

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Vývoj ukázkové aplikace pro rozšířenou realitu
Diplomová práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Petr Dolejš**
Osobní číslo: **I21272**
Studijní program: **N0613A140007 Informační technologie**
Téma práce: **Vývoj ukázkové aplikace pro rozšířenou realitu**
Zadávací katedra: **Katedra softwarových technologií**

Zásady pro vypracování

Cílem práce je navrhnout a vytvořit aplikaci pro rozšířenou realitu ideálně ve vnitřním prostředí (např. uvnitř místnosti). Aplikace by měla sloužit jako ukázka možností, které rozšířená realita skýtá. V rámci práce by mělo dojít k návrhu samotného procesu tvorby aplikace společně s návrhem serveru, kde budou uložena data potřebná pro zobrazení. Předpokládá se využití platformy Unity3D pro mobilní zařízení se systémem Android. V práci by měla být popsána teoretická východiska jako jsou např. typy a technologie pro rozšířenou realitu, hledisko bezpečnosti atp. V praktické části by mělo dojít k vývoji aplikace, která by ukázala možnosti rozšířené reality v praxi např. pro edukační účely.

Rozsah pracovní zprávy: **50-60 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

Linowes, J., Babilinski, K. *Augmented Reality for Developers: Build Practical Augmented Reality Applications with Unity, Arcore, Arkit and Vuforia*. Packt Publishing, 2017.

Kebo, V., Kodym, O. *Virtuální realita a řízení procesů: (odborná publikace)*. Ostrava: VŠB-TU Ostrava ve vydavatelství Montanex, 2011. ISBN 978-80-7225-361-6

Vedoucí diplomové práce: **Ing. Jan Panuš, Ph.D.**
Katedra informačních technologií

Datum zadání diplomové práce: **8. listopadu 2022**
Termín odevzdání diplomové práce: **19. května 2023**

Ing. Zdeněk Němec, Ph.D. v.r.
Děkan

L.S.

prof. Ing. Antonín Kavička, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 30. listopadu 2022

PROHLÁŠENÍ

Prohlašuji:

Práci s názvem Vývoj ukázkové aplikace pro rozšířenou realitu jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 24. 08. 2023

Bc. Petr Dolejš v.r.

PODĚKOVÁNÍ

Rád bych touto cestou vyjádřil poděkování Ing. Janu Panušovi, Ph.D., za jeho cenné rady, doporučení a trpělivost při vedení mé diplomové práce. Taktéž bych chtěl poděkovat všem, kteří mě podporovali, zejména rodičům a bratrovi.

ANOTACE

Tato diplomová práce se zabývá možnostmi využití rozšířené reality ve vnitřním prostředí. V práci jsou představeny jednotlivé typy rozšířené reality, technologie pracující s touto realitou a enginy, ve kterých lze vytvářet aplikace využívající rozšířenou realitu. Dále jsou popsána zařízení, na kterých můžeme s rozšířenou realitou pracovat. Z praktického hlediska je zde vysvětlen návrh a vývoj mobilní aplikace pro zařízení s operačním systémem Android, která byla vytvořena v enginu Unity. Konečná aplikace ukazuje možnosti rozšířené reality, které nabízí technologie ARCore. Spolu s aplikací byl také navržen a představen server pro ukládání dat.

KLÍČOVÁ SLOVA

rozšířená realita, ARCore, mobilní aplikace, Android, Unity, GLB, server

TITLE

Development of a sample application for augmented reality

ANNOTATION

This thesis deals with the possibilities of using augmented reality in indoor environments. The thesis introduces different types of augmented reality, technologies working with augmented reality, and engines in which applications using augmented reality can be created. Furthermore, the devices on which we can work with augmented reality are described. From a practical point of view, the design and development of a mobile application for Android devices, created in the Unity engine, is explained. The final application demonstrates the possibilities of augmented reality offered by ARCore technology. Along with the application, a custom data storage server was also designed and presented.

KEYWORDS

augmented reality, ARCore, mobile application, Android, Unity, GLB, server

OBSAH

OBSAH.....	7
SEZNAM OBRÁZKŮ.....	9
SEZNAM TABULEK.....	9
SEZNAM ZDROJOVÝCH KÓDŮ.....	10
ÚVOD.....	11
1 ROZŠÍŘENÁ REALITA.....	12
1.1 Ve vnitřním prostředí.....	14
1.2 Ve vnějším prostředí.....	15
1.3 Typy AR.....	16
1.3.1 Marker-based AR.....	16
1.3.2 Markerless AR.....	17
1.3.3 Projection-based AR.....	18
1.3.4 Location-based AR.....	18
1.4 AR Technologie.....	19
1.4.1 ARCore.....	19
1.4.2 ARKit.....	20
1.4.3 AR Foundation.....	20
1.4.4 Vuforia.....	21
1.4.5 MARS.....	22
1.5 Bezpečnost.....	23
2 ENGINY AR.....	25
2.1 Unity.....	25
2.2 Unreal Engine.....	26
3 AR ZAŘÍZENÍ.....	28
3.1 Mobilní zařízení.....	28
3.2 Headset, chytré brýle.....	28
3.3 AR kontaktní čočky.....	29
4 NÁVRH APLIKACE.....	30
4.1 3D modely.....	30
4.2 Použité technologie.....	32
4.2.1 Unity.....	32
4.2.2 C#.....	33
4.2.3 Android.....	34
4.2.4 AR Foundation, ARCore.....	35

4.3	Analýza	36
4.3.1	Funkční požadavky	36
4.3.2	Nefunkční požadavky	36
4.3.3	Use case diagram	36
4.3.4	Use case specifikace	38
4.3.5	Diagram aktivit	38
5	VÝVOJ APLIKACE	40
5.1	HTTP server	40
5.2	Import GLB modelů	42
5.2.1	Siccity.GLTFUtility	42
5.2.2	atteneder.gltFast	43
5.3	Export GLB modelů	43
5.4	Struktura aplikace AR Magic	43
5.4.1	AR Marker	44
5.4.2	AR Face	49
5.4.3	AR Area	50
5.4.4	AR Video	50
5.4.5	UtilityApp	52
5.5	Uživatelská příručka	53
5.5.1	Aplikace Model Converter	53
5.5.2	Aplikace AR Magic	54
6	VYUŽITÍ APLIKACE	59
7	TESTOVÁNÍ	60
	ZÁVĚR	62
	POUŽITÁ LITERATURA	63
	OBSAH CD	69

SEZNAM OBRÁZKŮ

Obrázek 1: Digitální reality. Zdroj [4].	12
Obrázek 2: Aplikace IKEA Place. Zdroj [6].	14
Obrázek 3: Aplikace PeakVisor. Zdroj [11].	16
Obrázek 4: Příklad Marker-based AR. Zdroj [15].	17
Obrázek 5: Příklad Markerless AR. Zdroj [17].	17
Obrázek 6: Příklad Projection-based AR. Zdroj [20].	18
Obrázek 7: Příklad Location-based AR. Zdroj [21].	19
Obrázek 8: Zobrazení digitálního obsahu na statických prvcích. Zdroj [30].	22
Obrázek 9: Hierarchie ARCore, ARKit, AR Foundation a MARS. Zdroj [34].	23
Obrázek 10: AR headsety Microsoft HoloLens 2, Apple Vision Pro. Zdroj [50].	28
Obrázek 11: AR brýle Nreal Air. Zdroj [52].	29
Obrázek 12: AR kontaktní čočka společnosti Mojo Vision. Zdroj [53].	29
Obrázek 13: Rozhraní Unity editoru. Zdroj [59].	32
Obrázek 14: Vrstvená architektura OS Android. Zdroj [65].	34
Obrázek 15: Use case diagram aplikace AR Magic. Zdroj vlastní.	37
Obrázek 16: Diagram aktivit – přidání digitálního obsahu ve scéně AR Marker. Zdroj vlastní.	39
Obrázek 17: Vytvořená aplikace Model Converter. Zdroj vlastní.	54
Obrázek 18: Scény Menu, Options a zobrazení modelů. Zdroj vlastní.	55
Obrázek 19: Zobrazení části AR Marker. Zdroj vlastní.	56
Obrázek 20: Zobrazení části AR Face. Zdroj vlastní.	57
Obrázek 21: Zobrazení části AR Area. Zdroj vlastní.	57
Obrázek 22: Zobrazení části AR Video. Zdroj vlastní.	58

SEZNAM TABULEK

Tabulka 1: Funkcionality AR balíčků. Zdroj [25].	21
Tabulka 2: Funkcionality ARCore a ARKit v Unreal Engine AR. Zdroj [47].	27
Tabulka 3: Funkční požadavky aplikace AR Magic. Zdroj vlastní.	36
Tabulka 4: Nefunkční požadavky aplikace AR Magic. Zdroj vlastní.	36
Tabulka 5: Use case specifikace – Přidání digitálního obsahu do scény. Zdroj vlastní.	38

SEZNAM ZDROJOVÝCH KÓDŮ

Zdrojový kód 1: Nastavení konfiguračního souboru httpd.conf pro Apache server	40
Zdrojový kód 2: Výpis příkazu ipconfig v Příkazovém řádku	41
Zdrojový kód 3: Kód pro zpracování požadavku na serveru.....	42
Zdrojový kód 4: Ukázka kódu pro importování 3D modelu pomocí projektu GLTFUtility....	42
Zdrojový kód 5: Zápis stažených dat do zařízení.	43
Zdrojový kód 6: Ukázka kódu pro exportování 3D modelu pomocí projektu UnityGLTF.	43
Zdrojový kód 7: Přidání a odebrání funkce po rozpoznání značky	44
Zdrojový kód 8: Funkce prováděná po rozpoznání značky	44
Zdrojový kód 9: Přidání modelů konfigurace a třísového objektu po rozpoznání značky	45
Zdrojový kód 10: Vložení modelu při kliknutí prstem na rozpoznanou plochu.....	46
Zdrojový kód 11: Pohyb modelu po scéně pomocí pohybu prstem	47
Zdrojový kód 12: Načtení 3D modelu po jeho vybrání	48
Zdrojový kód 13: Přidání značky do knihovny značek	49
Zdrojový kód 14: Funkce prováděná po rozpoznání obličeje.....	49
Zdrojový kód 15: Nastavení VideoPlayer a AudioSource komponent.....	50
Zdrojový kód 16: Ukázka kódu pro přípravu videa, nastavení textury a spuštění videa.....	51
Zdrojový kód 17: Ukázka metod pro ovládání videa	51
Zdrojový kód 18: Uložení párů do zařízení ve formátu JSON.....	51
Zdrojový kód 19: Třída pro ukládání párů značka-video	52
Zdrojový kód 20: Ukázka kódu pro stažení dat ze serveru.....	52
Zdrojový kód 21: Ukázka kódu pro upload konfigurace na server	53

ÚVOD

V této diplomové práci bude čtenáři představena rozšířená realita a její možnosti využití ve vnitřním prostředí. Tato rozšířená realita umožňuje přidávat digitální obsah do prostředí reálného světa ve formě 3D modelů, a v dnešní době je rozšířena do velkého množství mobilních zařízení. V těchto zařízeních je možné ji využít v široké škále odvětví mezi které patří školství, výroba nebo armádní využití.

V teoretické části této práce budou popsány jednotlivé technologie, které je možné využít při vytváření aplikací s rozšířenou realitou. Budou také představeny již existující aplikace pro práci s rozšířenou realitou ve vnitřním a vnějším prostředí. Spolu s tím budou popsány jednotlivé typy rozšířené reality, které můžeme rozdělit na typy založené na rozpoznání značky (marker-based AR) a bez značky (markerless AR). Závěrem teoretické části budou představena zařízení, díky kterým můžeme s rozšířenou realitou pracovat.

Hlavním cílem praktické části je navrhnout a vytvořit mobilní aplikaci, která bude sloužit jako ukázka možností rozšířené reality. V této části bude nejprve provedena analýza, kde dojde k identifikaci funkčních i nefunkčních požadavků, a spolu s tím dojde k definici funkcionalit aplikace ve formě diagramů. Zároveň budou představeny jednotlivé technologie, které budou potřebné pro vývoj mobilní aplikace. Pro práci s rozšířenou realitou bude využita technologie AR Foundation spolu s ARCore, přičemž vývoj bude probíhat v Unity enginu. K této aplikaci bude také navržen server, na kterém budou uložena data pro zobrazení. Dále budou rozebrány důležité části kódu, které slouží pro práci s rozšířenou realitou a serverem. Na závěr praktické části budou představeny jednotlivé scény mobilní aplikace s popisem jejich funkcí.

Pro ověření správného fungování mobilní aplikace bude aplikace otestována na odlišných Android zařízeních.

1 ROZŠÍŘENÁ REALITA

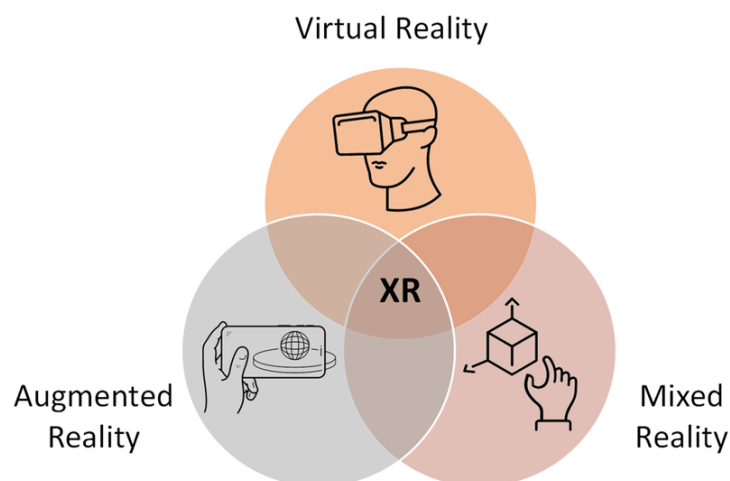
Rozšířená realita (AR) je technologie, která umožňuje vkládat digitální obsah do reálného prostředí v reálném čase. Zatímco Virtuální realita (VR) vytváří celé nové prostředí, v AR uživatelé vidí prostředí reálného světa s přidávanými digitálními objekty a modely [1].

Jedná se o obohacení skutečného světa pomocí přidání digitálních prvků, zvuků nebo jiných smyslových podnětů. Společnosti mohou tuto technologii využít k propagaci svých produktů a služeb, ale také při spouštění nových marketingových kampaní. Obchodní sektor také využil AR pro zlepšení zážitků při nakupování produktů. Firmy v této oblasti začlenily AR do svých katalogů, což umožňuje zaměření referenčního obrázku v katalogu a zobrazení příslušného produktu jako 3D modelu. Technologii lze také využít při nákupu nábytku, díky čemuž je uživatel schopný vložit virtuální stůl do svého pokoje [2].

V aktuální době se velmi rozrůstají digitální reality na trhu. Díky tomu může docházet k nesprávnému pochopení jednotlivých z nich.

Typy digitálních realit:

- Augmented reality (AR) – Vytvořena pro vkládání digitálních objektů do reálného světa s omezenou interakcí.
- Virtual reality (VR) – Prostřednictvím headsetu a sluchátek dokáže zcela izolovat uživatele od skutečného světa, přičemž se pomocí příslušenství cítí ve zcela jiné realitě.
- Mixed reality (MR) – Jedná se o kombinaci AR a VR prvků, která umožňuje interakci mezi uživatelem, digitálním obsahem a skutečným světem.
- Extended reality (XR) – Zahrnuje technologie zlepšující naše smysly, včetně AR, VR a MR [3].



Obrázek 1: Digitální reality. Zdroj [4].

AR se neustále rozrůstá a nabízí pohlcující zážitek pro všechny jeho uživatele. Mezi nejčastější formy AR patří využití pomocí chytrých brýlí nebo kamery (mobilní zařízení, tablet), kde ve všech zařízeních se využívá pěti významných složek AR:

- Umělá inteligence – Naprostá většina řešení AR vyžaduje ke zpracování informací umělou inteligenci.
- AR software – Pro přístup a práci s AR jsou potřebné odpovídající nástroje a aplikace.
- Zpracování – Je nutné mít dostatečný výpočetní výkon pro zpracování potřebných informací.
- Obrazová plocha – Pro zobrazení obsahu je nezbytné mít dostatečně kvalitní obrazovou plochu (mobilní zařízení, tablet).
- Senzory – Pro správnou synchronizaci skutečného a digitálního světa jsou potřeba senzory, které zachytí informace a odešlou se pomocí softwaru ke zpracování [3].

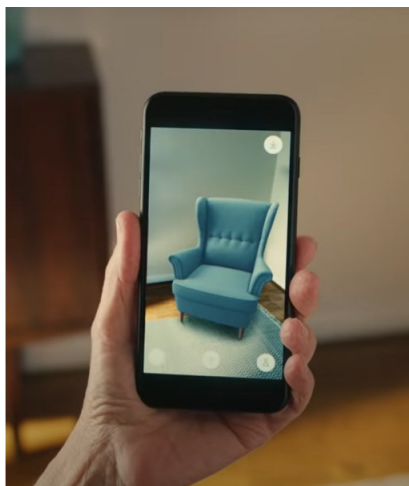
AR dokáže být velmi užitečné v pracovním prostředí. Umožňuje efektivní zaučení nových zaměstnanců stejně jako školení těch aktuálních. Díky AR můžeme zvýšit kvalitu vzdělávacích a školicích materiálů.

Mnoho pracovních odvětví již AR využívá:

- Výroba – AR dokáže poskytovat velmi přesné pracovní postupy, a navíc umožní školitelům okamžitou zpětnou vazbu na školeného.
- Zdravotnictví – Díky AR lze ve zdravotnictví získat praxi bez jakýchkoliv rizik, dokáže poskytnout přesné, praktické a bezpečné návody k operacím nebo informace o anatomii těla a tělesných systémech.
- Armáda – AR pomáhá v armádním prostředí při bojovém výcviku letounem.
- Automobily – Pomocí AR lze prozkoumat všechny komponenty vozidla a rozebrat je do jednotlivých částí pro lepší pochopení dané komponenty [3].

1.1 Ve vnitřním prostředí

Ve vnitřním prostředí je možné AR využívat při nakupování produktů. Pro tento účel vznikla v roce 2017 aplikace IKEA Place, která umožňuje zobrazit nábytek z katalogu přímo ve vašem pokoji. To vytváří lepší představu o tom, jak bude kus nábytku ve vašem pokoji vypadat. Aplikace aktuálně nabízí přes 2000 kusů nábytku k zobrazení a je dostupná pouze na operační systém iOS. Pro Android existuje alternativa s názvem Houzz [5].



Obrázek 2: Aplikace IKEA Place.
Zdroj [6].

AR můžeme využít také pro navigování ve výrobních halách nebo velkých obchodních domech. Ve vnitřních prostorech je obtížné využít klasické navigační systémy GPS nebo GLONASS, tudíž je nutné použít technologii uzpůsobenou tomuto prostředí. Pro správné fungování vnitřní navigace je potřeba vytvořit vhodnou infrastrukturu, která se může skládat z čárových kódů, značek nebo majáků tzv. *beaconů*. S vhodně zvolenou infrastrukturou je k navigování potřeba zařízení s kamerou, typicky se jedná o mobilní zařízení nebo tablet.

Typy infrastruktur:

- AR navigace s vestavěnými mobilními senzory – Pomocí mobilního zařízení měříme elektromagnetické pole, a díky naměřené hodnotě lze navigovat z bodu A do bodu B. Jsou zde nutné pravidelné kontroly, jelikož vnější vlivy mohou ovlivňovat magnetická pole a časem tyto hodnoty měnit.
- AR navigace s majáky – Po budově jsou rozprostřeny majáky s nízkou spotřebou disponující technologií Bluetooth, díky kterým se určí poloha uživatele. Toto řešení vyžaduje velké množství majáků pro co nejlepší výsledky, tudíž je velmi nákladné.

- AR navigace se značkami – Pomocí kamery se detekují značky, které jsou rozmístěny po budově. Na základě této značky víme přesnou polohu uživatele a jakým směrem je otočený. Toto řešení je cenově velmi přívětivé a zároveň přesné [7].

AR ve vnitřním prostředí lze využít také pro vzdělávání. Pro operační systém iOS existuje aplikace AR Anatomy 4D+, která zvládne zobrazit modely srdce, jater a jiných orgánů. Díky tomu si mohou studenti lépe vizuálně představit, jak daný orgán vypadá a jak funguje. K této aplikaci je nutné vytisknout značky, na kterých se po jejich rozeznání mobilním zařízením zobrazí požadovaný model [8].

Pro operační systém Android je dostupná aplikace AR-ANIMALS BOOK, která funguje na stejném principu jako předchozí aplikace. Tato aplikace zobrazuje vybraná zvířata, díky čemuž je vhodná pro základní školy [9].

1.2 Ve vnějším prostředí

AR není technologie, kterou lze použít pouze ve vnitřních prostorech, ale své využití dokáže nalézt i ve venkovním prostředí. Díky AR můžeme lépe prozkoumávat města, navigovat, nebo nám může pomáhat při pozorování hvězd.

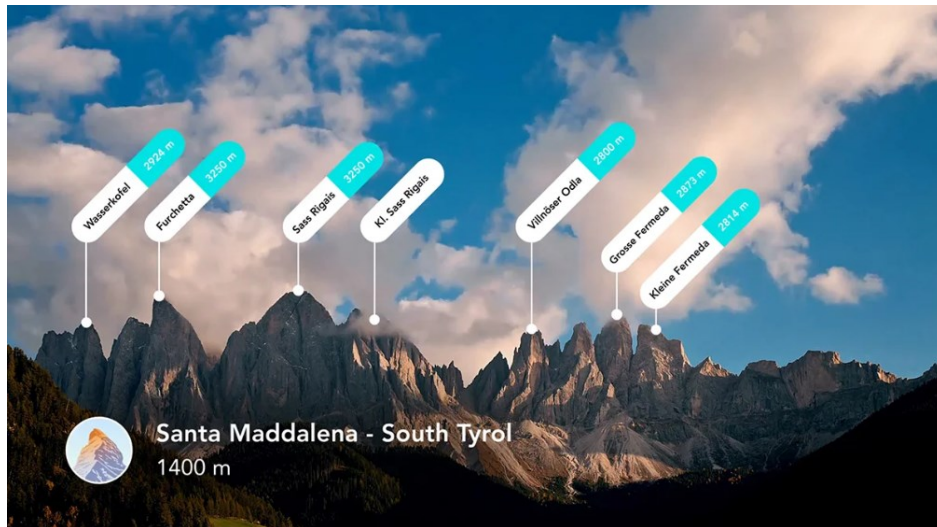
Mezi známé aplikace ve venkovním prostředí můžeme zařadit aplikaci Night Sky, která je vydána pouze pro operační systém iOS. Uživatelé díky ní mohou vidět na zařízení tisíce hvězd, satelitů, planet a konstelací po namíření telefonu na oblohu. Velmi podobnou aplikací je Skyview, která navíc dokáže zobrazit, jak vypadalo seskupení hvězd v minulosti, nebo jak bude vypadat v budoucnosti na základě aktuálních informací o pohybech jednotlivých objektů. Aplikaci lze nainstalovat na iOS i Android [10].

Další oblastí, kde lze využít AR ve venkovním prostředí je cestování. Při cestování do neznámé části světa je vhodné vědět, jaké typy služeb se nacházejí kolem nás. To můžeme zjistit pomocí aplikace World Around Me, která dokáže zmapovat vaše okolí a zobrazit blízké služby jako jsou restaurace, obchody nebo bankomat [11].

AR je také velmi užitečné při navigování. Při použití správné AR aplikace využívající GPS, můžeme skrz mobilní zařízení vidět přímo na ulici vybranou cestu, po které se dostaneme do určeného cíle. V těchto aplikacích lze také zobrazit šipky při změně směru a během navigování mohou být v okolí označena zajímavá místa nebo služby. Jednou z aplikací, která těmito vlastnostmi disponuje je ARCity, která je aktuálně dostupná ve více než 300 městech, a kromě

navigace umožňuje také zobrazit informace o okolních ulicích a památkách. AR navigaci do svých aplikací přidali také Sygic a Yahoo maps [12].

Zobrazení zajímavých míst nemusí být pouze ve městě. Při zdolávání hor je možné využít aplikaci PeakVisor, která nám při rozhlédnutí pomocí telefonu ukáže, jaké vrcholky hor máme v okolí. Tato aplikace ukazuje vrcholky hor po Evropě, Asii, Oceánii a Severní Americe. V aplikaci zatím nejsou registrovány žádné hory na území České republiky, a její nejbližší využití je možné v Rakousku nebo v Německu. Aplikace je určená pro iOS a Android [11].



Obrázek 3: Aplikace PeakVisor. Zdroj [11].

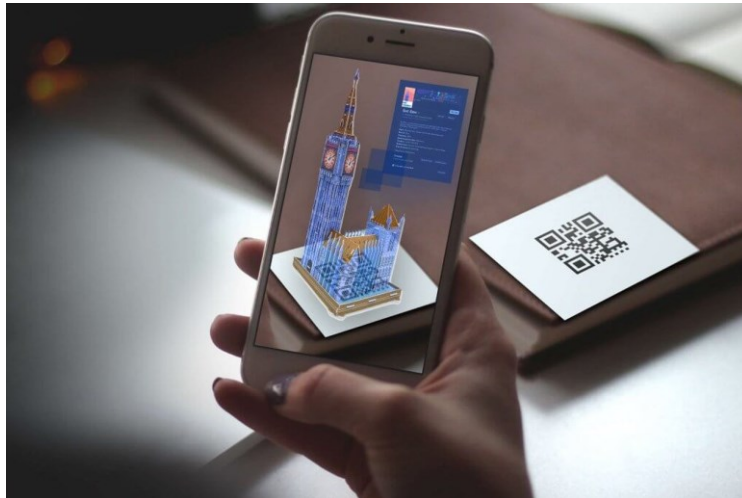
1.3 Typy AR

1.3.1 Marker-based AR

Pro tento typ AR je stěžejní rozpoznání značky neboli markeru. Marker-based AR vyžaduje značku, kdy se po jejím rozpoznání zobrazí příslušný model nebo se spustí příslušná akce. Značky jsou specifické obrazce, které jsou snadno rozpoznatelné a zpracovatelné. Zároveň jsou vizuálně nezávislé na svém okolním prostředí a mohou být vytisknuty na papír nebo je lze zobrazit na obrazovce počítače [13].

Pro spolehlivé rozpoznání by měla být značka také unikátní, jednoduchá a zaostřená. Spolu s tím by měla disponovat vysokým kontrastem, dostatečnou kvalitou a rozměry. Značka by

naopak neměla obsahovat opakující se vzory, prázdná místa a velké množství textu na malé rozloze. Pro značky je vhodné zvolit QR kódy, loga firem nebo personalizované vizitky [14].



Obrázek 4: Příklad Marker-based AR. Zdroj [15].

1.3.2 Markerless AR

Markerless AR neboli AR bez značky může být použita téměř v jakémkoli prostředí, jelikož nespolehá na specifickou značku pro zobrazení digitálního obsahu. Díky tomuto přístupu můžeme tento typ využívat v různých situacích a na různých místech. Markerless AR umožňuje vkládat 3D modely do skutečného světa tím, že si nejprve naskenuje okolní prostředí, ve kterém zjistí přítomnost rovinných ploch. Po kliknutí na jednu z těchto ploch se vloží vybraný 3D model do prostoru. Mezi známé aplikace, které tento přístup využívají patří IKEA Place nebo Pokemon GO [16].



Obrázek 5: Příklad Markerless AR. Zdroj [17].

1.3.3 Projection-based AR

V tomto typu AR není potřeba využívat mobilní zařízení nebo tablet jako v předchozích případech. Je zde využíván projektor, který zobrazuje digitální obsah na objekty a plochy reálného světa [18]. Díky tomu můžeme vytvářet iluze hloubky, polohy a orientace objektů [13].

Tento typ má i svá omezení. Díky tomu, že hlavní komponentou je projektor, lze tento typ používat pouze v zatemněném prostředí a je finančně nákladný. Zároveň pokud budeme provádět interakci s nasvícenou oblastí, bude ruka vykonávající akci vrhat stín na zobrazované prvky [19].

Existují dva typy projekce:

- Statická projekce – Promítá statické digitální prvky na objekty skutečného světa. Tímto přístupem je možné zobrazovat videa, obrázky nebo jiný statický, digitální obsah.
- Dynamická projekce – Tento typ umožňuje promítat interaktivní a pohyblivý digitální obsah na objekty skutečného světa. Díky tomu je možné pracovat s tímto obsahem, který následně reaguje na prováděnou akci v reálném čase [18].



Obrázek 6: Příklad Projection-based AR. Zdroj [20].

1.3.4 Location-based AR

Chytré telefony disponují funkcemi, díky kterým lze velmi přesně určit polohu zařízení. Location-based AR se váže na konkrétní místo, a zpracováním dat z kamery, GPS, digitálního kompasu a akcelerometru se snaží spárovat zjištěné informace se skutečným světem. Při úspěšném spárování se zobrazí příslušné informace, které jsou s daným místem spojeny. Díky využití dříve zmíněných komponent pro zjištění polohy, je tento typ velmi přizpůsobivý, jelikož nepotřebuje žádné značky a lze ho použít kdekoli [13].

Tento typ AR se často využívá pro aplikace zaměřené na turistiku nebo cestování. Kromě měst se dá využít i ve volné přírodě, kde se převážně nevyužívají data z kamery, ale informace o nadmořské výšce, terénu a místní flóře a fauně. Zpracováním těchto informací dokáže aplikace poskytnout uživatelům mapu okolí v rozšířené realitě [16].



Obrázek 7: Příklad Location-based AR. Zdroj [21].

1.4 AR Technologie

1.4.1 ARCore

Technologie ARCore je nástupcem technologie Tango, která byla vyvíjena společností Google. Tango potřebovalo pro své používání velmi drahé senzory pro zmapování svého okolí a vytvoření jeho 3D modelu, na který se následně vkládal digitální obsah. Díky tomu bylo použito pouze v omezeném množství zařízení, mezi který patřil Asus ZenFone AR, a přestalo být podporováno v roce 2018. Google se rozhodl plně soustředit na vývoj ARCore, který mnohem více spoléhá na kamery v chytrých telefonech, díky čemuž se dokázal rozšířit do velkého počtu těchto zařízení [22].

ARCore se zaměřuje na 3 hlavní vlastnosti:

- Sledování pohybu – ARCore využívá kameru chytrého zařízení pro sledování významných bodů v okolí a dat ze snímače IMU (Inertial measurement unity), díky kterým dokáže určit polohu a orientaci telefonu při jeho pohybu. Díky tomu zůstává digitální obsah přesně na svém místě.
- Pochopení okolního prostoru – Digitální obsah je běžné dávat na rovinné plochy jako podlahu nebo stůl. Tyto rovinné plochy je možné rozpoznat díky významným bodům v místnosti, které byly využity, již při sledování pohybu.

- Odhad světla – ARCore dokáže rozpoznat aktuální osvětlení scény a díky tomu také správně osvětlit digitální obsah. Tím vytváří vzhled digitálního obsahu mnohem realističtější [23].

1.4.2 ARKit

ARKit je technologie vyvíjena společností Apple. Byla představena v roce 2017 a na posledních verzích iOS se využívá ARKit 6. Tato technologie používá VIO (Visual Inertial Odometry), a kombinuje data z kamery spolu s CoreMotion daty pro zmapování okolního prostředí. Díky těmto datům dokáže zařízení přesně rozpoznat, jak se pohybuje v prostoru bez nutnosti jakékoliv další kalibrace. Stejně jako ARCore je ARKit schopný rozpoznat horizontální plochy, které mohou být využity pro vložení digitálního obsahu. ARKit také umožňuje s vysokou přesností rozpoznat obličej pomocí TrueDepth kamery, díky které lze vytvořit mesh obličeje a přidávat na něj digitální obsah. ARKit dokáže využívat technologie LiDAR, která je dostupná od iPhone 12. Ta umožňuje zmapovat místnost až do vzdálenosti 5 metrů v řádu jednotek sekund. Díky tomu je mapování okolního prostředí jednodušší, rychlejší a mnohem přesnější [24].

1.4.3 AR Foundation

AR Foundation je balíček, který obsahuje interface pro práci s AR, ale neimplementuje žádnou z jeho funkcí. Díky tomu je možné napsat jediný kód pro AR a na základě operačního systému zařízení se bude používat daná implementace. V projektu je třeba mít balíčky, které tyto funkce implementují. Unity oficiálně podporuje balíčky:

- Google ARCore XR plug-in pro Android.
- Apple ARKit XR plug-in pro iOS.
- OpenXR plug-in pro HoloLens 2.

Implementace jednotlivých funkcí nemusí být obsažena ve všech ze zmíněných balíčků [25]. Tabulka 1 nám ukazuje, jaké funkce implementují jednotlivé balíčky.

Tabulka 1: Funkcionality AR balíčků. Zdroj [25].

Funkce	ARCore	ARKit	OpenXR	Popis
Session	Ano	Ano	Ano	Povoluje, zakazuje, konfiguruje AR na cílové platformě.
Device tracking	Ano	Ano	Ano	Trackuje pozici a rotaci zařízení v reálném světě.
Camera	Ano	Ano		Vykreslování snímků z kamer zařízení a odhadování osvětlení.
Plane detection	Ano	Ano		Detekuje a trackuje plochy.
Image tracking	Ano	Ano		Detekuje a trackuje 2D obrázky.
Object tracking		Ano		Detekuje a trackuje 3D objekty.
Face tracking	Ano	Ano		Detekuje a trackuje obličej.
Body tracking		Ano		Detekuje a trackuje tělo.
Point clouds	Ano	Ano		Detekuje a trackuje významné body v okolí.
Raycasts	Ano	Ano		Pouští paprsky na trackované objekty.
Anchors	Ano	Ano	Ano	Sleduje libovolné body v prostoru.
Meshing		Ano	Ano	Generuje mesh okolí.
Environment probes	Ano	Ano		Generuje cubemapu okolního prostředí.
Occlusion	Ano	Ano		Překrývá digitální obsah fyzickými objekty a provádí segmentaci člověka.
Participants		Ano		Trackuje jiná zařízení pro sdílení AR relaci.

1.4.4 Vuforia

Vuforia je SDK pro vytváření aplikací s AR. Díky tomuto SDK je možné přidat do aplikací pokročilý proces zpracování obrazu, který umožňuje rozpoznávání obrázků, objektů a okolí. Tato technologie umožňuje vytvářet aplikace pro Android, iOS, Lumin a Universal Windows Platform zařízení [26].

SDK poskytuje funkce pro rozpoznání obrázků, čárových kódů nebo rozpoznání VuMarks, které představují značky pro ukládání dat nebo inicializování AR session. Oproti ARCore a ARKit umožňuje rozpoznat i cylindrické objekty jako jsou plechovky nebo hrnky. Vuforia také nabízí možnost ukládání cílů pro rozpoznání na cloudovou službu, a díky tomu šetří místo na zařízení.

Velkou předností tohoto SDK je možnost rozpoznání 3D modelů. SDK umožňuje rozpoznávat širokou škálu předmětů, pokud je 3D model dostatečně kvalitní. Mezi tyto předměty patří průmyslová zařízení, vozidla, hračky nebo domácí spotřebiče [27].

Pro vytvoření cílového modelu je možné využít program CAD, který je doporučený pro modely nářadí, lékařského vybavení nebo automobilů. Tento typ rozpoznávání je založený na rozpoznání tvaru, díky čemuž funguje i pokud objekt není zcela dobře texturovaný nebo se jeho povrch v určitých částech liší. Pokud není k dispozici model z programu CAD, je možné využít 3D skenování objektu u kterého je potřeba, aby byl objekt velmi dobře texturován, bez odlesků a bez matně tmavých ploch [28].

Mimo rozpoznání obrázků a 3D modelů umožňuje Vuforia také rozpoznání okolního prostředí. Pomocí přesného 3D skenu tohoto prostředí, je možné velmi jednoduše přidat digitální obsah na statické objekty ve svém okolí. Tato funkcionality je vhodná pro použití v muzeích, zajímavých veřejných místech, ale i v kancelářích nebo výrobních halách. Je potřeba aby skenování okolního prostředí bylo velmi přesné, proto je doporučeno použít jeden z podporovaných skenerů:

- Zařízení s LiDAR senzorem.
- Matterport Pro2 3D kamera.
- NavVis M6 a VLX skener.
- Leica BLK360 a RTC360 skener [29].



Obrázek 8: Zobrazení digitálního obsahu na statických prvcích. Zdroj [30].

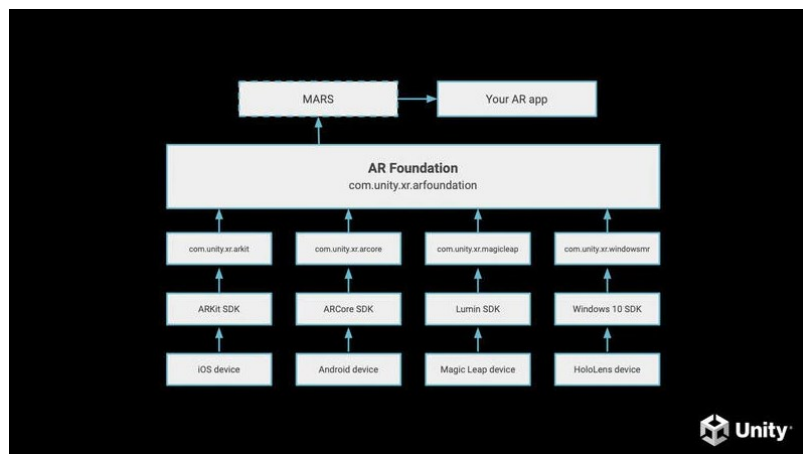
1.4.5 MARS

Mixed and Augmented Reality Studio neboli MARS přidává nové funkcionality pro tvorbu AR a MR. Toto rozšíření pro Unity umožňuje zpracovávat objekty reálného světa jako Unity objekty tzv. GameObject, a díky tomu na ně reagovat a interagovat s nimi. MARS také přináší nové ovládací prvky a uživatelské rozhraní oproti klasickému Unity. Jednou z velkých předností, kterou MARS nabízí, je režim Simulace. Tento režim poskytuje testování aplikací

v předem připraveném digitálním prostředí. Tyto aplikace interagují s digitálním okolím stejně, jako kdyby byla aplikace testována v reálném světě. Díky tomu lze testovat AR a MR funkce přímo v Unity editoru [31].

Toto rozšíření spolupracuje s AR Foundation, ze kterého využívá poskytovaná data a umožňuje vytvářet AR aplikace na více platformech. Díky této spolupráci je vytváření, testování a nasazování aplikací mnohem jednodušší [32].

MARS můžeme využívat v trial verzi po dobu 45 dní, poté zaplatíme za používání 600 \$ ročně [33].



Obrázek 9: Hierarchie ARCore, ARKit, AR Foundation a MARS. Zdroj [34].

1.5 Bezpečnost

Pro správné fungování potřebuje AR shromažďovat velké množství informací z kamery, o aktuální poloze a zároveň vidí, co uživatel dělá. Z toho důvodu je jednou z důležitých vlastností této technologie ochrana soukromí. Je třeba správně chránit svá data, aby k nim případný útočník neměl přístup a nemohl je zneužít.

Díky možnosti použít AR přímo ve webovém prohlížeči, se velmi usnadnil proces rozšiřování této technologie. Těmto webům však často dodává digitální obsah třetí strana, která může záměrně, nebo omylem, dodat škodlivý obsah. Tím může poškodit zařízení nebo otevřít zadní vrátka pro případného útočníka.

V dnešní době je také mnoho firem, které prodávají zboží a mají integrované AR ve své aplikaci. V téže aplikaci mají často přidanou platební kartu ve svém účtu, kterou by mohl útočník zneužít. Je potřeba zabezpečit veškerou komunikaci, například při stahování digitálního obsahu, a zároveň zaručit bezpečnost veškerého digitálního obsahu, který aplikace využívá.

Dalším možným útokem na aplikaci využívající AR je DOS (Denial of Service) útok. V jistých zaměstnáních je potřeba, aby AR neustále zpracovávalo přicházející tok dat. Pokud by AR přestalo zpracovávat data, přestalo by pracovat v reálném čase a škody by mohly být kritické. Jednat se může například o chirurgy, kteří mohou ztratit přístup k důležitým informacím na svých AR brýlích [35].

Mimo ohrožení na kybernetické úrovni, může AR přinášet i jiná rizika. Tato technologie může odvádět pozornost uživatelů od jejich aktuálního okolí. Pokud by uživatel využíval tuto technologii v automobilu, mohlo by jednoduše dojít k velkému riziku ve formě přehlédnutí chodce nebo srážky s jiným vozidlem. Stejným způsobem by chodec, využívající AR, mohl přehlédnout hrozby ve svém okolí [36].

Uživatelé AR nemusí být pouze obětí. Díky AR se může zvýšit bezpečnost na pracovišti například v bezpečnostní službě. Při použití AR brýlí má uživatel volné ruce, a v kombinaci s hlasovým ovládním může vidět a přepínat obraz mezi jednotlivými kamerami v celé budově, kterou hlídá. Zároveň může docházet k rychlé evakuaci, pokud by v brýlích byl integrován systém vnitřní navigace. AR brýle mohou také obraz odesílat v reálném čase na centrálu zabezpečení, která by díky těmto záběrům přímo z pohledu uživatele mohla rozhodovat, jaký zvolit další postup při evakuaci [37].

2 ENGINEY AR

2.1 Unity

Unity je vývojová platforma představena v roce 2005 pro tvorbu 2D a 3D aplikací pomocí programovacího jazyka C#. Můžeme zde vytvářet cross-platformní hry, aplikace i simulace pro více než 25 platform mezi které patří chytré telefony, počítače, konzole, televize i aplikace a hry pro VR a AR. Unity je zdarma a dostupné na operační systémy Windows a macOS [38].

Tato platforma poskytuje mnoho vestavěných funkcí, které jsou potřeba pro vývoj her a jejich správné fungování. Obsahuje funkce pro správné chování fyziky, 3D renderování nebo detekci kolizí. To usnadňuje práci vývojáře, jelikož tyto funkce nemusí sám implementovat. Značnou výhodou Unity je také Asset Store, který umožňuje vývojářům publikovat svou tvorbu. Díky tomu mohou usnadnit práci jiným vývojářům, kteří místo vytváření modelu ohně s částicovými efekty od nuly mohou model najít na Asset Storu a vložit do svého projektu [39].

Unity také nabízí velké množství animačních nástrojů pro 2D a 3D grafiku. Jednou z možností je importování animací z jiného programu jako je například Blender. Druhou možností, kterou Unity nabízí, je vytvoření vlastní animace přímo v tomto enginu. Díky tomu můžeme definovat pohyby, které bude postava ve hře využívat. Systém Animator nám umožní přehrávat a měnit jednotlivé animace na základě aktuálního stavu proměnných, kterými je hráč definovaný. Můžeme tak plynule přepínat například mezi animací chůze a skoku. Animator je zobrazen jako vizuální graf a umožňuje snadno pochopit souvislosti mezi jednotlivými animacemi [40].

Tento software je pro nezávislé vývojáře zdarma, pokud jejich roční výdělek z Unity aplikací nepřekročí 100,000 \$. Pokud chce vývojář vytvářet komerční hry, musí si zakoupit licenci Plus nebo Pro. Licence Plus může vývojářům pomoci se zlepšením výkonu aplikace, díky možnosti využití nových funkcí a tréninkových materiálů. Tato licence je vhodná pro nezávislé vývojáře, zatímco licenci Pro využívají herní studia a profesionální vývojářské týmy, které díky Unity aplikacím vydělají více než 200,000 \$ ročně [41].

Unity má i své nevýhody. V porovnání s Unreal Enginem nebo CryEnginem nejsou grafické schopnosti Unity na tak vysoké úrovni. Díky zaostávající grafice mohou dát přednost velká AAA herní studia jinému enginu. Méně efektivní je také osvětlení nebo vývoj 2D her, které lze snadněji vytvořit v Godotu nebo Gamemakeru [42].

Mezi nejznámější herní tituly, které byly vytvořeny pomocí Unity patří česká hra Beat Saber, světoznámá hra pro mobilní zařízení Pokemon GO nebo oblíbená karetní hra Hearthstone [43].

2.2 Unreal Engine

Unreal Engine je vývojový nástroj pro vytváření her, který vyvíjí společnost Epic Games. Vývojáři tento nástroj používají pro vytváření těch nejlepších a nejpobulárnějších her dnešní doby. Je možné ho také využívat pro vytváření simulací, editaci videí a zvuků nebo renderování animací. Tento nástroj se velmi rychle stal standardem pro AAA herní tituly. Unreal Engine byl celý napsán pomocí programovacího jazyka C++ a je možné díky němu vytvářet hry pro iOS, Android, Windows, PlayStation i Xbox.

Díky tomu, že lze v Unreal Engine vyvíjet hry pomocí jazyka C++, nabízí nástroj vysokou stabilitu a výbornou práci s pamětí. To je jeden z důvodů proč se tento nástroj hojně využívá. Tento jazyk je však považovaný za náročný na naučení, proto se doporučuje alespoň základní znalost tohoto jazyka před začátkem vývoje pomocí kódu [44].

Nicméně v tomto nástroji lze vytvořit hru i bez programování. Pomocí vizuálního editoru s názvem Blueprint, je možné vytvářet základní funkcionality hry, díky čemuž je tento nástroj vhodný i pro začátečníky [45]. Tyto Blueprints se skládají z posloupnosti uzlů, které slouží pro generování herních prvků. Díky tomu můžeme využívat celou řadu dostupných nástrojů pro vývojáře, a umožňuje nám hrát hry jednodušším způsobem a na vyšší vizuální úrovni. Tato metoda nenahrazuje kompletně programování v jazyce C++, pouze ho doplňuje, přičemž tyto dvě metodiky mezi sebou dokonale spolupracují. Metodika Blueprint umožňuje například vytváření kamer, předmětů nebo celých postav a také dokáže provádět interakci s nabídkou a prostředím [46].

Engine poskytuje framework pro práci s AR s názvem Unreal Engine AR. Tento framework poskytuje nástroje pro vývoj AR aplikací pro Android i iOS. Díky bohatému a jednotnému rámci tohoto frameworku, mohou vývojáři vyvíjet aplikace zároveň na obě tyto platformy. AR framework podporuje základní funkcionality pro AR jako jsou odhad okolního osvětlení, který následně pomáhá s realističností scény, aktuální stav relace nebo trackování. Unreal Engine podporuje i pokročilé funkcionality dostupné v ARCore a ARKit [47], které jsou uvedeny v Tabulka 2.

Tabulka 2: Funkcionality ARCore a ARKit v Unreal Engine AR. Zdroj [47].

ARCore	ARKit
Rozpoznání obličejů	Rozpoznání obličejů
Rozpoznání obrázků	Rozpoznání 2D obrázků a 3D objektů
Detekce vertikálních ploch	Motion capture
Cloud Anchors	Sdílení AR mezi více zařízení

Již v roce 1998 byla tímto nástrojem vytvořena první hra s názvem Unreal. V dnešní době mezi nejpopulárnější hry patří Fortnite, Rocket League, Batman Arkham nebo Gears of War [45].

3 AR ZAŘÍZENÍ

3.1 Mobilní zařízení

V dnešní době má téměř každý svůj chytrý telefon. Naprostá většina těchto telefonů je dostatečně vybavena pro práci s aplikacemi, které využívají AR. Díky tomu, že není potřeba dalšího zařízení, je AR v chytrých telefonech velmi rozšířená a oblíbená. Nicméně oproti chytrým brýlím je vždy omezen velikostí displeje. Navíc musí mít uživatelé vždy telefon v ruce, kdežto chytré brýle nebo headset nám nabízí větší volnost [48].

3.2 Headset, chytré brýle

AR headset je specializované zařízení na hlavu, které umožňuje uživateli vidět skutečný svět spolu s vloženým digitálním obsahem díky speciálním, transparentním LCD displejům. Headset umožňuje uživatelům procházet obsah ve formě virtuálních obrázků, videí, animací a dalších zdrojích informací. Toto zařízení umožňuje přidávat digitální obsah na základě toho, co uživatel vidí a kde se nachází. [49].

Mezi nejznámější AR headsety patří Microsoft HoloLens 2 nebo Apple Vision Pro, který byl zatím pouze představený a stále se čeká na jeho prodej [50].



Obrázek 10: AR headsety Microsoft HoloLens 2, Apple Vision Pro.
Zdroj [50].

Chytré brýle jsou oproti headsetu daleko lehčí a jednodušší na nošení, přibližně jako klasické brýle. Mají daleko méně funkcí, mezi které patří navigace, pořizování fotek a videí nebo čtení a překládání textu. Brýle dokážou i vybrané funkce chytrých telefonů jako zobrazení notifikací. Díky tomu, že mají brýle méně funkcí, jsou daleko levnější než AR headset. Cena se pohybuje v řádu stovek dolarů, kdežto AR headset se může vyšplhat až do řádů tisíců dolarů [51].

Mezi nejrozšířenější chytré brýle na trhu aktuálně patří Nreal Air, Ray-Ban Stories nebo Vue Lite 2 [52].



Obrázek 11: AR brýle Nreal Air. Zdroj [52].

3.3 AR kontaktní čočky

Dalším zařízením pro práci s AR jsou kontaktní čočky. Tato technologie zatím není tak pokročilá, jako AR v chytrém telefonu, nicméně společnost Mojo Vision již ukázala první prototyp, jak by tato technologie mohla pracovat. Kontaktní čočky obsahují jeden z nejmenších MicroLED displejů na světě, díky kterému můžeme vidět digitální obsah ve skutečném světě. Pro správné zobrazení tohoto obsahu využívají čočky také akcelerometr, gyroskop, magnetometr a ARM procesor. Uživatelské prostředí, které nám čočky nabídnou, lze zatím ovládat pouze pohybem očí. Tato technologie zatím není sériově vyráběna a prodávána, nicméně společnost Mojo Vision věří, že vyřešení všech potřebných záležitostí a zároveň úspěšné splnění klinických testů by mělo přijít do čtyř let [53].



Obrázek 12: AR kontaktní čočka společnosti Mojo Vision. Zdroj [53].

4 NÁVRH APLIKACE

V této práci bude navržena a vytvořena mobilní aplikace pro operační systém Android, která bude pracovat s AR a ukáže možnosti této technologie. Aplikace bude rozdělena do čtyř AR částí, přičemž každá z nich předvede jednu z funkcionalit, kterou ARCore nabízí.

V první části s názvem AR Marker je aplikace zaměřena na rozpoznání značky, kterou uživatel vybere přímo ze zařízení, a následně vloží vybraný model do scény. Další část aplikace tvoří AR Face, která umožňuje rozpoznání obličeje a přidání digitálního obsahu, kterým může být knír nebo brýle. V těchto dvou částech aplikace poskytuje možnost vytvoření konfigurace z aktuálních modelů ve scéně, která se následně uploaduje na server. Tuto konfiguraci lze později aplikovat zpět do scény, čímž se přidají všechny modely definované v konfiguraci s odpovídající pozicí, rotací a velikostí. Následující část s názvem AR Area umožňuje uživateli přidávat 3D modely do prostoru po rozpoznání horizontálních ploch. Poslední část pro práci s AR nese název AR Video. Tato část pracuje s rozpoznáním značky stejně jako AR Marker. Na rozdíl od AR Markeru však nevkládá 3D modely do scény, ale po rozpoznání značky se na ní začne přehrávat uživatelem definované video.

V aplikaci se také nachází část nevyužívající AR. Tato část s názvem Options umožňuje uživateli nastavit IP adresu serveru a zkontrolovat jeho dostupnost. Dále je možné stáhnout všechny 3D modely, konfigurace a videa ze serveru a uložit je do zařízení. Zároveň tato část umožňuje zobrazit modely uložené na serveru nebo přímo v zařízení, abychom věděli, jak budou vypadat ve scéně.

Mobilní aplikace rovněž funguje v offline režimu. K tomu je nejprve potřeba stáhnout všechny modely a videa do zařízení, poté je možné pracovat bez přístupu k internetu. Při pokusu o upload konfigurace v offline režimu se neuploaduje na server, ale dochází k lokálnímu uložení.

K mobilní aplikaci byla také vytvořena aplikace pro Windows, díky které můžeme nastavit velikost a rotaci jednotlivých modelů. Tato aplikace nebude probrána podrobně, jelikož se jedná pouze o doplněk k mobilní aplikaci.

4.1 3D modely

V mobilní aplikaci se digitální obsah do scény vkládá ve formě 3D modelů. 3D model je matematická reprezentace objektu ve třech dimenzích. Tyto modely se používají pro ztvárnění

objektů reálného světa a jejich využití najdeme v 3D tisku, VR a AR, počítačových hrách a dalších odvětvích [54].

3D model je tvořený vrcholy, které po propojení pomocí hran vytváří mesh. Tento mesh tvoří jádro celého modelu. Jednotlivé vrcholy mohou měnit svou pozici v prostoru a tím měnit celkový tvar 3D modelu. Nejčastěji se 3D model vytváří na základě primitivních tvarů, mezi které patří krychle, válec nebo koule [55].

Běžně se pro samotné modely vytváří také animace. V tom případě je nutné vytvořit 3D model velmi pečlivě, jelikož při pohybu modelu by mohlo dojít k neočekávaným stavům.

Modely mohou mít definovanou barvu, práci se světlem a další charakteristiky povrchu, pomocí programu zvaného shader. Bez použití shaderu je možné tyto charakteristiky definovat pomocí většího množství souborů ve formě 2D obrázků [54].

Ve vytvořené mobilní aplikaci se pracuje s 3D modely ve formátu GLB. GLB je zkratka pro GL Transmission Format Binary File a je vhodný pro práci s AR a VR. Formát byl vytvořen v roce 2015 jako binární formát k formátu glTF. GLB je binární formát, zatímco glTF je formát založený na textovém souboru typu JSON. Zároveň formát GLB je tvořený pouze jedním souborem, ve kterém jsou obsaženy veškeré komponenty. Mezi tyto komponenty patří textury, animace, materiál i celková hierarchie modelu. glTF je rovněž pouze jeden soubor, ale pro jeho správné fungování potřebuje i externí soubory. Typicky se jedná o soubory obsahující textury, ale mohou to být i shadery nebo animace [56].

Struktura glTF souboru:

- Složka modelu
 - Scene.bin
 - Scene.gltf
 - License.txt
 - Textures
 - material_0_diffuse.png
 - material_0_normal.png

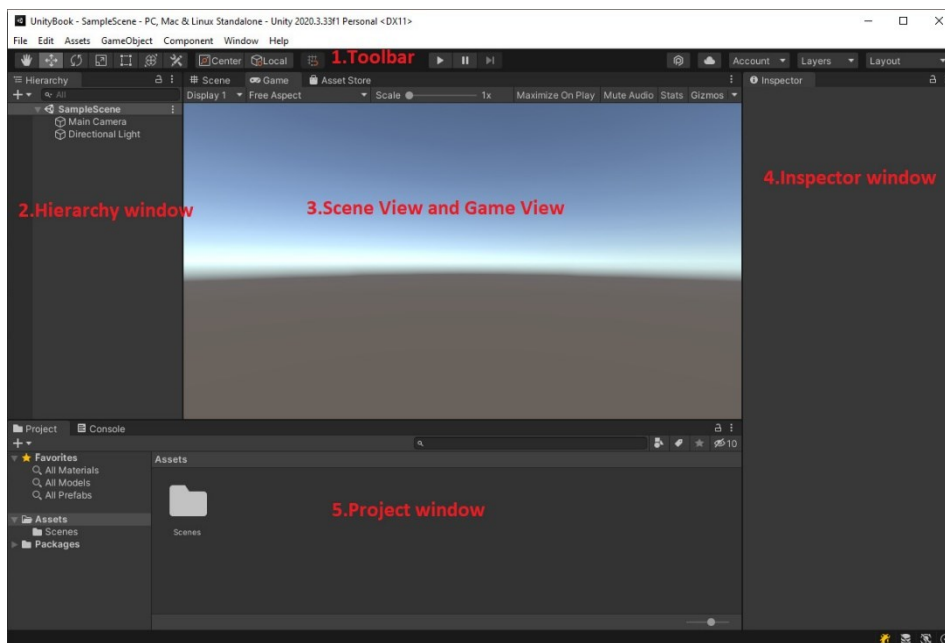
S 3D modely lze efektivně pracovat ve standardních, grafických programech 3ds Max, Maya nebo Blender [57]. V těchto programech je možné pracovat s širokou škálou formátů mezi které patří FBX, OBJ nebo STL [58]. Mimo programy k tomu určené lze pracovat s modely i v enginu Unity, který je využíván v této práci.

4.2 Použité technologie

4.2.1 Unity

Z teoretického hlediska bylo Unity popsáno v kapitole 2.1 Unity. V této části bude popsán praktický pohled na tento engine spolu s Unity editorem.

Při otevření projektu v Unity můžeme vidět obrazovku rozdělenou na 5 hlavních částí.



Obrázek 13: Rozhraní Unity editoru. Zdroj [59].

1. Panel nástrojů – Panel nástrojů obsahuje několik skupin ovládacích prvků. Je zde sada transformačních nástrojů pro pohyb ve scéně, rotaci objektů nebo změnu měřítek jednotlivých objektů. Dále se zde nachází trojice tlačítek pro spuštění projektu, jeho pozastavení a provedení dalšího kroku. Na konci řádku se nachází část pro verzování projektu.
2. Okno hierarchie – V této části se nachází všechny objekty, které se aktuálně nachází ve scéně. Můžeme zde najít objekty pro zobrazení světla, kamery, AR session nebo klasické modely pro zobrazení postav ve scéně.
3. Režim Scéna a režim Hra – V režimu Scéna můžeme manipulovat s jednotlivými objekty v prostoru. Je zde také vidět grid, díky kterému se můžeme lépe orientovat. Mezitím nám režim Hra přináší pohled z vytvořené kamery ve scéně, díky které vidíme zobrazení scény tak, jak bude vypadat v konečné aplikaci. Zároveň můžeme v tomto režimu měnit poměry stran, což je výhodné při vytváření aplikací pro zařízení s různě velkými displeji.
4. Okno Inspektor – Každý objekt ve scéně může obsahovat komponenty. Tyto komponenty nám definují tento objekt a jeho chování. Typickou komponentou obsaženou v objektech je

Transform, která nám definuje aktuální polohu v prostoru. Této komponentě můžeme měnit její hodnoty a tím pohybovat objektem. Všechny komponenty objektu můžeme vidět v tomto okně.

5. Okno projektu – V Okně projektu můžeme prohlížet a organizovat všechny dostupné prostředky v projektu, přičemž vypadá velmi podobně jako klasický prohlížeč souborů. Mohou se zde nacházet skripty, modely nebo animace [59].

Při spuštění kódu v Unity se automaticky volají předem definované funkce, přičemž vybrané jsou v této práci využity. První, volanou, předdefinovanou funkcí je *Awake* a je volána jen pokud je objekt, na kterém je kód přidán ve formě komponenty, aktivní. Následuje funkce *OnEnable*, která se volá hned po aktivaci objektu. Dokončením všech *Awake* funkcí přijde na řadu funkce *Start*, která se provede před prvním snímkem scény. Po této inicializační části přichází metoda s názvem *Update*, volaná každý snímek. Do této části se často vkládá nejdůležitější kód pro správnou funkci scény. Při deaktivaci objektu se jako poslední volá funkce *OnDisable*. Mezi další funkce, které nebyly v práci využity, patří *OnGUI*, *OnDestroy* nebo *OnApplicationQuit* [60].

4.2.2 C#

C# je programovací jazyk, který vyvinula společnost Microsoft v roce 2000, a která ho i nadále udržuje. Tento jazyk byl původně navržen jako konkurence pro jazyk Java a podle PYPL indexu i TIOBE indexu si tyto dva jazyky drží velmi podobnou úroveň popularity mezi vývojáři. Můžeme říct, že Microsoft vytvořil velmi úspěšnou konkurenci pro Javu, a navíc je při porovnání s Javou mnohem snadnější na naučení.

C# je velmi často používán v kombinaci s Windows .NET frameworkem, ve kterém je možné využít objektově orientované programování. Tento jazyk byl vyvinut Microsoftem pro Microsoft, a díky tomu je jednou z nejlepších voleb při vytváření aplikací pro operační systém Windows.

Mimo aplikace přímo pro Windows, se tento jazyk hojně využívá při tvorbě her, jelikož s ním pracuje herní engine Unity. Také lze tento jazyk použít pro vývoj webových stránek, které díky němu mohou dosahovat vysoké škálovatelnosti, efektivnosti a snadné údržbě [61].

Mezi nejznámější aplikace napsané tímto jazykem patří Microsoft Office (Word, Excel atd.), Skype nebo Visual Studio [62].

4.2.3 Android

Android je mobilní operační systém, založený na operačním systému Linux a vlastněný společností Google. Tento operační systém lze využít jak v chytrých telefonech a tabletech, tak je možné najít jeho využití v chytrých televizích pod názvem Android TV. Dále ho můžeme najít v automobilech, a svou verzi má i chytrých hodinkách pod názvem WearOS [63].

Tento operační systém má podle GlobalStats statCounteru podíl na trhu přibližně 72 % (k 1. 1. 2023), což z něj dělá nejrozšířenější operační systém na trhu [64].

Pro tento operační systém existuje na Google Play Storu přes 2,6 milionu aplikací a zároveň do něj lze stahovat a instalovat aplikace, které zde nenajdeme. Díky této otevřenosti systému je velmi mocný a upravitelný pro osobní potřeby, ale zároveň je třeba dát si pozor na aplikace, které by mohly obsahovat viry nebo jiný škodlivý kód [63].

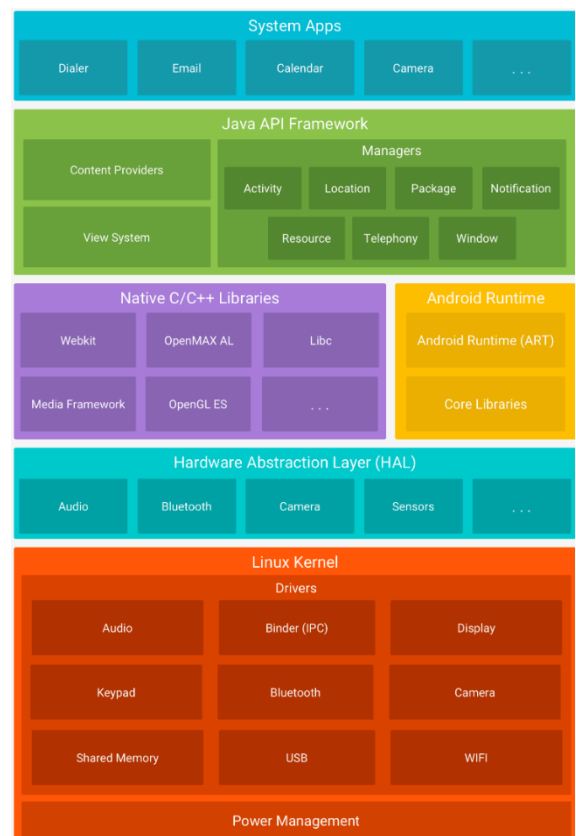
Architektura operačního systému Android je založena na pěti vrstvách, přičemž každá vrstva poskytuje odlišné funkce vrstvě nad sebou.

Linux Kernel

Linux Kernel je základem celé Android platformy [65]. Tato vrstva se nachází úplně ve spodu celé architektury a poskytuje základní funkce systému Android. Mezi tyto funkce patří nastavení zabezpečení, správa napájení, správa paměti a další ovladače hardwaru. S touto vrstvou nedochází k přímé interakci ze strany uživatele ani vývojáře [66].

Abstrakce hardwaru

Vrstva poskytuje standardní rozhraní, díky kterému může Java API framework přistupovat k hardwarovým možnostem zařízení. Vrstva se skládá z několika částí, kde každá část se stará o jednu hardwarovou komponentu. Mezi tyto komponenty patří kamera nebo Bluetooth [65].



Obrázek 14: Vrstvená architektura OS Android. Zdroj [65].

Vrstva knihoven

Tato vrstva obsahuje C/C++ knihovny a Java knihovny, které pomáhají při vývoji Android aplikací. V těchto knihovnách můžeme najít nástroje pro práci s OpenGL, SSL, Webkit ale i s médii [67].

Runtime vrstva

Vrstva poskytuje klíčovou komponentu Dalvik Virtual Machine pro správný běh operačního systému. DVM je speciálně navržený a přizpůsobený Java Virtual Machine, který je optimalizovaný pro operační systém Android. Také zajišťuje, aby každá aplikace běžela ve vlastním procesu s vlastní instancí DVM a využívá základní funkce Linuxu pro více vláknové zpracování a správu paměti. Díky knihovnám, které jsou v této vrstvě, mohou vývojáři vyvíjet Android aplikace pomocí jazyka Java [68].

Java API framework

Tato vrstva nabízí celou řadu funkcí, které jsou k dispozici pomocí Java API. Tato API usnadňují vývojářům vytváření Android aplikací, díky možnosti jednoduchého používání základních komponent a služeb. Mezi tyto služby patří:

- Manažer aktivit – Stará se o životní cyklus aplikací.
- Poskytovatel dat – Pomáhá aplikacím sdílet data online nebo je sdílet do jiných aplikací.
- Manažer notifikací – Umožňuje aplikacím zprostředkovávat notifikace do uživatelského prostředí [65].

Aplikační vrstva

Na této vrstvě jsou nainstalovány všechny aplikace. Může se jednat o předinstalované aplikace nebo nainstalované uživatelem. Tyto aplikace běží v Android prostředí a spolupracují s třídami a službami, které poskytuje vrstva Java API framework. S touto vrstvou typicky uživatel pracuje při běžné práci se zařízením, mezi kterou patří prohlížení webových stránek, provádění hovoru nebo prohlížení galerie [67].

4.2.4 AR Foundation, ARCore

Pro práci s AR je v mobilní aplikaci využito AR Foundation společně s ARCore. Tyto dvě technologie byly popsány v kapitolách 1.4.1 ARCore a 1.4.3 AR Foundation.

4.3 Analýza

4.3.1 Funkční požadavky

Funkční požadavky jsou funkce, které vývojář implementuje do aplikace. Tyto požadavky popisují možnosti chování celého systému za určitých podmínek. Je možné je vidět v konečné verzi daného systému nebo aplikace, a uživatel může tyto funkce provádět [69].

Tabulka 3: Funkční požadavky aplikace AR Magic. Zdroj vlastní.

ID	Požadavek
F1	Systém bude rozpoznávat horizontální plochy v okolí
F2	Systém bude rozpoznávat obličeje
F3	Systém bude rozpoznávat předem definované značky
F4	Systém bude schopný přidávat digitální obsah (modely, videa) na rozpoznané objekty (plochy, obličeje, značky)
F5	Systém bude schopný stahovat modely ze serveru
F6	Systém bude schopný provést upload konfigurace modelů na server
F7	Systém bude schopný manipulovat s modely (pohyb, rotace, změna velikosti)
F8	Systém bude schopný nastavit značky a videa přímo ze zařízení
F9	Systém bude schopný cachovat vybraná data
F10	Systém bude schopný nastavit adresu serveru
F11	Systém bude schopný otestovat připojení k serveru

4.3.2 Nefunkční požadavky

Nefunkční požadavky určují vlastnosti systému, které nesouvisí s jednotlivými funkcemi [70]. Určují nám kvalitativní faktory, které musí systém splňovat a byly domluveny před zahájením projektu. Tyto požadavky nejčastěji určují vývojáři, softwaroví architekti nebo technický teamleader [71].

Tabulka 4: Nefunkční požadavky aplikace AR Magic. Zdroj vlastní.

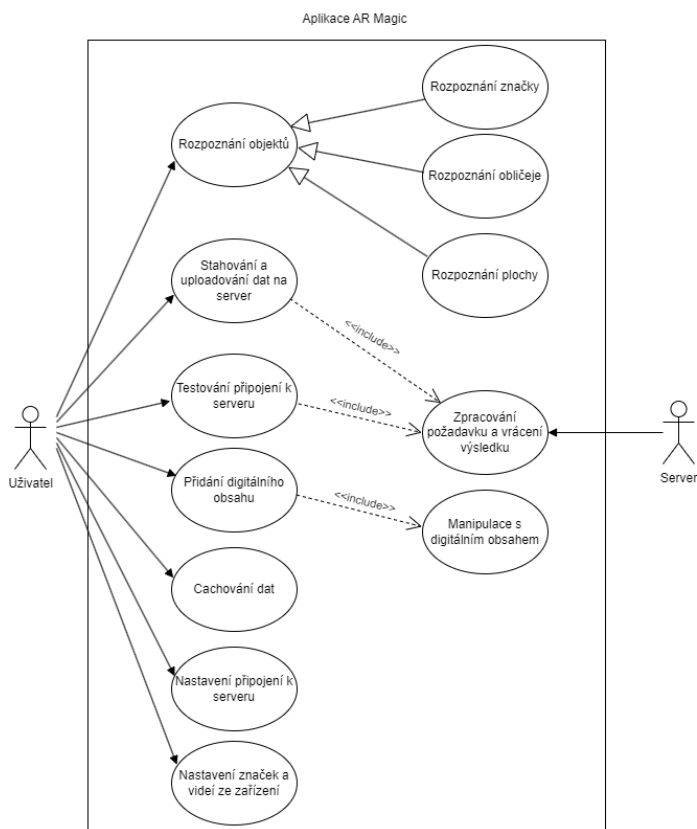
ID	Požadavek
N1	Systém bude naprogramován v programovacím jazyce C#.
N2	Systém bude vyvíjen pomocí engine Unity.
N3	Systém bude využívat modely typu GLB.
N4	Systém bude fungovat na zařízeních s OS Android 7.0 nebo vyšším, které podporují ARCore
N5	Systém bude využívat AR Foundation a ARCore pro práci s rozšířenou realitou.
N6	Systém bude schopný pracovat online i off-line

4.3.3 Use case diagram

Use case diagramy nám definují, jací aktéři se v systému nachází, jaké funkcionality bude systém schopný provádět a kteří aktéři k nim budou mít přístup.

Diagram se skládá ze tří částí:

- Use case – Use case neboli případ užití definuje jednu funkcionalitu systému, která obsahuje několik akcí. Tuto vnitřní logiku diagram nezachycuje.
- Aktéři – Akteřem může být uživatel nebo jiný systém, a definuje nám roli, která pracuje s jednotlivými případy užití.
- Vazby mezi objekty
 - Asociace – Spojení aktéra a jednoho use casu. Znáznorňuje, které funkcionality může aktér provádět.
 - Include – Spojení dvou use casů. Po spuštění prvního use casu se automaticky spustí i use case, který je připojený touto vazbou.
 - Generalizace – Spojení dvou aktérů nebo use casů. Používáme se pro znázornění dědičnosti v diagramu [72].
 - Extend – Spojení dvou use casů. Jedná se o rozšíření use casu, přičemž připojený use case lze provést pouze po splnění určité podmínky. Zároveň je závislý na use casu, ke kterému je připojen [73].



Obrázek 15: Use case diagram aplikace AR Magic. Zdroj vlastní.

4.3.4 Use case specifikace

Use case specifikace nám přináší detailnější pohled na jednotlivé use casey. Tato specifikace nemá žádnou pevně definovanou strukturu ani povinné části. Může být ve formě tabulky nebo obyčejného textu a nejčastěji se skládá z následujících částí:

- Název.
- Krátký popis.
- Aktéři.
- Podmínky pro spuštění.
- Základní tok.
- Alternativní toky.
- Podmínky pro ukončení [74].

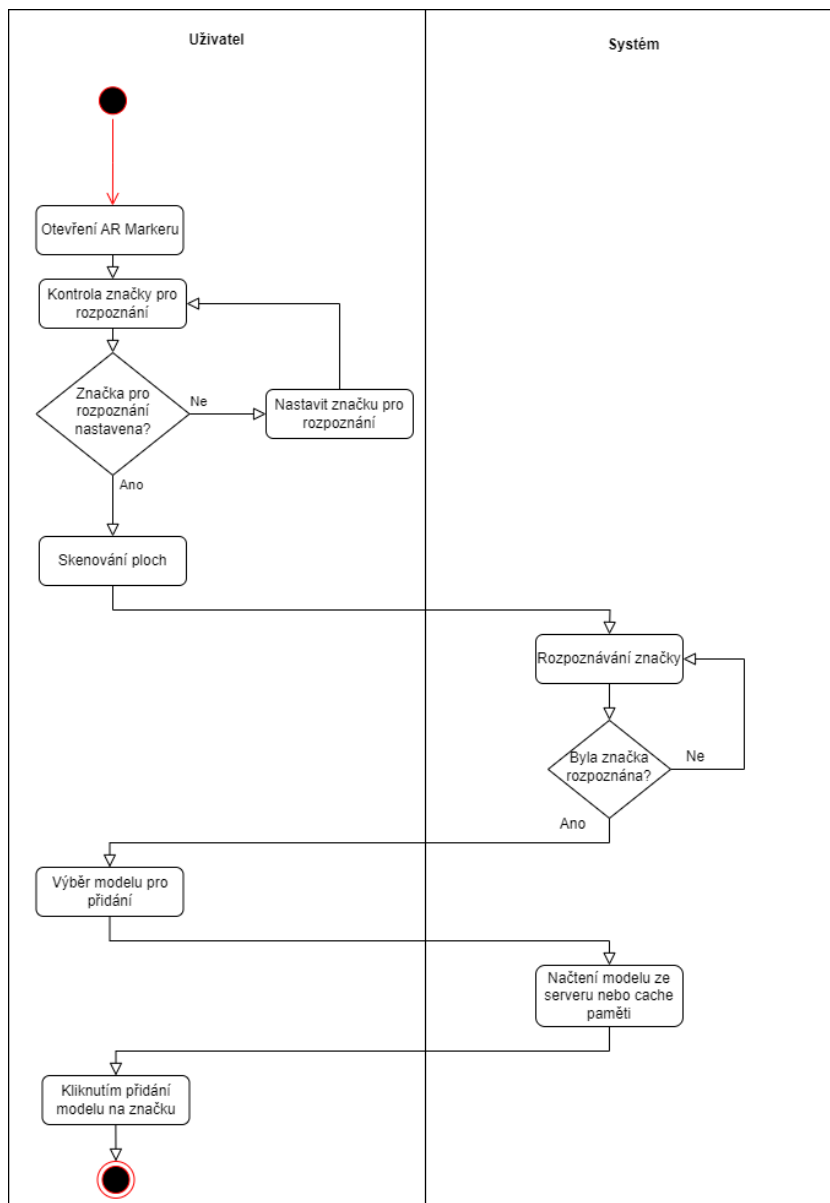
Tabulka 5: Use case specifikace – Přidání digitálního obsahu do scény. Zdroj vlastní.

Název	Přidání digitálního obsahu
Krátký popis	Use case umožňuje přidat digitální obsah (model) do scény
Aktéři	Uživatel, systém, server
Podmínky pro spuštění	1. Uživatel se nachází v části aplikace pro přidání digitálního obsahu (AR Marker, AR Face, AR Area)
Základní tok	1. Uživatel z menu vybere model pro vložení 2. Systém model připraví stažením ze serveru nebo načtením z cache paměti 3. Uživatel naskenuje okolní plochy 4. Systém tyto plochy zpracuje a informuje uživatele o možnosti vložení 5. Uživatel kliknutím na plochu přidá digitální obsah do prostoru
Podmínky pro ukončení	Digitální obsah je vložen do prostoru a připraven pro další manipulaci

4.3.5 Diagram aktivit

Diagram aktivit představuje dynamický diagram, ve kterém se pohybuje časová osa odshora dolů. Pomáhá nám lépe pochopit, jaké procesy v aplikaci existují, jak pracují a jak spolupracují

s ostatními. V těchto diagramech můžeme najít řadu odlišných prvků, mezi které patří počáteční a koncový stav, prováděná aktivita nebo rozhodovací prvek [75].



Obrázek 16: Diagram aktivit – přidání digitálního obsahu v části AR Marker.
Zdroj vlastní.

5 VÝVOJ APLIKACE

5.1 HTTP server

Pro tuto práci je potřeba vytvořit server, na kterém budou uložena data. Na serveru budou uloženy 3D modely, jednotlivé konfigurace těchto modelů a videa. Bude možné provádět pouze upload a download dat, z toho důvodu nám stačí vytvořit velmi jednoduchý server. Pro tyto účely byl vybrán balíček XAMPP, který obsahuje Apache HTTP server a umožňuje pracovat s MariaDB, PHP a Perlem [76].

Apache je open-source software, který je možné používat zdarma a běží na něm přibližně 40 % webových stránek po celém světě. Díky němu mohou vlastníci stránek tyto stránky zpřístupnit ostatním uživatelům. Tento software vyvinula a nadále i udržuje společnost Apache Software Foundation, která vydala první verzi již v roce 1995.

Apache je software, který běží na fyzickém serveru, a jeho hlavním cílem je vytvořit spojení mezi internetovým prohlížečem a fyzickým serverem. Pro tuto aplikaci je dostačující, aby se serverem stal počítač [77].

Po jednoduché instalaci balíčku XAMPP na počítač je nutné provést úpravy konfigurace, aby byl server přístupný i pomocí mobilního zařízení. Toho docílíme úpravou konfiguračního souboru httpd.conf, který lze otevřít pomocí tlačítka Config. Pro přístup k serveru pomocí mobilního zařízení je potřeba, aby počítač, na kterém je server zprostředkován, byl na stejné síti Wifi jako mobilní zařízení.

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
# Listen 12.34.56.78:80

Listen 80
Listen 192.168.107.43:8000
```

Zdrojový kód 1: Nastavení konfiguračního souboru httpd.conf pro Apache server.

Úpravu, kterou vidíme ve Zdrojovém kódu 1, je vhodné vložit pod zobrazený komentář, pod který je vložen řádek s červeným označením. Tento řádek nám zpřístupňuje server na dané IP adrese a portu. Tato IP adresa musí být shodná s IP adresou počítače, a později ji nastavíme i v mobilní aplikaci. IP adresu počítače lze zjistit po otevření Příkazového řádku pomocí příkazu

ipconfig, přičemž z výsledku vykopírujeme hodnotu řádku IPv4 Address. Výsledek tohoto příkazu vidíme ve Zdrojovém kódu 2.

```
C:\Users\Petr>ipconfig

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix. : 
    IPv6 Address. . . . . : fdf3:e82b:eea4:561c:912d:d393:d6ec:77a
    Temporary IPv6 Address. . . . . : fdf3:e82b:eea4:561c:a440:e5d:b211:aba6
    Link-local IPv6 Address . . . . . : fe80::6c84:c654:85d9:5c18%7
    IPv4 Address. . . . . : 192.168.107.43
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.107.1
```

Zdrojový kód 2: Výpis příkazu *ipconfig* v Příkazovém řádku.

Po tomto nastavení a spuštění serveru, pomocí tlačítka Start na řádku Apache, je vhodné vyzkoušet, zda server běží. Přístup k serveru můžeme ověřit pomocí počítače, kdy otevřeme webový prohlížeč a vložíme IP adresu s portem. Stejně lze přístup ověřit i pomocí mobilního zařízení.

Po správném nastavení IP adresy a spuštění serveru, potřebujeme pro práci s aplikací vytvořit na serveru složku, kam budeme ukládat data. Pokud proběhla instalace bez vlastní konfigurace, přesuneme se do složky C:\xampp\htdocs, ve které vytvoříme složku pro naši aplikaci, například s názvem Unity. V této složce je nutné vytvořit složky s přesnými názvy, které odpovídají kategoriím v aplikaci.

Názvy složek ve složce Unity:

- area_objects
- face_configurations
- face_objects
- marker_objects
- marker_objects_configurations
- videos

Tato struktura je připravena na přiloženém CD ve složce s názvem *Struktura_server*. V té jsou také připraveny modely, konfigurace a videa, které je možné v aplikaci využít.

Zároveň je potřeba, aby složka Unity obsahovala soubor upload.php, který přijímá a zpracovává požadavky při uploadu dat.

```

<?php
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        if(isset($_POST['ConfigurationName'])
            && isset($_POST['ConfigurationContent'])
            && isset($_POST['folder'])) {
            //upload konfigurace
            $myfile = fopen('./'.$_POST['folder'].'/'.
                $_POST['ConfigurationName'].'.json', "w")
                or die("Unable to open file!");
            fwrite($myfile, $_POST['ConfigurationContent']);
            fclose($myfile);
        }else if(isset($_FILES['model']) && isset($_POST['folder'])) {
            //upload modelu
            $tmp_name = $_FILES['model']["tmp_name"];
            move_uploaded_file($tmp_name, $_POST['folder'].
                $_FILES['model']["name"]);
        }else {
            echo('Fields not set!');
        }
    }
}
?>

```

Zdrojový kód 3: Kód pro zpracování požadavku na serveru v souboru upload.php.

V souboru upload.php, který vidíme ve Zdrojovém kódu 3, se při zpracování požadavku typu POST určí, zda se jedná o upload konfigurace nebo modelu, a dle toho se uloží text ve formátu JSON nebo binární soubor ve formátu GLB.

5.2 Import GLB modelů

5.2.1 Siccity.GLTFUtility

Pro práci s 3D modely ve formátu GLB je potřeba tyto modely importovat do aplikace. Jedním ze způsobů, jak toho docílit je využití projektu z Githubu od autora Siccity s názvem *GLTFUtility*. Tento projekt se zaměřuje na snadné použití a jednoduchost, a díky tomu je velice jednoduché importovat modely do aplikace pomocí jednoho řádku kódu.

```

GameObject model = Importer
    .LoadFromFile(Path.Combine(Application.persistentDataPath, folder, name));

```

Zdrojový kód 4: Ukázka kódu pro importování 3D modelu pomocí projektu *GLTFUtility*.

Projekt dále nabízí importování modelů asynchronním způsobem nebo přímo z pole bytů, a do budoucna by měl dokázat i exportování GLB modelů.

Projekt také zvládne pracovat s řadou komponent, díky kterým lze převést model formátu GLB na GameObject, s kterým můžeme pracovat v Unity. Mezi tyto komponenty patří práce se statickým meshem, normálami, tangentami, barvami vrcholů, materiály, texturami nebo kamerami.

Pro správné fungování je nutné vložit vybrané shadery do určité části v projektu, díky čemuž budou vždy součástí vytvořeného buildu. Shadery přidáme do části

Project Settings/Graphics/Always Included Shaders. Potřebné shadery pro vložení nalezneme v části *GLTFUtility/Materials/Built-in*. Jedná se o *Standard-Metallic-Blend*, *Standard-Metallic*, *Standard-Specular-Blend* a *Standard-Specular*. Do této části je také potřeba vložit shader z balíčku *glTFast* s názvem *glTFPbrMetallicRoughness* [78].

5.2.2 atteneder.glTFast

glTFast byla první z možností, která byla testována pro importování GLB modelů. Tento projekt se zaměřuje především na rychlost, efektivní práci s pamětí a malé nároky při tvorbě buildu. Spolu s tím nabízí snadné používání, robustnost, stabilitu a rozšiřitelnost pro pokročilé uživatele. Projekt by byl velmi výhodný pro naši aplikaci, jelikož umožňuje stahovat modely přímo z webu a také dokáže modely exportovat. Při testování na operačním systému Windows se tento projekt zdál jako ideální volba, bohužel po testování na operačním systému Android bylo načtení GLB modelů chybné. Díky ostatním uživatelům používajícím tento projekt bylo zjištěno, že tento problém přetrvává již nějakou dobu. Z toho důvodu bylo od tohoto projektu upuštěno a byl vybrán projekt *Siccity.GLTFUtility* [79].

5.3 Export GLB modelů

V mobilní aplikaci probíhá export modelů do formátu GLB velice jednoduše. Vybraný model se exportuje do mobilního zařízení po stažení ze serveru. Díky tomu, že exportujeme stažená data, máme data připravena v binární formě a stačí provést klasický zápis do binárního souboru.

```
File.WriteAllBytes(Path.Combine(Application.persistentDataPath, folder,
Path.GetFileName(name)),
webRequest.downloadHandler.data);
```

Zdrojový kód 5: Zápis stažených dat do zařízení.

V aplikaci pro Windows se neexportují stažená data, ale je potřeba vyexportovat model GLB z *GameObjectu*. K tomu slouží projekt *UnityGLTF* jehož autorem je *prefrontalcortex*. Tento projekt je založený na projektu od *KhronosGroup*, kteří jsou autorem 3D formátu GLB a *glTF*, a tento projekt přestali v roce 2020 udržovat. Pro exportování je potřeba nastavit jaký *GameObject* budeme exportovat, cestu pro export a název modelu [80].

```
GLTFSceneExporter exporter = new GLTFSceneExporter(go.transform,
new ExportOptions());
exporter.SaveGLB(path[0], modelNewName.text);
```

Zdrojový kód 6: Ukázka kódu pro exportování 3D modelu pomocí projektu *UnityGLTF*.

5.4 Struktura aplikace AR Magic

Jak bylo zmíněno v předchozí kapitole, aplikace obsahuje čtyři části pro práci s AR a Options. V této kapitole budou představeny jednotlivé části a detailně popsány vybrané funkce.

Pro každou část existuje v projektu vlastní složka, ve které jsou uloženy soubory s kódem. Z těchto souborů s kódem budou vždy popsány nejdůležitější části, které se přímo týkají AR a práce s ní. Zároveň bude popsána spolupráce aplikace se serverem.

5.4.1 AR Marker

Pro správné fungování části AR Marker obsahuje projekt dva soubory. Jedná se o soubory s názvem *RecognitionMarker.cs* a *MarkerManager.cs*.

V souboru *RecognitionMarker.cs* probíhá rozpoznání značky a práce se samotnými modely. V první řadě je nutné definovat funkci, která se bude provádět při rozpoznání značky, její aktualizaci nebo při odstranění. Tato funkce s názvem *OnImageChanged* se definuje v předdefinované funkci *OnEnable* a obdobně se při opuštění scény musí odebrat ve funkci *OnDisable*.

```
public void OnEnable()
{
    ARTrackedImageManager = FindObjectOfType<ARTrackedImageManager>();
    ARTrackedImageManager.trackedImagesChanged += OnImageChanged;
}

public void OnDisable()
{
    ARTrackedImageManager.trackedImagesChanged -= OnImageChanged;
}
```

Zdrojový kód 7: Přidání a odebrání funkce při rozpoznání značky.

Ve funkci *OnImageChanged* procházíme nově rozpoznané značky, které se nachází ve struktuře *List* s názvem *args.added*. Podobně se prochází i *List* s názvem *args.updated*, ve kterém jsou uloženy všechny značky, které již byly rozpoznány. Mezi všemi již rozpoznávanými značkami je třeba vybrat aktuálně rozpoznanou značku pomocí její *TrackingState*. Při rozpoznání nové nebo již rozpoznané značky se volá funkce *SetModel*, která má parametrem tuto značku.

```
public void OnImageChanged(ARTrackedImagesChangedEventArgs args)
{
    foreach (ARTrackedImage trackedImage in args.added)
    {
        SetModel(trackedImage);
    }
    foreach (var trackedImage in args.updated)
    {
        if (TrackingState.Tracking == trackedImage.trackingState)
        {
            SetModel(trackedImage);
        }
    }
}
```

Zdrojový kód 8: Funkce prováděná po rozpoznání značky.

Funkce *SetModel* je primárně určená pro vkládání modelů z uživatelem vybrané konfigurace. Při vybrání nové konfigurace se tato konfigurace prochází a daný model se podle jména najde v Listu *ModelInstances*, který obsahuje již načtené modely. Následně se přidá do scény ve správné poloze, rotaci a velikosti. Zároveň se na značku přidá tříosí objekt pro snazší orientaci.

```
private void SetModel(ARTrackedImage image)
{
    actualImage = image;
    if (AreNewModels)
    {
        AreNewModels = false;
        //Procházení všech modelů v konfiguraci
        foreach (ModelConfiguration modelConfig in ConfigurationsInScene)
        {
            GameObject model = ModelInstances.models
                .Find(item => item.name == modelConfig.name);

            if (model != null)
            {
                //Přidání modelu do scény
                GameObject go = Instantiate(model, actualImage.transform);
                go.transform.GetChild(0).gameObject.tag =
                    EnumModelType.MarkerObject.ToString();

                go.transform.localPosition = modelConfig.position;
                go.transform.localRotation = modelConfig.rotation;
                go.transform.localScale = modelConfig.scale;
                go.SetActive(true);
                go.transform.GetChild(0).gameObject.SetActive(true);
            }
            else
            {
                Debug.Log("Model for config NOT FOUND");
            }
        }
    }
    if (TriAxisInstantiated == null)
    {
        //Přidání tříosého objektu do scény
        TriAxisInstantiated = Instantiate(triAxis,
            image.transform.position,
            image.transform.rotation,
            image.transform);
    }
}
```

Zdrojový kód 9: Přidání modelů konfigurace a tříosého objektu po rozpoznání značky.

V tomto souboru je také velmi důležitá funkce *Update*, kterou lze rozdělit do dvou částí.

V první části, kterou můžeme vidět ve Zdrojovém kódu 10, probíhá přidání modelu do scény, po kliknutí prstem. Zde se nejprve provádí kontrola, zda uživatel vybral model pro vložení a zda bylo kliknuto na rozpoznanou plochu. Dále se model získá z *ModelInstances* na základě jména, vloží se do scény a nastaví se mu pozice.

```

if (IsTouchPhaseBegan() &&
    PlayerPrefs.GetString("NameOfEdisonModel").Length != 0)
{
    if (!IsPointerOverUIObject(Input.GetTouch(0).position))
    {
        if (aRRaycastManager.Raycast(Input.GetTouch(0).position,
                                      hits,
                                      TrackableType.PlaneWithinPolygon))
        {
            //Získání modelu
            GameObject prefab = ModelInstances.models
                .Find(item =>
                    item.name == PlayerPrefs.GetString("NameOfEdisonModel"));

            if (prefab != null)
            {
                //Přidání modelu do scény
                GameObject go = Instantiate(prefab, actualImage.transform);
                go.SetActive(true);
                go.transform.GetChild(0).gameObject.SetActive(true);
                go.transform.position = new Vector3(
                    go.transform.position.x,
                    actualImage.transform.position.y,
                    go.transform.position.z);

                go.transform.GetChild(0).gameObject.tag =
                    EnumModelType.MarkerObject.ToString();
            }
            else
            {
                Debug.Log("GameObject not found");
            }
        }
    }
}
}

```

Zdrojový kód 10: Vložení modelu při kliknutí prstem na rozpoznanou plochu

Ve druhé části se nachází kód, který nám umožňuje odstraňovat modely nebo měnit jejich polohu. Je rozdělen do tří částí definující začátek doteku, pohyb prstem a ukončení doteku. Při začátku doteku se model odstraní, pokud je nastavena proměnná *DeletingObjects*, která zde reprezentuje zaškrtačací políčko *Deleting objects*. Pokud se nejedná o odstranění modelu, tento model se přiřadí do proměnné *prefab*, kterou používáme v dalších částech. Ve druhé části vybraný model mění svou polohu na základě pohybu prstem. Při zvednutí prstu z displeje zařízení se dotek ukončí a proměnná se nastaví na hodnotu *null*. Tento proces můžeme vidět ve Zdrojovém kódu 11. Pro změnu měřítka a rotaci objektů je v práci využit balíček Lean Touch, který můžeme nalézt v Asset Storu.

```

if (IsTouchPhaseBegan() && prefab == null)
{
    //Začátek doteku
    if (Physics.Raycast(ray, out RaycastHit hit))
    {
        if (hit.collider.CompareTag(EnumModelType.MarkerObject.ToString()))
        {
            GameObject touchedObject = hit.collider.gameObject.transform
                .parent.gameObject;

            if (DeletingObjects)
            {
                //Odstranění modelu
                Destroy(touchedObject);
            }
            else
            {
                //Nastavení aktuálně vybraného modelu
                if (ActualPrefab != null)
                {
                    utilityApp.SetColorToModel(ActualPrefab, true);
                }
                utilityApp.SetColorToModel(hit.collider.gameObject, false);
                ActualPrefab = touchedObject;
                prefab = touchedObject;
                prefab.GetComponent<Lean.Touch.LeanTwistRotateAxis>()
                    .Axis = RotationOfPrefab;
            }
        }
        else
        {
            Debug.Log("Tag not found");
        }
    }
}
else if (HasTouchPhaseMoved() && prefab != null)
{
    //Pohyb prstu
    //Změna polohy modelu při pohybu prstu
    prefab.transform.position = new Vector3(
        hits[0].pose.position.x,
        actualImage.transform.position.y,
        hits[0].pose.position.z);
}
else if (HasTouchPhaseEnded())
{
    //Ukončení doteku
    prefab = null;
}
}

```

Zdrojový kód 11: Pohyb modelu po scéně pomocí pohybu prstem

Druhý soubor s názvem *MarkerManager.cs* se stará o funkce v menu. Mezi nejdůležitější funkce patří *OnClickButtonToChooseModel*, která se volá po vybrání modelu ze seznamu. Pokud uživatel kliknul na jeden z modelů, zjistí se, zda tento model již není připravený v *ModelInstances*. V případě že model chybí, stáhne ze serveru nebo načte z cache paměti pomocí metody *utilityApp.GetModel* a přidá se do *ModelInstances*. Na závěr se název

vybraného modelu uloží do PlayerPrefs s názvem NameOfMarkerModel, který se později použije při vkládání modelu. Ukázku této funkce můžeme vidět ve Zdrojovém kódu 12.

```
void OnClickButtonToChooseModel(string modelName)
{
    if (PlayerPrefs.GetString("NameOfMarkerModel").Equals(modelName))
    {
        PlayerPrefs.SetString("NameOfMarkerModel", "");
        return;
    }
    utilityApp.ClearButtonsColor(contentConfigurations);
    if (!ModelInstances.models.Exists(item => item.name == modelName))
    {
        GameObject progressCirclePrefab = Resources
            .Load("ProgressCircleBar") as GameObject;

        GameObject progressCircle = Instantiate(progressCirclePrefab,
            contentModels.parent.parent.transform);

        progressCircle.GetComponent<ProgressCircleBar>()
            .StartProgressBar();

        //Získání modelu ze serveru nebo ze zařízení
        utilityApp.GetModel(EnumFolders.marker_objects.ToString(),
            modelName, go =>
        {
            if (go != null)
            {
                //Přidání modelu do ModelInstances pro pozdější využití
                ModelInstances.AddModel(go, modelName);
            }
            else
            {
                Debug.Log("Model NOT FOUND");
            }
            progressCircle.GetComponent<ProgressCircleBar>()
                .StopProgressBar();

            Destroy(progressCircle);
        });
    }
    PlayerPrefs.SetString("NameOfMarkerModel", modelName);
}
```

Zdrojový kód 12: Načtení 3D modelu po jeho vybrání.

Pro rozpoznání značky je potřeba ji nejprve přidat do knihovny. Tato knihovna obsahuje všechny značky, které budeme chtít rozpoznat, a v našem případě bude vždy obsahovat pouze jednu. Knihovnu vytvoříme znovu, čímž vymažeme všechny aktuální značky. Tuto knihovnu nastavíme do *ImageManager* jako referenční a následně přetypujeme na *MutableRuntimeReferenceImageLibrary*. Do nově vytvořené knihovny se přidá vybraná značka a nastaví se hodnoty do proměnných. Přidání značky do knihovny můžeme vidět ve Zdrojovém kódu 13.


```

private void AddImageToLibrary(Texture2D image, float width)
{
    m_TrackedImageManager = FindObjectOfType<ARTrackedImageManager>();
    library = m_TrackedImageManager.CreateRuntimeLibrary();
    m_TrackedImageManager.referenceLibrary = library;
    if (library is MutableRuntimeReferenceImageLibrary mutableLibrary)
    {
        //Přidání značky do knihovny
        var jobState = mutableLibrary
            .ScheduleAddImageWithValidationJob(image, image.name, width/100.0f);

        PlayerPrefs.SetString("RecognitionImage", image.name);
        PlayerPrefs.SetFloat("RecognitionImageWidth", width);
    }
}

```

Zdrojový kód 13: Přidání značky do knihovny značek.

5.4.2 AR Face

Tato část obsahuje stejně jako AR Marker dva soubory, kdy *ARFaceRecognition.cs* se stará o rozpoznání obličeje a přidání modelů na tento obličej, zatímco *FaceManager.cs* poskytuje funkce pro vybírání modelů nebo konfigurací na obličej.

V souboru *ARFaceRecognition.cs* se podobně jako v předchozí části musí definovat funkce, která bude reagovat na rozpoznání nového obličeje nebo ztrátu aktuálního obličeje ve scéně. Při rozpoznání obličeje se přidávají modely na obličej pomocí funkce *SetModel*, naopak při ztrátě obličeje se všechny modely ze scény odstraní.

```

public void OnFaceChanged(ARFacesChangedEventArgs args)
{
    //Rozpoznání obličeje
    foreach (var trackedFace in args.added)
    {
        SetModel(trackedFace);
    }
    //Odstranění obličeje
    foreach (var trackedFace in args.removed)
    {
        DeleteModels(trackedFace);
    }
}

```

Zdrojový kód 14: Funkce prováděná po rozpoznání obličeje.

V metodě *SetModel* se prochází list modelů, které mají být aktuálně na obličej a přidávají se do scény.

V souboru *FaceManager.cs* se při výběru modelu volá funkce *OnClickButtonToChooseFace*, která je prováděna obdobně jako v části AR Marker.

5.4.3 AR Area

I v této části se nachází dva soubory pro práci s AR a s menu. Kód v těchto souborech se velmi podobá AR Markeru s tím rozdílem, že modely nejsou potomky dané značky, ale jsou umístěny volně do prostoru. Z toho důvodu zde není potřeba kód této části představovat.

5.4.4 AR Video

Tato část se skládá ze dvou scén. V první scéně lze rozpoznat značky, přidat na ně video a toto video ovládat. Ve druhé scéně je možné nastavit videa, která se budou zobrazovat po rozpoznání přiřazené značky.

Pro správné fungování této části je potřeba celkem čtyř souborů. Jedná se o soubory *ImageRecognisionForVideo.cs*, *ImageVideoContainer.cs*, *MenuScriptVideos.cs* a *SetImageAndVideo.cs*.

První scéna pracuje se souborem *ImageRecognisionForVideo.cs*, který se stará o rozpoznání značky, následné získání odpovídajícího videa a jeho spuštění. Rozpoznání značky probíhá na stejném základu jako v části AR Marker.

Pro přidání videa na značku a jeho spuštění je potřeba přidat komponenty *VideoPlayer* a *AudioSource* a provést jejich správné nastavení.

```
videoPlayerComponent = prefab.AddComponent<VideoPlayer>();
audioSourceComponent = prefab.AddComponent<AudioSource>();

videoPlayerComponent.playOnAwake = false;
audioSourceComponent.playOnAwake = false;

videoPlayerComponent.source = VideoSource.Url;
videoPlayerComponent.audioOutputMode = VideoAudioOutputMode.AudioSource;

videoPlayerComponent.EnableAudioTrack(0, true);
videoPlayerComponent.SetTargetAudioSource(0, audioSourceComponent);
```

Zdrojový kód 15: Nastavení komponent *VideoPlayer* a *AudioSource*.

Po konfiguraci komponent se nastaví cesta k videu, která může vést k lokálně uloženému videu nebo videu na serveru. Texturu videa je potřeba přiřadit objektu, který bude vložen na značku. Poté zbývá komponenty pro video i zvuk spustit, přičemž tento proces můžeme vidět ve Zdrojovém kódu 16.

```

videoPlayerComponent.url = "192.168.107.13/unity/videos/video1.mp4";

videoPlayerComponent.Prepare();
while (!videoPlayerComponent.isPrepared)
{
    yield return null;
}

prefab.GetComponentInChildren<MeshRenderer>().material = new
Material(material);

//Nastavení textury do vkládaného objektu
prefab.GetComponentInChildren<MeshRenderer>().material.mainTexture =
videoPlayerComponent.texture;

//Spuštění videa a zvuku
videoPlayerComponent.Play();
audioSourceComponent.Play();

while (videoPlayerComponent.isPlaying)
{
    yield return null;
}

```

Zdrojový kód 16: Ukázka kódu pro přípravu videa, nastavení textury a spuštění videa.

Pro ovládání videa je kód uložen v souboru *MenuScriptVideos.cs*, díky kterému lze video pozastavit, spustit, přetočit a měnit jeho velikost. Ve Zdrojovém kódu 17 jsou ukázány konkrétní funkce, které to umožňují.

```

if (videoPlayer.isPlaying)
{
    videoPlayer.Pause();
}
else
{
    videoPlayer.Play();
}

videoPlayer.time -= rewindSeconds; //rewindSeconds = 5
videoPlayer.time += rewindSeconds;

```

Zdrojový kód 17: Ukázka funkcí pro ovládání videa.

Pro druhou scénu je stěžejní soubor *SetImageAndVideo.cs*, díky kterému můžeme vytvářet páry mezi videi a značkami. V tomto kódu se tyto páry přidávají do Listu, se kterým se poté pracuje při rozpoznání značky. Tento List je také ukládán do zařízení ve formátu JSON.

```

string result = File.ReadAllText(Path.Combine(
    Application.persistentDataPath,
    EnumFolders.imagesVideosPairs.ToString() + ".json"));

videoImageContainers = JsonHelper.FromJson<ImageVideoContainer>(result);

```

Zdrojový kód 18: Uložení párů do zařízení ve formátu JSON.

List obsahující páry video-značka je tvořen objekty třídy *ImageVideoContainer.cs*.

```
public ImageVideoContainer(string video, string image, float width)
{
    VideoPath = video;
    ImagePath = image;
    ImageWidth = width;
}
```

Zdrojový kód 19: Třída pro ukládání párů značka-video.

5.4.5 UtilityApp

V aplikaci se nachází třída *UtilityApp*, která slouží k usnadnění práce se serverem a odstraňuje velké duplicity kódu. Tato třída se využívá ve všech funkčních scénách pro získání modelu, konfigurace nebo videa, uploadu konfigurací nebo ukládání modelů do cache paměti. Pro správné fungování této třídy, jsou zde využity i třídy *UtilityDownloader* pro stahování souborů a *UtilityUploader* pro uploadování konfigurací.

Pro stahování souborů ze serveru zde existuje funkce *DownloadFile*. V této funkci se vytvoří *UnityWebRequest*, jemuž se přiřadí URL adresa souboru pro stažení a určí se typ HTTP požadavku na GET. Tento požadavek se odešle a po úspěšném stažení se soubor zpracuje jako model, konfigurace nebo video.

```
public IEnumerator DownloadFile(string folder,
                                string name,
                                Action<UnityWebRequest> callback)
{
    using (UnityWebRequest webRequest = UnityWebRequest
        .Get(Path.Combine(UnityAddress.GetURLToWebserver(),
            folder,
            Path.GetFileName(name)))
    {
        yield return webRequest.SendWebRequest();

        if (webRequest.result == UnityWebRequest.Result.ConnectionError ||
            webRequest.result == UnityWebRequest.Result.ProtocolError)
        {
            //Neúspěšné stažení souboru
            callback(null);
        }
        else
        {
            //Úspěšné stažení souboru
            callback(webRequest);
        }
    }
}
```

Zdrojový kód 20: Ukázka kódu pro stažení dat ze serveru.

Uploadování konfigurace probíhá ve funkci *UploadConfiguration*, která se nachází ve třídě *UtilityUploader*. V této funkci se extrahuje z modelů název, pozice, rotace a měřítko. Tyto

hodnoty se vloží do Listu obsahující objekty *ModelConfiguration* a celý tento List se převede na JSON formát. Jeho obsah spolu s názvem konfigurace a umístěním na serveru se přidá k požadavku typu POST, který se následně odešle.

```
List<ModelConfiguration> configurations = new List<ModelConfiguration>();
foreach (GameObject obj in models)
{
    configurations.Add(new ModelConfiguration(
        obj.name.Replace(" (Clone) ", ""),
        obj.transform.localPosition,
        obj.transform.localRotation,
        obj.transform.localScale));
}

WWWForm wwwform = new WWWForm();
string jsonString = JsonHelper.ToJson(configurations);
wwwform.AddField("ConfigurationContent", jsonString);
wwwform.AddField("ConfigurationName", nameOfConfiguration);
wwwform.AddField("folder", folder);

UnityWebRequest www = UnityWebRequest
.Post(Path.Combine(UtilityAddress.GetURLToWebserver(),
    "upload.php"),
    wwwform);

StartCoroutine(SendRequest(www));
```

Zdrojový kód 21: Ukázka kódu pro upload konfigurace na server.

Třída *UtilityApp* také obsahuje funkce pro načtení značky ze zařízení, zjištění přítomnosti souboru na serveru nebo načtení názvů modelů ze serveru.

5.5 Uživatelská příručka

5.5.1 Aplikace Model Converter

Tato aplikace byla vytvořena jako doplňková pro AR Magic. Stejně jako AR Magic, byla aplikace vytvořena v prostředí Unity pomocí programovacího jazyka C#, ale není v něm využita technologie AR. V této aplikaci je možné importovat modely ve formátu glTF nebo GLB, které můžeme následně upravovat. Hlavním záměrem této aplikace je umožnit změnu velikosti modelu tak, aby byla přijatelná vůči velikosti značky v AR Markeru nebo měla odpovídající velikost při práci v AR Area. Kromě velikosti je zde také možné nastavit pozici i rotaci modelu, který umístíme na hlavu v rámci AR Face.

Správné velikosti modelu vůči značce docílíme nastavením velikosti referenčního čtverce na stejnou velikost jako má naše značka, přičemž velikost čtverce 1 odpovídá 1 metru ve skutečnosti. Pokud se jedná o model na obličej, je zde možné zobrazit referenční obličej, podle kterého můžeme nastavit pozici a rotaci.

Po nastavení všech potřebných parametrů můžeme model exportovat ve formátu GLB, který později nahrajeme na server. Je zde také možnost zadání IP adresy serveru, kdy po úspěšném navázání spojení bude svítit zelená tečka a můžeme exportovat model přímo na server do konkrétní složky.

V aplikaci můžeme pohybovat kamerou pomocí klávesnice a myši:

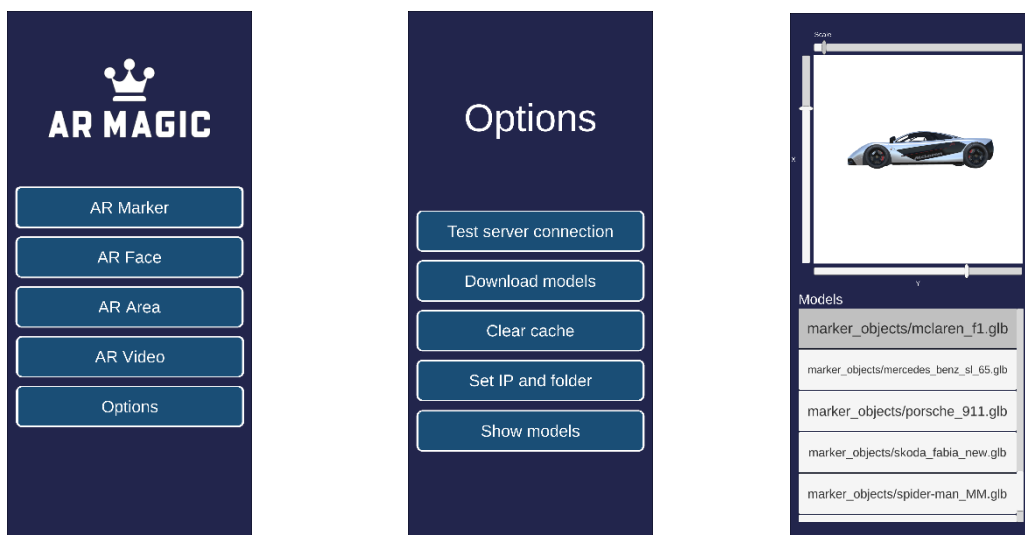
- Stisknuté kolečko myši + pohyb myši – pohyb kamery v prostoru.
- Stisknuté kolečko myši + pohyb myši + stisknutý levý shift – rotace kamery.
- Stisknutý levý shift + stisknuté F – reset kamery.
- Pohyb kolečka myši – přiblížení/oddálení kamery.



Obrázek 17: Vytvořená aplikace Model Converter. Zdroj vlastní.

5.5.2 Aplikace AR Magic

Scéna Menu tvoří úvodní scénu aplikace, která funguje jako rozcestník. V této scéně se nachází pět tlačítek, díky kterým se můžeme dostat do částí AR Marker, AR Face, AR Area, AR Video a Options. Ve scéně Options se podobně jako ve scéně Menu nachází pět tlačítek, pomocí kterých můžeme zjistit, zda je možné se připojit k serveru, případně mu nastavit IP adresu. Dále nám tato část umožňuje stáhnout všechny dostupné modely a videa ze serveru, a pro uvolnění místa můžeme všechna data v cache paměti smazat. Poslední tlačítko slouží k prohlížení všech dostupných modelů bez využití AR.



Obrázek 18: Scény Menu, Options a zobrazení modelů. Zdroj vlastní.

AR Marker

Scéna pro AR Marker obsahuje vysouvací menu z levé strany, díky kterému můžeme nastavit chování v této scéně.

Jako první se zde nacházejí dvě zaškrťovací políčka s názvy *Adding object* a *Deleting object*.

- *Adding object* – Toto políčko nám umožňuje vkládat modely na značku. Jeho zaškrtnutím se nám zpřístupní list *Models*, který obsahuje dostupné modely pro tuto část. Po výběru modelu a rozpoznání značky se model kliknutím přidá na rozpoznanou plochu.
- *Deleting object* – Díky tomuto políčku můžeme odebírat modely ze scény. Po zaškrtnutí políčka stačí kliknout na model ve scéně pro jeho odebrání.

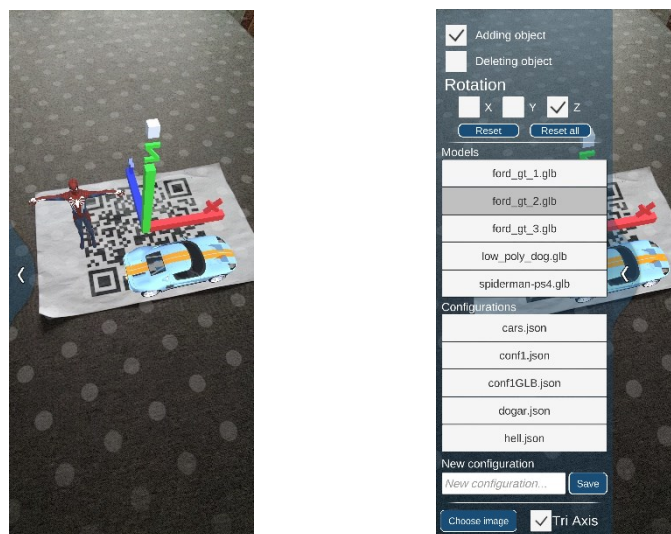
Pokud není vybráno ani jedno políčko, můžeme modelům pohybem prstů měnit polohu, rotaci nebo velikost.

V další části s názvem *Rotation*, lze vybrat osu, okolo které budeme model otáčet. Jsou zde i tlačítka pro vrácení modelu do původní rotace, kterou má uloženou na serveru nebo v cache paměti.

V listu *Models* jsou na výběr dostupné modely, které je možné přidat do scény. Tyto modely se nachází na serveru nebo v cache paměti. V podobně vypadající části s názvem *Configurations* se nachází všechny konfigurace, které má tato část k dispozici. Po stisknutí jedné z konfigurací se všechny aktuální modely ze scény odstraní, a přidají se modely definované v této konfiguraci. Konfiguraci lze vytvořit přidáním modelů do scény, nastavením jejich pozice,

rotace a měřítka, nastavením jména konfigurace a následně pomocí tlačítka Save konfiguraci uploadujeme na server. To nám ulehčí a urychlí vkládání více modelů do scény a zároveň již budou na předem definovaném místě se správnou rotací a velikostí.

Potřebujeme zde také nastavit značku, kterou bude naše zařízení rozpoznávat. Tuto značku lze nastavit pomocí tlačítka Choose image, po jehož stisknutí se otevře další okno. V tomto okně nastavíme značku, kterou vybereme přímo ze zařízení. Tato značka by měla být čtvercového tvaru nebo se tomu velmi blížit. Zároveň je potřeba nastavit její šířku ve skutečném světě pro správné fungování. Vedle tohoto tlačítka se nachází zaškrtnuté políčko, které nám umožňuje skrýt zobrazený tříosý objekt na značce.

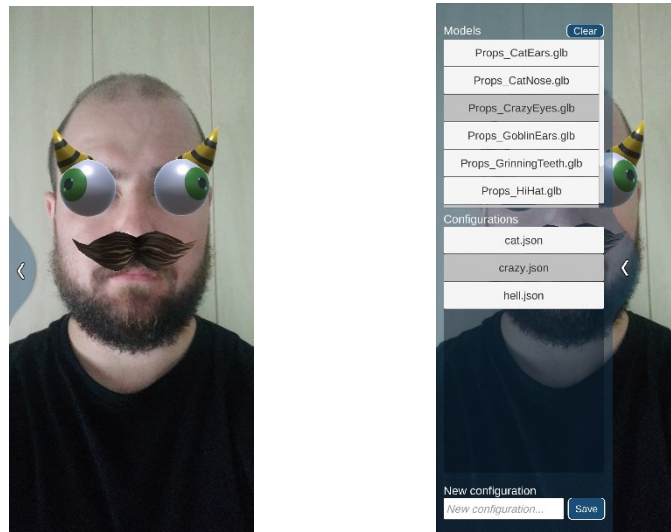


Obrázek 19: Zobrazení části AR Marker. Zdroj vlastní.

AR Face

V části AR Face je scéna zaměřena na obličej. Nejprve vybereme modely, které chceme zobrazit na obličejí nebo celou konfiguraci. Po výběru stačí přední kameru namířit na obličej a modely se zobrazí na předem definovaných místech.

Scéna AR Face obsahuje v menu dva listy, ve kterých vidíme všechny modely a konfigurace, část pro upload konfigurace a tlačítko pro smazání všech modelů ve scéně.



Obrázek 20: Zobrazení části AR Face. Zdroj vlastní.

AR Area

Část AR Area nepracuje s referenční značkou ani obličejem. Místo toho se modely volně vkládají do prostoru, kde musí být nejprve zmapovány horizontální plochy, na které se modely budou vkládat.

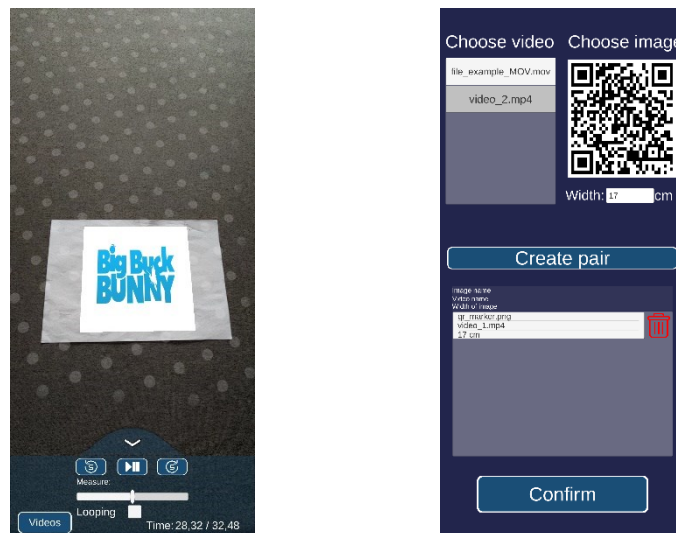
Nabídka menu je velmi podobná nabídce v AR Markeru, v této části ovšem chybí jakákoliv práce s konfiguracemi. AR Area nabízí přidávání, odebrání a manipulování s modely, a také vkládání tříosého objektu do prostoru, abychom případně věděli, jak budou modely rotovat.



Obrázek 21: Zobrazení části AR Area. Zdroj vlastní.

AR Video

Podobně jako v AR Markeru se v části AR Video pracuje s rozpoznanou značkou, na kterou se přidává digitální obsah. V tomto případě nejde o klasický model, ale o video uložené na serveru nebo v cache paměti. Tato část se skládá ze dvou scén, kdy v první scéně probíhá rozpoznání značky, spuštění videa a jeho ovládání. Ve druhé scéně se vytvářejí páry mezi značkou a videem, které na ní bude hrát.



Obrázek 22: Zobrazení části AR Video. Zdroj vlastní.

6 VYUŽITÍ APLIKACE

Mobilní aplikace AR Magic pro zařízení s operačním systémem Android, ukazuje možnosti využití rozšířené reality ve vnitřním prostředí. Tato aplikace se zaměřuje na zobrazení hlavních funkcionalit, které nabízí technologie ARCore, mezi které patří rozpoznání značky, rozpoznání obličeje a rozpoznání horizontálních ploch. Díky možnosti vkládání modelů na rozpoznané značky a horizontální plochy, je možné tuto aplikaci využít pro edukační účely. V této oblasti je aplikace vhodná pro základní školy, ve kterých mohou studenti detailně prozkoumávat modely tělesných orgánů. Pro studenty středních škol s technickým zaměřením by bylo možné aplikaci využít pro zobrazení potřebných součástí, což by bylo přínosnější než klasický 2D obrázek v knížce. Aplikaci může najít své využití i v muzeích, ve kterých by mohla fungovat jako interaktivní průvodce, který by po rozpoznání značky zobrazil poučné video nebo model s informacemi o vybraném exponátu.

Aplikace by nabízela větší množství využití po přidání vybraných funkcionalit. V budoucnu by bylo možné přidat animace k jednotlivým modelům, díky čemuž by mohlo dojít k výraznému zjednodušení při vysvětlování funkcí tělesných orgánů. Díky podpoře animací by bylo usnadněno také popsání historických událostí, kterými mohou být bitvy nebo jiné důležité události. V dalším vývoji bychom také mohli přidat možnost definování a následného zobrazení popisku u každého modelu.

Také by bylo možné směřovat AR Magic ke specifickému účelu. Nabízí se možnost využití spolu s výukovým robotem Edison, který dokáže jezdit po černé čáře nebo reagovat na světlo. Aplikaci by bylo možné rozšířit tak, aby se po dosažení cílové pozice objevil vybraný model, který by značil úspěšné dokončení úkolu robota. Tím by dětem bylo vizuálně potvrzeno úspěšné naprogramování robota a mohlo by to podpořit jejich další zájem o učení v této oblasti.

7 TESTOVÁNÍ

Mobilní aplikace AR Magic byla otestována na třech mobilních zařízeních. Pro testování byl vybrán LG G7 ThinQ s Android verzí 9, Mi MIX 2S s Android verzí 10 a Samsung Galaxy S21+ 5G s Android verzí 13. Každé z těchto zařízení disponuje odlišnou velikostí displeje, přičemž se aplikace těmto velikostem přizpůsobuje. Tím byla otestována správná změna velikostí jednotlivých objektů v podobě tlačítek nebo obrázků.

Na mobilní aplikaci bylo provedeno 6 scénářů, které otestovaly hlavní funkce jednotlivých částí. Každý scénář byl na jednotlivých zařízeních proveden 5x pro jistotu správnosti výsledku.

Scénář 1

Bylo provedeno otevření části AR Marker, nastavení referenční značky, její rozpoznání, stažení modelu ze serveru / načtení z cache paměti, přidání do scény a následné stažení konfigurace ze serveru / načtení z cache paměti a její nastavení do scény.

Scénář 2

Po přidání modelů v části AR Marker byla provedena jejich rotace po jednotlivých osách, reset všech rotací, pohyb modelů, změna měřítka a jejich následné odstranění.

Scénář 3

Bylo provedeno otevření části AR Face, stažení dvou modelů ze serveru / načtení z cache paměti, rozpoznání obličeje, přidání modelů na obličej a následné stažení konfigurace ze serveru / načtení z cache paměti a její nastavení na obličej.

Scénář 4

Po otevření AR Area bylo provedeno rozpoznání horizontálních ploch, stažení modelu ze serveru / načtení z cache paměti, jeho přidání do scény a provedení rotace, pohybu, změny měřítka a následné smazání.

Scénář 5

Po otevření AR Video byla otevřena scéna pro párování videa se značkou, kde byly nastaveny dva páry. Po potvrzení byla rozpoznána první značka, kde došlo k pozastavení, přetočení a změně měřítka. Po rozpoznání druhé značky se postup opakoval.

Scénář 6

Po otevření Options byla nastavena IP adresa serveru, otestováno připojení k serveru, následné stažení všech modelů ze serveru, kontrola existence těchto modelů, a nakonec byla odstraněna všechna data z cache paměti.

Výsledky

Po otestování jednotlivých scénářů bylo dosaženo uspokojivých výsledků, nicméně zde bylo zaznamenáno menší množství problémů. První z problémů se vyskytuje při načítání názvů modelů ze serveru. V ojedinělých případech nedokáže mobilní zařízení stáhnout názvy modelů ze serveru. To může být způsobeno pomalým internetovým připojením, jelikož v aplikaci je nastaveno čekání na odpověď serveru do 3 sekund. Pro správné načtení těchto názvů stačí zavřít a znovu otevřít scénu. Druhým z problémů bylo špatné načtení textury při importu 3D modelů. I přes vyzkoušení většího počtu nástrojů pro importování 3D modelů nebylo možné najít takový, který zvládne správně naimportovat každý 3D model. Z toho důvodu je vhodné importovat model nejprve do aplikace Model Converter a provést důkladnou kontrolu před jeho vložením na server. Aplikace zároveň neumožňuje pracovat s animacemi v modelech.

ZÁVĚR

Cílem této diplomové práce bylo vytvořit mobilní aplikaci, která by předvedla možnosti využití rozšířené reality ve vnitřním prostředí. Za tímto účelem byla vytvořena aplikace AR Magic pro zařízení s operačním systémem Android. Tato aplikace je rozdělena do čtyř částí, přičemž každá část je zaměřena na jednu z možností využití rozšířené reality. Díky využití technologie ARCore pro práci s rozšířenou realitou na operačním systému Android, byly tyto možnosti omezené oproti ARKitu. Při použití ARKitu by bylo navíc možné provádět rozpoznání objektů nebo rozpoznání těla. Mezi hlavní možnosti, které ARCore nabízí a byly využity v aplikaci, patří rozpoznání značky, obličeje a horizontálních ploch. Pro šetření úložiště na zařízení, byl také navržen server, který uchovává data ve formě 3D modelů, jejich konfigurací a videí. Pro zobrazení 3D modelů jsou v aplikaci využívány modely ve formátu GLB. Vytvořením aplikace AR Magic byl tento cíl splněn.

K mobilní aplikaci byla také vytvořena aplikace pro operační systém Windows. Tato aplikace s názvem Model Converter slouží ke změně velikosti a rotace modelů před nahráním na server. Její hlavní účel je změna velikosti jednotlivých modelů tak, aby korespondovaly s velikostí značky v AR Markeru nebo byly dostatečně velké v AR Area. Zároveň lze v této části umístit 3D model na referenční obličej, pomocí čehož nastavíme polohu modelu na hlavě pro část AR Face. Z této aplikace je možné model exportovat lokálně nebo přímo na server.

Teoretická část byla zaměřena na charakterizaci rozšířené reality a její aktuální využití ve vnitřním i vnějším prostředí. Spolu s tím byly definovány jednotlivé typy a technologie rozšířené reality, přičemž ve vytvořené mobilní aplikaci byl využit typ založený na rozpoznání značky (marker-based AR), ale i typ který značku nevyužívá (markerless AR). Z popsaných technologií byly využity AR Foundation spolu s ARCore pro práci s rozšířenou realitou. V další kapitole byly představeny enginy, ve kterých je možné vytvářet aplikace využívající rozšířenou realitu. Ze zmíněných enginů byl pro tvorbu aplikace vybrán Unity engine. V závěru teoretické části byla zmíněna zařízení, pomocí kterých můžeme pracovat s rozšířenou realitou.

POUŽITÁ LITERATURA

- [1] GILLIS, Alexander S. augmented reality (AR). In: *TechTarget* [online]. 11. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.techtarget.com/whatis/definition/augmented-reality-AR>
- [2] HAYES, Adam. Augmented Reality (AR) Defined, With Examples and Uses. In: *Investopedia* [online]. 15. 05. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.investopedia.com/terms/a/augmented-reality.asp>
- [3] Microsoft. What is augmented reality or AR?. In: *Microsoft* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://dynamics.microsoft.com/en-us/mixed-reality/guides/what-is-augmented-reality-ar/>
- [4] UOTINEN, Lauri. Visualization of 3D rock mass properties in underground tunnels using extended reality. In: *ResearchGate* [online]. 04. 2021. [cit. 2023-07-26]. Dostupné z: https://www.researchgate.net/figure/Extended-Reality-XR-technologies_fig1_354858662
- [5] Inter IKEA Systems. IKEA Place App. In: *IKEA*. [online]. ©1999–2023. [cit. 2023-07-26]. Dostupné z: <https://www.ikea.com/ch/en/customer-service/mobile-apps>
- [6] Say Hej to IKEA Place. In: *Youtube* [online]. 12. 09. 2017. [cit. 2023-07-26]. Dostupné z: https://www.youtube.com/watch?v=UudV1VdFtuQ&ab_channel=IKEA. Kanál uživatele IKEA.
- [7] Insider Navigation Inc. Top 3 Augmented Reality Wayfinding & Indoor Navigation Technologies. In: *Insider Navigation* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://insidernavigation.com/ar-indoor-navigation/>
- [8] Apple Inc. AR Anatomy 4D+. In: *App Store* [online]. 22. 04. 2018. [cit. 2023-07-26]. Dostupné z: <https://apps.apple.com/us/app/ar-anatomy-4d/id1315392436>
- [9] Google. AR-ANIMALS BOOK. In: *Google Play* [online]. 01. 12. 2014. [cit. 2023-07-26]. Dostupné z: https://play.google.com/store/apps/details?id=com.gt.aranimals&hl=en_US
- [10] CORPUZ, John. Best stargazing apps for looking at the night sky in 2021. In: *Tom's Guide* [online]. 14. 07. 2021. [cit. 2023-07-26]. Dostupné z: <https://www.tomsguide.com/round-up/best-stargazing-apps>
- [11] NIELD, David. These Are the Best AR Apps You Can Use Outside. In: *Gizmodo* [online]. 19. 04. 2021. [cit. 2023-07-26]. Dostupné z: <https://gizmodo.com/these-are-the-best-ar-apps-you-can-use-outside-1846696834>
- [12] VAKHNENKO, Helen. Augmented Reality Navigation: Killer Feature for Your Mapping App. In: *Agilie* [online]. 10. 04. 2022, poslední aktualizace 20. 03. 2023. [cit. 2023-07-26]. Dostupné z: <https://agilie.com/blog/augmented-reality-navigation-killer-feature-for-your-mapping-app>
- [13] Digital Promise. Types of AR. In: *Digital Promise* [online]. 13. 05. 2022. [cit. 2023-07-26]. Dostupné z: <https://digitalpromise.org/initiative/360-story-lab/360-production-guide/investigate/augmented-reality/getting-started-with-ar/types-of-ar/>
- [14] ZVEJNIEKS, Gatis. How to create marker-based augmented reality content?. In: *Overly* [online]. 18. 07. 2022. [cit. 2023-07-26]. Dostupné z: <https://overlyapp.com/blog/how-to-create-marker-based-augmented-reality-experiences-without-code/>

- [15] SZCZEPANIAK, Aleksandra. Real or not real? Welcome to Augmented Reality. In: *Leo Rover* [online]. 24. 01. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.leorover.tech/post/real-or-not-real-welcome-to-augmented-reality>
- [16] Program-Ace. 4 Key Types of AR and What Makes Them Special: What is Markerless Augmented Reality?. In: *Program-Ace* [online]. 19. 10. 2021. [cit. 2023-07-26]. Dostupné z: <https://program-ace.com/blog/types-of-ar/>
- [17] HUMPHERY-JENNER, Mark. What went wrong with Pokémon Go? Three lessons from its plummeting player numbers. In: *The Conversation* [online]. 18. 10. 2016. [cit. 2023-07-26]. Dostupné z: <https://theconversation.com/what-went-wrong-with-pokemon-go-three-lessons-from-its-plummeting-player-numbers-67135>
- [18] DUMBRE, Sakshi. Projection-based AR. In: *Medium* [online]. 10. 05. 2023. [cit. 2023-07-26]. Dostupné z: <https://medium.com/@sakshi.dumbre31/projection-based-ar-220240721883>
- [19] DAVIS, Todd. Explained! What is Projected Augmented Reality? (with examples). In: *EssentialPicks* [online]. 18. 01. 2021, poslední aktualizace 04. 08. 2022. [cit. 2023-07-26]. Dostupné z: <https://essentialpicks.com/projected-augmented-reality>
- [20] Pool Live Aid - Qualifica, Porto. In: *Youtube* [online]. 19. 03. 2014. [cit. 2023-07-26]. Dostupné z: https://www.youtube.com/watch?v=l2bzAmysjc8&ab_channel=Lu%C3%ADsCarlos. Kanál uživatele Luís Carlos.
- [21] Sergiy. Everything you need to know to build a location-based AR app. In: *Vakoms* [online]. 18. 04. 2023. [cit. 2023-07-26]. Dostupné z: <https://vakoms.com/blog/everything-you-need-to-know-to-build-location-based-ar-app/>
- [22] WILLIAMS, Robert. Google shuts down Tango AR platform. In: *Marketing Dive* [online]. 18. 12. 2017. [cit. 2023-07-26]. Dostupné z: <https://www.marketingdive.com/news/google-shuts-down-tango-ar-platform/513284/>
- [23] AMADEO, Ron. Google's ARCore brings augmented reality to millions of Android devices. In: *Ars Technica* [online]. 29. 08. 2017. [cit. 2023-07-26]. Dostupné z: <https://arstechnica.com/gadgets/2017/08/googles-arcore-brings-augmented-reality-to-millions-of-android-devices/>
- [24] Quiller Media Inc. ARKit. In: *AppleInsider* [online]. 2022. [cit. 2023-07-26]. Dostupné z: <https://appleinsider.com/inside/arkit>
- [25] Unity Technologies. AR Foundation. In: *Unity3D* [online]. 22. 05. 2023. [cit. 2023-07-26]. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.0/manual/index.html>
- [26] PTC Inc. Getting Started with Vuforia Engine in Unity. In: *Vuforia* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://library.vuforia.com/getting-started/getting-started-vuforia-engine-unity>
- [27] PTC Inc. Vuforia Engine Overview. In: *Vuforia* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://library.vuforia.com/getting-started/vuforia-features>
- [28] PTC Inc. Using CAD models vs 3D scans for Model Targets. In: *Vuforia* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://library.vuforia.com/objects/comparison-object-tracking-methods>
- [29] PTC Inc. Area Targets. In: *Vuforia* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://library.vuforia.com/environments/area-targets>

- [30] PTC Inc. Best Practices for Preparing and Scanning an Environment. In: *Vuforia* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://library.vuforia.com/area-targets/best-practices-scanning-environment>
- [31] Unity Technologies. Introduction. In: *Unity3D* [online]. ©2020. [cit. 2023-07-26]. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.mars@1.0/manual/index.html>
- [32] JOHNSON, Braelyn. How AR Foundation and Unity Mars work together to enable interactive, multiplatform AR experiences. In: *Unity* [online]. 11. 05. 2020. [cit. 2023-07-26]. Dostupné z: <https://blog.unity.com/technology/how-ar-foundation-and-mars-work-together-to-enable-interactive-multiplatform-ar>
- [33] Unity Technologies. Unity Mars. In: *Unity Store* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: https://store.unity.com/configure-plan/unity-mars?free_trial_product=true&check_logged_in=1
- [34] Unity Technologies. GET STARTED WITH UNITY MARS. In: *Unity* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://unity.com/products/mars/get-started>
- [35] AO Kaspersky Lab. What are the Security and Privacy Risks of VR and AR. In: *Kaspersky* [online]. 25. 05. 2021. [cit. 2023-07-26]. Dostupné z: <https://usa.kaspersky.com/resource-center/threats/security-and-privacy-risks-of-ar-and-vr>
- [36] AMBOLIS, Diana. Top 10 AR Devices That Are Riding The Wave Of Future. In: *Blockchain Magazine* [online]. 12. 06. 2023. [cit. 2023-07-26]. Dostupné z: <https://blockchainmagazine.net/top-10-ar-devices-that-are-riding-the-wave-of-future/>
- [37] ONVU Technologies Group AG. Augmented Reality: The Real Deal in Security | OnCam. In: *OnCam* [online]. ©2021. [cit. 2023-07-26]. Dostupné z: <https://blog.oncamgrandeye.com/featured/augmented-reality-the-real-deal-in-security>
- [38] Microsoft. Unity. In: *Microsoft* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://dotnet.microsoft.com/en-us/apps/games/unity>
- [39] SINICKI, Adam. What is Unity? Everything you need to know. In: *Android Authority* [online]. 20. 03. 2021. [cit. 2023-07-26]. Dostupné z: <https://www.androidauthority.com/what-is-unity-1131558/>
- [40] Zenva. What is Unity? – A Guide for One of the Top Game Engines. In: *Zenva* [online]. 04. 07. 2023. [cit. 2023-07-26]. Dostupné z: <https://gamedevacademy.org/what-is-unity/>
- [41] PETTY, Josh. What is Unity 3D & What is it Used For?. In: *Concept Art Empire* [online]. 14. 03. 2019. [cit. 2023-07-26]. Dostupné z: <https://conceptartempire.com/what-is-unity/>
- [42] SHAH, Mit. What is Unity3D? Everything You Need to Know. In: *ViitorCloud* [online]. 30. 06. 2022. [cit. 2023-07-26]. Dostupné z: <https://viitorcloud.com/blog/what-is-unit-3d/>
- [43] DRAKE, Jeff. 19 Great Games That Use The Unity Game Engine. In: *TheGamer* [online]. 17. 03. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.thegamer.com/unity-game-engine-great-games/>
- [44] MOSS, Elizabeth. What Is Unreal Engine?. In: *BairesDev* [online]. 25. 08. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.bairesdev.com/blog/what-is-unreal-engine/>
- [45] Computer Hope. Unreal Engine. In: *Computer Hope* [online]. 02. 08. 2019. [cit. 2023-07-26]. Dostupné z: <https://www.computerhope.com/jargon/u/unreal-engine.htm>

- [46] Redacción Tokio. Unreal engine blueprint: the simple way to develop videogames. In: *Tokio School* [online]. 26. 12. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.tokioschool.com/en/news/unreal-blueprint/>
- [47] Epic Games. Augmented Reality Overview. In: *Unreal Engine* [online]. ©2004–2023. [cit. 2023-07-26]. Dostupné z: <https://docs.unrealengine.com/4.26/en-US/SharingAndReleasing/XRDevelopment/AR/HandheldAR/AROverview/>
- [48] SHAPIRA, David. Augmented Reality: A User’s Guide to Smart Phones vs. Smart Glasses. In: *dormakaba* [online]. 14. 09. 2020. [cit. 2023-07-26]. Dostupné z: <https://blog.dormakaba.com/augmented-reality-a-users-guide-to-smart-phones-vs-smart-glasses/>
- [49] ROUSE, Margaret. Augmented Reality Headset. In: *Techopedia* [online]. 27. 02. 2012. [cit. 2023-07-26]. Dostupné z: <https://www.techopedia.com/definition/23143/augmented-reality-headset-ar-headset>
- [50] ELIAÇIK, Eray. Comparison: Apple Vision Pro vs HoloLens 2. In: *Ghacks* [online]. 10. 06. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.ghacks.net/2023/06/10/comparison-apple-vision-pro-vs-hololens-2/>
- [51] FISHER, Tim. VR/AR Headsets vs. Smart Glasses: What's the Difference?. In: *Lifewire* [online]. 16. 01. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.lifewire.com/headsets-vs-smart-glasses-6891703>
- [52] ALLISON, Conor. Best smart glasses and AR specs 2023: Tested picks from Snap, Meta and Amazon. In: *Wareable* [online]. 06. 06. 2023. [cit. 2023-07-26]. Dostupné z: <https://www.wareable.com/ar/the-best-smartglasses-google-glass-and-the-rest>
- [53] LANG, Ben. Hands-on: Mojo Vision’s Smart Contact Lens is Further Along Than You Might Think. In: *Road To VR* [online]. 05. 07. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.roadtovr.com/mojo-vision-smart-contact-lens-ar-hands-on/>
- [54] TechTarget Contributor. 3D Model. In: *TechTarget*. [online]. 09. 2016. [cit. 2023-07-26]. Dostupné z: <https://whatis.techtarget.com/definition/3D-model>
- [55] FutureLearn. What is 3D modelling and what is it used for?. In: *FutureLearn* [online]. 18. 03. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.futurelearn.com/info/blog/general/what-is-3d-modelling>
- [56] RUEL, Alexia. What is a GLB file? Learn more about GLB file. In: *Visao* [online]. 06. 01. 2022. [cit. 2023-07-26]. Dostupné z: <https://visao.ca/what-is-glb-file/>
- [57] CALVELLO, Mara. 15 Best 3D Modeling Software Platforms in 2022. In: *G2* [online]. 27. 06. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.g2.com/articles/best-3d-modeling-software>
- [58] CROWDER, Ashley. Top 3D File Formats For 3D Commerce, Social & More. In: *VNTANA* [online]. 01. 10. 2020. [cit. 2023-07-26]. Dostupné z: <https://www.vntana.com/blog/demystifying-3d-file-formats-for-3d-commerce-and-more/>
- [59] Packt Publishing Limited. Getting started with the Unity Editor. In: *Packt* [online]. 04. 2022. [cit. 2023-07-26]. Dostupné z: <https://subscription.packtpub.com/book/game-development/9781801078078/2/ch021v11sec04/getting-started-with-the-unity-editor>
- [60] Unity Technologies. Order of execution for event functions. In: *Unity3D* [online]. 18. 03. 2019. [cit. 2023-07-26]. Dostupné z: <https://docs.unity3d.com/Manual/ExecutionOrder.html>

- [61] Pluralsight LLC. What is C# Programming? A Beginner's Guide. In: *Pluralsight* [online]. 14. 11. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c->
- [62] STENIS, Luke. Windows apps. In: *Austin Coding Academy* [online]. 08. 04. 2019. [cit. 2023-07-26]. Dostupné z: <https://austincodingacademy.com/blog/2017/05/17/5-awesome-apps-you-can-create-with-c-sharp/>
- [63] BROWN, C. Scott. What is Android? Here's everything you need to know. In: *Android Authority* [online]. 19. 02. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.androidauthority.com/what-is-android-328076/>
- [64] StatCounter. Mobile Operating System Market Share Worldwide. In: *Statcounter Global Stats* [online]. ©1999-2023. [cit. 2023-07-26]. Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [65] Google. Platform architecture. In: *Android developers* [online]. 04. 05. 2023. [cit. 2023-07-26]. Dostupné z: <https://developer.android.com/guide/platform>
- [66] JOHARI, Aayushi. Android Tutorial – Learn Android From Scratch!. In: *Edureka* [online]. 09. 02. 2021. [cit. 2023-07-26]. Dostupné z: <https://www.edureka.co/blog/android-tutorial/>
- [67] InterviewBit. Android Architecture – Detailed Explanation. In: *InterviewBit* [online]. 03. 06. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.interviewbit.com/blog/android-architecture/>
- [68] Tutorials Point. Android – Architecture. In: *Tutorials Point* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: https://www.tutorialspoint.com/android/android_architecture.htm
- [69] AltexSoft. Functional and Nonfunctional Requirements: Specification and Types. In: *AltexSoft* [online]. 23. 07. 2021. [cit. 2023-07-26]. Dostupné z: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>
- [70] Scaled Agile Inc. Nonfunctional requirements. In: *Scaled Agile Framework* [online]. 13. 03. 2023. [cit. 2023-07-26]. Dostupné z: <https://scaledagileframework.com/nonfunctional-requirements/>
- [71] chitrasingla2001. Functional vs Non Functional Requirements. In: *GeeksForGeeks* [online]. 02. 12. 2022. [cit. 2023-07-26]. Dostupné z: <https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/>
- [72] ČÁPKA, David. Lekce 2 - UML - Use Case Diagram. In: *itnetwork.cz* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>
- [73] Nishadha. Use Case Diagram Relationships Explained with Examples. In: *Creately Blog* [online]. 13. 12. 2022. [cit. 2023-07-26]. Dostupné z: <https://creately.com/blog/diagrams/use-case-diagram-relationships/>
- [74] ČÁPKA, David. Lekce 3 - UML - Use Case Specifikace. In: *itnetwork.cz* [online]. ©2023. [cit. 2023-07-26]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-specifikace-diagram>
- [75] JENÍČKOVÁ, Kateřina. Zmapování procesů a vazeb v informačním systému. In: *vovcr* [online]. 26. 04. 2021. [cit. 2023-07-26]. Dostupné z: <https://www.vovcr.cz/odz/tech/213/page04.html>

- [76] JavaTpoint. XAMPP TUTORIAL. In: *JavaTpoint* [online]. ©2011-2021 [cit. 2023-07-26]. Dostupné z: <https://www.javatpoint.com/xampp>
- [77] JavaTpoint. What is Apache?. In: *JavaTpoint* [online]. ©2011-2021 [cit. 2023-07-26]. Dostupné z: <https://www.javatpoint.com/what-is-apache>
- [78] BRIGSTED, Thor. GLTFUtility. In: *GitHub* [online]. 30. 03. 2019, poslední aktualizace 07. 12. 2022. [cit. 2023-07-26]. Dostupné z: <https://github.com/Siccity/GLTFUtility>
- [79] ATTENEDER, Andreas. glTFast. In: *GitHub* [online]. 13. 11. 2020, poslední aktualizace 09. 12. 2022. [cit. 2023-07-26]. Dostupné z: <https://github.com/atteneder/glTFast>
- [80] prefrontal cortex. UnityGLTF. In: *GitHub* [online]. 09. 03. 2017, poslední aktualizace 16. 03. 2023. [cit. 2023-07-26]. Dostupné z: <https://github.com/prefrontalcortex/UnityGLTF>

OBSAH CD

Přiložené CD obsahuje:

- Elektronickou verzi diplomové práce DolejsP_VyvojUkazkove_JP_2023.
- Složku *Struktura_server* – obsahuje strukturu složek pro server včetně modelů, konfigurací a videí.
- Složku *AR_Magic* – obsahuje zdrojové kódy a apk pro mobilní aplikaci AR Magic.
- Složku *Model_converter* – obsahuje zdrojové kódy a spustitelnou aplikaci Model converter pro Windows.
- Složka *Markers* – obsahuje značky, které lze využít pro videa ve složce *Struktura_server/unity/videos*.