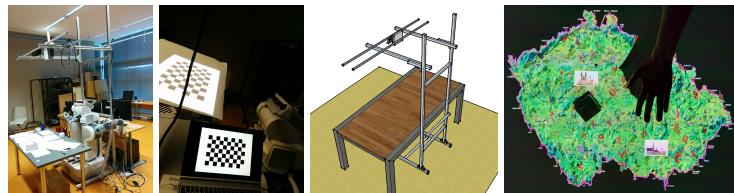


Deskové hry na inteligentním stole

Petr Svoboda*



Abstrakt

Tato práce se zaměřuje na využití principů hmatatelného uživatelského rozhraní (Tangible User Interface) a prostorové rozšířené reality (Spatial Augmented Reality) ke hraní deskových her. Za tímto účelem vznikl systém nazvaný inteligentní stůl. Jedná se zároveň o hardwarové i softwarové řešení. Hardwarová část řeší uchycení hloubkové kamery a projektoru nad stolem. Softwarové řešení se na jedné straně zabývá kalibrací systému hloubkové kamery a projektoru použitím známých postupů pro kalibraci kamery a na straně druhé řeší úlohy z oboru počítačového vidění. Tato percepční část je jádrem celé práce a pro účely hraní deskových her poskytuje informace o poloze fyzického herního předmětu, fyzické herní desky a výpočtu masky pro maskování rušivých objektů. Pro účely demonstrace možností systému vznikla výuková demonstrační aplikace. Inteligentní stůl nabízí nové možnosti hraní deskových her v reálném prostředí kombinací prvků rozšířené reality a reálných objektů a s tím spojené nové uživatelské zkušenosti.

Klíčová slova: spatial augmented reality, tangible user interface, projector-camera calibration, board games

Přiložené materiály: [Downloadable Code](#)

*xsobo91@stud.fit.vutbr.cz, Faculty of Information Technology, Brno University of Technology

1. Úvod

Příchod nových technologií (Microsoft Kinect, Nintendo Wii, Oculus Rift) velmi výrazně ovlivnil herní průmysl. Nové generace herních konzolí nabízejí nové možnosti ovládání a interakce s uživatelem. Hráč nyní může konzoli ovládat klasickým ovladačem s hapstickou odezvou, hlasem nebo samotným pohybem těla. Poslední zmíňovaný způsob ovládání reaguje na specifické hráčovy pohyby (dřepy, výskoky) audio-vizuální zpětnou vazbou podle typu hry a tím přináší do her nové zážitkové a multimediální prvky. Předchůdci těchto her, hry deskové, takovéto multimediální prvky nenabízí. Tato práce se snaží rozšířit možnosti hraní deskových her a to právě v oblasti multimédií a za použití podobných senzorů jako v případě konzolových her. Takovýto přístup je výzvou, neboť spojuje dva

různé světy hraní her. První, virtuální, s počítačem kreslenou grafikou a nedotykovým ovládáním a druhý, bez jakéhokoliv použití výpočetní techniky a kompletně založeného na ovládání pomocí pohybu s reálnými objekty. Cílem práce je tedy navrhnout a realizovat systém, inteligentní stůl, který využívá reálných objektů jako ovládacích prvků (Tangible User Interface) a dále poskytuje zpětnou vazbu pomocí počítačem kreslené grafiky se zvuky. Jako prezentace dosažených výsledků pak slouží výuková demonstrační aplikace zaměřená na dvě největší města Prahu a Brno a dále obsahující kvíz s tématikou pohoří České republiky. Deskové hry nabírají na inteligentním stole nový dech, jsou atraktivnější, nabízí podobné možnosti jako hry digitální a přitom si zachovávají zřejmě to nejdůležitější, sociální aspekt.

2. Existující řešení

Prostorová rozšířená realita (Spatial Augmented Reality) využívá projekci světla na reálné objekty. Mezi nejzajímavější a nejpodobnější existující projekty se řadí IllumiRoom [1] a RoomAlive [2]. IllumiRoom představuje obývací pokoj s televizorem a projektorem umístěným naproti televizoru. Projektor rozšiřuje obraz televizoru tím, že promítá obraz do okolí televizoru a vizuálním obsahem se přizpůsobuje obsahu programu televize (prezentace fotek, film, hra atd.). RoomA-



Obrázek 1. IllumiRoom při hraní her. Převzato z [3]

live tento koncept dále rozšiřuje. Projekt RoomAlive je místnost, ve které je umístěno několik projektörů a Kinectů tak, aby pokrývali celou místnost. RoomAlive nabízí několik her, které si mohou hráči v místnosti zahrát. Hry využívají objekty v místnosti, na které je promítáno herní prostředí (láva, řeka) a informace o poloze uživatele při provádění herních akcí (střelba). RoomAlive přímo využívá kalibraci projektoru a Kinectu, resp. kalibraci projektörů a Kinectů. Dalším principem využitým v rámci inteligentního



Obrázek 2. RoomAlive. Ukázka mapování virtuálního prostředí do fyzické místnosti a hraní hry. Převzato z [4]

stolu je hmatatelné uživatelské rozhraní (Tangible User Interface). To využívá jako prvky interakce mezi uživatelem a počítačem reálné objekty, které uživatel zná a sám používá (kostka). Reactable¹ je přístroj ve

tvaru bubnu, kde na jeho vrchní části je umístěn displej a herní kostky. Každá herní kostka představuje určité



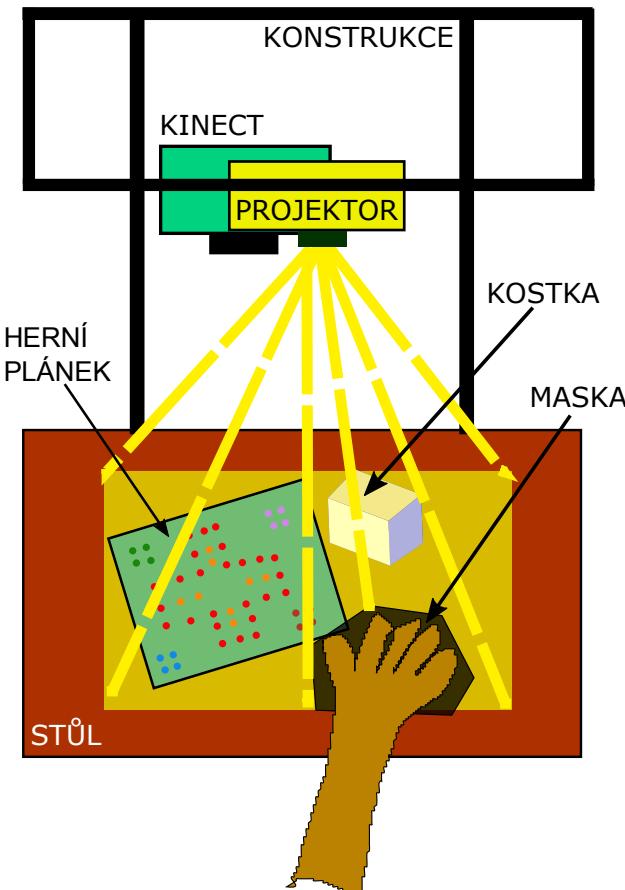
Obrázek 3. Reactable při vytváření rytmu. Převzato z 1

tempo nebo melodii a pomocí těchto kostek a jejich pozic na displeji se vytváří hudební doprovod.

3. Návrh inteligentního stolu

Ke hraní deskových her s rozšířenou realitou potřebuju systém, který sleduje probíhající hru na stole a adekvátně reaguje na změnu stavu herní scény. Aby byl systém schopen sledovat hru, musí umět detektovat herní desku umístěnou na stole a také herní předmět, který slouží jako hlavní ovládací prvek. Takovýto prvek by měl mít větší rozměry než např. herní figurka kvůli jednodušší a přesnější detekci. Ideální se jeví větší herní kostka (dále označovaná jako Kostka) kvůli jednoduchému tvaru a 6 stranám, které lze dál využít. Pro sledování herní scény je nejlepší volbou hloubková kamera Kinect. Využitím Kinectu získám kromě RGB obrazu i hloubková data, kde hloubkou rozumíme vzdálenost bodu od kamery. Kombinací RGB dat a hloubkových dat pak vzniká mračno bodů (pointcloud), ve kterém lze pro daný objekt určit jeho hloubku. O rozšířenou realitu navrhovaného systému se postará projektor, který bude promítat do herního prostředí virtuální grafické prvky. Nevýhodou promítání na stůl je to, že při manipulaci s Kostkou se uživatelovi na ruce promítá grafické prostředí, což může působit rušivě a kazit výsledný zážitek. Inteligentní stůl proto musí umět vytvořit masku kolem rušivého objektu tak, aby k tomuto nedocházelo jak je naznačeno na obr. 4. Dalším požadavkem na systém je schopnost promítat grafické prvky na jednotlivé části fyzického herního plánu podle bitmapy tohoto plánu uloženého v paměti počítače. Hlavním přínosem použití hloubkové kamery a projektoru je možnost tyto dva přístroje zkalibrovat. Díky kalibraci pak můžu přecházet z prostoru Kinectu do prostoru projektoru, což v praxi znamená, že jakýkoliv viditelný objekt

¹Reactable – Music Knowledge Technology, <http://reactable.com/>



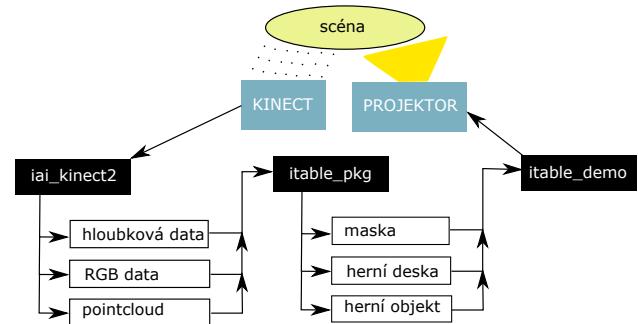
Obrázek 4. Herní scéna

kamerou mohu osvítit projektem. Praktickým problémem je uchycení Kinectu a projektoru do prostoru nad stolem. Konstrukce musí být nejen pevná, aby obě zařízení udržela, ale také stabilní a proto ji navrhuji zhotovit z hliníkových profilů, které by měly dané požadavky uspokojit. Celý systém je vhodné rozdělit do několika samostatných částí. První část se stará pouze o komunikaci se senzorem a získáváním dat a kalibračních údajů z něj. Druhá část pak tato data zpracovává a poskytuje výše zmíněné informace o herní scéně. Takto zpracovaná data jsou pak určena přímo pro použití v aplikacích.

4. Realizace systému

K implementaci jednotlivých částí systému zmíněných v kapitole 3 byl použit robotický operační systém ROS². Z pohledu ROSu pak na samostatné funkční celky nahlížíme jako na uzly (nodes), které mezi sebou komunikují pomocí kanálů (topics). Na tyto kanály mohou publikovat (publish) data a podobně mohou data z těchto kanálů odebírat (subscribe). Implementační a jiné informace o uzlu lze pak zabalit jako balík (package), který lze jednoduše distribuovat v rámci ROS ekosystému. Prvním balíkem použitým v řešení

je iai_kinect2 [5]. Ten představuje první krok řetězce zpracování dat jak je ilustrováno na obr. 5 a stará



Obrázek 5. Řetězec zpracování dat

se o získávání RGB dat, hloubkových dat a mračna bodů. Dále poskytuje kalibrační nástroje pro kalibraci hloubkové a RGB kamery. Dalším balíkem řetězce je pak itable_pkg, který je nejdůležitější součástí této práce a který se zaměřuje na zpracování dat z kamery. Funkčně jej lze rozdělit na dvě části, část pro kalibraci systému projektor a Kinect a na percepční část, která zpracovává data z kamery.



Obrázek 6. Konstrukce z hliníkových profilů osazená Kinectem a projektem

4.1 Kalibrace hloubkové kamery a projektoru

Princip kalibrace takového systému je založen na nalezení korespondujících 3D a 2D bodů. Dvoudimensionální body získáme promítáním známého a lehce

²The Robot Operating System www.ROS.org

detekovatelného vzoru, šachovnice, projektorem do prostoru (na stůl). Třídimenzionální body získáme nejprve určením bodů promítané šachovnice v RGB obrazu Kinectu. Vzhledem k tomu, že jsme provedli díky balíku iai_kinect2 kalibraci mezi hloubkovou a RGB kamerou, můžeme pro dané body šachovnice v RGB obrazu vyčist i jejich hloubku z hloubkové mapy. Tyto trojice pak převedeme na základě modelu dírkové kamery (Pinhole Camera Model) na body mračna bodů. Dvojice odpovídajících si 3D a 2D bodů pak slouží jako vstup do funkce calibrateCamera knihovny OpenCV³. Výsledkem je matice vnitřních parametrů (intrinsic matrix) systému projektor a kamery a dále rotační a transformační matice pro perspektivní projektci 3D bodů v souřadném systému Kinectu do 2D bodů v souřadném systému projektoru. Výše popsaná funkčnost byla implementována jako samostatný balík itable_calib.

4.2 Detekce a projekce na herní desku ve scéně

Cílem je nejen detektovat samotnou herní desku ve scéně, ale navíc i promítat na určité její části grafické prostředí. Tuto úlohu z prostředí počítačového vidění řeším pomocí výpočtu matice homografie, která popisuje perspektivní transformaci mezi dvěma rovinami. Podobně jako u kalibrace projektoru a kamery je zde nutné najít odpovídající si dvojice bodů. K tomu využívám nejprve SURF detektor pro výpočet klíčových bodů a následně SURF extraktor pro výpočet vektoru rysů. Pro porovnání obou vektoru rysů a získání odpovídajících si dvojic bodů jsem použil FLANN KNN matcher. Všechny uvedené výpočty využívají implementaci knihovny OpenCV.

4.3 Výpočet masky

Základem potřebným k výpočtu masky je rozsah vzdálenosti směrem od kamery ke stolu, ve které se má maska počítat. Ve svém řešení navrhoji celkem 3 režimy jak tento rozsah definovat. První režim, statický, počítá s pevně zadáným rozsahem hodnot. Druhý režim má rozsah definován na základě polohy herní desky na stole a jisté vzdálenosti nad stolem. Třetí režim je pak kombinací obou, kdy pokud je poloha herní desky známa, tak se použije druhý režim, v opačném případě se použije první režim. Body, které leží v tomto rozsahu hodnot se vyjmou z mračna bodů a transformují do prostoru projektoru. Nyní chceme kolem každé větší skupiny bodů vytvořit konvexní obálku. Navržená metoda promítne 3D body do obrazové roviny, využije morfologickou operaci dilatace k zacelení mezer mezi body, ke vzniklým shlukům

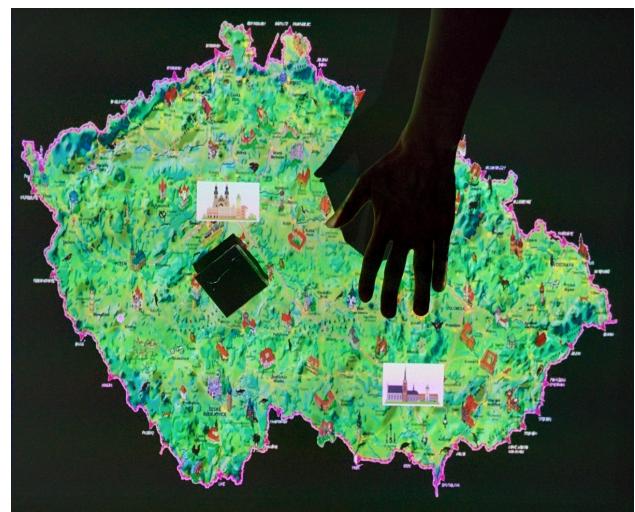
naleze kontury těchto shluků a vypočte konvexní obálku z nalezených kontur.

4.4 Detekce herní kostky

Před samotnou detekcí herní kostky je nutné mračno bodů vzhledem k jeho velikosti předzpracovat a tím snížit celkový počet bodů. K tomuto slouží struktura VoxelGrid implementovaná knihovnou PCL⁴. Podobně jako u výpočtu masky i u detekce herní kostky lze rozsah vzdálenosti určit pomocí 3 režimů. V podzorkované části mračna pak chceme naleznout největší plochu (stůl) a tuto plochu odstranit, aby ve výsledném mračnu bodů zbyly pouze objekty umístěné na stole. K nalezení parametrů roviny stolu jsem použil metodu RANSAC. Dalším krokem je pak extrakce jednotlivých shluků bodů ve zbývajícím mračnu tak, abychom dostali shluky odpovídající jednotlivým objektům na stole. Toto bylo dosaženo hledáním shluků v 3D bodech (Euclidian Cluster Extraction). Jednotlivé shluky bodů jsou dále porovnávány s modelem hrací kostky iterativním hledáním nejbližších bodů (Iterative Closest Point). Podle výsledného skóre je určen objekt, který je nejpravděpodobněji známá hrací kostka.

5. Výuková hra

Cílem aplikace je ukázat na jednoduché výukové hře možnosti inteligenčního stolu. Po spuštění aplikace se na stůl promítá mapa České republiky se zvýrazněnými městy Prahou a Brnem (viz obr. 7). Na herní Kostku se



Obrázek 7. Výuková hra na inteligenčním stole s detekovanou Kostkou a vykreslenou maskou kolem ruky.

grafika nepromítá a stejně tak se nepromítá na rušivé objekty mezi stolem a projektorem využitím masky. Kostka je polepená AR kódy, kde každý z nich má ve

³OpenCV <http://opencv.org/>

⁴PCL -Point Cloud Library <http://pointclouds.org/>

hře svůj význam. K detekci AR kódů jsem využil další ROS balík ar_tracky_alvar⁵. Pokud uživatel přiloží Kostku na obrázek jednoho z měst, pak se spustí akce podle významu AR kódu. Může to být film o daném městě nebo jednoduchá informační tabule. Na Kostce je i kód pro návrat na hlavní mapu. Opravdovou hrou v pravém slova smyslu je až kvíz, který se spouští natočením Kostky směrem ke Kinectu s posledním typem AR kódu. Pro hraní kvízu je nutné mít vytištěnou mapu České republiky umístěnou na stole. Postup hry je následující: nejprve se vypíše název pohoří a čeká se, až uživatel položí Kostku na fyzickou mapu tam, kde si myslí, že se dané pohoří nachází. Po ustálení polohy Kostky se rozhodne, zda ji uživatel položil na správnou pozici. Pokud ano, přehraje se oslavná znělka a vypíše se text „Správně!“, v opačném případě se spustí smutná melodie, vypíše se „Špatně“ a na mapě se vykreslí projektorem správná pozice. Při potvrzování každé volby, ať už se jedná o položení



Obrázek 8. Kvíz a reakce hry na nesprávné a správné určení polohy pohoří

ovládacího prvku na město nebo ustálení prvku u kvízu se kolem fyzické herní Kostky vykreslují postupně 4 červené čtverce zohledňující i rotaci fyzické Kostky jako vizuální zpětná vazba.

⁵http://wiki.ros.org/action/fullsearch/ar_track_alvar

6. Závěr

Tato práce se zaměřila na využití principů prostorové rozšířené reality (Spatial Augmented Reality) a hmatatelného uživatelského rozhraní (Tangible User Interface) pro hraní deskových her. Pro realizaci byla využita hloubková kamera Kinect a projektor, kteří byli zkalirováni známým postupem využívaným pro kalibraci kamery. Celé řešení využívá filosofie a možnosti robotického operačního systému ROS, jmenovitě implementace jednotlivých funkčních celků jako uzel v distribuovaném výpočetním systému. Uzel itable_pkg je hlavní součástí celého systému, zpracovává data Kinect a s využitím knihoven pro zpracování obrazu OpenCV a knihoven pro práci s mračny bodů, PCL, poskytuje dalším uzel informace o situaci na herním stole. Součástí práce byl i návrh a konstrukce hliníkového stojanu pro upevnění projektoru a Kinectu a tento stojan se softwarovou vrstvou itable_pkg je pak nazýván inteligentní stůl. Práce je zaměřena na využití inteligentního stolu pro hraní deskových her a výsledky prezentuje jako výukovou aplikaci. Výuková hra oživuje klasické herní schéma deskových her skládající se z herní desky a herních předmětů o rozšířenou realitu s multimediální zpětnou vazbu. Nový zážitek ze hraní vrcholí ve kvízu, kde uživatel lokalizuje pohoří České republiky pomocí reálné Kostky na fyzické mapě a podle jeho volby ho hra pochválí nebo opraví osvícením správné části fyzické mapy. Možnosti inteligentního stolu lze dále rozšířit jak přidáním dalších senzorů, tak zdokonalením použitých postupů, zejména detekce předmětů v mračnu bodů. Základním stavebním kamenem však zůstává zkalirovaný systém hloubkové kamery a projektoru, který nabízí do budoucna širokou paletu využití nejen pro hraní deskových her.

Poděkování

Děkuji panu Ing. Vítězslavovi Beranovi, Ph.D. za vedení, rady a motivaci při řešení této práce.

Literatura

- [1] Brett R Jones, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. Illumiroom: peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 869–878. ACM, 2013.
- [2] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. Roomalive: Magical experiences enabled by scalable, adaptive projector-

camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 637–644. ACM, 2014.

- [3] Brett Jones. Illumiroom: Immersive experiences beyond the tv screen, 2014.
- [4] Brett Jones. Roomalive: Magical experiences enabled by scalable, adaptive projector camera units, 2014.
- [5] Thiemo Wiedemeyer. IAI Kinect2. https://github.com/code-iai/iai_kinect2, 2014 – 2015. Accessed June 12, 2015.