



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ROZŠÍŘENÁ REALITA: HISTORICKÉ VERZE BUDOVY**

AUGMENTED REALITY: HISTORICAL VERSIONS OF A BUILDING

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. JURAJ MEDVEC**

**VEDOUcí PRÁCE**

SUPERVISOR

**prof. Ing. ADAM HEROUT, Ph.D.**

BRNO 2019

## Zadání diplomové práce



21643

Student: **Medvec Juraj, Bc.**  
Program: Informační technologie    Obor: Počítačová grafika a multimédia  
Název: **Rozšířená realita: Historické verze budovy**  
**Augmented Reality: Historical Versions of a Building**  
Kategorie: Počítačová grafika

Zadání:

1. Seznamte se s problematikou rozšířené reality na mobilních telefonech.
2. Experimentujte s nástroji a dílčími prvky řešení, které umožní přes reálnou budovu zobrazit její historické podoby jako 3D modely.
3. Testujte zkoumané prvky uživatelského rozhraní a rozšířené reality na uživateliích a iterativně své dílčí řešení vylepšujte.
4. Navrhněte a implementujte řešenou aplikaci, která přes reálnou budovu umožní zobrazit její historické verze jako 3D modely.
5. Testujte řešenou aplikaci na uživateliích a iterativně ji vylepšujte.
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Dieter Schmalstieg, Tobias Hollerer: Augmented Reality: Principles and Practice, ISBN: 978-0321883575
- Steve Krug: Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, ISBN-13: 978-0321965516
- Steve Krug: Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability, ISBN-13: 978-0321657299
- Susan M. Weinschenk: 100 věcí, které by měl každý designér vědět o lidech, Computer Press, Brno 2012

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 3, rozpracování bodů 4 a 5.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**

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

Datum zadání: 1. listopadu 2018

Datum odevzdání: 22. května 2019

Datum schválení: 1. listopadu 2018

## Abstrakt

Táto práca je zameraná na prácu s rozšírenou realitou v mobilných aplikáciach. Cieľom práce bolo vytvorenie mobilnej aplikácie, ktorá poskytuje jej užívateľom pohľad do minulosti vybraných historických budov. Na začiatku práce sú čitateľovi predstavené jednotlivé metódy a technológie pre prácu s rozšírenou realitou. Na základe získaných informácií je navrhnutá a implementovaná mobilná aplikácia. Táto aplikácia umožňuje jej užívateľom zobrazovať historickú podobu budov.

## Abstract

This thesis is focused on Augmented reality in mobile applications. The aim of the thesis is to create mobile application that provides its users with the view of the past of selected buildings. At the beginning of the thesis, the reader is presented with various methods and technologies for working with augmented reality. Based on the theories obtained in first chapters, a mobile application was designed and implemented. The final mobile application allows its users to view historical buildings.

## Kľúčové slová

Rozšírená realita, Mobilné Aplikácie, Swift, iOS, ARKit, Firebase

## Keywords

Augmented reality, Mobile Apps, Swift, iOS, ARKit, Firebase

## Citácia

MEDVEC, Juraj. *Rozšírená realita: Historické verze budovy*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

# Rozšířená realita: Historické verze budovy

## Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána prof. Ing. Adama Herouta. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....  
Juraj Medvec  
21. mája 2019

## Podakovanie

Chcel by som poďakovať vedúcemu diplomovej práce pánovi prof. Ing. Adam Heroutovi, Ph.D. za odborné vedenie pri vypracovávaní tejto diplomovej práce. Ďalej by som chcel poďakovať svojej priateľke a mojím rodičom za morálnu podporu, ktorou ma zahŕňali počas celého obdobia práce. Nakoniec by som chcel poďakovať priateľom, na ktorých som sa mohol vždy spoľahnúť počas celého štúdia.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Rozšírená realita</b>	<b>3</b>
2.1	Potenciál rozšírenej reality . . . . .	5
2.2	Najznámejšie aplikácie . . . . .	7
2.3	Existujúce platformy . . . . .	9
2.4	Visual inertial Odometry . . . . .	9
<b>3</b>	<b>Úvod do knižnice ARKit</b>	<b>12</b>
3.1	Princíp fungovania ARKitu . . . . .	12
3.2	Doporučenia pre prácu s knižnicou Arkit . . . . .	14
3.3	Programovanie v jazyku Swift . . . . .	16
<b>4</b>	<b>Návrh aplikácie pre zobrazovanie historických budov</b>	<b>17</b>
4.1	Prínos navrhovaného riešenia . . . . .	19
4.2	Požiadavky na funkcionality aplikácie . . . . .	19
4.3	Grafické rozhranie aplikácie . . . . .	25
<b>5</b>	<b>Implementácia aplikácie</b>	<b>31</b>
5.1	Technický popis aplikácie . . . . .	31
5.2	Výpočet správnej polohy vkladanejho objektu . . . . .	32
5.3	Interakcia s virtuálnymi objektami . . . . .	35
5.4	Ukladanie 3D modelov na zariadení užívateľa . . . . .	38
<b>6</b>	<b>Testovanie výslednej aplikácie</b>	<b>43</b>
6.1	Vytvorenie testovacej sady objektov . . . . .	43
6.2	Návrh testovacieho scenáru . . . . .	43
6.3	Testovanie aplikácie na koncových užívateľoch . . . . .	44
6.4	Vyhodnotenie testov . . . . .	45
<b>7</b>	<b>Záver</b>	<b>46</b>
	<b>Literatúra</b>	<b>47</b>
<b>A</b>	<b>Obsah priloženého pamäťového média</b>	<b>49</b>
<b>B</b>	<b>Plagát</b>	<b>50</b>

# Kapitola 1

## Úvod

Cieľom tejto diplomovej práce je vytvorenie mobilnej aplikácie, ktorá bude zobrazovať historickú podobu budov alebo iných objektov v našom okolí. Pre zobrazovanie 3D modelov bude výsledná aplikácia využívať technológiu rozšírenej reality. V dnešnej dobe existuje niekoľko nástrojov, ktoré sú určené pre mobilné aplikácie pracujúce s rozšírenou realitou. Samotná technológia bola detailnejšie špecifikovaná v roku 1997 *Ronaldom T. Azumom* [4]. Z toho vyplýva, že sa nejedná o novú technológiu. V posledných rokoch sa ale dostáva čoraz viac do povedomia ľudí, čo je zapríčinené mobilnými zariadeniami, ktoré čoraz viac podporujú danú technológiu.

V úvodných kapitolách bude čitateľovi detailnejšie predstavená technológia rozšírenej reality. S tým, ako sa daná technológia dostáva čoraz viac do povedomia ľudí, rozširuje sa aj jej využitie naprieč rôznymi odvetvami. Z pomedzi existujúcich frameworkov určených pre prácu s rozšírenou realitou bol pre implementáciu navrhovanej aplikácie vybraný *ARKit*. Jedná sa o framework vyvinutý spoločnosťou Apple. Daná knižnica poskytuje vývojárom funkcionalitu, ktorá značne uľahčuje prácu s rozšírenou realitou. Jediným obmedzením danej knižnice je podpora operačných systémov iOS. *ARKit* je podporovaný len na systémoch iOS 11 a novších.

V kapitole určenej návrhu aplikácie budú detailnejšie popísané jednotlivé požiadavky na systém. Medzi tieto požiadavky sa radí napríklad presnosť vloženia alebo možnosť úpravy polohy 3D objektu zobrazeného v rozšírenej realite. Pri počítaní výslednej polohy objektu bude aplikácia využívať GPS polohu užívateľa. Samozrejme nepresnosť GPS súradníc alebo samotného výpočtu môže spôsobovať nepresné zobrazenie 3D objektu v rozšírenej realite. Z toho dôvodu je potrebné, aby aplikácia poskytovala užívateľovi možnosť úpravy pozície 3D objektu.

Samotná implementácia navrhovanej aplikácie je realizovaná na operačnom systéme iOS 12 zariadenia tablet Apple. V kapitole venovanej vývoju aplikácie sú detailnejšie popísané jednotlivé funkcie využité k implementácii požiadavkov na systém popísaných v návrhu aplikácie. Výsledná aplikácia bola následne testovaná na koncových užívateľoch, pričom testovanie bolo zamerané na orientáciu užívateľa v grafickom rozhraní aplikácie.

## Kapitola 2

# Rozšírená realita

Pojem Rozšírená realita sa v dnešnej dobe často mieša s pojmom Virtuálna Realita. Je to z toho dôvodu, že obe technológie sú si určitým spôsobom podobné, no predsa zásadne odlišné. Základný rozdiel je v tom, že rozšírená realita pridáva do reality virtuálne objekty, ako je možné vidieť na obrázku 2.7. Zatiaľ čo virtuálna realita zobrazuje len virtuálne prostredie, ktoré nie je prepojené s realitou.



Obr. 2.1: Princíp Rozšírenej reality, kedy video zachytené kamerou obsahuje virtuálne objekty, pričom tieto objekty sú vhodným spôsobom vsadené do reality [zdroj: [4]].

Na to, aby mohlo byť povedané, že daná technológia predstavuje rozšírenú realitu, musia byť splnené tri základné podmienky:

- **kombinovanie virtuálnych a reálnych vecí**
- **práca v realtime**
- **registrovanie 3D okolia**

Základné podmienky definície AR<sup>1</sup> systému nezachytávajú všetky prípady systémov, ktoré pracujú s rozšírenou realitou. Môžeme ale vidieť, že kompletný AR systém potrebuje aspoň tri základné komponenty : **sledovaciú komponentu** (kameru), **registrovaciu komponentu** a **vizualizačnú komponentu** (displej). V systéme sa môže vyskytovať štvrtá komponenta, ktoré nie je tak kľúčová pre samotný systém ako prvé tri. Táto časť sa nazýva priestorový model. Môže sa jednať napríklad o istú formu databázy, kde sa ukladajú informácie o skutočnom svete a informácie o virtuálnom svete. Pre správne fungovanie systému je veľmi dôležité, aby informácie o oboch svetoch boli uložené v rovnakej koordinačnej sústave.

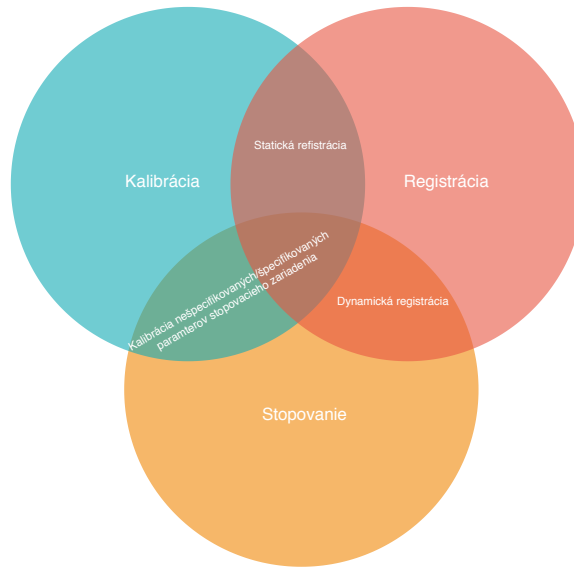
V kontexte rozšírenej reality sa stretávame s tromi dôležitými pojmami, ktoré sú spájané s meraním a zarovnaním objektov. Medzi tieto pojmy sa radia:

- **Kalibrácia:** jedná sa o proces porovnávania výsledkov dosiahnutých referenčným a kalibrovacím zariadením. Pričom referenčné zariadenie môže byť nahradené známou referenčnou hodnotou.
- **Registrácia:** pri práci s rozšírenou realitou znamená registrácia odkaz na zarovnanie koordinačných systémov medzi virtuálnym a reálnym svetom.
- **Stopovanie:** je pojem, ktorý je používaný k popisu dynamického snímania a počítania systému rozšírenej reality. Na to, aby bol niekto schopný zobraziť virtuálne objekty musím vedieť aspoň približnú pozíciu a orientáciu displeja voči reálnemu svetu.[17]

---

<sup>1</sup>Augmented Reality





Obr. 2.2: Systém rozšírenej reality pozostáva z troch dôležitých častí (stopovanie, registrácia, kalibrácia), ktoré sú navzájom prepojené [Inšpirované z [4]].

## 2.1 Potenciál rozšírenej reality

Rozšírená realita sa v poslednej dobe dostáva čoraz viac do povedomia ľudí. To je z časti zapríčinené rastúcou popularitou mobilných zariadení podporujúcich práve prácu s rozšírenou realitou. Vkladaním virtuálnych objektov do reality poskytuje táto technológia využitie v praxi naprieč širokým spektrom odvetví.

### 2.1.1 Využitie v školstve

Rozšírená realita môže byť veľkým pomocníkom pri vzdelávacom systéme, pretože dokáže prostredníctvom učenia založenom na aktivitách naučiť radu rôznych predmetov. Interaktívne steny, outdoorové aktivity sú len jedny z veľa iných spôsobov ako môžu pedagógovia prinútiť celú triedu k aktívnemu zbieraniu informácií.

Jednou z takýchto aplikácií je Blippar, ktorý poskytuje svojim užívateľom jednoduchý prístup k rôznorodým informáciám. Blippar taktiež uľahčuje učenie zložené na aktivitách tým, že ponúka študentom priateľské pomôcky a jednoduchý prístup k informáciám v rámci skupiny. Ako môžeme vidieť na obrázku 2.3, tak aplikácia poskytuje vizuálne zobrazenie objektov, ktoré sú pre študentov niekedy ťažko predstaviteľné ako je to napríklad snečná sústava [11].



Obr. 2.3: Úkážka využitia rozšírenej reality vo vzdelávaní [zdroj: [11]].

### 2.1.2 Využitie v priemysle

V priemyselných odvetviach najmä ako spracovateľský priemysel, si zamestnanci nemôžu dovoliť urobiť chybu, pretože táto chyba by ich mohla stáť veľmi veľa peňazí a času.

Práve rozšírená realita je správnym riešením tohto problému. Táto technológia s potenciálom digitalizácie prototypových produktov v 3D uľahčuje prístup k prototypu, ako je to možné vidieť na obrázku 2.4 a jeho následnej manipulácii alebo jeho pochopení. Vedúci tímov môžu tak správne identifikovať problémy alebo hrozby a na základe týchto informácií robiť správne rozhodnutia. To zvyšuje rýchlosť procesu spolu s mierou efektivity, čo v konečnom dôsledku zvyšuje dosiahnutý zisk [1].

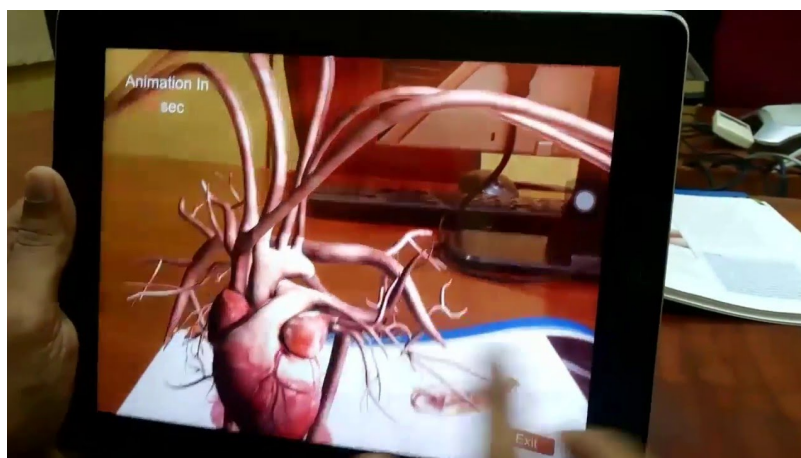


Obr. 2.4: Ukážka praktického využitia rozšírenej reality v priemysle [zdroj: [14]].

### 2.1.3 Využitie v zdravotníctve

Tak isto ako technológia umelej inteligencie alebo strojového učenia, tak aj rozšírená realita prináša nové možnosti do sveta zdravotníctva. Táto technológia rozkladá jednotlivé komplexné medicínske postupy na interaktívne 3D formy. Týmto spôsobom umožňuje zdra-

votníkom všetko jednoducho popísať svojim pacientom. Ďalším veľkým prínosom je zobrazovanie jednotlivých orgánov v 3D modeloch. To znamená, že študenti budú mať možnosť lepšie pochopiť anatómiu jednotlivých orgánov v ľudskom tele [1].



Obr. 2.5: Ukážka praktického využitia rozšírenej reality v medicíne [zdroj: [5]].

## 2.2 Najznámejšie aplikácie

Keďže témou tejto práce je vytvorenie aplikácie pre vkladanie modelov historických budov do reality, tak pri analýze boli vynechané, hry ktoré pracujú s rozšírenou realitou a práca bola zameraná na aplikácie, ktoré majú buď vzdelávací smer alebo sa zaoberajú vkladáním modelov reálnych vecí do reality.

### 2.2.1 BBC Civilisations

Jedná sa o aplikáciu od televízneho kanálu BBC. Táto aplikácia bola vytvorená za účelom vzdelávať ľudí pomocou rozšírenej reality. Využíva detekciu okolia, pričom hľadá rovné plochy, na ktorých neskôr zobrazuje virtuálny objekt zeme. Táto aplikácia dosahuje veľmi dobré recenzie na ApStore ako aj na GoodlePlay. Nevýhodou tejto aplikácie je to, že neexistuje verzia, ktorá by bola neplatená. Ak ju chce užívateľ používať, musí za ňu zaplatiť. [8]



Obr. 2.6: Vizualizácie aplikácie BBC Civilizations [zdroj: [6]].

### 2.2.2 Ikea Place

Ikea Place je aplikácia, ktorá umožňuje jej užívateľom vkladať virtuálne objekty v podobe ikea nábytku do reality, čím umožňuje zobrazíť vizuálnu stránku miestnosti s vybraným nábytkom. Táto aplikácia dosahuje priemerných hodnotení na AppStore a GooglePlay. Užívatelia sa najviac sťažujú na nesprávny rozmer vkladaných objektov, ktorý môže byť zapríčinený nesprávnou detekciou priestoru.[12]



Obr. 2.7: Vizualizácie aplikácie Ikea Place [zdroj: [12]].

## 2.3 Existujúce platformy

V dnešnej dobe existuje celý rad frameworkov, ktoré zľahčujú prácu s rozšírenou realitou. Tieto frameworky sa v rôznych oblastiach líšia, no jedno majú spoločné. Všetky sú zamerané na mobilné zariadenia. Z toho dôvodu existujú dva typy frameworkov, jeden ktorý je určený len pre jeden typ mobilných operačných systémov a druhý ktorý je podporovaný viacerými mobilnými platformami[7].

### 2.3.1 ARKit

Jedná sa o framework, ktorý bol vyvinutý spoločnosťou Apple. Je určený výhradne pre zariadenia od danej spoločnosti. Bol vytvorený za cieľom jednoduchšej práce s rozšírenou realitou. Má pomáhať developerom pri vyvíjaní mobilných aplikácií alebo hier. Bohužiaľ podporuje len operačné systémy iOS 11 a iOS 12. Jeho hlbšia analýza sa bude nachádzať v ďalších kapitolách [7].

### 2.3.2 CoreKit

CoreKit bol vytvorený spoločnosťou Google ako odpoveď na framework vytvorený od Apple. Najohromujúcejšia vec na tomto frameworku je to, že podporuje obe Android a iOS platformy. Je založený na dvoch veciach. Prvá je detekovanie real-time pozície a tá druhá je integrácia virtuálnych a reálnych objektov[7].

### 2.3.3 Vuforia

Jedná sa o framework, ktorý podporuje radu platforiem ako napríklad iOS, Android, UWP a Unity Editor. Radí sa medzi jeden z najpopulárnejších frameworkov pre prácu s rozšírenou realitou. Implementuje nasledujúcu funkcionálnosť :

- rozpoznávanie rôznych objektov
- rozpoznávanie textu
- rozpoznávanie prostredia

Tento framework je voľne dostupný. Pri jeho používaní má len jednu nevýhodu, že pri jeho používaní sa na obrazovke zobrazujú vodové znaky Vuforia[7].

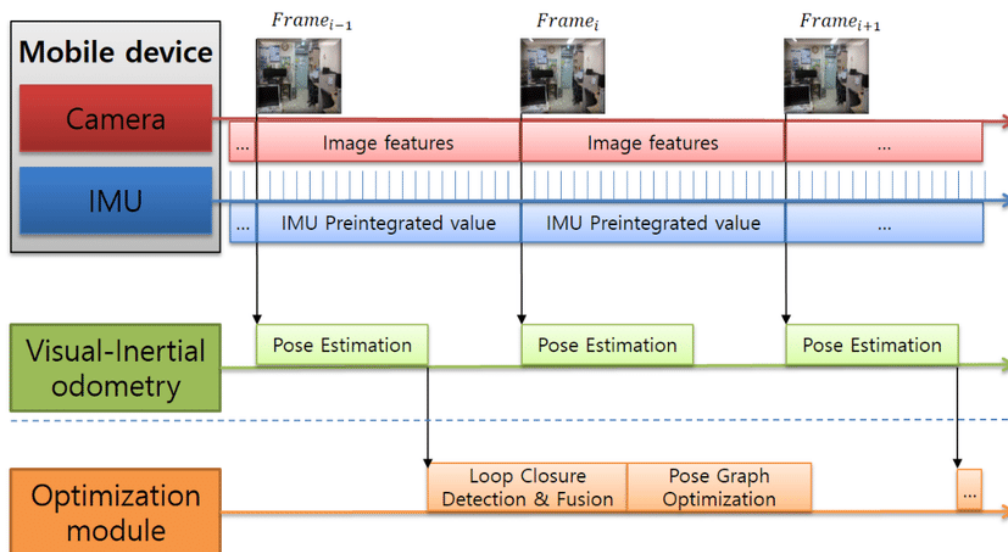
## 2.4 Visual inertial Odometry

Visual inertial Odometry (VIO) je bežne používaný proces orientácie sa v priestore, ktorý je používaný robotickými zariadeniami. S jeho praktickým použitím sa stretávame aj pri používaní frameworku ARKit od firmy Apple. Pri orientácii v priestore pracuje VIO s informáciami z rôznych zariadení ako napríklad : Akcelerometer, GPS, Systémy pre spracovanie obrazu.

Najčastejšie používaným systémom, ktorý sa využíva pri spracovaní obrazu a orientácii v priestore je algoritmus Simultaneous Location and Mapping (SLAM), ktorý sa niekedy

nazýva **visual odometry**. Funguje ako kontinuálne stopovanie 6DOF relatívnej pozície kamery k počiatočnej pozícii. Vizuálna odometria počíta s 3D rekonštrukciou prostredia, ale k tomu používa len podporu inkrementálneho stopovania. Základný proces SLAM je možné vyjadriť neformálne pomocou krokov [17]:

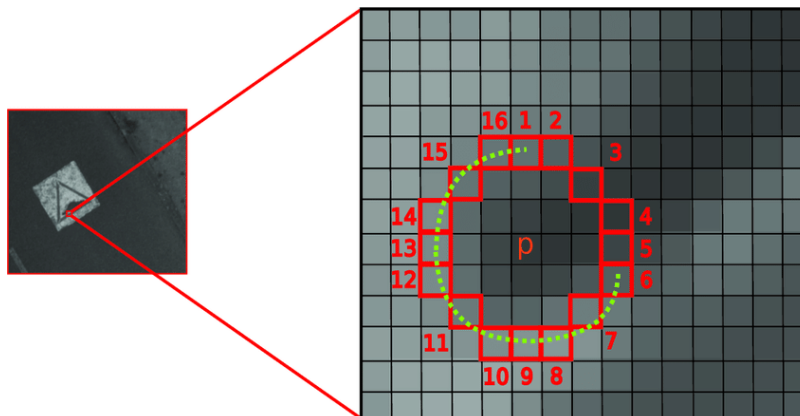
- Detekovanie bodov záujmu (Keypoints): napríklad pomocou Harris alebo FAST rohov
- Sledovanie bodu záujmu v 2D z predošlého snímku, použitím napríklad metódy KTL (Kabade-Lucas-Tomasi Tracking). Určenie základnej matice medzi aktuálnou a predošlou pozíciou z vlastnosti korešpondujúcej na 5-bodový algoritmus (Five-Point Algorithm) pre základnú maticu v rámci metódy RANSAC)
- Prepočítanie inkrementálnej pozície kamery pomocou základnej matice
- Keďže základná matica určuje prekladovú časť pozície len do stupnice, musí byť táto stupnica odhadnutá oddelene, aby bola zachovaná konzistentnosť v sledovanej sekvencii obrazov. Na dosiahnutie tohto cieľa sú 3D lokalizačné body triangulované z viacerých pozorovateľských pozícií rovnakého obrazu. Tento princíp sa nazýva SFM (structure from motion)
- Získanie ďalšieho snímku



Obr. 2.8: Princíp fungovania Visual inertial odometry [zdroj: [18]].

### 2.4.1 FAST, Detekcia bodov záujmu (Keypoints)

V roku 2006 bola predstavená metóda Rosten a Dortmundom pre detekovanie bodov záujmu v obraze, ktorú nazvali **Feature from accelerated segment test**(FAST). Predstavená metóda je vhodná pre real-time spracovanie obrazu, pričom využíva diskretný kruh umiestnený v strede potencionálneho kandidáta. Kandidát alebo inak povedané vybraný bod je klasifikovaný ako roh, pokiaľ existuje spojitý oblúk pixelov výrazným kontrastom okolo centrálného pixelu, ktorý zaberá aspoň tretinu kruhu ako je to zobrazené na 2.9. Existuje viacero variant metódy FAST ako napríklad : FAST9, FAST10, FAST11 a FAST12[17].



Obr. 2.9: FAST - metóda pre detekciu bodov záujmu v obraze [zdroj: [16]].

Na obrázku 2.9 je možné vidieť, ako pracuje algoritmus FAST12. Táto varianta vyberá zo 16 pixelov kruhu okolo potenciálneho bodu, ktorý je reprezentovaný bodom  $p$ . Hrana je detekovaná, pokiaľ 12 za sebou idúcich pixelov je markantne svetlejších alebo tmavších ako stred  $p$ . Ako je možné vidieť v rovniciach 2.1, 2.2, 2.3, tak tmavé body sú reprezentované množinou  $S^D$  a svetlé body sú reprezentované množinou  $S^L$ [17].

$$S^D = \{s_i(x, y) | s_i(x, y) \leq I(x, y) - d\} \quad (2.1)$$

$$S^L = \{s_i(x, y) | s_i(x, y) \geq I(x, y) + d\} \quad (2.2)$$

$$p(x, y) = \max\left(\sum_{s_i \in S^D} |s_i - I(x, y)| - d, \sum_{s_i \in S^L} |s_i - I(x, y)| - d\right) \quad (2.3)$$

#### 2.4.2 KLT

Jedná sa o klasický prístup inkrementálneho stopovania, ktorý extrahuje body z počiatočného obrazu a potom ich stopuje pomocou možného toku. Stopovanie pomocou KLT znamená nájdenie parametrov  $\mathbf{x}$  z obalu  $\mathbf{w}$  ktorý transformoval vzorový obraz  $T$  do vstupného obrazu  $I$ . Obal bude často obmedzený na afinnú transformáciu, čo je dostatočné pre namodelovanie deformácie obrazovej časti pozorovanej po malom pohybu kamery [17].

#### 2.4.3 Five-point Algorithm for essential Matrix

Tento algoritmus sa používa na určenie relatívnej polohy kamery pomocou piatich rozličných korespondenčných bodov. Pre vypočítanie relatívnej polohy využíva SVD. Ohniská dvoch potrebných optických kamier sa musia líšiť, aby výsledok by relevantný [17].

## Kapitola 3

# Úvod do knižnice ARKit

Pre vytvorenie navrhovanej aplikácie bol vybraný framework *ARKit*, ktorý ako bolo uvedené vyššie bol vytvorený spoločnosťou Apple. Pre vytvorenie spojenia medzi reálnym a virtuálnym prostredím využíva *ARKit* technológiu, ktorá sa volá *Visual-inertial odometry*. Tento proces kombinuje informácie z pohybového zariadenia ako napríklad aceleometer a informácie získané zo spracovania obrazu, ktorý je dostupný z kamerového zariadenia. Metóda *Visual-inertial odometry* je detailnejšie popísaná v kapitole 2. *ARKit* rozpoznáva špecifické oblasti v obraze, sleduje ich a následne na základe toho sleduje zmeny polohy týchto vybraných bodov naprieč video snímkam.

*ARKit* podporujú len určité zariadenia s procesormi : A9, A10, A11. Nie je možné ho spustiť na notebooku alebo počítači. Je výhradne určený pre mobily a tablety od spoločnosti Apple. Vďaka knižnici *ARKit* vie aplikácia efektívne detekovať prostredie a orientovať sa v ňom. Ďalej umožňuje jednoduché vkladanie virtuálnych objektov do reality.

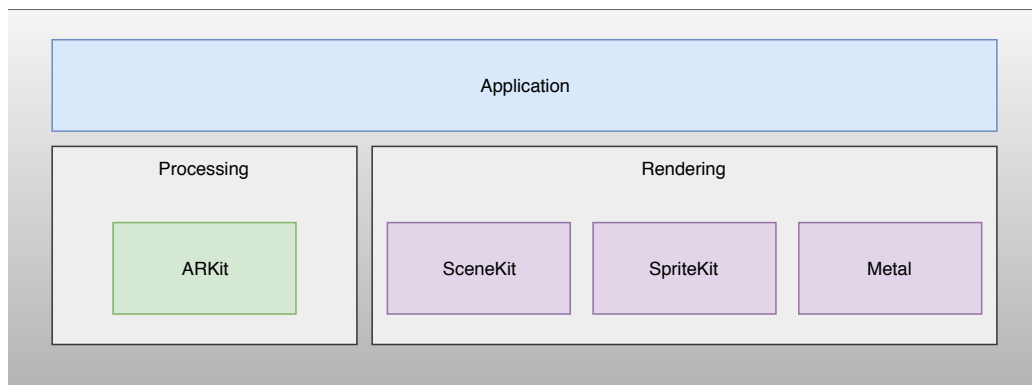
### 3.1 Princíp fungovania ARKitu

Samotný framework je možné rozdeliť do troch kategórií z ktorých pozostáva: *Tracking*, *Scene understanding*, *Rendering*. Každá jedna z týchto kategórií sa stará o to, aby zážitok z používania *ARKitu* bol čo najlepši.

*ARSession* [3] je základný objekt, ktorý spracuje vývojár pri práci s *ARKitom*. Tento objekt koordinuje hlavné procesy, ktoré *ARKit* vykonáva v užívateľovom mene, tak aby vytvoril skúsenosť s rozšírenou realitou. Medzi tieto procesy sa radí napríklad čítanie informácií z hardware-u snímania pohybu, ovládanie vstavaného fotoaparátu a analýza zaznamenaného obrazu. *ARSession* je zodpovedný za prepojenie reálneho sveta s tým virtuálnym, v ktorom sa zobrazujú virtuálne objekty. Spustenie relácie vyžaduje určitú konfiguráciu. Podtriedy abstraktnej triedy *ARConfiguration* určujú ako bude *ARKit* zaznamenávať pozíciu mobilného zariadenia v reálnom priestore.

*ARKit* používa pre vykresľovanie virtuálnych objektov do reality knižnice *SceneKit*, *SpriteKit*, *Metal* ako je to možné vidieť na obrázku 3.1. Knižnica *SceneKit* je využívaná na vykresľovanie 3D objektov, zatiaľ čo knižnica *SpriteKit* je využívaná na prácu s 2D objektami. Knižnica *Metal* slúži na prácu s vytváraním vlastného renderu. Všetky objekty, ktoré sú zobrazované *ARKitom* sú kontrolované objektom *ARSession*.

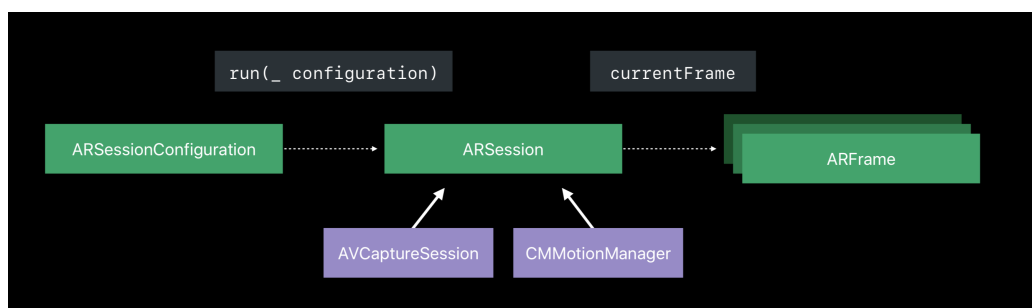




Obr. 3.1: Zobrazenie princípu práce knižnice ARKit. Na obrázku sú znázornené jednotlivé komponenty, ktoré využíva aplikácia pri vytváraní scény rozšírenej reality [Inšpirované z [13]].

Ako bolo uvedené vyššie, tak trieda *ARConfiguration* [2] je zodpovedná za nastavenie *ARSession*. Obrázok 3.2 reprezentuje ako abstraktná trieda *ARConfiguration* koordinuje vyššie spomínanú triedu *ARSession*. Pomocou tejto abstraktnej triedy nastavujeme parametre:

- **ARWorldTrackingConfiguration:** poskytuje vysokokvalitný zážitok s rozšírenou realitou, ktorý používa zadnú kameru zariadenia pre precízne detekovanie pozície a orientácie v priestore. Dovoľuje užívateľovi detekciu rovných plôch a osvetlenie virtuálnych objektov založené na reálnom prostredí.
- **AROrientationTrackingConfiguration:** poskytuje len detekciu polohy zariadenia v reálnom priestore.
- **ARImageTrackingConfiguration:** poskytuje základné detekovanie obrazu pomocou zadnej kamery.
- **ARFaceTrackingConfiguration-** poskytuje detekciu tváre pomocou prednej kamery zariadenia.
- **ARObjectScanningConfiguration:** poskytuje využitie zadnej kamery k zhromaždeniu priestorových údajov s vysokou kvalitou. Tieto informácie môžu byť použité k vytváraniu referenčných objektov.



Obr. 3.2: Popis návaznosti jednotlivých tried *ARSessionConfiguration*, *ARSession* a *ARFrame*. Obrázok zachytáva princíp fungovania knižnice ARKit [Prevzaté z [13]].

Obrázok 3.2 ďalej zobrazuje, ako jednotlivito na sebe závisia triedy *ARSessionConfiguration*, *ARSession* a *ARFrame*. Trieda *ARSessionConfiguration* vytvára instanciu triedy *ARSession*. Objekt tejto triedy existuje počas celej doby vykresľovanie virtuálnych objektov a pokiaľ chce užívateľ pridať nový objekt do scény, potrebuje k nemu pristúpiť. *ARSession* vykresľuje zobrazovanú scénu pomocou objektov *ARFrame*. Vďaka *ARFrame* dokáže aplikácia vykonávať hint testy, na základe ktorých určuje, či užívateľ mieri telefónom na virtuálny objekt alebo nie.

## 3.2 Doporučenia pre prácu s knižnicou ARKit

Apple pri predstavení frameworku *ARKit* poskytol svojim užívateľom rady, ako čo najlepšie pracovať s knižnicou *ARKit* a celkovo ako dobre navrhovať aplikácie pre prácu s rozšírenou realitou. Predtým, než človek začne vôbec niečo programovať a navrhovať, je dobré si prečítať aspoň zbežne rady, ktoré nám môžu ušetriť veľa času pri refactoringu kódu. Apple tieto rady rozdelil do niekoľkých kategórií podľa toho, na čo sa tieto rady zameriavajú.

### 3.2.1 Navrhovanie dizajnu aplikácie

Pri navrhovaní dizajnu aplikácie je potrebné dbať na nasledovné postupy:

**Používanie celého displeja** pre zobrazovanie virtuálnej scény. Nepreplniť obrazovku zbytočnosťami a nechať tam len najpodstatnejšie veci.

**Vytváranie presvedčivých ilúzií** pri umiestňovaní realistických objektov. Nie všetky ilúzie s virtuálnou realitou vyžadujú virtuálne objekty. Tie ktorý tak robia by mali zahŕňať len objekty, ktoré súvisia s prostredím do ktorého sú umiestňované. Je potrebné využívať dostupné prostriedky od knižnice *ARKit* k tomu, aby aplikácia správne vkladala 3D objekty do detekovaného prostredia.

**Zváženie ako sú virtuálne objekty s odrazovými plochami zobrazované.** Odrazy v knižnici *ARKit* sú aproximácie založené na prostredí zachytenom fotoaparátom. Mále reflexné povrchy pomáhajú udržaniu ilúzie.

**Používanie aplikácie len v prostrediach, ktoré nie sú optimálne pre AR.** Ľudia môžu aplikáciu otvoriť na mieste, kde nie je veľa priestoru. Preto treba uvažovať aj o sade funkcií, ktoré budú určené pre rôzne prostredia.

**Dbáť na pohodlie užívateľa.** Dlhodobé držanie zariadenia v určitej vzdialenosti alebo uhle môže byť únavné. Preto je potrebné zvážiť, ako musia ľudia držať zariadenie pri používaní navrhovanej aplikácie. Napríklad v predvolenom nastavení je možné vkladať objekty v takej vzdialenosti, ktorá znižuje potrebu presunúť zariadenie bližšie k objektu.

**Použite zvukovej a hmatovej spätnú väzbu na zlepšenie pohlcujúceho zážitku.** Zvukový efekt alebo vibrovanie je skvelý spôsob, ako potvrdiť, že virtuálny objekt dosiahol kontakt s fyzickým povrchom alebo iným virtuálnym objektom. V pohlcujúcej hre môže hudba na pozadí pomôcť obklopiť užívateľa vo virtuálnom svete.

**Dbajte na bezpečnosť používateľa.** Príliš veľa pohybu môže byť nebezpečné. Preto treba zvážiť spôsoby ako zabezpečiť bezpečnosť navrhovanej aplikácie. Napríklad je vhodné sa vyhýbať k motivácii užívateľa robiť náhle pohyby.

### 3.2.2 Vkládanie virtuálnych objektov

Pri vkladaní 3D objektov do scény rozšírenej reality by sa mala aplikácia riadiť pravidlami:

**Pomáhať užívateľom pochopiť, kde nájsť povrch na umiestnenie objektu.** Používanie vizualizátorov je skvelý spôsob komunikácie, keď je aktívny režim zacielenia na povrch. Akonáhle je cieľom povrch, indikátor by sa mal zmeniť tak aby naznačoval, že umiestnenie objektu je teraz možné.

**Vyhnuť sa snahe presne zarovnať vložené objekty.** V rozšírenej realite sú hranice aproximácie, ktoré sa môžu meniť, keď sa okolie používateľa ďalej analyzuje.

**Umožniť ľuďom priamo interagovať s objektami pomocou známych gest.** Uvažovať napríklad o podporovaní gesta ťahania jedným prstom pre pohybujúce sa objekty a gestom otáčania dvoch prstov pre otáčanie objektov. Rotácia by sa vo všeobecnosti mala vzťahovať na povrch, na ktorom leží objekt. Napríklad objekt umiestnený na vodorovnom povrchu by sa typicky otáčal okolo vertikálnej osi objektu.

**Vo všeobecnosti udržiavať interakciu s objektami čo najjednoduchšie.** Dotykové gestá sú v podstate dvojrozmerné, ale skúsenosť s rozšírenou realitou zahŕňa tri dimenzie reálneho sveta.

### 3.2.3 Spracovanie prerušení

**Vyhnuť sa zbytočnému prerušeniu zážitku z rozšírenej reality.** ARKit nedokáže sledovať polohu a orientáciu zariadenia ak je neaktívna funkcia AR. Jedným spôsobom, ako sa vyhnúť prerušeniam, je nechať ľudí upraviť objekty a nastavenia v rámci aktívnej funkcionality AR. Napríklad, ak užívateľ umiestni stoličku, ktorú zvažuje kúpiť do svojej obývačky, a stolička je k dispozícii v rôznych farbách, aplikácia umožní farbu tejto stoličky zmeniť bez opustenia AR.

**Povolenie užívateľovi zrušiť relokáciu.** Ak nie je užívateľ schopný umiestniť a orientovať svoje zariadenie do pozície pred prerušením, relokácia pokračuje bez úspechu. Aplikácia naviguje užívateľa k tomu aby dokázal obnoviť AR úspešne alebo poskytnite tlačítko na reštartovanie AR.

### 3.3 Programovanie v jazyku Swift

Programovací jazyk Swift sa stále radí medzi pomerne nové programovacie jazyky. Prvé predstavenie toho programovacieho jazyka bolo v roku 2014, kedy bola uvedená verzia Swift 1.0 . Od vtedy do dnešnej doby bolo predstavených niekoľko verzii, pričom najnovšia verzia je v dnešnej dobe Swift 5.0 . Samotný jazyk vychádza z jeho predchodcu, ktorým bol programovací jazyk Objective C. Jazyk sa prezentuje ako výsledok najnovšieho výskumu programovacích jazykov v kombinácii s niekoľko ročnými skúsenosťami s budovaním platformy Apple. Programovanie v jazyku Swift je založené na používaní čistej syntaxe, ktorá uľahčuje čítanie a údržbu kódu.

Swift sa snaží eliminovať výskyt nebezpečného kódu aplikáciach. Premenné sú vždy inicializované pred použitím, polia a cele čísla sú kontrolované na preplnenie. Pamäť je automaticky riadená a žiadosti o výhradný prístup k zdieľanej pamäti je chránený pred vznikom konfliktov. Celková syntax je v jazyku Swift definovaná tak, aby bol programátor schopný jednoducho a efektívne vyjadriť svoje myšlienky. Ako príklad používa Swift jednoduché trojznakové kľúčové slová definujúce premennú (*var*) alebo konštantu (*let*).

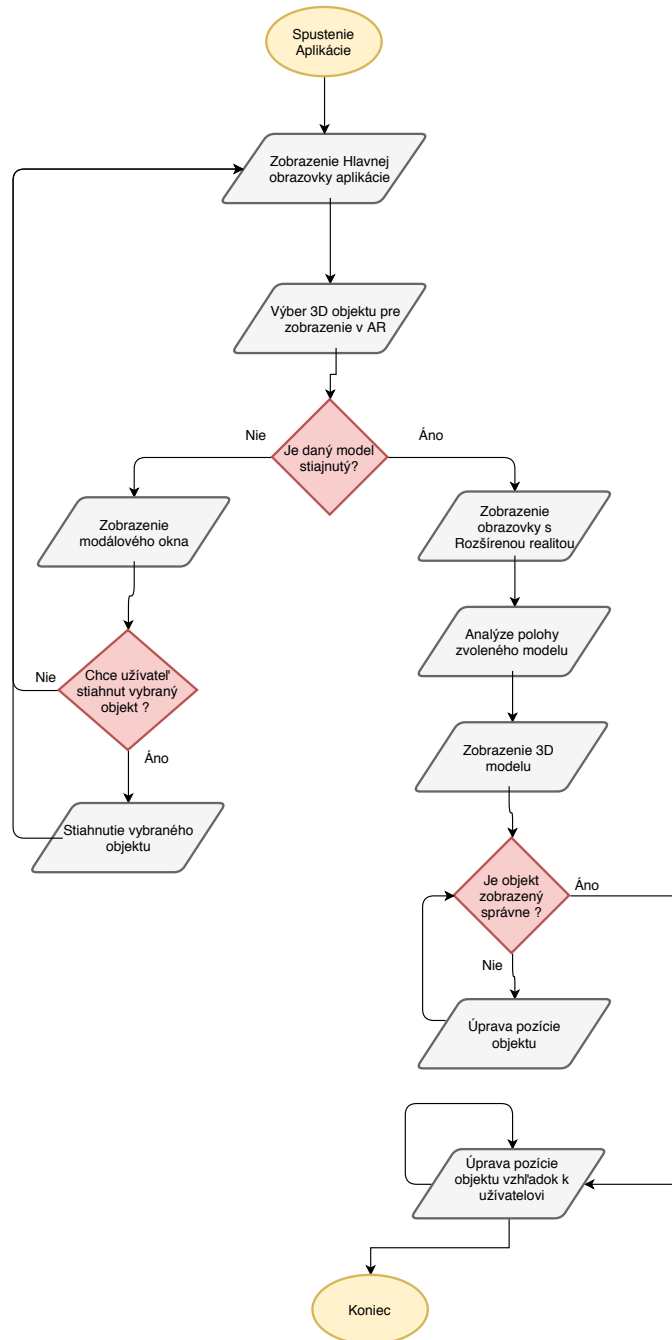
## Kapitola 4

# Návrh aplikácie pre zobrazovanie historických budov

Cieľom práce je vytvorenie aplikácie, ktorá bude zobrazovať historickú podobu budov v užívateľovom okolí pomocou rozšírenej reality. Hlavným cieľom je zatriktívnenie histórie, ktorá sa nedochovala do dnešnej doby. Pomocou rozšírenej reality priblížiť podobu budov, alebo scenérii hradov, tak ako vyzerali v minulosti. To znamená, že úlohou aplikácie bude zobrazovanie jednotlivých historických 3D modelov na ich reálnom mieste s čo najmenšou interakciou s užívateľom. Výsledný systém by mal byť schopný identifikovať polohu užívateľa voči reálnej pozícii vybraného 3D modelu a následne tento model vložiť na správne miesto do rozšírenej reality. Výpočet polohy 3D modelu v scéne nemusí vždy viesť k žiadanej pozícii objektu. Z toho dôvodu je potrebné aby aplikácia umožňovala užívateľovi vhodným spôsobom upraviť pozíciu virtuálneho objektu na správne miesto.

Potenciálne zariadenie, na ktoré je samotná aplikácia navrhovaná je tablet, ktorý poskytuje užívateľovi displej široký až 13". Toto zariadenie je vhodné pre účely aplikácie práve vďaka veľkosti displeju, na ktorom bude možné vhodne zobrazovať jednotlivé modely budov alebo iných objektov. Pri návrhu grafického rozhrania aplikácie bude kladený dôraz na jeho intuitívnosť. Tak aby užívateľ nebol zťažovaný zbytočnými grafickými prvkami, ktoré by ho pripravovali o zážitok z používania samotnej aplikácie. Ako vyplýva z pravidiel o používaní knižnice *ARKit* v kapitole 3, tak obrazovka zobrazujúca video z kamery obsahujúce virtuálne prvky, by mala zaberáť čo najväčšiu plochu displeja. Vďaka tomu bude zaistené zobrazenie detailov, ktoré by inak mohli užívateľovi uniknúť.

Navrhovaná aplikácia by nemala byť obmedzená len na určitý počet 3D modelov, ktoré dokáže zobrazovať užívateľovi v rozšírenej realite. Užívateľ pri používaní výslednej aplikácie musí mať možnosť výberu jednotlivých objektov, ktoré ho určitým spôsobom zaujali a chce ich zobrazovať v rozšírenej realite. Z toho vyplýva, že navrhovaná aplikácia musí riešiť distribúciu jednotlivých objektov medzi užívateľov na základe ich voľby.



Obr. 4.1: Diagram zobrazujúci jednotlivé činnosti aplikácie. Činnosti predstavujú určitú funkčnosť navrhovanej aplikácie, pričom samotný graf zobrazuje ich časovú následnosť pri používaní aplikácie užívateľom [zdroj: vlastná tvorba].

## 4.1 Prínos navrhovaného riešenia

Navrhovaná aplikácia je prevažne určená pre bežné používanie. Keďže jej cieľom je zobrazenie historických budov v ich predošlej podobe, jej primárnymi užívateľmi budú turisti alebo ľudia so záľubou pre históriu. Ďalšia skupina užívateľov, pre ktorých je aplikácia navrhovaná sú deti vo vekovom rozmedzí 12-18 rokov. Výsledný systém by mohol byť použiteľný ako dobrá učebná pomôcka pre základné a stredné školy pri výučbe predmetov histórie. Na Slovensku ako aj v Českej Republike sa nachádza množstvo zrúcanín, ktoré sú často navštevované hlavne v letných mesiacoch pomerne veľkými počtami turistov. Výsledná aplikácia by bola veľmi prínosná pre tento typ turistiky. Keďže je veľmi ťažké rekonštruovať zrúcaniny do pôvodnej podoby a zväčša sa praktizuje len konzervácia zachovalých častí, navrhovaná aplikácia je niečo, čo by mohlo dané zrúcaniny zatriktívniť. Pri porovnaní s rekonštrukciou je to oveľa lacnejšie riešenie.

Navrhovaná aplikácia má oveľa väčší potenciál ako len zobrazovanie historických budov. Tým, že sa aplikácia zaoberá zobrazovaním virtuálnych objektov do reality, je vhodné ju napríklad použiť pri stavebníctve. To znamená, že by pomocou navrhovanej aplikácie bolo možné zobrazovať stavby ešte pred tým, ako sú dokončené a tým by sa užívateľovi naskytl pohľad ako bude daná budova vyzerat po dokončení.

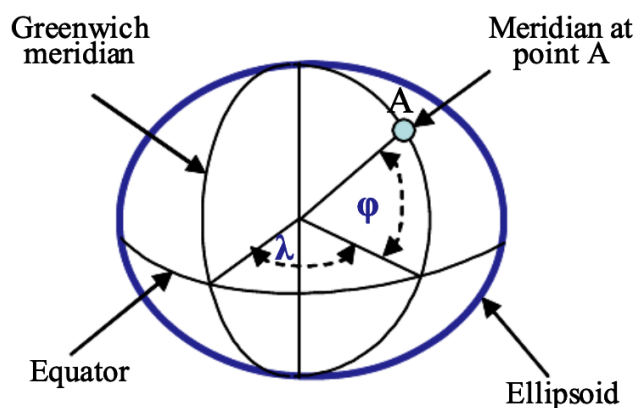
## 4.2 Požiadavky na funkcionality aplikácie

Hlavnou úlohou navrhovanej aplikácie bude zobrazovanie 3D modelov na ich konkrétnej pozícii v reálnom svete. Ďalej by mala aplikácia umožňovať užívateľovi určitú interakciu so zobrazovanými objektami v podobe zmeny ich pozície. Táto požiadavka vyplýva z odchýliek, ktoré môžu nastať pri vypočítaní výslednej pozície vybraného objektu. V neposlednom rade by mala aplikácia podporovať určitý spôsob distribúcie objektov medzi jednotlivými užívateľmi. Na základe tejto funkcionality je možné zhrnúť požiadavky na výslednú aplikáciu do bodov:

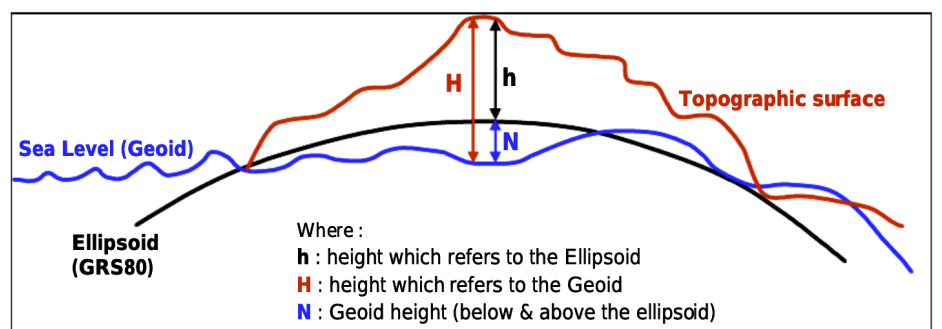
- vkladanie objektov na ich pozíciu v reálnom svete
- úprava pozície objektu
- rotácia objektu okolo vlastnej osi
- sťahovanie nových objektov z databázy

### 4.2.1 Vkladanie objektov na ich pozíciu v reálnom svete

Jedná sa o hlavnú funkcionality navrhovanej aplikácie. Existuje niekoľko spôsobov ako vkladať 3D objekt do reality na konkrétnu pozíciu. Jeden možný spôsob ako daný problém riešiť je vkladať objekty do lokálnej karteziánskej súradnicovej sústavy. V tomto prípade sa predpokladá, že aplikácia bude poznať GPS súradnice polohy užívateľa a GPS súradnice polohy objektu, ktorý bude vkladatý do rozšírenej reality. Poloha užívateľa predstavuje počiatok karteziánskej súradnicovej sústavy. Následne je potrebné vypočítať referenčnú pozíciu objektu v súradnicovej sústave na základe jeho GPS polohy. Ako je možné vidieť na obrázku 4.2, tak je možné previesť GPS súradnice z formátu WGS84 do formátu karteziánskej súradnicovej sústavy. V tomto prípade je zemepisná šírka reprezentovaná pomocou uhla  $\phi$  a zemepisná dĺžka je reprezentovaná pomocou uhla  $\lambda$ . Osi karteziánskej súradnicovej sústavy sú v obrázku vyjadrené rovinami, od ktorých sa vyjadrujú oba spomínané uhly.



Obr. 4.2: Geografické súradnice bodu na elipsoide [zdroj: [9]].



Obr. 4.3: Zobrazenie vzťahu medzi elipsoidom a geoidom [zdroj: [9]].

Pre koncový výpočet referenčných súradníc je potrebná aj GPS informácia o výške, ktorá je vyjadrená pomocou parametra altitude. Výsledné rovnice pre výpočet jednotlivých súradníc sú vyjadrené ako [9]:

$$X = (N + h) * \cos \phi \sin \alpha \quad (4.1)$$

$$Y = (N + h) * \sin \phi \cos \alpha \quad (4.2)$$

$$Z = [N * (1 - e^2) + h] * \sin \phi \quad (4.3)$$

kde :

$$N = \frac{a}{\sqrt{1 - e^2 * \sin^2 \phi}} \quad (4.4)$$

$$e = \frac{a^2 - b^2}{a^2} \quad (4.5)$$

$$f = \frac{a - b}{a} \quad (4.6)$$

$$a = 6378137 \quad (4.7)$$

$$b = 6356752 \quad (4.8)$$



## 4.2.2 Úprava pozície objektu

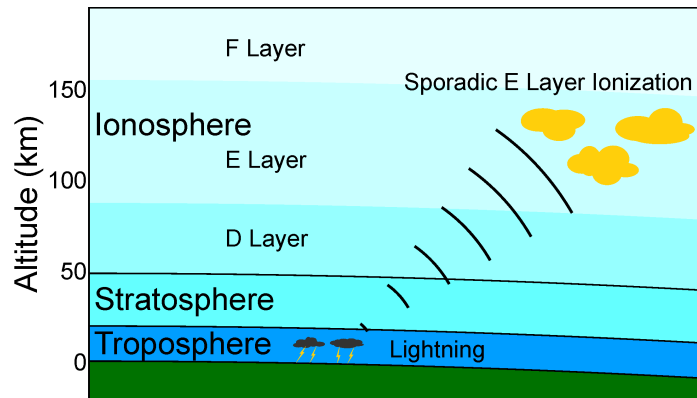
Presnosť zobrazenia 3D objektu v realite dosť závisí na presnosti GPS súradnice polohy užívateľa. Medzi faktory, ktoré ovplyvňujú presnosť GPS signálu sú radené:

Efekt	Mierka nepresnosti
Ionosferické efekty	+/- 5 metrov
Posuny satelitných dráh	+/- 2.5 metrov
Chyby hodín satelitov	+/- 2 metrov
Tróposférické efekty	+/- 0.5 metrov
Chyby zaokrúhľovania	+/- 1 metrov

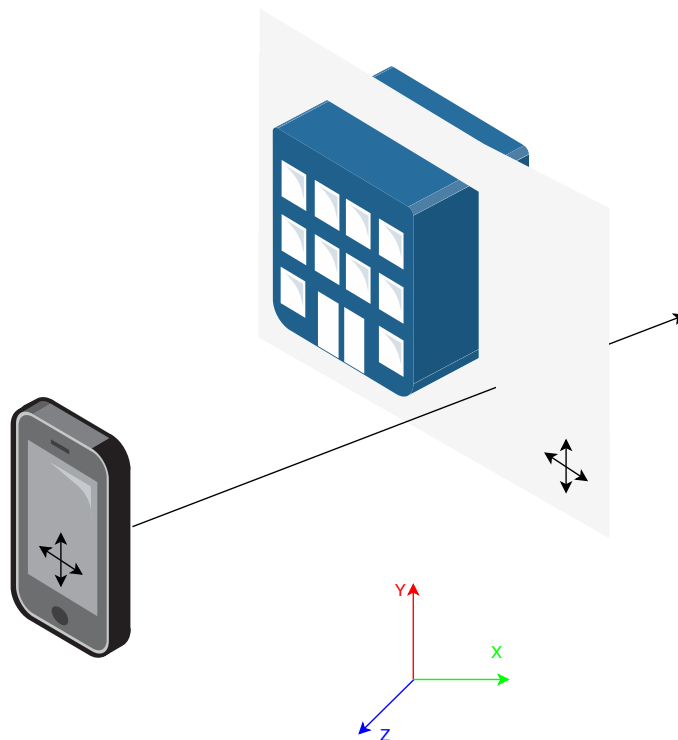
Tabuľka 4.1: Efekty ovplyvňujúce presnosť GPS signálu [zdroj: [10]].

V tabuľke 4.1 je možné vidieť, že najväčšiu nepresnosť GPS signálu spôsobujú Ionické efekty. Medzi ďalšie efekty ktoré ovplyvňujú presnosť GPS signálu sú budovi v okolí užívateľa. Ak sa užívateľ nachádza medzi zástavbou vysokých budov, tak dochádza k nepresnosti GPS signálu pravé odrazom z týchto budov. Eliminovať jednotlivé faktory ovplyvňujúce presnosť signálu by bolo veľmi komplikované. Z toho dôvodu je potrebné, aby navrhovaný systém bol schopný reagovať na prípady, kedy objekt nebude vložený sa správnu pozíciu. To znamená, že medzi základnú funkcionality systému je potrebné zaradiť aj posun objektu po jeho vložení do rozšírenej reality. Ako vhodné riešenie sa ukazuje implementácia posunu objektu užívateľom. Je predpoklad, že užívateľ je schopný upraviť pozíciu objektu tak, aby zodpovedala jeho reálnej pozícii. Úpravu pozície objektu v rozšírenej realite je možné realizovať viacerými spôsobmi, ako napríklad posúvanie objektu po osiach súradnicového systému, posun na základe zmeny pozície zariadenia alebo posun objektu po ploche rovnobežnej s obrazovkou zariadenia. Pre navrhovanú aplikáciu bude najvhodnejšie, ak výsledný systém zmeny pozície bude pre užívateľa čo najintuitívnejší použitím jednoduchého gesta. Ďalším dôležitým faktorom, ktorý musí spĺňať navrhovanie funkcionality je komplikovanosť pohybu. Užívateľ by nemal byť zbytočne zaťažovaný premýšľaním, ako daný objekt presunie na vybranú pozíciu. Výsledný pohyb by mal byť pre užívateľa prirodzený.

Jedna z metód, ktorá spĺňa požadované parametre, je metóda posunu objektu v rovine rovnobežnou s obrazovkou užívateľského zariadenia, ako to môžeme vidieť na obrázku 4.5. Pri danom pohybe je možné objekt jednoducho posúvať do strán, alebo meniť jeho výšku. Každý bod na obrazovke je reprezentovaný bodom na danej ploche, ktorá prechádza zobrazeným 3D objektom. Pre výpočet bodu na ploche je možné využiť techniku priesečníka vektora s rovinou. V tomto prípade rovinu reprezentuje vytvorená plocha a vektor je reprezentovaný dvoma bodmi. Počiatočný bod má súradnice užívateľa a koncový bod daného vektora obsahuje súradnice bodu v priestore, na ktorý klikne užívateľ na obrazovke. Tento spôsob posúvania zaručuje, že objekt nebude posúvaný smerom od alebo k užívateľovi.



Obr. 4.4: Zobrazenie jednotlivých vrstiev atmosféry zeme, ktorých vplyvom dochádza k náhodnému ovplyvňovaniu GPS signálu [zdroj: [15]].

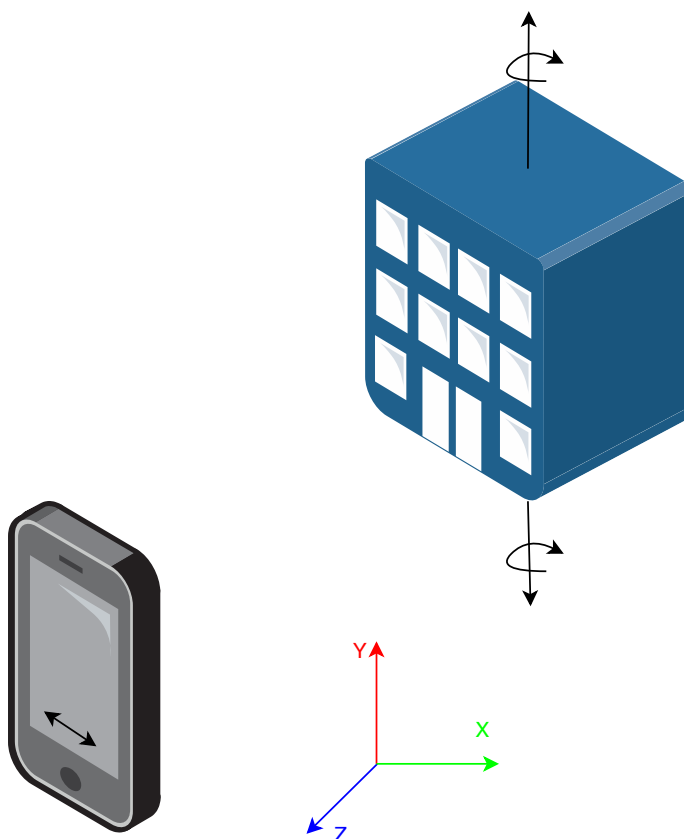


Obr. 4.5: Ukážka posunu objektu po ploche, ktorá je rovnobežná s displejom zariadenia [Vlastná tvorba].

### 4.2.3 Rotácia objektu okolo vlastnej osi

Pri vkladaní objektu do rozšírenej reality je potrebné myslieť aj na jeho správne natočenie. Na to, aby aplikácia vedela objekt správne natočiť, tak aby zodpovedal reálnej budove, ktorú sa snaží prekryť, je potrebné určiť osiam karteziánskej súradnicovej sústavy vždy rovnaký smer. Pomocou knižnice *ARKit* je systém schopný pri inicializácii karteziánskej sústavy určiť smer z-ovej osi na sever. Bohužiaľ, presnosť severu jednotlivých mobilných zariadení nie je dokonalá a je bežné, že jednotlivé zariadenia ukazujú iný smer severného pólu. Spôsob ako daný smer spresniť je len jeden, a to je kalibrácia zariadenia, ktorá dokáže čiastočne daný problém eliminovať ale nie ho úplne odstrániť. Z toho dôvodu pri používaní aplikácie môže dôjsť k prípadom, kedy objekt nebude správne natočený a bude potrebná korekcia natočenia užívateľom.

Tak ako pri zmene polohy objektu, tak aj pri metóde rotácie je veľmi dôležité dbať na to, aby výsledná metóda bola pre užívateľa jednoduchá čo sa týka používania. Existuje veľa spôsobov ako rotovať objekt okolo vlastnej osi. Ako príklad je možné uviesť metódu rotácie objektu na základe rotácie zariadenia, alebo rotácia objektu okolo vlastnej osi na základe užívateľského gesta. Pre celkové riešenie navrhovanej aplikácie je vhodnejšie použiť metódu rotácie na základe jednoduchého gesta zadaného užívateľom pohybom prstov po obrazovke. To znamená, že pri spustení módu pre rotáciu objektu, bude môcť užívateľ jednoduchým pohybom prsta (vpravo, vľavo) po displeji meniť natočenie vybraného 3D modelu ako je to znázornené na obrázku 4.6.



Obr. 4.6: Ukážka rotácie objektu okolo svojej osi na základe užívateľom zadaného gesta pohybu prstov po obrazovke v smere šípok znázornených na displeji telefónu [Vlastná tvorba].

#### 4.2.4 Návrh štruktúry databázy pre ukladanie objektov

Ako ďalšia neodeliteľná funkcionálna potrebná pre navrhovanú aplikáciu je možnosť pridávania alebo odoberania 3D objektov zobrazovaných v danej aplikácii. Keďže nie je vhodné, aby výsledná aplikácia obsahovala len obmedzené množstvo historických 3D modelov, je potrebné navrhnúť metódu distribúcie jednotlivých objektov medzi užívateľov. Ako najideálnejšie riešenie danej funkcionality sa javí využitie verejnej databázy *Firestore*. Táto databáza poskytuje široké portfólio cielené pre webové a mobilné aplikácie. Medzi základné produkty *Firestore* databázy patria napríklad *Cloud Firestore*, *ML Kit*, *Cloud Functions*, *Cloud Storage* a *Realtime Database*.

Z dôvodu, že navrhovaná aplikácia bude bežať na platforme iOS a pri práci s rozšírenou realitou využívať knižnicu *ARKit*, tak jediný podporovaný formát 3D modelov bude **.scn**. Jedná sa o formát súboru, ktorý je určený pre ukladanie 3D scén. Každá scéna uložená v databáze musí obsahovať samotný model ako aj informácie o tomto modeli. Tieto informácie sú nevyhnutné pre navrhovanú aplikáciu, pretože budú využívané k výpočtu správnej pozície a natočeniu objektu v scéne rozšírenej reality. Medzi informácie o jednotlivých objektoch je možné zaradiť:

- **Identifikačné číslo scény** je potrebné pre filtrovanie scén v aplikácii. Na základe tohto atribútu dokáže aplikácia efektívne rozlišovať scény, a tak dokáže predísť viacerým konfliktom.
- **GPS poloha objektu** je jedna z hlavných informácií, vďaka ktorej dokáže aplikácia vkladať objekty na správnu pozíciu v rozšírenej realite.
- **Natočenie objektu voči severnému pólu** reprezentuje popri GPS súradniciach ďalší údaj potrebný pre správne vkladanie objektov do rozšírenej reality. Vďaka tomuto atribútu dokáže aplikácia správne určiť natočenie vybraného objektu voči užívateľovi.
- **Odkaz na 3D scénu** predstavuje atribút, ktorý je potrebný pri sťahovaní samotnej scény. Z dôvodu, že táto scéna nie je uložená priamo v súbore typu **JSON**, aby nedochádzalo k zbytočnému sťahovaniu dát, tak bolo potrebné vytvoriť prepojenie medzi **JSON** súborom a samotnou 3D scénou. O toto prepojenie sa bude starať spomínaný odkaz, pomocou ktorého bude užívateľ schopný danú scénu stiahnuť.
- **Objekty nachádzajúce sa v scéne** predstavujú informácie potrebné pre aplikáciu pri samotnom vkladaní scény do rozšírenej reality. Každá scéna môže pozostávať z viacerých objektov, ktoré majú rôznu polohu v danej scéne. Tieto informácie sú pre aplikáciu kľúčové k tomu, aby bola schopná vložiť jednotlivé objekty scény do rozšírenej reality. Medzi tieto informácie radíme názov objektu, ktorý je taktiež aj identifikátor objektu a mal by byť unikátny identifikátor v rámci jednej scény. Ďalšie údaje obsiahnuté v tomto objekte sú informácie o pozícii jednotlivých objektov v scéne.

Z návrhu aplikácie vyplýva, že systém by mal byť schopný zobraziť základne informácie o objekte užívateľovi aj bez toho, aby bol tento objekt stiahnutý na zariadení užívateľa. Z toho dôvodu budú informácie o objekte uložené v databáze pomocou JSON formátu ako môžeme vidieť na ukážke 1.

Základným a zároveň jediným objektom v danej štruktúre bude objekt **scenes**. Tento objekt obsahuje pole jednotlivých modelov, ktoré budú užívateľovi k dispozícii na stiahnutie. Jednotlivé modely sú reprezentované ako samostatné objekty obsahujúce všetky potrebné informácie pre správne vloženie objektu do rozšírenej reality.

```
1 {
2   "scenes" : [
3     {
4       "name" : "sceneName",
5       "Coordinates" : {
6         "lat" : Latitude,
7         "long" : Longitude
8       },
9       "link" : "linkToScene",
10      "id" : "sceneId",
11      "containedObjects" : [
12        {
13          "name" : "ObjectName",
14          "positionX" : 0.0,
15          "positionY" : 0.0,
16          "positionZ" : 0.0
17        }
18      ]
19    }
20  ]
21 }
```

Kód 1: Ukážka štruktúry dát, ktoré budú uložené v databáze *Firebase*. Tieto dáta predstavujú informácie o objektoch, ktoré sú k dispozícii užívateľovi.[Vlastná tvorba]

### 4.3 Grafické rozhranie aplikácie

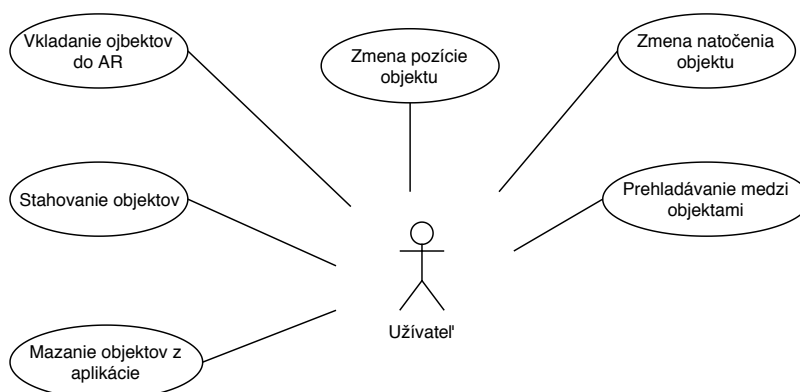
Navrhovanie grafického rozhrania aplikácie je taktiež dôležitou súčasťou vývoja mobilných, webových alebo iných aplikácií. Medzi prvé kroky, ktoré je potrebné urobiť pri navrhovaní grafického rozhrania je vytvorenie počiatočného návrhu, ktorý sa odborné nazýva *Wireframe*. Tento návrh predstavuje prvú ukážku, ako bude vyzeráť výsledné grafické rozhranie aplikácie. Väčšinou sa jedná o ukážky návrhov, ktoré sú nakreslené na papier. Po tom, čo je *Wireframe* dostatočne premyslený, prichádza na rad prenesenie daného návrhu do elektronickej formy.

Pre návrh grafického rozhrania navrhovanej aplikácie bol zvolený nástroj Draw.io, ktorý poskytuje dostatočnú funkcionálnosť a preddefinované grafické objekty, pomocou ktorých je veľmi jednoduché navrhovať požadovaný dizajn aplikácie.

### 4.3.1 Prípady použitia navrhovanej aplikácie

Ako je možné vidieť na obrázku 4.7, tak pre vyjadrenie jednotlivých prípadov použitia bol použitý *UseCase diagram* vytvorený v nástroji Draw.io . Jedná sa štandardný UML diagram, ktorý bol navrhnutý práve na tento účel.

Obrázok 4.7 zobrazuje, že aplikácia nebude poskytovať veľa prípadov použitia. Je to z dôvodu zamerania aplikácie na vkladanie objektov do rozšírenej reality ako bolo spomenuté v podkapitole 3.2, tak jednou z dobrých praktík pri vyvíjaní aplikácii s rozšírenou realitou je maximalizácia zážitku užívateľa z rozšírenej reality. Navrhovaná aplikácia neposkytuje autentifikačný systém alebo funkcionality pre vytvorenie účtu, na základe ktorého by bolo možné filtrovať objekty, ktoré si bude môcť užívateľ zobraziť. Pri návrhu aplikácie bol skôr kladený dôraz na základnú funkcionality, ktorá je potrebná pre splnenie účelu aplikácie, tak aby nezatažovala užívateľa.



Obr. 4.7: Obrázok zobrazujúci jednotlivé prípady použitia navrhovanej aplikácie [zdroj: vlastná tvorba].

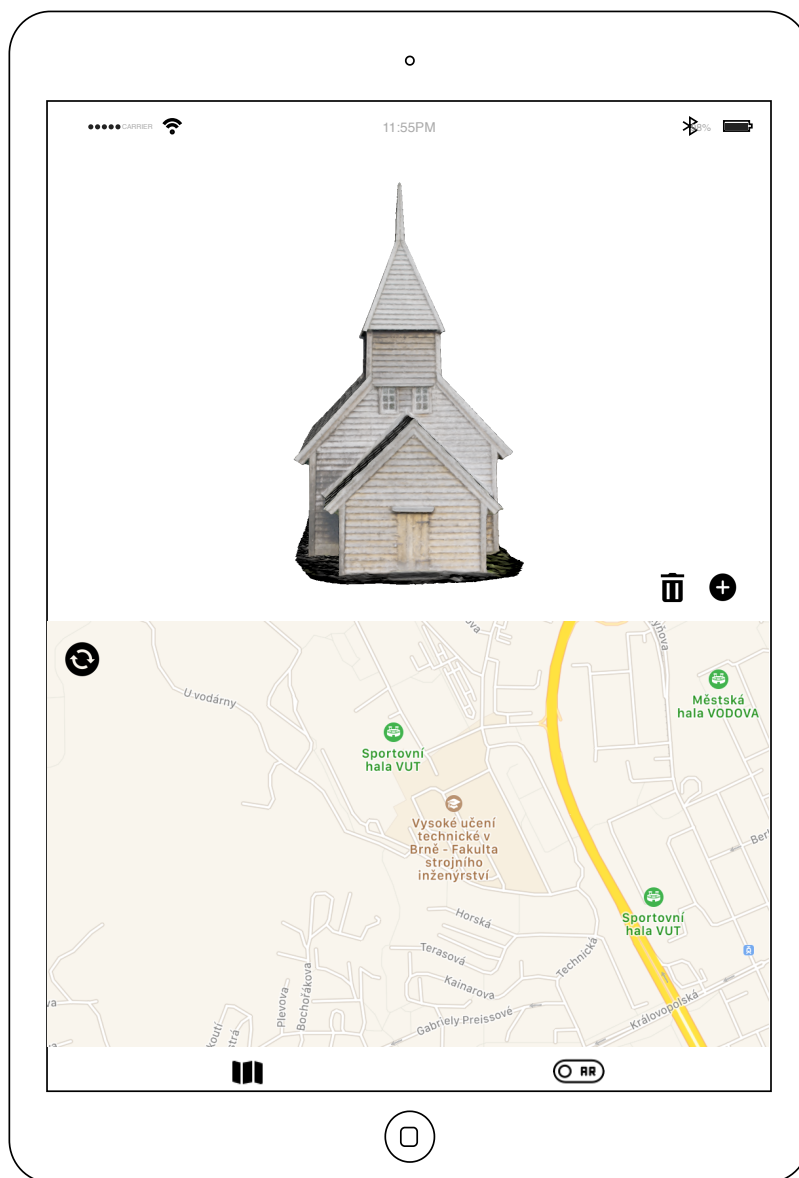
Ako to vyplýva z názvu aplikácie, tak najzákladnejším prípadom použitia aplikácie bude vkladanie objektu do rozšírenej reality. Ostatné prípady z obrázku 4.7 je možné rozdeliť do skupín ako interakcia s 3D objektom, komunikácia s databázou.

### 4.3.2 Návrh hlavnej obrazovky aplikácie

Návrh hlavnej obrazovky navrhovanej aplikácie prešiel niekoľkými zásadnými zmenami pri interaktívnom vývoji samotnej aplikácie. Spočiatku mala navrhovaná aplikácia obsahovať len jednu obrazovku, ktorá by obsahovala potrebnú funkcionality pre splnenie hlavného prípadu použitia aplikácie.

Obrazovka ktorá prináša zážitok užívateľovi z rozšírenej reality by nemala obsahovať veľa prvkov, ktoré by zbytočne zaberali miesto na obrazovke používaného zariadenia. Z toho dôvodu došlo k zmene návrhu grafického rozhrania hlavnej obrazovky, tak aby plocha zobrazujúca video z kamery bola čo najväčšia. Tak vznikol návrh novej hlavnej obrazovky, ktorú môžeme vidieť na obrázku 4.8, pričom táto obrazovka obsahuje funkcionality, s ktorou užívateľ prichádza do styku pred samotným používaním rozšírenej reality. Obrazovka, ktorá obsahuje obraz kamery s virtuálnymi objektami je ďalej popisovaná ako "ARScreen".

Samotná obrazovka obsahuje dve hlavné časti (hornú a dolnú). Ako je možné vidieť na obrázku 4.8, tak horná časť obrazovky bude určená na zobrazenie 3D modelu budovy, ktorý sa nachádza na danej pozícii. Z užívateľského pohľadu je táto časť veľmi dôležitá, pretože prináša možnosť zobrazenia 3D modelu pred samotným zobrazením v rozšírenej realite. To znamená, že užívateľ si bude môcť jednoducho vyberať spomedzi ponúkaných objektov.



Obr. 4.8: Návrh hlavnej obrazovky, ktorá sa zobrazí užívateľovi po spustení aplikácie [zdroj: vlastná tvorba].

V dolnej časti obrazovky sa nachádza mapa, ktorej hlavnou úlohou bude zobrazovať jednotlivé modely na ich GPS pozícii. Budú sa tam nachádzať modely, ktoré má užívateľ k dispozícii stiahnuté na svojom zariadení, ako aj modely ktoré sú dostupné k stiahnutiu z databázy. Jednotlivé modely sú farebne rozlíšené. Vďaka čomu bude mať užívateľ jednoduchý a rýchly prehľad o tom, v akom stave sa jednotlivé 3D objekty nachádzajú. Informácie v databáze sa môžu samozrejme meniť aj počas aktívneho používania aplikácie. Na aktu-

alizáciu dát v zariadení slúži tlačítko umiestnené v hornom rohu mapy nachádzajúcej sa na obrazovke.

Navrhovaná aplikácia bude obsahovať dve hlavné obrazovky medzi ktorými bude užívateľ prechádzať pri používaní aplikácie. Z pohľadu užívateľa je veľmi dôležité, aby prechod medzi obrazovkami bol čo najjednoduchší. Ako je možné vidieť na obrázku 4.8, tak v dolnej časti obrazovky sa nachádza prvok *TabBar*, ktorého úlohou bude práve spomínaný jednoduchý prechod medzi obrazovkami. Ikony reprezentujúce jednotlivé obrazovky boli vybrané na základe hlavného obsahu alebo prípadu použitia. Hlavná obrazovka je reprezentovaná ikonou mapy, keďže hlavnou úlohou danej obrazovky je zobraziť užívateľovi mapu a jednotlivé modely. Obrazovka s rozšírenou realitou je reprezentovaná ikonou, ktorá v sebe obsahuje názov AR čo reprezentuje skratku od slova Augmented Reality.

### 4.3.3 Návrh obrazovky pre interakciu s rozšírenou realitou

Návrh obrazovky pre interakciu s rozšírenou realitou prešiel niekoľkými zásadnými zmenami pri iteratívnom vývoji navrhovanej aplikácie. Jedným z hlavných faktorov, na ktorý sa kládol dôraz pri samotnom návrhu, bol faktor veľkosti zobrazovanej plochy pre rozšírenú realitu. Táto plocha by mala zaberáť čo najviac z danej obrazovky zariadenia, na ktorom je spustená aplikácia. Z toho dôvodu došlo k zásadným zmenám návrhu grafického rozhrania aplikácie.

Obrázok 4.9 zobrazuje finálny návrh danej obrazovky. Ako je možné vidieť na prvý pohľad, tak na obrazovke sa nenachádza veľa prvkov, ktoré by zaberali miesto výhľadu užívateľa, a tým znižovali zážitok z rozšírenej reality. Nachádzajú sa tam len prvky, ktoré priamo súvisia s interakciou užívateľa s objektami zobrazovanými v rozšírenej realite. Ako môžeme vidieť na obrázku 4.9, tak tieto prvky sa nachádzajú na pravej strane obrazovky približne v strednej výške. Zvolenie danej pozície je odôvodnené tým, ako užívateľ drží zariadenie pri používaní aplikácie. Keďže sa jedná o aplikáciu určenú primárne na tablety, tak uchopenie daného zariadenia je prirodzene odlišné ako pri mobilnom zariadení. Z pohľadu tabletu, užívateľ drží zariadenie oboma rukami v dolných rohoch, tak aby svojím palcom bol schopný interagovať s prvkami na obrazovke. Na základe toho bola zvolená pozícia prvkov v strednej výške obrazovky, tak aby bol na ne jednoduchý dosah.



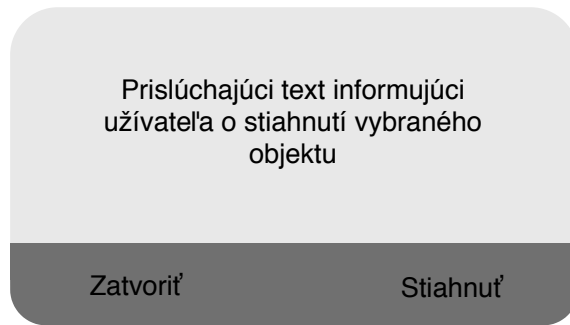


Obr. 4.9: Obrázok zobrazujúci jednotlivé prípady použitia navrhovanej aplikácie.[Vlastná tvorba]

#### 4.3.4 Návrh dialógové okna pre sťahovanie objektov

Jednou z hlavných funkcionalít navrhovanej aplikácie je aj sťahovanie objektov z verejne prístupnej databázy. Sťahovanie nových objektov nie je vždy žiadúca činnosť, keďže objekty, ktoré sú sťahované majú určitý objem dát. Je užívateľsky prívetivé ak sa aplikácia pred samotným stiahnutím objektu spýta, či si naozaj užívateľ praje daný objekt stiahnuť.

Pre tieto účely bolo navrhnuté dialógové okno, ktorého vizuálna stránka je zobrazená na obrázku 4.10. Daná ukážka zobrazuje dialógové okno obsahujúce text, ktorý má za úlohu informovať užívateľa, že sa chystá stiahnuť objekt z databázy. Navrhované okno má hlavne za úlohu zabrániť prípadom, kedy užívateľ omylom klikne na objekt, ktorý nie je lokálne dostupný na zariadení, a tento objekt by bol bez jeho vedomia stiahnutý.



Obr. 4.10: Dialógové okno, ktoré ma za úlohu upozorniť užívateľa, že daný objekt na ktorý klikol, nie je dostupný v lokálnej pamäti a či si želá jeho stiahnutie. Po stlačení tlačítka *Zatvoriť* sa dané okno zatvorí [zdroj: vlastná tvorba].

## Kapitola 5

# Implementácia aplikácie

Táto kapitola bude venovaná implementácii navrhovanej aplikácie pre vkladanie 3D objektov do rozšírenej reality. Samotná implementácia postupovala podľa technických požiadaviek definovaných v predošlej kapitole. Na základe týchto technických parametrov je rozdelená aj samotná kapitola implementácie, kde jednotlivé sekcie predstavujú technické požiadavky na výsledný systém. V každej podkapitole bude detailnejšie popísaná implementácia jednotlivých technických parametrov, pričom bude uvedená realizácia daného parametru a odôvodnenie vybranej realizácie.

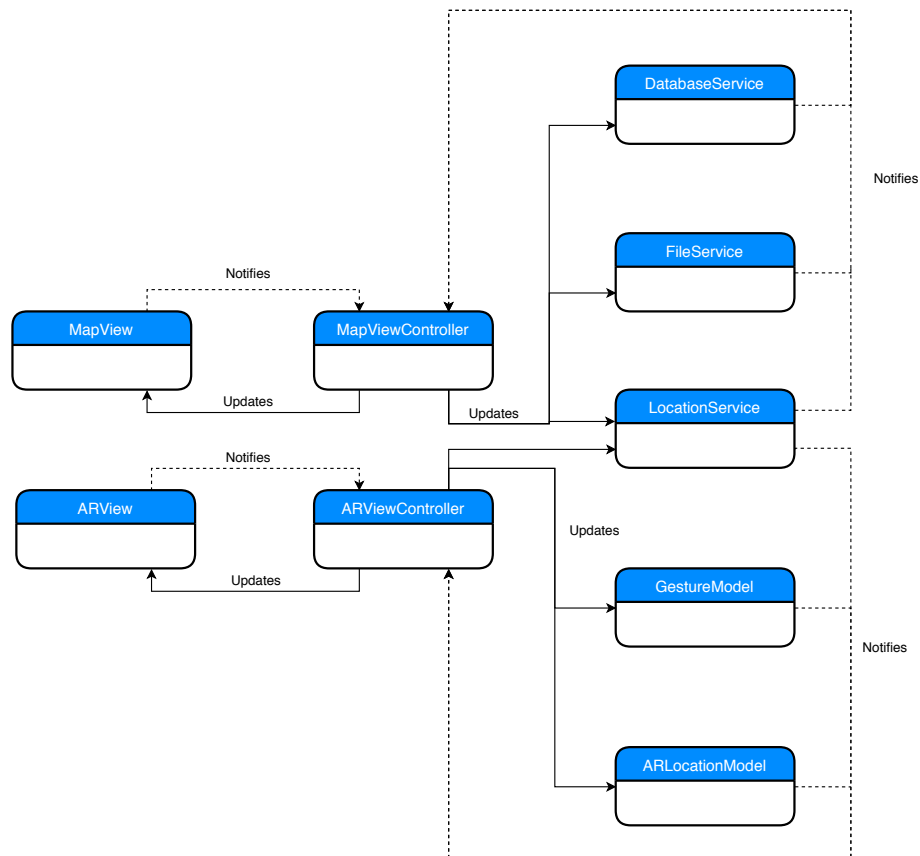
### 5.1 Technický popis aplikácie

Pre implementáciu aplikácie na vkladanie historických modelov budov do rozšírenej reality bola vybraná platforma iOS. Táto platforma poskytuje vhodnú knižnicu *ARKit* na riešenie daného problému. Táto knižnica bola detailnejšie popísaná v podkapitole *ARKit*. Hlavným zariadením na ktorom bol daný systém škalovaný je tablet. Je to z dôvodu, že toto zariadenie poskytuje svojim užívateľom veľký displej, ktorý je potrebný pre vytvorenie potrebného zážitku z rozšírenej reality.

Keďže systém bude používať knižnicu *ARKit*, ktorá je podporovaná len na systémoch iOS od verzie 11, je to jedno z hlavných obmedzení aplikácie. Knižnica *ARKit* predstavuje špeciálnu knižnicu od spoločnosti Apple, ktorá bola vytvorená za účelom zjednodušenia vývoja aplikácii pracujúcich s rozšírenou realitou. Samotná knižnica má v sebe implementované prvky pre detekovanie okolia užívateľa pomocou technológie *Visual Inertial Odometry*. Táto technológia bola detailnejšie popísaná v kapitole 2, preto sa v tejto kapitole budeme venovať len metódam, ktoré boli použité pri implementácii navrhovanej aplikácie.

#### 5.1.1 Štruktúra navrhovanej aplikácie

Pri návrhu štruktúry aplikácie bol kladený dôraz na voľbu typu architektúry, tak aby výsledný systém bol čo najľahšie udržiateľný a v budúcnosti bolo jednoduché dotvárať novú funkcionálnosť. Z toho dôvodu bol pre navrhovanú aplikáciu použitý návrhový model *Model-View-Controller*, čo môžeme vidieť aj na obrázku 5.1. Na danom diagrame sú zobrazené jednotlivé vrstvy návrhového modelu a triedy, ktoré reprezentujú jednotlivé vrstvy architektúry aplikácie.



Obr. 5.1: Model reprezentujúci architektúru *Model-View-Controller* implementovanej aplikácie [zdroj: vlastná tvorba].

## 5.2 Výpočet správnej polohy vkladaneho objektu

Spracovanie GPS súradníc užívateľa a samotný výpočet správnej polohy vkladaneho objektu je jednou z hlavných funkcionalít aplikácie.

Na to, aby bola aplikácia schopná vypočítať kde sa presne bude nachádzať vkladaneý objekt v rozšírenej realite, je potrebné určiť polohu užívateľa. GPS poloha užívateľa bude použitá ako referenčná poloha, na ktorú bude vždy pri spustení rozšírenej reality vložený počiatočný súradnicový systém. Polohu užívateľa získava aplikácia pomocou triedy `CLLocationManager`, ktorá je súčasťou knižnice ***CoreLocation***. `CLLocationManager` umožňuje aplikácii konfiguráciu lokalizačného modulu, pričom detailnejšiu ukážku danej konfigurácie môžeme vidieť v časti kódu 2.

```

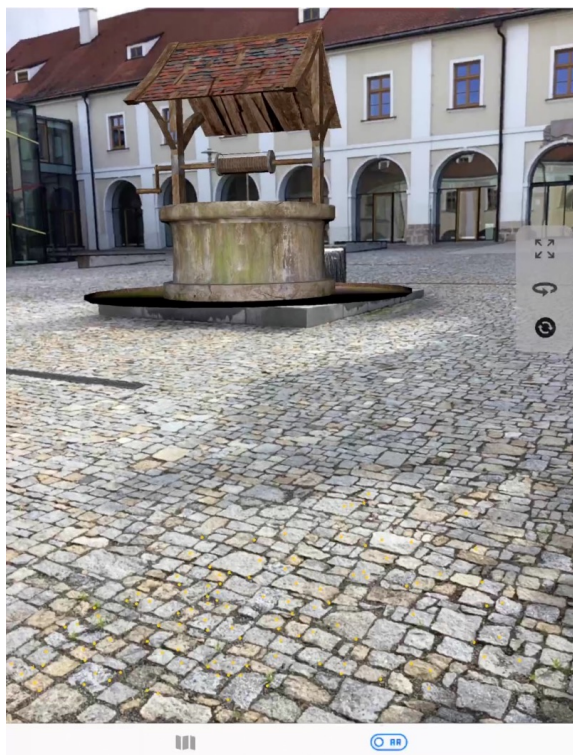
1 //Location Manager confoguration
2 locationManager.desiredAccuracy =
3     kCLLocationAccuracyBestForNavigation
4 locationManager.distanceFilter = kCLLocationDistanceFilterNone
5 locationManager.headingFilter = kCLHeadingFilterNone
6 locationManager.pausesLocationUpdatesAutomatically = false
7 locationManager.delegate = self

```

Kód 2: Nastavenie lokačného servisu pred spustením zaseliana polôh zariadenia [zdroj: vlastná tvorba].

Kód v ukážke 2 predstavuje základné nastavenie lokalizačného modulu. Ako môžeme vidieť pri prvom parametri, tak aplikácia nastavuje presnosť polohy získanej z lokalizačného modulu na hodnotu *kCLLocationAccuracyBestForNavigation*. Tento parameter nastavuje lokalizačný model na získavanie GPS súradníc pomocou všetkých dostupných senzorov na danom zariadení. Ďalej je možné vidieť, že aplikácia nastavuje parameter **headingFilter** na hodnotu *kCLHeadingFilterNone*. To je z dôvodu, aby bola aplikácia oboznámená o každej zmene polohy užívateľa, čo môže viesť k väčšej presnosti určenia pozície.

Aplikácia bude pre zobrazovanie virtuálnych objektov používať karteziánsku súradnicovú sústavu. Počiatok tejto sústavy reprezentuje počiatočnú polohu užívateľa, ktorá bola zachytená pri spustení rozšírenej reality. Samotný prepočet medzi GPS súradnicami a súradnicami karteziánskej sústavy je implementovaný pomocou metódy *GeodeticToEcef* v triede **ARModel**. Táto metóda je v aplikácii použitá na prepočet GPS súradníc užívateľa ako aj na prepočet súradníc vkladáneho objektu. Výsledok danej metódy je rozdiel vypočítaných súradníc, ktorý predstavuje výslednú polohu zobrazovaného objektu v karteziánskej súradnicovej sústave.



Obr. 5.2: Ukážka vloženia objektu na správnu polohu v reálnom priestore [zdroj: vlastná tvorba].

## 5.3 Interakcia s virtuálnymi objektami

Zo špecifikácie požiadavkov na systém vyplýva, že systém musí podporovať určitú interakciu užívateľa s objektami, ktoré sú zobrazované v rozšírenej realite. Samotná funkcionálna je postavená na rozpoznávaní jednotlivých gest, ktoré užívateľ vykonáva pohybom prstov po obrazovke. Pre detekciu pohybu prstov po obrazovke využíva aplikácia modul ***UIPanGestureRecognizer***, ktorý je dostupný v knižnici ***UIKit***. Nastavenie jednotlivých parametrov daného modelu je zobrazené v ukážke kódu 3.

```
1 let panGesture = UIPanGestureRecognizer(target: gestureModel ,
2         action: #selector(GestureModel.swipe(_:)))
3 self.sceneView.addGestureRecognizer(panGesture)
4 self.sceneView.isUserInteractionEnabled = true
5 self.sceneView.isMultipleTouchEnabled = true
```

Kód 3: Nastavenie modulu pre rozpoznávanie gést vykonaných užívateľom na obrazovke zariadenia [zdroj: vlastná tvorba].

O samotné spracovanie informácií získaných z vyššie uvedeného modulu sa v systéme stará trieda ***GestureModel***. Táto trieda má za úlohu filtrovať gesta, pre ktoré nie je definovaná v systéme žiadna vyššie špecifikovaná funkcionálna. Pri spracovaní jednotlivých informácií o pohybe prsta po obrazovke systém rozoznáva dva základné stavy :

- ***UIGestureRecognizer.State.begin***: stav reprezentujúci počiatočné súradnice, ktoré modul zachytil pri posune prsta po obrazovke
- ***UIGestureRecognizer.State.changed***: stav reprezentujúci súradnice bodu na obrazovke, ktorý nereprezentuje počiatočný ani koncový bod daného gesta.

Tieto stavy sa pri spracovaní gesta cyklicky prepisujú. Systém si vždy ukladá informáciu o počiatku aktuálne spracovávaného gesta. Pri získaní informácií o zmene, ktorú reprezentuje stav "changed", systém prepočíta zmenu pozície alebo rotácie objektu a následne nahradí pôvodný počiatočný stav aktuálnym. Vďaka tomu vie systém docieľiť priebežnú zmenu pozície alebo rotácie objektu.

### 5.3.1 Posun Objektu po rovnobežnej ploche s displejom zariadenia

Na to, aby bol systém schopný posunu objektu po rovnobežnej ploche s obrazovkou zariadenia je potrebné získať informácie o aktuálnom natočení obrazovky voči osiam súradnicovej sústavy. Informácie o natočení poskytuje aplikácii objekt ***ARSCNView*** v parametri ***pointOfView*** ako je zobrazené na ukážke 4.

```
1 sceneView.pointOfView?.simdEulerAngles.x
2 sceneView.pointOfView?.simdEulerAngles.y
3 sceneView.pointOfView?.simdEulerAngles.z
```

Kód 4: Získanie informácií o natočení zariadenia voči osiam súradnicovej sústavy. Jednotlivé hodnoty objektu ***simdEulerAngles*** predstavujú natočenie voči danej osi [zdroj: vlastná tvorba].

Na základe vyššie uvedených informácií a polohy objektu v priestore vytvára systém plochu, ktorá prechádza osou 3D objektu a je rovnobežná s obrazovkou zariadenia. Vytvorená plocha je v systéme reprezentovaná pomocou triedy *Plane*. Po vypočítaní danej plochy systém určuje vektor smerujúci od užívateľa do zvoleného bodu. Pozíciu užívateľom zvoleného bodu určuje systém pomocou funkcie *hitTest*. Táto funkcia je súčasťou knižnice *ARKit*. Ako posledný krok, ktorý robí systém pri určovaní novej pozície objektu, je vypočítanie priesečníku plochy objektu a vektora smerujúceho z obrazovky do zvoleného bodu. Na obrázku 5.3 je možné vidieť reálnu ukážku ako vyzerá posun daného objektu. Objekt na pravom obrázku predstavuje pozíciu objektu pred posunom na správne miesto, zatiaľ čo objekt na druhom obrázku predstavuje výslednú pozíciu objektu v scéne po zmene polohy užívateľom.



Obr. 5.3: Ukážka zmeny pozície objektu v scéne. Pravý obrázok popisuje počítačnú pozíciu objektu v scéne pred zmenou pozície užívateľom. Ľavý obrázok reprezentuje výslednú polohu [zdroj: vlastná tvorba].



### 5.3.2 Rotácia objektu okolo svojej osi

Rotácia objektu predstavuje popri zmene pozície objektu ďalšiu funkcionálnosť aplikácie, kedy dochádza k interakcii užívateľa s objektom zobrazeným v rozšírenej realite. Na základe návrhu aplikácie bol implementovaný spôsob rotácie objektu okolo osi, kedy aplikácia zachytáva vybrané gestá pohybu prstov na obrazovke a následne mení uhol 3D modelu.

Podobne ako pri spracovaní jednotlivých gest pri zmene polohy objektu, tak aj pri rotácii aplikácia využíva na detekovanie pohybu prsta po obrazovke modul *UIPanGestureRecognizer*. Užívateľ jednoduchým pohybom prsta vodorovne smerom doprava a doľava po obrazovke určuje veľkosť uhla o ktorý bude daný objekt rotovaný a taktiež aj smer rotácie. Samotná úprava uhla objektu nachádzajúceho sa v rozšírenej realite je realizovaná pomocou knižnice ARKit nastavením hodnôt *eulerAngles*, ako je zobrazené na ukážke 5.



Obr. 5.4: Ukážka rotácia objektu okolo svojej osi. Jednotlivé obrázky zobrazujú natočenie objektu pred a po užívateľom zvolenom otočení, pričom pravý obrázok predstavuje pozíciu objektu pred natočením, zatiaľ čo ľavý obrázok predstavuje natočenie objektu skončení rotácie [zdroj: vlastná tvorba].

```
1 let node = self.sceneView.scene.rootNode.childNode(  
2     withName: object.name, recursively: true)!  
3 let oldOrientation = node.eulerAngles  
4 node.eulerAngles = SCNVector3((oldOrientation.x),  
5     (oldOrientation.y) + angle,  
6     (oldOrientation.z))
```

Kód 5: Rotovanie objektu okolo svojej osi nastavením uhla zvieraného s osou y [zdroj: vlastná tvorba].

## 5.4 Ukladanie 3D modelov na zariadení užívateľa

Distribúcia 3D modelov medzi užívateľmi bola v návrhu aplikácie určená cez zdieľaný databázový systém. Jedná sa o jednoduchý princíp zdieľania informácií medzi užívateľov vďaka čomu má každý jeden jednoduchý prístup k novým objektom. Ako referenčná databáza pre ukladanie objektov bola zvolená databáza *FireBase*. K tejto voľbe došlo z dôvodu, že daná databáza poskytuje potrebné *API* rozhranie, ktoré dokáže komunikovať s iOS aplikáciou cez zabezpečený kanál. Ďalej poskytuje daná databáza rozsiahlu dokumentáciu, kde sa nachádza detailný popis integrácie spomínanej databázy s mobilnými aplikáciami bežiacich na platforme iOS.

### 5.4.1 Komunikácia s databázou

Databáza *Firestore* poskytuje rôzne možnosti pre uloženie dát. Ako bolo spomenuté v návrhu aplikácie, tak jednotlivé modely sú uložené v cloude *Storage* a informácie o jednotlivých modeloch sú dostupné v *RealTimeDatabase*. Z pohľadu aplikácie sa o komunikáciu s databázou stará trieda *DatabaseService*. Pri načítaní hlavnej obrazovky aplikácie dochádza k automatickej aktualizácii informácií o dostupných modeloch z databázy, kedy dochádza k získaniu dostupných JSON záznamov pomocou metódy zobrazenej v ukážke 6. Sťahovanie objektov prebieha dotazom na referenciu databázy. *API* databázy *Firestore* poskytuje niekoľko metód pre stiahnutie snapshotu *RealTime* databázy. Pre konkrétny prípad bola zvolená metóda *observeSingleEvent*, pretože sa jedná o špecifickú metódu určenú pre načítavanie dát pri inicializácii grafického rozhrania, keďže dochádza k získaniu informácií, ktoré sa po stiahnutí okamžite premietnu užívateľovi na základnej obrazovke aplikácie.

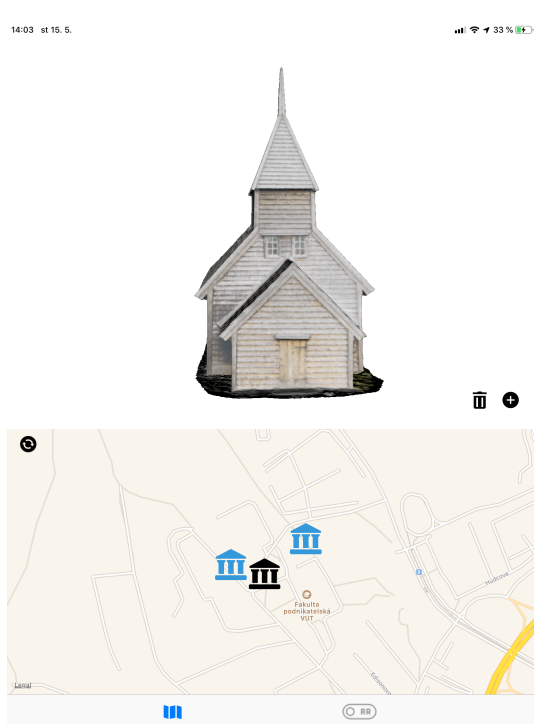
```
1  ref.child("scenes").observeSingleEvent(of: .value,
2    with: { (snapshot) in
3      let value = snapshot.value as? NSArray
4      for obj in value ?? [] {
5        guard let ns_scene = obj as? NSDictionary else {
6          continue
7        }
8        let scene = self.parseScene(scene: ns_scene)
9        scenes.append(scene)
10     }
11     self.delegateCaller(scenes: scenes)
12   }) { (error) in
13     self.errorHandling(error: error)
14   }
```

Kód 6: Stiahnutie informácií o objektoch, ktoré sú dostupné z databázy. Ako môžeme vidieť, tak hneď po stiahnutí dochádza ku konvertovaniu dát z obecného typu *Data* na typ *NSArray*. Toto pretypovanie je potrebné, aby samotná aplikácia neskôr dokázala čítať získané dáta [zdroj: vlastná tvorba].

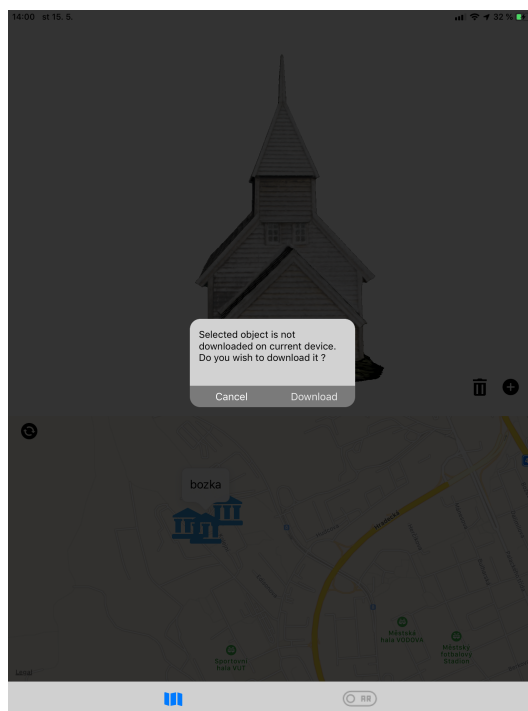
Po stiahnutí JSON záznamov je daný *Snapshot* konvertovaný do triedy *Scenes*, ktorá bola vytvorená za účelom serializácie a deserializácie objektov obsahujúcich informácie o jednotlivých 3D modeloch. Objekty typu *Scenes* v sebe obsahujú informáciu, či 3D

modely, ktoré reprezentujú, sú lokálne stiahnuté na konkrétnom zariadení alebo sa nachádzajú len v databáze. Na základe tejto informácie následne systém rozlišuje medzi týmito dvoma stavmi, v ktorom sa môžu jednotlivé objekty nachádzať.

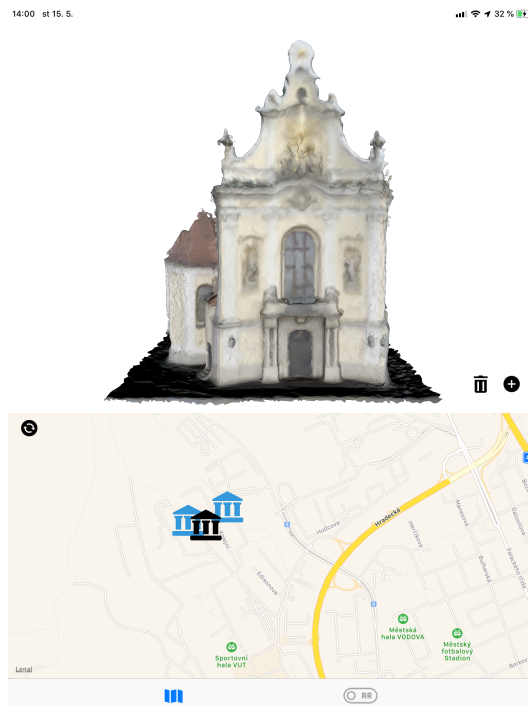
Stiahovanie 3D objektov z databázy *FireBase* vždy prebieha na žiadosť užívateľa. Aby aplikácia predchádzala určitým nedorozumeniam, tak predtým než dôjde k samotnému stiahnutiu objektu z databázy je užívateľ upozornený dialógovým oknom 5.6 na to, že objekt ktorý sa chystá zobraziť nie je stiahnutý na jeho zariadení a pre jeho zobrazenie je nutné daný objekt stiahnuť.



Obr. 5.5: Hlavná obrazovka aplikácie, ktorá zobrazuje na mape 3D modely, ktoré sú užívateľovi k dispozícii uložené v databáze.[Vlastná tvorba]



Obr. 5.6: Ukážka dialógového okna, ktoré sa objaví na obrazovke len vtedy, ak sa užívateľ snaží zobraziť objekt, ktorý nie je stiahnutý na jeho zariadení [zdroj: vlastná tvorba].



Obr. 5.7: Stav hlavnej obrazovky po stiahnutí užívateľom zvoleného objektu. Po dokončení sťahovania zmizne z obrazovky dialógové okno a v hornej časti určenej pre zobrazenie vybraného modelu sa otvorí práve stiahnutý model [zdroj: vlastná tvorba].

Samotné stiahnutie 3D objektu je implementované pomocou metódy *getData*, ktorá je zavolaná nad odkazom na Storage databázu pričom ako jeden z hlavných parametrov pre danú metódu je maximálna veľkosť súboru, ktorý bude sťahovaný. Po stiahnutí objektu systém daný objekt najprv uloží na dané zariadenie a následne ho konvertuje do požadovaného typu *SCNSceneSource*.

#### 5.4.2 Ukladanie objektov do lokálnej pamäte

Ukladanie jednotlivých dát, či už sa jedná o informácie o objektoch alebo o samotné objekty, je implementované v triede *FileService*. Táto trieda má na starosti všetky operácie, ktoré sa týkajú práce s informáciami uloženými na zariadení konkrétneho užívateľa.

Pre samotné ukladanie dát využíva aplikácia modul *FileManager*, ktorý je súčasťou knižnice *Foundation*. Ako je možné vidieť v ukážke 7, tak pri čítaní dát zo súboru dochádza k deserializácii informácii pomocou triedy *JSONDecoder*. Výsledok deserializácie je objekt typu *Scene*. Funkcie pre čítanie dát sa nachádzajú v bloku *do-catch*. Vďaka tomu je systém schopný odchytiť výnimku, ktorá môže byť vyvolaná pri čítaní dát zo súboru alebo pri deserializácii. Načítavanie jednotlivých scén je v aplikácii riešené veľmi jednoduchým spôsobom pomocou funkcie *SCNSceneSource*. Táto funkcia má na vstupe URL parameter, ktorý odkazuje na konkrétne miesto v pamäti, kde sa dané dáta nachádzajú.

```

1  func readJSON(fileName: String) -> Scene{
2      let fileURL = createJSONFileURL(fileName: fileName)
3      var result : Scene?
4      do{
5          let data = try Data(contentsOf: fileURL)
6          let decodedScene = try JSONDecoder()
7              .decode(Scene.self, from: data)
8          result = decodedScene
9      }catch{
10         handleReadError()
11     }
12     return result!
13 }

```

Kód 7: Funkcia pre čítanie informácií o jednotlivých 3D modeloch zo súboru. Po načítaní dát dochádza dekodovaniu informácií pomocou triedy *JSONDecoder* [zdroj: vlastná tvorba].

Zápis informácií o jednotlivých scénach prebieha veľmi podobne ako čítanie, ktoré bolo popísané vyššie. V prípade JSON súborov sa jedná o serializáciu dát a následný zápis do pamäte. Zápis jednotlivých 3D modelov do pamäte je v objekte *FileModel* implementovaný vo funkcii *saveScene*. Na vstupe danej funkcie sa nachádza 3D model typu *Data* a lokácia na ktorú bude zapísaný daný model.

## Kapitola 6

# Testovanie výslednej aplikácie

Cieľom tejto diplomovej práce bolo vytvoriť mobilnú aplikáciu, ktorá bude svojim užívateľom poskytovať zážitok zo zobrazovania historických budov alebo iných objektov v rozšírenej realite. Miera zážitku aplikácie je ovplyvnená tým, ako veľmi spokojný je užívateľ s grafickým rozhraním a funkcionalitou aplikácie. Preto je dôležité otestovať aplikáciu na koncovom užívateľovi, vďaka ktorému je možné získať reálne ohodnotenie vytvorenej aplikácie. Na základe získanej spätnej väzby je možné určiť či daná implementácia aplikácie obsahuje skryté nedostatky a či sa jedná o rozsiahle chyby alebo skôr o malé nedostatky, ktoré môžu byť jednoducho odstránené v ďalšej iterácii vývoja aplikácie.

### 6.1 Vytvorenie testovacej sady objektov

Jednotlivé objekty použité pri testovaní boli získané z verejne dostupných databáz 3D modelov<sup>1</sup> alebo boli vytvorené z fotiek pomocou technológie fotogrametrie. Na to aby bolo možné tieto objekty používať v aplikácii, museli byť konvertované do formátu .dae. Jedná sa o formát umožňujúci uloženie jednotlivých objektov spolu s informáciami o textúre. Následne po exportovaní do formátu .dae bolo potrebné pomocou nástroja Xcode konvertovať tieto súbory do formátu .scn, ktorý je podporovaný knižnicou ARKit.

### 6.2 Návrh testovacieho scenáru

Testovanie bolo zamerané na orientáciu užívateľa v grafickom rozhraní aplikácie. Na to aby výsledky boli čo najviac relevantné, tak testovacia skupina pozostávala z technicky zameraných užívateľov ako aj z užívateľov, ktorý mali väčšie technické zázemie. Keďže aplikácia je cieleňá ma ľudí v aktívnom veku, tak testovacia vzorka užívateľov bola vo vekovom rozpätí 16-30 rokov.

Základným faktorom, ktorý bol skúmaný pri testovaní aplikácie je práca užívateľa s metódami určenými pre interakciu užívateľa s objektom zobrazeným v rozšírenej realite. To znamená, že bol kladený dôraz na správnu voľbu gest, pomocou ktorých užívateľ ovládal jednotlivé metódy. Ďalším faktorom bolo sledovanie citlivosti metód, konkrétne ako reagujú na signály, ktoré im zadá užívateľ pohybom prstov po obrazovke. Výsledok tohto testovania potvrdí alebo vyvráti správnosť návrhu jednotlivých metód pre interakciu užívateľa s objektami v rozšírenej realite.

---

<sup>1</sup><https://free3d.com>

Ako posledný faktor, ktorý bol skúmaný pri testovaní je samozrejme orientácia užívateľa v grafickom rozhraní aplikácie. To znamená, že do akej miery užívateľ rozumie jednotlivým prvkov grafického rozhrania a ich samotnej pozícii na obrazovke. Pri tomto faktore bol sledované ako rýchlo sa užívateľ dokázal zorientovať v aplikácii.

Samotné testovanie prebiehalo v dvoch fázach. V prvej časti dostal užívateľ do ruky tablet a jeho úlohou bolo spustiť aplikáciu, vybrať si objekt ktorý chcel zobrazit v rozšírenej realite a bol zároveň v jeho blízkosti. Pri plnení daného testovacieho scenára bol sledovaný čas za aký užívateľ správne splnil vyššie špecifikované ciele. Pričom je dôležité aby sa pri vykonávaní tohto testovacieho scenára jednalo o prvý kontakt užívateľa s aplikáciou.

Druhá fáza testovania pozostávala z práce užívateľa s metódami pre zmenu pozície objektu v scéne. Úlohou užívateľa bolo presunutie objektu na vopred ukázanú správnu pozíciu, pričom samotný presun objektu zahrňoval aj správne natočenie objektu voči užívateľovi. Pri vykonávaní daného testovacieho scenára bol sledovaný čas, za ktorý užívateľ presunul daný virtuálny objekt na správnu pozíciu. Tento testovací scenár zahrňoval komplikovanejšiu prácu užívateľa s aplikáciou, z toho dôvodu bol vykonávaný až po prvej fáze testovania.

### **6.3 Testovanie aplikácie na koncových užívateľoch**

Navrhnuté testovacie scenáre boli vykonané na náhodnej vzorke ľudí, ktorí spĺňali vyššie uvedené požiadavky a boli ochotní sa podrobiť testovaniu. Zúčastnení užívatelia predstavovali skupinu cca 10tich ľudí.

#### **6.3.1 Prvá fáza testovania**

Prvá fáza testovania predstavovala analýzu orientácie užívateľa v grafickom rozhraní aplikácie. Testovanie prebiehalo tak, že užívateľ dostal do rúk tablet a bola mu zadaná úloha zobrazenia vybraného objektu v rozšírenej realite. Jednalo sa o prvý kontakt užívateľa s aplikáciou. Tým, ktorým nebola aplikácia pred testovaním detailne predstavená, potrebovali určitý čas na orientáciu v systéme. Títo užívatelia mali tendenciu skúsiť najprv jednotlivé grafické prvky zobrazené na displeji zariadenia a až potom sa venovali zadanej úlohe. To znamená, že v budúcnosti by bolo dobré doimplementovať v aplikácii možnosť nápovedy, ktorá by usmerňovala užívateľov pri používaní aplikácie. Užívatelia, ktorým bola aplikácia pred danou testovacou fázou predstavená dosahovali dobrú orientáciu a nemali problém splniť zadanú úlohu.

#### **6.3.2 Druhá fáza testovania**

V druhej fáze testovania sa užívatelia venovali presunu objektu na zvolenú pozíciu. Ako bolo spomenuté v úvode kapitoly, tak samotný presun v sebe zahrňoval aj rotáciu objektu okolo vlastnej osi. V tejto časti testovania boli výsledky veľmi rôzne. Keďže počiatočná poloha objektu nebola vždy rovnaká, tak užívatelia nemali rovnaké počiatočné podmienky. Užívatelia, ktorí potrebovali posunúť objekt smerom k sebe alebo od seba mali zo začiatku problém vykonať daný pohyb. To bolo zapríčinené tým, že metóda pre zmenu polohy objektu predstavuje zmenu pozície po ploche rovnobežnej s displejom zariadenia. Táto komplikácia spôsobila pomalšie splnenie danej úlohy ale celkovo nezabránila k dosiahnutiu požadovaného výsledku.



## 6.4 Vyhodnotenie testov

Testovanie aplikácie bolo vyhodnotené ako úspešné, keďže nebol nájdený žiadny závažný nedostatok v implementácii systému. Užívatelia, ktorí sa podrobili testovaniu podali kladnú spätnú väzbu a s aplikáciou ako takou boli veľmi spokojní. Medzi tie najväčšie nedostatky, ktoré boli odhalené testovaním radíme problém zmeny pozície objektu smerom k alebo od užívateľa. Pri danej implementácii metódy zmeny pozície objektu nie je tento pohyb nemožný ale užívateľom nie je prirodzený, a je potrebný určitý čas na to, aby si na daný pohyb objektom zvykli. Pre zvýšenie spokojnosti užívateľa by bolo dobré v budúcnosti na danom probléme zapracovať a vyriešiť ho úpravou metódy pre zmenu pozície objektu v scéne. Pri zmene natočenia objektu neboli zaznamenané žiadne komplikácie, a preto môžeme povedať, že táto metóda bola implementovaná úspešne.

Pri testovaní grafického rozhrania aplikácie užívatelia narazili na pár nedostatkov. Ako prvý nedostatok bol identifikovaný pri používaní jednotlivých módov úpravy pozície 3D objektu v scéne. Užívatelia častokrát chceli po zmene pozície upraviť natočenie objektu, ale neprepli aplikácie do patričného módu. Z toho dôvodu došlo pri úprave grafického rozhrania a k dôraznejšiemu zvýrazneniu aktívneho módu pre úpravu 3D objektu.

# Kapitola 7

## Záver

Cieľom diplomovej práce bolo navrhnutie a následné implementovanie aplikácie, ktorá pomocou rozšírenej reality zobrazuje užívateľovi historickú podobu bodov alebo iných objektov nachádzajúcich sa v našom každodennom okolí. Táto aplikácia mala byť špecifická tým, že zobrazované objekty sa vždy budú nachádzať na ich reálnej polohe, tak aby užívateľ s čo najmenším príspevom dosiahol požadovanú vizualizáciu vybraného 3D objektu.

Ako prvý krok pri navrhovaní aplikácie bolo oboznámenie sa s danou problematikou. To znamená, že bolo potrebné preštudovanie literatúry týkajúcej sa rozšírenej reality a existujúcich frameworkov pre prácu s rozšírenou realitou. Na základe získaných informácií bola vybraná platforma iOS pre navrhovanú aplikáciu. Vybraná platforma poskytuje pre prácu s rozšírenou realitou knižnicu *ARKit*. Na základe vybranej knižnice bola navrhnutá funkcionálnosť aplikácie, ktorá bola popísaná v kapitole 3. Návrh aplikácie približuje funkcionálnosť, ktorá je potrebná pre splnenie hlavného prípadu použitia aplikácie. Najkomplikovanejšiu časť návrhu predstavovali metódy pre výpočet správnej polohy objektu v rozšírenej realite a spôsob zmeny pozície objektu po ploche rovnobežnej s displejom užívateľského zariadenia.

Na základe návrhu bola implementovaná výsledná aplikácia s názvom LookAR. V aplikácii boli implementované jednotlivé metódy spomenuté v návrhu aplikácie. Výsledná aplikácia obsahuje dve základné obrazovky. Prvá obrazovka, ktorá sa zobrazí užívateľovi po spustení aplikácie, je určená pre výber 3D objektu. Užívateľ má možnosť stiahnutia nových objektov z databázy *Firestore*, alebo ich zmazanie z lokálnej pamäte. Druhá obrazovka aplikácie poskytuje užívateľovi obraz z kamery obsahujúci rozšírenú realitu o vložený 3D objekt. Ako vyplýva z návrhu aplikácie, tak užívateľ má možnosť úpravy polohy a natočenia objektu.

Vytvorená aplikácia bola testovaná na náhodne vybranej vzorke oslovených ľudí. Testovanie ukázalo malé nedostatky pri návrhu metódy pre posun objektu. Nejednalo sa však o závažnú chybu. Z celkového hľadiska dopadlo testovanie dobre a užívateľa zanechali pozitívnu spätnú väzbu na vytvorenú aplikáciu. Z celkového hľadiska je prácu možné považovať za veľmi prínosnú, keďže vytvorená aplikácia má veľký potenciál v rôznych odvetviach ako napríklad školstvo alebo stavebníctvo.

# Literatúra

- [1] *10 industries utilizing augmented reality* . [Online; navštívené 25.4.2019].  
URL <https://www.clickz.com/10-ways-your-industry-uses-augmented-reality/214953/>
- [2] *ARConfiguration*. [Online; navštívené 23.12.2018].  
URL <https://developer.apple.com/documentation/arkit/arconfiguration>
- [3] *ARSession*. [Online; navštívené 23.12.2018].  
URL <https://developer.apple.com/documentation/arkit/arsession>
- [4] *Augmented Reality In Social Media, Marketing, Mobile Apps and More* . [Online; navštívené 7.1.2019].  
URL <https://www.gc-solutions.net/blog/augmented-reality-in-social-media-marketing-mobile-apps-and-more/>
- [5] *Augmented reality iPad app used in liver surgery to aid tumour resection* . [Online; navštívené 28.4.2019].  
URL <https://www.imedicalapps.com/2013/08/augmented-reality-ipad-app/>
- [6] *BBC Civilisations AR: This app is like a virtual exhibition with over 30 curated artefacts* . [Online; navštívené 7.1.2019].  
URL <https://economictimes.indiatimes.com/magazines/panache/bbc-civilisations-ar-this-app-is-like-a-virtual-exhibition-with-over-30-curated-artefacts/articleshow/63345134.cms>
- [7] *Best AR SDK for development for iOS and Android in 2018* . [Online; navštívené 23.12.2018].  
URL <https://thinkmobiles.com/blog/best-ar-sdk-review/>
- [8] *Civilisations AR* . [Online; navštívené 23.12.2018].  
URL <https://itunes.apple.com/us/app/civilisations-ar/id1350792208#?platform=ipad>
- [9] *CONVERSION OF GPS DATA TO CARTESIAN COORDINATES VIA AN APPLICATION DEVELOPMENT ADAPTED TO A CAD MODELLING SYSTEM* . [Online; navštívené 21.4.2019].  
URL [https://www.researchgate.net/publication/30871852\\_CONVERSION\\_OF\\_GPS\\_DATA\\_TO\\_CARTESIAN\\_COOR-DINATES\\_VIA\\_AN\\_APPLICATION\\_DEVELOPMENT\\_ADAPTED\\_TO\\_A\\_CAD\\_MODELLING\\_SYSTEM](https://www.researchgate.net/publication/30871852_CONVERSION_OF_GPS_DATA_TO_CARTESIAN_COOR-DINATES_VIA_AN_APPLICATION_DEVELOPMENT_ADAPTED_TO_A_CAD_MODELLING_SYSTEM)

- [10] *GPS SYSTEMS LITERATURE: INACCURACY FACTORS AND EFFECTIVE SOLUTIONS* . [Online; navštívené 21.4.2019].  
URL <http://www.airconline.com/ijcnc/V8N2/8216cnc11.pdf>
- [11] *HOW TO BRING AUGMENTED REALITY TO YOUR SCHOOLS* . [Online; navštívené 25.4.2019].  
URL <https://www.blippar.com/blog/2016/10/20/how-bring-augmented-reality-your-school>
- [12] *Ikea Place*. [Online; navštívené 23.12.2018].  
URL <https://itunes.apple.com/us/app/ikea-place/id1279244498?mt=8>
- [13] *Introducing ARKit: Augmented Reality for iOS* . [Online; navštívené 7.1.2019].  
URL <https://developer.apple.com/videos/play/wwdc2017/602/>
- [14] *Key benefits of Augmented Reality for Architecture Projects* . [Online; navštívené 26.4.2019].  
URL <https://www.augment.com/blog/key-benefits-augmented-reality-architecture-projects/>
- [15] *Troposphere-Ionosphere Interactions* . [Online; navštívené 25.4.2019].  
URL <https://sess.stanford.edu/talis>
- [16] Audi, A.; Deseilligny, M.; Meynard, C.; aj.: Implementation of an IMU Aided Image Stacking Algorithm in a Digital Camera for Unmanned Aerial Vehicles. *Sensors*, ročník 17, 07 2017: str. 1646, doi:10.3390/s17071646.
- [17] Dieter, S.: *Augmented Reality, principles and practices*. Pearson Addison Wesley Prof, 2015, ISBN 0321883578.
- [18] Piao, J.-C.; Kim, S.-D.: Adaptive Monocular Visual-Inertial SLAM for Real-Time Augmented Reality Applications in Mobile Devices. *Sensors*, ročník 17, 11 2017: str. 2567, doi:10.3390/s17112567.

## Príloha A

# Obsah priloženého pamäťového média

/	
_ srcXcode/ .....	Adresár obsahujúci zdrojové súbory mobilnej aplikácie
_ srcTex/ .....	Adresár obsahujúci zdrojové texty textu práce
_ documents/	
_ poster.pdf/ .....	Plagát propagujúci vytvorenú aplikáciu
_ videos/ .....	Adresár obsahujúci záznamy obrazoviek
_ DP_Promo.mp4 .....	Propagačné video vytvorenej aplikácie
_ dp.pdf .....	Text diplomovej práce

# Príloha B

## Plagát

**Rozšírená realita:  
Historické verzie  
budov**

**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ**

**Mobilná Aplikácia  
LookAR**

**AR**

*Rýchly výber spomedzi 3D objektov v okolí*

Ukážka vybraného 3D modelu

Mapa zobrazujúca 3D modely, ktoré si môže užívateľ zobrazit'

*Jednoduché sťahovanie nových objektov*

3D objekt vložený do Rozšírenej reality

Posun po ploche rovnobežnej s displejom zariadenia.

Rotácia 3D modelu okolo vlastnej osi

Možnosť znovu načítať aktuálne zobrazovanú scénu

*Zobrazenie objektov na ich reálnej pozícii*

Autor : Bc. Juraj Medvec  
Vedúci práce:  
prof. Ing. Adam Herout, Ph.D.  
2019

Obr. B.1: Plagát prezentujúci vytvorenú mobilnú aplikáciu [Vlastná tvorba].