



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

APLIKACE ROZŠÍŘENÉ REALITY V ÚDRŽBĚ VÝROBNÍCH STROJŮ

APPLICATION OF AUGMENTED REALITY IN MAINTENANCE OF PRODUCTION MACHINES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Blaha

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Zdeněk Tůma, Ph.D.

BRNO 2020

Zadání diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Student: **Bc. Jan Blaha**
Studijní program: Strojní inženýrství
Studijní obor: Výrobní stroje, systémy a roboty
Vedoucí práce: **Ing. Zdeněk Tůma, Ph.D.**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Aplikace rozšířené reality v údržbě výrobních strojů

Stručná charakteristika problematiky úkolu:

V rámci diplomové práce bude zpracována rešeršní část, obsahující technické a softwarové možnosti rozšířené reality. V následující fázi diplomant provede návrh aplikace pro údržbu stroje MCV Kovosvit pomocí software Unity.

Cíle diplomové práce:

Rešerše a zhodnocení současného stavu.
Systémový rozbor řešené problematiky.
Návrh aplikace v prostředí Unity.
Vlastní závěry a zhodnocení.

Seznam doporučené literatury:

Unity Real-Time Development Platform | 3D, 2D VR & AR ... [online]. [cit. 2019-09-13]. Dostupné z: <https://unity.com/>

GOMES DE SÁ, Antonino and Gabriel ZACHMANN. 1999. Virtual reality as a tool for verification of assembly and maintenance processes. Computers & Graphics. 23(3), 389-403.

DANGELMAIER, Wilhelm, Matthias FISCHER, Jürgen GAUSEMEIER, Michael GRAFE, Carsten MATYSCZOK and Bengt MUECK. 2005. Virtual and augmented reality support for discrete manufacturing system simulation. Computers in Industry. 56(4), 371-383.

ŽÁRA, Jiří. VRML 97 – Laskavý průvodce virtuálními světy. Brno: Computer Press, 1999. ISBN 8072261436.

Augmented Reality SDK Comparison [online]. [cit. 2019-10-04]. Dostupné z:
<http://socialcompare.com/en/comparison/augmented-reality-sdks>

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Diplomová práce se zabývá návrhem aplikace rozšířené reality v údržbě obráběcích strojů. Práce obsahuje část rešeršní a část návrhovou. V první části diplomové práce je shrnuta problematika rozšířené reality z obecného hlediska. Součástí je i představení předních výrobců této technologie. Praktická část práce se zabývá návrhem aplikace. Je shrnut myšlenkový proces návrhu od tvorby databáze po výsledný prototyp. Je uvedeno několik možných řešení a z nich je vybráno nejvhodnější. Závěr práce je věnován diskuzi dosažených výsledků a praktickým doporučením.

ABSTRACT

The master's thesis deals with the design of an augmented reality application for production machine maintenance. The thesis consists of two parts: the theoretical and the design part. The first part of the thesis summarizes the subject matter of augmented reality from a general point of view. It also includes an overview of leading manufacturers. The practical part of the thesis deals with application design. It contains the thought process from the database creation to the final prototype. Several possible solutions are listed, and the most suitable one is selected. The conclusion of the thesis contains a discussion of the results obtained, and practical recommendations for future uses.

KLÍČOVÁ SLOVA

obráběcí stroje, údržba, rozšířená realita, Unity, Vuforia

KEYWORDS

production machines, maintenance, augmented reality, Unity, Vuforia

BIBLIOGRAFICKÁ CITACE

BLAHA, J. *Aplikace rozšířené reality v údržbě výrobních strojů* Brno, Vysoké učení technické v Brně, Fakulta strojního inženýrství. 2020, 74 s., Vedoucí diplomové práce Ing. Zdeněk Tůma, Ph.D.

PODĚKOVÁNÍ

Děkuji tímto vedoucímu diplomové práce, kterým byl Ing. Zdeněk Tůma, Ph.D. za cenné rady a připomínky. Zejména za čas strávený na konzultačních hodinách.

Dále bych chtěl poděkovat všem zaměstnancům firmy SIGMATRONIC za vstřícný přístup. Zejména pak děkuji Bc. Janu Forejtníkovi, Bc. Petrovi Fašianokovi a Ing. Markovi Gonzálezovi.

Největší dík však patří mým rodičům, kteří mi byli oporou po celou dobu mého studia.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedení Ing. Zdeňka Tůmy, Ph.D. a s použitím literatury uvedené v seznamu.

V Brně dne 26.6.2020

.....

Bc. Jan Blaha

Obsah

1	ÚVOD	15
2	MOTIVACE	16
3	ROZŠÍŘENÁ REALITA	17
3.1	Virtuální realita	17
3.1.1	Technické požadavky na VR	18
3.2	Mixed Reality – MR	18
4	HISTORICKÝ POHLED NA ROZŠÍŘENOU REALITU	19
5	OBLASTI POUŽITÍ ROZŠÍŘENÉ REALITY	22
5.1	Medicína	22
5.1.1	NuEyes.....	23
5.1.2	Brain Power	24
5.1.3	SCOPIs.....	25
5.2	Vojenské použití	26
5.2.1	Výcvik pilotů a výsadkářů	26
5.2.2	Tactical Augmented Reality	26
5.2.3	ARES	27
5.3	Průmyslové použití	28
5.3.1	Dokumentace a manuály	28
5.3.2	Návrh a konstrukce	31
6	SYSTÉMOVÝ ROZBOR PROBLEMATIKY	32
6.1	Definice aktuální situace	32
6.2	Definice problému	32
6.2.1	Definice zkoumaného systému	33
6.3	Formulace cílů	34
6.4	Systém veličin.....	34
6.5	Definice a výběr vývojářského prostředí	34
6.5.1	Grafické rozlišení	34
6.5.2	Výpočetní výkon.....	35
6.5.3	Velikost vývojářského týmu	35
6.5.4	Použití převzatých entit a databází	35
6.5.5	Typ aplikace	36
6.5.6	Zhodnocení a výběr platformy	36
7	VÝBĚR KNIHOVNY	37
7.1	Výběr trackovací SDK metody	37
7.1.1	Cena a licence	37
7.1.2	Kompatibilita s Unity	37
7.1.3	Marker tracking	37
7.1.4	Natural Feature tracking	37
7.1.5	3D Object tracking.....	37
7.1.6	IMU Senzory	38
7.1.7	Visual Search.....	38
7.1.8	Face tracking.....	38
7.1.9	GPS	38
7.2	Posouzení metod	39
7.3	Zhodnocení výsledků a výběr knihovny	39

8	NÁVRH APLIKACE	40
8.1	Výběr knihoven a jejich implementace	41
8.1.1	Databáze Vuforia	41
8.1.2	Posouzení vhodnosti Image Target Databáze	43
8.2	Tvorba 3D modelů	44
8.2.1	Úprava 3D modelů	44
8.2.2	Rozdělení 3D modelů na podskupiny	46
8.3	Výběr Model Target technologie	46
8.4	Import databází do prostředí Unity	48
8.5	Úprava trackovacích scriptů	48
8.6	Tvorba UI	49
8.6.1	Canvas	50
8.6.2	Soustava panelů	51
8.6.3	Menu	52
8.6.4	Zobrazení 3D indikátorů	54
8.6.5	Zobrazení textu	56
8.6.6	Dodatečná tlačítka UI	59
8.7	Tvorba databáze poruch	61
8.7.1	Všechny informace obsaženy ve scriptech	61
8.7.2	Použití vlastní databáze	61
8.7.3	SQL databáze	62
8.7.4	Použití externího modulu	62
8.8	Testování a debugging	63
9	ZHODNOCENÍ A DISKUZE VÝSLEDKŮ	65
10	ZÁVĚR	66
11	SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK	71
11.1	Seznam tabulek	71
11.2	Seznam obrázků	72
12	SEZNAM PŘÍLOH	74

1 ÚVOD

Diplomová práce se zabývá návrhem aplikace rozšířené reality v údržbě výrobních strojů. Rozšířená a virtuální realita zažívá v poslední době nebývalý rozvoj. Oblastí využití této technologie je nepřeberné množství. Největší popularitě se však těší v průmyslu a videoherním odvětví. Velké množství moderních výrobních závodů začíná vyvíjet tlak na implementaci technologie rozšířené reality do svých společností. Důvodem tohoto tlaku je úspora času a energie, kterou může rozšířená realita nabídnout. Dalším důvodem je omezení rizika chyby a eliminace jejího dopadu. Dá se předpokládat masivní nasazení této technologie ve všech velkých průmyslových závodech v následujících letech.

2 MOTIVACE

Motivací k této práci je zejména možnost podílení se na vývoji moderních technologií. Akademická půda k tomu vytvořila vhodné prostředí a byla by škoda této možnosti nevyužít. Společností zabývajících se touto problematikou je pomálu. Je tedy záhodno využít trendu a připojit se k těmto výrobcům.

Část motivace je však i osobní. Tuto práci jsem pojal jako výzvu. Chtěl jsem dokázat, že i člověk nepoznamenaný programováním, je schopen vyvinout moderní aplikaci s reálným použitím. Bylo prokázáno, že při investici dostatečného množství času je možné dosáhnout zajímavých výsledků.

3 ROZŠÍŘENÁ REALITA

Rozšířená realita pochází z anglického slova Augmented reality, dále jen AR. Jedná se moderní způsob zachycování a zobrazování reálného světa. Záznam obrazu reálného světa je obvykle promítán spolu s počítačem generovanou grafikou přímo uživateli. V některých případech je mimo grafického znázornění možno doplnit AR o další vjemy jako například zvukové efekty. Pokud chceme mluvit o rozšířené realitě, musí systém splňovat následující prvky:

- AR reprezentuje prvky reálného světa současně s prvky vygenerovanými počítačem.
- Promítání i snímání předmětů probíhá v reálném čase.
- Přesná prostorová reprezentace objektů reálného světa, jakožto i objektů promítaných.

Hlavní přínos AR spočívá v možnosti promítnout data zároveň s prvky reálného světa. Data se obvykle neomezují pouze na uživatelský interface, ale je velmi často promítán určitý 3D objekt. Díky možnosti trackování a možnosti rozeznávání objektů je možné na 3D promítaný objekt nahlížet z různých úhlů, či ho různě modifikovat. To dává uživateli větší přehled o situaci, než by bylo možné dosáhnout pouhým 2D zobrazením.

Velké oblibě se AR těší také díky rozvoji chytrých telefonů a všeobecné dostupnosti internetu na těchto zařízeních. Většina nových smartphonů je dnes vybavena silným procesorem, který běžně má i 8 jader a dostatečným grafickým zobrazením. To přispívá k rozvoji AR hlavně na těchto zařízeních. Mobilní telefon je běžně vybaven poměrně kvalitní kamerou, která dokáže snímat okolí a grafická karta se potom postará o vykreslení promítaného předmětu. Do mobilních aplikací často bývá přidávána funkce zjišťování polohy GPS. Mobil tak přesně ví, kde se nachází a díky tomu dokáže zobrazovat relevantní údaje. Může se jednat například o promítnutí trasy k nemocnici či obchodu. Turisti v cizích městech si jednoduše mohou promítnout potřebnou trasu a řídit se pouze videokamerou v telefonu.

Dostatečný výpočetní výkon mobilních telefonů je také důvod pro masivní rozvoj AR určené pro mobilní telefony. AR hru pro mobilní telefon v podstatě není nijak obtížné vytvořit. Proto celosvětově sklízí úspěch, hlavně u mladších hráčů [1].

3.1 Virtuální realita

Dalším druhem promítání je Virtuální realita, dále jen VR. Na rozdíl od rozšířené reality je výsledný obraz tvořen pouze počítačem generovanou grafikou. Není tedy nutné snímat okolní prostor kamerovým systémem.

Osoba využívající VR je obvykle vybavena speciálními brýlemi, které jsou propojeny s počítačem. Ten je zodpovědný za veškeré výpočty jakožto i za renderování obrazu a jeho promítání v dostatečné kvalitě.

Uživatel obvykle setrvává ve stacionární pozici. Jeho výhled plně omezují nasazené brýle. Vidí před sebou tedy pouze počítačovou simulaci a jeho schopnost orientace v prostoru je velmi silně omezena. Většina headsetů je rovněž vybavena sluchátky. To jen podporuje pocit odloučení od reality. Na druhou stranu tato vlastnost usnadňuje možnost plně se ponořit do simulovaného prostředí. Při tomto způsobu využívání VR je však hráč často natolik ponořen do

simulace, že je schopen narazit do okolních překážek, či zranit sebe nebo jinou osobu. Z toho důvodu jsou některé simulace spouštěny pouze v sedu. Další možností je omezení pohybu různými pružnými zábranami v podobě jakéhosi plotu okolo uživatele.

3.1.1 Technické požadavky na VR

Nákladnějším řešením VR je doplnění prostoru o různé senzory, jež je možné trackovat a zobrazovat v simulovaném prostoru. K tomu je však nutné pořízení jak výše zmíněných senzorů, brýlí, tak i speciálního snímacího zařízení. Proto se může stát spuštění VR na této úrovni velice nákladnou zábavou.

Uživatelé VR je možné vybavit dodatečnými controlery. Obvykle v podobě ergonomicky tvarovaných ovladačů. Tyto ovladače umožňují určitým způsobem interagovat s prostředím. A to zejména díky tlačítkům, jimiž jsou vybaveny. Většina moderních výrobců je navíc vybaví senzory k zajištění jejich polohy. Důvodem je, že hráč poté před sebou vidí dané kontrolery a je schopen se lépe orientovat v prostoru. V herním průmyslu se těchto kontrolerů využívá k simulaci různých zbraní, nástrojů aj. Aby se zamezilo zběsilému pohybu uživatele, je hojně využíváno funkce „teleportu“. Místo toho, aby uživatel musel fyzicky dojít na požadované místo, stiskne tlačítko a na místo se prostě teleportuje.

VR přináší mnoho benefitů, k bezproblémovému fungování je však zapotřebí značné technické zázemí. Mimo jiné se jedná o dostatečný výpočetní výkon počítače. Musí být také vybaven dostatečně silnou grafickou kartou s možností připojení VR. Kvalita promítaného obrazu je přímo úměrná kvalitě použité grafické karty. Ta musí dostatečně rychle vykreslovat požadovanou grafiku. Dalším kritickým parametrem je i rychlost obnovovací frekvence. Udávána je jednotkách fps (frame per second). Pro plynulou simulaci se doporučuje alespoň 120 fps, ale ani ta v určitých situacích nemusí stačit. Proto se většina výrobců VR headsetů zaměřuje právě na zvýšení této frekvence.

3.2 Mixed Reality – MR

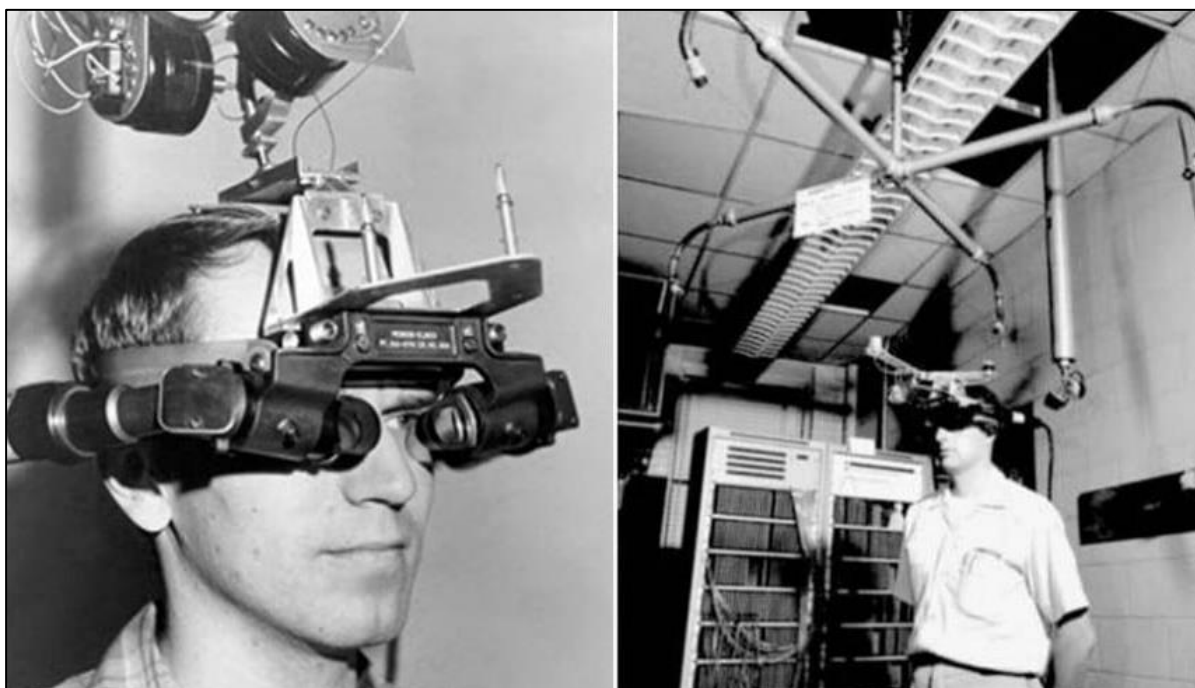
Zatímco VR kompletně nahrazuje reálný svět digitálním prostředím, Mixed reality, je jakýmsi mezistupněm mezi VR a AR. Principiálně fungování MR je postaveno na stejných principech jako výše zmíněné. V některých ohledech se však dá odlišit. Digitálně vytvořené prostředí v MR může plně překrývat reálný svět. Prostředí je však na reálný svět stále vázáno. V praxi to znamená, že digitální objekty v prostoru zaujímají neustále stejné místo, v závislosti na reálném okolí. To mimo jiné znamená, že uživatel je schopen interagovat jak s tímto prostředím, tak s reálným světem. Jinými slovy o MR mluvíme v případě prolnutí reálného světa s digitálním, za účasti lidské obsluhy [2] [3].

4 HISTORICKÝ POHLED NA ROZŠÍŘENOU REALITU

Termín rozšířené reality (Augmented reality) byl poprvé použit již v roce 1960. V této době se však stále mluví o Virtuální realitě. Není totiž dobře možné od sebe odlišit tyto dvě technologie, a to z důvodu dosud nezavedené terminologie a strohé počítačové nástavby, která by je dokázala jednoznačně oddělit.

Jako první mechanismus používaný v oblasti AR lze považovat přístroj zvaný The Sword of Damocles. Jedná se rameno, na které bylo zavěšeno promítací zařízení složené ze dvou objektivů. Zdrojem promítacího materiálu byl počítač. V této době však značně primitivní. Promítací zařízení bylo schopno zobrazovat pouze jednoduché geometrické modely v podobě třírozměrné sítě a bylo příliš těžké, na ro aby mohlo být umožněno dlouhodobé používání.

Zařízení však již bylo opatřeno určitým trackovacím zařízením v podobě polohového senzoru. Ten měl za úkol monitorovat pozici a natočení hlavy uživatele. Podle těchto údajů byl poté zpracovaný obraz modifikován, tak aby bylo docíleno správné perspektivy obrazu. Vývoj zařízení The Sword of Damocles přineslo pozitivní výsledky a umožnilo další pokrok v této oblasti. Následující prototyp tohoto zařízení pak bylo možné přizpůsobit požadavkům zákazníka a bylo možné ho namontovat na vojenský vrtulník. V této situaci usnadňovalo pilotům práci při nočních operacích. Pilot byl schopen promítat i ve tmě kontury přistávací plochy. Zařízení tak vlastně fungovalo jako určitý pradědeček nočního vidění [4].



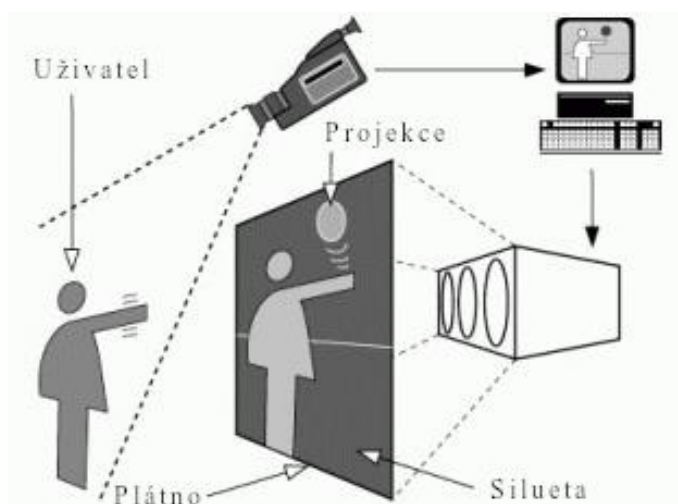
Obr. 1) Zařízení The Sword of Damocles [4].

Další velkým milníkem v oblasti Rozšířené reality bylo založení „Videoplace“ Myronem Kruegerem na konci 70. let. Jednalo se o experimentální a zároveň inovátorskou metodu zobrazování a promítání předmětů. K tomuto účelu bylo použito silného světelného zdroje, který ve ztemnělé místnosti promítal na promítací plátno siluetu uživatele. Tato silueta byla snímána kamerou a vyhodnocována ve speciálním počítači. Po zpracování signálu bylo možné promítnout na správné místo promítacího plátna určený objekt. Nejprve bylo experimentováno s jednoduchými objekty typu míč, krychle etc. Nicméně postupem času bylo možné promítnout i jednotlivé herní prvky a začít tvořit zárodky herního průmyslu v oblasti rozšířené reality.

Metoda promítání siluety byla unikátní také tím že ke snímání obsluhy a následnému zobrazování nebylo použito žádných externích prostředků jako jsou brýle, rukavice či senzory. Ovládání tak bylo pro uživatele velmi jednoduché a intuitivní. Nebyl tedy zatěžován jakýmkoliv jiným zařízením a mohl se plně soustředit na práci se siluetou.

Další iterace projektu Videoplace přinesly nové poznatky a funkce. Bylo již možné zajistit interakce více osob v rozdílných místnostech. Samotné funkce zařízení pak byly rozšířeny na možnost psaní, jednoduché promítání geometrických tvarů a nenáročné hry. Samotný akt psaní byl vyřešen velmi elegantně. Na projekční plochu byla promítnuta řada písmen a operátor pak díky siluetě mohl prstem vybrat jaký symbol si přeje zapsat. Funkce trackování prstů bylo možné jednoduše zrušit zatnutím dlaně v pěst. Napsaný text bylo možné zobrazit na projekční ploše libovolného účastníka simulace,

Pomyslného vrcholu bylo dosaženo po návrhu herní funkcionality „Small Planet“. Jednalo se o velmi ambiciózní projekt. Jak napovídá název, hra byla zasazena do prostředí podobné planetě Zemi, která však byla výrazně zmenšena, aby bylo hráči umožněno prozkoumat v kratším časové úseku větší herní prostor. Systém byl opět postaven na ověřeném konceptu snímání nasvícené siluety uživatele. Hráč mohl svého avatara ovládat pouze pohybem svých končetin. Rozpažení rukou určovalo pohyb hráče kupředu či do stran. Letovou výšku pak bylo možné měnit natočením paží. V samotné herní simulaci bylo opět možné interagovat s ostatními hráči, kteří byli umístěni v podobných místnostech a ovládali obdobné avatary [5].



Obr. 2) Způsob snímání siluety v projektu Videoplace [6].

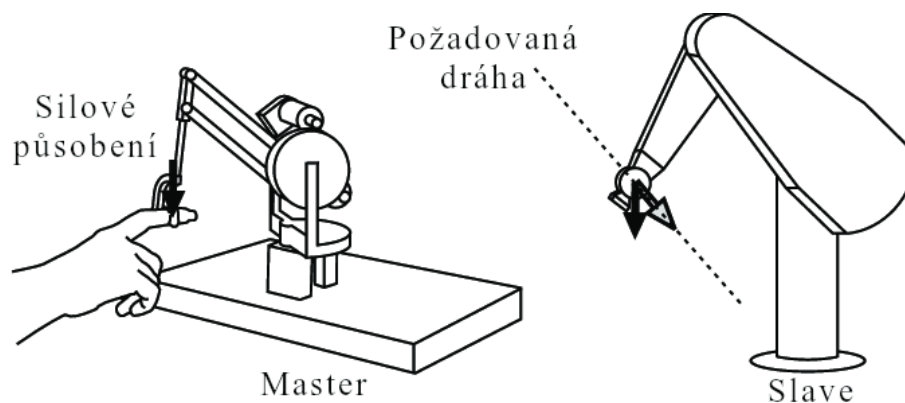
První rozšířená realita moderního druhu byla vytvořena na počátku devadesátých let ve vědecké laboratoři Amerických vzdušných sil. U zrodu stál Luis Rosenberg, který tak v podstatě vytvořil rozšířenou realitu, jak ji známe dnes.

Původní myšlenkou projektu Virtual Fixtures byl ulehčit lidskou práci v oblasti fyzické aktivity. K tomu byla mimo jiné použita robotika a princip promítání rozšířené reality. Funkce byla na dnešní poměry relativně obtížně realizovatelná. K uskutečnění projektu byla použita dvojice reálných robotických ramen vyšší generace. Ta byla uzpůsobena k operování v těžce dostupných, či nebezpečných místech. Samotná ramena byla ovládána uživatelem za pomoci speciálního exoskeletu, který postupně snímal všechny pohyby uživatele. Exoskelet se omezoval pouze na horní část těla. Dokázal tedy snímat v podstatě pouze pohyby paží obsluhy. Daný exoskelet byl poměrně masivní a musel být k člověku upnut po celou dobu provozu.

Uživatel však byl dodatečně vybaven speciálními brýlemi poskytující vhléd do rozšířené reality. Brýle byly vybaveny speciální optikou, určenou k promítání potřebné digitální geometrie. Samotné promítání potom bylo zajištěno projekcí robotických ramen do oblasti paží člověka. V brýlích bylo vidět lidské tělo, ovšem s robotickými rameny namísto paží. Díky tomu bylo možné v reálném čase pozorovat natočení jednotlivých kloubů robotických ramen. Speciální důraz byl kladen na bezproblémový pohyb koncového efektoru robotu.

Mimo jiné bylo možné promítnout i prvky prostředí ve kterém bylo právě operováno. Tento aspekt zvýšil bezpečnost práce s roboty. Nejčastěji se totiž promítaly různé překážky, stěny, popř. zábrany. Značně se také využívalo určit not-go zón. Tedy zón, do kterých se robot z určitých důvodů nesměl dostat. K tomuto účelu bylo mimo jiné využíváno audiovizuálních podnětů. Obsluha tak jednoduše díky zvukovému efektu věděla, kde se zrovna koncový efektor nachází a do čeho nesmí za žádnou cenu narazit.

Systém používání fyzických robotů a mechanického spojení byl nutný zejména z důvodu nízkého výpočetního výkonu grafických zařízení. Spojení fyzického mechanického modelu vedení robotu a robotických ramen, lze do dnešní terminologie částečně převést jako metoda Master-Slave. U takového systému je vždy master veden lidskou obsluhou za použití určité zmenšené síly. Slave prvek potom vykonává danou trajektorii, kterou mu Master přednastavil. V dnešní době se na podobném principu využívají průmyslové roboty kolaborativního typu. Tyto roboty mohou běžně pracovat v pracovním prostředí hned vedle člověka. Není u nich potřeba komplikovaného vedení exoskeletu. Program je možné zadat pouhým vedením robotického ramena. Navíc jsou tyto roboti vybaveny senzory, které zastaví jeho pohyb při kontaktu s překážkou [7] [8] [9].

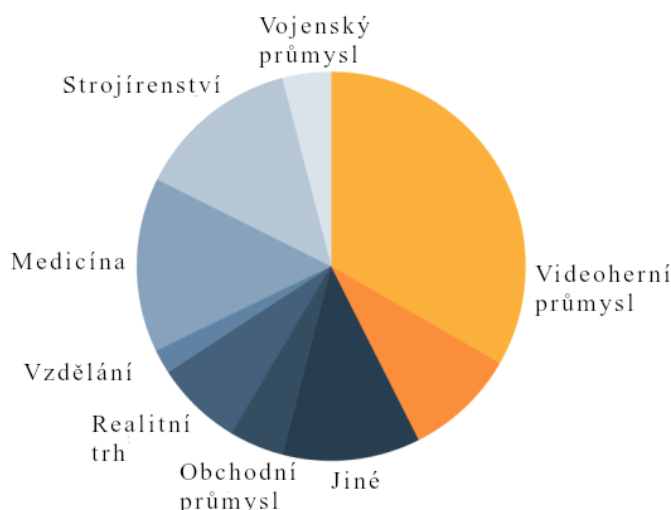


Obr. 3) Příklad soustavy Master-Slave [7].

5 OBLASTI POUŽITÍ ROZŠÍŘENÉ REALITY

Oblasti používání AR i VR se neustále zvětšují. Musíme mít na paměti, že toto odvětví se v moderní podobě rozvíjí teprve poslední desetiletí. Navzdory této skutečnosti se však stále vyvíjí a je implementována do nejrůznějších odvětví. Největší zastoupení má v herní oblasti. Z průmyslových sekcí potom dominuje oblast medicíny, strojírenství či vojenské aplikace. Je zde předpoklad, že trend v těchto oblastech bude i nadále pokračovat.

Předpokládané oblasti použití VR/AR do roku 2025



Obr. 4) Trendy v používání VR/AR [10].

5.1 Medicína

Rozšířená realita, popřípadě i virtuální realita, získává v lékařství stále větší prostor a začíná být dokonce vyvíjen tlak na implementaci této funkcionality do lékařského odvětví. Důvodů k implementaci může být hned několik. Mezi nejdůležitější lze zařadit například:

- Pro značnou část lékařských zákroků je vytvořen, alespoň částečný 3D model pacienta, který může být jednoduše použit i v oblasti AR.
- Překotný rozvoj umělé inteligence a její možné využití v medicíně jakožto i v aplikacích AR.
- Rozvoj 3D tiskáren a jejich snadná dostupnost. Opět je možné využívat stejná data pro tisk modelů jako pro geometrii v AR.
- Používání typizovaných senzorů a trackerů v lékařské vědě. Podobné senzory je možné s úspěchem využít i pro AR.

Aplikace využívající rozšířenou realitu najdou uplatnění i díky postupné digitalizaci pracovišť. Většina nemocnic přechází na kompletní vedení záznamů v digitální podobě. Týká se to zejména personálu a pacientů. Obdobně může být nakládáno i s daty lékárenského vybavení, léčiv, popřípadě jiných nutných prostředků. Lékař si může jednoduše ve vhodné

aplikaci vyhledat potřebné informace o pacientovi a přizpůsobit jeho léčbu aktuálním informacím.

Podobná data pak mohou být předkládána i pacientům za účelem jejich samostudia onemocnění. Většina pacientů již dnes vlastní chytrý telefon, do kterého je možné nahrát potřebné informace. Vizualizace v podobě promítání dodatečných modelů může být v tomto ohledu velice prospěšná a bude mít za následek menší vytížení nemocničního personálu.

Poměrně hojně zavedenou možností používání AR je školení zaměstnanců či studentů na lékařských fakultách. Je relativně snadné vytvořit jednoduchou simulaci, například rutinního lékařského zákroku a tu pak dále implementovat do systému počítačové projekce. Předpokládá se, že operací již bylo provedeno velké množství a není problém získat dostatečné množství dat pro tvorbu simulace. Může se jednat například o 3D modely kostry či svalstva. Jejich vlastnosti, a zejména vhodnou léčbu.

Aplikaci rozšířené reality je možné využívat přímo na operačním sále. Mezi již zavedené a plně funkční technologie patří trackování trasy nástroje. Jednotlivé 3D kamery lze upnout k používaným přístrojům na operačním sále. Díky tomu nezabírají místo, které potřebuje využívat lékařský personál. Často se na velmi malém místě v okolí operovaného tělesa i desítky lidí. Není tedy prostor pro omezení dalšího místa. Kamery, instalované obvykle u stropu, v reálném čase trackují používaný nástroj, či přímo ruce lékaře. Při pohybu nástroje je počítač schopen dopočítat trajektorii nástroje a díky tomu predikovat, jestli nenapáchá škody na těle operovaného. Tuto trajektorii potom v podobě linie zobrazuje buď na displej či do speciálních brýlí operujícího lékaře [11].

5.1.1 NuEyes

Jedním z přístrojů využívajících rozšířenou realitu je produkt NuEyes. Jedná se o zdravotnickou pomůcku, která má za úkol pomáhat pacientům se zrakovými potížemi. Jde o lehké brýle, které mohou být nošeny i dlouhodobě. Obsahují videokameru umístěnou na přední části brýlí. Ta se stará o snímání obrazu a jeho následné promítání do vnitřního prostoru.

Pacienti trpící zrakovými obtížemi velmi často nejsou schopni ovládat podobné pomůcky, a to pouze z důvodu, že na ně prostě nevidí. Proto byla vyvinuta funkce ovládání hlasem. Aparát je možné ovládat jednoduchými pokyny: „Zapnout, zvětšit, přiblížit“ etc. Maximální přiblížení je dvanáctinásobek původního obrazu. Využití lze nalézt při četbě, kdy se kamera automaticky fokusuje na čtený text, a ten následně zvětšuje a promítá. Další použití je u sledování televize, či fotografií. Je možné také dodatečně modifikovat obraz například změnou poměru barev či kontrastu [12].

Mimo jiné je v některých státech, díky zdravotnímu pojištění, možné získat finanční podporu až do výše 50%. Díky tomu se přístroj stává dostupným pro široké spektrum pacientů.

Důležité vlastnosti přístroje:

- 1300 mAh baterie zajišťující provoz po dobu 10 hodin.
- Celková fyzická paměť 64 GB.
- Dva stereoskopické displeje s rozlišením 720p při 120 fps.
- Senzory zajišťující gyroskopickou stabilitu ve 3 osách.
- Software Android, s implementovanou knihovnou Vuforia.



Obr. 5) Zdravotní pomůcka NuEyes Pro [13].

5.1.2 Brain Power

Další zajímavou aplikaci přináší projekt Brain Power. Ve spolupráci s Google glasses je vyvíjena aplikace, speciálně zaměřena na děti postižené autismem. Tyto děti mají obtíže při rozpoznávání emocí okolních lidí a z toho plynoucí problémy s reakcí na tyto emoce. Ruku v ruce s tím jdou i problémy s projevy vlastních emocí. Toto je pak velmi znevýhodňuje v kolektivu

Funkce přístroje je opět velice podobná na přístroj NuEyes. Kamera snímá místa, na které se pacient dívá a řídicí systém vyhodnocuje obdržené informace. V tomto případě snímá obličeje okolních lidí a podle stahů obličejových svalů jim přiřazuje určitou emoci. Nejčastěji radost, smutek, rozčilení a jiné. Následně systém promítne na promítací část brýlí různé možnosti emocí a dětský pacient se následně musí rozhodnout o jakou emoci se jedná. Svou odpověď poté potvrdí nakloněním hlavy. Při správné odpovědi je odměněn kladnými body, zatímco při nesprávné odpovědi ohodnocen není a dostane možnost svoji odpověď opravit.

Aplikace je také vybavena Bluetooth systémem, k propojení s mobilem rodičů. Rodič tak může v reálném čase sledovat průběh léčby. Mimo jiné také má možnost „odměnit“ svou ratolest bonusovými body či jinou digitální odměnou.

Díky tomu je dítě učeno hrou a za své úspěchy je odměněno. To posiluje pocit zadostiučinění ze zdolaného úkolu. Cílem tohoto projektu je pomoci pacientům zvýšit sociální citění, jejich motivaci a možnost plně fungovat ve společnosti. Zároveň tlumí projevy hyperaktivity a agresivního chování [14].

5.1.3 SCOPIS

Systém Scopis nachází své místo v oblasti lebečních zákroků. Při provádění takto závažné operace není přípustné udělat jakoukoliv chybu. Proto se také lékaři snaží omezit toto riziko moderními metodami a přístupy.

Jedná se o poměrně obsáhlý SW, který je navzdory své velikosti a složitosti uživatelsky dostatečně přijatelný. Lékař před začátkem samotné operace vyvolá z databáze předem vytvořené 3D modely pacienta. Ty jsou pořízeny s časovým předstihem. Nadále zadá bod, který vybere pro vstup nástroje do lebky. SW už poté s jemnými zásahy uživatele automaticky dopočítává potřebnou trajektorii pracovního nástroje.

Po dokončení přípravné procedury je daný plán operace uložen do počítače a je přistoupeno k samotnému zákroku. Systém trackuje pozici nástroje a zároveň lékaři zobrazuje potřebnou trajektorii. Ta je transparentní a je ohraničena různě zbarvenými elementy pro lepší orientaci v prostoru. Při dosažení určitého bodu změní značka barvu a lékař může pokračovat ve zbývajících částech zákroku [15] [16].

5.2 Vojenské použití

AR se využívá ve vysoké míře pro účely školení vojenského personálu, nejčastěji pilotů či výsadkářů. Cílem je poskytnout autentický zážitek. Zároveň je kladen důraz snížení nákladů spojených s výcvikem.

Díky simulacím proto není nutné vyzbrojit celou jednotku, převést ji na určené místo cvičení a provést výcvik. Některé části výcviku je možné provádět přímo v kasárnách v jedné místnosti. Jedná se zejména o drilové akce, kdy je kladen důraz na správné provedení úkonů, a hlavně na to, aby v kritické situaci voják nezpanikařil. Může se jednat o simulace různých nepřehledných situací. Například střelba na nepřítele, popř. zastavené střelby v okolí civilistů, nácvik boje v extrémním počasí, orientace v neznámém terénu etc.

5.2.1 Výcvik pilotů a výsadkářů

Piloti stíhaček či vrtulníků jsou tradičně vybaveni helmou s příslušenstvím. Toho je vhodně využito právě v oblasti AR. Pilotovi je možné do brýlí rovnou promítat potřebné informace, jako je rychlost, výškoměr, popřípadě komunikaci s týmem. Omezí se tím zbytečné ohlížení pilota a zvýší se jeho ostražitost.

Výsadkáři mohou být naopak cvičeni na různé typy poruch padáků. Při poškození padáku je nutné nejprve poškozený padák odříznout a následně otevřít padák rezervní. Je jasné že tyto druhy akcí by byly v reálu velice obtížně cvičitelné. Právě proto je v těchto oblastech úspěšně používáno AR/VR systémů.

5.2.2 Tactical Augmented Reality

TAR je zvláštní způsob použití AR ve vojenském průmyslu. Jedná se zařízení velmi podobné brýlím pro noční vidění. Je však možné ho napojit na GPS a díky tomu zobrazovat nositeli jeho přesnou polohu. Při vybavení celé jednotky je pak možnost zobrazovat jejich pozice, jakožto i umístění nepřátel, či jiných cílů.

TAR je také možno vybavit speciálními teplotními senzory umístěnými na zbraní a zápěstí. Promítací zařízení tyto senzory detekuje a je schopno zobrazit trajektorii střelby i vzdálenost cíle.

Jeho dalším benefitem je možnost rozdělit displej na více částí. Ty mohou být tvořeny záznamem z kamery usazené na přilbě a další na hlavní zbraně. Natočením zbraně je tedy v podstatě možné vidět za roh, aniž by došlo k přímému vystavení nebezpečí střelby.

Systém TAR začíná být hojně využívána součástí výbroje amerických vojenských složek. Eliminuje nebezpečí, dovoluje zobrazovat taktické informace a zmenšuje množství vybavení, jež sebou musí voják nést [17].

5.2.3 ARES

Poměrně jednoduchou aplikaci AR nabízí systém ARES. Jedná se o zobrazovací zařízení podobné mapě. Skládá se ze senzoru, projektoru, počítače a speciální nádoby s pískem. Pro detailnější popis taktické situace se dříve používaly poměrně jednoduché pískové „mapy“, doplněny o plastové figurky, reprezentující situaci. ARES využívá této skutečnosti a díky senzorům je schopen monitorovat množství písku navrstveného na určitém místě. Prohlubeň v písku potom znamená kaňon či řeku. Navrstvený písek reprezentuje pohoří, či budovu. Projektor umístěný nad touto mapou promítá potřebná data přímo na písek. Je schopen promítat jak komplexy budov, tak vojenská vozidla a letouny. Vše probíhá v reálném čase. Může být navíc doplněn o animace, ozbrojené střety, pohyby jednotek etc.

V principu jde o velmi jednoduchý a levný systém, který však pro svou jednoduchost a přehlednost nalézá značné uplatnění [17].



Obr. 6) Využití systému ARES v praxi [18].

5.3 Průmyslové použití

Použití v průmyslu je jednou ze stěžejních složek AR. Často se používá za účelem úspory času v montovnách. Eliminuje možnost chyby obsluhy a dává supervizorům schopnost lépe řídit personál. Právě z těchto důvodů se AR začíná stále více uplatňovat v průmyslovém odvětví.

5.3.1 Dokumentace a manuály

Některé společnosti postupně přechází k vedení dokumentace pouze v digitální podobě. Vede je k tomu zejména Digitalizace a Průmysl 4.0.

Stávající vedení výkresů a dokumentace ve 2D podobě přestává být vhodné. Je běžné že ve firmě se nachází velké množství dokumentů a výkresů. Při změně některého z nich musí být zbytek dohledán a aktualizován. Často však dochází k tomu, že některé výkresy jsou špatně založeny. Jsou zastaralé nebo jinak nevyhovující. To pak může vést ve výrobě k fatálním chybám, kdy se výrobek sice vyrábí správně, ale podle neaktuálního výkresu. Tím pak dojde k vyrobení špatného kusu, který není možné smontovat se zbytkem sestavy.

Řešením může být mimo pečlivosti například využití AR namísto některých výkresů či manuálů. Personál si jednoduše promítne na tabletu či mobilu potřebnou součást. Zabrání se tak chybám spojených s chybnými výkresy a manuály.

Další nespornou výhodou jest, že se informace v digitálním prostředí mohou rychle a snadno aktualizovat. Pokud konstruktér provede na některém prvku změnu, je jednodušší ji dostat na potřebná místa než při využití stávající metody 2D výkresů, kde může trvat i týdny, než zprávu se změnou obdrží všechny instance.



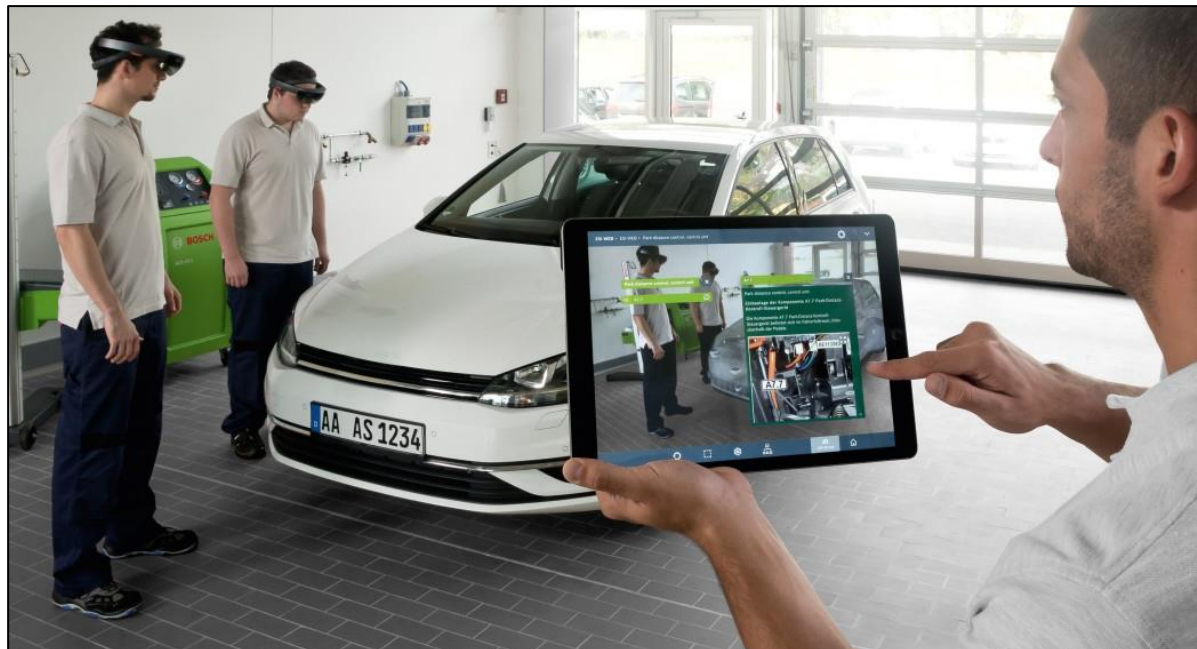
Obr. 7) Příklad použití AR jako montážního manuálu [19].

Jako jednoho z hlavních představitelů implementace digitální dokumentace do výrobních podniků lze uvést společnost Re'flect. Tato firma se specializuje na průmyslovou výrobu převážně v automobilovém a chemickém průmyslu.

Specializuje se zejména na převádění či tvorbu 3D modulů pro potřeby AR. Využití nalézá jak při zaučování nových zaměstnanců, tak při údržbě a servisu stávajících zařízení. Hlavní výhodou je možnost udat vnějším stěnám produktu průhlednost. Je tak možné v podstatě vidět „dovnitř“ zařízení jako například čerpadla, motoru, rozvaděče etc. Při zobrazování je také možné vybrat variantu zobrazení sestavy. Zařízení se vlastně rozloží na jednotlivé součásti a je

tak pro operátora snadnější sestavovat zařízení, když vidí postupně, jaké kroky musí podniknout.

Re'flect spolupracuje s velkým spektrem světových výrobců. Lze jmenovat například Microsoft, ABB nebo BOSCH. Není tak s podivem, že při objednávce dostane zákazník mimo samotného programu i Hololens, právě z dílny Microsoft [20] [21].



Obr. 8) Školení zaměstnanců společnosti BOSCH při použití AR [22].

Jako dalšího výrobce produktů na bázi AR lze uvést společnost DAQRI. Tato uvedla na trh vlastní software worksense, který je implementován do jedinečných brýlí lakonicky pojmenovaných Smart Glasses. Je zde tedy předpoklad tvorby SW přímo na míru. V brýlích je totiž předinstalovaná celá sada nástrojů a aplikací, které pomohou urychlit implementaci AR do chodu firmy.

Samotné brýle jsou velmi zdařile provedeny. Nabízí ergonomický design relativně vysokou výdrž baterie a adekvátní kamery s dostačujícím rozlišením. Součástí výbavy je taktéž poměrně výkonný procesor i7, šesté generace. Od koupi by mohla odrazovat poněkud vyšší váha téměř 500 g, či nižší obnovovací frekvence obrazu. Toto je však současný problém většiny moderních výrobců zařízení AR/VR.

V systému je implementována možnost videohovorů. Díky tomu může obsluha rychle dostat adekvátní rady na pracovišti od přiděleného supervizora. Operátor hovoru totiž jednoduše vidí, na co se pracovník dívá a může ho instruovat. V případě potřeby je možnost zobrazení dodatečných informací. Například umístění komponenty, datasheet, rozvody etc.

Zajímavou funkcionalitou je i možnost skenování okolí. Kamera jednoduše snímá okolí a ukládá ho s určitou vzorkovací frekvencí. Tímto způsobem je možné vytvořit 3D model pracoviště v měřítku 1:1. Do tohoto modelu je pak možné vkládat 3D prvky, varování, či kritická místa, která vyžadují zvláštní pozornost. Uplatnění nalézá převážně v obsluze a servisu [23].



Obr. 9) Použití AR DAQRI při identifikaci vadného ventilu [24].

5.3.2 Návrh a konstrukce

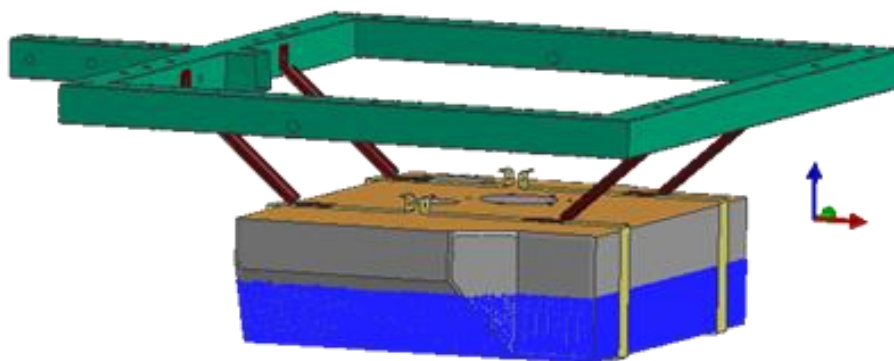
VR se výrazně uplatňuje i na poli konstruování a návrhu výrobku. Nemusí se při tom jednat pouze o zobrazování součástí. Běžnou praxí je návrh konceptu přímo ve VR. Jako dílčí činnost je možné použít i výpočet různých pevnostních výpočtů nebo řešení matematických rovnic.

Dá se v podstatě mluvit o určité nehmotném prototypování. Primární koncept a design je proveden pouze v digitální formě a může být zkontrolován se zákazníkem. Užitím tohoto konceptu se sníží náklady potřebné na vývoj produktu. Dále se sníží možnost chyby návrhu a je tak možné dodat zakázku průměrně dříve než při použití konvenčního vývoje.

Uplatnění je převážně v automobilkách s větším množstvím pracovníků. Popřípadě ve stavitelství a podobných oborech, kde je nutné řešit systémy o velkém množství prvků a neznámých.

Jako jednoho z představitelů lze uvést firmu IC.IDO. Jedná se o zástupce zaměřujícího se na VR v oblasti automobilového průmyslu. Uplatnění je převážně v digitálním prototypování automobilů. Hlavní výhodou je omezení iterací návrhu. V současné době většina automobilek cílí na snížení hmotnosti a emisí při zachování bezpečnosti a výkonu auta. Tomuto trendu je možné čelit pouze za pomoci moderních technologií.

S rozvojem elektromobilů přichází trend „čisté“ energie. Klasická auta jsou koncepčně přepracována na elektromobily. Největší výzvou však stále zůstává napájení auta. Je nutné zajistit dostačený dojezd, jakožto i rychlost nabíjení. Mezi inovativní koncepty patří vybavení auta přívěsem s dodatečným akumulátorem. Na to cílí společnost EP Tender. Při vývoji a výzkumu tohoto zařízení využívá právě VR. A to při návrhu konceptu, a při zkouškách bezpečnosti. Ve VR je možné navržený systém zkoumat z mnoha hledisek. Je také možné nasimulovat nehodu nebo srážku s jiným vozidlem. V případě přívěsu s baterií šlo o návrh speciálního nosiče baterie, který sníží riziko poškození přívěsu při autonehodě. Díky tomuto přístupu se dostala před konkurenci, při použití menších finančních prostředků nutných na vývoj [25] [26].



Obr. 10) Návrh konstrukce nosiče baterie [27].

6 SYSTÉMOVÝ ROZBOR PROBLEMATIKY

Při návrhu aplikace rozšířené reality je nutné zvážit velké množství řešení a variant. Do samotného postupu řešení vstupuje velké množství neznámých, které musí být definovány a vyřešeny.

K vyřešení těchto problémů je vhodné do procesu řešení zařadit systémový rozbor dané problematiky. Systémový rozbor dává do spojitosti věci, které by měly být vyřešeny a jsou esenciální pro správné řešení návrhu. Použití této metody výrazně zvyšuje pravděpodobnost bezchybného řešení daného problému.

6.1 Definice aktuální situace

Jak již bylo popsáno v kapitole 5, mechanismy AR/VR jsou na vzestupu. Světové společnosti na tuto realitu reagují zvýšenou poptávkou po těchto produktech. Dalším aspektem je implementace AR do výroby a dále vybavení pracovního personálu touto technologií. Před výrobci AR systémů nyní stojí nelehký úkol. Tím je zaprvé navržení potřebné technologie přímo na míru daného podniku. Druhou možností je navržení univerzálního produktu, který bude splňovat požadavky většího spektra společností. Je zřejmé, že univerzální produkt nikdy nebude možné nasadit v každé situaci. Výrobní závod strojírenské společnosti bude mít jistě odlišné požadavky na funkcionalitu a bezpečnost než například pracovník administrativní činnosti. Dále lze předpokládat, že i jednotlivé výrobní závody budou požadovat své specifické odlišnosti.

6.2 Definice problému

Návrh aplikace rozšířené reality pro obráběcí stroj MCV bude navržen ve vhodném vývojářském prostředí. V našem případě se jedná o prostředí UNITY. Bude obsahovat uživatelský interface a trackovací/promítací systém. Ten se bude odvíjet od výpočetního výkonu zařízení a zástavbovým rozměrům stroje.

6.2.1 Definice zkoumaného systému

Návrh aplikace je určen pro obráběcí stroj MCV 754 QUICK. Jedná se vertikální obráběcí centrum. Pohyb je možný ve 3 osách. Přičemž pohyb X/Y je realizován posuvem pracovního stolu a pohyb v ose Z zajišťuje vertikální pohyb vřetene. Stroj je možné navíc vybavit otočným či sklopným stolem, jež zajistí pohyb ve 4 či 5-ti osách.



Obr. 10) Obráběcí stroj MCV 754 QUICK [28].

Stroj je také vybaven přímým odměřováním, který zajišťuje přesné polohování. Na odměřovacím pravítku jsou speciální značky. Stroji stačí najet na jednu z těchto značek pro každou osu. Díky tomu poté „ví“ kde je a dokáže najít referenční polohu. Nedílnou součástí je také zásobník nástrojů, který pojme až 24 nástrojů, v závislosti na jejich velikosti. Samotná výměna je realizovaná výměníkem nástrojů v podobě ruky s dvěma polohami.

Stroj je vybaven řídicím systémem HEIDENHAIN a SIEMENS SINUMERIK.

Tab. 6.2 Technické parametry stroje MCV [29].

Technické vlastnosti	MCV 754 QUICK	Jednotky
Zástavbové prostory	2320 × 2590 × 2560	mm
Maximální příkon stroje	20	kVa
Výkon motoru (S1/S6 – 40%)	9/13	kW
Jmenovitý kroutící moment (S1/S6 – 40%)	57/83	Nm
Upínací plocha stroje	1000 x 500	mm
Pracovní rozsah X-osa	754	mm
Pracovní rozsah Y-osa	500	mm
Pracovní rozsah Z-osa	500	mm
Hmotnost stroje	4 000	kg
Maximální otáčky	10 000	min ⁻¹

6.3 Formulace cílů

Diplomová práce si dává za cíl vyřešit následující:

- Výběr vhodného vývojářského prostředí pro návrh aplikace rozšířené reality a zhodnocení jeho parametrů.
- Výběr vhodné knihovny a databáze pro trackování a promítání součástí.
- Návrh aplikace AR.
- Návrh interface pro danou aplikaci.
- Testování aplikace a zhodnocení funkčnosti.
- Závěrečné shrnutí a doporučení do budoucna.

6.4 Systém veličin

Ke vhodnému provedení systémové rozboru je nutné nadefinovat hlavní veličiny vstupující do koncepce návrhu. Jedná se především o veličiny výrazně ovlivňující celý systém. Zanedbáním některé veličiny můžeme způsobit fatální chybu v našem systémovém pojetí. Při návrhu aplikace AR bude hrát největší roli výběr platformy vývojářského prostředí. K tomu bude nutné stanovit parametry jednotlivých možností a vybrat z nich v závislosti na požadavcích tu nejvíce vhodnou [30].

6.5 Definice a výběr vývojářského prostředí

Při návrhu aplikace pro údržbu obráběcího stroje je nejdůležitější výběr vhodného vývojářského prostředí. Tuto volbu je nutné učinit na počátku návrhu. Kdybychom chtěli změnit platformu v průběhu tvorby, vyvstanou neočekávané problémy a spoustu entit nebude možné bez poškození přesunout z jednoho prostředí do druhého. Nejprve je tedy nutné se zamyslet nad vlastnostmi, které považujeme za důležité a které jsme naopak ochotni upozadit. Je jasné, že například možnost tvorby otevřeného fotorealistického světa pro nás nebude mít takovou váhu jako možnost jednoduše naimportovat knihovny a databáze.

V současné době jsou k dispozici hlavně systémy UNITY a UREAL ENGINE. Dá se říci, že jde o dlouholeté rivaly na poli vývojářských platform. Obě platformy mají své klady i zápory, byť na první pohled se může zdát, že jde o relativně identické prostředí.

V následujících odstavcích budou podrobně vypsány stěžejní prvky a vlastnosti těchto systémů. Bude posouzena jejich vhodnost při použití na AR aplikaci v průmyslovém sektoru.

6.5.1 Grafické rozlišení

Jedním z hlavních faktorů pro posouzení vývojářského prostředí je grafické rozlišení. Jedná se zejména o kvalitu zobrazovaných textur. Popřípadě i další utility. Například tvorba Assetů, Shaderů nebo tvorba a kvalita jednotlivých materiálů. Jako vstup zde může být i tzv. fotorealismus. Jde o schopnost systému dosáhnout kvality textury objektu, která je k nerozeznání od reálného světa. K tomu herní platformy využívají různé renderovací procesy. Důležitým faktorem je také tzv. microdisplacement. Při aplikaci textury, například kovového povrchu, je na 3D objekt v podstatě pouze nalepena určitá barevná paleta. Nebere v potaz různé úhly, pod kterými by měla odrážet světelné paprsky. Objekt se tak zdá plochý a bez hloubky. Neinteraguje s okolním světem tak jak bychom očekávali. Při aplikaci microdisplacementu

musíme však brát v potaz vyšší renderovací čas, což může být kritickým faktorem při tvorbě AR aplikace. Podobného efektu lze dosáhnout i tzv. fake microdisplacement. Jde převážně o funkce typu bump. Ty pouze simulují správné odrazení světelných paprsků. Dosahují adekvátních grafických výsledků za zlomkový renderovací čas. Při bližším pohledu je však jasné, že se nejedná o reálný objekt.

V grafické oblasti poměrně s přehledem vede UNREAL ENGINE. Dosahuje průměrně lepšího obrazu než UNITY. Je taktéž nesrovnatelně jednodušší vytvořit nový shader nebo texturu.

Další výhodou UNREAL ENGINU je možnost snadnějšího importu, již vytvořeného materiálu například z Blenderu. UNITY má o něco horší možnost tvorby adekvátního grafického rozlišení. Má sice implementovaný vlastní modul pro jeho tvorbu. Je však pomalý, tvorba trvá dlouhou dobu a kvalita není uspokojivá.

Z těchto důvodů je pro aplikaci, která sází na grafickou kvalitu vhodné zvolit UNREAL ENGINE. To je také důvod, proč je oblíben u herních vývojářů. Uplatnění může nacházet i u filmových snímků, které používají velké množství CGI prvků. Příkladem budiž snímek Mandalorian, u kterého bylo v podstatě veškeré prostředí vytvořeno v UNREAL ENGINU.

6.5.2 Výpočetní výkon

Ruku v ruce s vysokým grafickým rozlišením jdou i nároky na výpočetní výkon zařízení. UNREAL požaduje poměrně výkonné zařízení. Většina vývoje se provádí na stolním PC. Výsledná aplikace je ve výsledku určena pro podobné stolní zařízení. To je také důvod častého využití vývojářů PC her či filmařů.

Na druhou stranu UNITY je více zaměřena na zařízení s menším výkonem. Mezi ně patří zejména mobilní telefony, tablety, konzole a zařízení pro AR. Není však nutností využívat UNITY pouze k těmto aplikacím. Lze vytvořit i plnohodnotný PC program. V naší aplikaci je však výhodné použít právě UNITY z důvodů jednoduché realizace na mobilním telefonu či tabletu.

6.5.3 Velikost vývojářského týmu

Při návrhu a vývoji aplikace je jedno z nejdůležitějších hledisek velikost, složení a zkušenosti vývojářského týmu. Ne vždy znamená, že větší je lepší. V dnešní době dávají vývojáři přednost spíše menšímu týmu, složenému ze specializovaných odborníků. V takovém případě se například jeden vývojář zabývá pouze logikou, další pouze texturami. Někdy vstupují do hry i specializace jako particle effect (exploze, kouř, oheň etc.). Tohoto rozložení využívá nejvíce UNREAL ENGINE. Důvodem je větším komplexnost systému a požadavek na větší znalosti problematiky. Odměnou za dodatečné znalosti je lépe vizuálně vypadající aplikace.

UNITY na druhou stranu lze relativně bezproblémově obsluhovat jako jedinec, či jako malý vývojářský tým např. o 5 lidech. Důvodem je intuitivní ovládání a rozdělení UNITY do tzv. Assetů. Jedná se o oddělené bloky logiky. Při vývoji se tedy vytvoří blok např. 3D modelů. Jakmile je blok uspokojivě vytvořen, pokračuje se k dalšímu a předpokládá se, že předcházející blok funguje a nemusí se nijak výrazně měnit.

6.5.4 Použití převzatých entit a databázi

Nespornou výhodou Unity je taktéž Asset Store. Jedná se o jakýsi „obchod“ který je implementován do samotného systému UNITY. Zde je možné si zakoupit různé prvky nebo bloky, které můžeme využívat ve vlastní aplikaci.

Ne všechny prvky jsou však zpoplatněny. Značnou část je možné stáhnout a importovat zcela zdarma. Díky tomu se vývojáři nemusí zdržovat s vývojem něčeho, co již bylo vyvinuto někým jiným. Například různé druhy kamery. Logika převádění datových typů nebo i 2D textury. Toto velmi usnadní práci jednotlivce.

6.5.5 Typ aplikace

Důležitým faktorem výběru je druh aplikace, který vyvíjíme. Z výše zmíněných poznatků je očividné, že po UNREALu sáhnou spíše grafičtí designéři a vývojáři herních aplikací. Pro jedince, kteří vyhledávají spíše jednoduchost a funkčnost je zde Unity.

Je však nutné poznamenat že obě značky si neustále snaží konkurovat. V praxi to znamená že UNREAL je každým rokem intuitivnější a uživatelsky přívětivější. Naproti tomu UNITY přidává do svého buildu profesionální nástroje, díky nimž dosahuje obdobných výsledků jako UNREAL.

V budoucnu se předpokládá trend, kdy obě platformy dosáhnou zhruba obdobné kvality a budou využívány podobným procentem uživatelů [31] [32].

6.5.6 Zhodnocení a výběr platformy

Pro posouzení a zhodnocení daných parametrů byla vytvořena přehledná tabulka 6.5. Dává do souvislosti výše zmíněné poznatky. Pro obě platformy UNITY i UNREAL byly stanoveny stěžejní parametry. U těchto parametrů byl stanoven jejich dopad jakožto i číselná hodnota váhy důležitosti. Dané číselné hodnoty jsou subjektivní a budou hrát roli pouze při návrhu této specifické aplikace.

Váha byla stanovena na stupnici: 1 (nejméně vhodné) – 5 (nejvíce vhodné).

Tab. 6.5 Porovnání parametrů UNITY a UNREAL ENGINE.

Typ platformy:	Unity		UNREAL ENGINE	
	Dopad	Váha	Dopad	Váha
Grafické rozlišení:	Střední	3	Vysoké	4
Požadavek na výpočetní výkon:	Nízký	5	Vysoký	2
Požadavek na vývojářský tým:	Nízký	4	Střední	3
Vlastní databáze:	Ano	3	Ne	1
Typ aplikace:	Mobilní	4	PC	2
Σ	19		12	

Porovnání daných sumací dává jasný závěr. A to že pro návrh naší zvolené aplikace je vhodné použít vývojářské prostředí UNITY. Nejdůležitější faktory jsou nízký požadavek výpočetního výkonu zařízení a možnost jednoduché tvorby mobilní či konzolové aplikace.

7 VÝBĚR KNIHOVNY

Při samotném návrhu aplikace je dále nutné vybrat vhodnou metodu trackování a promítání. Výhodou UNITY je mimo jiné i možnost využívat externích knihoven. Není proto nutné vyvíjet vlastní metodu snímání obrazu. Stačí zvolit vhodnou variantu z nabízených. Většina variant přichází s v tzv. SDK verzi. Jde o zkratku Software Development Kit. Z názvu plyne že jde o systém funkcionalit, které vývojáři značně usnadní práci. Ty mohou být děleny podle obsahu, nebo možnosti použití. Např. iOS, android etc.

7.1 Výběr trackovací SDK metody

Pro adekvátní a podložený výběr metody je opět nutné srovnat jednotlivé možnosti a posoudit jejich vhodnost použití. V tomto případě máme na výběr hned několik variant. Pro přehlednost budou vynechány varianty zastaralé a na první pohled nevhodné. Varianty budou posuzovány z několika hledisek.

7.1.1 Cena a licence

Pro aplikaci v oblasti údržby obráběcích strojů je rozhodující cena dané knihovny. Většina knihoven je zdarma pro nekomerční použití nebo pro účely studia. Pro komerční použití pak výrobci nabízejí speciální licenci, tomuto uzpůsobenou. Na trhu se však nalézají i knihovny, které jsou zcela zdarma, bez ohledu na oblast použití.

7.1.2 Kompatibilita s Unity

Nejdůležitějším faktorem ve výběru vhodné varianty je jednoznačně možnost implementování knihovny do prostředí Unity. Při nesplnění tohoto faktoru nemá cenu brát danou metodu v potaz.

7.1.3 Marker tracking

Možnost knihovny načítat Markery. Marker je předem definovaná značka, kterou má systém v paměti. Jde obvykle o černou plochu s bílými nesouměrnými značkami po obvodu. Kamera snímá marker a systém dopočítává vektor natočení vzhledem ke kameře.

7.1.4 Natural Feature tracking

Alternativa k trackování markerů. Opět se jedná o image-based tracking. Systém trackuje uživatelem zvolený obraz. V něm vytvoří definiční body. Ty se obvykle objeví na ideálně zvolených místech. Můžou to být rohy, hrany, nebo jasné barevné přechody mezi tmavou/světlou barvou. Výhodou je možnost trackovat jakýkoliv obrázek. Nevýhodou jest možnost využití nevhodné předlohy. Například obraz nedostatečné kvality, neostrý snímek nebo snímek příliš barevný [33].

7.1.5 3D Object tracking

Jde o rozličné možnosti detekovat požadovaný 3D objekt. Pro výběr knihovny jde o opět kritický faktor. Může se jednat o objekty vytvořené uživatelem s rozmanitou geometrií. Nebo o jednoduché geometrické objekty předdefinované knihovnou, u kterých uživatel pouze zvolí několik jednoduchých atributů. Příkladem budiž válec se vstupními parametry poloměr a výška.

7.1.6 IMU Senzory

Jedná se o možnost přístupu k tzv. Inertial sensors. Jde obvykle o zabudované senzory v použitém zařízení. V našem případě jde o senzory zaznamenávající natočení a zrychlení mobilního telefonu a tabletu. V dnešní době jde o poměrně běžnou výbavu všech zařízení, a proto není potřeba dodatečně programovat tyto senzory.

7.1.7 Visual Search

Schopnost zadat do systému tvar objektů, které chceme trackovat. Při použití kamery potom systém vyhodnotí obdržený obraz a zhodnotí, zda je přítomný podobný předmět, který byl zadán. Jedná se kritérium důležité, ne však nezbytné.

7.1.8 Face tracking

Možnost vybavit systém schopností rozpoznávat obličeje, popřípadě jejich změnu, například úsměv. Při některých aplikacích může být kladen důraz i na tuto technologii. V oblasti obráběcích strojů však není příliš relevantní.

7.1.9 GPS

Vybavení systému dodatečným ukládáním polohy a souřadnic přístroje. Některé moderní aplikace této možnosti hojně využívají. Jedná se však o aplikace z oblasti herního průmyslu a různé navigátory. V našem případě jde opět o funkci, kterou není třeba implementovat do systému.

7.2 Posouzení metod

Do užšího výběru bylo zahrnuto celkem 7 možných kandidátů. Jedná se o knihovny, které jsou aktivní a pravidelně aktualizovány. Knihovny jsou posuzovány z výše zmíněných hledisek. Knihovny, které nesplňují požadavek implementace do Unity anebo možnost trackování 3D objektů jsou vyřazeny z výběru. Jednotlivým kritériím byla opět přiřazena číselná hodnota, popisující její důležitost. Některá kritéria jsou navíc doplněna o znaménko ±. To vyjadřuje, jestli je kritérium splněno nadstandardně (+), či s výhradami (-). Na výslednou váhu však nemá vliv.

Váha byla stanovena na stupnici 1 (nejméně vhodné) – 5 (nejvíce vhodné).

Tab. 7.2 Posouzení jednotlivých knihoven [34].

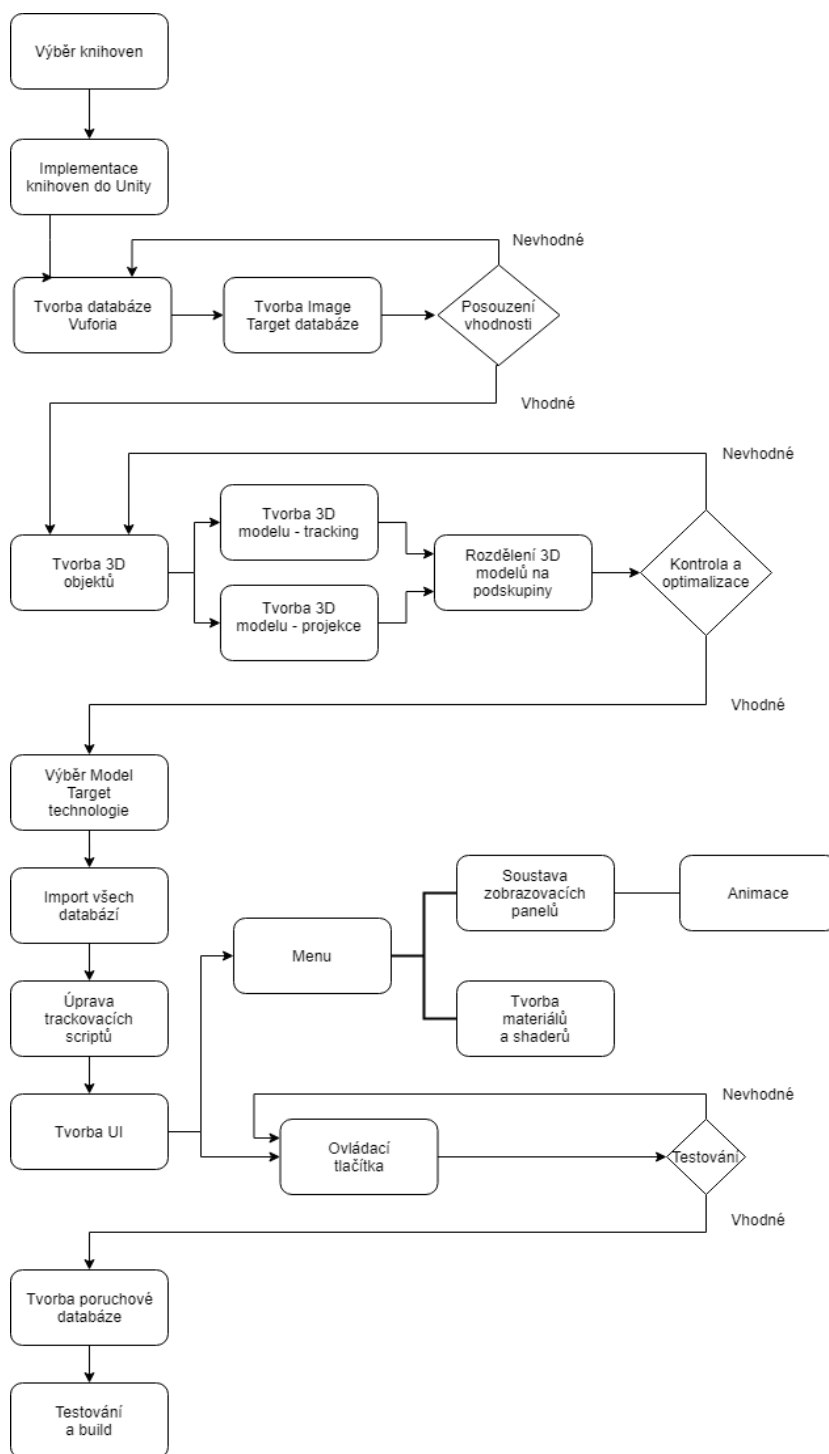
		Vuforia	DeepAR	XZIMG	EasyAR	DroidAR	IN2AR	ARmedia
Zdarma	5	ANO-	ANO-	ANO+	ANO-	ANO+	ANO-	ANO-
Unity	5	ANO	NE	ANO	ANO	NE	ANO	ANO
Marker	4	ANO+	NE	ANO	ANO	ANO	NE	ANO
Natural Feature	4	ANO	NE	ANO	ANO	ANO	ANO	ANO
Object Tracking	5	ANO	NE	ANO-	NE	ANO	NE	ANO
IMU	3	ANO	NE	NE	NE	ANO	NE	ANO
Visual Search	2	ANO	NE	NE	NE	Brzy	NE	NE
Face Tracking	1	NE	ANO	NE	NE	NE	NE	NE
GPS	1	ANO	NE	NE	NE	ANO	NE	ANO
Σ		29	0	23	0	0	0	27

7.3 Zhodnocení výsledků a výběr knihovny

Po vyloučení nevhodných variant, z důvodu nesplnění některého z kritických kritérií, zbývají v úvahu 3 možné knihovny. Jako nejvhodnější se jeví použít metodu společnosti Vuforia. Splňuje všechny stanovené podmínky. Licence použití je zdarma. Tato licence má sice jistá omezení, která budou podrobně rozebrána v následující kapitole, nejedná se však o omezení podstatného rázu. Zabraňuje spíše sériovému nasazení.

8 NÁVRH APLIKACE

Na začátku návrhu je nutné předem definovat, co vlastně má aplikace umět. Kde bude nasazena a jaké jsou její atributy. To bylo shrnuto v předcházející kapitole systémového rozboru. Pro jednodušší postup návrhu byl stanoven diagram vývoje aplikace. Diagram obsahuje všechny důležité milníky, kterých je nutno dosáhnout. Na konci každého takového milníku je posouzení vhodnosti daného řešení.



Obr. 11) Postup operací při návrhu.

8.1 Výběr knihoven a jejich implementace

V předchozí kapitole bylo jednoznačně prokázáno že bude jako nejvhodnější varianta použita knihovna Vuforia. Ta je instalována přímo při prvním spuštění Unity editoru. Mimo tuto knihovnu je nutné zahrnout do výběru i další knihovny dle našeho výběru. V této aplikaci byli mimo jiné zahrnuty knihovny Android Studio, z důvodu možnosti buildu aplikace přímo na android. Dále byla vybrána knihovna NET. Důvodem je možnost psaní skriptů a jejich úprava v prostředí Visual studia. Je velmi důležité nepodcenit prvotní nastavení. Může se stát, že by chyběly některé důležité prvky se kterými nebylo na začátku počítáno.

8.1.1 Databáze Vuforia

Na portálu Vuforia je možné si založit developerský účet. Je záhodno vybrat účet bez poplatků pro jednotlivce. Budeme tím sice ochuzeni o některé funkce, ale tyto nebudou mít žádný vliv na vývoj aplikace.

Vuforia portál je rozdělen na dvě části. První se nazývá License Manager a spravuje údaje o účtu. Dále generuje speciální licenční kód, který je nutné zadat do Unity, tak aby „věděla“ o jakou databázi se jedná a s jakým účtem je propojena.

Druhou částí je Target Manager. Zde tvoříme samotnou databázi objektů, které budeme chtít trackovat. Je možné si vybrat z následujících:

- Single Image – Jedná se o trackování jednoduchého obrázku ve 2D rovině.
- Cuboid – Trackování objektu ve tvaru krychle.
- Cylinder – Trackování objektu ve tvaru válce.
- 3D Object – Teoreticky jakýkoliv 3D model.

Pro použití v našem případě byla vybrána možnost první, a to trackování 2D obrázku. Ten byl vytvořen v aplikaci Gimp. Důraz byl kladen zejména na jasnou geometrii, ostré hrany a předěly mezi tmavými a světlými místy. Díky tomu dojde ke snadnému rozpoznání 2D struktury.



Obr. 12) 2D obrázek pro Image Tracking.

Obrázek je čtvercový a zabírá celou šířku stránky formátu A4. Rozlišení je tedy definováno jako 1080×1080 px. Pro použití v Unity je nutný přepočet na metry ze vztahu:

$$1080 * 0,026458333 = 28,575 \text{ cm} \cong 0,28575 \text{ m}$$

Dále je nutné vybrat a vložit hledaný objekt a určit jeho velikost. Vuforia pracuje s metry jako základní jednotkou. Při zadání nesprávné hodnoty je možné, že by Vuforia objekt nenašla. Důvodem je zaměření na objekt jiné velikosti. Posledním krokem je vhodně pojmenovat objekt. Toto jméno již nebude možné měnit a bude se vyskytovat ve všech instancích programu.

Add Target

Type:



File:

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Obr. 13) Vytvoření 2D Image Target databáze.

8.1.2 Posouzení vhodnosti Image Target Databáze


Následuje nahrání obrázku na portál Vuforia. Zde dojde k posouzení jeho vhodnosti. Vuforia opět většinu práce udělá za nás. Po nahrání následuje několika minutová prodleva v závislosti na počtu a kvalitě posílaných údajů.

Po uploadu jsou vyznačeny trackovací markery. Jde o vizualizaci důležitých bodů nalezených na vloženém obrázku. Je vhodné jich mít co nejvíce na pomezí tmavá/světlá barva.



Obr. 14) Vyznačení trackovacích markerů.

Následně portál automaticky zhodnotí kvalitu nahraného obrázku a udá mu určité hodnocení. Pokud je hodnocení dostatečné, nastaví status jako aktivní a umožní uživateli stáhnout databázi a importovat ji do Unity.

<input type="checkbox"/>	Target Name	Type	Rating ⓘ	Status ▾
<input type="checkbox"/>	 Door_Tracking_final	Single Image	★★★★★	Active

Obr. 15) Hodnocení Image Target databáze.

Tento obrázek Vuforia hodnotila maximálním možným hodnocením. Vše je tedy připraveno na Import do prostředí Unity.

8.2 Tvorba 3D modelů

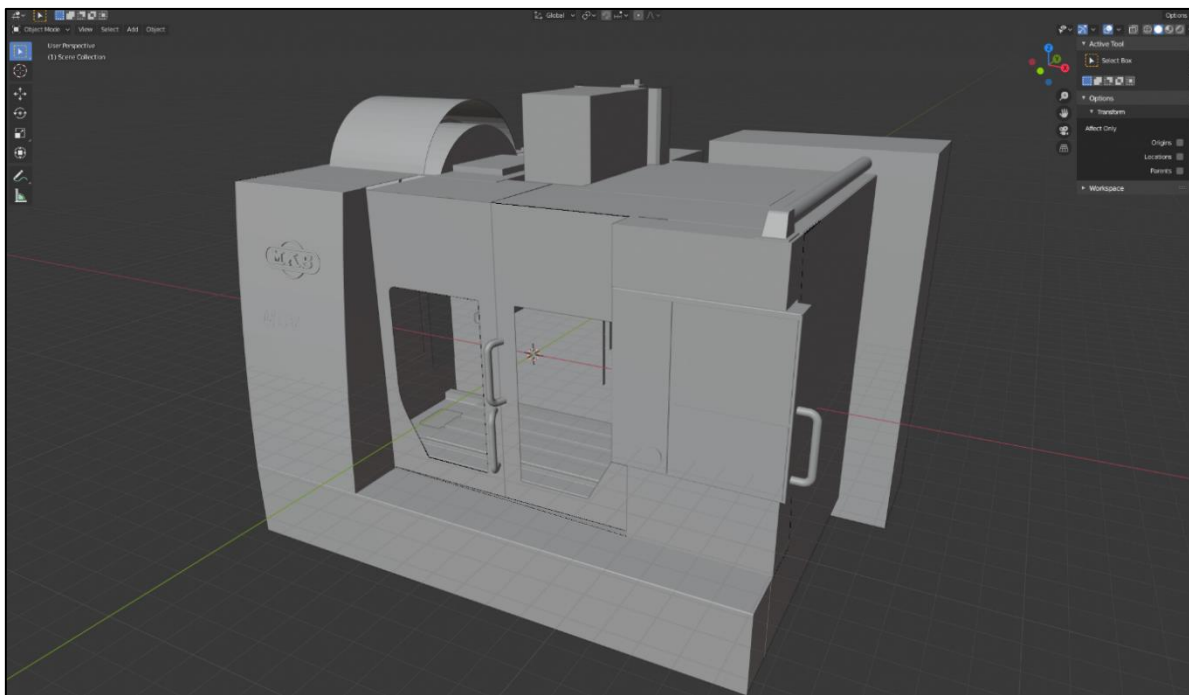
Jako další logický krok je tvorba potřebných 3D modelů. Jedná se jak o modely promítané, tak o modely snímané. V tomto případě jde o modely stroje MCV. Modely budou principiálně stejné. Je však nutné při jejich tvorbě přistupovat odlišným způsobem. Předlouhou jsou oficiální modely přímo od společnosti MCV. Jedná se tedy o ideální modely v měřítku 1:1.

8.2.1 Úprava 3D modelů

Pro úpravu modelů byl použit 3D modelář Blender. Tento se sice nepoužívá v oblasti průmyslu a strojní výroby. Je to však velmi silný nástroj, který umožňuje pokročilou práci se strukturami, má obsáhlou knihovnu a je zcela zdarma. Výhodou je také relativně vysoká stabilita (od verze 2.81) a možnost poradit si i s velmi obsáhlými modely.

Do Blenderu byl nejprve vložen původní model MCV. Následně byl upraven referenční bod tzv. Origin. Lze si jej představit jako těžiště předmětu. Tento byl posunut k patě stroje pro lepší práci s modelem. Následně byly aktualizovány jednotlivé parametry modelu. Tzn. pozice a rotace jako $[0,0,0]$. Pro zajištění rozměrové stálosti bylo upraveno i měřítko Scale na $[1,1,1]$.

Následně byly opraveny osy modelu stroje, tak aby se globální souřadný systém shodoval s referenčním. Posledním krokem byla změna polohy osy. A to tak, aby osa Y vždy mířila nahoru. Většina modelářů pracuje v systému osa Z nahoru. Unity však vyžaduje přístup jiný.



Obr. 16) Importovaný 3D model stroje MCV do prostředí Blender.

Na řadu přichází samotná geometrická úprava. Dodaný model byl veden pouze jako skořepina. Proto bylo použito několika speciálních funkcí k jeho úpravě. Jedná se základní funkce SW Blenderu.

- Solidify
- Remesh
- Decimate

Nejprve bylo použito funkce solidify. Ta vzala celou skořepinu a vytvořila z ní prostou aproximaci plný model. Následně bylo zapotřebí funkce remesh. Ta se postarala o vytvoření nové topologie struktury. Jedná se o trojúhelníkové, popř. čtvercové plochy. Konečná funkce potřebná pro vhodný model byla Decimate. Ta určitým faktorem podělí všechny prvky struktury. Díky tomu, že bylo nejprve použito funkce remesh, jsou všechny prvky relativně stejně velké. Funkce Decimate potom jednoduše zmenší počet polygonů. Faktor je definovaný uživatelem. Díky tomuto přístupu dokážeme omezit velikost exportovaného modelu. Při zbytečně velkém počtu polygonů musí AR aplikace příliš dlouho počítat a vykreslovat nepotřebné vrstvy modelu. U mobilní aplikace je toto kritickým faktorem. Tímto přístupem byl omezen počet stavebních prvků modelu. Jedná se o:

- Vertices – Body
- Edges – Hraný
- Faces – Plochy – spojnice mezi hranami a body. To co vidíme.

Scene Collection | Verts:81,007 | Faces:160,790 | Tris:160,790 | Objects:0/7 | Mem: 56.3 MiB | v2.81.16

Obr. 17) Parametry vytvořené topologie.

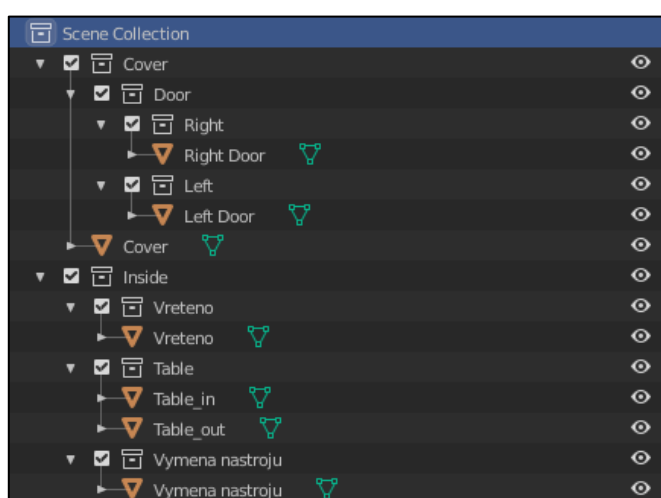
Použitím těchto metod bylo dosaženo redukce počtu ploch na cca 160 000. To je relativně obstojný výsledek. Lze akceptovat hodnoty pod hranici 400 000. Paměť potřebná pro práci s modelem je cca 55 Mb. To je opět velmi přijatelná hodnota. Hravě jí dosahuje většina moderních mobilních telefonů a tabletů se systémem android.

8.2.2 Rozdělení 3D modelů na podskupiny

Posledním krokem bylo „nařezat“ MCV model na jednotlivé podskupiny. Opět byl použit SW Blender.

Na modelu vždy stačí označit jeden bod a použít funkci Link, která automaticky označí požadovaný objekt. Možnost označení podléhá nastavení uživatele. Obecně však lze říci, že Blender označí body, které jsou určitým způsobem propojené.

Poté bylo použito funkce separate by selection. Tímto způsobem byl postupně model rozdělen na jednotlivé podskupiny. Ty byly přesunuty do jednotlivých souborů tzv. collection. Díky tomuto přístupu bylo možné vyexportovat každý jednotlivý segment stroje MCV ve formátu STL. Další výhodou je, že po importu do Unity bude každý jednotlivý segment mít své ideální místo v prostoru a nebude nutné je dodatečně přesouvat.



Obr. 18) Rozdělení původního modelu na jednotlivé segmenty.

8.3 Výběr Model Target technologie

Aplikace se sestává ze dvou technologií trackování. První z nich, Image target, již byla popsána v předcházející kapitole. Nyní je načase vybrat technologii, která se bude starat o trackování samotného stroje MCV v prostoru. Při použití databáze Vuforia se nabízí 3 možnosti provedení trackování.

- 1) Využití přímo databáze pro 3D objekty, stejně jak tomu bylo u Image target databáze.

Výhody	Nevýhody
+ Snadná aplikace. + Využití známého prostředí. + Jednoduchý import do Unity.	- Omezení velikosti modelu. - Obtížné nahrání více částí. - Absence 2D náhledu. - Obtížná zpětná úprava. - Možnost malého zásahu uživatele.

- 2) Použití Vuforia Object Scanneru. Jedná se o program pro mobilní telefony. Scanovaný objekt je umístěn na speciální podložku s markery po stranách. Pohybem telefonu kolem objektu dojde k jeho oskenování. Jedná se v podstatě o 3D scanner. Naskenovaný objekt je poté nahrán na cloud a je možné ho pohodlně použít.

Výhody	Nevýhody
<ul style="list-style-type: none"> + Vytvořená trackovací podložka. + Výhodné u reálného předmětu. + Nemusím tvořit 3D model. + Použití mobilního telefonu. 	<ul style="list-style-type: none"> - Zdlouhavá metoda. - Velmi nepřesné. - Ovlivněno materiálem předmětu. - Získám pouze mračno bodů. - Není uzavřena geometrie. - Vhodné pouze pro malé předměty. - Neintuitivní ovládání. - Náchylné na chybu uživatele.

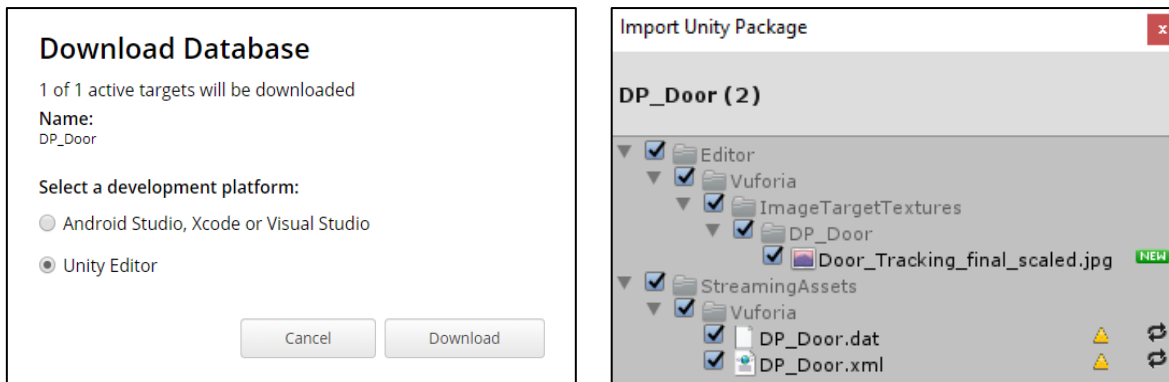
- 3) Použití Object Target Generatoru. Opět jde o aplikaci společnosti Vuforia. V tomto případě se však jedná o aplikaci desktopovou. Do této aplikace je nahrán obj. soubor trackovaného objektu. Jsou stanoveny jeho rozměry a vzdálenost ze které bude objekt trackován. Je možné přiložit i referenční 2D obrázek, pro snadnější trackování.

Výhody	Nevýhody
<ul style="list-style-type: none"> + Propojeno s databází Vuforia. + Intuitivní ovládání. + Možnost tvorby 2D náhledů. + Snadný import do Unity. + Vhodné pro jakýkoliv model. + Velmi přesné. 	<ul style="list-style-type: none"> - Nutnost vytvořit 3D model. - Nutnost přihlášení. - Lze použít pouze pro 20 modelů. - Maximální velikost 400 000 polygonů. - Obtížné stažení a instalace SW.

Jako nejvhodnější se bezpochyby jeví použití třetí metody a to Model Target Generatoru. 3D model je již vytvořen. Stačí ho tedy pouze nahrát, vytvořit Target a ten poté importovat do Unity. Ostatní metody nejsou příliš vhodné pro tuto aplikaci. Hlavním důvodem je možnost příliš malého zásahu do procesu tvorby Targetu.

8.4 Import databází do prostředí Unity

V tomto okamžiku je vše připraveno pro import. Jak již bylo zmíněno jednotlivé databáze se zabalí do speciálního souboru tzv. package. Ten se pak automaticky importuje do prostředí Unity.



a)

b)

Obr. 19) Import databáze: a) export databáze z portálu Vuforia, b) import package databáze do Unity.

8.5 Úprava trackovacích scriptů

Jak již bylo zmíněno, je nutné použít více scriptů ke trackování Image a Object Targetu. V tomto případě se jedná o dva scripty. Po importu databáze do Unity se tyto scripty automaticky vytvoří. Jedná se o poměrně pokročilé programování. Proto se dají se štěstím přejmout a upravit pouze minoritu kódu.

Účelem Image Tracking targetu je nalezení 2D textury. Při nalezení dojde k zobrazení dveří stroje. Při ztrátě jsou dveře zneviditelněny. Jde v podstatě pouze o emulování procesu zavírání a otvírání dveří.

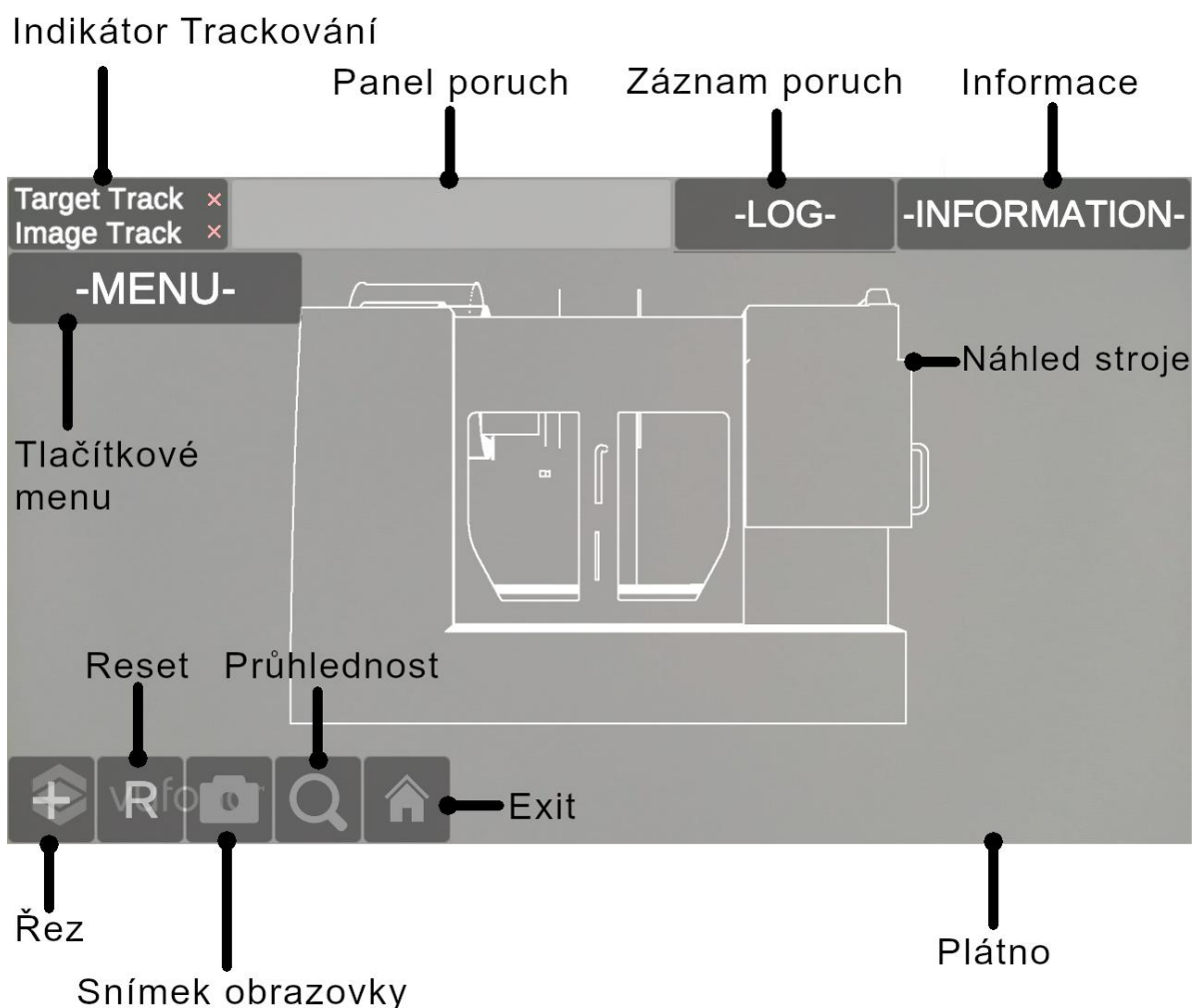
Ve výsledku je tedy řídicí script téměř identický. Přibyla pouze jedna funkcionality. Logika scriptu je následující. Při nalezení objektu vykonej funkci A. Při ztrátě objektu vykonej funkci B. Když nenastane ani jedna skutečnost vykonej Error. Po úpravě tedy scripty vypadají následovně. Byla vytvořena proměnná DoorTrack defaultně nastavená na setActive(True). Při prvotním zapnutí aplikace bude proměnná kladná a nebude vykonána žádná funkce. Dveře stroje budou stále viditelné. Při nalezení Objektu bude skutečnost stejná tzn. setActive(True). Při ztrátě trackovacího obrázku bude vykonána funkce setActive(False). Ta se postará o zneviditelnění dveří. V případě Object Targetu je situace obdobná. Místo viditelnosti dveřního prostoru však pouze obstarává viditelnost indikátoru v GUI. To je podrobněji popsáno v kapitole UI.

8.6 Tvorba UI

Tvorba vhodného prostředí je pro uživatele klíčová. UI musí být přehledné a snadno čitelné. Zároveň však musí obsahovat všechny potřebné prvky. Tento oříšek je poměrně častým problémem při tvorbě nových aplikací. Je běžnou praxí, že se autor snaží vměstnat na obrazovku co nejvíce prvků, zatímco uživatel jich chce používat jen několik.

Při návrhu aplikace jsem se setkal se stejnými problémy. Každý prvek UI má své místo a nesmí bránit ve funkci ostatním prvkům. To se často stává při překrývajících se vrstvách a layoutech. Je nutné brát v potaz pořadí v jakém můžeme na určité objekty klikat a jak budou volat jednotlivé funkce jim přiřazené. Nakonec jsem přistoupil k řešení pomocí vysouvacích panelů a poloprůhledných prvků.

Uživatel bude primárně používat prsty levé ruky, pravá ruka bude přidržovat telefon. Z toho důvodu je většina prvků, zasluhujících zásah uživatele koncipována na levé spodní části rozhraní. Jde o Menu panel a 5 tlačítek s rozličnými funkcemi.



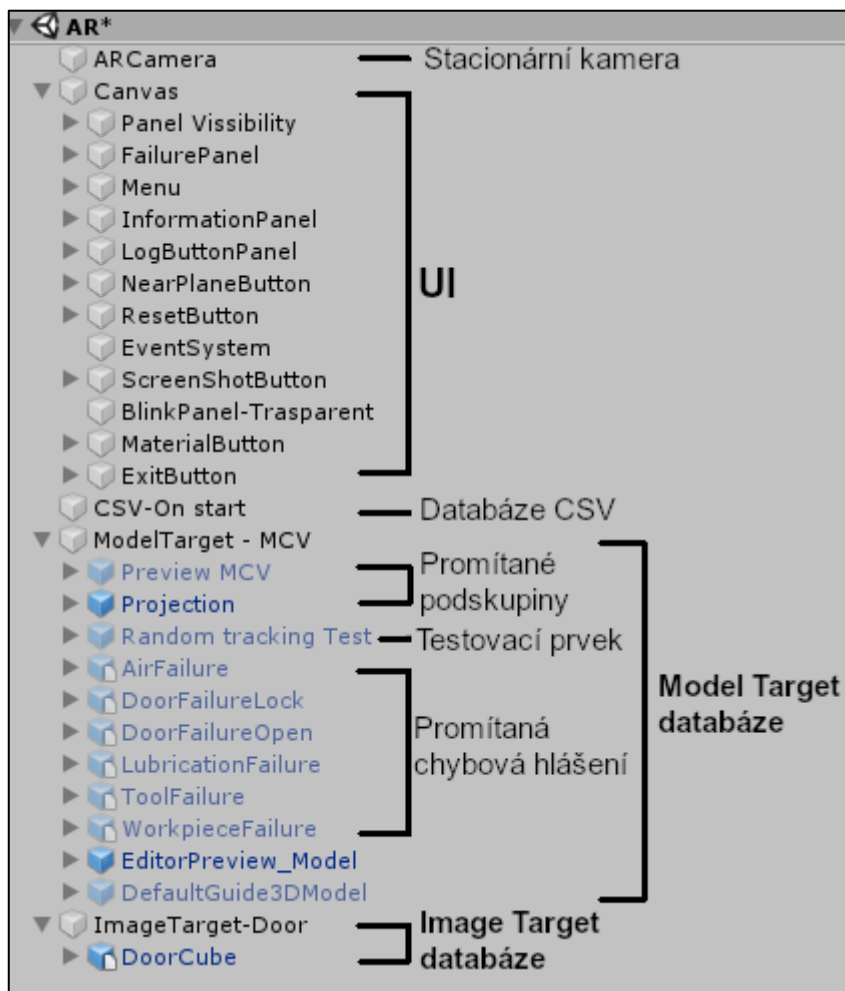
Obr. 20) Uživatelské rozhraní.

8.6.1 Canvas

UI se skládá z mnoha dílčích prvků. Z panelů, tlačítek a různých objektů. Všechny jsou ale podmnožinou prvku plátno, dále Canvas.

Canvas je definován jako plocha, která bude zobrazována po spuštění aplikace. Určuje, co se ve výsledku zobrazí na displeji. Je určena počtem pixelů. Tento počet je však možné měnit pro každé zařízení, podle jeho maximálního rozlišení. Plátno se proto vždy roztáhne na celou obrazovku. Dále byla udána restrikce na otáčení obrazovky, přičemž defaultní nastavení je v poloze horizontální nehladě na nastavení uživatele. Neočekávané otočení obrazovky vede k nesprávnému rozvržení UI. Není možné kliknout na jakékoliv tlačítko. Neblaze ovlivní také promítání součásti. Z těchto důvodů je restrikce otáčení nutná.

Canvas je osazen dalšími objekty. K těmto se chová jako parent. Ostatní prvky jsou v logice definovány jako child. Náleží tedy objektu Canvas. Tímto způsobem lze dělit prvky UI do hierarchického systému uspořádání. Pro další práci s programem je to stěžejní metoda. Je možné rychle přesunout či opravit větší množství objektů jednoduchým zásahem od objektu parent.



Obr. 21) Kompletní přehled prvků systému.

8.6.2 Soustava panelů

Pro komunikaci aplikace s uživatelem byla vytvořena soustava několika panelů. Do těchto panelů je vypsáno sdělení, výstraha nebo informace, která má být sdělena. Je sice možné ke stejnému účelu používat funkci GUI, ale toto není příliš vhodné. Je to metoda, kdy je sdělení prostě vypsáno na obrazovku bez možnosti editace. Proto jsem přistoupil k obtížnější metodě, která ovšem slibuje lepší výsledky. Hlavní výhodou je fakt, že opět definujeme gameObjekty v podobě panelů a těm přiřkládáme hierarchický status.

V této práci jsou použity rozličné panely. Některé stacionární, některé mají přiřazenou animaci zobrazení. Všechny jsou ale definovány stejným způsobem. U všech je také určen komponent Text. K tomuto komponentu je postupně přistupováno a je měněn v závislosti na přání uživatele. Popřípadě je panel nulován. To nastává v okamžiku, kdy nechceme nic zobrazovat. Množina prvků komponentu Text je v případě nulování prázdná. V práci jsou použity tyto panely:

Visibility Panel

Jde o zobrazovací panel v levém horním rohu obrazovky. Plní funkci indikátoru zobrazování. Jedná se spíše o pomocný panel. Obsahuje dvě hodnoty. První je Target track. Druhý je označen jako Image track. Při spuštění aplikace jsou obě hodnoty nastaveny na nulu. Při nalezení Target Objektu je hodnota nastavena na jedna a je zobrazen zelený indikátor. Znamená to, že je objekt nalezen a neustále trackování. Totéž platí i při nalezení Image Targetu.

Failure Panel

Panel poruch. V tomto panelu se zobrazují poruchy, které si uživatel přeje zobrazit. Název poruchy i její číselné označení je načteno z databáze poruch.

Menu Panel

Animovaný panel je skrytý, až do okamžiku kdy je otevřen poklepnem na tlačítko menu. Obsahuje hlavní část programu. Tím se rozumí logika zobrazování a několik tlačítek, jež načtou databázi a umožňují zobrazit požadovanou poruchu.

Information Panel

Jedná se o stěžejní panel. Opět je vybaven animací. V panelu je uvedena možná příčina načtené poruchy a její možné řešení. Text příčiny i řešení je načten z poruchové databáze.

Log Panel

Další z výsuvných panelů. V panelu jsou uvedeny poruchy, které byly v minulosti zobrazeny. Mimo poruchy se do Log panelu promítne i den a čas, kdy byla porucha volána. Při zapnutí aplikace se zobrazí poruchy pouze z minulých zapnutí. Log file se neaktualizuje v reálném čase. Aktualizace se provede pouze po novém spuštění aplikace.

Blink Panel

Jde o nejméně důležitý panel, který je po celou dobu provozu neviditelný a zabírá rozměry obrazovky. Je zde uveden pouze jako reference k funkci snímku obrazovky.

8.6.3 Menu

Stěžejním prvkem UI je výběrové Menu. V tomto oddíle uživatel vybere závadu, kterou obdržel na stroji a kterou chce zobrazit v aplikaci. Menu je tvořeno výsuvným panelem a tlačítkem, které udává povel k jeho otevření. Jedná se tak o dva separované objekty. V panelu jsou umístěna i další tlačítka kontrolující logiku zobrazování poruch.

Animator

Po kliknutí na tlačítko Menu je zavolána funkce k otevření panelu PanelOpener. Ta zkontroluje bool konstantu open. Při zapnutí systému je hodnota 0. Tato funkce ji vždy při zavolání převrátí na opačnou.

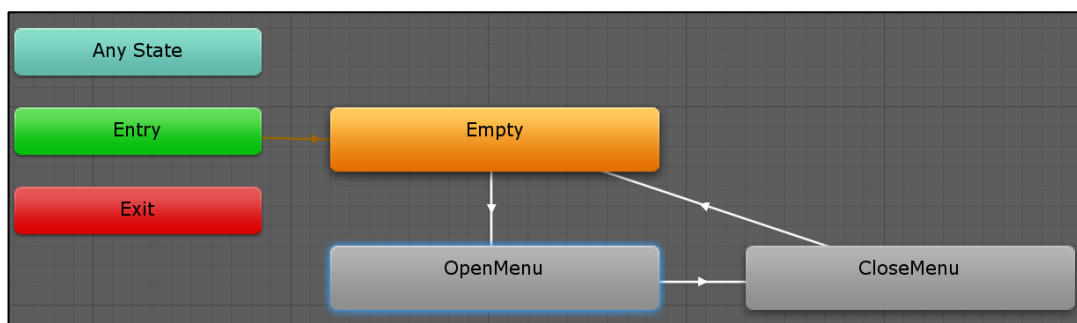
```
public class PanelOpener : MonoBehaviour
{
    public GameObject Panel;

    public void OpenPanel()
    {
        Animator animator = Panel.GetComponent<Animator>();
        if (animator != null)
        {
            bool isOpen = animator.GetBool("open");
            animator.SetBool("open", !isOpen);
        }
    }
}
```

Obr. 22) Funkce sloužící k otevření panelu.

Přepsání konstanty open ovlivní logiku prvku Animator. V něm jsou nastaveny stavy, ve kterých se objekt může nacházet. V tom případě je to:

- Empty – Defaultní stav, při kterém je, kterém je bool open == false. Panel je tedy zavřený.
- OpenMenu – Po zavolání funkce OpenPanel je změněna hodnota open a panel se otevře.
- CloseMenu – Poslední stav, který zavírá menu a přechází opět do Empty stavu.

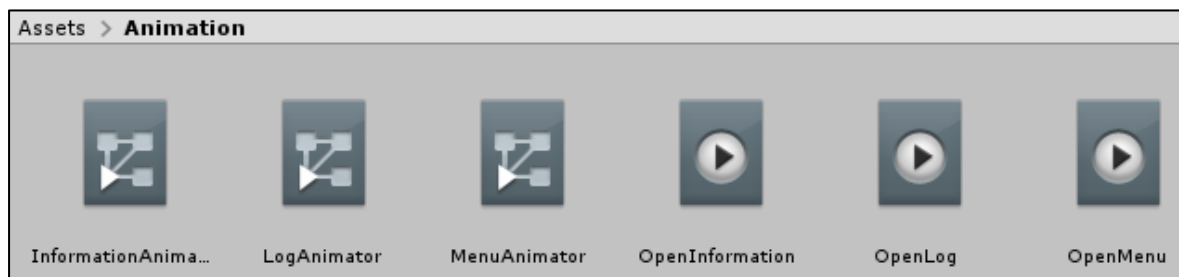


Obr. 23) Logika stavů otevírání panelu.

Pro správné fungování metody otvírání a zavírání panelů je nutné jednotlivé prvky uložit do stejného adresáře. Jedná se o výše zmíněnou logiku stavů a o samotnou animaci otevření a

zavření. Ta se musí vytvořit a uložit separovaně. V tomto případě jde však o prostou translaci panelu. Animaci tedy neuvádím. Stejného principu bylo využito i při zbylých panelech.

Jde o výsuvný panel Informací a Log panel. Ve výsledku je tedy nutné vytvořit 6 souborů (3 stavy a 3 animace).



Obr. 24) Uložené soubory animátoru v jednom adresáři.

Zobrazování poruch

Hlavní funkcí Menu je ale dát možnost uživateli zobrazit různé poruchy na stroji. K tomu je určena soustava tlačítek v panelu. Klepnutím na tlačítko menu, se panel rozvine do plné velikosti. Panel je parent dalších prvků. V tomto případě je to 6 separovaných tlačítek s názvy poruch, které se mají zobrazit. Díky faktu že jsou vedeny v logice jako child, je možné je přesouvat zároveň s výsuvným panelem. Jedinou podmínkou je, že musí být zavazben ve směru pohybu panelu. Je to poměrně logické. Když se panel vysouvá zleva doprava v ose X. Musí mít i tlačítka translaci pouze v ose X. Ostatní osy jsou v tomto případě irelevantní a nic se s objektem neděje v těchto rozměrech.

Samotná tlačítka jsou nastavena jako transparent. To znamená že jsou průhledná, ale jejich funkce je stále stejná. Průhlednost všech prvků je zvolena s ohledem na lepší přehlednost aplikace. Tlačítko tak vizuálně nezabírá tolik místa v UI. Jediná neprůhledná část je text tlačítka, tak aby uživatel mohl snadno přečíst co tlačítko vlastně dělá.

Ovládací scripty tlačítek, jsou připnuty k jejich parent objektu. V tomto případě je to Menu tlačítko. Tlačítka pak při jejich stisknutí volají specifickou funkci z vybrané třídy scriptu.

Z důvodu omezení tzv. špagetového kódu byla funkce zobrazování poruchy rozdělena na dvě části.

- 1) Script obsahující funkce pro manipulaci s gameObjekty, jako je zobrazování 3D modelů a zobrazování indikátorů ve 3D prostoru.
- 2) Script určený pro práci s textem a databází. Jednotlivé funkce jsou určeny k načtení textového pole z databáze a zobrazení tohoto textu v potřebných panelech.

8.6.4 Zobrazení 3D indikátorů

Po stisknutí určitého tlačítka je zobrazen doplňující 3D model. Jedná se o červený kruhový model indikátoru a popis dané poruchy. Logika zobrazování je definována ve funkci MenuButtons. Ve scriptu je nutné definovat následující:

Nejprve je nutné deklarovat proměnné a objekty. Jedná se o objekty Air, DoorLock etc. Jde o 3D objekty, které se budou zobrazovat v okolí stroje MCV. V programu je možné zavést pouze parent objekt. Další objekty, jako je text a pozadí textu se přesune společně s indikátorem, který je zde veden jako parent.

Dále jsou uvedeny objekty typu Text. Jde o objekty, které definují logiku ukládání Log informací do databáze. V případě ukládání musíme použít proměnnou Text Asset. Ta je schopná komunikace s textovými dokumenty. Poslední proměnnou je integer, které plní pouze funkci čítače pro všechna tlačítka.

Nakonec je nutné definovat jednotlivé funkce. Nejprve je vykonána funkce Start(). Následně je možné volat funkce odpovídající jednotlivým tlačítkům. Po stisku tlačítka se vykoná pouze jedna z 6 možných zobrazení. Poté se systém vrátí k počátečnímu stavu [35].

```
public class MenuButtons : MonoBehaviour
{
    public GameObject Air;
    public GameObject DoorLock;
    public GameObject DoorOpen;
    public GameObject Lubrication;
    public GameObject Tool;
    public GameObject Workpiece;

    public Text textPanel;
    private Text textLog;

    private string content;
    private string path;

    public TextAsset LogRead;

    public int counter = 0;

    public void Start()...
    private void LogDisplay()...
    private void Clear()...
    private void Log()...
    public void AirButton()...
    public void DoorLockButton()...
    public void DoorOpenButton()...
    public void LubricationButton()...
    public void ToolButton()...
    public void WorkpieceButton()...
```

Obr. 25) Logika pro zobrazení 3D objektů.

Při startu aplikace dojde ve funkci Start() automaticky k deklarování cesty objektu ve stávající scéně. Dojde k přiřazení textu panelu k vytvořené proměnné textLog. Proměnná je typu Text, proto se musíme odkazovat ke komponentu text. Následně dojde k zobrazení Log databáze v panelu Log.

```

public void Start()
{
    textLog = GameObject.Find("LogButton/Panel_movin/LogText").GetComponent<Text>();
    LogDisplay();
}
    
```

Obr. 26) Funkce Start().

Jako příklad zobrazování poruchy, byla vybrána funkce AirButton(). Ta při svém zavolání provede zviditelnění indikátoru pro nedostatečný tlak vzduchu. Zároveň také zapíše do Log databáze, že byla tato porucha zobrazena. Při dalším stisku tlačítka, dojde k provedení funkce Clear() a všechny dodatečně zobrazované objekty jsou opět zneviditelněny.

<pre> public void AirButton() { counter++; if (counter % 2 == 1) { Clear(); Air.SetActive(true); Log(); } else { Clear(); Air.SetActive(false); } } </pre>	<pre> private void Clear() { Air.SetActive(false); DoorLock.SetActive(false); DoorOpen.SetActive(false); Lubrication.SetActive(false); Tool.SetActive(false); Workpiece.SetActive(false); } </pre>
--	--

a)

b)

Obr. 27) Funkce AirButton(): a) volání funkce stiskem tlačítka, b) funkce zneviditelnění promítaných objektů.

Po provedení funkce AirButton(), je nutné tuto akci uložit. K tomu je uzpůsobena funkce Log(), volaná vždy na konci sekvence. Nejprve do proměnné typu string s názvem path uloží instanci .txt souboru, do kterého budeme ukládat data. Soubor je nalezen podle jména Log.txt.

Následně do proměnné content uloží aktuální položku poruchy zobrazovanou v panelu poruch. Zároveň uvede čas zobrazení a oddělí další položku novým řádkem. To vše je uloženo to souboru Log.txt.

```
private void Log()
{
    path = Application.dataPath + "/Log.txt";
    content = textPanel.text + System.DateTime.Now + "\n";
    File.AppendAllText(path, content);
}
```

Obr. 28) Ukládání zobrazovaných poruch do souboru Log.

Pro snadnější zobrazování a čtení textu byla stanovena speciální funkce LookAtCamera(). Ta zajišťuje otáčení všech indikátorů směrem ke kameře. Je otáčen panel s textem zobrazující místo a název poruchy. Otáčení probíhá kolem vlastní osy panelu. Uživatel tak může s přístrojem pohodlně obcházet stroj a indikátory budou vždy přístupné a jasně viditelné.

```
public class LookAtCamera : MonoBehaviour
{
    public Camera cameraToLookAt;

    void Start()
    {
        //transform.Rotate( 180,0,0 );
    }

    void Update()
    {
        Vector3 v = cameraToLookAt.transform.position - transform.position;
        v.x = v.z = 0.0f;
        transform.LookAt(cameraToLookAt.transform.position - v);
        transform.Rotate(90, 0,0);
    }
}
```

Obr. 29) Funkce otáčení panelu směrem ke kameře.

8.6.5 Zobrazení textu

Po zobrazení 3D modelů je nutné dát uživateli informace v podobě textu. Ten je zobrazen v panelech informací a v panelu poruch. Zobrazovaná data jsou načtena z databáze, která má podobu textových souborů.

Nejprve je nutné definovat proměnné. Je zjevné že bude nutné více proměnných typu text, než tomu bylo v minulém případě. Proměnné textInf etc. opět přistupují ke komponentům Text daných panelů. TextAsset je použit z důvodu načítání textu z databáze. Jako poslední je definován List, který je tvořen proměnnými typu string. Ten plní úlohu řadiče. Načtená data z textového souboru rozdělí podle řádků a vytvoří z nich list.

Dále byly vytvořeny funkce ovládající zobrazení jednotlivých řádků Listu do odpovídajících panelů. Každá porucha má svou vlastní funkci. Jsou zde uvedeny i dodatečné funkce jako Clear() a Write(). Ty se starají o samotné zapisování údajů do panelu.

<pre> private List<string> TextAssetToList; private Text textInf1; private Text textInf2; private Text textPanel; private TextAsset LoadAsset; </pre>	<pre> private int counter = 0; public void Start()... public void AirFailure()... public void DoorFailureLock()... public void DoorFailureOpen()... public void LubricationFailure()... public void ToolFailure()... public void WorkpieceFailure()... private void Write()... private void Clear()... </pre>
a)	b)

Obr.30) Logika zobrazování textu: a) deklarované proměnné, b) jednotlivé funkce.

Opět bylo vytvořeno šest řídicích funkcí. Příkladem budiž funkce indikující nedostatečný tlak vzduchu. Při jejím zavolání je nejprve vykonán příkaz Clear(). Ten vymaže text ze všech panelů. Následuje načtení .txt souboru. Ten je uložen v adresáři Resources v podsložce Failure Text. Správný soubor je nalezen díky hledání názvu souboru AirFailure. Tento je uložen jako TextAsset. Po načtení souboru s poruchami je nutné je zobrazit. K tomu slouží funkce Write(). Proto je volána až jako poslední prvek. Po opětovném stisknutí tlačítka je znovu volána funkce Clear().

```

public void AirFailure()
{
    counter++;
    if (counter % 2 == 1)
    {
        Clear();
        LoadAsset = Resources.Load("FailureText/AirFailure") as TextAsset;
        Write();
    }
    else
    {
        Clear();
    }
}
        
```

Obr. 31) Příklad funkce načítající poruchy z textového dokumentu.

Příkaz Write() slouží ke zpracování načtené databáze z textového souboru. Textový soubor je načítán znovu pro každou jednotlivou volanou poruchu. Po načtení je uložen do Listu a rozdělen podle řádků. Každý řádek mřížky má své specifické označení. Díky tomu je možné volat jednotlivá pole mřížky. Každému panelu je přiřazen odpovídající řádek Listu a ten je i zobrazen [35].

```
private void Write()
{
    var listToReturn = new List<string>();
    var arrayString = LoadAsset.text.Split("\n"[0]);
    foreach (var line in arrayString)
    {
        listToReturn.Add(line);
    }
    textPanel.text = listToReturn[0];
    textInf1.text = listToReturn[1];
    textInf2.text = listToReturn[2];
}
```

Obr. 32) Vytvoření Listu a zobrazení textu.

Funkce Clear(), slouží k vymazání textu v panelech. Je velmi jednoduchá a mohla by být začleněna do jiné funkce. Z hlediska přehlednosti kódu je však daleko výhodnější funkce vždy účelně oddělovat. Clear() v podstatě přistoupí ke komponentům text jednotlivých GameObjectů a nastaví jejich hodnotu na prázdnou množinu.

```
private void Clear()
{
    textPanel.text = null;
    textInf1.text = null;
    textInf2.text = null;
}
```

Obr. 33) Funkce vymazání textu.

8.6.6 Dodatečná tlačítka UI

Pro dosažení lepších výsledků a větší přehlednosti byla vytvořena pětice dodatečných tlačítek. Ty mají rozličné funkce. Jsou však cíleny na uživatele a na jeho možnosti ovlivnit zobrazované prvky.

Jedním z těchto tlačítek je možnost nastavení kamery na tzv. řezu. Při přiblížení kamery k modelu začne v určitém bodě clipovat. Nebude tedy již vykreslovat dané modely. Vzdálenost clipování lze ovlivnit. Právě na to cílí funkce `CameraNearPlane()`. Nejprve načte objekt kamery a přiřadí mu AR kameru z aktuální scény. Při startu aplikace jsou všechny změny smazány voláním funkce `Reset()`. Následně při stisku tlačítka je aktivována funkce `NearPlanePlus()`. Ta vyhledá komponent AR kamery `NearClipPlane` a upraví jeho hodnotu tak, že k ní přičte 0,25 m. Jelikož se jedná o float hodnotu je nutné ji tak také zapisovat. Příkaz `Reset()` je možné volat separovaně. K tomuto účelu je určeno vedlejší tlačítko `Reset`.

```

public class CameraNearPlane : MonoBehaviour
{
    public GameObject ObjektCamera;

    public void Start()
    {
        Reset();
    }

    public void NearPlanePlus()
    {
        Camera MyCamera = ObjektCamera.GetComponent<Camera>();
        MyCamera.nearClipPlane = MyCamera.nearClipPlane + 0.25f;
    }

    public void Reset()
    {
        ObjektCamera.GetComponent<Camera>().nearClipPlane = 0.05f;
    }
}
    
```

Obr. 34) Přepnutí kamery do funkce řez.

Dalším z tlačítek je tlačítko transparentnosti. Funkce je opět relativně jednoduchá. Nejprve načte komponent `Renderer` ze všech potřebných objektů. Po jeho načtení je komponentu `Renderer` střídavě přiřazován určitý materiál. Proto je nutné přímo v Unity vytvořit dva identické materiály. Jejich barva je vždy stejná, liší se ovšem ve transparentnosti. Transparentnost je v Unity označena jako `alpha` a lze k ní přistoupit přímo z Unity.

Dále jsou definovány poslední dvě tlačítka. Obě jsou ukotvena ke spodní části obrazovky. První z nich zastává úlohu screenshotu. Po stisknutí tlačítka je automaticky uložen obrázek obrazovky. Využívá k tomu funkci `TakeScreen()`. Velikost zachycené fotografie bude stejná jako pomocný transparentní panel. Fotografie je nakonec uložena do adresáře s aplikací, pod hlavičkou názvu aplikace.

Posledním tlačítkem je tlačítko vypnutí. Příkaz ukončení aplikace je nutné provést manuálně, proto je zde speciální tlačítko k tomu uzpůsobené. Funkce je koncipovaná na platformu android. Proto v pravidelných intervalech „hlídá“ zda došlo ke stisknutí klávesy ukončení. Bez této funkce by nebylo možné aplikaci řádně ukončit [36] [37].

```
public class ExitGame : MonoBehaviour
{
    public void ExitNow()
    {
        Application.Quit();
    }
    void Update()
    {
        if (Application.platform == RuntimePlatform.Android)
        {
            if (Input.GetKeyDown(KeyCode.Escape))
            {
                Application.Quit();
            }
        }
    }
}
```

Obr. 35) Funkce ukončení aplikace.

8.7 Tvorba databáze poruch

Aby bylo vůbec možné zobrazovat text v panelech nebo v podobě 3D objektů, je nutné nejprve vytvořit databázi možných poruch a jejich řešení a příčiny. Toho je principiálně možno dosáhnout hned několika způsoby. Některé jsou vhodné, některých je však třeba se vyvarovat. Principálně jsem zpracoval 4 možné druhy řešení databáze.

8.7.1 Všechny informace obsaženy ve scriptech

Jednou z možností, jak uchovat data poruch je jejich „tvrdé“ uložení do scriptu C#. Pro všechna tlačítka by vznikla jedna velká třída, která by obsahovala všechny informace.

Princip funkce by byl zhruba obdobný jako při stávajícím řešení. Nenačítal by však žádné externí soubory. Jen by zobrazoval text, který byl zapsán do scriptu. Pouze by měnil textový komponent na předepsaný text. Nevýhodou tohoto řešení je nutnost zavedení speciální funkce pro každý jednotlivý stav a panel. Z toho důvodu je dané řešení nevhodné.

8.7.2 Použití vlastní databáze

Elegantnějším řešením databáze je její vlastní vytvoření. Toho je možné dosáhnout rozličnými způsoby. Může se jednat o jeden veliký balíček dat v jednom souboru, či několika malých. Lze využít vlastní SQL databázi. V některých případech se dá použít Excel soubor nebo jemu podobný formát.

Unity sama o sobě není schopná načíst excel soubor. Je proto nutné použít jiný způsob jak z excelu dostat data do Unity. Se štěstím lze použít Asset store. Ten nabízí velké množství funkcionalit, které je možné použít. Převod dat z Excelu do Unity je tak možný za použití CSV2Table. Jedná se o funkci, která převede formát CSV na C#, se kterým již umí Unity pracovat.

Původně bylo zamyšleno, aby tento způsob převládá a byl využíván v maximální míře v celé diplomové práci. Má však řadu nevýhod, které byly zjištěny až v procesu návrhu a vývoje. Mezi ně se řadí následující:

- Nutnost stahovat cizí funkce z Asset Storu.
- Převodění je možné pouze jednorázově. Nefunguje v reálném čase a je vždy nutný zásah programátora, při modifikaci databáze.
- Je nutné ponechat reziduální soubory ve scéně. Při vymazání nějaké části je možné že převod přestane fungovat.
- Neintuitivní ovládání a možnost zadávání informací.
- Některé části musí být spuštěny při startu aplikace, což může brzdit její plynulý rozběh.

Z těchto důvodů bylo od použití této metody opuštěno. Metoda je příliš komplikovaná a nabízí pouze málo výhod.

Daleko elegantnějším řešením vlastní databáze je přeskočit krok zapisování do Excelu a začít už na začátku návrhu tvořit databázi v prostředí textových souborů. Principiálně jde o stejný postup, jako při zápisu dat do Excelu. Není však nutné zavádět žádný vnější script z Asset Storu. Vše je možné naprogramovat v závislosti na přání uživatele.

Jednoduše se tedy vytvoří textový dokument pro každou jednotlivou možnou poruchu. Tyto se poté uloží do společného adresáře. Tvorba dokumentů je triviální. Celý text je pouze zapsán do textového dokumentu. Pokud chceme aby se text zobrazil v dalším panelu, oddělíme text řádkem (Enter) a spokojeně můžeme tvořit další textový zápis.

8.7.3 SQL databáze

Jednou z dalších možností vedení databáze je použití SQL. Lze využít profesionální databáze vedené externími firmami nebo je možné postupovat s vlastní SQL databází. Propojení s Unity je poměrně jednoduché a řada mobilních aplikací ji využívá. V samotné databázi se poměrně jednoduše vyhledávají potřebné informace. Hodí se však spíše pro použití v oblasti velkého množství prvků. Ideální oblast použití je například při přihlašování. V databázi je pak vedeno jméno a heslo uživatele. Další výhodou je možnost šifrování uloženého hesla, tak aby nemohlo být zneužito třetí stranou.

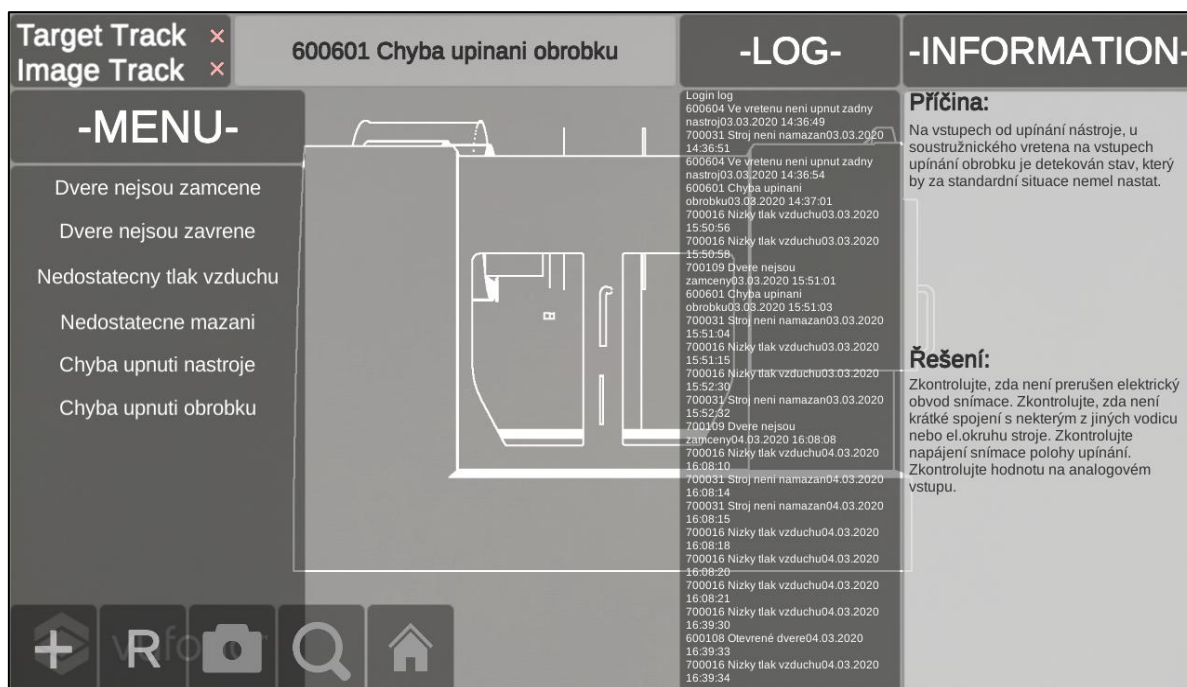
8.7.4 Použití externího modulu

Pro profesionální zavedení do výroby by byly výše zmíněné metody příliš kostrbaté a vyžadovaly by příliš častý zásah vývojářů. Za nejnávýspěšnější metodu tvorby databáze je možné považovat databázi na externím cloudu.

Při potřebě provést změnu v databázi se může programátor jednoduše připojit ke cloudu, provést změnu a uložit ji. Při dalším spuštění se pak změna automaticky provede.

Tento způsob je však náročnější na provedení. Do Unity se musí implementovat další knihovny. Celkový systém by tak byl méně přehledný a daleko více komplexní.

Dalším aspektem by mohlo být například ukládání Logu do databáze. Vedoucí výroby by si mohl snadno zobrazit nejčastější poruchy na pracovišti. Mimo jiné by bylo možné implementovat systém přihlašování uživatelů. Díky tomu by bylo patrné, jaký pracovník vyvolal jaké poruchy. Jako nejvhodnější zástupce uvedu databázi Firebase. Zpracování online databáze ale nebylo cílem této diplomové práce. Z tohoto důvodu tato nebude použita.



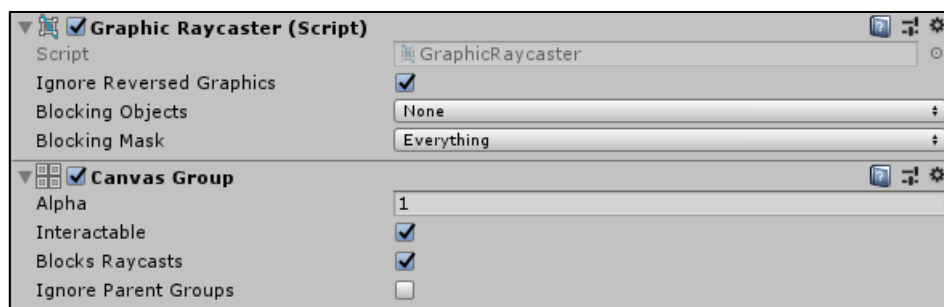
Obr. 36) Hotový návrh aplikace se všemi funkcemi.

8.8 Testování a debugging

Posledním krokem při návrhu aplikace je její testování a debugování. Jedná se o proces odhalování chyb, které mohli při vývoji vzniknout. Nejčastěji jde o chyby syntaxe. Pokud je chyba závažná a brání spuštění aplikace, Unity chybu obvykle odhalí a zabrání spuštění. Do konzole je také napsána lokace chyby a řádek kódu, kde byla poprvé nalezena.

Chyby které nejsou prioritní však ponechá netknuté. Takové chyby je pak nutné manuálně odhalit a opravit. Nemusí se jednat pouze o chyby syntaxe. Některá chybová hlášení nemají s kódem co dočinění. Jde spíše o interní nastavení Unity. Může se taky jednat o chybně nastavené UI. Při návrhu aplikace jsem se potýkal s následujícími chybami:

- Při pilotním spuštění bylo na první pohled jasně vidět, že rozložení UI je nevhodné. Bylo chybně nastaveno otočení obrazovky. Prvky UI se nezobrazovaly, protože jednoduše neměli místo kam se zobrazit. Tato chyba bylo opravena vhodným nastavením objektu Canvas. Dalším problémem byla neviditelnost některých prvků UI. To bylo způsobeno absencí některých komponentů objektu Canvas. V procesu vývoje byly zřejmě nevědomky smazány určité objekty. Jedná se o komponenty Raycaster() a Canvas Group. Tyto vyjadřují, jak bude prvek zobrazován. Po jejich opětovném vložení byl problém zažehnán.



Obr. 37) Chybějící komponenty objektu Canvas.

- Další chybou, kterou bylo nutné opravit je chyba nevhodného zobrazování panelů. V několika případech byly všechny panely růžové a neumožňovali přístup. To bylo vyřešeno resetováním shaderů v Unity. Evidentně byl někde zaveden shader, který nebyl definován.
- Jako další chyba lze považovat nefunkční event systém. Jedná se o komponentu Unity, která udává jaké operace a v jakém pořadí se budou vykonávat. Problém vyřešilo jednoduché smazání a opětovné vložení event systému.
- V prostředí Unity již je vše v pořádku. Nastal však problém v prostředí android. Původně byla k separaci poruchové databáze použita funkce z knihovny Unity. Jednalo se o funkci sReader(). Tato je implementována v Unity, není ji však možné používat v mobilním přístroji. Místo sReaderu() bylo nutné metodu čtení a zápisu databáze zpracovat odlišně. A to s ohledem na zavedení vlastní invence do této metody. Místo sReaderu() byl použit princip rozdělení textu do tabulky Listu. Metodu bylo nutné změnit u funkcí pracující s textem, včetně funkce Log(), která jen text ukládá.
- Poslední a také fatální chybou byl kolaps celé scény. Důvod jsem nebyl schopen identifikovat. Jediným řešením tohoto problému je vytvoření duplicitní scény a smazání scény originální. Tímto se sice vytvořili některé dílčí problémy, ale bylo možné scénu spustit.

Tímto analytickým přístupem byly postupně eliminovány všechna chybová hlášení a bylo dosaženo maximální stability aplikace.

Testování aplikace proběhlo na reálném stroji MCV a prokázalo, že aplikace je dostatečně stabilní a nevykazuje žádné výrazné chyby. Promítaný model byl správně ustaven v prostoru. Načítání modelu proběhlo v řádech milisekund. 3D model se rovněž udržel na správném místě při pohledu z rozličných stran. Trackování bylo přerušeno až při přiblížení na několik centimetrů ke stroji, kdy nebylo možné trackovat celý stroj, ale pouze jeho část s nedostatečnou plochou geometrie. Dá se tedy prohlásit, že testování aplikace bylo úspěšné a splnilo očekávání.



Obr. 38) Testování konečného produktu.

9 ZHODNOCENÍ A DISKUZE VÝSLEDKŮ

V diplomové práci byl zpracován návrh a vývoj aplikace rozšířené reality pro údržbu výrobních strojů. Bylo zjištěno že jednotlivec, bez větších znalostí programování, je schopen vytvořit relativně pokročilou aplikaci rozšířené reality.

Aplikace je dostatečně stabilní a nevykazuje žádné kritické chyby. Její funkce jsou omezeny pouze na 6 základních poruch stroje. Databázi poruch lze však snadno rozšířit na několik desítek. Samotné promítání a trackování objektů je dostatečně rychlé a přesné. Test aplikace byl proveden na zařízení Nokia 5. Výkon průměrného mobilního telefonu se ukázal být dostatečným.

Vývojové prostředí Unity je velmi vhodným nástrojem pro tvorbu podobných aplikací. Unity je velmi stabilní a uživatelsky přívětivé prostředí. Nespornou výhodou jsou také pravidelné aktualizace. Klíčovou vlastností Unity je však velké množství vláken na fórech, jakožto i nepřehledné množství tutoriálů a návodů přímo na oficiálních stránkách. Toto velmi usnadňuje práci na rozličných projektech. Použitá knihovna Vuforia zdárně plní svůj účel. Je dostatečně dynamická a nevykazuje žádné závažné chyby. Její použití doporučuji pro všechny podobné projekty.

V následujících letech očekávám rapidní zvýšení počtu aplikací rozšířené reality ve výrobních závodech. Podobné aplikace snižují riziko chyby a úrazu na pracovišti. Dále také zvyšují efektivitu celého podniku. Z tohoto důvodu doporučuji jako další téma závěrečné práce rozšíření stávající aplikace. Jako ideální téma se nabízí použití aplikace pro větší spektrum obráběcích strojů nebo implementování výše zmíněné databáze Firebase.

10 ZÁVĚR

Tématem diplomové práce byl návrh aplikace rozšířené reality v údržbě obráběcích strojů. Návrh aplikace byl proveden ve vývojářském prostředí Unity. Aplikace je uzpůsobena k usnadnění obsluhy a údržby vertikálního obráběcího centra firmy Kovošvit MCV 754 QUICK.

Práce se sestává ze dvou částí. V první, rešeršní části, je zpracována teoretická část práce. Důraz je kladen zejména na historické pojetí rozšířené reality. Jsou uvedeny její počátky jakožto i její postupný vývoj. Součástí rešeršní části je i nastínění možného vývoje. V práci je dále zpracováno portfolio moderních výrobců aplikací s rozšířenou realitou a oblastí jejich použití.

Hlavní část práce popisuje návrh a vývoj vlastní aplikace. Ten je popsán v jednotlivých, na sebe navazujících, krocích. Pro usnadnění návrhu byl vypracován blokový diagram se všemi důležitými prvky. Nejprve bylo zpracováno porovnání možných vývojářských prostředí. Srovnávány byly prostředí Unity a UNREAL ENGINE. Byly porovnány jejich klady a zápory. Po důkladném zvážení jejich předností bylo přistoupeno k použití prostředí Unity. Obdobným přístupem bylo zpracováno i porovnání možných trackovacích knihoven. Jako nejvhodnější knihovna bylo použita knihovna Vuforia.

Navržená aplikace je určena pro mobilní telefony a tablety s operačním systémem Android. Nutnou podmínkou je fotoaparát telefonu, ke kterému aplikace přistupuje. Princip funkce je vyhledání stroje MCV za pomoci fotoaparátu. Stroj je v mobilním telefonu překryt digitálním 3D modelem. Při poruše obsluha použije svůj telefon k naskenování stroje MCV a může si přímo na svém telefonu zobrazit řešení dané poruchy. Součástí aplikace je i 3D indikátor s popisem, který se zobrazí v místě závady. To usnadňuje obsluhu orientaci v problematické situaci.

Nedílnou součástí návrhu aplikace bylo taktéž vytvoření databáze nejčastějších poruch stroje MCV. Bylo uvedeno několik možných přístupů k vytvoření databáze a následně byl vybrán přístup nejvhodnější.

Posledním krokem bylo vytvoření přehledného uživatelského rozhraní. To obsahuje všechny potřebné prvky pro danou aplikaci. Součástí bylo i testování aplikace na reálném stroji. To ukázalo vhodnost této technologie pro použití v oblasti obráběcích strojů.

Seznam použitých zdrojů

- [1] KAIWU, Hsin, Silvia WEN-YU LEE, Hsin YI-CHANG a Jyh CHONG-LIANG. *Current status, opportunities and challenges of augmented reality in education* [online]. Taiwan, 2012 [cit. 2020-02-13]. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0360131512002527?via%3Dihub>. _ . Institute of Science Education.
- [2] The Difference Between Virtual Reality, Augmented Reality And Mixed Reality: Mixed Reality. In: *Forbes* [online]. 2018: Quora, 2018 [cit. 2020-02-17]. Dostupné z: <https://www.forbes.com/sites/quora/2018/02/02/the-difference-between-virtual-reality-augmented-reality-and-mixed-reality/>
- [3] The mixed reality spectrum. In: *Microsoft* [online]. _, 2018 [cit. 2020-02-17]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/mixed-reality>
- [4] POETER, Bridget. Historie Rozšířené reality. In: *Learn G2* [online]. Chicago: Learn G2, 2017 [cit. 2020-02-03]. Dostupné z: <https://learn.g2.com/history-of-augmented-reality>
- [5] Small planet Videoplace. In: *Aboutmyronkrueger* [online]. [cit. 2020-02-04]. Dostupné z: <https://aboutmyronkrueger.weebly.com/small-planet.html>
- [6] VideoPlace. In: *Virtual reality definition* [online]. _ : Word Press, 2016 [cit. 2020-02-03]. Dostupné z: <https://virtualrealitydefinition.files.wordpress.com/2016/12/history-of-augmented-reality.gif?w=840>
- [7] OKAMURA, Allison. Master slave Virtual Fixtures. In: *Research Gate* [online]. 2007 [cit. 2020-02-05]. Dostupné z: https://www.researchgate.net/figure/Guidance-virtual-fixtures-assist-the-user-in-moving-the-manipulator-along-desired-paths_fig1_220122686
- [8] Virtual Fixtures. In: *HistoryofInformation* [online]. [cit. 2020-02-05]. Dostupné z: <http://www.historyofinformation.com/detail.php?entryid=4696>
- [9] Fanuc. In: *Fanuc* [online]. 2016 [cit. 2020-02-05]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/str%c3%a1nka-filtru-robot%c5%af/spolupracuj%c3%ad%c3%ad-roboty>
- [10] BIEGUN, Steve. Software Assumption by Use. In: *Pulsedesigngroup* [online]. 2016 [cit. 2020-02-17]. Dostupné z: <https://images.squarespace-cdn.com/content/v1/5734ac97b654f943bb4e1329/1466442081337-HCH0DSKFN5MNC2UFDK4/ke17ZwdGBToddI8pDm48kI26Eq3Op>

WhoCDKCjiUrtxpZw-
zPPgdn4jUwVcJE1ZvWxYb2krSYGNt8NUBfJA7VeEJFbgE-
7XRK3dMEBRBhUpw9U9jR-
YsMu2aNDC_Abswd3t4AIRVY2VKnLL1XGU5YnpF14us02ALrtHZKR
8Nn-U8/image-asset.png

- [11] AR in medicine Archer soft. In: *Archer soft* [online]. New York [cit. 2020-02-05]. Dostupné z: <https://archer-soft.com/en/blog/how-augmented-reality-used-medicine>
- [12] NuEyes. In: *NuEyes* [online]. [cit. 2020-02-08]. Dostupné z: <https://nueyes.com/products/nueyes-pro/>
- [13] NuEyes Picture. In: *NuEyes* [online]. 2017 [cit. 2020-02-08]. Dostupné z: <https://nueyes.com/wp-content/uploads/2017/12/nueyes-electronic-magnification-glasses-low-vision1200x800.jpg>
- [14] Brain Power. In: *Brain Power* [online]. 2015 [cit. 2020-02-08]. Dostupné z: <http://www.brain-power.com/autism/>
- [15] CITARDI, MJ, A AGBETOBA, JL BIGCAS a A LOUNG. *Augmented reality for endoscopic sinus surgery with surgical navigation: a cadaver study*. [online]. Houston, 2015 [cit. 2020-02-08]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4856592/>. Cadaver Study. University of Texas Medical School at Houston.
- [16] SCOPIS. In: *Scopis* [online]. Berlin: Scopis GmbH, 2015 [cit. 2020-02-08]. Dostupné z: <https://tracing.scopis.com/>
- [17] MOROZOV, Michael. Augmented Reality in Military: AR Can Enhance Warfare and Training. In: *Jasoren* [online]. USA: Jasoren, _ [cit. 2020-02-17]. Dostupné z: <https://jasoren.com/augmented-reality-military/>
- [18] ARES Interface. In: *Youtube* [online]. _: DignitasTechnologies, 2015 [cit. 2020-02-22]. Dostupné z: <https://i.ytimg.com/vi/AxJmFj7nbMA/maxresdefault.jpg>
- [19] MCKENZIE, Jen. AR manuals. In: *Next reality* [online]. _: next reality, 2019 [cit. 2020-02-22]. Dostupné z: https://nextreality.com/wp-content/uploads/2019/08/augmented_reality-1-768x434.png
- [20] Reflekt AR. In: *Re-flekt* [online]. Re-flekt, 2019 [cit. 2020-02-22]. Dostupné z: <https://www.re-flekt.com/portfolio-item/bosch-common-ar-platform>
- [21] MCKENZIE, Jen. AR Applications in the Car Industry. In: *Next Reality* [online]. _: next reality, 2019 [cit. 2020-02-22]. Dostupné z: <https://nextreality.com/2019/08/13/augmented-reality-automobile/>
- [22] Bosch training-picture. In: *Bosch* [online]. _: Bosch, 2018 [cit. 2020-02-22]. Dostupné z: <https://www.bosch->

presse.de/pressportal/de/media/dam_images/pi10719/aa_as_ar_trainer_m
ode_77123_img_h720.jpg

- [23] DAQRI. In: *Daqri* [online]. _: daqri [cit. 2020-02-27]. Dostupné z: https://assets.ctfassets.net/rf6r9wh4bnrh/1qqj875VQMGWWMqQ0IKiKi/7b312efce249e81e810299ca8adaa70f/DQDSGDS2_DSG_DataSheet_Letter.pdf
- [24] Smart Glasses. In: *Daqri* [online]. _, 2019 [cit. 2020-02-27]. Dostupné z: <https://images.ctfassets.net/rf6r9wh4bnrh/14fx3AY4MIUwgIOSuEegew/a64bc195765ce827d2a5a9689486f10e/shah1105x880.png?w=1024>
- [25] Virtual Performance Solution ESI. In: *ESI Group* [online]. [cit. 2020-03-15]. Dostupné z: <https://www.esi-group.com/cz/softwarovareneni/virtualni-realizace/virtual-performance-solution-0>
- [26] SEGARD, Jean-Baptiste. *ROAD TRAILER WITH ORIENTABLE SECONDARY WHEELSET*. 2013. Francie. WO/2013/132468. Uděleno 09.12.2013. Zapsáno 03.08.2013.
- [27] EP Tender Crash test. In: *EP Tender* [online]. France: Technoparc, Espace Cristal, 2013 [cit. 2020-03-15]. Dostupné z: <http://eptender.com/wp-content/uploads/2018/03/r%C3%A9servoir-crash-test-2.png>
- [28] MCV 754. In: *Kovosvit* [online]. Sezimovo Ústí: KOVOSVIT MAS, 2016 [cit. 2020-02-29]. Dostupné z: <https://www.kovosvit.cz/mcv-754-quick-p3.html>
- [29] MCV 754 informace. In: *Kovosvit* [online]. Sezimovo Ústí: KOVOSVIT MAS, 2016 [cit. 2020-02-29]. Dostupné z: <https://www.kovosvit.cz/mcv-754-quick-p3.html#features>
- [30] MAREK, Jiří. *Expertní inženýrství v systémovém pojetí*. První. Praha: Grada, 2013. Expert (Grada). ISBN 978-80-247-4127-7.
- [31] Unity vs Unreal Engine. In: *Creative Bloq Staff* [online]. 3D World: 3D World, 2019 [cit. 2020-03-02]. Dostupné z: <https://www.creativebloq.com/advice/unity-vs-unreal-engine-which-game-engine-is-for-you>
- [32] PLURALSIGHT, . Unreal Engine 4 vs. Unity: Which Game Engine Is Best for You. In: *Pluralsight* [online]. _: pluralsight, 2020 [cit. 2020-03-02]. Dostupné z: <https://www.pluralsight.com/blog/film-games/unreal-engine-4-vs-unity-game-engine-best>
- [33] ČUKOVIĆ, Saša, Michele GATTULLO, Frieder PANKRATZ, Goran DEVEDZIC, Ernesto CARRABBA a Khelifa BAIZID. *Marker Based vs. Natural Feature Tracking Augmented Reality Visualization of the 3D Foot Phantom*. 2015.

- [34] NILS, David. Augmented Reality SDK Comparison. In: *Social Compare* [online]. _, 2020 [cit. 2020-03-08]. Dostupné z: <https://socialcompare.com/en/comparison/augmented-reality-sdks>
- [35] GRIFFITHS, Ian, Matthew ADAMS a Jesse LIBERTY. *Programming C# 4.0: Building Windows, Web, and RIA Applications for .NET with C# 4.0*. 6. Sebastopol, CA 95472: O'Reiillz Media Inc., 2010. ISBN 978-0-596-15983-2.
- [36] FEATHERS, Michael. *Údržba kódu převzatých programů.: Vylepšete výkon, funkce, spolehlivost i ovladatelnost aplikace*. 1. Prentice Hall: Pearson Education Inc., 2009. ISBN 978-80-251-2127-6.
- [37] MARTIN, Robert. *The Clean Coder: A code of Conduct for Professional Programmers*. 1. Old Tappan Road, New Jersey: Pearson Education Inc., 2011. ISBN 978-0-13-708107-3.

11 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

Zkratka	Význam
VR	Virtuální realita
AR	Augmented reality
Fps	Frame per second
GPS	Global positioning system
2D	Dvojměrný
3D	Trojměrný
SW	Software
TAR	Tactical augmented reality
CGI	Computer generated imagery
SDK	Software development kit
IMU	Inertial measurement unit
GUI	Graphical user interface
UI	User interface
CSV	Comma separated values

11.1 Seznam tabulek

TAB. 6.2 TECHNICKÉ PARAMETRY STROJE MCV.	33
TAB. 6.5 POROVNÁNÍ PARAMETRŮ UNITY A UNREAL ENGINE.	36
TAB. 7.2 POSOUZENÍ JEDNOTLIVÝCH KNIHOVEN.	39

11.2 Seznam obrázků

OBR. 1) ZAŘÍZENÍ THE SWORD OF DAMOCLES [4].	19
OBR. 2) ZPŮSOB SNÍMÁNÍ SILUETY V PROJEKTU VIDEOPLACE [6].	20
OBR. 3) PŘÍKLAD SOUSTAVY MASTER-SLAVE [7].	21
OBR. 4) TRENDY V POUŽÍVÁNÍ VR/AR [10].	22
OBR. 5) ZDRAVOTNÍ POMŮCKA NUEYES PRO [13].	24
OBR. 6) VYUŽITÍ SYSTÉMU ARES V PRAXI [18].	27
OBR. 7) PŘÍKLAD POUŽITÍ AR JAKO MONTÁŽNÍHO MANUÁLU [19].	28
OBR. 8) ŠKOLENÍ ZAMĚSTNANCŮ SPOLEČNOSTI BOSCH PŘI POUŽITÍ AR [22].	29
OBR. 9) POUŽITÍ AR DAQRI PŘI IDENTIFIKACI VADNÉHO VENTILU [24].	30
OBR. 10) NÁVRH KONSTRUKCE NOSIČE BATERIE [27].	31
OBR. 10) OBRÁBĚCÍ STROJ MCV 754 QUICK [28].	33
OBR. 11) POSTUP OPERACÍ PŘI NÁVRHU.	40
OBR. 12) 2D OBRÁZEK PRO IMAGE TRACKING.	41
OBR. 13) VYTVOŘENÍ 2D IMAGE TARGET DATABÁZE.	42
OBR. 14) VYZNAČENÍ TRACKOVACÍCH MARKERŮ.	43
OBR. 15) HODNOCENÍ IMAGE TARGET DATABÁZE.	43
OBR. 16) IMPORTOVANÝ 3D MODEL STROJE MCV DO PROSTŘEDÍ BLENDER.	44
OBR. 17) PARAMETRY VYTVOŘENÉ TOPOLOGIE.	45
OBR. 18) ROZDĚLENÍ PŮVODNÍHO MODELU NA JEDNOTLIVÉ SEGMENTY.	46
OBR. 19) IMPORT DATABÁZE: A) EXPORT DATABÁZE Z PORTÁLU VUFORIA, B) IMPORT PACKAGE DATABÁZE DO UNITY.	48
OBR. 20) UŽIVATELSKÉ ROZHRANÍ.	49
OBR. 21) KOMPLETNÍ PŘEHLED PRVKŮ SYSTÉMU.	50
OBR. 22) FUNKCE SLOUŽÍCÍ K OTEVŘENÍ PANELU.	52
OBR. 23) LOGIKA STAVŮ OTEVÍRÁNÍ PANELU.	52
OBR. 24) ULOŽENÉ SOUBORY ANIMÁTORU V JEDNOM ADRESÁŘI.	53
OBR. 25) LOGIKA PRO ZOBRAZENÍ 3D OBJEKTŮ.	54
OBR. 26) FUNKCE START().	55
OBR. 27) FUNKCE AIRBUTTON(): A) VOLÁNÍ FUNKCE STISKEM TLAČÍTKA, B) FUNKCE ZNEVIDITELNĚNÍ PROMÍTANÝCH OBJEKTŮ.	55
OBR. 28) UKLÁDÁNÍ ZOBRAZOVANÝCH PORUCH DO SOUBORU LOG.	56
OBR. 29) FUNKCE OTÁČENÍ PANELU SMĚREM KE KAMERĚ.	56
OBR.30) LOGIKA ZOBRAZOVÁNÍ TEXTU: A) DEKLAROVANÉ PROMĚNNÉ, B) JEDNOTLIVÉ FUNKCE.	57
OBR. 31) PŘÍKLAD FUNKCE NAČÍTÁJÍCÍ PORUCHY Z TEXTOVÉHO DOKUMENTU.	57
OBR. 32) VYTVOŘENÍ LISTU A ZOBRAZENÍ TEXTU.	58
OBR. 33) FUNKCE VYMAZÁNÍ TEXTU.	58
OBR. 34) PŘEPNUTÍ KAMERY DO FUNKCE ŘEZ.	59

OBR. 35) FUNKCE UKONČENÍ APLIKACE.	60
OBR. 37) CHYBĚJÍCÍ KOMPONENTY OBJEKTU CANVAS.....	63
OBR. 38) TESTOVÁNÍ KONEČNÉHO PRODUKTU.	64

12 SEZNAM PŘÍLOH

Aplikace MCV.apk