

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Zaznamenávání akcí ve virtuální realitě pro účely výuky

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Vítek POÓR**
Osobní číslo: **A19N0115P**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Počítačová grafika**
Téma práce: **Zaznamenávání akcí ve virtuální realitě pro účely výuky**
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Seznamte se se softwarem pro výuku ve virtuální realitě VR Classroom vyvíjeným na ZČU.
2. Implementujte do softwaru záznam a přehrávání prováděných akcí a zvukového výkladu. Záznamový subsystem navrhnete tak, aby umožňoval snadné doplnění další funkcionality do softwaru včetně záznamu akcí provedených s využitím doplněných funkcí.
3. Navrhnete metodiku pro (polo)automatické ověření funkčnosti celého systému a odstraňte všechny chyby, včetně těch které se netýkají nově zavedené funkcionality.
4. Přidejte do systému podporu zařízení platformy SteamVR a ověřte funkcionality a kompatibilitu.
5. Hotový systém důkladně zdokumentujte.

Rozsah diplomové práce: **doporuč. 50 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Doc. Ing. Libor Váša, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání diplomové práce: **10. září 2021**
Termín odevzdání diplomové práce: **19. května 2022**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

Doc. Ing. Přemysl Brada, MSc., Ph.D.
vedoucí katedry

V Plzni dne 11. října 2021

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 19. května 2022

Vítek Poór

Abstract

The thesis describe the VRClassroom software, which is being developed at the University of West Bohemia at the Department of Computer Science and Engineering. VRClassroom software is a tool to complement teaching where spatial imagination is required. Using virtual reality and real-time synchronization, it makes it easier for the student to understand the subject. This work describes the software as a whole. Adds new functionality, especially creating a record within a session. Extends software support with the SteamVR platform, which is used especially with Oculus Rift and HTC Vive devices. Finally, the work proposes a set of test scenarios for user testing of basic usage scenarios.

Abstrakt

Práce se věnuje popisu softwaru VRClassroom, který je vyvíjen na Západočeské Univerzitě na katedře informatiky a výpočetní techniky. VRClassroom slouží jako nástroj pro doplnění výuky, kde je zapotřebí prostorová představitivost. Pomocí virtuální reality a synchronizace v reálném čase usnadňuje studentovi pochopit danou látku. Práce popisuje software jako celek. Přidává novou funkcionalitu, zejména vytvoření záznamu v rámci sezení. Rozšiřuje podporu softwaru o platformu SteamVR, která se používá při práci zejména se zařízeními Oculus Rift a HTC Vive. V poslední řadě práce navrhuje sadu testovacích scénářů pro uživatelské testování základních scénářů použití.

Obsah

1	Úvod	8
2	Virtuální realita	10
2.1	Kategorie virtuální reality	10
2.2	Zařízení pro virtuální realitu	11
2.2.1	Oculus	14
2.2.2	HTC	16
3	Související práce	19
4	Software VRClassroom	20
4.1	Architektura	20
4.2	Spustitelné programy	22
4.3	Uživatelské rozhraní	24
4.3.1	Přihlášení k serveru	26
4.3.2	Funkce klientské aplikace	27
4.4	Technické rozhraní	35
4.4.1	Sestavení	37
4.4.2	Sdílená knihovna VRClassroom.Common	39
4.4.3	Serverová aplikace	43
4.4.4	Klientské aplikace	45
5	Záznamový subsystém	49
5.1	Pořízení záznamu	49
5.2	Distribuce záznamu	52
5.3	Přehraní záznamu	53
5.4	Rozšíření subsystému	54
6	Integrace SteamVR	60
6.1	Přidání OpenVR do Unity	60
6.2	Objekt hráče	61
6.3	Sjednocení uživatelských vstupů	62
6.4	Práce s GUI	64
7	Ověřování funkčnosti celého systému	66
7.1	Prohlížeč testovacích případů	66
7.2	Testovací scénáře	68

8 Závěr	70
Literatura	72
9 Přílohy	76
9.1 Barevná krychle	76
9.2 Šablona testovacího scénáře	77

1 Úvod

V rámci studia aplikovaných věd, zejména počítačové grafiky či matematiky, je drtivá většina vykládané látky prezentována nějakou formou 2D vizualizace (obrázky, knihy, tabule, prezentace...). Děje se tak zejména z technických důvodů. Je potřeba určitá úroveň studentovi představivosti pro plnohodnotné pochopení dané látky, zejména pokud studuje nějaký prostorový jev.

Metody počítačové grafiky usnadňují prostorovou vizualizaci. Pomocí zobrazování v reálném čase (real-time rendering) je možné například zobrazit řadu problémů v interaktivním režimu, a tím výrazně obohatit výsledný vjem. Není však jednoduché takovýto vizualizátor vytvořit pro každý vizualizovaný jev tak, aby zůstal implementačně stejný. Pro standardní výuku by to bylo velmi časově náročné, proto se od těchto metod upouští. Vezměme například oblast matematiky, ve které existuje mnoho nástrojů pro vizualizaci grafů funkcí. Interakce s objekty je však omezena na pohyb myši. Pohled může být někdy, z důvodu 3D projekce, na monitoru zkreslený.

Tyto problémy překonává technologie virtuální reality (VR) a augmentované reality (AR). Technologie virtuální reality dovoluje zobrazovat prostorové objekty v přirozenějším vjemu, než je tomu v klasické zobrazovací metodě. Používá vlastní hardware, především speciální brýle (headset).

V rámci Západočeské Univerzity vznikl na fakultě aplikovaných věd software pro virtuální realitu (VRClassroom - viz Kapitola 4), který se používá při usnadnění vizualizace některých fenoménů v rámci kurzu zpracování polygonálních sítí (KIV/ZPOS). Jedná se o pomocný nástroj k probírané látce. Nástroj je nezávislý na daném kurzu a je použitelný i pro jiné kurzy. Obecně slouží jako interaktivní vizualizátor prostorových objektů v reálném čase pro více uživatelů. Software usnadňuje studentům pochopit specifické problémy jejich prostorovou vizualizací. Vizualizace je tvořena vyučujícím v reálném čase tak, jak kdyby problém vysvětloval na tabuli. Doposud měl software pouze několik základních funkcí. V rámci této práce se software výrazně rozšířil a byla vydána nová verze.

Následující kapitola představuje technologii virtuální reality. (2) Zaměřuje se na to, jak tato technologie funguje a jak je dostupná. V kapitole (3) se dozvídáme o dalších studiích, které byly v této oblasti zpracovány a vydány. Kapitola (4) dokumentuje existující řešení softwaru VRClassroom. Popisuje architekturu projektu, uživatelské použití, dostupnou funkcionalitu a programátorské řešení. Na to navazují kapitoly (5) a (6), které předsta-

vují nové funkcionality softwaru a popisují jejich integraci. V poslední řadě kapitola (7) řeší problém testování aplikace. Popisuje nově vzniklou aplikaci pro usnadnění uživatelského testování a představuje sadu uživatelských testovacích scénářů.

2 Virtuální realita

Díky rozvoji různých oblastí informatiky (počítačová grafika, umělá inteligence) a se sníženými náklady na chytré brýle (headset, head-mounted display - HMD) se virtuální (VR) a rozšířená realita (augmented - AR) stávají součástí našeho každodenního života. Odhaduje se, že velikost trhu s VR headsety v následujících letech poroste. Lidé nejspíš budou taková zařízení používat pravidelněji v různých denních rutinách, včetně zábavy, vzdělávání a školení. V současné době jsou k dispozici různá zařízení s širokou škálou technických možností. V této práci se zaměříme především na zařízení od firem Oculus a HTC. S použitím zařízení od těchto firem je dále používaný software vyvíjen a používán. [6]

Kvalita systému s virtuální realitou je určena vlastnostmi známými jako 3-I. Jsou to imerze (Immersion), interakce (Interaction) a představivost (Imagination). Imerze (také ponoření, přítomnost - presence) je pocit přítomnosti bytí jako součást ve virtuální scéně. Je to důsledek stimulace vizuálních, sluchových a haptických lidských smyslů systémem virtuální reality. Interakce je prostředek komunikace se systémem virtuální reality. Nejedná se o tradiční interakci člověka s počítačem (Human-Computer Interaction - HCI), která typicky používá 1D či 2D prostředky (myš, klávesnice, monitor). Jedná se o interakci prostřednictvím prostorově snímaných 3D prostředků (HMD, prostorově snímané ovladače). Představivost slouží k efektivnějšímu vyjádření komplexních myšlenek za pomoci komponent systému s virtuální realitou. Je účinnější než při použití tradičních 2D zobrazovacích metod. [2]

2.1 Kategorie virtuální reality

Systémy s virtuální realitou můžeme, z pohledu úrovně ponoření (imerze), rozdělit do 3 hlavních kategorií: neimerzivní (non-immersive), imerzivní (immersive) a z části imerzivní (semi-immersive). Úroveň ponoření ovlivňuje zejména typ rozhraní a používané komponenty.

Neimerzivní virtuální realita je typ virtuální reality, ve které probíhá komunikace s virtuálním prostředím obvykle prostřednictvím počítače či jiného výpočetního zařízení (herní konzole...). Uživatel může nějakým způsobem zasahovat do virtuálního světa, ovládá postavy nebo činnosti v rámci zážitku, ale virtuální prostředí s uživatelem přímo neinteraguje. Jako příklad uvádíme běžné počítačové hry, kdy hráč hraje za postavu ve virtuálním světě hry. Hráč komunikuje s virtuálním prostředím nepřímo skrze postavu,

za kterou hraje.

Z části imerzivní virtuální realita stojí na rozhraní neimerzivní a plně imerzivní virtuální reality. Pomocí obrazovky počítače nebo brýlí pro virtuální realitu se uživatel může pohybovat ve virtuálním prostředí. Kromě vizuálního zážitku nebude mít žádné fyzické vjemy, které by zážitek umocnily. Do této kategorie se řadí virtuální prohlídky či různé simulátory.

Na druhém konci od neimerzivní virtuální reality stojí imerzivní virtuální realita. Ta se snaží navodit realistický virtuální zážitek. V uživateli probudí pocit fyzické přítomnosti ve virtuálním světě. Uživatel má dojem, že události, které se odehrávají ve virtuálním světě, se dějí přímo jemu. Pro poskytnutí realistického virtuálního zážitku je vyžadováno speciální vybavení (HMD, ovladače, snímače prostředí či možné dodatečné detektory těla). Každé vybavení, které uživatel přímo používá, je vybaveno senzory pohybu, na které systém reaguje v reálném čase. Tím je uživateli poskytnut realistický virtuální zážitek. Zde si představme jakoukoliv hru odehrávající se ve virtuální realitě. [34]

2.2 Zařízení pro virtuální realitu

Jak již bylo zmíněno výše, na trhu je mnoho zařízení s virtuální realitou. Produkty nižší třídy většinou obsahují pouze brýle, které fungují jako schránka pro chytré mobilní zařízení. Neobsahují žádný dodatečný software ani hardware. Obvykle mají pouze dva displeje. Celková funkcionality a uživatelský zážitek závisí na možnostech použitého mobilního telefonu.

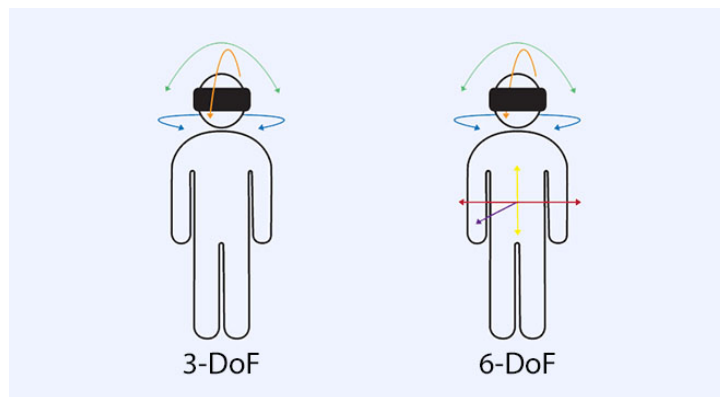
V dnešní době nejrozšířenějším zařízením pro plně imerzivní zážitek je head-mounted display (viz Obrázek 2.1). Pro toto zařízení jsou typické dva stereoskopické displeje a sledovací funkcionality. Uživatel vidí trojrozměrnou scénu a pohybuje se v ní podle polohy hlavy. Stereoskopické vidění zajišťují dvě virtuální kamery ve scéně. Každá kamera se zobrazuje na jeden displej (oko). Obvykle brýle navíc obsahují pro umožnění kvalitnějšího pohybu gyroskop, akcelerometr či magnetometr. [4]



Obrázek 2.1: Ukázka zařízení head-mounted display. [23]

Výsledné trojrozměrné zobrazení je dosaženo pomocí stereoskopie. Obecně se stereoskopie zabývá trojrozměrným zobrazováním, kdy se k dvojrozměrnému obrázku přidává hloubka jako třetí dimenze. Zjednodušeně stereoskopie funguje na stejném principu jako lidské oko, kdy se při pohledu na scénu pořídí dva obrázky z různých úhlů a pozic (vzdálenost určena roztečí očí). Tyto obrázky se poté promítají pro každé oko zvlášť. [8]

Zařízení disponují třemi nebo šesti stupni volnosti (Degrees of Freedom - DoF). Stupně volnosti označují počet způsobů, jak se může objekt pohybovat trojrozměrným prostorem. Existuje šest celkových stupňů volnosti, které popisují každý možný pohyb objektu. Tři způsoby rotace a translace. U rotace se jedná o sklon (pitch), stáčení (yaw) a naklánění (roll). U translace jde o pohyb vpřed nebo vzad, doleva i doprava a nahoru či dolů (Obrázek 2.2).



Obrázek 2.2: Rozdíl mezi třemi (vlevo) a šesti (vpravo) stupněmi volnosti. [3]

Další technologie, kterou je dobré na tomto místě zmínit, je CAVE (Cave Automatic Virtual Environment). Jedná se o místnost, v níž projektory pokrývají stěny místnosti stereoskopickým obrazem (viz Obrázek 2.3). Uživatel používá speciální synchronizované brýle s projektory, které střídají obrazy. Místnost doplňují reproduktory pro zvukovou kulisu. [4]



Obrázek 2.3: Ukázka technologie CAVE. [7]

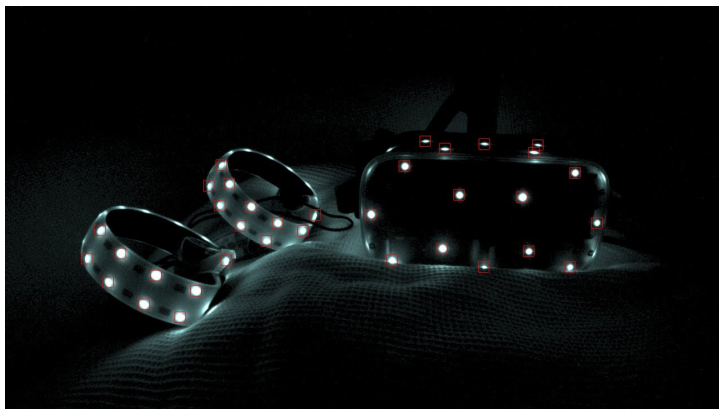
Technologie CAVE je drahá a ve srovnání s HMD pro každodenní běžné používání v domácnosti nepraktická. Tato práce se zaměřuje především na zařízení HMD cenově vyšší třídy. Brýle se distribují jako ucelený produkt s vlastním hardwarem a softwarem. Zejména jsou zde představeny vybrané produkty od společností Oculus a HTC. Od společnosti Oculus jsou to produkty Rift, Quest 1 a 2. Od společnosti HTC se jedná o Vive a jeho verze.

Zvolené produkty jsou doporučena a podporovaná zařízení pro práci s aplikací VRClassroom. S pomocí těchto produktů je prováděn vývoj aplikace.

2.2.1 Oculus

Společnost Oculus vznikla v roce 2012. Postupem času se jejím korporátním nástupcem stala společnost Reality Labs spadající pod Meta Platforms (známé také jako Facebook Inc.). Prvotním produktem byl Oculus Rift. Pod společností Facebook vznikaly produkty Oculus Go (2018), Oculus Quest (2019), Oculus Rift S (2019) a Oculus Quest 2 (2020). [25]

Produkty řady Rift jsou navrženy pro připojení k výkonnému počítači. Ten umožní pokročilé vykreslování grafiky. Rift je v základu složen z headsetu, senzorů pohybu a ovladačů. Sledování je dosaženo pomocí technologie známé jako constellation positioning technology. Jedná se o optické sledování. Senzory (zpravidla dva) jsou staticky umístěny v místnosti a obsahují videokameru. Headset a ovladače jsou označeny markery (infračervené LED diody), které periodicky svítí (Obrázek 2.4).



Obrázek 2.4: Snímací systém produktu Oculus. [15]

Tento způsob sledování se nazývá outside-in tracking system (externí kamery snímající dílčí komponenty). [5] HMD disponuje OLED displeji s rozlišením 1080x1200 a obnovovací frekvencí 90Hz (novější Rift S má LCD 2560x1440 s 80 Hz) [26]. Zorné pole (FOV) se udává 110 stupňů a oblast sledování je obdélník 5 x 11 stop (~152 x ~335 cm). Headset obsahuje akcelerometr, gyroskop a magnetometr. Jak již bylo zmíněno výše, systém potřebuje ke svému chodu výkonný počítač s grafickou kartou. Doporučené specifikace jsou: grafika NVIDIA GTX 970 a vyšší, procesor Intel i5 a vyšší, paměť RAM 8 GB a více, minimálně 2x USB port verze 3.0 a operační systém

Windows 7 a vyšší. Pro novější verze Rift se tyto parametry mohou měnit. Roku 2020 Facebook ukončil výrobu produktu Rift [14].



Obrázek 2.5: Rozbalený produkt Oculus Rift S. [18]

Produkty řady Quest (podobně jako jeho předchůdce Oculus Go) jsou samostatná bezdrátová zařízení (Obrázek 2.6), která v sobě mají operační systém Android. Podporují sledování polohy se šesti stupni volnosti. Dále disponují interními senzory a řadami kamer v přední části headsetu, namísto externích sledovacích senzorů. Kamery se také používají jako součást bezpečnostního prvku „Passthrough“, který ukazuje šedotónový pohled z kamer v případě, že uživatel opustí hraniční oblast virtuálního prostředí. Sledovací systém je známý jako inside-out (kamery jsou integrovány v brýlích a neexistuje žádné externí snímací zařízení). Sledovací systém je založen na konceptu algoritmu počítačového vidění zvaného SLAM (Simultaneous Localization And Mapping). Sledovací systém si pomocí kamer zvolí několik pevných bodů v prostředí podle předem určených vlastností (vzniklé oblak bodů - point cloud). Mezi vlastnosti, které hledá Quest z pořízených obrázků kamer, patří křivost, respektive rohovitost objektů (jelikož detekce rohů je snadná a účinná). SLAM v první řadě odhadne pozice určených pevných bodů v rámci prostředí vůči pozorovateli. Poté odhadne vlastní pozici pozorovatele vůči pevným bodům, za předpokladu, že se nehýbou. Při určování pozice pevných bodů se provádí mapování (mapping) a při určování vlastní pozice se provádí lokalizace (localization) vůči vzniklé mapě. [38] [31]



Obrázek 2.6: Rozbalený produkt Oculus Quest 2. [30]

Oculus Quest má rozlišení OLED displejů 1600x1440 pixelů a obnovovací frekvenci 72 Hz. Oproti tomu novější verze Quest 2 má displeje LCD 1832x1920 pixelů s obnovovací frekvencí 72 Hz. Procesorem je tříjádrový Qualcomm Snapdragon 835 s maximální frekvencí 2.3 GHz (oproti tomu Quest 2 má tříjádrový Qualcomm Snapdragon XR2 Platform). Operační paměť Questu má 4 GB (Quest 2 má 6 GB). [26]

Společnost Oculus kromě vlastního hardwaru dodává i vlastní programátorské rozhraní. Rozhraní slouží k vývoji aplikací pro zmíněné produkty. Při vývoji aplikace s podporou virtuální reality je tedy důležité zvážit, zdali je Oculus podporovaná platforma a až poté aplikaci přizpůsobit pro kompatibilitu s Oculus zařízeními. Zásadní rozdíl je v podpoře bezdrátových zařízení řady Quest, protože navíc běží na platformě Android. Kompletní dokumentace je k dispozici online [27].

2.2.2 HTC

HTC (High Tech Computer) je tchajwanská společnost zabývající se spotřební elektronikou. HTC zpočátku vyrábělo převážně chytrá mobilní zařízení. V roce 2016 začalo HTC diverzifikovat své podnikání mimo mobilní telefony. Díky partnerství se společností Valve vznikla platforma virtuální reality známá jako HTC Vive.

Za představení stojí hlavně produkty řady Vive (Obrázek 2.7). Jsou navrženy pro připojení k výkonnému počítači, který umožňuje pokročilé vykreslování grafiky. V základu se jedná o headset, sensory pohybu a ovladače. K tomuto hardwaru je k dispozici software, podobně jako u Oculus, pro instalaci a konfiguraci prostředí s názvem VivePort.



Obrázek 2.7: Balení HTC Vive Pro. [35]

Sledování je dosaženo pomocí technologie lighthouse (Lighthouse tracking system), která je navržena firmou Valve a integrována do rozhraní SteamVR. SteamVR je aplikace pro manipulaci s HTC Vive na platformě Windows a zároveň balíček pro vývoj software (taktéž od firmy Valve). Snímání funguje na bázi outside-in. Senzory, základové stanice (base stations), jsou staticky umístěny v místnosti. Základové stanice vyzařují neviditelné infračervené paprsky z mnoha LED diod a dvou laserových emitorů. Laserové emitory produkují horizontální a vertikální plošné vlny. Na LED paprsky a plošné vlny reagují fotosenzory obsažené na headsetu a ovladačích. Jeden průchod algoritmu snímání je následující:

- (1) Obě základní stanice pošlou infračervený paprsek z LED diod.
- První základní stanice zaplaví prostor horizontální vlnou z emitujícího laseru.
- Opakuje se (1).
- První základní stanice zaplaví prostor vertikální vlnou z emitujícího laseru.
- Opakuje se (1).
- Druhá základní stanice zaplaví prostor horizontální vlnou z emitujícího laseru.
- Opakuje se (1).
- Druhá základní stanice zaplaví prostor vertikální vlnou z emitujícího laseru.

Takto zjednodušeně popsaný průchod se opakuje ve frekvenci 60 Hz. Světelný impuls LED diody slouží jako synchronizační mechanismus. Poloha snímaného bodu (jednoho fotosenzoru) se určí jako průsečík dvou paprsků. Každý paprsek odpovídá průniku dvou ploch získaných z vertikálního a horizontálního průchodu laserových vln prostorem (pro každou základní stanici zvlášť). Tento systém pracuje při vyhodnocování rychleji, protože nepoužívá vizuální snímání prostoru. Headset a oba ovladače na sobě mají řádově desítky fotosenzorů. [29]

HTC Vive Pro má rozlišení displejů 1440x1600 pixelů a obnovovací frekvenci 90 Hz. Zorný pozorovací úhel má 110 stupňů. Minimální požadavky pro desktopový počítač jsou: grafika NVIDIA GeForce GTX 1060 nebo vyšší, procesor Intel Core i5 nebo vyšší, operační paměť 4GB a více, minimálně jeden USB 3.0 port, operační systém Windows 7 nebo novější.

HTC Vive Pro 2 má rozlišení displejů 2448x2448 pixelů (4K) a obnovovací frekvenci 120 Hz. Zorný pozorovací úhel má 120 stupňů. Minimální požadavky pro desktopový počítač jsou: grafika NVIDIA GeForce GTX 1060 nebo vyšší, procesor Intel Core i5 nebo vyšší, operační paměť 8GB a více, minimálně jeden USB 3.0 port, operační systém Windows 10 nebo novější. [35]

3 Související práce

Software pro výuku ve virtuální realitě dominuje zejména v oblastech zdravotnictví, inženýrství a všeobecných vzdělávacích nástrojích [19]. Aplikací je velké množství. Na tomto místě pro čtenářovu představu zmiňujeme pouze některé. [20].

Na Lékařské fakultě Jordánské Univerzity vznikl nástroj pro lékařský výcvik ve virtuální realitě, který nabízí 3D zobrazení srdce v reálném čase v interaktivním prostředí [1]. Dalším příkladem je nástroj Simodont [37]. Simodont je 3D haptický simulátor virtuální reality pro preklinický dentální výcvik. Jeho hlavní myšlenkou a vizí je to, že pokud studenti uvidí ukázky výsledků své přípravy v reálném čase na monitoru, mohou objektivně a vizuálně lépe porozumět nedostatkům ve svých manuálních dovednostech. Do systému simulátoru byla začleněna zejména technika 3D obrazu. Ta měla zlepšit schopnost posuzování polohy, vzdálenost a velikost objektu. Vznikla také simulace [12] v sektoru ošetrovatelského vzdělávání, která simuluje nemocniční prostředí s virtuálními pacienty trpícími neurodegenerativním onemocněním a s ostatními postavami. Uživatel plní roli práce zdravotní sestry.

V oblasti robotiky vznikl systém, který umožňuje uživateli ponořit se do simulace virtuální reality skutečného programovacího prostředí robota. Vylepšuje zážitek pomocí hmatové (haptické) a 3D vizuální zpětné vazby [36]. Dalším zajímavým systémem je ViMeLa (Virtual Mechatronics Laboratory). ViMeLa vytváří virtuální mechatronickou laboratoř pro výuku studentů v mechatronice. Součástí projektu jsou tři scénáře, a to: návrh, konstrukce a principy fungování elektromotorů; řešení průmyslové automatizace pro řízení procesu třídění balíků ve skladu; linka na třídění průmyslového a domovního odpadu s pásem [11].

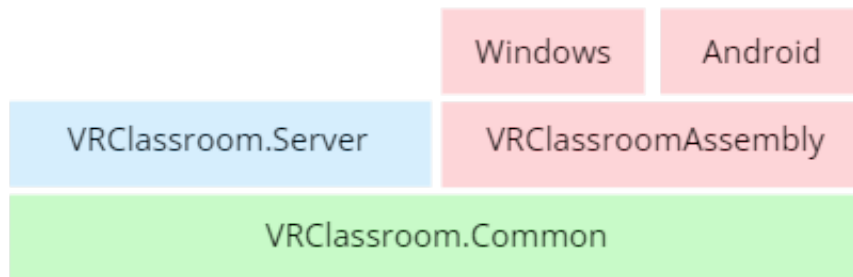
Mezi všeobecný vzdělávací nástroj patří například aplikace Google Expeditions, která díky propojení s databází mapy světa skrze technologii Google Street View, utváří jedinečný zážitek vizualizace reálného světa z pohledu 360 stupňového videa a fotografií [13]. Na Univerzitě v Georgii vznikl software 'VR Classroom', který slouží jako nástroj pro lektory k rozvoji vlastního stylu přednášek ve virtuální realitě [22]. Nutno dodat, že software se stejnojmenným názvem VR Classroom, který vzniká na Západoděské Univerzitě v Plzni se od toho softwaru liší. Pokud není řečeno jinak, je v této práci pod názvem VRClassroom myšlen pouze software vyvíjený pod záštitou Západoděské Univerzity v Plzni.

4 Software VRClassroom

Software s názvem ‘VRClassroom’ je software vyvíjený Západočeskou Univerzitou v Plzni. Vznikl jako doplňující nástroj pro výuku témat náročných na prostorovou představivost. Využívá běžné nástroje pro virtuální realitu, jako jsou desktopový počítač a brýle od výrobců Oculus nebo HTC. Navíc existuje verze pro desktopového klienta, která vykresluje sdílenou scénu na monitoru. Software dovoluje interakci s 3D objekty jednoduchým a intuitivním způsobem. Studenti mohou sledovat virtuální scénu z libovolně zvoleného úhlu pohledu, a interagovat se sdílenou scénou přihlášením se o slovo a použitím sdíleného ukazovátka. Software byl testován na předmětu Zpracování polygonálních sítí (KIV/ZPOS). Odhalilo se mnoho chyb a nedostatků, které se posléze povedlo vyřešit i v rámci této práce. [16]

4.1 Architektura

Řešení softwaru VRClassroom je založeno na architektuře klient-server [9]. Software VRClassroom je složen z dvou hlavních komponent, přičemž jedna z nich je dělitelná na více dalších (momentálně dvě další). Hierarchický náhled dekompozice řešení je na Obrázku 4.1. Společnou vrstvou je knihovna VRClassroom.Common, která definuje společné prvky pro všechny projekty. Samostatnou komponentou je projekt VRClassroom.Server, který představuje server formou konzolové aplikace. Klientské komponenty jsou postaveny nad společným assembly VRClassroom.Assembly. Jedná se v zásadě o knihovnu, nad kterou jsou stavěny výsledné klientské aplikace. V současné době existují aplikace odladěné pro platformy Windows a Android. Aplikace pro Windows je klasický okení program, který funguje buď v režimu s podporou virtuální reality (např. HTC Vive nebo Oculus Rift), nebo bez podpory virtuální reality. Aplikace pro Android je odladěna pro bezdrátové headsety řady Oculus Quest.



Obrázek 4.1: Architektura projektů tvořící software VR Classroom.

Spustitelné programy projektu řešení VRClassroom lze rozdělit do tří kategorií. Jedná se o klientské aplikace, serverové aplikace a doplňkové nástroje. Zároveň je software navržen pro dva typy uživatelů. Máme na mysli studenty a vyučující.

Celý projekt je implementován v objektově orientovaném jazyce C# [10] nad platformou Microsoft .NET [24]. Klientská aplikace je navíc implementována pomocí herního engine Unity [33]. Z tohoto důvodu mají výsledné spustitelné soubory společnou assembly a překlad na jednotlivé platformy je zajištěn engine Unity. Klientské aplikace tedy vycházejí ze stejného projektu, který obsahuje platformně závislé implementace, které se upřednostňují při samotném překladu. Případné rozšíření o podporu další platformy by znamenalo v ideálním případě pouze rozšířit daný projekt a platformně závislé prvky (zejména uživatelské vstupy a zobrazení), jelikož základní logika aplikace je sdílena.

Řešení projektu obsahuje dodatečné nástroje pro interní ulehčení práce. Jedná se o konzolovou aplikaci VRClassRoom.RecordFileConsoleApp. Tato aplikace kompletuje pořízené uživatelské záznamy akcí s pořízeným zvukovým záznamem. Server její spuštění vyvolává automaticky při požadavku na kompletaci. Je však použitelná i jako samostatný nástroj při manuální kompletaci záznamu či jeho úpravě. Pro správný zautomatizovaný chod musí být její spustitelný soubor v adresáři spouštěného serveru (viz Sekce 4.4).

Dalším nástrojem je desktopová aplikace VRClassRoom.TestApp s grafickým uživatelským rozhraním. Tato aplikace slouží jako prohlížeč testovacích případů pro manuální testování aplikace. Obsahuje předvytvořené testovací případy pro komplexní testování synchronizace uživatelských akcí mezi klientskými aplikacemi různých verzí (viz Kapitola 7).

4.2 Spustitelné programy

Kategorie s klientskými aplikacemi zahrnuje program *vrclassroom.exe* a aplikaci *vrclassroom.apk*. Program *vrclassroom.exe* je klient s grafickým uživatelským rozhraním spustitelný na systému Windows. Tento klient dominuje automatickou detekcí existence zařízení s virtuální realitou. Spouští se ve dvou režimech. Pokud je k desktopovému počítači připojen headset a je aktivní v aplikaci SteamVR (například HTC Vive nebo Oculus Rift), potom se klient spustí v režimu s podporou virtuální reality přímo v daném headsetu přes SteamVR. Pokud počítač nemá připojené zařízení pro virtuální realitu, potom se klient spustí v režimu bez virtuální reality jako klasická 2D aplikace. Aplikace *vrclassroom.apk* je spustitelná na systému Android upraveném pro virtuální realitu (například Oculus Quest 1 a Oculus Quest 2). Aplikaci je nutné na systém Android nainstalovat nejčastěji přes nějaký externí program skrze jiný operační systém. Aplikace je poté spustitelná přes sekci neznámé zdroje. Tento proces je komplikovanější, protože aplikace doposud nebyla publikována na klasický obchod aplikací pro dané zařízení. Kategorie s klientskými aplikacemi mohou používat studenti i vyučující.

Pro klientské aplikace (zejména desktopový klient) existuje konfigurační soubor *configuration.properties*, který obsahuje základní konfiguraci pro prvotní spuštění. Tento soubor je načten při spuštění. Některé konfigurace (ip adresa a port server) jsou použity pouze pokud neexistují lokální preference uživatele. Soubor je textový a jednotlivé elementy jsou odděleny novou řádkou. Konfigurační soubor obsahuje:

- `IMAGE_MAGICK_PATH=C:/Program Files/ImageMagick-7.0.10-Q16-HDRI/magick.exe`
- `MAX_FPS=60`
- `PREFILLED_IP_ADDRESS=127.0.0.1`
- `PREFILLED_PORT=5555`

Atribut `IMAGE_MAGICK_PATH` reprezentuje absolutní cestu k externímu softwaru ImageMagick. Atribut `MAX_FPS` udává maximální omezení vykreslovací frekvence. Údaje `PREFILLED_IP_ADDRESS` a `PREFILLED_PORT` určují výchozí hodnoty pro vyplnění formuláře panelu pro přihlášení k serveru.

Kategorie serverových aplikací zahrnuje jednu konzolovou aplikaci *VRC-lassRoom.Server.exe*, která reprezentuje server. Aplikace je spustitelná na platformě Windows buď přes samotný `.exe` soubor, nebo přes příkazovou

řádku. Aplikace potřebuje k bezchybnému běhu práva pro vytváření adresářů a zápis na disk. Server je spustitelný s parametry nastavení. Parametry se berou buď jako argumenty příkazové řádky, nebo se parsují z konfiguračního souboru *VRClassRoom.Server.exe.config*. Pokud nejsou zadány argumenty příkazové řádky, je konfigurace načtena z konfiguračního souboru. Pokud konfigurační soubor neexistuje, jsou vybrány výchozí hodnoty. Parametry jsou následující:

- **port** (výchozí hodnota: 5555) - celočíselná hodnota portu, na kterém server běží.
- **records_path** (výchozí hodnota: aktuální adresář) - relativní cesta k adresáři se záznamy pořízenými serverem a synchronizovaným zvukem.
- **records_distribution_path** (výchozí hodnota: aktuální adresář) - relativní cesta k adresáři s kompleťovanými pořízenými záznamy (viz Kapitola 5), které jsou určeny k distribuci skrze klientské rozhraní.
- **record_file_console_app_path** (výchozí hodnota: 'VRClassRoom.RecordFileConsoleApp.exe') - relativní cesta ke spustitelnému souboru (.exe) konzolové aplikace VRClassRoom.RecordFileConsoleApp.

Při zadávání parametrů jako argumentů v příkazové řádce je nutné dodržet pořadí: 'port', 'records_path', 'records_distribution_path' a 'record_file_console_app_path'. Argumenty se zadávají pouze hodnotou. Serverové aplikace jsou použitelné vyučujícími. Server není interaktivní aplikace. Uživatel může ovlivnit spuštění pomocí konfiguračního souboru. Server průběžně vypisuje provedené operace do konzole. Využívá doplňkový nástroj VRClassRoom.RecordFileConsoleApp.

Kategorie doplňkových nástrojů zahrnuje konzolovou aplikaci *VRClassRoom.RecordFileConsoleApp.exe* a aplikaci s grafickým uživatelským rozhraním *VRClassRoom.TestApp.exe*. Na platformě Windows je spustitelná konzolová aplikace VRClassRoom.RecordFileConsoleApp přes .exe soubor. Aplikace načte soubory s příponami *.record* a *.mp3* a zkompletuje je do jednoho výsledného souboru s příponou *.record*. Jedná se o doplnění zvukového záznamu do souboru se záznamem provedených akcí. Vstupními argumenty jsou cesty k oběma souborům a název výstupního souboru. Pokud nejsou argumenty zadány, aplikace prohledává aktuální adresář a hledá v něm soubory s těmito příponami. Při více shodách bere poslední nález. Jako název výstupního souboru zvolí název vstupního souboru s řetězcem '_new' na konci. Doplňkové nástroje využije zejména vyučující.

Aplikace `VRClassRoom.TestApp` je spustitelná na platformě Windows přes souboru `VRClassRoom.TestApp.exe`. Aplikace načítá testovací scénáře, které jsou definované v adresáři `TestCases`. Tento adresář navíc obsahuje přílohy k vybraným scénářům. Jednotlivé scénáře jsou definové jako `.json` soubory. Aplikace načítá všechny existující `.json` soubory v daném adresáři. Pokud si uživatel potřebuje přidat další scénář, stačí vytvořit nový `.json` soubor se stejnou strukturou a restartovat aplikaci.

4.3 Uživatelské rozhraní

Uživatelské rozhraní se snaží být konzistentní napříč všemi klientskými verzemi softwaru. Je však závislé na použité platformě. Rozdíly jsou zejména mezi desktopovou verzí aplikace bez virtuální reality a verzemi aplikace s podporou virtuální reality. U desktopového klienta bez virtuální reality je uživatelské rozhraní vykreslováno v popředí virtuální scény (Obrázek 4.2 nahoře). Ve virtuální realitě je uživatelské rozhraní umístěno ve virtuální scéně s pevně danou pozicí ve světě (Obrázek 4.2 dole). Jedná se o klasický přístup, kvůli odlišnému způsobu vykreslování na dva displeje headsetu.



Obrázek 4.2: Porovnání uživatelského rozhraní aplikace VRClassroom u desktopové verze (nahore) a verze s virtuální realitou (dole). V obou případech je ukázána úvodní obrazovka s přihlašovacím panelem na server.

Veškerá funkcionalita napříč uživatelským rozhráním je tvořena ze znovupoužitelných komponent, jejichž podoba se může lišit v závislosti na platformě. Funkcionalita, pokud to jde, je jednotná. Kompletní uživatelské rozhraní se skládá z následujících komponent:

- Panel pro přihlášení k serveru (1)
- Indikátor stavu přihlášení k serveru (2)
- Panel s výčtem přihlášených uživatelů (3)

- Panel s ovládacími tlačítky (4)

Jednotlivé prvky výčtu komponent uživatelského rozhraní jsou viděny na Obrázku 4.3.



Obrázek 4.3: Popis uživatelského rozhraní na desktopovém klientu.

4.3.1 Přihlášení k serveru

Přihlášení k serveru je řešené přes panel pro přihlášení. Pro správné přihlášení jsou zapotřebí informace o IP adrese (IP Address) stroje, na kterém server běží, a informace o portu (Port), na kterém je server publikován. Toto jsou jediné vyžadované informace pro připojení. Uživatelské jméno (Username) a učitelské zaškrtačací políčko (Teacher) společně s heslem (Password) jsou volitelné informace. Pokud není uživatelské jméno vyplněno, je vygenerováno automaticky serverem ve formátu 'user_connectionId', kde connectionId je celočíselný identifikátor připojení k serveru. Vstupní pole s vyplněným hesla je dostupné pouze, pokud je zaškrtnuto zaškrtačací políčko učitele. Uživatel se tímto přihlašuje v roli učitele, pokud je vyplněné heslo správné. Uživatel je přihlášen v roli studenta, pokud heslo není zadané. Obecně nemusí být role rozděleny na učitele a studenta. V této práci jsou dané role používány v kontextu školství. V aplikaci však není explicitně dán výčet rolí, jelikož do této doby nebyla podpora uživatelských rolí zadavatelem projektu vyžadována, a proto jednoznačné rozdělení nebylo implementováno. Aplikace rozděluje práva uživatelům pouze na základě autorizace. Uživatel autorizovaný heslem (učitel) má k dispozici více funkcionalit než neautorizovaný

uživatel (student). Je doporučeno přihlášení pouze jednoho učitele v jeden čas. Přihlášení pro více autorizovaných účtů není podporováno.

Při prvotním startu klientské desktopové aplikace jsou informace mimo jiné o IP adrese a portu načteny ze souboru *configuration.properties*. Uživatelem vyplněné údaje v celém přihlašovacím formuláři jsou při validním přihlášení k serveru uloženy do lokálního uživatelského úložiště. Tyto informace jsou persistentní napříč životními cykly aplikace (tj. jsou k dispozici i po vypnutí a znovuspuštění aplikace).

Funkcionalita klientské aplikace je závislá na stavu připojení k serveru. Aplikace funguje v ‘online’ a ‘offline’ režimech. Režimy, a tedy i funkcionalita, se mění dynamicky podle stavu připojení k serveru. Indikace stavu připojení k serveru je zobrazena v levé spodní části obrazovky (pro klienta bez VR) nebo levé spodní části hlavního panelu (pro klienta s VR). Červená ikona společně s textem ‘no connection’ indikuje offline režim. Zelená ikona s textem ‘connected’ indikuje online režim.

4.3.2 Funkce klientské aplikace

Veškerá funkcionalita klientské aplikace je přístupná ovládacími tlačítky. Ovládací tlačítka jsou dynamicky filtrována na základě režimu aplikace (online nebo offline), na základě autorizace uživatele (autorizován heslem nebo neautorizován heslem) a na základě verze klientské aplikace (s podporou virtuální reality nebo bez podpory virtuální reality). Veškerou funkcionalitu shrnují následující sekce.

Načtení záznamu

Režim: Offline

Autorizace: Není

Verze aplikace: Desktop a VR

Tlačítko: Load Record

Popis: Funkce načtení záznamu načte z lokálního úložiště soubor s příponou *.record* a zobrazí scénu pro přehrávání načteného záznamu. Při stisknutí tlačítka je zobrazen průzkumník souborů, kde je předvybrán adresář lokálního úložiště aplikace. Pro načtení je nutné si daný soubor vybrat přes vyvolaného průzkumníka souborů. Po výběru a úspěšném načtení souboru se zobrazí speciální uživatelské rozhraní pro manipulaci se záznamem. Popis je k dispozici v sekci 5.

Načtení 3D modelu

Režim: Online

Autorizace: Vyučující

Verze aplikace: Desktop a VR

Tlačítko: Load Mesh

Popis: Funkce načtení 3D modelu načte z lokálního úložiště soubor s příponou *.obj* a zobrazí jej uprostřed scény. Kromě klasického formátu *.obj* souboru je podporován navíc formát, který obsahuje barevnou informaci pro každý vrchol. Barvy bodů mimo vrcholy jsou interpolovány lineárně. Informace o barvě je v souboru zapsána přímo za definicí pozice vrcholu, např.: `1.000000 1.000000 -1.000000 1.000000 0.000000 0.000000` je vrchol na pozici $x = 1$, $y = 1$, $z = -1$ a má rgb barvu $r = 1$, $g = 0$, $b = 0$ (červená barva). Kompletní ukázkou takto vytvořeného souboru je barevná krychle v Příloze 9.1.

Načtení prezentace

Režim: Online

Autorizace: Vyučující

Verze aplikace: Desktop

Tlačítko: Load Presentation

Popis: Načtení a zobrazení *.pdf* souboru. Tato funkce je dostupná pouze pro desktopového klienta na platformě Windows. Pro načtení *.pdf* souboru je potřeba součinnost externího nástroje pro práci s obrázky s názvem ImageMagick ve verzi 7.0.10. ImageMagick je software pro vytváření, upravování, skládání nebo převádění digitálního obrazu [17]. Pro správné fungování je nutné vyplnit před startem klientské aplikace absolutní cestu ke spustitelnému *.exe* souboru nástroje ImageMagick v souboru *configuration.properties*. Při stisknutí tlačítka je zobrazen průzkumník souborů, ve kterém je nutné vybrat požadovaný *.pdf* soubor. Soubor je následně načten a pomocí ImageMagick konvertován na sadu obrázků, které jsou následně vykresleny na virtuální tabuli. Virtuální tabule je umístěna přes celou obrazovku (u desktop klienta bez VR) nebo ve scéně (pro VR). U desktop klienta bez VR se lze přepnout mezi virtuálním světem a prezentací pomocí tlačítka Scene / Presentation.

Zahájení záznamu

Režim: Online

Autorizace: Vyučující

Verze aplikace: Desktop a VR

Tlačítko: Start Recording

Popis: Zahájení pořizování záznamu akcí společně se zvukem, viz Sekce 5.

Ukončení záznamu

Režim: Online

Autorizace: Vyučující

Verze aplikace: Desktop a VR

Tlačítko: Stop Recording

Popis: Ukončení pořizování záznamu akcí společně se zvukem, viz Sekce 5.

Odeslání záznamu zvuku

Režim: Online

Autorizace: Vyučující

Verze aplikace: Desktop a VR

Tlačítko: Send Record Audio

Popis: Odeslání nahraného zvuku na server, kde se záznam následně zkompletuje, viz Sekce 5.

Správce stahování

Režim: Online

Autorizace: Vyučující / Student

Verze aplikace: Desktop a VR

Tlačítko: Download Records

Popis: Otevření správce pro stahování zkompletovaných záznamů ze serveru, viz Sekce 5.

Synchronizace pohledu vyučujícího

Režim: Online

Autorizace: Student

Verze aplikace: Desktop a VR

Tlačítko: Sync Teacher View

Popis: Synchronizaci lze rozdělit do 4 scénářů použití.

- Vyučující v klientské aplikaci bez virtuální reality
 - Student v klientské aplikaci bez virtuální reality - V tomto případě je viewpoint streamován a interpolován na straně aplikace studenta. Synchronizace se tedy jeví spojitě.

- Student v klientské aplikaci s virtuální realitou - V tomto případě je synchronizace jednorázová, tj. viewpoint aplikace studenta je nastaven na pozici a y-ovou rotaci viewpointu aplikace vyučujícího.
- Vyučující v klientské aplikaci s virtuální realitou
 - Student v klientské aplikaci bez virtuální reality - V tomto případě je viewpoint streamován a interpolován na straně aplikace studenta. Je sdílena pozice, x-ové, y-ové a z-ové složky rotace jsou brány z headsetu vyučujícího (tzn. student by měl vidět přesně to, co učitel v brýlích).
 - Student v klientské aplikaci s virtuální realitou - V tomto případě je synchronizace jednorázová, tj. viewpoint aplikace studenta je nastaven na pozici a y-ovou rotaci viewpointu aplikace vyučujícího. Je to z důvodu pohodlí. Pokud by totiž byla synchronizace streamována, potom by uživatel viděl obraz z jiného úhlu, než ze kterého se dívá. To má za následek fyzickou nevolnost.

Ve Windows je přístupné tlačítko ‘Sync Teacher View’. Po jeho stisknutí se pošle na server požadavek na synchronizaci viewpointu učitele. Server pošle učiteli požadavek na aktivaci streamování a založí nové vlákno, které periodicky (nyní každých 1000ms) posílá viewpoint všem klientům, kteří mají aktivní streamování. Streamování se dá vypnout tlačítkem ‘Stop Synchronization’, které se zobrazí namísto tlačítka startovacího.

Ve VR je synchronizace prováděna pomocí tlačítka, které je k dispozici v menu na levé ruce. Ve VR je synchronizace pouze jednorázová, tudíž po stisknutí na tlačítko ‘Sync Teacher View’ se pošle požadavek na server k získání viewpointu učitele. Server pošle požadavek učiteli a ten vrátí požadovaná data, která se přepošlou klientovi.

Obnovení transformace 3D modelu

Režim: Online

Autorizace: Student

Verze aplikace: Desktop a VR

Tlačítko: Reset Mesh Transform

Popis: Obnovení transformací zobrazovaného 3D modelu. 3D model je načítán zpravidla autorizovaným uživatelem a je sdílen napříč všemi klienty. Transformace modelu provedené uživatelem, který model načítal, je také sdílena napříč všemi klienty. Model je tedy synchronizován z pohledu vyučujícího. Klientská aplikace (student), která si daný model prohlíží, může

pomocí funkce ‘Manipulace s 3D modelem’ jeho transformace lokálně měnit. Tato změna se ostatním uživatelům neprojevuje. Po stisknutí tlačítka ‘Reset Mesh Transform’ se na model aplikuje poslední známá transformace modelu, která byla synchronizována. Model se tedy nastaví tak, jak jej vidí vyučující.

Manipulace s 3D modelem

Režim: Online

Autorizace: Vyučující / Student

Verze aplikace: Desktop a VR

Tlačítko: Model

Popis: Manipulace s načteným 3D modelem. Transformace, které jsou vyvolány autorizovaným uživatelem (vyučujícím) jsou synchronizovány napříč všemi přihlášenými uživateli (studenty). Transformace, které jsou vyvolány neautorizovanými uživateli (studenty) nejsou synchronizovány a jsou viditelné pouze lokálně danému uživateli. Manipulace se liší pro verze aplikace s virtuální realitou a bez virtuální reality. Aplikace bez virtuální reality má k dispozici v levé horní části obrazovky skupiny tlačítek pod označeními ‘Translate’ (pro translační pohyb) a ‘Rotate’ (pro rotační pohyb). Tlačítka pro translační pohyb jsou ‘Up’ (nahoru), ‘Down’ (dolu), ‘Left’ (doleva), ‘Right’ (doprava), ‘Closer’ (dopředu) a ‘Further’ (dozadu). Tlačítka pro rotační pohyb jsou ‘Up’ (nahoru), ‘Down’ (dolu), ‘Left’ (doleva), ‘Right’ (doprava). Směry translace a rotace odpovídají pravotočivému souřadnému systému s y-ovou složkou směřující nahoru. Velikost jednoho kroku translace je jeden metr. Velikost jednoho kroku rotace je pět stupňů.

Aplikace s virtuální realitou používají pro manipulaci s načteným objektem ovladač umístěn v pravé ruce. Manipulace je intuitivní formou namíření ukazovatele pravé ruky na objekt ve virtuálním světě a následném držení tlačítka pravého ukazováčku (viz Kapitola 6). Objekt se přichytí k ukazovátku a jeho transformace je ovlivněna transformací ovladače pravé ruky. Při uvolnění tlačítka se ukončí manipulace s objektem.

Změna barvy

Režim: Online / Offline

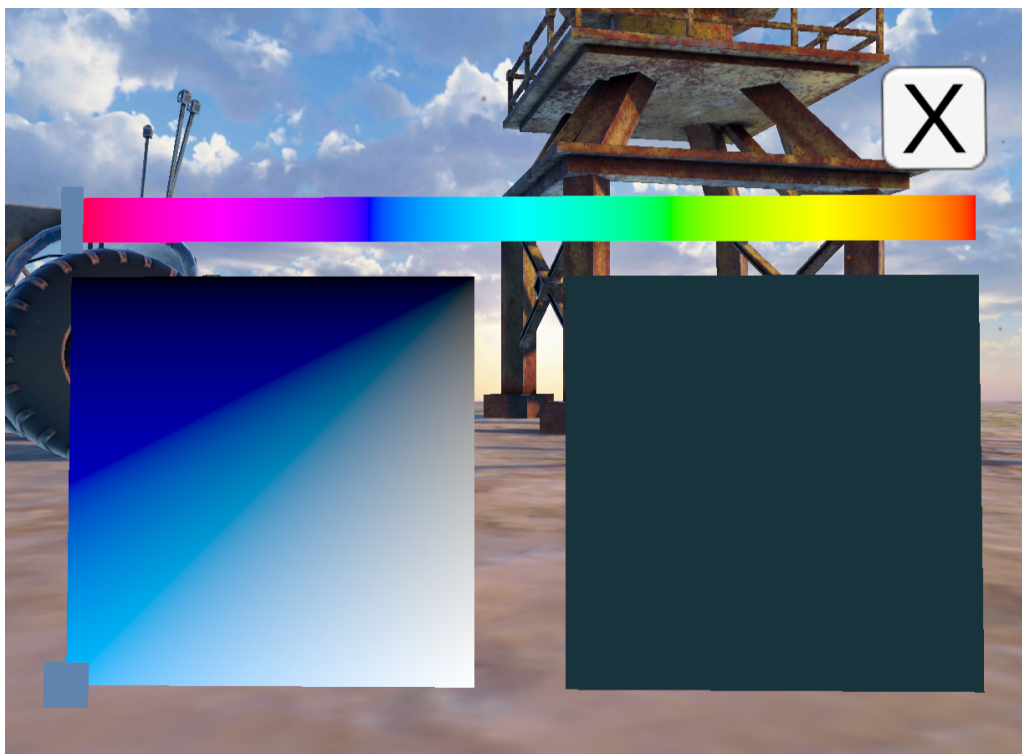
Autorizace: Vyučující / Student

Verze aplikace: VR

Tlačítko: Závislé na kontextu použití.

Popis: Dialog pro změnu barvy (viz Obrázek 4.4) je zobrazitelný přes různá tlačítka nastavující barevnost (např. nastavení štětce, rovin, objektů ...). Nastavení barvy je globální. Dialog obsahuje posuvník a dvě čtvercová okna.

Horní posuvník ovlivňuje barevný odstín. Posuvník na levém čtvercovém okně ovlivňuje jas saturaci. Pravé čtvercové okno zobrazuje výslednou barvu. Manipulace s posuvníky funguje přes namíření ukazovátka pravé ruky na posuvník a stisknutím hlavního tlačítka pod palcem pravé ruky (viz Kapitola 6). Vybraná barva je navíc propagována na tlačítko, které dialog pro změnu vyvolalo. Díky tomu je možná vizuální kontrola použité barvy před kreslením v nastavení štetce.



Obrázek 4.4: Dialog pro globální změnu barvy.

Prostorové kreslení

Režim: Online / Offline

Autorizace: Vyučující / Student

Verze aplikace: VR

Tlačítko: Brush

Popis: Kreslení prostorových křivek v prostoru scény 3D štetcem (viz Obrázek 4.5). Tato funkcionality je dostupná pouze v aplikacích s virtuální realitou. Prostorové křivky, které vytvoří autorizovaný uživatel (vyučující) jsou sdíleny napříč všemi přihlášenými uživateli. Prostorové křivky vytvořené neautorizovaným uživatelem (studentem) jsou viditelné pouze lokálně

danému uživateli. Sdílení funguje pouze v online režimu. Po stisknutí na tlačítko 'Brush' se aktivuje prostorový štětec a zobrazí se okno s nastavením štětce po pravé straně hlavního menu. Štětec se zobrazí namísto modelu pravé ruky. Model štětce je koule o velikosti zvolené šířky prostorové křivky. Křivka se kreslí držením tlačítka na ovladači pod ukazovátkem pravé ruky (viz Kapitola 6). Při uvolnění tlačítka se ukončí kreslení křivky. Křivka je kreslena po částech spojitými úsečkami o velikosti jednoho centimetru. Okno s nastavením křivky obsahuje posuvník ovlivňující šířku křivky a tlačítko se změnou barvy. Šířka křivky je ve výchozím nastavením zvolena na minimum, které odpovídá jednomu centimetru. Maximální šířka je deset centimetrů. Krok pro změnu šířky je jeden centimetr. Tlačítko pro změnu barvy vyvolá globální funkci pro výběr barvy (viz funkce 'Změna barvy'). Štětec je neaktivní, pokud ukazovátko míří na grafické uživatelské rozhraní.

Prostorové kreslení rovných čar

Režim: Online / Offline

Autorizace: Vyučující / Student

Verze aplikace: VR

Tlačítko: Arrow

Popis: Kreslení prostorových úseček v prostoru scény 3D štětcem (viz Obrázek 4.5). Tato funkcionalita má stejné vlastnosti jako funkce 'Prostorové kreslení'. Aktivuje se tlačítkem 'Arrow'. Jediný rozdíl je v průběhu kreslení. Počátek úsečky se vytvoří v bodě, kde uživatel stisknul tlačítko pro kreslení. Konec úsečky je bod, kde uživatel pustil tlačítko pro kreslení.

Prostorové kreslení rovin

Režim: Online / Offline

Autorizace: Vyučující / Student

Verze aplikace: VR

Tlačítko: Plane

Popis: Vytváření rovin v prostoru scény 3D štětcem (viz Obrázek 4.5). Tato funkcionalita má stejné vlastnosti jako funkce 'Prostorové kreslení rovných čar'. Aktivuje se tlačítkem 'Plane'. Rozdíly jsou v průběhu kreslení a okna nastavení. Při držení tlačítka pro kreslení se na místo štětce zobrazí rovina, která opisuje transformace ovladače pravé ruky. Rovina se umístí do prostoru v momentě uvolnění tlačítka pro kreslení. V okně nastavení navíc přibyl posuvník (Transparency) nastavující průhlednost dané roviny. Hodnoty jsou v rozmezí intervalu 0 (plně průhledné) až 1 (neprůhledné). Rovina je velikostně omezena. Je zobrazena jako čtverec o hraně dané hodnotou v nastavení. Vý-

chozí hodnota pro velikost nabývá jednoho metru. Rozsah hodnot velikost hrany rovinného čtverce je od 10 centimetrů až po 2 metry.



Obrázek 4.5: Ukázka objektů vytvořených prostorovým štětcem. Vlevo prostorová křivka, uprostřed prostorová úsečka a vpravo nakloněná čtvercová rovina.

Kreslení na 2D tabuli

Režim: Online / Offline

Autorizace: Vyučující / Student

Verze aplikace: VR

Tlačítko: Paint

Popis: Kreslení na 2D tabuli pomocí ukazovátka pravé ruky. Po stisknutí tlačítka 'Paint' se aktivuje válcové ukazovátko namísto ovladače pravé ruky. Kreslení má obdobné vlastnosti jako předchozí funkce kreslení. Pro kreslení slouží tabule umístěná po pravé straně scény. V okně pro nastavení štětce je navíc zaškrtačovací okénko (Eraser) pro přepnutí na mazání. Mazání funguje stejně jako kreslení, ale kreslí se barvou, která je na pozadí plátna, což evokuje vjem smazání barvy. Barva štětce je černá a nelze ji změnit.

Prostorová guma

Režim: Online / Offline

Autorizace: Vyučující / Student

Verze aplikace: VR

Tlačítko: Eraser

Popis: Mazání prostorových objektů vytvořených štětcem. Mazání funguje obdobně jako kreslení objektů. Při stisknutí tlačítka 'Eraser' se místo prave ruky zobrazí průhledná koule, která symbolizuje gumu. Koule má průměr závislý na parametru velikosti v okně nastavení. Mazání je aktivní při držení tlačítka pro kreslení. Smaže se jakýkoliv nakreslený objekt, který se dostane do kolize s gumou (tzn. uživatel musí přes objekt přejet gumou). Okno nastavení navíc obsahuje tlačítko 'Erase All', které odstraní všechny nakreslené prostorové objekty ve scéně.

Změna ovladače na ruku

Režim: Online / Offline

Autorizace: Vyučující / Student

Verze aplikace: VR

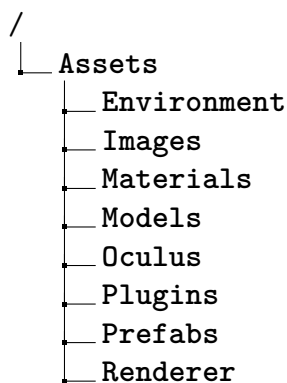
Tlačítko: Hand

Popis: Vrácení modelu pravé ruky namísto ovladače. Pokud je zobrazen model ruky, jsou zároveň vypnuty jakéhokoliv štětce či guma.

4.4 Technické rozhraní

Projekt je tvořen nad platformou .NET s pomocí herního enginu Unity. Projekt je konstruován pomocí integrovaného vývojového prostředí Visual Studio verze 2019. V tomto nástroji je doporučena práce, protože nastavení projektu obsahuje specifické příkazy systému Windows, které se spouští po sestavení projektu. Klientské aplikace jsou vytvářeny pomocí nástroje Unity ve verzi 2019.4.24f1. Zmíněné nástroje jsou používány na operačním systému Windows 10. Jako verzovací software je použit git.

Adresářová struktura projektu VRClassroom je následující:



```

├── Resources
├── Scenes
├── Scripts
├── Shaders
├── SteamVR
├── SteamVR_Input
├── SteamVR_Resources
├── StreamingAssets
├── TextMesh Pro
├── Textures
├── XR
├── Builds
├── Documentation
├── Packages
├── ProjectSettings
├── Solutions
│   ├── VRClassroom.Common
│   ├── VRClassroom.RecordFileConsoleApp
│   ├── VRClassroom.Server
│   ├── VRClassroom.TestApp
│   ├── VRClassroom.Server.Setup
│   └── VRClassroom.sln
├── .gitignore
└── configuration.properties

```

Projekty implementované jako balíček řešení nástrojem Visual Studio jsou umístěny v adresáři **Solutions**. Projekty se otevírají klasickým způsobem otevřením souboru *VRClassRoom.sln* v nástroji Visual Studio. Klient-ské aplikace tvořené pomocí engine Unity jsou umístěny v adresáři **Assets**, který je typický při použití právě Unity. Tento projekt se otevírá přes výběr tohoto adresáře v editoru Unity.

Adresář **Builds** je nastaven jako výstup sestavení jednotlivých aplikací. Adresář **Documentation** obsahuje návod pro připojení k serveru a tuto práci. Adresáře **Packages** a **ProjectSettings** obsahují dodatečné nastavení editoru Unity a projektu pro Unity.

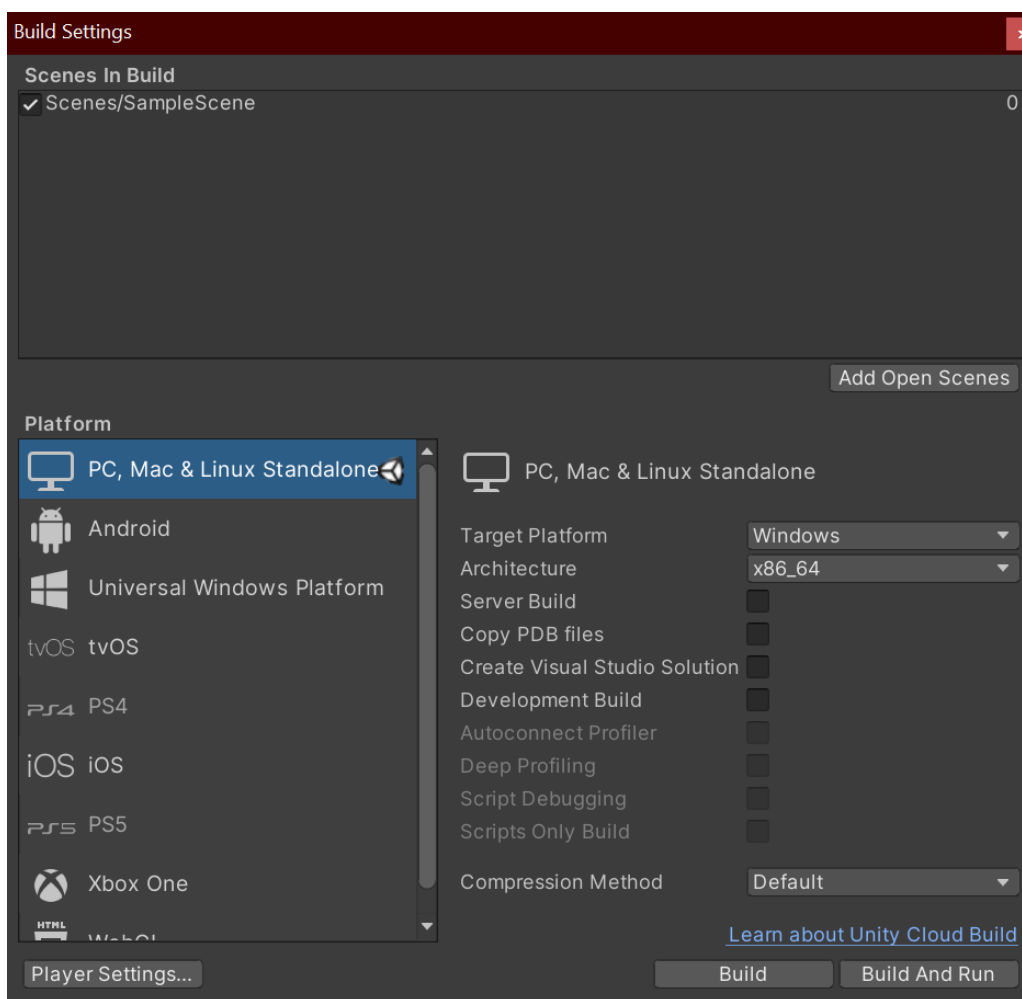
Kazdý projekt v řešení pro Visual Studio je vytvářen nad různým .NET frameworkem. Je tedy nutné mít pro práci s nimi tyto verze nainstalované na počítači. Projekt *VRClassRoom.Common* je knihovna tvořena nad frameworkem .NET Standard 2.0. Projekt *VRClassRoom.RecordFileConsoleApp* je konzolová aplikace tvořena nad frameworkem .NET Core 3.1. Projekt *VRClassRoom.Server* je konzolová aplikace tvořena nad frameworkem .NET Framework 4.7.2. Projekt *VRClassRoom.TestApp* je WPF aplikace použí-

vající framework .NET Framework 4.7.2. Volba verzí nebyla řešena systematicky a její různorodost je důsledek historie vývoje a použité verze engine Unity. Doposud nebyl proveden pokus o sjednocení verzí, avšak sjednocení konzolové aplikace VRClassRoom.RecordFileConsoleApp společně se serverem by mělo být přímočaré. Sjednocení verze knihovny VRClassRoom.Common s ostatními projekty již přímočaré být nemusí, protože je silně závislé na použité verzi engine Unity, která udává výčet verzí, se kterými dokáže pracovat.

4.4.1 Sestavení

Závislost jednotlivých projektů je především na společném projektu VRClassRoom.Common (viz Obrázek 4.1). Sestavení serverové aplikace VRClassRoom.Server je intuitivní a stačí pouze spustit build klasickým způsobem v nástroji Visual Studio (Project -> Build). Server pro správný chod potřebuje přístup k aplikaci VRClassRoom.RecordFileConsoleApp a konfigurační soubor *configuration.properties*. Aby programátor nemusel po každém sestavení serveru navíc kopírovat tyto zmíněné objekty do cílového adresáře sestavení serveru, byly vytvořené automatické události spouštěné po sestavení serverového projektu. Tyto automatické události vyvolávají příkazy pro kopírování skrze příkazovou řádku systému Windows a kopírují zmíněné soubory.

Sestavení klientské aplikace je realizováno skrze editor Unity klasickým způsobem (viz 4.6). Proces sestavení je jednotný a liší se pouze výběrem cílové platformy. Projekt je optimalizován pro platformy Windows a Android. Je použito výchozí nastavení. Pro správné sestavení je nutné mít v klientské aplikaci k dispozici sdílenou knihovnu VRClassRoom.Common. Aby Unity dokázalo pracovat s externími knihovny, je nutné je umístit do adresáře **Plugins**. Z tohoto důvodu byla vytvořena obdobná automatická událost po sestavení projektu VRClassRoom.Common jako u serverové části. Tato událost kopíruje výstup sestavení (.dll soubor) do zmíněného adresáře v projektu pro Unity. U serverové aplikace tento krok není explicitně nutný, protože knihovna je nastavena v závislostech projektu a Visual Studio tento krok dělá automaticky.



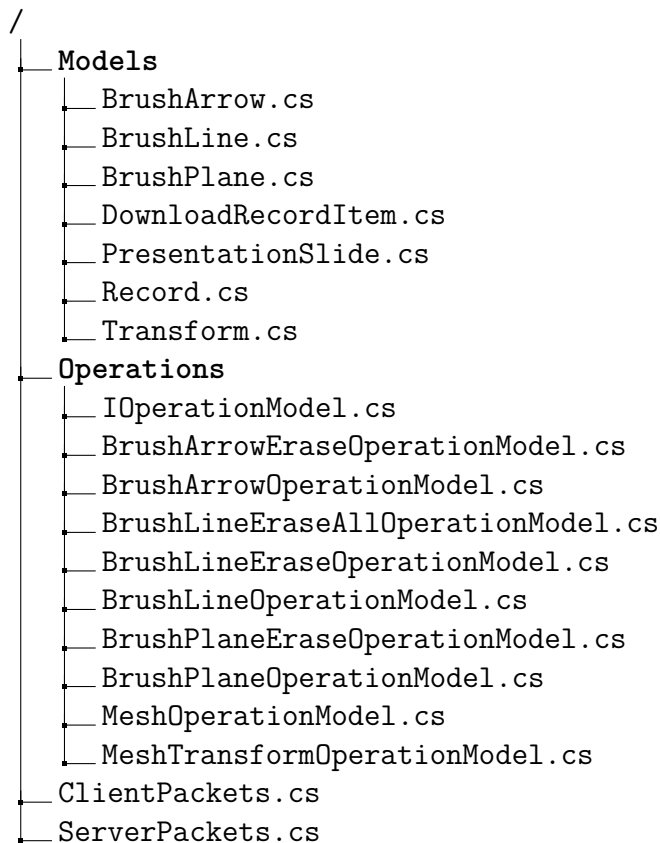
Obrázek 4.6: Okno pro sestavení klientské aplikace v editoru Unity. Vybraná platforma je Windows.

Sestavení klientské aplikace pro systém Windows nebo Android se liší pouze výběrem platformy i přes to, že celý projekt je tvořen v rámci jedné scény. Vytvořené objekty ve scéně disponují speciální komponentou, která ovládá jejich aktivaci v závislosti na zvolené platformě pro sestavení. Ve scéně tedy existují objekty společné a specifické pro každou platformu. Objekty specifické pro platformu se zapínají/vypínají dle zvolené platformy po spuštění aplikace. Jinými slovy, strom scény je dynamicky filtrován v závislosti na platformě. V klientské aplikaci pro Windows se navíc zohledňuje zda je připojen headset přes SteamVR. Sestavená aplikace pro Windows je pouze jedna a podporu nebo nepodporu virtuální reality rozhoduje validní nastavení VR v aplikaci SteamVR. Pokud je aplikace SteamVR zapnutá a komunikuje s headsetem připojeným k počítači, potom se klientská aplikace

spustí v režimu s podporou virtuální reality. V opačném případě se spustí jako klasická desktopová aplikace s grafickým uživatelským rozhraním.

4.4.2 Sdílená knihovna VRClassroom.Common

Sdílená knihovna VRClassroom.Common obsahuje především definici paketů pro síťovou komunikaci. Dalšími celky jsou použité modely objektů a operace nad nimi. Struktura projektu je následující:



Adresáře **Models** a **Operations** obsahují třídy, které reprezentují společné vlastnosti objektů posílaných po síti (data transfer object - DTO). Logicky jsou od sebe odděleny, jelikož adresář **Operations** je součástí záznamového subsystému (viz Sekce 5). Třídy s operacemi reflektují sdílené funkce, které jsou dostupné uživateli v klientské aplikaci. Adresář **Models** byl historicky implementován dříve a jednotlivé třídy reprezentují objekty sdílené po síti.

Klientské pakety slouží jako identifikace příchozí komunikace z klienta na server. Mají prefix 'C' jako 'client'. Výčet výčtového typu ClientPackets, identifikátorů a popisku výsledného použití shrnuje Tabulka 4.1.

1	CPing	Připojení klienta na server.
2	CClearSlides	Smazání prezentace.
3	CSlide	Vytvoření prezentace.
4	CSlideNumber	Přepnutí stránky prezentace.
5	CMeshData	Vytvoření 3D objektu.
6	CBrushLineCreated	Vytvoření křivky.
7	CBrushLineErased	Smazání křivky.
8	CBrushLineErasedAll	Smazání všech křivek.
9	CPaint	Přidání 2D křivky na tabuli.
10	CAuthorization	Autorizace uživatele heslem.
11	CMeshTransform	Transformace 3D modelu.
12	CPointer	Aktivace/deaktivace a transformace ukazovátka.
13	CSignalInfo	Informace o signalizaci studentů.
14	CBrushArrowCreated	Vytvoření čáry.
15	CBrushArrowErased	Smazání čáry.
16	CBrushPlaneCreated	Vytvoření roviny.
17	CBrushPlaneErased	Smazání roviny.
18	CSynchronizeViewpoint	Synchronizace pohledu vyučujícího.
19	CStreamSynchronizeViewpointActive	Aktivace streamování synchronizace pohledu.
20	CStreamSynchronizeViewpoint	Stream pohledu vyučujícího.
21	CRecordActive	Zahájení/ukončení záznamu akcí.
22	CRecordAudioBegin	Zahájení posílání audio záznamu.
23	CRecordAudio	Posílání audio záznamu.
24	CRecordAudioEnd	Ukončení posílání audio záznamu.
25	CRecordDownloadItemsFetch	Získání informací o dostupných záznamech ke stažení.
26	CRecordDownloadItems	Stažení konkrétního záznamu

Tabulka 4.1: Tabulka shrnující jednotlivé výčtové hodnoty typy ClientsPackets společně s celočíselnými hodnotami a popisem reálného použití.

Serverové pakety slouží jako identifikace příchozí komunikace na klienta ze serveru. Mají prefix 'S' jako 'server'. Výčet výčtového typu ServerPackets, identifikátorů a popisku výsledného použití shrnuje Tabulka 4.2.

1	SWelcomeMessage	Připojení k serveru.
2	SInstantiatePlayer	Vytvoření síťového uživatele.
3	SSlide	Vytvoření prezentace.
4	SSlideNumber	Přepnutí stránky prezentace.
5	SMeshData	Vytvoření 3D objektu.
6	SBrushLineCreated	Vytvoření křivky.
7	SBrushLineErased	Smazání křivky.
8	SBrushLineErasedAll	Smazání všech křivek.
9	SPaint	Přidání 2D křivky na tabuly.
10	SAuthorized	Přihlášení jako autorizovaný uživatel.
11	SClearSlides	Smazání prezentace.
12	SMeshTransform	Transformace 3D modelu.
13	SPointer	Informace o ukazovátku.
14	SUserAdd	Přihlášený další uživatel.
15	SUserRemove	Odhlášený další uživatel.
16	SSignalInfo	Přihlášení o slovo.
17	SSignalReset	Smazání všech přihlášení o slovo.
18	SBrushArrowCreated	Vytvoření rovné čáry.
19	SBrushArrowErased	Smazání rovné čáry.
20	SBrushPlaneCreated	Vytvoření roviny.
21	SBrushPlaneErased	Smazání roviny.
22	SSynchronizeViewpoint	Požadavek na sync pohledu vyučujícího.
23	SStreamSynchronizeViewpoint	Stream pohledu vyučujícího.
24	SStreamSynchronizeViewpointActive	Aktivace streamu pohledu.
25	SRecordActive	Aktivace/Deaktivace pořizování záznamu.
26	SRecordDownloadItemsFetch	Informace o možných záznamech ke stažení.
27	SRecordDownloadItems	Stažení konkrétního záznamu.

Tabulka 4.2: Tabulka shrnující jednotlivé výčtové hodnoty typy Server-Packets společně s celočíselnými hodnotami a popiskem reálného použití.

4.4.3 Serverová aplikace

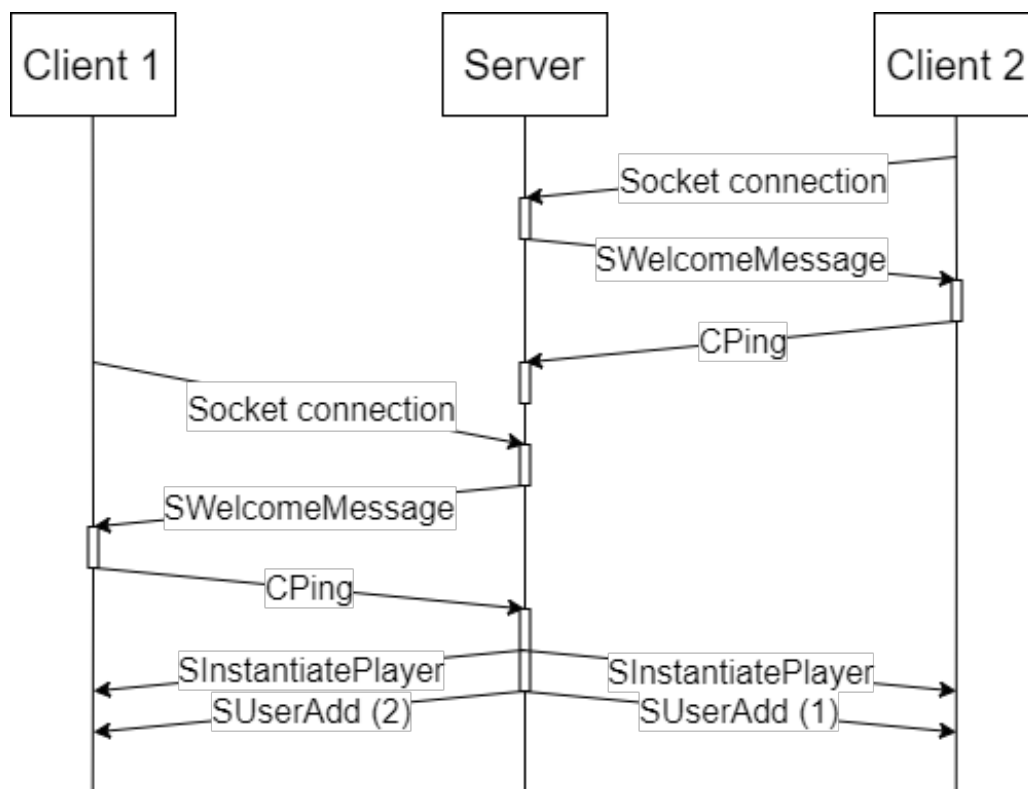
Vstupní bod této aplikace je v souboru *Program.cs*, avšak jádro logiky serveru je ve třídě *ClassRoomManager.cs*. Program načte konfiguraci serveru a inicializuje TCP socket. *ClassRoomManager* slouží jako správce uživatelů a sdílených objektů scény.

Server pro komunikaci používá síťový protokol TCP. Síťová komunikace je zajištěna knihovnou třetí strany *KaymakNetwork.dll*. Rozhraní této knihovny definuje datový typ *KaymakNetwork.Network.Server*, který slouží jako socket. Přes socket probíhá veškerá komunikace. Konfigurace socketu je definována ve třídě *NetworkConfig.cs*. Definice socketu zahrnuje počet přijímaných paketů. Pakety se dodatečně definují ve třídě *NetworkReceive.cs*. Jednotlivé pakety jsou k socketu asociovány přes celočíselný identifikátor a delegát na metodu. Identifikátor je definován ve společné knihovně *VRClassRoom.Common* jako výčtový typ *ClientsPackets* (viz Sekce 4.4.2). Delegát na konkrétní metodu se spustí při zpracování paketu s konkrétním identifikátorem. Tyto metody jsou deklarovány a implementovány v této třídě společně s definicí příchozích paketů. Odesílání zpráv funguje obdobně. Do odesílané zprávy se jako první zapíše celočíselná hodnota výčtového typu serverového paketu a poté zbytek zprávy. Na klientské straně funguje parsování příchozí zprávy stejným způsobem jako na serverové straně. Implementace dílčích zpráv pro jednotlivé serverové pakety jsou k dispozici ve třídě *NetworkSend.cs*.

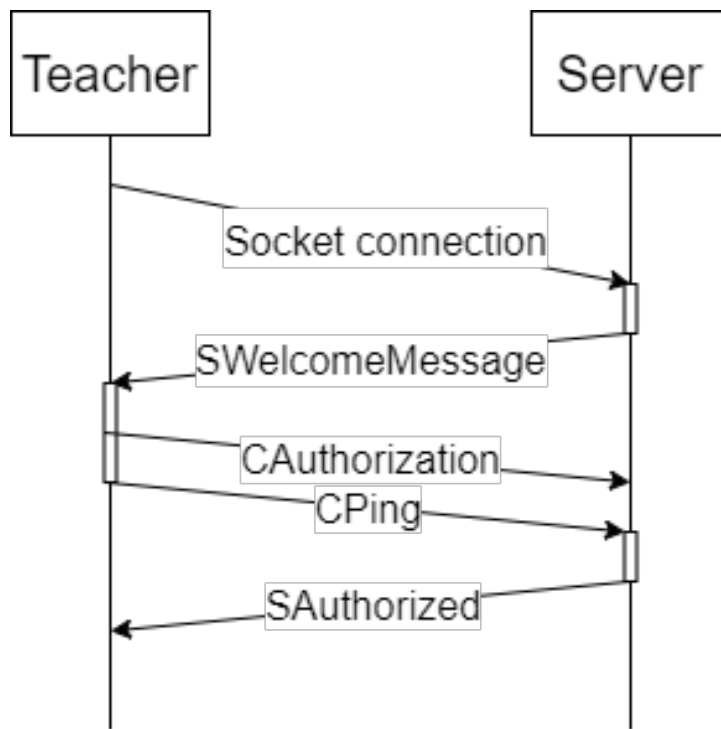
Přihlášení k serveru

Přihlášení k serveru shrnuje Diagram 4.7. Prvotní známka komunikace na serveru je vyvolána přes událost připojení na socketu (Socket connection). Zde je předáván identifikátor připojení. Server následně odešle zprávu *SWelcomeMessage*, která navíc obsahuje verzi serveru. Pokud klientská aplikace v konfiguračním souboru disponuje stejnou verzí, potom komunikace pokračuje. Pokud se verze liší, připojení je zamítnuto. Klientská aplikace pošle zprávu *CPing*, která obsahuje řetězec s volitelným jménem uživatele. Pokud je na serveru více uživatelů, pošlou se po zprávě *CPing* zprávy *SInstantiatePlayer* a *SUserAdd*. *SInstantiatePlayer* vytváří nový objekt síťového uživatele na straně klienta. *SUserAdd* přiřazuje vytvořenému uživateli jméno. Diagram 4.8 shrnuje přihlášení heslem. Pokud je v panelu pro přihlášení zaškrtnut checkbox detekující uživatele v roli vyučujícího, potom je navíc se zprávou *CPing* ještě poslána zpráva *CAuthorization* společně s heslem. Pokud se heslo shoduje s heslem v konfiguračním souboru serverové aplikace, potom je uživatel přihlášen. Odpověď autorizace je poslána formou zprávy

SAuthorized.



Obrázek 4.7: Diagram přihlášení více klientských aplikací k serveru.



Obrázek 4.8: Diagram autorizace heslem při přihlašování k serveru.

4.4.4 Klientské aplikace

Zdrojový kód pro klientské aplikace je dostupný přes editor Unity, který vygeneruje .sln soubor z definovaného assembly VRClassroomAssembly. Doporučená práce s tímto souborem projektu je přes Visual Studio. Hlavní a jediná scéna definující virtuální prostředí a funkcionalitu je MainScene v adresáři **Scenes**. Tato scéna je logicky rozdělena do 4 částí (viz Obrázek 4.9: kamery (Main Camera), správců (Managers), grafického rozhraní (GUI) a hráče (Player).



Obrázek 4.9: Strom objektů popisující rozdělení hlavní scény v editoru Unity.

Sekce se správci obsahuje společnou logiku. Sekce GUI obsahuje objekty uživatelského rozhraní specifické pro verze bez a s virtuální realitou. Sekce Player definuje objekty pro virtuální realitu. Objekt PlayerInput zobecňuje uživatelské vstupní zařízení z různých zařízení a tvoří tak vrstvu mezi funkcemi a ovládáním. Objekt OVRPlayerController je objekt hráče přizpůsob-

bený pro headset Oculus Quest. Objekt `SteamVRPlayerController` je objekt hráče přizpůsobený pro zařízení se SteamVR (Oculus Rift, HTC Vive).

Adresářová struktura assembly `VRClassroomAssembly` je následující:

```
Scripts
├── Brush
├── Compression
├── Configuration
├── Extensions
├── Helpers
├── Lame
├── Network
├── OBJImport
├── Operations
├── Platform
├── Player
├── SteamVR
├── UI
└── ...
```

Adresář **Brush** obsahuje skripty pro manipulaci se všemi typy štětců. Tyto komponenty jsou použity na potomcích objektu `PlayerInput`. V adresáři **Compression** je jediná pomocná komponenta pro kompresi 3D objektu posílaného přes síť. Definice konfigurací (reprezentující souboru `configuration.properties`) je v adresáři **Configuration**. Adresáře **Extensions** a **Helpers** definují pomocné komponenty. Zahrnují například práci s audio konverzí `.wav`, logováním výjimek, správou verze aplikace, limitací fps a další. Adresář **Lame** je skupina externích souborů pro manipulaci s `.mp3` záznamy. Adresář **Network** obsahuje vlastní implementaci zdrojových souborů, které se používají na straně serveru. Obsahuje soubory pro nastavení TCP spojení, přijímání a odeslání zpráv. Implementace komunikace je přizpůsobena pro klientské aplikace a je vytvářena stejným způsobem jako u serveru. Navíc je přidán `NetworkManager`, který spravuje objekty ve scéně v závislosti na jejich stavu na serveru. Adresář **OBJImport** obsahuje sadu skriptů pro import `.obj` souborů. Adresář **Operations** definuje operace pro záznamový subsystém obdobně jako server. Vychází z definic ve společném projektu `VRClassRoom.Common` a obaluje existující entity dodatečnou implementací pro manipulaci s objekty ve scéně. Adresář **Platform** definuje skripty `AnroidOnly`, `VROnly` a `WindowsOnly`, které slouží k dynamické selekci objektů scény v závislosti na spouštěné platformě. Tyto komponenty mají na sobě všechny objekty, které nemají společné využití. Adresář **Player**

definuje vrstvu pro jednotné uživatelské vstupy z více zařízení. Adresář **SteamVR** dodefinovává komponenty potřebné pro práci v zařízeních s VR přes Windows. Adresář **UI** obsahuje kompletní uživatelské rozhraní pro všechny verze aplikací. Projekt mimo výše popsané adresáře obsahuje mnoho dalších skriptů. Jedná se o samostatné různorodé skripty, které nenašly místo v hierarchii výše, nebo tam z historických důvodů nebyly přidány.

5 Záznamový subsystém

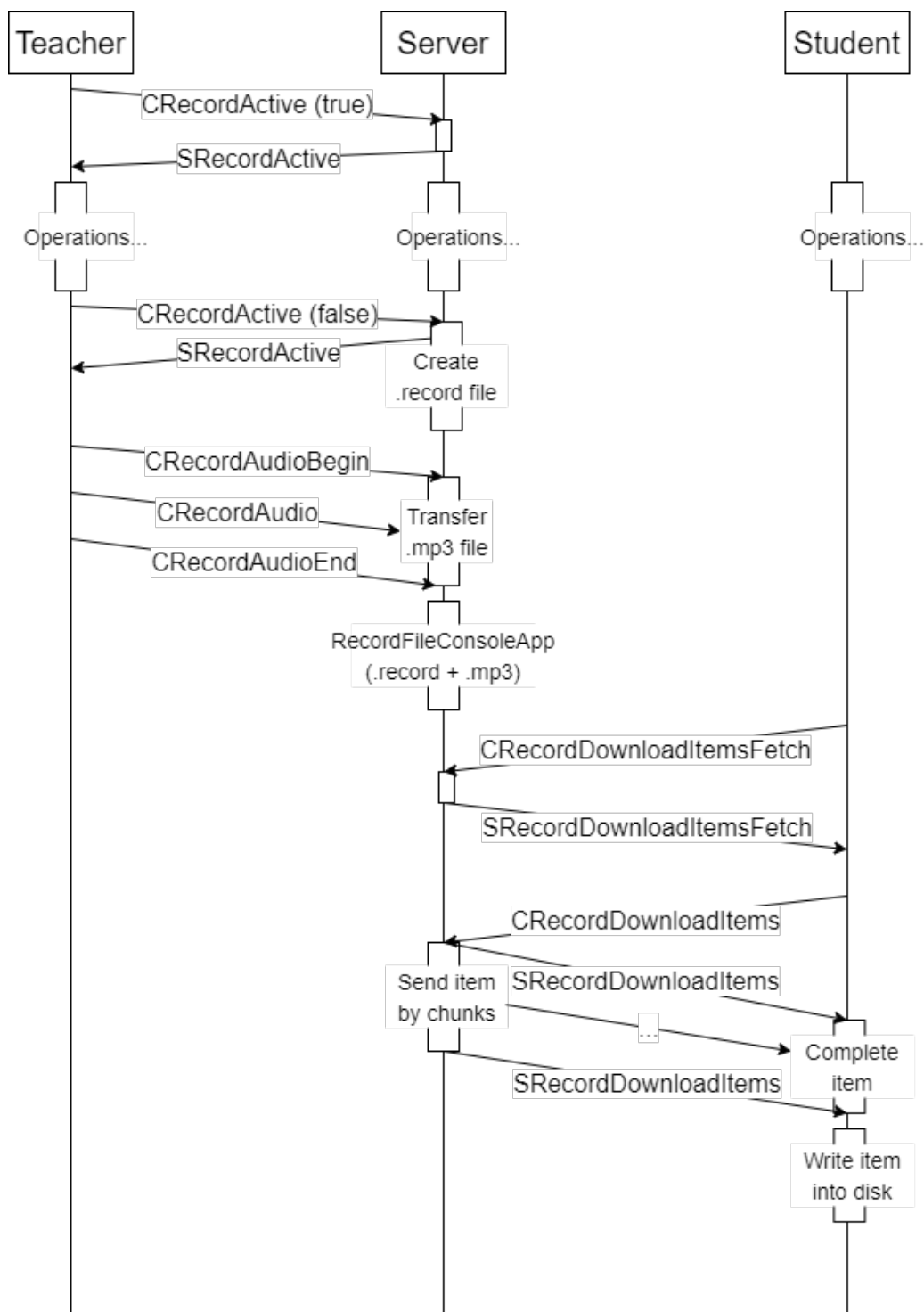
Záznamový subsystém dovoluje vytvářet a následně přehrávat záznamy akcí prováděných autorizovaným uživatelem (vyučujícím) v rámci online sezení. Subsystém lze koncepčně rozdělit do dvou dílčích celků: pořízení záznamu a přehrání záznamu. Mimo pořízení a přehrání byla řešena otázka distribuce záznamů koncovým uživatelům (studentům). Vznikl správce stahování, který je integrován v uživatelském rozhraní klientských aplikací. Následující sekce popisují zmíněné celky subsystému.

5.1 Pořízení záznamu

Záznam může pořídit pouze autorizovaný uživatel přes mechanismus nahrávání. Nahrávání je tedy dostupné pouze v online režimu aplikace. Nahrávání se spustí tlačítkem ‘Start Record’ buď v levém horním rohu obrazovky (pro Windows), nebo v menu na levém ovladači (pro VR). Pokud je klientskou aplikací detekován aktivní mikrofon, potom se při nahrávání také nahrává zvuk pořízený z mikrofonu. Bezdrátové headsety mají integrovaný mikrofon. Nahrávání se ukončí stisknutím tlačítka ‘Stop Record’. Nahrávají se operace:

- Načtení 3D modelu,
- Manipulace s 3D modelem,
- Prostorové kreslení,
- Prostorové kreslení rovných čar,
- Prostorové kreslení rovin,
- Prostorová guma.

Kompletní proces pořízení záznamu společně s jeho kompletací a distribucí je shrnut na Digramu 5.1. Záznam akcí se tvoří na straně serveru, jelikož všechny operace jsou synchronizované. Zvukový záznam se nahrává na straně klientské aplikace a po ukončení pořizování záznamu se dodatečnou funkcionalitou uživatelsky nahraje na server.



Obrázek 5.1: Diagram popisující pořízení záznamu, jeho kompeltaci a následnou distribuci.

Pořízení záznamu začíná požadavkem CRecordActive poslaného z kli-

entské aplikace autorizovaným uživatelem. Server se přepne do stavu nahrávání záznamu. Jakákoliv podporovaná operace se kromě synchronizace navíc uloží, společně s časovou značkou, do datové struktury fronty. Nahrávání se ukončí stejným požadavkem jako zahájení. Při ukončení se na straně serveru serializuje fronta operací do souboru s příponou .record (viz Obrázek 5.2). Tento soubor sám o sobě není kompletní a nelze jej přehrát v klientské aplikaci. Soubor ke své úplnosti potřebuje doprovodný zvukový záznam. Záznam vyučující pošle na server skrze tlačítko ‘Send Record Audio’. Tlačítko otevře prohlížeč souborového systému s přednastaveným adresářem, do kterého se audio záznamy ukládají. Uživatel vybere audio záznam a ten se pošle po částech na server. Server po přijetí poslední části záznamu spustí externí konzolovou aplikaci RecordFileConsoleApp s parametry souborů .record a .mp3. Aplikace následně vytvoří kompletní .record soubor v adresáři pro distribuci, který byl zadán v rámci konfigurace. Tento adresář je viditelný pro klientské aplikace skrze správce stahování. Klientská aplikace si přes požadavek CRecordDownloadItemsFetch nechá vypsát obsah tohoto adresáře. Následně se přes požadavek CRecordDownloadItems zahájí stahování konkrétního záznamu.

```
using System;
using VRClassRoom.Common.Operations;

namespace VRClassRoom.Common.Models
{
    [Serializable]
    1 reference
    public class Record
    {
        public IOperationModel[] Items;

        public byte[] Audio;
    }
}
```

Obrázek 5.2: Zdrojový soubor definující obal pořízeného záznamu. Skládá se z pole operací a bytové nahrávky.

5.2 Distribuce záznamu

Distribuce záznamu probíhá skrze správce stahování v klientské aplikaci. Správce stahování je dostupný pouze v online režimu. Správce se otevírá přes tlačítko ‘Download Records’, které otevře panel s dostupnými záznamy (viz Obrázek 5.3 vlevo). Jednotlivé záznamy jsou reprezentovány zaškrťávacím okénkem a názvem. Záznam lze stáhnout, pokud jde zaškrtnout. Neaktivita zaškrťávacího okénka symbolizuje již stažené záznamy. Záznamů lze zaškrtnout více v rámci jednoho stažení. Stahování se zahájí intuitivním způsobem přes tlačítko ‘Download’. Při stahování je zobrazen průběh právě stahovaného souboru (viz Obrázek 5.3 vpravo).



Obrázek 5.3: Vlevo je panel správce stahování s dostupnými záznamy na serveru. V pravo je panel správce stahování se zobrazeným průběhem právě stahovaného souboru.

Informace o dostupných záznamech ke stažení jsou získávány při každém otevření správce stahování. Jedná se o .record soubory, které jsou uloženy v adresáři určeném pro distribuci záznamů. Cesta k adresáři je nastavena v konfiguračním souboru serveru. Tento adresář je vytvořen při spuštění serveru, pokud neexistuje. Výchozí hodnota je relativní cesta ‘Records/Distribution’. Stejný adresář je použit jako parametr pro výstup kompletace záznamů konzolovou aplikací RecordFileConsoleApp. V ideální situaci je záznam dostupný ke stažení hned po nahrání zvukového záznamu. Server se při přijetí .mp3 audio záznamu snaží najít stejnojmenný .record soubor v konfiguraci definovaném adresáři s pořízenými záznamy. Při kompletaci výsledného záznamu musí mít .record a .mp3 soubory stejný název. Název je proto generován systematicky ve formátu ‘Session_[rok]_[mesic]_[den]_[hodiny]_[minuty].[pripona]’. Jednotlivé podřetězce v závorkách se nahrazují následovně:

- [*rok*] - celočíselná hodnota symbolizující rok. Například ‘2022’.
- [*mesic*] - celočíselná hodnota symbolizující měsíc. Například ‘3’.
- [*den*] - celočíselná hodnota symbolizující den. Například ‘1’.
- [*hodiny*] - celočíselná hodnota symbolizující hodiny. Například ‘15’.
- [*minuty*] - celočíselná hodnota symbolizující minuty. Například ‘8’.
- [*pripona*] - textová hodnota symbolizující typ souboru. Například ‘record’ nebo ‘mp3’.

Tento časový údaj je pořízen při zahájení nahrávání na serveru a je synchronizován s klientem, který nahrávání začal. Název by tedy měl být jednotný v obou adresářích. Příklady záznamů, které byly pořízeny 1. března roku 2020 v 15 hodin a 8 minut jsou ‘Session_2022_1_3_15_8.record’ a ‘Session_2022_1_3_15_8.mp3’. Stahované záznamy se ukládají do lokálního uložení zařízení, které je rezervováno aplikací v uživatelském prostoru systému. Cesta k tomuto adresáři je na každém systému jiná. Adresář je nastaven jako výchozí v prohlížeči souborů při otevření v rámci funkcí načtení a přehrání záznamu.

5.3 Přehrání záznamu

Funkce přehrání záznamu umožňuje interaktivní přehrání dříve pořízeného záznamu skrze přehraňč záznamů. Přehraňč disponuje speciálním uživatelským rozhraním, které se aktivuje po načtení záznamu. Přehraňč je dostupný pouze v offline režimu. Načtení záznamu se aktivuje přes tlačítko ‘Load Record’. Tlačítko otevře známý prohlížeč souborů s přednastaveným adresářem se staženými záznamy. Vybrat lze pouze jeden soubor. Potvrzením výběru se soubor načte do aplikace. Pokud bylo načtení úspěšné, přepne se uživatelské rozhraní do přehraňče (viz Obrázek 5.4). Jakékoliv uživatelsky vytvořené objekty se ze scény odstraní. Přehraňč je velice jednoduchý a pro práci intuitivní. Disponuje dvěma tlačítky a posuvnou lištou. S posuvnou lištou lze libovolně manipulovat během zapnutého i zastaveného přehrávání.

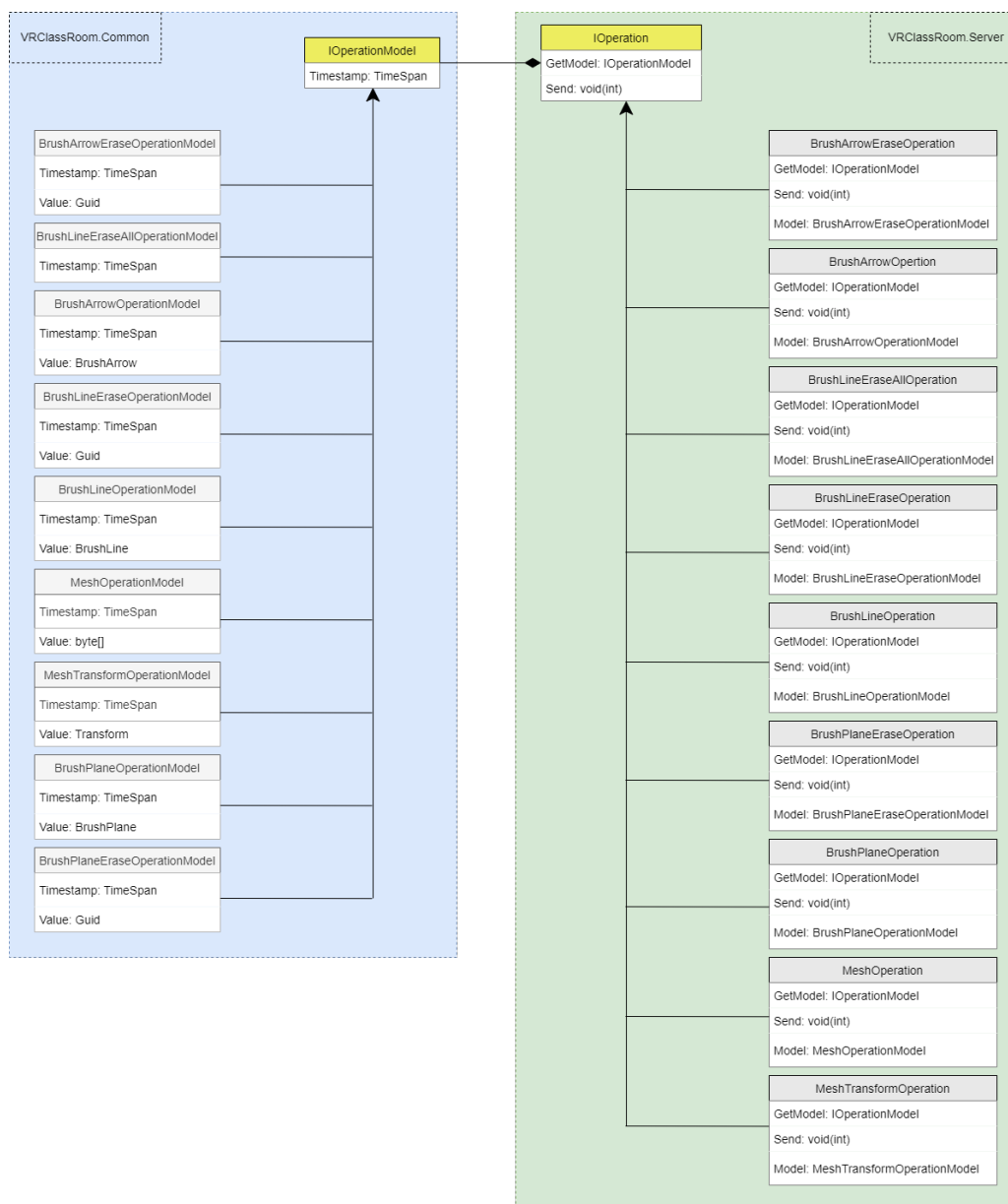


Obrázek 5.4: Příklad přehraavače záznamů na desktopovém klientovi bez podpory virtuální reality.

Přehrávání se spustí tlačítko ‘Start’ a pozastaví tlačítkem ‘Stop’. Posuvník navíc zobrazuje uplynulý čas záznamu. Jednotlivé operace se ve scéně aplikují dle časových značek. Uživatel se může ve scéně libovolně pohybovat a s objekty manipulovat. Ve VR klientovi lze použít tradiční štetce. Přehraavač se zavře tlačítkem ‘Exit’. Při zavření přehraavače se odstraní vzniklé objekty ze scény a aplikace se přepne do klasické úvodní scény.

5.4 Rozšíření subsystému

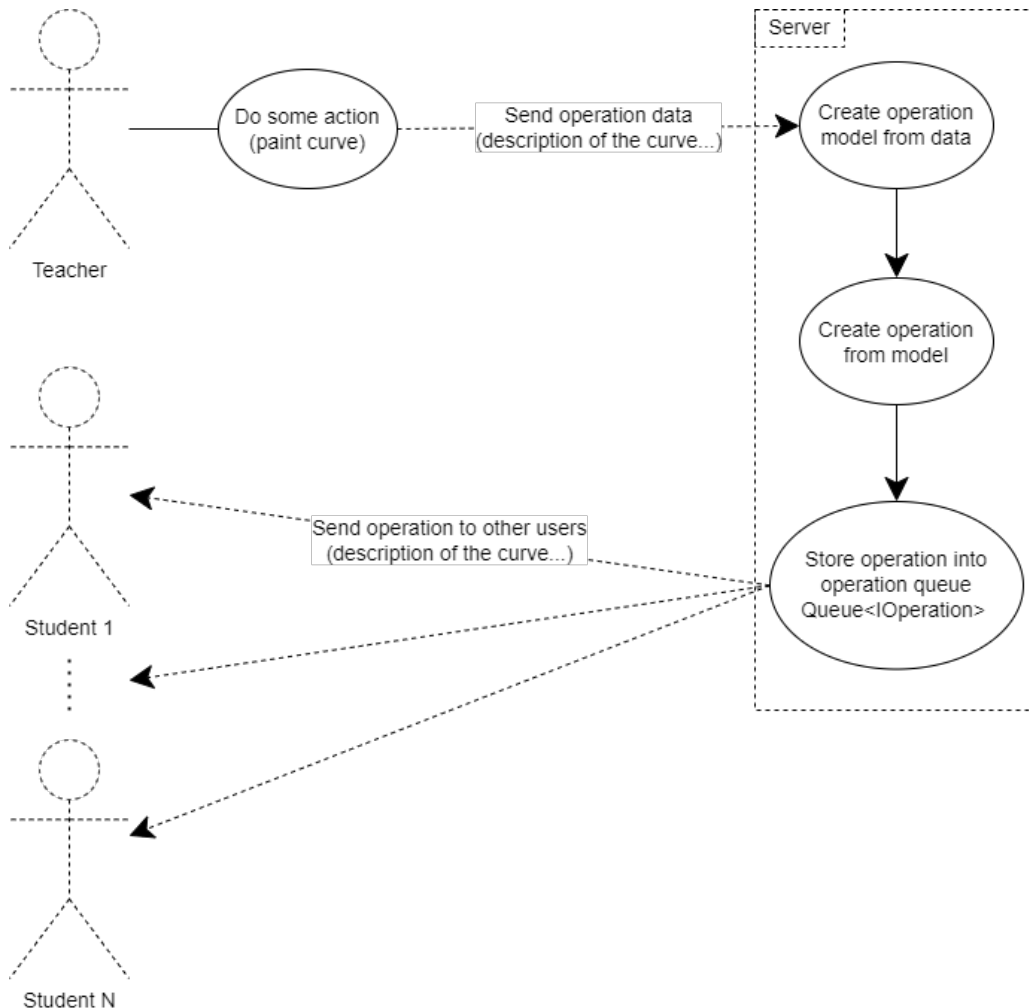
Při implementaci subsystému byl kladen důraz na možné rozšíření o další operace. Jednotlivé akce, které se zaznamenávají v rámci nahrávání jsou ve zdrojovém kódu označené jako operace. Každá sdílená činnost v aplikaci má přiřazenou odpovídající operaci. Modely obalující konkrétní data provedené akce jsou společné pro klientskou a serverovou část. Mají jednotné rozhraní `IOperationModel` a jsou součástí společného balíčku `VRClassRoom.Common`. Rozhraní po potomcích pouze požaduje informaci o časové značce. Samotné operace, které s těmito modely pracují jsou odlišné pro serverovou a klientskou část softwaru. Diagram 5.5 shrnuje veškeré operace s jejich závislosti serverové části softwaru.



Obrázek 5.5: Diagram tříd zobrazující vztahy mezi operacemi na serverové části aplikace.

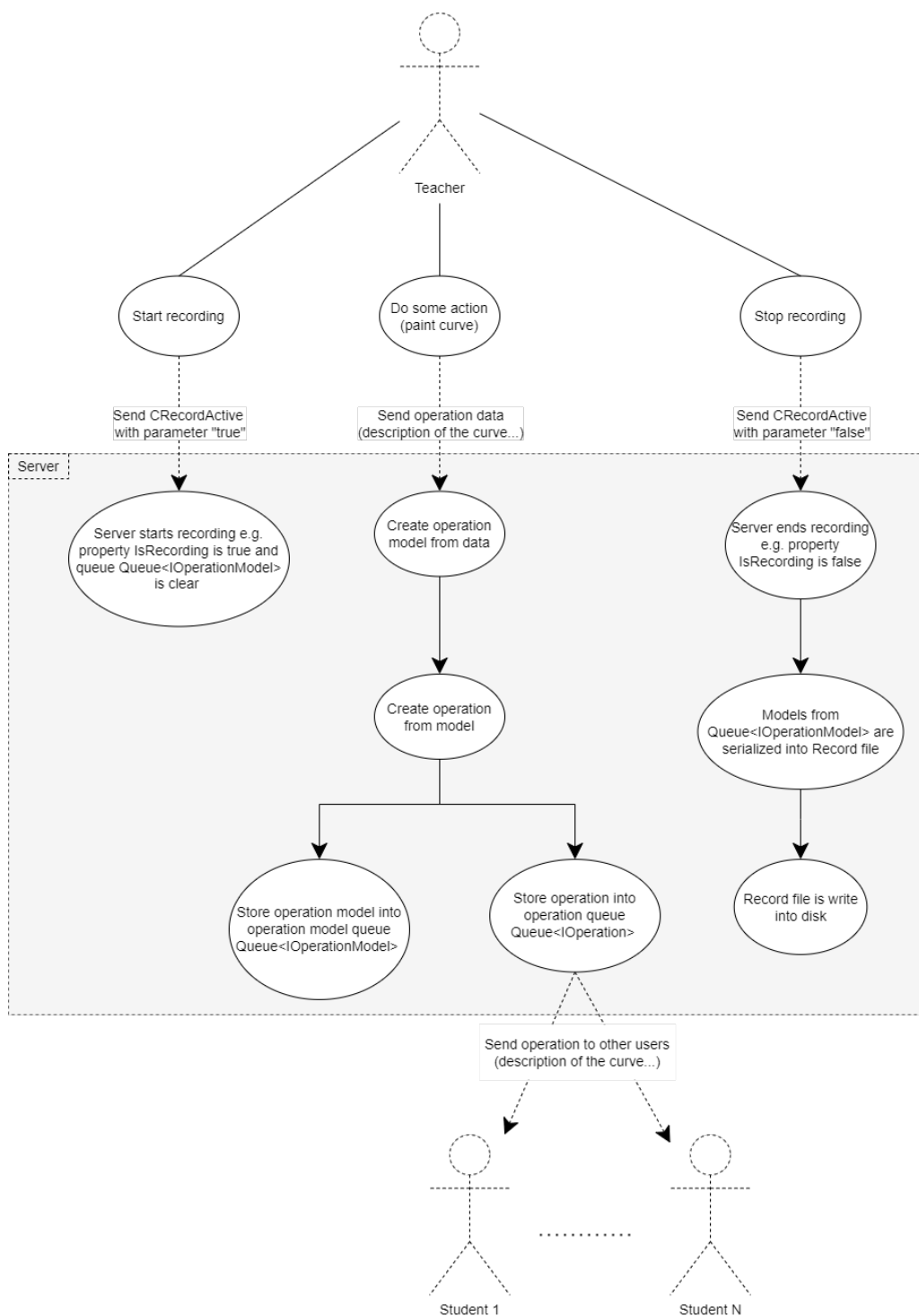
Na serverové straně existuje jednotné rozhraní `IOperation`. Jedná se o kompozici s rozhraním `IOperationModel`. Navíc rozhraní deklaruje metodu `Send`, která má jeden celočíselný parametr reprezentující identifikátor připojení. Tato struktura je zobecněna na synchronizaci akcí v rámci celé aplikace a nikoliv pouze v rámci nahrávání. Diagram 5.6 ukazuje případ synchronizace jedné akce. Tyto operace slouží k synchronizaci v reálném čase. Fronta

operací navíc dovoluje synchronizovat doposud sestavenou scénu s nově připojeným uživatelem.



Obrázek 5.6: Obecný popis (s konkrétním příkladem v závorkách) synchronizace operace mezi klienty.

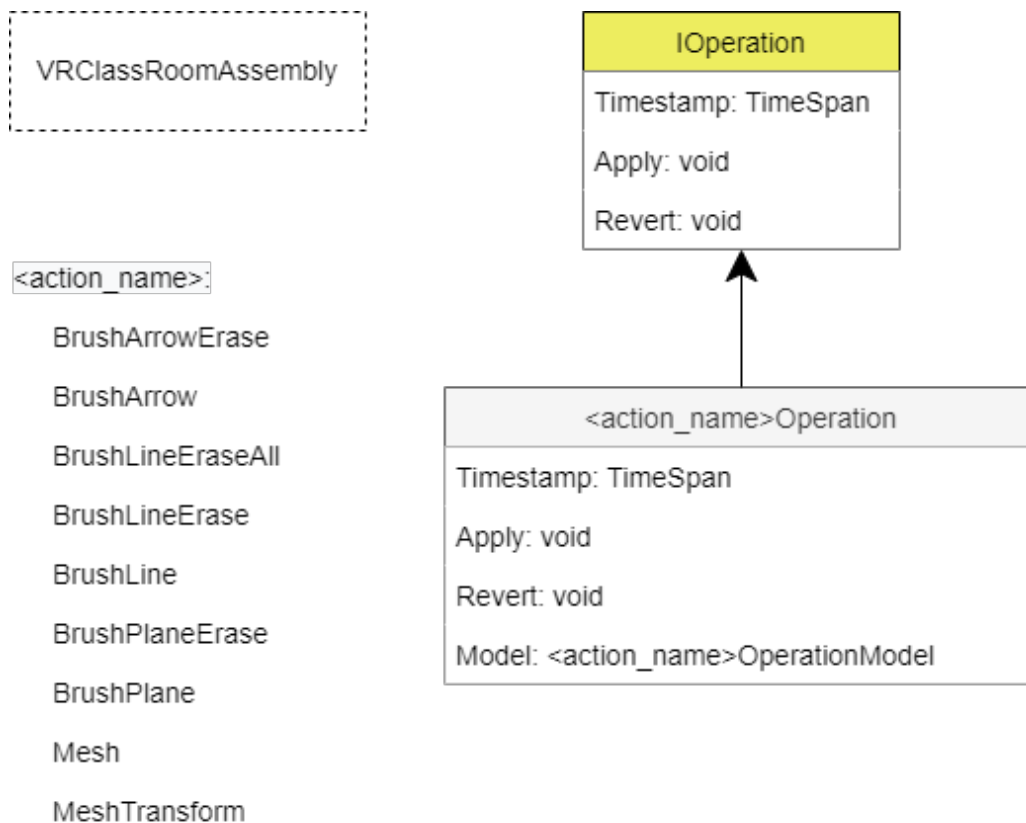
Vedle fronty operací existuje další fronta modelů operací, která se plní jenom v rámci zapnutého nahrávání. Diagram 5.7 popisuje proces nahrávání jako případ užití. V rámci nahrávání se k již existujícímu toku dat navíc ukládají modely jednotlivých operací do speciální fronty. Fronta se při ukončení nahrávání serializuje jako pole modelů operací do třídy Record. Třída Record se posléze serializuje na disk jako samostatný soubor s příponou .record. Takto vzniklý soubor slouží jako základ pro následnou kompletní se zvukovým záznamem a vytvoření plnohodnotného záznamu.



Obrázek 5.7: Diagram případu nahrávání akcí při synchronizaci v reálném čase.

Diagram 5.8 shrnuje veškeré operace s jejich závislostmi v klientské části

softwaru. Klientské aplikace používají vlastní implementaci rozhraní `IOperation`, které navíc definuje metody `Apply` a `Revert`. Třídy, které implementují rozhraní jsou použity pouze při práci s nahrávacím subsystémem. Nejsou používány při synchronizaci, jako je tomu u serveru. Metody rozhraní se používají v přehrazači záznamu. Posuvník v přehrazači určuje dobu přehrávání. Aplikace metod `Apply` a `Revert` je v závislosti na časové značce dané operace a času posuvníku. Metoda `Apply` se aplikuje při překročení časové značky operace posuvníkem. Obdobně se aplikuje metoda `Revert` při opačném posuvu (zpátky v čase).



Obrázek 5.8: Diagram tříd zobrazující vztahy mezi operacemi na klientské části aplikace. Diagram je zjednodušen na jednu generickou komponentu, která má proměnný název `<action_name>`.

Rozšíření o další operaci je realizovatelné třemi kroky:

- Vytvoření datové reprezentace nové operace nejlépe v adresáři `Models` projektu `VRClassRoom.Common`.

- Vytvoření nové operace a modelu v projektech VRClassRoom.Common, VRClassRoom.Server a VRClassRoomAssembly.
- Vytvoření metod pro přijetí a poslání těchto dat v rámci klientské i serverové aplikace. Tento krok zahrnuje rozšíření výčtových typů posílaných paketů a použití v určených třídách.

6 Integrace SteamVR

SteamVR [32] je nástroj pro Windows, který umožňuje používání virtuální reality na různém hardwaru (Valve Index, HTC Vive, Oculus Rift, Windows Mixed Reality headsety a další). Je zdarma dostupný přes digitální distribuční platformu Steam od společnosti Valve Corporation. Jedná se o běhové prostředí pro OpenVR API [28]. OpenVR je aplikační rozhraní, které umožňuje přístup k hardwaru VR od více dodavatelů, aniž by aplikace vyžadovaly specifické znalosti o hardwaru, na který se zaměřují.

Integrace SteamVR do softwaru VRClassroom znamená přizpůsobit software pro OpenVR. Před samotnou integrací neexistoval desktopový klient, který by podporoval virtuální realitu. Virtuální realita byla pouze na verzi pro Android a používala aplikační rozhraní Oculus. V rámci integrace bylo nutné nahradit objekty a funkce vyžadované Oculus API. Bylo nutné rozšířit logiku pro zjišťování platformy a podpory VR. Klientský projekt v Unity obsahuje pouze jednu scénu a platformně závislé objekty se dynamicky filtrují při startu aplikace. Za tímto účelem vznikla komponenta VROnly a informace o režimu byla navíc přenesena na globální úroveň. Proces integrace aplikačního rozhraní OpenVR zahrnoval: přidání OpenVR do projektu v Unity, vytvoření nového objektu hráče, sjednocení vstupů od uživatele a přizpůsobení OpenVR pro práci s GUI.

6.1 Přidání OpenVR do Unity

V unity existuje podpora pro OpenVR přes SteamVR Plugin, který je dostupný zdarma v Unity obchodě s assety. Po importu jsou objekty a zdrojové kódy pluginu dostupné v adresářích s prefixem ‘SteamVR’. Podporu aplikačního rozhraní OpenVR je nutné specifikovat v nastavení hráče pro Windows v sekci ‘XR Settings’. XR je zastřešujícím termínem, který zahrnuje aplikace pro augmentovanou, virtuální nebo smíšenou realitu. Nastavení XR je v současné době v Unity označeno jako zastaralé. Ve verzi projektu je však funkční a upgrade na vyšší verzi by vedl k širokosáhlým úpravám.

Ve stejné sekci hráče existuje i nastavení pro Android, které obsahuje Oculus API. Různé API jsou vybírána dle zvolené platformy pro sestavení. Na platformě Windows je nutné detekovat, zda je OpenVR vůbec použitelné. Jinými slovy se ptáme, jestli je nástroj SteamVR aktivní a je připojen headset k počítači. Za tímto účelem vznikla komponenta VROnly, která detekuje mimojiné existenci XR nastavení. Díky této komponentě je aplikace schopná

zjistit, zda je XR aktivní.

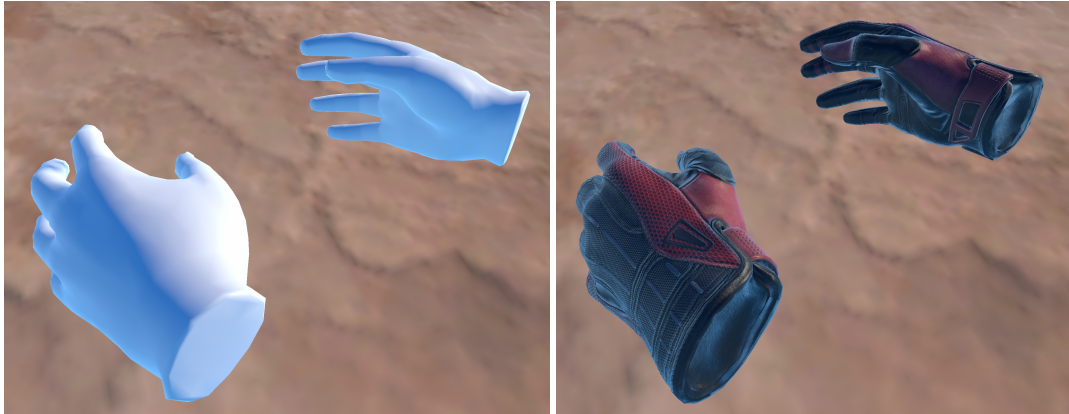
Tento přístup má reálně jeden problém, který je ignorován. Projevuje se při spuštění aplikace pro Windows bez podpory virtuální reality. Unity se při spuštění aplikace snaží inicializovat veškerou funkcionalitu pro OpenVR, protože je nastaveno jako výchozí pro Windows aplikace. Inicialize vyhodí výjimku, pokud není připojen headset. Tím pádem se podporu XR nepodaří aktivovat a aplikace běží v režimu bez podpory virtuální reality. Tento problém je možné vyřešit sestavením aplikace bez zapnuté XR podpory. Je to však krok navíc při sestavování aplikace, který bohužel nelze automatizovat. Je tedy závislý na programátorovi, jenž sestavení vytváří. Aplikačně je tato výjimka explicitně ignorována, takže by se při normálním běhu aplikace neměla projevit.

6.2 Objekt hráče

Doposud ve scéně s virtuální realitou existovala pouze jedna reprezentace uživatele (hráče). Jedná se o objekt `OVRPlayerController` (OVR - Oculus Virtual Reality), který je použitelný pouze na platformě Android. Objekt obsahuje všechny důležité modely a komponenty (kamera, ovladače, pohyb...) pro reprezentaci hráče ve scéně. Máme na mysli znovupoužitelný objekt, který je distribuován v rámci Oculus API. Kromě Oculus komponent obsahuje i vlastní komponenty pro patřičnou integraci do scény. Reprezentace hráče tímto způsobem je kompatibilní pouze s Oculus API a nikoliv s OpenVR API.

Balíček SteamVR obsahuje obdobnou reprezentaci hráče. Jde o objekt ve scéně známý jako `SteamVRPlayerController`. Komponenty tohoto objektu se nesou ve stejném duchu jako u objektu `OVRPlayerController`. `OVRPlayerController` je použitelný pouze na platformě Windows. Takto předpřipravená reprezentace hráče má omezenou funkcionalitu především na ovládání kamery a ovladačů. Nemá integrovanou komponentu pro pohyb, jako je tomu u verze od Oculusu. Pohyb je řešen pomocí snímačů prostředí a je fyzický v rámci vyhrazeného prostředí v reálném světě. SteamVR řeší pohyb ve virtuálním prostředí pomocí teleportace. Tato funkce však ve `VRClassRoom` není a pohyb je zde u všech předchozích klientů řešen plynulým přechodem. Za tímto účelem vznikla komponenta `SteamVRPlayerMovement`, která řeší plynulý pohyb v rámci virtuální scény pro `SteamVRPlayerController` obdobným způsobem jako u `OVRPlayerController`. Komponenta je umístěna v adresáři SteamVR se skripty aplikace. V tomto adresáři jsou umístěny všechny komponenty specifické pro práci se SteamVR.

Vizuálně se tyto dva objekty liší použitým modelem pro ovladače (ruce) a jeho animacemi. Objekt s podporou OpenVR řeší práci s modelem rukou jiným způsobem než objekt s podporou API Oculus. Synchronizovat tyto modely by vyžadovalo značnou práci navíc. Obrázek 6.1 porovnává modely obou použitých knihoven.



Obrázek 6.1: Porovnání modelů ovladačů (rukou) u aplikací s Android a Windows verzí. Android verze (vlevo) používá Oculus API a jeho model. Windows verze (vpravo) používá OpenVR API a jeho model.

6.3 Sjednocení uživatelských vstupů

Vstupy od uživatele jsou důležitým aspektem v systémech s odezvou v reálném čase. Engine Unity, ve verzi projektu, řeší vstupy pomocí správce vstupů (Input Manager). Používá přístup aktivního sledování vstupů namísto přístupu řízeného událostmi. Ve správci se definují názvy akcí a k nim přiřazené identifikátory tlačítek. V aplikaci se poté aktivně sleduje požadovaná akce a její stav. Jedné akci je možné přiřadit více identifikátorů tlačítek či kláves.

Vstupní zařízení se hardwarově liší. Klient bez VR používá jako vstup klávesnici a myš. Klient s VR používá ovladače dle použitého headsetu. Ovladače značek HTC a Oculus se liší. Navíc se liší způsob mapování jednotlivých ovladačů v Unity při použití OpenVR a Oculus API. OpenVR API jako propagaci vstupů od uživatele používá přístup řízený událostmi. Oproti tomu Oculus API používá přístup aktivního sledování vstupů.

Cílem sjednocení vstupů je vytvoření aplikační vrstvy, která propaguje jednotnou práci se vstupy napříč různým typem ovladačů a použitých API. Za tímto účelem vznikla komponenta PlayerInput. Komponenta pracuje zejména s ovladači pro virtuální realitu, jelikož funkce ovládané těmito spe-

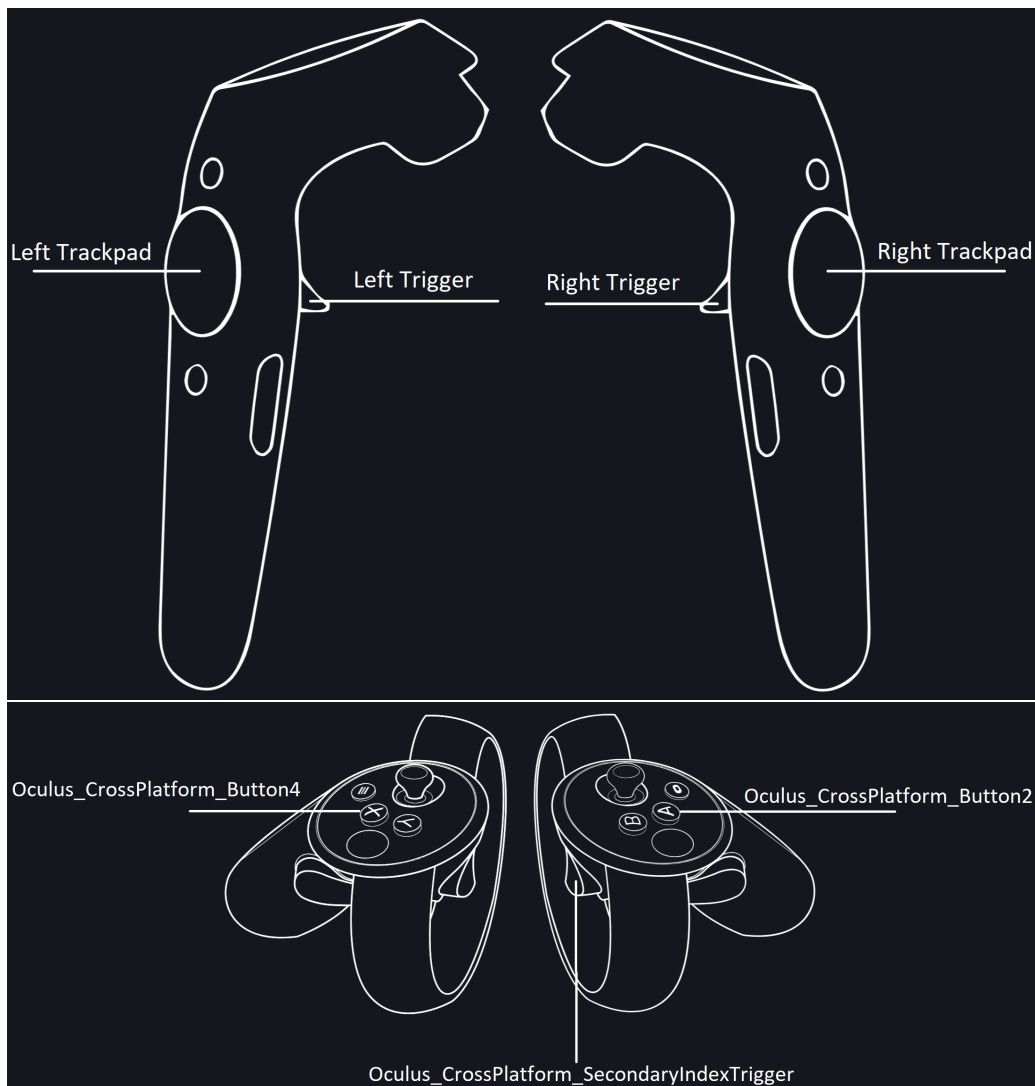
ciálními vstupy nejsou dostupné v desktopové verzi klienta bez VR. Komponenta preferuje propagaci vstupů přístupem řízeným událostmi. Definuje tři události:

- `OnButtonDownEvent` - událost stisknutí tlačítka. Jako parametr přebírá název tlačítka.
- `OnButtonUpEvent` - událost uvolnění tlačítka. Jako parametr přebírá název tlačítka.
- `OnTriggerEvent` - událost zmáčknutí posuvného tlačítka (trigger). Jako parametr přebírá název tlačítka a hodnotu s pohyblivou řádovou čárkou reprezentující úroveň stlačení.

Jakákoliv komponenta, která bude potřebovat informaci o stavu konkrétního tlačítka, se napojí na tyto události. Dle názvu tlačítka je poté schopna provést vlastní logiku. Tabulka 6.1 popisuje mapování názvů tlačítek s reálnými tlačítky na ovladačích. Obrázek 6.2 zobrazuje názvy tlačítek ovladačů HTC Vive a Oculus Touch, které jsou použity v mapovací tabulce.

Název tlačítek ve VRClassRoom	HTC Vive Ovladač	Oculus Touch Ovladač
LeftButton01	Left Trackpad (Press)	Oculus_CrossPlatform_Button4
RightButton01	Right Trackpad (Press)	Oculus_CrossPlatform_Button2
JoystickForward	Trackpad (Move Up)	-
JoystickBackward	Trackpad (Move Down)	-
RightIndex	Right Trigger	Oculus_CrossPlatform_SecondaryIndexTrigger

Tabulka 6.1: Tabulka mapování názvů tlačítek aplikace VRClassRoom a tlačítek ovladačů HTC Vive a Oculus Touch. Názvy tlačítek ovladačů jsou k dispozici na Obrázku 6.2.



Obrázek 6.2: Popis názvů tlačítek ovladačů HTC Vive (nahore) a Oculus Touch (dole).

6.4 Práce s GUI

Interakce s grafickým uživatelským rozhraním (GUI) v aplikacích s podporou virtuální reality funguje prostřednictvím laserového ukazovátka na ovladači pravé ruky. Implementace laserového ukazovátka je dodávána v některých případech použitým API. Jinými slovy Oculus API má jinou implementaci ukazovátka než OpenVR API. Je vyžadováno, aby práce s GUI byla totožná při použití různých ovladačů virtuální reality.

Interakce s grafickým uživatelským rozhraním pomocí laserového ukazo-

vátka OpenVR vyžaduje dodatečnou implementaci speciálních komponent. Laserové ukazovátko implementované pomocí rozhraní OpenVR interaguje s prostředím pomocí událostí vyvolaných kolizemi objektů. Rozhraní poskytuje události PointerIn, PointerOut a PointerClick. Implementace laserového ukazovátko pro SteamVR je ve skriptu, potažmo komponentě, CustomLaserPointer. Tato komponenta naslouchá na dané události a reaguje dle typu kolizních objektů. Pro každý element grafického uživatelského rozhraní (tlačítko, posuvník, rozbalovací seznam a další) je provedena vlastní logika v rámci vyvolané události. Pro jednoduchost byla vytvořena komponenta VRUIItem, která je přidána na všechny elementy grafického rozhraní. Tato komponenta dynamicky za běhu programu zjišťuje aktuální velikost daného elementu a aplikuje na něj komponentu Box Collider. Komponenta je aktivní pouze pokud je aplikace v režimu virtuální reality.

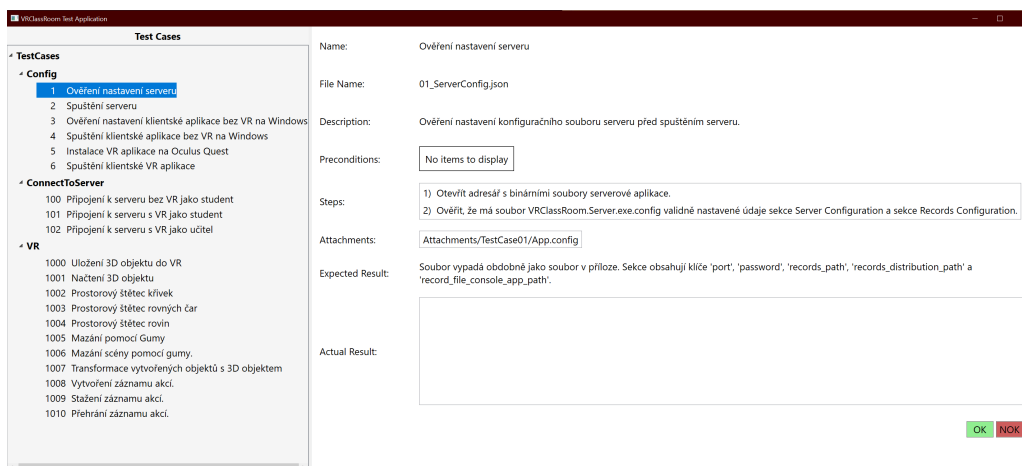
7 Ověrování funkčnosti celého systému

Testování softwaru je proces spouštění softwaru se záměrem hledání chyb. Jedná se o proces validace a ověřování, že software odpovídá sjednaným požadavkům. V rámci testování se provádějí jednotlivé testy, které porovnávají skutečný výsledek s očekávaným výsledkem. Účinnost testu je relativní vůči ceně nalezení chyb v testovaném softwaru. Od stráveného času se odvíjí cena. Tím je celý proces nákladný. Proces testování lze rozdělit z pohledu automatizace do kategorií ručního a automatizovaného testování. Provádění testovacích případů pomocí automatizačních nástrojů je známé jako automatizované testování. Manuální testování je proces, při kterém testeři provádějí testy ručně. Porovnávají očekávané výstupy programu se skutečnými výsledky. Ruční testování je vhodné pro testování komplexních scénářů případů použití. [21]

Automatizované testování jednotlivých funkcí synchronizovaných po síti napříč různými klientkými aplikacemi by bylo příliš náročné na vytvoření. Z tohoto důvodu byl zvolen přístup ručního testování. Ruční testování nejlépe ověří funkčnost celého systému. Za tímto účelem vznikla aplikace VRClassRoom.TestApp, která slouží jako prohlížeč předem vytvořených testovacích případů. Vznikla sada testovacích scénářů zaměřujících se zejména na funkce aplikace, které jsou synchronizované po síti.

7.1 Prohlížeč testovacích případů

Jako prohlížeč předem vytvořených testovacích případů slouží aplikace VRClassRoom.TestApp. Jedná se o aplikaci s grafickým uživatelským rozhraním pro systém Windows (viz Obrázek 7.1). Rozhraní aplikace je koncipováno do dvou částí. Na levé straně okna se nachází stromový seznam načtených testovacích scénářů. Hierarchické složení stromu odpovídá adresářové struktuře. Jako rodičovský prvek se považuje adresář definovaný v konfiguračním souboru. Jednotlivé položky načteného stromu lze vybírat. Jejich detail je zobrazen v druhé části okna. Jeden testovací případ je reprezentován jako soubor s příponou .json.



Obrázek 7.1: Grafické rozhraní aplikace VRClassRoom.TestApp.

Pro vytvoření nového scénáře a správné načtení prohlížečem je nutné splnit strukturu scénáře. Testovací scénář má následující strukturu (viz Příloha 9.2):

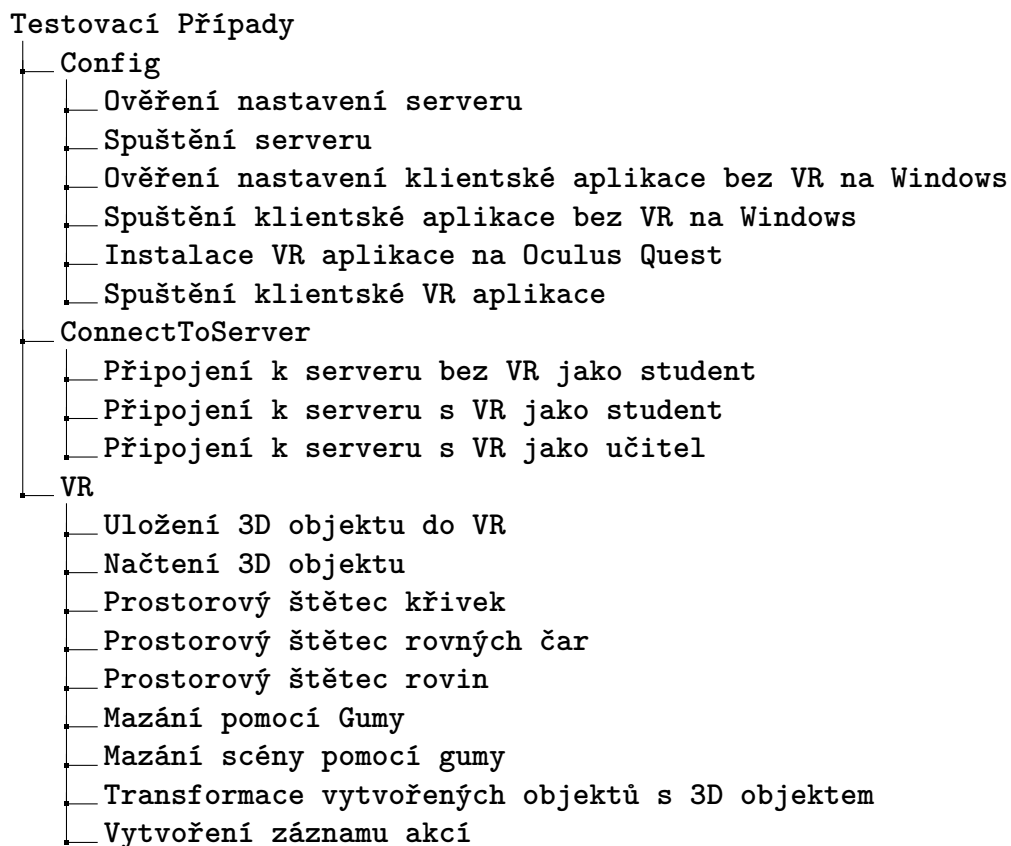
- **ID** - Celočíslný identifikátor testovacího scénáře.
- **Name** - Název testovacího scénáře.
- **Description** - Popis testovacího scénáře.
- **PreconditionIds** - Výčet identifikátorů testovacích scénářů, které je nutné splnit před tímto scénářem.
- **Steps** - Výčet kroků testovacího scénáře.
- **Attachments** - Výčet příloh testovacího scénáře.
- **ExpectedResult** - Předpokládaný výstup testovacího scénáře.
- **ActualResult** - Reálný výstup testovacího scénáře.

Nově vytvořené scénáře lze libovolně hierarchicky strukturovat do adresářů. Jednotlivé přílohy v sekci Attachments jsou reprezentovány relativní cestou k danému souboru ze spouštěného adresáře. Pro přílohy je dedikován adresář **Attachments** v rámci adresáře **TestCases**. Cesta k adresáři **TestCases** je modifikovatelná v konfiguračním souboru *VRClassRoom.TestApp.exe.config*. Při kliknutí na přílohu v aplikaci se vyvolá program, který je systémem určen jako výchozí pro soubor s danou příponou.

V poslední řadě aplikace obsahuje tlačítka OK, NOK a textové pole označené jako 'Actual Result'. Tlačítko OK signalizuje, že se daný test uživateli povedlo reprodukovat s požadovaným výsledkem. Signalizace je vizuální formou zelené ikony po levé straně názvu scénáře v seznamu scénářů. Tlačítko NOK signalizuje, že se reálný výstup liší od požadovaného. Pro tyto případy je vhodné reálný výstup důkladně popsat v textovém poli 'Actual Result'. Poté kliknout na tlačítko NOK. Kromě zobrazení červené ikonky vedle názvu scénáře v seznamu scénářů se navíc daný test uloží na disk i s uživatelem vyplněným textem. V adresáři, ze kterého je aplikace spouštěna, se vytvoří adresář **Results**. Do adresáře se uloží daný test se stejným názvem a ve stejném .json formátu.

7.2 Testovací scénáře

Jednotlivé testovací scénáře jsou děleny do tří kategorií. Testy jsou většinou zasazeny do situace, kdy probíhá práce v aplikaci s virtuální realitou. Práce je sdílena přes server do aplikace bez virtuální reality. Nutno dodat, že tato struktura není striktně dána a je lehce modifikovatelná přes úpravu adresářové struktury. Přehled testovacích scénářů je následující:



	Stažení záznamu akcí
	Přehrání záznamu akcí

První kategorií jsou scénáře ověřující nastavení konfigurací dílčích aplikací. Tyto scénáře jsou v adresáři s názvem **Config**. Jedná se o dvojice testů. První test ověřuje konfiguraci a v nějakých případech i instalaci. Druhý test je závislý na první testu. Ověřuje správnost spuštění aplikace. Pro testy byly vybrány aplikace: serverová aplikace, klientská aplikace bez VR a klientská aplikace s VR. Jako zařízení byly zvoleny: Oculus Quest 1 nebo Oculus Quest 2 a desktopový počítač s operačním systémem Windows.

Druhou kategorií jsou testovací scénáře, které ověřují základní propojení aplikací z různých zařízení pomocí serverové aplikace. Jsou obsaženy v adresáři **ConnectToServer**. Scénáře řeší především připojení z aplikací, které mají různé GUI (desktop a VR). Dále řeší připojení autorizací heslem.

Poslední doposud vytvořenou kategorií jsou scénáře, které pracují s aplikací v režimu virtuální reality z učitelského pohledu. Měly by odpovídat ryze praktickým případům užití. Scénáře jsou k nalezení v adresáři **VR**. Testují práci se štetci, 3D objekty a záznamovým subsystémem.

8 Závěr

Byla vytvořena nová verze 2.0.0 aplikace VRClassroom. Verze dodává novou funkcionalitu (zejména záznamový subsystém) a opravuje chyby, které nebyly explicitně zmíněné v této práci. Doposud (verze 1.0.0) software obsahoval server, klientskou aplikaci pro Windows bez VR a klientskou aplikaci pro Android (VR). Funkcionalita ve verzi s VR byla omezena pouze na kreslení prostorových křivek, kreslení na tabuli, práci s ukazovátkem a manipulaci s již načteným objektem. Načítání probíhalo pomocí desktopového klienta na Windows. V rámci této práce byly opraveny chyby reportované ve verzi 1.0.0 společně se změnovými požadavky. V rámci těchto úprav byl software rozšířen o následující:

- průzkumník souborů ve VR (pro různá načítání přímo ve VR),
- štětce pro rovné čáry a plochy,
- synchronizace pohledu,
- seznam přihlášených uživatelů v rámci online sezení,
- okno pro chybová hlášení,
- nové prostředí scény,
- indikátor aktivního připojení k serveru,
- tlačítko odpojení od serveru,
- načítání .obj souboru s obarvenými vrcholy,
- kontrola verze klientské aplikace vůči serveru,
- systém přihlášení se o slovo,
- a další drobné změny v rámci úprav.

Aplikace byla podrobně popsána. Byl navržen, implementován a posán záznamový subsystém. Záznamový subsystém disponuje rozhraním pro pořizování, stahování a přehrávání záznamů. Byla navržena, implementována a popsána integrace SteamVR do projektu. SteamVR byl do projektu přidán ve verzi 2.2.0. Z důvodu rozsáhlých změn provedených v této práci, slouží práce jako dokumentace k softwaru VRClassroom. Pro ověření funkčnosti

celého systému bylo zvoleno manuální testování pomocí testovacích scénářů. Pro usnadnění provádění manuálních testů vznikla externí aplikace. Testovací scénáře jsou jednoduše rozšiřitelné a hierarchicky strukturovatelné.

Vývoj softwaru VRClassroom stále probíhá. Tato práce byla koncipována jako vstupní bod pro nově příchozí členy týmu. Při tvorbě nových funkcionalit byl kladen důraz na jednoduché použití a rozšiřitelnost. Současná verze aplikace přiložená k této práci je 2.2.0.

Budoucí vývoj softwaru je závislý na vedoucím projektu. Naskytá se hned několik možných rozšíření. Jedním z nich je implementace vlastní síťové komunikaci založené i na bázi protokolu UDP. Díky této úpravě by bylo možné vytvořit jednoduché avatary reprezentující studenty a učitele v rámci online sezení, které by se navzájem viděli v reálném čase. Další možná úprava je přednačtení 3D modelů a dynamické vybírání za běhu aplikace z předem vytvořeného seznamu. Zároveň by bylo vhodné odstranit závislost na nástroji ImageMagick u desktopové aplikace a sjednotit verze aplikací.

V poslední řadě by bylo vhodné vytvořit systém pro polo-automatizované ověření funkčnosti celého systému. Výše doporučené testování je čistě manuální a vyžaduje simulaci problému a následné ověření správnosti. Tento přístup lze rozšířit pomocí nového klienta, která by například načítala vytvořené testovací scénáře a aplikovala je v roli učitele v rámci nově vzniklého online sezení. Uživatel by se poté pouze připojil do vzniklého online sezení a ověřil výsledek pomocí existující aplikace. Eliminována by se tak část simulace testovacího scénáře, která je časově náročná. V rámci této práce nevznikl takto pospaný klient, jelikož z časových důvodů byla upřednostněna cesta manuálního testování.

Literatura

- [1] ALFALAH, S. F. M. et al. A comparative study between a virtual reality heart anatomy system and traditional medical teaching modalities. *Virtual Reality*. 2018, s. 1–6.
- [2] BAMODU, O. – YE, X. Virtual Reality and Virtual Reality System Components. *Advanced Materials Research*. 04 2013, 765-767. doi: 10.2991/icsem.2013.192.
- [3] BARNARD, D. *Degrees of Freedom (DoF)* [online]. VirtualSpeech Ltd., 2019. Dostupné z: <https://virtualspeech.com/blog/degrees-of-freedom-vr>.
- [4] BOAS, Y. A. G. V. Overview of Virtual Reality Technologies. *Computer Science*. 2012.
- [5] BOGER, Y. *Overview of Positional Tracking Technologies for Virtual Reality* [online]. Road to VR, 2014. Dostupné z: <https://www.roadtovr.com/overview-of-positional-tracking-technologies-virtual-reality>.
- [6] BOZKIR, E. Towards Everyday Virtual Reality through Eye Tracking, 2022. Dostupné z: <https://publikationen.uni-tuebingen.de/xmlui/handle/10900/125488>.
- [7] *CAVE Immersive 3D Display* [online]. Visbox, 2020. Produkt CAVE od společnosti Visbox. Dostupné z: <http://www.visbox.com/products/cave/viscube-m4>.
- [8] CHLÁDECKÝ, T. Stereoskopická projekce. Diplomová Práce, Západočeská Univerzita - Fakulta Elektrotechnická, 2011.
- [9] *client-server architecture* [online]. Encyclopædia Britannica, 2021. Dostupné z: <https://www.britannica.com/technology/client-server-architecture>.
- [10] *C#* [online]. Microsoft, 2022. Dostupné z: <https://dotnet.microsoft.com/en-us/languages/csharp>.
- [11] CVETKOVSKI, G. et al. ViMeLa Project: An innovative concept for teaching mechatronics using virtual reality. *Przeglad Elektrotechniczny*. 05 2019, s. 18–21.

- [12] ELLIMAN, J. – LOIZOU, M. – LOIZIDES, F. Virtual Reality Simulation Training for Student Nurse Education. s. 1–2, 09 2016. doi: 10.1109/VS-GAMES.2016.7590377.
- [13] *Google Expeditions App* [online]. Google, 2022. Dostupné z: <https://sites.google.com/tcsnc.org/tcs-g-expeditions/google-expeditions-app>.
- [14] HEANEY, D. *Facebook Killing Oculus Rift Line, Aims To Make Quest 2 'Best PC VR Experience'* [online]. UVR Media LLC., 2020. Dostupné z: <https://uploadvr.com/facebook-killing-oculus-rift>.
- [15] *How do common virtual reality tracking systems work?* [online]. Mechatech, 2021. Dostupné z: <https://www.mechatech.co.uk/journal/how-do-common-virtual-reality-tracking-systems-work>.
- [16] HÁCHA, F. – VANECEK, P. – VÁŠA, L. A Virtual Reality Platform for Immersive Education in Computer Graphics. In SOUSA SANTOS, B. – DOMIK, G. (Ed.) *Eurographics 2021 - Education Papers*. The Eurographics Association, 2021. doi: 10.2312/eged.20211001. ISBN 978-3-03868-132-8.
- [17] *Image Magick* [online]. ImageMagick Studio LLC, 2022. Dostupné z: <https://imagemagick.org/index.php>.
- [18] JAMES, P. *Unboxing the New Oculus Rift Step-by-Step in Pictures* [online]. Road to VR, 2016. Dostupné z: <https://www.roadtovr.com/oculus-rift-unboxing-gallery>.
- [19] KAMIŃSKA, D. et al. Virtual Reality and Its Applications in Education: Survey. *Information (Switzerland)*. 10 2019, 10, s. 318. doi: 10.3390/info10100318.
- [20] KAVANAGH, S. et al. A systematic review of Virtual Reality in education. 2017.
- [21] MAILEWA, A. – HERATH, J. – HERATH, S. A Survey of Effective and Efficient Software Testing. 04 2015.
- [22] MELATTI, M. – JOHNSEN, K. Virtual Reality mediated instruction and learning. In *2017 IEEE Virtual Reality Workshop on K-12 Embodied Learning through Virtual Augmented Reality (KELVAR)*, s. 1–6, 2017. doi: 10.1109/KELVAR.2017.7961556.
- [23] MICHAEL-GRIGORIOU, D. et al. Impact of Immersion and Realism in Driving Simulator Studies. *International Journal of Interdisciplinary Telecommunications and Networking*. 01 2014, 6, s. 10–25. doi: 10.4018/ijitn.2014010102.

- [24] *.NET* [online]. Microsoft, 2022. Dostupné z:
<https://dotnet.microsoft.com/en-us>.
- [25] *Reality Labs* [online]. Wikipedia, 2022. Dostupné z:
https://en.wikipedia.org/wiki/Reality_Labs.
- [26] *Oculus Device Specifications* [online]. Facebook Technologies, LLC., 2022.
Dostupné z:
<https://developer.oculus.com/resources/oculus-device-specs>.
- [27] *Oculus Documentation* [online]. Facebook Technologies, LLC., 2022.
Dostupné z: <https://developer.oculus.com/documentation/>.
- [28] *OpenVR* [online]. Valve Corporation, 2022. Dostupné z:
<https://partner.steamgames.com/doc/features/steamvr/openvr>.
- [29] POÓR, V. Automatická analýza pohybu ramene pro účely rehabilitace ve virtuální realitě. Bakalářská Práce, Západočeská Univerzita - Fakulta Aplikovaných Věd, 2019.
- [30] *QUEST 2* [online]. Facebook Technologies, LLC., 2022. Dostupné z:
https://www.oculus.com/quest-2/?locale=cs_CZ.
- [31] RANGANATHAN, A. *The Oculus Insight Positional Tracking System* [online]. AI ACCELERATORS. Dostupné z:
<https://www.aiacceleratorinstitute.com/the-oculus-insight-positional-tracking-system>.
- [32] *SteamVR* [online]. Valve Corporation, 2022. Dostupné z:
<https://www.steamvr.com>.
- [33] *Unity Real-Time Development Platform* [online]. Unity Technologies, 2022.
Dostupné z: <https://unity.com>.
- [34] *Different Types of Virtual Reality* [online]. Stars & Strikes, 2020.
[cit. 10.12.2020]. Rozdílné typy virtuální reality. Dostupné z: <https://www.starsandstrikes.com/blog/different-types-of-virtual-reality>.
- [35] *VIVE Pro Series* [online]. HTC Corporation, 2022. Dostupné z:
www.vive.com/eu/product.
- [36] VÁZQUEZ, C. – ROJO, A. – SANCHEZ, L. Virtual Reality Robotics System for Education and Training. s. 162 – 167, 11 2010. doi:
10.1109/CERMA.2010.98.

- [37] WANG, F. et al. Application of a 3D Haptic Virtual Reality Simulation System for Dental Crown Preparation Training. In *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, s. 424–427, 2016. doi: 10.1109/ITME.2016.0101.
- [38] WIKIPEDIA. *Oculus Quest* [online]. Wikipedia, 2022. Dostupné z: https://en.wikipedia.org/wiki/Oculus_Quest.

9 Přílohy

9.1 Barevná krychle

o Cube

```
v 1.000000 1.000000 -1.000000 1.000000 0.000000 0.000000
v 1.000000 -1.000000 -1.000000 0.000000 1.000000 0.000000
v 1.000000 1.000000 1.000000 0.000000 0.000000 1.000000
v 1.000000 -1.000000 1.000000 1.000000 0.000000 0.000000
v -1.000000 1.000000 -1.000000 0.000000 1.000000 0.000000
v -1.000000 -1.000000 -1.000000 1.000000 0.000000 0.000000
v -1.000000 1.000000 1.000000 0.000000 1.000000 0.000000
v -1.000000 -1.000000 1.000000 0.000000 0.000000 1.000000
vt 0.625000 0.500000
vt 0.875000 0.500000
vt 0.875000 0.750000
vt 0.625000 0.750000
vt 0.375000 0.750000
vt 0.625000 1.000000
vt 0.375000 1.000000
vt 0.375000 0.000000
vt 0.625000 0.000000
vt 0.625000 0.250000
vt 0.375000 0.250000
vt 0.125000 0.500000
vt 0.375000 0.500000
vt 0.125000 0.750000
vn 0.0000 1.0000 0.0000
vn 0.0000 0.0000 1.0000
vn -1.0000 0.0000 0.0000
vn 0.0000 -1.0000 0.0000
vn 1.0000 0.0000 0.0000
vn 0.0000 0.0000 -1.0000
s off
f 1/1/1 5/2/1 7/3/1 3/4/1
f 4/5/2 3/4/2 7/6/2 8/7/2
f 8/8/3 7/9/3 5/10/3 6/11/3
f 6/12/4 2/13/4 4/5/4 8/14/4
```


f 2/13/5 1/1/5 3/4/5 4/5/5
f 6/11/6 5/10/6 1/1/6 2/13/6

9.2 Šablona testovacího scénáře

```
1  {  
2    "ID": 0,  
3    "Name": "",  
4    "Description": "",  
5    "PreconditionIds": [],  
6    "Steps": [],  
7    "Attachments": [],  
8    "ExpectedResult": "",  
9    "ActualResult": ""  
10 }
```