



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Aplikácia pre interaktívnu prezentáciu 3D modelov
Student:	Bc. Oliver Findra
Vedoucí:	Ing. Ján Vrábek
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

Cieľom práce je navrhnuť a implementovať multiplatformovú aplikáciu v Unity určenú na interaktívne prezentovanie 3D modelov. Aplikácia bude umožňovať posun a otáčanie modelu, kliknutie na preddefinované UI elementy na modeli pre zobrazenie podrobností – textu, obrázku alebo videa, prípadne spustenie animácie. Model bude tiež možné prezentovať pomocou rozšírenej reality (ak to bude zariadenie umožňovať).

Postup:

- Analyzujte už existujúce aplikácie s podobnou funkcionalitou.
- Naštudujte a popíšte herný engine Unity.
- Navrhните concept aplikácie na základe predošlých analýz.
- Navrhните uživateľské rozhranie aplikácie.
- Implementujte funkčný prototyp v Unity.
- Aplikáciu otestujte.
- Zhodnoťte použiteľnosť riešenia a diskutujte možné rozšírenia.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 15. února 2020



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Diplomová práce

Aplikácia pre interaktívnu prezentáciu 3D modelov

Bc. Oliver Findra

Katedra softwarového inženýrství

Vedúci práce: Ing. Ján Vrábek

30. júla 2020

Pod'akovanie

Chcel by som pod'akovať svojej rodine za ich podporu počas mojich štúdií a písaní práce a taktiež Ing. Jánovi Vrábľovi za vedenie práce, cenné rady a odbornú pomoc.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 30. júla 2020

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2020 Oliver Findra. Všetky práva vyhradené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Findra, Oliver. *Aplikácia pre interaktívnu prezentáciu 3D modelov*. Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Táto diplomová práca sa venuje vývoju mobilnej multiplatformovej aplikácie pre vytváranie interaktívnych prezentácií k 3D modelom. Práca je zameraná hlavne na tvorbu popisných bodov modelov a prezentovanie v rozšírenej realite. Aplikácia je vyvíjaná cez nástroj Unity s využitím Firebase ako serverovej časti pre správu používateľov na demonštráciu fungovania v rámci spoločnosti.

Kľúčová slova Unity, rozšírená realita, interaktívna prezentácia, mobilná aplikácia, 3D modely

Abstract

This Master's thesis describes the process of development of a multi-platform application for mobile devices, which serves for creation of interactive presentations for 3D models. Thesis is focused mainly on creation of description points for models and presenting the models in augmented reality. Application is developed in Unity with the use of Firebase as the server side for user management to demonstrate possible use in companies.

Keywords Unity, augmented reality, interactive presentation, mobile application, 3D models

Obsah

Úvod	1
1 Cieľ	3
2 Technológie	5
2.1 Rozšírená realita	5
2.2 Unity3D	8
3 Analýza	15
3.1 Existujúce riešenia	15
3.2 Špecifikácia požiadaviek	21
4 Návrh	25
4.1 Modelovanie prípadov použitia	25
4.2 Doménový model tried	31
4.3 Používateľské rozhranie	33
5 Implementácia	39
5.1 Použité nástroje a knižnice	39
5.2 Firebase server	42
5.3 Unity projekt	42
5.4 Používateľský vstup a gestá	45
5.5 Tvorba interaktívneho bodu	47
5.6 Rozšírená realita	49
6 Testovanie	51
6.1 Testovanie použiteľnosti	51
Záver	57
Možnosti rozšírenia	57

Literatúra	59
A Zoznam použitých skratiek	63
B Diagramy	65
C Wireframy	67
D Ukážky aplikácie	71
E Obsah priloženého CD	77

Zoznam obrázkov

2.1	Pokémon GO - populárna mobilná hra, ktorá využíva technológiu rozšírenej reality [8]	8
2.2	Porovnanie herných enginov podľa ich znovupoužitelnosti [11]- čím viac vpravo, tým je viac engine oddelený od konkrétnej implementácie a môže byť použitý na tvorbu väčšej škály hier	10
2.3	Zjednodušený diagram hernej slučky a preddefinovaných udalostí, na ktoré je možné reagovať v skriptoch Unity [10] (niektoré časti sú vynechané pre veľkosť diagramu)	11
2.4	Hierarchia objektov v Unity z [17]	13
3.1	Prezentácia 3D modelu v programe PowerPoint	16
3.2	Tvorba 3D prezentácie v nástroji Aurora 3D	17
3.3	3D prezentácia v nástroji Savree	19
3.4	Aplikácia Augment (iOS verzia)	20
4.1	Doménový model tried	33
5.1	Diagram ukazujúci štruktúru XR framework-u v Unity, a ako fungovanie s rôznymi poskytovateľmi implementácií [28]	41
5.2	Proces prihlasovania cez Firebase	43
5.3	Diagram tried vstupu	47
5.4	Raycast z kamery na collider	48
5.5	Vizuál interaktívneho bodu	49
B.1	Graf úloh	66
C.1	Prihlasovacia obrazovka	67
C.2	Zvolenie počiatočného módu po prihlásení	68
C.3	Obrazovka s produktami a prezentáciami	68
C.4	Obrazovka prezentácie	69
C.5	Obrazovka prezentácie s otvoreným interaktívnym bodom	69

D.1	Prihlasovacia obrazovka	71
D.2	Registračný formulár	72
D.3	Počiatkové zvolenie režimu	72
D.4	Výber modelu a prezentácie	73
D.5	Používateľské menu	73
D.6	Schvaľovanie používateľov administrátorom	74
D.7	Tvorba prezentácie - 3D režim	74
D.8	Tvorba prezentácie - AR režim	75

Zoznam tabuliek

4.1	Pokrytie funkčných požiadaviek	32
-----	--	----

Úvod

V posledných rokoch zaznamenala rozšírená realita veľký nárast na popularite. Postupný vývoj tejto technológie dovedol aj najväčšie firmy ako Google a Apple k vydaniu nástrojov s ich podporou, čo viedlo k oveľa lepšej dostupnosti pre vývojárov.

Rozšírená realita dokáže jednoznačne zaujať. Vidieť produkt naživo v prostredí cez mobilné zariadenie prináša oveľa lepší zážitok ako ho vidieť len na obrázku či videu.

Čím ďalej sa objavuje viac aplikácií, ktoré sa snažia používateľovi priblížiť cez túto novú technológiu. Väčšinou ide o špecifické riešenia jednej spoločnosti, ktorá sa rozhodla takto prezentovať svoje produkty (napríklad spoločnosť *IKEA* s aplikáciou pre umiestňovanie nábytku v rozšírenej realite). Pre takúto formu prezentácie je potrebná veľká investícia a dlhý čas na vývoj.

Prezentovať svoje produkty klasickým spôsobom môže byť pre firmy zložité. Mať prístupné všetky produkty na konferenciách, výstavách alebo pobočkách dokáže byť nákladné až nemožné. Dobrá prezentácia produktu umožňuje tento deficit aspoň čiastočne napraviť. Interaktivita a nové technológie môžu zákazníka určite zaujať a vniesť oživenie oproti klasickému prístupu.

Preto som sa rozhodol vytvoriť aplikáciu, ktorá by umožňovala vytvárať a vykonávať takéto prezentácie na mobilnom zariadení. Interaktívne prezentácie s popismi častí produktu môžu uľahčiť prácu firmám a ich predajcom, pre ktorých chcem túto prácu zamerať.

Prvá časť práce je venovaná popisom hlavných použitých technológií - rozšírenej reality a nástroja pre tvorbu multiplatformových aplikácií Unity, rešerši už existujúcich podobných riešení interaktívnych prezentácií a špecifikácii požiadaviek aplikácie. Druhá časť sa zaoberá návrhom aplikácie, vrátane prípadov použitia, doménového modelu a návrhu používateľského rozhrania. V tretej časti sú opísané použité nástroje a detaily implementácie prototypu. Posledná časť je venovaná testovaniu z pohľadu používateľa.

Cieľ

Cieľom práce je vytvoriť aplikáciu, ktorá umožní jednoduché vytváranie interaktívnych prezentácií pre 3D modely produktov. Primárnymi užívateľmi pre vytváranie prezentácií sú predajcovia, ktorí ponúkajú produkt pre zákazníka. Produkt môže mať podobu akéhokoľvek fyzického objektu, pre ktorý je vytvorený počítačový 3D model.

V aplikácii budú k dispozícii virtuálne modely produktov firmy. Predajca by mal byť schopný jednoducho a intuitívne vytvoriť k zvolenému 3D modelu interaktívne body, ktoré budú schopné poskytnúť dostatok informácií o rôznych častiach produktu. Základná funkcionality interaktívneho bodu by mala byť zobrazenie textu, obrázku, videa alebo prehranie animácie modelu.

Zákazníkovi by malo byť umožnené prezeráť si ponuku produktov. Po vybratí by mal mať možnosť prezeráť si model pomocou intuitívneho ovládania a čo najjednoduchšie získať informácie o častiach produktu, ktoré ho zaujímajú. Aplikácia by mala umožňovať prechod do rozšírenej reality, kde používateľ môže vidieť, ako by produkt vypadal v priestore okolo neho.

Keďže aplikácia má slúžiť firmám, v práci by mal byť zahrnutý aj jednoduchý prihlasovací systém, aby bol obmedzený prístup len pre schválených používateľov.

Technológie

2.1 Rozšírená realita

Rozšírená realita (anglicky *Augmented reality*, skrátene *AR*) je technológia, ktorá kombinuje virtuálnu realitu s reálnym svetom. Hlavný princíp AR je pripojiť počítačovo generované virtuálne informácie, ako text, obrázok, 3D model, hudbu, video atď., do reálneho sveta. Tieto 2 druhy informácií sa navzájom dopĺňajú a dosahujú rozšírenie reálneho sveta o nový obsah. [1]

V posledných rokoch je stále väčší záujem výskumných inštitúcií, univerzít a podnikov investovať do štúdií spojených s rozšírenou realitou. Neustále zlepšovanie počítačového softvéru a hardvéru posunulo AR z teoretického výskumu do praktického využitia, kde poskytuje ľuďom nový spôsob ako vidieť a skúmať svoje okolie. [1]

Vo vývoji AR v 3D priestore sú kľúčovými technológiami podľa [1] nasledujúce:

- **Inteligentné obrazovky**

Na prácu s AR je potrebné mať správny hardware. Najviac intuitívny spôsob interakcie človeka s reálnym svetom je zrak - pochádza z neho až 65% prijatých informácií z okolitého prostredia. Pokročilý vývoj inteligentného displeja umožnil lepší a kvalitnejší vývoj AR pre tieto zariadenia. Podľa [2] sú 3 hlavné kategórie zariadení, ktoré sú používané s AR:

- **Obrazovky pripevnené k hlave** - väčšinou sa jedná o zariadenia, ktoré zobrazujú obsah na obrazovke tesne pred okom používateľa, ako napríklad smart okuliare alebo kontaktné šošovky
- **Malé príručné zariadenia** - malé zariadenia s displejom, ktoré používateľ drží v ruke, napríklad smart mobilné telefóny

- **Priestorové zobrazenia** (*angl. spatial displays*) - zariadenia, ktoré projektujú grafické informácie priamo na fyzické objekty bez nutnosti používania obrazovky, napríklad smart projektory

- **Registrácia 3D prostredia**

3D registrácia je základnou technológiou v systémoch s rozšírenou realitou. Účelom tejto technológie je presne vypočítať pozíciu a rotáciu kamery v reálnom čase. Na základe toho je vypočítaná pozícia virtuálneho obsahu, ktorý má byť pridaný v projekcii. Spôsob, akým je 3D registrácia riešená, priamo ovplyvňuje výkon systému, preto je dôležité, aby bola technológia efektívna, rýchla a v budúcnosti naďalej skúmaná a zlepšovaná. [3]

- **Inteligentná interakcia**

Od začiatku bol výskum rozšírenej reality sústredený hlavne na oblasti, ktoré AR poskytujú (ako napríklad 3D registrácia a zariadenia), než na metódy, ktoré umožňujú používateľovi lepšie interagovať s virtuálnym obsahom. Interakcia s AR prostredím bola väčšinou limitovaná len na pasívne sledovanie alebo jednoduché listovanie informácií zasadených do reálneho sveta. Viacero aplikácií, ktoré vznikali, neumožňovali používateľovi obsah vytvárať, upravovať alebo meniť za behu, aj keď by sa to logicky v aplikácii hodilo a vylepšilo by to zážitok používateľa. [4]

Všetky tri z vyššie spomenutých technológií sú podstatné z hľadiska tvorby aplikácií, ktoré chcú zobrazovať obsah v AR. Smart mobilné telefóny sú dobrým prostriedkom pre ich realizáciu. V posledných rokoch integrovali rozšírenú realitu do svojich repertoárov aj najväčšie firmy v oblasti mobilných telefónov - Apple a Google. To umožňuje pre vývojárov oveľa jednoduchší prístup k technológii a tým aj signifikantne väčší nárast aplikácií, ktoré využívajú AR. Dve najlepšie SDK (skratka pre *software development kit* - sada vývojových nástrojov) pre prácu s AR sú podľa [5] nasledujúce:

ARKit

V roku 2017 vydala spoločnosť Apple operačný systém iOS 11 a s ním bola pridaná aj sada vývojových nástrojov ARKit, ktorá znamenala veľmi podstatnú udalosť z hľadiska histórie rozšírenej reality. ARKit je výnimočný nástroj, ktorý zaručuje kompatibilitu s mobilmi iPhone a tabletmi iPad (musia byť ale vybavené procesorom A9 alebo lepším). Rovnako ako pri iných SDK pre AR, umožňuje pridávať digitálne informácie a 3D objekty do reálneho sveta.

V [5] sú zhrnuté funkcionality takto:

- *SLAM* (skratka z angl. *Simultaneous Localization and Mapping*) so senzormi - detekcia okolia spolu so zaznamenávaním polohy a orientácie zariadenia

- Odhad okolitého osvetlenia - informácie o osvetlení scény umožňujú osvetliť vložený objekt tak, aby lepšie zapadal do okolitého prostredia
- Odhad mierky a vzdialenosti (angl. *scale estimation*)
- Detekcia vertikálnych a horizontálnych plôch v okolí
- Stabilná a rýchla detekcia pohybu

ARKit využíva VIO - vizuálnu inerciálnu odometriu, ktorá sa používa hlavne v robotike a v počítačovom videní na detegovanie pozície a orientácie v priestore pomocou analýzy obrazov z kamery. Tento prístup je kombinovaný s knižnicou *CoreMotion*, ktorá disponuje dátami z hardware na snímanie pohybu (napríklad z akcelerometrického a gyroskopického senzora). ARKit rozpoznáva dôležité body v obraze scény, zaznamenáva zmeny v pozíciách daných bodov a porovnáva tieto informácie z dátami o pohybe zariadenia z *CoreMotion*. Výsledkom sú informácie o pozícii a pohybe zariadenia s vysokou presnosťou. Takisto je analyzované a zaznamenané aj okolité prostredie a obsah scény. [6]

ARCore

ARCore, vydané v roku 2018, je SDK pre rozšírenú realitu od spoločnosti Google. Podobne ako ARKit, umožňuje jednoduchší vývoj AR aplikácií na spoločnosťou podporovaných smartfónoch a tabletoch. Okrem zariadení Android, poskytuje ARCore aj podporu pre zariadenia s operačným systémom iOS.

Google popisuje v [7] tri kľúčové vlastnosti integrácie virtuálneho obsahu do reálneho sveta:

- Detekcia pohybu - umožňuje sledovanie polohy a orientácie zariadenia vzhľadom na reálny svet
- Pochopenie prostredia - umožňuje zariadeniu detegovať veľkosť a polohu rôznych typov povrchu, ako napríklad horizontálne, vertikálne a šikmé plochy
- Odhad osvetlenia

ARCore využíva podobne ako ARKit vizuálnu inerciálnu odometriu. Funkcionality sú taktiež veľmi podobné a keďže sa jedná o 2 najväčšie spoločnosti na trhu, dá sa čakať, že sa tieto spoločnosti budú vo vývoji navzájom predbiehať a bude ťažké určiť, ktorý z týchto nástrojov má navrch.

Jedným z najznámejších príkladov použitia AR je hra *Pokémon GO* (obrázok 2.1). Hra umožňovala používateľom bojovať proti virtuálnym bytostiam v rozšírenej realite. Hráč videl to, čo sníma kamera jeho mobilného telefónu s tým, že v prostredí boli pridané virtuálni protivníci - pokémoni, ktorých musel chytať.



Obr. 2.1: Pokémon GO - populárna mobilná hra, ktorá využíva technológiu rozšírenej reality [8]

2.2 Unity3D

Unity (tiež známe aj ako Unity3D) je herný engine a vývojové prostredie pre tvorbu intelektívnych aplikácií, hlavne videohier. Výkonný riaditeľ David Helgason definoval Unity ako sadu nástrojov pre tvorbu hier a ako technológiu, ktorá uľahčuje proces z hľadiska grafiky, zvuku, fyzikálneho chovania, interakcií a sieťových prvkov. Unity je známe hlavne ako nástroj pre rýchly vývoj prototypov a veľký počet cieľových platforiem. [9]

Prvá verzia Unity (1.0) vyšla v roku 2005. Pracovali na nej 3 vývojári v Dánsku - David Helgason, Joachim Ante and Nicholas Francis. Boli to herní nadšenci a uvedomovali si, že herný priemysel je na vzostupe. Cieľom bolo vytvoriť dostupný vývojový prostriedok, ktorý umožní vytvárať hry a aplikácie aj jednotlivcom a menším skupinám vývojárov bez nutnosti kupovania drahých licencií a používania komplikovaných riešení. Na začiatku bežalo Unity na operačnom systéme MacOS a od verzie 1.1 podporovalo vývoj hier pre niekoľko platforiem (MacOS, Windows a web). Postupnou prácou na optimalizácii, pridávaní nových funkcionalít a rozšírením na ďalšie platformy sa Unity tešilo čím ďalej väčšej popularite. [9]

Dnes Unity podporuje vývoj aplikácií pre tieto platformy (zoznam je z dokumentácie pre verziu 2019.4 v [10]):

- Windows, MacOS a Linux
- tvOS
- iOS
- Lumin
- Android
- WebGL
- PS4
- Xbox One

Ako už bolo naznačené vyššie, primárnym cieľom Unity je uľahčiť vývoj produktov a umožniť multiplatformový vývoj hier. Unity ponúka integrovaný editor, ktorý je ľahký na používanie. Umožňuje import väčšiny herných objektov ako zvuky, obrázky, 3D modely, animácie, atď. a priamo s nimi pracovať v editore. Obsahuje 3D a 2D pohľad, v ktorom sa dá zostaviť herný svet a používateľské rozhranie. Projekt sa dá spustiť priamo v editore a tiež cez editor na cieľovom zariadení.

Pri práci s importovanými súbormi, využíva Unity *asset conditioning pipeline* [11] - transformuje dáta tak, aby neobsahovali zbytočné informácie a bolo v nich len to, čo sa reálne engine schopný použiť. Súbor môže byť importovaný a exportovaný vo formy tzv. balíčkov, ktoré umožňujú ľahký prenos medzi rôznymi projektmi. Takto "zabalené" súbory môžu obsahovať okrem obrázkov, modelov, materiálov, zvukov a iných aj špecifickú funkcionálnosť, ako AI, sieťové operácie, ovládanie postavy, atď., čím sa dokáže významne zrýchliť vývoj. [12]

Unity taktiež obsahuje kvalitný systém pre animácie modelov, ktorý podporuje aj presmerovanie (z angl. *retargeting*) - prehrávanie animácie z jedného modelu na úplne inom modeli. [11]

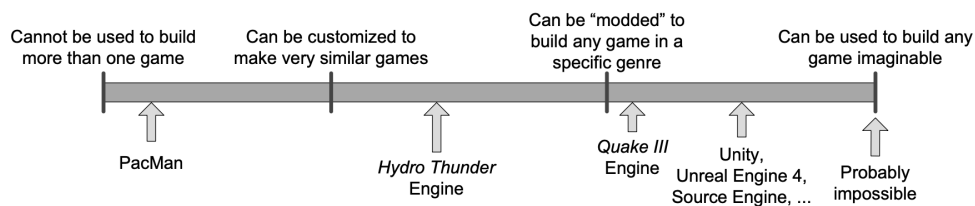
Všetky funkcie v Unity sú dostupné cez prívetivé používateľské rozhranie ovládané väčšinou myšou. Väčšina práce programátora spočíva v písaní Unity skriptov, čo sú komponenty, ktoré určujú chovanie objektov.

Unity primárne podporuje jazyk C#. V minulosti podporovalo aj jazyk Boo, ktorý odstránili z podpory vo verzii Unity 5 [13] a taktiež Javascript, ktorý bol označený ako zastaralý vo verzii 2017.1 [14].

2.2.1 Herný engine

Termín "game engine" (herný engine) sa objavil v polovici 90. rokov 20. storočia v spojení s veľmi populárnou hrou *Doom*. Hra bola navrhnutá tak, aby boli oddelené jej hlavné softvérové komponenty (ako napríklad 3D grafický vykresľovací systém, detekcia kolízií a zvukový systém) od modelov, obrázkov a logiky hry. Úspech tejto architektúry bol zjavný, pretože vývojári začali vytvárať nové hry pomocou tohto prvého engine-u, ktorý menili len minimálne. [11]

Hranica medzi konkrétnou hrou a jej enginom môže byť nejasná. V niektorých riešeniach sú tieto časti jasne oddelené, niekde však vývojári na logické oddelenie časti nedbajú. Ak zdrojový kód obsahuje pevne zakódovanú hernú logiku, pravidlá hry alebo má špeciálne prípady vykresľovania určitých objektov z hry, je ťažké ho znova použiť na tvorbu iného produktu. Termín "game engine" by mal byť teda používaný pre softvér, ktorý je rozšíriteľný a môže slúžiť ako základ pre väčší počet rôznych hier. [11]



Obr. 2.2: Porovnanie herných enginov podľa ich znovupoužiteľnosti [11]- čím viac vpravo, tým je viac engine oddelený od konkrétnej implementácie a môže byť použitý na tvorbu väčšej škály hier

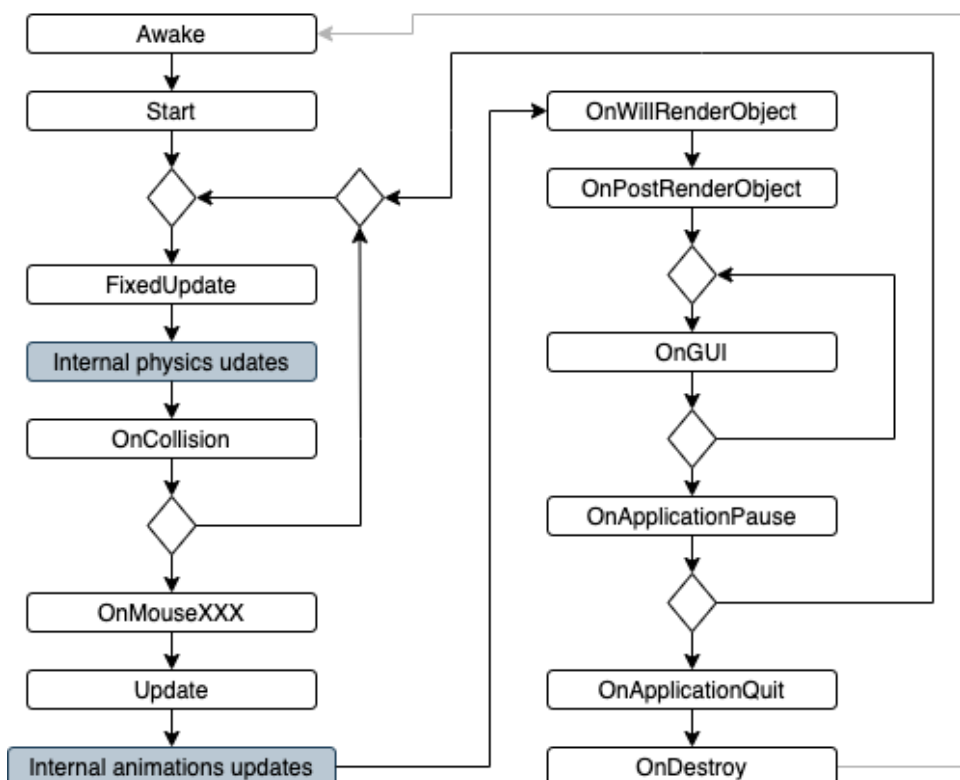
2.2.2 Herná slučka

V klasických grafických používateľských rozhraniach, ktoré používajú napríklad Windows a MacOS je väčšina obrazovky statická - iba malá časť okna sa v jednej chvíli mení. Na toto stačí technika zvaná *rectangle invalidation*, kde sa prekresľuje iba malá časť obrazovky pri zmene. [11]

3D počítačová grafika v reálnom čase je ale implementovaná úplne inak. Kamera a objekty sa neustále pohybujú a je potrebné prekresľovať väčšinu obrazovky veľmi často, aby mal používateľ vidieť plynulý pohyb. Takisto je pri pohybe objektov potrebné kontrolovať ich kolízie a riešiť, čo sa má s objektami stať. [11]

Tento problém rieši v herných enginoch herná slučka. S nejakou určenou frekvenciou volá udalosti každého subsystému - rendering, používateľský vstup, detekciu kolízií, audio, a mnoho ďalších. Je mnoho architektúr, ako toto riešenie

implementovať - Unity využíva architektúru riadenú udalosťami (angl. *event-driven architecture*). To znamená, že pri zmene sa vyvolá globálna udalosť a objekty môžu implementovať metódu, ktorá na danú udalosť reaguje. [12]



Obr. 2.3: Zjednodušený diagram hernej slučky a preddefinovaných udalostí, na ktoré je možné reagovať v skriptoch Unity [10] (niektoré časti sú vynechané pre veľkosť diagramu)

Na zachytávanie väčšiny udalostí hernej slučky slúži v Unity trieda *MonoBehaviour*. Najhlavnejšie časti sú nasledovné:

- **Awake** - funkcia, ktorá je volaná potom, ako je objekt instancovaný (vytvorený).
- **Start** sa vykoná pred Update funkciou prvého snímku po vytvorení objektu.
- **FixedUpdate** - funkcia, ktorá slúži pre fyzikálne výpočty v scéne. Má fixnú frekvenciu a môže byť volaná viac krát v rámci jedného snímku - záleží to od frekvencie hlavnej hernej slučky.

- **Update** je volaná v rámci každého snímku, hlavná funkcia pre aktualizáciu objektov pri novom snímku.
- **LateUpdate** sa volá hneď po funkcii Update. Slúži na výpočty potrebné až po aktualizovni ostatných objektov - napríklad výpočet polohy a rotácie kamery v hre po tom, ako sa postava posunula v rámci daného snímku.
- **OnDestroy** sa vykoná na konci posledného snímku existencie objektu.

2.2.3 Komponentová architektúra

Komponentová architektúra pomáha zvládnuť komplexnosť a vnútornú integritu systémov. Je založená na nezávislých a nahraditeľných komponentách, ktoré zaisťujú budúcu rozšíriteľnosť a udržateľnosť systému. [15]

Táto architektúra sa využíva hlavne preto, aby sa vyhlo zbytočne veľkým a mäťucim hierarchiám dedenia tried, ktoré v komplexnejších programoch vznikali. [16] Tento princíp sa nazýva kompozícia pred dedičnosťou (anglicky *Composition over inheritance*). Spočíva v dosiahnutí chovania podobného polymorfizmu pomocou kompozície objektov namiesto dedičnosti.

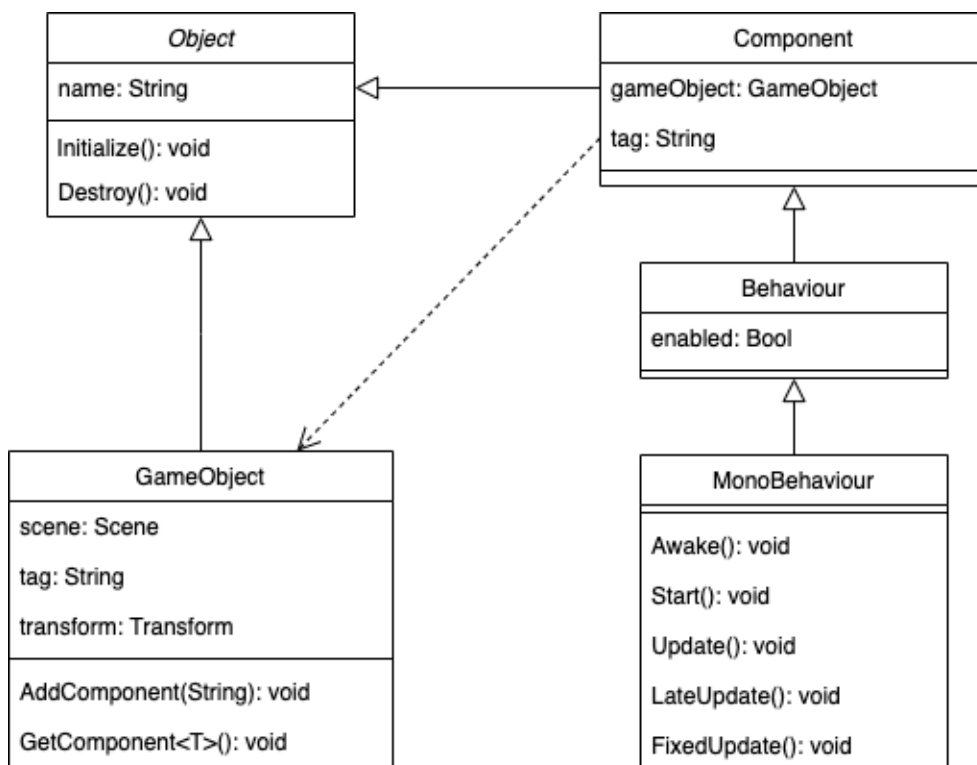
V Unity tvoria komponenty hlavnú časť vývoja. Každý herný objekt v scéne môže obsahovať ľubovoľný počet komponent. Hlavnou myšlienkou tohto princípu je rozdeliť a zapuzdriť funkcionality do samostatných objektov a tým umožniť znovupoužiteľnosť a flexibilitu. Navyše sa tým umožní jednoduché pridávanie a odoberanie funkcionality objektov priamo pri behu programu. Herný objekt je tým pádom kolekciou komponent, ktoré určujú jeho chovanie, vzhľad a funkcionality. [16]

Výhody používania komponent [17]:

- Škálovateľnosť
- Dátovo orientovaná štruktúra
- Komponenty sú ľahko zameniteľné a znovupoužiteľné
- Je ľahké spraviť nový typ objektu
- Polymorfné operácie pre komponenty

Nevýhody komponentovej architektúry [17]:

- Dynamická typová kontrola
- Všetky závislosti musia byť prepojené
- Ťažší debug
- Refaktoring môže byť o dosť zložitejší



Obr. 2.4: Hierarchia objektov v Unity z [17]

Diagram ukazuje kompozíciu objektov v Unity, ktoré tvoria základ komponentového prístupu. Trieda **Component** (komponenta) má svoj herný objekt ku ktorému je pripojená a pridáva mu funkcionalitu. V prípade triedy **MonoBehaviour** sú to okrem iného aj metódy ktoré sa volajú v daných úsekoch hernej slučky.

Najesenciálnejšou komponentou v Unity je *Transform*. Určuje polohu, rotáciu a mierku herného objektu v scéne a obsahuje rôzne metódy na jej zmenu. Bez tejto komponenty nemôže objekt existovať v scéne.

2.2.4 Kľúčové pojmy

V tejto časti sú definované dôležité pojmy pri vývoji v Unity, ktoré sú spomínané v ďalších častiach práce a ktoré neboli bližšie opísané v predošlých sekciách.

- **Projekt**

Projektom sú myslené všetky zdrojové súbory, s ktorými pracuje vývojár. Patria k tomu assety, skripty, a rôzne ostatné špecifické súbory Unity (scény, prefabry).

- **Scéna**

Kolekcia objektov, ktorá sa používa ako celok. S načítaním scény sa odstránia všetky nepotrebné herné objekty z aktuálnej scény a vytvorí sa hierarchia objektov z novej scény.

- **Asset**

Zdroj používaný v aplikácii - zvuk, textúra, animácia, model, atď. Môže to byť čokoľvek, čo môže byť znovupoužitý vývojárom.

- **Herný objekt (GameObject)**

Základná trieda všetkých entít, ktoré majú inštanciu v scéne.

- **Prefab**

Asset, kde je uložený herný objekt spolu s jeho komponentami a vlastnosťami. Prefab predstavuje šablonu na základe ktorej môže vývojár vytvárať inštancie predpripravených herných objektov v scéne.

- **Renderer**

Renderer je komponenta, ktorá sa stará o vykresľovanie herného objektu v scéne.

- **Collider**

Collider je neviditeľná komponenta, ktorá definuje tvar herného objektu pri kontrole a vyhodnocovaní fyzikálnych kolízií. Tvar collideru zvykne byť jednoduchší ako tvar herného objektu pre jednoduchší výpočet kolízií v scéne. V Unity existujú nasledovné typy:

- BoxCollider - má tvar kvádra
- SphereCollider - tvar gule
- CapsuleCollider - pozostáva z valca a dvoch pologuľových koncoviek
- MeshCollider - zložitejšie štruktúry, môže aj presne kopírovať tvar objektu

BoxCollider, SphereCollider a CapsuleCollider sú najjednoduchšie a významne urýchľujú detekciu kolízií. CapsuleCollider je používaný hlavne preto jeho tvar, ktorý dobre imituje postavu človeka. MeshCollider je najzložitejší a používa sa, keď potrebná vysoká presnosť detekcie. Táto presnosť však prináša vyššiu réžiu spracovania a výrazne ovplyvňuje výkon. Použitie tohto typu príliš veľa krát môže urobiť hru nehrateľnou, preto sa odporúča MeshCollider používať len zriedka.

Analýza

3.1 Existujúce riešenia

V tejto sekcii sú analyzované viaceré známe existujúce aplikácie, ktoré umožňujú používateľovi vytvoriť prezentácie z 3D modelov alebo zobraziť obsah v AR. Cieľom tejto rešerše je preskúmať už existujúce metódy riešenia tohto problému.

Skúmanie aplikácií bolo zamerané hlavne na interaktivitu, jednoduchosť používania a celkový dojem s používateľského rozhrania aplikácie, ale taktiež aj na podporu platforiem a cenu.

3.1.1 Microsoft PowerPoint

Cena: 3500 Kč

Platformy: MacOS, Windows, web

PowerPoint je jeden z najznámejších nástrojov na tvorbu prezentácií. Bol vydaný už v roku 1987, najprv len pre platformu Macintosh. Neskôr bol odkúpený spoločnosťou Microsoft a stal sa jedným z hlavných programov balíčku Office. [18]

Nástroj bol vytvorený na uľahčenie vizuálnej ukážky pre skupinové prezentácie, hlavne v podnikateľskom prostredí. Prezentácie sú usporiadané ako séria individuálne navrhnutých slajdov, ktoré obsahujú obrázky, text alebo iné objekty. [18] Na konci roku 2017 pridal Microsoft podporu aj pre 3D modely.

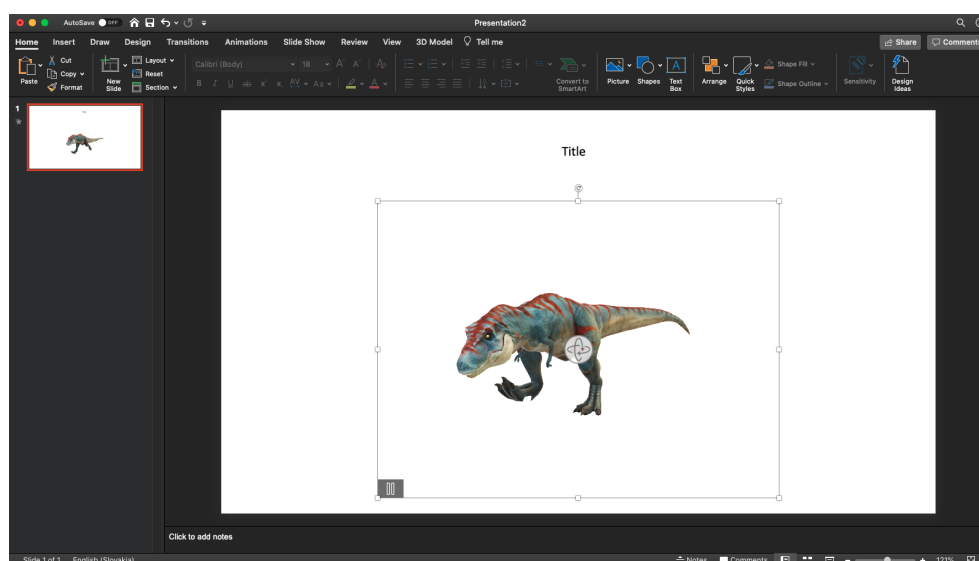
Keďže je tento nástroj primárne určený pre skupinové prezentácie, 3D model nie je možné ovládať interaktívne počas prezentácie. Používateľ vie model otočiť, zväčšiť a posunúť do polohy potrebnej pre prezentáciu, a taktiež mu nastaví predvolenú animáciu, ktorá sa pri spustení ukážky prehrá (napríklad otáčanie, poskočenie, zväčšenie, atď.).

PowerPoint je bohatý na funkcie. V jeho hornej časti sa nachádza 11 záložiek (ich počet však závisí na verzii a nastaveniach), kde sú v každej desiatky ďalších funkcionalít, každá väčšinou s vlastnou ikonou. To síce otvára používateľovi veľké spektrum možností, ale zase ho zahľucuje komplexitou.

3. ANALÝZA

V knihe [19] je ukázané, že UI bohaté na funkcie je efektívne pri vykonávaní rôznych zložitejších úloh, avšak zbytočne predlžuje výkon jednoduchších prác. Násť funkciu pridania 3D modelu môže trvať dlhšie aj pre skúsenejšieho používateľa.

Zaradenie tohto softvéru do rešerše je dôležité aj na ukážku, že 3D modely stávajú čím ďalej populárnejšie. PowerPoint v posledných rokoch pridal funkcionalitu na ich prezentáciu, avšak neposkytuje interaktivitu s používateľom. Ukážka pridania modelu v programe PowerPoint je na obrázku 3.1



Obr. 3.1: Prezentácia 3D modelu v programe PowerPoint

3.1.2 Aurora 3D Presentation

Cena: 2490 Kč

Platformy: MacOS, Windows

Aurora 3D Presentation je softvér na vytváranie 3D interaktívnych biznisových prezentácií. Nástroj je súčasťou rodiny produktov od Aurora3D Software, kde patria aj Aurora 3D Animation Maker, Aurora 3D Text & Logo Maker a Barcode Generator. Tieto programy tvoria spolu balíček, ktorý je vhodný na vytváranie dobrého dojmu pre klientov, či už pri väčších podujatiach alebo podnikateľských prezentácií.

Podľa stránky organizácie v [20] využíva Aurora 3D Presentation výkonných a vysoko efektívnych techník k vytváraniu pôsobivých, profesionálnych a presvedčivých prezentácií v rôznych formátoch. S nástrojom sa dá vytvárať bohaté a efektívne riešenia, ktoré kombinujú obrázky, text, videá a ostatné dáta takým spôsobom, že zaručene pritiahnu pozornosť publika.

Program ponúka veľké množstvo funkcionality. Podobne ako v programe PowerPoint prezentácia sa skladá z usporiadaných slajdov, ktoré obsahujú rôzny multimediálny obsah. Aurora 3D pridáva každému objektu polohu v 3D priestore. Používateľ si môže vytvárať pridávať vlastné tlačidlá a ich funkcionality, importovať 3D modely a meniť pohľad kamery pri rôznych akciách a prechodoch medzi slajdmi. Umožňuje to okrem klasických skupinových prezentácii vytvárať aj interaktívne.

Užívateľské rozhranie je takmer rovnaké ako v programe PowerPoint (opísaný v sekcii 3.1.1) - v ľavom paneli je rozvrhnutie častí prezentácie a v hornom paneli je väčšie množstvo ikoniek, ktoré majú rôznu funkcionality (napr. pridanie a vlastnosti textu, pridanie obrázkov, 3D modelu, a iné). Použitie tohto rozloženia UI je vhodné vďaka popularite programu PowerPoint - používatelia sú na to zvyknutí a je to pre nich prívetivejšie a uľahčuje to proces učenia.

Aurora 3D umožňuje vytvárať kvalitné prezentácie rôznych druhov. Ponúka veľké množstvo funkcionality, ktoré ale robí používateľské rozhranie komplexné a matúce. Verzia aplikácie v obchode App Store bola naposledy aktualizovaná v roku 2014 a taktiež UI programu a ukázkových projektov vypadajú zastaralo. S pridaním podpory 3D objektov do programu PowerPoint pridáva Aurora 3D až príliš málo funkcionality oproti najpopulárnejšiemu programu, ktorý je stále pravidelne aktualizovaný. Ukážka UI a tvorby prezentácie je na obrázku 3.2.



Obr. 3.2: Tvorba 3D prezentácie v nástroji Aurora 3D

3.1.3 Savree

Cena: 220-780 Kč mesačne (v závislosti od balíčku)

Platformy: Web

Spoločnosť Savree vznikla v roku 2017 a zameriava sa na produkciu kvalitných 3D modelov, animácií, videí a prezentácií, ktoré by vyplnili medzeru medzi pasívnou a aktívnou formou prezentácie. Sústreďujú sa hlavne na výučbu a výcvik pomocou interaktívnej prezentácie na webe a vo virtuálnej realite. [21]

Stránka produktu v [21] popisuje nasledovné hlavné funkcie:

- Interaktívne 3D modely - podporuje otáčanie, pohyb, priblíženie a od-dialenie modelu
- 3D animácie - prehrávanie animácií na modeli
- Rozloženie a zloženie modelu - špeciálna animácia, ktorá rozloží celý model na menšie časti, aby bolo vidno jeho zloženie
- Anotácie - interaktívne body, ktoré obsahujú text s popisom danej časti
- Pohľad prierezom modelu
- Kvízy - používateľ otestuje svoje znalosti tým, že odpovedá na otázky ohľadom daného zariadenia
- Virtuálna realita - 3D model sa dá prezeráť aj vo virtuálnej realite
- Prístup z internetového prehliadača - aplikácia beží priamo cez pre-hliadač a podporuje aj mobilné platformy

Webová stránka poskytuje knižnicu s veľkým množstvom 3D modelov. Každý môže (ale nemusí) podporovať niektoré z vyššie uvedených funkcií.

Používateľské rozhranie aplikácie je jednoduché. Aplikácia poskytuje dolný panel s hlavnými funkciami a prirodzené gestá pre ovládanie pohybu modelu. Prezentácie obsahujú interaktívne body ktoré obsahujú popis daných častí modelu.

Savree ale neposkytuje softvér na tvorbu týchto prezentácií. Tvorba 3D modelov, animácií a prezentácií k nim je internou súčasťou procesu, ktorú vykonáva samotná firma na základe návrhu zákazníka. Ten má prístup len k finálnym podobám modelov a prezentácií. Pridanie 3D modelu do aplikácie je tým pádom zdĺhavý proces, ktorý môže stáť aj peniaze navyše. Ukážka prezentácie v nástroji Savree je na obrázku 3.3.



Obr. 3.3: 3D prezentácia v nástroji Savree

3.1.4 Augment - 3D Augmented Reality

Cena: Zadarmo

Platformy: iOS, Android

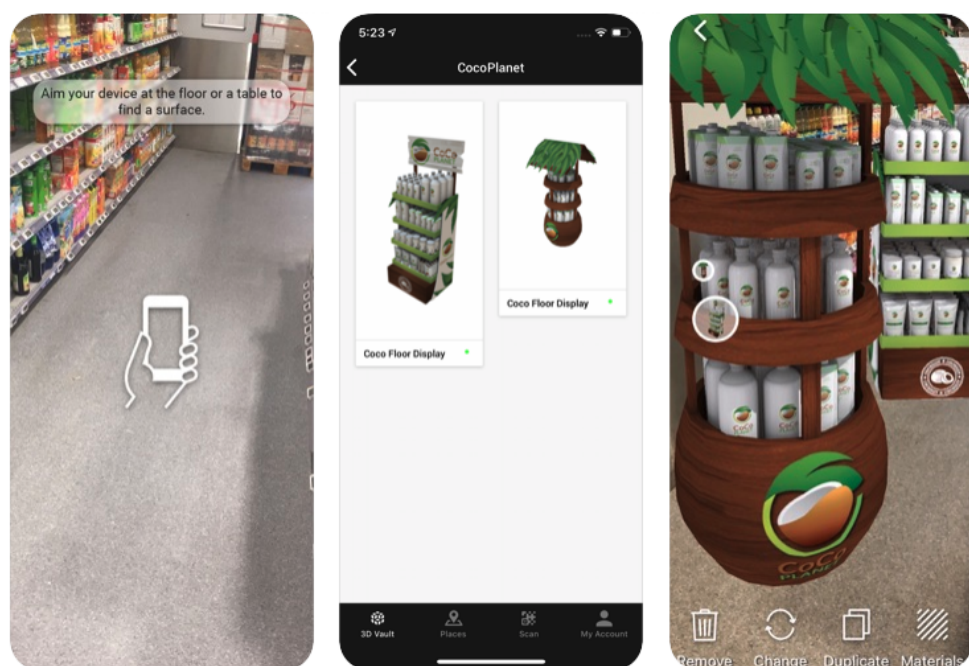
V obchodoch s aplikáciami App Store a Google Play existuje mnoho aplikácií na zobrazovanie 3D modelov na zariadení a prípadne aj v AR. Každé z nich poskytuje podobnú funkčnosť s malými rozdielmi. Do rešerše je zahrnutá ako príklad aplikácia Augment, ktorá je jednou z najpopulárnejších aplikácií tohto druhu.

Augment je spoločnosť ktorá sa zameriava na vizualizáciu produktov v 3D a rozšírenej realite. Poskytujú vlastnú aplikáciu pre zobrazovanie 3D modelov pomocou AR.

Aplikácia obsahuje verejnú knižnicu s množstvom 3D modelov. Používateľ má vlastnú knižnicu, kam si môže pridávať nové 3D modely cez webovú verziu aplikácie. Užívateľ si môže zvoliť model a následne ho vidieť cez zariadenia v priestore okolo seba pomocou rozšírenej reality.

Keďže aplikácia slúži len čisto na zobrazovanie v AR, používateľské rozhranie neobsahuje množstvo prvkov. Umožňuje používateľovi zapnúť svetlo pre lepšiu funkčnosť technológie AR a tiež vytvorenie fotografie s daným modelom v prostredí. Ukážka z aplikácie Augment je na obrázku 3.4.

3. ANALÝZA



Obr. 3.4: Aplikácia Augment (iOS verzia)

3.1.5 Zhrnutie

Zobrazovanie 3D modelov v aplikácii a v AR je už známa téma. Existuje mnoho aplikácií, ktoré túto funkcionality umožňujú. Pridávanie 3D modelu do prezentácie je už taktiež základom väčšiny známych nástrojov ako napríklad PowerPoint. Avšak existuje len málo aplikácií, ktoré by umožňovali kombináciu prezentácie s interaktivitou a jej tvorbu priamo na mobilnom zariadení. Cieľom tejto aplikácie by malo byť poskytnúť jednoduchší prístup pre tvorenie takéhoto obsahu a jeho prenos do rozšírenej reality.

3.2 Špecifikácia požiadaviek

V [22] sú požiadavky definované ako špecifikácia toho, čo má byť implementované. Existujú dva základné druhy požiadaviek:

- funkčné požiadavky - aké chovanie by mal systém ponúkať
- nefunkčné požiadavky - špecifické vlastnosti a obmedzenia pre systém

Požiadavky sú základ každého systému. Mali by hovoriť o tom, čo by mal daný systém robiť. Nemali by ale uvádzať ako by to malo byť dosiahnuté - spôsob dosiahnutia danej požiadavky by mal byť spomenutý len ak je to potrebné pre pochopenie.

3.2.1 Nefunkčné požiadavky

Táto časť obsahuje zhrnutie základných nefunkčných požiadaviek systému.

- **Multiplatformovosť**

Cieľom je vytvoriť aplikáciu, ktorá bude fungovať hlavne s podporou pre iOS, Android. Týmto spôsobom firmy nemusia nakupovať nové zariadenia a uľahčí im to proces prechodu k aplikácii. Navyše pri použití nástroja Unity je vývoj multiplatformových aplikácií jednoduchší.

- **Intuitívnosť používania**

Aplikácia by mala byť jednoduchá na pochopenie pre bežných používateľom mobilných a stolných zariadení. Mala by poskytovať prvky, na ktoré sú používatelia zvyknutí z iných známych aplikácií. Firmy sa týmto môžu vyhnúť zbytočne zložitému a drahému zaškoleniu predajcov.

- **Rýchlosť a jednoduchosť tvorby prezentácie**

Prezentácia sa dá vytvoriť rýchlo. Používateľ má k dispozícii pár esenciálnych nástrojov, ktoré sú potrebné k popisu produktu. Pár kliknutiami vie vytvoriť popisné body a pridať potrebné informácie bez zbytočného hľadania daných funkcií (oproti iným aplikáciám ako napríklad Aurora 3D presentation).

- **Unity**

Aplikácia má byť vyvinutá vo vývojom softvéri Unity. Hlavným dôvodom je moja vlastná preferencia a taktiež jednoduchosť vytvárania multiplatformových aplikácií v tomto nástroji.

3.2.2 Funkčné požiadavky

Funkčné požiadavky sú požiadavky, ktoré sú kladené na systém z pohľadu používateľa.

Zo zadania vyplývajú tieto funkčné požiadavky:

- **F01 - Prihlásenie a registrácia**

Pri prvom spustení aplikácie sa používateľ musí registrovať pomocou svojej emailovej adresy. Aplikácie neumožňuje pokračovať neregistrovaným používateľom.

- **F02 - Role a schvaľovanie registrácie**

Aplikácia má byť využiteľná v prostredí firiem, kde si môžu účet zakladať len zamestnanci. Na demonštráciu tohto využitia je potrebné implementovať správu účtov. Po registrácii potrebuje byť účet potvrdený administrátorom danej firmy. Po potvrdení administrátorom je umožnený používateľovi prístup do aplikácie.

- **F03 - Prehliadka produktov**

Používateľ má k dispozícii zoznam produktov, s ktorými je možné pracovať a prezentovať. Tieto produkty sú pridávané interne v rámci každej firmy a používateľ ďalšie produkty nepridáva.

- **F04 - Vytvorenie, úprava a mazanie prezentácií**

Hlavnou funkcionalitou systému je vytvorenie prezentácie a jej uloženie. Používateľ si môže kedykoľvek otvoriť 3D model produktu a vytvoriť k nemu interaktívnu prezentáciu. Existujúce prezentácie by malo byť možné upravovať a mazať.

- **F05 - Gestá pre prezeranie modelu**

Pri tvorbe a zobrazovaní prezentácie by mala aplikácia umožňovať používať gestá pre rotáciu, približovanie a odd'áľovanie 3D modelu produktu. Na základe nefunkčných požiadavkov by tieto gestá mali byť intuitívne - známe a používané v iných známych aplikáciách.

- **F06 - Vytvorenie, úprava a mazanie interaktívnych bodov v prezentácii**

Hlavnou časťou prezentácie sú interaktívne body. Sú to viditeľné prvky na 3D modeli produktu na ktoré môže používateľ kliknúť pre zistenie informácií o danej časti produktu. Do popisu bodu by malo byť možné pridávať okrem textu aj obrázkov, video a spustenie animácie modelu. Existujúce interaktívne body by malo byť možné upravovať a mazať pri upravovaní prezentácie.

- **F07 - Prehrávanie animácií modelu**

Používateľ bude mať k dispozícii animácie modelu (mimo interaktívneho bodu) a môže ich spúšťať kedykoľvek počas prezentácie.

- **F08 - Prezentácia ovládaná zákazníkom**

Aplikácia by mala umožňovať spustenie prezentácie. V tomto momente môže predajca nechať zariadenie zákazníkovi. Aplikácia bude umožňovať zvolenie modelu, následné spustenie prezentácie a prechod do AR. Nebude ale dovoliovať odhlásenie používateľa, tvorenie nových prezentácií alebo ich upravenie, mazanie verzií, tvorenie nových interaktívnych bodov alebo ich úpravu a mazanie.

- **F9 - Rozšírená realita**

Všetky 3D modely má používateľ možnosť vidieť aj pomocou rozšírenej (augmentovanej) reality. Pre zákazníka bude týmto spôsobom prezentácia oveľa zaujímavejšia a umožní mu to vidieť produkt v životnej veľkosti v prostredí okolo neho bez potreby prítomnosti reálneho produktu.

Návrh

Nasledujúca kapitola pokrýva témy, ktoré sú potrebné k detailnejšej špecifikácii chovania aplikácie a nasledovnej implementácii. Venuje sa analýze používateľov a ich spôsobom používania systému. Ďalej je v kapitole opísaný doménový model a diagram aktivít. Na základe toho je potom v poslednej sekcii opísaný návrh používateľského rozhrania.

4.1 Modelovanie prípadov použitia

Prípady použitia sú ďalšou technikou vytvárania požiadaviek systému. Pri tomto postupe sa v rámci systému vymedzia **aktéri** - role, ktoré majú ľudia alebo veci, ktoré interagujú so systémom. Každéj role sa potom priradia **prípady použitia** - akcie ktoré môžu aktéri robiť v rámci systému. [22]

4.1.1 Aktéri

Aktéri sú role ľudí alebo vecí, ktoré priamo interagujú so systémom. Môže to byť používateľ, hardware alebo aj externý systém. [22] V rámci tejto aplikácie môžeme identifikovať týchto aktérov:

- **Predajca**

Predajca môže byť napr. zamestnanec firmy, ktorá potrebuje prezentovať svoje produkty. Má vytvorený účet, pomocou ktorého sa prihlási, vidí všetky dostupné modely a môže vytvárať a spúšťať prezentácie pre zákazníkov.

- **Administrátor**

Administrátor je správca používateľov. Okrem funkcií predajcu má k dispozícii zoznam čakajúcich registrácií, ktoré môže schváliť alebo zamietnuť.

- **Zákazník**

Zákazník sa dostane k aplikácii v čase, keď je prihlásený predajca a zároveň je spustená prezentácia. Môže si prezerat' modely, ale nemôže vytvárať ani meniť prezentácie.

4.1.2 Prípady použitia

Prípady použitia sú v [22] definované ako špecifikácie akcií, ktoré systém, subsystém alebo trieda môže vykonávať pri interakcii s aktérmi. Je to niečo, čo chce aktér vykonať so systémom. Platia pri tom dve základné pravidlá:

- prípad použitia stále začína aktérom
- prípad použitia je stále napísaný z pohľadu aktéra

Najlepší postup ako identifikovať prípady použitia je uvažovať o tom, ako každý z aktérov bude systém používať. V rámci systému môžeme identifikovať tieto prípady použitia:

Zákazník

- **UC01 - Zobrazenie dostupných produktov**

Používateľ vidí zoznam 3D modelov produktov, ktoré vie otvoriť.

1. Predpoklad **UC09** (prihlásenie)
2. Používateľ si vie prezerat' modely potiahnutím prsta po zozname modelov. Zoznam by mal ukazovať aj náhľad modelu.

- **UC02 - Prezentácia 3D modelu**

1. Predpoklad **UC01**
2. Používateľ klikne na zvolený model
3. Načíta sa najaktuálnejšia prezentácia zvoleného modelu

- **UC03 - Otočenie 3D modelu**

1. Predpoklad **UC02**
2. Používateľ potiahne prstom po obrazovke
3. 3D model produktu sa otočí v danom smere podľa rýchlosti ťahu

- **UC04 - Priblíženie a oddialenie 3D modelu**

1. Predpoklad **UC02**
2. Používateľ priloží 2 prsty na obrazovku a potiahne ich od seba, resp. k sebe
3. 3D model sa zväčší, resp. zmenší podľa rýchlosti ťahu

- **UC05 - Zobrazenie interaktívneho bodu**

Používateľ je schopný vidieť obsah interaktívnych bodov.

1. Predpoklad **UC02**
2. Používateľ klikne na ikonu interaktívneho bodu na modeli
3. Zobrazí sa okno s obsahom interaktívneho bodu

- **UC06 - Spustenie animácií modelu**

1. Predpoklad **UC02**
2. Používateľ klikne na animáciu v ľavom menu
3. 3D model prezentácie prehrá animáciu

- **UC07 - Zobrazenie produktu v rozšírenej realite**

1. Predpoklad **UC02**
2. Používateľ klikne na tlačidlo AR v dolnej časti obrazovky
3. Aplikácia prejde do režimu kamery a zobrazí používateľovi pokyn na nájdenie plochy
4. Používateľ hýbe mobil po okolí až kým sa mu nezobrazí nájdená plocha
5. Používateľ klikne na bod, kde chce model umiestniť
6. Model sa umiestni na dané miesto v priestore aj s interaktívnymi bodmi

Predajca

Predajca môže používať aplikáciu rovnakým spôsobom ako zákazník, ale navyše má tieto prípady použitia:

- **UC08 - Registrácia**

Pri prvom spustení aplikácie sa používateľ musí registrovať, aby mohol systém používať. Aplikácia neumožňuje pokračovať neregistrovaným používateľom.

1. Používateľ spustí aplikáciu
2. Zobrazí sa úvodná prihlasovacia obrazovka
3. Používateľ klikne na *Register*
4. Zobrazí sa registračná obrazovka
5. Používateľ vyplní údaje a potvrdí registráciu

- **UC09 - Prihlásenie**

1. Používateľ spustí aplikáciu
2. Zobrazí sa úvodná prihlasovacia obrazovka
3. Používateľ vyplní svoje údaje a potvrdí prihlásenie
4. Zobrazí sa hlavné menu aplikácie

- **UC10 - Odhlásenie**

1. Predpoklad **UC09**
2. Používateľ klikne na ikonku profilu
3. Zobrazí sa menu s možnosťami
4. Používateľ klikne na možnosť *logout*

- **UC11 - Prechod do prezentačného režimu**

Prezentačný režim by mal byť režim aplikácie, kde predajca môže nechať zariadenie zákazníkovi. Zákazník má obmedzené funkcie - mód slúži len na prezeranie, nemôže nič meniť.

1. Predpoklad **UC09**
2. Používateľ klikne ikonu profilu
3. Zobrazí sa menu s možnosťami
4. Používateľ klikne na možnosť prepnutia do prezentačného módu
5. Menu sa zavrie a aplikácia od tohto momentu nedovolí používateľovi vytvárať, editovať, mazať prezentácie a ani sa odhlásiť.

- **UC12 - Prechod späť do editačného režimu**

Prechod naspäť do prezentačného režimu musí sprevádzať vyžiadanie hesla, inak by odovzdanie zariadenia mohlo byť riskantné - používateľ by sa mohol preklikať až k editačnému módu.

1. Predpoklad **UC11**
2. Používateľ klikne na ikonu profilu
3. Zobrazí sa menu s možnosťami
4. Používateľ klikne na možnosť prepnutia do editačného módu
5. Vytvorí sa okno s políčkou na vyplnenie hesla
6. Používateľ vyplní heslo prihláseného účtu
7. Okno sa zavrie a používateľovi sa vrátia naspäť právomoci vytvárania, editácie, mazania prezentácií a odhlásenia

- **UC13 - Zobrazenie vytvorených prezentácií k 3D modelu**

1. Predpoklad **UC01**
2. Používateľ klikne na vybraný 3D model
3. Zobrazí sa zoznam vytvorených prezentácií

- **UC14 - Vytvorenie novej prezentácie modelu**

1. Predpoklad **UC13**
2. Používateľ klikne na tlačidlo vytvoriť novú prezentáciu
3. Zobrazí sa nová prázdna prezentácia so zvoleným 3D modelom

- **UC15 - Upravenie prezentácie**
 1. Predpoklad **UC13**
 2. Používateľ klikne na riadok s vybraným názvom prezentácie
 3. Načíta sa zvolená prezentácia daného modelu
- **UC16 - Zmazanie prezentácie**
 1. Predpoklad **UC13**
 2. Používateľ klikne na ikonu odpadkového koša v riadku s danou prezentáciou
 3. Zobrazí sa potvrdzovacie okno
 4. Používateľ potvrdí zmazanie prezentácie
 5. Okno zmizne a daná prezentácia sa vymaže zo zoznamu prezentácií
- **UC17 - Vytvorenie interaktívneho bodu**

Je dôležité aby vytváranie bodu bolo čo najjednoduchšie. Kliknutie na dané miesto, kde sa aplikácia postará o to, aby bol bod správne umiestnený na modeli je pre používateľa prívetivé.

 1. Predpoklad **UC14** alebo **UC15**
 2. Používateľ sa dotkne prstom tlačidla *Create interaction point*
 3. Používateľ si vyberie miesto na modeli, kde chce vložiť interaktívny bod a potom sa na miesto dotkne prstom
 4. Na modeli produktu pribudne grafika interaktívneho bodu a otvorí sa editácia prázdneho interaktívneho bodu
- **UC18 - Upravenie existujúceho interaktívneho bodu**
 1. Predpoklad **UC14** alebo **UC15**
 2. Používateľ klikne na interaktívny bod
 3. Zobrazí sa obsah interaktívneho bodu
- **UC19 - Zmazanie interaktívneho bodu**
 1. Predpoklad **UC17** alebo **UC18**
 2. Používateľ klikne na ikonu zmazania interaktívneho bodu
 3. Interaktívny bod zmizne a zmaže sa z prezentácie
- **UC20 - Pridanie položky do interaktívneho bodu**
 1. Predpoklad **UC17** alebo **UC18**
 2. Používateľ klikne na tlačidlo pridania bodu
 3. Zobrazia sa ikony pre pridanie videa, obrázku, animácie a textu
 4. Používateľ klikne na obsah ktorý chce pridať

- **UC21 - Pridanie videa**
 1. Predpoklad **UC20**
 2. Zobrazí sa okno, ktoré umožňuje používateľovi vybrať video z jeho galérie na zariadení
 3. Používateľ zvolí video
 4. Video sa pridá medzi položky interaktívneho bodu
- **UC22 - Pridanie obrázku**
 1. Predpoklad **UC20**
 2. Zobrazí sa okno, ktoré umožňuje používateľovi vybrať obrázok z jeho galérie na zariadení
 3. Používateľ zvolí obrázok
 4. Obrázok sa pridá medzi položky interaktívneho bodu
- **UC23 - Pridanie textu**
 1. Predpoklad **UC20**
 2. Položka textu sa pridá do zoznamu a na zariadení sa otvorí klávesnica pre vpísanie obsahu
 3. Po napísaní textu používateľ potvrdí obsah
- **UC24 - Pridanie animácie**
 1. Predpoklad **UC20**
 2. Zobrazí sa okno s dostupnými animáciami
 3. Používateľ klikne na položku
 4. Vytvorí sa položka animácie s ikonou na spustenie animácie
- **UC25 - Zmazanie položky interaktívneho bodu**
 1. Predpoklad **UC17** alebo **UC18**
 2. Používateľ klikne na ikonku zmazanie pri položke
 3. Položka zmizne z obsahu interaktívneho bodu

Admin

Admin je vyššia funkcia v rámci spoločnosti, ktorá spravuje predajcov a produkty, ktoré daná spoločnosť prezentuje. Okrem prípadov použitia predajcu a zákazník má admin navyše nasledujúce položky:

- **UC26 - Potvrdenie používateľa**
 1. Predpoklad **UC09**
 2. Administrátor zvolí ikonu profilu
 3. Zobrazí sa menu s možnosťami
 4. Administrátor klikne na možnosť *manage users*

5. Zobrazí sa zoznam používateľov
 6. Administrátor klikne na ikonku potvrdenia pri zvolenom používateľovi
- **UC27 - Zamietnutie registrácie používateľa**
 1. Predpoklad **UC09**
 2. Administrátor zvolí ikonu profilu
 3. Zobrazí sa menu s možnosťami
 4. Administrátor klikne na možnosť *manage users*
 5. Zobrazí sa zoznam používateľov
 6. Administrátor klikne na ikonku zamietnutia pri zvolenom používateľovi

Je dôležité skontrolovať, či prípady použitia pokrývajú všetky funkčné požiadavky. Toto pokrytie je ukázané v tabuľke 4.1.

4.2 Doménový model tried

Modelovanie domény je spôsob ako opísať entity z reálneho sveta a vzťahy medzi nimi. Definovanie entít a ich vzťahov, ktoré by malo vychádzať z požiadaviek aplikácie, ponúka efektívny základ pre pochopenie problémovej domény, čo môže pomôcť pri ďalšom vývoji, testovaní a pri zachovaní udržateľnosti aplikácie. [23]

Doménový model sa nachádza na obrázku 4.1 a sú v ňom nasledujúce entity:

- **Company (spoločnosť)**

Reprezentuje firmu, ktorá je v aplikácii registrovaná. Spoločnosť má v aplikácii svoje produkty (ich 3D modely) a svojich zamestnancov, ktorí môžu k daným modelom pristupovať.

- **User (používateľ)**

Je to akákoľvek osoba, ktorá je v aplikácii registrovaná. Každý používateľ musí mať spoločnosť, ktorá ho pokrýva a taktiež definovanú rolu (pole *Role*), ktorá určuje jeho právomoci v rámci systému a firmy.

- **ProductModel (3D model produktu)**

Reprezentuje počítačový 3D model produktu firmy.

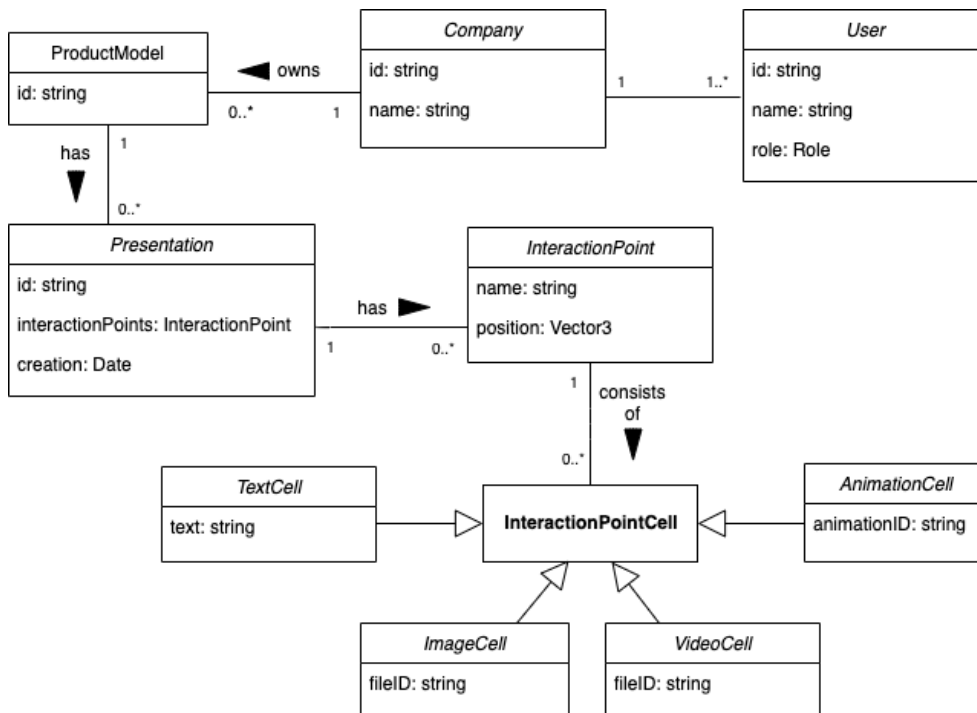
- **Presentation (prezentácia)**

Ku každému modelu môže byť vytvorených viacero prezentácií. Prezentácia je definovaná interaktívnymi bodmi, ktoré sú pridané k modelu.

4. NÁVRH

	F01	F02	F03	F04	F05	F06	F07	F8	F9
UC01			■						
UC02			■						
UC03					■				
UC04					■				
UC05						■			
UC06							■		
UC07									■
UC08	■								
UC09	■								
UC10	■								
UC11								■	
UC12								■	
UC13				■					
UC14				■					
UC15				■					
UC16				■					
UC17						■			
UC18						■			
UC19						■			
UC20						■			
UC21						■			
UC22						■			
UC23						■			
UC24						■			
UC25						■			
UC26		■							
UC27		■							

Tabuľka 4.1: Pokrytie funkčných požiadaviek



Obr. 4.1: Doménový model tried

- **InteractionPoint (interaktívny bod)**

Interaktívny bod je základná časť prezentácie. Obsahuje popisné informácie k danej časti modelu, ktoré sa v ňom nachádzajú v podobe buniek.

- **InteractionPointCell (bunka interaktívneho bodu)**

Bunka interaktívneho bodu môže byť buď text, obrázok, video alebo spustenie animácie na modeli. Interaktívny bod sa môže skladať z viacerých buniek umiestnených za sebou.

4.3 Používateľské rozhranie

Používateľské rozhranie (*anglicky User Interface, skratene UI*) je dôležitá časť návrhu aplikácie. Dobre navrhnuté rozhranie zlepšuje efektívnosť a rýchlosť používania aplikácie a taktiež zlepšuje dojem, ktorý aplikácia zanechá na používateľovi.

Pri návrhu používateľského rozhrania ale nejde len o grafické prvky a ich vzhľad. UI je vrstva, v ktorej sa vykonáva interakcia medzi človekom a nástrojom, ktorý používa. Cieľom je umožniť efektívne využitie nástroja

človekom a zároveň vytvárať takú spätnú väzbu pre používateľa, ktorá podporí jeho rozhodovací proces.

Avšak definovať, ako vypadá ideálne používateľské rozhranie, je zložité, pretože každý človek vníma a používa aplikáciu iným spôsobom. Je preto dôležité držať sa už zaužívaných spôsobov, na ktoré sú ľudia zvyknutí z iných aplikácií a nástrojov.

4.3.1 Princípy dizajnu používateľského rozhrania

Jakob Nielsen, významný odborník v oblasti UI, vytvoril vo svojej publikácii [24] na uľahčenie tohto procesu desať zásad pre dizajn používateľského rozhrania. Avšak nie sú to špecifické pokyny, ale slúžia len ako pomôcka pri dizajne aplikácie.

1. Viditeľnosť stavu systému

Používateľ by mal byť vždy informovaný o tom, čo sa v systéme deje - pomocou zmysluplnej spätnej väzby v rozumnom čase.

2. Zhoda medzi systémom a realitou

Systém by mal komunikovať s používateľom cez frázy, slová a koncepty, ktoré už používateľ pozná. Pri zobrazovaní informácií by sa mali dodržiavať známe a používané pravidlá z reálneho sveta.

3. Minimálna zodpovednosť a sloboda voľby

Používateľ si môže náhodou zvoliť nesprávnu funkciu a malo by mu byť umožnené chybu jednoducho napraviť. Systém by mal podporovať funkciu "krok späť" všade, kde je to možné.

4. Konzistencia a zhoda s použitou platformou

Systém by mal dodržiavať rovnaké dizajnové prvky a postupy v celej aplikácii a zároveň rešpektovať konvencie danej platformy. Tým by mal vytvoriť konzistentné prostredie pre používateľa.

5. Prevencia chýb

Lepšie ako dobré chybové správy je dizajn systému, ktorý zabráni aby k problému vôbec došlo. Buď je treba sa vyhnúť stavu, v ktorom môže dojsť ku chybe, alebo by mal byť používateľ upozornený ešte predtým ako sa daná akcia vykoná.

6. Pozriem a vidím

Používateľ by nemal byť nútený pamätať si informácie z predošlých obrazoviek alebo hlásení. Informácie by mali byť jednoducho dostupné a viditeľné.

7. Flexibilita a efektivita používania

Akcelerátory sú prvky, ktoré umožňujú rýchlejšiu efektivitu interakcie pre skúsenejších používateľov. Noví a neskúsení používatelia o týchto prvkoch vedieť nemusia a ani nepotrebujú. Systém by tiež mohol umožňovať používateľom prispôbiť si systém tak, aby boli často vykonávané akcie prístupnejšie.

8. Estetický a minimalistický dizajn

Dialógy by nemali obsahovať nadbytočné informácie, ktoré sú irelevantné alebo zriedka potrebné. Nadbytočné informácie znižujú relatívnu viditeľnosť podstatným informáciám, ktoré sú pre používateľa dôležitejšie.

9. Pomoc používateľom pri rozpoznaní, diagnostike a opravení chýb

Chybové hlásenia by mali byť napísané prostým jazykom, presne a zrozumiteľne opísať problém a navrhnúť riešenie situácie.

10. Nápoveda a dokumentácia

Vždy je lepšie, ak je systém použiteľný aj bez dokumentácie a nápovedy, avšak niekedy to môže byť nevyhnutné. Informácie by mali byť ľahko dohľadateľné, stručné a mali by byť zamerané na úkony používateľa.

4.3.2 Analýza úloh

Úlohy sú podobné ako prípady použitia s tým rozdielom, že berú akcie viac z pohľadu používateľa. Rozdeľujú väčšie procesy na menšie časti a snažia sa ich logicky prepojiť a tým uľahčiť proces ďalšieho dizajnu. Identifikácia úloh, ktoré musí aplikácia podporovať je dôležité pre dosiahnutie cieľov s ktorými bola aplikácia navrhnutá. [25]

Úvodná obrazovka

Vzhľadom na viditeľnosť stavu systému, bolo rozhodnuté pridať po prihlásení používateľa výber režimu aplikácie (prezentačný alebo editačný). Tým pádom je od začiatku jasné, v akom režime sa používateľ nachádza a že sa to dá zmeniť.

- Zobrazíť prihlasovací formulár
- Prihlásenie
- Zobrazíť registračný formulár
- Registrácia
- Výber počiatočného módu

Hlavná obrazovka

Hlavná obrazovka poskytuje náhľad dostupných modelov a prezentácií a taktiež otvorenie profilového menu.

- Zobrazit' dostupné modely
- Vybrať model
- Zobrazit' zoznam prezentácií
- Vytvorenie novej prezentácie
- Zmazanie prezentácie
- Úprava prezentácie
- Zobrazit' profilové menu

Profilové menu

Obsahuje všetko čo súvisí s používateľom a chodom aplikácie.

- Odhlásenie
- Zmena módu
- Správa používateľov
- Vypnutie aplikácie

Prezentácia

Pokrýva prípady použitia pri tvorení, úprave a zobrazovaní prezentácií.

- Vytvorenie interaktívneho bodu
- Úprava interaktívneho bodu
- Zobrazenie interaktívneho bodu
- Pridanie položky v bode
- Vymazanie položky v bode
- Zavretie interaktívneho bodu
- Zmazanie interaktívneho bodu
- Prechod do AR
- Prehranie animácie
- Uloženie
- Ukončenie prezentácie

Úlohy boli rozdelené do 4 častí podľa funkcionality a logického poradia. Ako užitočné je podľa [25] zostaviť si graf týchto úloh. Graf úloh nie je štandardizovaný UML diagram, ale vie dobre ukázať závislosti medzi úlohami pre ďalší dizajn drôtových modelov. Graf úloh sa nachádza v prílohe B.

4.3.3 Wireframe

Wireframe (alebo drôtový model) slúži na zobrazenie používateľského rozhrania aplikácie v skorších štádiách vývoja. Primárne zameranie sa kladie na umiestnenie elementov, dostupnú funkčnosť a zamýšľané správanie. Aj preto drôtové modely neobsahujú farby a grafiku. Môžu mať podobu rukou nakreslených obrázkov, ale existuje aj mnoho nástrojov na tvorbu wireframe na počítači.

V modeloch sú obrazovky rozdelené podľa grafu úloh, kde každá umožňuje používateľovi danú funkčnosť. Wireframey k aplikácii sú priložené v prílohe C.

Implementácia

5.1 Použité nástroje a knižnice

V tejto kapitole sú opísané použité nástroje a knižnice pri implementácii aplikácie.

5.1.1 Unity Editor

Hlavným vývojovým nástrojom je Unity, ktoré je bližšie opísané v sekcii 2.2. Unity Editor poskytuje prostredie pre prácu s projektom - vytváranie a úprava scén, tvorba objektov, pridávanie assetov, komponent a ďalšie.

5.1.2 C# a JetBrains Rider

V súčasnosti jediným plne podporovaným programovacím jazykom v Unity je C#. C# je vysokoúrovňový objektovo orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft.

JetBrains je medzinárodná firma, ktorá ponúka širokú škálu vývojových prostredí (skrátene *IDE* - z anglického *Integrated Development Environment*) pre rôzne jazyky. Ich verzia Rider pre vývoj v jazyku C# je kvalitné vývojové prostredie, ktoré je kompatibilné s Unity. Ponúka veľa funkcionality a hlavne podporuje ladenie (*debug*) kódu s tým, že sa IDE pripojí priamo na proces v Unity.

5.1.3 Git

Git je distribuovaný systém pre správu verzií. V súčasnosti najpopulárnejší systém tohto typu je esenciálnou súčasťou každého väčšieho projektu. Systém zaznamenáva zmeny v súboroch počas vývoja a tým sa umožňuje vracieť k starým verziám implementácie. Roky boli štandardom centralizované systémy, kde boli všetky data na centrálnom serveri a prístup sa riešil cez zamykanie (ako napríklad v Subversion a CVS). Oproti tomu, v distribuovaných

systémoch klienti kopírujú celý repozitár s históriou, tým pádom každý vývojár má k dispozícii vlastnú kópiu a pracuje nezávisle. Komplikácie nastávajú pri kolíziách zmien - vtedy musí jeden z vývojárov spraviť merge - definovať, ktoré časti zmien sa zachovajú a ktoré zahodia. [26]

5.1.4 Native Gallery

Native Gallery je knižnica pre Unity. Pomáha s ukladaním a načítaním obrázkov a videí zo zariadení, ktoré používajú Android a iOS. Uľahčuje prácu pri vytváraní multiplatformových aplikácií.

5.1.5 AR Foundation

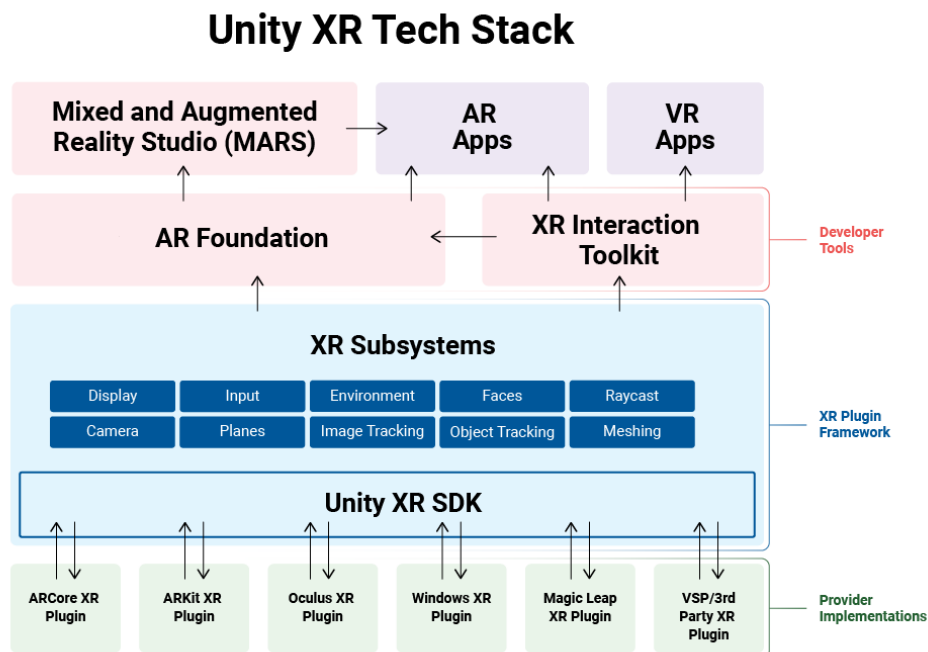
AR Foundation je oficiálny balíček od Unity, ktorý umožňuje prácu s rozšírenou realitou v Unity multiplatformovo. Využíva pri tom už existujúce SDK pre zariadenia ako ARKit a ARCore (viac je o nich opísané v sekcii 2.1). Knižnica poskytuje jednotné rozhranie, ktoré pomáha pracovať so zariadeniami podporujúcimi nasledovné koncepty [27]:

- Vnímanie okolia - zaznamenávanie pozície a rotácie zariadenia v priestore
- Detekcia roviny - detegovanie horizontálnych a vertikálnych povrchov
- Point cloud - množina bodov v priestore, ktoré sú väčšinou detegované 3D skenermi
- Referenčné body - ľubovoľná pozícia a orientácia, ktorú zariadenie monitoruje
- Odhad osvetlenia - odhady pre teplotu farby a jas v priestore
- Skúmanie prostredia (angl. *environment probes*) - dokáže vytvoriť reprezentáciu časti prostredia cez tzv. *cube mapping* - textúra uložená ako šesť stien kocky
- Detegovanie a sledovanie ľudskej tváre
- Detegovanie 2D obrázkov v priestore
- Detegovanie objektov

AR Foundation je vytvorená nad subsystémami. Subsystém je platformovo nezávislé rozhranie, ktoré sa stará o nejakú špecifickú funkcionálnosť systému. Napríklad rozhranie na detekciu roviny poskytuje *XRPlaneSubsystem*.

Konkrétnu implementáciu subsystemu obsahuje poskytovateľ (angl. *provider*). Aj keď je na každom poskytovateľovi, ako implementuje danú funkcionálnosť, väčšinou je to zaobalenie funkcionality natívnej knižnice pre danú platformu (napríklad ARKit pre iOS alebo ARCore pre Android). [27]

Celá systém sa nazýva *XR Plug-in Framework* (XR je skratka z angl. *extended reality* - výraz popisujúci rozšírenú aj virtuálnu realitu a ich kombinácie).



Obr. 5.1: Diagram ukazujúci štruktúru XR framework-u v Unity, a ako fungovanie s rôznymi poskytovateľmi implementácií [28]

5.1.6 Firebase Authentication a Firebase Database

Pre serverovú časť bola vybraná Firebase. Ponúka rýchle a efektívne riešenie, ktoré je ľahko integrovateľné do Unity projektu. Firebase Authentication a Firebase Database sú SDK, ktoré sú potrebné pre prácu s backend serverom Firebase. Každé z nich ponúka inú funkcionálnosť.

Authentication sa stará o správu účtov a umožňuje registráciu a prihlasovanie pomocou emailovej adresy, telefónneho čísla a rôznych iných účtov tretích strán ako Facebook, Google, atď.

Database je moderná NoSQL cloud databáza. Firebase poskytuje natívne SDK pre viacero platforiem a programovacích jazykov, vrátane Unity. Umožňuje to rýchlu a bezpečnú komunikáciu so serverovou časťou. [29]

5.2 Firebase server

Autentifikačná časť Firebase sa stará o vytváranie používateľov a generovanie ich identifikátoru. Takisto kontroluje prihlasovanie a odhlasovanie.

Databázová časť Firebase je veľmi jednoduchá. Na serveri sa ukladajú len 3 údaje o používateľoch - meno a údaje o schválení a roli používateľa.

O to sa stará kolekcia *users*. Každý používateľ má vlastný dokument s kľúčom podľa jeho identifikátora. Dokument používateľa pozostáva z mena, role a štádia procesu schvaľovania. Štruktúra vypadá to nasledovne:

```
users
├── <userId>
│   ├── approved
│   ├── type
│   └── name
```

5.3 Unity projekt

Projekt v Unity sa skladá z troch hlavných scén. Prvá scéna sa stará o prihlásenie a registráciu, ďalšia obsahuje zoznam produktov a ich prezentácií, a posledná scéna je prezentácia modelu, jej úprava a tiež prechod do rozšírenej reality.

Unity neobsahuje možnosť načítať scénu s parametrami a to komplikuje komunikáciu medzi jednotlivými scénami, pretože pri ich zániknutí sa vždy zničia všetky existujúce objekty. Na prenášanie informácií, ktoré sú potrebné (napríklad o prihlásenom používateľovi a prezentácii, ktorú je treba otvoriť) je vytvorená trieda *AppState* pomocou návrhového vzoru *singleton*, ktorý sa stará o to, aby stále existovala jedna inštancia triedy s aktuálnymi informáciami.

5.3.1 Prihlasovacia scéna

Táto scéna obsahuje funkcionality na prihlasovanie, registráciu a logiku prechodu medzi nimi. Použité boli natívne komponenty Unity - *InputField* pre zadávanie údajov a *Button* pre tlačidlá. Ukážka UI je v prílohe na obrázku D.1.

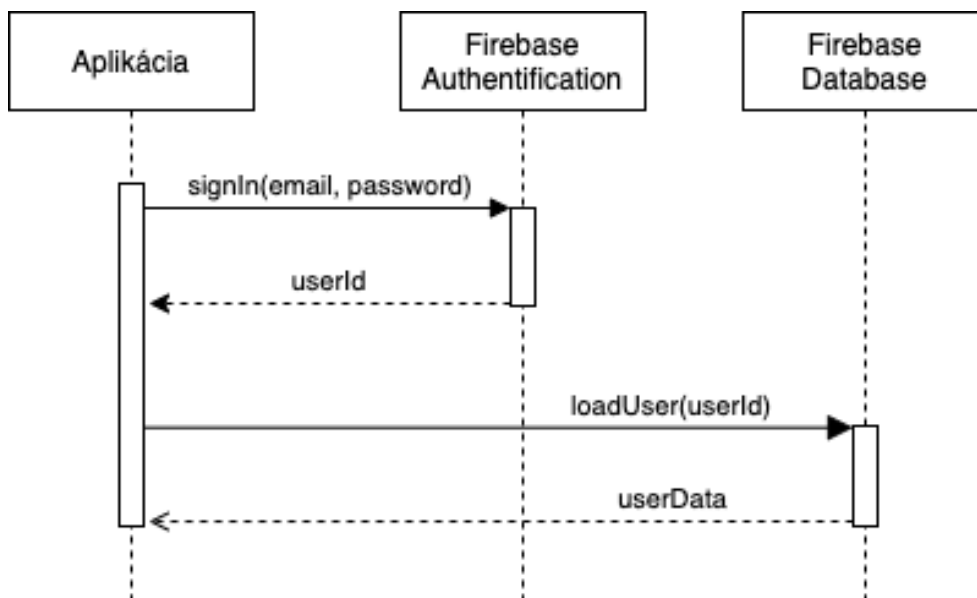
Prihlasovanie a registrácia prebieha cez Firebase Authentication SDK pomocou príkazov:

```

FirebaseAuth.DefaultInstance.SignInWithEmailAndPassword
Async(string email, string password);
FirebaseAuth.DefaultInstance.CreateUserWithEmailAndPass
wordAsync(string email, string password);

```

Po úspešnom prihlásení vráti server odpoveď s identifikátorom používateľa - *user ID*. User ID sa následne použije na dotazovanie do databázovej časti, z kadiaľ aplikácia zistí, aké právomoci má používateľ.



Obr. 5.2: Proces prihlasovania cez Firebase

Ak sa používateľ prihlási a jeho právomoci mu umožňujú vytvárať, mazať a upravovať prezentácie, aplikácia mu dá na výber z dvoch módov - editačný a prezentačný, a podľa jeho výberu načíta ďalšiu scénu. Ak má používateľ právomoci len na prezentáciu, aplikácia načíta ďalšiu scénu automaticky v prezentačnom móde.

5.3.2 Scéna hlavného menu

Scéna obsahuje 3 základné prvky:

- **Horný panel** obsahuje informáciu, v akom móde je aplikácia spustená (prezentačný alebo editačný mód) a taktiež obsahuje tlačidlo profilu, ktoré umožňuje používateľovi odhlásiť sa, vypnúť aplikáciu a taktiež spravovať používateľov a zmeniť mód - ak to používateľovi umožňujú právomoci.
- **Zoznam 3D modelov** je horizontálny posuvný zoznam. Vytvorený je pomocou Unity komponenty *ScrollView* v kombinácii s komponentami *ContentSizeFitter* a *VerticalLayoutGroup*, ktoré umožňujú správne rozloženie položiek zoznamu. Položka pozostáva z obrázku a mena produktu.
- **Zoznam prezentácií** po kliknutí na produkt sa pod položkou vytvorí zoznam prezentácií s možnosťou vymazania a vytvorenia novej prezentácie. Kliknutím na položku prezentácie alebo na tlačidlo vytvorenia novej prezentácie sa načíta scéna prezentácie modelu.

Ukážka obrazovky hlavného menu je v prílohe na obrázku D.4.

5.3.3 Scéna prezentácie

Grafické používateľské rozhranie je ukázané v prílohe na obrázku D.7 a pozostáva z nasledujúcich prvkov:

- **Zoznam animácií** - vľavo sa nachádza zoznam animácií, ktoré model obsahuje. Po kliknutí sa animácia prehrá. Keďže zoznam zaberá väčšiu časť obrazovky ako ostatné prvky, dá sa skryť kliknutím na ikonu animácie.
- **Prechod medzi AR a 3D** - v dolnej časti sa nachádza prepínač medzi rozšírenou realitou a klasickým 3D pohľadom.
- **Prednastavené pohľady** - umožňuje resetovať pohľad na 3D model - model sa otočí do zvolenej polohy. Predvolené pozície sú pohľad zhora, z boku a spredu. Aj keď táto funkcionálna nebola pôvodne v návrhu, pri vývoji sa pridala pre zjednodušenie pohľadu na model.
- **Vytvorenie interaktívneho bodu** - vykonáva sa tlačidlom vpravo dole. Je väčšia ako ostatné, pretože ide o hlavnú funkcionálnu obrazovky. Viac o riešení vytvárania interaktívnych bodov je v sekcii 5.5.
- **Uloženie** - uloží prezentáciu na zariadenie. Po kliknutí sa zobrazí okno, kde používateľ zadá meno prezentácie.

- **Ukončenie** - ukončí prezentáciu a vráti sa do hlavného menu.

Ukladanie a načítavanie prezentácií k 3D modelu sa rieši v rámci zariadenia. Prezentácie sa ukladajú do priečinku na zariadení, ktorý je určený pre aplikáciu. Priečinok má iné umiestnenie a názov pri každej platforme a Unity poskytuje premennú ktorá tento problém rieši za vývojára:

`Application.persistentDataPath`

Následne je určená adresárová štruktúra, kde sa dáta prezentácie (JSON súbory s obrázkami a videami k nim) ukladajú v nasledovnej hierarchii priečinkov:

```
Models
├── <modelID>
│   ├── <presentationName>
│   │   ├── interactionPoints.json
│   │   ├── image1-usedInInteractionPoints.png
│   │   ├── image2-usedInInteractionPoints.png
│   │   └── video1-usedInInteractionPoints.mp4
```

Štruktúra JSON súboru pre uloženie informácií o interaktívnych bodoch:

```
detailsData ..... objekt
├── position ..... objekt
│   ├── x ..... float
│   ├── y ..... float
│   └── z ..... float
├── title ..... string
└── cells ..... pole
    ├── type ..... string
    ├── textData ..... string(optional)
    ├── imageName ..... string(optional)
    ├── videoName ..... string(optional)
    └── animationId ..... string(optional)
```

5.4 Používateľský vstup a gestá

V aplikácii sa používajú na interakciu s 3D modelom nasledujúce gestá:

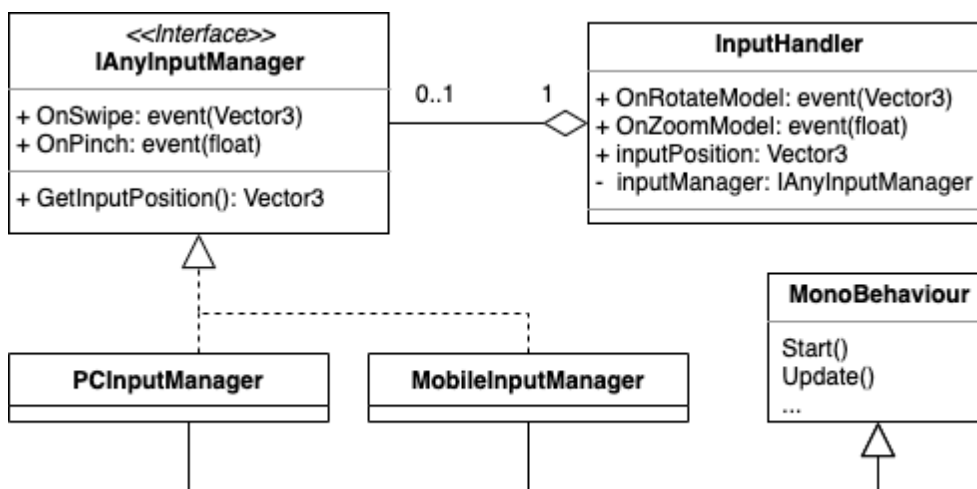
- **Drag** (rolovanie) - pohyb prstu po ploche displeja bez prerušenia kontaktu. Gesto sa väčšinou používa na listovanie dlhšieho obsahu, ktoré presahuje obrazovku. Keďže v prezentačnej časti aplikácie nie je žiadna komponenta, ktorá by sa mohla prekryvať s týmto gestom, jednoduchý drag sa dá využiť na rotáciu 3D modelu.

- **Pinch** (zväčšovanie) - dotyk obrazovky dvoma prstami a ich následný pohyb k sebe. Gesto sa najviac využíva pri prezeraní dokumentov, obrázkov alebo mapy, kde si používateľ týmto gestom dokáže zväčšiť menej viditeľný obsah. [30] Toto gesto je tým pádom prirodzené pre bežného používateľa mobilného telefónu na zväčšovanie 3D modelu.
- **Spread** (zmenšovanie) - dotyk obrazovky dvoma prstami a ich následné oddialenie od seba. Gest slúži ako opačné gesto ku zväčšovaniu a teda obsah zmenšuje.

Užívateľský vstup bol zameraný hlavne na mobilné zariadenia, ale na testovanie pri vývoji a tiež možné rozšírenia aplikácie v budúcnosti bolo implementované detegovanie gest aj pre počítače.

Unity poskytuje triedu *Input* pre prácu s používateľským vstupom pre počítače aj mobilné telefóny. Trieda obsahuje funkcie na zistenie polohy a počtu dotykov obrazovky. Riešenie reakcie na vstup by ale malo byť v pri každom snímku (pri zavolaní funkcie *Update*), čo pridáva na komplexite. V práci je teda rozdelený vstup na 2 triedy, kde každá rieši vstup pre inú platformu - štruktúra je ukázaná v diagrame na obrázku 5.3. Hlavná trieda (*InputHandler*) vytvorí na začiatku triedu pre danú platformu a následne reaguje na udalosti, ktoré sú triedou vyvolané. Zistenie platformy prebieha cez preddefinované direktívy Unity, ktoré sú vyhodnocované pri kompilácii:

```
#if UNITY_EDITOR || UNITY_STANDALONE_WIN || UNITY_STANDALONE_OSX
    inputManager = gameObject.AddComponent<PCInputManager>();
#elif UNITY_IOS || UNITY_ANDROID
    inputManager = gameObject.AddComponent<MobileInputManager>();
#endif
```



Obr. 5.3: Diagram tried vstupu

5.5 Tvorba interaktívneho bodu

Proces tvorby interaktívneho bodu je opísaný v sekcii prípadov použitia 4.1.2 pri prípade **UC17**. Používateľ sa dotkne tlačidla a následne sa dotkne miesta na modeli, kde chce vytvoriť bod. V mieste dotyku na najbližšom bode 3D modelu sa následne vytvorí grafika bodu.

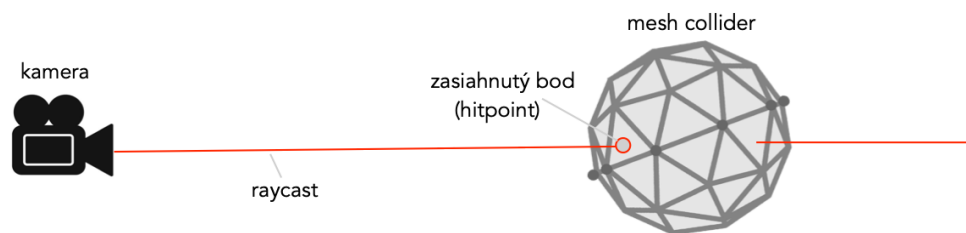
5.5.1 Detekcia miesta dotyku modelu

Pri tomto riešení je potrebné zaznamenať miesto na okraji modelu, kde sa môže daná grafika vytvoriť. V aplikácii sa využíva kombinácia komponent *raycast* a *collider* (opísaný v sekcii 2.2.4).

Raycast slúži na detekciu prieniku polpriamky cez objekt. Používa sa na zistenie, čo sa nachádza v nejakom smere, ktorý je určený vektorom. Na využívanie techniky *raycast* má Unity vlastnú knižnicu.

Keďže aplikácia vyžaduje vysokú presnosť miesta kliknutia na model, je ideálne využiť *MeshCollider*, pretože môže mať rovnaký tvar ako model. Vyššia réžia tohto typu komponenty ale nevlplyva na výkon, keďže detekcia zasiahnutia sa vykonáva len pri používateľskom vstupe (teda je omnoho menej častá ako pri každej *Update* metóde snímku).

Keď používateľ zvolí miesto, na ktorom chce vytvoriť interaktívny bod, z miesta kamery kde bol dotyk sa vytvorí *raycast* smerom dopredu do scény. Na mieste zásahu sa následne vytvorí interaktívny bod. Proces je naznačený na obrázku 5.4.



Obr. 5.4: Raycast z kamery na collider

5.5.2 Pridávanie obsahu interaktívneho bodu

Interaktívny bod slúži na zobrazovanie informácií o danej časti modelu. Oproti návrhu sa pri implementácii pridalo tlačidlo na posun bodu, kvôli blokovaniu pohľadu na model. Okno s obsahom interaktívneho bodu má teda nasledujúce časti:

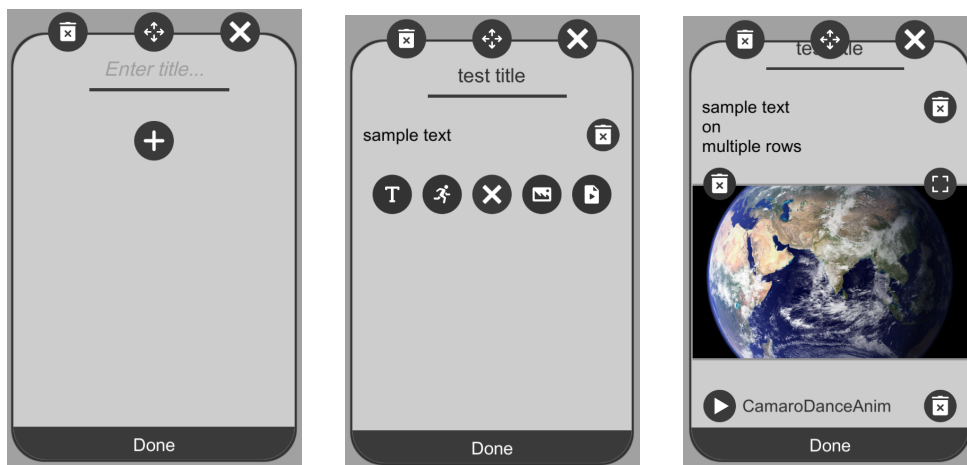
- Manipulačné tlačidlá na vrchu - obsahuje tlačidlo vymazania, posunu a zavretia interaktívneho bodu
- Rolovací zoznam, ktorý obsahuje všetky informácie o danom bode, ktorý tam tvorca prezentácie pridal. Prvá bunka zoznamu je vždy nadpis bodu. Pri editačnom móde je posledná bunka vždy tlačidlo na pridanie nového obsahu.

Bunka zoznamu je jeden typ informácie. Pri stlačení tlačidla na pridanie bunky, sa zobrazia všetky dostupné typy obsahu. Používateľ si následne zvolí typ a pridávanie obsahu potom pokračuje podľa scenárov z predošlej kapitoly. V aplikácii boli implementované nasledujúce typy buniek:

- **Text** - jednoduchá bunka, ktorá obsahuje textovú informáciu
- **Obrázok** - pridanie obrázku zo zariadenia
- **Video** - pridanie videa zo zariadenia
- **Animácia** - pridanie možnosti prehrania špecifickej animácie na modeli

Pri editačnom móde obsahuje každá položka zoznamu tlačidlo na zmazanie. Na importovanie obrázku a videa zo zariadenia bolo použitá knižnica *NativeGallery* z obchodu s assetmi Unity (*AssetStore*) (opísaná v sekcii 5.1.4). Na prehrávanie videa sa využíva UI komponenta *VideoPlayer*. Bunky s obrázkom a videom obsahujú aj tlačidlá na zväčšenie obsahu na celú obrazovku pre lepšiu viditeľnosť.

Vizuál interaktívneho bodu si prešiel viacerými iteráciami zmeny a oproti prvotnému návrhu v podobe wireframu je viditeľná zmena. Ukážka finálnej verzie je na obrázku 5.5.



Obr. 5.5: Vizuál interaktívneho bodu

5.6 Rozšírená realita

Ako už bolo spomínané v sekcii 5.1.5, na použitie rozšírenej reality bola použitá knižnica *AR Foundation*.

Knižnica poskytuje komponenty, ktoré sa starajú o rôzne funkcionality pre AR. Pre ich využitie je potrebné pochopiť ich funkcionality a ako ich spojiť dohromady aby bola vytvorená požadovaná funkcionality. Nasledujúce sú tie, ktoré sú využité v projekte:

- **ARSession**

Každá scéna, ktorá využíva rozšírenú realitu by mala mať túto komponentu. Môže byť pripojená k ľubovoľnému hernému objektu a stará sa o detegovanie prostredia. Ak je deaktivovaná, tak systém okolie nedeteguje. Obsahuje taktiež stav inicializácie AR na danom zariadení a tiež informáciu o tom, ak dané zariadenie AR nepodporuje.

- **ARSessionOrigin**

AR zariadenia poskytujú údaje o okolí pomocou súradníc z priestoru relatívneho voči počiatkovej pozícii AR zariadenia. Komponenta *ARSessionOrigin* sa stará o to, aby tieto údaje boli správne prevedené do priestoru scény Unity - počíta pozíciu, rotáciu a mierku nájdených plôch a bodov v okolí.

- **ARCameraManager**

Hlavná komponenta kamery, ktorá slúži pre AR pohľad. Umožňuje prístup k videu z kamery a tiež k metódam na odhad osvetlenia.

- **ARCameraBackground**

Stará sa o správne vykresľovanie obsahu z kamery zariadenia na pozadie v scéne. Pripája sa k objektu kamery, na ktorom je ARCameraManager.

- **TrackedPoseDriver**

Taktiež sa pripája sa na objekt kamery v scéne a slúži na zmenu polohy a rotácie kamery v scéne podľa pohybu zariadenia po priestore v reálnom svete.

- **ARInputManager**

Bez tejto komponenty nedokáže TrackedPoseDriver získať dáta o polohe zariadenia.

- **ARRaycastManager**

Kontroluje či "lúč"(ktorý je definovaný počiatočným bodom a smerom) zasiahne niektoré z detegovaných AR objektov (bodov a plôch). Má podobný princíp ako klasický raycast z modulu Unity Physics, ale keďže AR objekty nemusia mať fyzikálne vlastnosti v scéne, bolo pre nich vytvorené špeciálne rozhranie.

- **ARPlaneManager**

Slúži ako rozhranie pre detekciu plôch v okolí. Poskytuje udalosť *planeUpdated*, ktorá sa vyvolá pri zmene a predáva informácie o tom, či plocha bola pridaná, vymazaná alebo aktualizovaná. Iné skripty môžu na udalosť reagovať a s poskytnutými informáciami napríklad vytvoriť grafiku danej plochy.

- **ARPointCloudManager**

Rozhranie pre detekciu dôležitých bodov okolia.

Po stlačení tlačidla prechodu do rozšírenej reality sa zapnú komponenty pre chod AR. Po inicializácii komponenty ARSession sa ukáže grafika, ktorá radí používateľovi, aby hýbal mobilom po okolí. Keď sa podarí nájsť rovná plocha v okolí a zavolá sa príslušná udalosť z ARPlaneManager, na obrazovke sa objaví grafika plochy. V tomto momente môže používateľ kliknúť na ľubovoľnú časť plochy a 3D model sa na ňu umiestni - raycasting je podobný v časti umiestňovania interaktívnych bodov, len s rozdielom, že je použitá trieda *ARRaycastManager* opísaná vyššie. Po umiestnení stále fungujú gesta na manipuláciu s modelom, takže používateľ môže meniť veľkosť a rotáciu (v AR iba horizontálnu rotáciu) rovnako ako v klasickom 3D zobrazení. Ukážka aplikácie v režime AR je v prílohe na obrázku D.8.

Testovanie

Testovanie je dôležitou súčasťou vývoja softvéru. Testovať treba čo najčastejšie, aby predišlo chybám a zaručila sa určitá úroveň kvality produktu. Počas vývoja bola aplikácia pravidelne testovaná a na základe toho sa aj menilo používateľské rozhranie. Na testovanie bol použitý tablet iPad 5. generácie (samozrejme s podporou AR) s operačným systémom iOS 13.5 a Samsung Galaxy Tab S5e s operačným systémom Android 9. V skorších častiach vývoja sa používal aj iPhone SE so systémom iOS 13.4, ale zariadenie má príliš malú obrazovku a na účel prezentácie 3D modelu je menej praktický. Avšak kvalita testovania je dlhý a náročný proces, preto sa nedá zaručiť aplikácia bez chýb.

6.1 Testovanie použiteľnosti

Testovanie použiteľnosti je spôsob testovania, ktoré sa snaží zistiť ako je aplikácia jednoduchá na použitie. Aplikácia sa testuje na reálnych používateľoch.

Používatelia vykonávajú určené úlohy pod dohľadom pozorovateľa, ktorý sleduje, kde narazia na problémy. Ak viac ľudí narazí na rovnaký problém, malo by sa hľadať riešenie, ako upraviť danú časť aplikácie. [31]

6.1.1 Testovacie úlohy

Dôležité je si určiť úlohy, aké budú testeri vykonávať. Úlohy sú navrhnuté tak, aby prešli celou aplikáciou a vykonali čo najviac funkcionality. Počas testovania sa vypne nutnosť potvrdzovania novoregistrovaných používateľov, aby testeri nemuseli čakať po registrácii a rovno sa mohli prihlásiť. V testovacích úlohách sa taktiež nenachádza potvrdzovanie registrácie používateľom - je to špecifická funkcia pre administrátorov, ktorých je len malé percento používateľov

- **Úloha 1** - Registrujte sa v aplikácii
- **Úloha 2** - Prihláste sa pomocou vašich údajov

- **Úloha 3** - Budeme pokračovať vo vytváraní prezentácií - zvolte režim editácie
- **Úloha 4** - Vytvorte novú prezentáciu pre model Chevrolet Camaro 1960
- **Úloha 5** - V prezentácii vytvorte interaktívny bod na mieste zadného svetla auta
- **Úloha 6** - Pridajte do bodu ľubovoľný text a obrázok zo zariadenia
- **Úloha 7** - Uložte bod a zavrite ho
- **Úloha 8** - Spustite animáciu s názvom CamaroDanceA na modeli
- **Úloha 9** - Uložte prezentáciu s ľubovoľným názvom
- **Úloha 10** - Prejdite do módu rozšírenej reality
- **Úloha 11** - Umiestnite model niekam do priestoru
- **Úloha 12** - Prejdite naspäť na hlavnú obrazovku s modelmi
- **Úloha 13** - Prejdite do prezentačného módu
- **Úloha 14** - Odhláste sa

6.1.1.1 Testeri

Ďalej je dôležité uviesť kto boli testeri aplikácie. Uvádžam ich vek, zamestnanie a mobilný telefón, ktorý pravidelne používajú. Všetci testeri testovali aplikáciu na tablete iPad 5. generácie.

- Tester A
 - 27 rokov, programátor
 - používa iPhone X, iOS
- Tester B
 - 23 rokov, študent
 - používa Huawei 59 Light, Android
- Tester C
 - 24 rokov, programátor
 - používa Nokia 6, Android
- Tester D
 - 27 rokov, projektový manažér
 - používa iPhone X, iOS

6.1.2 Výsledky testovania

- **Úloha 1** - Registrujte sa v aplikácii

V tomto procese veľké problémy neboli. Registrácia je podobná ako v iných aplikáciách.

- **Úloha 2** - Prihláste sa pomocou vašich údajov

Takisto ako pri úlohe 1, aj tu je proces podobný ako vo väčšine aplikácii. Tester 2 ale zabudol napísať znak zavináča v emailovej adrese a chvíľu mu trvalo prísť na to, prečo sa nevie prihlásiť. Určite by sa hodilo zlepšiť hlášky pri chybe a pridať napríklad kontrolu formátu už pri vyplňaní polí.

- **Úloha 3** - Zvoľte režim editácie

Pri tejto úlohe nebol problém. Jednoduchá obrazovka, kde je jasne ukázané, aký režim sa spustí.

- **Úloha 4** - Vytvorte novú prezentáciu pre model Chevrolet Camaro 1960

Používateľom chvíľu trvalo, kým našli tlačidlo novej prezentácie. Bolo to aj formuláciou tejto úlohy - v aplikácii je totiž tlačidlo "new version", čo viac znie ako nová verzia modelu, a nie prezentácie. Toto tlačidlo by sa malo určite opraviť. Hľadanie modelu avšak nebol problém. Všetci pochopili, že ide o horizontálny list a model našli.

- **Úloha 5** - V prezentácii vytvorte interaktívny bod na mieste zadného svetla auta

Prvým krokom v tomto procese je otočiť model tak, aby videli zadné svetlá. Gesto potiahnutia je intuitívne a hneď to všetci vykonali. Počas otáčania sa ale stávalo, že model sa otočil trochu viac do strany a nedalo sa ho jednoducho vrátiť naspäť. Je to hlavne preto, že gesto potiahnutia otáča model len okolo 2 os.

Pridanie bodu trvalo trochu dlhšie. Používateľom väčšinou trvalo, kým zistili, že treba najprv použiť tlačidlo plus. No keď už klikli na tlačidlo, bolo im jasné, že treba kliknúť na model - pomohlo aj okno, ktoré radí používateľovi aby po zvolení tlačidla klikol na model.

- **Úloha 6** - Pridajte do bodu ľubovoľný text a obrázok zo zariadenia

Otvorenie a pridanie obsahu pochopili všetci. Kliknutie na bod je jednoduché a tlačidlo plus v interaktívnom bode má jasnú funkcionálnosť. Takisto ikonky pre rôzne typy buniek v interaktívnom bode boli všetkým jasné.

- **Úloha 7** - Uložte bod a zavrite ho
Na uloženie bodu treba najprv kliknúť na tlačidlo done v dolnej časti. Väčšina testerov to tak spravila, avšak tester 2 klikol len na zatvorenie bodu. Pravdepodobne by sa hodilo vyskakovacie okno s upozornením na neuložené zmeny, prípadne automatické ukladanie bodov pri zmene.
- **Úloha 8** - Spustite animáciu s názvom CamaroDanceA na modeli
Nájdenie tlačidla na animácie bol celkom zdĺhavý proces pre všetkých testerov. Niektorí nepokladali ikonu postavičky za animáciu a preto boli zmätení. Nakoniec to ale všetci našli a spustili animáciu modelu.
- **Úloha 9** - Uložte prezentáciu s ľubovoľným názvom
Úlohu zvládli všetci testeri bez problémov. Ikona ukladania (disketa) je veľmi dobre známa a všetci ju identifikovali správne.
- **Úloha 10** - Prejdite do módu rozšírenej reality
Všetci testeri poznali skratku AR, preto nebol problém s nájdením funkcionality.
- **Úloha 11** - Umiestnite model niekam do priestoru v okolí
Tester 1 mal problém s umiestnením na stôl, pretože nebola nájdená plocha. Potom to ale skúsil na podlahu a model umiestnil. Ostatní tento problém takisto vyriešili.
- **Úloha 12** - Prejdite naspäť na hlavnú obrazovku s modelmi
Tento úkon bol jasný, niektorí chvíľu váhali, ale všetci tlačidlo vcelku rýchlo našli a opustili prezentáciu.
- **Úloha 13** - Prejdite do prezentačného režimu
Väčšina testerov klikalo na menu režimu vľavo hore - čo je asi najintuitívnejšie chovanie. Avšak zmena režimu sa nachádza pod používateľským menu. Po chvíli pátrania ho všetci našli a prepli režim aplikácie.
- **Úloha 14** - Odhláste sa
V používateľskom menu sa nachádza aj tlačidlo na odhlásenie, takže problém s nájdením tu nebol.

6.1.3 Zhodnotenie testov

Vzhľadom na menší počet testerov, je vhodné venovať sa každému náznaku chyby. Testerí zvládali prechod aplikácie dobre, objavilo sa len pár menších náznakov zaváhania.

Najčastejšie sa vyskytol problém s pridaním interaktívneho bodu. Všetci testerí najprv skúsili pridať bod bez kliknutia na tlačidlo plus. Aj keď každý následne našiel správne tlačidlo, používateľom pripadal prvý spôsob intuitívnejší a prirodzenejší. Takéto chovanie by ale znamenalo risk náhodného pridania bodu napríklad pri otáčaní. Ako zjednodušenie tohto ovládania by sa dalo napríklad implementovať gesto dlhého dotyku ako alternatíva na pridanie bodu. Riešenie ostatných menších problémov bolo navrhnuté priamo pri hodnotení výsledkov daných úloh.

Video záznamy z testovania sú priložené k práci na CD.

Záver

Výsledkom práce je funkčná aplikácia pre operačné systémy iOS a Android, pre ktoré bola práca zameraná. Aplikácia taktiež funguje aj na systémoch s Windows a MacOS, ale pre tieto zariadenia samozrejme nefunguje časť s AR a tiež by bolo potrebné viac testovania a prispôbenie ovládania. Splnené boli všetky funkčné požiadavky, ktoré sa určili na začiatku práce. V kapitolách sú postupne opísané analýza, návrh, implementácia a testovanie. Aplikácia ponúka základ softvéru pre tvorenie interaktívnych prezentácií k 3D modelom pre prezentáciu produktov v rámci firmy.

Riešenie obsahuje systém prihlasovania a správy používateľov s využitím Firebase ako serverovej časti. Ponúka vytváranie interaktívnych bodov k modelom pomocou jednoduchého kliknutia na model. V rámci interaktívnych bodov umožňuje pridávanie textu, obrázku, videa a spúšťača animácie pre lepší popis daných častí a ich funkcionalít. Aplikácia taktiež poskytuje tieto funkcie v rozšírenej realite.

Možnosti rozšírenia

Práca sa nezaoberala importovaním vlastných modelov do aplikácie. Import rôznych formátov za behu aplikácie je zložitý proces, ktorý je ťažké implementovať pre širokú škálu rôznych 3D modelov. Pri budúcich zlepšeniach by sa ale pridávanie modelov malo určite zväziť.

Súčasnú riešenie predstavuje aplikáciu pre jednu spoločnosť. Už pri builde aplikácie je určená sada 3D modelov, ktorú bude obsahovať, a serverová časť jedná len s používateľmi v rámci jednej firmy (všetci administrátori schvaľujú všetkých, používatelia nie sú rozdelení na spoločnosti). Možné rozšírenie je umožniť viacerým firmám naraz používať ten istý build aplikácie a pridať entitu firmy, podľa ktorej budú používatelia rozdelení. Toto riešenie by už ale pravdepodobne vyžadovalo funkciu pridávania 3D modelov.

Literatúra

- [1] Chen, Y.; et al: An overview of augmented reality technology. *Journal of Physics: Conf. Series 1237 022082*, 2019. Dostupné z: <https://iopscience.iop.org/article/10.1088/1742-6596/1237/2/022082/pdf>
- [2] Carmigniani, J.; Furht, B.; Anisetti, M.; aj.: Augmented reality technologies, systems and applications. *Multimed Tools Appl 51*, 2011: str. 341–377. Dostupné z: <https://doi.org/10.1007/s11042-010-0660-6>
- [3] Xu, Z.; Wang, G.: Research on the Method of 3D Registration Technology. *IOP Conf. Ser.: Earth Environ. Sci. 440 032139*, 2020. Dostupné z: <https://iopscience.iop.org/article/10.1088/1755-1315/440/3/032139>
- [4] Billinghamurst, M.; Kato, H.; Myojin, S.: Advanced Interaction Techniques for Augmented Reality Applications. *Lecture Notes in Computer Science. 5622. 07 2009*, s. 13–22, doi:10.1007/978-3-642-02771-0_2.
- [5] Romilly, M.: 12 Best Augmented Reality SDKs. 2019. Dostupné z: <https://dzone.com/articles/12-best-augmented-reality-sdks>
- [6] Apple, Inc.: *Apple Doc: Understanding World Tracking*. 2020. Dostupné z: https://developer.apple.com/documentation/arkit/world_tracking/understanding_world_tracking
- [7] Google Developers: *ARCore Overview*. 2019. Dostupné z: <https://developers.google.com/ar/discover>
- [8] Augmented Reality UI Design: What, Why and How? 2019. Dostupné z: <https://arvrjourney.com/augmented-reality-ui-design-what-why-and-how-84c6b4b66e24>

- [9] Haas, J. K.: A History of the Unity Game Engine. 2014. Dostupné z: <https://digitalcommons.wpi.edu/iqp-all/3207>
- [10] Unity Technologies: *Unity User Manual*. 2020. Dostupné z: <https://docs.unity3d.com/2019.4/Documentation/Manual>
- [11] Gregory, J.: *Game Engine Architecture*. CRC Press, 2015, ISBN 978-1-4665-6006-2.
- [12] Xie, J.: Research on key technologies base Unity3D game engine. 2012, ISBN 978-1-4673-0241-8, s. 695–699, doi:10.1109/ICCSE.2012.6295169.
- [13] Unity Technologies: *What's new in Unity 5.0*. 2015. Dostupné z: <https://unity3d.com/unity/whats-new/unity-5.0>
- [14] Fine, R.: *UnityScript's long ride off into the sunset*. Unity Technologies Blog, 2017.
- [15] Štědroň, B.: *Manažerské řízení a informační technologie*. Praha: Grada, 2007, ISBN 978-8-024-72052-4, 90 s.
- [16] Gaul, R.: Component Based Engine Design. Dostupné z: <https://www.randygaul.net/2013/05/20/component-based-engine-design/>
- [17] Vesecký, A.: *Architecture of Computer Games, Component Architecture 1 [Prednáška]*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.
- [18] Encyclopaedia Britannica: *Microsoft PowerPoint*. 2013. Dostupné z: <https://www.britannica.com/technology/Microsoft-PowerPoint>
- [19] Campos, P.; Clemmensen, T.; Nocera, J. A.; aj.: *Human Work Interaction Design. Work Analysis and HCI: Third IFIP 13.6 Working Conference, HWID 2012, Copenhagen, Denmark, December 5-6, 2012, Revised Selected Papers*. Springer, 2013, ISBN 978-3-642-41144-1, 31-33 s.
- [20] Aurora3D Software: *Creative 3D Interactive Business Presentation Software*. 2020. Dostupné z: <https://www.presentation-3d.com/products/presentation-3d.html>
- [21] SaVRee 3D Interactive Media: *SaVRee: About us*. 2019. Dostupné z: <https://savree.com/en/about/>
- [22] Arlow, J.; Neustadt, I.: *UML 2 and the unified process: practical object-oriented analysis and design*. Addison-Wesley Professional, 2006, ISBN 0-321-32127-8.
- [23] *Domain Modeling*. 2018. Dostupné z: <https://www.scaledagileframework.com/domain-modeling/>

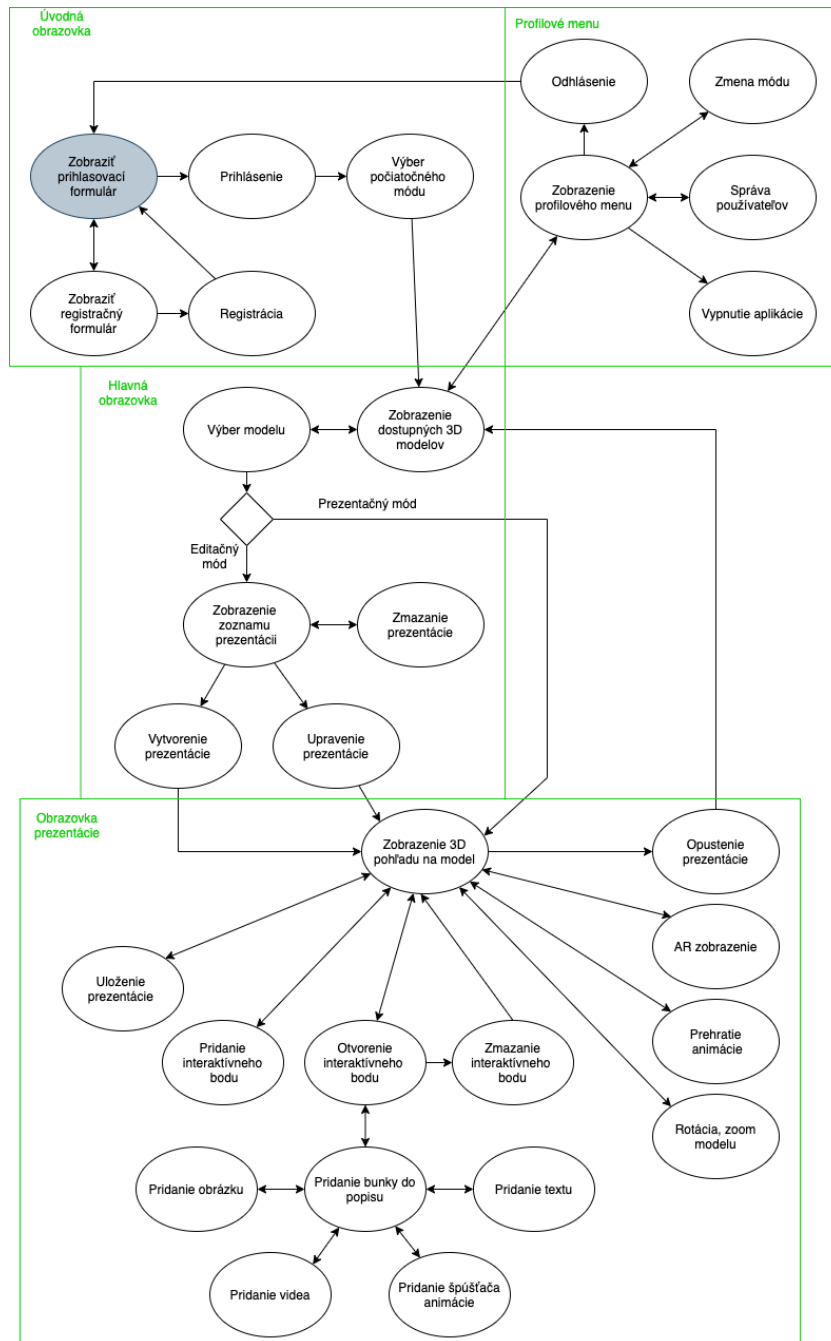
-
- [24] Nielsen, J.: Ten Usability Heuristics. 2005, ISSN 1548-5552. Dostupné z: <http://informaticathomas.nl/heuristicsNielsen.pdf>
- [25] Shur, G.: *"Coviator" - application for shared trip planning - OS Android*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2016.
- [26] Chacon, S.; Straub, B.: *Pro Git*. Apress, 2014, ISBN 1-484-20077-2.
- [27] Unity Technologies: *Unity manual: About AR Foundation*. 2019. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/index.html>
- [28] Unity Technologies: *Unity manual: XR Plug-in Framework*. 2020. Dostupné z: <https://docs.unity3d.com/2019.4/Documentation/Manual/XRPluginArchitecture.html>
- [29] Google Developers: *Google Firebase Documentation*. 2020. Dostupné z: <https://firebase.google.com/docs>
- [30] Apple, Inc.: *Príručka užívateľa pre iPhone*. 2020. Dostupné z: <https://support.apple.com/sk-sk/guide/iphone/iph75e97af9b/ios>
- [31] U.S. Dept. of Health and Human Services: *The Research-Based Web Design Usability Guidelines, Enlarged/Expanded edition*. Dostupné z: <https://www.usability.gov/how-to-and-tools/methods/usability-testing.html>

Zoznam použitých skratiek

- AR** Augmented reality
- CVS** Concurrent versions system
- GUI** Graphical user interface
- ID** Identification
- IDE** Integrated development environment
- QA** Quality assurance
- SDK** Software development kit
- UC** Use case
- UI** User interface
- XR** Extended reality

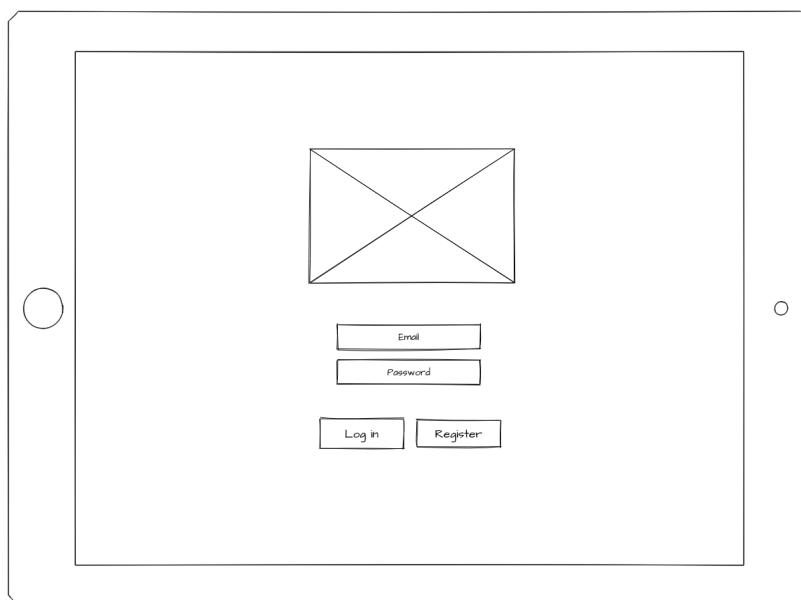
Diagramy

B. DIAGRAMY



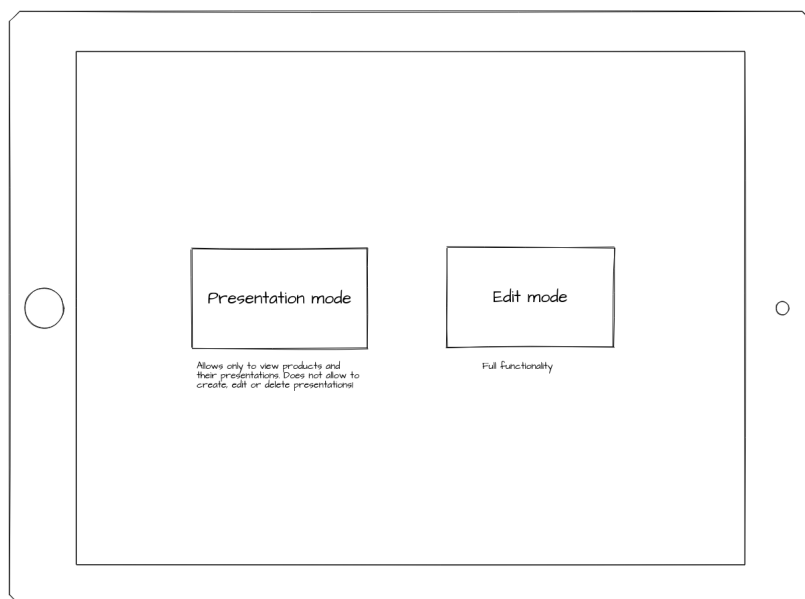
Obr. B.1: Graf úloh

Wireframy

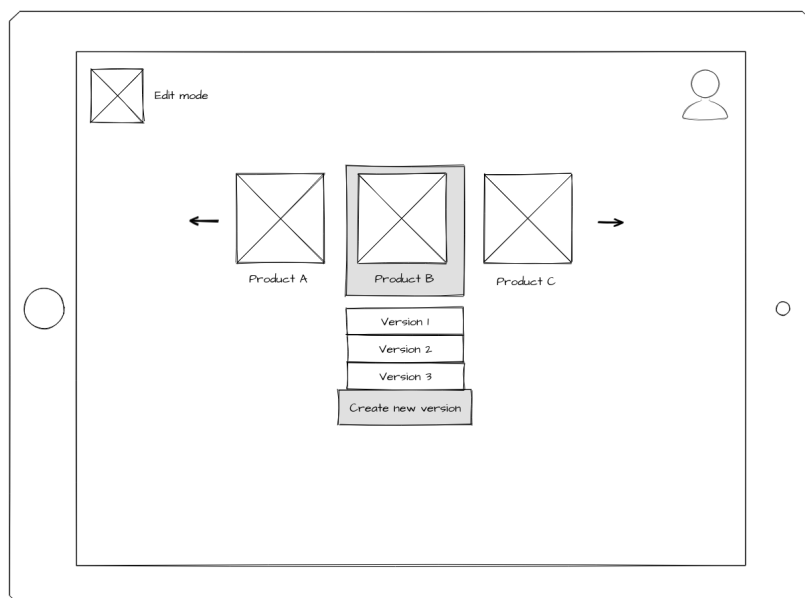


Obr. C.1: Prihlasovacia obrazovka

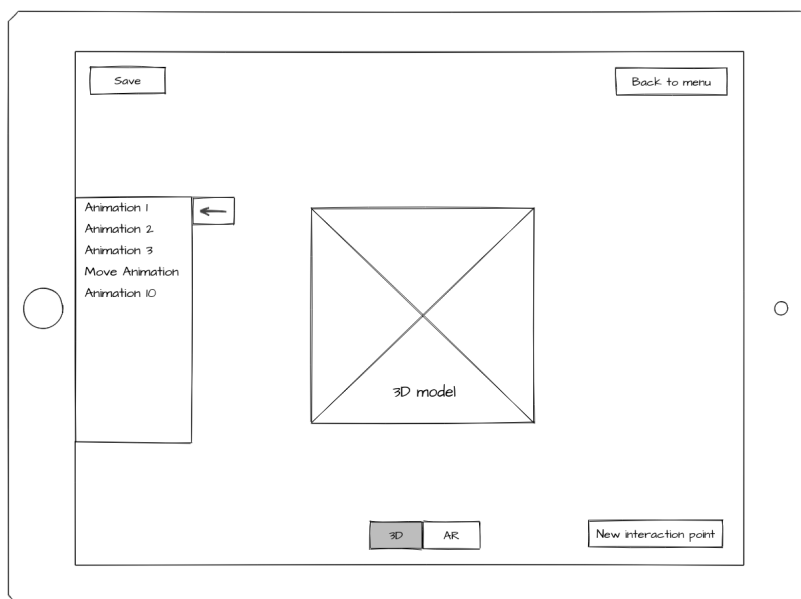
C. WIREFRAMY



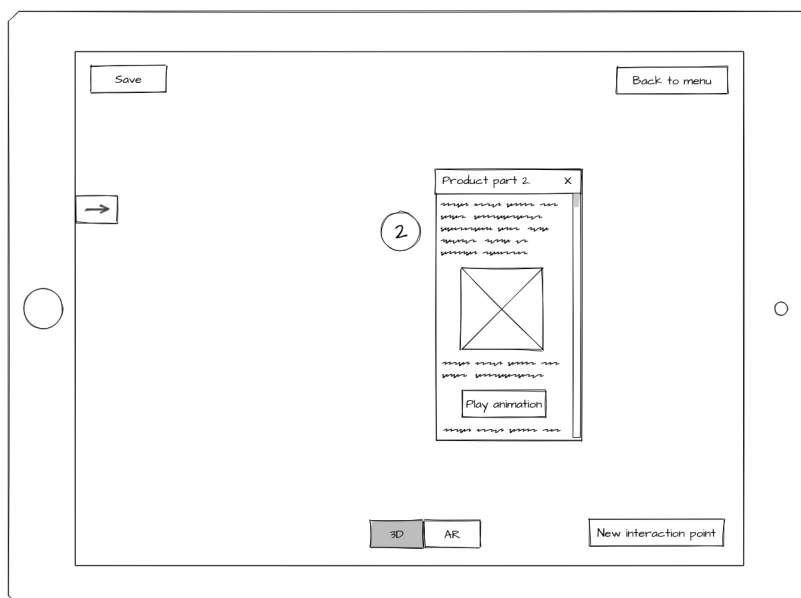
Obr. C.2: Zvolenie počiatočného módu po prihlásení



Obr. C.3: Obrazovka s produktami a prezentáciami

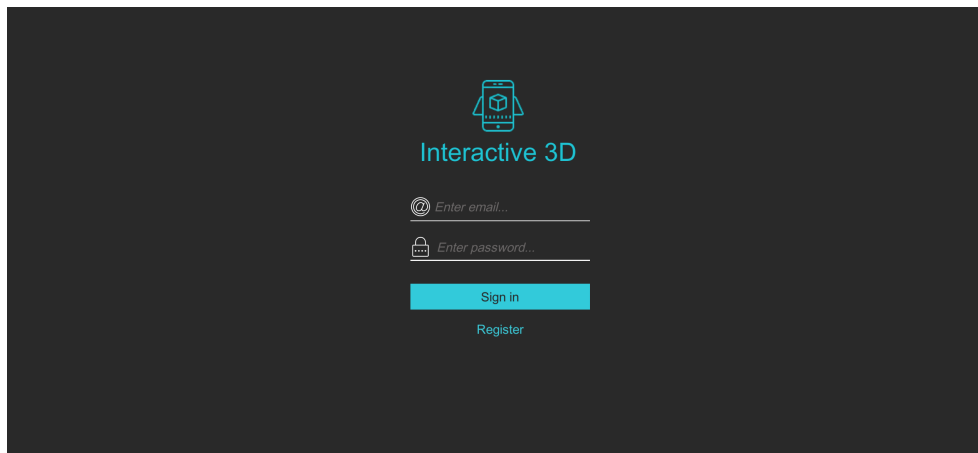


Obr. C.4: Obrazovka prezentácie



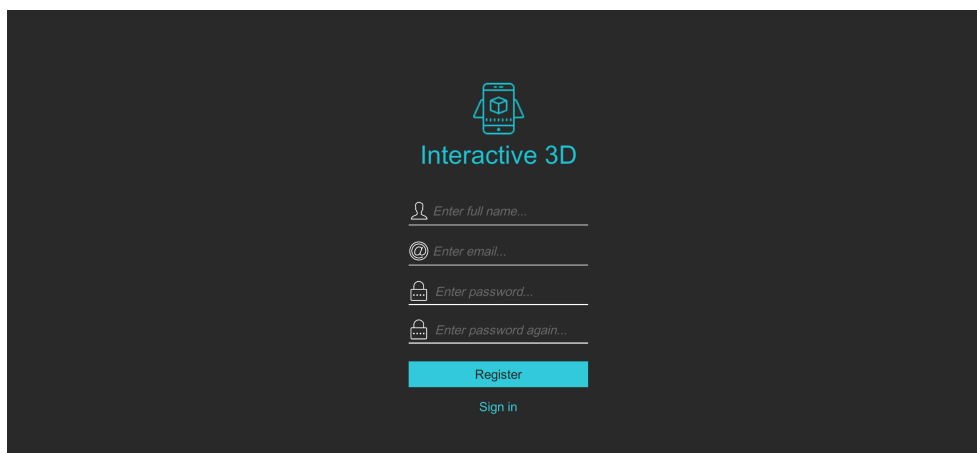
Obr. C.5: Obrazovka prezentácie s otvoreným interaktívnym bodom

Ukážky aplikácie

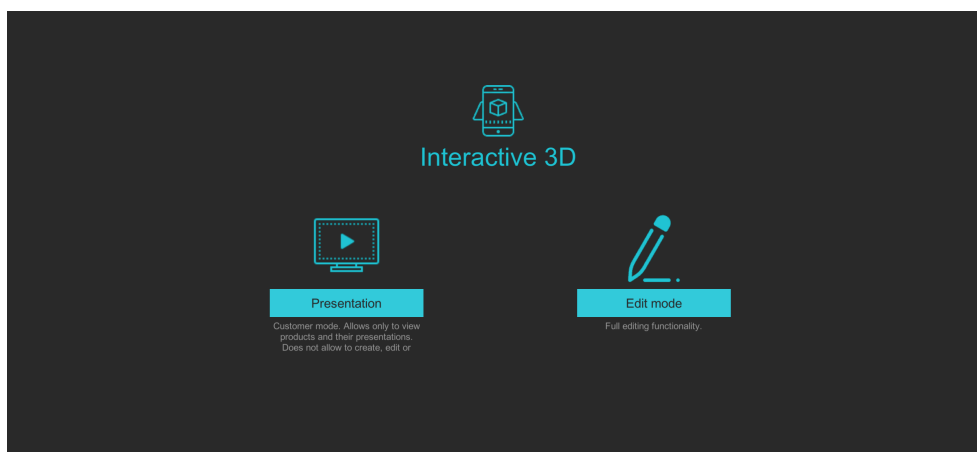


Obr. D.1: Prihlasovacia obrazovka

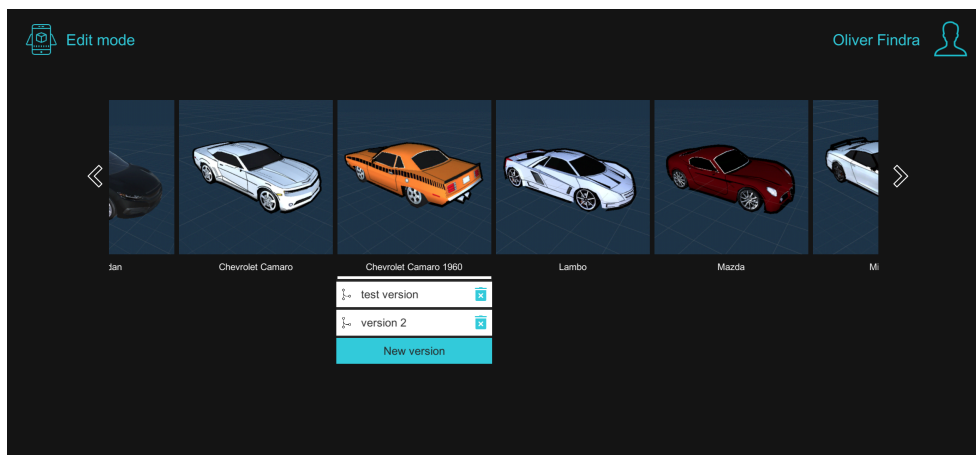
D. UKÁŽKY APLIKÁCIE



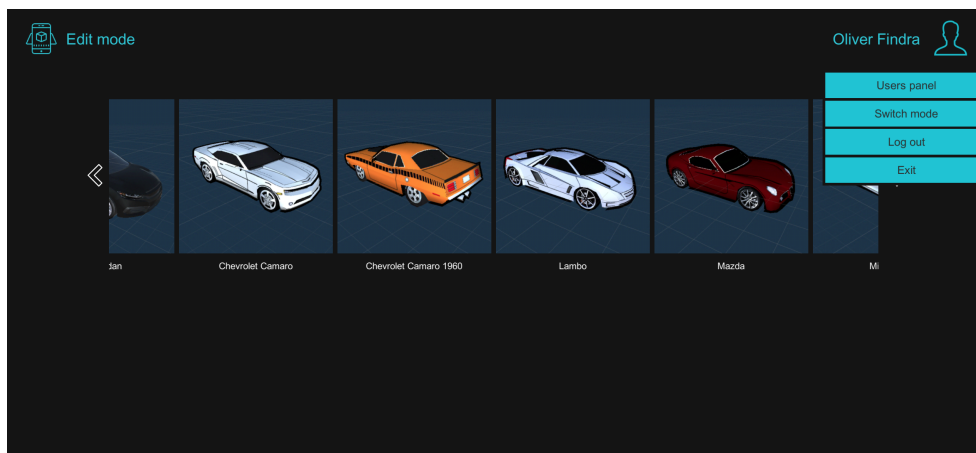
Obr. D.2: Registračný formulár



Obr. D.3: Počiatočné zvolenie režimu

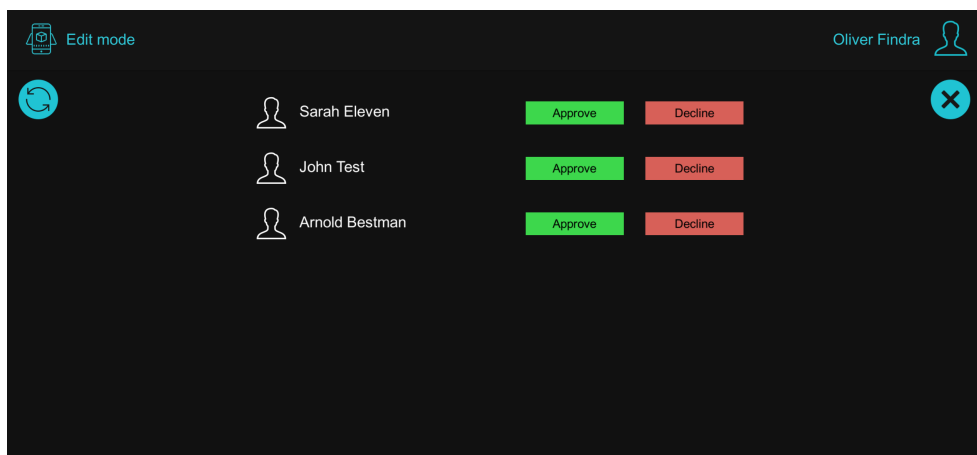


Obr. D.4: Výber modelu a prezentácie



Obr. D.5: Používateľské menu

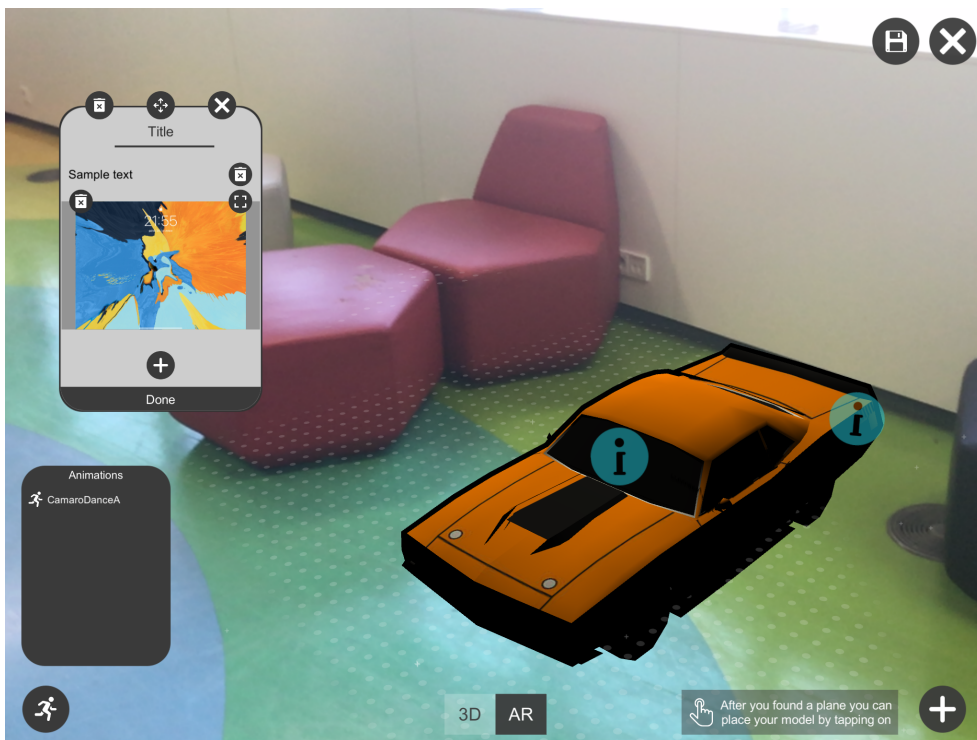
D. UKÁŽKY APLIKÁCIE



Obr. D.6: Schvaľovanie používateľov administrátorom



Obr. D.7: Tvorba prezentácie - 3D režim



Obr. D.8: Tvorba prezentácie - AR režim

Obsah priloženého CD

readme.txt	stručný popis obsahu CD
exe.....	adresár so spustiteľnou formou implementácie
└─ interactive3d.exe	Windows verzia (obmedzená funkcionálnosť)
src	
└─ impl	zdrojové kódy implementácie
└─ interactive-3d-cvut	zložka Unity projektu
└─ thesis	zdrojová forma práce vo formáte L ^A T _E X
text	text práce
└─ thesis.pdf	text práce vo formáte PDF
testing.....	video záznamy z testovania
└─ tester1.mov	
└─ tester2.mov	
└─ tester3.mov	
└─ tester4.mov	