VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

PROMÍTANÉ UŽIVATELSKÉ ROZHRANÍ – DESKOVÁ HRA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE PETR VIZINA
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

PROMÍTANÉ UŽIVATELSKÉ ROZHRANÍ - DESKOVÁ HRA

PROJECTED USER INTERFACE – BOARD GAME

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE PETR VIZINA
AUTHOR

VEDOUCÍ PRÁCE Ing. JIŘÍ ZAHRÁDKA
SUPERVISOR

BRNO 2015

Abstrakt

Tato práce se zabývá studií možností a použití promítaných uživatelských rozhraní, jež jsou založeny na systému kamera-projektor. Cílem je vytvoření funkční aplikace, využívající pro zobrazení uživatelského rozhraní projektor a interakce s uživatelem je zaznamenána na základě obrazu z kamery. Vytvořená aplikace má být simulací deskové hry „Mlýn“. Výsledkem je fungující aplikace deskové hry, která využívá vytvořené nástroje pro kalibraci systému kamera-projektor s využitím fyzického kalibračního vzoru a detekci rukou v obraze.

Abstract

This paper deals with study of possibilities and use of projected user interfaces, which are based on a camera-projector system. The purpose is to create functional application, that using projector to display user interface of application and camera for capturing user interaction with the projected interface. Created application is supposed to be a simulation of board game called „Mill“. The result is a functional application of board game, that utilizes tools for calibration of the camera-projector system with physical calibration pattern and hand detection in captured scene.

Klíčová slova

kamera, projektor, kalibrace, homografie, desková hra, mlýn, detekce rukou, java, opencv

Keywords

camera, projector, calibration, homography, board game, three men‘s morris, hand detection, java, opencv

Citace

Vizina Petr: Promítané uživatelské rozhraní – desková hra, bakalářská práce, Brno, FIT VUT v Brně, 2015

Promítané uživatelské rozhraní – desková hra

Prohlášení

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

……
Petr Vizina
20. 5. 2015

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Jiří Zahrádkovi za ochotnou odbornou pomoc a užitečné rady k řešení projektu.

© Petr Vizina, 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

Obsah 1
1 Úvod 3
2 Snímání obrazu 4
2.1 Počítačové vidění 4
2.2 Zachycení obrazu 4
2.3 Zpracování obrazu 5
2.4 Porozumění obrazu 5
3 Promítané uživatelské rozhraní 6
3.1 Aplikace využívající promítané rozhraní 6
3.2 Interaktivní hry 7
4 Detekce rukou v obrazu 12
4.1 OpenCV 12
4.2 JavaCV a EmguCV 14
4.3 Microsoft Kinect 14
5 Kalibrace kamery a projektoru 16
5.1 Kalibrace kamery 16
5.2 Kalibrace projektoru 18
6 Desková hra Mlýn 20
6.1 Pravidla hry Mlýn 20
6.2 Varianty hry 21
7 Návrh aplikace 22
7.1 Hardwarová sestava 22
7.2 GUI a ovládání aplikace 23
7.3 Aplikace 24
8 Implementace 27
8.1 Aplikace 27
8.2 Detekce rukou a prstů 28
8.3 Kalibrace kamery 32
8.4 Kalibrace projektoru 33
9 Testování a validace 34
9.1 Testování rozpoznání detekce ruky 34
9.2 Kontrola kalibrace kamery 36
9.3 Kontrola homografie 37
9.4 Testování aplikace 37

9.5 Možné rozšíření 38
10 Závěr 39
Literatura 40
Seznam příloh 43
Příloha 1. Manuál – Spuštění aplikace 44
Příloha 2. Manuál – Ovládání aplikace 46
Příloha 3. DVD 49

1 Úvod

V současné době s rychle se rozvíjejícím počítačovým odvětvím se stále více společností snaží prosadit a zaujmout novými nápady. Klasické vstupní periferie pro ovládání počítačů, klávesnice a myš, jsou zde s námi již půl století a nesou sebou řadu nevýhod. Obtížná je zejména interakce více uživatelů s počítačem. Po nástupu dotykového ovládání u mobilních telefonů se společnosti snaží vyvinout nový způsob i pro ovládání počítačů, jež by nahradil myš s klávesnicí. Hlavní zaměření zde směřuje právě na tzv. *Motion sensing input devices*, což lze přeložit jako vstupní zařízení se snímáním pohybu a *Projected user interface*, to znamená promítané uživatelské rozhraní, viz. kapitola 3.

Úspěšným se zde stal Microsoft se svým doplňkem k herní konzoli Xbox 360 s názvem Kinect. Jedná se o vstupní zařízení obsahující kameru, která snímá prostor před sebou a detekuje uživatele včetně jeho gest. Uživatelé poté mohou provádět interakci za pomocí rukou nebo také celého těla, případně může program ovládat i více uživatelů najednou. Uživatelé ocení tuto možnost zejména při hraní her, kdy je využití gest a přirozených pohybů vtáhnou více do děje. Spolu s tímto způsobem ovládání se nabízí provázání právě s promítáním uživatelského rozhraní programu nebo hry. Samotné promítání poté může probíhat na stěnu, desku stolu či jiný povrch vhodný pro konkrétní aplikaci.

Cílem této práce je implementace deskové hry Mlýn. Hra bude využívat systému kamera-projektor, kdy je použit dataprojektor pro zobrazení hrací plochy a kamera pro snímání promítané hrací desky. Snímání hry pomocí kamery umožní uživateli ovládání hry za pomocí gest rukou viz. 7.2.

Další kapitola přibližuje obor počítačového vidění. Třetí kapitola je zaměřena na promítané uživatelské rozhraní a popis již existujících aplikací založených na bázi kamera-projektor. Čtvrtá kapitola obsahuje popis dostupných knihoven, určených pro zpracování obrazu. Část pátá obsahuje informace o kalibraci systému kamera-projektor. Informace o implementované hře Mlýn jsou poskytnuty v šesté kapitole. Kapitola sedmá již pojednává o samotném návrhu aplikace a uživatelského rozhraní. Implementace aplikace je popsána v osmé kapitole a o fázi testování pojednává kapitola devátá.

2 Snímání obrazu

Téma promítaného uživatelského rozhraní v sobě kombinuje dvě odlišné skupiny oblastí. Následující část se bude snažit představit první z nich a tou je snímání obrazu.

2.1 Počítačové vidění

Jedná se o odvětví výpočetní techniky využívající metody zachycení, zpracování, analýzy a porozumění obrazu reálného světa. Úkolem je zpracovat kamerou zachycený obraz tak, aby byl počítač schopen obrazu porozumět a na základě zachycené akce provést patřičnou událost.

[1] Vzhledem k obtížnosti zpracování obrazových dat se obor začíná formovat až na konci sedmdesátých let 20. století. Vznikají velmi specifické metody pro dané úlohy, které nemají obecné využití. S tím, jak se obor vyvíjí, se stále více metod dostává do komerčních produktů, kde tvoří část celku řešícího komplexní problémy.

Na co se používá počítačové vidění [2]:

- detekce událostí (detekce chodců v automobilu, kamerově střežené objekty)
- detekce, segmentace, popisování, rozpoznávání (porozumění vnímání) objektů
- součást řídicích systémů (autonomní vozidla, průmyslový robot)
- modelování objektů nebo okolního světa (obrazová kontrola kvality výrobku v průmyslu, analýza lékařských obrazů)
- organizace informací (např. třídění fotografií)
- interakce mezi člověkem a strojem (ovládání počítačové hry)

2.2 Zachycení obrazu

K zachycení obrazu reálného světa se obvykle využívá kamery. K dispozici jsou dva druhy kamer.

- RGB kamera
- IR (infračervená) kamera

Nejrozšířenější je kamera typu RGB, která snímá záření z viditelné oblasti světelného spektra dopadající na snímač. Nevýhodou RGB kamery v počítačovém vidění je neschopnost pořídit snímek bez okolního osvětlení. Zde nachází výhodu infračervená kamera, která snímá pouze záření infračervené, jež vyzařuje každý objekt s teplotou vyšší než je absolutní nula[1]. Kamerou zachycený obraz není v tomto případě nijak závislý na okolním osvětlení a lze tedy infračervenou kamerou snímat scénu i v noci. Pro počítačové vidění je tato vlastnost zásadní (např. možnost detekce chodců v noci). Ve většině případů se však používá kombinace obou typů kamer (např. Kinect).

[1] Absolutní nula je teplota 0K, což je -273,15°C.

2.3 Zpracování obrazu

Cílem tzv. image processingu, což v překladu znamená zpracování obrazu, je provedení potřebných úprav pořízeného snímku, aby bylo možné provést požadované rozpoznání dat ve snímku. Zpracování probíhá nejčastěji na kamerou pořízené digitální fotografii, jež je uložena jako obrazový soubor. Mezi časté úpravy snímku patří:

- odstranění šumu
- detekce hran
- úprava jasu, kontrastu, expozice
- převod barevného obrazu na stupně šedi
- zostření, rozmazání
- prahování obrazu
- transformace snímku (vodorovné nebo svislé otočení, zvlnění snímku, zmenšení, ořezání)

[obr. 1] Ukázka zpracování snímku. A) Původní snímek. B) Snímek z RGB modelu převeden do stupňů šedi. C) Provedení prahování.

2.4 Porozumění obrazu

Jedná se o klasický problém počítačového vidění, kdy je potřeba určit, zda zachycená data obsahují hledaný objekt nebo nikoliv. Výstupem takové úlohy může být například označení nalezeného objektu ve snímku, získání souřadnic nalezeného objektu nebo pouze odpověď nalezeno, nenalezeno. Proces detekce a klasifikace objektů zatím není zcela automatizovaný [3]. Probíhá na základě programátorem definovaných pravidel. V oboru umělé inteligence, která je s porozuměním obrazu úzce provázáno, je snaha vyvíjet pokročilé učící se algoritmy. Takové algoritmy poté nacházejí uplatnění například v robotice, kde se robot poté sám naučí detekovat nové objekty.

[3] Literatura definuje různé varianty poznávání obrazu, patří mezi ně:

- *Rozpoznání objektů* (také nazýváno klasifikace objektů) – rozpoznání jednoho či více předem specifikovaného objektu.
- *Identifikace* – analýza rozpoznaného objektu, například za účelem identifikace podle obličeje, otisku prstů, nebo přepis psaného písma.
- *Detekce* – obrazová data jsou prohledána pro nalezení určité událostí. Například se může jednat o detekci vozidla.

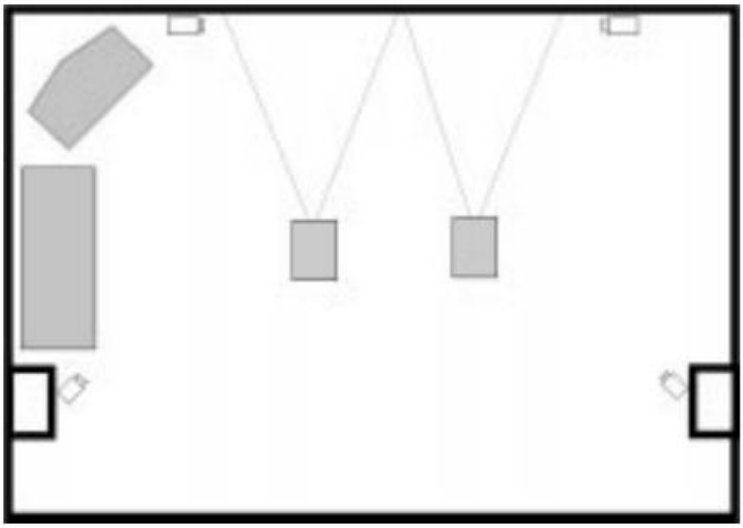
3 Promítané uživatelské rozhraní

Rostoucí dostupnost a také menší rozměry dataprojektorů na dnešním trhu mají za následek hledání jejich širšího využití. Zajímavým jej proti klasickému monitoru činí možnost promítat obraz na každý objekt (např. stěnu, stůl, plátno), pokud to dovolí okolní světelné podmínky. Spojení dataprojektoru a kamery do jednoho celku poté nabízí možnost vytvořit každou plochu interaktivní s tzv. hmatovou odezvou. Aby bylo možné vytvořit díky kameře onu interaktivitu, je potřeba využít znalostí z počítačového vidění (detekce, rozpoznání, sledování). Na základě tohoto principu vznikají jak pro práci užitečné aplikace (interaktivní pracovní stoly) 3.1, tak aplikace určené pro zábavu (hry) 3.2.

3.1 Aplikace využívající promítané rozhraní

The intelligent room project

Jedním z prvních projektů využívající systému kamera-projektor byl *The intelligent room (inteligentní místnost)* z roku 1997 [4]. Tvůrce tohoto projektu se držel myšlenky *enhanced reality*, tzv. vylepšené reality: „Místo vnést do virtuálního světa člověka, se pokusit do reálného světa přenést počítač.“ V interiéru místnosti se podle návrhu nacházejí dva dataprojektory a po stranách projektorů jsou umístěny čtyři kamery, pro snímání interakce uživatele pomocí rukou a pohybových gest.

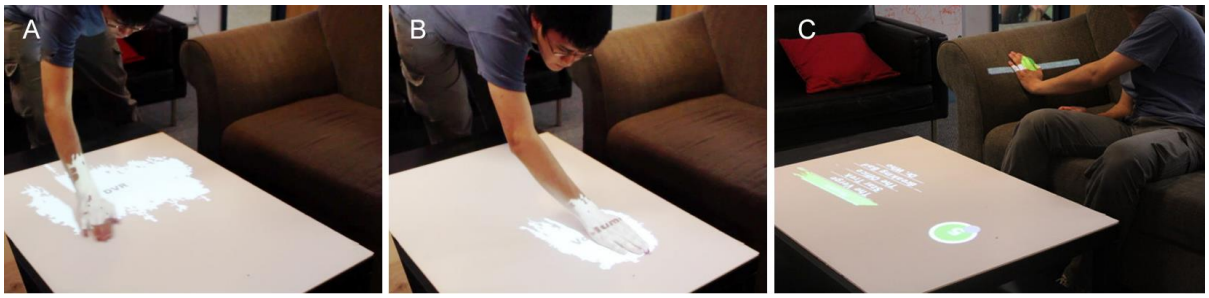


[obr. 2] Návrh inteligentní místnosti obsahující: dva projektory (uprostřed) zavěšené na stropu místnosti a čtyři kamery snímající promítanou plochu. [4]

Systém počítal také s detekcí více uživatelů před projekční plochou. Pro rozpoznání akcí uživatele je využito konečného stavového automatu, jenž znázorňoval možné pohyby uživatele. Boční kamery jsou použity zvláště k detekci dotyku s plochou, na níž je obraz promítán a zpřesnění výpočtu místa, na které uživatel míří rukou. Zajímavostí je také i možnost použití laserového ukazovátka pro ovládání aplikace. K detekci ukazovátka bylo použito jednoduchého prahování obrazu.

WorldKit

Projekt WorldKit [5] má za cíl vytvořit interaktivní plochu na každodenním objektu v domě. Systém je silně inspirován již zmíněným *The inteligent room project*.



[obr. 3] Použití systému WorldKit na povrchu stolu a pohovce. [5]

Jak lze vidět na obrázku [obr. 3], systém uživateli nabízí možnost „namalovat“ interaktivní elementy přímo na promítanou plochu dle potřeby [obr. 4: A, B]. Poté se již na místech objeví „dotykové“ prostředí, ihned připravené pro interakci s uživatelem. Pro snímání prostoru je použito zařízení Kinect [4.3 Microsoft Kinect]. Kinect je namontován přímo na promítacím projektoru, umístěném před projekční plochou. Jeho schopnost tvorby hloubkové mapy je zde využita při detekci dotyku s promítanou plochou.

Limitací systému je v současné době nízké rozlišení pro snímání i promítání obrazu, než by bylo u některých aplikací optimální. Vzhledem k použití pouze jedné kamery a projektoru může být problémem zastínění obrazu uživatelem.

3.2 Interaktivní hry

Pro deskové hry, jež jsou tématem této práce, je promítané rozhraní zajímavým prostředkem, jak uživatele zaujmout. Pomocí kamery a projektoru jsme schopni sloučit prvky reálného světa společně s tím virtuálním a do poněkud „statické“ deskové hry vnést určitou interaktivitu, efekty a prvky rozšířené reality, dále jen AR (Augmented Reality). Použitím projektoru, jež promítá obraz, se také vyhneme potřeby brýlí určených pro AR (např. Oculus rift).

Pro tvorbu rozhraní hry lze využít řadu možností:

1. Rozhraní postavené pouze na softwarovém základě, jež je promítáno na hrací plochu.
2. Kombinace promítaného obrazu a reálných objektů.

Pro druhý případ se může jednat např. o fyzickou herní desku, na kterou se promítají ovládací prvky nebo figurky. Konkrétním případem by mohlo být využití reálné šachovnice pro hru šachy, na kterou by byly promítány figurky. Případně by bylo možné využít reálné objekty pro určení místa, na které má být promítán obraz (např. hra HideOut).

Ovládání takto vytvořené hry je možné pojmout více styly a záleží zde na vývojáři:

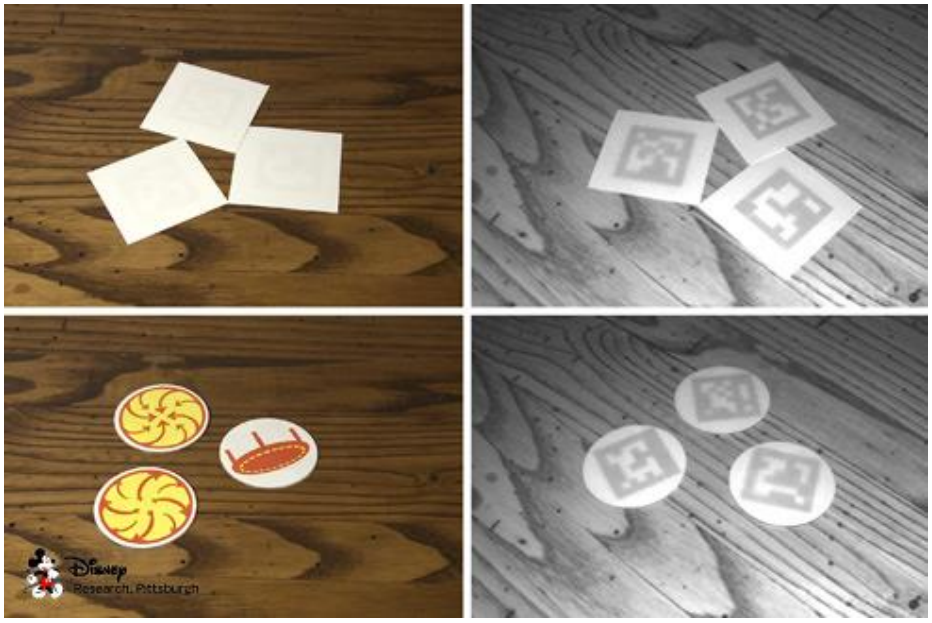
1. Čistě „dotykové“ ovládání, kamera snímá ruce hráče a detekuje jeho gesta (gesto pro posun, označení, …)
2. Ovládání založené na detekci a sledování fyzických objektů. Uživatel v tomto případě přemisťuje reálné předměty po hrací ploše, čímž může hru ovládat.
3. Není vyloučena ani kombinace obou předchozích přístupů.

Hra, využívající kamery a projektoru, může uživateli poskytnout řadu výhod oproti klasické deskové nebo čistě počítačové hře, jež není promítaná projektorem:

- Přirozená a intuitivní interakce, není třeba využití myši, klávesnice.
- Možnost pohodlné hry více hráčů.
- Možnost potenciálně většího obrazu než při použití počítačového monitoru.

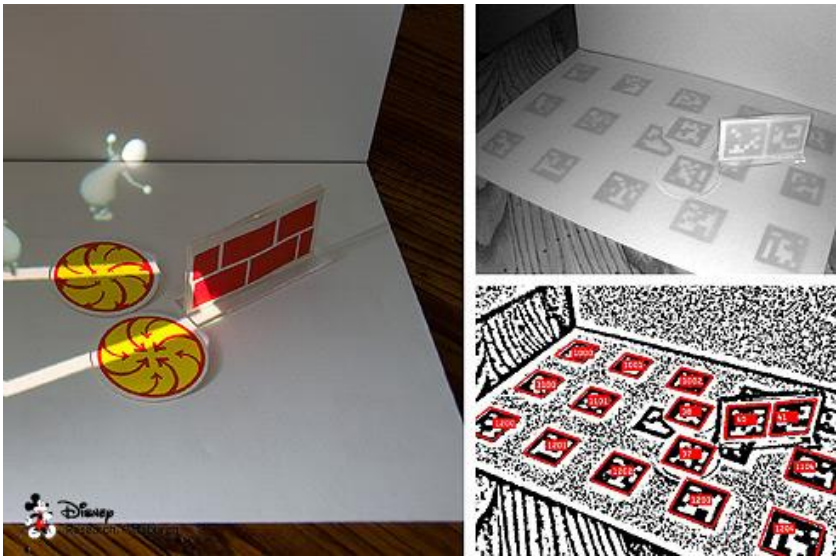
HideOut

„Herní projektor“ HideOut [6] byl představen na konferenci *Tangible, Embedded and Embodied Interaction (Hmatová, vestavěná a ztělesněná interakce)* v roce 2013 společností Disney. Zařízení obsahuje LED projektor a IR kameru. Uživatel má možnost provádět interakci s hrou pomocí fyzických objektů obsahujících natištěnou *značku*. Tisk značky je proveden speciální barvou pohlcující infračervené záření, pro lidské oko je neviditelná, ale IR kamera ji dokáže zachytit. Každá značka nese svůj význam.



[obr. 4] Ovládací prvky s natištěnou značkou. Vlevo snímek z klasické kamery, vpravo záznam z IR kamery. Značky jsou zde viditelné. [6]

Tyto objekty může uživatel umístit na hrací plochu, posouvat s nimi, nebo je odebrat. V závislosti na rozpoznané značce se provede událost odpovídající významu, jež značka nese. Promítnutí patřičného obrazu na značku nebo přidání určité vlastnosti do hry. Projektor s kamerou je v tomto případě umístěn před hrací plochou, která je pokrytá rovněž natištěnými značkami [obr. 5]. Mříž značek na hrací desce má využití při kalibrací projektoru. Jeho kalibrace je prováděna vůči těmto značkám a v závislosti na jejich velikosti a natočení, z níž jsou snímány, lze získat vzájemnou polohu a orientaci mezi hrací deskou a projektorem. Této vlastnosti je využito při promítání prostředí hry, kdy obraz „vystupuje“ z hrací desky a navozuje trojrozměrný dojem.

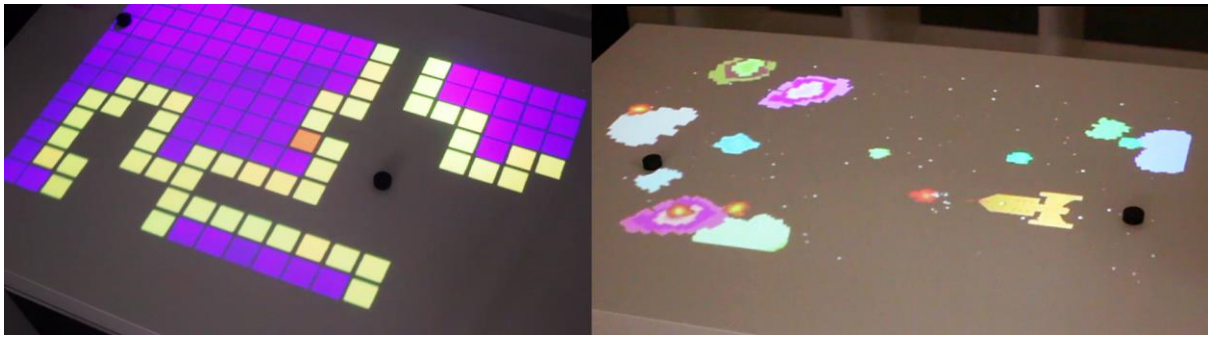


[obr. 5] Herní deska vlevo společně s umístěnými značkami a promítáním. Vpravo snímek plochy z IR kamery a zpracovaný snímek s detekovanámi značkami. [6]

Lichtspiel

[7] Jedná se o rozhraní pro využití u interaktivní deskové hry. Cílem projektu bylo spojit pocit ovládání klasické deskové hry společně s dynamickými možnostmi počítačových her. Hrací deska je promítána na bílou plochu (např. světlý stůl) pomocí malého LED projektoru. Hra využívá klasickou barevnou kameru, popis uvádí možnost použití webkamery nebo také Raspberry Pi kamery. Kamera je umístěna společně s projektorem před promítanou plochou na flexibilním 60cm dlouhém ramenu, hra tedy využívá přední projekce.

Ovládání hry probíhá pomocí tzv. žetonů. Jedná se o malé, černé fyzické objekty, kterými uživatel může pohybovat po hrací ploše. Při posunu žetonu je hře poslána jeho nová pozice ve 2D souřadnicích *x*, *y*. Interakce pomocí gest rukou není v projektu implementována. Zpracování vstupního obrazu z kamery, sledování žetonů na hrací ploše a rozpoznání aktuálního stavu hry má na starost knihovna OpenCV.



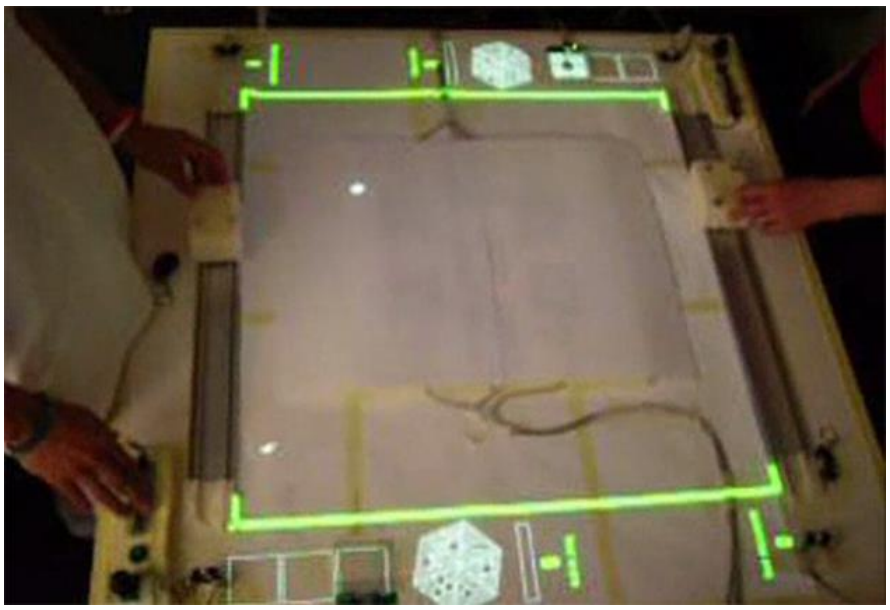
[obr. 6] Implementované hry. Vpravo například hra Asteroids. Žetony zde slouží jako ovládací prvky. [7]

Na obrázku [obr. 6] lze vidět dvě hry, jež jsou v projektu implementovány. Do hry se mohou zapojit dva hráči. Každý hráč má k dispozici vlastní žeton, kterým může pohybovat. Například u hry Asteroids jeden hráč pomocí žetonu pohybuje lodí a druhý svým žetonem určuje cíl střelby. Zdrojové kódy projektu jsou zveřejněny a k dispozici ke stažení na serveru GitHub[2].

[2] Dostupné na adrese: https://github.com/marwie/Lichtspiel-Interactive-Digital-Board-Game

Phong

Hra Phong [8] je dalším příkladem použití systému projektor-kamera. Implementace využívá opět přední projekce obrazu na desku stolu, spolu se snímáním scény klasickou kamerou. Návrh hry je postaven na principu prolnutí elementů virtuálního a reálného světa. Ovládání probíhá pomocí na kolejích připevněných posuvných kvádrů. Aby bylo možné kvádr sledovat kamerou, jsou všechny jeho rohy polepeny značkami. V zachyceném obraze se poté detekují tyto značky. Díky jejich využití je také možné určit přesné hranice a velikost kvádru. Okraje hrací plochy jsou spolu s hracím pukem promítány projektorem. Hra je určena pro dva hráče a neobsahuje možnost interakce pomocí rukou.

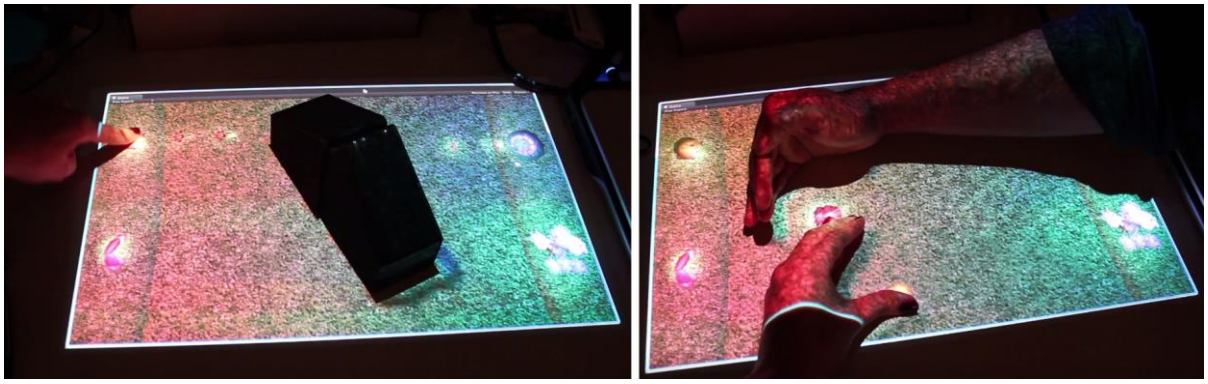


[obr. 7] Ukázky hry Phong. [8]

Dota

Jedná se o experiment dvojice vývojářů Chrise Rojase a Sterlinga Orstena [9]. Vývojáři v tomto případě využili piko projektor, jež je umístěn před promítanou plochou. Obraz je promítán na desku stolu. Zajímavým tento projekt činí využití *Intel Creative kamery* místo klasické webkamery. Ta obsahuje stejně, jako například zařízení Kinect, dvě kamery. Jednou z nich je klasická barevná kamera a druhou je IR kamera schopná detekce hloubkové mapy prostředí.

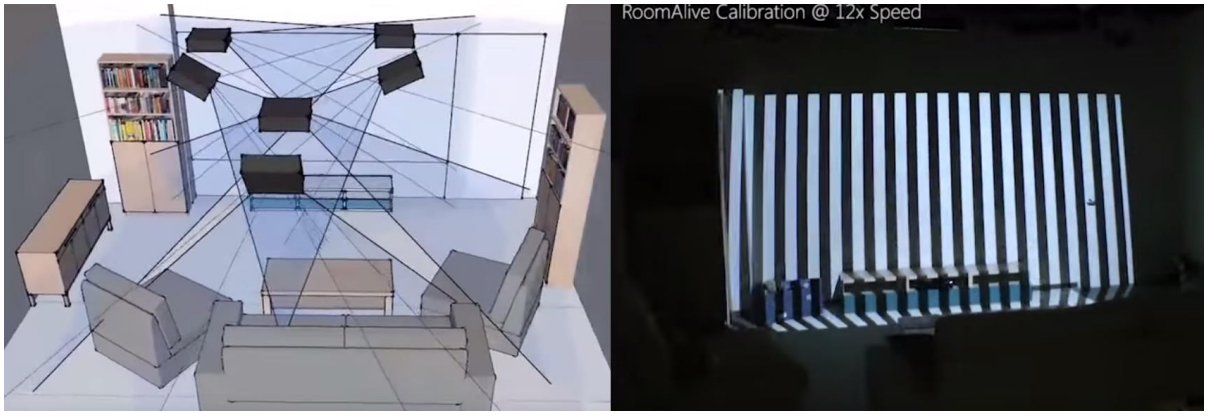
Proti předchozím příkladům má uživatel možnost hru ovládat pomocí rukou. Promítané prostředí hry obsahuje ovládací prvky, na které má uživatel možnost „kliknout“. Pozice rukou je sledována pomocí kamery a právě pro detekci onoho „kliknutí“, je zde využita získaná hloubková mapa scény. Podle hloubkové mapy lze určit, zda se uživatel již dotýká desky stolu nebo nikoliv. Kromě interakce rukou nabízí hra možnost využití fyzických objektů, které jsou po umístění na hrací plochu detekovány a slouží ve hře jako překážky, které je potřeba obejít. Překážka může být vytvořena také pomocí rukou.



[obr. 8] Ukázka hry Dota. Vlevo je zachycena detekce kliknutí a uprostřed vložená překážka ve formě reálného objektu. Vpravo je překážka vytvořena pomocí rukou, herní postava v tomto případě obchází překážku. [9]

RoomAlive

V tuto chvíli jde o koncept [10] společnosti Microsoft. Jedná se spíše o demonstraci možností her založených na systémech s projektorem a kamerou. Vytvořený systém využívá pro projekci 6 projektorů, které pokrývají svým obrazem celou plochu pokoje. Prostor pokoje je naskenován pomocí IR kamery zařízení Kinect, jež vytvoří hloubkovou mapu pokoje. Na jejím základě je poté postaven 3D model místnosti včetně nábytku. V modelu je také provedena detekce orientace povrchů (svislá nebo vodorovná). Na základě této mapy je poté celá místnost pokryta obrazem herního prostředí. Systém dokáže v záběru detekovat i více osob.



[obr. 9] RoomAlive. Vlevo je znázorněno umístění projektorů spolu s kamerami po místnosti. Vpravo je ukázka vzoru pro kalibraci systému, jež se dokáže zkalibrovat automaticky. [10]

[obr. 10] Vlevo rozpoznané plochy pokoje. Vpravo promítaná obraz a interakce uživatele. [10]

4 Detekce rukou v obrazu

Zadáním tématu je vytvoření deskové hry, s níž bude možné provádět interakci za pomocí rukou. Následující kapitola se tedy zabývá možnostmi a dostupnými nástroji pro zpracování obrazu a detekci objektů ve videosekvenci. Použití těchto specializovaných knihoven zjednoduší a zrychlí implementaci.

4.1 OpenCV

Celým svým názvem *Open Source Computer Vision* [11] je volně šiřitelná knihovna (pod BSD licencí) pro zpracování obrazu. Její výhodou je snadnost užití a i programátor, který nemá zkušenosti v tomto oboru, velmi rychle přijde na to, jak dostupné nástroje využívat. Zásluhu na tom nese také výborná dokumentace celého projektu se spoustou příkladů a ukázek. Nesporným pozitivem a důvodem, proč zvolit tuto knihovnu, je její multiplatformnost. Knihovna je k dispozici pro všechny majoritní počítačové systémy Windows, Linux a OS X, spolu s mobilními systémy Android a iOS. Bohaté jsou i možnosti, co se týká programovacích jazyků, které mají podporu OpenCV. Dříve podporované pouze v jazycích C a C++, plynoucí z implementace samotné knihovny OpenCV v těchto jazycích, jsou dnes doplněny o jazyky Java a Python.

Téma této práce nabízí možné využití OpenCV v situacích:

1. Detekce hran, šachovnice v obraze.
2. Kalibrace kamery a projektoru
3. Rozhraní pro připojení k webkameře.
4. Uložení, načtení obrazových souborů, včetně videa.
5. Zpracování obrazu (rozmazání, eroze, dilatace)

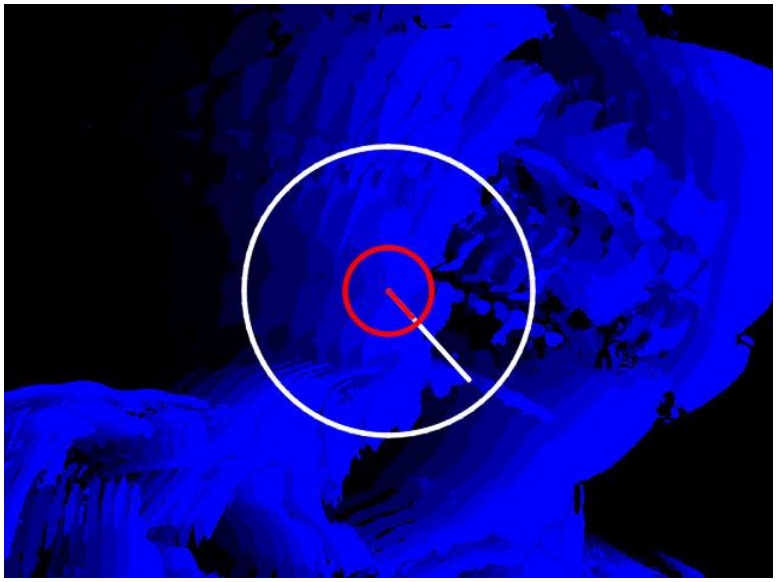
Pro případ implementace detekce ruky, nabízí samotné OpenCV více možností přístupů k docílení požadovaného výsledku.

Match template

Dostupná možnost pro detekci rukou, ale také jiných objektů je využití *match template* [12] (hledání shody). Algoritmus pracuje na nalezení shody mezi vstupním snímkem a šablonou, kterou má ve vstupních datech naleznout. Po vstupním snímku se v cyklu posouvá tzv. hledací okno, které musí být větší než šablona. V tomto okně se poté detekují shodné body se šablonou. Oblast s největším počtem shodných bodů je také nejlepší shodou s hledanou šablonou.

Motion template

Funkce pro detekci pohybujících se objektů ve videosekvenci. Po sobě jdoucí snímky jsou vůči sobě porovnány a místo s rozeznanou změnou v obraze se do snímku vykreslí barvou. Snímek kromě aktuálního místa změny, má zaznamenány i změny předchozí, jejichž barva postupně slábne. Podle „histogramu“ změn lze určit směr pohybu objektu.



[obr. 11] Detekce pohybu hlavy pomocí motion template. [13]

Haarovy kaskády

Jednou z možností je implementace rozpoznávání pomocí Haarových kaskád [14]. Detekce je zde založená pomocí trénování detektoru, jejímž výstupem je soubor typu xml, obsahující popis objektu. Je potřeba mít k dispozici tzv. positive samples (Pozitivní snímky) a negative samples (negativní snímky). Pozitivní snímky musí obsahovat ořezaný objekt, který máme v plánu detekovat. Naopak negativní snímky nesmí požadovaný objekt ve svých datech obsahovat. Pro detekci jednoduššího objektu je možné si vystačit s kombinací 40 pozitivních a 600 negativních snímků [14]. Pro složitější úlohy jako rozpoznání obličeje toto číslo roste. Výstupem detekce je návratová hodnota „1“ pokud snímek obsahuje daný objekt, v opačném případě „0“.

```
<_> <!-- tree 353 -->
     <_> <!-- root node -->
         <feature>
             <rects>
                <_>13 0 3 10 -1.</_>
                <_>13 5 3 5 2.</_></rects>
             <tilted>0</tilted></feature>
          <threshold>7.5196991674602032e-003</threshold>
          <left_val>-0.0286043900996447</left_val>
          <right_val>0.0983678027987480</right_val></_></_>
```

[alg. 1] Ukázka z xml souboru pro detekci obličeje pomocí Haarovy klasifikace. [15]

HSV (Odstín, sytost, hodnota)

Druhý způsob je využití principu prahování [16] vstupního snímku. Před samotným provedením prahování je potřeba provést převod dat z barevného modelu RGB do modelu HSV. Nad těmito daty

je poté, s vhodnými hodnotami, možné provést operaci prahování[3]. Získaný binární obraz [obr. 12] obsahuje hledaný objekt, zde je již možné nalézt jeho kontury. Popis implementace tohoto přístupu je popsán v sekci 8.2.

[obr. 12] Obraz dlaně po provedení prahování.

4.2 JavaCV a EmguCV

Pro zpracování tématu nebyl určený konkrétní programovací jazyk pro implementaci. OpenCV nemusí být dostupné pro všechny programovací jazyky, proto existují i alternativní projekty JavaCV a EmguCV. Tyto knihovny vycházejí z již popsaného OpenCV. Nabízejí obdobné možnosti, metody a datové typy. Jejich použití je totožné s OpenCV, liší se pouze syntaxí konkrétního jazyka. Vznik těchto knihoven pramenil zejména z nedostupnosti OpenCV v jazycích Java a vývojovém prostředí Microsoft .NET.

JavaCV, jak již název napovídá, je vytvořena právě pro použití v jazyce Java. Dnes již je OpenCV pro prostředí Java k dispozici, ale JavaCV má stále čím zaujmout. Ačkoliv JavaCV přináší „jen rozhrání“, které umožní volání funkcí OpenCV na pozadí, tak JavaCV tímto způsobem podporuje řadu dalších populárních knihoven pro zpracování obrazu (FFmpeg, OpenKinect a další). JavaCV se tak pro jazyk Java stává univerzální knihovnou, jež nabídne široké možnosti ze všech odvětví. Důvodem v upřednostnění OpenCV v jazyce Java může být u některých funkcí o něco lepší výkon, kterého je dosaženo skvělou optimalizací jádra programu.

EmguCV je poté implementace provedená v jazyce C#, jež je jazyk z rodiny .NET. Určení této knihovny je tedy pro jazyky Visual basic a C#. EmguCV nabízí dobrou dokumentaci jako OpenCV a je snadné pro naučení. Pro .NET jazyky nabízí také výborné uživatelské rozhraní.

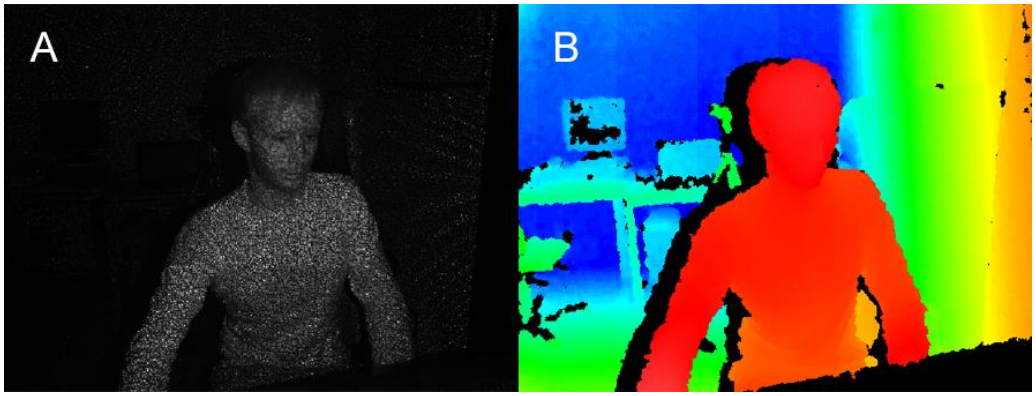
4.3 Microsoft Kinect

Nejedná se přímo o knihovnu jako OpenCV, ale jde o zařízení nabízející, s vývojářským prostředím Microsoftu, podobné možnosti jako OpenCV v oblasti detekce objektů. Pro potřeby zpracování obrazu v tématu této práce by jej bylo možné využít. Kinect je velice populární zařízení nabízející

[3] Zpracování snímku, kdy se prochází jednotlivé pixely a testuje se, zda daný pixel splňuje definovaný rozsah hodnot nebo nikoliv. Při nesplnění se zapíše „0“ při splnění „1“. Výsledkem je černobílý obraz, složený z hodnot „0“, „ 1“.

možnost snímání obrazu a jeho následného zpracování. Kinect [17] byl představen v roce 2009 jako doplněk pro herní konzoli Xbox 360. Nabízel využití převážně pro interaktivní ovládání her, kdy uživateli umožňoval aplikaci ovládat pomocí pohybů a gest. Dnes je Kinect možné připojit i k počítači se systémem Windows. Od roku 2011 jsou k dispozici také oficiální vývojářské nástroje (*Kinect for Windows SDK*) [17], poskytující možnost vyvíjet aplikace s využitím Kinectu. Díky velké základně fanoušků, které zaujal vysoký potenciál a široká možnost využití tohoto zařízení, vznikla komunita *OpenKinect*. Komunita čítající přes 2000 členů zdarma poskytuje otevřené knihovny, umožňující práci se zařízením Kinect i na operačních systémech Linux a OS X.

Pro snímání obrazu zařízení obsahuje RGB a IR kameru. Obě kamery jsou schopny pořizovat záznam v rozlišení 640×480 pixelů při obnovovací frekvenci 30Hz. IR kamera je využita společně s emitorem infračerveného záření, jež promítá do prostoru před kamerou vzor paprsků [obr. 13]. IR kamera snímá tyto odražené paprsky s drobným zpožděním. Na základě zpoždění se vypočítá vzdálenost objektů a zařízení je schopné vytvořit hloubkovou mapu prostoru před kamerou. Zařízení tedy poskytuje zachycení pohybu celého těla ve 3D, rozpoznání gest nebo obličeje. Současně je schopno detekovat až 6 osob. Na těle je schopný rozpoznat a sledovat až 20 částí pro dva uživatele současně.



[obr. 13] A) Obraz z IR kamery Kinectu pokrytý laserovou mříží. B) Získaná hloubková mapa prostoru. [18]

Kinect je pro své možnosti u vývojářů velice oblíbený a spousta projektů zabývajících se tématem promítaného uživatelského rozhraní jej právě využívá. Vývojář v tomto případě může využít dostupných funkcí pro pokročilou detekci objektů, které by musel při použití klasické RGB kamery sám vytvořit.

5 Kalibrace kamery a projektoru

Aby bylo dosaženo kvalitního rozpoznání obrazu a propojení kamery s projektorem, je potřeba tento systém podrobit tzv. kalibraci.

5.1 Kalibrace kamery

Cílem kalibračního procesu kamery je odstranění nedostatků optoelektrického systému. Nedostatky jsou radiální nebo tangenciální zkreslení čočky. Kamera je popsána matematickým modelem, jenž je vázán na interní a externí parametry použité kamery. Kalibrací kamery je získána sada těchto parametrů. Na jejich základě je možné vypočítat prostorovou pozici objektu vzhledem ke kameře, pokud známe jeho velikost, případně opravit zkreslení snímku.

Interní parametry

$$A = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & \upsilon_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$\alpha_x = f \cdot m_x \tag{2}$$

$$\alpha_y = f \cdot m_y \tag{3}$$

Matice v rovnici (1) [19] znázorňuje interní parametry kamery:

- α_x, α_y - ohnisková vzdálenost vyjádřena v pixelech, lze vyjádřit pomocí rovnic (2) a (3), kde f znázorňuje metrickou ohniskovou vzdálenost kamery a m_x, m_y vyjadřují měřítko mezi velikostí pixelu a metrické vzdálenosti
- υ_0, u_0 - reprezentují polohu hlavního snímkového bodu
- γ - koeficient radiálního a tangenciálního zkreslení

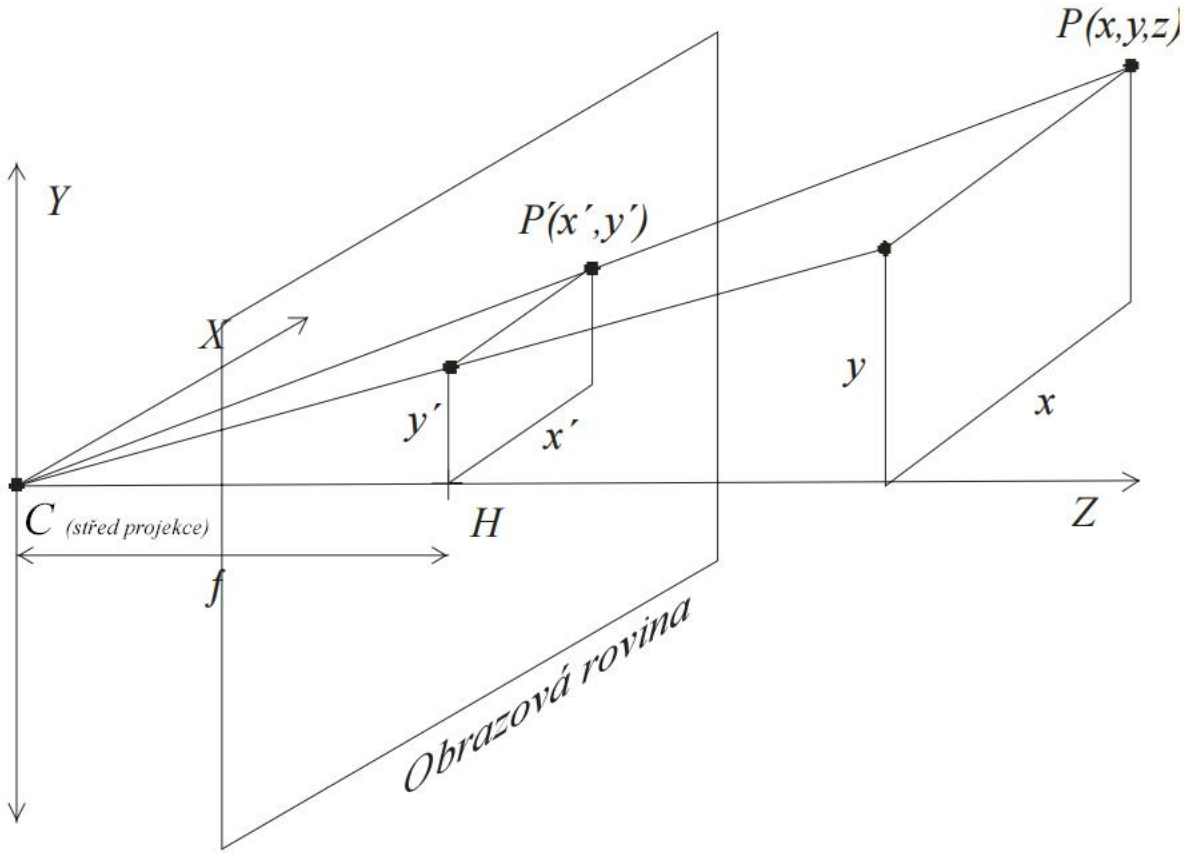
Tyto parametry charakterizují optické vlastnosti kamery. Každá kamera má vlastní vnitřní parametry, po vypočítání je můžeme uložit a znovu použít. Jejich využitím lze provést rekonstrukci paprsků středového promítání, transformací z metrických do pixelových souřadnic a odstranění geometrického zkreslení obrazu [20].

Model dírkové kamery

Pro získání kalibrace se využívá tzv. dírkové kamery. [21] Její model nám dává možnost pomocí lineární transformace promítnout bod z prostorové roviny do obrazové roviny [obr. 14]. Souřadnice bodu *P'(x', y')* lze získat z *P(x, y, z)* v prostorové rovině pomocí vzorce (4). Přepočítání získaných koeficientů již je možné provést pomocí výrazů (5).

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4}$$

$$x' = \frac{u}{w},\ y' = \frac{v}{w} \tag{5}$$

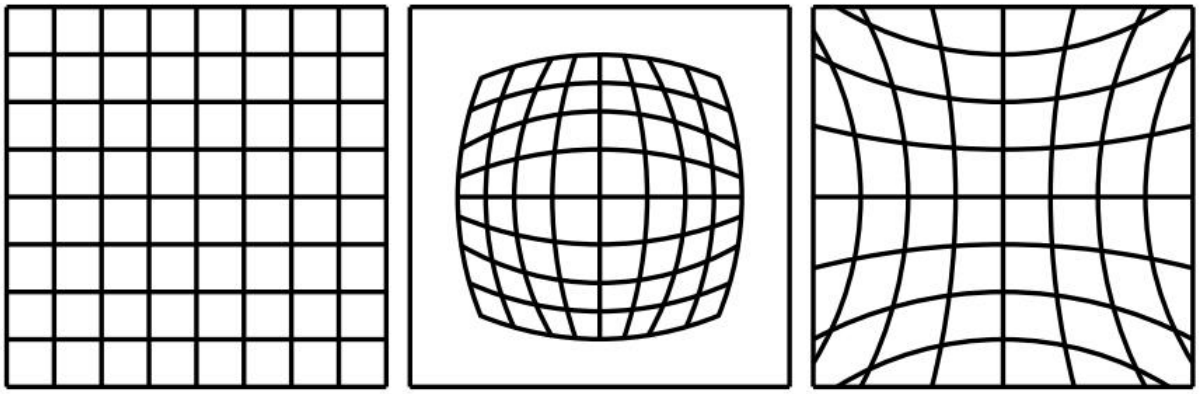


[obr. 14] Princip dírkové kamery. [21]

Zkreslení obrazu

Při průchodu světla čočkou fotoaparátu dochází ke zkreslení výsledného obrazu. Použitím fyzikálně ideálních čoček lze tomuto geometrickému zkreslení snímku předejít. Vyrobit ideální čočku však není možné a zkreslený obraz je potřeba softwarově upravit. Zkreslení snímku můžeme pozorovat na rovných přímkách, které jsou vlivem zkreslení zdeformovány. Přímka se ve snímku zobrazuje jako křivka.

Největším podílem se na deformaci podílí radiální zkreslení. Jedná se o posunutí bodů v obrazové rovině [22]. V případě soudkovitého zkreslení se tato vzdálenost směrem od středu snímku zvyšuje. U polštářkového zkreslení se naopak vzdálenost bodů od středu snižuje, ukázka [obr. **15**]. Typickým příkladem radiálního zkreslení jsou snímky přezdívané jako „rybí oko“. Toto zkreslení lze úspěšně softwarově odstranit, pokud známe koeficienty zkreslení získané kalibrací. Koeficienty znázorňující radiální zkreslení jsou k_1, k_2, k_3.



[obr. 15] Vlevo obraz bez zkreslení. Uprostřed soudkovité zkreslení. Vpravo polštářkové zkreslení. [22]

Dalším typem je tangenciální zkreslení obrazu. Toto zkreslení je způsobeno nerovnoběžně za sebou uskládanými čočkami kamery. Koeficienty tohoto zkreslení jsou označovány p_1 a p_2 [20]

Externí parametry

Vnější parametry kamery slouží k získání pozice středu kamery ve světových souřadnicích, vůči nějakému bodu. Přesněji, parametry znázorňují transformaci mezi souřadným systémem kamery a světovým souřadným systémem, vůči němuž je kamera kalibrována.

[19] Označení parametrů je *R* a *T*. Parametrem *T* je označena pozice počátku souřadnic reálného systému vyjádřené v souřadnicích souřadného systému kamery. *R* označuje rotaci. Vyjádření pozice kamery *C*, je znázorněna ve vztahu (6).

$$C = -R^{-1}T \tag{6}$$

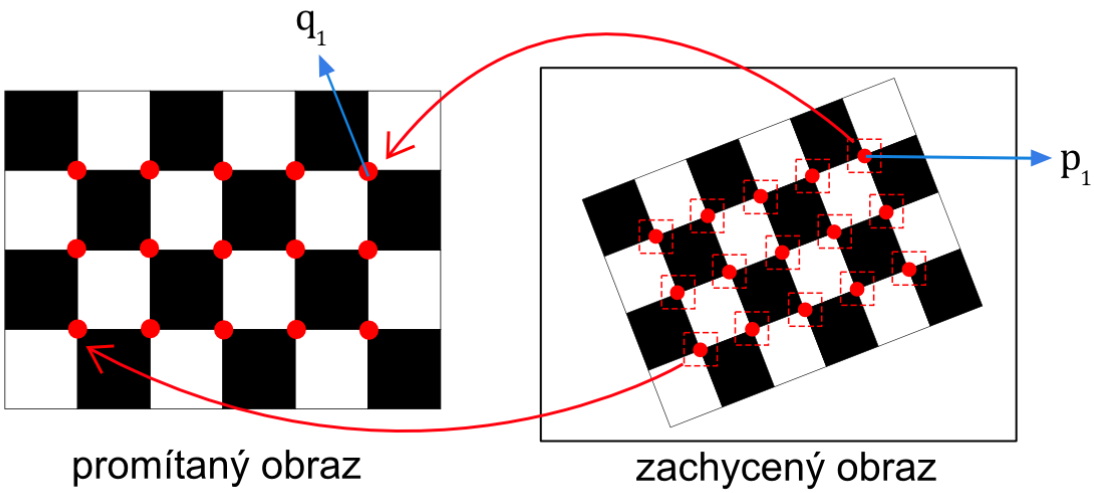
5.2 Kalibrace projektoru

Nedílnou operací v systému kamera-projektor je kalibrace projektoru vzhledem ke kameře. Každé z těchto zařízení má jiný střed promítání scény. Souřadnice stejného objektu se proto v obou rovinách liší. Aby bylo možné korektně převádět souřadnice určitého objektu, jak z promítané roviny do snímané, tak i opačně, je potřeba zjistit vzájemnou polohu mezi těmito zařízeními.

Homografie

[20] Kalibraci projektoru je možné provést nalezením homografie *H* mezi promítanou a snímanou rovinou. Homografie je reprezentována maticí *H*, jež vyjadřuje projektivní transformaci mezi rovinami. Promítnutí bodů je znázorněno na [obr. 16]. Převod bodu z jedné roviny do druhé je definován vztahem (7). Matice H se získá výpočtem vztahů mezi množinami bodů z roviny kamery a projektoru.

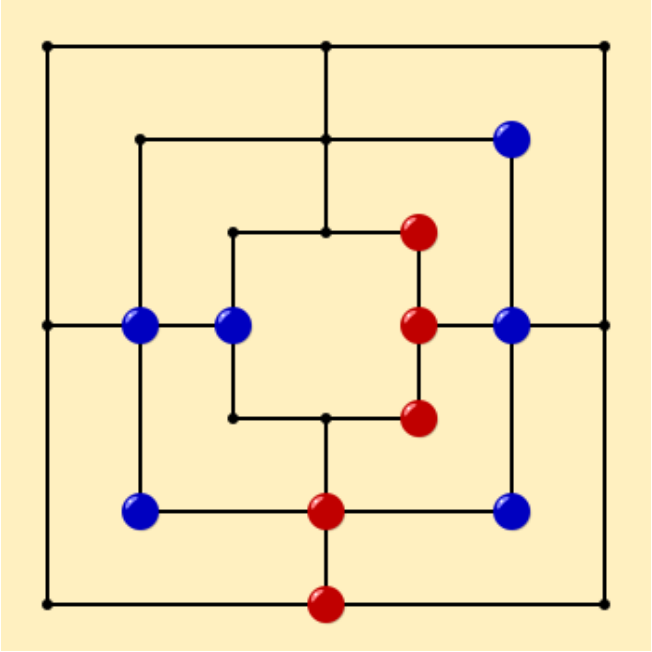
$$q_x = H \cdot p_x = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \\ w_q \end{pmatrix} \tag{7}$$



[obr. 16] Ukázka promítnutí bodů mezi rovinami.

6 Desková hra Mlýn

Obsah této práce je implementace deskové hry, jež bude využívat systému kamera-projektor. Pro implementaci jsem zvolil zjednodušenou verzi deskové hry Mlýn s názvem Three Men's Morris. Jedná se o tahovou hru určenou dvěma hráčům.



[obr. 17] Rozehraná hra mlýn. V rozehrané hře lze vidět, že oba hráči tvoří svými kameny svislý mlýn. [23]

6.1 Pravidla hry Mlýn

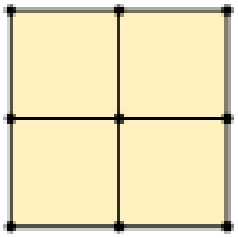
Každý hráč má k dispozici 9 kamenů, které hráči nejprve střídavě kladou na prázdnou hrací desku. Deska je tvořena ze tří čtverců se společným středem, které jsou propojeny čtyřmi liniemi [obr. 17]. Kámen může být umístěn pouze na volnou pozici hrací desky. Jakmile jsou kameny rozmístěny na hrací desce, hráči můžou v jednom tahu posunout jedním svým kamenem na sousední volný bod. Zbývají-li hráči pouze tři poslední kameny, smí kámen posunout na libovolné volné místo. Posunutím se hráč snaží docílit postavení tří vlastních kamenů do jedné svislé nebo vodorovné linie, takové trojici se říká mlýn. Hráč, který získal mlýn, odstraní jeden protihráčův kámen z hrací desky. Hra končí porážkou hráče, který nemůže posunout žádným kamenem nebo mu zůstanou pouze dva kameny a již není schopen vytvořit mlýn.

6.2 Varianty hry

Kromě klasické verze hry Mlýn existuje i několik jeho modifikací.

Three Men’s Morris

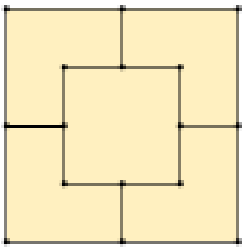
Zjednodušená verze, kde má každý hráč k dispozici 3 kameny. Posunutí kamenu již nemusí být pouze na sousední bod, ale na jakékoliv volné místo na hrací ploše. Zde již není potřeba odstraňovat kameny, ale hra končí ve chvíli, kdy jeden z hráčů získá mlýn.



[obr. 18] Hrací deska modifikace Three Men's Morris. [23]

Six Men’s Morris

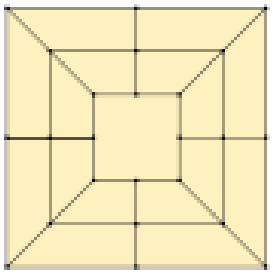
Verze hry, v níž má každý hráč 6 kamenů. Hrací plocha je v tomto případě redukována o jeden vnější čtverec. Pravidla hry jsou stejná jako pro klasickou verzi.



[obr. 19] Modifikovaná herní deska pro Six Men's Morris. [23]

Twelve Men’s Morris

[23] Proti klasické verzi přidává tato alternativa na herní pole uhlopříčky spojující rohové body desky. Každý hráč má 12 kamenů. V této variantě může dojít k zaplnění herní plochy již při počátečním rozmisťování kamenů. V tomto případě hra končí remízou.



[obr. 20] Verze Twelve Men's Morris nabízí i pohyb po úhlopříčkách. [23]

7 Návrh aplikace

Návrh aplikace je možné rozdělit na několik úkolů:

- návrh uživatelského rozhraní (dále GUI)
- návrh ovládání aplikace
- návrh implementace programu

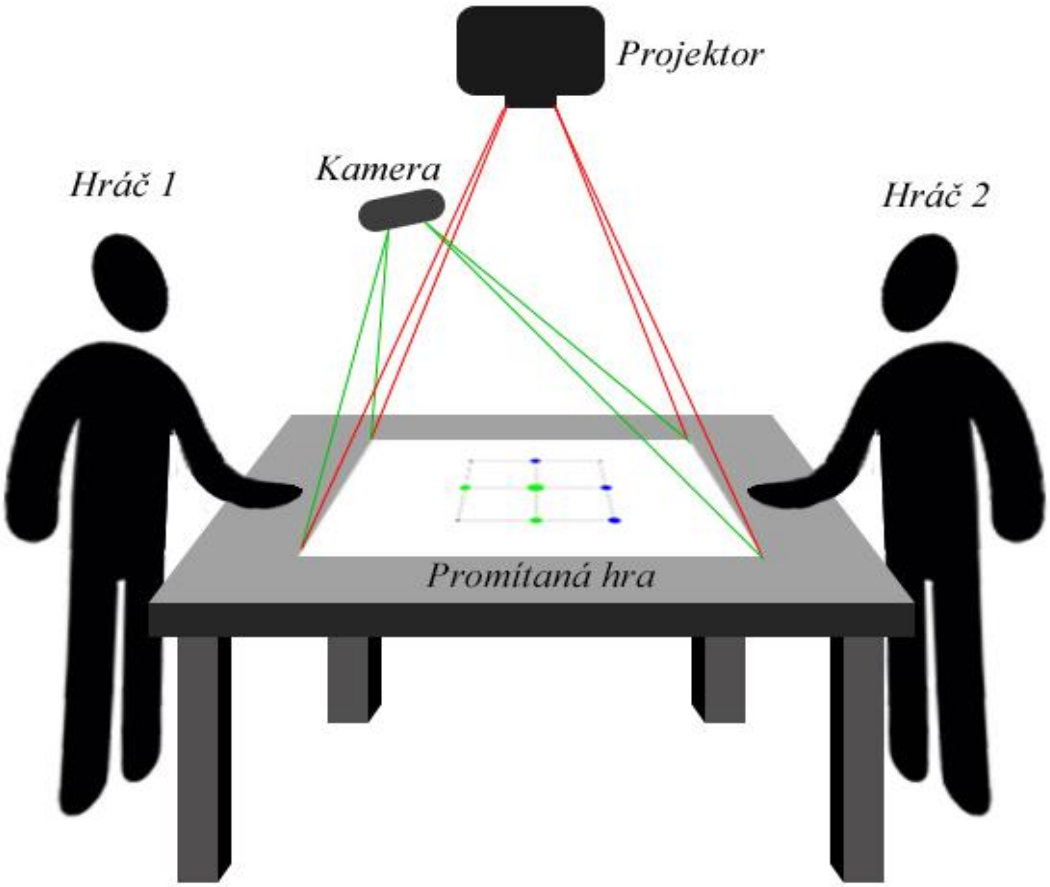
7.1 Hardwarová sestava

Hlavní součásti systému:

- Kamera - Snímá herní desku ve spektru viditelného světla.
- Projektor - Promítá rozhraní hry na promítací plochu.
- Počítač - Řídicí jednotka celého systému. Provádí zpracování obrazu z kamery, zpracovává hru a generuje promítaný obraz.

Mezi vedlejší součásti poté patří:

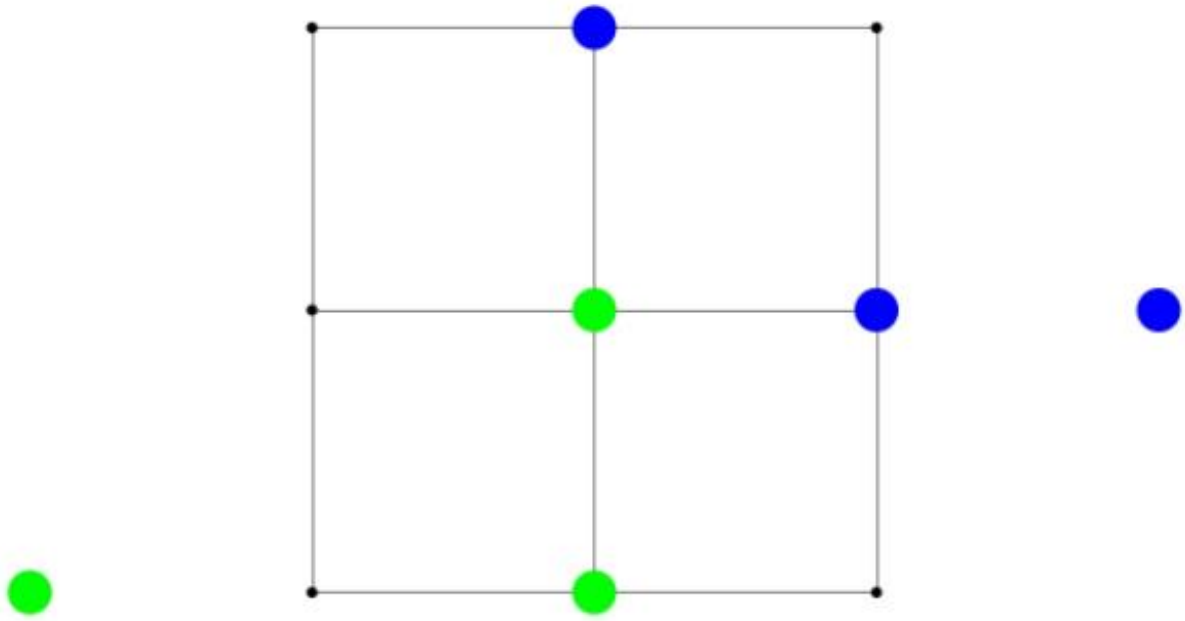
- Stůl - Slouží jako promítací plocha.
- Kalibrační vzor - Šachovnicový vzor pro kalibraci kamery.



[obr. 21] Znázornění systému kamery s projektorem. Hra je promítána na promítací plochu. Uživatelé provádějí interakci pomocí rukou.

7.2 GUI a ovládání aplikace

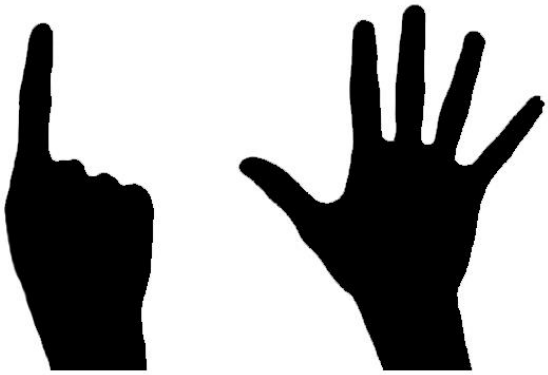
Obecně u aplikací, jež jsou ovládány dotykem, je potřeba dbát na návrh GUI tak, aby bylo možné jej pohodlně ovládat. Ovládací prvky ve hře musí být dostatečně velké pro zaručení pohodlné interakce.



[obr. 22] Návrh GUI deskové hry mlýn.

Na [obr. 22] je ukázka použitého návrhu hry. Vzhledem k využití přední projekce[4] a přihlédnutí k jejím nevýhodám je navržené GUI minimalistické. Tímto krokem jsem se snažil předejít následným problémům s detekcí rukou, jež by byly překryty promítaným obrazem. GUI tak obsahuje hrací desku a kameny. Na začátku hry jsou kameny jednotlivých hráčů svisle seřazeny po boku hrací desky. Tyto kameny poté hráč umisťuje na hrací plochu.

Interakce s kameny a hrací deskou je založena na detekci rukou a prstů. Pro ovládání jsou hráči k dispozici dvě gesta [obr. 23].



[obr. 23] Vlevo gesto ukázání pro výběr. Vpravo potvrzovací gesto.

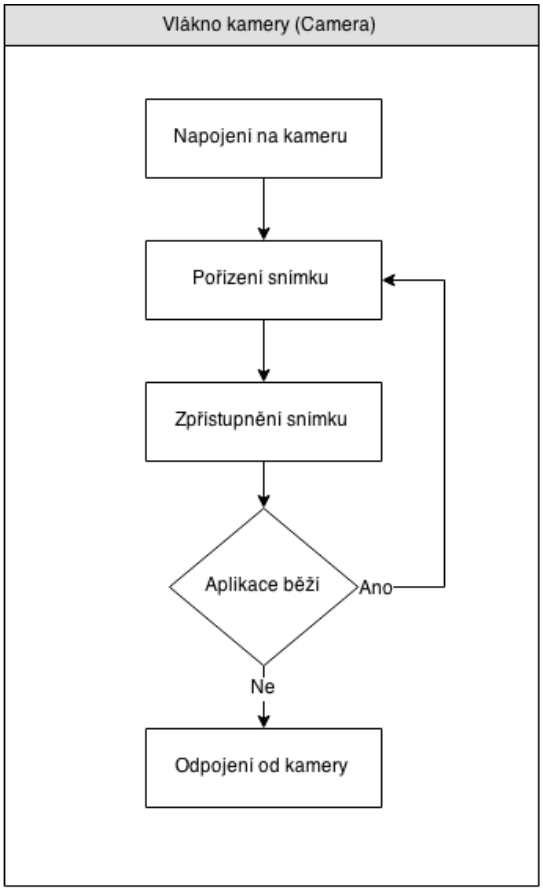
[4] Projektor promítající obraz se nachází před promítanou plochou. Opakem je zadní projekce, kdy se projektor nachází za plátnem, na které promítá obraz.

Použitím gesta pro výběr uživatel ukáže prstem na kámen, který bude chtít posunout. Po ukázání se vybraný kámen zvýrazní jeho zvětšením. Než uživatel výběr potvrdí rozevřením dlaně, má možnost ještě ukázáním na jiný kámen změnit výběr. Když je hráč s výběrem kamene spokojen, potvrdí výběr pomocí potvrzovacího gesta. Poté obdobným způsobem probíhá výběr bodu na hrací ploše, který určuje nové umístění kamene. Potvrzením vybraného bodu na desce se poté provede přesun označeného kamene na cílové místo. Po úspěšném provedení tahu je na řadě hráč druhý. V situaci, kdy hráč získal mlýn, jsou kameny tvořící mlýn zvýrazněny a hra je ukončena.

7.3 Aplikace

Návrh aplikace je rozdělen do tří částí, jednotlivé části jsou znázorněny na diagramech [diag. 1], [diag. 2] a [diag. 3]. Každá z částí běží jako vlastní vlákno. Rozdělení na samostatná vlákna je hlavně z důvodu logického oddělení jednotlivých částí, ale také zrychlení běhu programu. Informace mezi vlákny jsou předávány na základě implementace adaptéru.

Vlákno kamery



[diag. 1] Vlákno provádějící obsluhu kamery.

Životnost vlákna je určena dobou běhu programu. Jeho úkolem je pořizovat nové snímky scény pomocí kamery, na kterou je napojeno. V případě úspěšného pořízení snímku jej uvolní k poskytnutí pro ostatní vlákna programu.

Vlákno pro zpracování snímku

[diag. 2] Vlákno pro zpracování obrazu a detekci prstů.

Jedná se o nejvíce časově náročnou část programu. Vlákno poběží po celou dobu, kdy je hra spuštěná. Jejím úkolem je získání kamerou poskytnutého snímku. Následně projde získaný snímek korekcí, na základě parametrů kamery vypočítaných při její kalibraci.

Na snímku je poté provedena úprava snímku a detekce obrazu, jež je podrobně popsána v sekci 8.2. Poté, co je ve snímku provedena detekce prstů, je na základě jejich detekovaného počtu vyhodnoceno, zda se jedná o potvrzovací gesto, gesto výběru nebo nebylo detekováno žádné gesto uživatele.

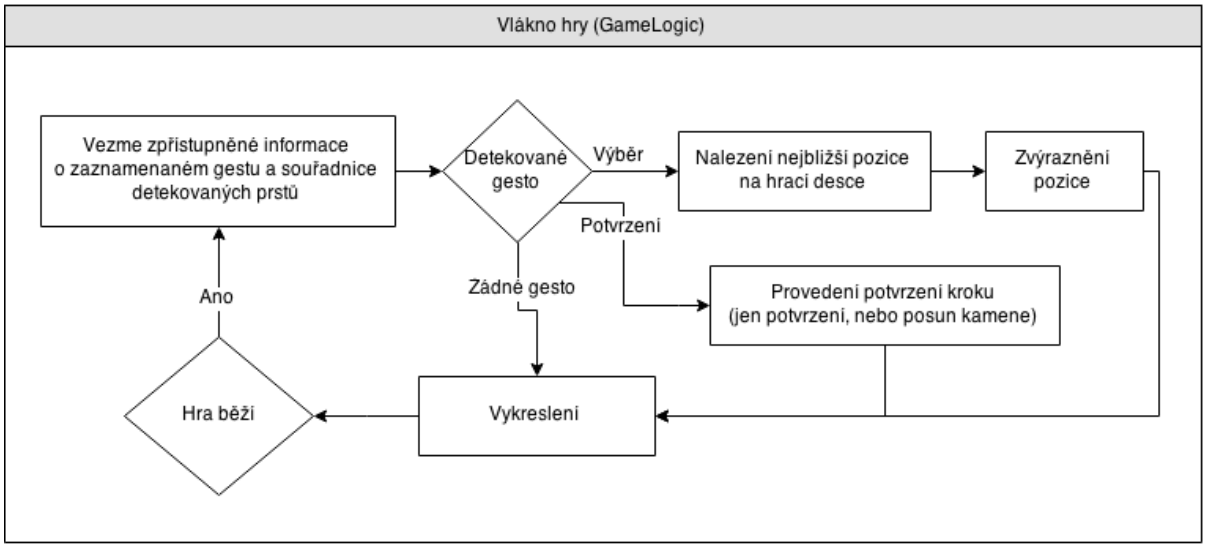
Pro potvrzovací gesto musí být detekováno 5 prstů. V případě jeho detekce se do adaptéru uloží informace o zaznamenaném potvrzovacím gestu.

Gesto výběru obsahuje pouze 1 detekovaný prst. V tomto případě jsou souřadnice detekovaného prstu převedeny z obrazové roviny kamery do roviny projektoru pomocí transformace bodu podle nalezené homografie. Převodem souřadnic umožníme použití bodu přímo v aplikaci hry, kdy hrou vykreslená herní deska má stejný souřadný systém jako projektor. Převodem pouze jednoho bodu se implementace stává také efektivnější. Do adaptéru je nahrán transformovaný bod spolu s informací, že bylo zaznamenáno gesto výběru.

V případě nepovedené detekce dlaně a prstů je do adaptéru nahrána informace, že nebylo v tomto kroku nalezeno žádné gesto.

Uložením dat do adaptéru dojde k jejich zpřístupnění pro ostatní vlákna, která si můžou nahraná data vyzvednout.

Vlákno hry



[diag. 3] Vlákno, které má na starost herní logiku.

Hlavním úkolem herního vlákna je vykreslení GUI a reakce na detekovaná gesta. Vlákno si z adaptéru vyzvedne informace o detekované události v obrazu. V případě, že nejsou k dispozici nová data, vlákno se uspí. Notifikace vlákna proběhne v případě nahrání nových dat do adaptéru. Na základě dat získaných z adaptéru se poté odvíjí následné akce hry.

Gesto potvrzení

V případě detekovaného gesta potvrzení, provede herní logika akci, kdy:

- Potvrdí aktuální výběr kamenu. V dalším kroku bude logika očekávat označení místa na herní desce.
- Potvrdí výběr místa na hrací ploše a provede přesun vybraného kamenu.
- V případě, kdy předtím nebyl proveden žádný výběr, neprovádí žádnou další akci.

Gesto výběru

Při detekci gesta výběru se z adaptéru načte bod, který znázorňuje pozici špičky prstu. Tento bod je již z fáze zpracování obrazu transformován do roviny projektoru, proto je možné jeho následné využití v herní logice. Na základě souřadnic získaného bodu se provede vyhledání nejbližší pozice na herní desce. Nejblíže nalezená pozice je následně otestována, zda leží v určitém, poloměrem definovaném kruhu. Situace, kdy bod leží mimo kruhem definovaný rozptyl, není provedena žádná akce. V případě očekávání výběru kamene musí nejbližší pozice obsahovat i položený kámen, v tomto případě dojde ke zvýraznění kamene. Pokud již je kámen vybrán, herní logika očekává označení volné pozice na herní desce. Nejbližší volná pozice bude opět zvýrazněna.

Žádné gesto

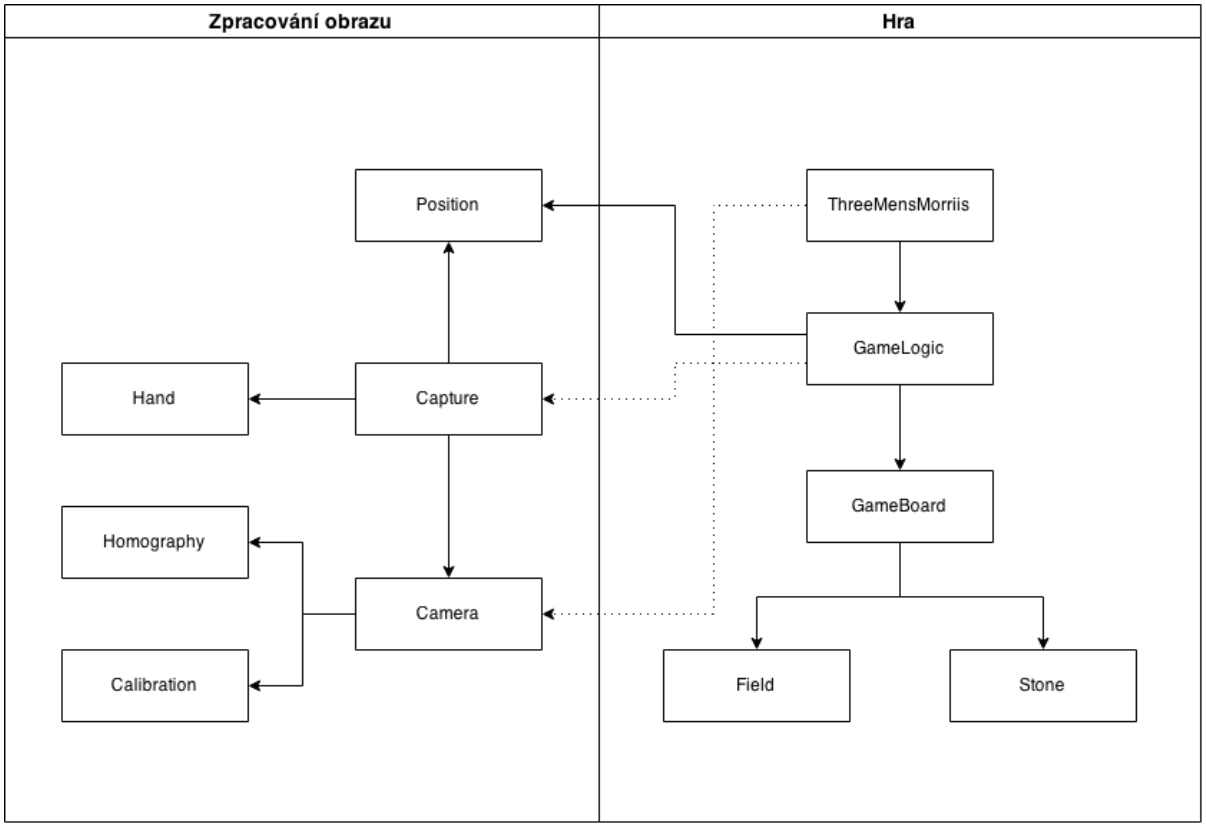
Při detekci akce, kdy nebylo při zpracování nalezeno žádné gesto, logika hry neprovádí žádnou obslužnou metodu.

Po provedení všech obslužných metod, pro detekovanou akci, je provedeno překreslení herní desky.

8 Implementace

Pro implementaci aplikace byl zvolen jazyk Java spolu s knihovnou pro zpracování obrazu OpenCV, jež byla popsána v kapitole 4.1. Vykreslení rozhraní programu je prováděno pomocí nástroje Swing.

8.1 Aplikace



[diag. 4] Diagram tříd aplikace.

Implementace aplikace je založena na navrženém popisu 7.3 do dvou částí a několika tříd, jež jsou znázorněné v [diag. 4]. Část první obsahuje samostatnou hru společně s herní logikou. Druhá část se stará o zpracování obrazu. Herní část spolu se zpracováním obrazu a kamerou, jsou navrženy tak, aby fungovaly jako samostatné vlákna.

Část zpracování obrazu:

- Camera: Třída obsluhující kameru. Nabízí možnost zavolání kalibrace, nalezení homografie, vyfocení snímku (samostatné vlákno).
- Homography: Nabízí nástroje pro nalezení homografie a transformaci bodu z roviny kamery do roviny projektoru.
- Calibration: Nástroje pro kalibraci kamery a transformaci snímku.
- Capture: Obsahuje metodu, která běží jako samostatné vlákno. Metoda má na starost zpracování obrazu. Komunikuje s rozhraním kamery, přes které získá pořízený snímek, nebo také možnost použít nástroje v třídách Homography a Calibration.

- Position: Slouží jako adaptér pro předání dat mezi třídou Capture a GameLogic.

Herní část obsahuje třídy:

- ThreeMensMorris: Hlavní třída, která vykresluje menu. Nabízí spuštění hry (GameLogic) nebo kalibraci kamery.
- GameLogic: Obsahuje herní logiku deskové hry. Komunikace se zpracováním obrazu probíhá pomocí třídy Position.
- GameBoard: Obsahuje implementaci herní desky složenou z objektů typu Field.
- Field: Třída znázorňující pozici na herní desce, na kterou je možné umístit kámen (Stone).
- Stone: Implementace kamenu. Umisťuje se na pozici (Field).

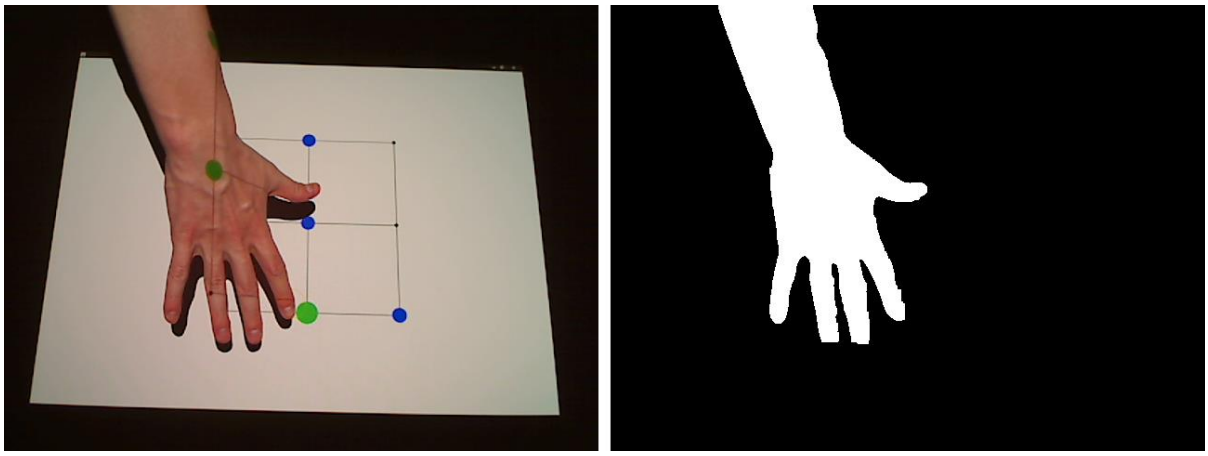
8.2 Detekce rukou a prstů

Pro rozpoznání rukou v obraze kamery jsem se rozhodl využít knihovny OpenCV a metody založené na základě prahování obrazu v HSV modelu. Důležitou třídou pro zpracování a detekci dlaně v obraze je třída Hand.

Zpracování snímku

```
Imgproc.medianBlur(image, image, MEDIAN_SIZE);
Imgproc.cvtColor(image, image, Imgproc.COLOR_BGR2HSV);
inRange(image, new Scalar(hHandMin, sHandMin, vHandMin),
                new Scalar(hHandMax, sHandMax, vHandMax), image);
image = morphology(image, HAND_EROSION, HAND_DILATION);
```

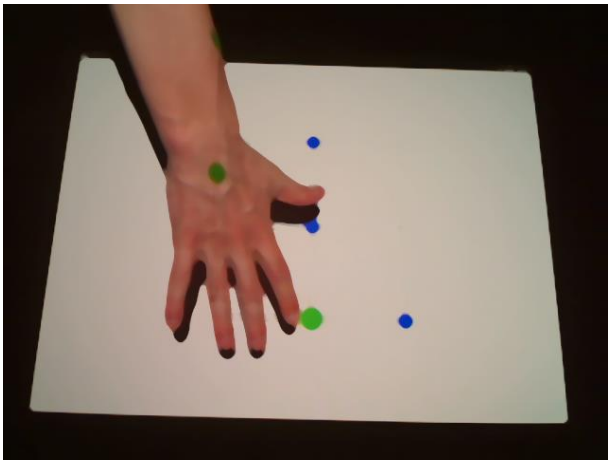
[alg. 2] Znázorněná sekvence pro zpracování snímku.



[obr. 24] Vlevo surový snímek z kamery, vpravo snímek zpracovaný.

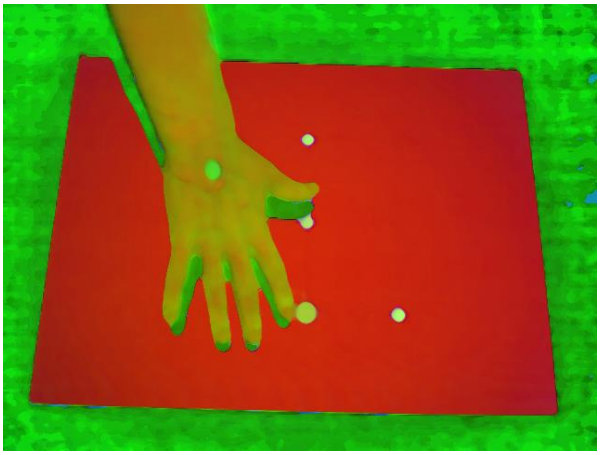
Ve fázi zpracování jde o získání binárního obrazu objektu určeného k detekci, ukázka je na [obr. 24]. Po získání surového obrazu z kamery se nad obrazem jako první provede rozmazání pomocí funkce `medianBlur()`. Filtr pracuje na základě přepočítání hodnoty jednotlivých pixelů v závislosti na průměru hodnot okolních pixelů. Okolí, z kterého je počítán medián, definuje matice.

Experimentálním zkoušením různých velikostí matic se dospělo k optimální velikosti matice o rozměru 3×3. Rozmazáním obrazu se docílí získání celistvějších barevných ploch, odstranění šumu a artefaktů z pořízených snímků.



[obr. 25] Na snímku lze vidět, že aplikací mediánového filtru se povedlo ze snímku odstranit herní desku.

Aby bylo dosaženo kvalitního výsledku segmentace ruky z obrazu, je potřeba provést segmentaci nad barevným modelem, kde bude možné dobře rozeznat odstín kůže od prostředí. Pro tento účel je vhodný barevný model HSV [16]. Převedení z RGB do HSV se provede pomocí funkce `cvtColor()`.



[obr. 26] Snímek převedený do barevného modelu HSV.

Nejdůležitější částí zpracování snímku je provedení prahování. Pro provedení segmentace je potřeba určit rozsah hodnot pro jednotlivé složky modelu HSV. Každá ze tří složek modelu má 0-255 hodnot. Experimentálně zvolené rozsahy hodnot pro jednotlivé složky jsou uvedeny v [tab. 1]. Prahování je následně provedeno pomocí funkce `inRange()`. Na [obr. 27] je vidět získaný obraz. Body snímku, jež byly v určeném rozsahu, jsou zaznamenány bílou barvou a ostatní černou.

	Min.	**Max.**
H (odstín)	0	50
S (sytost)	50	255
V (hodnota)	80	255

[tab. 1] Zvolený rozsah složek HSV pro provedení prahování.

[obr. 27] Bíle zvýrazněná ruka v obraze po provedení segmentace.

Segmentace nemůže být příliš agresivní, abychom nepřišli o data, se kterými chceme pracovat. Aby se předešlo chybné detekci kontur ve snímku, je potřeba rušivá data, která ve snímku zbyla, odstranit jiným způsobem. Zde se nabízí využití morfologické operace eroze a dilatace. Erozi lze použít ke zjednodušení struktury objektů. Jejím použitím se z obrazu odstraní objekty jednotkové tloušťky. Aplikováním metody na snímek [obr. 27] se provede redukce segmentovaného objektu [obr. **28**]. Redukcí dojde ke ztrátě dat, které nebyly zcela odstraněny již při prahování. Při aplikaci eroze může v některých případech dojít ke vzniku trhlin v segmentovaném objektu. V našem případě se potenciální problém nachází u prstů ruky. Objekt v místě prstů není tak silný a aplikací eroze může v tomto místě vzniknout trhlina. Následkem by byla zhoršená, nebo také zcela nemožná detekce prstů. Proto je potřeba po provedení eroze aplikovat operaci dilatace. Na rozdíl od eroze, která objekt redukuje, dochází u dilatace ke zvětšení objektu. Zvětšením je možné zacelit potencionální trhliny. Nyní se již jedná o cílový snímek [obr. 28], na kterém je možné provést rozpoznání ruky a prstů.

[obr. 28] Vlevo obraz po provedení eroze. Vpravo po následné aplikace dilatace získaný cílový snímek.

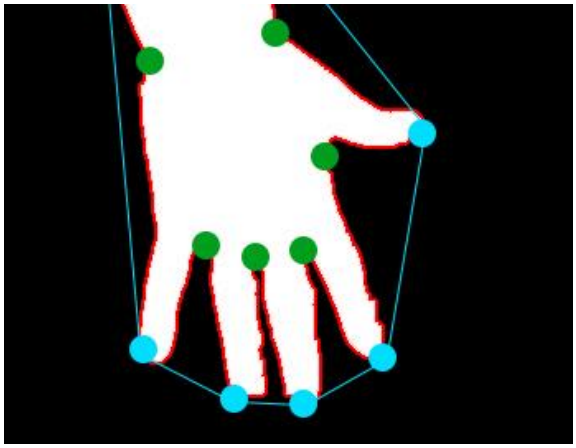
Rozpoznání dlaně a prstů

O rozpoznání objektů se stará metoda `detect(Mat imageThreshold)`, jež na vstupu očekává zpracovaný snímek z předchozích kroků, přesněji řečeno masku ruky.

Jako první se v získané masce provede vyhledání obrysů pomocí `findContours()`. Použitá funkce umožňuje nastavit mód hledání obrysů. Jedním z módů je vyhledání pouze externích obrysů, případně druhá možnost, umožňuje zanořené hledání obrysů tzv. více úrovňové. V tomto případě parametrem RETR_EXTERNAL určujeme použití jednoduššího hledání obrysů, tzv. první úrovně.

Získané obrysy, kterých může být nalezeno více, protřídíme podle jejich velikosti. Největší nalezený obrys otestujeme, zda splňuje požadovanou minimální velikost. Pokud testem projde, předpokládá se, že se jedná o obrys dlaně hráče. V případě, že kontura testem neprojde, je vyhodnoceno, že ve scéně není žádná dlaň.

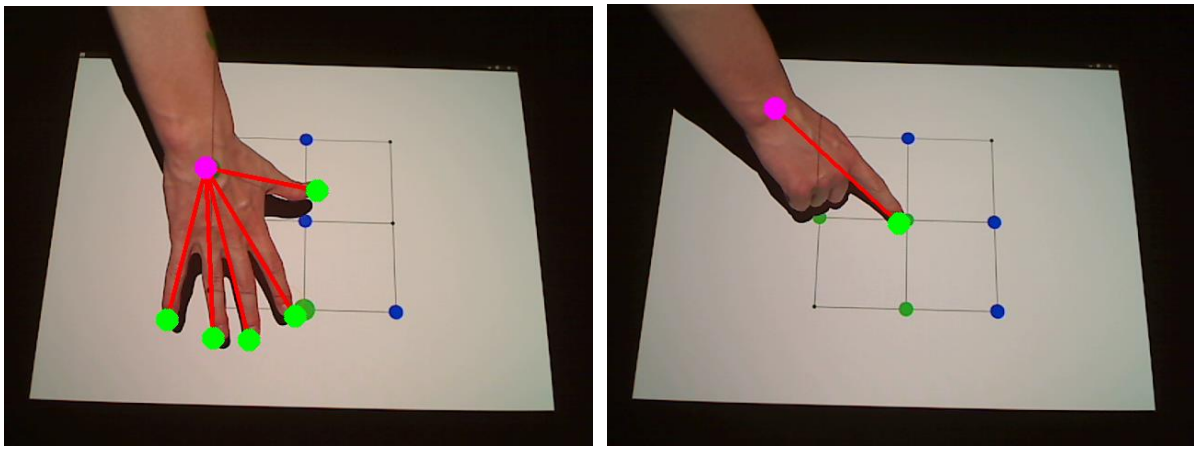
Jakmile získáme obrys dlaně, je možné jej využít k detekci konečků prstů, které budou jedny z bodů obrysu. Aby bylo možné s body obrysu pracovat, je potřeba nalézt vrcholové body obrysu tzv. konvexní obal. K tomu slouží funkce `convexHull()`. Následně funkcí `convexityDefects()`, získáme list nalezených defektů v konvexním obalu. Defektem je oblast, ve které dochází k propadu mezi konvexním obalem a obrysem dlaně [obr. 29].



[obr. 29] Červenou barvou je znázorněn získaný obrys dlaně, modrá patří konvexnímu obalu a zelená znázorňuje defekty.

Využití nalezených defektů lze využít pro detekci konečků prstů. Jak lze vidět na [obr. 29], defekty se nacházejí mezi prsty ruky a body konvexního obalu korespondují s konečky prstů. Ale defektů společně s konvexními body může být více na různých místech. Proto je potřeba vhodně vyfiltrovat list konvexních bodů, aby zbyly pouze body znázorňující špičky prstů. Jako první je vhodné určit minimální velikost defektu, lze to brát jako minimální délku prstů. Tímto pravidlem dojde k odstranění malých defektů. Dalším krokem při filtraci je kontrola, zda úhly mezi jednotlivými prsty svírají menší než je maximálně možný úhel [24].

Jakmile je filtrace dokončena, zbylé body definující konečky prstů. Na základě jejich počtu se určí, zda jde o gesto potvrzovací (rozevřená dlaň, detekováno 5 prstů), nebo gesto výběru (detekován pouze jeden prst).



[obr. 30] Detekování prstů. Gesto potvrzení vlevo. Gesto pro výběr vpravo.

8.3 Kalibrace kamery

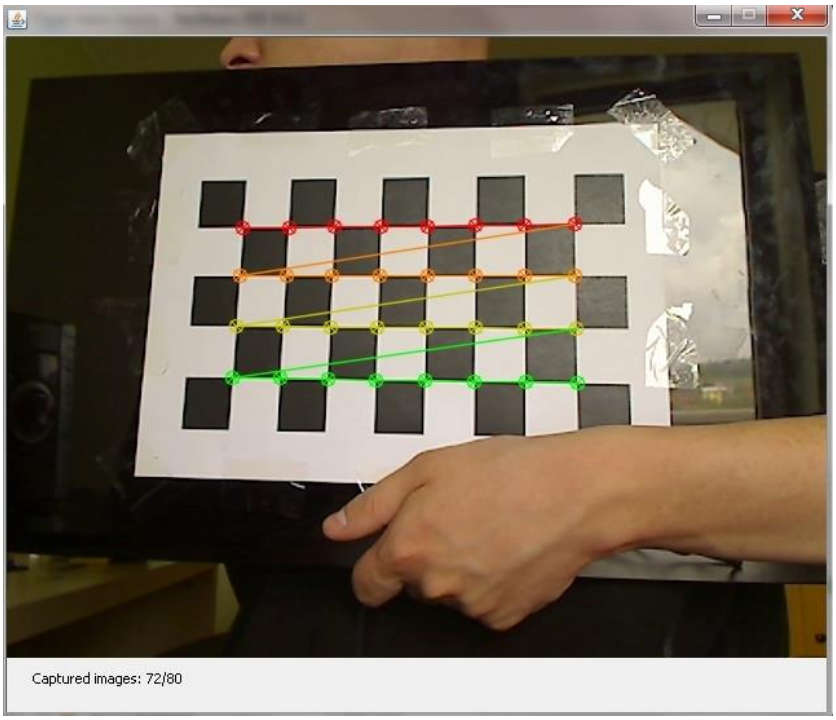
Ke kalibraci je využito vytištěného šachovnicového vzoru s 8×4 vnitřními body. Při kalibraci je potřeba umístit vzor na rovné a tvrdé ploše, aby se předešlo jeho deformaci a tudíž špatnému výsledku kalibrace. Za tímto účelem je kalibrační vzor umístěn na skleněné tabuli, která zaručuje potřebnou rovnost.

Samotná kalibrace má dvě fáze:

- snímání kalibračního vzoru
- výpočet parametrů kamery

Snímání

V této fázi je potřeba získat snímky kalibračního vzoru v různých pozicích. Kamera je umístěna pevně a pohybuje se pouze kalibračním vzorem. Zachycení snímků probíhá v reálném čase. Po zachycení snímku s kalibračním vzorem se provede jeho detekce pomocí funkce `findChessboardCorners()`. Detekování spočívá v nalezení rohů uvnitř šachovnice [obr. 31], ty jsou poté uloženy jako matice bodů. Aby bylo nalezené body vůči čemu kalibrovat, je potřeba vygenerovat vlastní „dokonalý“ šachovnicový vzor ve 3D souřadnicích, jež bude sloužit jako předpis pro nalezený vzor [25]. Jakmile je získaný dostatečný počet snímků, je možné přejít ke kalibraci. Za účelem této práce postačuje při kalibraci získání pouze vnitřních parametrů.



[obr. 31] Ukázka kalibračního programu. Na ukázce probíhá snímání scény a nalezení kalibračního vzoru.

Výpočet parametrů

Vnitřní parametry kamery jsou získány pomocí funkce `CalibrateCamera()`. Ta pro všechny snímky, získané z předchozí fáze, vypočítá jejich rotace a transformace mezi nalezeným a generovaným vzorem. Výsledkem je získaná matice kamery a matice s koeficienty zkreslení. Návratová hodnota funkce je průměrná chyba re-projekce [25]. Číslo udává odhad přesnosti

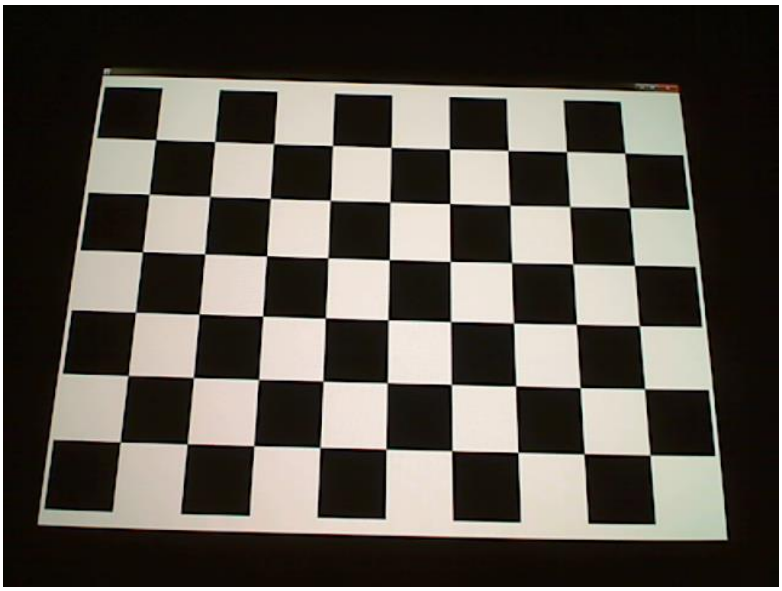
nalezených parametrů. Toto číslo by mělo být co nejblíže číslu 0. Následně již je možné pomocí získaných parametrů provést korekci snímku.

8.4 Kalibrace projektoru

Pro kalibraci projektoru vůči kameře je využito metody pro získání homografie mezi jednotlivými rovinami. Homografie je nalezena vždy při spuštění hry a odpovídá aktuální pozici mezi kamerou a projektorem. V případě manipulace se zařízením je potřeba provést výpočet homografie znovu.

Homografie je získána na základě šachovnicového vzoru (s vnitřními body 9×6), jež je zobrazen projektorem na promítací ploše [obr. 32]. Kamera pořídí snímek tohoto vzoru a nalezne zobrazený vzor. Aby bylo možné vypočítat vzájemnou pozici mezi rovinou kamery a projektoru, je potřeba ještě získat obraz šachovnice, jež je promítaná. Promítaná šachovnice je generovaná algoritmem tak, aby pokryla vždy obsah celého okna. Při generování se tedy uloží jednotlivé rohy generované šachovnice.

Jakmile jsou získány body rohů šachovnice v obrazové rovině kamery i rovině projektoru, je možné provést výpočet homografie mezi rovinami. Pomocí funkce `findHomography()` se následně získá transformační matice, kterou lze převést souřadnice bodu z jedné projekční roviny do druhé.



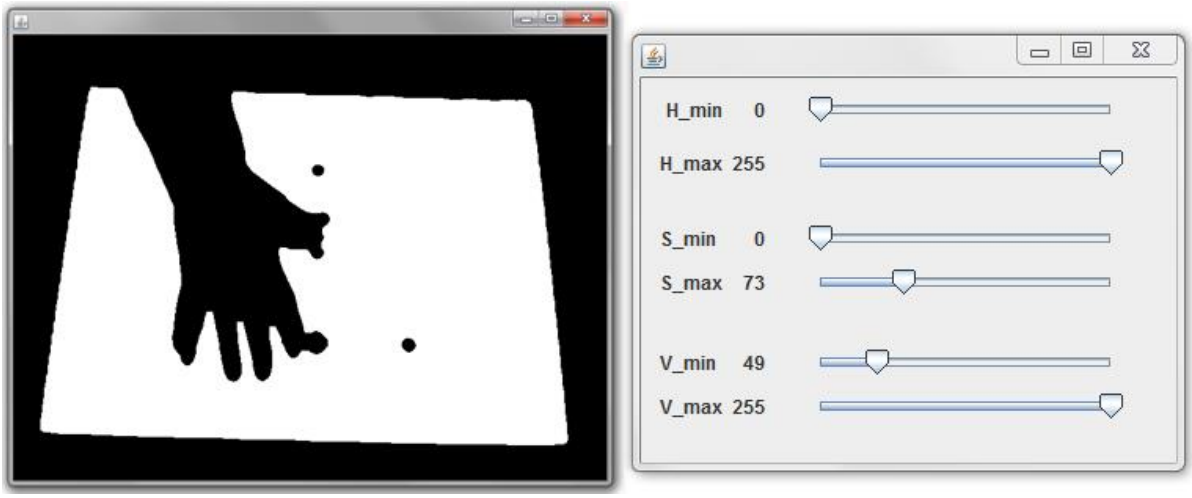
[obr. 32] Projektorem promítaný šachovnicový vzor, zachycený kamerou.

9 Testování a validace

Aplikace byla testována po jednotlivých logických celcích, ze kterých je složena. Otestovaná byla i funkčnost aplikace jako celku. Za účelem testování bylo využito LED projektoru Acer K11 a webkamery Logitech Webcam Pro 9000. Obraz byl promítán v rozlišení 1400×1050px a promítací plocha byla ve vzdálenosti 115cm. Kamera scénu snímala ze vzdálenosti 70cm v rozlišení 640×480px. Testování probíhalo na operačním systému Windows 7 Professional 64bit. Pro překlad aplikace bylo využito JDK a JRE verze 1.8.0.45.

9.1 Testování rozpoznání detekce ruky

Pro testování rozpoznání rukou v obrazu bylo vytvořeno rozhraní, které umožnilo v reálném čase měnit parametry pro snímaný obraz. Jednotlivé kroky úpravy snímku byly zobrazeny v samostatných oknech. Pomocí posuvníků bylo umožněno s jednotlivými parametry manipulovat.

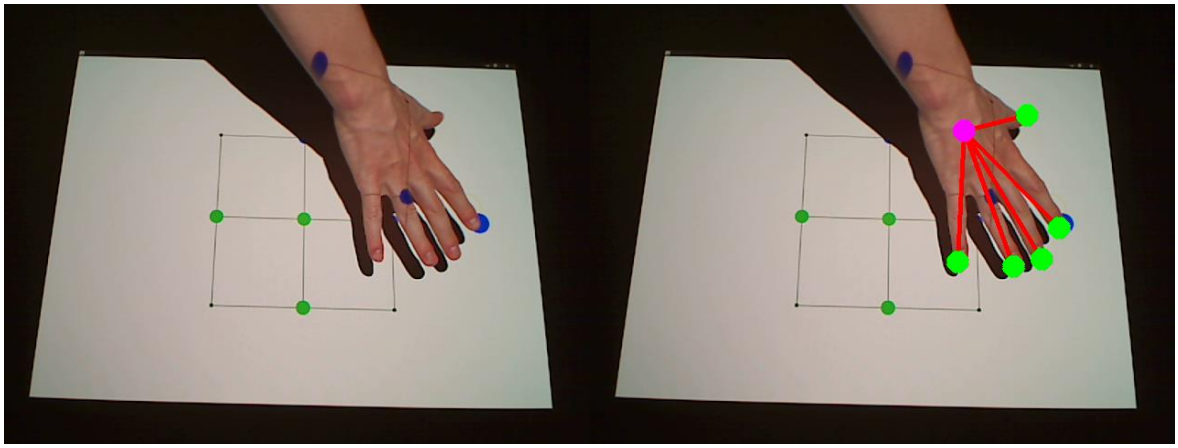


[obr. 33] Ukázka nástroje pro změnu rozsahu prahování snímku.

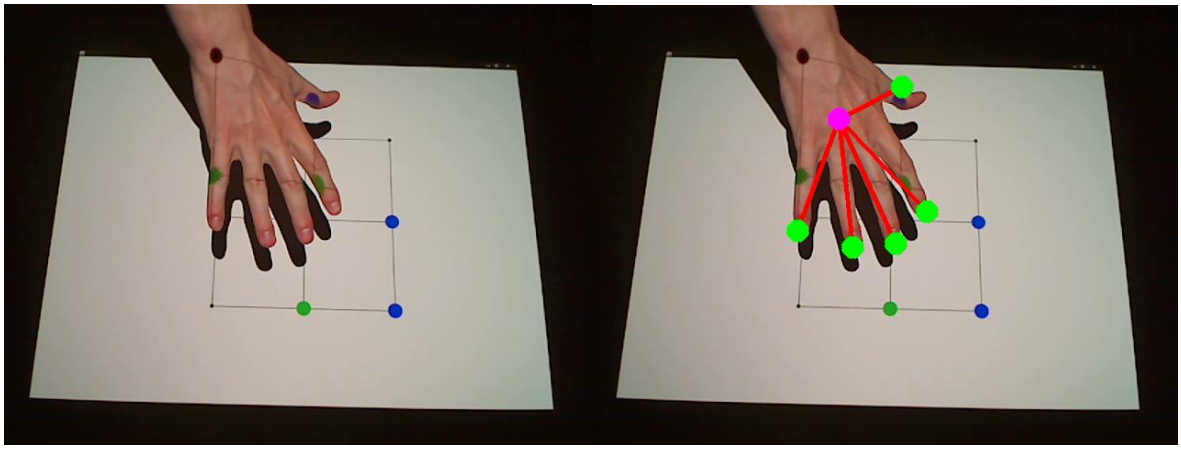
Hledání nejvhodnějších hodnot parametrů pro jednotlivé kroky probíhalo na sadě snímků. Snímky obsahovaly různé situace:

- čistý obraz rozevřené dlaně, nerozevřené dlaně, jednoho, dvou prstů…
- obraz ruky překrytý hrací deskou
- snímek bez ruky

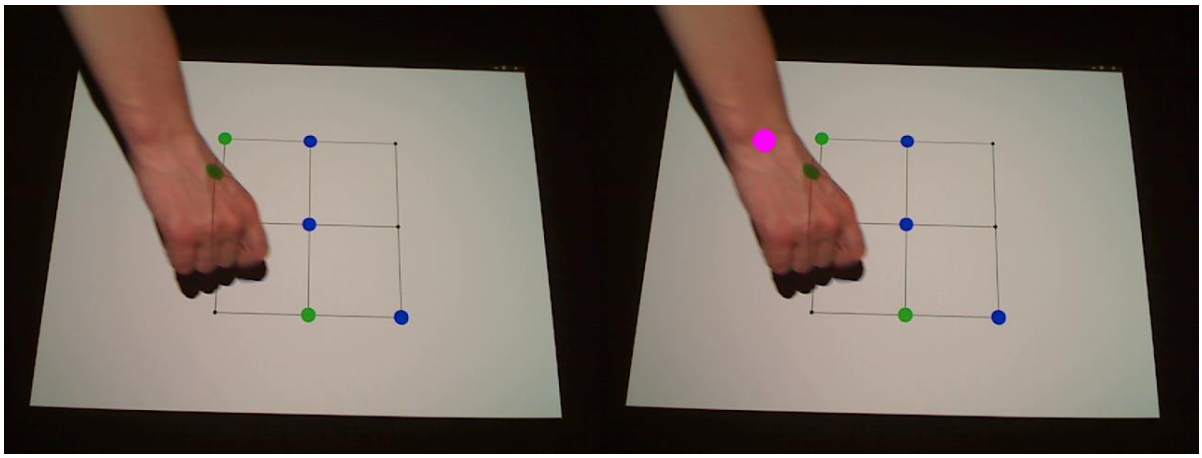
Výstupem testu byly obrazové soubory, jež obsahovaly detekované objekty. Kontrola těchto souborů probíhala ručně. Při testech si implementovaná metoda pro rozpoznání vedla dobře. Problémem bylo v několika případech částečné překrytí prstu promítaným kamenem [obr. 35].



[obr. 34] Označení prstů proběhlo v pořádku.



[obr. 35] U tohoto vstupního snímku nastal problém s překrytím palce modrým kamenem. Detekce palce není v tomto případě přesná.



[obr. 36] Nalezení kontury ruky a vykreslení pouze bodu gravitace. Korektně není detekován žádný prst.

9.2 Kontrola kalibrace kamery

Kalibrace kamery je velice náchylná na kvalitu pořízených snímků a v závislosti na jejich kvalitě je odvozena také kvalita výsledné kalibrace. Pro provedení kvalitní kalibrace kamery je potřeba dbát na následující body:

- Fyzický kalibrační vzor je potřeba mít připevněný na rovném a neohebném povrchu.
- Důležité je dostatečné a hlavně rovnoměrné osvětlení.
- Provést alespoň 40 snímků kalibračního vzoru v různých pozicích a vzdálenostech od kamery.
- Nemanipulovat s kamerou, pohybuje se pouze kalibračním vzorem.
- Vypnutí automatického ostření kamery.
- Kalibrační vzor by měl obsahovat alespoň 28 vnitřních bodů.

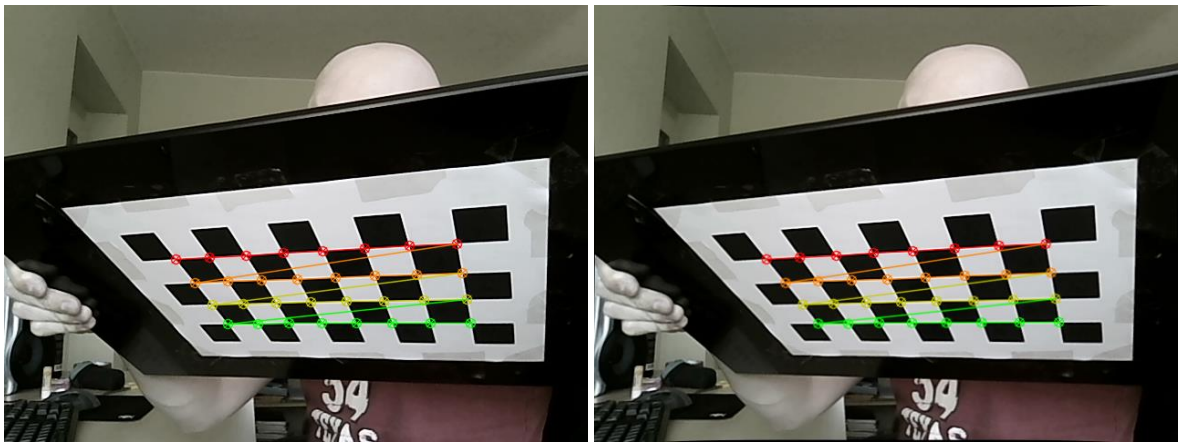
Splněním těchto podmínek lze předpokládat, že kvalita kalibrace kamery bude pro použití této práce na dostatečné úrovni.

	20 snímků	**40 snímků**	**60 snímků**	**80 snímků**
Re-projekční chyba	0,1-0,5	0,45-0,6	0,5-0,8	0,5-0,8
Doba zpracování [s]	1	15	35	85

[tab. 2] Závislost získaných parametrů na počtu vstupních snímků.

V implementovaném nástroji, je pro manuální kalibraci zvoleno použití 60 snímků. Pro takto velký vzorek dat se hodnota re-projekční chyby kalibrace pohybovala v rozmezí 0,5-0,8 bodu. Test probíhal dále s 20, 40 a 80 vzorky. Získané re-projekční chyby pro jednotlivé vzorky jsou spolu s dobou zpracování uvedené v **[tab. 2] Závislost získaných parametrů na počtu vstupních snímků.**[tab. 2]. Nejnižší chyba byla překvapivě získána u kalibrace pouze 20 vzorky. To bylo ovšem způsobeno nedostatečným pokrytím všech míst v rozsahu kamery. To často vedlo deformaci rohů v kalibrovaném snímku. Tento problém se u ostatních vzorků nevyskytoval, pokud byly splněny výše uvedené doporučení.

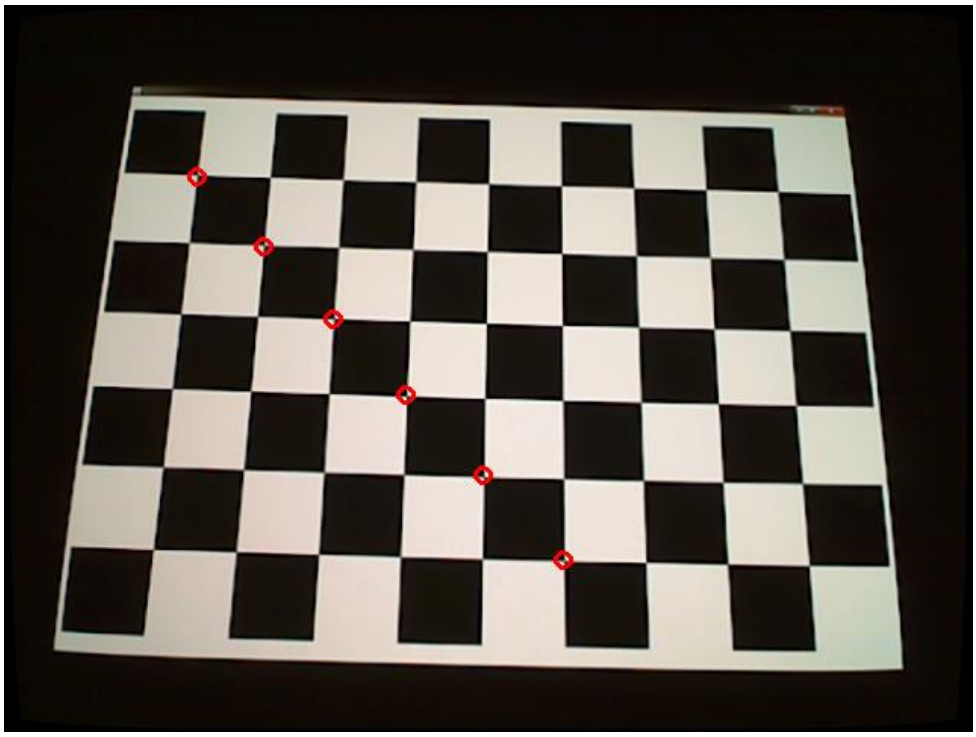
Doba výpočtu interních parametrů kamery roste společně s počtem snímků nepřímo úměrně. Ačkoliv s rostoucím počtem snímků kvalita kalibrace stoupá, velký časový rozdíl mezi 60 a 80 snímky rozhodl, pro využití 60 kalibračních snímků.



[obr. 37] Vlevo snímek bez provedení kalibrace, vpravo s kalibrací. Zkreslení obrazu u použité kamery není nijak výrazné. Rozdíl lze zpozorovat u horní hrany skleněné desky.

9.3 Kontrola homografie

Pro funkčnost aplikace je kalibrace projektoru vůči kameře zásadní. Kontrola správnosti získané homografie je prováděna promítnutím série bodů z roviny projektoru do roviny kamery [obr. **38**]. Z obrazu změřená odchylka mezi promítnutým bodem a rohem čtverce je v rozmezí 0-2 pixelů. Pro potřeby této aplikace je dosažená přesnost dostačující.



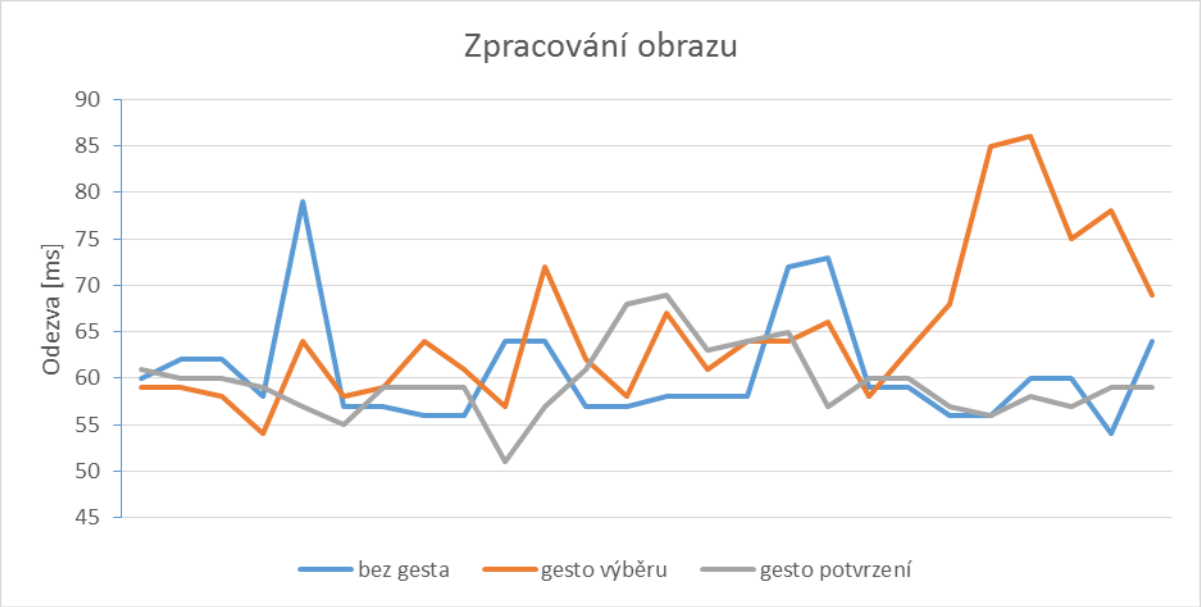
[obr. 38] Promítnuté značky by měly odpovídat přesně hranám na šachovnici.

9.4 Testování aplikace

První verze aplikace byla konzultována s Ing. Zadhrádkou (FIT VUT v Brně). Na základě námětů a postřehů bylo lehce upraveno GUI aplikace. Původní návrh GUI obsahoval herní kameny modré a červené barvy. Použití červené barvy kamene v některých situacích kolidovalo s barvou kůže. Jako vhodnější se jevilo použití barvy zelené, která má odlišný odstín proti barvě kůže. Pro lepší odezvu směrem k uživateli bylo také implementováno označení detekovaného gesta v promítané scéně. Při detekci gesta potvrzení se kolem detekované dlaně vykreslí kružnice žluté barvy. V případě detekce gesta vybrání je kolem detekovaného konečku prstu vykreslena malá žlutá kružnice. Uživatel tímto získá větší přehled o rozpoznání jeho gesta.

Samotné testování bylo zaměřeno na celkovou funkčnost aplikace a její odezvu vůči interakci uživatele. Testováním první verze aplikace byla změřena velká odezva na uživatelův vstup, průměrná hodnota odezvy činila 400ms. Pro ovládání aplikace v reálném čase se jedná o problém, interakce není v takovém případě komfortní. Původní návrh aplikace neobsahoval implementaci snímání scény kamerou, jako samostatné vlákno. Následkem bylo, že před každým pořízením snímku bylo potřeba otevřít kameru. Změřená režie pro otevření kamery se pohybuje kolem 300ms. Proto byla provedena změna implementace metody pro snímání scény. Nově byla metoda samostatné vlákno, kde nemusí být provedeno opětovné připojení ke kameře při zachycení dalšího snímku. Tímto krokem došlo

k razantnímu snížení odezvy aplikace. Po provedené korekci se změřená odezva pohybuje mezi 50-90ms, však průměrná hodnota odezvy činí 63ms. Časově nejnáročnější částí aplikace je zpracování obrazu [graf. 1]. Průměrná doba jeho zpracování je 61ms. Odezva ostatních částí systému činí v průměru 1,5ms. Pro pohodlné ovládání se jedná o dostatečně rychlou odezvu.



[graf. 1] Znázorňující rozdíly odezvy mezi detekcí jednotlivých gest ve snímku.

Funkčnost finální aplikace byla předvedena za účelem testování několika lidem, kteří se mohli do hry zapojit. Aplikace si s odlišnými tóny kůže poradila a její běh byl bezproblémový. Videoukázka běžící aplikace je k dispozici na přiloženém DVD.

Případným problémem by potenciálně mohlo být přímé sluneční světlo. Za těchto světelných podmínek není detekovaný odstín kůže ruky v požadovaném rozsahu a detekce je v tomto případě zhoršena. Jedná se o jednu z nevýhod použitého způsobu detekce rukou. Avšak při zatažení rolet na oknech již daný problém nebyl patrný. V noci za umělého osvětlení nebylo zhoršení detekce patrné.

9.5 Možné rozšíření

Případné možnosti rozšíření aplikace:

- Implementace automatické korekce rozsahu pro prahování snímku.
- Odstranění potvrzovacího gesta a jeho nahrazení formou posunu „Drag and Drop“ (detekce doteku plátna a tažení).
- Zjištění, zda došlo ve scéně k pohybu. Pokud ano, provést zpracování snímku, jinak neprovádět žádnou akci.

10 Závěr

Tato práce se zabývá tématem promítaného uživatelského rozhraní a možností interakce s rozhraním pomocí gest rukou. Uvedeny jsou zde principy zpracování obrazu a ukázky aplikací, jež jsou založeny na principu použití kamery pro snímání scény a projektoru za účelem promítání rozhraní aplikace. Obsažen je také popis kalibrace těchto zařízení, jež je důležitý pro korektní funkčnost výsledné aplikace. Uvedeny jsou také dostupné nástroje, jež je možné využít pro zpracování obrazu. Cílem práce je návrh a implementace volně zvolené deskové hry.

Implementovaná aplikace bude využívat systému projektor-kamera a nabídne nástroje pro kalibraci použité kamery a projektoru. Součástí aplikace má být také implementace nástroje pro detekci rukou a prstů v obraze. Cíl se podařilo splnit. Implementovaná desková hra „Mlýn“ nabízí rozhraní, jež je možné ovládat pomocí gest rukou. Hra na detekovaná ovládací gesta reaguje změnou stavu hry a následným vykreslením změny v promítaném rozhraní.

Funkčnost nástrojů pro kalibraci a zpracování obrazu. Kalibrace kamery a projektoru dosahuje dobrých výsledků a jejich využití v implementované hře je bezproblémové. Detekce rukou a prstů je obsažena v nástrojích pro zpracování obrazu. Detekce ovládacích gest funguje spolehlivě a přesně. Za určitých okolností však může dojít k chybné detekci rukou a prstů. Jednou z takových okolností je značné překrytí rukou promítaným obrazem nebo nevhodné světelné podmínky, kdy je scéna osvícena ostrým slunečním světlem. V takových případech dochází ke špatné segmentaci objektů v obrazu a následně jejich chybné detekci. Rozhraní hry je však navrženo tak, aby překrytí ruky promítaným obrazem nezpůsobovalo při detekci problémy.

Vytvořená desková hra nabízí rychlou interakci s uživatelem. Průměrná hodnota odezvy je 63ms. Značnou část odezvy tvoří systém pro zpracování obrazu, který se na době odezvy podílí 96%. Přesnost rozeznání místa, s kterým uživatel provádí interakci je přímo závislá na kvalitě provedené kalibrace snímků a nalezené homografii. Získaná přesnost je na dobré úrovni a nečiní problémy při hraní hry. Ukázka interakce s rozhraním je zachycena na videu a k dispozici na přiloženém DVD.

Výsledná aplikace může sloužit pro zábavní účely, kde může nahradit klasické stolní hry. Rozhraní hry je navrženo pro projekci na desku stolu, aby byla interakce s ní pro hráče pohodlná. Hráči poté můžou sedět kolem stolu a interakci provádět pomocí rukou, jako u klasické deskové hry.

V porovnání s existujícími aplikacemi systému kamery a projektoru je unikátní především interakce pomocí gest rukou a použití obyčejné kamery. Většina uvedených aplikací je ovládaná formou fyzických značek, které jsou detekovány kamerou. V případě rozšíření aplikace by bylo dobré využít pro snímání scény dvě kamery, aby bylo možné vytvořit hloubkovou mapu scény a rozeznat dotyk prstu s projekční plochou.

Literatura

[1] Wikipedie: Otevřená encyklopedie: *„Počítačové vidění“* [online]. 2010, aktualizováno 2014-2-17 [cit. 2015-05-12]. Dostupné na URL: <http://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A9_vid%C4%9Bn%C3%AD>

[2] Hlaváč, H.: Digitální zpracování obrazu × počítačové vidění zakotvení [online]. Praha: 2013, aktualizováno 2013-10-20 [cit. 2015-05-12]. Dostupné na URL: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/012IntroDigImProcVsCompVisionCz.pdf>

[3] Wikipedie: Otevřená encyklopedie: *„Computer Vision“* [online]. 2001, aktualizováno 2015-5-2 [cit. 2015-05-12]. Dostupné na URL: <http://en.wikipedia.org/wiki/Computer_vision>

[4] Brooks, R. A.: The Intelligent Room Project [online]. Cambridge: 1997 [cit. 2015-05-13]. Dostupné na URL: <http://people.csail.mit.edu/brooks/papers/aizu.pdf>

[5] Xiao, R., Harrison, C., Hudson, S. E.: WorldKit: Rapid and Easy Creation of Ad-hoc Interactive Applications on Everyday Surfaces [online]. Pittsburgh: 2013 [cit. 2015-05-13]. Dostupné na URL: <http://chrisharrison.net/projects/worldkit/worldkit.pdf>

[6] Willis, K. D. D., Shiratori, T., Mahler, M.: HideOut [online]. Pittsburgh: 2012 [cit. 2015-05-13]. Dostupné na URL: <http://www.disneyresearch.com/project/hideout/>

[7] Wiessler, M.: Lichtspiel [online]. 2014 [cit. 2015-05-13]. Dostupné na URL: <http://gaisterhand.de/#Lichtspiel>

[8] Radgohar, M.: Phong [online]. Fribourg: 2006 [cit. 2015-05-13]. Dostupné na URL: <https://diuf.unifr.ch/main/diva/download/overview/demos>

[9] Rojas, C., Orsten, S.: Projected Board Game Demo [online]. 2014 [cit. 2015-05-13]. Dostupné na URL: <http://artofmaking.tumblr.com/post/97605180705/projected-board-game-demo>

[10] Jones, B.: RoomAlive: Magical Experinces Enabled by Scalable, Adaptive Projector Camera Units [online]. 2014 [cit. 2015-05-13]. Dostupné na URL: <http://projection-mapping.org/roomalive-uist/ >

[11] OpenCV: OpenCV [online]. 2015 [cit. 2015-05-14]. Dostupné na URL: <http://opencv.org/>

[12] Brownlee, J.: Using OpenCV, Python and Template Matching to play „Where's Waldo?“ [online]. 2014 [cit. 2015-05-13]. Dostupné na URL: <http://machinelearningmastery.com/using-opencv-python-and-template-matching-to-play-wheres-waldo/>

[13] Andol, L.: An example of detecting object motions using OpenCV [online]. 2011 [cit. 2015-05-13]. Dostupné na URL: <http://www.andol.info/hci/1714.htm>

[14] Ball, T.: Train your own OpenCV Haar Classifier [online]. 2013 [cit. 2015-05-14]. Dostupné na URL: <http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>

[15] GitHub: opencv/data/haarcascades/haarcascade_frontalface.xml [online]. 2014 [cit. 2015-05-14]. Dostupné na URL: <https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_frontalcatface.xml>

[16] Kang, K. K., Nam, M. Y., Rhee, P. K.: Color Based Hand and Finger Detection Technology for User Interaction [online]. Washington: 2008 [cit. 2015-05-14]. Dostupné na URL: <http://dl.acm.org/citation.cfm?id=1438229>

[17] Wikipedie: Otevřená encyklopedie: *„Kinect“* [online]. 2010, aktualizováno 2015-2-26 [cit. 2015-05-15]. Dostupné na URL: <http://en.wikipedia.org/wiki/Kinect>

[18] Burrus, N.: Kinect RGB Demo v0.4.0 [online]. 2011 [cit. 2015-05-15]. Dostupné na URL: <http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV4?from=Research.KinectRgbDemo>

[19] Wikipedie: Otevřená encyklopedie: *„Camera resectioning“* [online]. 2006, aktualizováno 2014-12-28 [cit. 2015-05-16]. Dostupné na URL: <http://en.wikipedia.org/wiki/Camera_resectioning>

[20] Zahrádka, J.: Rozšířené uživatelské rozhraní [online]. Brno: 2011 [cit. 2015-05-16]. Dostupné na URL: <http://www.fit.vutbr.cz/study/DP/DP.php?id=11321&file=t>

[21] Král, E.: Analýza obrazové informace kamerového systému [online]. Zlín: 2006 [cit. 2015-05-16]. Dostupné na URL: <http://digilib.k.utb.cz/bitstream/handle/10563/1370/kr%C3%A1l_2006_dp.pdf?sequence=1>

[22] Gaura, J.: Odstranění geometrických zkreslení [online]. Ostrava: 2012 [cit. 2015-05-16]. Dostupné na URL: <http://mrl.cs.vsb.cz/people/gaura/dzo/geom_distortion_cz.pdf>

[23] Wikipedie: Otevřená encyklopedie: *„Mlýn (hra)“* [online]. 2008, aktualizováno 2015-4-19 [cit. 2015-05-17]. Dostupné na URL: <http://cs.wikipedia.org/wiki/Ml%C3%BDn_%28hra%29>

[24] Andresen, S.: Hand Tracking And Recognition with OpenCV [online]. 2013, [cit. 2015-05-18]. Dostupné na URL: <http://sa-cybernetics.github.io/blog/2013/08/12/hand-tracking-and-recognition-with-opencv/>

[25] OpenCV 2.4.11.0 documentation: Camera calibration With OpenCV [online]. 2011, [cit. 2015-05-18]. Dostupné na URL: <http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html>

Seznam příloh

Příloha 1. Manuál – Spuštění aplikace
Příloha 2. Manuál – Ovládání aplikace
Příloha 3. DVD

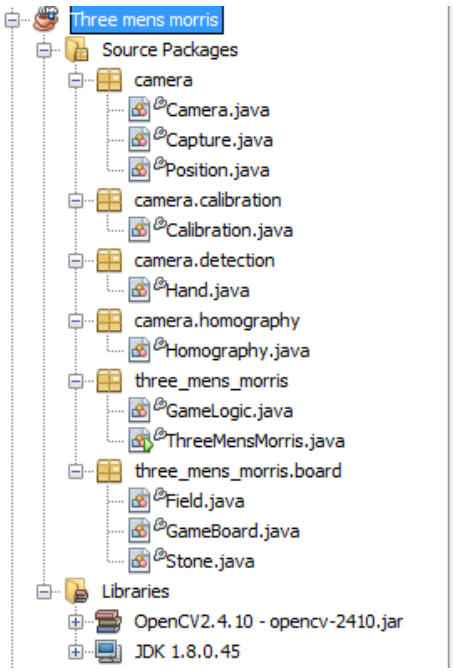
Příloha 1. Manuál – Spuštění aplikace

Návod pro spuštění aplikace v prostředí Windows:

Pro bezproblémový běh je potřeba nainstalovat do počítače knihovnu OpenCV, jež je přiložena na DVD. Instalaci knihovny umístit např. na systémový disk *„C:\opencv\“*.

Na DVD přiložený soubor „*Three_mens_morris.zip*“ jež obsahuje zdrojové kódy aplikace je potřeba importovat do programu NetBeans, ze kterého bude provedeno spuštění. Návod pro import:

1. Spustit aplikaci NetBeans.
2. Vybrat v záložkách: *„File -> Import Project -> From ZIP“*
3. V zobrazeném okně v položce *„ZIP File“* vybereme zip se zdrojovými kódy („*Three_mens_morris.zip*“) z přiloženého DVD. Do *„Folder“* umístění pro rozbalení projektu.
4. Jakmile je import dokončen, projekt se zobrazí ve stromu projektů v aplikaci NetBeans pod názvem „Three mens morris“.



[obr. 39] Importovaný projekt do programu NetBeans.

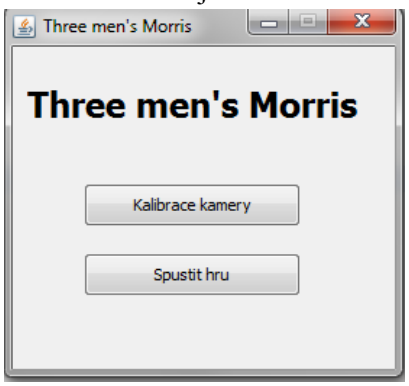
5. Nyní klikneme na projekt pravým tlačítkem myši a vybereme možnost „Properties“.
6. Vybereme položku *„Libraries“* a v záložce *„Compile“* klikneme na tlačítko *„Add JAR/Folder“*.
7. V zobrazeném okně vyhledáme soubor JAR, patřící knihovně OpenCV. Soubor JAR se nachází v adresáři, kde byla nainstalována knihovna OpenCV. *„C:\opencv\build\java\opencv-2410.jar“*. Nalezenou knihovnu přidáme pomocí *„Open“*.
8. Po kroku 7. by měla být knihovna OpenCV přidána do projektu.
9. Nyní v položce *„Run“* zvolíme parametr aplikace. Do kolonky *„VM Options“* připíšeme následující parametr: *„ -Djava.library.path="C:\opencv\build\java\x64" “*. Cestu ke složce

zvolíme podle adresáře, kam bylo nainstalováno OpenCV. Na základě verze operačního systému vybereme buď *x64* nebo *x86*.

10. Okno *„Properties“* ukončíme tlačítkem *„OK“*.
11. Nyní již je možné projekt spustit. Klikneme pravým tlačítkem myši na projekt a zvolíme možnost *„Run“*.

Příloha 2. Manuál – Ovládání aplikace

Po spuštění aplikace se objeví hlavní menu obsahující dvě možnosti (Kalibrace kamery, Spustit hru).



[obr. 40] Menu hry.

Kalibrace kamery

Stiskem tlačítka *„Kalibrace kamery“* je provedeno spuštění manuální kalibrace kamery. Na obrazovce se zobrazí okno s živým náhledem kamerou snímané scény, kde je zvýrazněn detekovaný kalibrační vzor [obr. 31]. Ihned po zobrazení je spuštěna detekce kalibračního vzoru (šachovnice), který je potřeba mít připevněn na rovném a neohebném povrchu. Ideální se jeví použití tabule skla. Při detekci je potřeba s kalibračním vzorem pohybovat, aby byla kalibrace kvalitní. S kamerou nepohybujeme. Pro ukončení snímání je potřeba získat 60 snímků šachovnice. Aktuální počet získaných snímků je zobrazen pod živým náhledem.

Po získání 60 snímků je spuštěn výpočet vnitřních parametrů kamery. Po dokončení kalibrace dojde k uzavření okna (asi po 35 sekundách). V tuto chvíli je kalibrace provedena. Vypočítané parametry kamery se vypíšou do konzole programu NetBeans.

Před samotným spuštěním procesu kalibrace je potřeba nastavit soubory *„calibration.xml“* a *„calibration_input_images.xml“*, jež jsou k dispozici v adresáři projektu ve složce *„camera_data“*.

Nastavení souboru calibration.xml

Ukázka obsahu souboru:

```
<?xml version="1.0" encoding="UTF-8"?>
<opencv_storage>
      <Settings>
            <!-- Pocet vnitrnich hodu pouzite sachovnice. -->
            <BoardSize_Width>8</BoardSize_Width>
            <BoardSize_Height>4</BoardSize_Height>
            <!-- Velikost ctverce sachovnice -->
            <Square_Size>28</Square_Size>
            <!-- Cesta k xml souboru, obsahující vsetupní snímky (pro
            automatickou kalibraci) -->
            <Input>"camera_data\calibration_input_images.xml"</Input>
      </Settings>
</opencv_storage>
```

V souboru je potřeba nastavit informace o šachovnici, jež bude použita ke kalibraci:

- *BoardSize_Width*: Počet vnitřních rohů šachovnice v orientaci na šířku.
- *BoardSize_Height*: Počet vnitřních rohů šachovnice v orientaci na výšku.
- *Square_Size*: Velikost čtverce na šachovnici (např. v mm).
- *Input*: Cesta k xml souboru, jež obsahuje vstupní snímky pro automatickou kalibraci. Není potřeba nastavovat v případě manuální kalibrace.

Nastavení souboru calibration_input_images.xml

Ukázka obsahu souboru:

```
<?xml version="1.0"?>
<opencv_storage>
      <images>images/image01.png</images>
      <images>images/image02.png</images>
      <images>images/image03.png</images>
</opencv_storage>
```

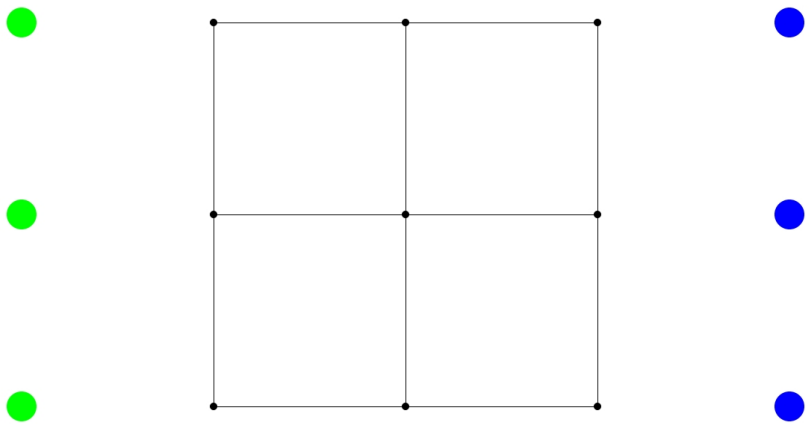
Soubor obsahuje seznam snímků, jež mají být použity pro automatickou kalibraci kamery. Automatická kalibrace kamery probíhá v případě spuštění hry, bez předchozího provedení manuální kalibrace kamery.

- *images*: Definuje cestu ke snímku.

Spuštění hry

Po spuštění hry tlačítkem *„Spustit hru“* se na obrazovce zobrazí nové okno. V případě, že nebyla před spuštěním provedena manuální kalibrace kamery, se automaticky spustí automatická kalibrace, která využívá definovaných snímků v souboru *„calibration_input_images.xml“*. Otevřené okno po dobu kalibrace neobsahuje žádný obsah. Doba automatické kalibrace závisí na počtu vstupních snímků, např. pro 40 snímků bude kalibrace trvat v průměru 20 sekund.

Poté, co je dokončena automatická kalibrace kamery nebo v případě, že byla provedena kalibrace manuální, se v okně zobrazí šachovnicový vzor. V tuto chvíli je potřeba umístit spuštěné okno do obrazu projektoru. Toto je potřeba provést ještě před začátkem hledání homografie, které se automaticky spustí do 5 sekund. Před promítanou plochou je potřeba mít umístěnu kameru, která provede snímek promítnuté šachovnice. Po úspěšném nalezení homografie mezi kamerou a projektorem se v okně zobrazí herní deska [obr. 41], hra je připravena. Od této chvíle se již nesmí s kamerou, projektorem, ani oknem hry pohybovat, aby nedošlo ke ztrátě kalibrace systému. V případě, že se po procesu hledání homografie nezobrazila v okně herní deska, došlo k chybě při výpočtu homografie. Při výskytu chyby se okno uzavře a zobrazí se opět hlavní menu. Při zaznamenání chyby je potřeba zkontrolovat, zda kamera snímá celou scénu (zobrazenou šachovnici) a zda je v promítacím okně viditelná celá šachovnice o počtu vnitřních rohů 9×6.



[obr. 41] Promítnuté rozhraní deskové hry Mlýn.

Ovládání hry

Po úspěšném spuštění hry, kdy se v okně zobrazí herní deska, je již možné provádět interakci s hrou pomocí rukou. Gesta pro ovládání jsou (podrobnější popis ovládání se nachází v kapitole 7.2):

- roztažení dlaně po potvrzení výběru
- ukázání prstem pro výběr

Prvním na tahu je hráč se zelenými kameny. Pro provedení tahu je potřeba provést následující posloupnost gest. Vybrání kamenu gestem pro výběr, vybraný kámen je zvýrazněn. Následný výběr potvrdit gestem potvrzení. Jakmile je vybraný kámen potvrzen, je potřeba vybrat gestem pro výběr, na jakou pozici na herní ploše bude umístěn. Vybraná pozice je opět zvýrazněna. Potvrzením vybrané pozice se provede přesun kamene na novou pozici. Hráčův tah je ukončen a na tahu je hráč druhý. V případě, že hráč získá mlýn, jsou kameny tvořící mlýn zvýrazněny a hra je zastavena. Po několika sekundách dojde k automatickému návratu do menu aplikace.

Příloha 3. DVD

Obsah DVD:

- Zdrojové kódy projektu.
- Plakát reprezentující tuto práci.
- Text této práce.
- Video prezentující interakci s vytvořenou deskovou hrou.
- Instalační soubor knihovny OpenCV.