Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informačních technologií

Interaktivní vizualizace pro mobilní zařízení

Diplomová práce

Autor práce: Bc. Michal Černý
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Karel Mls, Ph.D.

Hradec Králové 2021

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

. . . . . .

Michal Černý

30. dubna 2021

Poděkování

Rád bych poděkoval svému vedoucímu Ing. Karlu Mlsovi, Ph.D. za cenné rady a připomínky při vedení mé diplomové práce. Také bych chtěl poděkovat mé rodině a přítelkyni za podporu v průběhu studia.

Anotace

Tato práce se zabývá porovnáním nástrojů ARKit a ARCore pro vývoj aplikací s augmentovanou realitou na mobilních platformách a realizací aplikace na vizualizaci obrazu. V úvodní části je popsáno, co to augmentovaná realita je, jaké jiné druhy rozšířených realit existují a nástroje jsou porovnány na základě parametrů. Následuje porovnání obou nástrojů na reálných příkladech, konkrétně je porovnána detekce ploch reálného prostředí a odhad intenzity osvětlení. Poté je provedena analýza, návrh a implementace aplikace pro interaktivní vizualizaci obrazu v prostoru umožňující umístění obrazu na zeď, změnu jeho pozice, velikosti, rámu a vzhledu.

Annotation

Title: Interactive visualization for mobile devices

This thesis deals with comparing ARKit and ARCore software development kits for development of augmented reality applications on mobile platforms and development of an application for interactive picture visualization. The first part contains description of what augmented reality is, what other types of extended reality exist and the tools are compared based on their parameters. Both tools are compared in real-world scenarios, to be more specific the plane detection and the light estimation is compared. Following parts describe an application analysis, implementation design and implementation itself. The application allow user to place a picture on a wall, change its size, frame and appearance.

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Bývaly doby, kdy počítače byly velké jako celé místnosti a na cestu na Měsíc stačily 4KB operační paměti. Dnes se již nesrovnatelně výkonnější zařízení vejdou do batohu a pokud není požadavkem velký displej, stačí pouhé zápěstí. Je možné, že za pár (desítek) let nahradí tato zařízení pouhé chytré brýle s displejem pokrývajícím většinu zorného pole. Navigace zobrazená přímo na cestě, předpověď počasí dostupná při pohledu na nebe, druhá osoba stojící přímo před vámi i při vzdáleném hovoru, obrovská pracovní plocha nahrazující monitor, či nepřehlédnutelné notifikace, to jsou funkce, které by chytré brýle mohly nabízet.

Pokud se toto stane realitou, bude to zprostředkovávat dnes již existující technologie - augmentovaná realita (AR). Tato technologie, rozšiřující reálné prostředí o virtuální objekty, je v dnešní době dostupná převážně na chytrých mobilních zařízeních. Je možné se s ní setkat například při výběru nábytku ze známého švédského nábytkářského řetězce, nebo při přidávání různých obličejových filtrů a masek na známé sociální síti, ale i v oblastech jako je například vzdělávání, navigace, hry, průmysl a nakupování.

Augmentovaná realita stále není mezi širokou veřejností příliš rozšířená. Její propagaci napomohl příchod a rozvoj chytrých telefonů. Velký pokrok také zaznamenala, když v roce 2017 společnosti Apple a Google představily vlastní řešení pro implementaci této technologie a tím nabídly vývojářům silné nástroje pro tvorbu kvalitních AR aplikací. V případě společnosti Apple se jedná o nástroj ARKit a u společnosti Google se nazývá ARCore. V této práci jsou oba nástroje porovnány jak z hlediska dostupných funkcí a technických parametrů, tak i na reálných příkladech.

Úvodní část práce tvoří vysvětlení, co přesně augmentovaná realita je, jaké jiné druhy rozšířených realit existují, jaká je její historie a na jakém principu funguje. Následuje teoretické porovnání všech zásadních funkcí nabízených těmito nástroji spolu s podpůrnými nástroji od společností Google a Apple pro tvorbu augmentované reality.

Pro správnou funkci aplikací s augmentovanou realitou je klíčové správné a rychlé rozpoznání prostředí, konkrétně detekce ploch v reálném prostředí a odhad intenzity osvětlení. Tyto funkce budou porovnány na reálných příkladech.

Po provedené analýze a porovnání ARKit s ARCore bude vybrán jeden z nástrojů a pomocí něho bude vyvinuta aplikace umožňující interaktivní vizualizaci obrazu v prostoru.

Konkrétně umožní umístění 2D obrazu na reálnou zeď, úpravu jeho pozice, velikosti, otočení, rámu a vzhledu. Tato aplikace nalezne využití například pro fotografy, malíře a jejich zákazníky, kteří si budou moci dílo zkusit pověsit ve svém bytě ještě předtím, než ho koupí.

Téma jsem si zvolil, neboť věřím, že augmentovaná realita bude dříve či později součástí každodenního všedního života, stejně jako je tomu u počítačů a mobilních zařízení. Také z důvodu, že není mnoho prací a článků porovnávající ARKit s ARCore a v neposlední řadě z důvodu, že vyvinutá aplikace může být prospěšná pro umělce a jejich zákazníky.

2 Cíl práce

Cílem této diplomové práce je provést analýzu a porovnání mobilních platforem pro vývoj aplikací s augmentovanou realitou. Po tomto porovnání si jednu z platforem vybrat a implementovat aplikaci na interaktivní vizualizaci 2D objektů v 3D scéně. Při porovnání platforem bude cílem provést teoretické i praktické porovnání. Teoretické na základě dostupných informací o parametrech a funkcionalitách a praktické pomocí implementace shodných aplikací, které budou vizualizovat tvar rozpoznaných ploch a umístí 3D objekty do prostoru, aby bylo možné provést porovnání odhadu osvětlení. Cílem implementace aplikace pro vizualizaci 2D objektů v 3D scéně bude umožnit uživateli vybrat obraz z galerie a umístit ho na zeď, upravovat jeho pozici, rozměr, otočení, orámování a vzhled.

3 Analýza hlavních platforem AR

V této kapitole bude vysvětleno, co je to augmentovaná realita, jaké další typy rozšířených realit existují, pomocí jakých nástrojů se aplikace využívající augmentovanou realitu v mobilních zařízeních vyvíjejí a následně budou tyto nástroje porovnány.

3.1 Typy rozšířených realit

Augmentovaná realita bývá mylně nazývána rozšířenou realitou, nebo realitou virtuální. Rozšířená realita je ovšem pojem zastřešující augmentovanou realitu, augmentovanou virtualitu a virtuální realitu. V následujících bodech budou jednotlivé pojmy vysvětleny dle definic uvedených v [1].

Augmentovaná realita (AR) je označení používané pro reálný obraz světa doplněný o digitálně vytvořené prvky například pomocí mobilního zařízení. S prvky je typicky možné určitým způsobem interagovat - například měnit jejich pozici, velikost nebo vzhled. Příklad augmentované reality je na obrázku 1.

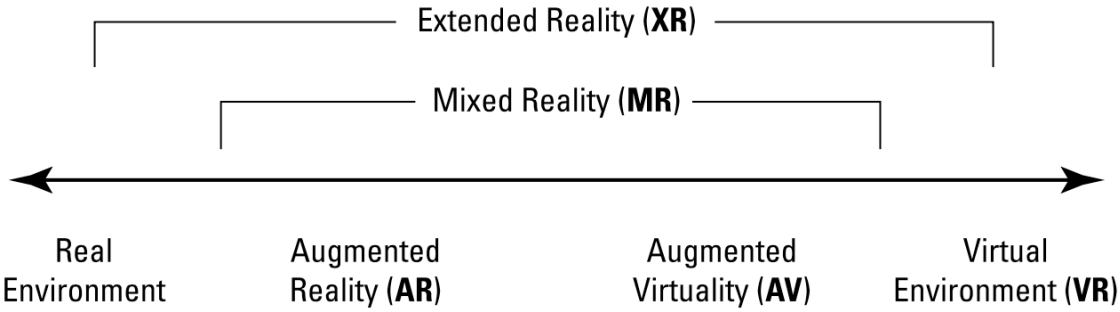
Obrázek 1: Augmentovaná realita [2]

Augmentovaná virtualita (AV) značí opak typické augmentované reality. Zatímco AR značí převážně reálný svět rozšířený o digitální objekty, AV značí převážně digitálně vytvořené prostředí s vloženými prvky z reálného světa. Příkladem může být projekt Alloy od firmy Intel, který do virtuální scény vkládá obraz uživatelových reálných rukou.

Virtuální realita (VR) vytváří fyzické prostředí, které neexistuje. Prostředí VR jsou typicky uzavřená od fyzického světa ve smyslu, že daná prostředí jsou kompletně nová, přestože digitální prostředí může být inspirováno reálným prostředím.

Mixovaná realita (MR) může vzít obraz reálného světa a vložit do něj digitálně vytvořený obsah, který může interagovat s obrazem reálného světa. Nebo může vzít plně digitální prostředí a propojit ho s objekty reálného světa. V tomto ohledu může být MR občas podobná VR a občas AR. V AR založené na MR není obsah digitálního světa pasivně položen na reálný svět, ale může interagovat jako kdyby byl jeho součástí. Digitální objekty vypadají jako by existovaly ve fyzickém prostředí a je možné s nimi interagovat jako kdyby tam opravdu byly.

Rozšířená realita (XR) je pojem zastřešující celé spektrum popsaných technologií. Dobře tyto pojmy popisuje virtuální kontinuum na obr. 2, které je měřítkem množství reality a virtuality. Na jedné straně je prostředí kompletně reálné a na druhé kompletně virtuální. XR zahrnuje celé toto spektrum.



Obrázek 2: Virtuální kontinuum [1]

3.2 Historie rozšířené reality

Za první představy o rozšířené realitě by se dala považovat povídka *Pygmalion's Spectacles* [3] z roku **1935** od Amerického spisovatele sci-fi Stanleyho Graumana Weinbaumana o profesorovi, který vynalezl brýle, které umožní uživateli spustit *„film, který poskytuje pohled a zvuk ... chuť, vůně a dotek. ... Jste v příběhu, mluvíte s postavami a oni odpovídají a místo*

toho, aby byli na obrazovce, příběh je o vás a vy jste v něm.“. [3] Weinbaum tím předpověděl nejen rozšířenou realitu, ale i vynález počítačů.

V roce **1955** kinematograf Morton Heilig, považovaný za otce virtuální reality, začal pracovat na vynálezu nejprve označovaném jako kino budoucnosti, později pojmenované jako Sensorama (obr. 3) [4]. Jednalo se o velký přístroj se sedačkou a prostorem pro umístění hlavy, který stimuloval vícero smyslů. Obsahoval mnoho funkcí moderních VR zařízení, jako je 3D displej umožňující prostorové vidění, stereo reproduktory a haptickou zpětnou vazbu pomocí vibrací skrze židli, na které uživatel seděl. Krátce po vynálezu Sensoramy si Heilig také patentoval první displej, který se umisťoval na hlavu, poskytoval prostorové vidění a stereo zvuk. Tento na tu dobu relativně malý displej se již více podobal dnešním VR zařízením.

Obrázek 3: Sensorama [4]

V roce **1965** Ivan Sutherland napsal esej pojednávající o takzvaném ultimátním displeji. Konkrétně ho popisoval následovně:
„Ultimátní displej by, samozřejmě, byla místnost ve které počítač může řídit existenci hmoty. Židle vyobrazená v takové místnosti by byla dostatečně dobrá na sezení. Pouta v takové místnosti by byla omezující a kulka by byla fatální. S vhodným naprogramováním takového displeje by mohl být doslova říše divů do kterého Alenka vstoupí.“ [5]
Další zajímavá část této eseje zmiňuje následující: *„Uživatel jednoho z dnešních displejů může jednoduše udělat plné objekty průhlednými - může vidět skrz hmotu!“* [5] Sutherland tím tedy předpověděl existenci virtuální i augmentované reality. O tři roky později, v roce

1968, dokončil první náhlavní displej přezdívaný Damoklův meč (obr. 4) [6]. Kvůli jeho váze musel být zavěšen na stropě. Tento displej již obsahoval snímaní pozice hlavy a optiku, skrze kterou se dalo vidět.

Obrázek 4: Damoklův meč [4]

V roce **1990** byl Tom Caudell, zaměstnanec výzkumného centra Boeingu, pověřen vytvořením náhrady za tehdejší systém velkých překližkových desek s instrukcemi zapojení elektronických komponent pro každé vyráběné letadlo [7]. Caudell a jeho kolega David Mizell navrhli displej na hlavu (obr. 5) pro zaměstnance montující letadla, který znázorňoval pozici kabelů skrze brýle a zobrazoval je na znovupoužitelných deskách. Místo použití různých desek pro každé letadlo byly instrukce v přístroji každého zaměstnance.

Obrázek 5: Přístroj zobrazující elektroinstalaci v letadlech vyvinutý společností Boeing [7]

V roce **1994** Paul Milgram a Fumio Kishino napsali semestrální práci „Taxonomy of Mixed Reality Visual Displays“ [8], kde definují virtuální kontinuum, které je měřítkem množství reality a virtuality (obr. 2). V roce **1997** Ronal Azuma představil svůj průzkum Augmentované reality [9], ve kterém definoval augmentovanou realitu třemi charakteristikami: kombinuje reálné a virtuální, je interaktivní v reálném čase a pracuje se třemi dimenzemi. Dodnes jsou definice od Milgrama, Kishina a Azumy považovány za definice augmentované reality.

V roce **1999** vydal Hirokazu Kato knihovnu ARToolKit na vývoj aplikací s augmentovanou realitou. V době psaní této práce je tato multiplatformní knihovna dostupná jako otevřený software a stále udržovaná komunitou.

V roce **2003** představili Daniel Wagner a Dieter Schmalstieg systém pro navigaci ve vnitřních prostorách [10]. Tento systém fungoval na PDA (kapesní počítač) bez nutnosti dalších podpůrných zařízení, takže byl dostupný relativně široké škále uživatelů. Tento systém používal na pozadí knihovnu ARToolKit.

V roce **2007** byl představen iPhone, první chytrý telefon s displejem podporujícím několik dotyků ve stejnou chvíli.

V roce **2010** byl podnikatel Palmer Luckey nespokojený s existujícími VR brýlemi na trhu. Téměř všechny byly drahé, extrémně těžké, měly malý zorný úhel a dlouhou latenci vedoucí k velmi špatnému uživatelskému zážitku. Luckey se rozhodl postavit prototyp levných a lehkých brýlí pro virtuální realitu s nízkou latencí a velkým zorným úhlem. Jeho šestá generace se jmenovala Oculus Rift a propagoval ji pomocí služby Kickstarter jako Rift Development Kit. Kampaň měla velký úspěch. Luckey dostal příspěvky ve výši 2,4 milionu dolarů, což bylo téměř 980 procent oproti původnímu cíli. Podstatné na této kampani bylo také to, že sloužila k celkovému pozvednutí zájmu o virtuální realitu.

V roce **2014** společnost Google představila projekt Tango, který byl dostupný pouze na dvou zařízeních [11]. Jednalo se o předchůdce aktuálního nástroje pro augmentovanou realitu od společnosti Google - ARCore.

V červnu roku **2017** společnost Apple na své každoroční konferenci pro vývojáře představila nástroj pro implementaci rozšířené reality ARKit [12]. V srpnu téhož roku představila obdobný nástroj s názvem ARCore i společnost Google [13]. Tím poskytly kvalitní nástroje pro jednoduchou implementaci na široké škále zařízení.

3.3 Využití AR v mobilních aplikacích

Augmentovaná realita v mobilních aplikacích má široké spektrum oblastí použitelnosti. Její možnosti by se mohly ještě rozšířit s příchodem brýlí pro augmentovanou realitu, které by se podařilo dostatečně rozšířit mezi běžné uživatele, podobně jako chytré telefony. O něco takového se v roce 2012 neúspěšně pokoušela společnost Google s brýlemi Google Glasses [14]. V následujícím seznamu jsou vypsány hlavní kategorie dle [15], kde aktuálně nalézá augmentovaná realita využití v případě mobilních aplikací.

- **Nakupování:** aplikace v této kategorii umožňují vizualizovat nabízený produkt na jeho budoucím místě. Příkladem může být aplikace Ikea Place (obr. 6), ve které má zákazník možnost si umístit produkty do svého interiéru, nebo ŠKODA AR App [16], pomocí které lze umístit Škoda Karoq například do vlastní garáže.

- **Zaměstnání:** hlavním cílem této kategorie je usnadnění a zefektivnění pracovního procesu. Příkladem může být aplikace společnosti GE (obr. 6), vizualizující průmyslová zařízení pro servisní pracovníky, kteří je mají opravovat.

- **Navigace:** rozšířená realita nalézá dobré uplatnění při navigaci. Přestože dnešní 2D navigační aplikace dokáží zobrazit směr, kterým uživatel směřuje, nemusí být pro všechny uživatele dostatečně intuitivní. V případě, že uživatel na obrazovce zařízení vidí reálné prostředí a šipky, nebo body zájmu, je navigace o poznání intuitivnější. Body zájmu mohou obsahovat jednoduše dostupné informace, například pokud se uživatel nalézá u obchodu, může být určitým způsobem vizualizován a po kliknutí na něj zobrazena například otevírací doba. Toho využívá aplikace společnosti American Airlines (obr. 6), která dokáže navigovat v prostorách letišť a dovést k odletové bráně nebo do obchodu.

- **Hry:** herní zážitek může být rozšířenou realitou umocněn nebo může přinášet herní zážitek úplně nový. Příkladem takových her je Monster Park (obr. 6), simulující dinosaury v reálném světě, nebo EVE: Project galaxy (obr. 6), hra pro více hráčů umožňující vést vesmírné bitvy i doma v obývacím pokoji.

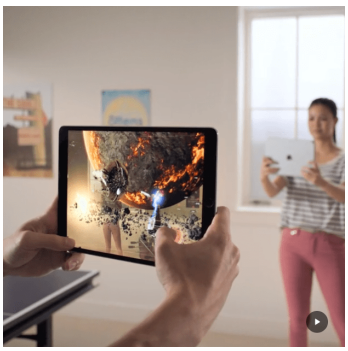
- **Vzdělání:** některé věci je jednodušší vyobrazit a pochopit v 3D prostoru, kde se objekty mohou pohybovat a lze s nimi libovolně manipulovat. Takovou aplikací je Complete Anatomy (obr. 6) vizualizující detaily lidského těla.
- **Turismus:** Podobně jako v případě kategorie navigace dokáží aplikace zabývající se turismem využívající augmentovanou realitu uživatele jednoduše navést na zajímavá místa a v těchto místech zobrazit dodatečné informace nebo vizualizace. V případě historických míst může být například vizualizováno, jak dané místo vypadalo v minulosti. Místo hledání v průvodcích tak stačí mít vhodnou aplikaci.

Ikea

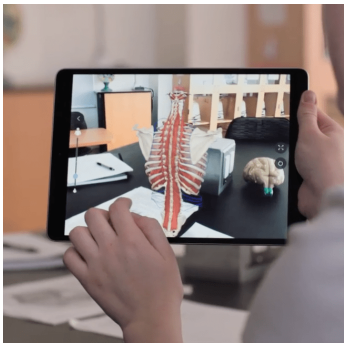
GE

American Airlines

Monster Park



EVE: Project galaxy



Complete Anatomy

Obrázek 6: Ukázky aplikací s augmentovanou realitou [17]

3.4 Princip funkce AR

V této části budou popsány základní principy fungování augmentované reality. Tyto principy jsou platné pro mobilní platformy využívající softwarové vývojářské nástroje ARKit a ARCore, které budou podrobněji popsány později. Většina z těchto principů je ale platných i pro jiné platformy a nástroje.

3.4.1 Detekce kotev

Aby bylo možné do reálného prostředí pozorovaného na mobilním zařízení pomocí fotoaparátu umístit virtuální objekt, musí mít vývojář referenci na objekty v reálném světě. Těmto objektům reálného světa se říká *anchors* neboli kotvy. Kotvami může být například horizontální a vertikální plocha, obraz, konkrétní objekt nebo obličej. Aby nástroj dokázal rozpoznat nějakou kotvu, detekuje v prostředí pomocí kamery tzv. *feature points* neboli body, které v prostředí rozpozná a dokáže sledovat jejich pozici i při změně polohy zařízení. Příklad takových detekovaných bodů je na obr. 7, kde jsou označené žlutými tečkami. Tyto body se vývojářský nástroj snaží shlukovat a vytvořit z nich kotvu. Například když body leží ve stejné horizontální rovině, vytvoří z nich vodorovnou plochu. Detekce obrazu probíhá v ARKit a v ARCore od verze 1.11 na vybraných zařízeních s frekvencí 60 obrazů za vteřinu.

Obrázek 7: Detekované body, tzv. feature points [18]

Pro detekci prostředí ale nestačí pouze kamera. Záznam z kamery je nutné zkombinovat se záznamem z gyroskopu a akcelerometru. Tím zařízení dokáže sledovat změnu natočení a změnu pozice a tím i svou polohu. ARKit i ARCore zaznamenává údaje z těchto senzorů s frekvencí 1 000 záznamů za vteřinu.

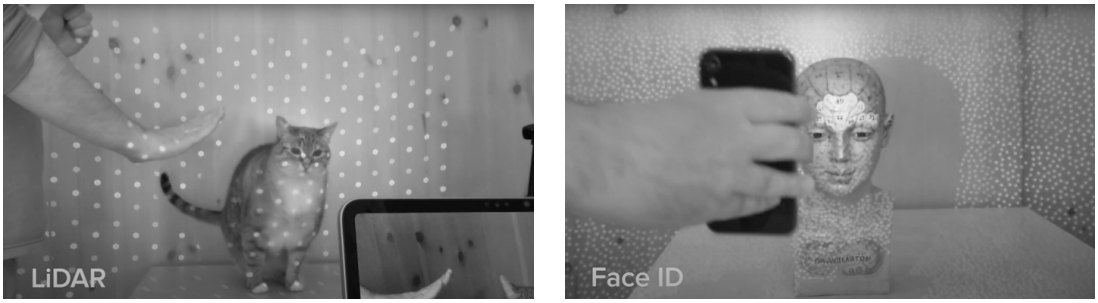
3.4.2 Time-of-Flight senzor

„Time-of-Flight je typ senzoru, který měří vzdálenost mezi sebou samým a všemi předměty, jež spadají do jeho úhlu záběru, a to pomocí světelných paprsků nebo ultrazvuku.“ [19]. V

případě mobilních zařízení se jedná o senzor fungující pomocí vysílání světelných paprsků a označuje se jako LiDAR, což je zkratka pro „Light Detection And Ranging“, v překladu světelné rozpoznávání a zaměřování. Obecně se dá říci, že LiDAR funguje na velmi podobném principu jako radar, s tím rozdílem, že radar vysílá rádiové vlny a LiDAR vyzařuje pulzy infračerveného záření ve vysoké frekvenci. Tyto pulzy se odráží od okolí a měří se doba mezi vysláním pulzu a jeho navrácením do senzoru. Z této doby se dá vypočítat vzdálenost v daném místě a kombinací údajů těchto pulzů se dát vytvořit mapa prostředí. Oproti radaru má LiDAR výhodu, že je mnohem přesnější a nevýhodu, že nedosahuje takové vzdálenosti. Konkrétně Apple LiDAR dosahuje vzdálenosti maximálně 5 metrů [20].

Tento senzor je dodávaný v posledních letech s nejlepšími mobilními zařízeními pro zlepšení detekce prostředí. Konkrétní Apple zařízení obsahující LiDAR v době psaní této práce jsou iPad Pro čtvrté generace (2020) a iPhone 12 Pro a Pro Max. Android zařízení s ToF senzorem a zároveň podporou ToF senzoru nástrojem ARCore je dle [21] 9 typů. Konkrétně to jsou LG V60 ThinQ 5G a určité specifikace Samsung Galaxy A80, Note10+, S10 a S20.

Záření vyzařované LiDAR senzorem je běžným okem neviditelné a pro jeho zobrazení je potřeba například speciální kamera. Záznam tohoto záření speciální kamerou je na obrázku 8, kde je navíc porovnáno s technologií Face ID pro rozpoznávání obličeje při přihlášení, které je na Apple zařízeních dostupné už od iPhone X představeného v roce 2017. Face ID je založené na podobném principu s tím rozdílem, že má menší dosah, více skenovacích bodů a senzor se umísťuje na přední stranu zařízení, oproti LiDARu, který je na zadní straně.



Obrázek 8: Srovnání Apple LiDAR a Apple Face ID technologií [22]

3.5 Vývojářské nástroje pro vývoj AR aplikací

Na trhu existuje několik nástrojů, pomocí kterých se dá augmentovaná realita implementovat. Liší se například množinou podporovaných zařízení, dostupnou funkcionalitou, programovacím jazykem a složitostí implementace. Každý z těchto nástrojů je dostupný jako takzvaný *Software Development Kit*, zkráceně SDK. V následujících podkapitolách budou srovnány dvě nejvýznamnější SDK pro mobilní platformy - ARKit a ARCore. Jedná se o nativní nástroje vyvinuté společnostmi Apple v případě ARKit a Google v případě ARCore.

3.6 Apple ARKit

ARKit [2] je vývojářský nástroj pro tvorbu aplikací s augmentovanou realitou od společnosti Apple, poprvé představený v červnu 2017 na vývojářské konferenci WWDC a v září 2017 se stal dostupný uživatelům. K vytvoření virtuálního světa na základě reálného (tzv. World Tracking) využívá techniku zvanou „visual-inertial odometry“. ARKit dokáže zaznamenat pozici a orientaci v šesti stupních volnosti. Záznam z kamery provádí s frekvencí šedesáti obrazů za sekundu a záznam z gyroskopu s frekvencí tisíc obrazů za sekundu.

ARKit 1.0

ARKit 1.0 [12], dostupný od iOS 11, představený na WWDC 2017 přinesl následující funkce:

- „World Tracking“ - využití kamery a pohybových senzorů zařízení pro kontinuální detekci jeho polohy v prostředí.
- „Scene Understanding“ - porozumění scéně, pojem zastřešující následující pojmy:
  - „Plane detection“ - detekce horizontálních ploch v reálném prostředí.
  - „Hit-Testing“ - slouží k nalezení kotvy v reálném prostředí pomocí vyslání pomyslného paprsku.
  - „Light estimation“ - odhad intenzity ambientního osvětlení. Odhad osvětlení je detailněji popsán v sekci 4.2.2.
- Podpora vykreslovacích nástrojů SpriteKit, SceneKit, Metal, Unity a Unreal. Všechny tyto nástroje jsou podrobněji popsány v sekci 3.6.1.

ARKit 1.5

ARKit 1.5 [23], dostupný od iOS 11.3, poskytl zlepšení v přesnosti odhadu tvaru detekovaných ploch, lepší rozlišení obrazu a automatické zaostřování obrazu. Kromě toho také přinesl následující funkce:

- Rozšíření „Plane detection“ - přidána možnost detekce vertikálních ploch.
- „Image detection“ - rozpoznání předdefinovaného statického 2D obrazu.

ARKit 2.0

ARKit 2.0 [24], dostupný od iOS 12, představený na WWDC 2018 přinesl následující funkce:

- „Multiuser colaboration“ - simultánní interakce v jedné scéně z více iOS zařízení a možnost pozorovatele, tj. sledování scény z jiného zařízení.
- „ARWorldMap“ - objekt poskytující aktuální mapování prostoru a rozpoznané kotvy. Tento objekt je možné uložit a později načíst, modifikovat a sdílet v rámci simultánní interakce.
- Možnost vrácení se k objektu po jeho umístění do prostoru například i po několika dnech.
- „Image tracking“ - kontinuální detekce 2D obrazů. V ARKit 1.5 bylo možné detekovat jeden statický obraz. ARKit 2.0 rozšiřuje tuto možnost na detekci až dvou obrazů zároveň, které se mohou pohybovat.
- „3D Object detection and scanning“ - detekce a skenování 3D objektů.
- Podpora formátu usdz pro 3D objekty vyvinutý společností Pixar, který lze sdílet a pomocí funkce QuickLook zobrazit i ve vybraných nativních aplikacích, kterými jsou Zprávy, Safari, Mail, Soubory a Novinky.
- Aplikace Měření, ve které je možné změřit velikost různých objektů.
- „Realistic reflections“ - využití strojového učení ke generování textury prostředí, která se dá využít jako reflexe objektu.
- „Face tracking“ - kontinuální detekce jednoho obličeje.

ARKit 3.0

ARKit 3.0 [25], dostupný od iOS 13, představený na WWDC 2019 přinesl následující funkce:

- Vylepšení detekce ploch pomocí metody zvané „Raycasting“, která umožňuje lepší výběr plochy pomocí specifikace parametrů plochy (např. typu a zarovnání). Raycasting nahrazuje „Hit-Testing“, který se již doporučuje nepoužívat.
- „People Occlusion“ - částečné, případně úplné zakrytí virtuálního objektu, pokud se mezi ním a zařízením objeví člověk.
- „RealityKit“ - vykreslovací nástroj pro usnadnění vkládání objektů do scény. Společně s ním přichází i aplikace „Reality Composer“ pro modelování různých objektů a jejich chování. Oba nástroje jsou detailněji popsány v sekci 3.6.1.
- „Body Motion Capture tracking“ - detekce lidského pohybu.
- „ARCoachingOverlayView“ - asistent rozpoznání prostoru. Před tím, než uživatel může umístit virtuální objekt na reálnou plochu, musí být tato plocha nástrojem ARKit rozpoznána. ARKit od verze 3.0 nabízí kvalitně zpracovaného průvodce pro navedení uživatele, co má pro správnou detekci prostředí udělat, tedy jakým způsobem má pohybovat se zařízením. Před verzí 3.0 si museli vývojáři implementovat tohoto průvodce sami.
- Současné použití přední a zadní kamery.
- „Image tracking“ - rozšíření kontinuální detekce obrazů až na 100 zároveň.
- „Face tracking“ - rozšíření kontinuální detekce obličeje až na 3 zároveň.

ARKit 4.0

ARKit 4.0 [26], dostupný od iOS 14, představený na WWDC 2020 přinesl následující funkce:

- „Location anchors“ - umístění objektu specifikací nadmořské výšky, šířky a délky. Funguje na základě lokalizační mapy, což jsou body z aplikace Apple Mapy.

- „Scene geometry“ - vytváří topologickou mapu prostředí. Umožňuje umístění virtuálních objektů přidaných do scény za reálné objekty (tzv. „object occlusion“, ne jen za osoby jako tomu bylo v ARKit 3). Je možné pouze na zařízeních s LiDAR senzorem.
- „Depth API“ - přístup k „Scene geometry“ datům.
- „Object placement“ - vylepšený Raycasting pomocí LiDAR senzoru.
- „Face tracking“ - kontinuální detekce obličeje pomocí přední kamery. Podporováno na všech zařízeních s „TrueDepth“ kamerou (platí pro ARKit 1-3). ARKit 4 podporuje kontinuální detekci obličeje i na zařízeních bez „TrueDepth“ kamery, pokud mají procesor A12 a novější.

3.6.1 Vykreslovací nástroje pro ARKit

ARKit sám o sobě nemá žádný vykreslovací nástroj a slouží pouze k detekci reálného světla a porozumění scéně. Je tedy nutné nějaký vykreslovací nástroj zvolit. Je možné vybírat z nástrojů, které jsou vyvíjeny společností Apple - tedy RealityKit, SceneKit, SpriteKit a Metal. Další možností je využít některý z neoficiálních nástrojů implementujících funkce ARKit, kterým je například Vuforia.

SceneKit

SceneKit je vysokoúrovňový vykreslovací nástroj. Ze všech vykreslovacích nástrojů pro ARKit je tím nejstarším - byl přestaven v roce 2012. Podporuje oba programovací jazyky, ve kterých se dají psát aplikace pro iOS, konkrétně tedy Objective-C a Swift, což je výhoda oproti nástroji RealityKit, který je novější a podporuje pouze Swift. Největší výhodou je vysoký rozsah jeho možností nastavení a změna geometrie a materiálů za běhu. Provádí vykreslování scény v rozsahu 30 až 120 obrazů za vteřinu a nabízí šest možností výpočtu osvětlení.

SpriteKit

SpriteKit je vysokoúrovňový nástroj pro vykreslování 2D grafiky představený v roce 2013. Pro ARKit může být použit samostatně nebo v kombinaci se SceneKit. Jeho hlavní funkce

jsou vykreslování obrazů, textu, tvarů, videa a simulace fyzikálních vlastností objektů. SpriteKit je možné používat s Objective-C i Swift programovacími jazyky.

Unity a Unreal

Unity a Unreal jsou nástroje využívané hlavně při tvorbě her. Pro vývoj jsou používaná odlišná vývojová prostředí, než je tomu u běžných nativních aplikací. Takto vyvinuté aplikace pak mohou být vloženy jako celek do běžných aplikací nebo mohou být samostatně distribuovány. Unity i Unreal podporují většinu funkcí nástroje ARKit a zprostředkovávají je v rámci svého prostředí.

RealityKit

RealityKit je vykreslovací nástroj, který Apple představil na WWDC 2019. Od základů byl budován pro vývoj ARKit aplikací s důrazem na maximální zjednodušení jejich vývoje. Toho bylo docíleno pomocí velmi jednoduchého vysokoúrovňového rozhraní, které RealityKit nabízí v kombinaci s programovacím jazykem Swift, který podporuje.

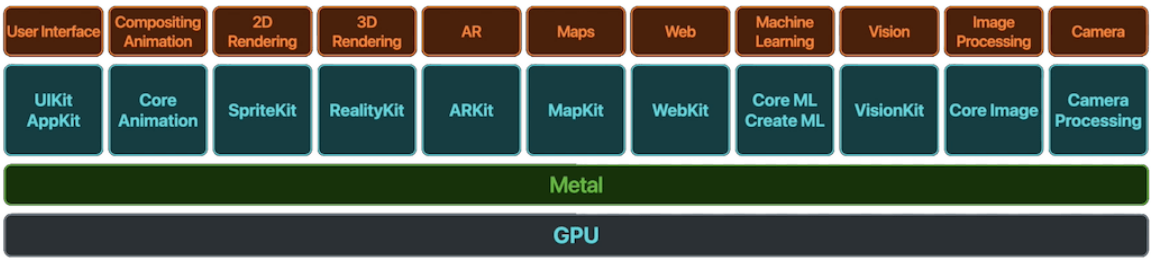
RealityKit obsahuje kvalitní vykreslovací technologii schopnou vytvářet velmi realistické, fyzikálně založené modely s přesnou simulací fyziky a kolizí s reálným světem. Tento nástroj dělá obsah tak dobře vypadající, jak jen to je možné. Seznam jeho hlavních funkcí je následující:

- **Animace:** Podpora animací nabízí možnost animace kostry modelu a transformaci objektu, takže je například možné jednoduše rozpohybovat postavu, přesunout ji na jiné místo, měnit její velikost a rotovat.
- **Synchronizace:** Framework má podporu kolaborace a zajišťuje automatickou synchronizaci mezi připojenými zařízeními.
- **Vykreslování:** RealityKit nabízí výkonné vykreslovací jádro postavené na Apple frameworku Metal, který je plně optimalizovaný pro všechny Apple zařízení.
- **Fyzika:** RealityKit podporuje přidávání fyzikálních parametrů objektům, díky kterým je možné tvořit reálně vypadající kolize.

- **Zvuk:** Jednotlivým objektům ve scéně je možné přidat zvuk, který se následně mění v závislosti na poloze a vzdálenosti od pozorovatele.
- **Video:** Objektům je možné jako materiál nastavit nejen barvu a statickou texturu, ale i video.

Metal

Metal je nízkoúrovňový nástroj vydaný v roce 2014 a používaný k vykreslování ostatními nástroji jako je SceneKit, RealityKit, AVFoundation, CoreML a další (obr. 9). Metal by se dal přirovnat svou funkcionalitou například k OpenGL. Využití nachází především u her s komplikovaným prostředím, při zpracování videa nebo při využívání výkonu grafické karty například u vědeckých výzkumů.



Obrázek 9: Nástroje využívající rozhraní Metal [27]

3.6.2 Další podpůrné nástroje pro ARKit

Apple na WWDC v roce 2019 představil několik nástrojů pro usnadnění vývoje aplikací s augmentovanou realitou. Konkrétně se jedná o popisovaný RealityKit, Reality Composer a Reality Converter.

Reality Composer

Reality Composer [28] je nástroj od společnosti Apple na jednoduchou tvorbu interaktivní augmentované reality bez nutnosti předchozí znalosti tvorby 3D objektů. Tento nástroj je dostupný v App Store pro iOS a iPadOS a přímo ve vývojovém prostředí Xcode pro MacOS. Vytvořené objekty je možné vyexportovat a používat v nástroji „Quick Look“ k jednoduché vizualizaci 3D objektů na Apple zařízeních nebo lze exportované objekty použít v ARKit aplikaci.

Při tvorbě modelu je možné použít některý ze stovek připravených přímo v tomto nástroji nebo naimportovat vlastní. U přidaných modelů je pak možné upravovat například velikost, styl, přidávat animace kostře modelu nebo celý model rozpohybovat, přidávat interakce například po kliknutí na objekt, když se k němu uživatel přiblíží a také objektům přiřadit nějaký zvuk. Hotový model je následně možné vyexportovat do formátu USDZ.

Obrázek 10: Reality Composer [28]

USDZ Tools

SceneKit a RealityKit mají pouze jeden společný podporovaný formát 3D souborů, kterým je formát usdz vyvinutý společností Pixar. Tento formát 3D souborů je tedy preferovaný při vývoji ARKit aplikací a společnost Apple připravila sadu nástrojů pro příkazový řádek, které s tímto formátem pracují. Tato sada se nazývá „USDZ Tools“ a obsahuje nástroj „usdzconvert“ pro konverzi do usdz z formátů obj, abc, usd, usda, usdc, glTF, glb a fbx. Také umožňuje kontrolu struktury usdz souborů, porovnání usdz souborů a další.

Reality Converter

Reality Converter [28] slouží podobně jako nástroj „usdzconvert“ ke konverzi 3D souborů do formátu usdz. Výhodou Reality Converter je, že má grafické rozhraní, v němž je možné si modely zobrazit s různým osvětlením, v různých prostředích a editovat jejich vlastnosti. Nevýhodou je, že podporuje méně formátů než USDZ Tools. Konkrétní formáty z kterých lze provést konverze do formátu usdz jsou obj, usd a glTF.

3.7 Google ARCore

ARCore je vývojářský nástroj pro implementaci rozšířené reality od společnosti Google pro jeho mobilní platformu Android. ARCore dokáže zaznamenat pozici a orientaci v šesti stupních volnosti pomocí techniky zvané „Concurrent Odometry and Mapping“. Záznam z kamery provádí s frekvencí šedesáti obrazů za sekundu a záznam z gyroskopu s frekvencí tisíc obrazů za sekundu. V následujícím seznamu budou představeny nejzásadnější vydané verze dle [29].

ARCore demo 1

ARCore byl poprvé přestaven jako demo v srpnu 2017 v článku na blogu společnosti Google [13]. Tato první verze byla dostupná pouze na zařízení Google Pixel a Samsung S8 a vycházela z platformy Tango, vyvíjené od roku 2014. Podporovala následující funkce:

- „Motion tracking“ - využití kamery a pohybových senzorů zařízení pro zachování umístění virtuálního objektu i při změně polohy zařízení.
- „Environmental understanding“ - rozpoznání horizontálních ploch.
- „Light estimation“ - detekce ambientní složky osvětlení umožňující upravit osvětlení virtuálních objektů.

ARCore demo 2

Druhé demo bylo představeno v prosinci 2017 a přineslo navíc podporu Unity a Unreal. Dále umožňovalo přerušit a později pokračovat v AR session. V neposlední řadě zlepšilo přesnost a výkonost [30].

ARCore 1.0

Představení první oficiální verze - ARCore 1.0 proběhlo v únoru 2018 [31]. Podporovalo 13 Android zařízení a umožňovalo detekci nejen ploch, ale i objektu s texturou.

ARCore 1.2

ARCore 1.2, představený na Google I/O 2018 [32] přinesl následující funkce:

- „Cloud Anchors“ - funkce umožňující vývojářům vytvořit sdílený AR zážitek napříč iOS a Android zařízeními pomocí transformace kotev vytvořených na jednom zařízení na „Cloud anchors“ a jejich následné sdílení s ostatními zařízeními.
- „Augmented Images“ - umožňuje ARCore aplikacím detekovat obrazy a v reálném čase pracovat s jejich pozicí.
- „Vertical plane detection“ - detekce vertikálních ploch.

ARCore 1.7

Datum představení: březen 2019.

- Přední kamera a „Augmented Faces“ - podpora přední kamery pro detekce obličeje. S přední kamerou není možné rozpoznávat jiné kotvy než je obličej, tedy není možné rozpoznat plochy a obrazy.
- „Shared Camera access“ - umožňuje přístup ke kameře v AR módu a běžném módu současně.

ARCore 1.9

ARCore 1.9, představený na Google I/O 2019 [33] přinesl následující funkce:

- Vylepšení „Augmented images“ - nově umožňující sledovat pozici obrazů i když se pohybují.
- Vylepšení „Light estimation“ - nově využívající strojové učení pro věrohodnější vykreslení stínů, reflektujícího a výrazného světla.
- „Scene Viewer“ - otevření 3D scény z webové stránky.

ARCore 1.10

Datum představení: červen 2019.

- „Environmental HDR mode“ - rozšíření „Light estimation“ o tři rozhraní pro replikaci světelnosti reálného světa při použití zadní kamery. Tyto tři rozhraní jsou následující:
  - „Main Directional Light“ - pomáhá s vržením stínů správným směrem.

  - „Ambient Spherical Harmonics“ - pomáhá modelovat ambientní osvětlení ze všech směrů.
  - „HDR Cubemap“ - poskytuje zrcadlení a odrazy.

ARCore 1.11

Datum představení: srpen 2019.

- Podpora záznamu z kamery s frekvencí 60 obrazů za sekundu na vybraných zařízeních.

ARCore 1.18

Datum představení: červen 2020.

- „Depth API“ - poskytuje vypočítaný hloubkový obraz pro každý obraz z kamery. Toto rozhraní je podporované pouze vybranými zařízeními, v závislosti na výkonu jejich procesoru. LiDAR není nutný.

ARCore 1.19

Datum představení: září 2020.

- „Instant Placement API“ - rozhraní umožňující obejití kroku detekce povrchu pro rychlejší umístění objektů v dané scéně. Pokud se uživatel pohybuje dále v daném prostředí, toto rozhraní pozici objektu může změnit a zpřesnit.

ARCore 1.20

Datum představení: říjen 2020.

- „Persisten Cloud Anchors“ - možnost prodloužení doby zapamatování Cloud Anchors až na 365 dní.

ARCore 1.21

Datum představení: listopad 2020.

- Přidána možnost záznamu a přehrání ARCore dat. Funkce záznamu umožňuje zaznamenat data potřebná k přehrání AR session.

3.7.1 Vykreslovací nástroje ARCore

ARCore stejně jako ARKit neobsahuje vykreslovací nástroj a je třeba nějaký zvolit. Rozsah oficiálně doporučovaných možností společností Google je oproti ARKit omezenější, konkrétně je možné volit mezi vysokoúrovňovým nástrojem Sceneform, nízkoúrovňovým OpenGL, Unity a Unreal. Další možností je využít některý z neoficiálních nástrojů implementující podporu ARCore, kterým je například Vufofia.

Sceneform

Sceneform je vykreslovací nástroj vyvinutý společností Google s vysokoúrovňovým rozhraním podobným nástroji SceneKit. Jeho hlavním cílem je co nejvíce usnadnit vývojáři implementaci rozšířené reality. Vývoj Sceneform byl ukončen v květnu roku 2020 a nebyl nahrazen žádným novým nástrojem.

OpenGL

OpenGL je multiplatformní, obecné, nízkoúrovňové rozhraní pro vykreslování 2D a 3D grafiky. ARCore nabízí podporu OpenGL již od svého vydání.

Unity a Unreal

Unity a Unreal popsané v 3.6.1 podporují ARCore podobně jako ARKit.

3.8 Podporované formáty 3D souborů

Každý z oficiálně doporučovaných vykreslovacích nástrojů s vysokoúrovňovým rozhraním, kterými jsou SceneKit a RealityKit pro ARKit a Sceneform pro ARCore, podporuje určité formáty 3D souborů a v této části bude tato podpora porovnána.

Jak lze vidět v tabulce 1, nástroje pro ARKit pokrývají větší množinu formátů. Některé z nich nepodporuje přímo, ale je možné je pomocí *USDZ Tools* nebo *Reality Converter* nástrojů konvertovat do formátu usdz, který podporuje SceneKit i RealityKit

Sceneform pro ARCore má seznam podporovaných formátů kratší. Stejně jako SceneKit podporuje formát obj v kombinaci s definicí materiálů ve formátu mtl, které nejsou součástí obj souboru. Dále podporuje glTF a glb formát (binárně zakódovaný glTF), ovšem pouze

bez animací. Dále nabízí podporu formátů fbx, sfa. Formát sfb Sceneform dříve podporoval, ve verzi 1.16 ho však označil za zastaralý.

Tabulka 1: Podporované formáty 3D souborů

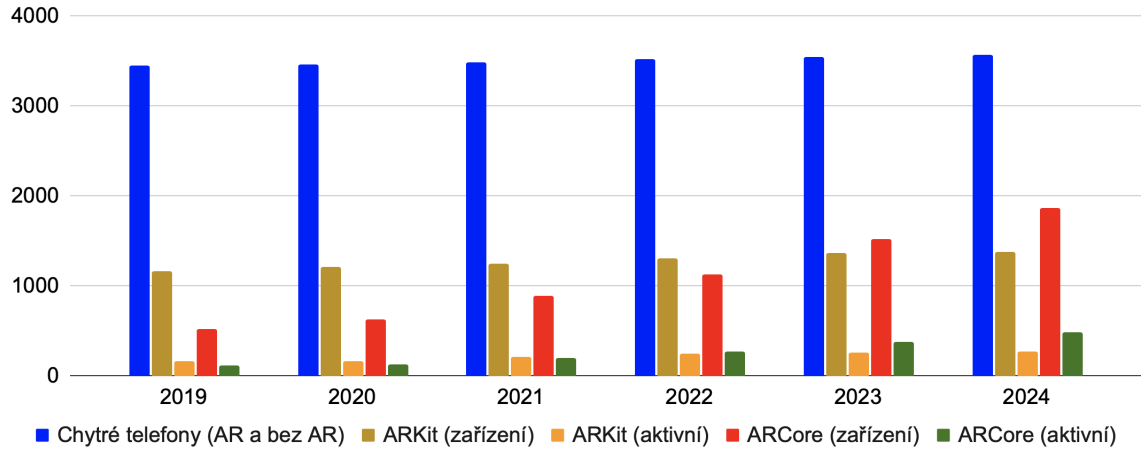
Formát	SceneKit	RealityKit	Sceneform
dae	✓		
obj	✓	* **	✓
abc	✓	*	
scn	✓		
ply	✓		
stl	✓		
usd	✓	* **	
usda	✓	*	
usdc	✓	*	
usdz	✓	✓	
reality		✓	
rcproject		✓	
glTF	* **	* **	✓(bez animací)
glb	*	*	✓(bez animací)
fbx	*	*	✓
sfa			✓
sfb			✓(zastaralé)
* možná konverze do usdz formátu pomocí USDZ Tools			
** možná konverze do usdz formátu pomocí Reality Converter			

3.9 Podporovaná zařízení

Při srovnávání ARKit s ARCore je důležité neopomenout možnou velikost cílové skupiny, respektive počet podporovaných zařízení těmito nástroji. Tuto statistiku pravidelně zpracovává společnost Artillery Intelligence [34], která se speciálně zabývá sledováním vývoje rozšířené reality a tvorbou reportů. Počty zařízení od roku 2019 a jejich odhady do roku 2024 jsou znázorněny v grafu 11. Tento graf udává celkový počet chytrých zařízení s podporou augmentované reality i bez ní, celkový počet AR aktivních zařízení, počet aktivních a celkový počet ARKit a ARCore zařízení.

Za povšimnutí stojí, že celkový počet chytrých zařízení se téměř nemění, ale počet AR aktivních se zvýší téměř čtyřnásobně. Dále je zajímavé, že počty ARKit zařízení se téměř nemění vzhledem k tomu, že již v roce 2019 měl ARKit rozsáhlou podporu svých zařízení. Naopak je tomu u ARCore, který při svém uvedení podporoval pouze 13 typů zařízení, ale

postupně jejich seznam rozšiřuje a podle Artillery Intelligence v roce 2023 překoná celkovým počtem podporovaných zařízení nástroj ARKit.



Obrázek 11: Počty ARKit a ARCore podporovaných zařízení v milionech. Zdroj dat: [34], vlastní tvorba.

3.10 Závěr porovnání

ARKit i ARCore nabízejí velmi podobnou sadu základních funkcí. Ty nejvýznamnější jsou znázorněny v tabulce 2. Oba nástroje detekují prostředí na podobném principu (popsaném v 3.4) a nabízejí podobné porozumění prostředí - kontinuální detekci pohybu mobilního zařízení, horizontálních i vertikálních ploch, obličeje, 2D obrazů a 3D objektů. ARKit nabízí možnost detekce více obličejů zároveň a také více 2D obrazů, než je tomu u ARCore. Také nabízí možnost kontinuální detekce postavy a vývojáři poskytuje kostru pohybujícího se těla, se kterou je možné dále pracovat.

Oba nástroje provádějí odhad osvětlení. Vzhledem k rozsáhlosti problematiky odhadu osvětlení bude podrobný popis nabízených funkcí a srovnání na reálném příkladu popsáno později v kapitole 4.2.

Některé funkce jsou podporované jen u jednoho z nástrojů. Konkrétně je tomu tak u současného použití přední a zadní kamery zároveň, kterou podporuje pouze ARKit. Spustit aplikaci na virtuálním simulátoru bez nutnosti použití reálného zařízení zase podporuje pouze ARCore.

Společnost Apple má výhodu, že vyvíjí současně software i hardware a proto má možnost jejich perfektního propojení. V případě použití obrazových dat z kamery a polohových

dat z akcelerometru a gyroskopu výhoda tohoto propojení není tak důležitá, jako je tomu u senzoru LiDAR, u kterého je propojení se software zásadní, aby byl schopný možnosti LiDAR senzoru naplno využít. Pomocí LiDARu je možné detekovat nejen standardní kotvy, kterými jsou plochy, obrazy, objekty, obličej a tělo, ale dokáže vytvořit hloubkovou mapu celého prostředí. ARCore poskytuje také hloubkovou mapu prostředí ovšem bez využití LiDAR senzoru. Má tedy výhodu, že je dostupný na větším množství zařízení, ovšem není tak kvalitní.

Dosud popsané funkce byly u obou konkurentů relativně srovnatelné. Jinak tomu je v porovnání nástrojů podporujících vývoj, kde zřetelně dominuje ARKit. Ať už jde o vykreslovací nástroje, kterými je SceneKit, SpriteKit a RealityKit, nebo nástroj pro tvorbu 3D scén Reality Composer nebo nástroje na konverzi 3D formátů Reality Converter a USDZ Tools. ARCore nabízí pouze již dále nevyvíjený vykreslovací nástroj Sceneform.

Tabulka 2: Funkce jednotlivých nástrojů pro augmentovanou realitu

	ARKit	ARCore
Kontinuální detekce pohybu zařízení	✓	✓
Kontinuální detekce horizontálních ploch	✓	✓
Kontinuální detekce vertikálních ploch	✓	✓
Kontinuální detekce obličeje	✓(max. 3)	✓(max. 1)
Kontinuální detekce 2D obrazů	✓(max. 100)	✓(max. 20)
Kontinuální detekce 3D objektu	✓	✓
Kontinuální detekce pohybu těla	✓	
Odhad světla	✓	✓
Sdílení relace	✓	✓
Skrytí virtuálního objektu za osobu	✓	✓
Použití přední a zadní kamery zároveň	✓	
Podpora simulátoru		✓
Podpora Time-of-Flight senzoru	✓(3 typy zařízení)	✓(8 typů zařízení)
Podpora detekce hloubky	✓(pouze s ToF s.)	✓(165 typů zařízení)
Unreal a Unity podpora	✓	✓
Dokumentace	kvalitní	špatná
Operační systém	iOS	Android, částečně iOS
Cena	zdarma	zdarma
Programovací jazyk	Objective-C, Swift	Java, Kotlin
Nástroje:		
ARKit: SceneKit, SpriteKit, RealityKit, Reality Composer a Converter, USDZ Tools		
ARCore: Sceneform		

4 Praktické porovnání platforem AR

V kapitole 3 byl porovnán ARKit s ARCore z hlediska specifikací a parametrů. V této kapitole budou nástroje porovnány hlavně prakticky na reálných příkladech. Konkrétně budou porovnány jedny z nejdůležitějších funkcí těchto nástrojů, kterými jsou detekce ploch a odhad osvětlení.

4.1 Rozpoznání prostředí

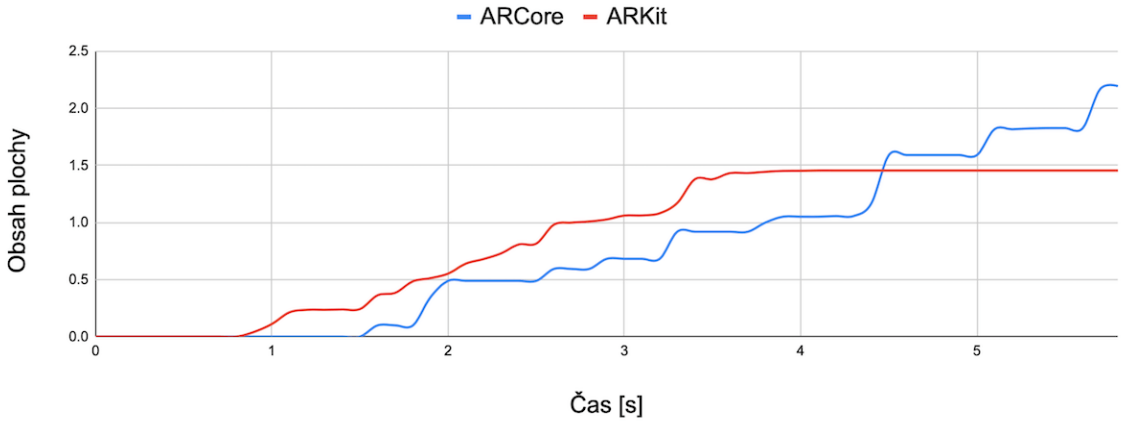
V této části bude testováno rozpoznání prostředí pomocí nástrojů ARKit a ARCore. Pro testování ARKit byl použit iPhone XR a v případě ARCore byl použit Xiaomi Redmi 7. Testování probíhalo na běžném objektu pro mnoho AR aplikací, kterým je běžný stůl. Testovací stůl má rozměry 180 x 70 cm, tedy celkový obsah je 1,24 m^2. Měření byla prováděna za běžného denního osvětlení v místnosti.

Obsah plochy

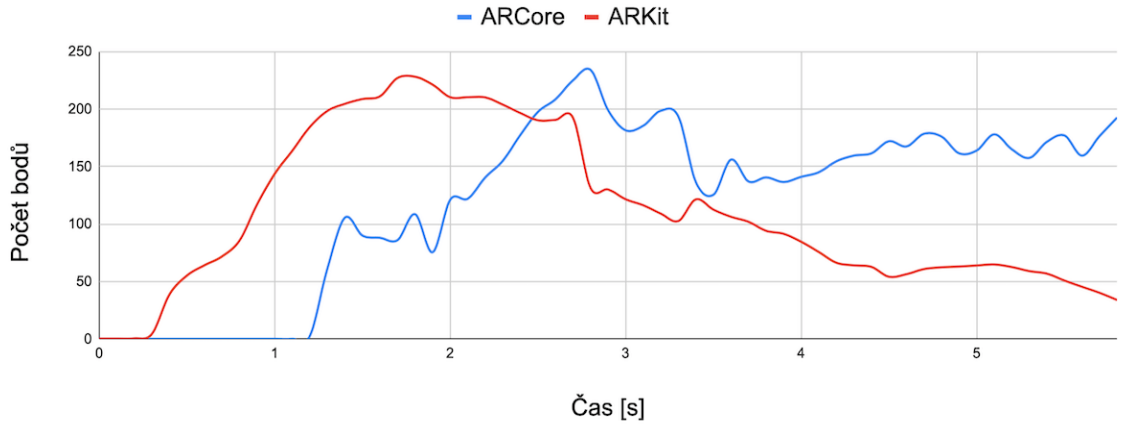
První parametr, který bude testován je obsah rozpoznané plochy objektu v čase. Na grafu 12 jsou vidět průměrné výsledky měření. Je na nich patrné, že první rozpoznání plochy proběhne v případě ARKit po 0,9 s a v případě ARCore po 1,5 s. Celá plocha je rozpoznána v případě ARKit po 3,3 s a v případě ARCore po 4,5 s. Nejzajímavější ale na tomto grafu je, že ARKit zaznamená o něco větší plochu, než ve skutečnosti stůl má (1,4 m^2 místo 1,24 m^2), ale tato plocha se již dále nerozšiřuje. V případě ARCore to tak není a plocha se rozšíří až na 2,2 m^2.

Počet bodů

Druhým testovaným parametrem je počet rozpoznaných bodů prostředí v čase. Počet a rychlost jejich rozpoznání je klíčová pro správnou funkci nástrojů pro augmentovanou realitu. Jak může být vidět v grafu 13, ARKit začne body rychle rozpoznávat již po 0,3 s, po 1,7 s jich má maximum a začíná je shlukovat do ploch, čímž některé z nich vyhodnotí jako nedůležité pro další sledování a počet těchto bodů klesá. V případě ARCore dochází k prvnímu rozpoznání bodů po 1,2 s a nejvíce jich má po 2,8 s.



Obrázek 12: Obsah rozpoznané plochy v čase [vlastní tvorba]



Obrázek 13: Počet rozpoznaných bodů v čase [vlastní tvorba]

4.2 Osvětlení

I když je virtuální model přidaný do reálného prostředí propracovaný do posledního detailu a může působit sebevíc reálně, ve chvíli kdy vlastnosti jeho osvětlení, stíny a odrazy v případě lesklých předmětů neodpovídají prostředí, je na první pohled zřejmé, že do prostředí nepatří. Například při absenci stínů není lidský mozek v určitých situacích schopný rozhodnout, kde přesně se objekt v prostoru nachází a může se zdát, že levituje. Odhad parametrů osvětlení prostředí tedy patří mezi základní funkce vývojářských nástrojů a proto ho ARKit i ARCore obsahují od prvních verzí a průběžně vylepšují. V této části budou srovnány řešení odhadu parametrů osvětlení obou těchto nástrojů.

4.2.1 Složky osvětlení

Na obrázku 14 jsou popsány důležité složky určující výsledný vzhled objektu. Jejich popis je následující:

- **Ambientní osvětlení:** všudypřítomné osvětlení - projevuje se na všech částech objektu a jeho intenzita se dá nejlépe odhadnout například ve stínu tělesa bez odlesků, protože tam není násobeno ostatními složkami osvětlení.
- **Odraz:** promítá v sobě okolní prostředí. Přítomnost, případně ostrost odrazu závisí na vlastnostech materiálu objektu.
- **Difúzní osvětlení:** závisí na poloze pozorovatele vůči objektu. Určitý objekt v prostoru může mít na stejném místě různé osvětlení v závislosti na úhlu dopadajícího a odraženého světla vůči pozorovateli, tedy v závislosti na jeho poloze.
- **Stín:** pozorovatel dokáže pomocí stínu lépe určit přesnou polohu objektu, hlavně ve vertikální ose a také zdroj světla.
- **Lesklé osvětlení:** lesklé části povrchu objektu. Stejně jako difúzní osvětlení závisí na poloze pozorovatele vůči objektu.
- **Barva ambientního osvětlení:** lesklé materiály reflektují barvu osvětlení. Například bílý objekt s modrým osvětlením bude mít namodralý povrch.

Obrázek 14: Složky osvětlení [35]

4.2.2 ARKit

ARKit obsahuje tři třídy popisující odhad osvětlení a okolního prostředí, u kterých jsou vždy uvedeny parametry popisující odhad osvětlení a prostředí:

- **AREnvironmentProbeAnchor:** jedná se o potomka *ARAnchor* a obsahuje mapu prostředí okolo určitého bodu.
  - *environmentTexture:* textura prostředí kolem této kotvy.
  - *extent:* oblast kolem polohy kotvy, která obsahuje texturu.
- **ARLightEstimate:** pokud je povoleno *isLightEstimationEnabled* v nastavení ARSession, ARKit poskytuje odhad osvětlení pro každý záznam obrazu kamery ARFrame.
  - *ambientIntensity:* odhad ambientní složky osvětlení v lumenech.
  - *ambientColorTemperature:* odhad teploty barvy ambientního osvětlení v kelvinech.
- **ARDirectionalLightEstimate:** poskytuje podrobnější informace o odhadu osvětlení než *ARLightEstimate*, ale je dostupný pouze při skenování obličeje.
  - *sphericalHarmonicsCoefficients:* data popisující odhad světelných podmínek ve všech směrech.
  - *primaryLightDirection:* vektor popisující směr nejsilnějšího směrového osvětlení ve scéně.
  - *primaryLightIntensity:* odhad intenzity nejsilnějšího směrového osvětlení ve scéně v lumenech.

4.2.3 ARCore

ARCore nabízí tři módy odhadu osvětlení v rámci výčtového typu *Config.LightEstimationMode*:

- **ENVIRONMENTAL_HDR:** tento mód se skládá z několika oddělených rozhraní, která poskytují informace o osvětlení pro dosažení realisticky vypadajících stínů, odlesků, odrazů a difúzního osvětlení. Používá strojové učení pro analýzu obrazu v reálném čase. Konkrétně poskytuje následující rozhraní:

  - **Hlavní směrové světlo:** reprezentuje hlavní zdroj světla. Může být použito pro správné vržení stínů.
  - **Ambientní osvětlení:** reprezentuje zbylé ambientní osvětlení.
  - **HDR mapa:** reprezentuje texturu prostředí, která může být použita pro odrazy lesklých objektů.

- **AMBIENT_INTENSITY:** Tento mód poskytuje průměrnou intenzitu a barvu osvětlení pro konkrétní zaznamenaný obraz. Může být použito v případech, kdy není potřebné znát přesné parametry osvětlení.

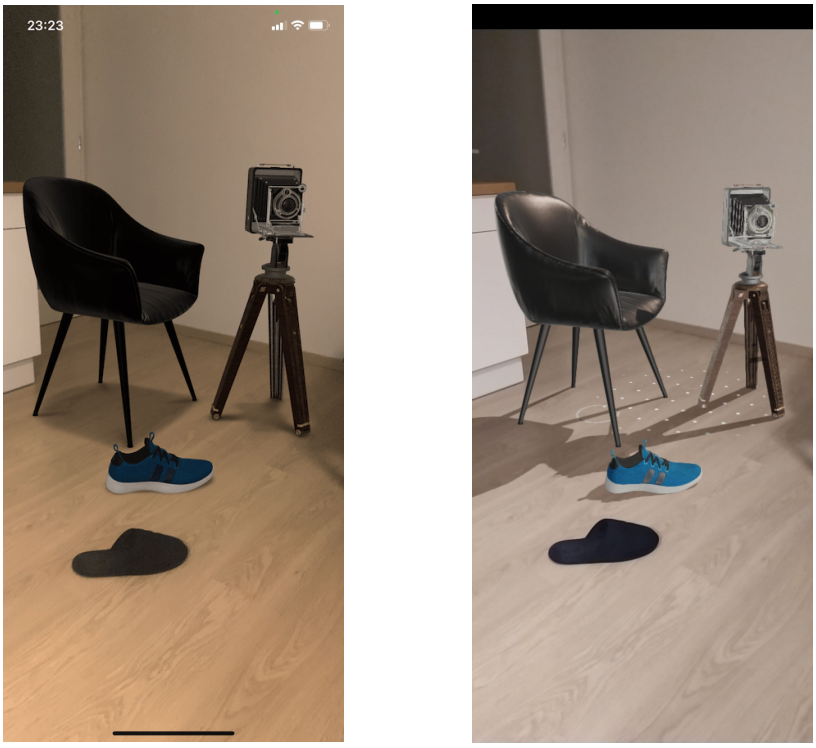
- **DISABLED:** odhad osvětlení je vypnutý a ARCore o něm neposkytuje žádné informace.

4.2.4 Porovnání při reálném použití

ARKit i ARCore řešení bylo porovnáno s použitím Apple RealityKit a Google Sceneform nástrojů. RealityKit i Sceneform provádí odhad osvětlení a jeho aplikaci již ve výchozím nastavení, které bylo zachováno. Testování probíhalo za denního i umělého světla.

Pro testování byly vybrány tři objekty. Jelikož RealityKit a Sceneform nemá ani jediný společný podporovaný formát 3D objektů, byly vybrány objekty ve formátu glb, které podporuje nástroj Sceneform a zkonvertovány do formátu usdz pro RealityKit pomocí nástroje *Reality Converter*, který byl vyvinut společností Apple.

Prvním testovacím objektem je kožené křeslo černé barvy, které má velmi lesklý a zároveň drsný povrch. Má tedy na sobě mnoho odlesků, ale bez odrazů. Druhým objektem byla modrá bota. Pro porovnání byla v reálném prostředí umístěna pantofle také modré barvy. Virtuální bota má převážně matný povrch, kromě lesklých pruhů na jejím boku. Poslední objekt je historický fotoaparát. Tento objekt je složený z různých materiálů - má dřevěné nohy, skleněnou čočku, kovový podstavec a gumovou clonu. Výsledky porovnání jsou na obrázcích 15 a 16, na kterých je vidět, že Sceneform odhaduje silnější světlo a objekty působí reálněji při použití nástroje RealityKit. Na obrázcích je také vidět, že Sceneform se snaží odhadnout umístění zdroje světla a přizpůsobit mu zobrazení stínů. RealityKit vrhá stíny vždy pod objety jako by byl zdroj světla vždy nad nimi.



Obrázek 15: RealityKit (vlevo) vs. Sceneform (vpravo) s umělým osvětlením [vlastní tvorba]

Obrázek 16: RealityKit (vlevo) vs. Sceneform (vpravo) s denním osvětlením [vlastní tvorba]

5 Analýza aplikace

Výsledkem této práce bude mobilní aplikace určená pro umístění 2D objektů do 3D scény. V této kapitole bude navržen seznam požadavků, které bude tato aplikace splňovat a také budou analyzovány aplikace s podobnými požadavky.

5.1 Analýza požadavků

Analýza požadavků slouží hlavně k vymezení hranic systému, umožnění přesnějšího odhadu pracnosti, vyjasnění si zadání a zachycení omezení, která jsou na systém kladena.

5.1.1 Funkční požadavky

Funkční požadavky specifikují funkce, které jsou od aplikace očekávány.

FP1 Výběr obrazu z galerie Bude umožněno vybrat libovolný obraz z galerie pro následné umístění do 3D prostoru.

FP2 Detekce ploch Aplikace bude detekovat plochy v reálném prostředí, které bude následně možné využít pro vizualizaci.

FP3 Umístění obrazu Vybraný obraz bude možné umístit na jednu z detekovaných ploch.

FP4 Odebrání obrazu Po umístění obrazu do prostoru jej bude možné i odebrat.

FP5 Změna pozice Pozici obrazu bude možné změnit i po jeho umístění.

FP6 Změna velikosti Velikost obrazu bude možné změnit. Poměr stran bude při změně velikosti zachován.

FP7 Zobrazení detekovaných ploch Detekované plochy bude možné vhodným způsobem zobrazit.

FP8 Rám Obrazu ve scéně bude možné nastavit rám. Rám bude mít konfigurovatelnou šířku a barvu.

FP9 Efekty Obraz ve scéně bude možné upravit pomocí efektů, například změnit do sépiových odstínů.

FP10 Otáčení Obraz bude možné otáčet po devadesáti stupních.

FP11 Stín Ve scéně budou stíny za umístěným objektem.

FP12 Obnova Scénu bude možné obnovit, tedy všechny detekované plochy a umístěné objekty bude možné smazat.

5.1.2 Nefunkční požadavky

Nefunkční požadavky popisují vlastnosti, které by měla aplikace mít.

NP1 Podporované verze systému Aplikace bude podporovat zařízení s aktuální a předešlou verzí operačního systému.

5.2 Analýza podobných řešení

V této části bude analyzováno několik aplikací podporujících přidávání virtuálních objektů s hlavním zaměřením na aplikace, které konkrétně podporují přidávání obrazů na zeď. Výjimkou bude aplikace Ikea Place, která umožňuje přidávání nábytku, nikoliv obrazů. Do analýzy byla zařazena z důvodu, že je považována za jednu z nejlépe zpracovaných aplikací týkajících se augmentované reality [36].

Etsy

Etsy je služba, kde mohou běžní lidé nabízet, případně koupit převážně umělecké produkty. Ve třech ze svých kategorií, konkrétně kresby, fotografie a výtisky je umožňují ve své mobilní aplikaci vizualizovat pomocí augmentované reality. Při otevření detailu produktu, patřícího do jedné z těchto kategorií, se v horní části zobrazí ikona označující augmentovanou realitu. Po kliknutí na ni se zobrazí instrukce pro detekci prostoru. Pro tyto instrukce nebylo využito nativní hotové řešení, ale Etsy si ho implementovala sama. Je pouze textové, ale velmi dobře napovídající, co má uživatel dělat. Před rozpoznáním první plochy se zobrazují rozpoznané body. Po rozpoznání plochy se na ní zobrazí čtverec, který se pohybuje podle nasměrování telefonu, tedy je vždy uprostřed obrazovky. Se zobrazením čtverce se zároveň zobrazí instrukce, aby uživatel klikl na místo, kam chce objekt umístit. Toto chování může být pro některé uživatele trochu matoucí, protože by se dalo předpokládat, že když se čtverec na zdi pohybuje společně se zařízením, bude i poloha objektu při jeho umístění shodná

s polohou čtverce. Po umístění objektu je možné měnit jeho velikost běžným gestem, pokud je zvolené dílo dostupné ve více velikostech. Zvolený rozměr se vždy zarovná na nejbližší dostupnou velikost. Seznam všech dostupných velikostí se zároveň zobrazí jako rozbalovací seznam, ze kterého je možné velikost vybrat. Celkově je uživatelský zážitek velmi dobrý až na trochu překvapující umístění objektu a nemožnost vizualizovat své umění v jakékoliv velikosti.

Shutterstock

Shutterstock je jedna z nejznámějších fotobank, tedy služba poskytující zpoplatněné fotografie a obrazy ke komerčnímu použití. V její mobilní aplikaci je možné tyto obrazy vizualizovat v prostoru. Její zpracování je velmi podobné Etsy, tedy v detailu díla je ikona označující augmentovanou realitu, po kliknutí na ni započne detekce doprovázená průvodcem, který si Shutterstock implementoval sám. Při detekci se také zobrazují detekované body. Po rozpoznání první plochy zobrazí instrukce nabádající ke kliknutí na tuto plochu. Již se na zdi nezobrazuje žádný čtverec, pouze je zeď vizualizována jemnou mřížkou, která je relativně špatně viditelná. Kliknutím na plochu se objekt umístí a dále je s ním možné běžnými gesty pohybovat a měnit jeho velikost.

Artsy

Artsy je podobná služba jako Etsy. V detailu každé položky má tlačítko „Zobrazit v místnosti“. Po kliknutí na něj začne detekce s textovým průvodcem. Po rozpoznání první horizontální plochy se na dané ploše zobrazí přímka, která se pohybuje společně se zařízením a uživatel je instruován k jejímu zarovnání s vertikální stěnou. Poté, co provede zarovnání, se na vertikální stěně zobrazí bíle orámovaný poloprůhledný obraz, který se po zdi pohybuje spolu se zařízením tak, že je vždy uprostřed obrazovky. Po kliknutí na tlačítko „Umístit práci“ orámování zmizí, dílo se stane neprůhledným a ukotví se na dané pozici. S objektem není možné dále manipulovat. Lze předpokládat, že nutnost umístění přímky je z důvodu, že tato aplikace byla implementována s první verzí ARKit, kde ještě nebylo dostupné rozpoznání vertikálních ploch.

Paul Reiffer

Paul Reiffer je fotograf, který na svých webových stránkách nabízí své fotografie k prodeji. V detailu každé z těchto fotografií má tlačítko „zobrazit ve vašem prostoru“, které otevře nativní řešení pro vizualizaci trojrozměrných objektů. Tuto funkci podporuje platforma iOS i Android. Toto řešení má výhodu, že je velmi jednoduché na implementaci a nevyžaduje stažení mobilní aplikace. Nevýhodou je, že lze nastavit pouze vizualizovaný objekt a několik parametrů, ale není možné dále toto řešení upravovat.

Ikea place

Ikea place je aplikace určená k vizualizaci jejich produktů u sebe doma. Je označována za jednu z nejlepších na rozšířenou realitu, například Forbes ji zařadil mezi tři nejlepší na trhu [36]. Svým záměrem se liší od předchozích aplikací, které se zaměřují hlavně na vizualizaci obrazů. Ikea place byla zařazena do analýzy kvůli jejímu zpracování, které je nesrovnatelně lepší než mají všechny předchozí aplikace. První hlavní funkcí je naskenování a rozpoznání produktu. K tomu slouží ikona v seznamu produktů, která je velmi obecná a není zpočátku zcela zřejmé k čemu daná funkce slouží. Po kliknutí na ni se otevře kamera a v horní části obrazovky se zobrazí zpráva „Klikněte na předmět, který chcete najít“ velmi dobře vystihující danou funkci. Po kliknutí se zobrazí obdélník, označující hledaný produkt a lupa, která vyhledá a zobrazí nalezené produkty. Druhou hlavní funkcí je vizualizace produktu. Po vybrání některého z produktů se v rohu zobrazí standardní ikona označující augmentovanou realitu. Po kliknutí na ni se otevře kamera a nejprve se zobrazí pokyn pro detekci místnosti pomocí vizualizace pohybujícího se mobilního telefonu nad plochou. Ikea nezvolila nativní hotové řešení, ale udělala si svoje, které je také velmi dobře zpracované. Před rozpoznáním první plochy se zobrazují tzv. „feature points“ - rozpoznané body v prostoru. Po rozpoznání plochy se na ni zobrazí kolečko, označující budoucí umístění produktu a ve spodní části obrazovky tlačítko pro umístění. Toto kolečko je v místě, kam směřuje telefon, tedy vždy ve středu obrazovky. Pokud se s mobilním telefonem pohybuje rychle, kolečko se posouvá s mírným zpožděním, ve výsledku to ale působí velmi plynule. Po kliknutí na tlačítko, se s haptickou odezvou označující úspěšnou akci, umístí produkt na požadované místo. Po kliknutí na objekt se vznese a začne levitovat nad kolečkem pro umístění, které se

znovu zobrazilo. Objekt je možné přesouvat buď přejížděním prstem po obrazovce, nebo přidržením prstu na objektu a pohybem telefonu. Zároveň je zde zobrazená ikona popelnice umožňující objekt odstranit. Celkově je celá zkušenost s aplikací rychlá, intuitivní a plynulá.

Závěr porovnání podobných řešení

V této části budou shrnuty zdařilé funkce analyzovaných aplikací, které by se daly použít i pro aplikaci, která bude výsledkem této práce. Všechny z analyzovaných aplikací, kromě aplikace Paula Reiferra, začínají detekci prostředí vlastním průvodcem. Aplikace Paula Reiferra využívá průvodce poskytovaného v rámci nástroje ARKit, který by se dal považovat za nejlepší spolu s průvodcem v aplikaci Ikea place, který je graficky velmi podobný. V aplikacích Etsy, Shutterstock a Ikea place se před detekcí první plochy v prostoru zobrazují detekované body dobře indikující probíhající proces. Po rozpoznání plochy se v aplikacích Etsy, Artsy a Ikea place zobrazí ve středu obrazovky indikátor pozice na zdi a ve spodní části tlačítko pro potvrzení umístění objektu. Toto řešení by se dalo považovat za velmi intuitivní. Po umístění objektu je s ním možné ve všech aplikacích interagovat běžnými gesty.

6 Návrh

V této kapitole budou popsána rozhodnutí, která byla klíčová před započetí vývoje. Konkrétně bude vybrána platforma, vývojářský a vykreslovací nástroj pro augmentovanou realitu, vývojové prostředí, programovací jazyk a architektura. Dále bude popsána tvorba uživatelského rozhraní a interakce s virtuálním obrazem. V závěrečné části kapitoly bude zvoleno vhodné osvětlení pro virtuální obraz.

6.1 Platforma, vývojářský a vykreslovací nástroj

Nejzásadnějším rozhodnutím před započetím vývoje byla volba mezi ARKit a ARCore, která následně určila i platformu. Vzhledem k výsledkům analýzy platforem z kapitoly 3 a výsledkům praktického porovnání z kapitoly 4, kde téměř ve všech ohledech zvítězil ARKit, hlavně tedy při srovnání vykreslovacích nástrojů, byl ARKit zvolen i jako nástroj pro implementaci aplikace, která bude výsledkem této práce.

ARKit sám o sobě nemá žádný vykreslovací nástroj, slouží pouze k detekci reálného světla a porozumění scéně. Je tedy nutné nějaký nástroj zvolit. Vybírat lze z několika oficiálních možností, kterými jsou Unity, Unreal, SpriteKit, Metal, RealityKit a SceneKit. Unity a Unreal se používají převážně pro tvorbu her, SpriteKit se používá pro vizualizaci dvourozměrných objektů a Metal je nízkoúrovňový nástroj, který je pro účely této práce zbytečně komplikovaný a jeho přidané možnosti by nepřinášely žádnou výhodu. Použití RealityKit je oproti SceneKit o něco jednodušší, ale možnosti jsou omezenější například v dostupných světlech a změny materiálu objektu za běhu programu. Jako vykreslovací nástroj byl tedy vybrán SceneKit.

6.2 Vývojové prostředí

Za vývojové prostředí byl zvolen Xcode [37] vyvíjený společností Apple, čímž je zaručena jeho plná kompatibilita s podporovanými vývojovými nástroji. Alternativou by mohlo být například vývojové prostředí AppCode, které ale v pozadí využívá Xcode [38] pro překlad aplikace, a proto nemůže nabízet znatelně lepší funkcionalitu.

6.3 Programovací jazyk

Nativní iOS aplikace je v současnosti možné vyvíjet pomocí programovacích jazyků Objective-C a Swift [39]. Objective-C byl několik let jedinou volbou a Swift se k němu přidal až v roce 2014. Swift [40] je programovací jazyk vyvíjený společností Apple pro vývoj aplikací na všech jeho platformách, kterými jsou iOS, iPadOS, tvOS, watchOS a macOS. Dle [41] má Swift oproti Objective-C výhody například ve větší bezpečnosti, snazší udržovatelnosti, automatické správy alokované paměti a stručností psaného kódu oproti Objective-C. Z těchto důvodů byl programovací jazyk Swift vybrán.

6.4 Architektura

Architektura byla zvolena MVVM (Model-View-ViewModel). Jedná se o návrhový vzor velmi rozšířený v oblasti vývoje mobilních aplikací. Alternativou může být návrhový vzor MVC (Model-view-controller), který doporučuje společnost Apple. Jednou z největších výhod MVVM oproti MVC je oddělení logiky od uživatelského rozhraní, což umožní jednodušší testovatelnost a způsobí větší přehlednost.

Jednotlivé části MVVM architektury:

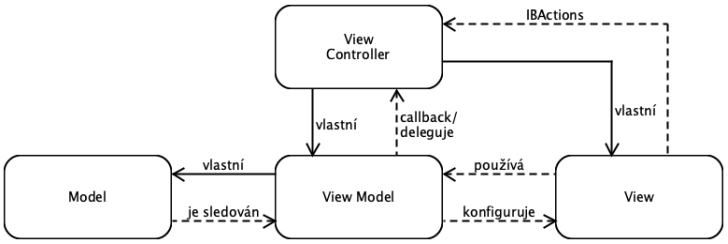
Model Reprezentuje data aplikace. Většinou je tvořen strukturami nebo jednoduchými třídami.

View Reprezentuje rozvržení prvků a vzhled obrazovky. Zachycuje uživatelovy interakce s prvky a posílá je komponentě View Controller, která se stará o jejich přeposílání do View Modelu.

View Model Uchovává si aktuální stav aplikace a na jeho základě filtruje data z modelu pro jednotlivé prvky view.

View Controller Slouží jako propojení View Modelu s View.

Komunikace mezi popsanými částmi je znázorněna na obr. 17



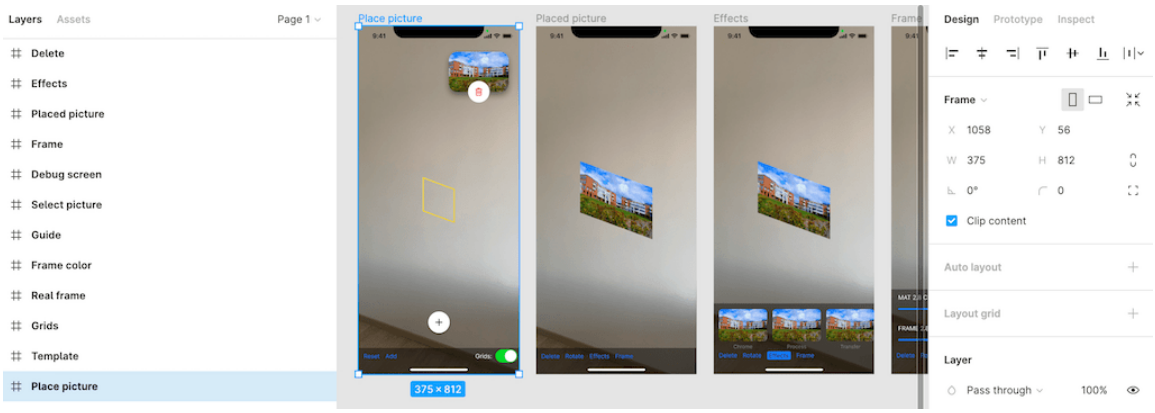
Obrázek 17: Návrhový vzor MVVM [42]

6.5 Reaktivní programování

Jedná se o paradigma používané na mnoha platformách, a i proto se stává oblíbeným tématem poslední doby v oblasti vývoje mobilních aplikací [43]. *„Reaktivní programování je programování s asynchronními datovými toky. Datový tok je sekvence po sobě jdoucích událostí seřazených v čase. Může produkovat tři věci: hodnotu, chybu a „completed“ signál“* [44]. Hodnota může být například datová struktura, proměnná, vlastnost, uživatelský vstup, atd. Toky je možné kombinovat a filtrovat tak, aby byly výstupem události, které nás zajímají. Jeden tok může být použit jako vstup pro druhý. Tento tok vytváří takzvaný Publisher a reaguje na něj takzvaný Observer, který na jeho základě může například měnit uživatelské rozhraní. Nejznámější knihovny umožňující použití reaktivního programování při vývoji pro iOS jsou nativní Combine a knihovny třetích stran RxSwift a Reactive Swift. Všechna řešení nabízejí funkce pro vytváření, kombinování, filtrování a další práci s datovými toky. Tento přístup při vhodném použití dělá zdrojový kód přehlednější a proto bude použit i v této práci.

6.6 Uživatelské rozhraní

Po zvolení požadavků nic nebrání vytvoření grafického návrhu. Byť bude aplikace relativně malá, co se počtu obrazovek a grafických komponent týče, vyplatí se vytvořit si grafický návrh v některém z dostupných editorů, a to z důvodu urychlení vývoje jak samotných grafických komponent a jejich celků, tak i z hlediska ujasnění si výsledné podoby aplikace a tím i odladění průchodu aplikací a jednotlivých stavů aplikace. Ke grafickému návrhu byl použit program Figma [45], který je pro nekomerční účely zdarma, byť je to profesionální nástroj, který se v praxi běžně používá pro grafické návrhy aplikací. Návrh aplikace v tomto programu je znázorněn na obrázku 18.



Obrázek 18: Grafický návrh v programu Figma [vlastní tvorba]

6.7 Detekce prostředí

Jedna z prvních interakcí uživatele s AR je detekce prostředí. V ideálních podmínkách, za které by se dalo považovat dobře osvětlené prostředí s kontrastní strukturou, je tento proces téměř instantní. Ne vždy se ale uživatel nachází v ideálních podmínkách, a proto je ho třeba informovat o tom, proč interakce zatím není možná a jaké kroky je třeba provést pro rozpoznání prostředí. Všechny analyzované aplikace implementovaly vlastního průvodce detekcí prostředí a nevyužily nativní řešení, které nabízí ARKit od verze 3.0, tedy od iOS 13. Žádná z analyzovaných aplikací neměla průvodce zpracovaného znatelně lépe, než je tomu u nativního řešení a vzhledem k nefunkčnímu požadavku na podporu iOS 13.1 a novější, byla pro tuto práci zvolena implementace nativního průvodce.

6.8 Umístění objektu

Pro umístění objektu do prostředí se nabízí několik možností. První z nich je instantní umístění po rozpoznání podporované plochy, například po výběru přidávaného objektu nebo okamžitě po rozpoznání první odpovídající plochy. Toto řešení je většinou uživatelem neočekávané a následuje přemístění na požadovanou lokaci. Druhá možnost je kliknutí na jednu z naskenovaných ploch. V tomto případě je nutné vhodným způsobem vizualizovat všechny dostupné plochy a edukovat uživatele o akci, kterou musí provést, konkrétně tedy kliknutí na plochu. Třetí možnost je zobrazit indikátor umístění uprostřed obrazovky a tlačítko indikující přidání objektu s vhodným jménem, nebo ikonou. Tento indikátor mění polohu v

reálném světě v závislosti na pohybu zařízení - po nasměrování na podporovanou plochu zobrazí indikátor na dané ploše, pokud plocha v daném bodě nebyla rozpoznána, indikátor změní svou podobu a informuje tím uživatele o tom, že v tomto místě není možné objekt přidat. Tato volba by se dala považovat za nejintuitivnější a bude tedy implementována v této práci. Po umístění objektu bude dále možná změna jeho pozice, velikosti, otočení a případně bude možné objekt ze scény odebrat. Změna pozice a velikosti bude možná běžnými gesty, na které jsou uživatelé chytrých mobilních zařízení zvyklí, tedy změna pozice bude možná potažením a změna velikosti roztažením nebo přiblížením prstů. Rotace bude možná pomocí tlačítka vzhledem k faktu, že vizualizovaný obraz může být k pozorovateli natočen a gesto by nemuselo být tak intuitivní jako je tomu u dvourozměrných obrazů. Smazání obrazu bude taktéž možné pomocí tlačítka.

6.9 Grafické úpravy umístěného obrazu

Po umístění a nastavení velikosti obrazu bude možné modifikovat jeho vzhled a orámování. Vzhled bude možné modifikovat aplikováním některého ze standardních filtrů, který zahrnuje například sépiový, monochromatický nebo chromový odstín. Filtry bude možné měnit tak, aby byla změna viditelná v reálném čase a bude možné vrátit původní vzhled, tedy odebrat filtr.

Orámování obrazu bude možné provést dvěma způsoby, které je možné modifikovat - změnou samotného rámu a pasparty. Pasparta je jednobarevná plocha mezi obrazem a rámem, která je stejně tenká jako samotný obraz. Barvu pasparty bude možné modifikovat. Orámování obrazu bude možné výběrem jednoho z dostupných rámů ze seznamu, který bude obsahovat náhledový obrázek pro každý z rámů. Po vybrání rámu bude možné měnit jeho barvu.

6.10 Osvětlení

Pro docílení co možná nejlepšího osvětlení objektů a vržení stínů je důležité znát všechny možnosti vykreslovacího nástroje a následně z nich zvolit vhodnou kombinaci. V případě nástroje SceneKit to je šest druhů zdrojů osvětlení [46] a jeden druh pro sběr informací o světle. Všechny druhy jsou popsány v následujícím seznamu. SceneKit je v tomto ohledu

napřed oproti nástroji RealityKit, který nabízí pouze tři druhy osvětlení, kterými jsou *point* - v případě SceneKit označované jako *omni*, *directional* a *spot* [47].

- **ambient:** Ambientní osvětlení by se dalo popsat jako všudypřítomné. Pokud bude ve scéně pouze ambientní zdroj světla, budou objekty osvětleny všude stejně, nezávisle na vzájemné pozici světla a objektu. Pozice zdroje ambientního osvětlení tedy není důležitá.

- **directional:** Směrové světlo, které by se dalo v reálném světě z pohledu pozorovatele na Zemi přirovnat ke Slunci. Pozice stejně jako u ambientního zdroje osvětlení není důležitá a intenzita nezávisí na vzdálenosti od zdroje. Nejdůležitějším parametrem je tedy jeho směr.

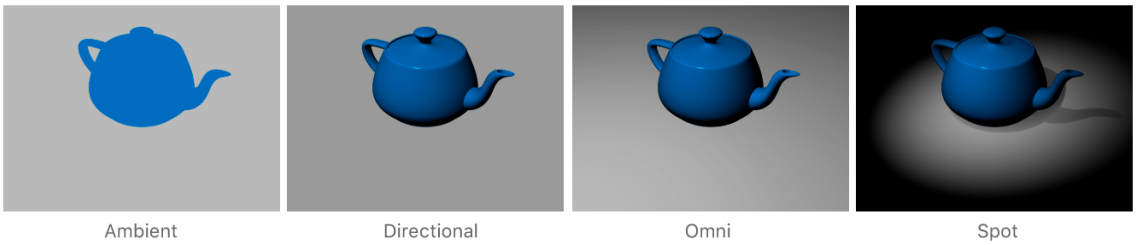
- **omni:** Omni je zkratka pro omnidirectional, v překladu všesměrový, také označované jako point light. Jedná se o zdroj světla, který by se dal přirovnat k žárovce. Světlo tedy vyzařuje všemi směry a jeho intenzita klesá se vzdáleností od zdroje.

- **spot:** Spotlight, v překladu svítilna má vlastnosti stejné jako reálná svítilna. Tedy světelný zdroj kuželovitého tvaru vychází z určitého bodu, jeho konus je pevně definovaný zvoleným úhlem a objekty vzdálenější od zdroje světla budou slaběji osvětlené, než ty u zdroje světla.

- **IES:** Světelný zdroj jehož vlastnosti jako je například tvar, směr a intenzita jsou definované pomocí souboru s příponou IES. Tyto soubory se typicky dají stáhnout na stránkách výrobce osvětlení.

- **area:** Tento zdroj světla byl přidaný až s iOS 13. Jedná se o zdroj světla porovnatelný se spot osvětlením, ale na rozdíl od spot typu nevychází světlo z jednoho bodu, ale vyzařuje ho kruhovitá nebo čtvercová plocha. V reálném světě by se dalo přirovnat k zářivkovému svítidlu.

- **probe:** Probe není zdroj světla, ale s osvětlením úzce souvisí. Jedná se o objekt, který sbírá informace o osvětlení a vlastnostech prostředí v místě jeho umístění a objekty ve scéně pak těchto informací využívají při výpočtu barvy a intenzity svého osvětlení.



Obrázek 19: Základní zdroje světla v nástroji SceneKit [46]

Dokumentace věnovaná světelným zdrojům nástroje SceneKit uvádí u parametru *castsShadows* [46], že pouze světelné zdroje *spot* a *directional* podporují vržení stínu. Jelikož je vržení stínu jeden z funkčních požadavků, musí být vybráno z těchto typů. V tomto případě byl vybrán směrový zdroj světla, protože není nutné vytvářet efekt svítilny. Směrové světlo bude navíc doplněno ambientním zdrojem světla jehož intenzita bude upravována podle odhadu osvětlení poskytnutým nástrojem ARKit popsaným v sekci 4.2.2.

7 Implementace

V této kapitole budou popsány nejzásadnější kroky při implementaci aplikace. Kapitola začíná tématem nastavení scény, což je prvotní krok při implementaci aplikace s augmentovanou realitou a pokračuje detekcí prostředí, přidáváním objektů do scény, ovládáním scény, zakrytím objektů, úpravami umístěného obrazu a je zakončena popisem zobrazování rozpoznaných ploch.

7.1 Nastavení scény

Nastavení scény je nezbytné v každém projektu používajícím SceneKit nebo Sceneform nástroje. V této části bude popsáno, jaké nastavení bylo použito v aplikacích, které jsou výsledkem této práce.

V první řadě je třeba specifikovat několik pojmů:

- SceneKit - framework kombinující vysoce výkonný render engine s API pro import, manipulaci a renderování 3D objektů. Na rozdíl od nízkoúrovňových API jako je Metal a OpenGL, které vyžadují implementaci detailního algoritmu, který vykresluje scénu, SceneKit vyžaduje pouze popis obsahu scény a akce, nebo animace, které chce vývojář dosáhnout.

- ARSession - objekt koordinující hlavní procesy, které ARKit provádí za vývojáře pro vytvoření AR scény. Tyto procesy zahrnují čtení dat z pohybových senzorů zařízení, ovládání zabudované kamery a provádění analýzy zachycených obrazů kamerou. ARSession syntetizuje všechny tyto výsledky k vytvoření propojení mezi reálným světem a tím virtuálním, kde vývojář modeluje AR obsah.

- ARSCNView - view, které integruje ARSession renderování do SceneKitu, respektive umožňuje zobrazení AR obsahu ze SceneKitu.

- ARWorldTrackingConfiguration - konfigurační objekt, který je předáván ARSCNView při spuštění jeho ARSession.

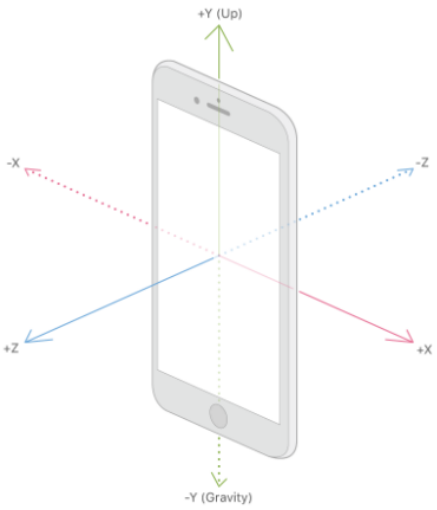
V ukázce zdrojového kódu 1 je zobrazena základní konfigurace ARSCNView. Třída obsahuje objekt ARWorldTrackingConfiguration a referenci na ARSCNView. Funkce *setupSce-*

neView pak obsahuje samotnou konfiguraci, kde se nastavuje jaké plochy se mají detekovat a následně se s danou konfigurací spouští ARSession na objektu ARSCNView.

Je důležité zmínit, že v ARWorldTrackingConfiguration se dá nastavit souřadnicový systém pomocí parametru *worldAlignment*, který je pro tuto práci zachován v základním nastavení, tj. *gravity* - tento souřadnicový systém je vyobrazen na obrázku 20.

```
private let config = ARWorldTrackingConfiguration()
private weak var sceneView: ARSCNView!
...
private func setupSceneView() {
    config.planeDetection = [.vertical, .horizontal]
    sceneView.session.run(config)
    sceneView.delegate = self
}
```

Výpis 1: Nastavení ARSCNView



Obrázek 20: ARKit souřadnicový systém [48]

7.2 Detekce prostředí

Jak bylo popsáno v 6.7, před detekcí první plochy bude uživatel instruován pomocí nativního řešení, kterým je takzvaný „ARCoachingOverlayView“ - info grafika s popisem, co má uživatel dělat. Konfigurace je pro vývojáře velmi jednoduchá - minimální konfigurace,

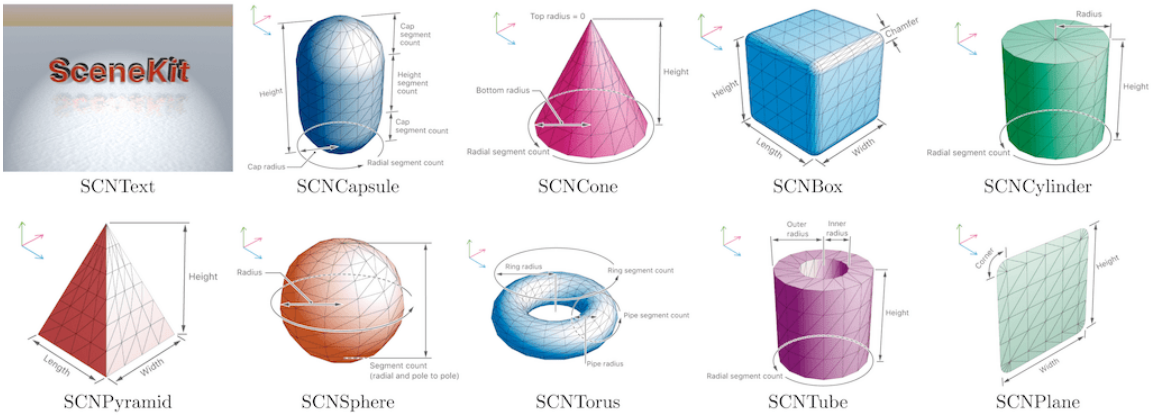
vyobrazená v ukázce 2, je na tři řádky, kde první je vytvoření instance view, následně se nastaví cíl a nakonec instance ARKit session.

```
let coachingView = ARCoachingOverlayView()
coachingView.goal = .verticalPlane
coachingView.session = sceneView.session
```

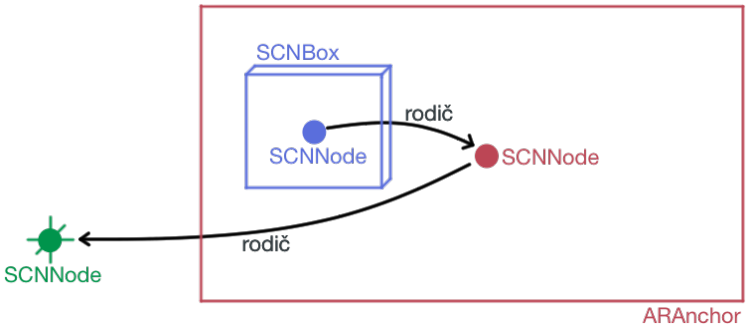
Výpis 2: Nastavení ARCoachingOverlayView

7.3 Objekty ve scéně

Pro porozumění následujícím odstavcům je třeba vysvětlit, na jakém principu funguje uspořádání objektů v ARKit scéně. Toto uspořádání se označuje jako *graf scény*. Základní stavební kámen všech objektů je uzel *SCNNode*. Jedná se o element grafu scény reprezentující pozici a transformaci v 3D souřadnicovém systému, kterému lze přiřadit geometrii, světla, kamery a jiný zobrazitelný obsah. V ARKit scéně může být pouze jeden kořenový uzel, který má umístění a transformaci shodnou se zařízením použitým pro detekci při započetí ARSession. Pro zmíněnou geometrii nabízí SceneKit jedenáct předdefinovaných objektů *SCNGeometry*. Deset z nich je na obrázku 21, jedenáctý je pak SCNShape, založený na dvou-dimenzionální křivce, volitelně škálovatelný na 3D objekt. Aby bylo možné jednoduše umísťovat do scény nové uzly, ARWorldTracking kontinuálně vyhledává a aktualizuje kotvy označované jako *ARAnchor*. Kotvou může být horizontální nebo vertikální plocha, zadefinovaný objekt, obrázek nebo tvář. Každá z těchto kotev má také svůj uzel, který má jako rodiče kořenový uzel. Pro přidání například kvádru do prostoru je potřeba vytvořit nový uzel SCNNode, kterému se nastaví geometrie SCNGeometry, konkrétně SCNBox, tento uzel se přidá jako potomek jednoho z uzlů ve scéně - tím může být kořenový uzel, uzel některé z nalezených kotev nebo nějaký jiný již přidaný uzel. V neposlední řadě je třeba nastavit pozici přidaného objektu, kterou lze nastavit relativně k rodičovskému uzlu nebo kořenovému uzlu. Nákres takové scény je vidět na obr. 22. Kromě předdefinovaných materiálů umožňuje SceneKit importovat i podporované formáty 3D souborů popsané v sekci 3.8.



Obrázek 21: SNCMaterials [49]



Obrázek 22: SceneKit scéna [vlastní tvorba]

7.4 Ovládání scény

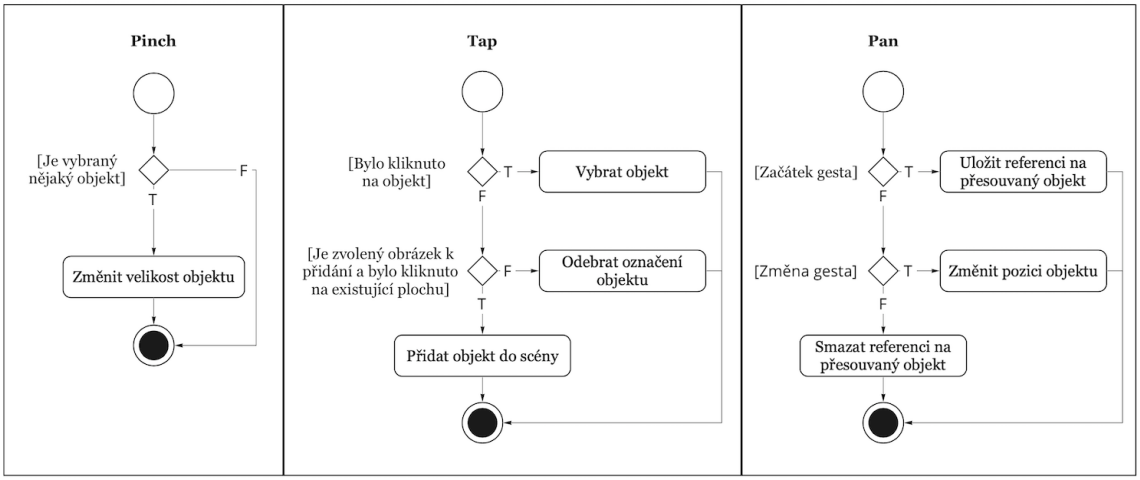
Vzhledem k tomu, že s objekty by mělo být možné interagovat, je nezbytné implementovat gesta pro ovládání scény. V této části bude popsána implementace těchto gest.

7.4.1 Gesta pro ovládání scény

Pro ovládání scény je nutná implementace gest. Konkrétně v této práci je implementována podpora následujících gest:

- *Pinch* pro změnu velikosti objektu.
- *Tap* vybrání a zrušení výběru objektu.
- *Pan* pro změnu pozice objektu.

Podrobnější nákres funkce jednotlivých gest je na obrázku 23.



Obrázek 23: Podporovaná gesta [vlastní tvorba]

Při implementaci podpory gest je nejzásadnější správné rozpoznání objektu, se kterým má gesto interagovat. Toto rozpoznání je potřeba provést při rozpoznání *tap* gesta pro korektní změnu výběru objektu a při začátku *pan* gesta pro rozpoznání objektu, se kterým se má pohybovat.

ARKit nabízí dvě možnosti, kterými se dá tohoto rozpoznání docílit - tzv. *Hit-Testing* a *Raycasting*. V ukázce zdrojového kódu 3 jsou zobrazeny obě tyto možnosti. Hit-Testing bude od iOS 14 považován za zastaralý a doporučuje se použití Raycastingu. Největší výhodou Raycastingu je možnost výběru orientace ploch, tj. zda se mají testovat horizontální plochy, vertikální plochy anebo obě možnosti. Další výhodou je možnost tzv. *trackedRaycast*, který opakovaně provádí Raycast query a upozorňuje při změně povrchů v reálném prostředí.

```
let tapLocation = recognizer.location(in: sceneView)
let hitTestingResults = sceneView.hitTest(
    tapLocation, types: [.existingPlaneUsingGeometry])
guard let query = sceneView.raycastQuery(
    from: tapLocation,
    allowing: .existingPlaneGeometry,
    alignment: .any) else { return nil }
let raycastingResults = sceneView.session.raycast(query)
```

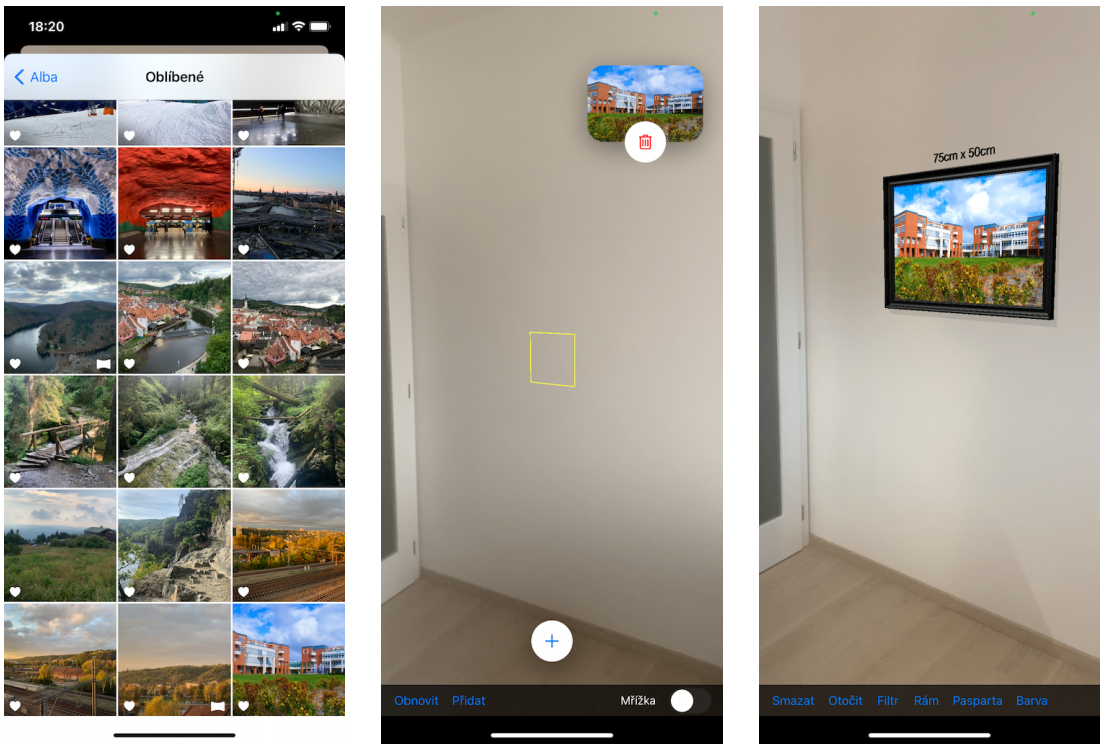
Výpis 3: Nalezení objektu po detekci gesta v ARKitu

7.4.2 Umístění objektu

Jak bylo popsáno v 6.8, pro umístění objektu se bude uprostřed obrazovky zobrazovat indikátor a ve spodní části obrazovky bude tlačítko pro umístění objektu.

Na obrázku 24 je vidět implementace umístění obrazu na zeď. Prvním krokem je vybrání obrazu z galerie. Následně se zobrazí pomocný obrazec určující pozici umístění. Tento obrazec je vždy uprostřed obrazovky a mění svou polohu v reálném prostředí po změně polohy zařízení. Také je na obrazovce vidět náhled obrazu, který bude umístěn na danou pozici a tlačítko pro umístění. Obrazec pro umístění a tlačítko mění svůj vzhled v závislosti na dostupnosti naskenované plochy ve středu obrazovky. Pokud zařízení směřuje do prostoru, kde v jeho středu pohledu není žádná vertikální zeď, nebo není žádná rozpoznaná, čtverec nebude transformovaný a bude obsahovat mezery. Tím indikuje, že by uživatel měl pokračovat ve skenování nebo změnit směr zařízení. Pokud zeď v daném směru je rozpoznaná a uživatel klikne na tlačítko pro umístění objektu, přidá se obraz na dané místo.

Při umístění objektu je využit Raycasting popsaný v 7.4, který vrátí uživatelem vybranou plochu a přesnou pozici, kde byla plocha vybrána. Na tuto pozici je umístěn nový objekt.



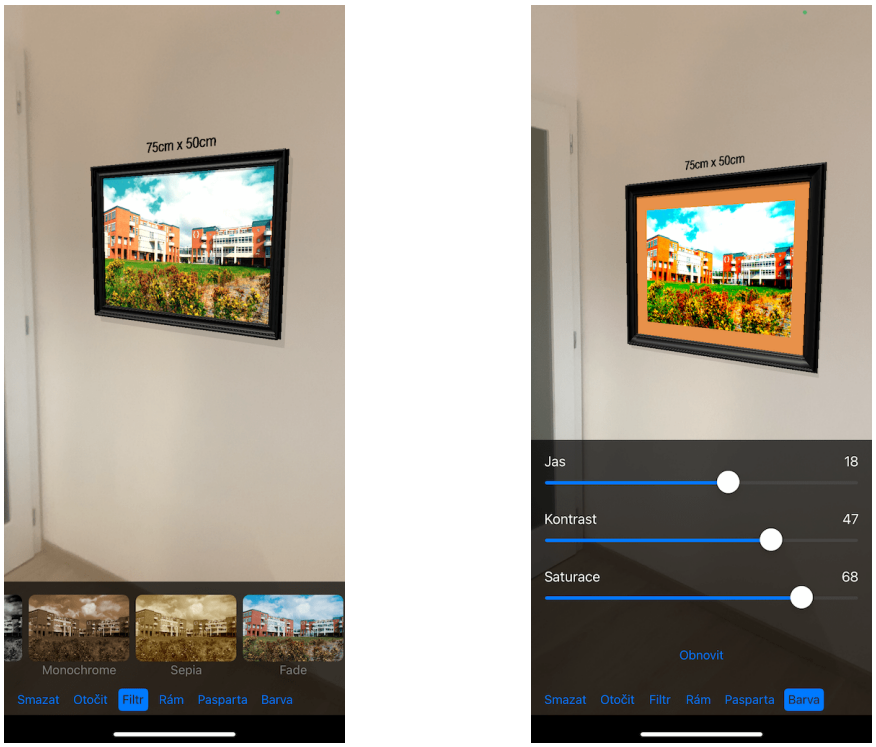
Obrázek 24: Výběr a umístění obrazu [vlastní tvorba]

7.5 Úpravy umístěného obrazu

V aplikaci bylo implementováno několik možností úpravy umístěného obrazu. Nabídka úprav se zobrazí po vybrání obrazu kliknutím na něho. První z možností je odstranění ze scény. Po kliknutí na tuto volbu se zobrazí dialog a po jeho potvrzení je objekt odebrán. Další možností je rotace. Po výběru této volby je objekt rotován ve směru hodinových ručiček po devadesáti stupních.

Třetí volbou je změna filtru (obr. 25). Apple nabízí několik filtrů jako podtřídy třídy CIFilter. V této práci bylo implementováno 10 filtrů - například mono, tonal, noir a další. Při zvolení této možnosti se zobrazí seznam s náhledy aplikovaných filtrů na původním obrazu. Kliknutím na konkrétní náhled se filtr aplikuje na obraz. Dále je také možné upravovat jas, kontrast a saturaci obrazu. Úpravy těchto vlastností bylo také docíleno pomocí třídy CIFilter.

Uzavření detailu konkrétního nastavení je možné pomocí kliknutí na právě vybranou volbu. Tím se pouze skryje detail a obraz zůstane označen. Další možností je kliknutí kamkoli do prostoru mimo daný obraz. Tím se zavře celé nastavení. Po znovu vybrání některého z objektů se otevře stejné nastavení jako bylo naposledy vybrané.

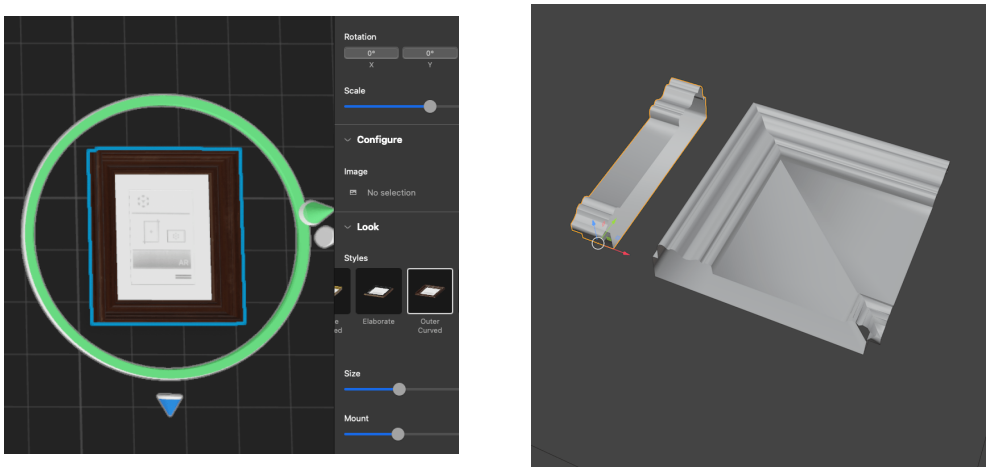


Obrázek 25: Změna filtru, jasu, kontrastu a saturace [vlastní tvorba]

7.6 Rám a pasparta

Jednou z možností modifikace obrazu je úprava pasparty a rámu. Pasparta byla implementována jako objekt SCNBox, který je o něco nižší, než objekt pro obraz. Šířku pasparty je možné měnit pomocí posuvníku a změna se v reálném čase aplikuje na vizualizovaný obraz. Kromě šířky pasparty je také možné měnit její barvu.

Změnu rámu lze provést výběrem z nabídky. Každý rám je definovaný dvěma 3D objekty. První objekt definuje rohy a druhý strany. Bylo by možné definovat rám pouze jedním objektem, ale při změně velikosti obrazu by se měnila i šířka rámu, což je nežádoucí, protože by rám přestával vypadat reálně při velké odchylce od jeho zadefinované velikosti. Definice 3D objektů rámů byly použity z nástroje Reality Composer (obr. 26). Tyto definice by se daly použít pomocí vykreslovacího nástroje RealityKit, ale nebylo by možné za běhu aplikace měnit obrazy, jejich velikost a pasparty. Reality Composer v kombinaci s RealityKit je tedy ideální volba pro scénu se statickými prvky, což ale není případ aplikace, která je výsledkem této práce. Z toho důvodu byly definice rámu upraveny v programu Blender (obr. 26) jehož použití je zdarma a i přesto se jedná o profesionální nástroj na úpravu 3D objektů. Pro podporu usdz souborů, které jsou exportovány z nástroje Reality Composer a které podporuje i SceneKit, bylo nezbytné nainstalovat plugin. Každý z rámů byl oříznut na roh, byla mu nastavena správná rotace a také pivot - bod, který určuje pozici objektu nezávisle na jeho geometrii, tedy pivot může ležet i mimo geometrii objektu.



Obrázek 26: Příprava 3D objektů rámu. Vlevo před exportem z nástroje Reality Composer, vpravo po úpravě v nástroji Blender [vlastní tvorba]

7.7 Osvětlení

První složkou osvětlení, která byla dle návrhu implementována, bylo ambientní osvětlení. Zdroj ambientního osvětlení je přidán do scény ihned po jejím vytvoření. V konfiguraci scény byl povolen odhad osvětlení, a tedy s každou obnovou obrazu, která probíhá s frekvencí 60 Hz, jsou provedeny odhady intenzity osvětlení a jeho barvy v reálném prostředí. Tyto odhady jsou aplikovány na ambientní zdroj osvětlení a virtuální objekty, tedy lépe zapadají svým jasem do prostředí.

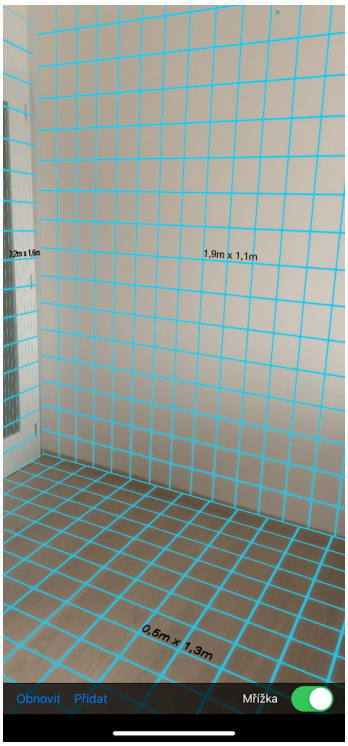
Dalším zdrojem osvětlení je směrové světlo. Jeho směr je nastavený tak, aby dopadal pod úhlem 45 stupňů vůči ose x a pod úhlem 30 stupňů vůči ose y a tím vytvořil stín pod obrazem. Aby se tento stín měl na čem promítnout, má každý přidaný virtuální objekt pod sebou průhlednou plochu. Pro dosažení jeho reálného dojmu bylo potřeba upravit několik jeho parametrů. Konkrétně to byla jeho barva, která je nastavena na černou se 45% průhledností a parametr *shadowMapSize* určující velikost obrazu, který SceneKit renderuje při vytváření stínů na hodnotu 1024 x 1024 pixelů.

Obrázek 27: Stín [vlastní tvorba]

7.8 Zobrazení rozpoznaných ploch

Detekce jednobarevných ploch, kterými zdi pro umístění obrazu zpravidla jsou, může být komplikovaná. Pro vizualizaci procesu detekce uživateli a usnadnění ladění aplikace tedy bylo implementováno zobrazení rozpoznaných ploch. SceneKit nemá vlastní implementaci zobrazení rozpoznaných ploch, jako tomu je například u Sceneform a je tedy nutné si vytvořit vlastní implementaci. Toho bylo docíleno pomocí ARSCNView delegáta, který informuje o každé změně rozpoznaných ploch. Konkrétně obsahuje dvě funkce, první informuje o rozpoznání nové kotvy (`renderer(_:didAdd:for:)`) a druhá informuje o aktualizaci některé z již přidaných kotev (`renderer(\textunderscore,didUpdate:for:)`).

Zobrazení bylo vizualizováno pomocí čtvercové mřížky (obr. 28), na které se zobrazuje i rozměr rozpoznané plochy. Zobrazení a skrytí ploch je možné měnit pomocí přepínače v hlavním menu.



Obrázek 28: Zobrazení rozpoznaných ploch [vlastní tvorba]

8 Shrnutí výsledků

V rámci této diplomové práce byly analyzovány a porovnány nativní nástroje hlavních platforem pro vývoj mobilních aplikací s augmentovanou realitou, nástroje Apple ARKit a Google ARCore. Porovnány byly jak teoreticky na základě uváděných parametrů a dostupných funkcí, tak i na praktických příkladech porovnáním rozpoznávání ploch a odhadu osvětlení.

Z teoretického hlediska nabízejí oba nástroje podobné funkce. Rozdíly jsou pouze v tom, že ARKit nabízí podporu detekce pohybu těla, umožňuje také detekci několika obličejů či 2D obrazů zároveň a má kvalitně zpracovanou dokumentaci. ARCore má výhodu oproti ARKit v možnosti spuštění aplikací na simulátoru a umožňuje tvorbu hloubkové mapy i bez Time-of-Flight senzoru.

ARKit a ARCore jsou nástroje na detekci a odhad vlastností prostředí a jejich běžná funkce není možná bez vykreslovacích nástrojů. Zde má Apple zřetelný náskok s nástroji SceneKit, SpriteKit a RealityKit. Dále také nabízí Reality Composer na jednoduchou tvorbu scén pro interaktivní augmentovanou realitu a nástroje pro konverzi a úpravu 3D objektů Reality Converter a USDZ Tools. Google nabízí pouze vykreslovací nástroj Sceneform, který již není dále vyvíjen. Apple s nástroji SceneKit a RealityKit podporuje 12 formátů 3D souborů a další 3 formáty je možné pomocí nástrojů Reality Converter a USDZ Tools zkonvertovat do podporovaného formátu USDZ. Sceneform bez omezení podporuje pouze 3 formáty. Apple má aktuálně více podporovaných zařízení, což by se mělo dle předpovědi ARtillery Intelligence [34] v nejbližších letech změnit a do vedení by se měl dostat ARCore.

Nástroje byly porovnány i na reálných příkladech, konkrétně v rozpoznávání prostředí a odhadu osvětlení. Zde zvítězil ARKit s rychlejší a přesnější detekcí. ARKit téměř přesně detekoval pozici plochy a s malou odchylkou určil její ohraničující obdélník a tedy i obsah. ARCore dobře odhadoval pozici, ale znatelně se lišil v obsahu plochy oproti skutečnosti. Lepšího odhadu intenzity osvětlení dosahoval také ARKit oproti ARCore, s kterým byly objekty přesvícené vzhledem k okolnímu prostředí. Nevýhodou ARKit je ovšem jeho umístění stínů, které jsou vždy pod objekty jako by zdroj světla byl nad nimi. ARCore odhaduje pozici zdroje osvětlení, a tedy má snahu vrhat stíny odpovídajícím směrem.

V rámci této práce byla také vyvinuta aplikace pro interaktivní vizualizaci obrazu v prostoru s využitím nástroje ARKit v kombinaci s vykreslovacím nástrojem SceneKit. V této aplikaci je možné vybrat obraz z galerie, umístit ho na zeď, měnit jeho pozici, velikost, otočení, vzhled, rám a paspartu. Pro umístění obrazu byl na základě analýzy konkurenčních aplikací volen nejvhodnější možný postup tak, aby byl proces rychlý a intuitivní a v případě problémů s detekcí a umístěním obrazu, je možné zobrazit mřížku vizualizující rozpoznané plochy. Po umístění na zeď je dále možné měnit jeho pozici, velikost a otočení běžnými gesty. Změnu vzhledu lze provést pomocí aplikace filtrů, změnou jasu, kontrastu a saturace. Rám je možné volit mezi několika 3D objekty s podobou reálného rámu, dále měnit jeho barvu stejně jako u pasparty obrazu, u které lze měnit i její šířku.

9 Závěry a doporučení

Cílem této práce bylo porovnat mobilní platformy pro augmentovanou realitu a implementovat aplikaci na umístění 2D objektů do 3D scény. Oba vytyčené cíle byly splněny.

Při porovnání nativních vývojářských nástrojů mobilních platforem Apple ARKit a Google ARCore, bylo dospěno k závěru, že oba nástroje jsou z hlediska funkčnosti velmi podobné. Velký rozdíl je ale ve vykreslovacích a podpůrných nástrojích, kde Apple nabízí více možností. ARKit a ARCore byly také porovnány na praktických příkladech. Zde ARKit vykazoval lepší výsledky jak v detekci horizontální plochy, tak v odhadu osvětlení.

V rámci této práce byla také vyvinuta aplikace umožňující umístění 2D objektů do 3D scény. Konkrétně byla implementována aplikace umožňující vybrat obraz z galerie a umístit jej na vertikální plochu. Umístěný obraz je možné dále upravovat změnou jeho pozice, velikosti, rotace, filtru, rámu, pasparty, jasu, kontrastu a saturace. Jedním z nejzásadnějších bodů implementace byla vhodná volba postupu umístění obrazu tak, aby byl tento proces uživatelsky přívětivý. Výběru vhodného postupu bylo docíleno analýzou aplikací s podobnou funkcionalitou. Vzniklá aplikace nalezne uplatnění především v umělecké sféře, umožní totiž autorům, prodejcům obrazů a jejich zákazníkům pověsit dílo na konkrétní zeď.

Na tuto práci je možné navázat rozšířením úprav vzhledu vizualizovaného obrazu. Například by bylo možné přidat úpravu barvy změnou její teploty, odstínu, sytosti, živosti, dále pak úpravu ostrosti, prokreslení, redukce šumu, vinětace a další možnosti standardních grafických editorů. Další možností navázání na tuto práci je rozšíření praktického porovnání obou nástrojů. Porovnání by mohlo být provedeno na více příkladech, například detekcí různých typů kotev, v různých světelných podmínkách a na vícero zařízeních.

Seznam použité literatury

1. MEALY, Paul. *Virtual and Augmented Reality For Dummies*. Hoboken, New Jersey: John Wiley a Sons, Inc, 2018. ISBN 978-1119481348.
2. APPLE INC. Augmented Reality. *Apple Developer portal* [online] [cit. 2020-01-19]. Dostupné z: `https://developer.apple.com/augmented-reality`.
3. WEINBAUM, Stanley. *Pygmalion's Spectacles*. 1935.
4. CHRISTOPHER ZIMMERMANN. The History of VR - In VR. *Magnolia* [online]. 2017 [cit. 2021-04-06]. Dostupné z: `https://blog.magnolia-cms.com/blog/history-of-vr-in-vr.html`.
5. SUTHERLAND, Ivan E. The Ultimate Display. In: *Proceedings of the IFIP Congress*. 1965, s. 506–508.
6. SCHMALSTIEG, Dieter; HÖLLERER, Tobias. *Augmented Reality - Principles and Practice*. United States: Addison-Wesley Professional, 2016. ISBN 0321883578.
7. CAUDELL, Thomas P. Introduction to augmented and virtual reality. In: DAS, Hari (ed.). *Telemanipulator and Telepresence Technologies*. SPIE, 1995, sv. 2351, s. 272–281. Dostupné z DOI: `10.1117/12.197320`.
8. MILGRAM, Paul; KISHINO, Fumio. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Information Systems*. 1994, roč. vol. E77-D, no. 12, s. 1321–1329.
9. AZUMA, Ronald T. A Survey of Augmented Reality. *Presence: Teleoper. Virtual Environ.* 1997, roč. 6, č. 4, s. 355–385. ISSN 1054-7460. Dostupné z DOI: `10.1162/pres.1997.6.4.355`.
10. WAGNER, Daniel; SCHMALSTIEG, Dieter. First steps towards handheld augmented reality. In: 2005, s. 127–135. ISBN 0-7695-2034-0. Dostupné z DOI: `10.1109/ISWC.2003.1241402`.
11. SALMAN SH. Google Unveils New Augmented Reality Platform 'ARCore'; What About Tango? *MEDIANAMA* [online]. 2017 [cit. 2021-04-06]. Dostupné z: `https://www.medianama.com/2017/08/223-google-augmented-reality-platform/`.

12. APPLE INC. ARKit 1. *Apple Developer portal* [online] [cit. 2020-01-19]. Dostupné z: https://developer.apple.com/videos/play/wwdc2017/602.

13. GOOGLE INC. ARCore: Augmented reality at Android scale. *Google blog* [online] [cit. 2017-08-29]. Dostupné z: https://blog.google/products/google-ar-vr/arcore-augmented-reality-android-scale/.

14. JOHN FINN. Why Did Google Glass Fail? *Screen Rant* [online]. 2019 [cit. 2021-04-17]. Dostupné z: https://screenrant.com/google-glass-fail-why-enterprise-explorer-edition/.

15. FURHT, Borivoje. *Handbook of augmented reality*. New York: Springer, 2011. ISBN 1461400635.

16. ŠKODA AUTO A.S. ŠKODA AR [online] [cit. 2020-05-21]. Dostupné z: https://www.skoda-auto.com/world/ar-app.

17. APPLE INC. Apple Augmented Reality [online] [cit. 2020-01-19]. Dostupné z: https://www.apple.com/augmented-reality/.

18. APPLE INC. Understanding ARKit Tracking and Detection. *Apple Developer portal* [online] [cit. 2021-03-15]. Dostupné z: https://developer.apple.com/videos/play/wwdc2018/610/.

19. ALZA. Time of Flight senzor [online] [cit. 2021-04-16]. Dostupné z: https://www.alza.cz/time-of-flight-senzor.

20. APPLE INC. Apple unveils new iPad Pro with breakthrough LiDAR Scanner and brings trackpad support to iPadOS. *Apple Newsroom* [online] [cit. 2021-03-12]. Dostupné z: https://www.apple.com/newsroom/2020/03/apple-unveils-new-ipad-pro-with-lidar-scanner-and-trackpad-support-in-ipados/.

21. GOOGLE INC. ARCore supported devices. *ARCore dokumentace* [online] [cit. 2021-04-17]. Dostupné z: https://developers.google.com/ar/devices.

22. STEFAN PFEIFER. What ARKit 3.5 and the new iPad Pro bring to the table and how you can export it's LIDAR scans. *Stefan Pfeifer Blog* [online] [cit. 2021-03-08]. Dostupné z: https://medium.com/zeitraumgruppe/what-arkit-3-5-and-the-new-ipad-pro-bring-to-the-table-d4bf25e5dd87.

23. APPLE INC. ARKit 1.5. *Apple Developer portal* [online] [cit. 2020-01-19]. Dostupné z: https://developer.apple.com/news/?id=01242018b.

24. APPLE INC. ARKit 2. *Apple Developer portal* [online] [cit. 2020-01-19]. Dostupné z: https://developer.apple.com/videos/play/wwdc2018/602.

25. APPLE INC. ARKit 3. *Apple Developer portal* [online] [cit. 2020-01-19]. Dostupné z: https://developer.apple.com/videos/play/wwdc2019/604.

26. APPLE INC. ARKit 4. *Apple Developer portal* [online] [cit. 2020-01-19]. Dostupné z: https://developer.apple.com/videos/play/wwdc2020/10611.

27. APPLE INC. Unleash Your Creativity with Augmented Reality session. *Apple Developer portal* [online] [cit. 2021-04-10]. Dostupné z: https://developer-sessions.apple.com/s/1JFgq9ja.

28. APPLE INC. AR Creation Tools [online] [cit. 2020-05-21]. Dostupné z: https://developer.apple.com/augmented-reality/tools/.

29. GOOGLE INC. arcore-android-sdk. *GitHub* [online] [cit. 2021-04-10]. Dostupné z: https://github.com/google-ar/arcore-android-sdk/releases.

30. GOOGLE INC. ARCore Developer Preview 2. *Google blog* [online] [cit. 2017-12-15]. Dostupné z: https://blog.google/products/google-ar-vr/arcore-developer-preview-2/.

31. GOOGLE INC. Announcing ARCore 1.0 and new updates to Google Lens. *Google blog* [online] [cit. 2018-02-23]. Dostupné z: https://blog.google/products/google-ar-vr/announcing-arcore-10-and-new-updates-google-lens/.

32. GOOGLE INC. Experience augmented reality together with new updates to ARCore. *Google blog* [online] [cit. 2018-05-08]. Dostupné z: https://blog.google/products/google-ar-vr/experience-augmented-reality-together-new-updates-arcore.

33. GOOGLE INC. Updates to ARCore Help You Build More Interactive and Realistic AR Experiences. *Google blog* [online] [cit. 2019-05-07]. Dostupné z: https://developers.googleblog.com/2019/05/ARCore-IO19.html.

34. Mobile AR Global Penetration. *ARtillery Intelligence* [online] [cit. 2021-04-02]. Dostupné z: https://arinsider.co/about/.

35. GOOGLE INC. Using ARCore to light models in a scene. *ARCore dokumentace* [online] [cit. 2021-04-10]. Dostupné z: https://developers.google.com/ar/develop/java/light-estimation.

36. RECCHIA, Chad. Augmented Reality Marketing: Three Companies That Have Done It Best. *Forbes* [online] [cit. 2021-02-25]. Dostupné z: https://www.forbes.com/sites/forbesagencycouncil/2018/03/01/augmented-reality-marketing-three-companies-that-have-done-it-best/?sh=b0fada92dc8c.

37. APPLE INC. Xcode. *Apple Developer portal* [online] [cit. 2020-12-27]. Dostupné z: https://developer.apple.com/xcode/.

38. SMITH, David. AppCode and Xcode. *JetBrains* [online] [cit. 2020-12-27]. Dostupné z: https://www.jetbrains.com/help/objc/appcode-and-xcode.html.

39. SOLT, Paul. What coding language is commonly used for native iOS app development? *Livewirelabs* [online] [cit. 2020-12-27]. Dostupné z: https://www.livewirelabs.co/native-ios-app-development/.

40. APPLE INC. Swift. *Apple Developer portal* [online] [cit. 2020-12-27]. Dostupné z: https://developer.apple.com/swift/.

41. Swift vs. Objective-C: 10 reasons the future favors Swift. *Infoworld* [online] [cit. 2020-12-27]. Dostupné z: https://www.infoworld.com/article/2920333/swift-vs-objective-c-10-reasons-the-future-favors-swift.html.

42. Design Patterns by Tutorials: MVVM [online] [cit. 2020-12-27]. Dostupné z: https://www.raywenderlich.com/%5C%2034-design-patterns-by-tutorials-mvvm.

43. PILLET, Florent; BONTOGNALI, Junior; TODOROV, Marin; GARDNER, Scott. *RxSwift: Reactive Programming with Swift*. 2. vydání. Razeware LLC, 2017. ISBN 194287846X.

44. STALTZ, André. The introduction to Reactive Programming you have been missing. *Blog* [online] [cit. 2020-12-27]. Dostupné z: https://gist.github.com/staltz/868e7e9bc2a7b8c1f754.

45. Figma [online] [cit. 2020-12-27]. Dostupné z: https://www.figma.com.

46. APPLE INC. SCNLight.LightType. *Apple Developer portal* [online] [cit. 2021-04-12]. Dostupné z: `https://developer.apple.com/documentation/scenekit/scnlight/lighttype`.

47. APPLE INC. RealityKit. *Apple Developer portal* [online] [cit. 2021-04-12]. Dostupné z: `https://developer.apple.com/documentation/realitykit`.

48. APPLE INC. ARConfiguration.WorldAlignment.gravity. *Apple Developer portal* [online] [cit. 2020-12-27]. Dostupné z: `https://developer.apple.com/documentation/arkit/arconfiguration/worldalignment/gravity`.

49. GOOGLE INC. Built-in Geometry Types. *ARCore dokumentace* [online] [cit. 2021-04-28]. Dostupné z: `https://developer.apple.com/documentation/scenekit/built-in_geometry_types`.

Seznam zkratek

AR Augmentovaná Realita

MVVM Model-View-ViewModel architektura

SDK Software Development Kit

ToF Time-of-Flight senzor

WWDC Worldwide Developer Conference

Přílohy

A Obsah přiložených souborů

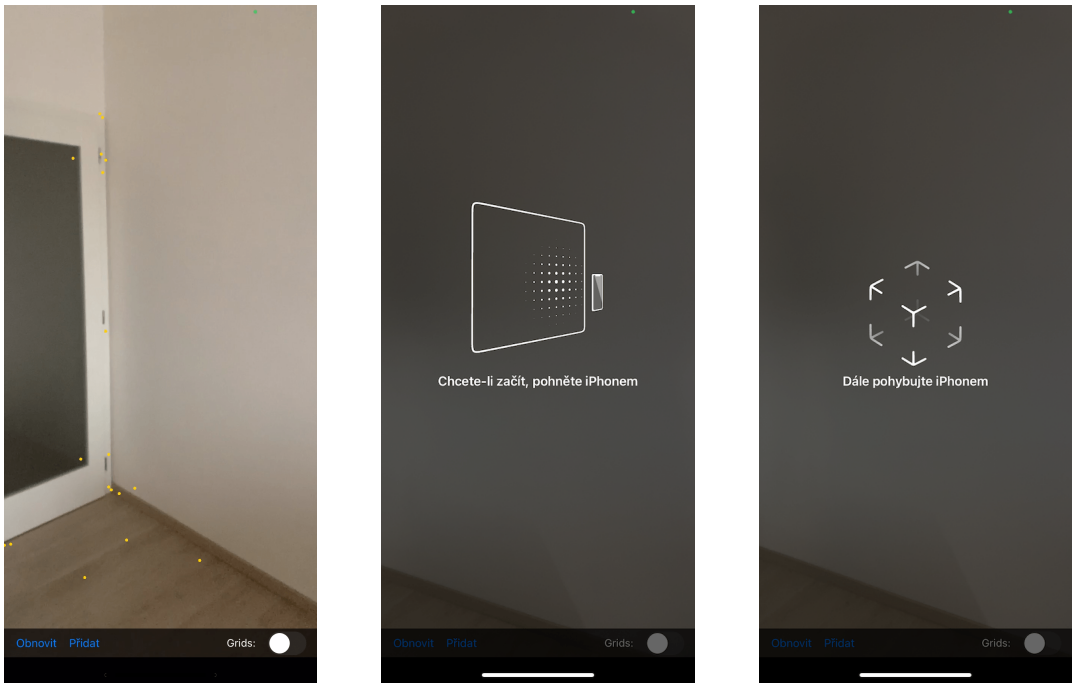
- `readme.txt` ..... stručný popis obsahu přiložených souborů
- `src` ..... zdrojové kódy
  - `impl` ..... zdrojové kódy implementace
    - `ARKit-app` ..... hlavní ARKit aplikace
    - `git-links.txt` ..... odkazy na GIT repozitáře
    - `frames.blend` ..... rámy v Blender 3D formátu
    - `readme.txt` ..... instalační příručka
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `thesis.pdf` ..... text práce ve formátu PDF

B Odkazy na GIT repozitáře

Jednotlivé aplikace, které jsou výsledkem této práce, jsou dostupné na následujících odkazech:

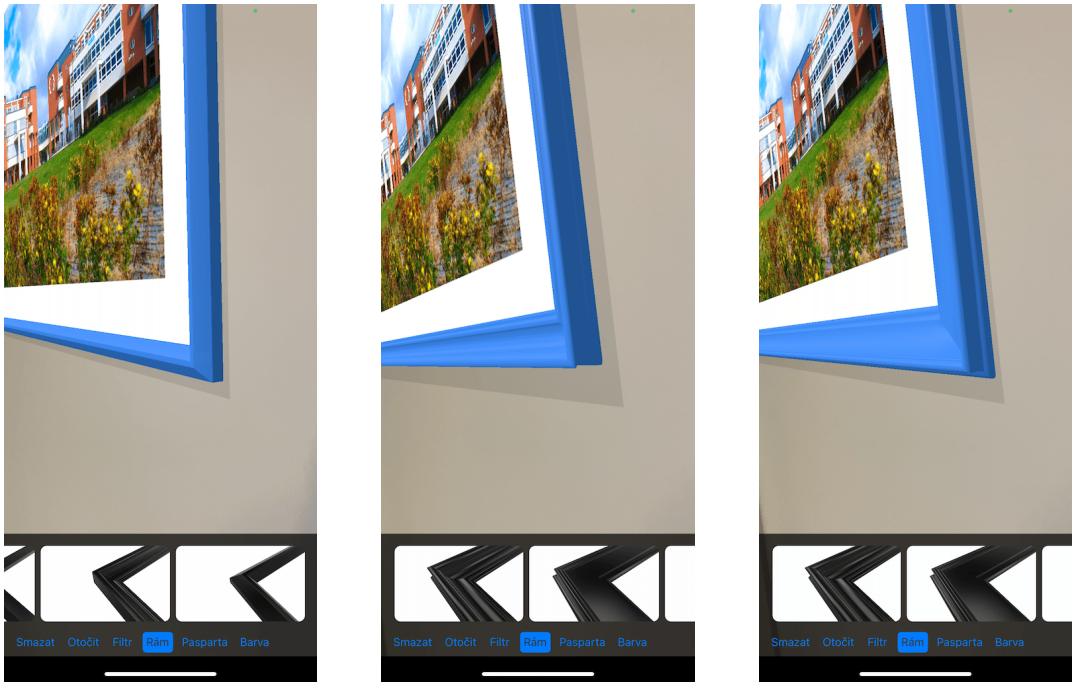
- **Hlavní ARKit aplikace:**
  https://gitlab.com/Cerny/arkit-app
- **ARKit aplikace pro porovnání odhadu osvětlení:**
  https://gitlab.com/Cerny/arkit-light-estimation-app
- **ARCore aplikace pro porovnání odhadu osvětlení:**
  https://gitlab.com/Cerny/lightning-arcore

C Snímky obrazovky ARKit aplikace



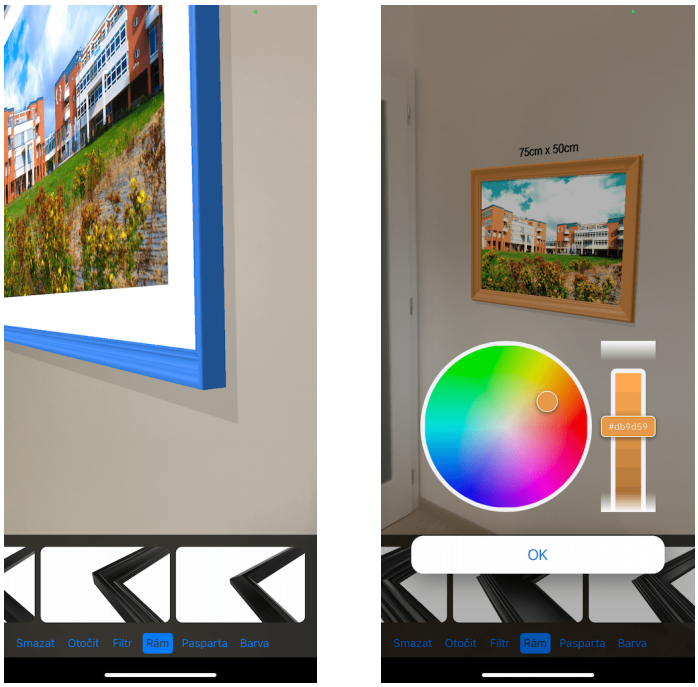
Obrázek 29: Detekce prostředí [vlastní tvorba]

Obrázek 30: Úprava pasparty [vlastní tvorba]

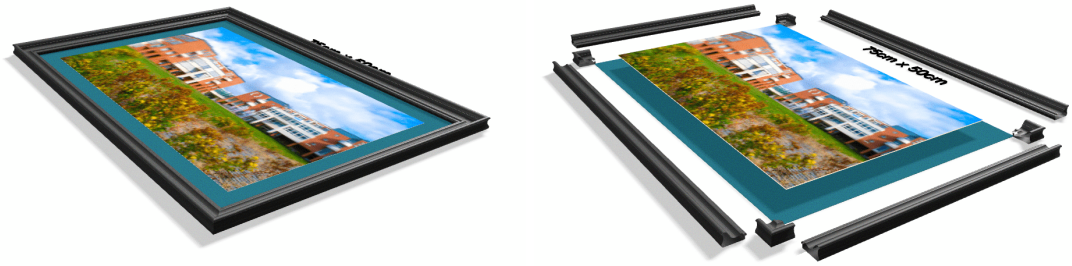


Obrázek 31: Rámy [vlastní tvorba]

Obrázek 32: Rámy [vlastní tvorba]



Obrázek 33: Rám a úprava jeho obarvení [vlastní tvorba]



Obrázek 34: Komponenty 3D modelu [vlastní tvorba]

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2019/2020

Studijní program: Aplikovaná informatika
Forma studia: Kombinovaná
Obor/kombinace: Aplikovaná informatika (ai2-k)

Podklad pro zadání DIPLOMOVÉ práce studenta

Jméno a příjmení: **Michal Černý**
Osobní číslo: **I1900536**
Adresa: Dukelská třída 632, Nový Bydžov, 50401 Nový Bydžov, Česká republika

Téma práce: Interaktivní vizualizace pro mobilní zařízení
Téma práce anglicky: Interactive visualization for mobile devices

Vedoucí práce: Ing. Karel Mls, Ph.D.
Katedra informačních technologií

Zásady pro vypracování:

Cíl: Provést analýzu a porovnání platforem pro rozšířenou realitu a navrhnout a implementovat mobilní aplikaci na umístění 2D objektů do 3D scény.
Osnova:
- Úvod
- Cíl a metodika práce
- Analýza hlavních platforem AR
- Návrh a Implementace
- Závěr, zhodnocení

Seznam doporučené literatury:

AZUMA, Ronald T. A survey of augmented reality. *Presence: Teleoperators & Virtual Environments*, 1997, 6.4: 355-385. FURHT, Borko (ed.). *Handbook of augmented reality*. Springer Science & Business Media, 2011. SCHMALSTIEG, Dieter; HOLLERER, Tobias. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016.

Podpis studenta: Datum:

Podpis vedoucího práce: Datum:

© IS/STAG, Portál – Podklad kvalifikační práce , cernymi3, 30. dubna 2021 08:53