VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MALOVÁNÍ V 3D PROSTORU S VYUŽITÍM ROZŠÍŘENÉ REALITY

PAINTING IN 3D SPACE USING AUGMENTED REALITY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Milan Kořínek
AUTHOR

VEDOUCÍ PRÁCE — Ing. Vítězslav Beran, Ph.D.
SUPERVISOR

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Kořínek Milan**
Obor: Informační technologie
Téma: **Malování v 3D prostoru s využitím rozšířené reality**
Painting in 3D Space Using Augmented Reality
Kategorie: Uživatelská rozhraní

Pokyny:

1. Seznamte se s existujícími aplikacemi pro kreslení ve virtuální realitě. Prostudujte nástroje a postupy pro realizaci rozšířené reality.
2. Navrhněte UI systému umožňující tvorbu a editaci geometrických 3D objektů pomocí rozšířené reality. Dále navrhněte architekturu systému včetně integrace vybraných existujících dílčích řešení.
3. S využitím vybraných existujících nástrojů navržený systém implementujte.
4. Proveďte experimenty s uživateli na efektivitu a použitelnost řešení.
5. Vytvořte plakát a krátké video prezentující klíčové výsledky vašeho řešení.

Literatura:

- R. Hartson, P. Pyla. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*, Morgan Kaufmann, ISBN-10: 0123852412, 2012
- Dále dle pokynů vedoucího.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Beran Vítězslav, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2017
Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této práce je popsat a následně vytvořit aplikaci, ve které může uživatel malovat v 3D prostoru virtuální a rozšířené reality. V první části práce je objasněno řešení nástrojů, které poskytují možnost malování v 3D prostoru. Dále jsou uvedeny prvky a komponenty, díky nimž je možné vytvořit prostředí, ve kterém se uživatel pohybuje. A konečně je popsán způsob, jak jednotlivé části řešení spojit se speciálnímu hardwarem pro virtuální a rozšířenou realitou.

Abstract

The goals of this work are described and create an application in which the user can paint in 3D space of virtual and augmented reality. First, the solution of the elements that provide the possibility of painting is clarified. It also describes the processing of the environment in which the user moves and interaction with the application. Finally, it is described how to combine individual parts of the solution with special hardware for a virtual and augmented reality.

Klíčová slova

Virtuální realita, rozšířená realita, malování v 3D prostoru, HTC Vive, Unreal Engine 4, ZED kamera.

Keywords

Virtual reality, augmented reality, painting in 3D space, HTC Vive, Unreal Engine 4, ZED camera.

Citace

Milan Kořínek: Malování v 3D prostoru s využitím rozšířené reality, bakalářská práce, Brno, FIT VUT v Brně, 2018

Malování v 3D prostoru s využitím rozšířené reality

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Vítězslava Berana, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Milan Kořínek
3. 4. 2018

Poděkování

Tímto děkuji Ing. Vítězslavu Beranovi, Ph.D. vedoucímu mé bakalářské práce, za vedení a zájem, který mi věnoval. Dále bych chtěl poděkovat všem, kteří mi pomáhali svými připomínkami, radami i náměty.

© Milan Kořínek, 2018
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod ... 6
2 Virtuální realita ... 8
2.1 Historie ... 8
2.2 Současnost ... 8
2.3 Rozdělení virtuální reality ... 13
2.4 Software pro práci s virtuální realitou ... 14
2.5 Využití ... 15
2.6 Existující řešení využití virtuální reality v oblasti malby a kresby ... 18
3 Návrh řešení ... 20
3.1 Popis výsledné aplikace ... 20
3.2 Architektura systému ... 20
3.3 Různé způsoby malování v 3D ... 21
3.4 Datové struktury ... 22
3.5 GUI ... 22
4 Konkrétní řešení ... 24
4.1 Výběr vhodného softwaru ... 24
4.2 Malování ... 25
4.3 Malířská paleta ... 29
4.4 Prostředí ... 31
4.5 Hlavní menu ... 33
5 Testování a vyhodnocení ... 35
5.1 Způsob testování ... 35
5.2 Známé problémy ... 36
5.3 Možná rozšíření ... 37
6 Závěr ... 38
Zdroje ... 40
Literatura ... 40
Elektronické zdroje ... 40
Příloha A - Obsah přiloženého DVD ... 42
Příloha B – Výsledky dotazníku ... 43
Příloha C – Návod k použití aplikace ... 45
Příloha D – Návrh GUI ... 47

1 Úvod

První pokusy s virtuální realitou vznikaly již ve 20. století. Tehdejší zařízení však byla těžká, zobrazování bylo nevalné a technika umožňující tyto nové aplikace byla z důvodu vysoké ceny pro většinu lidí nedostupná. Ale byl to krok, díky kterému bylo možné tyto technologie dále zdokonalovat. Zvláště v posledních několika letech se výkon hardwaru mnohonásobně zlepšil a také jeho cena se již dostala na úroveň, kdy si jej může dovolit téměř každý. Existuje také řada firem, které vyrábějí hardware a aplikace nejen pro virtuální, ale i rozšířenou a mixovanou realitu. V současnosti je už dokonce možné spouštět aplikace pro virtuální realitu nejen na osobních počítačích, ale i v chytrých telefonech, které jsou pro značnou část lidí již běžným standardem. Právě zde se v plném rozsahu potvrzuje známá skutečnost, že rychlost vývoje nových technologií je do značné míry závislá na vytvoření výhodných obchodních podmínek, které se následně promítají do stále větších počtů jejich uživatelů.

Od vytvoření prvního zařízení v roce 1962 se virtuální realita mnohonásobně zlepšila ve všech směrech. Zobrazení je kvalitnější a pohyb uživatele je možné určit s přesností desetin centimetrů. Mnoho aplikací již dokáže uživatele docela úspěšně znejistit v jeho vnímání a snaží se ho přesvědčit, že to, co vidí je skutečné. Nicméně to bude zřejmě ještě nějakou dobu trvat, než se podaří vytvořit takovou realitu, která dokáže na všechny smysly zapůsobit tak komplexně, aby člověk spuštěnou aplikaci vnímal jako skutečnost.

Obrázek 1 – Sensorama 1962[1]

Obrázek 2 – HTC Vive 2016[2]

[1] https://commons.wikimedia.org/wiki/File:Sensorama-morton-heilig-virtual-reality-headset.jpg
[2] http://www.esa.int/spaceinimages/Images/2017/07/Reality_check

Tato práce je rozdělena do tří částí. V první části je vysvětleno, co je to virtuální realita. Jsou zde popsány první historické pokusy a směr, kterým se ubírala až do dnešní doby. Dále je zde provedeno rozdělení virtuální reality do kategorií, a nakonec jsou zde popsána některá zařízení a vývojářský software, díky kterým je možné přenést se do světa virtuální reality.

Druhá část se zabývá dílčími částmi návrhu a jeho řešením. Jsou zde popsány jednotlivé malířské komponenty a části aplikace. Také jsou zde vypsány vzniklé problémy a jejich řešení nebo odklonění od původního návrhu. Vše je rozděleno do samostatných kapitol podle logických celků.

V poslední části práce je popsána výsledná aplikace a její testování. Také jsou zde popsány známé problémy. Jedna kapitola je věnována směrům, kterými by bylo možné aplikaci dále rozvíjet.

2 Virtuální realita

Virtuální realita je způsob zobrazení určitých prvků tak, aby se uživateli nebo divákovi jevily jako skutečné. Může jít o simulaci obrazu, zvuku, chuti, ale i o předstírání čichového nebo hmatového zážitku [17]. K tomu je vždy potřeba nějakého zařízení.

2.1 Historie

Prvním, kdo vytvořil zařízení, které je možno považovat za virtuální realitu byl Morton Heilig. Bylo to v roce 1962 a nazval je Sensorama. V tomto zařízení přehrával pět krátkých filmů, které působily na všechny smysly člověka, kromě chuti [6].

V roce 1968 vytvořili I. Sutherland a B. Sproull zařízení, které měl mít uživatel umístěné na hlavě. Na to bylo však příliš těžké, a proto je museli odlehčit zavěšením na strop. Přes tuto úpravu je považováno za první zařízení pro virtuální realitu typu „Head-mounted display“. Protože umožňovalo zobrazovat geometrické tvary na skutečném pozadí, dalo by se považovat i za zařízení zobrazující rozšířenou realitu [12].

Na MIT vznikl v roce 1978 projekt Aspen Movie Map. Jednalo se o aplikaci, ve které mohl uživatel procházet městem Aspen. Na výběr měl ze dvou ročních období a 3D modelu města. Aplikace fungovala obdobně jako dnešní virtuální prohlídky (Google maps, panoramatická prohlídka mapy.cz…). Všechny přístupné ulice města byly fotografovány a snímky byly poté skládány tak, aby na sebe správně navazovaly. Uživatel si poté na obrazovce vybíral trasu procházky a podle toho se načítaly odpovídající snímky. Město bylo fotografováno v obou ročních obdobích [4].

V dalších letech do konce století pak bylo vytvořeno mnoho aplikací a zařízení pro virtuální realitu. Některé byly úspěšné, jiné skončily jako propadák. Z neúspěšných lze uvést například konzoli Virtual Boy vytvořenou firmou Nintendo v Japonsku v roce 1995 [14]. Nejenže byla drahá, ale její zobrazovací schopnosti a 3D efekt byly velice špatné. Kromě toho způsobovala uživatelům bolesti hlavy, krku a nevolnost. Pro tuto konzoli bylo vyvinuto pouhých 22 her, a i jejich kvalita byla, v porovnání s většinou jiných her, podprůměrná.

2.2 Současnost

V dnešní době jsou aplikace pro virtuální realitu zobrazovány uživatelům přes zařízení, která mají umístěné na hlavě. Rozdíl mezi jednotlivými aplikacemi je v rozsahu a způsobu zobrazování. Z jednotlivých smyslů převažuje zaměření na zrak a hmat.

Smart Glasses

Jedním z typů jsou zařízení, která doplňují skutečný pohled na svět. Takovým zařízením mohou být například speciální brýle, které vypadají jako klasické, ale mají přidaný LCD display, na kterém zobrazují rozšiřující informace. Tyto brýle mají v sobě buď zabudovaný malý počítač, nebo mohou komunikovat s telefonem či klasickým počítačem bezdrátově pomocí Wi-Fi nebo Bluetooth. Takovýchto zařízení existuje mnoho. Mezi známé firmy, nabízející brýle pro rozšířenou realitu, patří například Samsung, Lenovo nebo Toshiba. Nejzajímavější z těchto produktů jsou ale určitě Google Glass nebo Microsoft HoloLens.

Google Glass jsou brýle vyvinuté pro běžné uživatele a jejich každodenní potřeby. Mají silnou podporu ze strany Google. Lze je použít jako fotoaparát, kalendář, komunikátor s přáteli, a dokonce je možné zobrazit si v nich mapu a GPS navigaci, včetně informací o nejbližších obchodech nebo restauracích. Obsahují tedy v podstatě stejné funkce jako webová aplikace Google. Ovládají se pomocí dotykových tlačítek a hlasu a mobilní aplikace, dále obsahují gyroskop, akcelerometr, kompas, světelný senzor a moduly pro připojení přes Wi-Fi nebo Bluetooth. Součástí brýlí je také 5Mpx kamera. Tyto brýle byly vyvinuty v roce 2013, nejsou bohužel zatím dostupné pro běžné spotřebitele, ale pouze pro vývojáře [7].

Zařízení **HoloLens** je větší a nepřipomíná zcela klasické brýle jako Google Glass, ale jeho využití je obdobné. Podporuje ovšem o mnoho více funkcí a aplikací, přičemž podpora funkcí ze strany Microsoftu je samozřejmostí. Patří do řady produktů Windows 10, takže obsahuje také virtuální pomocnici Cortanu. Dále obsahuje několik senzorů (gyroskop, akcelerometr…), 2Mpx kameru a moduly pro připojení stejné jako Google Glass. Zařízení reaguje na hlas a gesta ruky. Pomocí stisku ukazováčku a palce je možné přehrát a pozastavit video nebo spustit zachytávání videa nebo snímku. Poklepáním prstem do prostoru je možné simulovat stisk levého tlačítka myši a tím klikat třeba na webovou stránku virtuálního prohlížeče. Na trhu je od roku 2016 [9].

Obrázek 3 – Google Glass[3]

Obrázek 4 – Microsoft HoloLens[4]

[3] https://x.company/glass/
[4] https://www.microsoft.com/cs-cz/hololens

Virtuální realita pro chytré telefony

Velice levnou variantou virtuální reality jsou headsety pro chytré telefony. Jedná se o pouzdro za pár set korun, které obsahuje držák na telefon a dvě brýlová skla. Do držáku se umístí telefon, ve kterém běží aplikace určená pro tyto headsety. Aplikace se vykresluje na display dvakrát – pro levé a pravé oko. Některé aplikace umožňují pouze naklánět hlavu a tím i vnímaný obraz, jiné zvládnou i otáčení kolem osy, takže se uživatel může rozhlížet kolem sebe. Takto je možné si prohlížet obrázky, video pořízené 360° kamerou, nebo hrát různé hry.

Problémy však někdy bývají s ovládáním, protože telefon je uzavřený uvnitř headsetu (Obrázek 5). Komunikace s uživatelem je u některých aplikací řešena tak, že se nejprve zapnou a poté se telefon vloží do headsetu. Jiný způsob komunikace je možný pomocí ovládacího bodu, umístěného uprostřed obrazovky, který uživatel ovládá pohybem hlavy, a tak směřuje bod na určité místo. Po zastavení bodu, začíná načítání a komponenta se aktivuje obdobně, jako při dotyku prstem. Další možností je ovládání aplikace pomocí ovladače připojeného přes Bluetooth nebo Wi-Fi. Tento ovladač má na sobě několik tlačítek a joystick. Některé druhy chytrých telefonů však s tímto ovladačem nespolupracují dobře.

Obrázek 5 - Chytrý telefon vložený do headsetu[5]

Výkon a kvalita tohoto druhu virtuální reality závisí pouze na výkonu telefonu a jeho výbavě. Například proto, aby mohl uživatel otáčet hlavou kolem osy je potřeba, aby telefon obsahoval komponentu gyroskop, bez které telefon, a tím pádem ani aplikace, nepozná, že uživatel otočil hlavou. Proto je s telefony bez tohoto prvku možné pouze prohlížení a spouštění těch aplikací, při kterých se uživatel dívá jen dopředu, což ale značně snižuje zážitek.

[5] http://www.bobovr.com/product/bobovrz4/

Některé headsety mají v přední části pouzdra otvor, přes který je možno kamerou telefonu vidět reálný svět, a tedy spouštět i aplikace pro rozšířenou realitu.

Headsety jsou různé. Liší se způsobem instalace telefonů a další výbavou, například polstrováním, nebo zabudovanými sluchátky. Primitivní headset lze vyrobit i amatérsky podle předlohy z kartonového papíru.

Zařízení tohoto typu umožňují využívat virtuální realitu pouze s minimálními náklady, a proto jsou velice oblíbená a rozšířená.

Virtuální realita pro „nepřenosná“ zařízení

Nejlepších zážitků z virtuální reality lze dosáhnout pomocí zařízení připojeného k počítači nebo herní konzoli (díky vylepšením, jako je například MSI VR One Backpack, je možné odstranit nevýhodu nepřenositelnosti). Headset tohoto zařízení obsahuje dva samostatné displeje, na které se renderuje obraz posunutý pro každé oko zvlášť. Počítač (konzole) však musí být velice výkonný, protože je třeba, aby překresloval obraz dvakrát rychleji než na klasickém monitoru. Headset obsahuje buď čidla, nebo je snímán speciální externí kamerou a je schopen rozpoznávat nejen pohyb a otáčení hlavou, ale také pohyb v prostoru. Tento typ virtuální reality bývá obvykle doplněn o další zařízení (ovladač) pro komunikaci s aplikacemi. Možnosti využití těchto aplikací jsou velmi široké a lze v nich provádět rozličné úkoly. Mohou sloužit pro pouhé prohlížení, ale zároveň dávají uživatelům i možnost pohybu a provádění různých úkonů v prostoru. S pomocí přídavných ovladačů se mohou předmětů ve virtuální realitě nejen dotýkat, ale také s nimi manipulovat.

Asi největší nevýhodou tohoto typu virtuální reality je její vyšší cena, a ne vždy možná přenositelnost. Za problém lze považovat i vyvolávání kinetózy u některých uživatelů. Ten by však měl být v budoucnu z větší části eliminován postupným zlepšováním kvality headsetů a optimalizací aplikací.

Prvním produktem nové generace nepřenosných zařízení byl **Oculus Rift**, vlastněný společností Facebook a dostupný od roku 2016. Je to headset se dvěma displeji s rozlišením 1080x1200 bodů pro každé oko a se zabudovanými sluchátky. Zpočátku neměl ovladače a interakce probíhala pomocí herního gamepadu. Aktuální verze neumožňuje ani pohyb v prostoru. Především tyto dva uvedené nedostatky snižují jeho kvalitu a řadí ho až na druhé místo za HTC Vive. Na druhou stranu má však v porovnání s ním výrazně nižší cenu [10].

Nejlepším produktem v oblasti virtuální reality je zařízení **HTC Vive**. Bylo vyrobeno ve spolupráci společností HTC a Vive Corporation. Zařízení je podobné Oculusu, ale v některých oblastech jej výrazně předčí. Součástí kompletu jsou dva bezdrátové ovladače, částečně nahrazující pohyb rukou uživatele, což značně zvyšuje zážitek z virtuální reality.

Dalším prvkem jsou dvě snímací kamery, díky kterým lze sledovat pohyb hlavou i pohyb v prostoru omezeného na plochu 5x5 metrů [15].

Obrázek 6 - Oculus Rift[6]

Obrázek 7 - HTC Vive[7]

Vstupní zařízení pro mixovanou realitu

K zobrazení skutečného světa ve formě pozadí bez navazující interakce virtuálních objektů, stačí obyčejná kamera, která pořizuje snímky prostředí a posílá je do aplikace. Pro plnohodnotné napojení virtuálních objektů do prostředí skutečného světa je pak nutné další zařízení schopné vypočítat vzdálenosti a vytvořit tak hloubkovou mapu s přesným určením míst, kde se jednotlivé prvky v prostředí nachází. Aplikace využívá takto získaná data k vytvoření vzájemných vztahů mezi skutečnými a virtuálními objekty. Ve výsledku vznikají v podstatě tři možnosti. V prvním případě je virtuální objekt zobrazován na popředí. Ve druhém případě prochází virtuální objekt objektem skutečným, přičemž je vždy zobrazen pouze částečně. Ve třetím případě, je virtuální objekt umístěn dál než objekt skutečný a v takovém případě aplikace musí zajistit, aby se nevykresloval, což znamená, že z daného směru pohledu musí být skrytý.

Kinect je zařízení určené pro herní konzole Xbox 360, Xbox One a pro počítače s operačním systémem Windows. Má mnoho funkcí, z důležitějších je třeba uvést vytváření hloubkové mapy. K tomu využívá dva komponenty – infračervený laser a snímač. Laser promítá do prostoru uskupení teček, které snímač zpracovává a následně z nich vypočítává hloubku obrazu, která je základem pro vytváření aplikací mixované reality [8].

Jiný způsob vytváření hloubkové mapy pro tvorbu mixované reality využívají stereoskopické kamery. Zařízení mají dvě (může být i víc) samostatné kamery, které jsou posunuté tak, aby simulovaly lidské oči. Objekt je každou kamerou snímán z jiného úhlu, díky

[6] https://commons.wikimedia.org/wiki/File:Oculus_Consumer_Version_1.jpg
[7] https://www.vive.com/us/product/vive-virtual-reality-system/

čemuž se poté výsledný obraz jeví jako plastický nebo je možné pomocí triangulace vypočítat hloubku obrazu.

Na tomto principu pracuje **ZED kamera**, která dokáže určovat hloubku až do vzdálenosti 20 metrů. K tomu využívá přepočtů ze získaných barevných obrazů dvou kamer.

Zároveň dokáže díky senzorům 6DOF (z anglického six degrees of freedom) určovat svoji vlastní pozici v prostoru, což je důležité v případech, kdy je k dispozici pouze headset bez tohoto vybavení [11].

Obrázek 8 – Kinect[8]

Obrázek 9 - ZED kamera[9]

2.3 Rozdělení virtuální reality

Oblast virtuální reality je velice široká a lze ji rozdělit do čtyř hlavních kategorií, které se vzájemně více či méně liší obsahem, rozsahem a způsobem toho, co zobrazují.

Virtuální realita

Jedná se o aplikace, kde je celé prostředí nasimulované počítačem. To znamená, že vše, co uživatel vidí, bylo předem vytvořeno programátorem a reálně tedy neexistuje. Tento pojem se také používá k obecnému popisu všech kategorií.

Rozšířená realita

Tyto aplikace využívají kamery a uživateli tak zobrazují jako základ skutečné prostředí. Do tohoto prostředí pak ale přidávají virtuální prvky (objekty, okna, postavy), které jsou vykreslovány na popředí tak, aby co nejvíce zapadaly do skutečného promítaného prostředí [16].

Mixovaná realita

Je to stejný způsob zobrazování jako předchozí, to znamená reálný svět s virtuálními prvky. Rozdíl je však v tom, že virtuální prvky na svět skutečný přímo reagují. Například virtuální

[8] https://commons.wikimedia.org/wiki/File:Xbox-360-Kinect-Standalone.png
[9] https://www.stereolabs.com/zed/

kulička se bude pohybovat po stole, dokud nenarazí na skutečnou překážku, o kterou se zastaví [16].

Hyper realita

Jedná se o způsob virtuální reality, kde jsou virtuální objekty v aplikaci přesně mapovány na ty skutečné. Příkladem může být 3D model stolu, na který si může uživatel skutečně sáhnout, protože před stejným stolem právě stojí.

2.4 Software pro práci s virtuální realitou

Při práci s virtuální realitou může být využit například jeden z následujících softwarů.

Unity

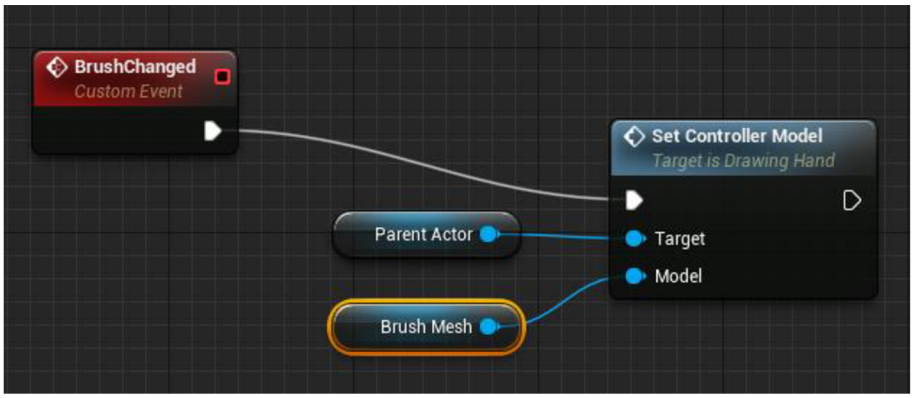
Unity je herní engine vyvinutý v roce 2006 firmou Unity Technologies, původně pouze pro OS X a postupně vylepšovaný a modifikovaný pro různé další platformy. S jeho pomocí je možné vytvářet 2D a 3D hry. Základem vývoje aplikací je vnitřní Unity Script, který je nadstavbou JavaScriptu doplněný o programovací jazyk C#. Podporuje grafické knihovny Direct3D, OpenGL, WebGL a další. Protože patří mezi nejpoužívanější herní enginy, existuje mnoho poznatků, návodů, webových stránek a diskusních fór, kde jeho uživatelé řeší své praktické zkušenosti. Tak vzniklo i mnoho zásuvných modulů a knihoven, které jsou velmi nápomocné při řešení specifických problémů jednotlivých uživatelů. Ani v současné době, kdy se rychle vyvíjí virtuální realita, nezůstává tento engine pozadu a stále je v něm možno nalézt prostor pro vytváření originálních aplikací pro virtuální a rozšířenou realitu [13].

Unreal Engine

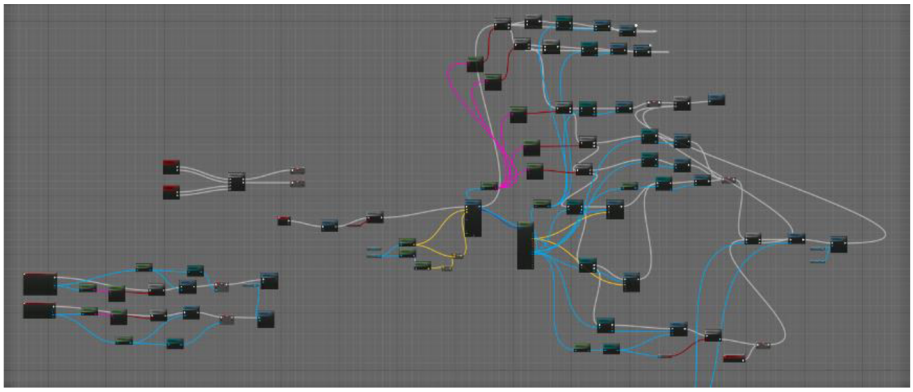
První verze tohoto herního enginu vnikla již v roce 1998 ve firmě Epic Games a původně byla vyvíjena pouze pro „střílečky“. Současná verze je již čtvrtá a její využití je podstatně širší od již zmíněných „střílček“, přes strategické až po masivně multiplayerové hry. Aplikace vzniklé na jeho bázi lze vyvíjet nejen pro osobní počítače s různými operačními systémy a konzole (Xbox, Playstation), ale i pro web nebo chytré telefony. Nově je přidána podpora virtuální a rozšířené reality. K této poslední verzi pravidelně vycházejí nové updaty a vylepšení [18].

Tvorba aplikace je možná dvěma způsoby. Jedním z nich je skriptování v Blueprintech. Blueprinty jsou grafické editory znázorňující třídu. Tento způsob funguje na jednoduchém přetahování hotových uzlů na pracovní plochu a jejich následné spojování. Naskriptovaná část pak vypadá jako graf (Obrázek 10). Zároveň odpadá potřeba dlouhého vypisování zdrojového

kódu a také se tím současně zajišťuje jistá kontrola, například při přetypování nebo syntaxi. Určitá nevýhoda tohoto způsobu programování se projevuje zejména v případech rozsáhlejší logiky, kdy začne být graf méně přehledný (Obrázek 11).



Obrázek 10 - Jednoduchý graf v Blueprintu



Obrázek 11 - Méně přehledný graf v Blueprintu

Druhý způsob je klasické programování a psaní zdrojového kódu v jazyce C++. Ten se hodí zejména pro výše zmíněné případy rozsáhlejší logiky. Unreal Engine je propojen s Visual Studiem od firmy Microsoft, jehož funkce jsou již delší dobu na vysoké úrovni a odhalování chyb se tím značně zjednodušuje.

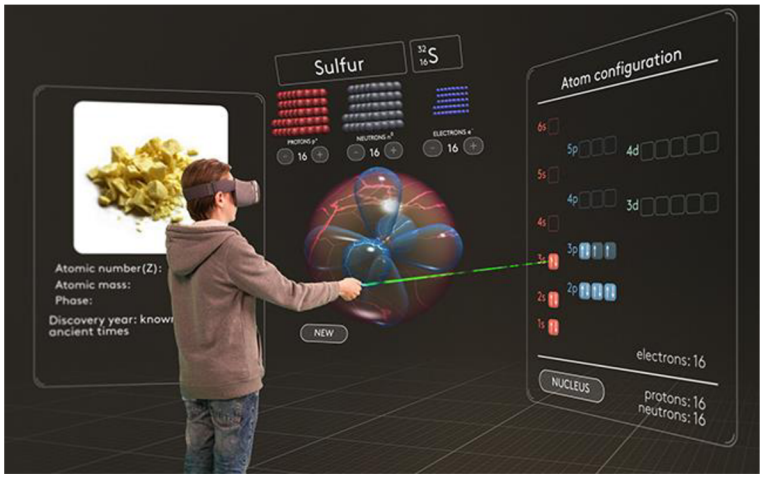
2.5 Využití

Virtuální realita nalezla využití v mnoha oborech. V některých se využívá více v jiných méně a v jiných je její zakomponování velice složité. Dávno už ale neslouží pouze pro zábavu.

Školství

Ve školství může být virtuální realita využita pro oživení nudné látky a tím příznivě ovlivňovat její pochopení studenty. Příkladem může být výuka historie, kdy učitelka bude třeba vyprávět

o Spojených státech, zatímco vedle ní bude Benjamin Franklin podepisovat deklaraci nezávislosti. V přírodních vědách se mohou zase studenti objevit uprostřed chemické reakce nebo si prohlížet a skládat kostru dinosaura. Určitý problém ale spočívá v tom, že výuka probíhá většinou hromadně, zatímco virtuální realita je zaměřena především na jednotlivce.



Obrázek 12 - Využití VR ve školství[10]

Strojírenství

Aplikace se uplatňují zejména při konstrukci nových výrobků. V současnosti se prototypy strojů většinou vytvářejí pomocí 3D modelů, které jsou však často složité a na monitoru počítače mohou být některé jejich části přehlédnuty. Ve virtuální realitě je možné použít stejný 3D model u kterého je však možné provést mnohonásobné zvětšení a tím i zjistit nedostatky těch nejmenších detailů.

Obchod a marketing

Z této oblasti je v současnosti virtuální realita již běžně využívána např. v oblasti projektování konkrétních zařízení bytů a také při prodeji aut s výbavou odpovídající v detailech přání jednotlivých zákazníků.

Trenažéry

Místo konstrukce trenažéru je možné vytvořit pouhou aplikaci. Existují třeba aplikace nahrazující trenažéry pro jízdu v autě, a dokonce je možné simulovat i let raketou a přistání na měsíci. Obdobné využití mohou mít aplikace také v armádě, např. při seznamování vojáků s technikou před tím, než nasednou do skutečného stroje [17].

[10] https://steamuniverse.com/articles/2017/06/15/mel-science-launches-virtual-reality-chemistry-lessons.aspx

Lékařství

Zde se nabízejí velké možnosti využívání virtuální reality pro simulaci operací. Další oblastí jsou rehabilitace, kde je možné díky snímání pohybu pacientů podstatně lépe vyhodnocovat postup změn a hodnotit jejich celkový stav. Jako velmi perspektivní se jeví také využívání virtuální reality pro léčbu některých fobií [17].

Obrázek 13 - Využití VR v lékařství[11]

Zábava

V této oblasti existuje řada her a aplikací, které slouží pouze k zábavě od prohlížení videa, přes odreagování na pláži na druhém konci světa, až třeba po střílení zombie.

[11] https://medialist.info/en/2018/01/26/vr-healthcare-virtual-reality-in-medicine/

2.6 Existující řešení využití virtuální reality v oblasti malby a kresby

Nejnovější generace virtuální reality je teprve na počátku své cesty. Dobrých aplikací není mnoho, a těch z nich, které umožňují malování, je ještě méně. Například specializovaný internetový obchod Steam nabízí cca čtyři, funkčnost dvou z nich je však problematická. Naproti tomu kvalita dvou dalších, níže uvedených, je na poměrně vysoké úrovni.

KingSpray Graffiti VR[12]

Tato placená aplikace je velice realistický simulátor sprejování. Uživatel si může vybrat prostředí a plochu, na které bude vytvářet své dílo. Jeden ovladač slouží jako výběrové menu a druhý ovladač je sprej. Aplikace věrně napodobuje mimo jiné i částečky sprejované barvy, přilnavost k povrchu a zejména konečné změny vzhledu při různých vzdálenostech sprejů od povrchu.

Tilt Brush

Další placenou aplikací, která využívá virtuální realitu a zároveň nabízí mnoho funkcí je aplikace Tilt Brush[13]. Byla vytvořena v roce 2016 firmou Google. Obsahuje řadu nástrojů, díky kterým mohou vznikat velice zajímavá díla. Jedná se třeba o více typů štětců, rozdílné styly malování nebo různé doplňkové efekty. Tato aplikace je velmi dobře koncipovaná, k čemuž navíc přidává nesporné výhody elektronického zpracování, neumožňuje však malbu dílčích prvků kresby (jednotlivých čar) ve formátu 3D.

Obrázek 14 - Tilt Brush[13]

[12] Dostupné z: https://store.steampowered.com/app/471660/Kingspray_Graffiti_VR/
[13] Dostupné z: https://store.steampowered.com/app/327140/Tilt_Brush/

Aplikace pro malování v mixované realitě

Tato oblast se v současnosti nachází zřejmě pouze ve fázi různých rozpracovaných projektů. Toto konstatování vychází ze skutečnosti, že žádná dokončená a vydaná aplikace nebyla nalezena. Rozpracované projekty jsou většinou orientovány na využití v chytrých telefonech, kde malbu zajišťuje (zatím nedokonale) malovací bod uprostřed displeje ovládaný pohyby telefonu. V průběhu malování je pouze možnost změny barev a přepínání tloušťky čáry.

Pro malování v mixované realitě na počítači byl nalezen pouze jeden, zatím téměř nefunkční projekt, v počáteční fázi rozpracování.

3 Návrh řešení

V následující kapitole je popsán návrh aplikace pro malování ve virtuální a rozšířené realitě. Nejdříve je popsána aplikace a její funkce jako celek. V další části je návrh řešení jednotlivých částí aplikace, a nakonec je popsán a zobrazen návrh uživatelského rozhraní.

3.1 Popis výsledné aplikace

Cílem práce je navrhnout a poté implementovat aplikaci, ve které bude moci uživatel malovat v 3D prostoru. K dispozici bude mít několik nástrojů, které budou svým vzhledem připomínat ty skutečné. Pomocí nich bude moci vytvářet barevné čáry, upravovat jejich vlastnosti a také je i mazat.

Uživatel si bude moci libovolně přepínat mezi prostředím, ve kterém se bude chtít pohybovat. Na výběr bude virtuální realita a kombinace skutečného světa s virtuálními prvky v podobě rozšířené reality. Pozadí virtuální reality bude tvořit jednotná barva nebo přednastavená fotografie.

Veškerou práci bude možné uložit do souboru tak, aby mohl uživatel kdykoliv v budoucnu pokračovat tam, kde skončil.

3.2 Architektura systému

Navrhovanou aplikace bude nutné spojit s hardwarem umožňujícím využití virtuální reality. Ten navíc musí poskytovat přesné určování pozice nejen uživatele, ale i jeho rukou pro realističtější zážitek z malování. Proto při výběru odpadají zařízení, která nemají ovladače nebo jiné „trackovací[14]“ prvky. Také virtuální realita pro chytré telefony není příliš vhodná v případě, že by nebylo připojeno zařízení pro zpracování gest.

Aby bylo možné zobrazit rozšířenou realitu je nutné přidat další hardware pro distribuci prvků ze skutečného světa. Tím může být kamera s vysokým rozlišením, aby se obraz jevil maximálně reálný, a dobrou přenosovou rychlostí pro zajištění plynulého běhu obrazu. Aby zážitek co nejvíce odpovídal realitě, je nutné umístit kameru, která snímá skutečný svět, do výšky uživatelových oči. Poté lze vytvořit před uživatelem virtuální obrazovku, na které se zobrazuje okolní svět.

Dalším úkolem bylo zamezit tomu, aby se virtuální objekt, který zůstává na své pozici, při pohybu dostal za zobrazovací plochu (nebyl by vidět). Tento problém je možné řešit dvěma

[14] Tracking – určování pozice objektu v prostoru

způsoby. První možnost je, že všechny virtuální objekty se budou renderovat na popředí. K tomu je možné použít hloubkový buffer. Druhý způsob by fungoval opačně, to znamená, že zobrazování snímků z kamery se bude vykreslovat vždy na pozadí.

Hlavní funkcí je malování v 3D prostoru. Jako nejlepší se jeví využít některý z herních enginů proto, že již obsahují základní herní prvky, které je možné využít v navrhované aplikaci, a také proto, že novější enginy obsahují navíc komponenty a knihovny pro práci s virtuální realitou. Díky tomu bude mnohem jednoduší propojit naprogramované prvky s navrženým hardwarem.

3.3 Různé způsoby malování v 3D

Výhodou malby na počítači je neomezená možnost úprav. Uživatel může svoje dílo kdykoliv smazat, odstranit část kresby, nebo se vrátit o několik kroků zpět. To tradiční malba neumožňuje vůbec, nebo jsou úpravy velice obtížně. V reálném ateliéru musí mít malíř dopředu naplánované, jaké nástroje a barvy bude používat [1]. V elektronické podobě má naproti tomu po ruce všechny nástroje, které si může vybírat podle potřeby pouhým stisknutím tlačítka. Dalším omezením při tradiční malbě jsou rozměry plátna. Obvykle se jedná o velikost v řádech nejvýše jednotek metrů. Naproti tomu na počítači je tento prostor prakticky neomezený, což je v některých případech další, nezanedbatelná výhoda.

Simulace tradiční malířské tvorby

Jednou z možností, jak malovat ve virtuální realitě, by mohlo být malování na virtuální plátno (nebo jiný povrch). Uživatel aplikace by pak maloval obvyklou malířskou technikou a také výsledek by byl podobný tradičnímu výtvarnému projevu. V tomto případě se tedy jedná v podstatě o obdobu klasické malby s tím rozdílem, že výtvarník není fyzicky vázán na místo nebo předměty, které zobrazuje, ale vytvoří si je libovolně kdekoliv ve virtuální podobě. Toto řešení navíc umožňuje i vytvoření patřičné atmosféry, připomínající skutečné prostředí, kterou každý výtvarník při své práci nezbytně potřebuje [3]. Přes tyto výhody, pro některé malíře, kteří jsou zvyklí tvořit ve skutečném prostředí, nebude zřejmě malba ve virtuální realitě plnohodnotnou alternativou. Nicméně i v jejich tvorbě by mohla plnit alespoň částečně doplňkovou funkci.

Při takovémto způsobu malování by bylo možné využít několika klasických grafických funkcí a odpadal by problém s 3D vzhledem. V takovém případě by se pravděpodobně využil nějaký způsob úprav textury za běhu nebo vzájemné překrývání nástroji typu *Decal*.

Trojrozměrné malování

Tradiční malbu výtvarníci realizují pouze na ploše. Své dílo a techniku jeho provedení, včetně výběru nástrojů, si předem připraví a rozvrhnou v mysli [3]. Co by se však stalo, kdyby měli k dispozici třetí rozměr? Jak by v takovém případě fungovaly jejich běžné nástroje, na které jsou zvyklí? To by mohl být značný problém, protože malíř zná pouze, jaký je například výsledek tahu štětcem na ploše, ale nedokáže si představit výsledek stejného tahu štětcem ve trojrozměrném prostoru. Proto je potřeba tyto netradiční 3D nástroje co nejvíce přiblížit těm skutečným, aby výsledek byl obdobný, ale zajímavější.

U samotné malby se postup výtvarníka nijak neliší od tradiční tvorby. Jednotlivé prvky malby se mění zejména v závislosti na tlaku a úhlu natočení použitých výtvarných nástrojů (štětec, pastel, sprej…) [3]. Otázkou zůstává, jaký bude výsledek použití těchto nástrojů ve třetím rozměru.

Při malování v 3D prostoru je zobrazování výsledných čar složitější než na monitoru. Uživatel se může na své dílo podívat všech stran. Proto je třeba klást větší důraz na detaily tak, aby byla malba kvalitní ze všech úhlů pohledu [2]. Aby bylo tohoto efektu dosaženo, bude potřeba vytvořit 3D model znázorňující čáru nebo umístit 2D čáru do prostoru tak, aby se natáčela podle pozice uživatele (tím by nepoznal, že je zobrazovaná čára plochá).

3.4 Datové struktury

Malování bude nejspíš realizováno vytvářením dílčích prvků na pozici ovladače, které po složení vytvoří dojem jednoho velkého objektu. Proto je vhodné vytvořit samostatnou třídu, která bude spravovat dílčí komponenty vzniklé čáry ať už se bude jednat o 2D nebo 3D řešení. Pokud by byla jen vytvořena čára v prostoru, další práce s ní by byla složitá.

Každý pracovní nástroj by bylo vhodné umístit do samostatné třídy, která má vlastní logiku a není přímo ovlivňována ostatními. Tyto menší části je opět vhodné obalit rodičovskou třídou, která bude zajišťovat jejich správné chování jako celku.

Aby mohla být zobrazována rozšířená realita, bude nutné vytvořit speciální typ materiálu (popř. textury), protože vlastnosti klasických materiálu, které se ve většině enginů vytvářejí již při tvorbě aplikace, není možné měnit.

3.5 GUI

Při návrhu je potřeba myslet na to, že uživatel se má soustředit na malování, a ne na hledání svých nástrojů. Výhodou virtuálních výtvarných nástrojů je, že jsou k dispozici v širokém

sortimentu na jednom místě. Uživatel si může vybírat nejen nejvhodnější nástroj, ale také do značné míry nastavit jeho vlastnosti, například barvu, velikost, tvar nebo různé efekty (průhlednost, zrnění…). To však předpokládá u uživatelů větší technické znalosti při práci s počítačem a využití těchto možností může být pro některé méně technicky zdatné značně limitující. Proto je nutné rozvrhnout komponenty tak, aby byly intuitivní a poskytovaly potřebnou funkčnost. V aplikaci budou dva hlavní prvky s možností výběru.

Prvním z nich bude pracovní nástroj, který bude velice často používán. Protože se jedná a uměleckou aplikaci, navržený vzhled připomíná malířskou paletu s rozmíchanými barvami, které budou sloužit k výběru barvy. Na paletě dále budou prvky, díky nimž se budou upravovat vlastnosti vybrané barvy. Pro odstranění chybných tahů byla navržena mistička s vodou, která bude sloužit k „vyčištění“ štětce od barvy. Touto paletou (Obrázek 28) bude nahrazen jeden z ovladačů.

Aplikace bude obsahovat několik funkcí, které není vhodné umístit na malířskou paletu jednak proto, že nebudou mít přímou návaznost na výslednou tvorbu a také proto, že nebudou často využívány. Patří mezi ně ukládání a načítání výtvoru, reset aplikace a přepínání mezi virtuální a rozšířenou realitou. Tyto funkce budou umístěny do hlavního menu, které bude možné vyvolat stiskem tlačítka (Obrázek 30).

Veškerá interakce bude probíhat pomocí ovladače. Je vhodné využít co nejméně tlačítek, protože uživatelé mívají problém si zapamatovat, co které dělá. Dále je vhodné rozdělit funkce podle důležitosti a pak je přiřadit k tlačítkům tak, aby nejdůležitější byly na hlavním tlačítku (budou se nejčastěji používat) a ty nejméně důležité funkce byly například na tlačítku, na které je nutné přesunout prst, čímž se zajistí, že uživatel nebude mít potřebu je používat.

Budou potřeba minimálně dvě tlačítka a směrový joystick či dotykový displej nebo šest tlačítek (hlavní akce, vyvolání menu, pohyb).

4 Konkrétní řešení

V procesu hledání možných řešení problémů je vždy vhodné vycházet i z názorů potencionálních uživatelů, v daném případě tedy aktivních výtvarníků. Před zahájením práce byl proto vytvořen dotazník, o jehož vyplnění byli požádání lidé z této skupiny. Odpovědi dotazníku jsou zaznamenány v příloze (Příloha B – Výsledky dotazníku).

4.1 Výběr vhodného softwaru

Jak již bylo popsáno v úvodní části, ze současně dostupných softwarů se jeví jako nejvhodnější Unity a Unreal Engine 4, protože oba obsahují komponenty pro práci s virtuální realitou. Po jejich vyzkoušení byl vybrán Unreal Engine 4. Jeho výhodou je zejména větší rozšířenost v různých skupinách uživatelů (studenti, malí vývojáři, ale také velké společnosti jako je například Square Enix nebo Capcom [5]). Další výhodou je snadnější ovladatelnost a lépe navržené a přehlednější vývojové prostředí. Velice dobře je také zpracován způsob programování a skriptování, kde řadu základních programovacích struktur lze vytvořit díky systému *Blueprintů* pouze několikerým kliknutím a zároveň je možné průchod naskriptovaným grafem sledovat za běhu, což značně usnadňuje hledání chyb.

Výhodou obou enginů je jejich podpora hardwaru pro virtuální realitu, který byl k dispozici.

Prvním byl Oculus Rift, u kterého byla nevýhodou nemožnost fyzického pohybu. Uživatel by musel pouze sedět nebo stát, což se jevilo jako velký handicap ve využívání aplikace. Příliš kvalitní nebylo ani zobrazování a velkým problémem byla silná kinetóza, která se projevovala již po krátké době používání.

Druhým testovaným produktem byly brýle HTC Vive, u kterých se žádný z výše uvedených problémů neobjevil. V porovnání s prvním zařízením se uživatel může pohybovat v prostoru až 5x5 metrů, což je důležité proto, aby mohl dílo, které tvoří, fyzicky obcházet ze všech stran. Pokud by zde tato možnost nebyla, musel by se pohybovat pouze programově (pomocí tlačítka), což však neumožňuje plnohodnotný zážitek. Z uvedených důvodů padla jednoznačně volba na tento hardware.

4.2 Malování

Jednou z možností při tvorbě čar bylo využití vestavěné funkce *DrawLine*. Tato možnost se však neosvědčila jednak proto, že se čára vykresluje pouze dvojrozměrně, a také protože čáru nelze zaoblit.

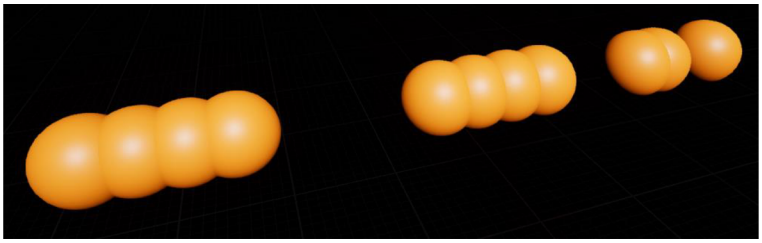
Zde se nabízí řešení, spočívající ve vytvoření 3D modelu, který bude čáru reprezentovat. Válec, koule a krychle byly modely vybrané pro experimentování.

Malování skládáním objektů

První navržená třída fungovala tak, že při každém tiknutí se vytvoří objekt na pozici ovladače. To umožňuje vestavěná funkce *EventTick*, která se aktivuje při každém snímku (největší možnou rychlostí), to znamená, při 90 FPS „tikne“ maximálně 90x za vteřinu. V případě potřeby lze tuto rychlost tikání nastavit i nižší.

Z výše uvedených testovaných 3D modelů se v tomto případě nejlépe osvědčily koule a krychle.

Tento způsob nebyl přesto zcela ideální, protože nebylo dosaženo hladké čáry, kdy ve všech případech bylo patrné, že se jedná o množinu dílčích objektů. Kromě toho, docházelo při velkém množství vytvořených objektů k poklesu FPS. Další problém se objevoval při rychlejším pohybu ovladače. Pokud se ovladač pohyboval normální rychlostí, funkce měla dostatek času na to se spustit a vytvořit objekt, při rychlejším pohybu ovladače se však všechny objekty nestihly vytvořit a na trase tak vznikaly mezery (Obrázek 15).



Obrázek 15 - Chyby při rychlém pohybu ovladače

V prvním návrhu byly použity pro zobrazení modelů obyčejné *Static Meshe*, což jsou samostatné objekty, s vlastními atributy. Pokud byla namalována dlouhá čára, ve scéně bylo mnoho objektů, které měly sice stejné vlastnosti, ale každý existoval autonomně. A právě to se projevilo negativně na výkonu. Proto byly místo *Static Meshů* použity instance jednoho objektu. Uživatel nepoznal rozdíl, ale výkon se mnohonásobně zlepšil. Při testování bylo vytvořeno přes 10 000 instancí a pokles výkonu téměř nenastal, naproti tomu při použití *Static Meshe* bylo možné bez ztráty výkonu vytvořit pouze několik set objektů.

Řešení vznikajících mezer zajistilo rozšíření třídy o novou doplňovací funkci, která po ukončení procesu malování vyhodnotila vzniklé mezery mezi sousedními objekty, a v případě že se do těchto mezer vešly další objekty, funkce je tam doplnila. Tím bylo dosaženo rovnoměrných čar sestávajících z plynule navazujících objektů bez mezer.

Tato třída však nebyla nakonec využita, protože ani výše popsané úpravy nezajistily dosažení hladké a bezchybné čáry. Proto byla vytvořena nová třída, která do značné míry eliminuje tyto nedostatky a ta je popsána v následujícím odstavci.

Malování pomocí objektu Spline

Další možností, jak malovat 3D čáru, bylo využít vestavěný objekt *Spline*. Ten funguje tak, že se vytvoří a spojí několik bodů, které mají svoji pozici a natočení určené pomocí tečny. Tyto dvě vlastnosti potom určují vzhled vzniklé křivky, na jejíž trasu jsou poté skládány objekty (*Spline Meshe*), které se deformují podél této trasy.

Vytvořená třída je podobná předchozí s tím rozdílem, že využívá *Spline* objekt. Vytváření bodů spouští každý tik a pozice bodů je určena ovladačem, stejně jako v předchozí třídě. Vždy po přidání dalšího bodu se vezmou poslední dva body křivky, jejich pozice a natočení a tyto dva parametry se použijí jako vstup pro deformaci *Spline Meshe*. Nevýhodou však je, že body, které jsou od sebe příliš vzdálené, objekt nepřiměřeně deformují.

Další možností bylo skládat *Spline Meshe* za sebe a nenatahovat je neomezeně mezi dva body, ale limitovat jim ručně maximální velikost natažení. Vzdálenost mezi body na křivce tak může být větší, protože tyto nastavené limity zamezují výše uvedeným nepřiměřeným deformacím objektů. Pokud však byly body na křivce vzájemně příliš blízko, vznikaly v těchto místech artefakty.

Každá z výše uvedených metod tedy vyhovovala pro jinou část křivky. Pro obě metody řešení jsou vhodné modely, které lze na sebe jednoduše napojit. Z uvedeného důvodu nebylo možné použít model koule.

Protože každá část křivky je specifická (délka, zaoblení), není možné použít instance objektů jako v předchozím řešení, a proto bylo potřeba nalézt jiné způsoby, jak tuto třídu optimalizovat.

První optimalizace byla ve výběru modelů

Výběr modelu

Bylo experimentováno s osmi modely tvaru válce s různými úpravami. Každý válec měl průměr 5 cm, který odpovídal středně tlusté čáře při malování. U jednotlivých modelů se měnila délka

a počet polygonů. Válce měly délky 5, 10, 25 a 50 cm. Na délku byl každý model rozdělen na menší a větší počet polygonů.

a) *Low-Poly* – První metoda

Tyto modely byly lepší, co se týče výkonu, ale byly velice nevhodné pro malování. V místech zakřivení nebylo možné dosáhnout hladkého přechodu.

Lepších výsledků dosahovaly modely při použití druhého algoritmu.

b) *Low-Poly* – Druhá metoda

Zde bylo zaoblení mnohem lepší. Objekty kopírovaly ve většině případů křivku tak, jak měly, což však bylo způsobeno zejména tím, že jich bylo ve scéně velké množství. Tím ovšem naproti tomu rapidně poklesl výkon.

c) *High-Poly* – První metoda

Dalším krokem bylo zjistit, jak metoda funguje při vyšším počtu polygonů. Pro tento algoritmus byly nejvhodnější modely s větší délkou. Bylo to proto, že algoritmus v místě zakřivení měl další části modelu, se kterými mohl pracovat. Nejvhodnější z hlediska celkové kvality se jevil model s délkou 50 cm.

d) *High-Poly* – Druhá metoda

Pro tento způsob se nehodil žádný z modelů s vyšším počtem polygonů. Síť polygonů byla velmi hustá a ani poté, co byla zvětšena délka mezi jednotlivými modely, aby jich nebylo na jednom úseku tolik, se nepodařilo zvýšit FPS. Vzhled křivky byl podobný jako při prvním algoritmu, ale výkon byl mnohonásobně nižší.

Další krok bylo omezení vytváření *Spline Pointů* na jednom místě, které bylo nutné z následujících důvodů. Pokud uživatel držel ovladač bez pohybu na jednom místě, vytvářely se v tomto prostoru přebytečné body, čímž docházelo ke vzájemnému překrývání *Spline Meshů*, které tak byly buď skryty, nebo vytvářely shluky divných obrazců. Navíc i objekty, které byly skryté, ve scéně fakticky existovaly, byly proto zahrnuté do veškerých výpočtů a tím zbytečně zatěžovaly aplikaci.

Kroky vedoucí k tomuto omezení byly realizovány hned na začátku malovací funkce. Dříve, než byl vytvořen *Spline Point* na pozici ovladače, funkce zjistila pozici posledního *Spline Pointu*. Pokud vektorová vzdálenost tohoto posledního bodu a bodu, který se měl nově vytvořit, nebyla alespoň dva centimetry, nevytvořil se žádný bod. Toto jednoduché omezení nemělo téměř žádný negativní vliv na vzhled malované čáry, naproti tomu velmi příznivý vliv na výkon.

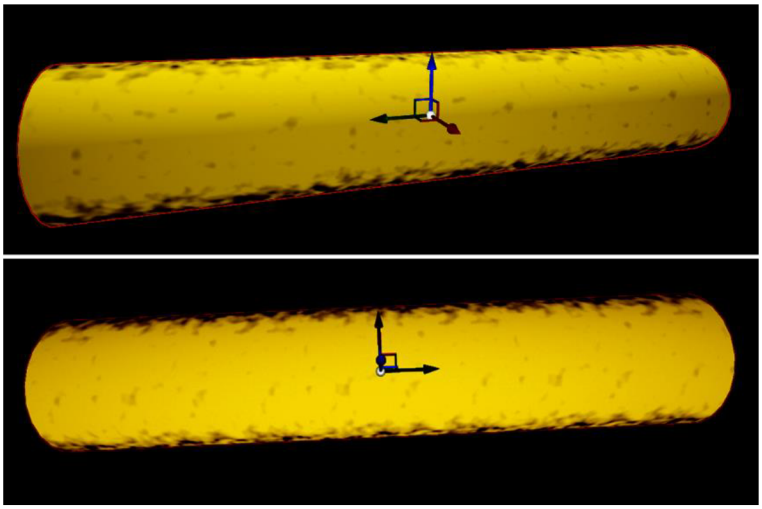
Poté následovaly drobné úpravy třídy, které však přispěly ke zlepšení výkonu pouze minimálně. U *Spline Meshů* byly vypnuto vykreslování stínů, které se při prvním testování ukázaly jako zbytečné.

Reálný vzhled malované čáry

Pro vytvoření reálně působící čáry byl vyvinut materiál, který měl simulovat vlastnosti klasické čáry vytvořené malířským štětcem [3]. Byly vytvořeny materiály vzhledem odpovídající použití voskovky a tempery.

Pro tvorbu materiálů bylo použito několik textur a texturových masek, které měly tento efekt vytvořit, včetně napodobenin nedostatků a vad obvyklých v reálných malbách. Například pomocí funkce *Noise* byly v malovaných čarách ponechány drobné mezery, které vznikají při tradiční malbě nerovnoměrným pokrytím barvou. Jiná nerovnoměrnost, tentokrát ve formě hrbolů, zase vzniká při malování temperou nebo olejem v místech, kde jsou nanesené barvy ve větší vrstvě [1]. Tento efekt byl vytvořen úpravou parametru *World Displacement*, který modifikuje fyzickou strukturu 3D modelu, na který je materiál aplikován.

Výše uvedené postupy však řeší problém jen částečně, protože 3D čára, která je ve své podstatě válec, navození tohoto efektu neumožňuje. Pokud by byla použita textura, která simuluje poškozený nebo jinak modifikovaný okraj, tento by se sice promítl v čáře, ale pouze při pohledu z jednoho směru, a proto by požadovaný efekt nenastal. K vyřešení tohoto problému proto byla do materiálu přidána další funkce *AxisAlignedFresnel*, která jej je schopna upravit podle pozice kamery. Výsledkem je vytvoření pomyslného konce čáry, který je vidět vždy, nezávisle na místě, kde uživatel stojí (Obrázek 16). Toto provedení je sice lepší než předchozí, ale bohužel ani takto upravená čára nepřipomíná většině uživatelů čáru skutečnou.



Obrázek 16 - Funkce AxisAlignedFresnel, pohled z různých směrů

Hlavním důvodem je zřejmě u uživatelů silně zafixovaný klasický vzhled 2D čáry a tento nový styl zobrazení jim pak připadá nepřirozený. Z toho důvodu nebyl nakonec doplňující materiál při tvorbě čáry aplikován.

4.3 Malířská paleta

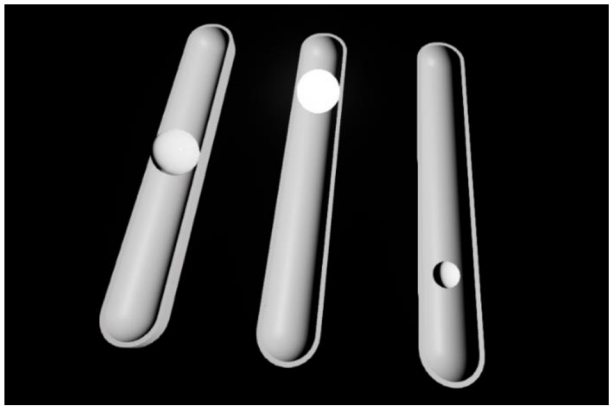
Uživatelům z oblasti malířského umění byla položena v dotazníku i otázka zaměřená na vzhled malířských nástrojů. Více než 60 % dotazovaných odpovědělo (Obrázek 24), že by ve virtuální realitě uvítali v rámci možností zachování pocitu spojeného s používáním skutečných nástrojů.

Varianta vybraná z více návrhů má vzhled malířské palety. Její první, do současnosti průběžně doplňovaná verze, obsahovala pouze rozmístěné barvy, posuvníky pro nastavení jejich parametrů a mističku s vodou pro případné odstraňování chyb. Touto paletou byl nahrazen jeden z ovladačů, přičemž uživatel měl pocit, že drží paletu v ruce. Druhým, malovacím ovladačem, uživatel vybíral z palety nástrojů. K němu byla připojena komponenta *RayTrace*, která při každém tiku vystřelila neviditelný paprsek a zkontrolovala, jestli nezasáhla objekt *PaletaActor*. Zásah znamenal podnět k výběru z nabídky malířské palety, přičemž každý objekt na paletě byl vytvořen samostatnou třídou a měl své chování.

Výběr barvy a nastavení dalších parametrů

Pro výběr barvy byla vytvořena třída *ColorSelector*. Jedná se o plochu pokrytou barvami. Každý bod na ploše vrací vypočítanou barvu z HSV barevného modelu. Pozice X určuje parametr *Hue* (H). Pozice Y určuje parametry *Saturation* (S) a *Value* (V) tak, že S je hodnota od nuly do jedné pro horní polovinu plochy a V je hodnota od jedné do nuly pro dolní polovinu plochy. Kombinací těchto parametrů vznikne barva.

Pro barvu, kterou si uživatel vybere, je možné nastavit další vlastnosti. Jsou to lesk, tvrdost a svítivost. Tyto parametry jsou upravovány pomocí posuvníků, které fungují na podobném principu jako vybírání barvy. Pokud je uživatel použije, táhne s nimi v předem daném směru, a tak nastavuje hodnotu parametru. Při úpravě hodnoty se mění vlastnosti kuličky posuvníku, které tak uživatele informují o provedených změnách (Obrázek 17).



Obrázek 17 - Posuvníky pro nastavení parametrů (lesk, svítivost, velikost)

Mazání objektů

Jednou z výhod elektronické aplikace je relativně snadná možnost odstraňování chyb. Konkrétně v případě malování není problém nepovedené části smazat. Nežádoucí čára může vzniknout z více důvodů. Uživatel namaloval čáru, která mu nevyhovuje například kvůli velikosti, barvě nebo umístění. Nežádoucí čáry vznikají i v důsledku nepřesné práce s ovladačem, například při jeho nedostatečném přiblížení k paletě či menu při výběru.

Mazací objekt byl v první verzi koncipován jako mistička s vodou. Po prvním testování byl však vyměněn za model gumy, protože většina testovaných uživatelů nepoznala, že se jedná o mističku s vodou k odstraňování chyb.

Pokud uživatel nyní vybere gumu, model ovladače, kterým maluje, se zneviditelní a na jeho pozici se vytvoří *ActorCleaner* mající vzhled klasické gumy. Ta funguje při držení akčního tlačítka tak, že odstraní části čar, kterých se dotkne. V průběhu vývoje byla funkce rozšířena o mazání předmětů.

V původní verzi bylo snahou mazat objekty napřímo. Při mazání *SplineMeshů* to nebylo z neznámého důvodu možné. Stejně tak nebylo možné mazat předměty, které uživatel držel a zároveň chtěl vymazat, kvůli neexistenci následných referencí.

ActorCleaner nyní funguje tak, že při každém použití gumy na objekt se pokusí o jeho přetypování na čáru (*SplineMesh)* nebo na předmět (*ItemToGrab*). Po úspěšném přetypování objektu zavolá jeho funkci *Destroyer*. Ta se postará o správné odstranění.

Pro vymazání *SplineMeshe* se v objektu *DrawSpline* zavolá funkce *Destroyer,* kde vstupním parametrem je konkrétní *SplineMeshe*. Ta poté porovná vstup se všemi *SplineMeshy* a požadovaný vymaže. U třídy *ItemToGrab* funkce *Destroyer* pouze označí, že předmět má být smazán a poté, co jej uživatel upustí, se teprve smaže.

Pomocí gumy je možné také čistit mističky s oblíbenými barvami.

Oblíbené barvy

Několik etap testů přineslo poznatek, že vybrat přesně stejnou barvu vícekrát po sobě bývá ve většině případů obtížné. K řešení tohoto problému byla dodatečně vytvořena a následně na paletu přidána třída pro uchování sortimentu oblíbených barev. Funguje tak, že si uživatel vybere barvu, nastaví její vlastnosti pomocí posuvníků a může využít jeden ze šesti slotů. Pro docílení pocitu reality uživatele mají tyto sloty pro uložení barvy vzhled malých mističek s namíchanou barvou, a aby působily přirozeně, jsou vytvořené v několika alternativách.

Třída DrawingHand

V první fázi řešení byl jeden ovladač určený pro malování a druhý představoval výběrovou paletu. Oba tyto prvky byly připnuty k třídě *Pawn* (znázorňující postavu uživatele), a také jejich logika byla v této třídě naprogramována. Po provedení změny, umožňující malování oběma ovladači zároveň, nebylo již toto řešení vhodné, protože nedostatečně informovalo o tom, jestli danou akci vyvolal levý nebo pravý ovladač. Stejně tomu bylo i v případech, kdy se mělo aktivovat nějaké omezení (například pokud je levý ovladač v mazacím módu, pravý ovladač může nadále malovat). Vše muselo být řešeno proměnnými, které uchovávaly informace o tom, jestli se jedná o levý nebo pravý ovladač či akci. Mnoho proměnných bylo duplicitních, a proto i jejich editace za běhu byla výrazně složitější. Nakonec to došlo do fáze, kdy jedno nastavení nepříznivě ovlivňovalo nastavení jiné. Proto byla vytvořena samostatná třída *DrawingHand*, do které byla přesunuta veškerá logika jednoho ovladače, přičemž nese informaci, ke kterému ovladači je konkrétně přiřazena. Tato třída si vše zajišťuje sama a není ničím nepříznivě ovlivňována. Navíc obsahuje komponentu *BoxCollision* a pokud je druhý ovladač v tomto prostoru, vypne se jeho malovací funkce, aby mohlo dojít k nastavení vlastností barvy a uživatel přitom nevytvářel nechtěné čáry, protože výběr na paletě a malování se provádí stiskem stejného tlačítka na ovladači. Po dalších testech byl přidán ještě 3D model kvádru, který znázorňuje oblast pro výběr na paletě. Ten rozsvícením a zhasínáním informuje uživatele o změnách funkce druhého ovladače z malovací na vybírací a naopak.

4.4 Prostředí

Původní návrh předpokládal možnosti malování ve třech typech prostředí. Prvním bylo jednobarevné pozadí, dalším přednastavená fotografie a třetí možnost přepnutí do rozšířené reality.

Výběr pozadí

Jako základní pozadí byla vybrána jednotná černá barva a jedna fotografie z 360° kamery. Po prvním testování se ukázalo, že by část uživatelů uvítala větší výběr možností. Proto byla přidána bílá barva a další dvě fotografie. Kromě toho bylo nově doplněno proměnlivé pozadí s oblohou a modifikovatelnou denní dobou, kterou si uživatel může nastavit v hlavním menu. Tato změna času byla vytvořena jako rozšíření vestavěné třídy *SkySphere*, která je poskytována enginem. Původně ovlivňovala i směr vrhaných stínu, ale jak bylo zmíněno dříve, stíny byly nakonec vypnuty.

Rozšířená realita

K přenosu obrazu reálného světa do aplikace, byl vybrán plugin využívající knihovnu *OpenCV* s názvem *UE4Webcamera*[15]. Ten zajišťuje přenos snímků z připojené kamery do aplikace, kde jsou jednotlivé snímky uloženy ve formě textury.

Pro zobrazování rozšířené reality byl vytvořen 3D model plochy a umístěn přímo před uživatele do aplikace tak, aby nebylo vidět nic jiného z virtuálního prostředí. Tato plocha byla připevněna na komponentu *Camera*, a uživatel tak při každém pohybu hlavy současně pohyboval i touto plochou. Na plochu poté byly promítány obrazy získané z kamery.

Při provozu aplikace se však objevilo několik problémů. Prvním z nich byl náhlý pokles výkonu, protože funkce na změnu textury se spouští při každém snímku kamery.

Problém byl také v samotném zobrazování, který byl způsobený tím, že kamera, která snímala vnější prostředí, měla jiné rozlišení, než bylo rozlišení brýlí. Pokud byla plocha dál od uživatele, což bylo nutné pro zobrazení objektů ve správně velikosti, objevily se současně i prvky z virtuálního prostředí po okrajích zobrazovací plochy. V opačném případě, při umístění plochy blíže k uživateli, byly zase skutečné předměty zobrazené na ploše větší než ve skutečnosti. Tento problém by však byl pravděpodobně řešitelný použitím kamery s jiným rozlišením.

Tento způsob zobrazování rozšířené reality nakonec nebyl použit a byl nahrazen lepším, mixovanou realitou, která pracuje s informacemi o hloubce obrazu.

Mixovaná realita

Předchozí návrh by aplikaci určitě oživil. Nejednalo by se jen o malování ve virtuálním prostředí, ale člověk by viděl do místnosti. Maloval by ve známém prostředí a aplikace by proto byla mnohem zajímavější. Také by se tím výrazně lišila od již existujících řešení, stále by to ale bylo pouhé zobrazení místnosti na pozadí. Interakce se skutečným prostředím by byla nulová.

Tento nedostatek lze ovšem odstranit využitím zařízení, které umí vytvářet hloubku obrazu. Pro toto řešení byla vybraná stereoskopická ZED kamera. Pro Unreal Engine 4 existuje plugin[16], který zpracovává získaný obraz a hloubku z této kamery formou vytvoření inicializační komponenty, která upraví pozici kamery a připraví zobrazovací plochu. Plugin sám už dokáže pracovat s hloubkou, proto nebylo potřeba již dále řešit zakrývání virtuálních objektů v případě kolize v reálném světě.

Protože původní řešení nebylo možné použít pro nově vytvořenou mixovanou realitu, bylo nutné provést úpravy v již existující třídě *Pawn*, která byla vytvořena pro realitu virtuální,

[15] Dostupné z: https://github.com/Temaran/UE4Webcamera
[16] Dostupné z: https://github.com/stereolabs/zed-unreal-plugin

a ve třídě *DrawingHand*, představující sady malířských komponent. Plugin vytváří vlastní *Pawn* (z důvodu správného nastavení kamery) a ani komponenty pro odchytávání pozice ovladačů nepracují stejně jako ty pro virtuální realitu. Proto bylo nutné třídu *DrawingHand* přepsat tak, aby fungovala nezávisle na tom, jestli bude připojena k *Pawnu* virtuální nebo mixované reality. Následně byla vytvořena ještě další třída *ARHand*, která obaluje třídu *DrawingHand* a zastupuje tak řídící *Pawn*. Tato třída *ARHand* byla umístěna do scény pro levý i pravý ovladač a není přímo připojena k *Pawnu* mixované reality. Protože neexistuje mezi *Pawnem* a třídou *ARHand* přímá reference není možná ani jejich vzájemná přímá komunikace. To však pro tuto konkrétní aplikaci není důležité, protože sada malířských nástrojů je upravena tak, aby fungovaly autonomně.

Další úpravu bylo nutné udělat ve třídě *MainMenu*. Pokud potřebuje uživatel vyvolat menu, třída spravující menu se nejdříve dotáže na pozici kamery. Protože ale v mixované realitě kamera funguje jinak, musela být do třídy *MainMenu* přidána úprava spočívající v tom, že před dotazem na pozici kamery se nejdřív zeptá, o jaký typ *Pawnu* se jedná a podle toho si upraví souřadnice vyvolaného hlavního menu.

4.5 Hlavní menu

Hlavní menu se vyvolá stiskem tlačítka na ovladači. Umístění menu se vypočítává z umístění kamery. Pokud se například uživatel přikrčí k zemi a vyvolá menu, funkce pro výpočet pozice nastaví menu přímo před uživatele, který tak nemusí pro následný výběr měnit svoji polohu.

Viditelná část hlavního menu je vytvořena komponentou *Widget*. Pro komunikaci s *Widgetem* slouží prvek emulující myš s názvem *WidgetInteractor* [2].

Od prvního návrhu přibylo několik dalších prvků a menu je nyní rozděleno na části seskupené podle funkčnosti (Obrázek 30).

První okno je hlavní menu, kde jsou uloženy všechny prvky a přesměrování na podokna.

První kategorie obsahuje tlačítka pro práci s aplikací. Jsou to tlačítka pro uložení a načtení výtvoru a restart aplikace. Po stisknutí tlačítka uložení/načtení (Save/Load) se otevře podokno, kde jsou sloty uložených pozic. Při stisknutí tlačítka slotu v ukládacím podokně se vytvoří fotografovací komponenta na vypočítané pozici (zde je opět funkce, která pozici přepočítá podle použitého *Pawnu*), obrázek uloží do složky v počítači a nastaví ho jako pozadí tlačítka slotu. Stejný obrázek je poté uložen i na tlačítko v podokně „Load“.

Ve druhé kategorii hlavního menu jsou přepínače týkající se vzhledu. První přepínač slouží k přepínání mezi virtuální a mixovanou realitou. Po jeho aktivaci se aplikace přepne do nového prostředí, které je připravené pro mixovanou realitu a pokusí se spustit ZED kameru.

Pokud z nějakého důvodu není možné kameru načíst (kamera není připojena, nedostačující grafická paměť, slabý signál USB…) uživateli je tento problém oznámen a aplikace se přepne zpět do prostředí virtuální reality. Druhý přepínač slouží k výběru modelu ovladačů. Z dotazníku vyplývá, že by uživatelé uvítali pocit simulující vzhled skutečných malířských nástrojů. Ne všem ale vyhovovalo, že vidí štětce, zatímco v ruce drží ovladače, a z toho důvodu byla do aplikace přidána zmíněná možnost výběru. Pokud je přepínač vypnutý, uživatel v aplikaci vidí ovladače odpovídající těm skutečným. Pokud jej zapne, modely ovladačů se změní na štětce.

Další samostatnou kategorií v hlavním menu je jediné tlačítko „Items“. Pomocí toho se uživatel dostane do podokna, které se stará o vytváření předmětů třídy *ItemToGrab*. Na výběr je několik základních geometrických tvarů a figurína. Po stisku tlačítka se vybraný předmět vytvoří vedle hlavního menu. Podokno nabízí i dva přepínače. Jeden z nich povoluje přesouvání předmětu ve scéně. Pokud je aktivní, může uživatel vytvořený předmět uchopit a přesouvat. Pokud jej uchopí oběma ovladači zároveň, mění jejich oddalováním a přibližováním velikost předmětu. Druhý přepínač slouží k povolení mazání předmětů gumou. Ten na jedné straně umožňuje vymazání přebytečných předmětů a na straně druhé zamezuje nechtěné náhodné odstranění předmětů v procesu malby.

Poslední kategorií v menu je změna pozadí. Tato funkce je aktivní pouze ve virtuální realitě a funguje pomocí dvou posuvníků, jejichž princip byl již popsán dříve.

5 Testování a vyhodnocení

V poslední části práce jsou uvedeny praktiky využité během testování. Dále jsou zde popsány odhalené chyby, které se nepodařilo ve výsledném řešení odstranit a také směr, kterým by bylo možné implementovanou aplikaci dále rozvíjet.

5.1 Způsob testování

Během vývoje aplikace proběhlo několik cyklů testování. To je důležité nejen pro odhalení chyb ve funkci programu, ale zároveň slouží jako zpětná vazba od uživatele, pro kterého mohou být některé funkce nesrozumitelné a nelogické, zatímco pro vývojáře, který s aplikací pracuje se zdá být vše v pořádku.

Výběr uživatelů

Pro testování byly vybíráni uživatelé spadající do jedné z třech kategorií. První skupinou byli uživatelé pohybující se v oblasti umění. Díky nim bylo možné získat zpětnou vazbu týkající se malířských funkcí a nástrojů.

Druhou skupinou byli uživatelé, kteří se nevěnují malování, ale mají zkušenosti s jinými aplikacemi virtuální reality. Jejich připomínky pomáhaly vylepšit řešení tak, aby chování odpovídalo tomu, na co jsou zvyklí z jiných aplikací.

Poslední skupinou uživatelů byli ti, kteří neměli žádné zkušenosti s virtuální realitou. Toto testování je náročnější, ale skryté problémy se odhalují mnohem snáz, protože díky technické nezručnosti uživatelů je mnohem větší šance problémy odhalit.

Funkce

Jedna část testů byla určena pro získání zpětné vazby k funkcím aplikace. Na základě těchto testů byly odhaleny nedostatky již implementovaných prvků a zároveň vznikly nové, které uživateli chyběly. Příkladem mohou být mističky oblíbených barev.

Vzhled

Další z testovaných oblastí byl vzhled jednotlivých částí aplikace. Položené dotazy se týkaly malířských nástrojů, výsledné malby a prostředí, ve kterém se uživatel pohyboval. Jak již bylo zmíněno dříve, uživatelé například nepoznali mističku s vodou, a proto je ani nenapadlo, že by mohla sloužit k mazání.

Zanesené chyby

Pro účel testování bylo umístěno do aplikace několik chyb, pro kontrolu, jestli si jich uživatel všimne, budou mu vadit při práci a bude k nim mít své připomínky. Chyby byly zaneseny různě při jednotlivých testech do grafického zpracování (splývající texty hlavního menu), do ovládání (nevhodně umístěné funkce na tlačítkách ovladače) a do komponent aplikace (komponenta nedělala, co by se od ní na první pohled očekávalo). Ve většině případů si chyb uživatelé všimli a zároveň je to donutilo více sledovat detaily.

Zpětná vazba

Zpětná vazba od uživatelů byla velice přínosná. Nejen, že bylo odhaleno několik chyb a nedostatků, které byly opraveny, ale zároveň měli uživatelé mnoho vlastních nápadů, které byly buď přímo implementovány, nebo popsány v možných rozšířeních.

5.2 Známé problémy

Níže jsou popsány známé problémy, na jejichž odstranění se při vývoji pracovalo, ale zatím se je nepodařilo zcela vyřešit. Dále je zde popsán možný směr, kterým by se aplikace mohla rozvíjet.

ZED kamera a mixovaná realita

Aby bylo možné využít ZED kameru, bylo nutné stáhnout a přeložit verzi Unreal Enginu 4, která byla modifikovaná firmou Stereolabs právě pro tyto účely. Tato modifikace však obsahuje nějaké chyby, které jsou pravděpodobně důvodem občasného pádu vytvořené aplikace, protože tento problém se nevyskytl do doby, než začala být používána tato modifikovaná verze enginu.

Dalším problémem jsou rozdíly v rozlišení ZED kamery a brýlí, které způsobují částečné zkreslení vnějšího prostředí a vznik černých okrajů kolem zobrazovací plochy.

Posledním problémem v režimu mixované reality je neustálé přepočítávání pozice kamery v prostoru. Důsledkem je, že zobrazené virtuální objekty se někdy náhle posunou i o několik centimetrů. To způsobuje problémy jednak přímo při malování, ale také při výběru ať už na malířské paletě nebo v hlavním menu. Dalším projevem této chyby je nesprávně provedený přepočet pozice ovladačů při rychlejším pohybu, který se sice ustálí, ale až po několika okamžicích.

Nepřesnost při výběru na malířské paletě

Výběr na paletě je prováděn neviditelným laserem objektu *RayTrace*. Uživatel míří na posuvníky nebo výběrovou paletu barev. Tyto komponenty fungují tak, že pokud je na ně ukázáno, začnou každý tik kontrolovat pozici a podle toho vrací vypočítanou hodnotu. Jakmile uživatel přestane laserem na tuto komponentu ukazovat, výpočet se ukončí. To způsobuje občas problém při výběru krajních hodnot, pro jejichž získání je nutné postupovat velmi pomalu a opatrně. Stejně tak může být problém při prudkém pohybu, kdy objekt *RayTrace* nestíhá posílat signál výběrovým komponentám.

5.3 Možná rozšíření

Vytvořenou aplikaci lze rozšiřovat několika směry. Nynější fáze vývoje prezentuje řadu možností, které je možné realizovat pomocí enginu a virtuální reality.

Jedním z možných směrů, kterým by se mohla aplikace dále ubírat je vylepšení uměleckého směru, například přidáním více druhů štětců, přidáním možností přepínat mezi několika typy malování, nebo možnost malby efektovým štětcem. Určité rozšíření by mohlo být ve vzájemném ovlivňování jednotlivých barev a jejich mixování.

Další oblastí pro rozšíření by mohl být import některých částí – vlastní pozadí, vlastní textury, možná i vlastní modely.

Možnosti jsou i v nových režimech prohlížení, umožňujících uživateli zvětšovat či zmenšovat celé dílo jako celek.

Opačným směrem by mohlo být přepracování této aplikace z umělecké na technickou aplikaci typu CAD. Zde by mohl uživatel kreslit čáry, které by se přichytávaly k mřížce, čímž by se zvýšila jejich přesnost až na několik setin milimetrů. Zde by také mohly být využity další nástroje typu pravítko, kružítko, možnosti výplně atd.

Součástí obou verzí (malířské i technické) by mohla být funkce exportu, která by vytvořila STL model, který by bylo možné vytisknout na 3D tiskárně. Na podobném principu by mohla být řešena také možnost sdílení výtvorů na sociálních sítích.

Obě aplikace by také mohly být rozšířeny o kooperativní režim, který by umožnil například více autorům malovat v jednom prostoru a zlepšit tak jejich zážitek z virtuální reality. V technickém směru by možnost využívat aplikaci více uživateli zároveň z různých míst ve světě umožnila v reálném čase konzultovat a vytvářet jejich společný produkt.

Toto je pouze několik návrhů rozšíření, o jejichž realizaci by bylo možná vhodné v budoucnu uvažovat. Je však také možné, a třeba i jednodušší, vytvořit aplikace zcela nové namísto rozšiřování této stávající.

6 Závěr

Cílem této práce bylo navrhnout a následně vytvořit aplikaci, kterou by bylo možné díky speciálnímu zařízení spouštět v režimu virtuální a rozšířené reality. Tato aplikace měla obsahovat sadu nástrojů, které by umožňovali uživateli malovat v 3D prostoru a zároveň by aplikace připomínala skutečný proces malování.

Před samotným řešením byl nejdříve v úvodu této bakalářské práce vylíčen průřez vývojem virtuální reality od jejího počátku po současnost. Po seznámení s prvotním hardwarem následovala kapitola, ve které byly vybráni zástupci zařízení umožňující spouštět aplikace nejen virtuální, ale také rozšířené a mixované reality. Tomuto rozdělení a detailnějšímu popisu se věnovala další podkapitola.

V implementaci aplikací, pro některý typ reality mnohdy pomáhají vývojářské softwary, které byly popsány dál.

Výsledné aplikace se mohou uplatnit v některém z oborů, které byly vylíčeny na konci úvodní části. Mnoho vytvořených aplikací slouží pouze pro zábavu, ale, uplatní se například i ve zdravotnictví. Také vytvořenou aplikaci je možné zařadit do kategorie zábavy.

Následující část práce se zaměřila na obecné postupy při návrhu řešení a způsoby, kterými by bylo možné docílit požadovaných výsledků. Nejdříve byly vyjmenovány požadavky na navrhovanou aplikaci.

Dále zde byly navrženy hardwarové a softwarové komponenty, které by mohly být využité při implementaci již zmíněných požadavků. Mezi tyto navrhované komponenty patřil například hardware pro odchytávání pozice uživatele nebo vývojářský software, který by bylo vhodné využít.

Jedna z kapitol se také věnovala možným způsobům zobrazení malovaných čar a popisu jejich výhod a nevýhod.

V poslední části byly vytvořeny grafické návrhy uživatelských rozhraní a funkce, které bylo nutné přiřadit k ovladači.

Kapitola vlastního řešení popisuje implementace navržených částí, vzniklých problémů a odkloněni od původních návrhů.

Výsledné řešení bylo vytvořeno v softwaru Unreal Engine 4, který byl velice nápomocný zejména proto, že se jedná o herní engine.

Jedním z důležitých implementovaných nástrojů byl ten, díky němuž bylo možné malovat. Což byla hlavní funkce aplikace. Bylo vytvořeno několik postupů, které nevyhovovaly a nebyly použity nebo musely být upraveny. Nakonec bylo vytvořeno řešení, které k malování využívá křivku a vytváření 3D modelů po její trase.

Další částí implementované aplikace, kterou je vhodné zmínit byl způsob přepínání mezi virtuální a rozšířenou realitou. Zde nastal odklon od původního plánu a místo rozšířené reality bylo řešení upraveno tak, aby mohl uživatel, díky stereoskopické kameře, pracovat ne s rozšířenou, ale s mixovanou realitou.

V poslední části této práce byly vylíčeny způsoby testování vytvořené aplikace a konzultace s uživateli. Jedna kapitola se věnuje známým chybám, na kterých se pracovalo, ale zatím se je nepodařilo odstranit a také možným směrům, kterými by se mohla vytvořená aplikace dále vyvíjet.

Zdroje

Literatura

[1] ALLBROOK, Colin. *Techniky malby*. Přeložila Světlana PAVLÍKOVÁ. Praha: Knižní klub, 2017. Universum (Knižní klub). ISBN 9788024257471.

[2] MCCAFFREY, Mitch. *Unreal engine VR cookbook: developing virtual reality with UE4*. 2017. Boston: Addison-Wesley, 2017. ISBN 978-0134649177.

[3] STANYER, Peter. *Vše o technikách kresby: [praktická příručka pro výtvarníky]*. Praha: Svojtka & Co., 2007. ISBN 9788073523831.

Elektronické zdroje

[4] Aspen Moviemap. *Michael Naimark Home Page* [online]. Dostupné z: http://www.naimark.net/projects/aspen.html

[5] Category:Games - Epic Wiki. [online]. Copyright © 2004. Dostupné z: https://wiki.unrealengine.com/Category:Games

[6] Forgotten genius: the man who made a working VR machine in 1957: A Gear VR for from the 1950s? | TechRadar. TechRadar | The source for tech buying advice | TechRadar [online]. Copyright ©. Dostupné z: https://www.techradar.com/news/wearables/forgotten-genius-the-man-who-made-a-working-vr-machine-in-1957-1318253/2

[7] Google Glass Help. *Google Help* [online]. Copyright ©2018 Google. Dostupné z: https://support.google.com/glass

[8] How It Works: Xbox Kinect. *Jameco Electronics - Electronic Components Distributor* [online]. Dostupné z: https://www.jameco.com/jameco/workshop/howitworks/xboxkinect.html

[9] Microsoft Store. *Microsoft Corporation* [online]. Dostupné z: https://www.microsoft.com/en-us/store/d/Microsoft-HoloLens-Commercial-Suite/944XGCF64Z5B

[10] Oculus Rift | Oculus. *Oculus* [online]. Copyright ©. Dostupné z: https://www.oculus.com/rift/

[11] Stereolabs. *Stereolabs* [online]. Copyright © 2018 Stereolabs. Dostupné z: https://www.stereolabs.com/

[12] SUTHERLAND, Ivan. *Head Mounted Three Dimensional Display*. 1968. Dostupné z: https://www.cise.ufl.edu/research/lok/teaching/ve-s07/papers/sutherland-headmount.pdf

[13] Unity. *Unity* [online]. Copyright © 2018 Unity Technologies. Dostupné z: https://unity3d.com/

[14] Unraveling The Enigma Of Nintendo's Virtual Boy, 20 Years Later. *Fast Company | The future of business* [online]. Copyright ©. Dostupné z: https://www.fastcompany.com/3050016/unraveling-the-enigma-of-nintendos-virtual-boy-20-years-later

[15] VIVE™ | Discover Virtual Reality Beyond Imagination. [online]. Copyright © 2011. Dostupné z: https://www.vive.com/eu/

[16] VR/AR/MR, what's the difference? | Virtual reality | Foundry. *Imagination Engineered | Foundry* [online]. Dostupné z: https://www.foundry.com/industries/virtual-reality/vr-mr-ar-confused

[17] What is Virtual Reality? VR Definition and Examples | Marxent. *Marxent: The Leader in ARKit apps and Virtual Reality for Furniture* [online]. Copyright ©. Dostupné z: https://www.marxentlabs.com/what-is-virtual-reality/

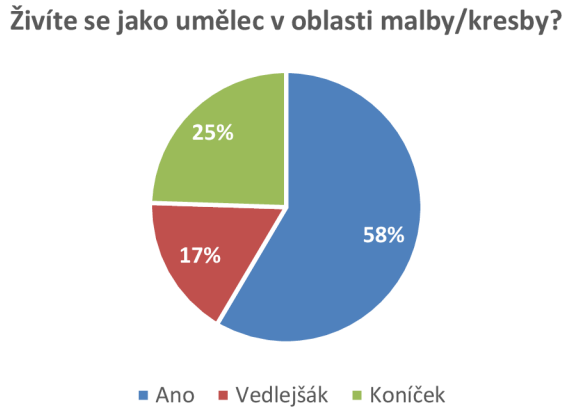
[18] *Wikipedia: The Free Encyclopedia: Unreal Engine* [online]. © 2017. Dostupné z: https://en.wikipedia.org/wiki/Unreal_Engine

Příloha A - Obsah přiloženého DVD

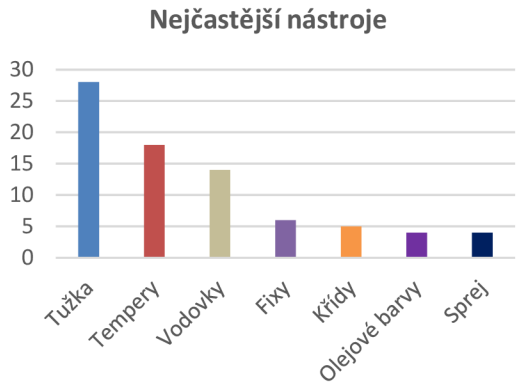
- *technicka_zprava.pdf* – Tato technická zpráva ve formátu PDF
- *technicka_zprava.docx* – Zdrojové texty této technické zprávy
- *zdrojove_soubory.zip* – Archiv se zdrojovými soubory aplikace
- *vysledna_aplikace.zip* – Archiv výsledné aplikace
- *reprezentativni_plakat.pdf* – Plakát ve formátu PDF
- *reprezentativni_video.mp4* – Video ukázka funkcí aplikace
- *ukazka_tvorby.mp4* – Video ukázka použití aplikace

Příloha B – Výsledky dotazníku

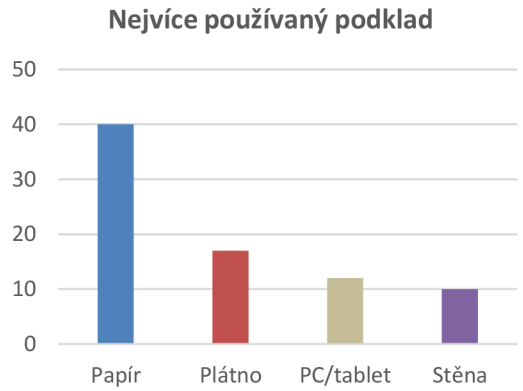
Před zahájením práce byl vytvořen dotazník a odeslán lidem zabývajícím se uměním v oblasti malby a kresby. Převážná většina odpovídajících byla ve věku od dvaceti do třiceti let a svému oboru se dotazovaní věnovali pět a více let.



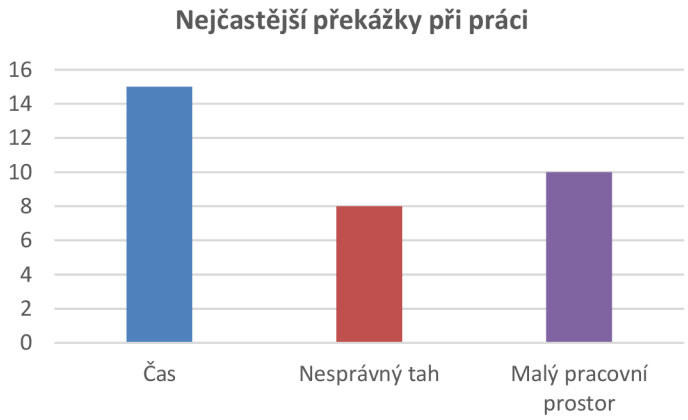
Obrázek 18 - Graf zobrazující, kolik dotazovaných se živí uměním



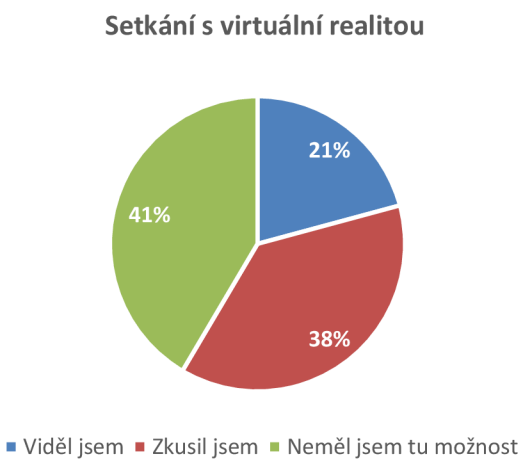
Obrázek 19 - Graf nejčastějších nástrojů



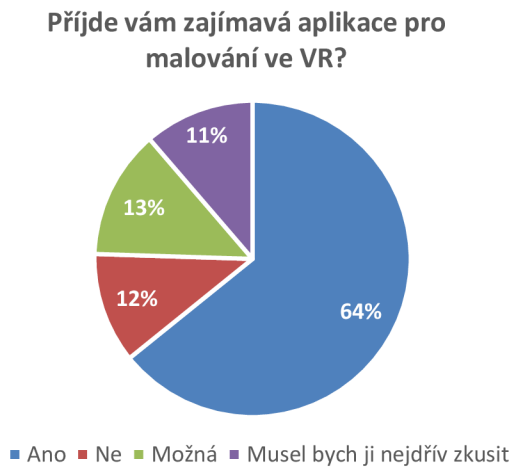
Obrázek 20 - Graf nejčastěji používaného podkladu



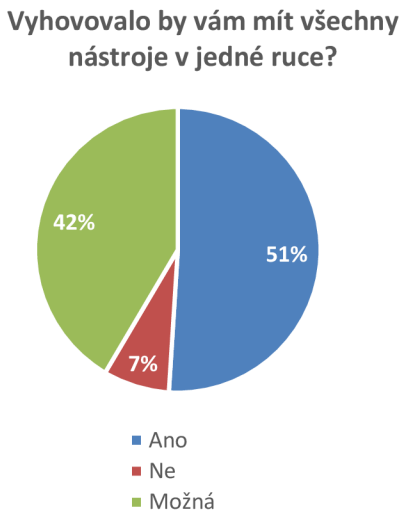
Obrázek 21 - Graf nejčastějších překážek při práci



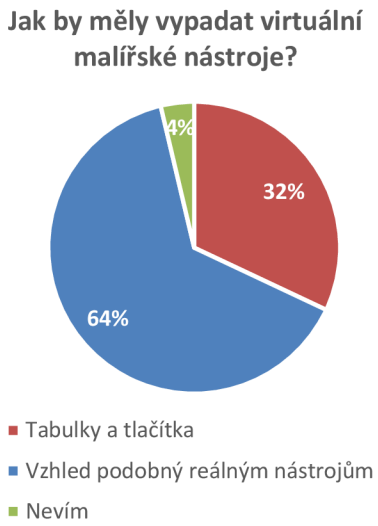
Obrázek 22 - Graf setkání s virtuální realitou



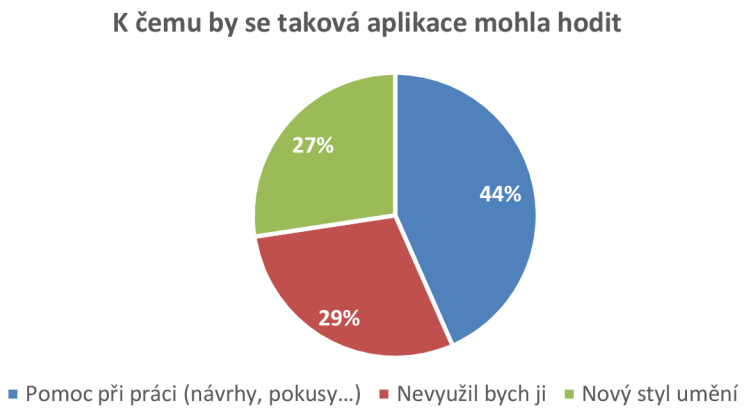
Obrázek 23 - Graf zajímavosti malířské aplikace



Obrázek 24 - Graf využití elektronických výhod



Obrázek 25 - Graf vzhledu nástrojů



Obrázek 26 - Graf možného využití aplikace

Příloha C – Návod k použití aplikace

V této příloze je popsáno ovládání aplikace a její funkce. V druhé části jsou informace k ovladači HTC Vive a funkce přiřazené tlačítkům.

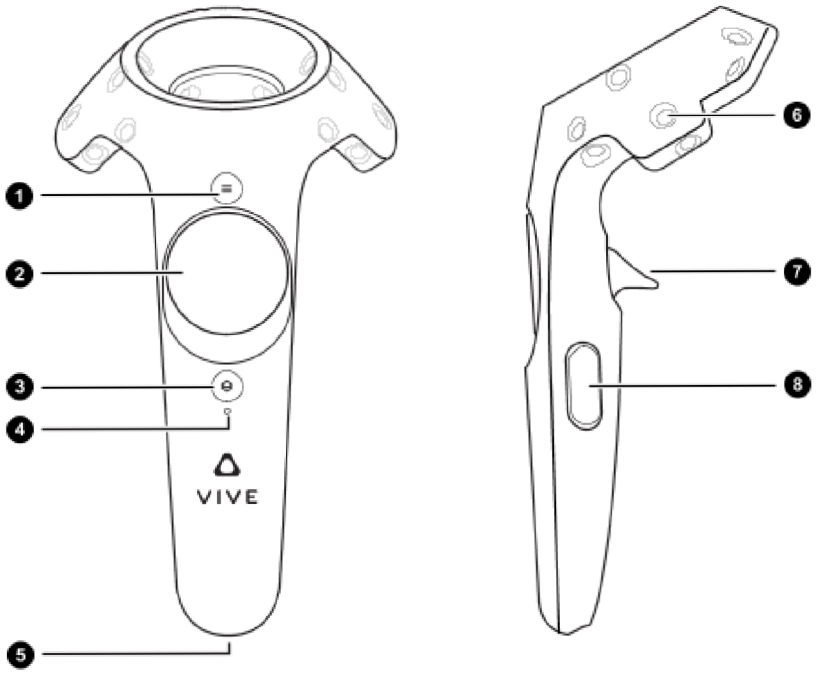
Popis aplikace

- Před spuštěním aplikace musí být headset položen na podlahu, jinak neproběhne správná inicializace výšky.
- Jako první se spustí mixovaná realita.
    - Pokud by z nějakého důvodu nastal problém v komunikaci se ZED kamerou, aplikace tento problém oznámí a přepne se do virtuální reality.
- Po spuštění aplikace má uživatel jeden malovací ovladač a jednu malířskou paletu.
    - Oba ovladače mají stejnou funkčnost – je možné mezi nimi přepínat a malovat oběma zároveň (tlačítko 1 pro přepnutí mezi paletou a ovladačem).
- Paleta obsahuje komponenty pro výběr barvy, nastavení vlastností barev, sloty pro uložení vybrané barvy, gumu na mazání a změnu velikosti.
    - Po výběru gumy se vytvoří její model, ale gumovat začne až stiskem tlačítka 7. Gumou je možné čistit pozice oblíbených barev. Guma se vypne po výběru barvy na paletě.
- Po přiblížení ovladače k aktivní malířské paletě se rozsvítí zelený kvádr indikující změnu z malování na výběr. Nastavení (barvy, vlastností…) se ukládá pro ovladač, kterým bylo vybíráno. Každá paleta/ovladač má svoje vlastní nastavení.
- Hlavní menu (tlačítko 8) obsahuje několik komponent, které je možné vybírat tlačítkem 7. Po přiblížení ovladače k menu se aktivuje výběrový laser, a ovladačem není možné malovat.
    - Podokna „Save/Load“ slouží k uložení/načtení výtvoru, pozadí a nastavení palet. Uložené pozice lze přenášet mezi VR/MR.
    - Přepínačem „Brush“ je možné vybírat mezi modelem štětce a ovladače.
    - Přepínačem „Mixed reality“ je možné spustit mixovanou realitu.
    - Podokno „Items“ slouží k vytváření předmětů. Obsahuje přepínače pro manipulaci s předměty a umožnění jejich mazání.
    - Dva posuvníky slouží ke změně pozadí, popř. ke změně denní doby (pro poslední pozadí). Změna pozadí je možná pouze ve virtuální realitě.
    - Tlačítko Reset vrátí vše do počátečního nastavení.
- Pokud je vytvořen předmět a povolena přepínačem jeho manipulace, je možné předmět uchopit a přemístit tlačítkem 7.
    - Pokud je předmět uchopen oběma ovladači zároveň, jejich přibližováním a oddalováním se mění velikost předmětu.

Popis jednotlivých tlačítek

Tlačítko 1 – Vyvolání/skrytí malířské palety
Tlačítko 2 – Chůze ve virtuální realitě při stisknutí v určitém směru
Tlačítko 7 – Malování v malovacím režimu, výběr na malířské paletě, výběr v hlavním menu
Tlačítko 8 – Vyvolání/skrytí hlavního menu

Tlačítko 3 – Systém
4 – Dioda, indikátor nabití a funkčnosti ovladače
5 – Nabíjecí konektor
6 – Snímané senzory



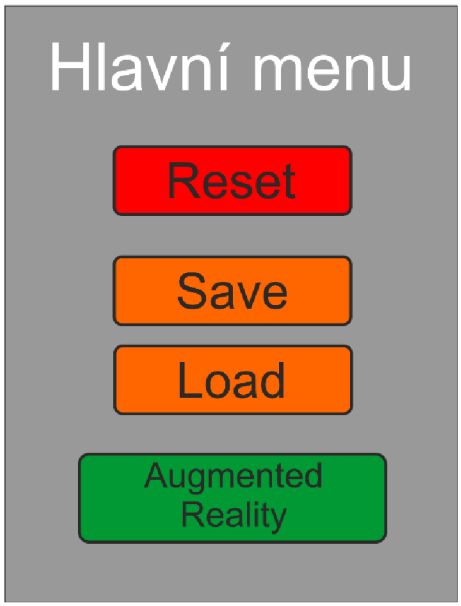
Obrázek 27 - Popis ovladače[17]

[17] https://www.vive.com/nz/support/vive/category_howto/about-the-controllers.html

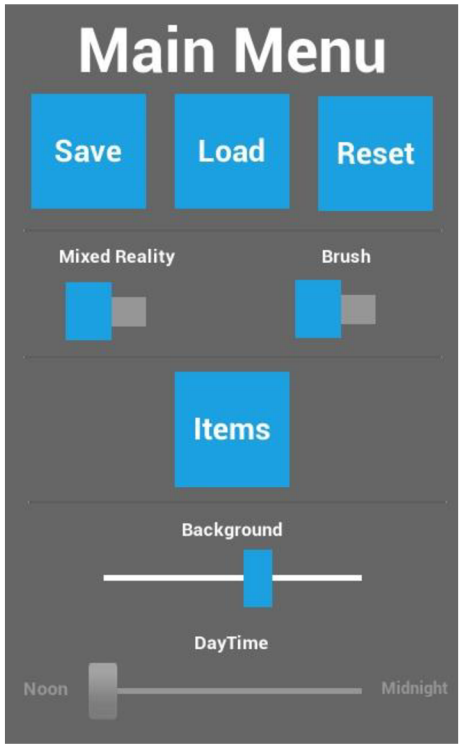
Příloha D – Návrh GUI

Obrázek 28 - Návrh vzhledu palety

Obrázek 29 - Finální vzhled palety



Obrázek 30 - Návrh vzhledu hlavního menu



Obrázek 31 - Finální vzhled hlavního menu