ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Bakalářská práce

Správa obsahu v rozšířené realitě

Jan Borecký
Program: Otevřená informatika
Obor: Počítačové hry a grafika

Květen 2022
Vedoucí práce: Ing. David Sedláček, Ph.D.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Borecký** Jméno: **Jan** Osobní číslo: **492303**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**

Studijní program: **Otevřená informatika**

Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Správa obsahu v rozšířené realitě

Název bakalářské práce anglicky:

Content management in augmented reality

Pokyny pro vypracování:

1) Prostudujte možnosti rozšířené reality (AR) na platformě Android s využitím rozšíření pro sdílení souřadného systému mezi více zařízeními (tzv. cloud anchors).
2) Navrhněte a implementujte systém využívající minimálně dvou zařízení (telefon/AR brýle + telefon/tablet) pro umístění a verifikaci virtuálních objektů v rámci rozšířeně realitní scény.
3) Navrhněte a implementujete interakční postupy pro přesné umístění objektů ve scéně (např. gizmo u objektu, raycast z telefonu, raycast z brýlí + menu modelů na telefonu, využití jednoho zařízení jako reference, ...)
4) Navrhněte a realizujte metodu pro ověření kvality umístění objektů v AR (jak v rámci změny světelných podmínek, tak v závislosti na čase a změnách v místnosti). Dle této metody porovnejte realizace bodu 2 a 3.
5) Implementujte aplikaci pro správu virtuální výstavy v rozšířené realitě využívající vybraných interkačních technik. Aplikace umožní umístění exponátů (3D modely) v rámci místnosti, uložení a načtení potřebných dat pro správu a prohlížení na více zařízeních.
6) Aplikaci otestujte s alespoň pěti uživateli.

Seznam doporučené literatury:

1] Joseph J. LaViola, Jr. et all. 3D User Interfaces: Theory and Practice, second edition. 2017. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
2] Steve Aukstakalnis. Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability). Addison Wesley 2017.
3] Dieter Schmalstieg, and Tobias Hollerer. Augmented Reality: Principles and Practice (Usability), Addison Wesley 2016
4] online documentation for: ARcore and Cloud Anchors, ARfoundation, Unity as a Library, Immersal (https://immersal.com/).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. David Sedláček, Ph.D. katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **10.02.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. David Sedláček, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání Podpis studenta

Poděkování / Prohlášení

Velmi děkuji vedoucímu své bakalářské práce, Ing. Davidu Sedláčkovi, Ph.D. za jeho velkou trpělivost a nečekanou vlídnost, a to zejména ve chvílích, když se zrovna něco nedařilo realizovat.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškerou použitou literaturu a zdroje.

V Praze dne 20. května 2022

.....

Abstrakt / Abstract

Bakalářská práce se zabývá využitím možnosti sdílení souřadného systému na platformě Android mezi více zařízeními pro koncept rozšířeně realitních expozic Národního muzea. Srovnává nejznámější dostupné technologie pro sdílení souřadného systému. Předkládá návrh aplikace pro umístění a následnou modifikaci pozice, rotace a velikosti virtuálních objektů v rámci rozšířeně realitní scény. Dále na základě návrhu implementuje jednoduchou aplikaci v herním enginu Unity postavenou na technologii Google Cloud Anchors. Aplikace umožňuje sdílení zmapovaných scén, do kterých lze přidat a modifikovat objekty uložené na serveru. To vše sdílené s dalšími uživateli v reálném čase. V neposlední řadě práce hodnotí kvalitu mapování a persistenci scény napříč zařízeními.

Klíčová slova: Cloud Anchor, Firebase, Google, AR, rozšířená realita, Národní muzeum

Bachelor thesis deals with the use of the possibility of sharing the coordinate system between multiple devices on the Android platform for the concept of augmented reality exhibitions of the National Museum. It compares the best known technologies for coordinate system sharing and presents an application design for placing and subsequent modifying the position, rotation and size of virtual objects within the augmented reality scene. Furthermore, based on the design, it implements a simple Unity application based on Google Cloud Anchors technology. The application allows sharing mapped scenes, to which you can add and modify objects stored on a server. All shared with other users in real time. Last but not least, the work evaluates the quality of mapping and scene persistence across devices.

Keywords: Cloud Anchor, Firebase, Google, AR, augmented reality, National museum

Title translation: Content management in augmented reality

Obsah /

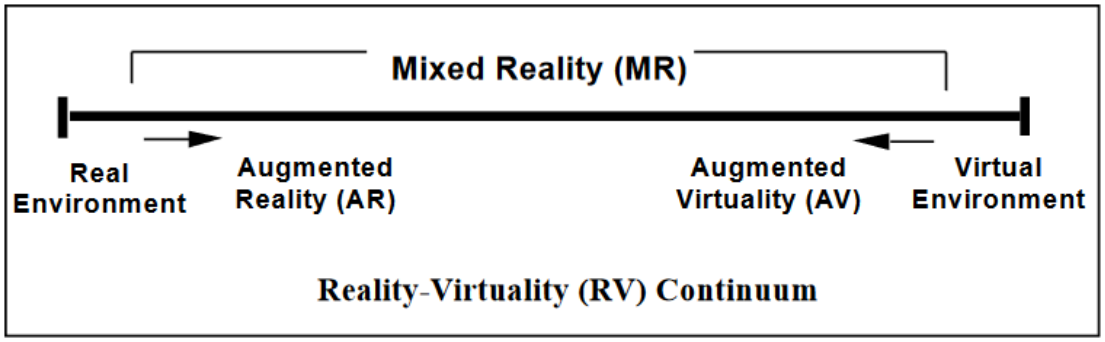
1 Úvod 1
1.1 Využití rozšířené reality 1
1.2 Struktura a obsah práce 2
2 Analýza a návrh řešení 3
2.1 Technologie pro sdílení souřadného systému 3
2.1.1 Azure Spatial Anchors 3
2.1.2 Stardust 4
2.1.3 ARwayKit 4
2.1.4 Google Cloud Anchors 5
2.1.5 Immersal 5
2.2 Vývojové prostředí 6
2.3 Návrh aplikace 7
2.3.1 Vytvoření/získání kotevního bodu 7
2.3.2 Práce s objekty 8
2.3.3 Grafické uživatelské rozhraní 9
2.3.4 Ukládání dat 11
3 Implementace 14
3.1 Nastavení a struktura projektu 14
3.2 Implementace grafického uživatelského rozhraní 16
3.3 Tvorba nového kotevního bodu 16
3.4 Rozpoznávání existujícího kotevního bodu 18
3.5 Tvorba, správa a mazání objektů 18
3.5.1 Vytvoření objektu ve scéně 18
3.5.2 Manipulace s objekty 19
3.5.3 Mazání kotevních bodů a objektů 19
3.6 Ukládání a vyhledávání dat 19
4 Testování 21
4.1 Testování persistence scény 21
4.1.1 Testování v muzeu 21
4.1.2 Testování v domácím interiéru 28
4.2 Testování uživatelské přívětivosti 29
5 Závěr 31
Literatura 32
A Návody ke spuštění 35
A.1 Aplikace 35
A.2 Unity projekt 35
B Dotazník uživatelského testování 36
B.1 Úkoly 36
B.2 Otázky a odpovědi 37

/ Obrázky

1.1. Realitně-virtuální kontinuum 1
2.1. Minecraft Earth 4
2.2. Webové rozhraní Immersal 6
2.3. Uživatelské rozhraní Unity.....7
2.4. Mapovací aplikace Immersal.....8
2.5. Rozhraní pro manipulaci s objekty v Unity 9
2.6. Rozhraní pro manipulaci s objekty v Blenderu.....9
2.7. Příklad 3D widgetu 10
2.8. Návrh grafického uživatelského rozhraní 11
2.9. Růst popularity databází 12
3.1. Balíčky v projektu 14
3.2. Nastavení XR Plug-in Management 15
3.3. GUI po zapnutí aplikace..... 16
3.4. GUI při selekci objektu 16
3.5. GUI po umístění objektu 16
3.6. Hostování kotevního bodu 17
3.7. Rozpoznání kotevního bodu ... 17
4.1. Model krakatice v Národním muzeu..... 22
4.2. Plánek místnosti s krakaticí ... 22
4.3. Místo hostování 1. kotevního bodu u modelu krakatice 23
4.4. Rozpoznání 2. kotevního bodu u modelu krakatice 23
4.5. Plánek pantheónu 24
4.6. První rozpoznání ve spodní části pantheónu 24
4.7. Výsledek rozpoznání na ochozu pantheónu 25
4.8. Druhé rozpoznání ve spodní části pantheónu 25
4.9. Třetí rozpoznání ve spodní části pantheónu 25
4.10. Výsledek rozpoznání skalky s mořským ptactvem..... 26
4.11. Referenční obrázek kostry plejtváka..... 27
4.12. První rozpoznání ve dvoraně .. 27
4.13. Plánek dvorany..... 28
4.14. Druhé rozpoznání ve dvoraně.. 28
4.15. Výsledky dotazníku 29

Kapitola 1
Úvod

V dnešní době je trendem náš reálný svět okolo nás nahrazovat interaktivním virtuálním prostředím, které si můžeme dle libosti navrhnout a měnit. Takový svět se nazývá virtuální realita (dále VR). Druhým trendem je pouze rozšiřovat náš reálný svět o virtuální prvky, což se nazývá rozšířená realita (dále AR). Mezistupněm mezi AR a VR je někdy označována rozšířená virtualita, ale dosud nejsou stanoveny přesné hranice, co je v dané kategorii, a co už není. Všechny tyto přístupy se zobrazují na takzvané realitně-virtuální kontinuum (Obrázek 1.1).



Obrázek 1.1. Zjednodušená reprezentace realitně-virtuálního kontinua. [1]

Abych mohl čtenáře obeznámit s problematikou sdílení souřadného prostoru v rozšířeně realitních scénách, je potřeba nejdřívě vysvětlit, co rozšířená realita je a proč má smysl se rozšířené realitě, respektive sdílení souřadného systému, věnovat.

Největší výhodou a zároveň problémem AR je, že uživatel vidí z větší části reálnou část scény oproti té virtuální. Výhoda je, že uživatel vidí reálný svět a tím je pro něj AR scéna velmi uvěřitelná, obzvláště pokud přidané virtuální objekty graficky dobře zapadají do reálného světa. Nevýhodou je technická a výpočetní náročnost tvorby virutálních prvků scény. Zařízení, které uživateli zobrazuje virtuální část scény, musí být velmi dobře zorientované v prostoru. Musí rovněž reagovat velmi rychle na každý pohyb uživatele. Např. při rychlém pohybu hlavy by se virtuální objekty měly stíhat posouvat a rotovat stejně, jako se posouvá a rotuje reálný svět kolem uživatele. Hardwarová řešení pro tyto problémy rychlosti a přesnosti umístění bývají i v dnešní technologicky velmi pokročilé době drahá a často nedostupná pro koncového uživatele. Často se tak v praxi nedostatek hardwaru kompenzuje chytrými softwarovými řešeními, která dokáží aproximovat potřebná data s využitím základní techniky, jako je např. kamera a inerciální měřící jednotka - kombinace akcelerometru, gyroskopu a magnetometru (zkráceně IMU).

1.1 Využití rozšířené reality

Díky zmíněné kompenzaci softwarovými řešeními se otevírá možnost oslovit mnohem větší množství potenciálních uživatelů. Převážná většina společnosti v dnešní době

vlastní chytrý mobilní telefon s fotoaparátem a IMU. Nebývá výjimkou kombinace více čoček s různými specifikacemi, např. hloubkový senzor nebo teleobjektiv. AR se tak dostala a stále dostává ke stále většímu množství uživatelů a nachází stále více možností uplatnění. Nejčastěji se můžeme setkat např. s obličejovými filtry v reálném čase (Instagram, Messenger), hrami (Pokémon GO) nebo aplikacemi pro rozvrhování a plánování prostor (IKEA Studio).

Poslední zmíněné kategorii se věnuje i tato bakalářská práce. Konkrétně se zabývá způsobem umístění virtuálních objektů do reálné scény, na které může v reálném čase spolupracovat více uživatelů. Myšlenka vznikla v Národním muzeu, kde zkoumají možnosti využití AR v muzejnictví, konkrétně tvorbu AR výstav. Muzeum navrhlo několik scénářů, ve kterých by se AR mohla uplatnit. V práci se těmto scénářům věnuji v sekci 4.1.

1.2 Struktura a obsah práce

Následující kapitola porovnává dostupná technologická řešení sdílení souřadného systému v AR. Dále v ní analyzuji existující aplikace, ve kterých je sdílení souřadného systému implementováno. Se získanými informacemi následně navrhuji vývojové prostředí a své řešení. V kapitole 3 prezentuji implementovanou aplikaci v herním enginu Unity. Kapitola 4 je rozdělená na dvě sekce. První se věnuje testování kvality persistence scény v domácích prostorách uživatelů, kteří aplikaci testovali, a v prostorách navrhnutých scénářů Národního muzea. Druhá ukazuje testování uživatelské přívětivosti, kde uživatelé hodnotí přívětivost aplikace, její obsluhu a možnosti.

Kapitola 2
Analýza a návrh řešení

Ze zadání bakalářské práce je nutno využít platformu Android, která však přináší několik výhod. První výhodou je rozšířenost a četnost uživatelů. S více než 70% zastoupením na trhu je zdaleka nejrozšířenějším operačním systémem pro mobilní zařízení na světě [2]. Druhou výhodou zmíněné platformy je velké množství technologií pro sdílení souřadného systému, které jsou dostupné právě pro Android. Neznamená to však, že každé Android zařízení umí pracovat s technologiemi pro sdílení souřadného systému.

2.1 Technologie pro sdílení souřadného systému

Všechny technologie umožňující sdílení souřadného systému se potřebují orientovat v prostoru. Na orientaci využívají IMU a data z kamery nebo více kamer. Při zpracování snímků z kamery je potřeba každý snímek analyzovat a získat z něj informace užitečné pro optimalizaci prostorové orientace zařízení, tzv. trackování. Nejčastěji se pro trackování využívá metoda SLAM (Simultaneous Localization And Mapping), která pracuje v tomto cyklu pro každý snímek [3]:

1. Detekce vizuálně zajímavých bodů (vysoký kontrast, roh, kulička v prostoru...) algoritmem pro detekci takových bodů, např. Harris corners nebo FAST.
2. Nalezení detekovaných bodů z předchozího snímku na aktuálním snímku. Zde mohou vstupovat data z IMU. Pokud se trackování nepodařilo, lze odhadnout pozici a rotaci pomocí RANSAC. (Random sample consensus - metoda pro odhad parametrů matematického modelu z množiny dat.)
3. Nalézt transformaci z předchozího snímku do aktuálního.
4. S pomocí SFM (Structure From Motion - estimace struktury podle změny pozice a rotace kamery) přidat trackované body do existující mapy okolí.

S pomocí SLAMu dochází k postupné tvorbě mapy okolí, kterou lze následně sdílet. Některé z technologií do této tvorby přidávají ještě umístění kotevního bodu uživatelem, díky kterému ostatní uživatelé ví, kam přibližně kamerou svého zařízení mířit pro co nejlepší nalezení mapy okolí a daného kotevního bodu.

V následujících sekcích stručně popisuji nejznámější a nejrozšířenější dostupné technologie pro sdílení souřadného systému.

2.1.1 Azure Spatial Anchors

Součástí cloudové platformy Azure od americké společnosti Microsoft je produkt Spatial Anchors. Hlavními přednostmi Spatial Anchors mají být [4]:

- práce v globálním měřítku - možnost udržovat miliony 3D objektů a pracovat s nimi
- připojení vlastních IoT (Internet of Things) dat - datové propojení se zaměstnanci v reálném čase
- multiplatformnost

Výše uvedená technologie nabízí nativní podporu pro herní engine Unity, Visual Studio v kombinaci s AR brýlemi Microsoft HoloLens, Android Studio, Xcode a Xamarin.

Mezi nejznámější aplikace využívající Azure Spatial Anchors patří AR hra pro mobilní zařízení Minecraft Earth (Obrázek 2.1), jejíž podpora skončila 30. června 2021 kvůli pandemii COVID-19. Používala zmíněné Spatial Anchors v kombinaci s OpenStreetMap (editovatelná geografická databáze světa) [5].

Obrázek 2.1. Záběry ze hry Minecraft Earth. Hra umožňovala do reálného světa přidat a postavit si objekty ze hry Minecraft. [zdroj obrázku: `https://www.actualapp.com/ya-puedes-descargar-apk-minecraft-earth-juega-antes-nadie-56087`]

2.1.2 Stardust

Hongkongská společnost Neogoma nabízí svůj SDK (Software Development Kit - balíček nástrojů pro vývoj) Stardust ve formě balíčku pro Unity. Předností Stardustu je multiplatformnost a fakt, že mu pro fungování stačí RGB kamera. K tomu nabízí možnost měnit namapované prostory podle změny světelných podmínek, počasí nebo množství lidí ve scéně [6]. Správu AR scén lze realizovat přes online editor, ve kterém může i designér bez hlubších technických znalostí rozmisťovat další objekty. Stardust využívá i Hobodream - druhý framework od Neogomy. Hobodream slouží ke standardizaci chování mezi všemi SDK od Neogomy a jeho součástí je např. rozhraní pro internetovou komunikaci, flexibilní lokalizační systém nebo implementace observer patternu.

2.1.3 ARwayKit

Vývojářský balíček od britské společnosti ARWAY má podobné možnosti jako výše zmíněný Stardust. Má vlastní aplikaci pro mapování prostředí, vlastní Web studio umožňující správu obsahu bez nutnosti programování a SDK pro Unity. (Brzy by měl být i pro Unreal, Android Studio a Xcode/Swift.) Využívá platformu ARCore pro Android zařízení, ARKit pro iOS zařízení a Azure Spatial Anchors pro HoloLens. Při mapování si v reálném čase ukládá všechny nalezené body do mračna bodů na cloudové uložiště. Poté během lokalizace zařízení odesílají data ze senzorů do nejbližšího cloudového uložiště a to navrací lokální pozici zařízení v předmapovaném prostředí. To by mělo vývojářům

umožnit integrovat lokalizační služby s minimální nutností programování [7]. V neposlední řadě má nativní podporu WebAR. (Technologie pro využívání AR bez nutnosti mít v mobilním zařízení dedikovanou aplikaci. Stačí připojení k internetu.)

2.1.4 Google Cloud Anchors

Cloud Anchor je součástí Google Cloud, což je cloudová platforma poskytující nástroje, služby a aplikace pro zvýšení produktivity a workflow. Obsahuje také cloudové uložiště, správu databází a mnohé další. Cloud Anchor funguje tak, že na uživatelem vybraném místě se vytvoří kotevní bod, kolem kterého uživatel naskenuje okolí a následně pošle veškerá data z kotevního bodu společně s informacemi o zmapovaném okolí přes Cloud Anchor API do databáze dle volby vývojáře. API neboli application programming interface je rozhraní pro interakci s vnitřními procesy aplikace ve formě funkcí a tříd. Jako vhodná databáze může velmi dobře posloužit služba Firebase od společnosti Google. Firebase je platforma pro tvorbu mobilních a webových aplikací, která podporuje nativní propojení Firebase projektu s Google Cloud projektem. Každý kotevní bod se zaznamenanými prostorovými informacemi má svůj identifikátor, na který se mohou odkazovat další uživatelé. Ti v oblasti, kde byl kotevní bod vytvořen, zažádají o napolohování daného kotevního bodu. Cloud Anchor sám začne mapovat okolí a porovnávat získané prostorové informace s daty uloženými v databázi. Pokud informace úspěšně spojí, zobrazí kotevní bod na místě, kde byl původním uživatelem vytvořen. Pro spolehlivé rozpoznání kotevního bodu je potřeba mít dostatečně zmapované okolí. K tomu má Cloud Anchor vlastní hodnocení Feature Map Quality (Kvalita mapy vizuálně významných bodů), které má 3 stavy:

- Insufficient (Nedostatečné) - Množství bodů je malé a pro aplikaci bude náročnější kotevní bod rozpoznat, natož správně naorientovat.
- Sufficient (Dostatečné) - Množství bodů je dostatečné a kotevní bod půjde dobře rozpoznat. Není ale zaručena úplně správná póza (pozice a rotace).
- Good (Dobré) - Aplikace rozpozná kotevní bod s velkou přesností a póza bude prakticky totožná s originálem.

Pro implementaci Cloud Anchor v Unity jsou potřeba tři věci. První je již zmíněný ARCore. Ve druhé řadě ARFoundation, což je framework přímo od Unity, který kombinuje základní vlastnosti ARKit, ARCore, Magic Leap a HoloLens. Umožňuje umisťování lokálních kotevních bodů, nemá však možnost tyto body sdílet s dalšími uživateli. K tomu slouží jako třetí ARCore Extensions balíček pro Unity, který konečně přidává funkcionalitu Cloud Anchors do Unity. Pro autorizaci aplikace a přístup k datům je potřeba nastavit autentifikaci. První možností autentifikace je API klíč, který stačí zadat do nastavení ARCore Extensions, ale zaručuje existenci Cloud Anchor na uložišti pouze po dobu 24 hodin od jeho vytvoření. Druhou možností je vytvoření OAuth klienta, který zaručuje existenci dat až na 365 dní.

ARCore má seznam zařízení, která podporuje [8]. U každého zařízení také uvádí, které z funkcí jsou na konkrétním mobilním zařízení dostupné. Jsou ale zařízení, která i přes to, že nejsou na seznamu podporovaných, mohou aplikace s Cloud Anchors spustit.

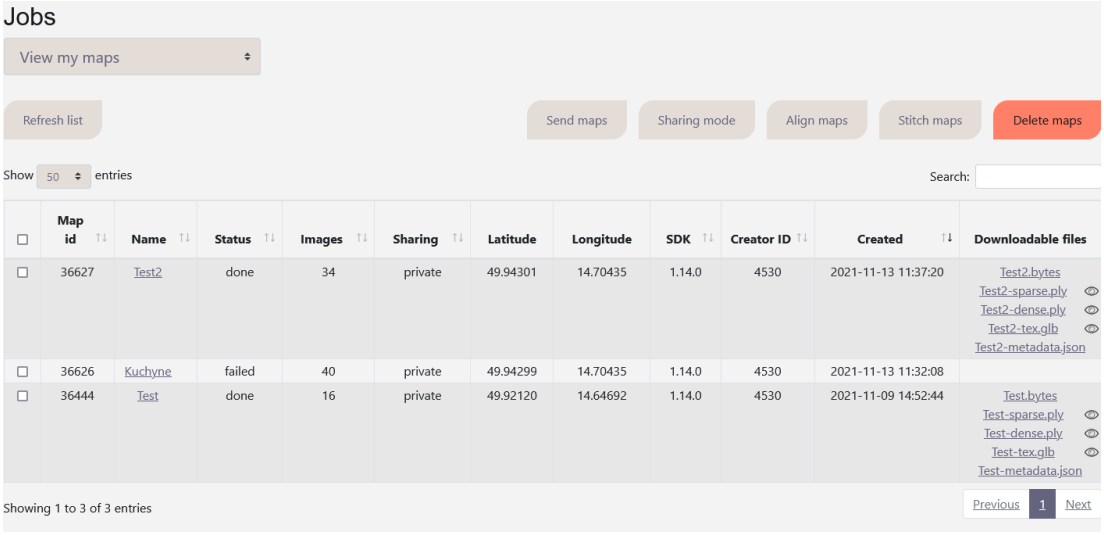
2.1.5 Immersal

Immersal je SDK pro Unity od finské stejnojmenné společnosti, která se soustředí na mapování velkých prostorů a budov [9]. V balíčku je několik demo aplikací, přičemž jednu z nich uživatel potřebuje ke zmapování prostoru. V mapovací aplikaci uživatel skenuje okolí buď pomocí fotek, které nafotí a aplikace spojí všechny dohromady, nebo

videa, kdy chodí kolem objektu/prostoru ke zmapování 30 sekund a aplikace sama skenuje okolí v reálném čase z videa. Zmapovaný prostor se nahraje na vývojářský portál na stránkách Immersal (Obrázek 2.2). Na vývojářském portálu se vytvoří mapa v několika verzích:

- binární soubor (formát .bytes)
- mračno bodů (formát .ply)
- trojúhelníkový mesh (formát .ply)
- texturovaný model (formát .glb)

Všechny soubory se dají stáhnout, trojúhelníkový mesh a texturovaný model Immersal umožňuje zobrazit online pomocí WebXR. (API pro přístup k VR/AR zařízením z internetového prohlížeče)



Obrázek 2.2. Vývojářský portál Immersal.

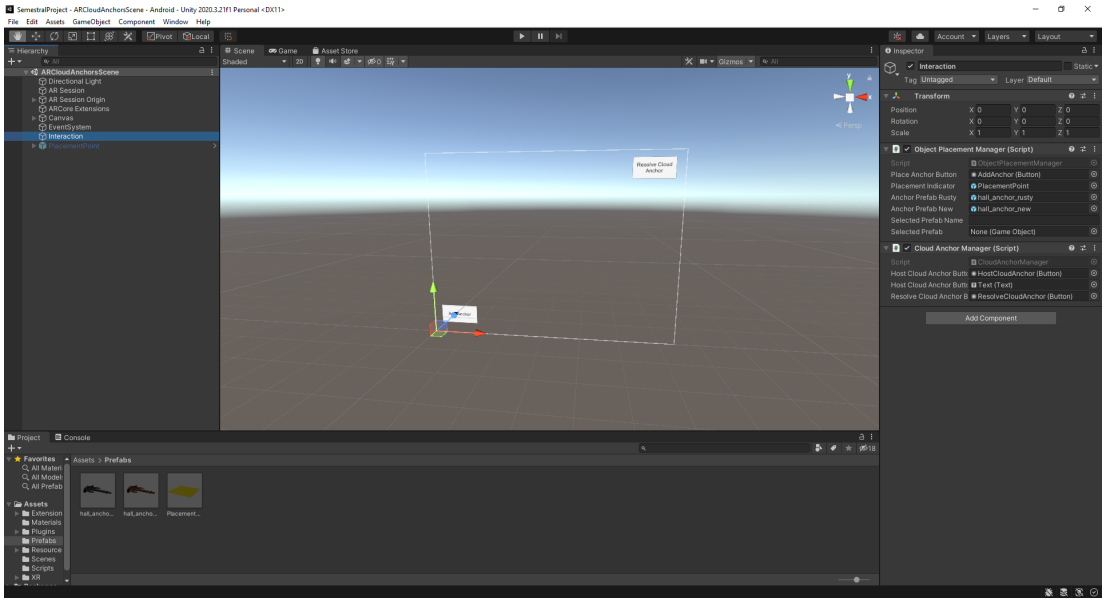
Zmapovaný prostor ve formě mračna bodů lze do Unity dostat pomocí pluginu, který je obsažen také v SDK. Tento plugin umožňuje synchronizaci s vývojářským portálem a nahrání mapy dle výběru. V Unity tak vznikne scéna, do které lze standardním postupem přidávat objekty, které si vývojář napozicuje do reálného světa přesně podle naskenované mapy.

Po otestování dvou dle mě nejlepších možností, Immersal a Google Cloud Anchors, jsem se rozhodl pro využití Cloud Anchors. Hlavní důvod byl nativní propojení s Google Firebase, kterou jsem později využil i pro uchovávání a propagování vlastností objektů náležícím k jednotlivým kotevním bodům, viz 2.3.4. Dále také pro jednodušší používání jednotlivých funkcionalit v kódu.

2.2 Vývojové prostředí

Pro tvorbu interaktivních aplikací na platformě Android se často využívají vybrané herní enginy. Jedná se o komplexní programy umožňující tvorbu interaktivní 3D scény, případně AR scény. Mezi známé herní enginy patří např. CryEngine, Unreal Engine nebo Unity. Je to právě Unity, který tvůrci technologií pro sdílení souřadného systému upřednostňují. Zmíněný herní engine Unity vyvíjí americká společnost Unity Technologies. Poprvé byl vydán v roce 2005 [10]. Společnost okolo enginu vytvořila infrastrukturu podpůrných softwarů a balíčků, které rozšiřují jeho využití nejen ve hrách, ale i

v průmyslu nebo vzdělávání. Unity je uživatelsky přívětivý a po vcelku krátkém čase tvorby se uživatel dobře zorientuje (Obrázek 2.3). Na obrázku vlevo je hierarchie objektů ve scéně, uprostřed pohled do scény, vpravo inspektor vybraného objektu a dole adresář projektu (aktuálně podsložka Assets/Prefabs). Unity má přehledné a obsáhlé API popsané do detailů na internetových stránkách Unity. Na stránkách je také velké množství návodů a instruktážních videí přímo od tvůrců Unity, což ještě více usnadňuje zorientování se při vlastním vývoji nové aplikace.



Obrázek 2.3. Uživatelské rozhraní v Unity.

2.3 Návrh aplikace

Aplikace by dle zadání měla mít několik funkcionalit:

- vytvořit nový kotevní bod na uživatelem zvoleném místě a jeho odeslání na cloud
- rozpoznat kotevní bod při namíření na prostor, kde byl vytvořen, a zobrazit ho
- přidávat, mazat a upravovat objekty patřící k jak nově vytvořeným, tak rozpoznaným kotevním bodům, přičemž všechny úpravy by měly být viditelné v reálném čase u ostatních uživatelů aktuálně pracujících se stejným kotevním bodem

2.3.1 Vytvoření/získání kotevního bodu

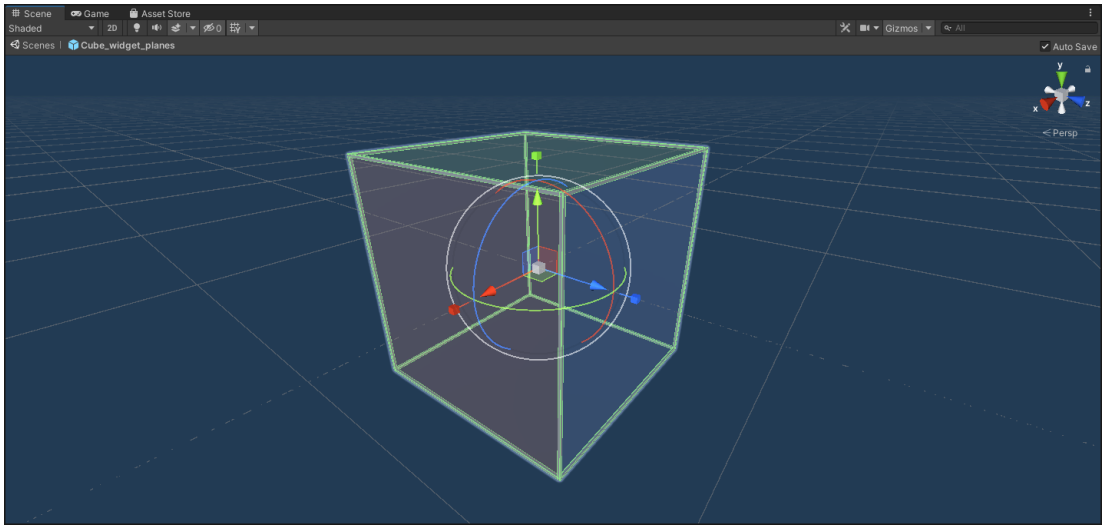
Místo, kde uživatel vytvoří nový kotevní bod, by mělo být nějakým způsobem označené. V mapovací aplikaci od Immersal (Obrázek 2.4) je například zvýraznění vizuálně zajímavých bodů červenými tečkami. V případě Cloud Anchors je vhodné zvýraznit primárně ten bod, který uživatel v prostoru umístí a označí ho jako kotevní. Umístění kotevního bodu je vhodné na rovné ploše (podlaha, stěna apod.). Využiji tedy možnost detekování vertikálních a horizontálních ploch, které nabízí ARFoundation (více v kapitole 3), aby nedocházelo k nevhodnému umístění na nerovné povrchy detekované mobilním zařízením.

Obrázek 2.4. Záběr z mapovací aplikace od Immersal [11]. Při úspěšné lokalizaci aplikace zvýrazní body, kterými se zorientovala, červenými tečkami.

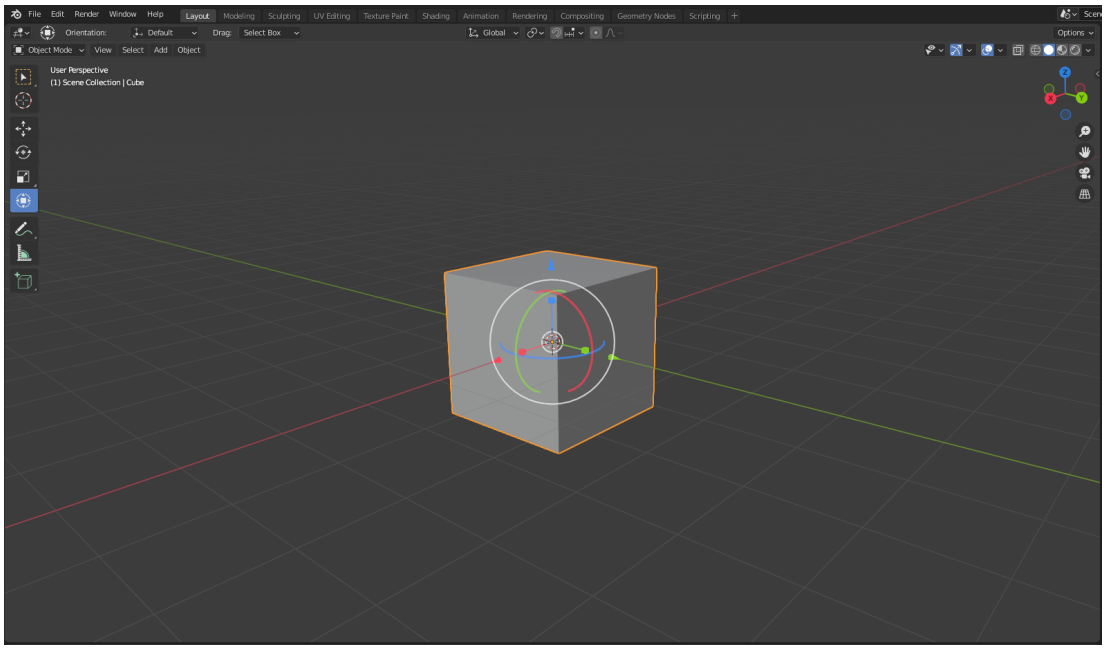
Druhou možností, jak získat kotevní bod, je rozpoznání již existujícího kotevního bodu. V takovém případě bude uživateli nabídnuto, který z uložených bodů chce vyhledat a po výběru bude aplikace informovat uživatele o stavu rozpoznávání. Pokud nedojde k úspěšnému rozpoznání do rozumné doby, aplikace oznámí neúspěšné rozpoznání a vrátí se do výchozího stavu.

2.3.2 Práce s objekty

Při získání kotevního bodu aplikace nabídne uživateli možnost umístit v okolí bodu virtuální objekty, případně je smazat. Po umístění objektu na místo, vyznačené podobně jako při pokládání nového kotevního bodu, se uživateli objeví rozhraní označující aktuálně vybraný objekt. Příklady rozhraní pro manipulaci jsou v herních enginech (Obrázek 2.5) nebo ve 3D modelovacích programech (Obrázek 2.6). Návrh vlastního GUI popisuji v následující sekci.



Obrázek 2.5. Grafické uživatelské rozhraní pro manipulaci s objekty ve 3D prostoru v Unity.



Obrázek 2.6. Grafické uživatelské rozhraní pro manipulaci s objekty ve 3D prostoru v programu Blender.

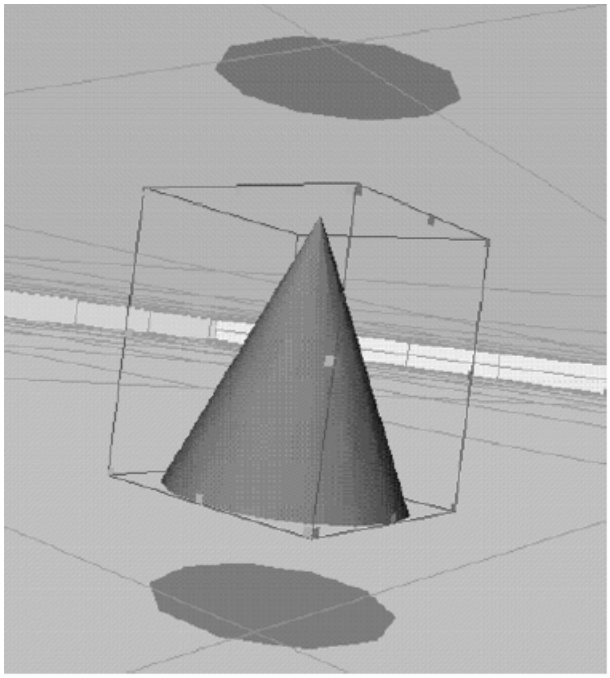
2.3.3 Grafické uživatelské rozhraní

Uživateli by měly být umožněny následující interakce:

- přidání/smazání kotevního bodu
- přidání/smazání objektu
- hostování kotevního bodu
- výběr kotevního bodu, který chce uživatel zobrazit
- výběr 3D modelu, který chce uživatel zobrazit
- úpravu názvu aktuálně zobrazeného kotevního bodu

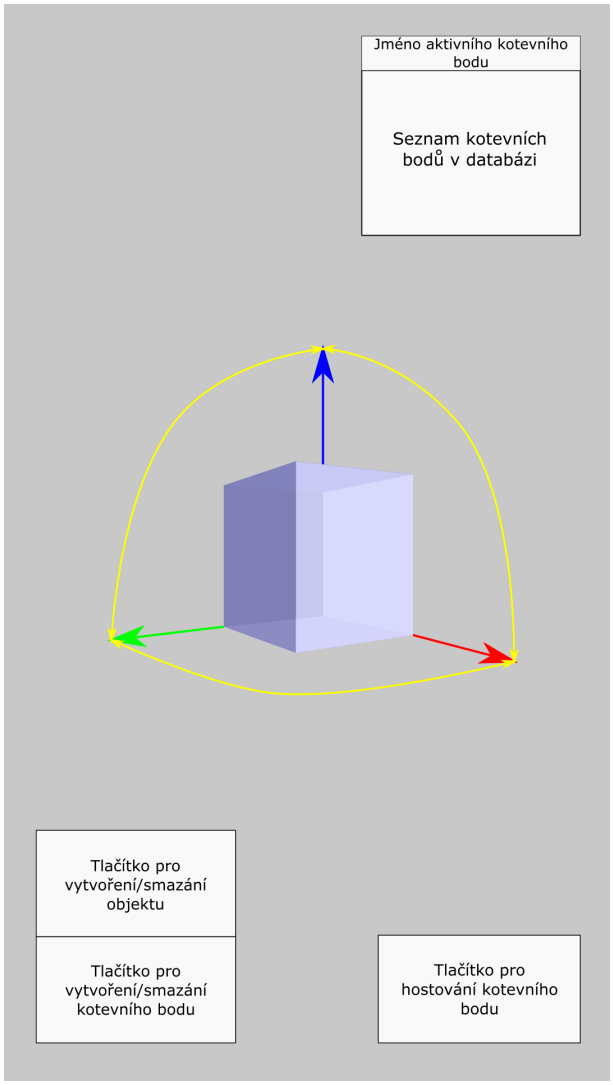
Rozhraní umožní uživateli translaci, rotaci a škálování velikosti objektů. Translace i rotace bude možná ve třech směrech souřadnicových os. Translace bude zároveň ale ro-

tací ovlivněna, tedy směry os budou natočené podle narotování objektu. Škálování bude vzhledem k povaze umisťovaných objektů (Ve výsledku modely exponátů v muzeu.) uniformní, tedy celý objekt se bude do všech směrů zvětšovat/zmenšovat stejně. U mobilních zařízení s dotykovým displejem je škálování nejčastěji prováděno dvěma prsty, kdy při prvním dotyku je zaregistrována vzdálenost mezi prsty jako výchozí a podle následného porovnávání aktuální vzdálenosti je objekt buď zmenšován, nebo zvětšován. Konkrétní vzhled GUI ve své aplikaci a možnosti interakce více popisuji v sekci 3.2. Kolem objektu se zobrazí bounding box. Bounding box ohraničuje objekt do nejmenšího možného prostoru tvaru kvádru zarovnanému dle lokálních souřadnic objektu. Okolo tohoto kvádru budou zobrazeny lokální souřadnicové osy pomocí šipek a také šipky pro rotaci. Tyto tzv. 3D widgety (Obrázek 2.7) jsou umístěny přímo do scény s manipulovanými objekty, díky čemuž je pro uživatele velmi jednoduché zvolit a provést požadované změny.



Obrázek 2.7. Příklad 3D widgetu pro manipulaci s objektem [12].

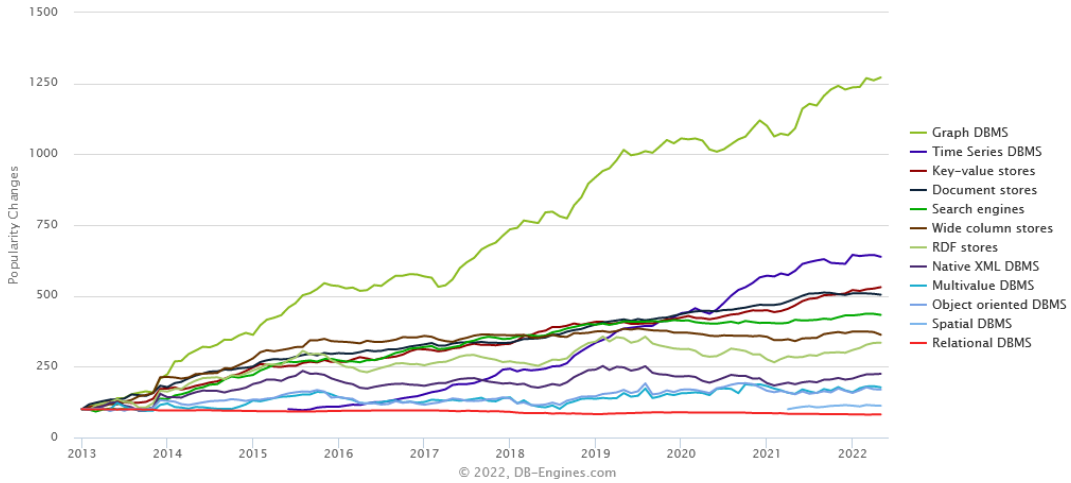
V návrhu uživatelského rozhraní (Obrázek 2.8) jsem využil tři tlačítka. První pro vytvoření/smazání kotevního bodu (vlevo dole), druhé pro vytvoření/smazání objektu (vlevo nad výše zmíněným tlačítkem) a třetí pro hostování kotevního bodu. Dále v horní části displeje je seznam kotevních bodů v databázi. Kolem aktivně zvoleného objektu je rozhraní pro manipulaci s objekty zmíněné v odstavci výše, tzv. gizmo. Skládá se ze tří různě barevných šipek pro translaci, kde každá znázorňuje jednu ze tří lokálních souřadnicových os objektu. Mezi konci translačních šipek jsou šipky pro rotaci kolem jednotlivých os. Rotační šipky jsou zakřivené pro znázornění směru, ve kterém s objektem rotují. Každá rotační šipka rotuje objektem kolem té osy, s jejíž šipkou se v gizmu nedotýká.



Obrázek 2.8. Návrh grafického uživatelského rozhraní v programu Inkscape.

2.3.4 Ukládání dat

Položení nového kotevního bodu a jakékoliv změny jemu náležících objektů budou ukládány v reálném čase na uložiště na internetu. Vzhledem k nativnímu propojení Cloud Anchors s Google Firebase zmíněnému v sekci 2.1 je právě Firebase prakticky ideální volbou. Firebase je široká platforma pro vývoj velkého spektra aplikací. Nabízí plnou správu backendu, databáze, sběr analytických dat, testování a mnoho dalšího. Pro můj případ je užitečná konkrétně Firebase Realtime Database, což je NoSQL databáze, tedy ukládá data jiným způsobem, než tradičními relačními SQL (Structured Query Language - jazyk používaný pro tvorbu tradičních relačních databází) tabulkami. NoSQL databáze získávají čím dál tím větší popularitu (Obrázek 2.9) díky své škálovatelnosti, univerzálnosti a absenci nutnosti perfektně znát SQL. Lze vidět, že tradiční relační databáze (na obrázku červená) si drží stabilně podobnou popularitu, zatímco mnoho druhům NoSQL databází roste popularita velmi rychle. Realtime databáze ukládá data ve formě JSON (Java Script Object Notation - strukturovaný způsob zápisu dat). Umožňuje také synchronizaci se všemi připojenými klienty v reálném čase.



Obrázek 2.9. Růst popularity databází podle druhu [13] od ledna 2013 do května 2022. Metoda výpočtu popularity: `https://db-engines.com/en/ranking_definition`

Aplikace potřebuje ukládat dva typy dat. První jsou 3D modely, které uživatel umisťuje do scény. Unity umožňuje mít ve scéně předpřipravené varianty třídy GameObject (Základní reprezentace čehokoliv, co může v Unity scéně existovat.), tzv. prefaby, tedy lze mít předpřipravené jednotlivé modely se správnou velikostí, orientací atd. Problém bývá, že musí být vše připraveno předem a nelze za běhu aplikace přidávat další nové prefaby. Dalším problémem je velikost těchto dat, protože uživatel nejčastěji nepoužije úplně všechny dostupné objekty. Nemá tedy smysl, aby měl všechny objekty uložené na svém zařízení. Z toho lze usoudit, že lepší variantou je mít data uložená na serveru. Nejjednodušší a v tomto případě nejvhodnější možností je FTP (File Transfer Protocol - standardní protokol pro přenos souborů) server, kde budou jednotlivé 3D objekty uložené a v případě, že uživatel bude chtít jeden z těchto objektů přidat do scény, aplikace si model stáhne a vytvoří ve scéně nový GameObject právě s tímto 3D modelem. Nevýhodou této varianty je velká rozmanitost formátů 3D modelů a nutnost mít v aplikaci také importér.

Aplikace bude tedy omezená na 3D modely formátu glTF (Graphics Language Transmission Format) a jeho binární verzi glb. glTF je mladý formát vyvinutý skupinou Khronos, která stojí např. za známými grafickými knihovnami OpenGL nebo Vulkan. Je označován jako „JPEG pro 3D“ [14]. Pro importování glTF modelů do Unity za chodu aplikace existuje několik importérů. Zvolil jsem jeden z vedoucím doporučených importérů GLTFUtility od Thora Brigsteda [15]. Umožňuje synchronní a asynchronní import modelu, který navrátí jako instancovaný GameObject. Postup bude tedy takový, že 3D model bude uložený na FTP serveru. Uživatel si zvolí přidat do scény konkrétní 3D model. Aplikace tento model stáhne a s pomocí GLTFUtility asynchronně naimportuje do scény.

Hostingů FTP serveru je k dispozici mnoho i zdarma. Zvolil jsem český hosting Endora, který umožňuje zdarma vytvořit FTP server a ukládat na něj až 2 GB dat [16].

Druhý typ dat jsou poziční a dodatečné informace ke kotevním bodům a jim náležícím objektům. Pro popis všech potřebných dat postupuji hierarchií objektů ve scéně zeshora. Světová pozice a rotace kotevního bodu (na vrcholu hierarchie) je zajištěna pomocí Cloud Anchors. Každý kotevní bod má také svůj unikátní identifikátor (Využiji GUID - Globally Unique Identifier, což je 128-bitový integer určený právě k tvorbě unikátních identifikátorů.), Cloud Anchor identifikátor (Vytvořený při hostování ko-

tevního bodu. Více v sekci 3.3) a jméno. U každého objektu patřící pod kotevní bod si aplikace pamatuje jeho pozici, rotaci a velikost v lokálních souřadnicích, název 3D modelu a identifikátor. Zde je vhodné využít funkci Unity GetInstanceID, která navrátí unikátní identifikátor v rámci scény. Ten mi stačí uložit jednou a pak se na něj už pouze odkazovat.

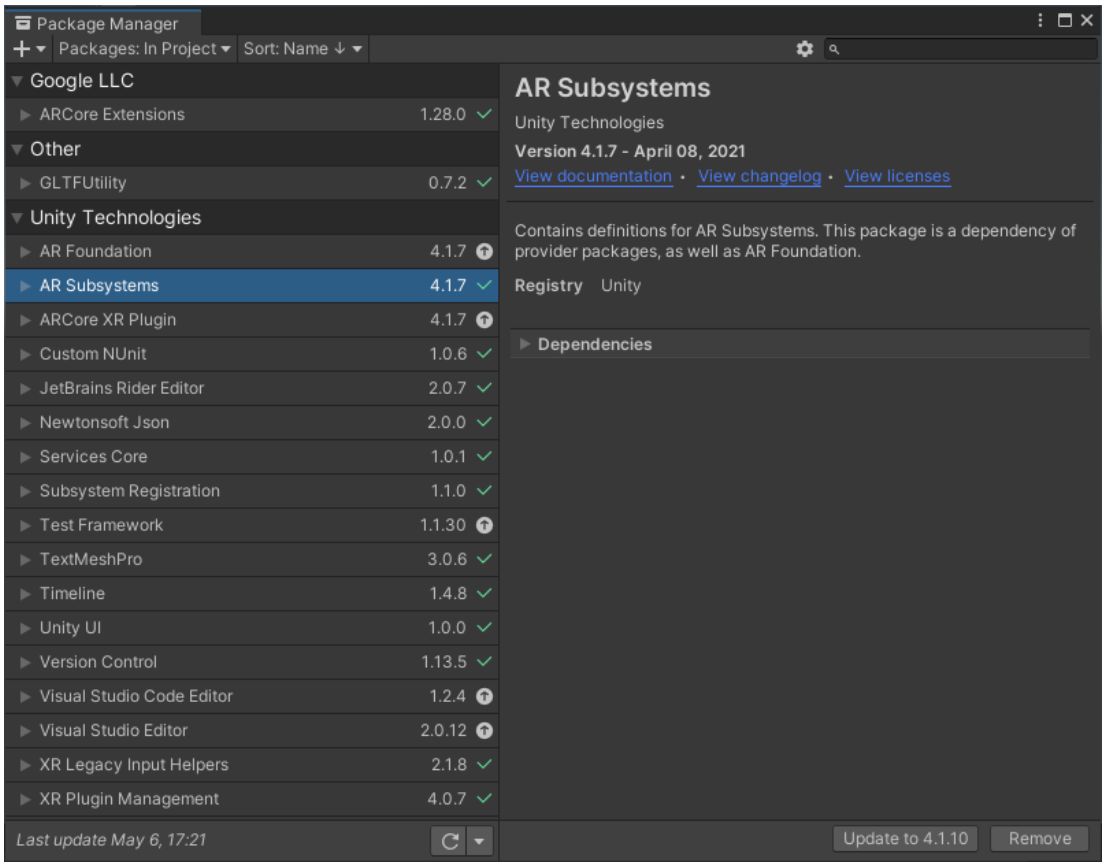
Všechna tato data jsou složená pouze z čísel a textu. Není tedy problém je poskládat do JSON formátu a odeslat/přijmout na/z Realtime Database. Pro převod ze serializovatelných objektů do JSON a opačně slouží Unity třída JsonUtility, která umí právě takové převody. Konkrétní strukturu dat a jejich převod uvádím v sekci 3.6.

Kapitola 3
Implementace

Základní funkcionalita Cloud Anchors a umisťování kotevních bodů do prostoru byly k dispozici již v mém semestrálního projektu. Využil jsem ho tedy jako základ pro bakalářskou práci. Z původního ukládání informací o kotevním bodu na FTP server jsem přešel na Firebase Realtime databázi (sekce 3.6.) Místo jednoho 3D modelu byla přidána možnost výběru z různých modelů a lze jednoduše přidat další. Semestrální projekt neumožňoval jakkoliv manipulovat s vytvořeným objektem, což nyní lze.

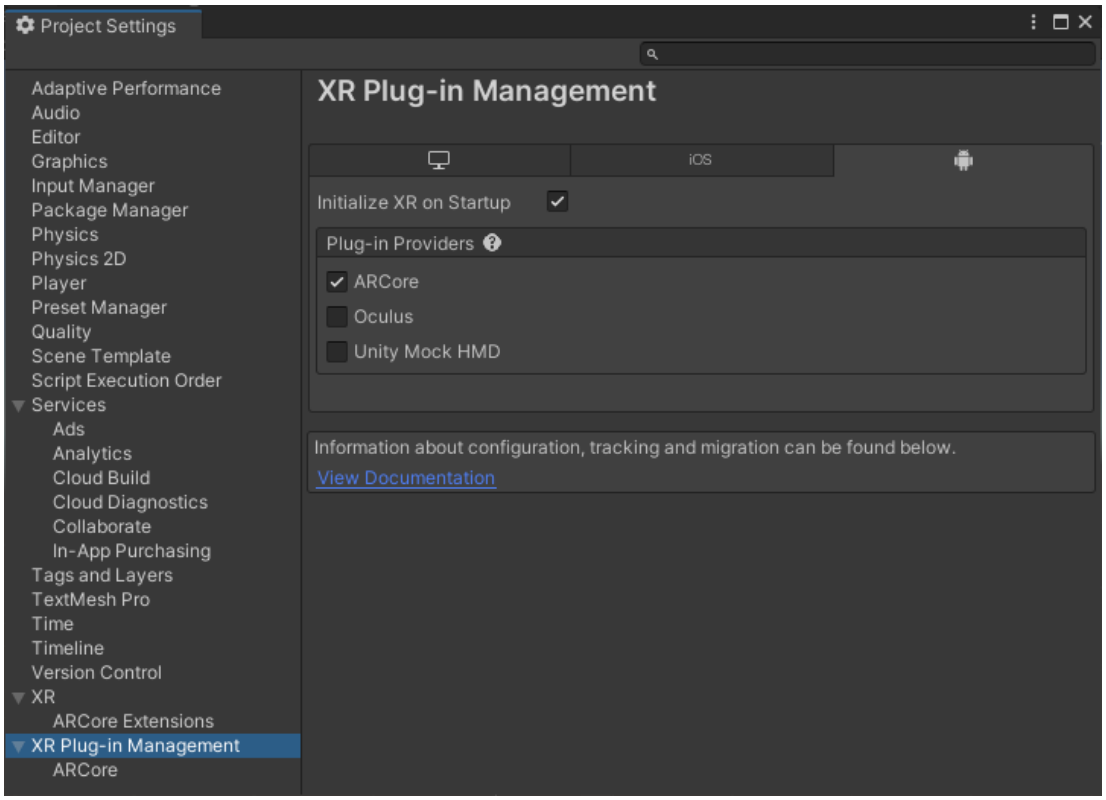
3.1 Nastavení a struktura projektu

Před samotným programováním a tvořením scény je potřeba správně nakonfigurovat nastavení projektu v Unity a stáhnout všechny potřebné balíčky a rozšíření (Obrázek 3.1). Patří mezi ně ARCore XR Plugin pro funkce ARCore, XR Plugin Management pro správné provázání platforem, AR Foundation společně s AR Subsystems a jejich nadstavbu ARCore Extensions, ve které jsou obsaženy i Cloud Anchors. V neposlední řadě pak GLTFUtility, což je importér .gltf/.glb 3D modelů v reálném čase.



Obrázek 3.1. Balíčky obsažené v projektu aplikace.

Po importování balíčků je potřeba provázat ARCore s Androidem. Je nutné, aby Android zařízení věděly, že mají použít ARCore (Obrázek 3.2). Aplikace vyvíjená v rámci bakalářské práce řeší pouze operační systém Android. Dále je v nastavení potřeba vypnout výchozí podporu Vulkanu (multiplatformní API pro 3D grafiku), protože ARCore ho zatím nepodporuje. Poté nastavit minimální požadovanou verzi Androidu na 7.0 'Nougat' a povolit podporu ARM64 architektury.



Obrázek 3.2. Nastavení pro Android, které zajistí používání ARCore při zapnutí aplikace.

V neposlední řadě je potřeba vytvořit nový Google Cloud projekt, zaktivovat ARCore Cloud Anchor API a propojení s Firebase.

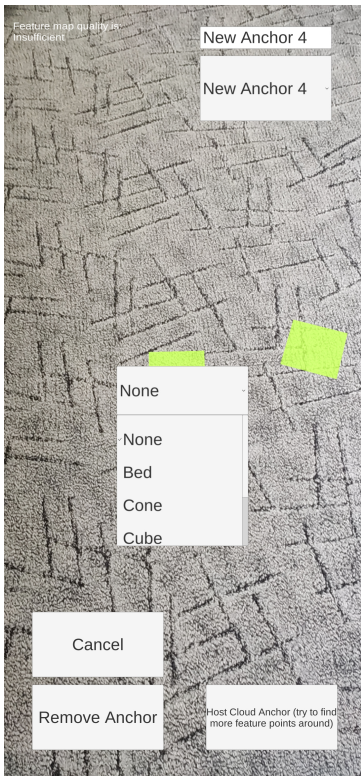
Aby se zařízení při spuštění aplikace lokalizovalo, je potřeba mít ve scéně předpřipravený objekt AR Session, který jednak funguje jako konfigurátor, jednak jako vstupní bod pro ARCore API. Vedle toho je potřeba mít AR Session Origin objekt, který obsahuje všechny ARCore skripty spravující různé AR prvky, které v aplikaci využívám. Zároveň spravuje kameru a všechny objekty vytvořené z detekovaných bodů, jako jsou mračna bodů nebo plochy. Pro potřeby této aplikace obsahuje AR Session Origin AR Raycast Manager, který spravuje vysílání paprsků do AR scény, AR Plane Manager pro detekci ploch a AR Anchor Manager pro správu kotevních bodů. V neposlední řadě se ve scéně nachází objekt ARCore Extensions, který řeší veškerou práci s Cloud Anchors.

Kód aplikace se skládá ze čtyř C# skriptů. Prvním je ObjectPlacementManager, který se stará o rozmístění ve scéně, pokládání kotevních bodů, objektů, změnu jejich pozice, rotace, velikosti atd. Druhý je CloudAnchorManager, ve kterém se provádí veškerá práce s Cloud Anchors - hostování, rozpoznávání a mapování prostoru kolem kotevního bodu. Třetí je FirebaseManager, který komunikuje s Realtime databází a synchronizuje uživateli veškerá data kotevních bodů a jejich objektů. Čtvrtý je DebugConsole od Ing. Tomáše Havlíka pro výpis debugové konzole při běhu aplikace [17]. Do debugové konzole jsou uživateli posílané všechny informace o chodu aplikace.

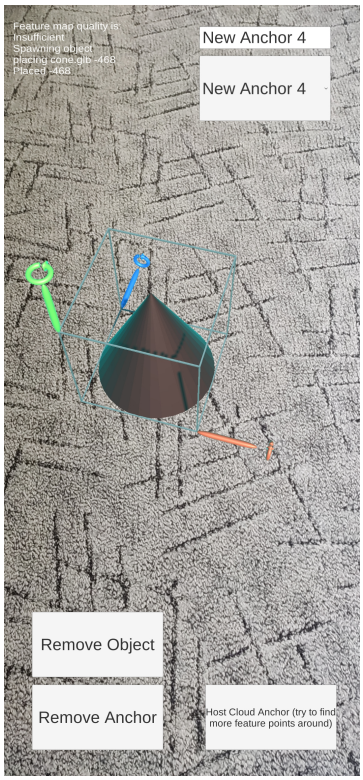
3.2 Implementace grafického uživatelského rozhraní

Při tvorbě uživatelského rozhraní byly využívány primárně tlačítka, případně rozbalovací seznamy např. pro volbu objektu, který chce uživatel umístit. Design tlačítek samotných není zaměřením této aplikace, která má sloužit hlavně jako otestování použitelnosti Cloud Anchors. Využil jsem tedy pouze výchozí vzhled tlačítek. Samozřejmostí je úprava velikosti tak, aby text na tlačítkách byl čitelný a tlačítka jednoduše stisknutelná i palcem. Veškerý text zobrazovaný uživateli jsem pro univerzálnost psal v angličtině. Realizované GUI se liší prakticky pouze v komonentě pro rotaci s aktivním objektem, kterou v rámci popisu celého rozhraní pro manipulaci s objekty (Obrázek 3.5) popisuji v sekci 3.5.

Obrázek 3.3. Vzhled GUI ihned po spuštění aplikace. V případě, že by již existoval nějaký kotevní bod v databázi, zobrazil by se v seznamu vpravo nahoře.



Obrázek 3.4. Po položení kotvy a požadavku přidání objektu, který chce uživatel umístit, se objeví seznam pro výběr.



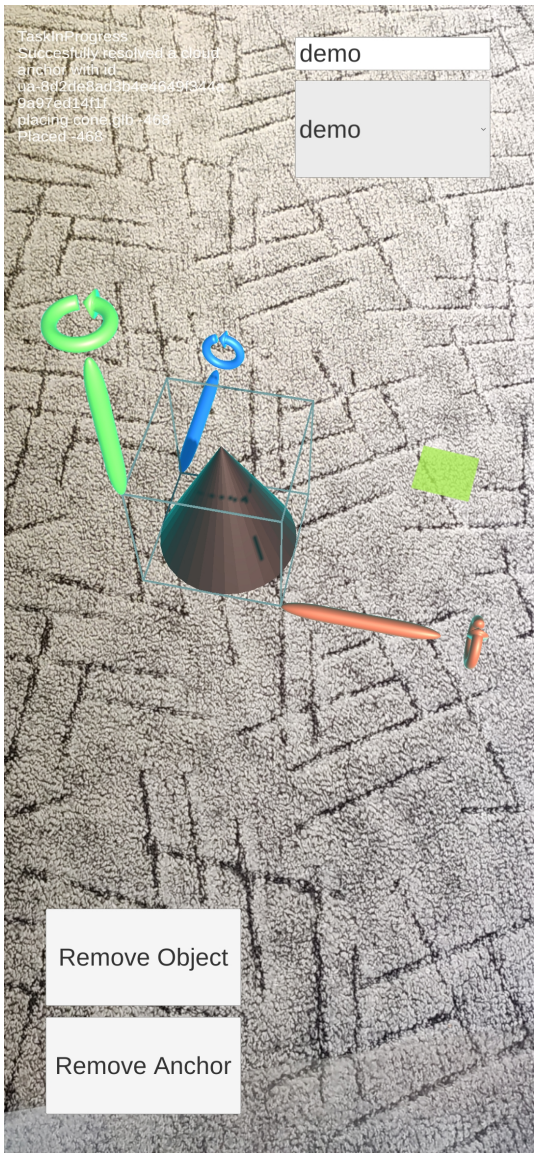
Obrázek 3.5. Po výběru objektu se ve vlákně na pozadí soubor stáhne, naimportuje a umístí do scény. Kolem vytvořeného objektu se zobrazí rozhraní pro manipulaci.

3.3 Tvorba nového kotevního bodu

Po otevření aplikace má uživatel dvě možnosti. První je vytvořit nový kotevní bod. Aplikace potřebuje chvíli času a pohyb zařízením pro zorientování se v prostoru a detekci ploch v okolí. Jakmile najde plochu (vertikální nebo horizontální) a zařízení bude namířeno tak, aby uprostřed obrazovky byla i daná plocha, zobrazí se na ni žlutý poloprůhledný čtverec. Od spuštění uživatel vidí jediné tlačítko, a to pro položení kotevního bodu (Obrázek 3.3). Po stisknutí tlačítka se na místě, kam zařízení aktuálně míří, vytvoří další stejný žlutý čtverec značící kotevní bod. S tím se zároveň odešlou aktuálně

dostupná data na Realtime databázi (Detailně popsáno v sekci 3.6.) Zároveň se zobrazí tlačítko pro umístění objektu do scény, jehož funkci popisuji v sekci 3.5. S položením kotevního bodu se začne periodicky mapovat okolí, které si kotevní bod ukládá do tzv. feature mapy. Cloud Anchors hodnotí kvalitu mapování okolí třemi konstantami zmíněnými v sekci 2.1.4. Pokud si uživatel rozmyslí umístění kotevního bodu, stejným tlačítkem, kterým bod vytvořil, ho může smazat.

Obrázek 3.6. Stav aplikace po úspěšném hostování: Vlevo nahoře je vidět výpis, co aplikace dělala.



Obrázek 3.7. Stav aplikace po úspěšném rozpoznání kotevního bodu.

Během mapování by měl uživatel mířit zařízením různě po nejbližším okolí kotevního bodu, aby kvalita mapování byla co největší a kotevní bod tak šel dobře rozpoznat. Uživatel se může kdykoliv rozhodnout začít kotevní bod hostovat. Hostování znamená, že se data z mapy okolí pokusí nahrát do interní Firebase databáze (ne Realtime používaná v této aplikaci). Neúspěch nastane tehdy, pokud je dat velmi málo, což vypíše i stejně znějící chybovou hlášku. Uživatel bude muset namapovat větší část okolí a pokusit se o hostování znovu. V případě úspěchu je uživateli vypsaná informace o úspěšném

hostování (Obrázek 3.6). Po hostování může uživatel dále s kotevním bodem pracovat dle libosti.

3.4 Rozpoznávání existujícího kotevního bodu

Druhou možností, kterou má uživatel po spuštění aplikace je výběr již existujícího bodu, který chce rozpoznat. V pravém horním rohu je rozklikávací seznam, který se při spuštění aplikace naplní názvy všech kotevních bodů již nahraných do Relatime databáze. Při výběru jednoho z nich si aplikace ověří, že daný kotevní bod již byl úspěšně hostován, tedy že v databázi má svůj Cloud Anchor identifikátor. Pokud identifikátor nemá, aplikace uživateli oznámí, že vybraný kotevní bod ještě nebyl hostován. V opačném případě aplikace vyšle požadavek na rozpoznání vybraného kotevního bodu, začne mapovat okolí a snaží se nalezené feature body napozicovat na mapu uloženou pro vybraný kotevní bod. V případě nenalezení shody může uživatel rozpoznávání zrušit a zopakovat. Pokud daný bod nalezne, zobrazí kotevní bod a všechny jemu náležící objekty tak, jak jsou uložené v databázi (Obrázek 3.7).

Podle kvality mapování může dojít k nepřesnému umístění kotevního bodu, tím pádem i veškerých objektů. V tom případě může uživatel zkusit rozpoznat kotevní bod znovu tím, že v seznamu kotevních bodů zvolí možnost None, která aktuálně rozpoznaný kotevní bod vymaže ze scény, a poté opět vybere stejný kotevní bod k rozpoznání.

3.5 Tvorba, správa a mazání objektů

Po vytvoření/rozpoznání kotevního bodu může uživatel stejným způsobem umístit jakýkoliv z objektů uložených na FTP serveru. Pro komunikaci se serverem je ve skriptu ObjectPlacementManager vytvořenou instanci třídy WebClient, která umožňuje odesílání a přijímání dat na zadanou adresu, v mém případě FTP server. Po namíření na místo, kde uživatel chce vytvořit nový objekt, stiskne tlačítko pro přidání objektu a otevře se mu seznam, kde vybere požadovaný model (Obrázek 3.4). Po výběru objektu se zavolá funkce WebClienta DownloadFileAsync, která na vedlejším vlákně stáhne soubor se 3D modelem a uloží ho do zařízení. Cesta, kam se model uloží, je definována pomocí statické proměnné Untity Application.persistentDataPath, která umožňuje napříč platformami persistentní cestu, kam je vhodné podobná data ukládat. V případě platformy Android je persistentní cesta /storage/emulated/0/Android/data/„název balíčku aplikace“/files [18]. Dokončení stažení souboru je klasifikováno jako tzv. event neboli událost. Události umožňují třídě nebo objektu upozornit další třídy nebo objekty, když k nějaké dojde [19]. K dané události můžeme připojit tzv. posluchače (event listener), tedy funkci, která se zavolá, když daná událost nastane. Další příklad využití událostí v této práci jsou prakticky veškeré interakce s uživatelským rozhraním, tedy stisk tlačítek, výběr v seznamech apod.

3.5.1 Vytvoření objektu ve scéně

K dokončení stažení je přidána funkce OnFinishDownloadAsync ve skriptu ObjectPlacementManager jako posluchač. Ta v případě úspěšného stažení zavolá glTF importér, konkrétně jeho funkci ImportGLTFAsync/ImportGLBAsync (podle formátu staženého 3D modelu), ke které je připojena další funkce jako posluchač OnFinishImportAsync. Ta po úspěšném asynchronním naimportování 3D modelu a vytvoření instance třídy GameObject nastaví správné parametry nově vytvořeného objektu, tedy pozici, rotaci, jméno, tag, přiřadí mu jako rodiče v hierarchii scény kotevní bod atd. Velmi podobně

aplikace funguje i při přidávání objektů z existujícího kotevního bodu do scény. Pro každý na databázi nalezený objekt rozpoznaného kotevního bodu (proces popsaný v předešlé sekci) aplikace daný model stáhne, nahraje do scény a nastaví všechny jeho parametry.

3.5.2 Manipulace s objekty

Po umístění objektu do scény se nastaví jako aktivní a aplikace kolem něj zobrazí rozhraní pro manipulaci (Obrázek 3.5 a 3.7.) Rozhraní je naorientováno podle lokální rotace objektu a zvýrazňuje lokální souřadnicové osy (X červená, Y zelená a Z modrá) šipkami pro translaci a stočenými šipkami pro rotaci. Šipky pro translaci jsou umístěny tak, aby vycházely z bounding boxu, který byl popsán v sekci 2.3.2. Translaci objektu ve směru jedné z lokálních souřadnicových os provede uživatel tak, že prstem uchopí jednu z dlouhých šipek pro translaci a začne pohybovat po nebo proti směru šipky. Tím se objekt posouvá po dané lokální souřadnicové ose. Stejně tak může uživatel s objektem rotovat podle jedné z os kulatými šipkami na koncích šipek pro translaci. Uživatel uchopí stejně jako v případě translace šipku té osy, ve které chce rotovat, a začne prstem hýbat nahoru a dolů, případně doleva a doprava, podle potřeby rotace objektu. Veškeré rotace probíhají stejně jako translace v lokálním souřadném systému. Třetí možností manipulace s objekty je změna velikosti. Uživatel položí dva prsty kamkoliv na displej a poté s nimi hýbe od sebe nebo k sobě, čímž zvětšuje/zmenšuje aktivní objekt. Změna velikosti je uniformní, tedy ve všech osách stejná.

3.5.3 Mazání kotevních bodů a objektů

Pokud chce uživatel některý z objektů ve scéně smazat, stačí na něj klepnout prstem a stisknout tlačítko Remove Object. To smaže objekt ze scény a následně i záznam v databázi. Stejně tak jdou smazat i kotevní body, kdy při stisknutí tlačítka Remove Anchor aplikace smaže kotevní bod a všechny jemu náležící objekty.

3.6 Ukládání a vyhledávání dat

Veškerá komunikace s Realtime databází a vše, co se týká dat, je ve skriptu FirebaseManager. Vytvořil jsem dvě serializovatelné třídy a jejich JSON varianty plus JSON variantu pole kotevních bodů pro získání seznamu všech kotevních bodů v databázi. První třídou je CloudAnchor, respektive CloudAnchorJSON. Obsahují vlastní identifikátor, Cloud Anchor identifikátor a jméno. CloudAnchor k tomu obsahuje přímo GameObject kotevního bodu, pod nímž jsou i všechny umístěné objekty. CloudAnchorJSON namísto GameObjectu obsahuje seznam objektů, respektive informací o objektech ve formě třídy ObjectJSON. Třída ChildObject a ObjectJSON reprezentují jednotlivé objekty pod kotevními body. Obsahují název 3D modelu a unikátní identifikátor, dále v případě ChildObject obsahuje GameObject samotného objektu (Ze kterého lze získat veškeré potřebné informace.) a u ObjectJSON pozici, rotaci a velikost, vše v lokálním souřadném systému.

Přístup k Realtime databázi funguje jako průchod stromovým grafem. V kódu je uchována reference do kořene databáze funkcí DefaultInstance.RootReference. Z reference kořene se lze přes funkci Child (”název uzlu”), která navrátí referenci daného podřadného uzlu, dostat kamkoliv v databázi.

Při vytvoření kotevního bodu je vytvořena nejdříve instance CloudAnchor, z ní posléze CloudAnchorJSON se jménem a unikátním identifikátorem. Instanci třídy

CloudAnchorJSON přetvořím do proměnné typu string s pomocí třídy Unity JsonUtility, která umí konverzi z a do JSON formátu. Vytvořená proměnná je odeslána jako nový uzel pod uzlem cloud_anchors pod kořenem (viz v kódu níže). Realtime databáze umožňuje veškerou komunikaci s databází asynchronně a umožňuje posílat buď konkrétní serializovatelné typy hodnot (Které v této aplikaci nejsou vhodné, protože neumí poslat trojsložkový vektor, kvaternion nebo vlastní třídu objektů.) nebo právě string ve formátu JSON. Celková struktura dat tedy vypadá takto:

```
unique-terminus-331513-default-rtdb.europe-west1.firebasedatabase.app
    cloud_anchors
       2f8f13aa-5a7d-4dd7-bf29-2c8a6b38221c
          cloudAnchorId: "ua-8d2de8ad3b4e4649f344a9a97ed14f1f"
          id: "2f8f13aa-5a7d-4dd7-bf29-2c8a6b38221c"
          name: "demo"
          objects
              -468
                  fileName: "cone.glb"
                  id: "-468"
                  position
                      x: -0.6328884363174438
                      y: -0.9196717739105225
                      z: 0.4932241141796112
                  rotation
                      w: 0.9988450407981873
                      x: 0
                      y: -0.048050493001937866
                      z: 0
                  scale: 0.2240743339061737
```

Při přidání nového objektu ke kotevnímu bodu podobně jako u samotného kotevního bodu vytvořím ChildObject, ve kterém je lokální pozice, rotace a velikost, jméno a identifikátor. Z instance ChildObject vytvořím ObjectJSON, který opět přetvořím do proměnné typu string a odešlu jako nový uzel pod daný kotevní bod (potomek uzlu `objects` v kódu výše).

Kapitola 4
Testování

V této kapitole zhodnocuji dva různé aspekty. Prvním je funkčnost samotných Cloud Anchors, tedy jak se s nimi pracuje, jak dobře funguje mapování a rozpoznávání kotevních bodů. Druhým je uživatelská přívětivost aplikace, kterou hodnotí výhradně vybraní uživatelé.

4.1 Testování persistence scény

Persistencí scény se myslí, jak přesně byly kotevní body rozpoznány a jak moc odpovídají pozici a rotaci při vytváření. Persistence může být ovlivněna hned několika faktory. Prvním je kvalita zmapování při vytváření kotevního bodu, jejíž tři možné výstupy byly zmíněny v sekci 2.1.4. Druhým faktorem jsou světelné podmínky, které mohou velmi ovlivnit schopnosti Cloud Anchors. Pro účely testování zobecním stav světelných podmínek na nízký (šero/tma), normální (denní světlo) a vysoký (přímý sluneční svit, zdroj ostrého světla). Třetím faktorem, který může ovlivnit fungování Cloud Anchors, je změna prostředí, které bylo namapováno. V případě interiéru může dojít např. k pohybu s nábytkem, v případě muzea mohou být ve zmapovaném prostředí návštěvníci, kteří budou blokovat dříve zmapované body v okolí. Jsou tedy stanoveny tři proměnné, které lze měřit. Dále jsou stanovena dvě prostředí, ve kterých budu tři stanovené proměnné měřit. První je v prostorách Národního muzea, kde byla možnost jedno odpoledne měřit a testovat. Druhým prostředím je domácí interiér, ve kterém testovala většina uživatelů.

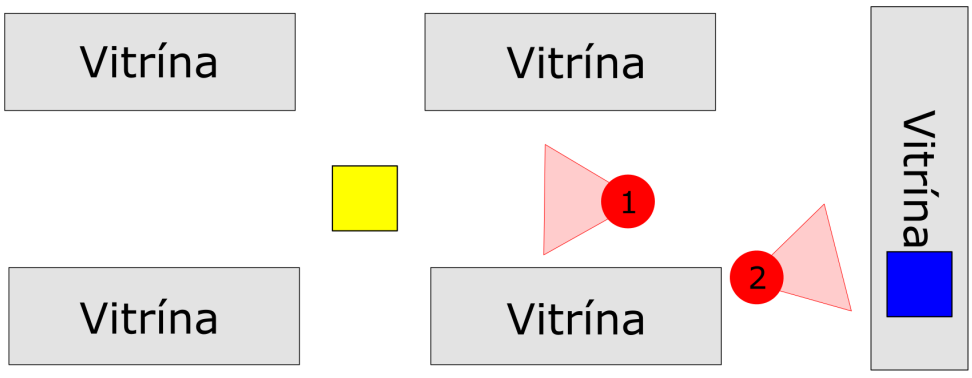
4.1.1 Testování v muzeu

Od Národního muzea byly dodány návrhy scénářů, ve kterých by muzeum chtělo využít AR se SLAM technologií. 9. března 2022 jsme společně s kolegou Bc. Dominikem Truongem řešícím podobnou problematiku měli možnost projít si aktuální expozice a prostory v Národním muzeu a testovat tak přímo v prostorách některých z navrhnutých scénářů. Během testování v muzeu aplikace nebyla ve finální podobě, uživatelské rozhraní se tedy liší a pozice kotevního bodu je znázorněna modelem modro-zelené kotvy. U tří scénářů jsou plánky místností, ve kterých byly snímky pořízené (Obrázky 4.2, 4.5 a 4.13). Červené kruhy s úhlem připomínající záběr kamery znázorňují pozici a směr snímku. Žluté čtverce znázorňují místo kotevního bodu při hostování a modré čtverce znázorňují místo kotevního bodu při rozpoznání.

Prvním scénářem je model krakatice obrovské v rámci expozice Zázraky evoluce (Obrázek 4.1). Model měří na délku 17 metrů a nachází se zavěšený nad návštěvníky ve slabě osvětlené místnosti s dvěma řadami vitrín a jednou vitrínou přes celou šířku místnosti na jednom z konců. Světlené podmínky ve scénáři jsou nízké. Změna prostředí byla větší, protože se v místnosti pohybovali návštěvníci muzea.

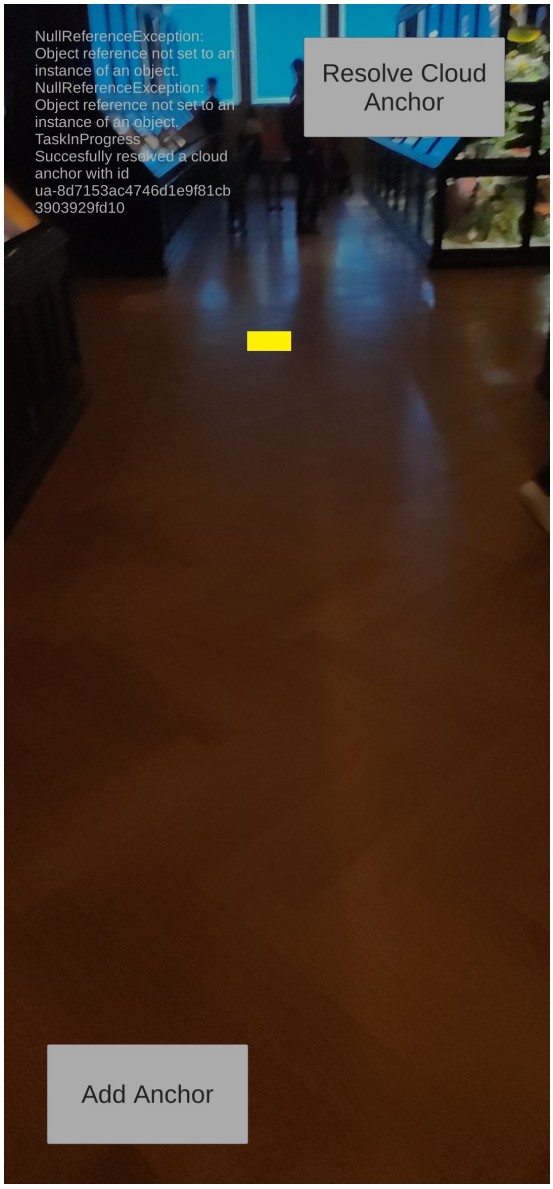
Obrázek 4.1. Referenční snímek modelu krakatice. Zdroj: `https://www.nm.cz/prirodovedecke-muzeum/zazraky-evoluce`

Na referenčním obrázku jsou rozsvícená velká stropní světla, která dobře osvětlují celou místnost. Během testování při standardním provozu muzea jsou tato světla zhasnutá a v místnosti jsou tím pádem výrazně horší světelné podmínky. Pro návštěvníky muzea jsou takové podmínky velmi příjemné a lidskému oku nedělá problém zaostřit na jemně nasvícené exponáty a popisky za vitrínami. Pro kameru mobilního zařízení to znamenalo výrazné ztížení mapování, protože bylo v místnosti mnohem méně kontrastu.



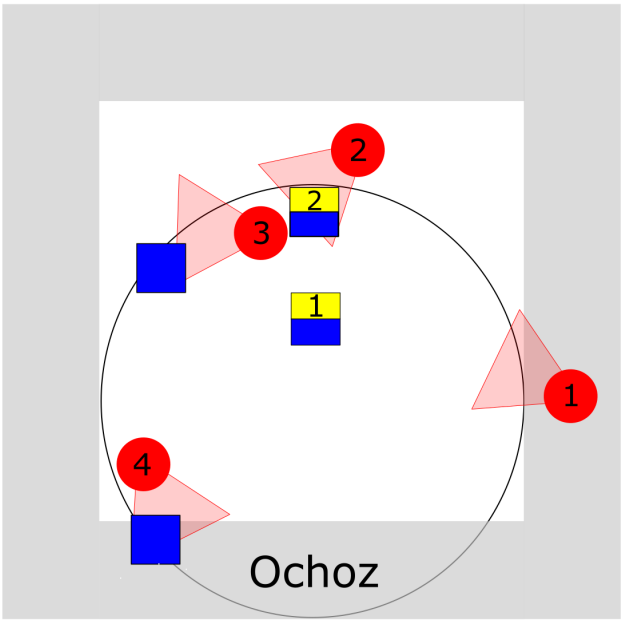
Obrázek 4.2. Plánek místnosti s krakaticí.

Kotevní bod byl vytvořen uprostřed místnosti mezi vitrínami (Obrázek 4.3, na plánku 4.2 žlutý čtverec). Kvalita mapování byla pouze nedostatečná. Po úspěšném a velmi dlouhém prvním rozpoznání byl kotevní bod posunut o několik metrů, rotaci si však zachoval podobnou originálu. Druhý pokus o rozpoznání zobrazil kotevní bod ve vitríně, ve které byly modely velkých vyšších korýšů (Obrázek 4.4). Právě tyto modely mohly svou členitostí a barevností dobře posloužit jako feature body. Rozpoznání však bylo od místa hostování několik metrů posunuté jak horizontálně, tak vertikálně a rotace byla také velmi odlišná (na plánku 4.2 modrý čtverec).



Obrázek 4.3. Místo hostování kotevního bodu v místnosti s krakaticí. Snímek odpovídá na plánku 4.2 číslu 1.

Obrázek 4.4. Místo rozpoznání kotevního bodu v místnosti s krakaticí. Na plánku 4.2 číslo 2.



Obrázek 4.5. Plánek pantheónu.

Druhým scénářem byl pantheón. Čtvercová místnost s ochozem v patře. Obsahovala převážně lesklé povrchy - mramor nebo kov. Na stropě a vrchních částech stěn jsou nástěnné malby, které mohou posloužit jako feature body. Zároveň jsou ale stropy vysoké a malby tím pádem daleko od zařízení. Testování začalo na horním ochozu. Kotevní bod byl položen doprostřed místnosti výškově v úrovni ochozu a namapovány byly prostory ochozu kolem jedné celé stěny. Kvalita mapování byla nedostatečná. Rozpoznání (Obrázek 4.7, na plánku modro-žlutý čtverec s číslem 1 a červený kruh s číslem 1) bylo oproti místnosti s krakaticí výrazně úspěšnější. Bod byl rozpoznán rychle, pozice s odchylkou maximálně desítky centimetrů, rotace přibližně o 45 stupňů jinak.

Při testování ve spodní části pantheónu byl hlavní problém s odrazem světel od všudypřítomného mramoru. Dalším problémem byl opakující se motiv na podlaze. Ze tří pokusů rozpoznání byl dobrý pouze jeden (Obrázek 4.6, na plánku modro-žlutý čtverec s číslem 2 a červený kruh s číslem 2). Zbylé dva byly hodně posunuté a špatně zrotované (Obrázky 4.8, na plánku číslo 3 a 4.9, na plánku číslo 4).

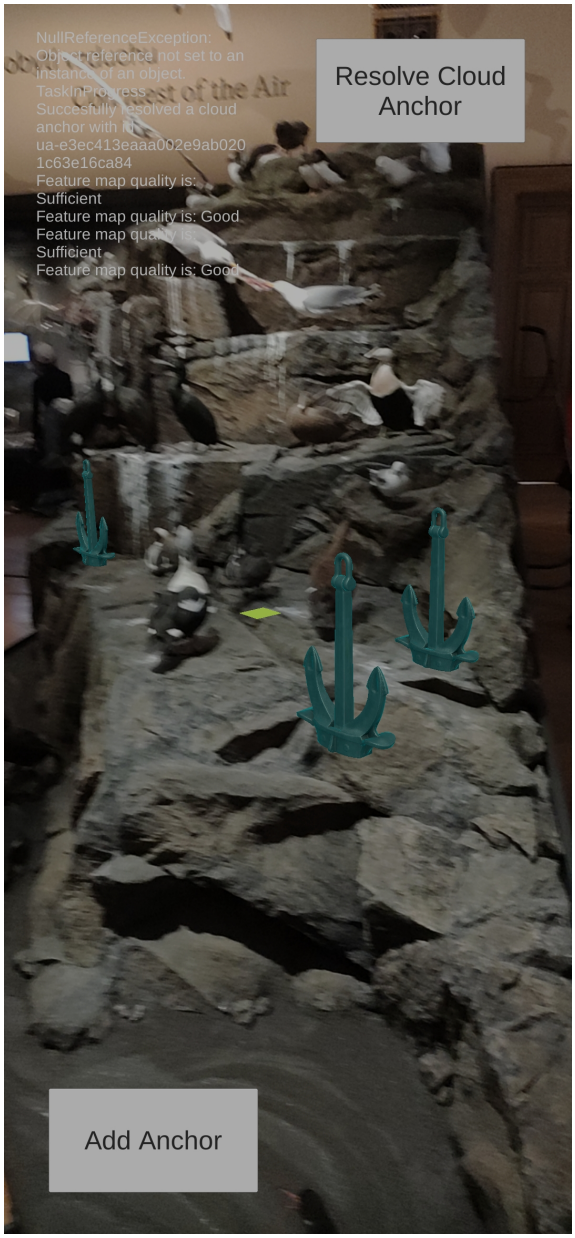
Obrázek 4.6. Nejlepší výsledek rozpoznání kotevního bodu ve spodní části pantheónu.

Obrázek 4.7. Výsledek rozpoznání kotevního bodu na ochozu pantheónu.

Obrázek 4.8. Oproti výsledku na obrázku 4.6 je posunutí o několik metrů a rotace o cca 110 stupňů.

Obrázek 4.9. Oproti výsledku na obrázku 4.8 je posunutí dvojnásobné. Rotace je posunutá o cca 30 stupňů.

Třetím scénářem byla skalka s mořským ptactvem opět z expozice Zázraky evoluce. Skalka samotná měla matně šedivou barvu a mnoho hran. Velké množství různobarevných ptáků také přispívalo k mnohem lepšímu zmapování než v předchozích dvou scénářích, díky čemuž bylo dosaženo dobré kvality mapování. Hostování bylo velmi rychlé, stejně tak i rozpoznání, po kterém byly objekty na totožných pozicích i se správnou rotací (Obrázek 4.10). Tento scénář byl zdaleka nejúspěšnější v celém muzeu.



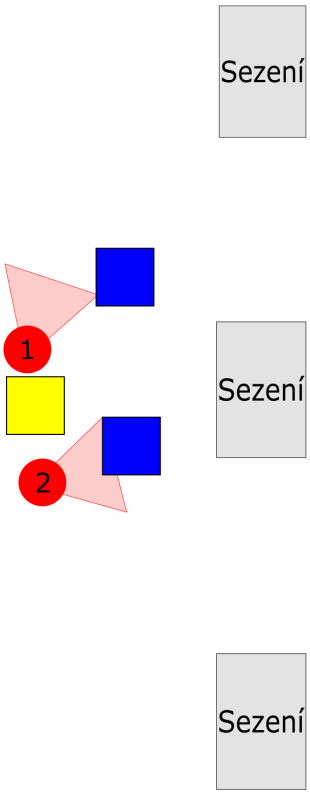
Obrázek 4.10. Výsledek rozpoznání skalky s mořským ptactvem totožný se stavem při hostování.

Čtvrtým scénářem byla kostra plejtváka myšoka ze stejné expozice (Obrázek 4.11). Světelné podmínky byly oproti místnosti s krakaticí lepší, protože místnost byla více osvětlená. Kvalita mapování byla maximálně dostatečná. Změna prostředí byla silná, protože místnost, ve které se kostra nachází, má největší otevřený prostor. Lidé se tedy pohybují prakticky vždy v celém záběru odkudkoliv. Rozpoznání bylo vcelku rychlé. Pozice byla o maximálně dva metry posunutá. Rotace byla výrazně odlišná. V otevřeném prostoru se nenacházely žádné vhodné orientační body, pouze jednobarevná podlaha a několik matných černých sedaček.

Obrázek 4.11. Kostra plejtváka myšoka v Národním muzeu. Zdroj: `https://www.nm.cz/prirodovedecke-muzeum/zazraky-evoluce`

Posledním scénářem byla dvorana. Velký otevřený prostor. Všechny stěny byly vcelku členité, ale stereotypní. Podlaha podobně jako v pantheónu lesklá s opakujícími se vzory. Světelné podmínky normální. Změna prostředí minimální, protože ve dvoraně nikdo nebyl. Kvalita mapování byla maximálně dostatečná. V místnosti jsou pouze dvě skupiny matných šedých krychlí, na které si mohou návštěvníci sednout. Z celkových čtyř pokusů o rozpoznání byly dva úplně neúspěšné a zbylé dva pozicí i rotací zdaleka neodpovídaly stavu při hostování ani sobě navzájem (Obrázky 4.12, na plánku číslo 1 a 4.14, na plánku číslo 2).

Obrázek 4.12. První rozpoznání ve dvoraně.



Obrázek 4.13. Plánek dvorany.

Obrázek 4.14. Druhé rozpoznání ve dvoraně.

4.1.2 Testování v domácím interiéru

Veškerá testování uživateli probíhala za denního světla nebo dostatečného interiérového osvětlení. Persistenci scény v domácím interiéru slovně ohodnotili tři uživatelé. Všem třem se nepodařilo namapovat prostředí alespoň na dostatečnou úroveň, i když se snažili mapovat různé prostory i delší dobu. Zároveň se všichni setkali s odlišnou pozicí a rotací po rozpoznání kotevního bodu. Tři uživatelé zmínili, že jim objekt tzv. plaval, tedy že se nedržel na stejné pozici, ale částečně kopíroval pohyb mobilního zařízení a pohyboval se. V praxi plavání objektu výrazně ztíží jeho obcházení a manipulaci s ním, protože při snaze ho jakkoliv obejít se pohybuje pomaleji společně s uživatelem.

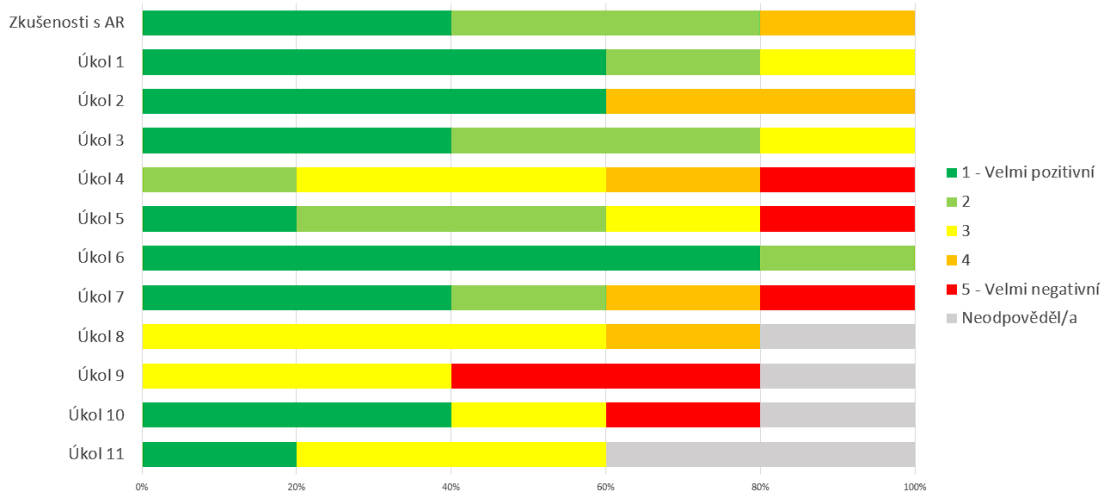
Při vývoji aplikace jsem zjistil, že pokud uživatel i při nedostatečné úrovni mapování rozpoznává kotevní bod ze stejného nebo velmi podobného místa a úhlu, získá většinou přesný výsledek s odchylkou maximálně do 20 centimetrů a prakticky stejnou rotací. V

praxi nelze takovou situaci očekávat a bude častěji docházet k nepřesným rozpoznáním, které získávali uživatelé během testování.

4.2 Testování uživatelské přívětivosti

První překážkou v testování aplikace se ukázaly být hardwarové a softwarové požadavky ARCore a Firebase balíčků pro Unity. Jeden z uživatelů chtěl aplikaci vyzkoušet na mobilním zařízení OnePlus 5T, které je v seznamu podporovaných [8]. Během ledna 2022 aplikaci k semestrálnímu projektu předcházejícímu této bakalářské práci neměl problém nainstalovat a otestovat. Finální verzi aplikace už ani nenainstaloval. Případná aktualizace balíčků by neměla mít vliv na dostupnost na mobilních zařízeních. Obzvlášť, když jsou stále napsané jako oficiálně podporované.

Aplikaci hodnotilo pět uživatelů různého věku a různého zaměření. Zastoupena byla obě pohlaví a také jak studenti, tak pracující. První z otázek cílila na zjištění zkušeností jednotlivých uživatelů s AR. Uživatelé dále dostali jedenáct jednoduchých úkolů a u každého zvlášť byli dotázání, jak jednoduchý/složitý pro ně úkol byl. Poslední tři úkoly byly pouze na zhodnocení rozdílu pozice a rotace rozpoznání oproti hostování. Úplné znění dotazníku naleznete v příloze B.



Obrázek 4.15. Odpovědi v testovacím dotazníku.

Předešlou větší **zkušenost s AR** měli čtyři z pěti uchazečů.

První úkol (umístění nového kotevního bodu) hodnotili pozitivně čtyři z pěti uchazečů a pátý neutrálně.

Druhý úkol (přejmenování kotevního bodu) sice tři z pěti hodnotili jako velmi jednoduchý, zbylí dva naopak negativně. Důvodem je, že bez postupu v dotazníku (Který tito dva konkrétní testující schválně nečetli. Četli pouze zadání úkolů.) v GUI nelze dobře rozeznat, kde se kotevní bod přejmenovává. Jednoduchým řešením je přidat vedle editovatelného textového pole pro jméno značku "Anchor Name:", kterým bude problém identifikace vyřešen.

Třetí úkol (vytvoření nového objektu) byl spíše pozitivně hodnocen a tedy způsob tvorby je vhodný.

Hodnocení **čtvrtého úkolu** (pohyb objektem ve scéně) poukazuje na nedostatky aplikace v tomto ohledu. Pro efektivní pohyb objektem je vhodné stát kolmo na osu, se kterou chce uživatel hýbat, což ho vede k chůzi kolem objektu, aby si správně stoupnul. V jednom slovním hodnocení je zmíněno, že pohyb s objektem byl možný pouze v

jednom směru osy, ale ne v opačném, což mohlo být způsobeno právě místem, odkud uživatel objektem hýbal. Řešením by bylo vytvořit GUI nezávislé na pozici uživatele, ve kterém by ale zároveň byl vidět směr, kterým lze s objektem hýbat. Jednou z možností je např. statické GUI ve formě dalších tlačítek / sliderů pro posun.

Pátý úkol (rotace objektem ve scéně) hodnotí uživatelé spíše pozitivně. Ze slovní zpětné vazby i z vlastních zkušeností je lepší mít pro rotaci oblouky (podobné jako např. na Obrázku 2.6), než malé šipky, které se z větší vzdálenosti a ze špatného úhlu velmi špatně stiskávají prstem na displeji.

Šestý úkol (změna velikosti objektu) je nejlépe hodnoceným úkolem. Změna velikosti dvěma prsty pohybem od sebe / k sobě je široce používaným způsobem interakce a z výsledků lze vidět, že většina uživatelů intuitivně věděla, co má dělat.

Sedmý úkol (hostování kotevního bodu) bylo celkově hodnoceno spíše kladně. Negativní hodnocení bylo dodáno s tím, že uživatel nemůže zjistit, jestli v danou chvíli hostuje. Ve výpisu z konzole se při hostování vypíše „TaskInProgress“ a po dokončení hostování (ať už úspěšném nebo selhaném) se doplní informace o výsledku. Jako lepší řešení se nabízí výpis „Trying to host anchor“ a např. načítací kolečko známé z většiny komerčních mobilních i počítačových aplikací.

Osmý úkol (rozpoznání kotevního bodu) bylo jedním uživatelem nevyplněné, protože testoval pouze úkoly pro jednoho uživatele. Zbylé úkoly se dají otestovat i v jednom, je ale výrazně příjemnější mít druhého testujícího. Celkově negativnější hodnocení je spojeno se silně negativním hodnocením následujícího **devátého a desátého úkolu** (porovnání pozice a rotace kotevního bodu a objektů od ostatních uživatelů). Problémem je nepřesné umístění a natočení kotevního bodu, která způsobí, že se objekty mohou zobrazit mimo zorné pole kamery. Uživatelé tak vědí o stavu rozpoznání pouze přes výpis z konzole, nemusí ale v první chvíli nic vidět. Řešením by bylo přidat grafické označení objektů mimo aktuální zorné pole kamery, např. malý kruh / kapku naznačující směr, kterým se musí uživatel natočit, aby objekt viděl.

Jedenáctý úkol (Porovnání přesnosti okamžité maniplace s objektem mezi uživateli, tedy jak se v reálném čase dalším uživatelům zobrazuje manipulace jiného uživatele.) byl hodnocen spíše neutrálně. Ze slovní zpětné vazby vyšlo, že testujícím vadila vzájemně odlišná pozice a rotace, tedy při manipulaci s objektem jednoho uživatele ji druhý viděl v jiných směrech. Řešením je přidat rychlou intuitivní možnost opakovaného rozpoznání kotevního bodu, aby se uživatelé mohli pokusit rozpoznat kotevní bod lépe a viděli vzájemně bod i objekty na stejných místech. Responzivita a rychlost synchronizace mezi uživateli byla hodnocena pozitivně.

Kapitola 5
Závěr

V práci byly představeny nejznámější technologie pro sdílení souřadného systému a poté byly mezi sebou porovnány. Následně byla vybrána ta nejvhodnější a společně s výběrem vývojového prostředí a způsobu ukládání dat navrhnuta a implementována aplikace pro správu objektů v AR scéně. Byla otestována jak funkčnost vybrané technologie Cloud Anchors v Národním muzeu i v domácím prostředí, tak obsluha aplikace uživateli. Technologie Cloud Anchors ukázala, že je náchylná na změny světelných podmínek a pro použitelnost vyžaduje co nejvyšší úroveň zmapování prostředí pro následné dobré rozpoznávání kotevních bodů. Uživatelé při testování aplikace poukázali na chyby a nedostatky aplikace samotné, která by po jejich odstranění mohla být uživatelsky přívětivější. Celkově lze říci, že při stabilním dobrém osvětlení a trpělivosti při tvorbě AR scén je Cloud Anchors na platformě Android zajímavou volbou, která určitě stojí za zvážení díky velkému počtu uživatelů mezi lidmi, v našem případě návštěvníků muzea. Práce ukázala výhody i nevýhody použitých technologií, způsobů a metod, ze kterých lze nadále stavět a dostat se k plně funkční a přívětivé aplikaci, která zpestří všední návštěvu muzea nevšedními techologiemi. V rozšířené realitě vidím velkou budoucnost. Tento pohled sdílí mnoho laiků i profesionálů. Proto jsem velmi rád, že jsem mohl pracovat na této bakalářské práci a podpořit Národní muzeum i formou vlastní kreativity a tvorby.

Literatura

[1] P. MILGRAM ET AL. Augmented reality: A class of displays on the reality-virtuality continuum. Proceedings of SPIE - The International Society for Optical Engineering. 1994, 2351, 10.1117/12.197321. Dostupné z: https://www.researchgate.net/publication/228537162_Augmented_reality_A_class_of_displays_on_the_reality-virtuality_continuum

[2] STATCOUNTER. Mobile Operating System Market Share Worldwide - April 2022 [Internet]. StatCounter, 2022. [4.5.2022]. Dostupné z: https://gs.statcounter.com/os-market-share/mobile/worldwide

[3] D. SCHMALSTIEG, T. HÖLLERER Augmented reality - Principles and Practice. 1. USA: Addison-Wesley, 2016. 978-0321883575.

[4] MICROSOFT AZURE. Azure Spatial Anchors [Internet]. Microsoft, 2022. [4.5.2022]. Dostupné z: https://azure.microsoft.com/en-us/services/spatial-anchors/

[5] WIKIPEDIA. Minecraft Earth [Internet]. Wikipedia, 2022. [4.5.2022]. Dostupné z: https://en.wikipedia.org/wiki/Minecraft_Earth

[6] NEOGOMA. Stardust SDK [Internet]. Neogoma limited, 2022. [5.5.2022]. Dostupné z: https://neogoma.com/

[7] ARWAY. On-Cloud Localization [Internet]. ARWAY, 2020. [5.5.2022]. Dostupné z: https://www.arway.app/

[8] GOOGLE DEVELOPERS. ARCore supported devices — Google Developers [Internet]. Google LLC, 2021. [19.11.2021]. Dostupné z: https://developers.google.com/ar/devices

[9] IMMERSAL. Immersal Cloud Service & SDK [Internet]. Immersal Ltd., 2021. [19.11.2021]. Dostupné z: https://immersal.gitbook.io/sdk/

[10] L. K. DALE. Unity - does indie gaming's biggest engine have an image problem? [Internet]. Guardian News & Media Limited, 2022. [11.1.2022]. Dostupné z: https://www.theguardian.com/technology/2015/jul/06/unity-indie-gamings-biggest-engine-john-riccitiello

[11] IMMERSAL. Immersal Mapper - SDK [Internet]. Immersal Ltd., 2022. [6.5.2022]. Dostupné z: https://immersal.gitbook.io/sdk/tutorials/how-to-map/immersal-mapper

[12] J. LAVIOLA, E. KRUIJFF, D. BOWMAN, R. MCMAHAN & I. POUPYREV. 3D User Interfaces: Theory and Practice. 2. USA: Addison-Wesley, 2017. 978-0134034485.

[13] DB-ENGINES. Popularity changes per category, May 2022 [Internet]. solid IT GmbH., 2022. [6.5.2022]. Dostupné z:
https://db-engines.com/en/ranking_categories

[14] KHRONOS. glTF Overview - The Khronos Group Inc [Internet]. The Khronos® Group Inc., 2022. [9.5.2022]. Dostupné z:
https://www.khronos.org/gltf/

[15] T. BRIGSTED. Siccity/GLTFUtility: Simple GLTF importer for Unity [Internet]. GitHub Inc., 2022. [9.5.2022]. Dostupné z:
https://github.com/Siccity/GLTFUtility

[16] ENDORA. Freehosting [Internet]. Freehosting ENDORA.cz, 2022. [11.5.2022]. Dostupné z:
https://www.endora.cz/

[17] T. HAVLÍK. DebugConsole.cs [Internet]. GitHub Inc., 2021. [9.5.2022]. Dostupné z:
https://gist.github.com/tomires/35f61063f2a38d596a5d125a5d11ba26

[18] UNITY. Unity - Scripting API: Application.persistentDataPath [Internet]. Unity Technologies, 2022. [11.5.2022]. Dostupné z:
https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html

[19] MICROSOFT. Events (C# Programming Guide) [Internet]. Microsoft, 2022. [11.5.2022]. Dostupné z:
https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/events/

Příloha A
Návody ke spuštění

Tato příloha kopíruje soubor README.md obsažený v přílohách a vysvětluje instalaci a spuštění aplikace, případně Unity pojektu.

A.1 Aplikace

Do telefonu s minimální verzí operačního systému Android 7.0 a podporou ARCore zkopírujte soubor cloudAnchors.apk, který získáte extrahováním stejnojmenného zip souboru. Klepněte na něj ve svém průzkumníku souborů. Tím se aplikace nainstaluje a můžete ji spustit jako jakoukoliv jinou aplikaci na telefonu.

A.2 Unity projekt

Unity projekt je rozdělen do 3 částečných zip souborů. Označte najednou všechny 3 a společně je rozbalte. Stáhněte si Unity Hub (`https://unity3d.com/get-unity/download`). Po stažení v Unity Hub nainstalujte verzi Unity 2020.3.21f1 nebo novější (V záložce Installs tlačítko ADD) a v možnostech instalace přidejte Android Build Support. Po stažení a nainstalování v sekci Projects klepněte na ADD a proklikejte se k rozbalenému projektu. Poté stačí zvolit verzi Unity, ve které se má projekt spustit, klikněte na něj, a Unity začne projekt otevírat. Po každém spuštění Unity projektu bude potřeba zadat hesla pro komunikaci s Firebase databází. V editoru klikněte vlevo nahoře na Edit a vyberte Project Settings, ve kterých vyberte Player. V Player vyberte ikonku Androidu a sjeďte dolů k Publishing Settings. V případě, že by se klíčový soubor nenačetl sám, v kořenové složce Unity projektu je soubor user.keystore, který nastavte jako Project Keystore. K zadání jsou dvě hesla, která najdete na konci souboru README. Tím byste měli být schopni projekt spustit v editoru i případně sestavit.

Příloha B
Dotazník uživatelského testování

V rámci své bakalářské práce testuji funkčnost a uživatelskou přívětivost vytvořené aplikace pro umisťování kotevních bodů a objektů v prostoru. Testování probíhá ve více lidech naráz pro otestování práce více uživatelů na jednom kotevním bodu v jednu chvíli.

Do svého mobilního zařízení (Musí být na platformě Android.) stáhněte instalační soubor cloudAnchors.apk a nainstalujte aplikaci. Po spuštění vás aplikace požádá o povolení přístupu ke kameře. Přístup povolte a začněte mobilním zařízením pomalu snímat své okolí.

B.1 Úkoly

Body 8-11 jsou nezávislé na bodu 1-7. Proto doporučuji plnit úkoly postupně od prvního nebo osmého. Lze vyplnit i všechny úkoly najednou. Pokud vám aktuální stav aplikace nevyhovuje, aplikaci zavřete, vymažte z paměti a otevřete znovu.

1. Umístěte libovolně do prostoru nový kotevní bod. Ten umístíte tak, že při zobrazení žlutého čtverečku uprostřed obrazovky namíříte zařízením na požadované místo a stisknete tlačítko Add Anchor. V případě, že chcete změnit pozici kotevního bodu, stiskněte tlačítko Remove Anchor a vytvořte nový kotevní bod na vybrané pozici.
2. Umístěný kotevní bod si přejmenujte. V pravém horním rohu najdete editovatelné textové pole, ve kterém nyní bude napsáno ”New Anchor x”, kde x bude nějaké číslo. Klepněte na textové pole a libovolně přejmenujte kotevní bod.
3. Umístěte libovolně do prostoru nový/é objekt/y. Namiřte zařízení tak, aby žlutý čtvereček uprostřed obrazovky byl ve vámi vybrané pozici a stiskněte tlačítko Add Object. Zobrazí se vám seznam, ze kterého vyberte libovolný 3D model. Po výběru chvíli vyčkejte a na zmíněném místě ve scéně se vám zobrazí vybraný 3D model. Pokud máte ve scéně více objektů, vyberete jeden z nich jako aktivní tak, že na něj klepnete prstem. Úspěšný výběr objektu poznáte tak, že se vám kolem objektu zobrazí rozhraní pro pohyb a rotaci. Pokud chcete objekt smazat, stiskněte tlačítko Remove Object, které smaže aktuálně vybraný objekt.
4. Pohněte s libovolným/i objektem/y s pomocí dlouhých barevných šipek kolem něj. Posun ve směru znázorněném šipkou provedete tak, že položíte prst na šipku a držíte. Během držení pohybujte prstem po displeji doprava a doleva / nahoru a dolů, čímž hýbete s celým objektem.
5. Otočte libovolný/é objekt/y kolem jeho lokálních souřadnicových os. Položte prst na jednu z kulatých šipek nacházejících se za špičkou rovných šipek pro pohyb a držte. Stejně jako v předešlém případě pohybujte prstem po displeji doprava a doleva / nahoru a dolů. Tím rotujete objekt kolem vybrané osy.
6. Libovolně zvětšete/zmenšete libovolný/é objekt/y. Po výběru libovolného objektu položte dva prsty na displej a pohybujte jimi od sebe a k sobě. Tím měníte velikost objektu.

7. Hostujte vámi vytvořený kotevní bod. Během chodu aplikace v levém horním rohu uvidíte jednu nebo vícero hlášek "Feature map quality is Insufficient/Sufficient/Good", které vám říkají, jak dobře máte namapované okolí kotevního bodu. Pokud chcete zlepšit kvalitu mapování, mířte a pohybujte zařízením různě skrz scénu, dokud nedostanete oznámení o lepší kvalitě mapování. Při dosažení libovolné úrovně mapování stiskněte tlačítko Host Cloud Anchor. Chvíli vyčkejte a zobrazí se vám informace o úspěšném/neúspěšném hostování. V případě neúspěšného hostování zlepšete kvalitu mapování a zkuste hostování znovu.
8. Zkuste rozpoznat kotevní bod vytvořený dalším uživatelem. V pravém horním rohu uvidíte rozbalovací seznam, ve kterém jsou všechny existující kotevní body. Vyberte jeden z těch, který uživatelé kolem vás hostovali a mířte zařízením přibližně do prostoru, kde byl kotevní bod vytvořen. Pokud ani po dlouhé době aplikace nenalezne bod, vyberte ze seznamu možnost None a zkuste rozpoznávání znovu.
9. Při úspěšném rozpoznání porovnejte pozici kotevní bodu a objektů s dalšími uživateli připojenými na stejný kotevní bod.
10. Při úspěšném rozpoznání porovnejte rotaci kotevní bodu a objektů s dalšími uživateli připojenými na stejný kotevní bod.
11. Stejně jako v bodě 4 a 5 pohybujte a otáčejte libovolným objektem u kotevního bodu, který ve stejnou chvíli vidí i další uživatel/é. Objekt nebo kotevní bod můžete také smazat pomocí tlačítka Remove Object, respektive Remove Anchor.

B.2 Otázky a odpovědi

Odpovědi jsou na lineární stupnici od 1 do 5. U první otázky 1 odpovídá velké zkušenosti a 5 žádné zkušenosti. Od druhé do deváté otázky 1 znamená velmi jednoduché a 5 velmi složité. Od desáté do dvanácté se význam stupňů mění na 1 - testovaná vlastnost byla stejná nebo velmi podobná a 5 - testovaná vlastnost byla úplně jiná.

1. Jaké máte zkušenosti s aplikacemi pro mobilní zařízení využívající rozšířenou realitu?
2. Jak jednoduché / složité pro vás bylo umístění kotevního bodu? (Úkol 1)
3. Jak jednoduché / složité pro vás bylo přejmenování kotevního bodu? (Úkol 2)
4. Jak jednoduché / složité pro vás bylo vytvoření nového objektu a jeho výběr? (Úkol 3)
5. Jak jednoduché / složité pro vás bylo pohybování vybraným objektem ve scéně? (Úkol 4)
6. Jak jednoduché / složité pro vás bylo otáčení vybraného objektu ve scéně? (Úkol 5)
7. Jak jednoduché / složité pro vás bylo měnění velikosti objektu ve scéně? (Úkol 6)
8. Jak jednoduché / složité pro vás bylo hostování kotevního bodu? (Úkol 7)
9. Jak jednoduché / složité pro vás bylo rozpoznání vybraného kotevního bodu? (Úkol 8)
10. Jak se lišila pozice kotevního bodu a jeho objektů od ostatních uživatelů? (Úkol 9)
11. Jak se lišila rotace kotevního bodu a jeho objektů od ostatních uživatelů? (Úkol 10)
12. Jak přesně byla vidět vaše manipulace s objektem ve scéně u dalších uživatelů sledujících stejný kotevní bod / objekt? (Úkol 11)
13. Máte k vytvořené aplikaci nějaké další poznámky, podněty a připomínky?