

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Používateľské rozhranie a externá
komunikácia pre neurorehabilitáciu
vo virtuálnej realite**

Bakalárska práca

**Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky**

**Používateľské rozhranie a externá
komunikácia pre neurorehabilitáciu
vo virtuálnej realite**

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiteľ:	doc. Ing. Štefan Korečko, PhD.
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)

Košice 2026

Tomáš Süč

Abstrakt v slovenčine

Virtuálna realita sa postupom času stala dôležitou súčasťou mnohých odvetví. Táto práca sa zaoberá jej využitím v oblasti medicíny, konkrétne pri neurorehabilitačných cvičeniach. Cieľom práce je implementácia základného prototypu a vylepšenie systémov interakcie, ako je ovládanie gestami umožňujúce interakciu bez použitia ovládačov. Súčasťou práce je tiež implementácia systému externej komunikácie s EEG/BCI platformou OpenViBE prostredníctvom protokolu WebSocket. Výsledný prototyp predstavuje základ novej verzie systému L-NeRVEn pripravený na ďalšie rozširovanie a optimalizáciu.

Kľúčové slová v slovenčine

neurorehabilitácia, virtuálna realita, hand tracking, Unity, WebSocket, EEG/BCI komunikácia, XR systémy, interakcia snímaním rúk, Meta Quest

Abstrakt v angličtine

Virtual reality has gradually become an important part of many industries. This thesis focuses on its application in the field of medicine, specifically in neurorehabilitation exercises. The aim of the thesis is to implement a basic prototype and improve interaction systems, such as gesture control enabling interaction without the use of handheld controllers. The thesis also includes the implementation of an external communication system with the EEG/BCI platform OpenViBE using the WebSocket protocol. The resulting prototype represents the foundation of a new version of the L-NeRVEn system, prepared for further expansion and optimization.

Kľúčové slová v angličtine

neurorehabilitation, virtual reality, hand tracking, Unity, WebSocket, EEG/BCI communication, XR systems, hand-tracking interaction, Meta Quest

Bibliografická citácia

SÜČ, Tomáš. *Používateľské rozhranie a externá komunikácia pre neurorehabilitáciu vo virtuálnej realite*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2026. 42 s. Vedúci práce: doc. Ing. Štefan Korečko, PhD.

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY
Katedra počítačov a informatiky

ZADANIE
BAKALÁRSKEJ PRÁCE

Študijný odbor: **Informatika**
Študijný program: **Informatika**

Názov práce:

**Používateľské rozhranie a externá komunikácia pre neurorehabilitáciu vo
virtuálnej realite**

**User Interface and External Communication for Neurorehabilitation in Virtual
Reality**

Študent: **Tomáš Süč**
Školiteľ: **doc. Ing. Štefan Korečko, PhD.**
Školiace pracovisko: **Katedra počítačov a informatiky**
Konzultant práce:
Pracovisko konzultanta:

Pokyny na vypracovanie bakalárskej práce:

1. Oboznámiť sa so súčasnou verziou systému L-NeRVEn pre kolaboratívnu neurorehabilitáciu vo virtuálnej realite (VR) a podobnými riešeniami.
2. Analyzovať existujúce riešenia pre externú komunikáciu na báze protokolu WebSocket a interakciu s používateľom, najmä pomocou snímania rúk.
3. Na základe analýzy a pokynov vedúceho práce navrhnuť a implementovať funkcionality pre externú komunikáciu a interakciu s používateľom pre prototyp novej verzie systému L-NeRVEn.
4. Návrh a implementáciu koordinovať s ďalšími riešiteľmi prác na systéme L-NeRVEn.
5. Implementované riešenie overiť vhodnými testami a na relevantnom VR hardvéri.
6. Vypracovať dokumentáciu podľa pokynov vedúceho práce.

Jazyk, v ktorom sa práca vypracuje: **slovenský**
Termín pre odovzdanie práce: **29.05.2026**
Dátum zadania bakalárskej práce: **31.10.2025**




prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som záverečnú prácu vypracoval samostatne s použitím uvedenej odbornej literatúry, ako aj s využitím nástrojov generatívnej umelej inteligencie v súlade s Metodickým pokynom o záverečných a kvalifikačných prácach a Etickým kódexom študenta Technickej univerzity v Košiciach. Umelá inteligencia bola použitá ako podporný nástroj bez porušenia zásad akademickej etiky a autorskej integrity.

Podakovanie

Na tomto mieste by som sa rád poďakoval svojmu vedúcemu práce, doc. Ing. Štefanovi Korečkovi, PhD., za jeho čas a odborné vedenie počas riešenia mojej záverečnej práce. Rovnako by som sa rád poďakoval svojim rodičom a priateľom za ich podporu a povzbudzovanie počas celého môjho štúdia.

Obsah

1	Úvod	1
2	VR v neurorehabilitácii	4
2.1	Využitie VR v neurorehabilitácii	4
2.2	Systémy XR	5
2.2.1	XR hardvér	5
2.2.2	Softvér pre vývoj XR aplikácií	5
2.3	Hand tracking	5
2.4	WebSocket komunikácia	6
3	Existujúce riešenia	7
3.1	Existujúce aplikácie pre neurorehabilitáciu	7
3.2	Existujúce nástroje pre vývoj	9
3.2.1	Prostredia	9
3.2.2	Balíky VR integrácie prostredia Unity	10
3.2.3	Balíky pre externú komunikáciu pomocou WebSocket	11
3.3	Vyhodnotenie záverov pre implementáciu	12
4	Aktuálny stav projektu	14
4.1	Aktuálna implementácia	14
4.2	Nedostatky systému	16
4.3	Záver a návrhy pre implementáciu	17
5	Konceptuálny návrh	20
5.1	Stavový systém novej verzie	21
5.2	Konceptuálny prototyp	22
5.2.1	MetaXR rig a test interakcie	22
5.2.2	Kompatibilita s OpenXR	23
5.2.3	Integrácia avatarov a desktop rozhrania	25
5.2.4	WebSocket komunikácia	25

5.2.5	Závery pre plnú implementáciu	26
6	Prototyp novej verzie	27
6.1	Komunikácia s OpenViBE serverom	29
6.2	Systém ovládania a zjednotenie rigov	31
6.3	Sieťová synchronizácia rigov a interakcie	34
6.4	Škálovanie avatara a rúk	35
6.5	HUD menu	37
7	Vyhodnotenie a záver	38
7.1	Vyhodnotenie implementácie	38
7.2	Nedostatky prototypu	39
7.3	Záver	40
	Zoznam použitej literatúry	41
	Zoznam príloh	43

Zoznam obrázkov

5.1	Petriho sieť navrhovaného nového systému	21
5.2	Základná štruktúra MetaXR rig komponentu v scéne	22
5.3	Nastavenia OVR Manager skriptu	22
5.4	Jednoduché testovacie objekty interakcie v scéne	23
5.5	Nastavenia RigSwitcher skriptu.	24
5.6	Avatar Ready Player Me z predchádzajúcej verzie v scéne	25
5.7	Testovanie WebSocket komunikácie	26
6.1	Diagram tried pre implementáciu prototypu novej verzie systému. . .	28
6.2	Štruktúra rigov v jednotnom prefabe	31
6.3	Funkčné snímanie rúk v MetaXR po škálovaní avatara	32
6.4	Ukážka hybridného ovládania pomocou jedného ovládača a jednej ru- ky v MetaXR a OpenXR rigu	32
6.5	Prototyp UI pre manuálnu kalibráciu škály avatara a rúk	37
6.6	Prototyp HUD menu pre terapeuta	37

Zoznam výpisov

5.1	Detekcia typu VR zariadenia	24
6.1	Reakcia herného automatu na úspešný BCI signál.	30
6.2	Prepínanie cieľov pre inverznú kinematiku pri hybridnom ovládaní.	33
6.3	Synchronizácia vlastníctva objektu pri uchopení v online scéne.	35
6.4	Úprava metód AvatarController	36

1 Úvod

Neurorehabilitácia je dôležitou súčasťou liečby ľudí trpiacich ochoreniami, ktoré obmedzujú ich fyzický pohyb. S rastúcim počtom pacientov v posledných rokoch vznikol impulz pre rýchly rozvoj technológií, ktoré majú za cieľ zlepšiť možnosti liečby. Jednou z najviac rozvinutých oblastí týkajúcich sa pokrokovej neurorehabilitácie je využitie virtuálnej reality, ktorá vie významne pomôcť pacientovi, zjednodušiť prácu terapeutovi a celkovo výrazne zefektívniť jeho liečbu.

Virtuálna realita sa za posledné roky začala rapídne vyvíjať. Obdobie, kedy bola táto technológia dostupná len pre veľmi malú skupinu ľudí a za veľmi vysokú cenu, je už dávno preč. V súčasnosti je táto technológia prístupná aj bežným užívateľom, čo dalo vývojárom možnosť vytvárať aplikácie využívajúce virtuálnu realitu na mnoho účelov, pre širokú verejnosť. Využitie virtuálnej reality v oblasti neurorehabilitácie sa preukázalo byť jednou z týchto možností.

S rýchlym vývojom týchto zariadení sa začala objavovať technológia hand-tracking, ktorá umožňuje sledovanie pohybu rúk používateľa bez nutnosti fyzického ovládača. Tento spôsob ovládania vie byť pre používateľov v mnohých ohľadoch pohodlnejší a intuitívnejší. Využitie tejto technológie v oblasti neurorehabilitácie môže preto výrazne pomôcť terapeutovi, ale aj pacientovi.

Dôležitou súčasťou týchto systémov je v mnohých prípadoch aj externá komunikácia pomocou protokolu WebSocket, ktorý umožňuje bezproblémovú komunikáciu a prenášanie dôležitých dát v reálnom čase počas rehabilitačného procesu.

Táto práca bude zameraná na analýzu a vylepšenie už existujúcej základnej implementácie ovládania gestami rúk a systému externej komunikácie v aplikácii zdieľaného virtuálneho prostredia využívaného na neurorehabilitáciu horných končatín, vytvorením prototypu novej verzie aplikácie. Cieľom je zvýšiť presnosť a kvalitu týchto systémov a zabezpečiť, aby boli tieto systémy celkovo pre používateľa prínosné a prirodzené. Okrem analýzy bude táto práca obsahovať možné optimalizácie a vylepšenia aktuálnej implementácie systémov, ktoré by mali zlepšiť ich využitie v praxi.

Táto práca bola riešená v rámci projektu APVV-21-0105 s názvom „Dôveryhodná interakcia človek–robot a terapeut–pacient vo virtuálnej realite“, ktorého cieľom je výskum a vývoj technológií virtuálnej reality a interakčných systémov pre terapeutické a rehabilitačné využitie.

Formulácia úlohy

Cieľom tejto práce je detailná analýza implementovaného systému ovládania gestami rúk a systému externej komunikácie protokolom WebSocket, v aplikácii zdieľanej virtuálnej reality určenej na neurorehabilitáciu horných končatín, ktorá je implementovaná vo vývojovom prostredí Unity pre platformu Meta Quest.

Úlohou je identifikovať nedostatky týchto systémov, či už je to ich používateľský zážitok, spoľahlivosť systémov alebo prirodzenosť ovládania. Následne je cieľom vytvoriť prototyp novej verzie aplikácie a tieto nedostatky opraviť a prísť s možnými vylepšeniami pre tieto systémy. Práca bude zameraná na implementovanie týchto vylepšení a ich otestovanie v praxi v procese neurorehabilitácie, kde tieto vylepšenia umožnia implementáciu prirodzenejšej interakcie s objektmi a manipuláciu s predmetmi, umožnia terapeutovi prirodzenejšiu interakciu v zdieľanom virtuálnom prostredí pomocou snímania rúk a bude spojená externá komunikácia so serverom na zber dát o procese rehabilitácie.

Konečný prototyp aplikácie by mal byť technicky spoľahlivý, prirodzený, ale aj jednoducho nastaviteľný pre veľkú škálu rehabilitovaných pacientov s cieľom zvýšenia efektivity liečby v následnej plnej implementácii pri klinickej praxi.

2 VR v neurorehabilitácii

Virtuálna realita sa za posledné roky v medicíne stala významnou technológiou, ktorá má potenciál zrýchliť a zefektívniť liečbu ochorení fyzického aj psychologického charakteru. Táto technológia sa zvlášť osvedčila v oblasti neurorehabilitácie, kde významne pomáha pacientom zotavujúcim sa po mozgovej mŕtvici a iných neurologických poruchách.

2.1 Využitie VR v neurorehabilitácii

Využitie virtuálnej reality v neurorehabilitácii prináša množstvo výhod. Toto tvrdenie sa preukázalo už vo viacerých prieskumoch, kde väčšina terapeutov potvrdila jej veľmi dobrú realizovateľnosť, pričom motivácia pacientov pri tejto terapii je taktiež veľmi vysoká [1].

Proces neurorehabilitácie často zahŕňa umiestnenie pacienta do imerzívneho virtuálneho prostredia, vytvoreného vo vývojovom prostredí ako napríklad Unity a prezentovaného prostredníctvom VR headsetu, napríklad Meta Quest. Pacient je pri cvičeniach navigovaný terapeutom, pričom sa sústreďuje na cvičenia a pohyby vykonávané postihnutou hornou končatinou [1]. Moderné systémy môžu navyše využívať snímanie mozgovej aktivity pacienta, ktoré poskytuje dáta pre lepšie prispôsobenie a optimalizáciu liečebného procesu [2].

Napriek mnohým výhodám sa u menšej skupiny pacientov vyskytujú nežiadúce reakcie, ktoré sú celkovo spájané s používaním VR, ako nevoľnosť, bolesť hlavy alebo iné sprievodné medicínske problémy, ktoré môžu obmedzovať použitie virtuálnej reality ako univerzálneho riešenia neurorehabilitácie [2].

2.2 Systémy XR

XR (Extended Reality) sa ako termín v súčasnosti využíva ako pomenovanie pre viaceré formy virtuálnej a rozšírenej reality, medzi ktoré patria podsystémy ako virtuálna realita (VR), zmiešaná realita (MR) a rozšírená realita (AR) [3]. Tieto systémy spoločne dokážu pri neurorehabilitácii vytvoriť dôveryhodné virtuálne prostredie, s ktorým môže pacient interagovať a byť pod kontrolou terapeuta počas procesu liečby.

2.2.1 XR hardvér

V posledných rokoch sa na trhu začali objavovať cenovo dostupné XR zariadenia, no žiadne z nich nezískalo takú popularitu ako zariadenia spoločnosti Meta, najmä jej modelový rad Quest [4]. Vďaka ich cenovej dostupnosti a technologickej výbave sa tieto zariadenia začali vo veľkom používať v klinických aplikáciách a rôznych druhoch rehabilitačnej liečby. Najčastejšie využívanými modelmi sú zariadenia Meta Quest 2 a Meta Quest 3. Obe tieto zariadenia obsahujú technológiu snímania pohybu rúk (hand tracking) a obnovovacia frekvencia ich displeja (72-120 Hz) minimalizuje nepohodlie pri používaní headsetu [4].

2.2.2 Softvér pre vývoj XR aplikácií

Vývojové prostredia ako Unity alebo Unreal Engine sa stali štandardom pre vývoj XR prostredí a aplikácií, vrátane tých určených pre neurorehabilitáciu. Ich hlavným benefitom je široká dostupnosť dokumentácie a mnohých balíkov poskytujúcich nevyhnutnú funkcionálnu pre imerzívne a interaktívne VR prostredia. To zefektívňuje ich vývoj a implementáciu novej funkcionality. Tieto prostredia tak tiež umožňujú vývoj zdieľaných prostredí pre viac hráčov, ktoré sú dôležité pre neurorehabilitáciu, kde je potrebná interakcia pacienta s terapeutom v reálnom čase [4].

2.3 Hand tracking

Vo väčšine prípadov sa pri tomto druhu terapie využívajú už spomenuté systémy rozšírenej reality (XR), ktoré podporujú technológiu hand tracking, teda technológiu umožňujúcu ovládanie vo virtuálnom prostredí bez potreby fyzických ovládačov, čím sa zvyšuje imerzia pacienta a umožňuje mu prirodzene interagovať s týmto prostredím, čo môže viesť k lepším výsledkom terapie.

Technologicky je v dnešnej dobe hand tracking dostupný v mainstreamových vývojových prostrediach ako Unity a Unreal Engine. Vývojári môžu využiť balíčky ako XR Hands a XR Interaction Toolkit, ktoré sú v Unity dostupné, ktoré poskytujú komplexné nástroje na zaznamenávanie dát o pohybe rúk a na interakciu s objektmi vo virtuálnom prostredí [4]. Tieto nástroje vedia zabezpečiť detailné dáta o aktuálnom stave ruky a tým výrazne zlepšiť proces ovládania vytvoreného prostredia.

Moderné VR headsety ako Meta Quest alebo Apple Vision sú taktiež vybavené integrovanou podporou pre túto technológiu, čo eliminuje potrebu externých senzorov a zjednodušuje používanie týchto systémov pre pacientov a terapeutov.

2.4 WebSocket komunikácia

Využitie spoľahlivých a rýchlych sieťových protokolov je nevyhnutné pre zabezpečenie plynulej komunikácie v reálnom čase medzi pacientom a terapeutom. V kontexte VR aplikácií pre neurorehabilitáciu sa na tento účel využíva v mnohých prípadoch protokol WebSocket [5]. Na rozdiel od bežného HTTP protokolu, ktorý funguje na princípe jednoduchých jednorazových požiadaviek a odpovedí, WebSocket komunikácia poskytuje plne obojsmernú komunikáciu s veľmi nízkou odozvou pomocou TCP spojenia.

V systémoch pre neurorehabilitáciu je protokol WebSocket využívaný pri prenose dôležitých telemetrických dát [6]. Môžu to byť napríklad dáta sledujúce motoriku a pohyby pacienta vo virtuálnom priestore, na základe ktorých vieme dynamicky upravovať náročnosť neurorehabilitačného cvičenia [5].

3 Existujúce riešenia

V tejto kapitole sú predstavené existujúce riešenia a technológie využívané v oblasti neurorehabilitácie vo virtuálnej realite. Súčasťou analýzy sú existujúce terapeutické aplikácie, vývojové prostredia a balíky používané pri implementácii systémov virtuálnej reality, hand trackingu a externej komunikácie pomocou protokolu WebSocket. Cieľom tejto analýzy je identifikovať vhodné nástroje a prístupy pre implementáciu novej verzie systému L-NeRVEn.

3.1 Existujúce aplikácie pre neurorehabilitáciu

CUREO

CUREO¹ je terapeutický softvér využívajúci technológiu virtuálnej reality na podporu motorických a kognitívnych funkcií pacientov počas neurorehabilitácie. Aplikácia obsahuje viac ako 50 interaktívnych cvičení rozdelených do siedmich kategórií a umožňuje prispôsobenie rehabilitácie pre každého pacienta individuálne.

Počas rehabilitácie pacienti používajú VR headsety Meta Quest 2, prostredníctvom ktorých sú umiestnení do virtuálneho prostredia, kde vykonávajú rôzne pohyby a cviky v motivujúcich scenároch, kde je pod dohľadom terapeuta, ktorý má kontrolu nad celým procesom pomocou tabletu [1].

Pokrok pacienta sa počas procesu liečby automaticky zaznamenáva, čo terapeutovi umožňuje mať prehľad o pokroku pacientovej liečby a nastaviť ďalší postup.

Systém umožňuje realizovať rehabilitáciu individuálne, ale aj skupinovo, kde sa jej môže zúčastniť až 6 pacientov súčasne s individuálnym nastavením pre každého z nich.

¹<https://www.cureosity.com>

ClaustrOFF: Overcome Your Fears

ClaustrOFF² je terapeutická aplikácia virtuálnej reality, ktorá sa zameriava na postupné prekonávanie klaustrofóbie (strachu z uzavretých priestorov) a s ňou súvisiacich stavov úzkosti. Riešenie využíva imerzívne virtuálne prostredie, kde pacienti čelia rôznym scenárom, ktoré simulujú typické klaustrofobické situácie (napr. výťahy, uzavreté miestnosti alebo magnetická rezonancia) a učia používateľa lepšie zvládať tieto stresové situácie.

Aplikácia je dostupná pre platformy Meta Quest a PICO, a je vyvinutá v prostredí Unreal Engine. Pacient je počas používania hravou formou motivovaný riešiť hádanky alebo prechádzať dobrodružným príbehom. Každý scenár je navrhnutý tak, aby pacient mohol postupovať vlastným tempom. Pacient taktiež získava priebežne informácie o svojom pokroku. Aplikácia je dostupná pre domáce použitie ale taktiež je možné túto aplikáciu použiť v sprievode mentora počas terapie.

Vďaka interaktívnemu dizajnu, ktorý využíva technológiu hand tracking a možnosti personalizácie cvičenia, táto aplikácia patrí medzi moderné terapeutické riešenia na liečbu duševného zdravia.

XRHealth

XRHealth³ je terapeutické riešenie, využívajúce virtuálnu realitu pri neurorehabilitácii pacientov liečiacich sa z rôznych ochorení ako Parkinsonova choroba, rôzne druhy autizmu a pacientov zotavujúcich sa po mozgovej príhode. Systém je taktiež možné využiť pri terapii úzkostných stavov a fóbii [7]. Systém umožňuje pacientom podstupovať personalizované terapeutické programy vo virtuálnom prostredí pod dohľadom terapeuta.

Pri rehabilitácii pacient používa VR headset, ktorý umožňuje interakciu s virtuálnym prostredím, ktoré je na základe vykonávaného cvičenia individuálne prispôsobené. Kompatibilné platformy s XRHealth sú zariadenia ako Meta Quest a taktiež zariadenia od HTC a PICO.

XRHealth taktiež disponuje aj mobilnou aplikáciou a webovým portálom, ktorý umožňuje pacientovi a terapeutovi komunikovať a sledovať pokrok počas liečby.

²<https://gametherapy.eu/claustroff>

³<https://www.xr.health>

Neuro Rehab VR

Neuro Rehab VR⁴ je VR systém určený na rehabilitáciu širokého spektra diagnóz od ortopedických, geriatrických až po vážnejšie neurologické stavy. Tento systém je navrhnutý ako prenosný a bezdrôtový a umožňuje teda jeho nastavenie v každom prostredí.

Systém obsahuje viac ako 30 aktivít zameraných na precvičenie horných a dolných končatín, krku a trupu, ale aj cvičenie koordinácie a rovnováhy a iných kognitívnych schopností. Systém umožňuje terapiu v sede, stojí, ale aj pri ľahu na lôžku a je preto vhodný pre mnohých pacientov nezávisle od ich fyzického obmedzenia. Terapeutovi poskytuje možnosť upravovať cvičenia úpravou parametrov jednotlivo pre každého pacienta a sledovať ich pokrok počas liečby aj za pomoci AI systémov.

3.2 Existujúce nástroje pre vývoj

3.2.1 Prostredia

Unreal Engine

Unreal Engine je výkonné a zároveň patrí medzi najpokročilejšie prostredia pre vývoj 3D aplikácií využívaných v hernom a filmovom priemysle, ale aj vo vede a architektúre. Tento engine sa vyznačuje možnosťou kvalitnej grafiky a realistických vizuálnych efektov a obsahuje tiež nástroje pre moderný rendering a optimalizáciu.

Pre vývoj aplikácií VR, Unreal Engine ponúka priamo vstavané nástroje a dedikovaný režim priamo v prostredí enginu. Ponúka možnosť programovania pomocou tzv. Blueprint vizuálneho skriptovacieho systému, ktorý výrazne urýchľuje vývoj bez potreby klasického programovania. Súčasťou sú aj nástroje na podporu technológie hand tracking pričom podporuje najpopulárnejšie VR headsety ako Meta Quest, HTC Vive, Valve Index a mnohé ďalšie

Vďaka jeho vlastnostiam je Unreal Engine vhodný na vytváranie detailných VR prostredí s vysokou mierou interaktivity s možnosťou efektívneho vývoja. Miernou nevýhodou sú náročnejšie systémové požiadavky, čo vo viacerých prípadoch neumožňuje beh aplikácie priamo na hardvéri headsetov ako Meta Quest. Jeho dokumentáciu a ďalšie informácie je možné nájsť na oficiálnej stránke [8].

⁴<https://neurorehabvr.com/smart-therapy-complete-solution>

Unity

Unity patrí medzi najpoužívanejšie multiplatformové enginy pre vývoj 2D a 3D interaktívnych aplikácií používaných v hernom priemysle a vedeckej oblasti. Je známy svojou dostupnosťou, rozsiahlou dokumentáciou a veľkou komunitou.

Podobne ako Unreal Engine, Unity má vstavané nástroje pre vývoj VR aplikácií a podporu pre mnohé headsety. Umožňuje taktiež podporu mnohých balíkov pridávajúcich funkcie potrebné pre XR prostredia a balíky potrebné pre technológiu hand tracking.

Jeho hardvérové požiadavky sú v porovnaní s Unreal Engine miernejšie a preto je tento engine ideálny pre VR aplikácie v neurorehabilitácii, ktoré nepotrebujú realistické grafické spracovanie a chcú dosiahnuť funkcionality priamo na headsete. Ďalšie informácie a dokumentáciu je taktiež možné nájsť na oficiálnej stránke [9].

Záver

Obe platformy poskytujú rozsiahlu podporu pre vývoj VR aplikácií a technológie hand tracking. Keďže systém L-NeRVEn je implementovaný v prostredí Unity, ďalší vývoj novej verzie prirodzene pokračuje využitím tohto enginu a jeho XR balíkov. Výhodou Unity v kontexte projektu zostávajú najmä nižšie hardvérové požiadavky, rozsiahla dokumentácia a dobrá podpora zariadení Meta Quest.

3.2.2 Balíky VR integrácie prostredia Unity

XR Hands

XR Hands je balík, ktorý poskytuje detailné sledovanie rúk pomocou 26 sledovateľných bodov na každej ruke [10]. Balík umožňuje aj vizualizáciu týchto rúk a poskytuje údaje o ich polohe v reálnom čase, a tak umožňuje efektívnu prácu so vstupmi, ktoré využíva hand tracking.

XR Interaction Toolkit

XR Interaction Toolkit je balík určený pre interakciu s virtuálnymi objektmi vo VR a AR prostrediach [11]. Umožňuje implementáciu interakcií, ako sú uchopenie, dotyk alebo otočenie. Poskytuje vstavané komponenty pre kompletný interakčný systém snímania pohybu rúk.

Final IK

Final IK je balík určený pre animáciu avatarov pomocou pokročilej inverznej kinematiky [12]. V prípade VR aplikácií je ho možné použiť pri animácii rúk a tela na základe dát z headsetu v reálnom čase.

Meta XR All-in-One SDK

Balík nástrojov oficiálne poskytnutý od spoločnosti Meta pre zariadenia Meta Quest. Poskytuje plnú podporu pre vývoj aplikácií virtuálnej reality. Zahrňa nástroje pre technológie sledovania pohybu, interakcie a snímania pohybu rúk. SDK je voľne dostupné na stránke spoločnosti [13].

Ready Player Me

Balík umožňujúci rýchle vytváranie a implementáciu 3D avatarov do aplikácií vytvorených v Unity alebo Unreal Engine [14].

Balík umožňuje personalizáciu avatarov podľa potrieb používateľa. Upravené môžu byť doplnky, oblečenie a mnohé ďalšie parametre. Následne sú tieto avatary jednoducho implementované vo vybranom prostredí.

3.2.3 Balíky pre externú komunikáciu pomocou WebSocket

Mirror

Mirror je open source sieťový framework, uľahčujúci implementáciu multiplayer funkcionality v prostredí Unity. Zabezpečuje prenos dát medzi používateľmi na princípe serverovej autority a dokáže fungovať aj v režime, kde je jeden používateľ zároveň aj serverom aj klientom. Dokumentácia ale aj zdrojový kód je vďaka open source licencií voľne dostupný [15]. Pre WebSocket komunikáciu ponúka robustné moduly pomocou, ktorých umožňuje implementovanie synchronizácie stavu virtuálneho prostredia.

FishNet

FishNet je moderný sieťový framework pre platformu Unity zameraný na vysoký výkon a škálovateľnosť. Je vhodný pre komplexné multiplayer riešenia, kde je potrebná robustná synchronizácia stavov, hernej logiky a objektov. Podobne ako Mirror, aj FishNet podporuje WebSocket komunikáciu prostredníctvom transportných modulov. Na rozdiel od frameworku Mirror nepodporuje princíp

server authority, no vďaka novej technológii je tento framework efektívne a výkonnejšie. Jeho dokumentáciu je možné nájsť na oficiálnej stránke vývojára [16].

Fleck

Na rozdiel od balíkov Mirror a FishNet, ktoré sú komplexnejšími sieťovými riešeniami, Fleck je odľahčená open-source knižnica napísaná v jazyku C#. Jej hlavný účel je jednoduché a rýchle implementovanie WebSocket servera. Server implementovaný v tejto knižnici je schopný veľmi dobre spracovať dáta z virtuálneho prostredia a preposielať ich na externý server, kde ich terapeut môže sledovať a ďalej využiť. Jej zdrojový kód je verejne dostupný na platforme GitHub [17].

3.3 Vyhodnotenie záverov pre implementáciu

Na základe vykonanej analýzy vývojových nástrojov a dostupných technológií je možné formulovať odporúčania pre ďalšiu implementáciu a vylepšenia systému. Použitý engine poskytuje širokú podporu pre VR technológie, nízke systémové požiadavky a zároveň zabezpečuje kompatibilitu s už existujúcim systémom, ktorý je v ňom implementovaný.

Čo sa týka samotnej implementácie systému hand tracking, jeho základ tvoria už spomínané balíčky XR Hands a XR Interaction Toolkit, ktoré sa preukázali byť ako dostatočné pre potreby implementácie tohto systému v neurorehabilitácii. Nástroje ako Meta XR All-in-One SDK ponúkajú vysoko optimalizované funkcie pre zariadenia Quest, no použitie univerzálnejšieho štandardu ako OpenXR zabezpečuje kompatibilitu s väčšou škálou zariadení.

Pre animáciu a polohu rúk naďalej zostáva vhodným riešením balík Final IK, ktorý pomocou inverznej kinematiky efektívne pomáha s výpočtom na základe vstupov z headsetu a ovládačov.

V oblasti externej komunikácie sa ako najvhodnejšie riešenie naďalej javí využitie protokolu WebSocket, ktorý umožňuje obojsmernú komunikáciu s nízkou odozvou v reálnom čase. Na základe vykonanej analýzy možno konštatovať, že na prenos telemetrických údajov sa ako najvhodnejšie riešenie javí využitie knižnice Fleck, ktorá je použitá aj v aktuálnej verzii systému. Čo sa týka sieťovej vrstvy zodpovednej za synchronizáciu virtuálneho prostredia medzi pacientom a terapeutom, plánovaná je refaktORIZÁCIA na framework FishNet, ktorý bude využívaný v novej verzii aplikácie. FishNet poskytuje modernejšie a výkonnejšie riešenie pre synchronizáciu stavov objektov a komunikáciu medzi klientom a serverom.

V závere možno konštatovať, že aktuálna implementácia systému síce overila technickú realizovateľnosť použitia týchto nástrojov, no stále je priestor pre značnú inováciu týkajúcu sa kvality ich integrácie do systému. Cieľom bude využiť potenciál týchto nástrojov a ich novších verzií na vylepšenie stability, precíznosti a celkového používateľského zážitku pacienta pri ovládaní systému počas terapie a odstránením nedostatkov zo súčasnej verzie systému.

4 Aktuálny stav projektu

V tejto kapitole je predstavený aktuálny stav projektu systému L-NeRVEn a jeho existujúca implementácia, na ktorú nadväzuje návrh prototypu novej verzie systému popísanej v nasledujúcich kapitolách.

Pojem aktuálna alebo súčasná verzia systému v rámci tejto práce označuje implementáciu systému vytvorenú v rámci predchádzajúcich záverečných prác Petra Nehilu, Adama Mikaela a Damiana Chmuru. Tieto práce sa venovali návrhu zdieľaného virtuálneho prostredia, implementácii hand trackingu, reprezentácii používateľa a systému externej komunikácie pomocou protokolu WebSocket. Táto práca na uvedené riešenia nadväzuje a rozširuje ich o novú architektúru používateľského ovládania a externej komunikácie.

4.1 Aktuálna implementácia

Aktuálny projekt obsahuje komplexnú implementáciu systému, ktorý je postavený na platforme Unity a poskytuje aj multiplayer funkcionality pomocou sieťového frameworku Mirror.

Hand tracking funkcionality

Samotný hand-tracking je implementovaný pomocou balíkov XR Hands a XR Interaction Toolkit. Ruky sú pripojené k avataru pomocou komponentu *VR IK* poskytnutý balíkom FinalIK, ktorý vie na základe pohybu rúk synchronizovať pohyby celého avatara [18]. Pohyb prstov zabezpečuje komponent *Finger Rig*, ktorý zároveň umožňuje nastavenie váh a rotácií jednotlivých kostí.

Hybridné ovládanie

Systém obsahuje aj možnosť hybridného ovládania, kde je možné ovládať prostredie pomocou rúk a ovládačov zároveň, a voľne medzi týmito dvoma spôsobmi ovládania prepínať. Bežiaci skript sleduje, či sú v danom momente aktívne

ovládače alebo ruky a na základe toho určuje cieľové body pre animáciu. Túto funkcionality sprostredkujú event handlers prostredia Unity pomocou *XR Interaction Modality Managera*

Interakčný systém

OpenXR rig implementuje interaktory Poke a NearFar, ktoré umožňujú interakciu s objektmi na blízko aj na diaľku. Druhy implementovaný rig Meta XR tak tiež implementuje interaktory ako Hand Poke, Hand Grab, Distance Hand Grab a Hand Ray Interactor, ktorých kombinácia zabezpečuje interakciu, otáčanie a teleportáciu hráča vo virtuálnom prostredí.

Škálovanie avatara a rúk

Implementovaný je aj systém škálovania, ktorý umožňuje detailné nastavenie veľkosti avatara a jeho jednotlivých rúk. Obsahuje triedy *AvatarSettings*, *AvatarController* a *AvatarSetup*, ktoré spravujú zmeny veľkosti avatara a ich následne aplikovanie na model.

Sieťová synchronizácia a externá komunikácia

Sieťová časť je implementovaná pomocou už spomínaného frameworku Mirror. Umožňuje synchronizáciu pohybov a prstov aby boli viditeľné pre všetkých pripojených používateľov v prostredí. Používajú sa komponenty Network Transform na serveri, ktoré s využitím UDP protokolu umožňujú synchronizáciu v reálnom čase.

Okrem hlavnej serverovej komunikácie systém implementuje aj externú komunikáciu protokolom WebSocket. Slúži na komunikáciu medzi Unity serverovým klientom a EEG komponentom cez platformu OpenViBE. Samotná implementácia tejto komunikácie používa už spomínanú knižnicu Fleck a zabezpečuje obojsmerný prenos telemetrických údajov v reálnom čase. Konkrétne je prostredníctvom tejto komunikácie možné prenášať údaje o stave rehabilitačného cvičenia, motor imagery udalosti a ďalšie informácie potrebné pre synchronizáciu terapeutického procesu medzi VR klientom a EEG systémom.

4.2 Nedostatky systému

Napriek tomu, že súčasná implementácia systému hand tracking je robustná a spĺňa funkčné požiadavky, stále naďalej vykazuje nedostatky identifikované už v predchádzajúcej práci, ktorá sa venovala implementácii tohto systému [4]. Tieto nedostatky predstavujú príležitosť pre optimalizáciu a na nájdenie efektívnych riešení pre tieto problémy.

Problémy výkonu

Hoci problémy výkonu nesúvisia priamo s funkcionalitou hand trackingu, vieme konštatovať, že jedným z výraznejších problémov predstavuje záťaž CPU spôsobená GC.Collect, čo naznačuje potenciálne neoptimálnu správu objektov prostredia [4]. Logovanie a aktualizácia UI inputov pomocou `UIInputModule.Update()` tiež predstavuje výraznejšie zaťaženie.

Sieťový prenos

Sieťová náročnosť hand tracking systému je relatívne vysoká. Synchronizácia 364 float hodnôt pre prsty oboch rúk predstavuje 65 kB/s na každého používateľa [4]. Spolu s ostatnými synchronizačnými požiadavkami a komponentmi *Network Transform* pri vyššom počte používateľov, môže byť celkový prenos dát a plynulosť sieťovej komunikácie negatívne ovplyvnená.

Limitácie rigov

Implementované rigy sa odlišujú najmä úrovňou integrácie jednotlivých funkcionalít a konzistenciou ovládania. OpenXR rig podporuje hand tracking aj hybridné ovládanie pomocou rúk aj ovládačov zároveň, no aktuálna verzia nemá tieto funkcionality plne integrované vo všetkých scénach aplikácie. Naopak MetaXR rig poskytuje natívnejší prístup k interakčným systémom, pričom hybridné ovládanie je v tomto prípade realizované efektívnejšie. Jeho implementácia je však momentálne obmedzená prevažne na testovacie prostredie.

Rozdiely medzi jednotlivými rigmi so sebou prinášajú značnú mieru nekonzistentnosti a komplikujú ďalší vývoj systému. Pri vývoji novej verzie systému by preto bolo vhodné zjednotiť funkcionality oboch riešení a implementovať jednotný systém ovládania naprieč celou aplikáciou.

Vizualizácia a škálovanie rúk

Aktuálne riešenie škálovania a vizualizácie rúk hráča vykazuje nedostatky vo forme plynulosti a už spomenutej sieťovej synchronizácie. Prepínanie medzi virtuálnymi rukami a rukami Ready Player Me avatara vyžaduje manuálne ovládanie komponentmi, čo pri rýchlych a častých zmenách môže spôsobiť nekonzistentnosti v ich vizualizácii.

Vylepšený scaler síce umožňuje detailnú úpravu veľkosti rúk a iných častí avatara, no propagácia týchto zmien na serveri, aby sa aktualizovali pre ostatných používateľov ešte nie je ideálna, keďže dochádza k nesprávnemu zobrazovaniu v niektorých prípadoch.

Nedostatok a limitácia scenárov

Aktuálne riešenie vykazuje značnú závislosť na jedinom implementovanom prostredí. Meta XR rig je integrovaný výlučne v testovacej scéne, zatiaľ čo jadro aplikácie naďalej operuje so starším OpenXR rigom.

Funkcionalita hand trackingu chýba v kľúčových častiach systému, ako je napríklad hlavné menu, čo narúša konzistentnosť ovládania. Súčasná implementácia zároveň predstavuje nevyužitý potenciál, keďže jediné dostupné prostredie je prevažne stacionárne a z hľadiska používateľského zážitku či terapeutickému praxe neponúka dostatočnú mieru interaktivity.

4.3 Záver a návrhy pre implementáciu

Na základe analýzy je zjavné, že napriek robustnej a funkčnej implementácii, existujú konkrétne nedostatky, ktorých odstránenie by vedelo priniesť výrazne zlepšenie implementácie systému. Optimalizácie by umožnili zvýšiť výkon, plynulosť a plne využiť potenciál technológie snímania pohybu rúk prostredníctvom jej hlbšej integrácie do aplikácie.

Návrhy na zlepšenie výkonu

Prvým krokom by malo byť určité riešenie výkonnostných problémov, ktoré aplikácia obsahuje. Konkrétne by to mohlo byť zníženie záťaže CPU spôsobenej správou objektov, a to optimalizovaním volaní GC.Collect a redukciou logovacích výpisov. Bolo by vhodné použiť viac asynchrónny prístup a pooling objektov, aby sa zabránilo zbytočnému ničeniu objektov za chodu aplikácie.

Čo sa týka sieťového prenosu a jeho optimalizácie bolo by vhodné namiesto zastaranejšieho sieťový frameworku Mirror, použiť modernejší a efektívnejší framework FishNet, ktorý poskytuje modernejší a efektívnejší systém synchronizácie a mohol by prispieť k zníženiu sieťovej aj procesorovej záťaže [16].

Pri implementácii WebSocket komunikácie môže byť samotná komunikačná vrstva pre prenos EEG/BCI údajov zachovaná aj v novej verzii systému. Keďže táto vrstva predstavuje samostatnú komunikačnú architektúru, pri budúcej implementácii bude potrebné upraviť prepojenie medzi frameworkom FishNet a WebSocket komunikáciou. Zachovanie existujúcej implementácie by zároveň umožnilo pokračovať vo využívaní už overeného WebSocket protokolu a existujúcej EEG/BCI pipeline.

Návrhy na zlepšenie systému hand trackingu

Veľkým prínosom by bolo zjednotenie implementácie hand trackingu a interaktívnych systémov na jeden univerzálny rig alebo komponent. Tento krok by vedel výrazne zjednodušiť implementáciu a jej následné rozširovanie v budúcnosti. Zároveň by zjednotenie umožnilo viac konzistentné a bezproblémové ovládanie pre používateľa.

Vzhľadom na to, že hlavné headsety, na ktoré je funkcionálna aplikácia primárne zameraná sú tie na platforme Meta Quest je logickým kandidátom pre VR klienta rig MetaXR, keďže využitím natívneho Meta XR All-in-One SDK je možné využiť pokročilejšie možnosti samotného headsetu ako napríklad spomínané hybridné ovládanie na lepšie integrovanej úrovni⁵. Týmto krokom by sa zachovala možnosť hybridného ovládania a celkovo by bol tento systém lepšie integrovateľný. Takáto integrácia by vedela odstrániť závislosť funkcionality od konkrétnych scén a testovacích prostredí. Umožnila by tým implementáciu hand trackingu naprieč celou aplikáciou so zachovaním plnej funkcionality. Avšak naďalej by bolo vhodné ponechať možnosť použitia rigu OpenXR, pre zariadenia iné ako Meta Quest. Pri OpenXR je taktiež možné implementovať hybridné ovládanie, ale

⁵<https://forum.godotengine.org/t/multi-modal-input-for-xr/110603>

za cenu nižšej spoľahlivosti a exkluzivity pre platformu Meta Quest pre túto funkcionálnosť.

Dôležité je taktiež vyriešiť problémy vizualizácie a sieťovej synchronizácie škálovania a pohybu rúk. Prepínanie by bolo možné riešiť automatizovaním prepínacích skriptov medzi virtuálnymi a mesh rukami avatara. Prepojenie týchto rúk by bolo vhodné riešiť ich lepšou integráciou k modelu Ready Player Me alebo využitím alternatívneho či vlastného modelu rúk. Serverová synchronizácia škálovania a pohybu by mohla byť riešená prostredníctvom optimalizovanej synchronizácie pomocou RPC funkcií.

Ďalšie návrhy na zlepšenie

V poslednom rade by aplikácii prospelo rozšírenie scenárov o vyššiu mieru interaktivity. Pre zvýšenie efektivity terapie je dôležité, aby pacient mal k dispozícii rozmanité a pútavé prostredia, ktoré naplno využívajú potenciál hand trackingu. Prirodzená interakcia s prvkami prostredia, ako sú dvere či ovládacie panely, spolu s plynulou interakciou s terapeutom, by výrazne zlepšili používateľský zážitok.

Z dlhodobého hľadiska by systém benefitoval z dôkladnej refaktorizácie a vytvorenia novej verzie, ktorá by tieto zmeny integrovala od základov do jednotného a stabilného znovupoužiteľného systému.

5 Konceptuálny návrh

Pred samotnou implementáciou plného prototypu novej verzie systému bol vytvorený samostatný projekt, ktorý slúžil na overenie kompatibility existujúcich systémov s novou architektúrou aplikácie a aktualizovaným prostredím Unity 6. Hlavným cieľom tohto konceptuálneho prototypu bolo otestovať funkcionality nového systému ovládania vo virtuálnom prostredí, kompatibilitu existujúcich komponentov hand trackingu a overiť možnosti integrácie sieťovej komunikácie a WebSocket vrstvy do pripravovanej verzie systému.

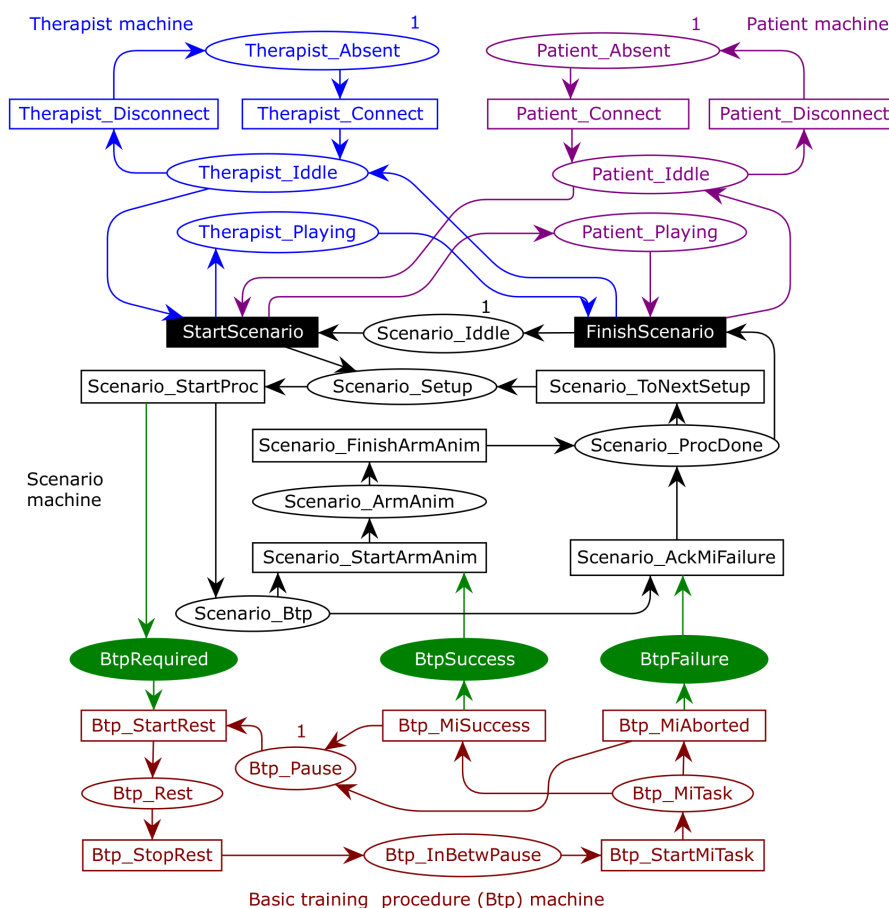
Použitie novej verzie prostredia Unity zároveň umožnilo využitie aktuálnych verzií XR balíkov a modernejších funkcionalít frameworkov MetaXR a OpenXR. V rámci prototypu boli použité novšie verzie balíkov *Meta XR All-In-One SDK* vo verzii 83, *FinalIK* vo verzii 2.4 a aktualizované XR balíky prostredia Unity. Oproti pôvodnej implementácii tak systém získal lepšiu kompatibilitu s novšími zariadeniami, stabilnejšiu podporu hand trackingu a modernejšiu architektúru XR funkcionality.

Na základe analýzy sa pri návrhu novej verzie začalo primárne uvažovať s implementáciou MetaXR rigu a využitím balíka *Meta XR All-in-One SDK*, ktorý poskytuje natívnu podporu zariadení Meta Quest a pokročilé funkcie hybridného ovládania. Súčasťou návrhu bolo zároveň zachovanie kompatibility s OpenXR rigom a jeho úpravy pre podporu ostatných VR zariadení.

V rámci prototypu bola taktiež overovaná kompatibilita novej WebSocket komunikačnej vrstvy s frameworkom *FishNet*, ktorý bol zvolený ako náhrada za pôvodný framework *Mirror*. Implementácia prototypu zároveň umožnila refaktoriáciu existujúcich funkcionalít a ich následné postupné presunutie do novej verzie systému.

5.1 Stavový systém novej verzie

Návrh nového stavového systému, poskytnutý vedúcim práce, predstavuje sústavu stavových diagramov vo forme Petriho siete, kde sú prechody medzi jednotlivými stavmi definované na základe vstupu a aktuálneho stavu v systéme. Na obrázku 5.1 sú tieto navrhované diagramy a prechod medzi nimi znázornené.



Obrázok 5.1: Petriho sieť navrhovaného nového systému

Tieto stavové automaty riadia chod systému od pripojenia terapeuta a pacienta, následne výber scenára počas ktorého prebieha konfigurácia EEG/BCI komponentu a samotná rehabilitácia pod dozorom prítomného terapeuta.

Výhodou tohto prístupu je predovšetkým separácia zodpovedností, keďže každý podsystém má presne definovanú svoju úlohu a dokáže s ostatnými komunikovať prostredníctvom udalostí a prechodov. Paralelný chod niektorých týchto automatov umožňuje vykonávanie rôznych akcií pre terapeuta a pacienta nezávisle od seba. Automat scenára terapie zase čaká na splnenie podmienok, predtým než pokračuje v ďalšom stave. Automaty obsahujú aj vetvy pri neúspešných akciách čo zabezpečuje robustnosť tohto systému.

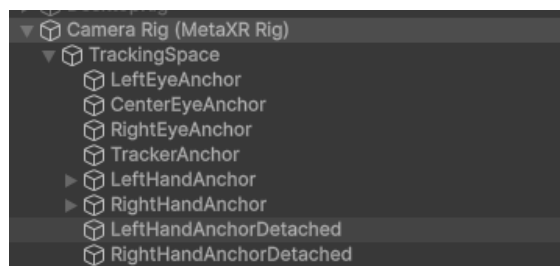
5.2 Konceptuálny prototyp

V prvej fáze implementácie bol vytvorený prototyp, ktorý slúžil na overenie funkcionality MetaXR rigu a zabezpečenie kompatibility s existujúcimi komponentmi predchádzajúcej verzie systému. Prototyp zároveň slúžil na otestovanie a overenie funkčnosti novej implementácie WebSocket komunikácie.

5.2.1 MetaXR rig a test interakcie

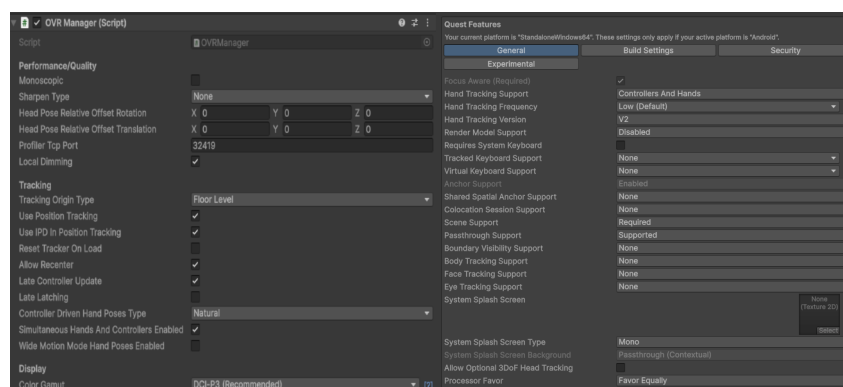
Do vytvoreného Unity prototypu projektu bola pridaná funkcionality MetaXR pomocou oficiálneho balíka Meta XR All-in-One SDK ponúkaného priamo na Unity Asset Store.

Vo vytvorenej scéne bol vytvorený základný XR rig pomocou predpripraveného komponentu *Camera Rig* ponúkaného priamo v balíku MetaXR. Tento komponent obsahuje kameru a controller skripty potrebné na základné ovládanie prostredia pomocou snímania rúk a fyzických ovládačov.



Obrázok 5.2: Základná štruktúra MetaXR rig komponentu v scéne

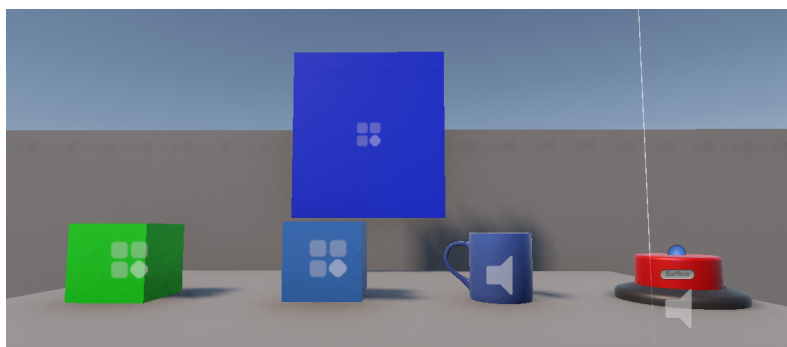
Dôležitou súčasťou tohto komponentu je *OVManager*, ktorý obsahuje dôležité nastavenia pre správanie rigu. Obsahuje nastavenia spôsobu ovládania, verzie snímania rúk, transformy pre rotáciu a polohu kamery a nastavenie hybridného ovládania.



Obrázok 5.3: Nastavenia OVManager skriptu

Ďalšou súčasťou tohto rigu je implementovanie jednotlivých rôznych spôsobov interakcie s objektmi v scéne. Tieto interakcie sú vo forme skriptov pre ovládanie ovládačmi, ale aj pre ovládanie snímaním rúk.

Scéna prototypu obsahuje aj jednoduché mesh objekty, ktoré umožňujú otestovanie týchto interakcií. Medzi implementovanými interakciami je klasické priame uchopenie objektu pomocou HandGrab interaktora, uchopenie objektu na diaľku pomocou DistanceHandGrab interaktora, interakcia pomocou tzv. raycastu, či interakcia s tlačidlami a UI elementami za pomoci Poke interaktora.



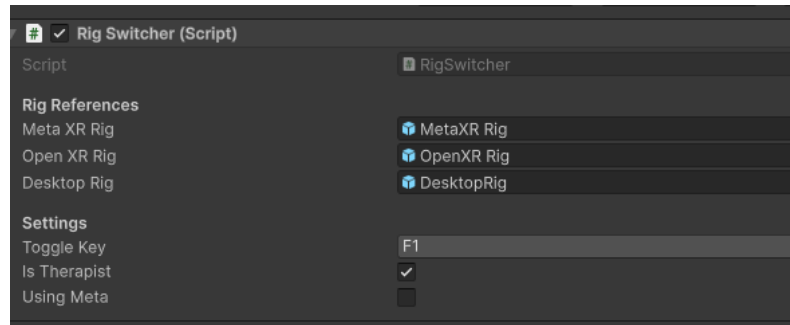
Obrázok 5.4: Jednoduché testovacie objekty interakcie v scéne

Pri vývoji ak nie je dostupné VR zariadenie, je na testovanie možné využiť funkciu simulátora, ktorú balíček Meta taktiež ponúka a bola tiež do prototypu zakomponovaná.

5.2.2 Kompatibilita s OpenXR

Hoci systém primárne cieľi na zariadenia Meta, z analýzy vyplynulo, že pre systém je dôležitá aj kompatibilita s headsetmi od iných výrobcov. Preto systém naďalej implementuje aj OpenXR rig, ktorý slúži ako alternatíva pre tieto zariadenia.

Na obrázku 5.5 je možné vidieť nastavenia skriptu *RigSwitcher*, kde hodnota *Using Meta* umožňuje určiť, či bude pri inicializácii systému použitý MetaXR rig alebo OpenXR rig na základe typu pripojeného VR zariadenia. Pri vypnutej hodnote *Using Meta* bude systém v danom builde používať výhradne OpenXR rig bez podpory MetaXR funkcionality.



Obrázok 5.5: Nastavenia RigSwitcher skriptu.

Samotný skript RigSwitcher zároveň pri štarte aplikácie deteguje prítomnosť VR zariadenia a na základe jeho typu inicializuje správny rig v scéne.

Výpis 5.1: Detekcia typu VR zariadenia

```
RigType DetectRigType()
{
    bool isVRActive = XRSettings.isDeviceActive && XRSettings.enabled;

    if (!isVRActive)
    {
        return RigType.Desktop;
    }

    if (IsMetaDeviceConnected())
    {
        return RigType.MetaXR;
    }

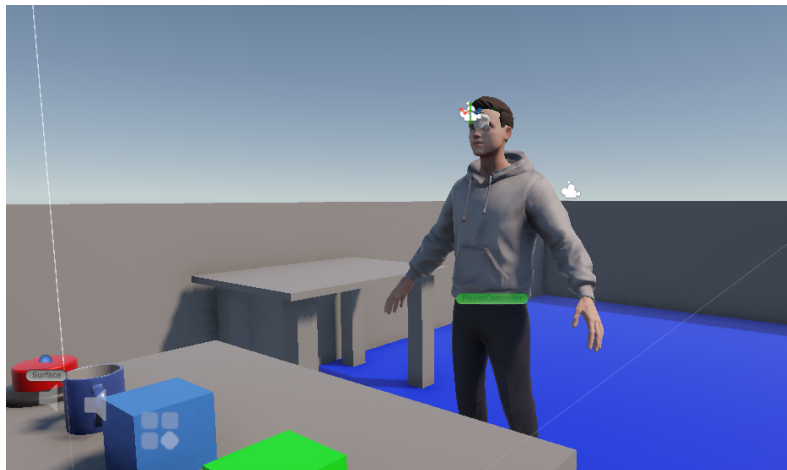
    return RigType.OpenXR;
}
```

Tam metóda `IsMetaDeviceConnected()` kontroluje názvy pripojených vstupných zariadení a inicializáciu *OVRPlugin* komponentu na určenie, či je pripojené Meta zariadenie. Ak nie, tak systém inicializuje OpenXR rig pre ostatné headsety.

Samotný OpenXR rig je v tomto prototype implementovaný len na základnej úrovni, keďže sa tu primárne testovala funkcionálnosť MetaXR rigu. Jeho implementácia v prototype slúžila predovšetkým na overenie kompatibility nových skriptov, správneho prepínania medzi rigmi a zachovania podpory pre VR zariadenia iné ako tie od spoločnosti Meta. Úpravy funkcionality a štruktúry OpenXR rigu sú súčasťou implementácie plnej verzie prototypu systému popísanej v nasledujúcej kapitole.

5.2.3 Integrácia avatarov a desktop rozhrania

V rámci prototypu bola otestovaná aj integrácia avatarov Ready Player Me z predchádzajúcej verzie. Už existujúci avatar bol presunutý do implementovaných rigov. Kde zatiaľ k hlbším upravám implementácie nedošlo a naďalej je využívaný balík FinalIK pri jeho animáciách využitím inverznej kinematiky.



Obrázok 5.6: Avatar Ready Player Me z predchádzajúcej verzie v scéne

Keďže pre tento systém je potrebná možnosť ovládania aj za pomoci klávesnice a myši bez VR zariadenia, bol do prototypu implementovaný aj vlastný desktop rig. Podobne ako XR rig obsahuje kameru a controller skripty, konkrétne *DesktopMovement* a *DesktopInteraction*, ktoré implementujú pohyb pomocou kláves a interakciu s objektmi a UI elementami pomocou myši. Za jeho vybratie pri štarte aplikácie je zodpovedný taktiež už vyššie spomenutý skript *RigSwitcher*, pomocou ktorého je tento rig aktivovaný, ak nie je momentálne pripojené VR zariadenie. Je taktiež implementovaná možnosť prepínania sa do tohto rigu klávesou, keď je aktívne VR zariadenie pri verzii pre Windows.

5.2.4 WebSocket komunikácia

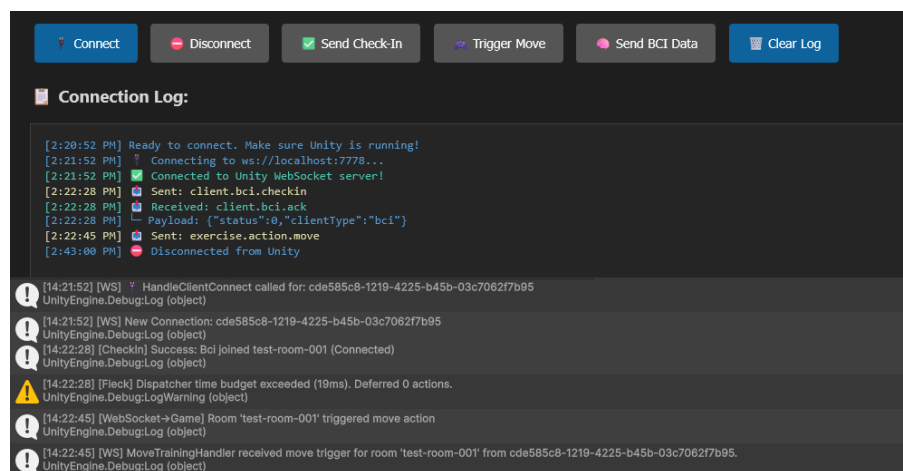
V tomto konceptuálnom prototypu bola taktiež otestovaná komunikácia pomocou protokolu WebSocket. Prototyp obsahuje implementáciu už refaktORIZOVANÉHO klienta WebSocket komunikácie, ktorú poskytol kolega Damian Chmura a je bližšie popísaná v jeho práci [19].

Keďže tento klient a jeho handlers boli pôvodne implementované pomocou knižnice Mirror, bolo potrebné ich refaktORIZOVAŤ tak, aby využívali volania frameworku FishNet, ktorý bude v novej verzii použitý na sieťovú komunikáciu.

Konkrétne bol upravený samotný WebSocket manažér a následne aj handlers zodpovedné za spracovanie jednotlivých typov správ.

V rámci prototypu bola otestovaná komunikácia medzi Unity klientom a externým EEG/BCI systémom OpenViBE pomocou vytvorenej testovacej stránky. Táto implementácia umožnila overiť správne spracovanie správ typu BCI, Check-In a Move, synchronizáciu klientov pomocou *roomId* identifikátorov a základnú funkcionálnu komunikáciu medzi klientom a serverom.

Výsledkom testovania bolo overenie kompatibility WebSocket komunikačnej vrstvy s frameworkom FishNet a pripravenie základnej architektúry pre implementáciu plnej EEG/BCI komunikácie v novej verzii systému.



Obrázok 5.7: Testovanie WebSocket komunikácie

5.2.5 Závery pre plnú implementáciu

Výsledky konceptuálneho prototypu a diskusia s mentorom práce a ostatnými riešiteľmi projektu, potvrdili vhodnosť navrhovanej architektúry pre implementáciu novej verzie systému. Overená bola funkcionálna MetaXR rigu, kompatibilita s OpenXR zariadeniami, základná integrácia desktop rozhrania a funkcionálna WebSocket komunikácia s frameworkom FishNet.

Prototyp zároveň umožnil refaktorizáciu existujúcich systémov a pripravil základ pre implementáciu plnej verzie aplikácie, ktorá je popísaná v nasledujúcej kapitole.

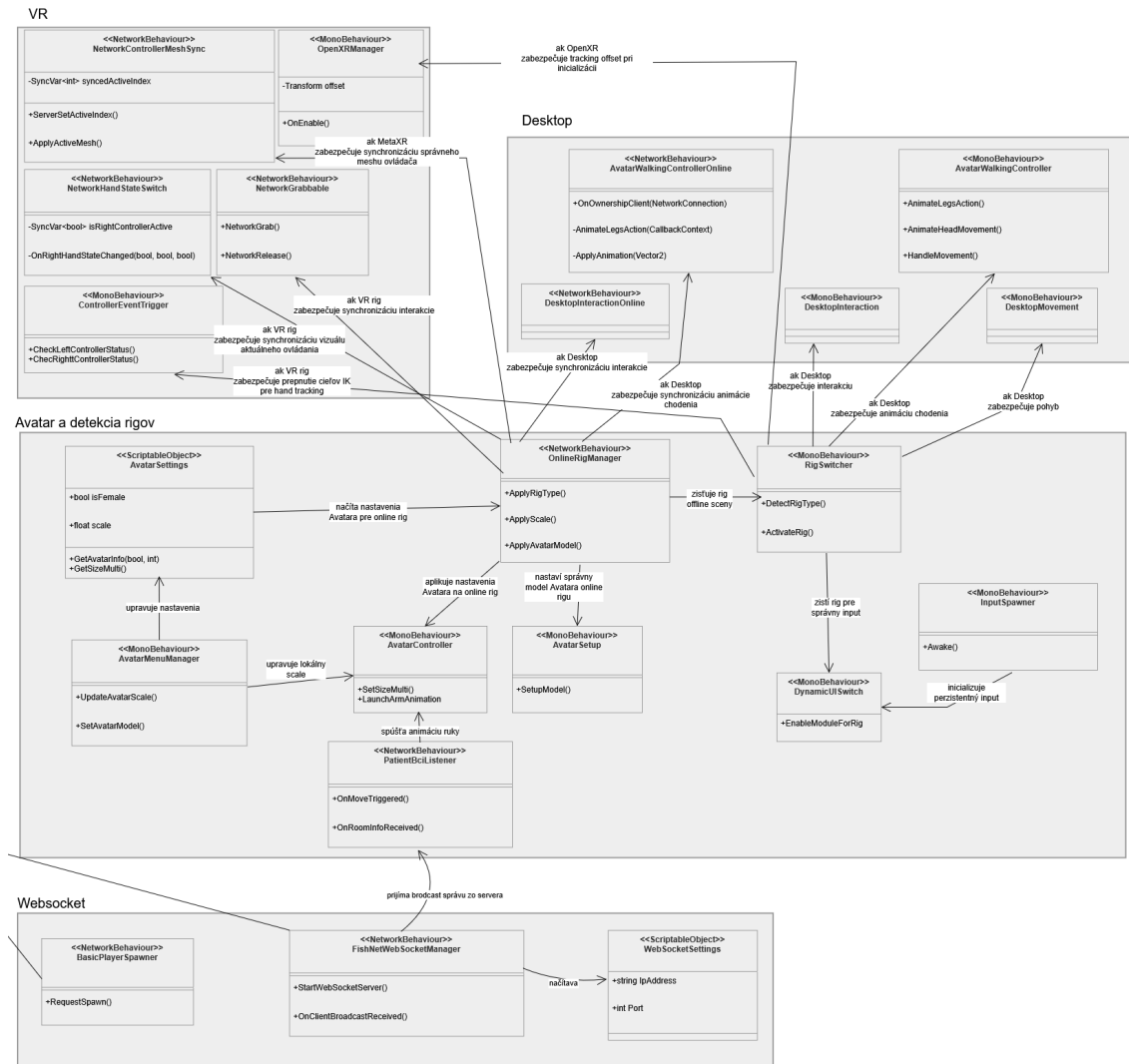
6 Prototyp novej verzie

Na základe konceptuálneho návrhu a výsledkov predchádzajúceho prototypu bol vytvorený prototyp novej verzie systému L-NeRVEn implementovaný v prostredí Unity 6. Vývoj tejto verzie bol koordinovaný s ostatnými riešiteľmi projektu, pričom implementácia nadväzuje na novú architektúru stavového systému a sieťovej komunikácie popísanú v predchádzajúcich kapitolách. Stavový systém bol implementovaný na základe už vyššie uvedenej Petriho siete. Detailnejší opis implementácie tohto systému je možné nájsť v práci spoluriešiteľa projektu, kolegu Martina Čižmárika.

Architektúra implementovaných systémov používateľského ovládania, synchronizácie interakcií a externej komunikácie je znázornená na obrázku 6.1 vo forme diagramu tried. Diagram zobrazuje hlavné triedy a komponenty implementované v rámci tejto práce a ich vzájomné vzťahy medzi offline a online časťou aplikácie. Súčasťou diagramu sú systémy pre správu rigov, hand tracking, desktop interakciu, synchronizáciu avatarov a integráciu WebSocket komunikácie s EEG/BCI platformou OpenViBE. Kompletný diagram tried je možné nájsť v CD prílohe tejto práce, pričom jeho druhá časť je popísaná v práci spoluriešiteľa Martina Čižmárika.

Implementácia novej verzie systému vznikala modulárnym spôsobom, pričom jednotlivé subsystémy boli rozdelené podľa svojej funkcionality na správu rigov, interakciu používateľa, avatar systém a externú komunikáciu. Do novej verzie boli postupne prenesené a upravené funkcionality overené už v konceptuálnom prototypu, najmä systémy ovládania vo VR prostredí, hand tracking, desktop rozhranie a externá komunikácia pomocou protokolu WebSocket.

Výsledný prototyp novej verzie tak obsahuje funkcionality potrebné pre interakciu vo virtuálnom prostredí pomocou VR aj desktop rozhrania, synchronizáciu klientov prostredníctvom frameworku FishNet a komunikáciu s externými systémami pomocou protokolu WebSocket.



Obrázok 6.1: Diagram tried pre implementáciu prototypu novej verzie systému.

6.1 Komunikácia s OpenViBE serverom

Bola implementovaná WebSocket komunikačná vrstva, ktorej základná funkcionálnosť bola overená už v konceptuálnom prototypu. V aktuálnej implementácii je za túto funkcionálnosť zodpovedný objekt *NetworkManager*, ktorý obsahuje skript *FishNetWebSocketManager*.

Samotná vrstva zabezpečujúca spracovanie WebSocket protokolu a JSON správ, nachádzajúca sa v mennom priestore *NeuroRehab.Networking.API*, bola, ako už bolo spomenuté, prevzatá z existujúcej implementácie systému. V rámci tejto práce však bolo potrebné implementovať prepojovaciu logiku medzi WebSocket komunikáciou, frameworkom FishNet a novým stavovým systémom aplikácie založeným na Petriho sieti.

WebSocket komunikácia je synchronizovaná so stavom FishNet servera pomocou udalosti *ServerManager.OnServerConnectionState*. Systém zároveň implementuje autorizáciu klientov pomocou udalosti *client.checkin*, pričom každý klient je priradený ku konkrétnej virtuálnej miestnosti identifikovanej pomocou hodnoty *roomId*.

Pri prichádzajúcej komunikácii systém spracováva najmä udalosti typu *exercise.action.move* a *mi.stream.fatigue*. Udalosť *exercise.action.move* reprezentuje úspešné vykonanie motorickej predstavivosti pacientom a spôsobí vykonanie prechodu stavového systému do ďalšej fázy rehabilitačného scenára.

Prijaté správy týkajúce sa rehabilitačného cvičenia a pacienta sú broadcastované na server pomocou triedy *FishNetWebSocketManager*. Tieto správy následne zachytáva klient pacienta pomocou triedy *PatientBciListener*, ktorý ich dokáže spracovať a vyvolať udalosti v scéne.

Výpis 6.1: Reakcia herného automatu na úspešný BCI signál.

```

public void TriggerMoveForRoom(string roomId)
{
    var room = RoomChangeManager.Instance.rooms
        .FirstOrDefault(r => r.roomName == roomId);

    if (room != null)
    {
        var msg = new BciMoveTriggeredMessage();

        foreach (NetworkConnection conn in room.playerConnections)
        {
            conn.Broadcast(msg);
        }

        StateCoordinator coordinator =
            UnityEngine.Object.FindFirstObjectByType<StateCoordinator>();

        if (coordinator != null)
        {
            coordinator.Server_TriggerTransition(
                "Btp",
                "MiSuccess"
            );
        }
    }
}

```

Okrem prijímania EEG/BCI udalostí systém implementuje aj odosielanie správ z Unity klienta smerom ku platforme OpenViBE. Unity klient prostredníctvom WebSocket komunikácie odosiela udalosti reprezentujúce jednotlivé fázy rehabilitačného procesu, napríklad správy typu *exercise.eval.start* alebo *exercise.rest.start*, ktoré platforma OpenViBE využíva pri riadení zberu EEG dát.

Odosielanie týchto udalostí tiež zabezpečuje trieda *FishNetWebSocketManager*, ktorá zároveň umožňuje posun stavu v scéne. Funkcionalitu je možné otestovať pomocou tlačidiel v online scéne, pričom prijaté správy sú viditeľné vo vytvorenej testovacej stránke *WebSocketTest*.

6.2 Systém ovládania a zjednotenie rigov

Vo finálnom prototypu novej verzie bol implementovaný systém obsahujúci všetky rigy, ktoré boli prítomné už v konceptuálnom prototypu. Jednotlivé rigy boli zjednotené pod univerzálny prefab, ktorý obsahuje všetky potrebné komponenty pre VR aj desktop ovládanie. Na obrázku 6.2 je možné vidieť štruktúru implementovaných rigov v systéme.

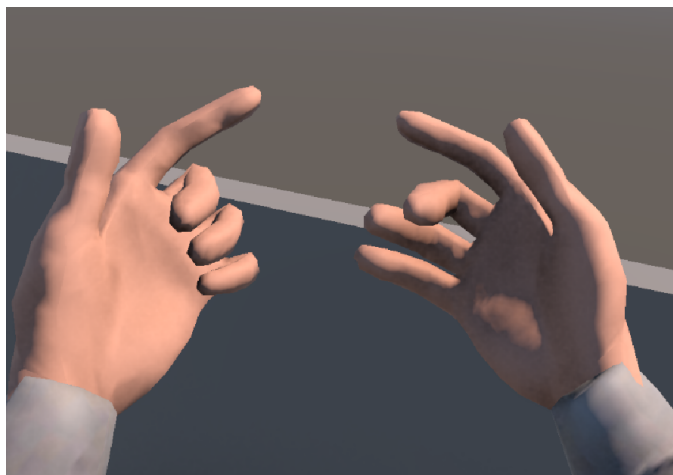


Obrázok 6.2: Štruktúra rigov v jednotnom prefabe

Po spustení aplikácie a automatickom výbere rigu pomocou skriptu *RigSwitcher* si používateľ v úvodnej Lobby scéne vyberá svoju rolu terapeuta alebo pacienta. Na základe tejto role server následne inicializuje príslušný prefab rigu pomocou triedy *BasicPlayerSpawner* metódou *RequestSpawn()* s upravenými funkcionalitami interakcie. Terapeut má prístup ku všetkým spôsobom ovládania vrátane desktop rozhrania, VR ovládania a snímania pohybu rúk. V prípade pacienta sa počas implementácie ukázalo, že možnosť ovládania pomocou hand trackingu neprináša výrazné zlepšenie používateľského zážitku a v niektorých prípadoch môže pôsobiť rušivo. Pacient preto môže naďalej využívať primárne klasické VR ovládanie pomocou ovládačov a funkcionality snímania rúk nemusí využívať.

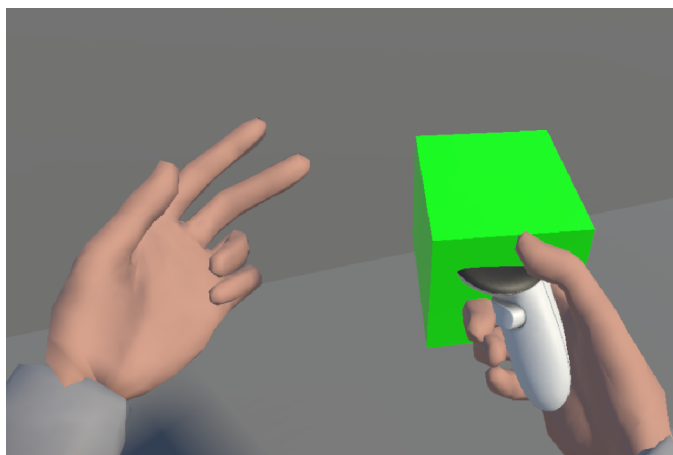
Systém snímania pohybu rúk bol oproti pôvodnej implementácii upravený tak, aby nedochádzalo k vizuálnemu odpojeniu rúk od modelu avatara pri rýchlych pohyboch. Na rozdiel od predchádzajúcej verzie systém využíva priamo ruky modelu avatara, čo zabezpečuje prirodzenejšiu vizualizáciu pohybu. Funkcionalita hand trackingu je implementovaná v oboch VR rigoch, teda v MetaXR aj OpenXR rigu, pričom používateľ môže medzi ovládaním pomocou rúk a ovládačov voľne prepínať.

Naďalej zostáva dôležitou súčasťou systému balík *FinalIK*, ktorý poskytuje komponenty *VRIK* a *FingerRig* zodpovedné za správnu animáciu a polohu rúk a prstov pomocou inverznej kinematiky. Tieto komponenty zabezpečujú synchronizáciu pohybov avatara so vstupmi zo snímania rúk a VR ovládačov.



Obrázok 6.3: Funkčné snímanie rúk v MetaXR po škálovaní avatara

Oba VR rigy zároveň implementujú hybridné ovládanie, ktoré umožňuje používateľovi ovládať aplikáciu jednou rukou a jedným ovládačom súčasne. V pôvodnej verzii systému bola táto funkcionálna dostupná len pre testovací MetaXR rig, no vďaka novším verziám OpenXR balíkov a úpravám štruktúry ovládania rúk bola implementovaná aj pre OpenXR rig.



Obrázok 6.4: Ukážka hybridného ovládania pomocou jedného ovládača a jednej ruky v MetaXR a OpenXR rigu

Konkrétne bol upravený skript *ControllerEventTrigger*, ktorý sa v oboch rigoch využíva na prepínanie cieľov inverznej kinematiky medzi hand tracking transformami a transformami VR ovládačov. Do OpenXR rigu bol zároveň pridaný komponent *FingerRig* samostatne pre každú ruku. V novej verzii skriptu bola zároveň optimalizovaná detekcia VR ovládačov pomocou cachovania zoznamu *InputDevice* zariadení, čím sa znížila potreba opakovaného vytvárania objektov pri každom frame aktualizáčnom cykle.

Výpis 6.2: Prepínanie cieľov pre inverznú kinematiku pri hybridnom ovládaní.

```
private List<InputDevice> inputDevices = new List<InputDevice>();

bool IsControllerCurrentlyTracked(XRNode node)
{
    InputDevices.GetDevicesAtXRNode(node, inputDevices);
    foreach (var device in inputDevices)
    {
        if (device.isValid &&
            device.characteristics.HasFlag(InputDeviceCharacteristics
                .Controller))
        {
            if (device.TryGetFeatureValue(CommonUsages.isTracked,
                out bool isTracked))
            {
                return isTracked;
            }
        }
    }
    return false;
}
```

Dôležitou súčasťou systému je aj persistentný prefab *GlobalEventSystem*, ktorý je inicializovaný už v offline Lobby scéne a následne prenášaný medzi scénami aplikácie. Obsahuje komponenty pre správu UI interakcie a používateľských vstupov pre VR aj desktop režim.

Súčasťou prefabu je skript *DynamicUISwitch*, ktorý dynamicky aktivuje alebo deaktivuje jednotlivé input a UI komponenty podľa aktuálne aktívneho rigu. Týmto spôsobom systém zabraňuje konfliktom medzi MetaXR, OpenXR a desktop interakčnými systémami pri súčasnej existencii viacerých rigov v scéne.

Prefab zároveň obsahuje komponent *OVRManager*, ktorý zabezpečuje inicializáciu MetaXR funkcionality aj pri použití OpenXR rigu na zariadeniach Meta Quest. Vďaka tomu je možné využívať funkcionality hybridného ovládania aj v OpenXR rigu.

6.3 Sieťová synchronizácia rigov a interakcie

Pre správnu synchronizáciu rigov medzi klientmi boli vytvorené sieťové verzie jednotlivých rigov určené pre terapeuta a pacienta. Tieto rigy obsahujú komponenty frameworku FishNet potrebné na sieťovú synchronizáciu pohybu, interakcií a vizualizácie jednotlivých častí avatara medzi klientom a serverom.

Dôležitou súčasťou online architektúry je trieda *OnlineRigManager*, ktorá zabezpečuje aktiváciu správneho rigu po pripojení klienta do online scény. Trieda prenáša informácie o zvolenom type rigu a nastaveniach avatara z offline časti aplikácie do online prefabov terapeuta alebo pacienta. Synchronizácia týchto údajov je realizovaná pomocou *SyncVar* premenných frameworku FishNet, ktoré zabezpečujú automatickú propagáciu zmien medzi serverom a klientmi.

Synchronizácia pohybu rúk, ovládačov a ich vizuálnych reprezentácií je realizovaná pomocou skriptov *NetworkControllerMeshSync* a *NetworkHandStateSwitch*. Skript *NetworkControllerMeshSync* zabezpečuje synchronizáciu viditeľnosti mesh objektov ovládačov medzi klientmi na základe aktuálneho spôsobu ovládania. V prípade prepnutia medzi hand trackingom a VR ovládačmi dochádza k automatickému prepínaniu vizualizácie jednotlivých modelov tak, aby ostatní klienti videli správny stav používateľovho ovládania.

Skript *NetworkHandStateSwitch* zabezpečuje synchronizáciu stavu virtuálnych rúk medzi klientom a serverom. Tento skript prepína medzi jednotlivými stavmi rúk na základe aktuálneho spôsobu interakcie používateľa a zároveň zabezpečuje správnu synchronizáciu týchto zmien medzi všetkými klientmi v miestnosti.

Dôležitou súčasťou synchronizácie interakcie s objektmi je aj skript *NetworkGrabbable*, ktorý umožňuje funkcionality XR grab interakcie a jej sieťovú synchronizáciu medzi klientmi a serverom. Skript zabezpečuje synchronizáciu pozície objektov, správu vlastníctva objektu a synchronizáciu fyzikálneho správania objektu počas jeho uchopenia alebo manipulácie používateľom.

Výpis 6.3: Synchronizácia vlastníctva objektu pri uchopení v online scéne.

```
private void ServerRequestOwnership(NetworkConnection newOwner)
{
    // SET OWNERSHIP
    base.NetworkObject.GiveOwnership(newOwner);

    // TURN OFF PHYSICS
    if (rb != null)
        rb.isKinematic = true;

    // SYNC
    ObserversSetKinematic();
}

[ObserversRpc(BufferLast = true)]
private void ObserversSetKinematic()
{
    if (!base.IsOwner && rb != null)
    {
        rb.isKinematic = true;
        rb.interpolation = RigidbodyInterpolation.None;
    }
}
```

V rámci desktop interakcie bola trieda *DesktopInteraction* rozšírená o sieťovú verziu *DesktopInteractionOnline*, ktorá pomocou frameworku FishNet zabezpečuje synchronizáciu interakcií používateľa s objektmi vo virtuálnom prostredí. Skript podporuje raycast interakcie, uchopenie objektov a synchronizáciu UI medzi klientmi.

Implementované systémy umožňujú konzistentnú synchronizáciu pohybu, interakcií a vizualizácie používateľov pri zachovaní podpory VR aj desktop ovládania.

6.4 Škálovanie avatara a rúk

O automatické škálovanie avatara a rúk sa v prototyp novej verzie škálovanie avatara a rúk zabezpečuje trieda *AvatarController*, ktorý je priradený priamo k prefabu avatara. V pôvodnej implementácii bola kalibrácia spúšťaná externe pomocou UI tlačidla alebo pomocného skriptu. V novej verzii systému prebieha kalib-

rácia automaticky pri štarte aplikácie.

Pri implementácii automatického škálovania bolo potrebné vykonať viaceré úpravy, pretože predchádzajúca verzia skriptu spôsobovala problémy pri použití MetaXR rigu. Počas prvých snímok po spustení aplikácie sa headset nachádza na pozícii (0,0,0), čo spôsobovalo nesprávny výpočet hodnoty škálovania blížiacej sa k nule. Úpravy boli vykonané v metódach *IsVRIKInitialized()* a *CalculateSizeMultiplier()*. Výpočet škálovania je pozastavený, kým výška headsetu nepresiahne hodnotu 0,5 metra. Následne je výsledná hodnota ohraničená na realistický rozsah, aby sa predišlo deformáciám modelu pri snímaní pohybu rúk.

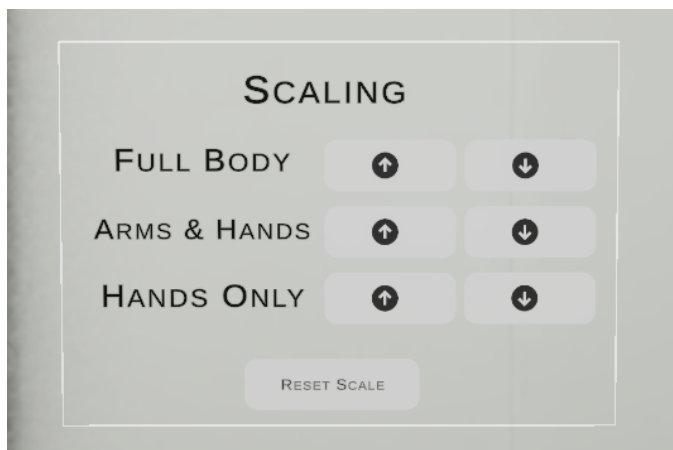
Výpis 6.4: Úprava metód AvatarController

```
bool IsVRIKInitialized()
{
    if (vrik == null || vrik.solver.spine.headTarget == null)
        return false;

    float headHeight = vrik.solver.spine.headTarget.position.y -
        vrik.references.root.position.y;
    return headHeight > 0.5f;
}

public float CalculateSizeMultiplier()
{
    float headsetHeight = (vrik.solver.spine.headTarget.position.y
        + 0.1f) - vrik.references.root.position.y;
    float rawMultiplier = Mathf.Clamp(headsetHeight / 1.725f, 0.75f
        , 1.15f);
    return Mathf.Round(rawMultiplier * 1000f) / 1000f;
}
```

Ďalej sa v scéne nachádzajú tlačidlá na manuálnu inkrementálnu kalibráciu škály avatara pre prípady, keď automaticky vypočítaná hodnota nie je úplne presná. Prostredníctvom triedy *AvatarMenuManager* a jej metód priradených k UI tlačidlám je možné upraviť škálu celého avatara alebo samostatne škálu jeho rúk. Trieda *AvatarMenuManager* najprv vyhledá aktuálne aktívny rig a následne aplikuje tieto hodnoty na triedu *AvatarController* aktívneho rigu, pričom ich zároveň uloží do objektu *AvatarSettings*.



Obrázok 6.5: Prototyp UI pre manuálnu kalibráciu škály avatara a rúk

6.5 HUD menu

Pre rolu terapeuta bol implementovaný základný prototyp HUD menu zobrazovaného zdvihnutím zápästia. Menu je implementované pomocou triedy *RaiseToWakeMenu*, ktorý je priradený k prefabom VR rigov.

Systém priebežne sleduje orientáciu a polohu ruky používateľa a pri vykonaní definovaného pohybu ruky automaticky zobrazí používateľské rozhranie vo virtuálnom priestore. Tento prístup umožňuje prirodzenejší spôsob interakcie bez potreby dodatočných tlačidiel alebo samostatných UI ovládačov.

Aktuálna verzia slúži predovšetkým ako koncept budúceho používateľského rozhrania pre správu terapie, výber scenárov a ovládanie jednotlivých častí systému. Funkcionalita menu zatiaľ nie je plne implementovaná, no prototyp overuje možnosti integrácie HUD rozhrania do VR prostredia a jeho ovládania pomocou hand trackingu a VR ovládačov.



Obrázok 6.6: Prototyp HUD menu pre terapeuta

7 Vyhodnotenie a záver

7.1 Vyhodnotenie implementácie

Cieľom práce bolo analyzovať existujúce riešenia systému L-NeRVEn, navrhnúť a implementovať funkcionality pre externú komunikáciu a používateľské ovládanie novej verzie systému a následne tieto riešenia overiť v prostredí virtuálnej reality. Na základe implementácie a vykonaného testovania možno konštatovať, že hlavné zadané ciele práce boli splnené.

V rámci tejto práce bola analyzovaná aktuálna implementácia systémov pre hand tracking, WebSocket komunikáciu a sieťovú synchronizáciu vo VR prostredí. Na základe tejto analýzy bol vytvorený konceptuálny prototyp novej verzie systému, ktorý následne slúžil ako základ pre implementáciu prototypu plnej verzie aplikácie. V tomto prototypu sa podarilo implementovať zjednotený systém ovládania pre MetaXR, OpenXR a desktop rozhranie. Funkcionalita hand trackingu bola rozšírená do oboch VR rigov a bola implementovaná podpora hybridného ovládania pomocou jednej ruky a jedného ovládača súčasne.

V oblasti sieťovej komunikácie bola úspešne vykonaná refaktorizácia pôvodného riešenia založeného na frameworku Mirror na framework FishNet. Nová implementácia zabezpečuje synchronizáciu rigov, interakcií a objektov medzi klientmi a serverom pri zachovaní kompatibility s VR aj desktop ovládaním. Zároveň bola implementovaná a overená funkcionality WebSocket komunikácie s EEG/BCI platformou OpenViBE pomocou protokolu WebSocket.

Implementované systémy boli priebežne testované na zariadení Meta Quest 3 a v desktop režime. Overená bola funkcionality hand trackingu, hybridného ovládania, sieťovej synchronizácie a komunikácie s externým systémom OpenViBE. Výsledný prototyp tak predstavuje základ novej verzie systému L-NeRVEn pripravený na ďalšie rozširovanie a optimalizáciu.

7.2 Nedostatky prototypu

Napriek úspešnej implementácii a overeniu hlavných funkcionalít systém stále obsahuje viaceré nedostatky, ktoré bude potrebné vyriešiť v budúcich verziách aplikácie.

Jedným z hlavných nedostatkov je naďalej limitovaný počet rehabilitačných scenárov a prostredí. Aktuálny prototyp sa sústreďoval prevažne len na overenie funkcionality systémov a tak obsahuje prevažne jednoduché testovacie prostredia, ktoré zatiaľ nevyužívajú plný potenciál virtuálnej reality v procese neurorehabilitácie.

V oblasti hand trackingu sa ukázalo, že použitý model avatara obsahuje limitovaný počet kostí a detailov v oblasti rúk, čo obmedzuje presnosť mapovania pohybu rúk pomocou inverznej kinematiky. Výsledkom je menej presná animácia prstov a pohybu rúk pri niektorých gestách. V budúcnosti by preto systém mohol benefitovať z využitia detailnejšieho modelu rúk alebo vlastného avatara optimalizovaného pre hand tracking.

Niektoré funkcionality OpenXR rigu sú menej stabilné v porovnaní s implementáciou MetaXR, najmä hybridné ovládanie pomocou jednej ruky a jedného ovládača. Hoci je táto funkcionalita implementovaná aj v OpenXR rigu, funguje exkluzívne na zariadeniach Meta Quest. Pri hybridnom ovládaní sa zároveň môžu v niektorých prípadoch vyskytnúť problémy s inicializáciou vstupných zariadení. V takom prípade je potrebné opätovne aktivovať vstupy pomocou systémového menu VR zariadenia a následne pokračovať v aplikácii.

Ďalším obmedzením je funkcionalita teleportácie používateľa. Spoľahlivo je momentálne implementovaná len v OpenXR rigu pri použití fyzických ovládačov. Teleportácia pomocou hand trackingu alebo v MetaXR rigu zatiaľ nie je plne funkčná vo finálnom headset builde aplikácie.

V aktuálnej implementácii sa tiež môže vyskytnúť problém pri kombinovanom použití desktop klienta a VR klienta v jednej online miestnosti. Ak sa ako prvý pripojí VR klient a následne desktop klient, môže v niektorých prípadoch dôjsť k narušeniu funkcionality VR interakcií. Počas testovania sa preto odporúča najprv pripojiť desktop klienta a až následne VR klientov. Tento problém súvisí s aktuálnou implementáciou inicializácie interakčných komponentov a predstavuje obmedzenie prototypu určené na odstránenie v budúcich verziách systému.

Limitácie sa prejavili aj pri automatickom škálovaní avatara, ktoré je menej presné najmä pri použití MetaXR rigu. V niektorých prípadoch je preto potrebná manuálna dodatočná kalibrácia pomocou UI tlačidiel.

Je možné spomenúť taktiež limitované testovanie systému. Implementácia bola overovaná najmä v kontrolovanom testovacom prostredí a na zariadení Meta Quest 3. Systém zatiaľ neprešiel rozsiahlejším testovaním s reálnymi pacientmi alebo v klinickej praxi.

7.3 Záver

Táto práca sa zameriavala na modernizáciu systému L-NeRVEn a návrh novej architektúry pre neurorehabilitáciu vo virtuálnej realite. Hlavným prínosom bolo úspešné prenesenie a rozšírenie interakčných a komunikačných systémov do prostredia Unity 6.

Vytvorený prototyp odstraňuje viaceré technologické obmedzenia pôvodného systému, zjednocuje metódy ovládania a zabezpečuje spoľahlivú synchronizáciu dát aj plynulú interakciu potrebnú pre terapeutické využitie.

Zároveň vytvára základ pre ďalší rozvoj systému, opravy nedostatkov prototypu a implementáciu nových terapeutických scenárov a budúce klinické testovanie v oblasti neurorehabilitácie vo virtuálnej realite.

Zoznam použitej literatúry

1. KÜHNE ESCOLÀ, Jordi; DEMIRDAS, Rumeysa; SCHULZE, Martin; CHAE, Woon Hyung; MILLES, Lennart Steffen; POMMERANZ, Doreen; OPPONG, Marvin Darkwah; KLEINSCHNITZ, Christoph; KÖHRMANN, Martin; FRANK, Benedikt. Virtual reality-guided therapy on a stroke unit: a feasibility study. *Neurological Research and Practice*. 2024, roč. 6, č. 1, s. 2–3. ISSN 2524-3489. Dostupné z doi: 10.1186/s42466-024-00357-w.
2. NEHILA, Peter. *Neurorehabilitácia v zdieľanej virtuálnej realite*. 2023. Dostupné tiež z: <https://opac.crzp.sk/?fn=detailBiblioForm&sid=880D977B60C8641DD3EE229C4FC3>.
3. RAUSCHNABEL, Philipp A.; FELIX, Reto; HINSCH, Chris; SHAHAB, Hamza; ALT, Florian. What is XR? Towards a Framework for Augmented and Virtual Reality. *Computers in Human Behavior*. 2022, roč. 133, s. 107289. ISSN 0747-5632. Dostupné z doi: <https://doi.org/10.1016/j.chb.2022.107289>.
4. MIKAEL, Adam. *Terapeutický systém na báze technológií eXtended (XR) reality*. 2025. Dostupné tiež z: <https://opac.crzp.sk/?fn=detailBiblioForm&sid=AF7DAECCF8461446C79043FD86FE>.
5. KOREČKO, Štefan; MATHES, Jaroslav; SOBOTA, Branislav; CHMURA, Damián; ROSIPAL, Roman; VANKÓ, Martin. *Open Computer Science*. 2025, roč. 16, č. 1, s. 20250049. Dostupné z doi: 10.1515/comp-2025-0049.
6. ŠARIĆ, Matko; RUSSO, Mladen; KRALJEVIĆ, Luka; METER, Davor. Extended Reality Telemedicine Collaboration System Using Patient Avatar Based on 3D Body Pose Estimation. *Sensors*. 2023, roč. 24, č. 1, s. 27. ISSN 1424-8220. Dostupné z doi: 10.3390/s24010027.
7. CALTABIANO, Anna; BURKE, Taylor; NESI, Jacqueline; DI SIMPLICIO, Martina; ZALK, Nejra van. Virtual reality delivered exposure for fear of needles: a small-scale pilot. *Frontiers in Psychiatry*. 2025, roč. Volume 16 - 2025. ISSN 1664-0640. Dostupné z doi: 10.3389/fpsy.2025.1642988.

8. UNREALENGINE. *Unreal Engine Documentation*. 2025. Dostupné tiež z: <https://dev.epicgames.com/documentation/en-us/unreal-engine/unreal-engine-5-6-documentation>.
9. UNITYTECHNOLOGIES. *Unity Documentation*. 2025. Dostupné tiež z: <https://docs.unity3d.com/Manual/index.html>.
10. UNITYTECHNOLOGIES. *XR Hands Documentation*. 2025. Dostupné tiež z: <https://docs.unity3d.com/Packages/com.unity.xr.hands@1.5/manual/index.html>.
11. UNITYTECHNOLOGIES. *XR Interaction Toolkit Documentation*. 2025. Dostupné tiež z: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.4/manual/index.html>.
12. ROOTMOTION. *Final IK Documentation*. 2025. Dostupné tiež z: <https://assetstore.unity.com/packages/tools/animation/final-ik-14290>.
13. META. *Meta XR SDK Documentation*. 2025. Dostupné tiež z: <https://developers.meta.com/horizon/downloads/package/meta-xr-sdk-all-in-one-upm/>.
14. READYPLAYERME. *Ready Player Me Documentation*. 2025. Dostupné tiež z: <https://docs.readyplayer.me/>.
15. MIRRORNETWORKING. *Mirror Documentation*. 2025. Dostupné tiež z: <https://mirror-networking.gitbook.io/docs>.
16. FISHNETNETWORKING. *FishNet Documentation*. 2025. Dostupné tiež z: <https://fish-networking.gitbook.io/docs>.
17. STATIANZO. *Fleck – C# WebSocket Implementation* [<https://github.com/statianzo/Fleck>]. 2024.
18. CHMURA, Damián. *Reprezentácia používateľa vo virtuálno-reálnom systéme pre neuror rehabilitáciu*. 2025. Dostupné tiež z: <https://opac.crzp.sk/?fn=detailBiblioForm&sid=A6A9937D4AB687F472CB8FA9E4AF>.
19. CHMURA, Damián. *Komunikácia, architektúra a ďalšie aspekty systému pre neuror rehabilitáciu vo virtuálnej realite*. 2026. Dostupné tiež z: <https://opac.crzp.sk/?fn=detailBiblioForm&sid=70D560FC111767F5366D79EF9293>.

Zoznam príloh

Príloha A Používateľská príručka

Príloha B Systémová príručka

Príloha C CD médium – záverečná práca v elektronickej podobe, zdrojový kód, systémová dokumentácia a ďalšie prílohy

A Používateľská príručka

1 Úvod

Táto príručka poskytuje základné informácie o inštalácii, spustení a používaní vytvoreného prototypu novej verzie systému L-NeRVEn určeného pre neurorehabilitáciu vo virtuálnej realite.

Systém umožňuje interakciu vo virtuálnom prostredí pomocou VR headsetov, snímania pohybu rúk (hand tracking) a desktop ovládania. Súčasťou systému je taktiež sieťová synchronizácia medzi klientmi a externá komunikácia s EEG/BCI systémom OpenViBE pomocou protokolu WebSocket.

Príručka popisuje základný postup inštalácie systému, jeho spustenie a spôsob používania jednotlivých funkcionálít prototypu.

2 Inštalácia

Keďže ide o základný prototyp verzie systému, aplikácia nie je voľne dostupná na žiadnej bežne používanej platforme na stiahnutie. Pre účely testovania a používania systému je potrebné kontaktovať autora práce a dohodnúť sa na spôsobe získania prístupových údajov k prototypu.

Stolný počítač

Pre spustenie desktop verzie systému je potrebný počítač s operačným systémom Windows 10 alebo novším. Systém je distribuovaný vo forme samostatného Unity buildu. Po získaní build súborov stačí rozbaľiť projekt do ľubovoľného priečinka a spustiť súbor l-nerven-neo.exe.

Meta Quest 3

VR verzia systému bola testovaná primárne na zariadení Meta Quest 3. Pre inštaláciu je potrebné mať aktivovaný vývojársky režim a povolenú možnosť inštalácie aplikácií z externých zdrojov. Inštalácia aplikácie prebieha pomocou nástroja Android Debug Bridge (ADB) nahraním APK súboru do zariadenia. Po úspešnej inštalácii je aplikácia dostupná v sekcii aplikácií v prostredí headsetu Meta Quest.

Pre správnu funkcionálnosť hand trackingu je odporúčané povoliť snímanie pohybu rúk v nastaveniach headsetu.

3 Spustenie

Pred spustením systému je potrebné zabezpečiť, aby boli všetky zariadenia pripojené do rovnakej lokálnej siete. V prípade využitia a testovania EEG/BCI komunikácie je taktiež potrebné mať spustenú platformu OpenViBE alebo využiť vytvorenú testovaciu stránku `WebSocketTest.html` v súboroch prototypu.

Spustenie servera

Pred spustením buildu je potrebné v priečinku *l-nerven-neo_Data* nakonfigurovať text súboru *server_config.json* aby sa IP adresa a port zhodovali so skutočnými hodnotami, na ktorých bude server bežať. Pre localhost komunikáciu sa používa IPv4 adresa 127.0.0.1. . Pre komunikáciu medzi zariadeniami v rovnakej sieti sa používa LAN IPv4 adresa, napríklad 192.168.0.250. Takáto adresa umožňuje pripojenie medzi viacerými zariadeniami pripojenými na rovnakú WiFi sieť alebo, napríklad medzi PC a VR headsetom. Adresa 0.0.0.0 sa používa na serverovej strane. Znamená, že server sa automaticky bindne na všetky dostupné sieťové adaptéry zariadenia. Vďaka tomu nie je potrebné nastavovať konkrétnu IP adresu pri každej zmene siete. Táto hodnota sa používa iba pre host/server aplikáciu, klienti sa na ňu nepripájajú priamo. Serverová časť systému je spúšťaná pomocou priloženého .exe súboru určeného pre serverový klient. Po jeho spustení sa otvorí terminálové okno, v ktorom sa zobrazujú informácie o stave servera, inicializácii sieťovej komunikácie a pripojených klientoch.

Po úspešnom spustení server inicializuje framework FishNet a následne aj WebSocket komunikačnú vrstvu používanú pre komunikáciu s EEG/BCI systémom OpenViBE. V terminálovom výpise je možné sledovať informácie o úspešnom vytvorení servera, pripojení klientov a spracovaní jednotlivých udalostí systému. Server musí byť spustený pred pripojením klientov do online miestnosti.

Spustenie klienta

Meta Quest

Pri použití VR headsetu Meta Quest je najprv potrebné nahráť klientsku aplikáciu do zariadenia. Po spustení aplikácie systém automaticky inicializuje MetaXR rig alebo OpenXR rig závisle od toho, aký build systému aktuálne máme, aktivuje sa funkcionality VR ovládania vrátane podpory hand trackingu a hybridného ovládania. Používateľ sa následne môže pripojiť k serveru a vstúpiť do virtuálneho prostredia ako terapeut alebo pacient.

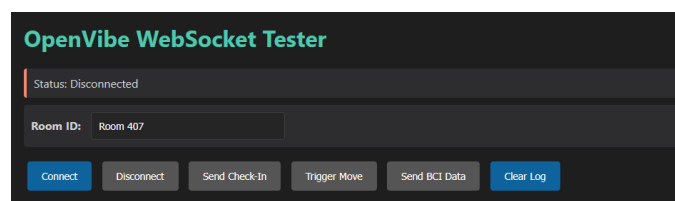
Režim stolného počítača

Na stolnom počítači je možné klienta spustiť pomocou priloženého .exe súboru. Pred spustením buildu je potrebné v priečinku *l-nerven-neo_Data* nakonfigurovať súbor *server_config.json* tak, aby IP adresa a port zodpovedali skutočným hodnotám, na ktorých bude server spustený. Ak pri štarte aplikácie nie je detegované VR zariadenie, systém automaticky inicializuje Desktop rig s podporou ovládania pomocou klávesnice a myši.

Po úspešnom spustení aplikácie je používateľovi umožnené pripojiť sa k serveru a následne ovládať virtuálne prostredie pomocou desktop rozhrania.

Externá komunikácia

Pre otestovanie externej komunikácie je potrebné mať spustenú platformu OpenViBE alebo testovaciu stránku WebSocketTest.html. Zároveň je potrebné zabezpečiť správne nastavenie WebSocket komunikácie medzi Unity aplikáciou a OpenViBE klientom.



Obrázok 1: Ukážka tlačidiel WebSocket testeru.

Po spustení servera je pri testovaní potrebné najprv stlačiť tlačidlo *Connect*, čím sa inicializuje spojenie medzi WebSocket a serverom. Následne je po pripojení do miestnosti potrebné zadať názov miestnosti do textového poľa *Room ID* a stlačiť tlačidlo *Send Check-in*. Týmto dôjde k autorizácii klienta a jeho priradeniu ku konkrétnej miestnosti pomocou identifikátora *roomId*.

V oblasti log výpisov je následne možné sledovať prijaté EEG/BCI správy, potvrdenia odoslaných správ a stav WebSocket komunikácie. Pomocou tlačidla *Trigger Move* je možné otestovať zmenu stavu stavového systému rehabilitačného procesu. Funkcionalitu je možné pozorovať najmä v prípade, keď sa aktuálny stav systému nachádza v stave *Setup*.

V prípade, že sa testovacia stránka *WebSocketTest* nedokáže pripojiť k serveru, je vhodné overiť, či port 7778 nie je obsadený iným procesom. Na systéme Windows je možné túto kontrolu vykonať príkazom:

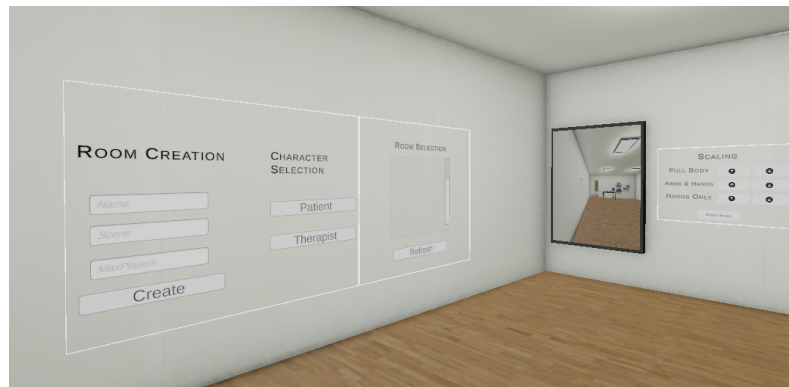
```
netstat -ano | findstr 7778
```

Ak je port obsadený, proces je možné ukončiť príkazom:

```
taskkill /F /PID <PID>
```

4 Použitie systému

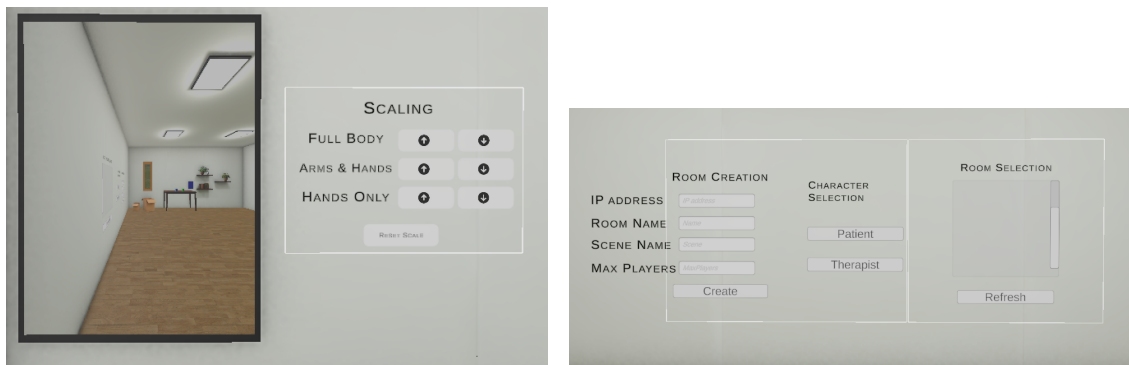
Po spustení aplikácie sa používateľ nachádza v úvodnej offline scéne. V tejto časti systému má možnosť upraviť škálu avatara, vytvoriť novú miestnosť alebo sa pripojiť do jednej z existujúcich miestností dostupných na serveri.



Obrázok 2: Prostredie offline scény.

Pri vytváraní miestnosti je možné nastaviť ID miestnosti, vybrať scénu, v ktorej bude prebiehať rehabilitácia, a určiť maximálny počet pripojených používateľov. V aktuálnej implementácii je dostupná iba testovacia scéna *Training_Test*. Pred pripojením alebo vytvorením miestnosti si používateľ vyberá rolu, s akou sa pri-

pojí do online miestnosti. Dostupné sú roly terapeuta a pacienta, pričom predvoľená hodnota je nastavená na terapeuta. Na základe zvolenej role systém inicializuje príslušný rig prefab a funkcionality ovládania.



Obrázok 3: Nastavenia škály avatara a vytvárania miestnosti.

V aktuálnej implementácii je IP adresu servera možné zmeniť vo vstupnom textovom poli *IP Address* ako aj vo vyššie popísanom konfiguračnom súbore pre server build a build stolného počítača. Po zadaní IP adresy servera je pre zobrazenie dostupných miestností potrebné stlačiť tlačidlo *Refresh*, ktoré načíta aktuálny zoznam miestností zo servera. Ak chce používateľ vytvoriť novú miestnosť, je potrebné po zadaní IP adresy stlačiť tlačidlo *Create*. Po pripojení do miestnosti je používateľ presunutý do rehabilitačnej scény. V prostredí je možné vykonávať základné interakcie s objektmi pomocou VR ovládania, hand trackingu alebo desktop rozhrania. Interakcie s objektmi sú synchronizované prostredníctvom servera medzi všetkými pripojenými klientmi.



Obrázok 4: Synchronizácia interakcií medzi klientmi v online scéne.

V režime stolného počítača je pohyb používateľa realizovaný pomocou kláves *W, A, S, D* a orientácia kamery pomocou pohybu myši. Interakcia s UI prvkami prebieha prostredníctvom kurzora myši a ľavým tlačidlom myši.

Vo VR režime je možné systém ovládať pomocou VR ovládačov alebo pomocou technológie snímania rúk. Pohyb je možný pomocou joysticku na ľavom ovládači. Interakciu s objektmi je možné otestovať v offline aj online scéne. V režime stolného počítača prebieha interakcia s objektmi pomocou podržania ľavého tlačidla myši. Pri použití OpenXR rigu je interakcia s objektmi realizovaná spodným trigger tlačidlom VR ovládača Meta Quest. Pri použití MetaXR rigu sa na interakciu využíva horný trigger ovládača. Pomocou týchto vstupov je možné vykonávať uchopenie objektov alebo interakciu s UI prvkami. OpenXR aj MetaXR rig podporujú ovládanie pomocou VR ovládačov, hand trackingu a hybridného ovládania umožňujúceho súčasné použitie jednej ruky a jedného ovládača. Funkcionalita hybridného ovládania je však v OpenXR rigu spoľahlivo podporovaná prevažne na zariadeniach od spoločnosti Meta.

Používateľ s rolou terapeuta má navyše možnosť meniť stavy stavového systému pomocou pripravených UI tlačidiel za predpokladu, že server vyhodnotí požadovaný stavový prechod ako validný. Terapeut má taktiež možnosť upravovať polohu objektov v scéne a ovplyvňovať priebeh rehabilitačného procesu. Na rozdiel od pacienta má navyše prístup k desktop rozhraniu.



Obrázok 5: Prostredie online scény dočasné UI pre stavy.

5 Známe obmedzenia

Aktuálna verzia systému predstavuje základný prototyp novej architektúry systému L-NeRVEn, a preto obsahuje viaceré obmedzenia a nekompletné funkcionality.

V rámci aktuálnej implementácie je dostupná iba základná testovacia scéna *Training_Test*. Systém zatiaľ neobsahuje pôvodne plánované terapeutické scenáre, s implementáciou ktorých sa počíta do budúca.

Systém bol primárne vyvíjaný a testovaný na zariadeniach Meta Quest 3. Hoci aplikácia obsahuje aj podporu OpenXR rigu, niektoré funkcionality, najmä hyb-

ridné ovládanie pomocou jednej ruky a jedného ovládača, fungujú spoľahlivo prevažne na zariadeniach Meta Quest.

Použitý model avatara obsahuje obmedzený počet kostí a detailov v oblasti rúk, čo môže pri hand trackingu spôsobovať menej presné mapovanie pohybov prstov a rúk pomocou inverznej kinematiky.

HUD menu terapeuta momentálne predstavuje len prototyp používateľského rozhrania a neobsahuje plnú funkcionálnosť.

Automatické škálovanie avatara nemusí byť v niektorých prípadoch úplne presné, a preto môže byť potrebná manuálna kalibrácia pomocou UI tlačidiel.

B Systémová príručka

Kompletnú systémovú príručku popisujúcu proces vytvárania spustiteľných súborov pre tento prototyp je možné nájsť aj v práci spoluriešiteľa projektu Martina Čižmárika.

1 Dokumentácia kódu

Zdrojový kód systémových súborov, ktoré boli v tejto práci spomenuté a zakreslené v architektúre systému, obsahuje dokumentačné komentáre vo formáte C#, z ktorých bola pomocou nástroja Doxygen vygenerovaná technická dokumentácia implementovaných tried a metód. Táto dokumentácia sa nachádza v CD prílohe práce.

Táto dokumentácia obsahuje aj popisy súborov a komponentov prevzatých z implementácie predchádzajúcich verzií systému, napríklad priečinkov *API* implementujúci WebSocket klient alebo vlastný logger nachádzajúci sa v priečinku *Logger*. Keďže tieto komponenty neboli hlavným cieľom tejto práce, dokumentácia nemusí obsahovať detailný popis všetkých prevzatých súborov a tried.