



Zadání bakalářské práce

Název:	Mobilní aplikace pro kreslení s využitím rozšířené reality
Student:	Adela Mišicáková
Vedoucí:	Ing. Tomáš Nováček
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Počítačová grafika
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Rozšířená realita se využívá čím dál více nejenom v průmyslu, ale i v každodenním životě. Nejčastější platformou, kde mohou lidé rozšířenou realitu využívat, jsou mobilní telefony. Jedna z možných aplikací je nástroj na přidávání virtuální vrstvy s obrázky do reálného světa, kterou by si potom mohl uživatel obkreslit, a ulehčit si tak malování obrazů.

Cíl práce:

- 1) Analyzujte možnosti vývoje aplikací využívajících rozšířenou realitu pro operační systém iOS.
- 2) Analyzujte požadavky cílové skupiny.
- 3) Analyzujte alespoň tři podobné aplikace.
- 4) Navrhněte architekturu aplikace.
- 5) Aplikaci naimplementujte.
- 6) Proveďte uživatelské testování.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Mobilní aplikace pro kreslení s využitím rozšířené reality

Adela Mišicáková

Katedra softwarového inženýrství
Vedoucí práce: Ing. Tomáš Nováček

1. května 2022

Poděkování

Ráda bych věnovala své poděkování vedoucímu práce Ing. Tomáši Nováčkovi za odborné vedení, trpělivost a ochotu při psaní této práce. Mé díky patří také rodině a všem, kteří mi při psaní pomáhali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (buť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 1. května 2022

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Adela Mišicáková. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Mišicáková, Adela. *Mobilní aplikace pro kreslení s využitím rozšířené reality*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací mobilní aplikace pro zařízení s operačním systémem iOS k usnadnění obkreslování obrázků pomocí rozšířené reality. K práci s rozšířenou realitou využívá aplikace framework ARKit a k implementaci uživatelského rozhraní framework SwiftUI. Výsledkem je mobilní aplikace umožňující uživateli vybírat obrázky (z vlastní či poskytnuté galerie) a následně jej promítat na danou vertikální nebo horizontální plochu. Obrázek lze pomocí filtrů upravovat a je možné měnit jeho velikost, natočení, polohu či průhlednost. V závěru jsou uvedeny možnosti rozšíření výsledné aplikace.

Klíčová slova mobilní aplikace, iOS, rozšířená realita, ARKit, Xcode, Swift, SwiftUI

Abstract

This bachelor thesis deals with the design and implementation of a mobile application for iOS devices to facilitate drawing of images using augmented reality. The application uses ARKit framework to work with augmented reality and SwiftUI framework to implement the user interface. The result is a mobile application that allows users to select images (from their own or provided gallery) and projects them on a given vertical or horizontal surface. The image can be edited using filters. Also, it is possible to change its size, rotation, position or transparency. Options for extending the resulting application are listed in the conclusion.

Keywords mobile application, iOS, augmented reality, ARKit, Xcode, Swift, SwiftUI

Obsah

Úvod	1
1 Rozšířená realita	3
1.1 Využití rozšířené reality	3
1.2 Nevýhody rozšířené reality	8
1.3 Vývoj AR aplikací	8
1.3.1 ARKit	10
1.3.2 ARCore	12
1.3.3 ARKit vs. ARCore	13
2 Vývoj aplikací pro zařízení s operačním systémem iOS	15
2.1 Zásady iOS	15
2.2 Architektonické vzory	16
2.3 UIKit, SwiftUI	18
2.4 Xcode	19
2.5 Swift, Objective-C	19
3 Analýza	23
3.1 Dotazníkové šetření	23
3.2 Cílová skupina	24
3.3 Funkcionalita aplikace	24
3.4 Existující aplikace	24
3.4.1 InkHunter	25
3.4.2 Etsy	25
3.4.3 SketchAR	26
4 Návrh	27
4.1 Případy užití	27
4.2 Uživatelské rozhraní	27
4.2.1 Hlavní menu	27

4.2.2	Show All obrazovka	28
4.2.3	Edit obrazovka	28
4.2.4	Draw obrazovka	29
5	Implementace	31
5.1	Uživatelské rozhraní	31
5.2	Výběr obrázku	33
5.3	Editace obrázku	34
5.4	AR scéna	34
5.4.1	Konfigurace a inicializace	35
5.4.2	Nápověda při detekci ploch	35
5.5	Vytvoření obrázku	36
5.6	Interakce s obrázkem	37
6	Testování	39
6.1	Testovací scénáře	39
6.2	Výsledky	40
6.3	Možná vylepšení	41
	Závěr	43
	Bibliografie	45
	A Seznam použitých zkratek	51
	B Obsah příloženého CD	53

Seznam obrázků

1.1	Rozpoznávání rostlin pomocí aplikace Pl@ntNet	4
1.2	Simulátor virtuální reality k výuce laparoskopické operace	5
1.3	Využití systému MEDARPA při brachyterapii	6
1.4	Aplikace IKEA Place, která do skutečného světa vkládá nábytek	6
1.5	Využití rozšířené reality v architektuře	7
1.6	Využití rozšířené reality v turismu k popisu budov	7
1.7	Ukázka použití filtru na obličej	8
1.8	3D model mračna bodů	9
1.9	3D síť vytvořená z 3D modelu mračna bodů	9
1.10	Ukázka špatné okluze na sochách Maji a Meryt	10
1.11	6DOF podporuje kromě rotace zařízení i pohyby nahoru, dolů, doprava, doleva, vpřed, vzad	11
1.12	Prst, který je k pozorovateli blíže nežli auto, má při pohledu z levého oka jinou pozici než při pohledu z oka pravého	12
1.13	Srovnání ARKit 2.0 (iPhone X) a ARCore 1.9.1 (Pixel 3)	13
2.1	Architektonický vzor MVC	17
2.2	Architektonický vzor MVVM	18
3.1	Výsledky dotazníkového šetření	23
3.2	Ukázka použití aplikace InkHunter	25
4.1	Diagram případů užití	28
4.2	Návrh grafického rozhraní	29
5.1	Mockup výsledné aplikace ve světlém a tmavém režimu	32
5.2	Ukázka obrazovky pro úpravu obrázků pomocí filtrů	35
5.3	Ukázka použití mobilní aplikace	37

6.1	Výsledek testovacího scénáře, který zkoumá použitelnost aplikace k obkreslování obrázků. Modrý obrys hvězdy byl nakreslen uživatelem, šedý obrys sloužil jako předloha k obkreslení.	40
-----	--	----

Úvod

Rozšířená realita (AR) se v posledních letech stala hojně diskutovaným pojmem. Dovoluje propojovat skutečný svět s uměle vytvořeným, a to bez nutnosti nákupu drahých zařízení, jako tomu je u reality virtuální. K objevování rozšířené reality nám stačí pouze mobilní telefon.

Mimo herní účely se AR využívá například v medicíně nebo školství, kde dopomáhá studentům nejrozličnějších oborů k procvičení svých znalostí či samotné výuce. Jistě se již každý z nás vyskytl v situaci, kdy měl nakreslit nějaký obrázek, ale nevěděl, kde začít. Mohlo se jednat o nakreslení beránka křídou na chodník nebo krychle do sešitu geometrie. V takových situacích byla zřejmě nutná improvizace a výsledek nemusel být příliš uspokojivý. Aplikace, jež bude výsledkem této bakalářské práce, má za cíl dopomoci nejen ve zmíněných situacích, a to právě pomocí rozšířené reality.

Téma jsem si vybrala z důvodu velké praktičnosti, jelikož jsem se sama v několika podobných situacích ocitla. Hledala jsem tedy aplikaci, která by mi jednoduše umožnila vybrat si libovolný obrázek z galerie a augmentovat jej na danou plochu. Narazila jsem na jedno řešení, které ale neodpovídá mé představě dostatečně. Augmentace vlastního obrázku v této mobilní aplikaci je měsíčně zpoplatněna částkou, která je dle mého názoru příliš vysoká. Mimo jiné mi aplikace přišla nepřehledná a proces k dosažení daného výsledku se zdál příliš zdlouhavý. Celkově se zabývá spíše výukou kresby jako takové. Krok za krokem vysvětluje uživateli, jak například proporčně rozvrhnout postavu člověka. Taktéž je řešena pomocí počítačového vidění, strojového učení a neuronových sítí, což značně ovlivňuje výši zpoplatnění aplikace.

Cílem této bakalářské práce je tedy implementace mobilní aplikace pro iOS zařízení, která umožňuje jednoduché obkreslování obrázků dle uživatelského výběru. Důraz je kladen především na návrh uživatelského rozhraní a jednoduchý průchod aplikací. Nevyužívá neuronových sítí k perfektní augmentaci, ale za účelem snížení nákladů je vytvořena pomocí knihoven zpřístupněných společnostmi Apple beze ztráty funkčnosti.

Úvod

V teoretické části nejdříve popisují vlastnosti a využití rozšířené reality. Analyzují možnosti vývoje aplikací pro operační systém iOS využívajících rozšířenou realitu. Dále analyzují již existující aplikace a pomocí dotazníkového šetření zjišťují požadavky cílové skupiny.

V praktické části této práce navrhuje architekturu aplikace a zmiňuje jednotlivé kroky implementace. Nakonec popisují postupy a výsledky provedení testování uživatelského rozhraní.

Rozšířená realita

Ve virtuální realitě se člověk ponoří do zcela uměle vytvořeného trojrozměrného prostředí. Tělesně se vyskytuje ve světě skutečném, avšak vědomím se nachází ve světě virtuálním. Technologie rozšířené reality (dále AR) nám umožňuje do skutečného světa přidávat vrstvu s virtuálními objekty v reálném čase. Klíčové vlastnosti rozšířené reality jsou tedy následující [1]:

- je interaktivní v reálném čase,
- využívá reálné 3D prostředí,
- kombinuje skutečné s virtuálním.

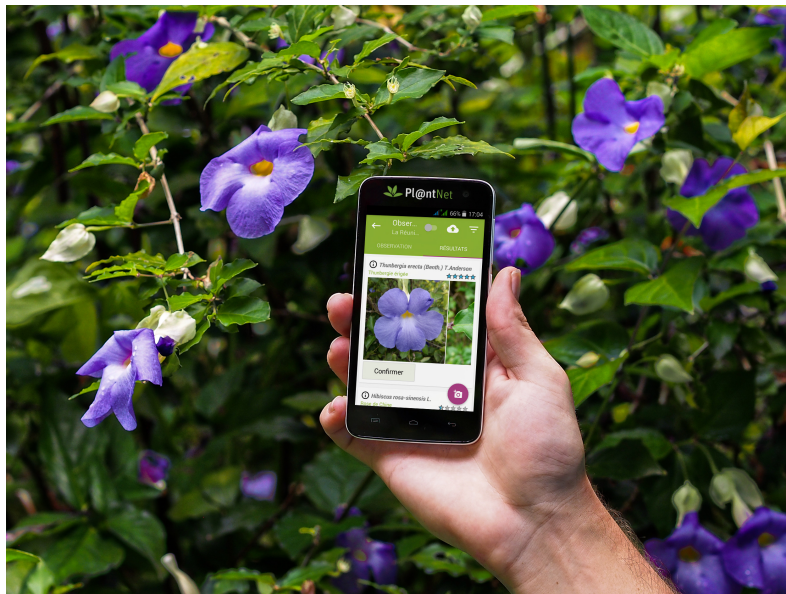
1.1 Využití rozšířené reality

AR zlepšuje interakci se skutečným světem, pomáhá provádět nejrůznější úkoly a zviditelňuje uživatelem nedetekovatelné informace. Mezi hlavní výhody rozšířené reality můžeme řadit především její rozmanité využití.

Vzdělávání

Poskytování informací o reálném světě s využitím AR je rychlejší, snadnější a zajímavější než vyhledávání na internetu či knihách. Studenti, kteří využívají AR ve výuce, pochopí danou látku mnohem lépe než-li studenti, kteří k výuce používají videa, knihy či jiný výukový software [2]. Pomocí AR můžeme rozpoznávat rostliny (viz obrázek 1.1), měřit velikosti předmětů či interagovat s objekty, se kterými by to v reálném světě nebylo možné (například s Věstonickou venuší, obrazem Mony Lisy, ...).

Hojné využití má i ve vzdělávání mediků. Mimo aplikace sloužící k výuce anatomie lidského těla, existuje dokonce i kurz laparoskopické operace na univerzitě Nippon Medical School v Tokiu (viz obrázek 1.2), ve kterém studenti



Obrázek 1.1: Rozpoznávání rostlin pomocí aplikace Pl@ntNet. Převzato z [4].

získávají základní dovednosti nejdříve ve virtuální realitě a následně na simulátoru s rozšířenou realitou. [3]

Medicína

AR se taktéž využívá mimo výuku mediků i při reálných operacích – k zobrazování vnitřní struktury operovaných objektů či k poskytnutí jiných anatomických informací. Na obrázku 1.3 můžeme vidět systém MEDARPA, jehož využití nalezneme například v brachyterapii. Základem je průhledný displej, který slouží k navigaci nástroje vysílajícího záření do bezprostřední blízkosti nádoru. [6]

Reklama

Rozšířená realita může být taktéž použita k reklamě zboží. Apple Inc. využívá AR k umístění nových produktů do prostředí potenciálního kupce a umocňuje v něm touhu si zařízení koupit. Na podobném principu funguje i aplikace IKEA Place, ve které uživatel vkládá pomocí rozšířené reality nábytek do své místnosti a to dokonce až s 98% přesností (viz obrázek 1.4). [7]



Obrázek 1.2: Simulátor virtuální reality k výuce laparoskopické operace. Převzato z [5].

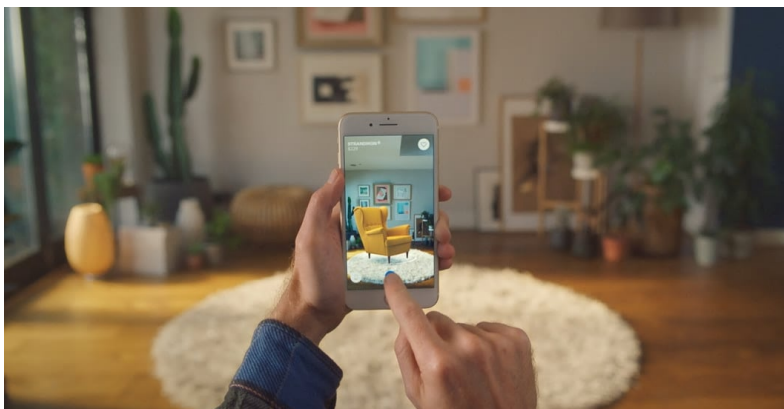
Architektura

Rozšířenou realitu v oboru architektury je možno využít v několika směrech. Například k prezentaci budov zákazníkům architektonických kanceláří (viz obrázek 1.5), kteří mohou nahlížet do vytvořených projektů. AR umožňuje například i interagovat s navrženými budovami (otevírat dveře, zapínat světla, měnit jejich velikost), nebo je dokonce vkládat do reálného prostředí, ve kterém by měly v budoucnu stát.

1. ROZŠÍŘENÁ REALITA



Obrázek 1.3: Využití systému MEDARPA při brachyterapii. Převzato z [6].



Obrázek 1.4: Aplikace IKEA Place, která do skutečného světa vkládá nábytek. Převzato z [8].



Obrázek 1.5: Využití rozšířené reality v architektuře. Převzato z [9].



Obrázek 1.6: Využití rozšířené reality v turismu k popisu budov. Převzato z [10].

Turismus

Turismus je na pomezí mnoha oborů jako například umění, architektura, edukace či archeologie. AR dopomáhá procitnout exponátům v muzeích, popisovat historické budovy či navigovat po okolí nevšedním způsobem (viz obrázek 1.6).



Obrázek 1.7: Ukázka použití filtru na obličej. Převzato z [13].

1.2 Nevýhody rozšířené reality

Zdánlivě největší nevýhodou rozšířené reality je její vliv na jedince a jeho vnímání skutečné reality. Hry obecně jsou známé svou návykovostí a jelikož technologie AR dokáže udržet pozornost uživatele i v rámci hodin, může na cestě k závislosti dopomoci. Taktéž schovává uživateli některé varovné signály skutečného světa, a ten je poté náchylnější k nepozornostem, což může vyústit až v dopravní nehody či smrt. [11]

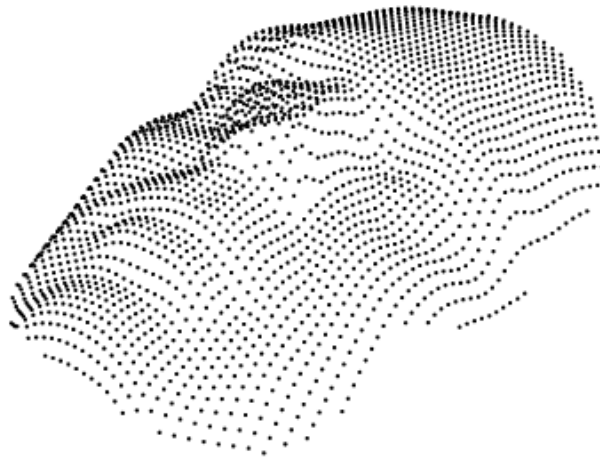
Úprava vzhledu pomocí filtrů v reálném čase se stala v posledních letech velmi populární na sociálních sítích a umožňuje uživateli kompletně změnit svou vizáž (viz obrázek 1.7). Někteří jedinci se trýzně zabývají myšlenkou, že nejsou dokonalí, což mnohokrát vede nejen k plastickým operacím, ale i anorexii. Tento fenomén byl již dokonce pojmenován a nese název „Snapchat dysmorphia“. [12]

Mezi dalšími nevýhodami rozšířené reality je jistě problém řešení okluze, což je schopnost skrývání virtuálních objektů za objekty v reálném světě. Touto problematikou se blíže zabývám v následující sekci.

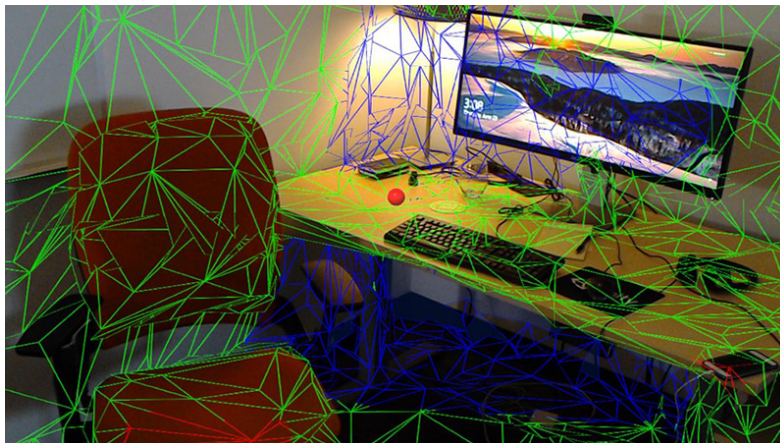
1.3 Vývoj AR aplikací

Abychom skutečný svět mohli rozšířit (augmentovat), musíme jej nejdříve nasnímat. Společnosti, které se zabývají tématem rozšířené reality, se k tomuto problému staví různě. Například Apple Inc. se zaměřuje spíše na vylepšování hardware (jmenovitě například LiDAR senzor, který vysílá paprsky laseru do okolního prostředí a zařízení poté počítá dobu, za kterou se vrátí zpět) [14]. Android se naopak snaží dosáhnout co nejlepšího snímání struktury 3D světa pouze pomocí informací z jedné kamery a jejího pohybu, popřípadě jejich kombinací společně s daty z hloubkového senzoru. [15]

Dále se očekává, že nasnímaný 3D svět nějakým způsobem rekonstruueme. Po nasnímání 3D struktury skutečného světa se nejčastěji vezmou 2D hloubkové mapy a přemění se ve 3D mračno bodů (viz obrázek 1.8), kde každý



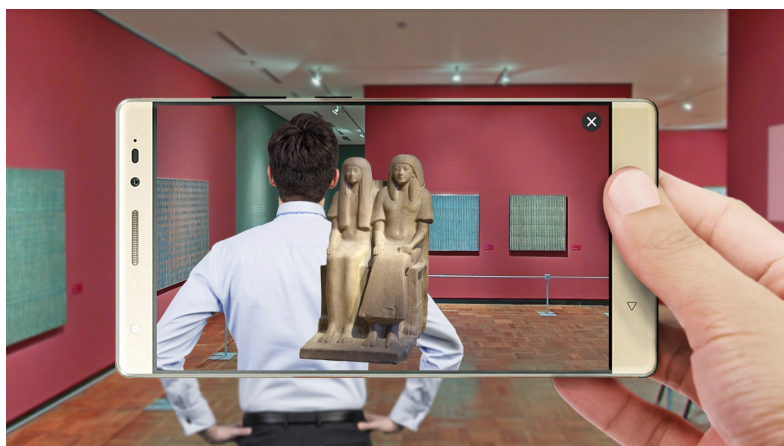
Obrázek 1.8: 3D model mračna bodů. Převzato z [17].



Obrázek 1.9: 3D síť vytvořená z 3D modelu mračna bodů. Převzato z [18].

pixel z toho snímku získá 3D polohu vzhledem ke kameře. Následně se z 3D modelu mračna bodů vytvoří méně výpočetně náročná 3D síť (viz obrázek 1.9), která ve výsledku toto 3D mračno bodů pokrývá. [16]

Ve virtuální realitě je docela snadné oklamat uživatelskou nedůvěru, jelikož máme kontrolu nad všemi prvky prostředí, ve kterém se nachází. V realitě rozšířené tuto kontrolu ztrácíme, protože drtivou většinu obrazovky tvoří reálný svět. Na obrázku 1.10, si můžeme všimnout, že sochy Maji a Meryt zcela nezapadají do skutečného světa. Za zvláštností obrázku stojí ne-



Obrázek 1.10: Ukázka špatné okluze na sochách Maji a Meryt. Převzato z [19].

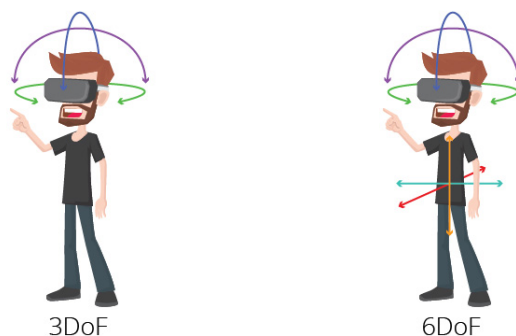
správné překrytí reálného objektu, v tomto případě postavy muže, objektem virtuálním. Jev, kdy vkládané virtuální objekty mohou být skryty za objekty skutečnými, nazýváme okluze.

V zásadě se okluze řeší tak, že se přestanou vykreslovat části virtuální scény, které mají být překryty, na základě znalostí 3D struktury reálného světa. 3D síť, kterou jsme získali z digitálního 3D modelu mračna, následně použijeme na vytvoření průhledné masky, kterou potřebujeme ke správnému vykreslení objektů scény. [16]

1.3.1 ARKit

Před vydáním operačního systému iOS 11 a zpřístupněním vůbec prvního frameworku pro iOS, který umožňuje pracovat v rozšířené realitě, museli vývojáři psát své vlastní algoritmy pro vyhledávání objektů v reálném světě a následnou interakci tohoto světa s objekty virtuálními. To se ale roku 2017 změnilo a Apple Inc. začátkem června vydal beta verzi frameworku s názvem ARKit, která umožňuje developerům vytvářet AR aplikace pro iPhone a iPady. [20] Minimálním požadavkem na spolehlivou funkčnost ARKitu je alespoň operační systém iOS 11.0 běžícím na iOS zařízení s čipem A9 (iPhone 6S, iPhone 6S Plus, iPad 9.7 2017) a vyšším. [21]

K trackování skutečného světa okolo iPhone či iPadu a zjištění, jak se v něm zařízení pohybuje, využívá ARKit vizuálně-inerciální odometrii (proces zpracování dat z kamery a inerciálních měřících jednotek k odhadnutí polohy, orientace a rychlosti zařízení [22]) v tomto případě pouze pomocí spojení obrazu z kamer s daty z akcelerometru a gyroskopu. [23] ARKit ji využívá nejen k analýze okolí, ale i k detekci vertikálních a horizontálních rovin.



Obrázek 1.11: 6DOF podporuje kromě rotace zařízení i pohyby nahoru, dolů, doprava, doleva, vpřed, vzad. Převzato z [24].

V AR implementacích podporujících three degrees of freedom¹ (3DOF), můžeme zařízením pouze rotovat okolo os x, y a z. Chceme-li objekt v rozšířené realitě pozorovat například tak, že jej budeme obcházet kolem dokola, 3DOF nám již stačit nebude, jelikož neřeší pohyb zařízením dopředu, dozadu, nahoru a dolů (viz obrázek 1.11). Avšak ARKit podporuje mimo 3DOF i 6DOF², jež tento problém řeší. Oficiální dokumentace dokonce doporučuje používat 3DOF pouze v případech, kdy 6DOF není nedostupné.

Ke správnému nasvícení a stínování překryvné grafiky můžeme využít `ARLightEstimate` – dopomůže k vytvoření podobných světelných podmínek jako ve skutečném světě. [25]

ARKit je také kompatibilní s mnoha renderovacími enginy — SceneKit a RealityKit pro 3D grafiku, SpriteKit pro 2D grafiku ale i Metal, Unity či Unreal Engine.

SceneKit

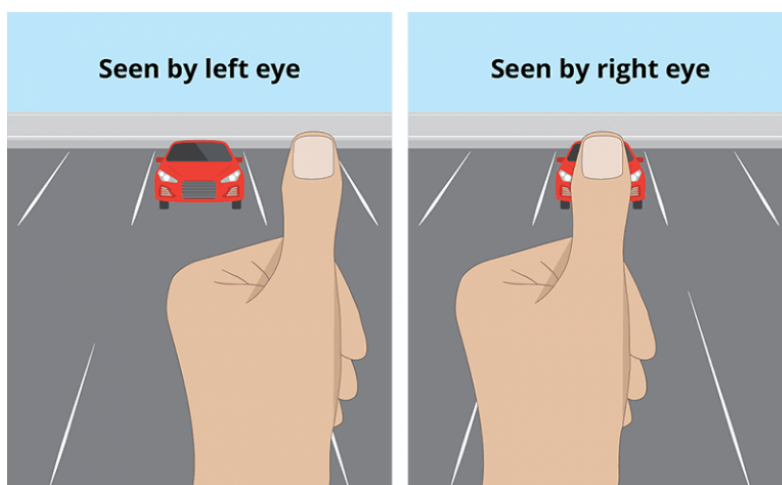
SceneKit je rozhraní, které vytváří 3D scény a přidává do aplikací 3D obsah pomocí high-level popisu scény. Pomáhá v iOS aplikacích vytvářet animace, fyzikální simulace a efekty, a může měnit geometrii a materiály objektů za běhu. Podporuje jazyk Objective-C i Swift. [26]

SpriteKit

Pomocí frameworku SpriteKit můžeme vykreslovat 2D objekty – obrázky, text, tvary, částice či dokonce videa. Podporuje také animace a simulace fyziky u 2D objektů, silová pole, detekci kolizí ale i generování nových osvětlení. [27]

¹tři stupně volnosti

²šest stupňů volnosti



Obrázek 1.12: Prst, který je k pozorovateli blíže nežli auto, má při pohledu z levého oka jinou pozici než při pohledu z oka pravého. Převzato z [32].

RealityKit

RealityKit je nejmladší z rodiny renderovacích Apple frameworků. Simuluje a vykresluje 3D objekty a pracuje s daty z ARKitu k vykreslování objektů do reálného světa. Narozdíl od SceneKit podporuje pouze jazyk Swift. [28]

Metal

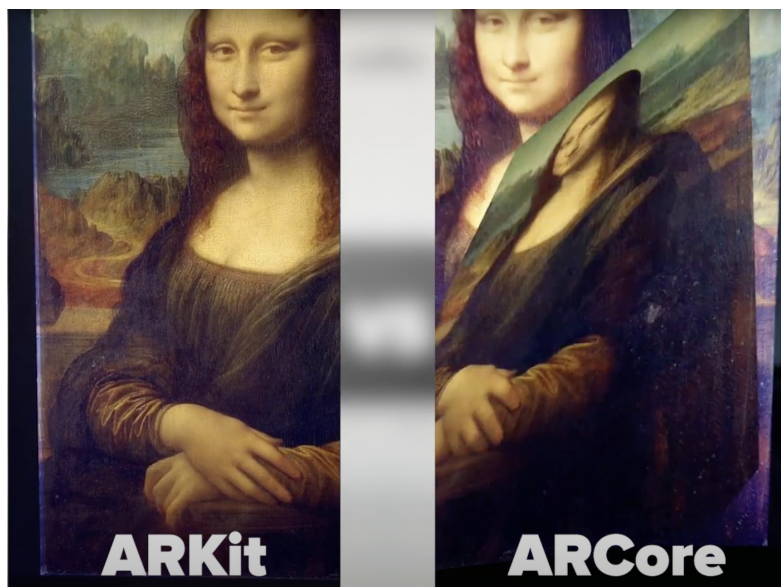
Pokud si chce uživatel vykreslování scény naprogramovat kompletně sám a komunikovat přímo s GPU, může použít speciální framework s názvem Metal. Jeho použití je vhodné například u aplikací, jež vykreslují komplexní scény či provádí složité vědecké výpočty. [29]

1.3.2 ARCore

Na začátku roku 2018 vydal Google svůj vlastní framework ARCore. K pochopení toho, jak se ve skutečném světě zařízení uživatele pohybuje, dopomáhá motion tracking, který využívá taktéž vizuálně-inerciální odometrii. [30] Je založen na podobném principu jako náš mozek – ten zpracovává mimo jiné i obraz z levého a pravého oka a poté odvozuje vzdálenost mezi námi a předmětem (viz obrázek 1.12). [31]

Mezi další technologie patří rozpoznávání velikostí a pozic horizontálních, vertikálních i zkosených rovin. Detekce takových rovin nám umožňuje registrovat například podlahu či stůl a následně na ně pokládat nejrůznější předměty.

ARCore detekuje informace o aktuálním osvětlení místnosti ve skutečném světě a v závislosti na tomto osvětlení poskytuje korekci virtuálního obsahu stejně jako ARKit. [33]



Obrázek 1.13: Srovnání ARKit 2.0 (iPhone X) a ARCore 1.9.1 (Pixel 3). Převzato z [36].

Narozdíl od ARKitu podporuje nejen zařízení s operačním systémem iOS 11 a více, ale i zařízení s operačním systémem Android 7 a více.

1.3.3 ARKit vs. ARCore

U ARKit a ARCore frameworky rozhodně nalezneme podobnosti. Oba jsou kompatibilní s neznámějším herním enginem Unity [34], podobně snímají změny osvětlení a k pohybovým sensorům přistupují taktéž stejně.

Mezi jejich odlišnosti jistě zařadíme způsob snímání světa. Pokud srovnáme rychlost a kvalitu mapování, ARCore je na tom mnohem lépe. ARKit používá metodu klouzavého okénka a uchovává pouze nedávno zažitá data, zatímco ARCore skladuje informaci o prostředí více. Na největší rozdíl však narazíme při výběru zařízení – iOS zařízení mají díky TrueDepth kameře a LiDAR senzoru obrovský náskok v rozpoznávání hloubky a následné augmentaci (viz obrázek 1.13), ale této výhody lze využít pouze u zařízení s čipem A12 a novějším. [35]

Vývoj aplikací pro zařízení s operačním systémem iOS

iOS je mobilní operační systém vyvinutý společností Apple Inc. a používaný na zařízeních iPhone a iPod Touch. Běžel taktéž na iPad zařízeních, avšak od roku 2020 mají svůj vlastní operační systém s názvem iPadOS. [37] Aktuální verze iOS je iOS 15, která byla představena na Apple Worldwide Developers Conference (WWDC) v červnu roku 2021. [38]

K vývoji aplikací pro zařízení s operačním systémem iOS můžeme využívat rozsáhlé množství vývojových prostředí, architektur, frameworků a knihoven, avšak nutností je vývoj na operačním systému MacOS. Další podmínkou vývoje pro iOS zařízení je instalace vývojového prostředí s názvem Xcode a založení Apple Developer Account. Registrace k účtu Apple Developer je zdarma, avšak pokud chceme aplikaci publikovat na obchod App Store, což je platforma k objevování a stahování aplikací [39], musíme zaplatit roční poplatek ve výši 99 USD ³. [40]

2.1 Zásady iOS

Apple's Human Interface Guidelines je soubor doporučení/zásad k vytvoření kvalitní aplikace pro Apple zařízení. Mezi obecné vlastnosti takové aplikace patří dle těchto doporučení následující:

1. **Přehlednost** – Text má být vždy čitelný, ikony přehledné a jasné, grafické efekty jemné. Důležitý obsah je podtržen pomocí prázdných míst, barev, fontů či grafických prvků.
2. **Úcta** – Uživatel by s aplikací nikdy neměl zápasit. Je nabádán k interakci pomocí plynulých animací a hezkému uživatelskému rozhraní.

³Platí k 10.02.2022

Obsah typicky překrývá celou plochu obrazovky. Interaktivní prvky jsou podtrhnuty rozostřením či průsvitností, čímž se zachová jemný a vřdušný vzhled. Aplikace obsahuje minimální počet stínů, rámečků či přechodů.

3. **Hloubka** – Hloubka napomáhá k celkovému porozumění a přehlednosti. Zvýrazníme ji pomocí zřetelně vizuálně oddělených vrstev či přechodů mezi jednotlivými stránkami. K objevu funkcí slouží interaktivní prvky.

Apple Inc. taktéž popisuje designové principy, které mají aplikaci dopomoci k většímu dosahu a dopadu.

1. **Estetická integrita** – Vzhled a chování aplikace se společně doplňuje s funkčností.
2. **Konzistence** – Aplikace se chová dle očekávání uživatele a aplikuje známé standardy a paradigmaty.
3. **Přímá manipulace** – Uživatel okamžitě vidí výsledky svých akcí (např. při používání gest či otáčení zařízení).
4. **Zpětná vazba** – Aplikace reaguje na každou akci uživatele a ukazuje výsledky, aby jej informovala.
5. **Metafory** – Aplikace využívá známých jevů z reálného či digitálního světa. Virtuální objekty jsou metaforami pro tyto známé zážitky – stiknutí tlačítka, otáčení stránek knihy apod.
6. **Kontrola uživatele** – Aplikace by nikdy neměla za uživatele rozhodovat sama, avšak může navrhnout postup nebo varovat před nebezpečnými důsledky. [41]

2.2 Architektonické vzory

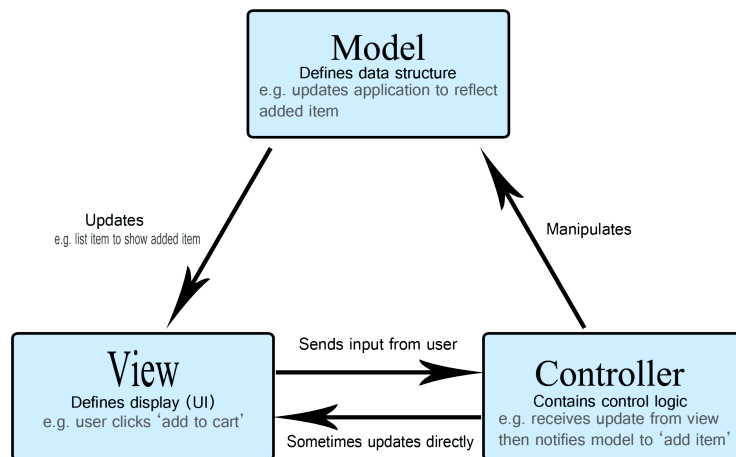
Mezi nejčastěji používané architektonické návrhové vzory patří MVC a MVVM. Řeší problém tzv. „špagetového kódu“, kdy v jednom souboru máme logiku i vykreslování výstupu. Takový kód je velmi špatně čitelný a špatně se udržuje, a proto vznikly architektonické návrhové vzory, který tento problém řeší.

MVC

Architektonický návrhový vzor Model-View-Controller (MVC) se skládá ze tří částí:

- **Model** – Odděluje logiku. Především udržuje data a nezajímá se absolutně o to, jak budou data, která poskytuje, zformátována.

- **View** – Stará se o zobrazení výstupu uživateli na základě informací z Model části a obsahuje minimální množství logiky.
- **Controller** – Controller získává a modifikuje data z Model části. Dále je posílá do View, které se postará o jejich prezentaci uživateli. Zodpovídá za komunikaci mezi View, Modelem a uživatelem. Jedná se o vstupní bod [42].

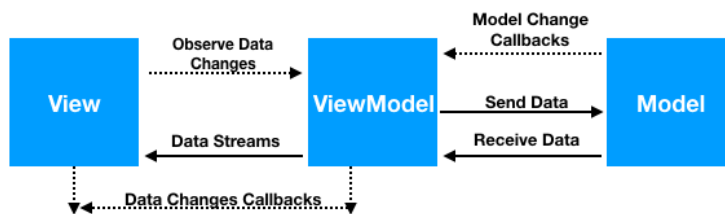


Obrázek 2.1: Architektonický vzor MVC. Převzato z [43].

MVVM

Architektonický návrhový vzor Model-View-ViewModel oproti MVC obsahuje část ViewModel. MVVM se tedy skládá z následujících [42]:

- **Model** – Stejně jako u MVC.
- **View** – Stejně jako u MVC, ale získává data pouze z ViewModel. Jedná se o vstupní bod.
- **ViewModel** – Stará se o logiku a data View – prezentující funkce, metody a podobně. Spojuje části Model a View.



Obrázek 2.2: Architektonický vzor MVVM. Převzato z [44].

2.3 UIKit, SwiftUI

Pro tvorbu uživatelského rozhraní můžeme využít dva frameworky, kde každý využívá jiný přístup k návrhu – UIKit nebo SwiftUI.

UIKit

Framework UIKit spravuje základní UI elementy (tlačítka, popisky, animace apod.), které se využívají ve vývoji aplikací pro Apple zařízení. Je postaven na programovacím jazyce Objective-C, vznikl dříve než SwiftUI a stojí na návrhovém vzoru MVC. Uživatelské rozhraní aplikace využívající UIKit mohou být vytvořeny několika různými způsoby [45]:

- V prostředí Interface Builderu bez nutnosti napsání jediného řádku kódu.
- Pomocí kódu, kde grafické prvky a jejich rozložení jsou popsány v programovacím jazyce Swift nebo Objective-C.
- Kombinací obou předchozích.

SwiftUI

SwiftUI je nový deklarativní framework sloužící k vývoji aplikací pro Apple zařízení. Veřejnosti byl představen roku 2019 a jeho největším rozdílem oproti UIKit je právě deklarativnost. [45] Nemá možnost vývoje v rámci Interface Builderu.

Se SwiftUI vývojář může přesně popsat, jak má uživatelské rozhraní vypadat a SwiftUI udělá těžkou práci za něj. V UIKit se každá změna na obrazovce musela implementovat úpravou modelu a překreslením obrazovky. Ve SwiftUI může mít každé view (základní stavební blok uživatelského rozhraní) stavovou proměnnou. S každou změnou této stavové proměnné se obrazovka automaticky překreslí. Takovou změnu stavu můžeme delegovat do dalších obrazovek,

což bylo pomocí UIKitu mnohem pracnější. SwiftUI je podporováno pouze na zařízeních s operačním systémem iOS 13 a novějším a je stále ve vývoji. [45]

2.4 Xcode

Xcode je vývojové prostředí vyvinuté společností Apple Inc., které slouží k vytváření, testování, optimalizování a publikování aplikací pro Apple zařízení.

Mezi nejvýznamější vlastnosti tohoto prostředí jistě patří schopnost simulovat všechna Apple zařízení, a umožňuje tím jednoduché testování kompatibility aplikace. Mimo simulaci může developer z pohodlí vývojového prostředí nainstalovat, testovat či krokovat aplikaci i na fyzickém zařízení.

Dalším nástrojem Xcode je tzv. Interface Builder, který umožňuje vytváření designu aplikace a následné propojení s kódem. Lze do něj přidávat grafické prvky (tlačítka, popisky, obrázky, ...) a následně je upravovat či propojovat. Takový návrh Xcode sám zpracuje, vytvoří z něj zdrojový kód a následně překládá překladačem. [46]

Mimo výše zmíněné vlastnosti obsahuje taktéž zabudovanou podporu verzování a vestavěný debugger.

2.5 Swift, Objective-C

Nejdříve se k vývoji pro Apple zařízení používal programovací jazyk s názvem Objective-C, který je nadstavbou jazyka C. Avšak na WWDC v roce 2014 představil Apple Inc. nový programovací jazyk Swift a odstartoval tak novou etapu programování pro zařízení Apple.

Swift se používá k vývoji aplikací pro zařízení s operačními systémy iOS, iPadOS, macOS, tvOS a watchOS. Jeho základy jsou postaveny na programovacím jazyku Objective-C a mnoha dalších jako například Rust, Haskell, Ruby, Python a C#. [47]

Swift je velmi přívětivý jazyk pro začínající programátory. Proměnné jsou vždy inicializovány před jejich použitím, integery jsou testovány na přetečení, pole jsou testována na out-of-bound error a paměť je spravována automaticky. [48]

Zajímavostí je tzv. Optionals. Optional rozšiřuje obor hodnot všech datových typů o hodnotu nil a přidává možnost „být prázdný“. Datové typy proměnných short a long v následujícím kódu jsou ekvivalentní (optional integer):

```
1 let short: Int? = Int("2000")
2 let long: Optional<Int> = Int("2000")
```

2. VÝVOJ APLIKACÍ PRO ZAŘÍZENÍ S OPERAČNÍM SYSTÉMEM IOS

Jestliže programátor s jistotou ví, že hodnota `nil` nemůže nikdy nastat, může optional proměnnou rozbalit pomocí „!“ . Tento způsob je nazýván force unwrap.

```
1 let myNumber = Int("2000")!  
2 print(myNumber)  
3 // Prints "2000"
```

Pokud však narazíme na situaci, kdy rozbalenou hodnotou `nil` být může, musíme specifikovat, jak dále pokračovat. Buď se pomocí `if-else` zeptáme, zdali náhodou není hodnota optional proměnné `nil` a dále pokračujeme dle potřeby, nebo využijeme tzv. optional binding či optional chaining.

```
1 // if-else statement  
2 if maybeNil != nil {  
3     // do something  
4 } else {  
5     // nil found, do something else  
6 }
```

Jestliže chceme přiřadit neprázdnou hodnotu optional proměnné do nové ne-optional proměnné, můžeme využít optional binding.

```
1 // Optional binding  
2 if let newNotNil = maybeNil {  
3     // maybeNil was not nil, its value is now assigned to newNotNil  
4 } else {  
5     // maybeNil was nil, do something  
6 }
```

Optional chaining využijeme například v situaci, kdy chceme bezpečně použít metodu zabalené instance.

```
1 // Optional chaining  
2 if imagePaths["pokemon"]?.hasSuffix(".jpg") == true {  
3     print("The pokemon image is in JPG format.")  
4 }
```

Pokud narazíme na situaci, kdy rovnou víme, čím chceme případný nil nahradit, použijeme tzv. nil-coalescing operátor „??“. [49]

```
1 // Nil-coalescing operator
2 let defaultImagePath = "/images/default.jpg"
3 let ikeaPath = imagePaths["pokemon"] ?? defaultImagePath
4
5 print(ikeaPath)
6 // No ikea path in imagePaths -> /images/default.jpg
```


Analýza

V této kapitole je obsažena analýza cílové skupiny uživatelů a požadavků na funkcionalitu aplikace, která byla vytvořena na základě výsledků z anonymního dotazníkového šetření. Na závěr kapitoly jsou zde stručně popsány již existující aplikace, které využívají rozšířené reality a mohou sloužit k inspiraci při návrhu a implementaci.

3.1 Dotazníkové šetření

Dotazníku se celkem zúčastnilo 98 respondentů. První tři otázky se zaměřovaly na průzkum cílové skupiny. 66,3 % respondentů uvedlo, že rádo kreslí. Celkem 69,4 % dotazovaných se nepovažuje za zdatné kreslíře a 68,4 % by se líbila možnost obkreslení obrázku pomocí mobilního zařízení. Z detailní analýzy vyplývá, že zhruba 45 % z dotazovaných, kteří nekreslí rádi, by se líbila možnost obkreslení obrázku pomocí mobilního zařízení.



Obrázek 3.1: Výsledky dotazníkového šetření

V další části dotazníku se otázky zaměřovaly na funkční požadavky aplikace. Z průzkumu vyplývá, že většina dotazovaných by uvítala dva módy obrázků – s původními barvami i pouze obrysy. Dále by si většina přála obres-

lovat obrázky z galerie uživatele, ale i galerie aplikace. Celkem 99 % respondentů by chtělo upravovat velikost a polohu obrázku promítnutého na papíře. Avšak měření času při obreslování by uvítalo pouze 34,7 % dotazovaných.

Poslední povinná otázka se zabývala nefunkčními požadavky aplikace. Stěžejním faktorem pro uživatele je funkčnost aplikace. Dále zajímá potenciální uživatele především přehlednost a až na třetím místě se nachází design.

Na nepovinné otázky odpovědělo celkem 90 dotazovaných. Zkoumaly, zdali je mezi dotazovanými větší povědomí o pojmu rozšířená realita či virtuální realita. Zhruba 91,1 % z dotazovaných zná pojem virtuální realita, avšak pouze 47,3 % ví, co znamená realita rozšířená.

3.2 Cílová skupina

Cílovou skupinou jsou běžní uživatelé mobilních zařízení. Především tedy amatérští umělci, kteří chtějí jednoduše nakreslit daný obrázek. Aplikace by měla být přehledná a snadno použitelná dítětem, které chce nakreslit kamarádovi obrázek do památníku, dospělým či seniorem.

Od uživatele se očekává znalost základní práce s mobilním zařízením, avšak předchozí zkušenosti s rozšířenou realitou nejsou předpokládány. Mezi uživateli se však mohou objevit i profesionální malíři. Někteří malíři si promítají základní prvky obrazů pomocí dataprojektorů, aby určili směr, kterým se dílo bude ubírat. Z dotazníku zároveň vyplynulo, že aplikaci by chtěli využívat i velmi zdatní kreslíři.

3.3 Funkcionalita aplikace

Klasický případ užití je následující. Uživatel si vybere obrázek ze své či poskytnuté galerie. Následně vybere mód, ve kterém chce obrázek obkreslit – originální obrázek či obrysy (popřípadě další úpravy). Uživatel krátce naskenuje plochu, kam chce obrázek promítnout, a kliknutím obrázek vloží na dané místo. Může jej následně po této ploše libovolně přesouvat či měnit velikost. Jakmile je hotov, opouští aplikaci a dotváří dílo v reálném světě.

3.4 Existující aplikace

Tato část se věnuje popisu a analýze existujících mobilních aplikací, které využívají rozšířenou realitu. Jsou záměrně vybírány tak, aby se co nejvíce přiblížily cíli této bakalářské práce. Tato analýza slouží k prozkoumání existujících řešení a inspiraci v oblasti návrhu uživatelského rozhraní. Taktéž se zabývá chybami, kterým by bylo vhodné se při implementaci vyhnout.



Obrázek 3.2: Ukázka použití aplikace InkHunter. Převzato z [50].

3.4.1 InkHunter

InkHunter je mobilní aplikace využívající rozšířené reality k vytvoření iluze tetování na těle uživatele. Uživatel si vybere tetování z galerie, nakreslí obrázek smajlíka dle předlohy na místo, kam by si jej přál, a InkHunter ho zde augmentuje.

Aplikace je zajímavá, avšak objevuje se v ní mnoho chyb. Na některých zařízeních bez vysvětlení nefunguje tlačítko „Try“, které uživatele naviguje k vyzkoušení tetování. Kreslení smajlíka je velmi nešikovné, jelikož pokud chce uživatel vyzkoušet tetování na ruce, jednou rukou drží telefon a míří s ním na ruku druhou. Smajlík se však musí obkreslit přesně podle předlohy, jinak je tetování pokřivené či se vůbec nezobrazí. Uživatel by tedy v takovémto případě potřeboval alespoň tři ruce či pomoc druhé osoby.

3.4.2 Etsy

Mobilní aplikace Etsy se zaměřuje především na prodej a inzerci rukodělné výroby. Možnost využití rozšířené reality byla přidána poměrně nedávno (2020). Lze ji využít jen k augmentaci obrazů, fotografií či tiskovin a to pouze na vertikální plochy, které uživatel musí předem naskenovat. [51]

3.4.3 SketchAR

Některé funkcionality mobilní aplikace SketchAR se dosti podobají funkcionalitám aplikace, která je cílem této bakalářské práce. Výuková aplikace obsahuje opravdu velké množství kurzů, kde se uživatel učí kreslit pomocí rozšířené reality. Kurzy jsou členěny do několika kategorií, mezi které například patří kresba květin, portrétů, částí těla apod.

Nejdříve uživatel namíří svůj fotoaparát na papír, na který chce kreslit či malovat, a v postupných krocích se na něj augmentuje daný obrázek. Okluze při překrytí obrázkem prstem či perem zde není řešena. Obrázek se zobrazí i přes objekt ve skutečném světě, což je uživatelsky velmi přívětivé, jelikož uživatel neustále vidí celou předlohu.

Funkcionalita výběru vlastního obrázku je zde (stejně jako mnoho dalších funkcionalit) měsíčně zpoplatněna, a to ve výši 209 Kč⁴. Uživatelské rozhraní mi nepřijde příliš přívětivé. Tlačítko k přidání vlastního obrázku lze najít velmi obtížně. Aplikace občas nereaguje a je nutné ji restartovat.

Vývojáři SketchAR nevyužívají standardních knihoven, nýbrž naimplementovali vlastní algoritmy, které si společnost nechala patentovat. Aplikace využívá počítačového vidění, strojového učení ale i neuronové sítě. Je dostupná na App Store, Google Play, Galaxy Store a AppGallery. [52]

⁴Platí k 10.03.2022

Návrh

Kapitola popisuje návrh aplikace. Jsou zde představeny případy užití aplikace a návrh uživatelského rozhraní. Návrh mobilní aplikace byl založen především na předchozí analýze cílové skupiny a výsledcích dotazníkového šetření, jež byly popsány v předchozí kapitole 3.

4.1 Případy užití

Cílem aplikace je augmentovat obrázek na uživatelem vybrané místo. Počet kroků, kterými uživatel musí projít, by měl být tedy minimální.

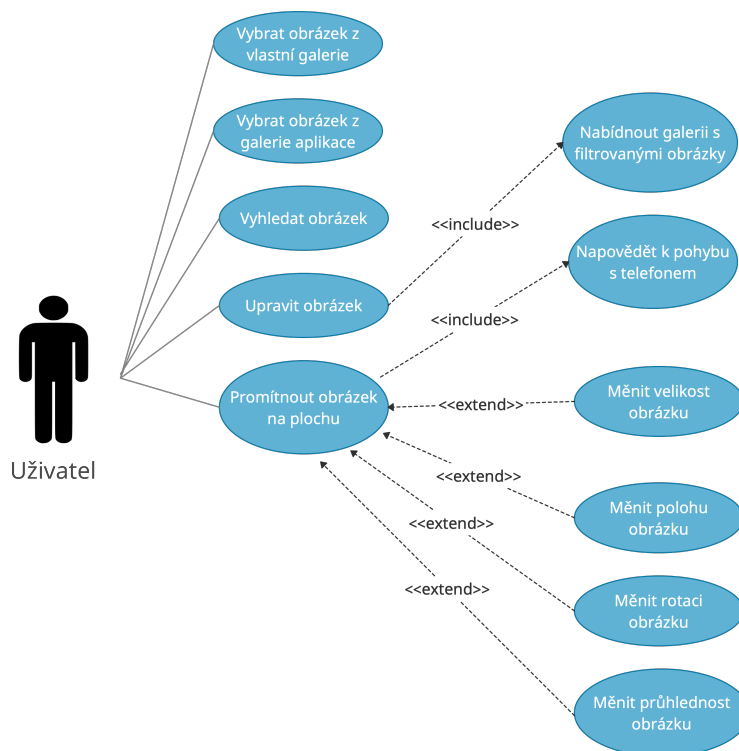
Nejdříve si uživatel vybere obrázek ze své galerie či z galerie aplikace. Následně barevně upraví daný obrázek dle potřeb, popřípadě převede pouze na obrázek s hranami a augmentuje jej na požadované místo. Dle výsledků dotazníku je pro uživatele velmi podstatné s augmentovaným obrázkem manipulovat. Uživatel jej může tedy libovolně posouvat, rotovat či měnit velikost. Z popsaného průchodu aplikací byl vytvořen diagram případů užití, zobrazený na obrázku 4.1.

4.2 Uživatelské rozhraní

Při tvorbě uživatelského rozhraní byl kladen důraz především na jednoduchost. Uživatel by se měl po aplikaci navigovat intuitivně a bez nutnosti jakýchkoli nápověd.

4.2.1 Hlavní menu

Hlavní menu tvoří několik prvků. Samozřejmě se zde nachází název aplikace, který je však nepodstatným elementem pro uživatele a není na něj kladen důraz. Následně uvítací zpráva, která motivuje uživatele ke kreslení. Dalšími prvky jsou galerie, ze kterých si uživatel vybírá obrázky, které by chtěl kreslit. Jestliže nechce využít obrázků nabídnutých aplikací, může pomocí tlačítka +



Obrázek 4.1: Diagram případů užití

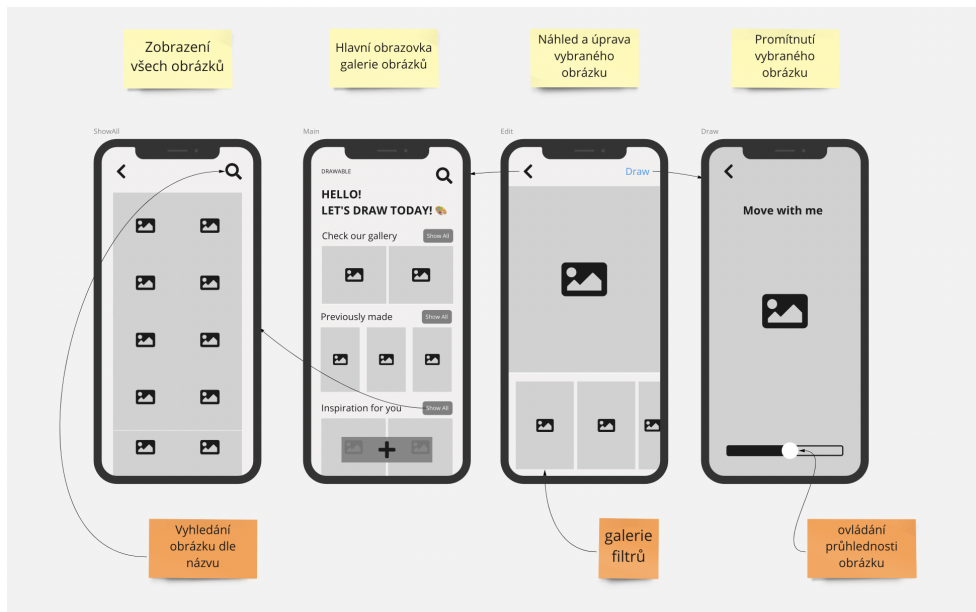
přidat svůj vlastní obrázek z galerie mobilního zařízení. Po vybrání obrázku je automaticky přesměrován na další obrazovku. Jestliže chce vidět více obrázků navrhovaných aplikací, může využít tlačítek „Show All“ či lupy v pravém horním rohu. Poté je přesměrován na novou obrazovku.

4.2.2 Show All obrazovka

Uživatel si na této obrazovce může prohlížet obrázky navrhované aplikací nebo v nich vyhledávat. Po vybrání jakéhokoli obrázku je přesměrován na další obrazovku k úpravě obrázku. Taktéž je zde možnost vrácení se na předchozí obrazovku.

4.2.3 Edit obrazovka

Uživatel zde může či nemusí upravit vybraný obrázek pomocí filtrů, které se nacházejí ve spodní části. Taktéž se může lehce vrátit pomocí tlačítka v levém horním rohu či pokračovat a začít kreslit.



Obrázek 4.2: Návrh grafického rozhraní

4.2.4 Draw obrazovka

Přes celou obrazovku uživatel vidí vstup z kamery. Může se vrátit na předchozí či pomocí slideru regulovat průhlednost obrázku. Na této obrazovce probíhá veškerá augmentace. Uživatel je upozorněn, že má před položením obrázku pohybovat zařízením.

Implementace

Tato kapitola pojednává o způsobu implementace mobilní aplikace pro iOS zařízení, jejíž cílem je dopomáhat pomocí rozšířené reality uživateli při kreslení obrázků. Aplikace je psána v programovacím jazyce Swift. K práci s rozšířenou realitou využívá aplikace framework ARKit a k implementaci uživatelského rozhraní deklarativní framework SwiftUI.

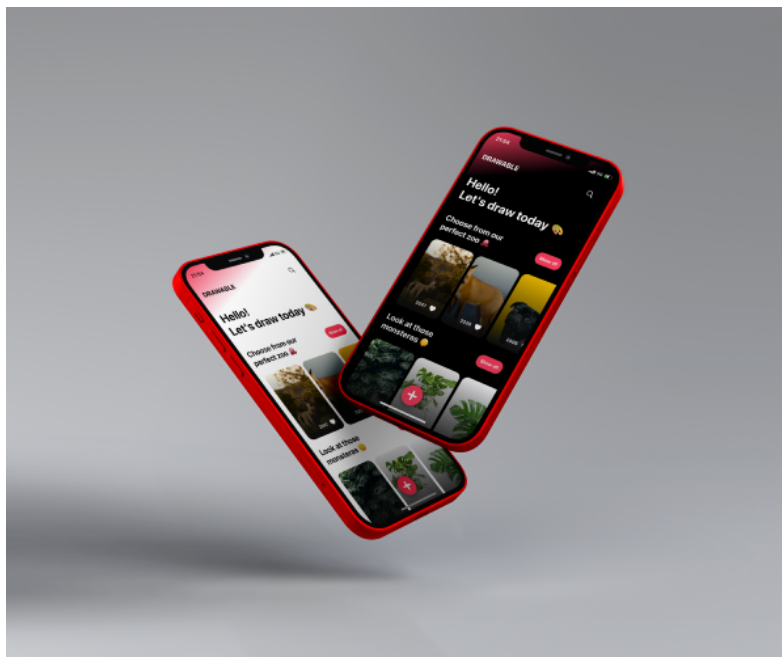
5.1 Uživatelské rozhraní

Implementace uživatelského rozhraní vycházela z návrhu, který je popsán v kapitole 4 a byl realizován pomocí frameworku SwiftUI. Taktéž se řídí Apple's Human Interface Guidelines, které jsou popsány v kapitole 2.

Aplikace využívá výchozího systémového fontu nabízeného iOS a jeho různých stylů k odlišení důležitosti jednotlivých nadpisů. Jednotlivé prvky uživatelského rozhraní jsou napříč celou aplikací zaoblené a dodržují tak konzistentnost. Tlačítko pro přidání uživatelova obrázku je lehce vyzdvihnuto stínem a navodí tak pocit hloubky. V rámci celé aplikace byla použita omezená paleta barev, která je sjednocená i s ikonou aplikace. Aplikace taktéž reaguje na barevné schéma zařízení – tj. tmavý či světlý režim, čehož bylo docíleno pomocí barevných setů v asset katalogu aplikace (viz obrázek 5.1).

Způsob, kterým jsou jednotlivé prvky uspořádány (texty, tlačítka, galerie...), je specifikován pomocí **Hstack**, **VStack** či **ZStack**. Horizontální **ScrollView** tvoří galerie posouvateľnými a interaktivními. Galerie obrázků nabízených aplikací jsou proporčně odlišeny od galerie obrázků uživatele, která se zobrazí až poté, co uživatel vybral alespoň jeden obrázek. Nápisys jsou od obrázků odstíněny pomocí tmavých gradientů pro lepší čitelnost.

Tlačítka jsou řešena jako obrázky, u kterých je specifikováno, co se s nimi má stát, jakmile na ně uživatel klikne. Tato deklarace mi přijde mnohem přehlednější při využívání obrázků jako pozadí tlačítek. K vyhledávání těchto ikon byla použita aplikace SF Symbols.



Obrázek 5.1: Mockup výsledné aplikace ve světlém a tmavém režimu

Obrázky v galeriích jsou deklarovány stejným způsobem jako tlačítka. K zaoblení jejich rohů jsem využívala **clipShape**. Důležité bylo následně specifikovat **contentShape**, který popisuje oblast, kam může uživatel kliknout, aby daný obrázek vybral. Bez této informace by se mohlo stát, že by obrázky do sebe vzájemně zasahovaly a při výběru by docházelo k chybám.

```
1      Image(uiImage: userGallery[index].image)
2          .resizable()
3          .scaledToFill()
4          .frame(width: imageWidth, height: imageHeight)
5          .clipShape(RoundedRectangle(cornerRadius: 15))
6          .contentShape(RoundedRectangle(cornerRadius: 15))
7          .onTapGesture {
8              pickedImage = userGallery[index].image
9              imageAuthorName = userGallery[index].authorName
10             imageAuthorUrl = userGallery[index].authorLink
11             imageIsChosen = true
12         }
```

5.2 Výběr obrázku

Z výsledků dotazníkového šetření popsaného v kapitole 3 vyplynulo, že by uživatelé měli zájem obkreslovat vlastní obrázky i obrázky poskytované aplikací. K výběru vlastních obrázků slouží **UIImagePickerController**. S jeho vytvořením se iOS automaticky zeptá uživatele, zdali dává přístup aplikaci k jeho fotografiím.

Aplikace navrhuje uživateli taktéž své obrázky, resp. k jejich poskytnutí využívá Unsplash API. Jedná se o JSON API, které obsahuje přes 3,8 milionu fotografií. K jejich získání byla vytvořena třída `PhotoManager`, speciálně tedy metoda `fetchPhoto`. Vstupními parametry jsou `photoName`, které slouží jako vyhledávací výraz, a celočíselné `numOfPhotos`, které specifikuje, kolik obrázků chceme získat. Na základě těchto informací se společně se soukromým klíčem vygenerovaným Unsplash API vytvoří **URL**. Pomocí něj zavoláme `dataTask` na **URLSession**. Získaná data jsou zpracována pomocí **JSONDecoder** a decodable struktury `PhotoData`, kde jsou specifikovány informace, které má získat.

Třída `PhotoManager` je vedena jako **ObservableObject**, což znamená, že jakékoli view, které jeho instanci uchovává jako **ObservedObject**, se obnoví, kdykoli je změněna hodnota jeho proměnné anotovaná pomocí **Published** [53]. V tomto případě je `Published` proměnná s názvem `photos`, která udržuje obrázky získané z Unsplash API. Zajišťuje tedy, že s každým přidáním nového obrázku do galerie, se obrazovka znovu překreslí.

Podmínkou použití Unsplash API je u jednotlivých fotografií uvést Unsplash a autora fotografie (implementace v aplikaci viz obrázek 5.2). Obě informace musí být včetně odkazů, které obsahují UTM parametr se jménem mobilní aplikace.

```

1  if let url = URL(string: urlString) {
2      let session = URLSession(configuration: .default)
3
4      let task = session.dataTask(with: url) { data, response, error in
5          if error != nil {
6              print(error!)
7              return
8          }
9          if let safeData = data {
10             if let safePhotos = self.parseJSON(photoData: safeData) {
11                 DispatchQueue.main.sync {
12                     self.photos.append(contentsOf: safePhotos.results)
13                 }
14             }
15         }
16     }
17     task.resume()
18 }

```

5.3 Editace obrázku

Uživatel má možnost před samotným augmentováním obrázků i upravit. To je řešeno galerií filtrů, která je implementována pomocí frameworku **CoreImage**.

Filtrování obrázků probíhá asynchronně na pozadí. Nejdříve z obrázku musíme vytvořit **CIImage**, který je následně předán jako vstupní parametr funkci **setValue** daného filtru. Výsledná data již hotového filtrovaného obrázku však nejsou zobrazitelná a musíme je převést zpět na **UIImage**.

Ve struktuře **FilteredImage**, která je identifiable, společně s obrázkem udržují i název použitého filtru a id, pomocí kterého následně lze iterovat ve **ForEach** cyklu k zobrazení výsledných obrázků.

Jelikož není možné publikovat změny z vláken běžících na pozadí, je nutné vrátit se zpátky do main vlákna a odsud přidat data filtrovaných obrázků, což je zajištěno pomocí **DispatchQueue.main.async**.

```

1  DispatchQueue.global(qos: .userInteractive).async {
2      // loading image into filter
3      let ciImage = CIImage(image: pickedImage)
4      filter.setValue(ciImage!, forKey: kCIInputImageKey)
5
6      // filtered image
7      guard let newImage = filter.outputImage else {return}
8
9      // filter name
10     let filterName = filter.name.dropFirst(2)
11
12     // creating UIImage
13     let cgImage = context.createCGImage(newImage, from: newImage.extent)
14     let filteredData = FilteredImage(image: UIImage(cgImage: cgImage!),
15                                     fileName: String(filterName))
16
17     DispatchQueue.main.async {
18         self.filteredImages.append(filteredData)
19     }
20 }

```

5.4 AR scéna

Stěžejní částí mobilní aplikace je AR scéna, ve které probíhá promítání a obkreslování daného obrázku. Byla naimplementována pomocí **ARSCNView**, která přidává virtuální objekty do skutečného světa. Nejdříve však musíme scénu nakonfigurovat, aby splňovala dané požadavky.



Obrázek 5.2: Ukázka obrazovky pro úpravu obrázků pomocí filtrů

5.4.1 Konfigurace a inicializace

V mobilní aplikaci je naimplementována detekce horizontálních i vertikálních ploch. Může nastat situace, kdy uživatel bude chtít kreslit nejen na papíry na stole či podlaze, nýbrž i na plochy vertikální – například stěny. Detekci těchto ploch je možné nastavit pomocí **ARWorldTrackingConfiguration**, která sleduje polohu zařízení ve scéně v šesti stupních volnosti, neboli 6DOF (viz 1.11), a proměnné **planeDetection**.

Dalším krokem je inicializace scény v rozšířené realitě. Tu provedeme zavoláním metody **run** objektu **session** instance třídy **ARSCNView**, jejíž vstupním parametrem je výše zmíněná konfigurace. Tato třída umožňuje zobrazování scény v rozšířené realitě pomocí frameworku **SceneKit**.

```
1 let configuration = ARWorldTrackingConfiguration()
2 configuration.planeDetection = [.horizontal, .vertical]
3 arView.session.run(configuration)
```

5.4.2 Náповěda při detekci ploch

K tomu, aby uživatel věděl, jak dopomoci aplikaci detekovat vertikální či horizontální plochy a zároveň tento proces urychlil, slouží **ARCoachingOverlayView**. Toto view poskytuje uživateli formou krátké animace instrukce k tomu, jak naskenovat své okolí pohybem telefonu. Je dostupné díky frameworku

ARKit a umožňuje specifikovat, zda chceme nápovědu pouze pro vertikální, horizontální či oba typy ploch.

Přidání této nápovědy je implementováno jako extension **ARSCNView** funkcí `addCoaching`. Nejdříve je vytvořen objekt `ARCoachingOverlayView`, následně specifikujeme cíl. Cílem mohou být vertikální či horizontální plochy – v tomto případě jsou to oba typy. V dalším kroce `coachingOverlayView` `session` přiřadíme naši AR `session` – tímto způsobem se zjistí, že plochy byly detekovány. V poslední části chceme zajistit, aby nápověda měnila velikost s natočením zařízení.

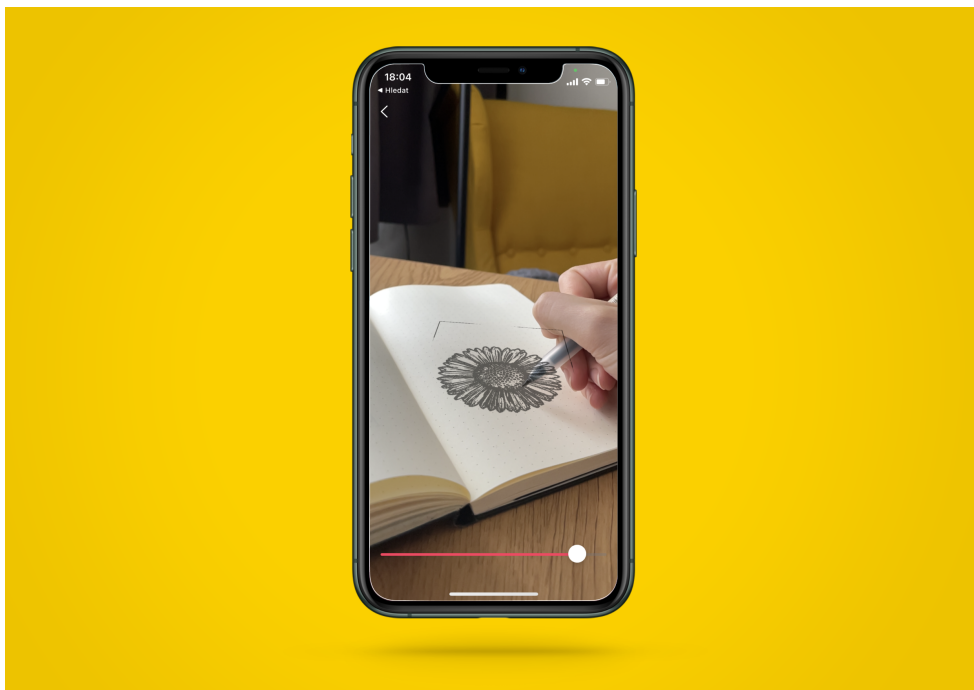
```
1 func addCoaching() {
2     // Create ARCoachingOverlayView object
3     let coachingOverlay = ARCoachingOverlayView()
4
5     // Set the AR goal
6     coachingOverlay.goal = .anyPlane
7
8     // Set session
9     coachingOverlay.session = self.session
10
11     // Rescale when device orientation changes
12     coachingOverlay.autoresizingMask = [.flexibleWidth, .flexibleHeight]
13
14     self.addSubview(coachingOverlay)
15 }
```

5.5 Vytvoření obrázku

Uživatel po dokončení detekce ploch vloží obrázek pomocí dotyku obrazovky na požadovaném místě. K umístění obrázku do prostředí využívá aplikace metodu `raycastQuery`. Vstupními parametry jsou pozice dotyku uživatele na obrazovku, typ plochy a zarovnání cílové plochy. Pokud se paprsek vytvořený ARKitem protne s danou plochou, metoda vrátí pozici tohoto protnutí, která se nachází v proměnné `worldTransform`.

Následně je vytvořena plocha, jejíž difúzní složkou materiálu je uživatelem zvolený obrázek. Vytvořená plocha má zprvu šířku 10 cm a dle původních rozměrů obrázku je dopočítána výška. Obrázek je nakonec zrotován dle typu plochy, na kterou je vkládán.

Při obkreslování obrázku může nastat situace, kdy se uživatel po delší dobu nedotkne obrazovky, a tím pádem dojde k jejímu zhasnutí. Tato vlastnost je v případě obrazovky s AR scénou zablokována a uživatel může nerušeně vytvářet své dílo.



Obrázek 5.3: Ukázka použití mobilní aplikace

5.6 Interakce s obrázkem

Aby si uživatel mohl přizpůsobit obrázek dle libosti, byly na základě analýzy naimplementovány 4 funkce. Z výsledků dotazníku vyplynulo, že uživatelé by rádi měnili polohu a velikost daného obrázku. Mimo implementaci těchto základních dvou funkcí je v aplikaci řešena i změna natočení a průhlednosti obrázku.

Změna průhlednosti

Změna průhlednosti obrázku je implementována pomocí slideru na obrazovce. S každým jeho posunem se volá metoda `ARView` delegáta s názvem `updatePlaneOpacity`.

Změna polohy

Pokud chce uživatel změnit polohu obrázku, stačí mu k tomu jednoduché gesto – podržet prst a následně jej posunout do cílové polohy. Obrázek se přesune, jestliže byla na cílovém místě detekována plocha. V opačném případě zůstane na posledním místě v tahu, které protínalo nějakou plochu. Tato funkcionalita je implementována pomocí `UIPanGestureRecognizer`.

Změna velikosti

Změna velikosti obrázku je řešena jednoduchým gestem pomocí dvou prstů měnící vzájemnou vzdálenost – stejným způsobem, jako když chce měnit velikost fotografie, textu apod. K tomu využívá opět **GestureRecognizer** speciálně tedy **UIPinchGestureRecognizer**.

Změna natočení

Pohybem dvou prstů rotujících proti či po směru hodinových ručiček se změní natočení obrázku na ploše. Tato funkcionality je naimplementována pomocí **UIRotationGestureRecognizer**.

Testování

V této kapitole jsou popsány výsledky uživatelského testování mobilní aplikace, jejíž implementace je nastíněna v kapitole 5. Před uživatelským testováním byl návrh aplikace několikrát konzultován s potenciálními uživateli, jejichž požadavky byly zohledněny z hlediska funkcionality i uživatelského rozhraní.

Uživatelské testování bylo provedeno s pěti uživateli na třech různých mobilních zařízeních – iPhone 7+, iPhone X, iPhone 12 mini. Všechna zařízení měla iOS 15.0 a vyšší. Testované osoby byly ve věku 15 až 53 let, z čehož tři osoby byly bez znalosti používání operačního systému iOS a dvě bez znalosti anglického jazyka. Předchozí zkušenost s rozšířenou realitou měl pouze jeden uživatel.

6.1 Testovací scénáře

Jednotlivé testy byly prováděny na různých zařízeních s předem nainstalovanou aplikací. Testovaly především funkčnost aplikace a vlídnost uživatelského rozhraní. Bylo vytvořeno celkem 5 scénářů testování.

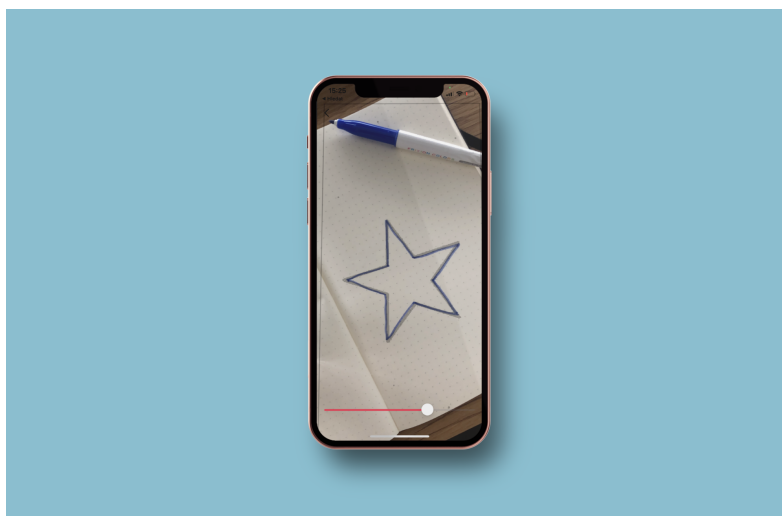
První scénář zkoumal přehlednost aplikace. Cílem bylo vybrat libovolný obrázek z galerie nabízené aplikací, upravit jej pomocí filtru a následně jej promítnout na horizontální plochu (stůl, podlaha apod.).

Druhý scénář testoval funkcionality výběru z různých galerií. Cílem bylo vybrat obrázek z galerie uživatele, galerie aplikace a následně i vyhledání obrázku pomocí daného výrazu.

Třetí scénář zkoumal interakci uživatele s promítnutým obrázkem. Jeho úkolem bylo obrázek zvětšit, natočit a přesunout po dané ploše na jiné místo.

Čtvrtý scénář měl za úkol otestovat různé typy ploch a jejich detekci. Uživatel měl promítnout obrázek i na plochy vertikální – stěnu, dveře, monitor apod. Následně s obrázkem na těchto plochách jakkoli interagovat.

Poslední scénář zkoumal obkreslování obrázku. Uživatel měl za úkol obkreslit jednoduchý obrázek hvězdy z uživatelské galerie na horizontální plochu.



Obrázek 6.1: Výsledek testovacího scénáře, který zkoumá použitelnost aplikace k obkreslování obrázků. Modrý obrys hvězdy byl nakreslen uživatelem, šedý obrys sloužil jako předloha k obkreslení.

6.2 Výsledky

Všichni uživatelé dokázali bez nutnosti nápovědy úspěšně dokončit zmíněné scénáře. Oba uživatelé bez znalosti anglického jazyka se dokázali pohybovat aplikací bez obtíží.

Největší problémy dělal čtvrtý scénář, který testoval různé typy ploch. Bylo zjištěno, že vertikální plochy bez textury nelze tak snadno naskenovat jako plochy horizontální. Promítání obrázků na stěny je tím pádem trochu složitější, avšak po dostatečném přiblížení ke stěně lze požadovaného výsledku dosáhnout.

Obkreslování obrázků probíhalo taktéž bez větších problémů, avšak v jednom případě se stalo, že uživatel dostatečně nenaskenoval prostředí a následně se mu obrázek pod rukou pohyboval. Po oddálení telefonu a přemístění obrázku byl tento problém vyřešen.

Zpětná vazba od uživatelů byla kladná. Uživatelské rozhraní se jim líbilo a vše jim přišlo dostatečně přehledné. Po následném rozhovoru o funkcích aplikace bylo zmíněno, že by uvítali více galerií s nabízenými obrázky na úvodní straně. Dále u jednoho uživatele bez znalosti anglického jazyka bylo zmíněno, že nerozuměl textu u nápovědy k naskenování ploch, avšak dle jeho slov byla animace dostatečně obsažná a pochopitelná.

6.3 Možná vylepšení

Po rozhovoru s cílovou skupinou uživatelů vzniklo několik zajímavých nápadů k rozšíření naimplementované aplikace.

Jedním z nich je zobrazení nejčastěji hledaných výrazů na vyhledávací obrazovce, které by sloužily k inspiraci ostatních uživatelů.

Další možnou funkcionalitou by mohla být možnost fotografovat a následně ukládat svá výsledná díla. Tato díla by si uživatelé mohli navzájem hodnotit a následně by se na hlavní obrazovce mohla zobrazovat ta nejlépe hodnocená.

Mezi galeriemi na hlavní obrazovce by se taktéž mohly zobrazovat nejoblíbenější či nejvíce obkreslované obrázky.

Závěr

Cílem této práce bylo vytvořit mobilní aplikaci pro zařízení s operačním systémem iOS, která usnadní uživateli obkreslování obrázků. K uskutečnění tohoto cíle bylo nutné nastudovat a zanalyzovat možnosti vývoje aplikací využívajících rozšířenou realitu pro operační systém iOS a následně zanalyzovat již existující aplikace. Na základě těchto znalostí bylo realizováno dotazníkové šetření a zanalyzovány požadavky cílové skupiny. Z výsledků byl vytvořen návrh, jenž zahrnoval uživatelské rozhraní, funkční požadavky aplikace a splňoval nefunkční požadavky definované Human Interface Guidelines. Implementovaná aplikace byla průběžně testována na funkčnost a poté byla podrobena uživatelskému testování na několika mobilních zařízeních pomocí pěti různých scénářů. Při testování nebyly zjištěny žádné větší nedostatky.

Výsledkem je mobilní aplikace napsaná v jazyce Swift. K práci s rozšířenou realitou využívá framework ARKit a k implementaci uživatelského rozhraní framework SwiftUI. Uživatel může vybraný obrázek upravit pomocí několika různých filtrů a promítnout jej na horizontální i vertikální plochy. S obrázkem lze následně interagovat – měnit jeho velikost, natočení, polohu či průhlednost. Uživatel si může vybírat mezi obrázky z vlastní galerie či galerií poskytovaných aplikací. Obrázky je možné vyhledávat taktéž pomocí výrazů. Aplikace k získávání obrázků komunikuje s Unsplash API a aplikuje zásady jeho použití.

Mezi možná vylepšení patří přidání nejčastěji vyhledávaných výrazů k inspiraci ostatních uživatelů či možnost zaznamenání výsledných děl. V budoucnu by taktéž mohl vzniknout žebříček nejoblíbenějších či nejvíce obkreslovaných obrázků. Nad aplikací by tak mohla vyrůst komunita, která si navzájem svá díla hodnotí a má možnost se zúčastnit nejrůznějších soutěží. Výsledky těchto soutěží by mohly být zobrazeny na hlavní obrazovce mezi současnými galeriemi.

Bibliografie

1. AZUMA, Ronald T. A survey of augmented reality. In: [online]. 1997, sv. 6, s. 355–385 [cit. 2022-04-21]. Č. 4. ISSN 10547460. Dostupné z DOI: <https://doi.org/10.1162/pres.1997.6.4.355>.
2. RADU, Iulian. Augmented reality in education: a meta-review and cross-media analysis. In: 2014, sv. 18, s. 1533–1543. Č. 6. Dostupné z DOI: <https://doi.org/10.1007/s00779-013-0747-y>. Copyright - Springer-Verlag London 2014; Poslední aktualizace - 2014-09-11.
3. NOMURA, Tsutomu; MAMADA, Yasuhiro; NAKAMURA, Yoshiharu; MATSUTANI, Takeshi; HAGIWARA, Nobutoshi; FUJITA, Isturo; MIZUGUCHI, Yoshiaki; FUJIKURA, Terumichi; MIYASHITA, Masao; UCHIDA, Eiji. Laparoscopic skill improvement after virtual reality simulator training in medical students as assessed by augmented reality simulator. In: Wiley, 2015, sv. 8, s. 408–412. Č. 4. Dostupné z DOI: <https://doi.org/10.1111/ases.12209>.
4. PL@NTNET. *Pl@ntNet*. Home [online] [cit. 2022-04-22]. Dostupné z: <https://plantnet.org/en/>.
5. GUY'S AND ST THOMAS' NHS FOUNDATION TRUST. *Education at Guy's and St Thomas'*. Laparoscopic Virtual Reality Simulators [online] [cit. 2022-04-22]. Dostupné z: <https://www.guysandstthomaseducation.com/project/laparoscopic-virtual-reality-simulators/>.
6. WESARG, Stefan; FIRLE, Evelyn A.; SCHWALD, Bernd; SEIBERT, Helmut; ZOGAL, Pawel; ROEDDIGER, Sandra. Accuracy of needle implantation in brachytherapy using a medical AR system: a phantom study. In: JR., Robert L. Galloway (ed.). SPIE, 2004, sv. 5367, s. 341–352. Dostupné z DOI: <https://doi.org/10.1117/12.535415>.
7. AYOUBI, Ayda. *ARCHITECT*. IKEA Launches Augmented Reality Application [online] [cit. 2022-04-22]. Dostupné z: <https://www.>

- architectmagazine.com/technology/ikea-launches-augmented-reality-application.
8. IKEA. *IKEA*. Say hey to IKEA Place [online] [cit. 2022-04-22]. Dostupné z: <https://www.ikea.com/au/en/customer-service/mobile-apps/say-hey-to-ikea-place-pub1f8af050>.
 9. INDIAMETRICS. *IndiaMetrics*. 7 Augmented Reality Use Cases in Architecture and Construction [online] [cit. 2022-04-21]. Dostupné z: <https://indiametrics.com/7-augmented-reality-use-cases-in-architecture-and-construction/>.
 10. MARR, Bernard. *Forbes*. The Amazing Ways VR And AR Are Transforming The Travel Industry [online] [cit. 2022-04-21]. Dostupné z: <https://www.forbes.com/sites/bernardmarr/2021/04/12/the-amazing-ways-vr-and-ar-are-transforming-the-travel-industry/?sh=4c8295ac36e0>.
 11. ACHARYA, Abhay. *Honest Pros and Cons*. 10 Pros and Cons of Augmented Reality [online] [cit. 2022-04-21]. Dostupné z: <https://honestproscons.com/pros-and-cons-of-augmented-reality/>.
 12. KEATS, Jonathon. *WIRED*. Snapchat Dysmorphia and the Real Dangers of Perceived Flaws [online] [cit. 2022-04-21]. Dostupné z: <https://www.wired.com/story/snapchat-dysmorphia-real-dangers-perceived-flaws/>.
 13. RAMOS, Diana. *LatinAmerican Post*. Snapchat filters: The enemy of mental health: A study from Boston University links the use of filters with a new phenomenon of body dysmorphia highly harmful to mental health [online] [cit. 2022-04-21]. Dostupné z: <https://latinamericanpost.com/22897-snapchat-filters-the-enemy-of-mental-health>.
 14. JELIČ, Pavel. *Letem světem Applem*. LiDAR senzor iPadu Pro: Vše, co jste o něm chtěli vědět [online] [cit. 2022-04-21]. Dostupné z: <https://www.letemsvetemapplem.eu/2020/03/18/lidar-ipadu-pro/>.
 15. GOOGLE DEVELOPERS. *ARCore: Documentation*. Use Depth in your Unity app [online] [cit. 2022-04-22]. Dostupné z: <https://developers.google.com/ar/develop/unity/depth/developer-guide>.
 16. MATHEW, Neil. *HACKERNOON*. Why is Occlusion in Augmented Reality So Hard? [Online] [cit. 2022-04-21]. Dostupné z: <https://hackernoon.com/why-is-occlusion-in-augmented-reality-so-hard-7bc8041607f9>.
 17. FABRY, Thomas; SMEETS, Dirk; VANDERMEULEN, Dirk. Surface representations for 3D face recognition. In: *IntechOpen*, 2010, kap. 16, s. 275. ISBN 978-953-307-060-5. Dostupné z DOI: <https://doi.org/10.5772/8951>.

18. UNITY TECHNOLOGIES. *Unity documentation*. HoloLens Spatial Mapping [online] [cit. 2022-04-22]. Dostupné z: <https://docs.unity3d.com/2018.4/Documentation/Manual/SpatialMapping.html>.
19. KUMAR, Pavan. *Selerio*. Occlusion is important in AR [online] [cit. 2022-04-21]. Dostupné z: <https://medium.com/selerio/occlusion-is-important-in-ar-9a672ff9ca2d>.
20. APPLE INC. *Apple*. IOS 11 brings powerful new features to iPhone and iPad this fall [online] [cit. 2022-04-22]. Dostupné z: <https://www.apple.com/newsroom/2017/06/ios-11-brings-new-features-to-iphone-and-ipad-this-fall/>.
21. APPLE INC. *Apple Developer: Documentation*. Verifying Device Support and User Permission [online] [cit. 2022-04-22]. Dostupné z: https://developer.apple.com/documentation/arkit/verifying_device_support_and_user_permission.
22. HOLLAND, Janet L; LEE, Sungwoong; DAOUK, Mohammad; AGBAJI, Daniel A. Higher education teaching and learning with augmented reality. In: *Handbook of Research on Fostering Student Engagement With Instructional Technology in Higher Education*. IGI Global, 2020, s. 229–248. Dostupné z DOI: <https://doi.org/10.4018/978-1-7998-0119-1.ch013>.
23. SCARAMUZZA, Davide; ZHANG, Zichao. Visual-Inertial Odometry of Aerial Robots. In: 2019, sv. abs/1906.03289. Dostupné z arXiv: 1906.03289.
24. VEATIVE LABS. *Veative*. Degrees of Freedom... 3DoF vs 6DoF [online] [cit. 2022-04-22]. Dostupné z: <https://veative.com/degrees-of-freedom-3dof-vs-6dof/>.
25. APPLE INC. *Apple Developer: Documentation*. ARLightEstimate [online] [cit. 2022-04-22]. Dostupné z: <https://developer.apple.com/documentation/arkit/arlightestimate>.
26. APPLE INC. *Apple Developer: Documentation*. SceneKit [online] [cit. 2022-04-22]. Dostupné z: <https://developer.apple.com/documentation/scenekit/>.
27. APPLE INC. *Apple Developer*. SpriteKit [online] [cit. 2022-04-22]. Dostupné z: <https://developer.apple.com/spritekit/>.
28. APPLE INC. *Apple Developer: Documentation*. RealityKit [online] [cit. 2022-04-21]. Dostupné z: <https://developer.apple.com/documentation/realitykit>.
29. APPLE INC. *Apple Developer*. Metal [online] [cit. 2022-04-22]. Dostupné z: <https://developer.apple.com/documentation/metal/>.

30. LANHAM, Micheal. *Learn ARCore - Fundamentals of Google ARCore: Learn to build augmented reality apps for Android, Unity, and the web with Google ARCore 1.0*. Packt Publishing, 2018. ISBN 9781788830409.
31. THE UNIVERSITY OF WAIKATO TE WHARE WĀNANGA O WAIKATO. *Science Learning Hub Pokapū Akoranga Pūtaiao*. Depth perception [online] [cit. 2022-04-22]. Dostupné z: <https://www.sciencelearn.org.nz/resources/51-depth-perception>.
32. CUIPIX. *CUIPIX*. How to Take 360° Photos at a Fixed NPP and Manually Stitch Them [online] [cit. 2022-04-22]. Dostupné z: <http://blog.cupix.com/take-360-photos-fixed-npp-manually-stitch/>.
33. GOOGLE DEVELOPERS. *ARCore: Documentation*. Fundamental concepts [online] [cit. 2022-04-21]. Dostupné z: <https://developers.google.com/ar/develop/fundamentals>.
34. ANDRÝSEK, Richard. *GAMESDEV*. TOP 5 Herních enginů pro indie vývojáře [online] [cit. 2022-04-22]. Dostupné z: <https://gamesdev.cz/top-5-hernich-enginu-pro-indie-vyvojare>.
35. MAKAROV, Andrew. *ARPost*. ARKit vs ARCore: Image Detection and Tracking [online] [cit. 2022-04-22]. Dostupné z: <https://arpost.co/2020/06/12/arkit-vs-arcore-image-detection-and-tracking/>.
36. MOBIDEV. *Youtube*. ARKit vs ARCore: "Recognize and Augment Images" feature comparison [online] [cit. 2022-04-22]. Dostupné z: <https://www.youtube.com/watch?v=uVW40FtMOAY>.
37. MACRUMORS STAFF. *MacRumors*. iPadOS 14 [online] [cit. 2022-04-21]. Dostupné z: <https://www.macrumors.com/roundup/ipados-14/>.
38. MACRUMORS STAFF. *MacRumors*. iOS 15 [online] [cit. 2022-04-21]. Dostupné z: <https://www.macrumors.com/roundup/ios-15/>.
39. APPLE INC. *Apple*. App Store [online] [cit. 2022-04-21]. Dostupné z: <https://www.apple.com/cz/app-store/>.
40. APPLE INC. *Apple Developer*. Enrollment [online] [cit. 2022-04-21]. Dostupné z: <https://developer.apple.com/support/enrollment/>.
41. APPLE INC. *Apple Developer*. Human Interface Guidelines [online] [cit. 2022-04-21]. Dostupné z: <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>.
42. SYROMIATNIKOV, Artem; WEYNS, Danny. A Journey through the Land of Model-View-Design Patterns. In: *2014 IEEE/IFIP Conference on Software Architecture*. 2014, s. 21–30. Dostupné z DOI: 10.1109/WICSA.2014.13.
43. MDN CONTRIBUTORS. *Mdn*. MVC [online] [cit. 2022-04-22]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>.

-
44. TECHTARGET CONTRIBUTOR. *Model-View-ViewModel (MVVM)* [online] [cit. 2022-04-21]. Dostupné z: <https://www.techtarget.com/whatis/definition/Model-View-ViewModel>.
 45. BARKER, Chris. *Learn SwiftUI*. Packt Publishing, Limited, 2020. ISBN 9781839215421.
 46. APPLE INC. *Apple Developer*. Xcode [online] [cit. 2022-04-21]. Dostupné z: <https://developer.apple.com/xcode/features/>.
 47. LATTNER, Chris. *Chris Lattner's Homepage*. Chris Lattner's Homepage [online] [cit. 2022-04-21]. Dostupné z: <https://nondot.org/sabre/>.
 48. APPLE INC. *Swift*. About Swift [online] [cit. 2022-04-21]. Dostupné z: <https://www.swift.org/about/>.
 49. APPLE INC. *Apple Developer: Documentation*. Optional [online] [cit. 2022-04-21]. Dostupné z: <https://developer.apple.com/documentation/swift/optional>.
 50. TECH NEWS 24H. *Tech News*. InkHunter - Try tattoos in real-time with augmented reality (Video) [online] [cit. 2022-04-22]. Dostupné z: <https://www.technews24h.com/2016/04/inkhunter-try-tattoos-in-real-time-with.html>.
 51. JURDI, Shadi. *Etsy*. Introducing Augmented Reality in the Etsy App [online] [cit. 2022-04-22]. Dostupné z: <https://www.etsy.com/seller-handbook/article/introducing-augmented-reality-in-the/854585081862>.
 52. SKETCHAR INC. *SketchAR*. Using deep tech to enable more creativity in people [online] [cit. 2022-04-22]. Dostupné z: <https://sketchar.io/technology/>.
 53. CAHILL, Bear. ObservableObjects. In: *UI Design for iOS App Development: Using SwiftUI*. Berkeley, CA: Apress, 2021, s. 63–74. ISBN 978-1-4842-6449-2. Dostupné z DOI: 10.1007/978-1-4842-6449-2_5.

Seznam použitých zkratk

3DOF Three degrees of freedom

6DOF Six degrees of freedom

AR Augmented reality

iOS iPhone operating system

MVC Model-View-Controller

MVVM Model-View-ViewModel

WWDC Worldwide Developers Conference

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
application	
├── tests.pdf	výsledky testování
└── video.mp4	video s průchodem aplikací
src	
├── impl	zdrojové kódy implementace
└── thesis	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
text	
└── thesis.pdf	text práce ve formátu PDF