

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Víceuživatelská prostředí v systémech virtuální reality**

Diplomová práce

Autor: Bc. Tomáš Bareš  
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Bruno Ježek, Ph.D.

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

*vlastnoruční podpis*

V Hradci Králové dne 28.4.2022

Jméno a Příjmení

#### Poděkování:

Tímto bych chtěl velice poděkovat fakultě FIM za možnost zapůjčení zařízení pro virtuální realitu HTC Vive pro možnost vývoje a testování praktické části této práce. Také bych chtěl srdečně poděkovat panu Ing. Brunovi Ježkovi, Ph.D. za ochotné a trpělivé metodické vedení mé diplomové práce až do fáze jejího dokončení. A nakonec bych chtěl poděkovat rodině a přátelům za podporu a pomoc při testování.

## **Anotace**

Virtuální realita je fenoménem dnešní doby, který nachází uplatnění v mnoha průmyslových oblastech. Tato diplomová práce se zabývá využitím virtuální reality, jako prostředku pro náhled do sdíleného víceuživatelského prostředí. Práce se snaží přijít s inovativním řešením pro zjednodušení distanční výuky, která je v aktuální době častým důsledkem šíření onemocnění COVID-19. Sdílené prostředí by mělo usnadnit předávání vědomostí formou přítomných interaktivních prvků, mezi které patří prostorové kreslení. Prostorové kreslení má za účel poskytnout nový způsob tvorby náčrtů, ke kterým při výuce často dochází. Kresby mohou představovat různé prostorové geometrické útvary a tím zjednodušit výuku prostorové geometrie. Práce v teoretické části představuje problematiku víceuživatelského prostředí a popisuje techniky a technologie pro její řešení. Praktická část se zabývá návrhem a implementací aplikace a představuje vlastní přístupy k řešení. Na závěr jsou shrnuty výsledky a zpětné vazby uživatelů testujících aplikaci.

## **Annotation**

### **Title: Multi-user virtual reality environments**

Virtual reality is a phenomenon of today, which finds application across many industrial areas. This diploma thesis deals with the use of virtual reality as a means of insight into a shared multi-user environment. The main goal of the thesis is to come up with an innovative solution to simplify distance learning, which is currently a frequent consequence of the spreading of COVID-19. The shared environment should allow to facilitate the transfer of knowledge in the form of interactive elements present, including spatial drawing. Spatial drawing is intended to provide a new way of creating sketches that are often used during teaching. Drawings can represent different spatial geometric shapes and thus simplify the teaching of spatial geometry. Theoretical part of the thesis presents the issue of multi-user environment and describes techniques and technologies useful for the solution of the problem. Practical part deals with the design and implementation of the application and presents its own approaches to the solution. In the end, results and feedback of users testing the application are summarized.



# Obsah

1	Úvod.....	1
2	Sdílené virtuálního prostředí .....	3
2.1	Sdílená víceuživatelská prostředí ve VR .....	4
2.2	Zpoždění synchronizace .....	6
3	Techniky synchronizace víceuživatelského prostředí.....	8
3.1	Architektura distribuovaného systému.....	8
3.1.1	Synchronizace s autoritou na straně klientů .....	8
3.1.2	Serverově autoritativní přístup .....	8
3.2	Vzdálené volání procedur .....	9
3.3	Synchronizované proměnné .....	10
4	Technologie pro síťovou synchronizaci v Unity.....	11
4.1	Photon Unity Networking .....	11
4.2	Mirror.....	11
4.3	Unity Netcode .....	12
4.3.1	Komponenty Unity Netcode .....	12
4.3.2	Vzdálené volání procedur v Unity Netcode .....	13
4.3.3	Unity Transport Package .....	16
4.3.4	Unity Relay .....	17
5	Definice cílů a metodika zpracování .....	19
5.1	Cíl systému.....	19
5.2	Navrhované komponenty systému.....	20
5.2.1	Virtuální tabule.....	20
5.2.2	Prostorové kreslení .....	20
5.2.3	Komunikační konzole.....	20

5.2.4	Síťová komunikace.....	20
5.3	Výběr technologií .....	21
6	Návrh a implementace systému .....	22
6.1	Vytvoření projektu a prostředí virtuální scény .....	22
6.1.1	Implementace síťové komunikace.....	23
6.1.2	Uživatelské charaktery .....	23
6.1.3	Připojení k relaci .....	24
6.2	Interaktivní prvky .....	24
6.2.1	Okno pro zasílání zpráv.....	24
6.2.2	Nastavení scény.....	25
6.2.3	Tabule.....	26
6.2.4	Prostorové kreslení .....	27
6.2.5	Panel připojených klientů .....	28
6.3	Správci stavů aplikace .....	28
6.3.1	Správce kontextu .....	29
6.3.2	Správce oken .....	30
6.3.3	Správce synchronizovaných dat .....	30
6.3.4	Správci scény .....	30
7	Výsledky.....	32
7.1	Testování vývoje.....	32
7.2	Spuštění aplikace a připojení k relaci.....	33
7.3	Virtuální prostředí a jeho prvky .....	36
7.3.1	Pohyb ve scéně .....	37
7.3.2	Komunikační konzole.....	37
7.3.3	Ovládací panel.....	38
7.3.4	Prostorové kreslení .....	39

7.3.5	Tabule.....	41
7.3.6	Panel připojených klientů .....	43
7.4	Synchronizované prostředí více uživatelů .....	44
8	Závěry a doporučení .....	47
	Seznam použité literatury .....	49

## Seznam obrázků

Obrázek 1: Tok volání ServerRPC (Unity Technologies 2022e) .....	15
Obrázek 2: Tok volání ClientRPC (Unity Technologies 2022) .....	16
Obrázek 3: Komunikace přes Relay server (Unity Technologies 2022f) .....	18
Obrázek 4: Přechody a operace stavů.....	30
Obrázek 5: Okno pro připojení k relaci prostřednictvím Unity Relay .....	33
Obrázek 6: Okno pro připojení k relaci bez využití Unity Relay.....	34
Obrázek 7: Okno pro založení relace prostřednictvím Unity Relay .....	35
Obrázek 8: Panel výběru charakteru pro hostitele relace .....	36
Obrázek 9: Okno s komunikační konzolí .....	38
Obrázek 10: Panel pro nastavení vlastností fixů .....	39
Obrázek 11: První pohled na načrtnuté objekty prostorovým kreslením.....	40
Obrázek 12: Pohled z jiného úhlu na náčrty objektů .....	41
Obrázek 13: Pohled na tabuli a ovládací tlačítka.....	42
Obrázek 14: Rozdíl kreseb v závislosti na použití interpolace .....	43
Obrázek 15: Panel zobrazující připojené klienty.....	44
Obrázek 16: Pohled hostitele do scény prostřednictvím VR .....	45
Obrázek 17: Pohled do scény připojeného klienta .....	46

# 1 Úvod

Fenoménem dnešní doby objevujícím se ve stále více oblastech našeho života je virtuální realita, která na sebe strhává čím dál větší pozornost. Technologie virtuální reality se stále více rozšiřuje mezi běžné uživatele díky její větší dostupnosti a pomalu snižující se ceně. Tato technologie poskytuje nový způsob pohledu do prostředí virtuálních scén, který umožňuje přicházet s inovativními řešeními běžných činností. Jednou z těchto činností je i téma této diplomové práce, která se inspiruje aktuální realizací různých typů událostí distanční formou a snaží se přijít s inovativním řešením distanční výuky, která v posledních letech postihuje většinu studentů a ztěžuje předávání vědomostí v průběhu výuky.

Práce se zabývá problematikou sdílené víceuživatelské scény v prostředí virtuální reality. Cílem je navrhnout a implementovat prototyp aplikace řešící distanční výuku formou virtuální scény poskytující připojení prostřednictvím sítě. Výuka má probíhat ve sdíleném virtuálním prostředí, jehož stav bude synchronizován mezi veškerými jeho účastníky. Pro zajištění přístupu všem uživatelům je sdílené virtuální prostředí dostupné i pro uživatele desktopového zařízení, kteří nemusí disponovat hardwarem pro virtuální realitu.

Práce v teoretické části představuje uplatnění sdíleného virtuálního prostředí v herním průmyslu a medicíně. Uvedeny jsou problémy spojené se zpožděním synchronizace, kdy délka zpoždění hraje velkou roli při provádění jistých úkonů aplikace a zobrazování aktuální dat v reálném čase. Uvedeny jsou dostupné techniky a technologie pro realizaci synchronizace víceuživatelské scény společně s popisem konkrétních typů architektur distribuovaného systému a technologií zprostředkovávajících síťovou komunikaci.

Získané znalosti z teoretické části jsou využity při návrhu a tvorbě praktické části této práce. Cílem praktické části je realizovat aplikaci, která umožňuje připojení se klientům prostřednictvím rozdílných platforem do sdílené virtuální scény, která má nabízet interaktivní prvky poskytující předávání znalostí jako v běžné učebně. Návrh systému konkretizuje způsob připojení účastníků do virtuální scény a představuje jednotlivé interaktivní prvky dostupné ve virtuálním prostředí.

Výsledky realizované praktické části této práce ve formě aplikace jsou uvedeny v kapitole Výsledky společně s představením způsobu testování akcí, které jsou synchronizovány mezi připojenými uživateli. Závěr představuje shrnutí zpětných vazeb testujících uživatelů a uvádí chybějící funkce, které by mohly zlepšit efektivitu aplikace.

## 2 Sdílené virtuálního prostředí

V současné době se platformy virtuální reality (VR) stávají běžnou technologií a jsou dostupné mnoha uživatelům, což vede k vývoji velkého množství aplikací VR. Většina těchto aplikací však uživatelům různých hardwarových platform VR neumožňuje vzájemnou interakci ve společném sdíleném prostředí. To je způsobeno tendencí vývojářů navrhovat své aplikace pro konkrétní platformu, která drží většinu podílu na trhu, jako je HTC Vive a Oculus Rift. Některé aplikace VR jsou exkluzivní pro platformu, která se běžně uzavírá na základě vzájemně výhodné smlouvy mezi společností zabývající se platformou VR a vývojářem klientské aplikace. (Abbas et al. 2019)

Dalším důvodem, proč vývojáři neimplementují do svých aplikací podporu pro interakci více VR platform, je nutnost integrace knihovny pro zprostředkování komunikace mezi aplikacemi. Její integrací do projektu stoupá složitost na rozvržení a správu projektu. A je nutné zvážit, které herní prvky mají být v rámci sdíleného virtuálního prostředí synchronizovány a kdo nad nimi bude mít výslednou kontrolu.

Cílem sdíleného virtuálního prostředí je umožnit interakci v reálném čase vzdáleným uživatelům. Všechny zúčastněné strany mohou vstoupit do stejného sdíleného prostředí nezávisle na jejich geografické vzdálenosti. Prostředí může být ve formě libovolné scény tvořené modelem skutečného reálného místa nebo místem smyšleným, které je limitováno pouze představivostí jeho tvůrce. V prostředí scény mohou být realizovány různé druhy interakcí, které lze vyvolat jak uživatelem samotným nebo některou ze vzdálených zúčastněných stran. Interakce dále zvyšují důvěryhodnost reálného prostředí a nabízí rozšiřující možnosti úpravy jeho grafické nebo funkční stránky. (Shi et al. 2016)

Článek autorů (Abbas et al. 2019) se zabývá návrhem a implementací aplikace pro virtuální realitu, která umožňuje více uživatelům používat různé VR hardwarové platformy na samostatných počítačích ke vzájemné interakci v univerzálním virtuálním prostředí přes síť. Integrované virtuální prostředí je v souladu s konceptem bezproblémového propojení zařízení v síti internetu věcí (IoT), které také zahrnuje VR.

Aplikace se dále zabývá přístupem k řešení rozdílů mezi platformami VR, které spočívá ve vývoji jediné sady pro vývoj softwaru (SDK), které dokáže komunikovat mezi sadami SDK platformy VR a samotnou aplikací VR. To znamená, že vývojáři musí napsat kód aplikace pouze jednou a následně vytvořit rozhraní se sadou nástrojů SDK. Výstupem práce je aplikace poskytující síťově synchronizované interakce s tabulí, na kterou je umožněno kreslit fixem, se kterým lze interagovat prostřednictvím různých typů hardwarových platform virtuální reality (Abbas et al. 2019)

## **2.1 Sdílená víceuživatelská prostředí ve VR**

Systémy virtuální reality se využívají v mnoha průmyslových odvětvích. Své využití nachází v široké míře u zábavního a vojenského průmyslu. Kromě toho je tato technologie začleněna i v řadě dalších odvětvích. Často se jejím prostřednictvím realizují virtuální konference nebo prezentace prototypů různých objektů a výrobků. V případě vzdělávání virtuální realita umožňuje rekonstrukci historických událostí nebo již neexistujících či těžko dostupných míst.

V rámci víceuživatelského prostředí je virtuální realita často využívána v oblastech herního průmyslu, kdy je uživatelům her poskytnuta nová forma interakcí a nejsou omezeni vstupním rozhraním jako je klávesnice a myš. Víceuživatelská prostředí virtuální reality nabízí geograficky více vzdáleným uživatelům být účastníky společného virtuálního prostředí, ve kterém mohou provádět různé interakce. (Thalmann a Pandzic 1997)

Práce autorů (Thalmann a Pandzic 1997) popisuje návrh synchronizovaného virtuálního prostředí přístupného prostřednictvím VR. Každý z účastníků má vlastní kopii prostředí na svém lokálním zařízení, která je aktualizována při každé události, která v prostředí nastane pro zajištění konzistence mezi všemi účastníky. V jejich práci je představeno možné využití systému v mnoha oblastech. Mezi zmíněná využití patří prezentace 3D návrhů, virtuální telekonference nebo distanční vzdělávání a školení.

Víceuživatelské prostředí virtuální reality může být využito při prototypování designových návrhů. Zjednodušuje komunikaci a kolaboraci mezi spolupracovníky, urychluje samotný proces a snižuje jeho náklady. (Kan et al. 2001)



Sdílená víceuživatelská prostředí nachází své uplatnění i v medicíně v oblasti neurorehabilitace, kde byly dříve aplikovány hry pro jednoho hráče s pomocí robotů a virtuální reality při neurorehabilitační léčbě. A nedávno byly jako doplněk neuromuskulární terapie testovány hry pro více hráčů, od kterých se očekává, že budou motivovat pacienty a zvýší potenciál neuromuskulární terapie asistované roboty a virtuální realitou. Hry pro více hráčů zahrnují sociální interakci, která podporuje zábavu zúčastněných hráčů. Další hráč přidává do herního prostředí nové možnosti, které při hraní pro jednoho hráče proti předem naprogramovaným výzvám nebo uměle řízeným protivníkům obecně chybí. Prostředí pro více hráčů a související herní mechanismy mohou usnadnit sociální interakci, od konverzace až po haptické interakce. Díky této přidané sociální interakci se herní zážitek považuje za lepší ve hře pro více hráčů ve srovnání s hraním pro jednoho hráče. (Baur et al. 2018)

Aplikaci sdíleného virtuálního prostředí pro více uživatelů za použití VR lze najít v oblasti správy objektů budov, která musí zajistit správnou funkčnost vybudovaného objektu. Správa životního cyklu objektu je vysoce důležitá pro udržitelnost zastavěného prostředí. Ukazuje se, že nedostatek komunikace je vždy překážkou pro zlepšení obecného chápání potřeb spravovaného objektu. Požadavky mezi zainteresovanými stranami bez dobrého procesu následně ovlivňují jeho efektivitu. Důsledkem bývají pozdní reakce na potřeby údržby objektu, zbytečné plýtvání časem a zdroji. To dále může vést k tvorbě nepřátelských vztahů mezi koncovými stranami. (Shi et al. 2016)

Problém správy budov je v práci autorů (Shi et al. 2016) řešen vyvinutým nástrojem pro náhled do prostředí prostřednictvím VR. Vzdáleně umístěné zainteresované strany – obyvatelé budovy, správci zařízení, dodavatelé a projektanti – se mohou procházet ve stejné virtuální budově. Nástroj umožňuje interakce vzdálených zúčastněných stran v reálném čase a nabízí alternativní způsob komunikace pro efektivnější správu objektu. Pro vývoj nástroje bylo k vytvoření multiplatformního interaktivního prostředí využito Unity 3D a Photon Unity Networking (PUN) k implementaci funkce pro více uživatelů. Případová studie ukázala, že tato alternativní komunikační metoda může pomoci zúčastněným

stranám prozkoumat stejné zařízení ve vzdálených lokalitách a zlepšit efektivitu komunikace.

## **2.2 Zpoždění synchronizace**

Hry pro více hráčů, kde několik osob komunikuje současně prostřednictvím sítě, jako je internet, se těší velkému zájmu. Jedním z důvodů je soupeření s lidskými protějšky, které je obvykle považováno za mnohem zajímavější a náročnější, než hraní pouze proti počítači. Velkým problémem síťových her pro více hráčů je zpoždění síťového přenosu. To znamená zpoždění předávané informace, např. o pohybu objektů protivníka a jejich nové poloze, se dostanou k jejich příjemcům až po přijetí informace prostřednictvím sítě. Toto zpoždění způsobuje několik potíží a vede k nepříznivým situacím. (Pantel a Wolf 2002)

Příkladem takové nepříznivé situace může být závodní hra se dvěma hráči. Krátce po začátku se oba domnívají, že mají vedení, protože chvíli trvá, než se pozice protihráče dostane k místnímu hráči. V aktuální době existují přístupy k poskytnutí globálně konzistentního stavu hry prezentováním místních akcí se síťovým zpožděním. Nicméně tento přístup zvyšuje také zpoždění aplikace v reakci na zadaný vstup. Příkladem může být zatočení auta na dráze se zatáčkou, při čemž by díky zpoždění nezatočilo v požadovaný čas, ale až po zpoždění, díky kterému by zatáčku mohlo přejet. (Pantel a Wolf 2002)

Online hry pro více hráčů představují dnes jednu z nejpoblárnějších forem síťové skupinové komunikace na internetu a přispívají ke stále většímu podílu síťového provozu (McCreary a Claffy 2000)

Autor (Henderson 2001) ve své práci analyzoval a charakterizoval vztah mezi zpožděním na úrovni aplikace a chováním hráče v síťových hrách pro více hráčů. Zvyšující se odezvou síťového připojení k serveru zprostředkovávající hru více hráčů mají hráči větší tendenci tento server opustit a vyhledat jiný s rychlejším připojením nebo v některých případech může vysoká odezva vést k úplnému ukončení hraní. Povaha odehrávání se mnoha z těchto her v reálném čase znamená, že doba odezvy je důležitý faktor, kdy v síťovém prostředí musí být zpoždění omezeno na minimum. Maximální doporučené hodnoty odezvy pro zaručení dobré uživatelské zkušenosti závisí na typu synchronizované scény. Jedná-li se o scénu

s důrazem kladeným na rychlý a přesný pohyb hráčů s pohledem z první osoby jako je u online FPS (First Person Shooter) her. Maximální doporučená odezva by neměla překračovat hodnotu sta milisekund. U jiných typů scén jako jsou například strategie odehrávané se v reálném čase, se mohou hodnoty odezvy pohybovat až do maxima jedné sekundy, aby byla stále zaručena dobrá uživatelská zkušenost (Suznjevic a Saldana 2016). Z tohoto zjištění je patrné, že síťová odezva připojení je velice důležitá a ovlivňuje výslednou uživatelskou zkušenost s hrou.

## **3 Techniky synchronizace víceuživatelského prostředí**

Kapitola se věnuje aktuálním přístupům a technikám zprostředkovávající synchronizaci víceuživatelského prostředí. Zmíněny jsou typy architektur distribuovaného systému nebo způsoby synchronizace dat pomocí vzdáleného volání procedur a synchronizovaných proměnných.

### **3.1 Architektura distribuovaného systému**

Architektura je důležitý parametr systému, který určuje, jakým způsobem bude implementována jeho síťová komunikace. Rozdílné architektury mají odlišné strategie distribuce synchronizovaného stavu prostřednictvím zasílaných zpráv.

#### **3.1.1 Synchronizace s autoritou na straně klientů**

Podle autorů (Pantel a Wolf 2002) mohou počítačové hry pro více hráčů mít různé obecné rysy architektury. Mohou být dynamicky založené na serveru určeném hlavním serverem nebo s jedním centrálním serverem. Dalším přístupem je následovat distribuované schéma bez jakékoli centrální součásti udržující kompletní stav relace. Distribuci synchronizovaných dat lze provést multicastovou komunikací, kde každý z účastníků relace vypočítává svůj vlastní pohled na globální stav. Je zřejmé, že druhý přístup je lepší, pokud jde o snížení přenosového zpoždění, protože všechny datové pakety dorazí k příjemcům po přímé cestě.

Ve hře s autoritou na straně klienta rozhoduje o klientovi A jiný uzel než u klienta B, přičemž jsou oba od sebe zpožděny síťovým zpožděním. Příkladným problémem s autoritou na straně klienta je stav, kdy klient A střílí a zasahuje hráče B na uzlu A, zatímco hráč B se na jeho uzlu již schovává za krytem. Tento stav by zapříčinil inkonzistenci mezi klienty. Výše zmíněný problém v případě autority na straně serveru nenastane. Operace se vykonají sekvenčně v pořadí, ve kterém dorazí na server, čím je zajištěna konzistentnost. (Unity Technologies 2022e)

#### **3.1.2 Serverově autoritativní přístup**

U serverově autoritativního přístupu jsou všechna konečná rozhodnutí prováděna serverem. Autoritu má ten, kdo má právo činit konečná rozhodnutí o objektech ve scéně. Jednou z mnoha výhod přístupu s autoritou na straně serveru

je konzistence synchronizované scény, kdy všechna vykonávaná logika je přijímána stejným uzlem v síti (serverem). Rozhodnutí jsou přijímána a vykonávána v pořadí, ve kterém jsou serverem přijata. (Unity Technologies 2022e)

Je-li důležitá konzistentnost údajů v rámci relace a zajištění validací při změně dat, je přístup se serverovou architekturou vhodným řešením. Serverová architektura poskytne možnost využít server jako centrální autoritu, která má kontrolu nad změnami v rámci relace, které mohou být po jejich průběhu distribuovány všem klientům. (Unity Technologies 2022e)

Inkonzistenci dat při serverově autoritativním přístupu lze způsobit manipulací kódu na straně klienta. Klienti mohou padělat falešné síťové zprávy odeslané na server. I když v logice na straně klienta je určeno, že nelze provést interakci s předmětem, pokud je hráč od předmětu vzdálený více než 10 metrů. Je-li obsah zprávy zasílané serveru prostřednictvím vzdáleného volání procedur „proved' interakci“ a na straně serveru není žádná kontrola vzdálenosti od předmětu, klienti mohou tuto síťovou zprávu zfalšovat a tím obejít logiku na straně klienta. Z toho důvodu nelze klientům plně věřit a je vhodné, aby server měl logiku pro ověřování akcí přicházejících od klientů. Poté lze při přijetí zprávy na serveru zjistit, zda je klient dostatečně blízko předmětu pro interakci a operaci vykonat nebo při nesplnění podmínky vykonat alternativní logiku. (Unity Technologies 2022e)

### **3.2 Vzdálené volání procedur**

Jednou z možných technik pro synchronizaci distribuovaného virtuálního prostředí je vzdálené volání procedur (RPC). Jedná se o snadné a oblíbené paradigma pro vývoj distribuovaných aplikací, které je široce využíváno jako komunikační mechanismus v distribuovaných systémech. I když je vzdálené volání procedur koncepčně jednoduché a přímočaré, je s ním spojeno mnoho různých komplikací a problémů. Jednou z takových komplikací je serializace složitějších typů parametrů, které někdy musí být vyměněny za jiné parametry primitivního typu nebo je třeba zařídit jejich vlastní serializaci. V důsledku toho je k dispozici mnoho různých implementací RPC, a to jak ve výzkumu, tak v průmyslovém prostředí. (Tay a Ananda 1990)

### **3.3 Synchronizované proměnné**

Synchronizované proměnné udržují hodnotu proměnné v kódu, která je synchronizována mezi klienty distribuovaného systému. Její hodnoty jsou pravidelně replikovány na ostatní klienty připojené k serveru. Připojí-li se klient k serveru, všechny nejaktuálnější hodnoty jsou replikovány na tohoto nového klienta. Následně je stav aktualizován v pravidelných intervalech dojde-li ke změně její hodnoty. (Unity Technologies 2022b)

U tohoto způsobu synchronizace opět dochází ke komplikaci v podobě serializace složitějších typů proměnných, která nastává při zasílání její hodnoty prostřednictvím sítě. Při serializaci jiných než primitivních datových typů, je potřeba zařídit serializaci vlastní.

## 4 Technologie pro síťovou synchronizaci v Unity

Pro zajištění synchronizace mezi účastníky virtuálního prostředí je nutné realizovat komunikaci mezi jednotlivými účastníky a objekty, jejichž stavy mají být v rámci scény synchronizovány.

Unity engine disponuje řadou vestavěných funkcí pro usnadnění vytváření her pro více hráčů. Mezi tyto funkce patří například synchronizace stavů, real-time networking, vzdálené volání procedur a backendová konektivita. Díky integraci webových požadavků, je ve hře umožněno komunikovat s kontejnerovými webovými stránkami a službami. Bohužel tyto funkce jsou již vývojáři editoru Unity označeny za zastaralé a pro vývoj hry více hráčů může být vyžadováno přidání jednoho z několika dostupných softwarových balíčků třetích stran navržených pro tento účel. (Patil a Alvares 2015)

### 4.1 Photon Unity Networking

Jedním z balíčků je plugin Photon Unity Networking (PUN), který se používá k poskytování podpory více hráčů pro aplikace za pomoci síťové komunikace. Tento plugin obsahuje průvodce nastavením, plně integrovaná rozšíření pro editor a síťovou knihovnu. Je kompatibilní se službou Photon Cloud Service pro poskytnutí dedikovaného serveru. Vývojář může použít tuto službu nebo nastavit vlastní server pro svou aplikaci. (Patil a Alvares 2015)

Balíček Photon existuje ve dvou verzích – Photon PUN Free a Photon PUN+. Verzi Photon PUN Free lze stáhnout zdarma z Unity Assets Store. S tímto balíčkem může vývojář exportovat svou aplikaci pro desktopová zařízení nebo WebGL. Potřebuje-li exportovat aplikaci pro Android nebo iOS, je potřeba si zakoupit Photon PUN+ nebo verzi Unity Pro. (Patil a Alvares 2015)

### 4.2 Mirror

Dalším z balíčků poskytujících síťovou komunikaci pro herní engine Unity je knihovna Mirror. Jde o vysokoúrovňové síťové rozhraní pro Unity, které je dostupné zdarma jako open-source software a podporuje různé transportní vrstvy. Balíček pro synchronizaci stavu scény poskytuje synchronizované proměnné a zaslání příkazů formou vzdálených procedur. (Mirror Networking 2022)

## 4.3 Unity Netcode

Unity Netcode je síťová knihovna postavena pro herní engin Unity vyvíjená společností Unity Technologies, která nabízí internetovou komunikaci mezi objekty scény využívající transportní vrstvu Unity Transport Package. Jedná o novou knihovnu v rané fázi vývoje, která má za cíl nahradit stávající zastaralé řešení UNET.

Její součástí jsou struktury pro volání vzdálených procedur společně s proměnnými, které jsou synchronizovány po síti. Tyto proměnné mohou být dvojího typu, jako jednoduchá sdílená proměnná nebo seznam proměnných určitého typu. Integraci Unity Netcode lze zajistit síťovou komunikaci mezi jednotlivými klienty relace napříč platformami a tím umožnit crossplay. Crossplay je moderní termín využívaný zejména v herním průmyslu, ale i v softwarovém vývoji, kde je poskytnuta komunikace napříč rozdílnými platformami. Knihovna Unity Netcode zajišťuje podporu pro modely host a listen-server, které využívají serverově autoritativního přístupu. (Unity Technologies 2022a)

Společně s knihovnou je vydán zhotovený open-source projekt Boss Room, ve kterém je knihovna Unity Netcode implementována za účelem demonstrace jejích dostupných funkcí. Ukázkové řešení má sloužit uživatelům k pochopení užitých principů síťové knihovny a inspiraci pro jejich vlastní projekty. Boss Room je v době psaní této práce oficiálně v předběžné verzi a má být nadále vyvíjen a podporován společně s vývojem knihovny Unity Netcode.

### 4.3.1 Komponenty Unity Netcode

Součástí knihovny je celá řada komponent připravených k užití. Následující popis zahrnuje nezbytné komponenty pro konfiguraci a zajištění funkčnosti včetně nejčastěji využívaných komponent pro zprovoznění síťové synchronizace objektů scény.

#### **NetworkManager**

NetworkManager je hlavní komponenta spravující síťovou komunikaci v rámci projektu. Využívá návrhový vzor Singleton pro snadný přístup k instanci její třídy z ostatních skriptů, které mohou potřebovat ke své funkčnosti její referenci.



## **NetworkTransport**

Jedná se o transportní vrstvu využívanou komponentou NetworkManager. Komponenta zpracovává předaná data od NetworkManageru a posílá je určeným uzlům v síti. Nebo naopak přijímá a zpracovává přijatá data od ostatních uzlů a předává je NetworkManageru.

## **NetworkObject**

Komponenta NetworkObject přidává objektu, se kterým je spojena jeho unikátní síťovou identitu, díky které je tento objekt synchronizován mezi veškerými klienty připojeními k serveru.

## **NetworkTransform**

NetworkTransform zajišťuje synchronizaci polohy, natočení a škály objektu, ke kterému je připojena. Pro správnou funkčnost této komponenty je nutné, aby její objekt nebo některý z předků, měl síťovou identitu v podobě připojené komponenty NetworkObject.

## **NetworkBehaviour**

NetworkBehaviour je abstraktní třída dědicí z třídy MonoBehaviour, která umožňuje tvorbu komponenty s vlastní logikou. Třída NetworkBehaviour jejímu předkovi přidává atributy a vlastnosti využitelné pro tvorbu síťové logiky nad touto třídou. Mezi přidané prvky patří informace o tom, zda aktuální klient je majitel komponenty nebo zda se jedná o klienta či server. Jelikož třída dědí z třídy MonoBehaviour, lze jí spojit s libovolným objektem ve scéně a užít ji jako jeho komponentu. Podobně jako NetworkTransform i třídy zděděné z NetworkBehaviour pro správnou funkčnost potřebují, aby objekt, se kterým je spojena nebo některý z předků, měl síťovou identitu v podobě přidělané komponenty NetworkObject.

### **4.3.2 Vzdálené volání procedur v Unity Netcode**

V závislosti na serverově autoritativním přístupu má Unity Netcode k dispozici dvě varianty vzdáleného volání procedur (RPC) pro provádění logiky na

straně serveru nebo na straně klienta. Při volání RPC je volána metoda na objektu vzdáleného zařízení, který by mohl být kdekoli na světě. Jedná se o události, které sami o sobě neudrží žádný stav a lze je v případě potřeby vzdáleně spustit. (Unity Technologies 2022)

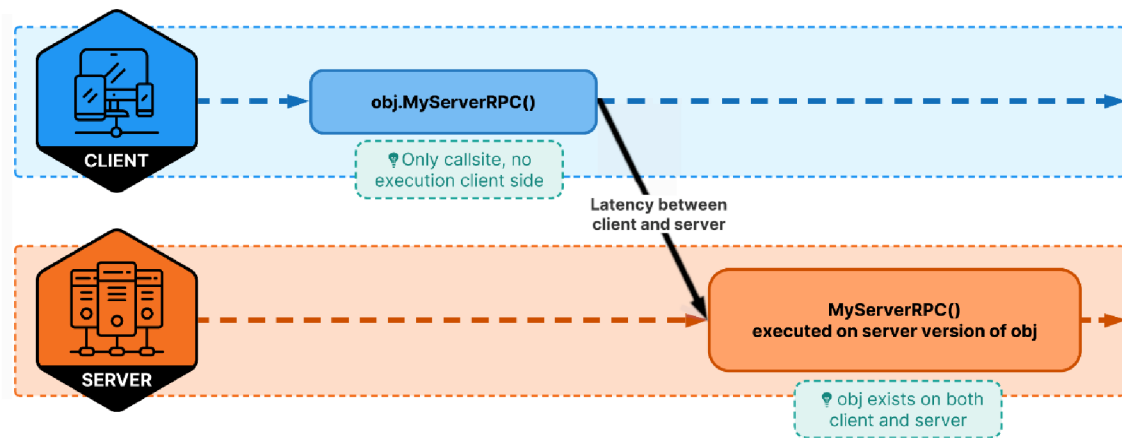
Na vysoké úrovni si balíček Unity Netcode při volání RPC ze strany klienta zaznamená objekt, komponentu, metodu a všechny parametry potřebné pro toto vzdálené volání a odešle tyto informace přes síť. Server informace přijme, najde zadaný objekt a zadanou metodu na odpovídající komponentě a zavolá ji s přijatými parametry.

Pro zavolání klientské nebo serverové vzdálené procedury je potřeba aby procedura byla volána z třídy dědicí z `NetworkBehaviour` a aby její objekt nebo některý z předků měl připojenou komponentu `NetworkObject`.

## **ServerRPC**

Jedním ze dvou druhů vzdáleného volání procedur je `ServerRPC`. Jedná se o proceduru, u které je na straně volajícího klient a na straně přijímajícího server. V této situaci klient předává v případě potřeby parametry pro vzdálené volání a inicializuje její vykonání na síťovém objektu. RPC je následně umístěna do místní fronty a poté odeslána na server, kde bude spuštěna na serverové verzi stejného síťového objektu. (Unity Technologies 2022)

# Server RPCs

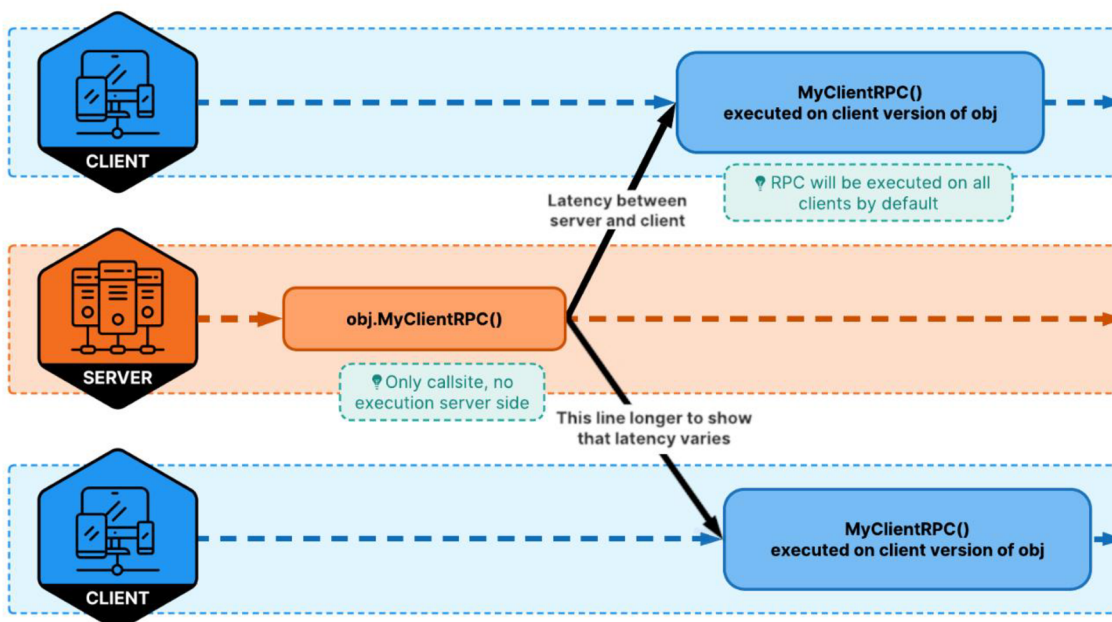


Obrázek 1: Tok volání ServerRPC (Unity Technologies 2022e)

## ClientRPC

Druhým typem vzdáleného volání je ClientRPC. ClientRPC je obdoba ServerRPC s rozdílem směru volání, kdy volající je server a na straně přijímajícího se nachází klient, který proceduru vykonává na svém síťovém objektu. Ve výchozím nastavení je přijímajícím každý klient, který je v době volání připojen k serveru. V případě nutnosti vykonat vzdálenou proceduru pouze na určitých klientech, lze vzdálené proceduře předat parametr určující konkrétní klienty, kteří mají být na straně přijímajícího. (Unity Technologies 2022)

# Client RPCs



Obrázek 2: Tok volání ClientRPC (Unity Technologies 2022)

## 4.3.3 Unity Transport Package

Unity Transport Package (UTP) je transportní vrstva umožňující síťovou komunikaci mezi aplikací a různými hostiteli v síti. Jedná se o výchozí transportní vrstvu balíčku Unity Netcode, pomocí které lze přidat multiplayerové nebo síťové funkce do projektu nezávisle na nutnosti přidávat balíček Unity Netcode. V případě potřeby je možné kromě výchozího řešení UTP získat komunitní transportní vrstvu nebo naimplementovat a použít vlastní. (Unity Technologies 2022d)

Transportní vrstva může poskytovat komunikaci založenou na spojení k zajištění robustního připojení před výměnou dat pomocí protokolu handshake. Dokáže spravovat pořadí doručení paketů za účelem opravy nesrovnalostí v případě výpadků paketů a zajistit integritu dat vyžádáním opětovného přenosu chybějících nebo poškozených dat pomocí kontrolních součtů. Přizpůsobuje datové toky pro přenos jako bajtové toky nebo pakety. Dále může spravovat přetížení sítě zprostředkováním průtoků a přetížení uzlů nebo ovládat datový tok síťového

připojení pro zabránění přetečení nebo podtečení vyrovnávací paměti, aby se předešlo zbytečným problémům s výkonem sítě. (Unity Technologies 2022d)

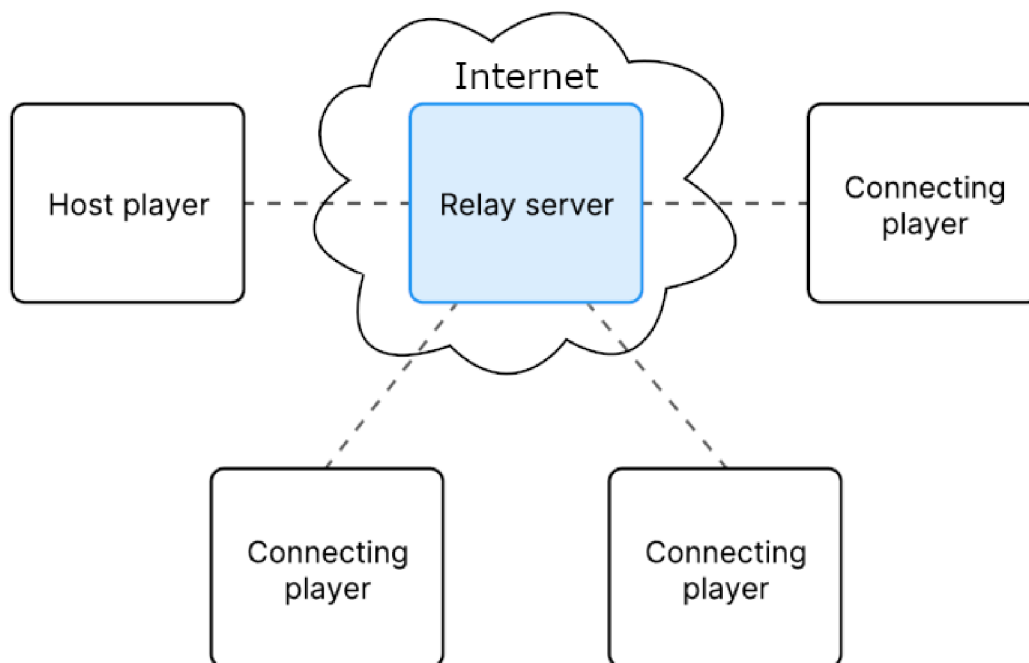
#### **4.3.4 Unity Relay**

Unity Relay je technologie v současné době v předběžném beta testování nabízející vývojářům způsob, jak bezpečně zajistit zvýšenou konektivitu mezi klienty, aniž by museli investovat do řešení třetích stran. Umožňuje vývojářům propojit klienty pomocí naslouchacího serveru. Namísto vyhrazených serverů poskytuje služba Relay konektivitu prostřednictvím univerzálního Relay serveru fungujícího jako proxy. (Unity Technologies 2022f)

Služba Relay usnadňuje podporu pro více hráčů bez vyhrazených herních serverů tím, že umožňuje hráčům komunikovat mezi sebou prostřednictvím serverů Relay. Relay servery doručují zprávy mezi připojenými hráči. Žádní dva hráči se nikdy nespojí přímo k sobě, veškerá komunikace probíhá přes Relay server.

Služba Relay má dvě klíčové součásti: Relay servery a službu Relay Allocations. Relay servery fungují jako veřejné koncové body dostupné všem hráčům. Tento přístup řeší běžné problémy se změnou sítě a IP adres, překlad síťových adres (NAT) a firewally mezi hráči. Každý hráč se může připojit ke stejné IP adrese a portu (IP vybraného Relay serveru) a herní klienti se mohou spolehnout, že informace o připojení zůstanou během herní relace stejné. Prostřednictvím této nepřímosti si hráči ve hře nemusí být vědomi své IP adresy, čímž se zvyšuje bezpečnost a soukromí. (Unity Technologies 2022f)

Služba Allocations funguje na backendu a umožňuje klientům vytvářet a připojovat se k relacím. Slouží jako vstupní bod pro interakci se službou Relay. Odhaluje veřejně přístupné API, které umožňuje klientům alokovat slot na Relay serveru. Klienti poté mohou využít přidělený Relay server ke komunikaci s ostatními klienty.



Obrázek 3: Komunikace přes Relay server (Unity Technologies 2022f)

Užití služby Unity Relay je pro správnou funkci podmíněno nastavením Unity Transport Package jako transportní vrstvy. Unity Relay rozšiřuje funkcionalitu UTP a přidává možnost připojit klienty, kteří by jinak nebyli schopni komunikovat kvůli omezením směrování, jako jsou omezující brány firewall. (Unity Technologies 2022f)

## 5 Definice cílů a metodika zpracování

Na základě znalostí, přístupů a technologií získaných z rešerše popsané v předchozích kapitolách bude v této kapitole definován cíl práce a odůvodněn výběr užitých technologií.

### 5.1 Cíl systému

Vzhledem k současné době, kdy je svět zahlcen nemocí COVID-19 přechází v mnoha případech řada činností, u kterých člověk může přijít do fyzického kontaktu s cizí osobou, do bezkontaktního režimu. Jedním z takových případů je výuka, která přešla na distanční režim a ztížila tím její průběh při předávání informací studentům. Z tohoto důvodu se práce zaměřuje na tvorbu prototypu aplikace, jejíž cílem je ověřit, jaké jsou možnosti víceuživatelské scény s klienty připojujícími se z rozdílných platforem. Aplikace by měla zjednodušit výuku formou virtuální učebny s interaktivními prvky umožňujícími předávání znalostí jako v běžné učebně.

Víceuživatelské prostředí má za cíl studentům poskytnout připojení se ze svých domácností pomocí desktopových zařízení do virtuální scény, která bude vytvořena za účelem efektivnějšího zprostředkování výuky. Jediným uživatelem, který bude mít umožněn přístup do scény prostřednictvím virtuální reality, bude hostitel síťové relace, kterým by měl být ve většině případů vyučující. Je to z důvodu, aby připojení klienti nemuseli disponovat nákladným hardwarem pro virtuální realitu. Vyučující bude mít možnost hostovat relaci a přistoupit do ní formou virtuální reality poskytující jednoduché interakce. Jeho pohyb bude v rámci virtuální scény realizován prostřednictvím teleportace. Studenti se budou moci do scény připojit jako klienti formou charakteru s pohledem z první osoby (FPS charakter), který se bude moci volně pohybovat po scéně.

## **5.2 Navrhované komponenty systému**

### **5.2.1 Virtuální tabule**

Vyučující bude mít ve scéně k dispozici virtuální tabuli jejíž změny obsahu budou synchronizovány napříč všemi připojenými klienty relace. Interakce s ní budou realizovány příslušným objektem v podobě fixu. Virtuální tabule bude podobně jako klasická tabule sloužit k výuce pro náčrty a kresby vysvětlující vyučovanou látku jako v klasické učebně.

### **5.2.2 Prostorové kreslení**

Virtuální prostředí bude poskytovat jeden objekt v podobě fixu, který bude umožňovat prostorové kreslení. Prostorové kreslení bude vhodné pro obrazce a geometrické útvary, které je snazší vyobrazit v prostoru namísto projekce na rovinu v podobě tabule. Příkladným využitím může být zjednodušená výuka prostorové geometrie.

Vlastnosti obou objektů reprezentující fixy budou moci být upraveny za běhu aplikace. Možná bude například výměna barvy, kterou mohou fixy kreslit.

### **5.2.3 Komunikační konzole**

Součástí realizované aplikace bude komunikační konzole jejíž prostřednictvím bude možné zasílat zprávy mezi všemi účastníky, kteří jsou přítomní ve virtuálním prostředí. Komunikační konzole bude reprezentována jednoduchým oknem nabízející zadání požadovaného textu zprávy k odeslání. Prostřednictvím tohoto okna bude poskytnut náhled, na již zaslané nebo přijaté zprávy.

### **5.2.4 Síťová komunikace**

Síťová komunikace bude uskutečněna pomocí knihovny Unity Netcode. K síťové knihovně bude připojena služba Unity Relay pro usnadnění vytvoření relace. Za jejího využití nebude potřeba disponovat veřejnou IP adresou nebo mít k dispozici dedikovaný server. Prostřednictvím služby Unity Relay bude poskytnuto jednoduché připojení k relaci formou šestimístného alfanumerického kódu, který



bude vygenerován hostiteli relace. Pro připojení klienta k relaci bude třeba pouze předat tento kód do příslušného pole připojovacího formuláře.

### **5.3 Výběr technologií**

Výběr technologie pro synchronizaci byl ovlivněn návazností této diplomové práce na bakalářskou práci s tématem Virtuální realita v Unity3D. Původní práce byla vytvořena v herním enginu Unity, což je multiplatformní vývojové prostředí pro tvorbu virtuálních 2D nebo 3D scén. Proto i tato práce bude herní engine Unity využívat k sestavení scény a demonstraci přístupů a interakcí ve virtuálním prostředí. Unity se v bakalářské práci velmi dobře osvědčilo pro tvorbu aplikace s podporou virtuální reality a díky předchozím zkušenostem s tímto vývojovým prostředím bylo zvoleno i pro praktickou část této práce.

Další z technologií potřebné k realizaci cíle praktické části této práce, je technologie umožňující komunikaci v rámci distribuovaného systému. Po prozkoumání dostupných možností byla zvolena knihovna Unity Netcode, která je přímo vyvíjena společností vyvíjející herní engine Unity. Jedná se o síťovou knihovnu, která je v době psaní této práce stále v aktivní vývojové fázi. Zvolena byla i navzdory tomuto faktu, protože poskytuje podporu další implementované službě Unity Relay, která nabízí založení relace hostujícímu klientovi a snadné připojení k relaci formou šestimístného alfanumerického kódu jejím klientům. Kombinace těchto dvou knihoven byla zvolena pro vývoj praktické části projektu právě z důvodu snadné formy připojení k relaci nevyžadující potřebnou síťovou infrastrukturu.

SteamVR Plugin je další zvolenou technologií zdarma dostupnou v obchodě Unity Asset Store. SteamVR nabízí jednoduchou správu a manipulaci se vstupy ovladačů virtuální reality společně s komponentami umožňující různé druhy interakcí, mezi které patří například tlačítka, posuvníky a uchopitelné objekty. Všechny tyto interaktivní objekty poskytují interakce prostřednictvím ovladače virtuální reality. (Valve Corporation 2022)

## 6 Návrh a implementace systému

Kapitola se zabývá samotným návrhem systému propojující desktopové stanice s počítačem s instalovaným zařízením pro virtuální realitu, a tak umožňující interakci všech uživatelů. Popisuje kroky, které byly potřeba provést při vytváření projektu, modelování virtuální prostorové scény a pro řešení jednotlivých funkcionalit dostupných ve výsledné aplikaci. Představeny jsou veškeré interaktivní prvky scény, spolu s jejich ovládáním a principy využívání připojených systémových komponent.

### 6.1 Vytvoření projektu a prostředí virtuální scény

Projekt je vytvořen v enginu Unity za použití šablony Universal Render Pipeline (URP). Jedná se o šablonu Unity projektu disponující jednoduchou vykreslovací technikou s podporou zařízení virtuální reality, které vykreslují veliký počet obrazových bodů, ale i méně výkonných zařízení, jako jsou mobilní telefony.

Do projektu byla integrována knihovna SteamVR, která nabízí různé interaktivní prvky ve formě uchopitelných objektů, tlačítek a posuvníků pro interakce pomocí ovladačů VR. Implementace je realizována prostřednictvím dodávaných komponent, které lze připojit k objektu scény a tím objektu přidat nabízené vlastnosti a chování. Projekt dále z knihovny SteamVR využívá techniku přemístění teleportem pro hostitele relace s virtuální realitou nebo tlačítka společně s posuvníky, které mají předem nastavené rozmezí hodnot a poskytují interakci prostřednictvím kontaktu s ovladačem virtuální reality.

Do uživatelského rozhraní bylo vytvořeno okno komunikační konzole nabízející zaslání zpráv mezi klienty scény. Okno je složeno z pole pro zadání textu zprávy a tlačítka pro odeslání zprávy. Odeslané zprávy se následně přidávají do panelu nad polem pro zadání textu zprávy, který má umožněn posun pro zobrazení starších zpráv v konverzaci. Druhým prvkem uživatelského rozhraní je panel s informacemi o připojených uživateli. Panel je volně umístěn do prostoru scény namísto obvyklého překrytí pohledu kamery, jako je tomu u komunikační konzole. Detailnější popis těchto objektů je v následujících podkapitolách.

### **6.1.1 Implementace síťové komunikace**

Pro implementaci síťové komunikace prostřednictvím knihovny Unity Netcode je potřeba do projektu zaintegrovat všechny její závislosti. Následně je ve scéně vytvořen nový objekt, kterému je přidána komponenta NetworkManager poskytující konfiguraci knihovny pro síťovou komunikaci. Pro zajištění funkčnosti služby Unity Relay je konfiguraci změněna transportní vrstva na Unity Transport, která je využita pro pozdější integraci připojení k serveru prostřednictvím kódu relace.

Integrace služby Unity Relay vyžaduje inicializaci služeb Unity a přihlášení se k autentizační službě alespoň formou anonymního přihlášení. Funkčnost Unity Relay může být následně rozdělena na dvě části. Jedna sloužící pro alokování nové relace, kterou v aplikaci může využívat hostující klient. Druhá zprostředkovává připojení se ke stávající relaci formou šestimístného kódu relace. Alokování nové relace poskytuje parametry pro výběr regionálních serverů, přes které bude komunikace procházet a volbu maximálního počtu souběžně připojených klientů do maxima sta klientů.

Nakonfigurováním projektu pro umožnění síťové synchronizace se pro každého připojeného klienta vytvoří vybraný charakter na serveru. To je dáno serverově autoritativním přístupem, jelikož nad synchronizovanou scénou má hlavní kontrolu server (v případě projektu hostující klient) a připojení klienti mohou pouze zasílat serveru požadavky na změnu synchronizované scény.

### **6.1.2 Uživatelské charaktery**

Virtuální scény se mohou účastnit dva typy charakterů. Charakter s pohledem z první osoby (FPS charakter) se smí volně pohybovat po virtuálním prostředí a otáčet se prostřednictvím myši. Tento charakter se ve scéně vyobrazuje jako kapsle s nastavenými kolizními zónami a jeho pohyb je ovládán za pomoci klávesnice.

Druhý typ charakteru, který může využívat pouze hostitel scény, je skupina objektů umožňující vstup do scény prostřednictvím virtuální reality (XR charakter). Jeho pohyb je v prostředí scény realizován formou teleportačního mechanismu poskytnutého knihovnou SteamVR.

### **6.1.3 Připojení k relaci**

Z pohledu uživatele je nutné po spuštění aplikace vybrat typ připojení. Prostřednictvím zobrazeného okna se aplikace dotáže uživatele, zda chce být připojen k nějaké ze stávajících relací nebo zda chce relaci založit novou. V závislosti na jeho výběru se zobrazí další okno upřesňující zvolenou možnost. Po zadání uživatelského jména nabídne aplikace volbu typu uživatelského charakteru v závislosti na typu připojení uživatele. Potvrzením se okno uzavře a aplikace uživateli vytvoří charakter, jehož prostřednictvím je mu poskytnut náhled do scény.

Výběr uživatelského charakteru je poskytnut pouze hostiteli relace, ten si smí vybrat mezi FPS charakterem, který se smí pouze volně pohybovat po virtuálním prostředí a zasílat zprávy prostřednictvím komunikační konzole nebo XR charakterem podporujícím virtuální realitu, kterému je dovoleno provádět akce s interaktivními prvky scény společně se zasíláním zpráv prostřednictvím komunikační konzole. Ostatním klientům, kteří jsou připojeni k hostiteli relace je automaticky přidělen FPS charakter.

## **6.2 Interaktivní prvky**

Virtuální scéna poskytuje množství interaktivních prvků nabízející rozdílné způsoby interakce. Jejich prostřednictvím lze v rámci scény vyvolat různé funkce ovládající vizuální nebo technickou stránku aplikace. Tato podkapitola se zabývá jejich popisem a vysvětlením principu jejich fungování. Popsány jsou jejich činnosti jak v rámci lokálního klienta, tak i v rámci synchronizované relace.

### **6.2.1 Okno pro zasílání zpráv**

Komunikační konzole je jedním z dalších interaktivních prvků přítomných ve scéně. Je zobrazena ve formě okna v levém spodním rohu okna aplikace a nabízí zasílání zpráv všem dalším klientům relace. Samotné okno se skládá ze dvou částí, které jsou umístěny pod sebou. Spodní z nich je složena ze dvou komponent, textového pole pro zadání textu zasílané zprávy a tlačítka pro její odeslání. Druhou horní částí okna je plocha s posuvníkem. Do které jsou zprávy přidávány a seřazeny pod sebe komponentou `VerticalLayout`. Posuvník slouží k vertikálnímu posunu plochy pro možnost zobrazení si starších zpráv v okně.

Zasílání zpráv je umožněno pouze prostřednictvím grafického uživatelského rozhraní aplikace. Rozhraní je renderováno pouze v okně aplikace a prostřednictvím brýlí virtuální reality ho nelze spatřit. Navíc aplikace neimplementuje žádný ze způsobů zadávání textu prostřednictvím VR, z toho důvodu je nutné si brýle VR sundat a až poté interagovat s oknem komunikační konzole prostřednictvím klávesnice.

Zasílané zprávy obsahují informaci o času odeslání, jméno odesílatele s identifikačním číslem relace a obsah samotné zprávy. Objekt zobrazované zprávy se skládá ze dvou potomků disponujících textovou komponentou. Pro přehlednou reprezentaci jsou tyto potomci horizontálně zarovnány vedle sebe pomocí komponenty `HorizontalLayout`. První z potomků zobrazuje čas poslané zprávy. Druhý z potomků vyobrazuje jméno odesílatele a jeho identifikační číslo spolu s textem zasílané zprávy.

Dojde-li k odeslání zprávy některým z účastníků relace. Zjistí se informace o lokálním klientovi v podobě jeho jména a identifikačního čísla. Tyto informace se společně s obsahem zprávy získaného z textového pole předají třídě obsluhující logiku pro zasílání zpráv mezi klienty relace. Třída předaná data zpracuje a vyvolá vzdálené volání procedury na serveru formou `ServerRPC` z knihovny `Unity Netcode`. Při přijetí dat na straně serveru dojde pouze k vyvolání další vzdálené procedury formou `ClientRPC`, která se odešle všem účastníkům probíhající relace. Každý klient si při přijetí tohoto vzdáleného volání zpracuje přijatá data, zkonstruuje z nich záznam zprávy a zobrazí jej v lokálním okně každého klienta.

### **6.2.2 Nastavení scény**

Ve scéně jsou přítomny dva panely, které umožňují úpravu nastavení jednotlivých fixů nebo kontrolu nad objekty přidanými do scény. První z panelů poskytuje přizpůsobení funkcionalit fixů dostupných ve virtuálním prostředí. Panel disponuje několika dotykovými plochami, dvěma posuvníky a jedním přepínačem.

Dotykové plochy poskytují výběr barvy, pro aktuálně držžený fix, kterou s ním lze kreslit. Změna barvy fixu se projeví po jeho uchycení do ruky pomocí ovladače virtuální reality a kontaktu plochou reprezentující vybranou barvu. Kromě změny barvy, kterou bude fix následně kreslit, se patřičně změní i barva jeho hrotu.

Posuvníky na panelu slouží pro úpravu parametrů prostorového fixu. Jeden z posuvníků nastavuje šířku, kterou bude mít příští nakreslená čára. Druhý z posuvníků mění minimální vzdálenost, po které je umožněno vykreslení dalšího bodu čáry. Jediným přepínacím tlačítkem na panelu je tlačítko přepínající zapnutí a vypnutí funkce interpolace pro fix umožňující kreslení na tabuli.

Druhý z panelů disponuje čtyřmi tlačítky, která jsou vyobrazena na obrázku 13. Jejich interakce je umožněna při kontaktu s ovladačem a stisknutím tlačítka spouště. Po interakci se spustí patřičná funkce nadepsaná na tlačítku. Mezi funkce tlačítek patří například smazání všech prostorových kreseb ve scéně nebo přepnutí viditelnosti panelu s připojenými klienty pro všechny účastníky relace vyjma hostitele.

### 6.2.3 Tabule

V rámci scény je jedním z interaktivních prvků tabule, na kterou je umožněno kreslit prostřednictvím fixu, k tomu určeného. Fix je možné při kontaktu s ovladačem uchopit patřičným tlačítkem do ruky a kreslit s ním na tabuli.

Funkcionalita tabule potom spočívá ve zjištění UV souřadnic do textury tabule pomocí vrženého paprsku z pozice hrotu fixu. Paprsek je vržen směrem špičky hrotu do maximální vzdálenosti dvou centimetrů virtuální scény. Při zasažení objektu se zjistí, zda zasažený objekt má na sobě přítomnou komponentu, která umožňuje úpravu textury a v kladném případě se komponentě předá zjištěná UV souřadnice. Po přijetí UV souřadnice komponenta vypočte pixelové souřadnice do textury a navrátí je formou seznamu pixelů. Tento přístup je výhodný pro pozdější využití interpolace kreslené čáry a pro rychlou synchronizaci mezi klienty. V případě zapnuté interpolace se zjistí, zda v předchozím snímku bylo kresleno na tabuli. Pokud bylo, vykreslí se lineárně interpolovaná čára z předchozí pozice kontaktu s tabulí do aktuální pozice kontaktu.

Dojde-li k obarvení pixelů tabule, je zavolána metoda pro přidání pixelů do fronty pro synchronizaci. Po přidání pixelů na konec fronty se v každém synchronizačním cyklu vezme patřičný počet pixelů, které se přidají do listu pixelů, který je synchronizován mezi všemi klienty scény. Patřičný počet pixelů se vybírá pro nezahlcení sítě a transportní vrstvy. Následně při přidání prvků do

synchronizovaného listu server notifikuje veškeré klienty o jeho změně a ty si v reakci na tuto skutečnost aktualizují svůj lokální stav a překreslí tabuli podle aktuálního stavu.

#### **6.2.4 Prostorové kreslení**

Další z možných interakcí ve scéně, je využití fixu pro prostorové kreslení. Fix po uchopení do ruky prostřednictvím ovladače virtuální reality poskytuje funkci kreslení v prostoru pomocí snímání jeho pozice, která následuje pozici ovladače v rámci virtuální scény. Po uchopení fixu, stisknutí a držení tlačítka spouště je možné začít kreslit prostorovou čáru.

Funkce prostorového kreslení spočívá v získání prostorové pozice hrotu fixu v každém snímku, je-li drženo tlačítko spouště. Jedná-li se o první pozici, je pozice přidána do listu s prostorovými body. Jinak je nejdříve zjištěna vzdálenost od poslední pozice a je-li od ní nová pozice vzdálena na větší vzdálenost, než je nastavena prahová hodnota, je přidána do listu k předchozím bodům, jinak je zahozena. Prahová hodnota je metodě předána prostřednictvím atributu minimální vzdálenosti pro vykreslení dalšího bodu, který je možno změnit ve scéně při běhu aplikace.

List bodů v prostoru je dále předán komponentě `LineRenderer` poskytovanou enginem Unity a za její pomoci jsou body vykresleny do scény. Z důvodů omezení přítomnosti pouze jedné komponenty `LineRenderer` na jednom objektu scény je nutné, aby každá prostorová kresba měla vytvořený vlastní objekt.

Implementace síťové synchronizace v případě prostorového fixu je méně komplikovaná oproti kreslení na tabuli. V tomto případě je potřeba před započítím kreslení vytvořit nový objekt s komponentou `LineRenderer` a vlastní komponentou udržující síťově synchronizovaný list bodů v prostoru, které mají být vykresleny. Poté stačí při kreslení získat referenci na vytvořený objekt a předat synchronizovanému listu body, které mají být vykresleny. Tím je docíleno stavu, kdy po každé změně synchronizovaného listu, jsou notifikováni klienti připojení k relaci a v reakci na notifikaci si překreslí vlastní lokální prostorovou čáru.

### **6.2.5 Panel připojených klientů**

Panel připojených klientů slouží k evidenci informací o připojených klientech v aktuální relaci. Každý záznam na panelu reprezentuje jednoho připojeného klienta. Záznam je složen ze zvoleného jména při připojení, přiděleného identifikačního čísla relace a času, kdy se klient k relaci připojil.

Panel má nastavenou formu vyobrazení v prostoru scény namísto klasického překrytí obrazu aktuálně vykreslované kamery. Tato skutečnost umožňuje panel volně umisťovat a natáčet v prostoru scény. Vlastnosti je využito formou vlastní komponenty, která po získání reference na objekt charakteru lokálního klienta rotuje panelem podél svislé osy. Hodnota rotace je v každém snímku běžící aplikace nastavena tak, aby panel byl otočený čelem k lokálnímu charakteru. Tato vlastnost panelu poskytuje každému z klientů přehlednější pohled na seznam připojených uživatelů a nevyžaduje přesun charakteru k pohledu na panel z jiného úhlu.

Jednotlivé záznamy panelu jsou tvořeny dvěma potomky s komponentou pro zobrazení textu a uspořádány komponentou `HorizontalLayout`. Záznamy jsou zároveň spravovány komponentou pro vertikální uspořádání, která patřičně nastavuje šířku každému ze záznamů. Při přidání se nové záznamy řadí chronologicky pod sebou podle času jejich přidání.

Záznamy jsou do panelu přidávány při každém připojení nového klienta a odstraněny při jeho odpojení. Pro synchronizaci napříč všemi klienty jsou jednotlivé informace o každém z nich ukládány na serveru do seznamu synchronizovaného po síti. S tímto seznamem může manipulovat pouze server, který právě při připojení nebo odpojení záznamy přidává nebo odebírá. Při každé změně tohoto seznamu jsou všichni klienti o této události upozorněni a v reakci na ní si aktualizují svůj lokální stav, tím je zaručena aktuálnost zobrazovaných informací u každého z klientů.

### **6.3 Správci stavů aplikace**

Za účelem kontroly a správy stavů aplikace, bylo do projektu potřeba naimplementovat několik typů správců pro správu jednotlivých systémů v aplikaci. Třídy těchto správců se v aplikaci vždy nachází v podobě maximálně jedné instance a ve většině případů implementují návrhový vzor `Singleton` pro usnadnění získání reference na jejich jedinou instanci a tím využití jejich poskytovaných funkcí.



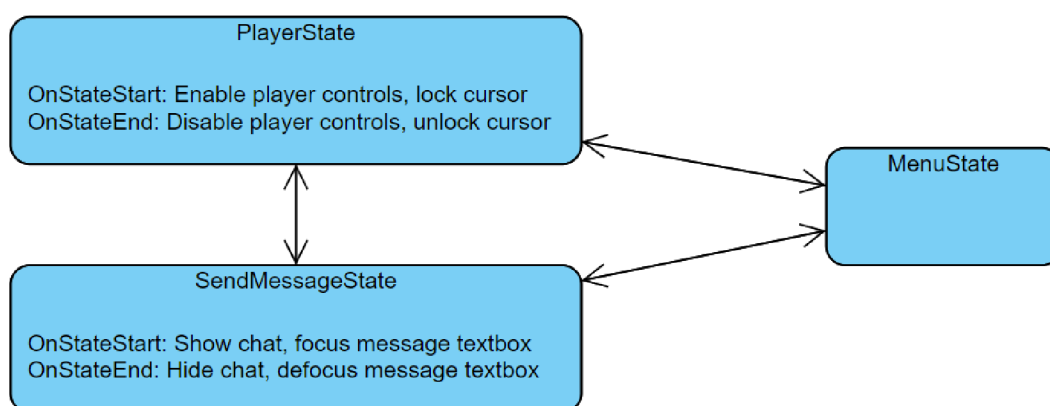
### 6.3.1 Správce kontextu

Správce kontextu se stará o přepínání stavů mezi ovládáním uživatelského charakteru v prostředí scény, zobrazením nastavení a stavu určeného pro posílání zprávy. Přepínání stavů je řešeno prostřednictvím implementace návrhového vzoru State-Machine (stavového stroje), který má kontrolu nad logikou přepínání stavů. Každý stav je dále reprezentován vlastní třídou, která poskytuje dvě metody, metodu pro volání při započítání stavu a metodu volanou při ukončení stavu.

Prostřednictvím změny stavu dokáže správce přepínat stav kurzoru myši, který musí být při ovládání uživatelského charakteru skrytý a uzamčený na jednom místě. To je umožněno zamčením a skrytím kurzoru při přepnutí správce do stavu ovládání hráče. Naopak při ukončení je kurzor opět zobrazen a uvolněn. Stav pro ovládání charakteru spravuje kurzor myši a přepínání kontroly pohybu nad charakterem.

Dalším ze stavů, do kterého se lze prostřednictvím správce přepnout, je stav pro zaslání zprávy. Započítáním tohoto stavu se ztmaví pozadí obrazovky, zvýrazní se okno chaty a umožní se interakce s jeho prvky. Ukončením stavu se uvede vše do předchozího nastavení před započítáním stavu.

Stav pro zobrazení nastavení nevyvolává žádné další operace, jeho správná funkce je zajištěna ukončením předchozího stavu.



Obrázek 4: Přejchody a operace stavů

### 6.3.2 Správce oken

V průběhu vývoje aplikace bylo potřeba zajistit jistou kontrolu nad zobrazovanými okny v aplikaci. Proto byl do projektu přidán správce oken, který má na starost přepínání jejich viditelnosti. Správce naslouchá změnám přepnutí viditelnosti jednotlivých oken a poskytuje notifikace formou událostí dalším komponentám.

### 6.3.3 Správce synchronizovaných dat

Správce synchronizovaných dat udržuje aktuální stav všech připojených klientů relace a poskytuje informace o lokálním klientovi dalším komponentám. Dále zajišťuje odesílání, přijímání a zobrazení zpráv v okně pro zasílání zpráv a vyvolává události pro panel záznamů s připojenými klienty v reakci na připojení nebo odpojení klienta.

### 6.3.4 Správci scény

V aplikaci se nachází dvě komponenty, které mají na starost operace spojené s aktuální scénou. Jeden z těchto správců zařizuje konfiguraci síťové komunikace prostřednictvím naslouchání na události poskytované knihovnou Unity Netcode a nabízí metody pro připojení nebo založení relace. Dále umožňuje rychlejší testování při vývoji aplikace automatizovaným založením nebo připojením k relaci v prostředí editoru Unity.

Druhým správcem je komponenta starající se o průběh aplikace. Poskytuje metody postavené nad správcem konfigurujícím síťovou komunikaci jako je založení a připojení k relaci, ukončení relace a ukončení aplikace. Komponenta zastřešuje dodatečnou logiku jako je například zobrazení kódu relace na uživatelském rozhraní při založení relace.

## 7 Výsledky

V této kapitole jsou shrnuty výsledky implementace navrženého systému a uvedeny problémy, které při testování nastaly. Nejprve je vysvětlen způsob testování synchronizovaných částí aplikace a dále je navázáno popisem interaktivních prvků implementovaných v aplikaci.

### 7.1 Testování vývoje

Testování aplikace bylo mnohem komplikovanější, než je tomu u aplikací, které nevyužívají ke svému chodu síťovou komunikaci. To je způsobeno faktem, kdy bylo potřeba otestovat nejen funkce na jedné běžící instanci aplikace, ale také funkce a akce, které mají být synchronizovány po síti. Tento stav vyžaduje mít alespoň dvě spuštěné instance projektu, které se k sobě vzájemně připojí. Bohužel tento přístup sebou nese nutnost sestavit projekt do spustitelné aplikace, což je velice časově náročné řešení.

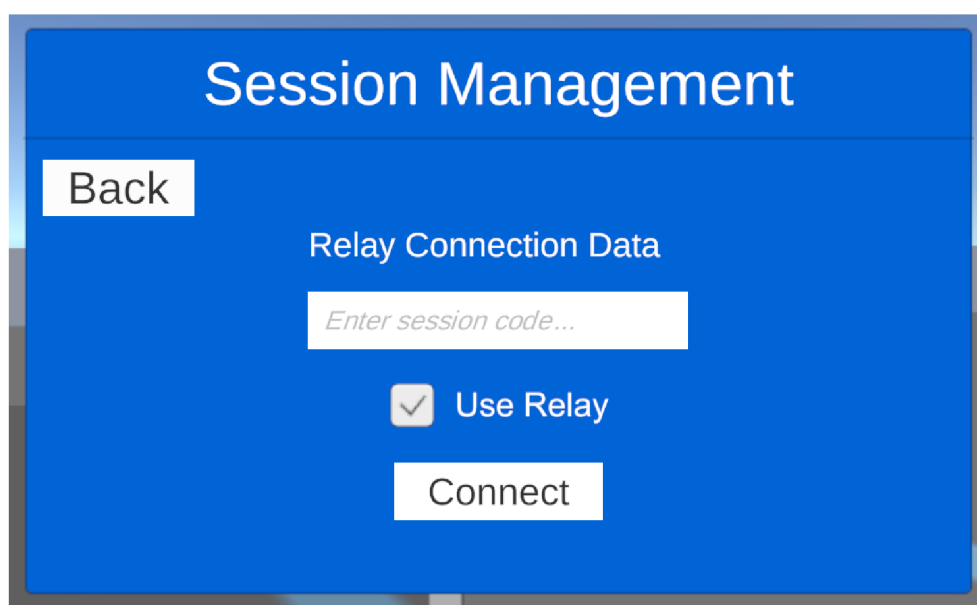
Na místo sestavování projektu při každé potřebě otestovat síťovou funkcionalitu bylo nalezeno řešení v dokumentaci Unity Netcode. Dokumentace zmiňuje open-source rozšíření jménem ParrelSync, které uživateli nabízí testovat hru pro více hráčů, bez nutnosti sestavení projektu, formou druhého okna editoru Unity, které zrcadlí změny z původního projektu. (VeriorPies 2022)

Rozšíření ParrelSync není podporováno přímo Unity, ale poskytuje značné urychlení testovacího procesu. Využitím této knihovny bylo možné sepsat logiku pro automatické spuštění a připojení k serveru. Aplikace v případě spuštění v rámci editoru zjistila z rozhraní knihovny, zda se jedná o klon projektu. V záporném případě spustila server a v kladném byl spuštěn klient a následně připojen k serveru.

Menší komplikace nastala při testování s možností připojení přes službu Unity Relay. V tomto případě musel spuštěný host po alokaci relay serveru uložit svůj kód relace ve formě textového souboru do adresáře projektu, ze kterého byl později načten při spuštění klonu projektu, který se automaticky pokusil k serveru připojit.

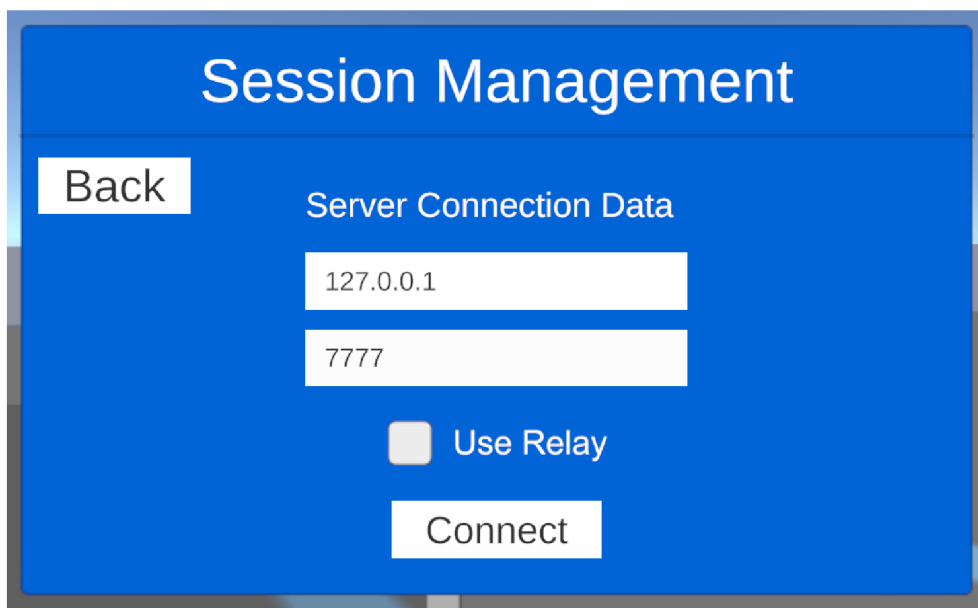
## 7.2 Spuštění aplikace a připojení k relaci

Po spuštění aplikace v prostředí operačního systému Windows 10 je uživatel nejdříve dotázán na poskytnutí síťového připojení aplikaci formou dialogového okna zobrazeného operačním systémem. Jeho potvrzením se aplikace spustí a v jejím okně zobrazí pohled do scény společně s dialogovým oknem aplikace dotazující se, zda uživatel chce hostovat novou relaci nebo zda se chce připojit k již stávající. Při výběru připojení se k relaci je uživatel dotázán, zda chce k připojení využít službu Unity Relay a jaký je připojovací kód relace.



Obrázek 5: Okno pro připojení k relaci prostřednictvím Unity Relay

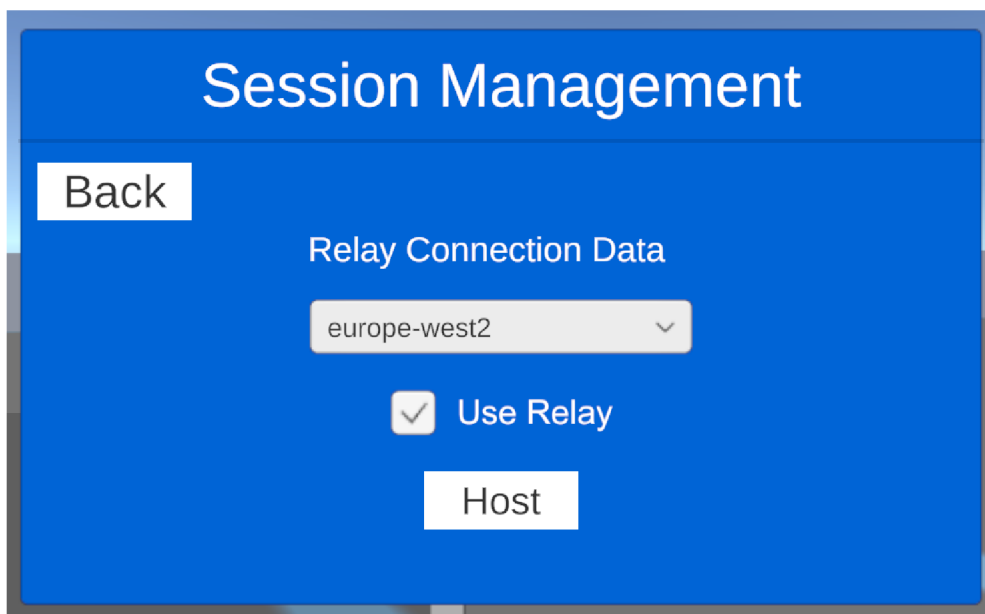
Druhým případem je, chce-li se uživatel připojit ke konkrétnímu uzlu v síti. V tomto případě je dotázán prostřednictvím textových polí na jeho adresu a port jako je vyobrazeno na obrázku 6.



Obrázek 6: Okno pro připojení k relaci bez využití Unity Relay

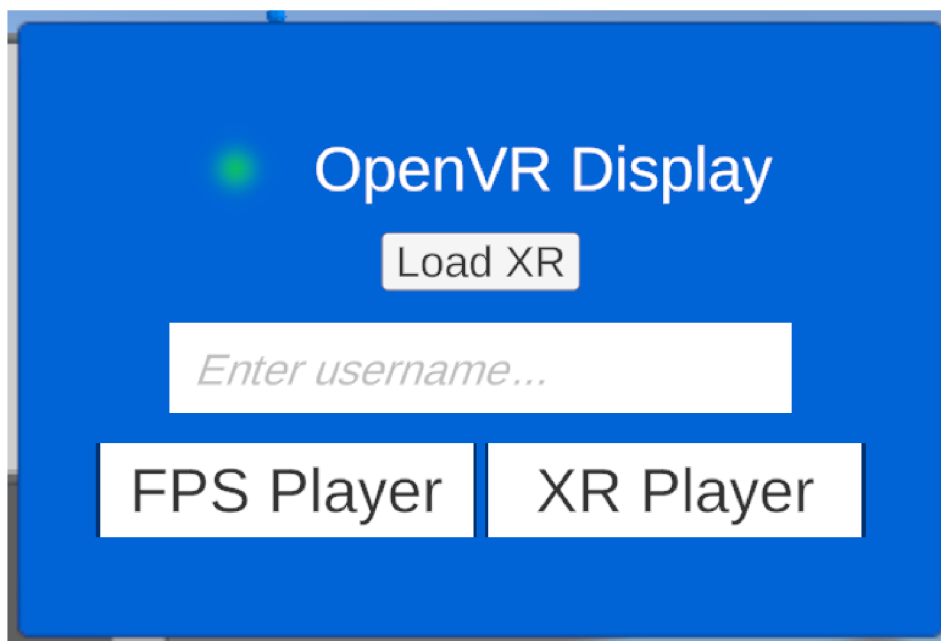
Pokračováním se aplikace pokusí připojit ke zvolenému uzlu a při neúspěšném připojení se proces opakuje. V případě úspěšného připojení si server zaeviduje nově příchozího klienta a na straně lokálního klienta se zobrazí další okno, které vyžaduje volbu jeho jména v rámci relace. Po zadání jména, které musí být v rozmezí 3 až 20 znaků může klient vstoupit do scény prostřednictvím FPS charakteru s pohledem z první osoby.

Pokouší-li se uživatel založit novou relaci, opět má možnost si zvolit, zda k připojení využít relay serveru od Unity. Ponechá-li volbu k použití Unity Relay nezměněnou, aplikace uživateli nabídne volbu serverů z různých lokalit, přes který bude komunikace probíhat. Po jeho výběru se vytvoří nová relace na zvoleném serveru a v okně aplikace se v jejím pravém horním rohu zobrazí šestimístní kód umožňující připojení klientům ke stejné relaci. Ukázkové znázornění kódu je vyobrazeno na obrázku 16.



Obrázek 7: Okno pro založení relace prostřednictvím Unity Relay

Po založení relace hostujícím klientem se aplikace pokusí zjistit, zda je k jeho zařízení připojena virtuální realita. Je-li připojena a rozpoznána načtou se její ovladače. Pokud jsou úspěšně načteny, okno s volbou jména v rámci relace zobrazí informaci o názvu připojené virtuální reality a umožní hráči zvolit, zda chce vstoupit do scény prostřednictvím FPS charakteru nebo XR charakterem podporujícího virtuální realitu. Po zadání vyžadovaného uživatelského jména je uživateli poskytnut přístup do scény prostřednictvím zvoleného charakteru.



Obrázek 8: Panel výběru charakteru pro hostitele relace

Výběrem charakteru se server dozví, jaký objekt reprezentující zvolený charakter má vytvořit a zobrazit ve scéně. Vytváření opět zprostředkovává server a klienti pouze posílají požadavky pro uskutečnění. Server tyto požadavky přijímá a vytváří zvolený objekt u každého z klientů v relaci.

V případě, jedná-li se o XR charakter, nastává speciální stav, kdy je potřeba vytvořit jeho objekt pouze lokálně u hostitele a ostatním klientům místo toho vytvořit vizuální reprezentaci. Tento krok je nutný pro zajištění funkčnosti aplikace na straně klientů, kteří nemají možnost účastnit se scény prostřednictvím XR charakteru. Problémový stav je způsoben situací, kdy běžný klient nepotřebuje lokálně charakter podporující virtuální realitu se všemi jeho funkcemi a komponentami umožňující jeho ovládání. Vytvořená vizuální reprezentace je proto zbavena všech nepotřebných komponent a je složena pouze z grafických prvků. Vizuální reprezentace XR charakteru je znázorněna na obrázku 17 po levé straně modrého prostorového textu.

### **7.3 Virtuální prostředí a jeho prvky**

Dokončením připojení k relaci je uživateli vytvořen charakter, jehož prostřednictvím se lze ve scéně volně pohybovat.



### 7.3.1 Pohyb ve scéně

Pohyb ve scéně je zpřístupněn rozdílnou formou v závislosti na typu aktuálního charakteru. Jedná-li se o XR charakter, je pohyb řešen formou teleportace na vybranou lokalitu. Uživatel stiskne tlačítko dotykové plochy ovladače, kterým lze inicializovat přesun. Při stisku se zobrazí křivka znázorňující pozici přesunu a při uvolnění tlačítka dojde ke skrytí křivky a k výslednému přesunu, pokud to zvolené místo umožňuje. Tato funkcionality byla do aplikace integrována za využití rozšíření SteamVR.

Jedná-li se o FPS charakter, který má do scény pohled z první osoby, je pohyb po scéně řešen pomocí kláves WSAD. Kdy klávesy WS ovládají dopředný (W) a zpětný (S) posun a klávesy AD pohyb do stran vlevo (A) a vpravo (D). Pohyb je zprostředkován pouhým posunem objektu v požadovaném směru. V každém snímku se zjistí, jaké klávesy umožňující pohyb jsou stisknuty, vypočte se vektor pohybu a charakter se v tomto snímku přemístí směrem vypočteného vektoru. Velikost výsledné vzdálenosti je určena velikostí vektoru, která je vynásobena délkou trvání předchozího snímku.

### 7.3.2 Komunikační konzole

Okno komunikační konzole je dostupné oběma uživatelským typům. Bohužel uživatel s charakterem podporujícím virtuální realitu má přístup k tomuto oknu ztížen potřebou sundat si brýle VR pro umožnění interakce s ním. Tento problém by bylo možné vyřešit zobrazením okna komunikační konzole v prostředí scény jako je tomu u panelu s připojenými klienty. Zadávání textu zprávy by mohlo být umožněno formou tlačítek virtuální klávesnice, které mají umožněny interakci prostřednictvím ovladačů VR.



Obrázek 9: Okno s komunikační konzolí

Na obrázku výše je vyobrazeno okno komunikační konzole. Okno v aktuálním stavu již zobrazuje dvě zaslání zprávy prostřednictvím sítě od dvou různých odesílatelů z rozdílných zařízení. První zpráva je zaslána hostitelem relace, od kterého byla přijata po síti. Druhá zpráva je od lokálního klienta. Jeho zpráva byla odeslána všem aktuálním účastníkům relace, ale v rámci jeho okna byla rovnou přidána mezi ostatní zprávy. Tento přístup odstraňuje zbytečné zpoždění při čekání na vyobrazení odesílané zprávy u lokálního klienta. Nevyužitím tohoto přístupu by byla zpráva po odeslání nejdříve zaslána serveru a následně ze serveru zpět lokálnímu klientovi a až poté přidána do seznamu zpráv.

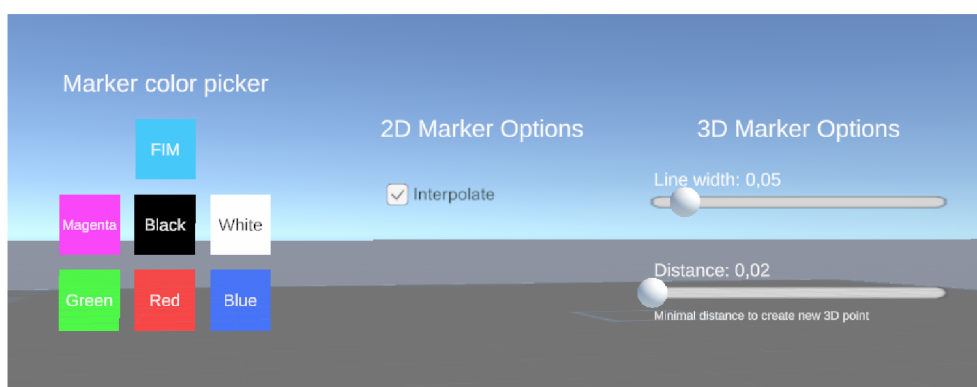
### 7.3.3 Ovládací panel

V prostředí scény byl vytvořen panel poskytující změny nastavení vlastností interaktivních fixů, kterými je umožněno kreslení v prostoru scény. S panelem lze interagovat pouze prostřednictvím ovladačů virtuální reality. Jak je vidět na obrázku 10 níže, panel je tvořen třemi částmi.

V první části se nachází sedm dotykových ploch pro výběr barvy aktuálně drženého fixu. Fixem je třeba se pouze dotknout dotykové plochy, která v reakci na tento úkon aktualizuje barvu fixu, kterou je s ním kresleno.

Druhá část panelu slouží pro nastavení vlastností při kreslení na tabuli ve scéně. Na ní je přítomno jedno přepínací tlačítko sloužící k přepínání funkce interpolace, která ovlivňuje průběh kreslení na tabuli.

Poslední část je určena úpravě parametrů prostorového kreslení. Nachází se na ní dva posuvníky. Z nichž první ovládá nastavení šířky kreslené prostorové čáry. Druhým posuvníkem lze změnit minimální vzdálenost pro vykreslení dalšího bodu trajektorie prostorové čáry.



Obrázek 10: Panel pro nastavení vlastností fixů

### 7.3.4 Prostorové kreslení

Prostorové kreslení se provádí uchopením patřičného fixu umístěném ve scéně. Tento fix má na sobě značení ve formě popisku s textem pro snadné a rychlé odlišení od druhého fixu určeného pouze ke kreslení na tabuli.

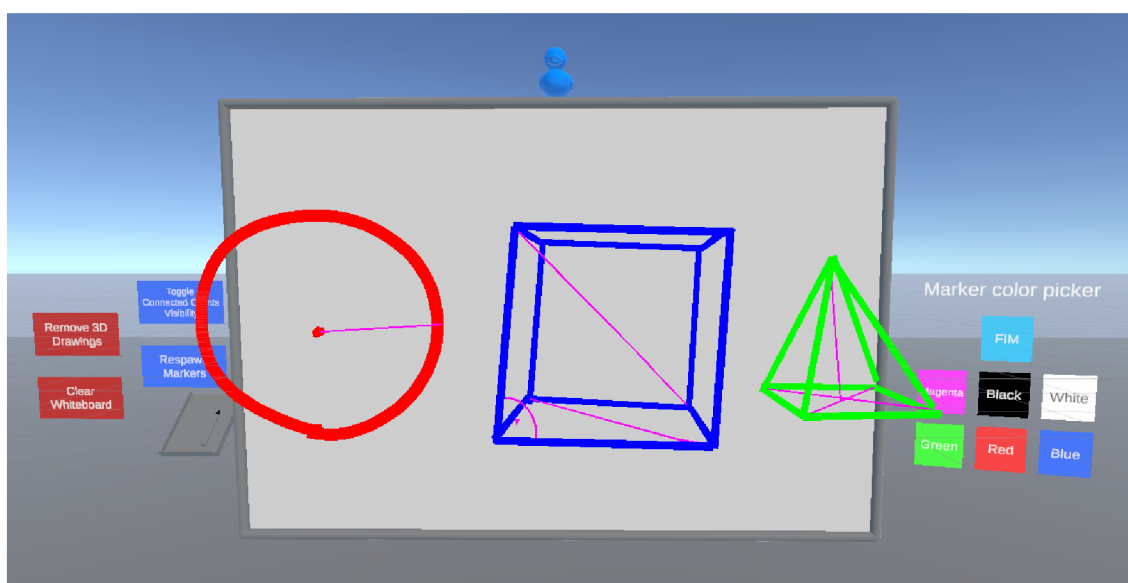
Funkce prostorového kreslení je zajištěna formou vytvořeného objektu scény, který reprezentuje prostorovou kresbu. Objekt musí být vytvořen pro každou kresbu nový, jinak by jednotlivé kresby byly propojeny prostorovou přímkou vedoucí z posledního bodu předchozí kresby do místa započetí kresby nové. To je zapříčiněno funkcí komponenty LineRenderer nabízené enginem Unity, která neposkytuje metody pro vytvoření přerušované prostorové čáry.

Společně s možností vytvářet prostorové kresby bylo do aplikace implementováno jejich smazání a opětovné přidání. Tato funkce poskytuje velmi

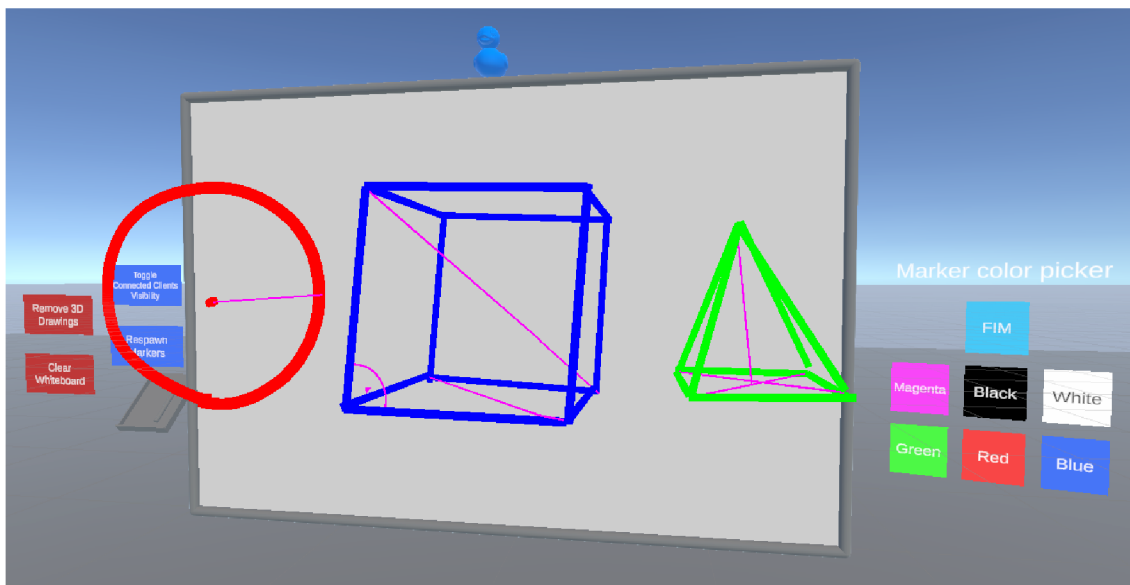
rychlou správu již vytvořených kreseb. Správa je umožněna tahem v určeném směru na dotykové ploše trackpadu ovladače VR. Při tažení směrem zprava do leva se zjistí, zdali jsou ve scéně přítomné nějaké prostorové kresby a v kladném případě se poslední z nich skryje. Tahem v opačném směru se naopak zjistí, zda některé z nakreslených kreseb již byly skryty a znovu se zobrazí poslední, která byla naposledy skryta. Dojde-li k započetí kreslení nové prostorové kresby, všechny kresby, které byly předtím skryty, se ze scény odstraní.

Kvalita výsledné prostorové kresby velice závisí na schopnosti vnímání virtuální prostorové scény prostřednictvím brýlí virtuální reality. Ovlivněn je vzhled a výsledná reprezentace kreslených tahů, které závisí na zkušenosti uživatele udržet stabilně ovladač VR s drženým fixem a následovat jím v prostoru trajektorii tahu, který je zamýšleno nakreslit.

Následující obrázky demonstrují funkci prostorového kreslení a vyobrazují pohled na náčrty objektů ze dvou různých úhlů s různými nastaveními použitými při jejich kreslení.



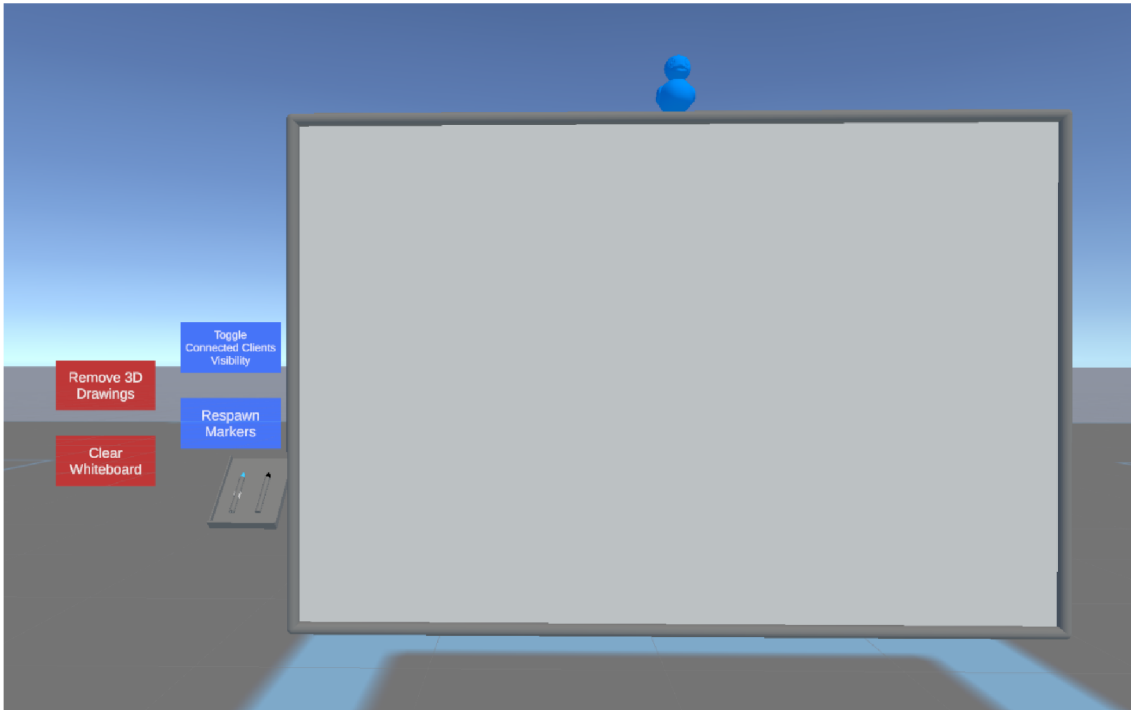
Obrázek 11: První pohled na načrtnuté objekty prostorovým kreslím



Obrázek 12: Pohled z jiného úhlu na náčrty objektů

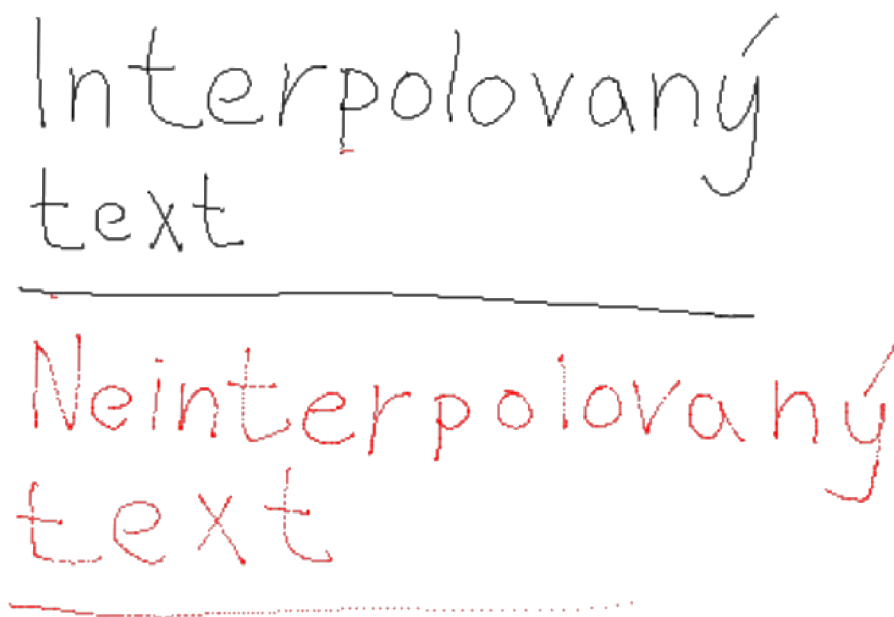
### 7.3.5 Tabule

V prostoru virtuální scény se nachází tabule, na kterou lze kreslit fixem, který při startu relace leží na její odkládací části, jak je vyobrazeno na obrázku 13. Na obrázku si lze také povšimnout čtyřech vytvořených tlačítek poskytující kontrolu nad sdílenou scénou.



Obrázek 13: Pohled na tabuli a ovládací tlačítka

Vlastnosti kreslených čar na prostředí tabule lze nastavit přepínačem interpolace, který se nachází na ovládacím panelu. Interpolace přímo ovlivňuje vzhled a viditelnost konečné reprezentace kreseb a tím i čitelnost textu v případě psaní.



Obrázek 14: Rozdíl kreseb v závislosti na použití interpolace

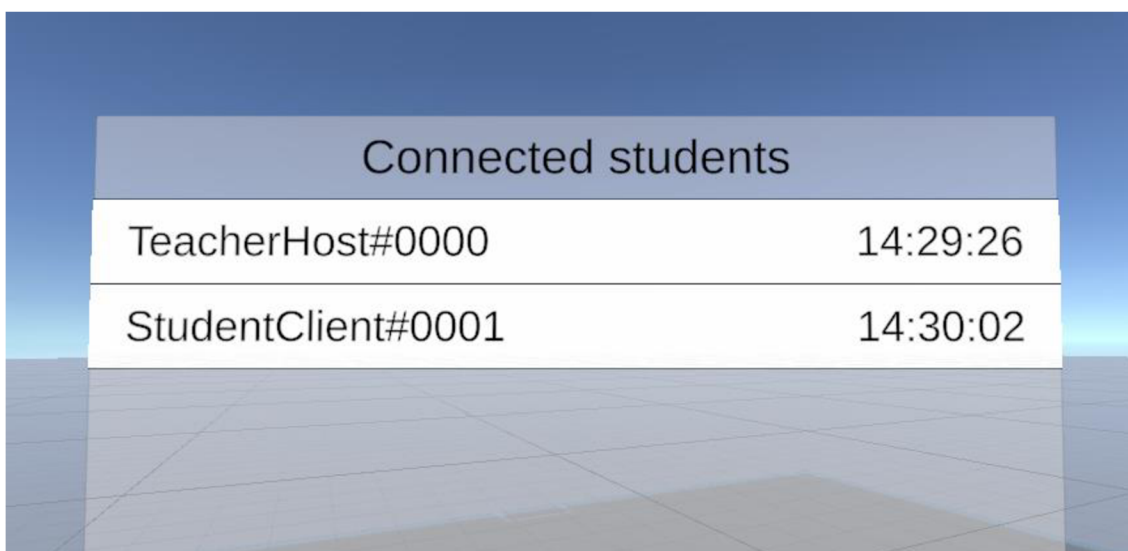
Na obrázku výše se nachází dva texty psané na tabuli fixem drženého ovladačem ve virtuální realitě. Na obrázku je patrný rozdíl mezi těmito texty. Horní černý text byl napsán fixem se zapnutou lineární interpolací mezi předávanými pozicemi. Díky tomu jsou jednotlivé tahy písmen textu jednolité a nenachází se mezi nimi žádné mezery. Naopak u spodního červeného textu byla interpolace vypnuta.

Rozdíl užití interpolace je nejvíce patrný u rychlých tahů. Příkladem jsou na obrázku 14 dvě vodorovné čáry. Tyto čáry byly kresleny postupně zrychlujícím se tahem pro demonstraci rozdílu. Horní černá čára byla kreslena za použití interpolace a je jednolitá bez zjevných přerušování. U spodní červené čáry kreslené s vypnutou interpolací si lze všimnout, že postupným zrychlováním tahu se četnost obarvených pixelů snižovala. Z toho důvodu se červený text psaný rychlejšími tahy jeví přerušovaný až tečkovaný.

### 7.3.6 Panel připojených klientů

V prostoru virtuálního prostředí byl vyhotoven panel udržující aktuální stav připojených klientů, který je sdílen mezi všemi účastníky připojenými k relaci.

Záznamy připojených klientů jsou chronologicky řazeny podle času připojení k relaci. Panel je možné prostřednictvím tlačítka vyobrazeného na obrázku 13 skrýt. Skrytí panelu je umožněno pouze prostřednictvím ovladače VR. Tím je jeho interakce poskytnuta pouze uživateli s charakterem VR. Interakcí s tlačítkem se přepne stav zobrazení panelu ostatním připojeným klientům, kterým panel ve scéně zmizí. Na straně interagujícího uživatele s tlačítkem se panel pouze zprůsvitní pro poskytnutí zpětné vazby o provedení jeho skrytí ostatním uživatelům. Na obrázku 15 je vyobrazen panel se dvěma připojenými uživateli, zakladatelem relace se jménem „TeacherHost“ a připojeným klientem „StudentClient“.



Connected students	
TeacherHost#0000	14:29:26
StudentClient#0001	14:30:02

Obrázek 15: Panel zobrazující připojené klienty

#### **7.4 Synchronizované prostředí více uživatelů**

Po založení síťové relace jedním z uživatelů aplikace je následně umožněno se kterémukoli dalšímu uživateli k relaci připojit zná-li přístupový kód, v případě komunikace prostřednictvím relay serveru, nebo zná-li adresu a port uzlu, který relaci hostuje.

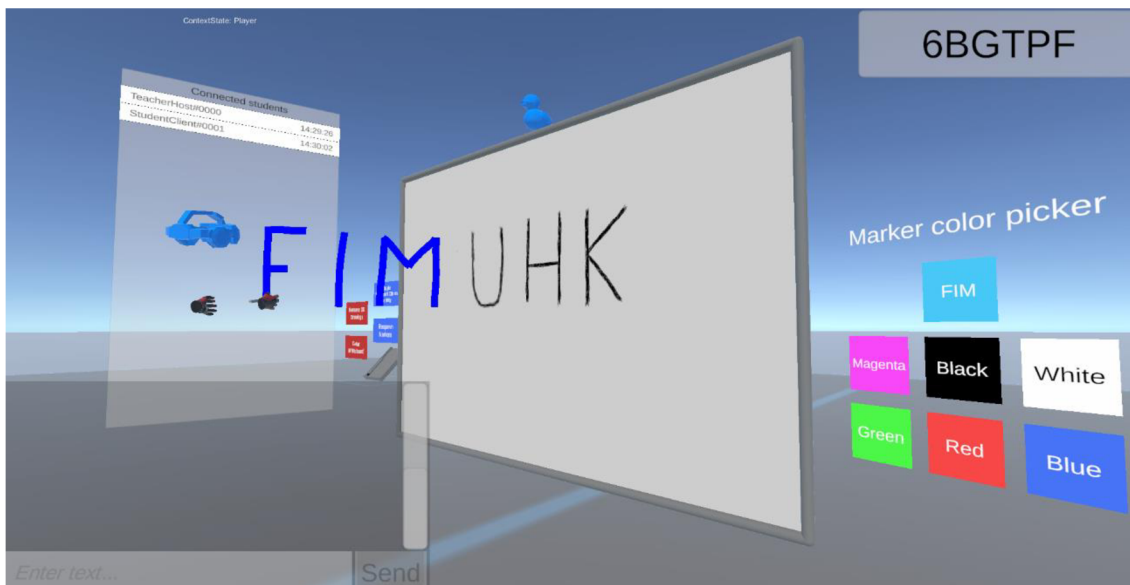
Připojení uživatelé mají možnost se uvnitř sdíleného virtuálního prostředí vzájemně vidět a nahlížet na interakce prováděné hostitelem relace, který má k dispozici interaktivní prvky. Interaktivní kresby ve virtuální scéně jsou synchronizovány mezi všemi klienty.





Obrázek 16: Pohled hostitele do scény prostřednictvím VR

Na obrázku 16 je vidět náhled hostitele do virtuální scény prostřednictvím charakteru virtuální reality. Na obrázku si lze povšimnout dvou kreseb, které byly vytvořeny hostitelem scény a jsou zobrazeny druhému klientovi relace, jehož pohled do scény je zobrazen na následujícím obrázku 17. Na obrázku 17 je také vidět vizuální reprezentace XR charakteru, který stojí u písmena F prostorové kresby. Vizuální reprezentace se skládá ze dvou rukavic představující ovladače VR a zobrazovacího zařízení virtuální reality, které znázorňují aktuální polohu v prostředí scény.



Obrázek 17: Pohled do scény připojeného klienta

## 8 Závěry a doporučení

V práci byly prozkoumány principy a metody víceuživatelského prostředí virtuální reality. Byl navržen a implementován prototyp aplikace poskytující sdílené virtuální prostředí určené pro interaktivní prezentace a případně výuky v distanční formě.

Vytvořená testovací aplikace umožňuje připojení vzdáleným uživatelům různých platform a sdílení prostředí virtuální prostorové scény. Scény se mohou účastnit dva typy uživatelských charakterů, které určují způsob chování jednotlivých uživatelů v rámci sdílené virtuální scény. Charakter XR určený pro systém virtuální reality může být vybrán pouze hostitelem relace a povoluje uživateli vykonávat veškeré dostupné interakce v prostředí scény. Tento uživatel se může volně pohybovat po scéně formou teleportace a má k dispozici interaktivní prvky, které umožňují provádění změn scény. Vzhled scény je sdílen po síti všem připojeným klientům aktuálně hostované relace. Charakteru FPS tvořenému pohledem první osoby je poskytnut pouze náhled do scény a interakce s komunikační konzolí pro zaslání zpráv.

K připojení není vyžadován dedikovaný server nebo nutnost hostitele vlastnit veřejnou IP adresu. Klient zakládající relaci si při startu aplikace zvolí server, přes který bude následná síťová komunikace probíhat a tím se stane hostitelem relace. Ostatním klientům je umožněno připojení formou unikátního kódu relace, který je vygenerován jejímu hostiteli.

Do aplikace byla implementována funkce prostorového kreslení, která umožňuje prezentaci kreseb všem klientům scény. Testování ukázalo, že přesnost funkce kreslení je velice ovlivněna schopností uživatelů vnímat prostor scény prostřednictvím brýlí VR a zkušeností manipulovat fixem uchopeném ovladačem ve virtuálním prostoru. Důsledkem toho může dojít ke tvorbě nepřesných modelů a špatnému předání informací ostatním účastníkům víceuživatelského prostředí.

Druhou interaktivní funkcí je typ fixu umožňující sdílené kreslení na tabuli, která je umístěna do prostředí scény. Tabule společně s fixem funguje velice podobně jako v běžné realitě. Fix při kontaktu s tabulí obarví místo jejich kontaktu. Funkce fixu a výsledná reprezentace kreslené čáry závisí na zapnutí funkce

interpolace, kdy pro prezentační a výukové účely je fix využitelný pouze při zapnuté interpolaci, která poskytuje kreslení souvislých nepřerušovaných čar.

Prototyp aplikace dále disponuje komunikační konzolí pro jednoduché zasílání zpráv mezi všemi účastníky relace nebo panel se záznamy aktuálních účastníků sdílené scény jehož stav je automaticky aktualizován při připojení či odpojení některého z klientů.

Koncept navrženého systému víceuživatelského prostředí byl odzkoušen při několika ukázkových prezentacích, které k popisu využívali oba implementované typy kreslení. Úkolem prezentací bylo vysvětlení a popis libovolného prostorového objektu pro demonstraci funkcí dostupných v rámci prototypu aplikace. Testující uživatelé následně poskytli jejich zpětnou vazbu k vlastnostem a interaktivním prvkům, které virtuální scéna nabízí.

Při testování byla velice kladně přijímána možnost připojení a hostování sezení prostřednictvím kódu relace, při kterém zakládající uživatel nemusí vlastnit veřejnou IP adresu nebo dedikovaný server. Nejvíce zajímavá přišla testujícím uživatelům funkce prostorového kreslení, jejíž využíváním bylo tráveno nejvíce času. Naopak nejčastěji zmiňovaným prvkem chybějícím ve scéně, byla absence možnosti smazání posledního tahu provedeného na ploše tabule, jako je tomu u prostorového kreslení. Dále byla zmiňována chybějící možnost pro kreslení přímek v prostoru nebo zjednodušený způsob pro nakreslení primitivních geometrických útvarů při využívání prostorového kreslení. Posledním často poznamenávaným nedostatkem byla absence možnosti interakce s komunikační konzolí pro zasílání zpráv ve virtuálním prostředí při nasazených brýlích VR.

Prototyp výukové aplikace by bylo možné rozšířit implementací výše zmíněných chybějících funkcí společně s dalšími jako je přesun prostorové kresby, změna škály nebo její export ve formě souboru. Dále by se nabízelo vytvoření objektu pro mazání obsahu plochy tabule nebo rozšířit podporované platformy o mobilní zařízení. Implementace výše zmíněných funkcí by mohla dále zvýšit dostupnost a efektivitu aplikace při předávání vědomostí v průběhu výuky.

## Seznam použité literatury

ABBAS, Sami, Anderson Augusto SIMISCUKA a Gabriel-Miro MUNTEAN, 2019. A Platform Agnostic Solution for Inter-Communication between Virtual Reality Devices. In: [online]. Dostupné z: doi:10.1109/WF-IoT.2019.8767230

BAUR, Kilian, Alexandra SCHÄTTIN, Eling D. DE BRUIN, Robert RIENER, Jaime E. DUARTE a Peter WOLF, 2018. Trends in robot-assisted and virtual reality-assisted neuromuscular therapy: a systematic review of health-related multiplayer games. *Journal of NeuroEngineering and Rehabilitation* [online]. **15**(1), 107. ISSN 1743-0003. Dostupné z: doi:10.1186/s12984-018-0449-9

HENDERSON, Tristan, 2001. Latency and User Behaviour on a Multiplayer Game Server. In: Jon CROWCROFT a Markus HOFMANN, ed. *Networked Group Communication* [online]. Berlin, Heidelberg: Springer, s. 1–13. Lecture Notes in Computer Science. ISBN 978-3-540-45546-2. Dostupné z: doi:10.1007/3-540-45546-9\_1

KAN, H.Y., Vincent DUFFY a Chuan-Jun SU, 2001. An Internet virtual reality collaborative environment for effective product design. *Computers in Industry* [online]. **45**, 197–213. Dostupné z: doi:10.1016/S0166-3615(01)00093-8

MCCREARY, Sean a Kc CLAFFY, 2000. *Trends in Wide Area IP Traffic Patterns - A View from Ames Internet Exchange*. leden 2000.

MIRROR NETWORKING, 2022. *Mirror Networking – Open Source Networking for Unity* [online]. [vid. 2022-04-19]. Dostupné z: <https://mirror-networking.com/>

PANTEL, Lothar a Lars C. WOLF, 2002. On the impact of delay on real-time multiplayer games. In: *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video* [online]. New York, NY, USA: Association for Computing Machinery, s. 23–29 [vid. 2022-04-18]. NOSSDAV '02. ISBN 978-1-58113-512-1. Dostupné z: doi:10.1145/507670.507674

PATIL, Pratik a Ronald ALVARES, 2015. Cross-platform Application Development using Unity Game Engine. *IJARCSMS, Issue 4*. **3**, 19.

SHI, Yangming, Jing DU, Sarel LAVY a Dong ZHAO, 2016. A Multiuser Shared Virtual Environment for Facility Management. *Procedia Engineering* [online]. **145**, ICSDEC 2016 – Integrating Data Science, Construction and Sustainability, 120–127. ISSN 1877-7058. Dostupné z: doi:10.1016/j.proeng.2016.04.029

SUZNJEVIC, Mirko a Jose SALDANA, 2016. Delay Limits for Real-Time Services. *IETF draft*.

TAY, B. H. a A. L. ANANDA, 1990. A survey of remote procedure calls. *ACM SIGOPS Operating Systems Review* [online]. **24**(3), 68–79. ISSN 0163-5980. Dostupné z: doi:10.1145/382244.382832

THALMANN, Nadia Magnenat a Igor S. PANDZIC, 1997. Collaborative work using our Virtual Reality Network System (VLNET). In: Frank PLONKA a Gustav OLLING, ed. *Computer Applications in Production and Engineering: IFIP TC5 International Conference on Computer Applications in Production and Engineering (CAPE '97) 5–7 November 1997, Detroit, Michigan, USA* [online]. Boston, MA: Springer US, IFIP — The International Federation for Information Processing, s. 727–735 [vid. 2022-04-22]. ISBN 978-0-387-35291-6. Dostupné z: doi:10.1007/978-0-387-35291-6\_65

UNITY TECHNOLOGIES, 2022a. *1st-Party Netcode Solution | Unity* [online] [vid. 2022-04-19]. Dostupné z: <https://unity.com/products/netcode>

UNITY TECHNOLOGIES, 2022b. *NetworkVariable | Unity Multiplayer Networking* [online] [vid. 2022-04-24]. Dostupné z: <https://docs-multiplayer.unity3d.com//netcode/current/basics/networkvariable>

UNITY TECHNOLOGIES, 2022c. *Sending Events with RPCs | Unity Multiplayer Networking* [online] [vid. 2022-04-19]. Dostupné z: <https://docs-multiplayer.unity3d.com//netcode/current/advanced-topics/messaging-system>

UNITY TECHNOLOGIES, 2022d. *Transports | Unity Multiplayer Networking. Unity Multiplayer Networking* [online] [vid. 2022-04-17]. Dostupné z: <https://docs-multiplayer.unity3d.com//netcode/current/advanced-topics/transports>

UNITY TECHNOLOGIES, 2022e. *Tricks and patterns to deal with latency | Unity Multiplayer Networking* [online] [vid. 2022-04-19]. Dostupné z: <https://docs-multiplayer.unity3d.com//netcode/current/learn/dealing-with-latency>

UNITY TECHNOLOGIES, 2022f. *Unity Relay (Beta). Unity Documentation* [online] [vid. 2022-04-17]. Dostupné z: <https://docs.unity.com/relay/introduction.html>

VALVE CORPORATION, 2022. *SteamVR Unity Plugin | SteamVR Unity Plugin. SteamVR Unity Plugin* [online] [vid. 2022-04-22]. Dostupné z: [https://valvesoftware.github.io/steamvr\\_unity\\_plugin/](https://valvesoftware.github.io/steamvr_unity_plugin/)

VERIORPIES, 2022. *ParrelSync* [online] [vid. 2022-04-19]. Dostupné z: <https://github.com/VeriorPies/ParrelSync>

## Zadání diplomové práce

**Autor:** Bc. Tomáš Bareš

**Studium:** I1900797

**Studijní program:** N1802 Aplikovaná informatika

**Studijní obor:** Aplikovaná informatika

**Název diplomové práce:** **Víceuživatelská prostředí v systémech virtuální reality**  
Název diplomové práce AJ: Multi-user virtual reality environments

### Cíl, metody, literatura, předpoklady:

#### Cíl

Prostudovat problematiku víceuživatelských prostředí v systémech virtuální reality (cross-platform a multiplayer). Na vhodném příkladu odzkoušet možnosti vzájemné interakce.

#### Osnova práce

1. Provést literární rešerši a seznámit se s aktuálním stavem oblasti cross-platform a multiplayer systémů v prostředí virtuální reality.\par
2. Vytvořit přehled metod a technologií používaných ve virtuální scéně s více uživateli.\par
3. Zaměřit se na možnosti interakce a sdílení virtuální scény.\par
4. Navrhnout testovací úlohy demonstrujících využití sdílení prostředí a interakce ve virtuální realitě.\par
5. Pro implementované řešení provést testování a zhodnotit dosažené výsledky.\par

PANTEL, Lothar a Lars C. WOLF. On the impact of delay on real-time multiplayer games. In: *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video - NOSSDAV '02* [online]. New York, New York, USA: ACM Press, 2002, 2002, s. 23- [cit. 2021-10-14]. ISBN 1581135122. Dostupné z: doi:10.1145/507670.507674

ABBAS, Sami, Anderson Augusto SIMISCUKA a Gabriel-Miro MUNTEAN. A Platform Agnostic Solution for Inter-Communication between Virtual Reality Devices. In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* [online]. IEEE, 2019, 2019, s. 189-194 [cit. 2021-10-14]. ISBN 978-1-5386-4980-0. Dostupné z: doi:10.1109/WF-IoT.2019.8767230

**Garantující pracoviště:** Katedra informatiky a kvantitativních metod,  
Fakulta informatiky a managementu

**Vedoucí práce:** Ing. Bruno Ježek, Ph.D.

**Oponent:** prof. RNDr. PhDr. Antonín Slabý, CSc.

**Datum zadání závěrečné práce:** 26.1.2021