



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ROZŠÍŘENÁ REALITA NA ANDROID - POTLAČENÍ
REÁLNÝCH OBJEKTŮ**

AUGMENTED REALITY ON ANDROID - HIDING REAL OBJECTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAROSLAV KATRUŠÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2020

Zadání bakalářské práce



Student: **Katrušák Jaroslav**
Program: Informační technologie
Název: **Rozšířená realita na Android - potlačení reálných objektů**
Augmented Reality on Android - Hiding Real Objects
Kategorie: Počítačová grafika

Zadání:

1. Seznamte se s problematikou rozšířené reality, zaměřte se na vývoj aplikací AR pro Android.
2. Navrhněte postup, jak ve scéně nějakým způsobem "skrýt" vybrané objekty a lépe tak vizualizovat virtuální objekty. Zaměřte se na vnitřní prostředí.
3. Navržený postup implementujte do uživatelské aplikace s využitím vhodných knihoven pro realizaci rozšířené reality na mobilním zařízení.
4. Vyhodnoťte vlastnosti výsledného řešení v reálném prostředí.
5. Prezentujte klíčové vlastnosti řešení formou plakátu a krátkého videa.

Literatura:

- Dieter Schmalstieg, Tobias Hollerer. *Augmented Reality: Principles and Practice*. Addison-Wesley, 2016. ISBN: 978-0321883575.
- Gary R. Bradski, Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*, ISBN 10: 0-596-51613-4, September 2008.
- Dále dle pokynu vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 a 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Beran Vítězslav, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 15. dubna 2020

Abstrakt

Cílem této bakalářské práce je návrh způsobu, jakým lze ve scéně rozšířené reality překrýt vybrané reálné objekty za účelem lepšího zobrazení virtuálních objektů a tento způsob následně implementovat v podobě aplikace pro operační systém Android. Výsledná aplikace, založená na technologii ARCore, umožňuje, kromě skrývání vybraných objektů, vkládání virtuálních objektů a následnou manipulaci s nimi. Práce se zaměřuje na návrh a implementaci takové aplikace a rozebírá vlastnosti zvoleného řešení.

Abstract

The goal of this bachelor thesis is design a way how to hide selected real objects to better display virtual objects in augmented reality and implement it in application for Android operating system. The resulting application, which is based on ARCore technology, allows, together with hiding selected objects, also insertion and subsequent handling of virtual objects. Thesis focuses on design and implementation of such an application and analyzes properties of the choosen solution.

Klíčová slova

Rozšířená realita, Android, ARCore, c#, model, mobilní aplikace, uživatelské rozhraní, skrývání objektů

Keywords

Augmented reality, Android, ARCore, c#, model, mobile application, user interface, hiding objects

Citace

KATRUŠÁK, Jaroslav. *Rozšířená realita na Android - potlačení reálných objektů*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vítězslav Beran, Ph.D.

Rozšířená realita na Android - potlačení reálných objektů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Jaroslav Katrušák

28. května 2020

Poděkování

Rád bych poděkoval panu Ing. Vítězslavu Beranovi, Ph.D. za veškeré rady a připomínky při psaní této práce.

Obsah

1	Úvod	2
2	Rozšířená realita a technologie ARCore	3
2.1	Rozšířená realita	3
2.2	Způsoby zobrazování rozšířené reality	4
2.3	Sledování polohy uživatele	6
2.4	Technologie ARCore	8
2.5	Přehled existujících aplikací	12
3	Návrh aplikace pro potlačení reálných a vkládání virtuálních objektů	17
3.1	Analýza řešeného problému	17
3.2	Možná řešení problému	20
3.3	Proces skrývání objektů	24
3.4	Uživatelské rozhraní	27
3.5	Manipulace s objekty	30
4	Implementace aplikace pro realističtější zobrazení virtuálních objektů	32
4.1	Volba programovacího jazyka, prostředí a platformy pro vývoj	32
4.2	Struktura aplikace	33
4.3	Výsledná aplikace	37
4.4	Testování	39
5	Závěr	47
	Literatura	48
A	Obsah přiloženého DVD	49

Kapitola 1

Úvod

Rozšířená realita, jejíž počátky lze datovat až do druhé poloviny dvacátého století, je obor, který zejména v tomto století zažívá rychlý rozvoj a začíná zasahovat do stále větší oblasti lidského života. Aniž si to mnozí uvědomují, tak se s rozšířenou realitou setkávají v každodenním životě například při sledování televizních přenosů sportovních událostí, v podobě výsledkových tabulek, nebo (při sledování) předpovědi počasí v podobě náhledů zobrazovaných před moderátorem.

Největší podíl na rozšíření povědomí o rozšířené realitě, v podobě interaktivních mobilních aplikací, však má rozmach chytrých telefonů. Tato zařízení nabízejí svým uživatelům vše potřebné pro rozšířenou realitu v podobě displeje, obrazového senzoru, výkonného procesoru pro zpracování obrazu a senzorů určujících polohu. Právě tyto vlastnosti přinášejí širokému spektru uživatelů do rukou nástroj, který je možné využít jako platformu pro aplikaci rozšířené reality ve vzdělávání, navigaci, marketingu, zábavě, nebo pro činnosti jako je měření rozměrů.

Cílem této práce je navrhnout způsob, jakým lze ve scéně rozšířené reality skrýt vybrané objekty a tento způsob následně implementovat v podobě aplikace pro operační systém Android. Kromě možnosti skrývání bude aplikace umožňovat do scény vkládat modely nábytku a následnou manipulaci s nimi.

Práce je rozdělena do 5 kapitol včetně úvodu. Kapitola 2. seznamuje čtenáře s pojmy týkajícími se tohoto tématu, jako jsou rozšířená realita, způsoby jejího zobrazování a sledování polohy uživatele. Kapitola se dále zabývá technologií ARCore využitou při realizaci aplikace a možnostmi dnes existujících aplikací s rozšířenou realitou pro operační systém Android. V kapitole 3. je popsán postup návrhu výsledné aplikace, který se skládá zejména z analýzy řešeného problému a návrhem jeho možných řešení. Pozornost je také věnována uživatelskému rozhraní a ovládání aplikace. Implementací výsledné aplikace se zabývá kapitola 4., která se zabývá volbou implementačního prostředí, strukturou implementované aplikace včetně některých jejích zajímavých detailů. Velká část kapitoly je věnována podobě výsledné aplikace a zejména pak jejího testování.

Kapitola 2

Rozšířená realita a technologie ARCore

Klíčovou technologií této práce je rozšířená realita, pro jejíž realizaci je potřeba dokázat zobrazovat virtuální objekty zarovnané s reálnými objekty tak, aby vytvářely dojem, že jsou součástí stejného prostoru. Dále je, pro realizaci rozšířené reality, potřeba lokalizovat zařízení sloužící pro její vizualizaci a porozumět okolnímu prostředí. Tato kapitola pojednává o základních stavebních prvcích rozšířené reality, kdy v první části této kapitoly je definován pojem rozšířená realita, následně jsou popsány technologie, které se používají pro její zobrazování. Rovněž jsou probrány přístupy používané ke sledování pozice uživatele v různých prostředích. Poslední část se věnuje knihovně ARCore¹ pro rozšířenou realitu používanou na zařízeních s operačním systémem Android². Zejména se věnuje způsobům, jakými je sledován pohyb uživatele, detekováno okolní prostředí a jak dohází k umístování virtuálních objektů.

2.1 Rozšířená realita

Rozšířená realita je technologie umožňující rozšíření reálného světa jeho doplněním o virtuální prvky. Na rozdíl od virtuální reality, která uživatele kompletně pohltí do virtuálního světa a uživatel tak nemá možnost vnímat reálné prostředí, je rozšířená realita [1] spojení dvou prostředí – skutečného světa a virtuálního světa (počítačem generovaného).

Vytvořený systém je interaktivní a pracuje v reálném čase. Objekty v obou světech jsou sledovány, aby bylo zajištěno jejich přesné umístění ve společném prostoru. Virtuální objekty mohou mít jak vizuální, tak i sluchovou, hmatovou nebo čichovou podobu stejně, jako reálné objekty. Virtuální objekty je také možné dělit na tzv. konstruktivní, jež doplňují reálný svět, nebo tzv. destruktivní, které naopak maskují některé části reálného světa.

Základním předpokladem pro fungování systému rozšířené reality je schopnost sledování pohybu uživatele a porozumění okolnímu prostředí, díky čemuž je poté možné umístovat virtuální objekty na pevně dané místo v prostoru a tím dát vzniknout výsledné scéně s rozšířenou realitou. Konceptem sledování polohy uživatele se blíže zabývá podkapitola 2.3. Porozumění okolnímu prostředí a umístování virtuálních objektů je věnován prostor v podkapitole 2.4 zabývající se základními koncepty rozšířené reality realizovanými na platformě

¹<https://developers.google.com/ar/discover>

²<https://www.android.com/>

ARCore, kdy většina, v této podkapitole, rozebraných konceptů platí za obecné koncepty rozšířené reality.

2.2 Způsoby zobrazování rozšířené reality

Možností, jak lze zobrazit rozšířenou realitu, je hned několik. Liší se od sebe jak způsoby, kterými dochází ke spojování reálného a virtuálního světa, tak i svým použitím. Technologickými prvky, které jednotlivé způsoby zobrazení mezi sebou spojují, jsou: procesor, senzory, displej (nebo jiné zobrazovací zařízení) a vstupní zařízení. Tato kapitola představuje tři různé způsoby zobrazování rozšířené reality, přičemž čerpá z článku Ronalda Azumy a kol. [2] a také z knihy Olivera Bimbera [3].

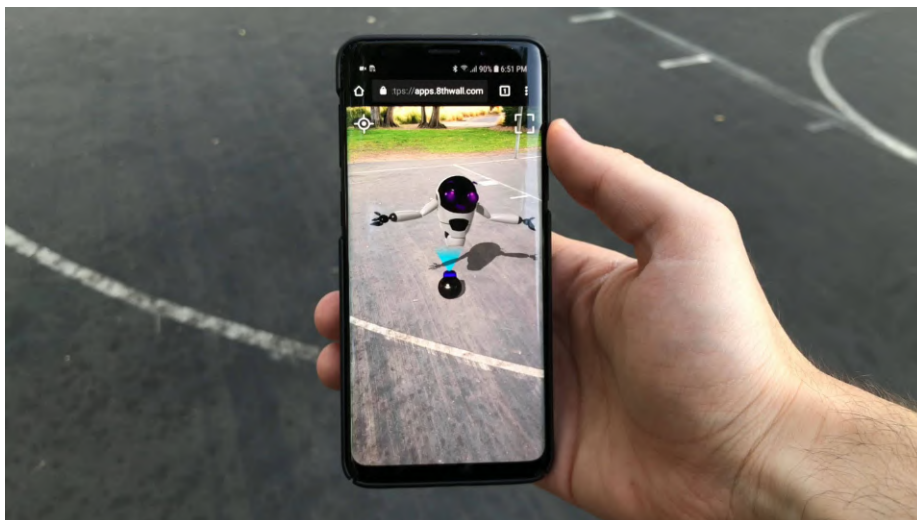
Displej mobilního zařízení

Za mobilní zařízení lze považovat takové zařízení, které je snadno přenosné a uživatel ho může mít neustále při sobě, ale nemá jej pevně umístěné někde na těle. V kontextu této práce se jedná o mobilní telefon, který uživatel při používání drží v rukou.

Mobilní telefon disponuje displejem, umístěným směrem k uživateli. Velikost displeje není nikterak stanovená, ovšem má nemalý vliv na zážitek z rozšířené reality, který nemusí být pozitivní v případě příliš malého displeje.

Typicky na zadní straně mobilního telefonu (je možné využít i přední kameru) se nachází kamera, která poskytuje potřebný obraz reálného světa. Další důležitou součástí jsou pohybové senzory a případně i GPS modul, které v kombinaci s obrazovými daty slouží k určování polohy zařízení v prostoru.

Obraz reálného prostředí získaný z kamery je následně, před svým zobrazením na displej, doplněn o virtuální objekty - obrázek 2.1. Velkou výhodou mobilních telefonů, pro účely rozšířené reality, je rozšířenost těchto zařízení mezi lidmi. Právě snadná dostupnost mobilních zařízení pro široké masy, jejichž technická výbava umožňuje využití pro účely zobrazování rozšířené reality, přináší velkou potenciální skupinu cílových uživatelů s mnoha oblastmi použití.



Obrázek 2.1: Rozšířená realita na mobilním zařízení (převzato z [9]).

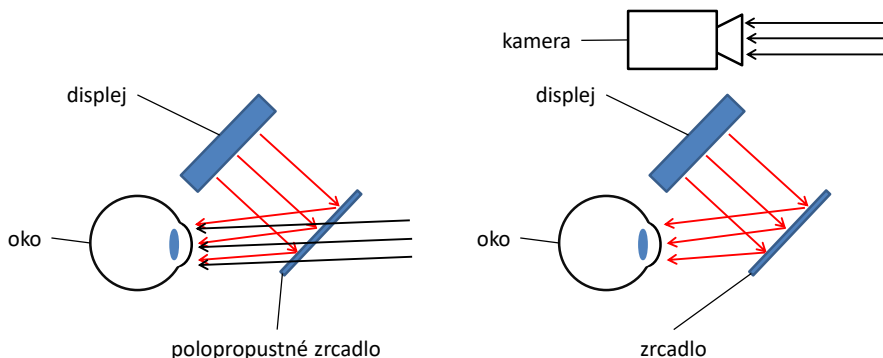
Nevýhoda zobrazování rozšířené reality na displej mobilního zařízení, plynoucí ze samotného způsobu fungování kamery a displeje, je možná ztráta hloubkového rozměru. Výsledné spojení reálného a virtuálního světa (které jsou oba trojrozměrné) je totiž na displeji zobrazováno pouze v podobě dvou rozměrného obrazu. Co se týče uživatelského komfortu, tak jistou nevýhodou může představovat potřeba držet mobilní zařízení, v průběhu používání, v ruce.

Displej umístěný na hlavu

Displej umístěný na hlavu [11] je zařízení, které má nejčastěji podobu brýlí, nebo je umístěné na přilbě. Obvykle se skládá z jednoho až dvou displejů, podle čehož lze daná zařízení rozdělit na monokulární (pro jedno oko), nebo binokulární (pro obě oči). Dalšími součástmi pak jsou polopropustné zrcadlo, nebo zrcadlo a kamera, kdy každá z těchto dvou variant využívá jiný princip zobrazování.

Zařízení využívající polopropustné zrcadlo umožňuje vkládání virtuálních objektů před obraz reálného prostředí, které je skrze toto zrcadlo sledováno. Na toto zrcadlo je promítán virtuální obraz z displeje, jenž se následně odráží na sítnici člověka, kde dochází ke spojení těchto dvou obrazů. Touto technikou je dosaženo zachování hloubkové informace o reálném prostředí.

V případě varianty zrcadla s kamerou je reálné prostředí sledováno kamerou umístěnou na úrovni očí uživatele. Získaný obraz je následně zpracován a rozšířen o virtuální část a tento celek je následně vykreslen na displej. Zrcadlo zde opět, tak jako v prvním případě, zajistí odraz tohoto obrazu na sítnici uživatele. Nevýhoda plynoucí ze samotného způsobu fungování této metody je ztráta hloubkové informace o reálném prostředí. Tento problém je však do určité míry možné eliminovat využitím dvou kamer – jedné pro každé oko. Obrázek 2.2 ukazuje rozdíl mezi variantou s poloprůhledným zrcadlem a variantou s kamerou.



Obrázek 2.2: Vlevo varianta s polopropustným zrcadlem, kde obraz reálného prostředí prochází přes toto zrcadlo, na které jsou promítány virtuální objekty. Vpravo varianta využívající kameru, která snímá reálné prostředí jež je následně, po doplnění o virtuální objekty, zobrazováno na displeji.

Oproti displeji mobilního zařízení přináší displej umístěný na hlavu výhodu v tom, že uživatel nemusí v ruce držet žádné zařízení. Navíc v případě zabudování tohoto systému např. do brýlí je také zachována vysoká diskrétnost, jelikož nemusí být patrné, že uživatel nějakou technologií aktuálně využívá. Ovšem takováto miniaturizace přináší i své nevýhody,

jako je nízký výkon hardware pro zpracování obrazu, nebo problém s napájením takového zařízení.

Projektor

Rozšířená realita, zobrazovaná za pomoci projektoru (obrázek 2.3) je, oproti již zmíněným způsobům, v určitých aspektech odlišná. Jedním z nejviditelnějších rozdílů je místo, na kterém dochází ke spojování reálného a virtuálního světa.

Jelikož je v tomto případě zařízením použitým pro zobrazování rozšířené reality projektor, tak dochází k promítání objektů z virtuálního světa na objekty z reálného světa. Uživatel tak výsledný obraz sleduje v reálném prostředí bez nutnosti držet v ruce, nebo mít někde na těle připevněné zobrazovací zařízení.

Protože je virtuální část rozšířené reality zobrazována do reálného světa, který je trojrozměrný, tak je potřeba v průběhu procesu mapování povrchu, sloužícího k promítání, analyzovat meze a hloubkový rozměr cílového objektu. Získané údaje slouží k úpravě promítaného obrazu tak, aby bylo dosaženo přesného zobrazení na daný objekt. Touto úpravou je dosaženo výsledného zobrazení se zachováním hloubkového rozměru reálného světa.

Ke správnému zobrazení rozšířené reality je tedy, kromě projektoru, nutné využít zařízení, jež zajistí mapování povrchu. Jedním z dostupných řešení je například zařízení nazvané LFC Kit³ od společnosti Lightform⁴.



Obrázek 2.3: Vlevo skutečná podoba boty použité pro zobrazení rozšířené reality prostřednictvím projektoru. Vpravo stejná bota na níž je prováděna projekce barevného vzoru (převzato z [8]).

2.3 Sledování polohy uživatele

Sledování polohy [10] a pohybu uživatele je jedním z nezbytných aspektů fungování rozšířené reality. Technologie pro sledování polohy a pohybu uživatele lze rozlišovat podle toho, jestli například umožňují sledovat více uživatelů současně či nikoliv. Další hlediskem může být skutečnost, zda je daná technologie určena pro sledování polohy v předem připraveném, nebo v libovolném prostoru. Každá z dostupných technologií má své silné i slabé stránky, přičemž pro nejlepší výsledky se v praxi používají kombinace různých technik, čímž je dosahováno potřebné přesnosti a spolehlivosti.

³<https://lightform.com/lfc>

⁴<https://lightform.com/>

Sledování zvenku dovnitř

Sledování zvenku dovnitř 2.4 lze charakterizovat použitím vysílačů typicky v uzavřené místnosti - je tedy zapotřebí prostor předem připravit. Pro určení pozice je v tomto případě použit přijímač, kterým může být přímo zařízení určené pro zobrazování rozšířené reality jako je mobilní zařízení, nebo displej umístěný na hlavě. Toto zařízení přijímá signál z vysílačů systematicky umístěných na několika různých místech, vůči kterým se měří vzdálenost. Ta může být určena na základě doby letu signálu od každého z vysílačů, nebo intenzity signálu. Pozice uživatele se následně, ze získaných vzdáleností, určí pomocí triangulace.

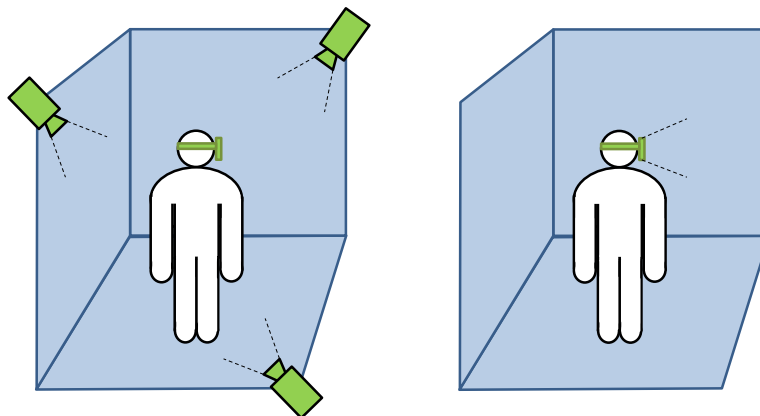
Výhodou tohoto přístupu je vysoká přesnost a spolehlivost určování pozice. Nevýhody této technologie vycházejí z její podstaty fungování a to v podobě potřeby dané prostory dopředu vybavit vysílači s čímž je spojeno omezení prostoru, ve kterém je možné pozici uživatele sledovat. Rovněž je nutné zajistit kompatibilitu vysílačů s přijímačem, což může omezit množství použitelných zobrazovacích zařízení.

Sledování zevnitř ven

Sledování zevnitř ven 2.4 přináší možnost sledovat pozici v libovolném prostoru, protože lze využít již existujících signálů v prostředí, jakými mohou být různé optické značky, vyhledávané pomocí počítačového vidění, nebo například magnetické pole Země.

Dalším přístupem je využití senzorů zrychlení v jednotlivých osách, z čehož lze integrací získat relativní pozici a rotaci. Toto umožňuje sledování v libovolném prostoru, ovšem přesnost určování pozice je závislá na přesnosti senzorů a typicky bývá nižší, než sledování pomocí aktivních vysílačů.

Výhodou je, kromě možnosti sledování v prakticky libovolném prostoru, možnost využít těchto technik sledování na všech zařízeních disponujících danými senzory. Navíc v případě, že je využíván i obraz z kamery pro lokalizaci, je možné dosáhnout velmi vysoké přesnosti zobrazování virtuálních objektů.



Obrázek 2.4: Vlevo varianta sledování zvenku dovnitř prostřednictvím externích senzorů rozmístěných po místnosti. Vpravo varianta sledování zevnitř ven, kdy je využíváno senzorů zařízení umístěného na uživateli.

2.4 Technologie ARCore

ARCore [6] je technologie od společnosti Google⁵, která je v současnosti používána na operačním systému Android⁶ pro realizaci aplikací s rozšířenou realitou. Představena byla jako nástupce projektu Tango. Oproti svému předchůdci nabízí zejména možnost použít v podstatě libovolné zařízení s operačním systémem Android, které splní určité podmínky.

Pro správné fungování ARCore musí dané zařízení disponovat operačním systémem Android verze 7.0, nebo novější. Kromě softwarové části kompatibility musejí mobilní zařízení podporující rozšířenou realitu splnit také požadavky na výkon procesoru, přesnost senzorů a disponovat kvalitní kamerou. Zařízení, která splní všechny požadavky, jsou certifikována jako podporovaná zařízení. Jejich úplný seznam je možné nalézt zde⁷.

V této kapitole budou dále rozepsány některé ze základních konceptů [7], které umožňují realizovat zážitky v rozšířené realitě prostřednictvím platformy ARCore.

Sledování pohybu

Sledování pohybu (*motion tracking*) využívá proces nazývaný simultánní lokalizace a mapování (zkráceně SLAM [4] - *Simultaneous Localization and Mapping*), pomocí kterého je zjišťována relativní pozice zařízení v reálném prostředí. Bez sledování polohy by nebylo možné zobrazovat virtuální objekty zarovnané s reálnými objekty tak, aby působily dojem, že patří do reálného prostředí.

ARCore pro zjišťování polohy používá detekci vizuálně význačných bodů ve snímcích poskytovaných kamerou, nazývaných *feature points*, na základě kterých je vypočítávána změna polohy. Zkombinováním těchto vizuálních informací a dat z inerciálních měření ze senzorů je zjištěna pozice a orientace kamery vůči reálnému prostředí (třída `Pose`). Prostřednictvím zarovnání virtuální kamery, zobrazující virtuální scénu, a kamery zařízení, sledující reálný svět, na základě relativní polohy a orientace, je virtuální obsah zobrazován ze správné perspektivy na snímky z kamery zařízení. Tímto je dosaženo výsledné scény, ve které se virtuální objekty jeví jako součást reálného světa.

Z programátorského hlediska je pro zarovnání využíváno toho, že po inicializaci subsystémů rozšířené reality je zařízení umístěno na výchozí souřadnice (0, 0, 0) v globálním souřadném systému. Posun, například o 1 metr ve směru osy *z*, způsobí ze strany ARCore úpravu polohy zařízení na souřadnice (0, 0, 1), přičemž polohu zařízení je nutné určovat po celou dobu běhu aplikace a to s dostatečnou přesností. Průběžné určování polohy znázorňuje obrázek 2.5.

Výhodou tohoto souřadného systému je jeho shoda se souřadnými systémy jakéhokoliv používaného 3D vykreslovacího enginu, kdy kamera daného enginu je umísťována na relativní pozici zařízení.

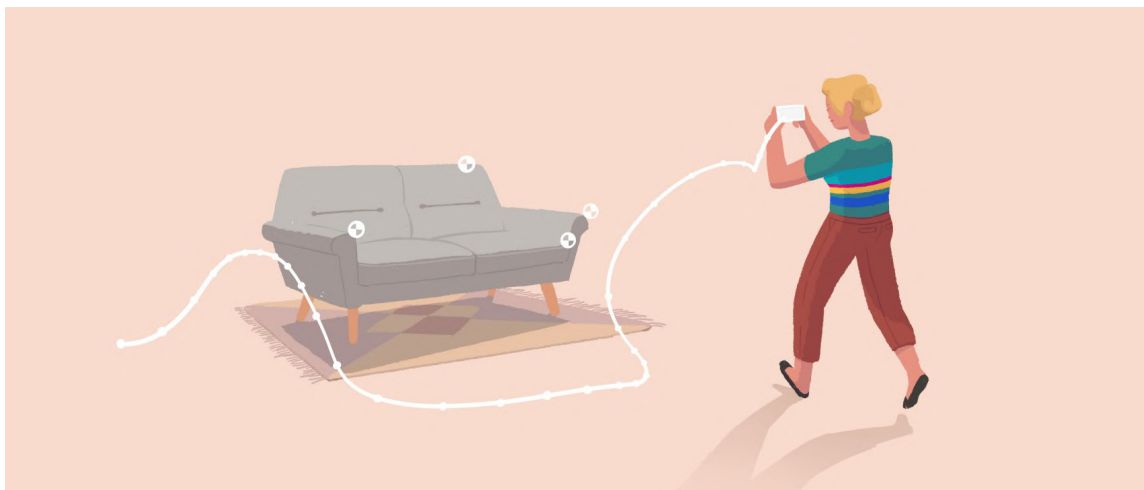
Porozumění prostředí

Mimo sledování pohybu zařízení knihovna ARCore řeší detekování a porozumění okolnímu prostředí (*Environmental understanding*). Okolní prostředí je mapováno v podobě tzv. *feature points*, což jsou vizuálně význačné body, nacházející se na objektech a površích s nehomogenní texturou jaká se nalézá například na dřevěných skříňkách, plovoucích podlahách nebo kobercích. Problémovými místy, pro hledání těchto bodů, jsou zejména jednobarevné

⁵<https://about.google/>

⁶<https://www.android.com/>

⁷<https://developers.google.com/ar/discover/supported-devices>



Obrázek 2.5: Obrázek znázorňující funkci relativního určování polohy. Trajektorie mobilního zařízení je znázorněná bílou křivkou, bílé body na pohovce znázorňují sledované body. Převzato z dokumentace ARCore [5].

stěny a další hladké jednobarevné povrchy, na kterých se nenachází vizuálně rozlišitelná místa.

Body nalezené na stejném horizontálním nebo vertikálním povrchu, například na podlaze, nebo stěnách tvoří shluky (*clusters*), které jsou dostupné aplikacím založeným na ARCore jako roviny (*planes*). O každé rovině jsou dostupné údaje o jejích hranicích, pozici, velikosti a typu povrchu.

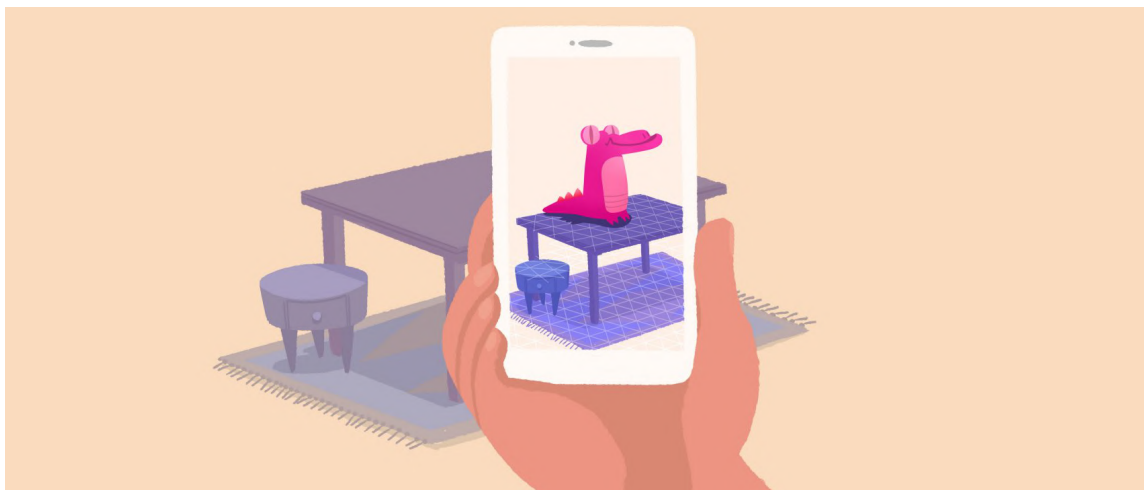
Pozice roviny je v ARCore reprezentována třídou *Pose*, v níž je poloha zaznamenána prostřednictvím souřadnic t_x, t_y, t_z a rotace roviny je zaznamenána v podobě quaternionu q_x, q_y, q_z, q_w .

Velikost je dostupná ve zjednodušené podobě jako výška h a šířka w . Typ roviny nabývá hodnot vertikální, horizontální natočená vzhůru (podlaha), horizontální natočená dolů (strop).

Detekované roviny slouží k umísťování virtuálních objektů do scény tak, aby vypadaly, jako by se přímo nacházely na daném místě – ležely přímo na daném povrchu - znázorněno na obrázku 2.6. Rovněž jsou detekované roviny využívány pro vykreslování stínů vrhaných na virtuální objekty reálnými objekty, což napomáhá poskytovat věrnější zobrazení.

Odhad osvětlení

K vykreslování virtuálních objektů prostřednictvím 3D engine je potřeba nastavit osvětlení, což může být prováděno manuálně, nebo automaticky. Odhad osvětlení (*Light estimation*) je jednou z důležitých funkcionalit poskytovaných knihovnou ARCore. Automatická detekce osvětlení umožňuje provádět korekci intenzity a barev v různých částech scény takovým způsobem, aby virtuální objekty vypadaly více realisticky a lépe zapadaly do reálného světa tak, jak je znázorněno na obrázku 2.7.



Obrázek 2.6: Detekce rovin ve scéně prostřednictvím ARCore. Detekované roviny jsou zvýrazněny trojúhelníkovou sítí na obrazovce mobilního zařízení. Převzato z dokumentace ARCore [5].

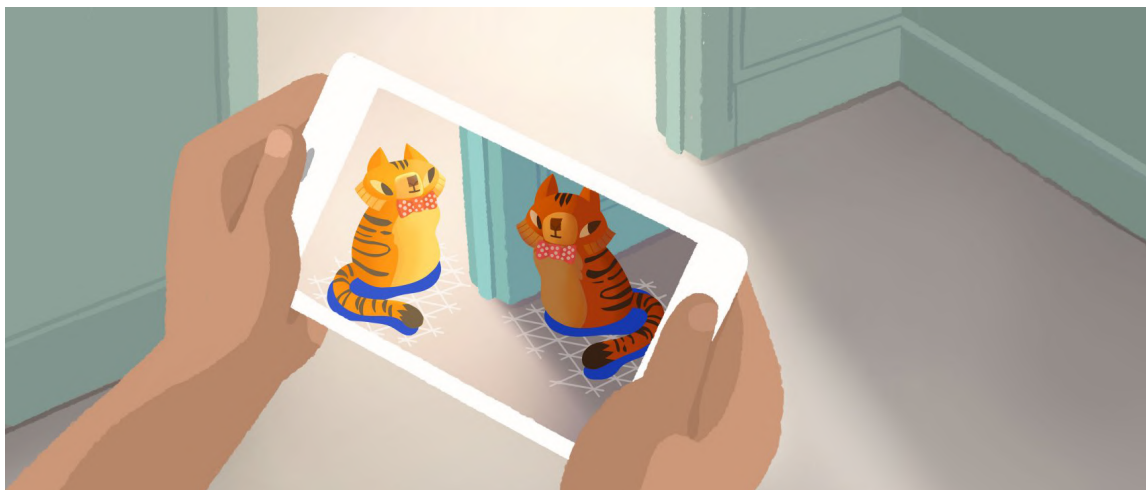
Umístění virtuálních objektů

Umístování virtuálních objektů do scény na určené místo v reálném prostředí je velmi často řešenou otázkou. Objekt je nutné umístit na pevně dané místo v reálném prostředí tak, aby bylo možné si scénu prohlížet z libovolného úhlu pohledu a případně s ní i interagovat.

Pro zajištění těchto požadavků jsou využívány tzv. kotvy (*anchors*), jež představují bod v prostoru, na němž bude umístěn virtuální objekt. Typickým příkladem použití je vytvoření kotvícího bodu na místě zásahu způsobeného uživatelem.

Testování zásahu (*hit testing*), což je rovněž funkcionalita poskytovaná knihovnou ARCore, na základě souřadnic (x, y) doteku na obrazovce mobilního zařízení, vyše paprsek (*ray*) pohledem na reálný svět, který je poskytován kamerou zařízení. Výsledek testování je ve formě rovin nebo feature pointů, které tento paprsek protne společně s pozicí průsečíků. Na místě průsečíku je typicky vytvořen kotvící bod, na který je připojen virtuální objekt, čímž se z kotvícího bodu stává rodič (*parent*) objektu. Právě provázání virtuálního objektu s kotvením bodem umožňuje knihovně ARCore sledovat pozici tohoto objektu v průběhu běhu aplikace.

Sledování pozice je důležité z hlediska možné úpravy pozice a rotace detekovaných rovin (*planes*) a feature pointů představujících speciální typ objektů souhrnně označovaných anglickým výrazem *trackables* (sledovatelné objekty). Tento typ objektů je sledován při běhu aplikace využívající knihovnu ARCore. V průběhu času běhu aplikace dochází jak k detekci nových ploch, tak i ke zpřesňování pozice, rotace a velikosti již detekovaných rovin. Pokud vložený virtuální objekt není ukotven na přesné místo detekované roviny pomocí kotvícího bodu, může při posunu roviny dojít k narušení zarovnání reálného a virtuálního světa, jelikož pozice objektu nebude změněna vzhledem ke změně pozice roviny. Vložený objekt se tak najednou může například vznášet nad deskou stolu, na níž byl původně umístěn.



Obrázek 2.7: V různých částech scény je rozdílné osvětlení pro realističtější vykreslení virtuálních objektů. Převzato z dokumentace ARCore [5].

Sdílená rozšířená realita

Pro účely sdílené rozšířené reality je možné v ARCore využít funkci *Cloud Anchors*⁸. Ta umožňuje více uživatelům interakci a sledování virtuálních objektů z různých úhlů pohledu ve stejném fyzickém prostředí. Využití sdílené rozšířené reality je možné zejména hrami pro více hráčů, nebo aplikacemi využívajícími spolupráci více uživatelů.

Základem této funkce je serverová služba *ARCore Cloud Anchor service*⁹. Pro správné fungování je nutné aby ARCore sestavilo 3D mapu (*feature map*) prostředí obklopujícího kotevní bod. Tato mapa bude vytvořena sledováním zájmové oblasti z různých úhlů a pozic po dobu 30 sekund. Vybraná vizuální data za posledních 30 sekund jsou prostřednictvím hostova volání (*host call*) odeslány na server. Ten ze získaných dat vytvoří 3D mapu a jako odpověď na volání vrátí unikátní *Cloud Anchor ID* pro vytvořený kotevní bod.

Ze zařízení druhého uživatele, které se nachází ve stejném prostředí jako zařízení uživatele, který vytvořil kotevní bod, jsou průběžně odesílány požadavky o porovnávání detekovaných vizuálních bodů na server. Na straně serveru je provedeno porovnání obdržovaných dat proti na serveru uložené 3D feature mapě. Jakmile dojde k nalezení shody mezi těmito daty – je zjištěno, že se oba uživatelé nacházejí ve stejném prostředí – jsou druhému uživateli odeslány údaje o vytvořených kotvících bodech.

Rozšířené obrázky

Jedná se o funkcionalitu umožňující vytvářet aplikace s rozšířenou realitou reagující na výskyt specifických obrázků třeba v podobě obalů, nebo plakátů. Při nalezení daného obrázku v obrazu kamery může být spuštěna předem definovaná akce. Například je možné při namíření fotoaparátu na obrázek v novinách spustit přehrávání videa a vytvořit tak interaktivní noviny.

Pro použití této funkce je nutné vytvořit databázi obrázků, která může obsahovat až 1 000 obrázků, zatímco počet těchto databází není limitován. Při nalezení obrázku začne jeho sledování prostřednictvím ARCore, které o detekovaném obrázku poskytuje údaje o pozici,

⁸<https://developers.google.com/ar/develop/java/cloud-anchors/overview-android>

⁹<https://console.cloud.google.com/apis/library/arcorecloudanchor.googleapis.com>

rotaci a jeho fyzické velikosti. Současně je možné sledovat pozici až 20 jedinečných obrázků, nacházejících se jak v zorném poli, tak i mimo zorné pole kamery.

2.5 Přehled existujících aplikací

Aplikací využívajících rozšířenou realitu je mnoho. Liší se od sebe svým účelem, kterým může být zábava, vzdělávání, marketing, případně usnadnění některých úkonů, a také tím, zda má daná aplikace reálné využití, nebo se jedná jen o demonstraci možností rozšířené reality. Aplikace, které budou zmíněny v této podkapitole jsem z hlediska toho, co u nich bylo sledováno rozdělil do dvou skupin - aplikace u nichž bylo sledováno zejména rozložení ovládacích prvků a aplikace u nichž byla sledována jejich funkcionalita. Pro účely vývoje vlastní aplikace byly kromě již zmíněných sledovány i možnosti interakce se scénou. Vyzkoušené aplikace byly voleny tak, aby ukázaly dnešní možnosti rozšířené reality, a přestože většinou je hlavním účelem těchto aplikací něco jejich uživateli usnadnit, tak problematika, kterou jednotlivé aplikace řeší, je často velmi rozdílná. Nejzajímavějším ze zmíněných aplikací bude věnován samostatný prostor pro podrobnější popis jejich funkce.

Aplikace u nichž bylo sledováno zejména rozložení ovládacích prvků

Níže zmíněné aplikace mají společné rozložení hlavních ovládacích prvků ve stylu aplikace fotoaparátu - v dolní části vertikální obrazovky. Takto umístěné ovládací prvky jsou snadno přístupné uživateli a působí ve scéně rozšířené reality nejméně rušivě, jelikož je střed obrazovky, na který se typicky zaměřuje pozornost uživatele, využíván pro zobrazování scény sledované uživatelem. Aplikace, které obsahovali větší množství ovládacích tlačítek, pak některá tlačítka měla umístěna i u horního okraje vertikální obrazovky. Umístění ovládacích prvků ve spodní a vrchní části obrazovky bude ze stejných důvodů využíváno i v případě implementované aplikace.

Aplikace z této skupiny často do obrazu reálného světa zobrazují zajímavé, avšak v mnoha situacích nevyužitelné informace. Jedná se například o zobrazování pozic navigačních satelitů aplikací **GPS Satellites Viewer**¹⁰, případně zobrazování informací o noční obloze tak, jako to provádí aplikace **SkyView Lite**¹¹. Mimo tyto aplikace byly v rámci této skupiny vyzkoušeny i aplikace, které mají, v případě většiny potencionálních uživatelů, lepší možnosti využití, jelikož se zaměřují na usnadnění nějaké činnosti. Takovými aplikacemi jsou **AR Ruler App**¹², sloužící pro měření rozměrů v obrazu reálného světa, a **Google Translate**¹³, umožňující překládat text snímáný kamerou v reálném čase do zvoleného jazyka.

SkyView Lite - jedná se o aplikaci, která do obrazu snímaného kamerou zařízení vkládá informace o hvězdné obloze - o hvězdách, souhvězdích a dalších tělesech. Ke svému správnému fungování vyžaduje zapnuté zjišťování polohy prostřednictvím systému GPS. Na základě zjištěné GPS polohy a natočení zařízení zjišťovaného prostřednictvím senzorů zrychlení - viz 2.3 je zjištěno, na jaké těleso na obloze uživatel zařízením míří a na obrazovce jsou následně zobrazeny příslušné informace o tomto tělese - obrázek 2.8. Aplikace při svém testování dosahovala z uživatelského pohledu plně dostačující přesnosti, co se pozice zobrazovaných informací a skutečné pozice daného tělesa na obloze týče.

¹⁰<https://play.google.com/store/apps/details?id=com.stgrdev.gpssatellitesviewer>

¹¹<https://play.google.com/store/apps/details?id=com.t11.skyviewfree>

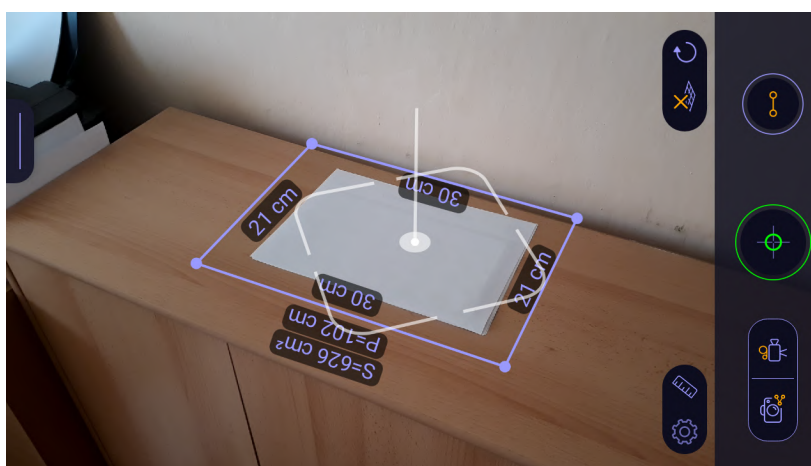
¹²<https://play.google.com/store/apps/details?id=com.grymala.aruler>

¹³<https://play.google.com/store/apps/details?id=com.google.android.apps.translate>



Obrázek 2.8: Po namíření fotoaparátu na souhvězdí Raka je na displeji toto souhvězdí vyznačeno a pojmenováno.

AR Ruler App - Tape Measure & Camera To Plan - aplikace sloužící pro měření rozměrů. Po nalezení rovin v obraze reálného světa umožňuje vybrat bod, od kterého se začne měřit a bod, po který se bude měřit. Následně je vypočítána vzdálenost mezi těmito dvěma body a ta je zobrazena na jejich spojnici. Dalším způsobem měření, které je k dispozici je automatické rozpoznání objektu, na který je namířen fotoaparát a jsou vypočítány jeho rozměry - viz obrázek 2.9. Aplikaci lze využít pro měření od nejrůznějších předmětů - stolů, skříní - až po celé místnosti. Jelikož aplikace vykazuje při měření odchylku průměrně do 10 %, v závislosti na světelných podmínkách a velikosti měřené vzdálenosti, tak je využitelná spíše pro orientační měření.

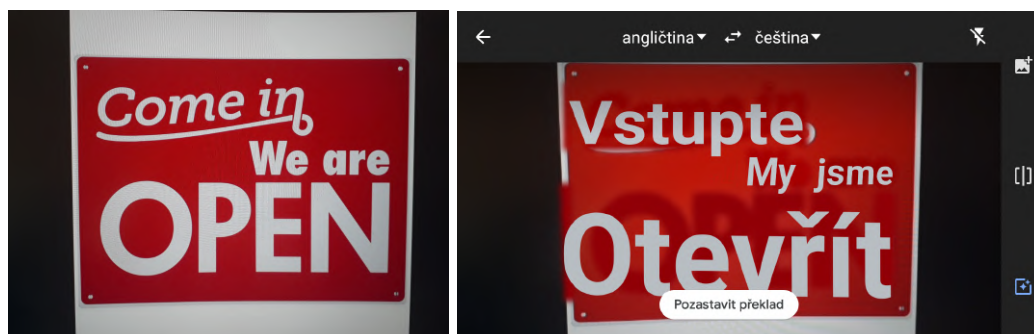


Obrázek 2.9: Automatické měření velikosti listu papíru formátu A4 prostřednictvím aplikace AR Ruler app.

Google Translate - jedná se o velmi rozšířenou aplikaci, která ve své základní podobě umožňuje překládat napsaný text do mnoha jiných jazyků. Samotný překlad je možné realizovat v online režimu, nebo v offline režimu prostřednictvím stažených jazykových sad. Z pohledu rozšířené reality je podstatné, že aplikace disponuje módem rozšířené reality, využívajícího kameru mobilního zařízení. Po jejím nasměrování na text dojde na obrazovce,

při nastavení správného zdrojového jazyka, v krátkém čase k překreslení snímaného textu přeloženým textem. Tento způsob překladu je vhodný pro překládání slovních spojení, nebo kratších textů - viz obrázek 2.10. Typickým problémem při překladu delších úseků textu je, že uživatel neudrží zařízení po dlouhou dobu na stejném místě a tento pohyb zařízením může vyvolat překreslování přeloženého textu, což při jeho větším množství chvíli trvá, než se text vrátí do čitelné podoby.

Pro účely překládání delšího textu je možné využít druhý mód, který je ovšem dostupný pouze v režimu online překladu, a tím je skenování. Zařízení je rovněž, jako v prvním případě, nasměrováno na text určený k přeložení, uživatel manuálně spustí skenování a po krátkém čase je zobrazen obraz, ve kterém jsou zářežkována všechna nalezená slova. Uživatel si následně může nechat přeložit vše nalezené, anebo si vybere úsek textu, který má být přeložen, pouhým přetažením prstu přes daná slova. Přeložený text je staticky zobrazen na displeji a mimo fázi skenování nemusí být zařízení namířeno na zdrojový text, díky čemuž je umožněno jednodušší čtení delších textů.



Obrázek 2.10: Obrázek vlevo ukazuje ceduli s anglickým nápisem vyzývajícím ke vstupu, jelikož je otevřeno. Vpravo je stejná cedule po provedení překladu prostřednictvím aplikace Google Translate - došlo zde k zamaskování původního textu červenou barvou.

Aplikace u kterých byla sledována jejich funkcionalita

Funkcionalita těchto aplikací byla sledována, poněvadž zmíněné aplikace nabízejí funkcionalitu, která bude obsažena i ve vytvářené aplikaci sloužící pro skrývání reálných objektů a vizualizaci virtuálních. Vytvářené aplikaci je svou funkcionalitou velmi blízká aplikace **IKEA Place**¹⁴, jelikož rovněž slouží pro vkládání a manipulaci s virtuálními objekty. Aplikace **Inkhunter - try tattoo designs**¹⁵, sloužící k vizualizaci tetování na vlastní kůži, ačkoliv se to nemusí jevit na první pohled, má s vyvíjenou aplikací také jednu společnou věc - skrývání reálných objektů. V podobném duchu však tuto funkcionalitu nabízí i mezi aplikace, u kterých bylo zkoumáno rozložení ovládacích prvků, zařazená aplikace **Google translate** - viz 2.5. Ta totiž v módu překladu v reálném čase (obrázek 2.10), za účelem toho, aby mohl být na stejné místo, kde se nachází původní text vykreslen překlad, provádí maskování původního textu.

IKEA Place - jedná se o aplikaci primárně sloužící pro účely marketingu od společnosti IKEA¹⁶. Aplikace umožňuje uživateli po naskenování cílového prostoru vkládat do obrazu reálného prostředí 3D modely nejrůznějších kusů nábytku a bytových doplňků z nabídky

¹⁴https://play.google.com/store/apps/details?id=com.inter_ikea.place

¹⁵<https://play.google.com/store/apps/details?id=tattoo.inkhunter>

¹⁶<https://www.ikea.com/cz/cs/>

společnosti IKEA - obrázek 2.11. S vloženými objekty je možné manipulovat ve smyslu změny jejich polohy a otáčení. Toto umožňuje uživateli si vyzkoušet, jak by produkt, o který má zájem, vypadal přímo v prostředí, do kterého má být umístěn a prohlédnout si jej bez nutnosti navštívit prodejnu.



Obrázek 2.11: Snímek obrazovky z aplikace IKEA Place, prostřednictvím které bylo do obrazu místnosti vloženo černé křeslo.

INKHUNTER - try tattoo designs - aplikace umožňující uživateli vyzkoušení si, jak by zvolené tetování vypadalo na jeho kůži. Pro svou funkci vyžaduje od uživatele, aby si na kůži na místo, kde má být tetování umístěno, nakreslil značku - na obrázku 2.12. Nakreslená značka je následně aplikací ve snímcích z kamery nalezena a překryta zvoleným tetováním. Díky využívání principu rozpoznávání specifických obrazců je dosahováno vysoké přesnosti zobrazování tetování na správné místo nezávisle na úhlu snímání mobilním zařízením, dokud je z daného úhlu viditelná celá značka. Přesné sledování polohy prostřednictvím značky také zajišťuje, vzhledem k úhlu pohledu, správné vykreslování perspektivy. Tetování tak z různých úhlů pohledu působí velmi realisticky a ve vytvořeném snímku scény se jeví, jako by bylo skutečně vytetované na kůži uživatele. Tomuto výsledku napomáhá i fakt, že aplikace provádí zamaskování značky nakreslené na kůži tak, aby tato značka nebyla viditelná pod vloženým tetováním - viz obrázek 2.12 vpravo.



Obrázek 2.12: Obrázek vlevo ukazuje značku namalovanou na kůži, zatímco na obrázku vpravo bylo na pozici této značky umístěno zvolené tetování a nakreslená značka byla zamaskována.

Kapitola 3

Návrh aplikace pro potlačení reálných a vkládání virtuálních objektů

Vkládání virtuálních objektů do scény je v rozšířené realitě často řešenou problematikou. Využití této funkcionality nachází v aplikacích sloužících zábavě, marketingu nebo vzdělávání.

Samotné vkládání objektů je, z programátorského hlediska, díky možnostem poskytovaným knihovnou ARCore, poměrně jednoduchá záležitost - viz 2.4. Základem pro vyřešení problému je detekce ploch zajišťovaná knihovnou společně s testováním zásahu, přes které je zvolený objekt umístěn na své místo. Jeho zobrazování ze správné perspektivy vzhledem k úhlu pohledu je pak již opět zajišťováno automaticky knihovnou.

V této kapitole bude v první části provedena analýza existujících aplikací pro vkládání virtuálních objektů, na kterých bude názorně vysvětlena řešená problematika, která se zabývá zlepšením vizualizace virtuálních objektů prostřednictvím potlačení reálných objektů. Následuje část zabývající se návrhem vlastního řešení v teoretické rovině. Poslední část kapitoly je věnována návrhu uživatelského rozhraní – rozložení jednotlivých ovládacích prvků a obrazovek.

3.1 Analýza řešeného problému

Aplikací, které vkládají virtuální objekty do scény je mnoho. Mezi nejznámější aplikace pro operační systém Android lze zařadit například **IKEA Place**¹, nebo **Augment**².

IKEA Place, blíže představená v kapitole 2.5 je aplikací, která bude v této kapitole využita pro představení řešeného problému, souvisejícího se zobrazováním virtuálních objektů, a stanovení základních požadavků na jeho řešení.

Na obrázku 3.1 je vidět případ, kdy virtuální objekt (hnědé křeslo) správně zapadá do scény – působí jako součást dané místnosti. Tohoto vjemu je dosaženo umístěním virtuálního křesla na prázdné místo – koberec, na kterém se nic nenachází. Je tak možné si nerušeně vyzkoušet, jak by zvolené křeslo vypadalo v dané místnosti.

Bohužel ne vždy je scéna pro rozšířenou realitu přívětivá, jako v předchozím případě. Pokud by uživatel aplikace již nějaké křeslo vlastnil a bylo by umístěné na stejném místě,

¹https://play.google.com/store/apps/details?id=com.inter_ikea.place&hl=cs

²<https://play.google.com/store/apps/details?id=com.ar.augment&hl=cs>



Obrázek 3.1: Snímek obrazovky z aplikace **IKEA Place** s vloženým modelem hnědého křesla.

na jakém se nyní nachází virtuální červené křeslo, dojde k problémové situaci – obrázek reálného křesla [3.2](#).



Obrázek 3.2: V místnosti se nyní nachází skutečné křeslo na místě, kde má být umístěno hnědé virtuální.

Problémem je, z hlediska principu rozšířené reality, právě ona přítomnost reálného objektu na místě, kde se má nacházet virtuální objekt. Virtuální objekty jsou totiž v rozšířené realitě vkládány na snímky reálného světa. Tato vlastnost nepředstavuje problém, pokud jsou virtuální objekty pokládány na povrchy reálných objektů – na podlahu, stěny, nebo stůl. Pokud však má být virtuální objekt umístěn na místo, na kterém se již nachází reálný objekt, tak vyvstává otázka, jak tuto situaci vyřešit.

Pokud se na tento problém podíváme pouze z pohledu jedné reality – skutečného světa, tak pokud si domů uživatel přinese nové křeslo, tak jej nemůže umístit na místo, na kterém se již nějaké křeslo nachází. Je možné jej umístit na místo, které je co nejbližší určenému místu, například vedle již přítomného křesla. Z fyzického hlediska jednoduše není možné v reálném světě umístit dva objekty na jedno místo, může tam být umístěn jeden, nebo druhý. Pro

přesné vyzkoušení, jak bude nové křeslo na vybraném místě vypadat, pak nezbyvá nic jiného, než původní křeslo fyzicky přemístit a nahradit jej novým.

Podobný postup je možné aplikovat i v případě použití rozšířené reality pro vyzkoušení virtuálního modelu křesla – přesunout reálné křeslo a na jeho místo vložit virtuální, nebo virtuální křeslo umístit vedle skutečného. Jak ale vyzkoušet vybrané křeslo, pokud uživatel nechce, nebo nemůže daný nábytek jednoduše přesunout a k přemístění těchto objektů by se uchýlil až ve chvíli, kdy je bude fyzicky nahrazovat?

Zde se nabízí myšlenka, že když aplikace **IKEA Place** umožňuje vyzkoušet si vybraný kus nábytku u sebe doma, aniž by uživatel musel si daný nábytek dopravit domů, zda není možné jej umístit na místo, kde se nachází skutečný nábytek.

Vzhledem k možnostem a principu fungování aplikace **IKEA Place** je možné, v určitých situacích, takového umístění dosáhnout. Uživatel aplikace je po jejím spuštění vyzván k naskenování podlahy místnosti, na kterou jsou následně umísťovány zvolené virtuální objekty. Naskenovaná podlaha pokrývá prakticky celou místnost a aplikace nebere přítomné objekty, nacházející se na podlaze, v potaz. Díky tomuto je možné umístit virtuální křeslo na místo na podlaze, na kterém se v reálném světě nachází skutečné křeslo.

Tato situace je znázorněna na obrázku 3.3, na kterém se nacházejí obě křesla na jednom místě. Poněvadž jsou při zpracovávání snímků, zobrazovaných na displeji, virtuální objekty vkládány na snímky z kamery, tak se virtuální objekty nacházejí nad reálným prostředím a dochází k jeho překrývání. Na ukázkovém obrázku jsou však stále viditelné některé části reálného křesla a zobrazení tak nebudí dojem, jakoby hnědé křeslo bylo skutečnou součástí interiéru.



Obrázek 3.3: Virtuální hnědé křeslo je umístěno na stejné místo jako reálné křeslo, přičemž dochází k jejich překrývání.

Shrnutím výše zmíněných skutečností lze stanovit požadavky na navrhovanou aplikaci. Nejpodstatnější funkcionalitou poskytovanou touto aplikací bude možnost vkládání virtuálních objektů do sledovaného prostoru. S vloženými objekty bude zapotřebí manipulovat, aby bylo uživateli umožněno provádět dodatečné úpravy libovolného virtuálního objektu a vyzkoušet si tak různé možnosti jeho umístění. Proto bude potřeba umožnit výběr objektu, dále provádět změnu jeho pozice, natočení a velikosti. Za účelem nerušeného zobrazení virtuálních objektů musí aplikace poskytovat způsob, jakým bude možné zamaskovat skutečné objekty ve scéně.

3.2 Možná řešení problému

Jak tento problém zobrazení vyřešit? Například z pohledu uživatele aplikace **IKEA Place** je možné daný reálný objekt odstranit fyzicky - přemístit jej mimo scénu, ovšem to vyžaduje jisté nepohodlí s používáním aplikace a nebude to v této situaci provedeno. Z technického hlediska je možné podobného efektu odstranění dosáhnout využitím pokročilých metod, například kdy bude celá scéna reprezentována jejím 3D modelem (polygonální sítí) a na polygony, které reprezentují objekt určený k odstranění bude vykreslována textura polygonů nacházejících se ve fyzickém prostoru za daným objektem. Tato práce se však bude zaměřovat na využití jednodušších principů pro maskování objektů a také na jejich vlastnosti.

Vzhledem ke skutečnosti, že virtuální objekty ve scéně zakrývají skutečné objekty, se nabízí možnost této vlastnosti využít za účelem skrytí reálného objektu. Této vlastnosti je v teoretické rovině je možné využít více způsoby, některé z nich jsou lépe a některé hůře aplikovatelné z hlediska uživatelského komfortu a možností knihoven pro rozšířenou realitu. V průběhu návrhu i následné implementace bylo do různých fází rozpracování dovedeno více způsobů s cílem zamaskovat reálné objekty, přičemž jeden z nich bude, společně se zvoleným způsobem, nastíněn v této kapitole.

Úvod do obou způsobů

Zvažované způsoby zakrytí byly založeny na využití ploch detekovaných knihovnou ARCore. Ta zajišťuje ihned po zapnutí aplikace vyhledávání jak vertikálních, tak i horizontálních rovin viz kapitola 2.4. Princip funkce bude vysvětlen na snímcích obrazovky z jedné z realizovaných mini aplikací, které byly v průběhu vývoje realizovány. Účelem této konkrétní aplikace bylo, kromě vizualizace rovin a detekovaných bodů, získaných přes ARCore, počítání plochy detekovaných rovin.

Na obrázku 3.4 je možné vidět dřevěnou skříňku, na jejímž povrchu bylo detekováno několik rovin. Vertikální rovina představuje přední stranu s dvířky, horizontální rovina je pak detekována na její horní desce. Tyto roviny pokrývají v podstatě celou skříňku, nepokrývají ji však zcela přesně, místy dochází k přesahům. Tyto přesahy jsou projevem nepřesné detekce, kdy na druhou stranu je možné narazit i na opačný případ – a to, že sledovaná plocha není detekována celá. Oba zmíněné jevy se při detekci rovin prostřednictvím knihovny ARCore vyskytují poměrně často, ačkoliv případ s rovinami přesahující sledovanou plochu je přeci jen častější. Tyto jevy jsou způsobovány více faktory, jakými jsou:

- přesnost senzorů použitého mobilního zařízení
- kamera zařízení
- úroveň osvětlení v sledované oblasti
- textura daného objektu a vzdálenost od něj
- algoritmy implementované v knihovně ARCore pro detekci rovin

Způsob využívající 2D masku

Úvodní myšlenka spočívala ve využití detekovaných rovin pro určení oblastí, ve snímku reálného prostředí z kamery, ve které se nachází zmíněný objekt. Uživatel aplikace by označil



Obrázek 3.4: Snímek obrazovky aplikace zobrazující detekované roviny - oranžově ohraničené polygony. Detekované vizuálně význačné body jsou označeny červenými čtverečky.

roviny, které představují objekt určený ke skrytí. Ze získaných rovin by byla vytvořena 2D maska pokrývající oblast objektu a s její pomocí by byla určená část scény překreslena zvolenou texturou, čímž by došlo k zakrytí daného objektu – znázorněno na obrázku 3.5.

Tato metoda však má několik problémů. Nejvýraznějším z nich je skutečnost, že dochází ke skrývání 3D objektu prostřednictvím 2D masky, kdy výsledek zakrytí nepůsobí příliš přirozeným dojmem. Je otázkou, zda je dosažený výsledek uspokojivější, než v případě, kdy by skutečný objekt nebyl vůbec zakryt a na jeho místo byl vložen virtuální objekt, tak jako je znázorněno na obrázku 3.3.

Další problém se nalézá v detekci rovin. Testovaná skříňka představuje pro knihovnu ARCore poměrně snadný úkol pro analýzu, jelikož je tvořena pouze horizontálními a vertikálními povrchy. Počet teoreticky detekovatelných ploch, ze kterých se daná skříňka skládá, je pouze 6, v závislosti na jejím umístění a úhlu pohledu i méně. V ukázkovém případě 3.4 byla skříňka detekována v podobě horizontální a vertikální roviny, kdy obě roviny přesahují meze skutečné skříňky a dochází tak k zakrytí větší části scény, než je nezbytně nutné.

V případě, že objektem určeným ke skrytí bude například křeslo, nebo pohovka s komplikovanějšími tvary – zejména s různými rozměry hloubky, bude detekce rovin poněkud problematická. Na obrázku 3.6 se nachází pohovka, na které byly detekovány dvě roviny. Horizontální rovina představuje sedací sekci a vertikální opěrnou, které společně představují značnou část plochy pohovky. Problém představují opěrky na předloktí a jiné malé výrazně tvarované části, které nejsou velmi často detekovány, tudíž by nebyly skryty a část pohovky by tak ve scéně zůstala.

Kombinace výše zmíněných nedostatků navrženého řešení vedla na přehodnocení způsobu, jakým budou objekty skrývány.



Obrázek 3.5: Obrázek znázorňuje situaci, kdy byla na základě detekovaných rovin vytvořena barevná maska, kterou byla překreslena oblast ve scéně, kde se nachází skříňka.

Způsob využívající virtuální objekt

Z mnoha hledisek je nově navržený způsob skrývání objektů podobný předchozímu. Rovněž využívá detekci rovin poskytovanou knihovnou ARCore a vlastnosti, že reálné objekty jsou zobrazovány pod virtuálními. Hlavní rozdíl je možné nalézt v přenesení větší části zodpovědnosti a tím i kontroly na uživatele.

Jako názornou situaci pro popis principu této metody bude použit obrázek 3.2. Zde se v místnosti nalézá reálné křeslo, které má být skryto, aby mohlo být zobrazeno virtuální. Jak již bylo zmíněno u způsobu využívajícího 2D masku, viz 3.2, tak samotný tvar tohoto křesla by byl poněkud komplikovaný pro použití způsobu využívajícího 2D masku.

Nově navržená metoda spočívá ve vykreslení uživatelem definovaného polygonu - virtuálního objektu, pro zjednodušení definovaného čtyřmi body, který zakryje oblast mezi těmito body. V ukázkové situaci bude naskenován zájmový prostor a podlaha, která se nalézá pod křeslem, bude detekována jako horizontální rovina. Na ní budou uživatelem zvoleny body, mezi kterými je následně vykreslena virtuální podlaha, která díky tomu, že je vykreslována na snímky z kamery ve výsledném obraze zakrývá zmíněné křeslo. Virtuální podlaha, případně stěna (dále jen clona) může mít barvu, případně i texturu odpovídající skutečné podlaze a ve výsledku tak vzbuzovat dojem, že je součástí reálného prostředí.

Tento způsob přináší výhodu v tom, že je možné si přímo zvolit, co má být skryto, bez ohledu na to, zda to bylo detekováno knihovnou ARCore jako rovina, či nikoliv. To umožňuje selektivní skrývání například obrazu na stěně, nebo malého koberce uprostřed místnosti.

Na druhou stranu jsou s tímto přístupem spojeny i nevýhody. První z nich je požadavek na zadávání bodů uživatelem, namísto pouhého výběru rovin, jako tomu bylo v prvním případě. Další nevýhoda spočívá ve vykreslování clony na úroveň rovin představující vir-



Obrázek 3.6: Snímek obrazovky aplikace zobrazující detekované roviny - oranžově ohraničené polygony. Detekované vizuálně význačné body jsou označeny červenými čtverečky.

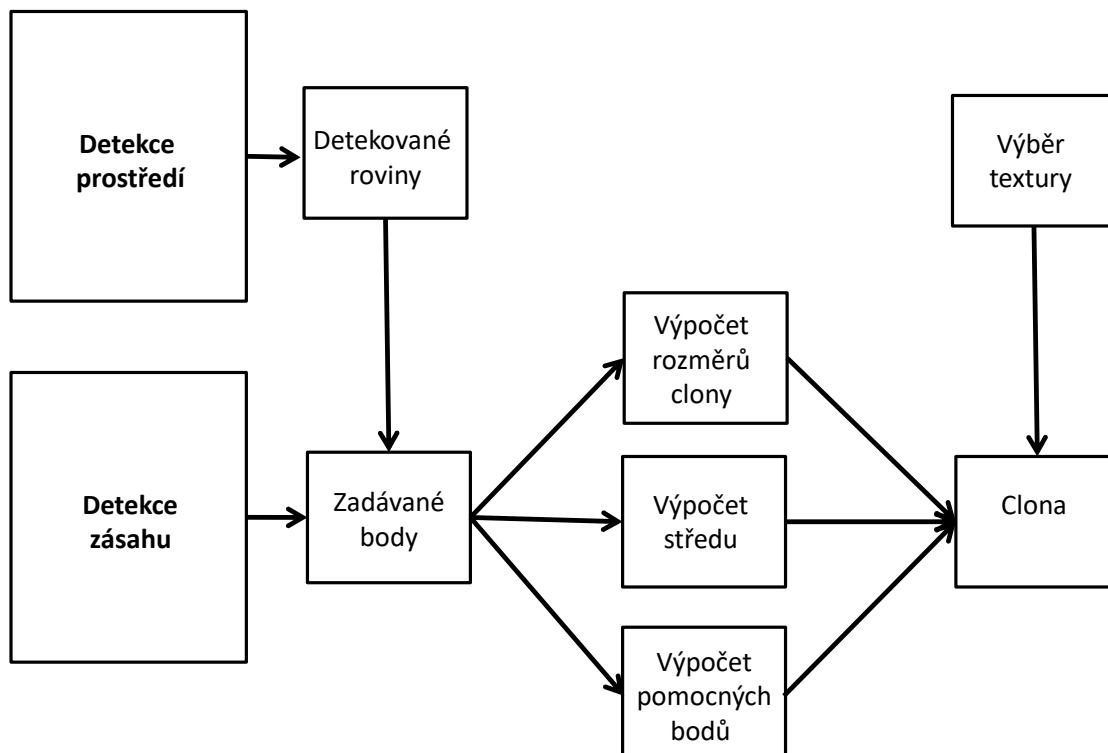
tuální podlahu. Z mnoha úhlů pohledu na danou scénu to nepředstavuje problém, ovšem pokud bude uživatel mobilní zařízení pomalu přibližovat k podlaze a stále sledovat místo, kde se nachází skrytý objekt, tak ten se bude postupně stávat více a více viditelným. Clona bude z těchto úhlů pohledu vykreslována stále na své pevně dané místo, avšak vzhledem ke změně pozice pozorování dojde k tomu, že z daného úhlu pohledu již clona nebude moci daný objekt ve scéně zakrýt. V určitých situacích může na druhou stranu zase docházet k opačnému případu - a to, že vytvořená clona bude ve scéně zakrývat částečně, nebo dokonce i zcela objekty, které uživatel maskovat nechce. Tyto, z uživatelského hlediska, nežádoucí projevy jsou důsledkem využití skutečnosti, že virtuální objekty ve scéně jsou zobrazovány nad obraz reálného prostředí, díky čemuž je prováděno maskování vybraných částí scény. Výsledek maskování vybraného objektu je tedy závislý na úhlu pohledu na maskovanou oblast.

Vyhodnocení obou způsobů

Ze dvou výše zmíněných způsobů přináší každý své výhody i nevýhody - rozebrané u každého z nich. Hlavním hodnotícím hlediskem toho, který způsob bude vhodnější využít, je potenciální množství situací, které je možné daným způsobem vyřešit. Způsob založený na využití 2D masky je do značné míry ovlivněn schopností ARCore detekovat okolní prostředí. Jak bylo zmíněno v části 3.2, tak problém představuje celá řada objektů, které se neskládají pouze z horizontálních a vertikálních ploch a v jejich případě bude problematické dosáhnout jejich úplného skrytí.

Právě možnost uplatnit způsob využívající ke skrytí skutečného objektu virtuální, uživatelem definovaný, objekt na více možných situacích, díky nižší závislosti na automatické detekci, vedla na volbu tohoto způsobu. Tím, že skrývaná oblast je na detekované roviny vázána nepřímo je také na druhou stranu možné skrýt jen část oblasti, která by jinak byla, v případě použití 2D masky vytvořené z detekované roviny, zakryta celá. Je tak možné například provést pouze zakrytí obrazu na stěně, která je detekována jako jedna velká rovina.

3.3 Proces skrývání objektů

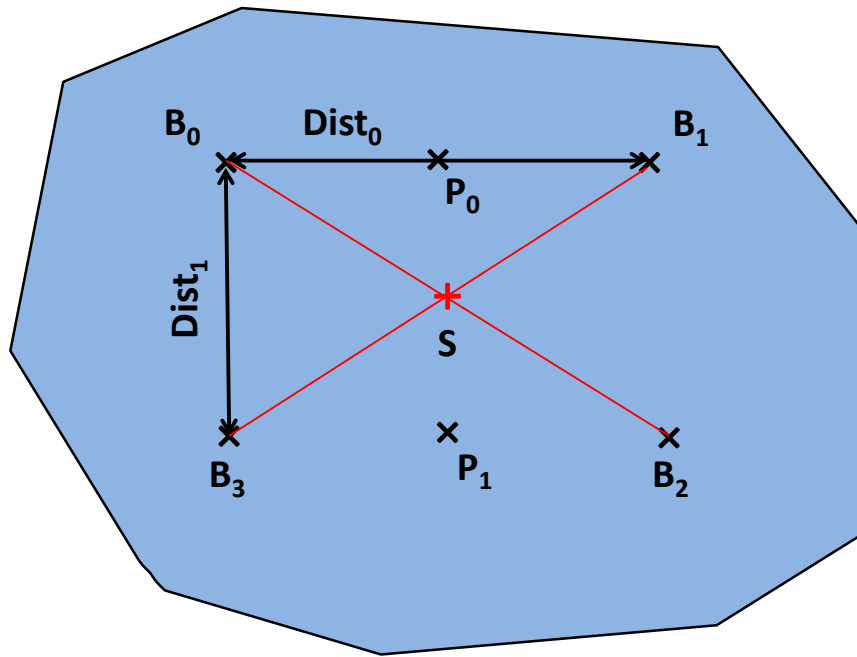


Obrázek 3.7: Proces skrývání objektů prostřednictvím virtuálního objektu. Části detekce prostředí a zásahu představují funkcionalitu realizovanou knihovnou ARCore.

Zjednodušená postup 3.7 znázorňuje základní stavební kameny procesu skrývání objektů založeného na využití virtuálního objektu - viz 3.2. Jak již bylo zmíněno v kapitole 3.2, tak je využívána detekce horizontálních a vertikálních ploch poskytovaná knihovnou ARCore – v grafu část nazvaná detekce prostředí. Získané roviny jsou vizualizovány uživateli, který na nich vytváří body, mezi kterými má dojít k vytvoření clony. Pozice bodů jsou získávány prostřednictvím detekce zásahu (*Hit testing*). Ze zadaných bodů jsou zjištěny rozměry potřebné clony. Výpočet středu představuje zjištění pozice, na kterou bude clona umístěna, zatímco její správné natočení je zajištěno za využití pomocných bodů.

Ke zjednodušení procesu je využíváno 4 uživatelem zadaných bodů. Vzhledem k tomu, že v typickém případě použití budou tyto čtyři body tvořit konvexní tvar přibližně v podobě čtverce, nebo obdélníku, nabízí se využití primitivního 3D objektu v Unity – roviny (*Plane*). Tuto rovinu lze na základě 4 získaných bodů vhodně umístit a upravit její velikost tak, aby pokrývala plochu mezi těmito body.

Na obrázku 3.8 se nalézají 4 body, přičemž z těchto bodů je potřeba zjistit 2 vyznačené vzdálenosti – jedná se mezi všemi 4 body o dvě nejmenší vzdálenosti, které určují délky stran roviny. Vzdálenost mezi dvěma body v 3D prostoru se vypočítá následovně:



Obrázek 3.8: Modrá ohraničená plocha představuje rovinu detekovanou ARCore. Uživatelem vytvořené body nesou označení $B_0 - B_3$. $Dist_0$ a $Dist_1$ jsou hledané vzdálenosti, střed zadaných bodů je označen písmenem S . Body označené jako P_0 a P_1 představují 2 pomocné body.

Souřadnice bodů jsou zapisovány ve formátu (x, y, z) :

$$B_n(x_n, y_n, z_n) \text{ kde } n = 4$$

Mějme funkci $d(B_0, B_1)$, která je definována jako Euklidovská vzdálenost dvou bodů v 3D prostoru:

$$Dist_0 = d(B_0, B_1) = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}$$

V návaznosti na zjištěné vzdálenosti je nutné vypočítat střed mezi 4 zadanými body, který představuje pozici, na níž bude objekt 3D roviny (clony) umístěn. Na obrázku je střed označen písmenem S , jeho pozice je vypočítána jako aritmetický průměr souřadnic bodů B_0 až B_3 :

Souřadnice středu jsou značeny následovně:

$$S(x_s, y_s, z_s)$$

Výpočet středu:

$$S = \left(\frac{x_0 + x_1 + x_2 + x_3}{4}, \frac{y_0 + y_1 + y_2 + y_3}{4}, \frac{z_0 + z_1 + z_2 + z_3}{4} \right)$$

Znalost rozměrů clony společně se středem již umožňuje její vytvoření a vykreslení na danou pozici. Nyní je nutné ji správně natočit, aby se její 2 delší strany nacházely mezi vzdálenější dvojicí bodů – na obrázku dvojice B_0, B_1 a B_2, B_3 – a její dvě kratší strany mezi dvojicemi B_0, B_3 a B_1, B_2 .

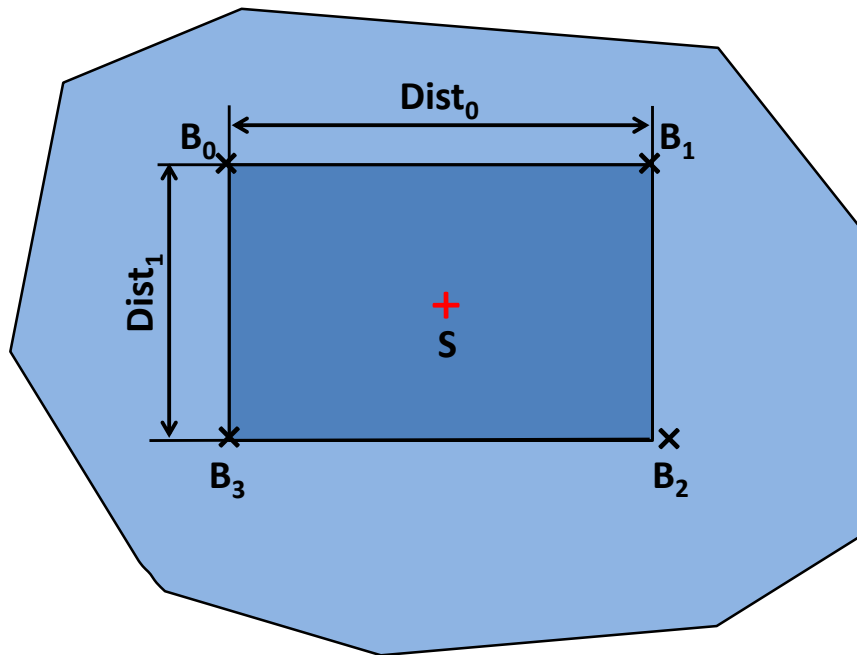
Za tímto účelem je nutné pro clonu umístěnou na vypočítaném středu určit potřebnou rotaci. Toho lze dosáhnout dvěma pomocnými body na obrázku označenými P_0 a P_1 , jež představují střed mezi body B na jejichž spojnici leží. Vektor spojující tyto dva pomocné body poté udává směr, jakým má být rovina natočena kolem osy y :

Souřadnice pomocných bodů P_0 a P_1 jsou:

$$P_0(x_{p_0}, y_{p_0}, z_{p_0}) = \left(\frac{x_0 + x_1}{2}, \frac{y_0 + y_1}{2}, \frac{z_0 + z_1}{2} \right)$$

$$P_1(x_{p_1}, y_{p_1}, z_{p_1}) = \left(\frac{x_3 + x_2}{2}, \frac{y_3 + y_2}{2}, \frac{z_3 + z_2}{2} \right)$$

Na základě provedení výše rozebraných operací je vytvořena clona s potřebnými parametry - znázorněno na obrázku 3.9.



Obrázek 3.9: Světle Modrá ohraničená plocha představuje rovinu detekovanou ARCore. Tmavě modrá plocha tvoří clonu vytvořenou na základě bodů $B_0 - B_3$, střed S je bod, na kterém je clona umístěna a velikost jednotlivých stran clony je pak určena na základě vzdáleností $Dist_0$ a $Dist_1$

3.4 Uživatelské rozhraní

Při návrhu uživatelského rozhraní jsem vycházel ze dvou základních funkcionalit, které má navržená aplikace poskytovat – maskování skutečných objektů a vkládání virtuálních. V obou případech je nutné uživateli poskytnout dostatečnou míru kontroly a přehlednosti.

Pro maskování skutečných objektů je nutné uživateli zobrazit stav porozumění okolnímu prostředí - vizualizovat roviny nalezené prostřednictvím ARCore. Jakmile bude nalezena rovina, na kterou chce uživatel umístit clonu, tak uživatel vytvoří body, mezi kterými bude clona vygenerována. Aby se uživatel neztratil při vytváření jednotlivých bodů v prostoru je nutné na místo vytvořeného bodu umístit vhodnou značku.

Vkládání objektů rovněž vyžaduje zobrazování detekovaných rovin, kdy po aktivaci této funkcionality bude umožněno do scény přidávat virtuální objekty. Avšak vzhledem k tomu, že uživatel nemusí zvolený objekt vložit, ať již úmyslně, nebo neúmyslně, na cílové místo, tak bude vhodné poskytnout možnost manipulovat s již vytvořenými objekty, kdy pro tento účel bude nutné umožnit výběr konkrétního objektu, se kterým bude následně manipulováno.

Uživatelské rozhraní bylo vyvíjeno iterativně společně s rozvíjející se funkcionalitou, zejména na základě mých osobních zkušeností s ovládáním vytvářené aplikace.

Vzhledem k tomu, že se jedná o aplikaci s rozšířenou realitou řešící vizuální otázku, tak většina testování v průběhu vývoje je realizována formou spouštění aplikace na mobilním zařízení a analýzou vizuálního výstupu aplikace na displeji. Právě časté spouštění aplikace vedlo na požadavek, aby bylo možné vytvářet stále stejnou scénu co nejrychleji a umožnit tak rychlé testování provedených změn.

Z hlediska realizované funkcionality je možné celý proces běhu aplikace rozdělit do několika fází:

- Detekce prostředí
- Vytváření clony
- Vkládání objektů
- Pozorování scény

Na základě těchto fází je možné stanovit jednotlivé stavy uživatelského rozhraní - obrázek 3.10.

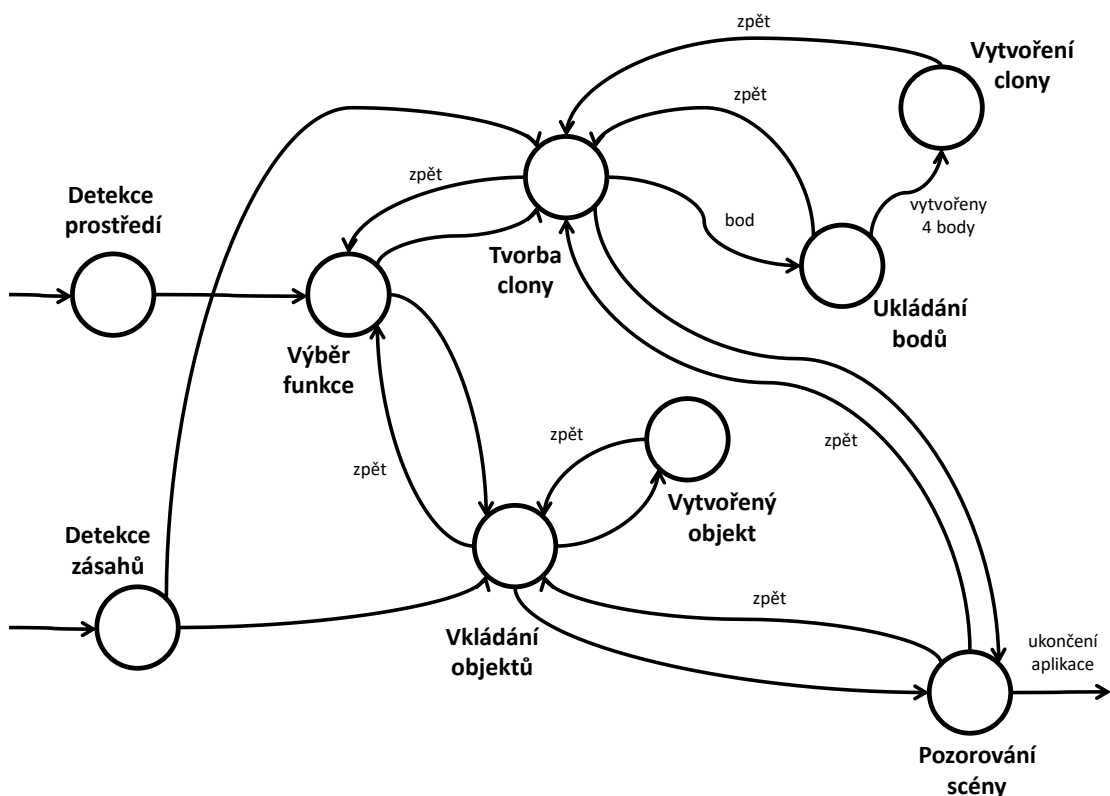
Fáze detekování prostředí

Detekce prostředí je část, která je kompletně v režii knihovny ARCore. Ze strany aplikace je řešena pouze vizualizace nalezených rovin a feature pointů. Při vizualizaci těchto prvků je nutné dodržet jednoduché základní požadavky. Roviny je třeba vizualizovat takovým způsobem, aby byly uživatelem aplikace na displeji snadno rozpoznatelné, ale aby přes ně bylo stále možné rozpoznávat objekty nacházející se v reálném světě a nedocházelo tak ke ztrátě orientace v prostoru při sledování obrazovky mobilního zařízení.

Na základě těchto požadavků a zkoušení různých variant byla nakonec zvolena vizualizace roviny v podobě polygonu bílé barvy s nastavenou úrovní průsvitnosti, jež zajišťuje viditelnost objektů nacházejících se pod ní.

Fáze vytváření clony

Vytváření clony představuje, společně s vkládáním virtuálních objektů, dvě hlavní funkce aplikace. Z tohoto důvodu je nutné zajistit, aby se tyto funkce nemíchaly do sebe a v jeden



Obrázek 3.10: Konečný automat znázorňující jednotlivé stavy uživatelského rozhraní při běhu aplikace.

čas byla aktivní pouze jedna z nich. Toho lze docílit ovládacími tlačítky pro přepínání hlavní funkcionality, kdy je možné mít aktivovanu pouze jednu funkci.

Po aktivaci funkce vytváření clony budou uživatelem generované zásahy na detekovaných rovinách zaznamenávány a na místo dotyku bude umístěna značka, představující vytvořený bod, pro snadnější orientaci. Jakmile jsou zadány 4 požadované body, je mezi nimi vygenerována clona. V průběhu testování aplikace bylo vyhodnoceno, že po vytvoření clony bude nejvhodnější umožnit ihned navázat vytvářením další clony, což dovoluje skrýt více vybraných oblastí - například dvě stěny a podlahu. Pokud uživatel nebude chtít vytvářet další clonu, musí funkci deaktivovat.

Fáze vkládání objektů

Vkládání objektů je funkce typicky využívaná po vytvoření clony, není to však pravidlem. Funkci je možné aktivovat nezávisle na tom, zda již uživatel vytvořil clonu, či nikoliv.

Pro vkládání objektů je rovněž, jako v případě vytváření clony, využívána detekce uživatelem generovaných zásahů, kdy na místě zásahu je vytvořen zvolený virtuální objekt. Na rozdíl od funkce vytváření clony je nutné si zvolit objekt z nabídky, který má být vložen, jelikož zde není nastavena výchozí volba z důvodu zvýšení přehlednosti aplikace, aby uživatel, který nechtěně aktivoval bez povšimnutí funkci vkládání objektů, nebyl překvapen tím, že se ve scéně najednou objevují nějaké objekty.

V průběhu testování aplikace právě pro tuto fázi vzešel požadavek na co nejrychlejší sestavení scény. Právě z důvodu urychlení bylo zvoleno řešení ve formě vertikálního výběrového menu sloužícího pro výběr objektu ke vložení, přičemž volba objektu zůstane uložena, dokud ji uživatel nezmění. Tímto je umožněno po návratu z výběrového menu na hlavní obrazovku na místo každého dotyku vložit virtuální objekt a rychle tak například umístit 4 modely vybrané židle a následně vybrat model stolu a ten také vložit. Tímto je dosaženo snížení množství kliknutí, které pro tento úkon musí uživatel vykonat oproti řešení užitého v jiných aplikacích vkládajících virtuální objekty, kdy zvolení objektu v menu umožňuje vložit pouze jednu instanci modelu a pro každou další je nutné znovu navštívit menu a vybrat stejný objekt.

Fáze pozorování scény

Pozorováním scény se rozumí fáze, kdy již uživatel vytvořil clonu a vložil do scény zamýšlené virtuální objekty, neplánuje ve scéně další změny a chce si vytvořenou scénu jen prohlížet. Pro zajištění co nejlepších pozorovacích podmínek je vhodné co nejvíce zvětšit plochu, na které je scéna zobrazována.

Zvětšení zobrazovací části displeje je dosaženo skrytím menu tlačítek, které již v této fázi nemají využití a pouze zabírají nemalou část obrazovky. Pokud by se uživatel chtěl vrátit do fáze vkládání objektů, nebo vytváření clony, je to možné prostřednictvím tlačítka, které vyvolá zpět nabídku se všemi funkcemi.

Základní rozvržení obrazovky aplikace

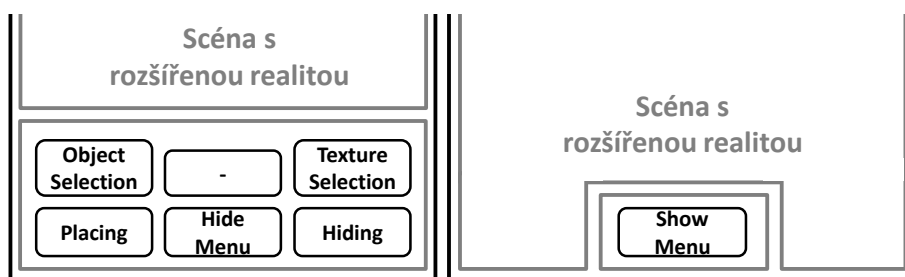
Při návrhu rozvržení aplikace jsem se snažil dodržet zvyklosti v rozložení ovládacích prvků užívaných na platformě Android a také poznatků zjištěných při analýze existujících aplikací - viz 2.5. Jelikož se jedná o aplikaci s rozšířenou realitou, je nutné nechat co možná nejvíce prostoru pro vykreslování scény. Rovněž je vhodné ovládací prvky umístit tak, aby byly snadno dosažitelné. Všem těmto požadavkům vyhovuje často používané umístění ovládacích prvků aplikace do spodní části obrazovky, tak jako to znázorňuje obrázek 3.11.

Část obsahující ovládací prvky (obrázek 3.12 vlevo) poskytuje přístup ke všem funkcím poskytovaným aplikací, přičemž jak pro vkládání objektů, tak pro vytváření clony jsou k dispozici 2 tlačítka. Jedno z tlačítek zapíná samotnou funkci a druhé tlačítko otevírá nabídku s objekty respektive texturami clony. Každá z těchto skupin tlačítek je umístěna u jedné strany displeje a v prostoru mezi nimi se nachází tlačítko určené ke skrývání menu. Umístění uprostřed bylo zvoleno, jelikož je jeho funkce blízká funkci prostředního tlačítka na mobilních zařízeních se systémem Android. Po jeho stisknutí dojde ke skrytí všech ostatních tlačítek, další stisk tohoto tlačítka vyvolá nabídku zpátky. Tímto je zajištěno zvětšení plochy displeje, za účelem lepšího pozorování scény - viz kapitola 3.4. Situaci znázorňuje obrázek 3.12.

Tlačítka nazvaná *Object Selection* a *Wall Material* vedou do vertikální rolovací nabídky, ve které je možné vybírat objekt pro vložení v prvním případě a texturu clony ve druhém případě. Po uskutečnění výběru je možné se vrátit na hlavní obrazovku prostřednictvím tlačítka *Return*. Nabídka je znázorněna obrázkem 3.13.



Obrázek 3.11: Hlavní obrazovka aplikace, kde největší část zabírá scéna s rozšířenou realitou.

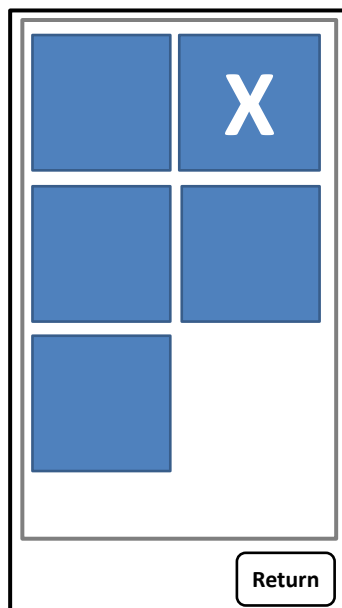


Obrázek 3.12: Vlevo plně zobrazený ovládací panel a vpravo skrytý.

3.5 Manipulace s objekty

Na základě povahy funkcionality a účelu vyvíjené aplikace vyvstal požadavek - viz 3.1, aby bylo možné s vloženými virtuálními objekty manipulovat. Mezi základní operace z hlediska úprav, jaké by mohl případný uživatel aplikace chtít realizovat, lze zařadit změnu pozice, otáčení a změnu velikosti virtuálního objektu. S cílem umožnit provádět tyto transformace došlo v průběhu vývoje na vyzkoušení více způsobů jak tohoto dosáhnout.

Zpočátku bylo experimentováno s vlastní implementací, která umožňovala posouvat virtuální objekt a měnit jeho velikost. Tato funkcionality byla, z důvodu jednodušší implementace, dostupná prostřednictvím tlačítek zobrazovaných na hlavní obrazovce po výběru některého virtuálního objektu. Tento přístup však přináší jisté nevýhody, zejména z hlediska složitosti ovládání na dotykové obrazovce. V případě změny pozice objektu při zachování jeho výškové úrovně, představované pozicí na ose y , může být pozice objektu změněna ve 2 osách (x a z) a na každé z os ve dvou směrech. Pro posun objektu je v takovém případě zapotřebí již 4 tlačítek, pokud však bude pohyb realizován i ve směru osy y , tak dokonce 6. Takové množství ovládacích prvků již zabírá značné množství plochy obrazovky, přičemž přidávání dalších transformací, pak situaci již jen dále komplikuje.



Obrázek 3.13: Výběrové menu - jednotlivé položky jsou představovány modrými čtverečky, velké písmeno X značí aktuálně vybranou položku.

Poněvadž je aplikace implementována pro mobilní zařízení, ovládané prostřednictvím dotykové obrazovky, tak se přímo nabízí možnost využití dotykových gest, které jsou mobilními aplikacemi široce používány. Pro tento přístup byly uvažovány dvě možnosti realizace. První možností bylo využití balíčku Lean Touch³ umožňujícího přidávat ovládání gesty bez nutnosti provádět vlastní implementaci rozpoznávání gest. Druhou možností bylo využít další ze vzorových aplikací poskytovaných v balíčku ARCore a to aplikace **Object Manipulation**⁴, která slouží pro demonstraci transformací virtuálních objektů a jako taková implementuje všechny poptávané možnosti.

Po zvážení všech aspektů byla zvolena druhá možnost, jelikož vyvíjená aplikace je založena na vzorové aplikaci **HelloAR**⁵, která má stejný programátorský styl, jako aplikace **Object Manipulation**, což usnadňuje orientaci v kódu a následné porozumění mu.

³<https://assetstore.unity.com/packages/tools/input-management/lean-touch-30111>

⁴<https://developers.google.com/ar/develop/unity/tutorials/object-manipulation-sample>

⁵<https://developers.google.com/ar/develop/unity/tutorials/hello-ar-sample>

Kapitola 4

Implementace aplikace pro realističtější zobrazení virtuálních objektů

Aplikace byla vyvíjena již v průběhu fáze navrhování, přičemž v této fázi pochopitelně došlo k mnoha změnám včetně změny hlavní myšlenky realizace skrývání skutečných objektů. Vývoj byl prováděn v podobě implementace různých malých aplikací řešících vždy jednu z budoucích funkcí aplikace tak, aby mohla být každá funkce odzkoušena nezávisle na další funkcionalitě, která by ji mohla ovlivnit.

Při implementaci aplikace byly využívány možnosti, které nabízí knihovna ARCore s cílem co nejjednoduššího vývoje. Využité technologie budou v této kapitole rozebrány podrobněji zejména z hlediska jejich konkrétního použití.

Kapitola se dále zabývá navrženými funkcemi, teoreticky rozebranými v kapitole 3.3, z hlediska jejich implementace, stejně jako volbě vývojového prostředí a programovacího jazyka.

4.1 Volba programovacího jazyka, prostředí a platformy pro vývoj

V průběhu implementace jednotlivých mini aplikací a zkoušení si práce s knihovnou ARCore a aplikacemi pro Android obecně, jsem vyzkoušel více vývojových prostředí i přístupů k vytváření aplikace s rozšířenou realitou. První pokusy byly prováděny v jazyce Java¹ a následně Kotlin² ve vývojovém prostředí Android Studio³, následně došlo k přesunu do prostředí Unity⁴. Hlavním důvodem pro tento krok byla má nezkušenost s programovacím jazykem Java, která v kombinaci s prakticky nulovými znalostmi vývoje pro operační systém Android značně komplikovala práci. Vývojové prostředí Unity využívá programovací jazyk C#⁵, který je velmi blízký jazykům C a C++.

¹<https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html>

²<https://kotlinlang.org/>

³<https://developer.android.com/studio>

⁴<https://unity.com/>

⁵<https://docs.microsoft.com/cs-cz/dotnet/csharp/>

V Unity byl vývoj zahájen ve frameworku AR Foundation⁶, který nabízí možnost programovat aplikaci s rozšířenou realitou bez ohledu na platformu, na kterou bude nasazena – Android nebo iOS⁷ (pozn. dalšími podporovanými platformami jsou Magic Leap⁸ a HoloLens⁹). To přináší výhodu programátorovi, který naprogramuje jednu aplikaci a následně ji jen sestaví pro jednu, nebo obě zmíněné platformy, přičemž je na obou platformách zajištěno stejné chování. Nevýhodou tohoto řešení je, že framework AR Foundation je vyvíjen Unity a vytváří zastřešující vrstvu, jejímž prostřednictvím je přístupováno k funkcionalitě platform ARCore a ARKit¹⁰.

Toto řešení přináší zmíněnou výhodu programování jedné aplikace pro dvě různé platformy, na druhou stranu je tímto způsobem programátorovi dostupná pouze ta funkcionalita, která je k dispozici v obou knihovnách. Funkcionalita, která je na jedné nebo druhé platformě jedinečná, případně řešená příliš odlišným způsobem pak není přístupná, nebo se do AR Foundation dostává jen s určitým zpožděním, což přináší komplikaci v tom, že aplikace stavěné na AR Foundation musejí někdy na novou funkcionalitu čekat, zatímco při vývoji v nativním ARCore, nebo ARKit je dostupná veškerá aktuálně vydaná funkcionalita.

Jelikož je aplikace vyvíjená pro operační systém Android, tak byl po zvážení všech kladů a záporů možných přístupů zvolen vývoj v nativním ARCore balíčku vyvíjeném přímo společností Google pro vývojové prostředí Unity. Jak již bylo zmíněno, tak vývoj v nativním balíčku přináší přímý přístup k nejnovější funkcionalitě, jakou je například sdílená rozšířená realita (*Cloud Anchors*) - představená v kapitole 2.4.

4.2 Struktura aplikace

Aplikace vyvíjená v prostředí Unity se skládá z prvků nazývaných `GameObject` (*herní objekt*), které jsou instancemi dvou druhů – `Asset file` (například 3D model) a `Prefab` (šablona vlastního herního objektu). Každý herní objekt se může skládat z velkého množství součástí nazývaných `Component` (*komponent*), jakými jsou například skripty, grafické efekty a elementy, prvky uživatelského rozhraní a mnoho dalších. Uspořádání herních objektů a jejich vzájemná závislost pak tvoří hierarchii scény aplikace.

Na obrázku 4.1 je hierarchie scény implementované aplikace nazvané `ObjectHider`. Aplikace je založena na ukázkové aplikaci HelloAR¹¹, která je součástí balíčku ARCore, tudíž zatímco některé objekty byly implementovány od začátku pro vytvořenou aplikaci, tak některé pocházejí z ukázkové aplikace, přičemž byly případně modifikovány pro implementovanou aplikaci. Jednotlivé herní objekty budou blíže představeny níže.

Controller

Controller představuje základní herní objekt z hlediska implementované funkcionality, jelikož je složen ze 3 důležitých komponentů, kterými jsou skripty `CanvasSwitcher`, `ObjectHider`, `SelectObjectToPlace`.

`CanvasSwitcher` je zodpovědný za přepínání mezi jednotlivými obrazovkami aplikace, jimiž jsou:

⁶<https://unity.com/unity/features/arfoundation>

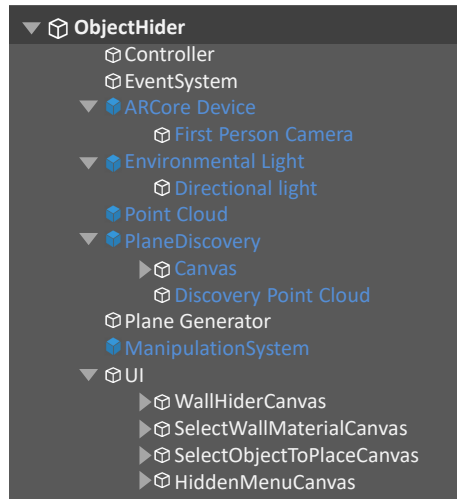
⁷<https://www.apple.com/cz/ios/ios-13/>

⁸<https://www.magicleap.com/en-us>

⁹<https://www.microsoft.com/en-us/hololens>

¹⁰<https://developer.apple.com/documentation/arkit>

¹¹<https://developers.google.com/ar/develop/unity/tutorials/hello-ar-sample>



Obrázek 4.1: Hierarchická struktura aplikace.

- hlavní obrazovka se scénou rozšířené reality
- menu pro výběr objektu
- menu pro výběr textury clony
- hlavní obrazovkou se skrytým menu

Výše zmíněné obrazovky aplikace jsou přímou implementací návrhu rozebraného a graficky znázorněného v kapitole 3.4, který byl v průběhu implementace v určitých ohledech upraven na základě nově získaných poznatků.

ObjectHider je skript zodpovědný za vytvoření clony s vybranou texturou. Funkce `ProcessTouches` zajišťuje pro detekované kliknutí na obrazovce vyslání paprsku směrem doteku, kdy je zjišťováno, zda bylo kliknuto na detekovanou rovinu - koncept blíže rozebrán v kapitole 2.4. Pokud byla zasažena detekovaná rovina, dojde na místě zásahu k vytvoření značky, která uživateli oznamuje zaznamenání bodu, pro značku je vytvořen na stejném místě kotvící bod a ze značky je učiněn jeho potomek. Touto dědičností je zajištěno, že AR-Core bude sledovat vytvořenou značku, která je následně uložena do pole značek pro pozdější použití. Získávání zadaných bodů realizuje následující kus zdrojového kódu:

```

if ((hit.Trackable is DetectedPlane) &&
    Vector3.Dot(firstPersonCamera.transform.position - hit.Pose.position,
                hit.Pose.rotation * Vector3.up) < 0)
{
    Debug.Log("Hit at back of the current DetectedPlane");
}
else
{
    GameObject prefab;
    prefab = pointPrefab;

    var pointObject = Instantiate(prefab, hit.Pose.position,

```

```

        hit.Pose.rotation);

    var anchor = hit.Trackable.CreateAnchor(hit.Pose);
    pointObject.transform.parent = anchor.transform;

    pointArray.Add(pointObject);
    anchorCount++;

}

```

Po vytvoření 4 bodů je zavolána funkce `CreateVirtualWall`, ve které je spočítána velikost vytvářené clony, pozice jejího umístění a po jejím vytvoření je zajištěno správné natočení. Také dochází k vytvoření instance manipulátoru (třída `Manipulator`) pro tuto clonu (clona je potomkem manipulátoru), jehož prostřednictvím jsou realizovány manipulace s objekty.

`SelectObjectToPlace` realizuje vkládání virtuálních objektů do scény. Podobně jako `ObjectHider` implementuje funkci `ProcessTouches`, která se zde v implementaci drobně liší. V tomto případě nedochází k ukládání bodů, ale na každém místě kliknutí je vytvořena instance zvoleného modelu. Vytvořený model je stejně, jako v případě clony vytvořené skriptem `ObjectHider`, potomkem třídy `Manipulator`.

Jedním ze zajímavých problémů souvisejícího s komponenty `Controlleru`, který se v průběhu vývoje vyskytl, byla skutečnost, že jednotlivé elementy uživatelského rozhraní, jako jsou například tlačítka, jsou propustná pro kliknutí. Kliknutí na tlačítko v aplikaci nejenom aktivovalo funkci daného tlačítka, ale také se přeneslo i do samotné scény s rozšířenou realitou. Tento jev v situaci, kdy se tlačítko nacházelo nad detekovanou rovinou a byla například aktivována funkce pro vkládání objektů, způsobil vložení nového objektu na pozici na rovině za daným tlačítkem. Tento efekt byl z uživatelského hlediska shledán nežádoucím a byl eliminován prostřednictvím následujícího kódu použitého ve skriptech `ObjectHider` a `SelectObjectToPlace`:

```

//if a button is hit no object will be placed on place behind button
//on AR plane
if (EventSystem.current.IsPointerOverGameObject() ||
    EventSystem.current.currentSelectedGameObject != null)
{
    return;
}

```

Výše uvedený kód umístěný ve funkci, která realizuje, na základě detekovaného kliknutí na obrazovce, nějakou funkci - například vkládání objektů - způsobí, že pokud je tímto kliknutím zasažen nějaký objekt typu `EventSystem`, případně je v aktivním stavu některý z objektů toho typu, bude proveden návrat z této funkce. Tímto je docíleno toho, že jsou kliknutí na tlačítka ignorována skripty pracujícími se scénou vy smyslu přidávání jakýchkoliv objektů.

EventSystem a UI

EventSystem tvoří důležitou součást pro fungování objektu **UI** (realizujícího uživatelské rozhraní aplikace), jelikož zajišťuje zpracovávání uživatelských vstupů v podobě interakce s tlačítky, ale hlavně také zajišťuje předávání událostí mezi objekty ve scéně.

UI se skládá ze čtyř hlavních obrazovek aplikace, které představují uživatelské rozhraní aplikace. Hlavní obrazovkou je **WallHiderCanvas** zobrazující scénu s rozšířenou realitou a umožňující vkládání vybraného objektu, nebo vytváření clony.

HiddenMenuCanvas představuje upravenou verzi hlavní obrazovky, na níž byly skryta všechna tlačítka, mimo tlačítka pro přepnutí do výchozího zobrazení.

SelectWallMaterialCanvas obsahuje menu s výběrem textury clony, zatímco **SelectObjectToPlaceCanvas** umožňuje vybrat model objektu ke vložení.

ARCoreDevice a Environmental Light

ARCoreDevice je objekt převzatý ze vzorové aplikace, jehož komponentem je skript **ARCoreSession** řídící běh **ARCore** v prostředí Unity scény. Pro správné fungování vyžaduje validní konfiguraci sezení (**ARCoreSessionConfig**), a také aby kamera zařízení poskytovala konfigurace zapsané v **ARCoreCameraConfigFilter**.

Potomkem **ARCoreDevice** je objekt **First Person Camera** představující hlavní kameru 3D scény. K němu jsou připojené 2 komponenty, kterými jsou **TrackedPoseDriver** a **ARCoreBackgroundRenderer**. **TrackedPoseDriver** se stará o aktualizaci pozice a rotace herního objektu na základě aktuální pozice a rotace zařízení, zatímco **ARCoreBackgroundRenderer** zajišťuje vykreslování snímků reálného prostředí získávaných kamerou jako pozadí pro připojenou 3D kameru scény.

Výše zmíněné objekty, vzhledem k tomu, že se podílejí na vytváření celkové scény, souvisejí s objektem **Environmental Light**. Tento objekt totiž nese zodpovědnost za automatické úpravy osvětlení ve scéně na základě hodnot detekovaných prostřednictvím **ARCore** – viz kapitola 2.4.

Point Cloud a Plane Generator

Převzaté objekty sloužící pro vizualizaci informací o detekovaném prostředí. V případě objektu **Point Cloud** jsou vizualizované informace tvořeny mračnem bodů, které je představováno detekovanými vizuálně význačnými body (*feature points*). Objekt **Plane Generator** posléze slouží pro řízení zobrazování detekovaných rovin (*planes*) ve smyslu jejich grafické podoby.

Oba objekty samy o sobě řeší pouze vizuální stránku věci a v průběhu vývoje aplikace bylo s těmito objekty experimentováno za účelem vyzkoušení různých možností vizualizace těchto informací – jejich množství, možností třídění a pochopitelně také grafického vzhledu.

Manipulation system

Objekt s názvem **ManipulationSystem** představuje z aplikace **ObjectManipulation**¹² převzatou součást, jejímž prostřednictvím jsou transformace realizovány za pomoci gest prováděných na displeji. Ovládání gesty bylo dále pro implementovanou aplikaci doplněno i o další funkce realizované prostřednictvím tlačítek.

¹²<https://developers.google.com/ar/develop/unity/tutorials/object-manipulation-sample>

Nejdůležitější komponentou pro fungování celého systému je prefab **Manipulator** (manipulátor), který je využíván skripty **Controlleru**, kde je pro každý vytvořený objekt vytvořena instance manipulátoru, ze které je učiněn rodič daného objektu. Prostřednictvím manipulátoru dochází k výběru objektů a jejich transformacím.

Samotný **Manipulator** se skládá z pěti komponent:

- SelectionManipulator
- ScaleManipulator
- TranslationManipulator
- RotationManipulator
- ElevationManipulator

Jednotlivé transformace realizované těmito komponenty využívají skriptů rozpoznávajících zadávaná gesta (*Gesture recognizers*) k ovládání.

SelectionManipulator zajišťuje zobrazení značky značící vybraný objekt a v případě, že dochází k posunu objektu ve směru osy y , tak se stará o to, aby značka vybraného objektu zůstala na detekované rovině bez ohledu na výšku objektu, ke kterému patří. Výběr objektu je realizován kliknutím na něj (*Tap gesture*).

ScaleManipulator slouží ke změně měřítka (*Scale*) vybraného objektu – zvětšování rozevíráním dvou prstů a zmenšování jejich svíráním (*Pinch gesture*). Skript disponuje možností nastavit mezní hodnoty měřítka. Výsledná hodnota měřítka objektu je určována pomocí určení poměru aktuálního měřítka objektu, které se určí jako rozdíl aktuálního měřítka objektu a minimálního měřítka děleno maximální rozsah měřítka. Hodnota zjištěného poměru je následně měněna na základě rozsahu zadávaného gesta a aktualizovanou hodnotou měřítka je průběžně transformován virtuální objekt, čímž dochází k plynulé změně velikosti.

TranslationManipulator zajišťuje změnu pozice objektu na základě jeho tažení (*Drag gesture*). Objekt je průběžně umísťován na pozici prstu na obrazovce, přičemž jsou zachovávány jeho ostatní parametry, jako jsou výška nad rovinou, měřítko a rotace.

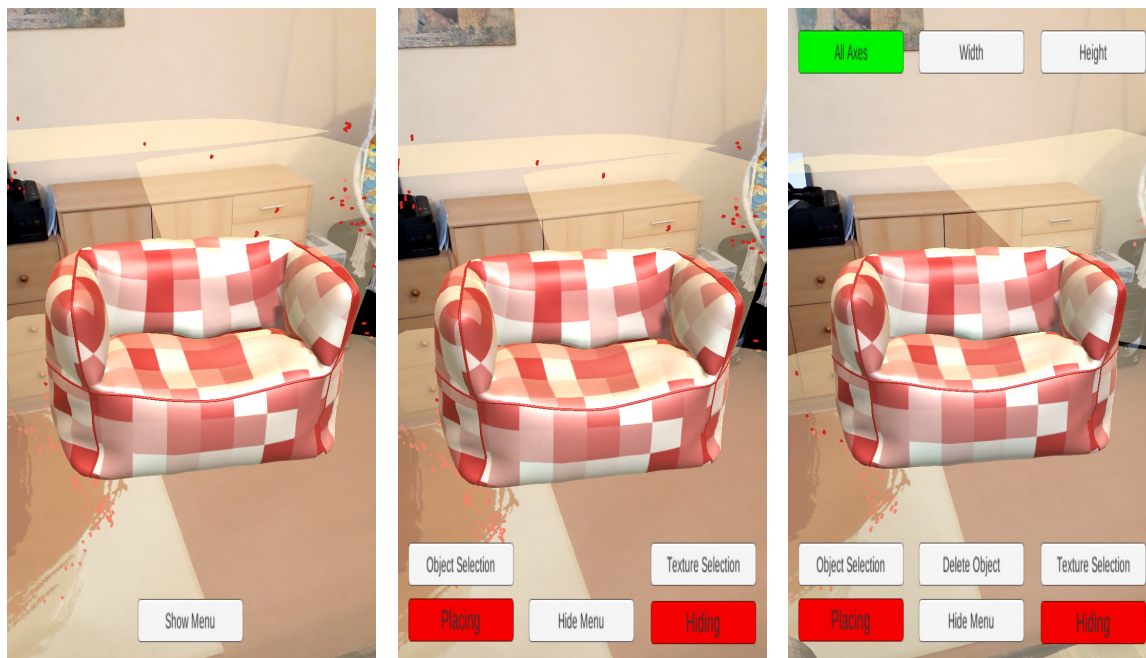
RotationManipulator provádí rotaci objektu okolo osy y , prostřednictvím gesta otáčení (*Twist gesture*). Změna rotace je určena na základě rozsahu gesta na displeji ve směru osy x vynásobeného rychlostí rotace. Průběžná aktualizace rotace objektu na ose y zajišťuje zobrazování plynulého otáčení.

ElevationManipulator aktualizuje výšku objektu nad horizontálními rovinami, prostřednictvím tažení dvěma prsty na výšku displeje. Hodnota pozice objektu na ose y je určována přes posun gestem na displeji ve směru osy y poděleného rozlišením displeje násobeného rychlostí posunu.

4.3 Výsledná aplikace

Výsledná aplikace nazvaná *ObjectHider* provádí vizualizaci detekovaných rovin a mračna bodů. Na detekované roviny umožňuje vkládat virtuální objekty z nabídky, se kterými je

následně možné provádět transformace v podobě změny jejich pozice, velikosti, rotace a elevace. Hlavní funkcionalita je představována možností skrývání reálných objektů, v tomto dokumentu označovaná jako vytváření clony. Clona je vytvořena na základě uživatelem zadané čtveřice bodů.

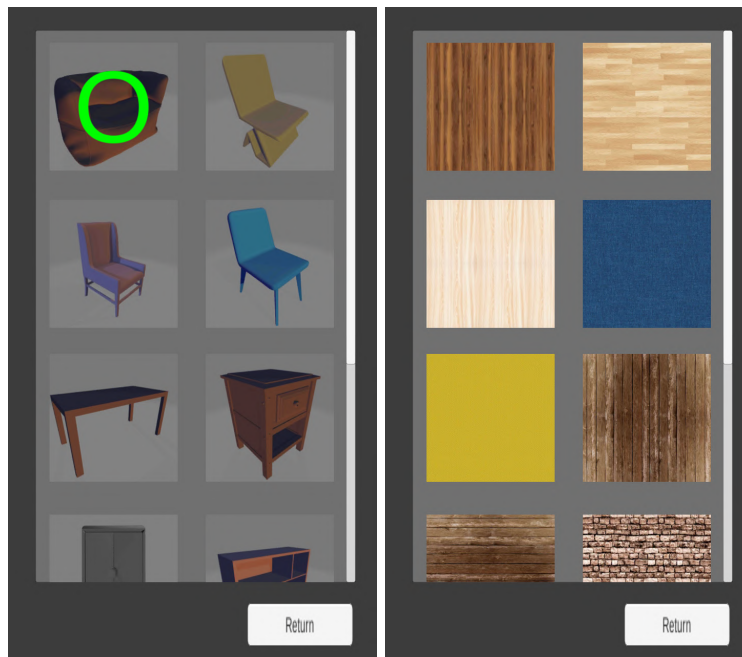


Obrázek 4.2: Červeno-bílé křeslo je virtuální objekt vložený prostřednictvím aplikace. Obrázek vlevo zobrazuje podobu pro mód prohlížení scény. Uprostřed se nachází výchozí podoba hlavní obrazovky. Vpravo je nabídka zobrazovaná v případě, že byl vybrán některý virtuální objekt ve scéně.

Obrázek 4.2 je tvořen snímky tří podob hlavní obrazovky aplikace. Jejich uspořádání vychází z návrhu sepsaného v kapitole 3.4, přičemž oproti návrhu došlo v průběhu vývoje k určitým úpravám. Jednou z viditelných změn je skryté tlačítko sloužící pro mazání vybraného objektu, jelikož není nutné toto tlačítko zobrazovat, pokud není vybrán žádný objekt.

Červeně obarvená tlačítka značí, že funkce ovládaná tímto tlačítkem je vypnuta, zelená značí aktivní funkci. Na obrázku vlevo je zobrazena podoba menu pro mód prohlížení scény, ve kterém jsou skryta všechna ovládací tlačítka, mimo tlačítko *Show Menu* pro zobrazení menu. Uprostřed se nachází výchozí podoba hlavní obrazovky - tlačítka *Placing* a *Hiding* slouží pro zapínání funkce vkládání objektů, respektive vytváření clony. *Object Selection* otevře nabídku s 3D modely, kdy vybraný model bude následně vkládán po aktivaci funkce vkládání objektů. *Texture selection* vyvolá nabídku s volbou textury pro vytvářenou clonu. Vpravo se nachází nabídka zobrazovaná v případě, že byl vybrán některý virtuální objekt ve scéně - v tomto případě červeno-bílé křeslo. Tato nabídka obsahuje oproti předchozí tlačítko *Delete Object* - které provede odstranění vybraného objektu. Tlačítka *All Axes*, *Width*, *Height* slouží pro přepínání režimu změny velikosti vybraného objektu - ta může být prováděna ve všech osách zároveň, pouze v ose x (*Width*) nebo pouze v ose z (*Height*). Tlačítka pro změnu šířky (*Width*) a výšky (*Height*) jsou primárně určena pro změnu velikosti clony, kdy je s jejich pomocí možné nastavit její přesnou velikost. Tlačítko *All Axes* je

typicky využíváno pro změnu velikosti modelů nábytku pro zachování poměru velikosti v jednotlivých osách.



Obrázek 4.3: Snímek vlevo ukazuje nabídku výběru objektu, který bude vkládán po aktivaci vkládání tlačítkem *Placing*. Zelený kroužek označuje aktuálně vybranou položku, tlačítko *Return* slouží k návratu na hlavní obrazovku. Vpravo se nachází velmi obdobná nabídka s tím rozdílem, že slouží pro výběr textury clony.

Snímek 4.3 ukazuje menu sloužící pro výběr objektu, který bude následně po kliknutí na nějaké místo na detekované rovině na toto místo umístěn prostřednictvím skriptu `SelectObjectToPlace`. Obdobnou podobu má i na stejném obrázku zobrazená nabídka sloužící pro výběr textury clony.

4.4 Testování

Testování aplikace bylo nedílnou součástí implementačního procesu, v jehož průběhu bylo nutné ověřovat funkčnost jednotlivých částí aplikace. Probíhalo v reálném prostředí, které je vzhledem k jejímu zaměření představováno vnitřními prostory budov. Pro účely testování byly používány scény s různou složitostí pro detekci rovin prostřednictvím platformy ARCore a také různou náročností pro implementovanou funkcionalitu.

Hlavním cílem testování bylo ladění implementovaných funkcí aplikace a uživatelského rozhraní a také ověřování toho, že implementované řešení má vlastnosti stanovené ve fázi návrhu. Za účelem ukázky možností implementovaného řešení byly zvoleny dvě testovací scény, jedna tvořena kuchyňskou stěnou a druhá jídelním stolem se židlemi. Na těchto dvou ukázkách budou prezentovány vlastnosti navrženého a následně implementovaného řešení. Budou rozebrány situace, ve kterých navržený způsob dokáže správně zakrýt cílovou oblast a skutečně zlepšit výsledný vjem ze scény. Na druhou stranu bude však věnována pozornost i situacím, pro které navržený způsob neposkytuje dobré výsledky maskování. Jak bylo

zmíněno v kapitole návrhu řešení - viz 3.2, tak výsledný vjem ze scény je závislý zejména na úhlu pohledu na maskovanou oblast - jedná se o vlastnost navrženého řešení.

1. testovací scéna

Na obrázku 4.4 se nachází testovací scéna skládající se z kuchyňské stěny, před níž je umístěna kuchyňská linka, nad kterou se nacházejí nástěnné skříňky. Z uživatelského pohledu je v této scéně cílem vyzkoušet, jak by na dané stěně vypadala nová nástěnná skříňka.



Obrázek 4.4: Snímek vlevo znázorňuje výchozí testovací scénu. Prostřední snímek vyznačuje z pohledu ARCore dvě vizuálně velmi výrazné roviny - přední strany skříňek, které jsou typicky velmi rychle rozpoznány. Vrchní nástěnné skříňky pak mají být nahrazeny virtuálními. Snímek vpravo vyznačuje část stěny místnosti, kterou je potřeba detekovat, aby na ní mohla být umístěna virtuální skříňka a ta tak vytvářela dojem, že se nachází na skutečné stěně.

Úkony vedoucími k realizaci tohoto úkolu jsou:

- Detekce skutečné stěny místnosti nacházející z pohledu uživatele za skříňkami
- Vytvoření clony tvořící virtuální stěnu místnosti
- Vložení virtuální skříňky na místo, na kterém se nacházejí skutečné skříňky

Tato testovací scéna obsahuje z pohledu ARCore mnoho vizuálně význačných bodů a rovin, které je možné detekovat. K nalezení první roviny proto dochází do několika vteřin od načtení aplikace. Rychlé rozpoznání rovin je z uživatelského pohledu kladnou záležitostí, ovšem na druhou stranu se ve scéně nachází značné množství různorodých objektů, tedy potenciačních rovin pro detekci - které v případě, že budou detekovány dokáží práci s aplikací jistým způsobem ztížit, jelikož je jednodušší pracovat s menším počtem větších rovin, než s větším počtem menších.

Obrázek 4.5 ukazuje roviny detekované prostřednictvím ARCore a zobrazené aplikací ObjectHider. Obrázek 4.6 znázorňuje postup, jakým dojde na detekované rovině k vytvoření clony. Vytvořená clona pokrývá část zdi nad dolními skříňkami a svojí plochou zcela zakrývá nástěnné skříňky. Žlutá barva byla zvolena na základě podobnosti se skutečnou stěnou.



Obrázek 4.5: Dva snímky z aplikace zobrazující tři detekované roviny - zadní stěnu, přední stranu nástěnných skříněk a kuchyňskou desku - ze dvou úhlů pohledu.

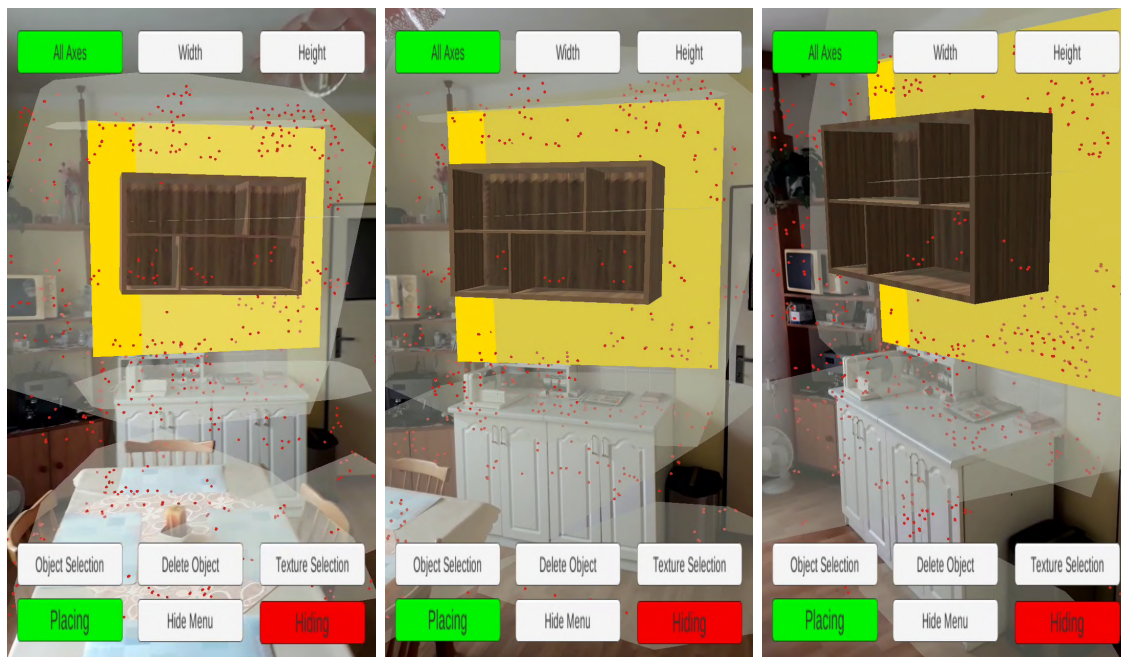


Obrázek 4.6: Na snímku vlevo byl vytvořen první bod sloužící pro vytvoření clony - označen zelenou šipkou. Na prostředním snímku byly vytvořeny další dva body. Obrázek vpravo pak znázorňuje situaci, kdy byly zadány všechny 4 potřebné body a byla vytvořena clona.

Na vytvořenou clonu byla následně umístěna virtuální skříňka a její velikost byla nastavena tak, aby svými rozměry odpovídala skutečným skříňkám nacházejícím se za clonou.

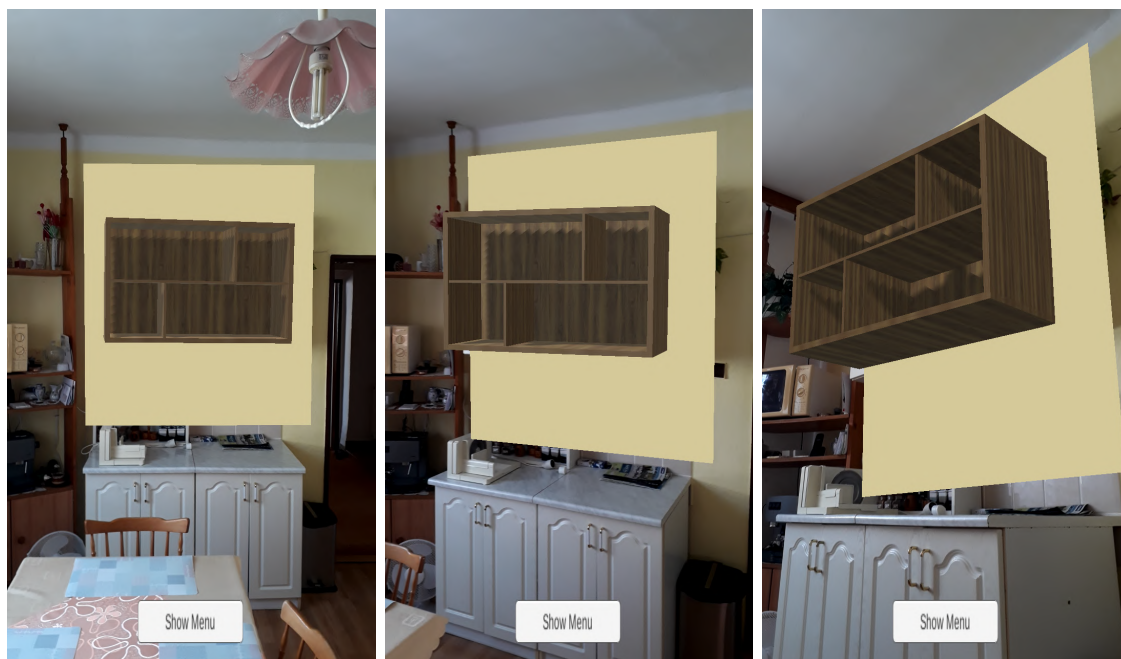
Obrázek 4.7 ukazuje celou scénu za použití výchozího pozorovacího módu aplikace - hlavní obrazovky, kdy jsou zobrazovány detekované roviny, body a také všechny ovládací prvky.

Obrázek 4.8 ukazuje podobné pohledy, avšak za použití prohlížečského módu, kdy po stisku tlačítka *Hide Menu* dojde ke skrytí ovládacích prvků a je deaktivováno zobrazování detekovaných rovin a bodů, což vytváří čistější scénu a umožňuje nerušené pozorování. Doplňující pohledy z levé strany (obrázek 4.9) znázorňují, podobně jako obrázek 4.8, úhly pohledu, ze kterých se scéna jeví přirozeně - dochází k zamaskování cílové oblasti. Ovšem na obrázku vpravo je možné pozorovat kousek clony, kterým již je, vzhledem k úhlu pohledu, překrývána část dveří.



Obrázek 4.7: Na všech snímcích je možné pozorovat rozdíl v zobrazení oproti obrázku 4.8, kde je skryta hlavní nabídka a je zvětšena plocha zobrazující scénu. Skrytí menu nabídky společně s deaktivací vykreslování detekovaných rovin přináší přehlednější scénu pro účely jejího prohlížení. Snímky znázorňují pohled na celou scénu z více úhlů.

Výše rozebraná testovací scéna ukazuje zejména možnosti navrženého řešení, kdy nejsou přímo patrné jeho nedostatky a vytvořená clona dokáže zamaskovat vybraný objekt tak, aby jeho přítomnost ve scéně nepůsobila rušivě po vložení virtuálního objektu na stejné místo. Avšak i na základě zde přítomných obrázků je možné nastínit některé z negativních vlastností, které jsou s tímto řešením spojené. Jedná se zejména o tu vlastnost, že v závislosti na úhlu pohledu na oblast umístění clony a virtuálního objektu může docházet k tomu, že clona bude zakrývat i ty části scény, které by uživatel chtěl nechat viditelné. Toto je možné si představit na obrázku 4.8 uprostřed, kdy pokud by byla clona zvětšena tak, aby její spodní hrana končila na přechodu kuchyňské linky a kachliček na stěně, došlo by na následujícím obrázku (4.8 vpravo) k situaci, že by clona zakrývala i část spodních skříněk, což je z uživatelského pohledu nežádoucí. Situacím, pro které navržené řešení nemůže, z hlediska svého principu fungování, poskytnout žádané zamaskování objektů se bude dále zabývat následující testovací scéna.



Obrázek 4.8: Pro vytvoření clony byla v případě těchto obrázků použita barva odpovídající barvě stěny. Obrázek vlevo je vzdálený pohled na celou scénu. Prostřední pohled znázorňuje pohled mírně z boku na celou scénu, ve které se virtuální skříňka nachází nad skutečnými skříňkami pod ní - je tedy na pozici nástěnných skříněk. Snímek vpravo zobrazuje scénu z nižší výšky.

2. testovací scéna

Obrázek 4.10 ukazuje testovací scénu, ve které je cílem zakrýt skutečný stůl s židlemi za účelem jejich nahrazení virtuálními objekty. Jejím účelem je zejména prezentovat, jak dokáže úhel pohledu na danou scénu ovlivnit výsledek zobrazení a ukázat situace, ve kterých navržená metoda nedokáže zakrýt vybraný objekt, nebo naopak dochází k nežádoucímu zakrytí i dalších objektů. Z tohoto důvodu bude, pro přehlednější demonstraci, ve scéně figurovat pouze clona, bez dalších virtuálních objektů.

Na obrázku 4.11 je vidět situace, kdy jsou z daného úhlu pohledu přesně skryty pouze vybrané objekty a je vytvářen dojem, jako by se v místnosti nacházel jen virtuální stůl s židlemi. Pokud se pozice pozorování posune směrem blíže k zemi 4.12, začne pomalu docházet k tomu, že clona umístěná na podlaze místnosti již nebude moci zcela zakrýt skutečné objekty. Na obrázku 4.13 jsou vidět další situace, ve kterých nedochází k plnému překrytí cílových objektů, nebo naopak dochází k překrytí i dalších objektů, které by zakryty být neměly.

Zhodnocení dosažených výsledků

Na dvou testovacích scénách byly ukázány možnosti výsledné aplikace, která dokáže do scény vkládat virtuální objekty, manipulovat s nimi a provádět maskování reálných objektů. Testovací scény prezentují jak situace, ve kterých navržené řešení přináší dobré vizuální výsledky, tak i situace, ve kterých již, vzhledem k principu svého fungování, nemůže metoda poskytnout žádané skrytí reálných objektů. Dosažený výsledek je, podle předpokladu sta-

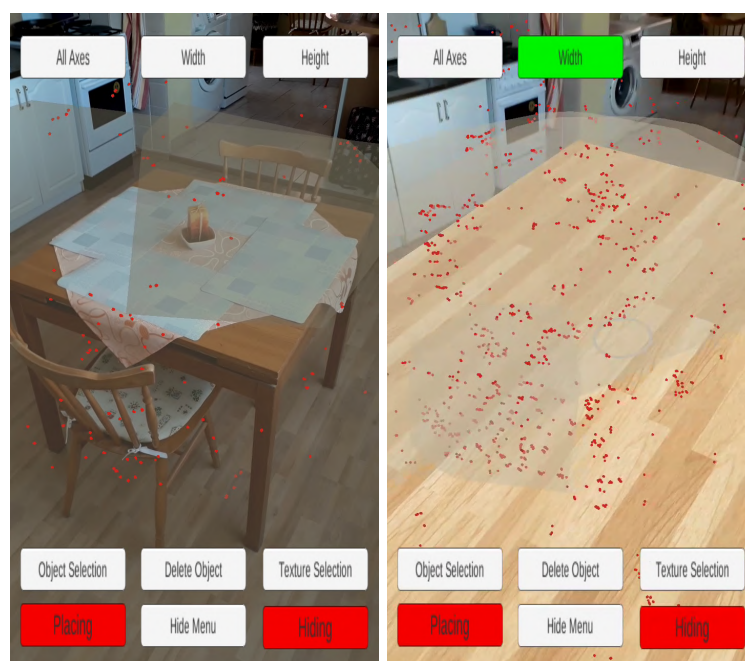


Obrázek 4.9: Pro vytvoření clony byla v případě těchto obrázků použita barva odpovídající barvě stěny. Na obrázku vlevo dochází, vzhledem k úhlu pohledu, k přesnému krytí cílové oblasti - clona nikde nepřesahuje. Vpravo je pohled z ostřejšího úhlu, kdy je vidět náznak toho, že clona již začíná zasahovat do prostoru dveří.

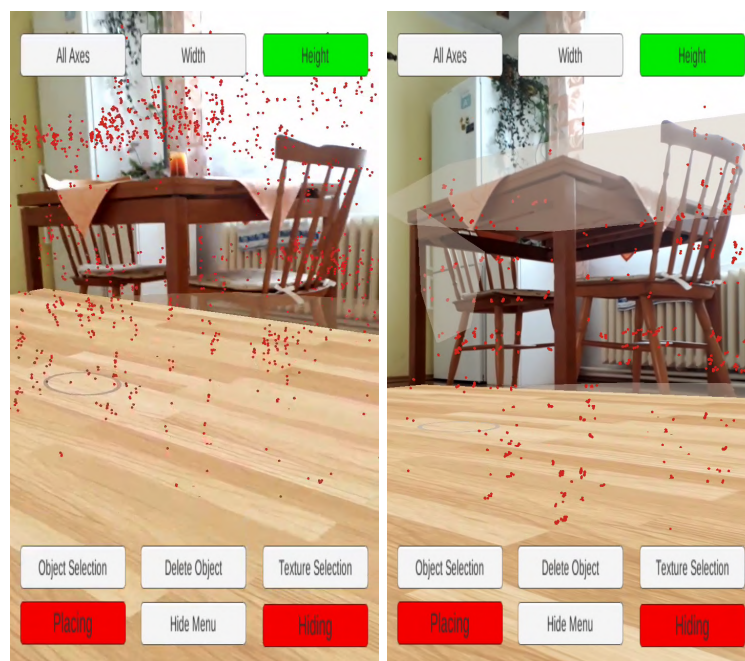


Obrázek 4.10: Obrázek vlevo zobrazuje celkový pohled na testovací scénu. Na obrázku vpravo je pak červeně označen jídelní stůl s židlemi určený k zakrytí.

noveného ve fázi návrhu (viz 3.2), závislý na úhlu pozorování scény, který ovlivňuje to, zda je skryta pouze cílová oblast, nebo dochází k maskování dalších částí, případně nedochází k úplnému maskování cílové oblasti.



Obrázek 4.11: Obrázek vlevo zobrazuje pohled na testovací scénu po nalezení rovin prostřednictvím ARCore. Vpravo je pohled ze stejného místa, přičemž zde dochází k úplnému zakrytí cílové oblasti.



Obrázek 4.12: Obrázek vlevo zobrazuje pohled na testovací scénu po posunu mobilního zařízení směrem k podlaze místnosti, což způsobí postupnou ztrátu překrytí. Obrázek vpravo je pohled téměř z úrovně podlahy, kdy jsou skrývané objekty již plně viditelné.



Obrázek 4.13: Obrázek vlevo zobrazuje pohled na testovací scénu z úhlu, ve kterém nedochází překrytí celého cílového objektu - červeně označená část stolu. Vpravo je pak vidět případ, kdy došlo k překrytí i dalších objektů - červeně vyznačená část bílé skříňky.

Kapitola 5

Závěr

Cílem této práce bylo nastudovat možnosti dnešních aplikací s rozšířenou realitou a navrhnout způsob, jakým lze ve scéně rozšířené reality skrýt vybrané reálné objekty, za účelem lepšího zobrazení virtuálních objektů.

V práci byla představena technologie rozšířené reality a její klíčové součásti včetně existujících aplikací s rozšířenou realitou, u kterých byly sledovány jejich vlastnosti a způsob ovládání. Na základě nastudovaných technologií a možností dnešních aplikací byla provedena analýza řešeného problému a navržen postup, jak umožnit skrýt vybrané reálné objekty ve scéně rozšířené reality. Kromě návrhu způsobu skrývání objektů je věnována pozornost i návrhu uživatelského rozhraní vyvíjené aplikace. Navržený způsob byl následně implementován do uživatelské aplikace s rozšířenou realitou s názvem *ObjectHider*, která v sobě kombinuje funkcionalitu umožňující vkládání virtuálních objektů, následnou manipulaci s nimi a vytváření virtuálního objektu nazvaného clona, určeného pro skrývání vybraných částí scény. Aplikace byla implementována s využitím možností poskytovaných knihovnou ARCore pro rozšířenou realitu na operační systém Android.

Nedílnou součástí práce bylo testování navrženého řešení v reálném prostředí a ověřování jeho vlastností, které prokázalo, že vytvořená aplikace disponuje jak kladnými, tak i negativními vlastnostmi, které byly identifikovány již v průběhu návrhu. V této části práce jsou prezentovány situace, ve kterých navržený postup poskytuje, z uživatelského hlediska, žádaný výsledek a dochází k vytvoření scény bez rušivého elementu, představovaného reálnými objekty. Rovněž jsou ukázány i případy, ve kterých navržený způsob, vzhledem k principu svého fungování, nemůže poskytnout uživatelem žádaný výsledek a dokonce může docházet i k tomu, že výsledná scéna působí méně přirozeněji, než kdyby se ve scéně nacházely pouze virtuální objekty bez clony.

Jako další pokračování vývoje by bylo zajímavé implementovat možnost provádět výřezy z vytvořené clony a tím například uživateli umožnit skrýt celou stěnu místnosti a následně v této cloně vyřezat otvor, který nebude překrývat reálné prostředí za clonou. Tímto bude možné ve výsledné scéně ponechat například okno, dveře, případně vypínač na zdi a docílit tak, v určitých případech, lepšího propojení mezi virtuálním a reálným prostředím.

Literatura

- [1] AZUMA, R. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*. 1996, sv. 6.
- [2] AZUMA, R., BAILLOT, Y., BEHRINGER, R., FEINER, S. K., JULIER, S. et al. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*. IEEE. 2001, sv. 21, s. 34–47.
- [3] BIMBER, O. a RASKAR, R. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. AK Peters/CRC Press, 2005. 71–92 s. ISBN 1568812302.
- [4] DURRANT WHYTE, H. a BAILEY, T. Simultaneous localization and mapping: part I. *IEEE Robotics automation Magazine*. 2006, sv. 13, č. 2, s. 99–110.
- [5] GOOGLE. *ARCore API reference* [online]. [cit. 2020-04-28]. Dostupné z: <https://developers.google.com/ar/reference/>.
- [6] GOOGLE. *ARCore overview* [online]. [cit. 2020-04-28]. Dostupné z: <https://developers.google.com/ar/discover>.
- [7] GOOGLE. *Fundamental concepts* [online]. [cit. 2020-04-28]. Dostupné z: <https://developers.google.com/ar/discover/concepts>.
- [8] HASBROUCK, C. *Breaking Augmented Reality out of the Screen* [online]. [cit. 2020-05-06]. Dostupné z: <https://blog.truthlabs.com/breaking-augmented-reality-out-of-the-screen-lightform-projection-mapping-ead91ff9844c>.
- [9] KURZER, R. *AR breaks out of the app onto the mobile browser* [online]. MarTech Today [cit. 2020-05-16]. Dostupné z: <https://martechtoday.com/ar-breaks-out-of-the-app-and-onto-the-mobile-browser-224018>.
- [10] PAGANI, A., HENRIQUES, J. a STRICKER, D. SENSORS FOR LOCATION-BASED AUGMENTED REALITY THE EXAMPLE OF GALILEO AND EGNOS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. IEEE. 2016, XLI-B1, s. 1173–1177.
- [11] SUTHERLAND, I. E. A head-mounted three dimensional display. *Proceedings of the AFIPS Fall Joint Computer Conference*. 1968, s. 757–764.

Příloha A

Obsah přiloženého DVD

/ObjectHider - zdrojové soubory výsledné aplikace

/zprava - zdrojové kódy tohoto dokumentu

plakat.png - plakát k práci

video-prezentace.mp4 - video ukazující činnost aplikace

readme.txt - popis struktury aplikace a její instalace

xkatru00.pdf - tento dokument

ObjectHider.apk - instalační soubor výsledné aplikace