiloženého DVD

- | | |
|---------------------|--|
| • plakat.pdf | Ilustrační plakát s výsledky této práce |
| • dokumentace.pdf | Tato práce ve formátu pdf |
| • source/ | Zdrojové soubory implementované aplikace |
| • source/readme.txt | Soubor s instrukcemi pro překlad a ovládání aplikace |
| • text/ | Složka se zdrojovými soubory k této práci i k ilustračnímu plakátu |