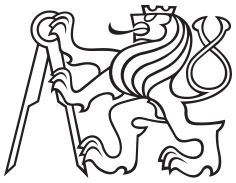
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Využití hlasového ovládání ve VR

Vít Gardoň

Vedoucí: Ing. David Sedláček, Ph.D
Studijní program: Otevřená informatika
Specializace: Počítačová grafika a hry
Květen 2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Gardoň** Jméno: **Vít** Osobní číslo: **499091**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**

Studijní program: **Otevřená informatika**

Specializace: **Počítačové hry a grafika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití hlasového ovládání ve VR

Název bakalářské práce anglicky:

Voice control in VR

Pokyny pro vypracování:

Seznamte se se způsoby a možnostmi rozpoznávání přirozené řeči využitelným pro virtuální realitu (VR). Navrhněte komponenty pro herní engine Unity, které umožní realizaci nabídky ovládané tradičními VR technikami (např. ukazování, přímá interakce) a současně hlasem (např. tradiční 2D nabídky nebo 3D menu). Navrhněte a implementujte testovou VR scénu, na které demonstrujete použitelnost implementovaných komponent. Navrhněte testové scénáře, kdy bude práce uživateli nějak ztížena, např. časová tíseň, nepřehlednost, omezený stupeň interakce rukama. Metodami uživatelského testování porovnejte tradiční techniky a ovládání hlasem (jednomodální i vícemodální kombinaci technik). Zhodnoťte preferované způsoby ovládání. Pro realizaci hlasového ovládání a VR aplikace se omezte na VR platformu Meta Quest, pro rozpoznávání řeči použijte knihovnu Mama-AI (https://themama.ai/).

Seznam doporučené literatury:

1] Jason Jerald, The VR Book: Human-Centered Design for Virtual Reality. 2015. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA.
2] Joseph J. LaViola, Jr. et all. 3D User Interfaces: Theory and Practice, second edition. 2017. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
3] Dan Jurafsky and James H. Martin, Speech and Language Processing, 3rd ed. draft. 2023. dostupné online: https://web.stanford.edu/~jurafsky/slp3/
4] A. Vaswani et al.. Attention Is All You Need. 2017. https://arxiv.org/abs/1706.03762

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. David Sedláček, Ph.D. katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **17.02.2023** Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. David Sedláček, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____
Datum převzetí zadání

_____
Podpis studenta

Poděkování

Děkuji panu Ing. Davidu Sedláčkovi, Ph.D., za odborné vedení práce, jeho připomínky, rady a vstřícnost při konzultacích bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně včetně všech podepsaných zdrojových kódů a také že jsem uvedl veškerou použitou literaturu v souladu Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

.....
Vít Gardoň

V Praze, 22. května 2023

Abstrakt

Bakalářská práce se zabývá hlasovým ovládání menu ve virtuální realitě. V první části této práce se věnuji způsobu zpracování zvuku a jeho transformaci na využitelná data. Podstatou řešení hlasového ovládání je převod zvuku na text a jeho pochopení programem. Ve druhé a třetí části své práce se zaměřuji na analýzu řešení a návrh komponent pro hlasové ovládaní v herním enginu Unity. Čtvrtá část obsahuje návrh a implementační detaily demonstrační aplikace pro ovládání hlasem s využitím mnou navržených komponent. V páté části se práce věnuje testování a zpracování výsledku testování. Na základě asistovaného monitoringu jsou zjištěna data, která shrnují a hodnotí preferované způsoby ovládání.

Klíčová slova:virtuální realita, VR, ovládání hlasem, hlasově ovládané menu, Unity

Vedoucí: Ing. David Sedláček, Ph.D

Abstract

The bachelor thesis deals with voice control of menus in virtual reality. In the first part of this thesis, I discuss how sound is processed and transformed into usable data. The essence of the solution is the conversion of sound to text and its understanding by the program.

In the second and third part of my thesis, I focus on the solution analysis and design of components for voice control in the Unity game engine. The fourth part contains the design and implementation details of a demonstration application for voice control using my proposed components. In the fifth part, the thesis focuses on testing and processing the result of testing. Based on assisted monitoring, gathered data are used to summarize and evaluate the preferred control methods.

Keywords:virtual reality, VR, voice control, menu, Unity

Title translation: Voice control in VR

Obsah

Obrázky

Tabulky

Kapitola 1

Úvod

Sluch a zrak jsou dva základní lidské smysly, které nám umožňují vnímání okolního světa. Pomocí nich zažíváme příběhy, které se dějí kolem nás. Virtuální realita je technologie, která nám umožňuje vytvořit iluzi skutečného světa, kterou naplno vstřebáváme právě zrakem a sluchem. Ve virtuálním světě se potřebujeme pohybovat a vykonávat určité akce. Jednou ze základních akcí je ovládání uživatelského rozhraní, které se v nějaké formě vyskytuje skoro v každé aplikaci. Toto ovládání je nejčastěji řešeno formou výběru z nabídky.

Lidská komunikace je založená na hlasu. Lidé tak spolu komunikují již od nepaměti. Hlas je jedním z důležitých prostředků k předávání informací a pokynů. Bez hlasových povelů se neobejde žádná lidská činnost. Ztráta hlasu vede k omezení komunikace. Počítačová komunikace se omezila ve svých počátcích zejména na programovací jazyky nebo na využívání psaných předem definovaných příkazů.

Nové technologie přináší větší komfort uživateli. Nabízejí zážitky v zrakové, sluchové i hmatové podobě, ale pro jejich ovládání se používají ovladače. Speciálně ve virtuální realitě se nabízí jako nový rozměr právě ovládání pomocí hlasu. Je tomu teprve několik desítek let, co počítače dokáží přijímat zvukové signály. To je první krok k hlasové komunikaci s počítačem. Ve virtuální realitě by hlasové ovládání přineslo větší, intenzivnější a pohlcující zážitek pro uživatele.

Cílem této práce je prostudovat možnosti hlasové komunikace ve virtuální realitě a navrhnout komponentu pro Unity s využitím analýzy hlasu pomocí knihovny firmy Mama-AI.

1.1 Analýza hlasové komunikace s počítačem

Hlasovou komunikaci s počítačem si můžeme rozdělit do několika po sobě jdoucích kroků. Tyto kroky jsou důležité pro zpracování informace do formy, která je snazší a srozumitelnější pro předávání dat počítači.

1. **Příjem zvukových signálů** – V prvním kroku máme mluvené slovo, které je získáno za pomoci mikrofonu v analogové formě. Následně je převedeno do digitální podoby. Digitální záznam je zpracován a uložen v počítači pro další využití.
2. **Převod řeči na text** – V druhém kroku je zaznamenaný digitální zvukový záznam analyzován a převáděn na výsledný text, který může být dále počítačem využíván k zjištění záměru uživatele.
3. **Zjištění smyslu vstupního textu** – V tomto kroku dochází k rozdělení textu na věty, u kterých musí proběhnout vyextrahování původní předávané informace. Tato část se dá považovat za nejkomplikovanější část hlasové komunikace s počítačem. K tomu, aby došlo k co nejpřesnějšímu pochopení informace, se primárně používá strojové učení. Strojové učení umožňuje počítači zvýšit schopnost zpracování komplexního zvukového signálu. K tomuto učení je zapotřebí velké množství dat.
4. **Reakce na uživatele** – V závěrečném kroku po zjištění smyslu vstupního textu, když program zjistí po analýze textu, co se v něm píše a co po něm uživatel chce, dojde k provedení příkazu nebo ke změně datového modelu. Tento krok závisí na doméně, o kterou se počítač stará. Může vykonat, to oč byl požádán (např. ukončit aplikaci) nebo může odpovědět na otázku, která mu byla položena. Jednou z reakcí na uživatele může být syntetizované mluvené slovo. Počítač musí sestavit text odpovědi, kterému bude uživatel rozumět a následně na základě natrénovaného modelu převést na mluvené slovo. Jedná se do značné míry o inverzní operaci ke kroku 3 a 2. Reakce na uživatele je závislá na kontextu a schopnostech používaného programu.

1.1.1 Převod řeči na text

Převod řeči na text STT (speech to text) je základní krok v komunikaci s počítačem. Při něm se analyzuje zaznamenaný digitální signál. Výstupem převodu řeči na text je přepis zaznamenaného mluveného slova, které je obsažené ve zvukové nahrávce. Vlastní rozpoznávání textu je prováděno pomocí

neuronových sítí. Tyto neuronové sítě se inspirovaly nervovou soustavou člověka. Jejím základem jsou neurony, které vytvářejí jednotlivé vrstvy. Neurony a tím pádem vrstvy jsou navzájem propojené a dochází mezi nimi k přenosu signálů [1].

Do převodu řeči na text je zapojeno velké množství neuronů. Proto nejsme schopni přesně určit, jak neuronová síť pracuje. Neuronová síť je označována za černou skříňku (Black box), což je funkce, u které přesně nevíme, jak vnitřně funguje nebo nás to nezajímá. Posuzujeme pouze přesnost výsledku. Vstupní data jsou převáděna na výstupní v závislosti na hodnotě vah a prahů. Hodnoty vah a prahů jsou závislé na tréninku neuronové sítě. Při trénování neuronové sítě se nastavují váhy a prahy jednotlivých neuronů tak, aby výsledky byly co nejpřesnější. Jak dobře si neuronová síť vede, se hodnotí pomocí funkce, jenž vrací chybovost na testovacích datech. Změnou vah a prahů se snažíme minimalizovat chybovost. Používá se třeba metoda postupného sestupu [2], kdy procházíme N – rozměrný prostor (N = počet vah a prahů). Snažíme se v něm najít minimum funkce určující chybovost. A to tak, že v každém opakování nalezne směr největšího spádu a učiníme krok tímto směrem. Poté tento proces znovu opakujeme až do dosažení lokálního minima. Pro trénování neuronové sítě je potřeba velké množství trénovacích dat. Rozpoznávací modely neuronových sítí pro převod řeči na text se dají rozdělit do dvou skupin. Jsou buď závislé nebo nezávislé [3].

- **Nezávislé** modely jsou obecné a jejich snahou je, aby fungovaly pro širokou škálu lidí s různými přízvuky a intonacemi.
- **Závislé** modely jsou naopak přizpůsobeny na míru specifickému uživateli. Díky takto omezené doméně je mnohem snazší model natrénovat. Je pak však mnohem méně přesný, když ho používá jiný uživatel. Schopnosti modelu také závisí na velikosti jeho slovní zásoby. Čím více slov, či dokonce čím více jazyků, má model obsahovat, tím větší jsou potřebná trénovací data [3].

Ještě před vložením signálu do modelu pro detekci slov se vyplatí jej rozdělit na několik částí. V signálu se najdou odmlky, které většinou symbolizují buď konec slova nebo slabiky. Je totiž snazší, když je rozpoznávaný signál kratší a obsahuje jen pár slabik, než když je v něm schovaná celá věta. Po rozdělení je signál analyzován pomocí Fourierovy transformace [4], která dokáže signál rozdělit na jednotlivé frekvence, které se v něm vyskytují. Tak vzniká spektrogram, který se teprve používá jako vstup pro model. Samotná konečná analýza může použít například neuronové sítě, vektorové kvantování nebo skrytý Markovův model. Dnes ty nejlepší modely využívají hluboké neuronové sítě. Příkladem takového modelu v akci je Google Speech-to-Text, ke kterému můžeme získat přístup na stránkách cloud.google.com [5].

Postup převodu řeči na text

1. Před vložením zvukového signálu do modelu pro detekci slov je vhodné jej rozdělit na několik částí. Ve zvukovém signálu se najdou odmlky, které většinou symbolizují buď konec slova nebo slabiky. Je vhodnější a také snazší, když rozpoznávaný signál je kratší a obsahuje jen pár slabik, než když je v něm schovaná celá věta nebo souvětí.

2. Po rozdělení je signál analyzován pomocí Fourierovy transformace, která dokáže zvukový signál rozdělit na jednotlivé frekvence, které se v něm vyskytují. Tak vznikne spektrogram, který se teprve použije jako vstup pro model.

3. K samotné konečné analýze jednotlivých frekvencí spektrogramu můžeme použít například neuronové sítě, vektorové kvantování nebo skrytý Markovův model. Dnes ty nejlepší modely využívají hluboké neuronové sítě, které dosahují nejlepších výsledků. Jedním z příkladů takového modelu v akci je Google Speech-to-Text. Přístup k němu můžeme získat na stránkách https://cloud.google.com/speech-to-text.

4. Výsledkem je text, který vznikl převodem zvukové nahrávky.

1.1.2 Pochopení textu programem - Natural language understanding

Pochopení textu programem je základním kamenem pro hlasové ovládání. Když již nějaký způsobem získáme výstup od uživatele v textové podobě, tak musíme zajistit, že program zareaguje tak, jak si uživatel představoval. Podstatou pochopení textu programem je zjistit uživatelův záměr. Ten je skrytý ve vstupním textu, ale program mu nerozumí.

K zjištění záměru se věnuje intent classification. V této úloze dokážeme analyzovat vstupní text a pomocí strojového učení určit záměr a entity, které text obsahuje. Text je rozdělen na jednotlivé části – tokeny, které slouží jako vstup pro rozpoznávací část algoritmu. Tomuto procesu se říká tokenizace [6].

Protože je velmi obtížné identifikovat všechny možné záměry, soustředí se model na předem určenou množinu záměrů. Pro správné přiřazení k záměrům musí být k dispozici dostatečné množství vět sloužících jako příklady – vzory. Čím více těchto vzorových vět je, tím přesnější bude model.

Tento proces je primární činností knihovny Mama-AI. Podle mnou vybraných záměrů a entit jsem nechal natrénovat model pro rozpoznávání záměrů [7].

1.2 Ukázky existujících řešení

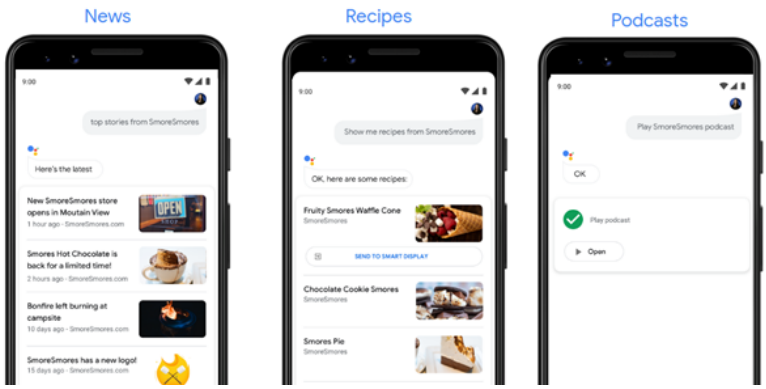
Hlasem ovládané menu ve VR má třeba v níže uvedené Voice commands na platformě Meta Quest. Další příklady hlasového ovládání můžeme nalézt u virtuálních hlasových asistentů jako jsou Google Assistant, Amazon Alexa nebo Siri od firmy Apple. Jde o nejznámější a nejrozšířenější řešení hlasového ovládání.

- **Voice commands - Meta Quest**

  Společnost Meta na svých headsetech Meta Quest, Meta Quest 2 a Meta Pro implementoval možnost ovládání headsetu pomocí hlasových příkazů. Jedná se o jednoduché příkazy, které umožňují například vypnout nebo zapnout wifi, spustit hru nebo zapnout passthrough mód. V nabídce hlasového ovládání na platformě Meta Quest není dostupná funkce pro ovládání jednotlivých nabídkových menu hlasem [8].

- **Google assistant, Alexa, Siri**

  Toto jsou tři nejznámější a nejrozšířenější konverzační hlasoví asistenti. Všichni fungují na principu popsaném výše [1.1], jen jejich doména se liší. Každý z nich dokáže odpovídat na obdobný, ale trochu se lišící soubor otázek a žádostí. Jsou schopni také vyhledávat informace na internetu. Dokážou tedy například zjistit, kdy se odehrála bitva u Slavkova a odpovědět vám na tuto otázku. Pokud však už nejsou schopni dále postupovat, odkážou na stránku vyhledávání, kde si odpověď může najít každý sám. Aby bylo dosaženo co nejlepšího efektu, který bude připomínat reálného člověka, je u nich možnost se zeptat na některé odlehčené otázky. Například: „Jak se máš?“ nebo je požádat, aby vám řekli vtip [5] [9].



Obrázek 1.1: Ukázka uživatelského rozhraní Google assistenta
Zdroj: [10]

Kapitola 2

Analýza řešení

Při řešení problematiky hlasové komunikace ve virtuální realitě jsem využíval stávající knihovny dostupné pro herní engine Unity. V této práci jsem využíval tři dostupné knihovny. První byla knihovna firmy Mama-AI a další dvě byly knihovny XR Interaction Toolkit a Oculus XR plugin.

2.1 Knihovna firmy Mama-AI

Knihovna, která mi byla poskytnuta firmou Mama-AI, je hlavní součástí tohoto projektu. Knihovna kombinuje cloudovou aplikaci Text-to-speech od Googlu (https://cloud.google.com/text-to-speech) s NLU modelem od firmy Mama-AI.

Knihovna Mama-AI

Knihovna je tvořena třemi Unity komponenty. Jedná se o následující komponenty:

- **AudioManager** - Tato komponenta je využívána k nastavení vlastností vstupu a délky nahrávek odesílaných na rozpoznání textu.
- **AudioSaveManager** - Tato komponenta umožňuje ukládání zvukových nahrávek z mikrofonu na disk.

- **Nexus** - Tato komponenta umožňuje ukládání zvukových nahrávek z mikrofonu na disk a je hlavní částí knihovny. Probíhá v ní komunikace se servery Google, kam se posílají zvukové nahrávky pro převod mluveného slova na text. Následně přijímá získané texty a ty odesílá na servery Mama-AI, kde se provádí rozpoznávání záměru. Získaný výsledný text a záměr je předáván dále pomocí události, ke které se může jakýkoliv script připojit. Na tento script musí být odkázáno v AudioManageru. Script se používá také pro vyžádání výstupů kdekoliv to je aplikací vyžadováno.

NLU model Mama-AI

NLU model je založen na platformě Rasa [11]. Rasa je open source framework, který slouží jako základ pro množství virtuálních asistentů. Pro správné fungování knihovny potřebujeme odpovídající přístupový klíč v souboru stt.json a také odkazy společně s autentifikačními klíči pro syntézu textu na řeč v config.txt a k modelu na rozpoznávání záměru. Mama-AI mi poskytla možnost využít syntézu řeči, která je přístupná pomocí komponenty Nexus. V případě správného nastavení se může začít používat samotná knihovna v jakékoliv Unity komponentě. Pomocí komponenty Nexus můžeme zapínat a vypínat nahrávání funkcemi Activate() a Deactivate(). Můžeme například zapnout rozeznávání, pouze když držíme nějaké tlačítko. Toto použití je preferované, jelikož se tím snižuje nárok na využívání serverů na cloudu a snižuje se tím i finanční zátěž aplikace. Používání cloudových serverů je totiž zpoplatněno.

Nová verze knihovny

Na začátku května 2023 mi byla poskytnuta nová verze knihovny. Její hlavní výhodou je zbavení se závislosti na Google speach to text. To výrazně snižuje velikost celé knihovny a také urychluje vytváření spustitelných souborů. Speech to text je však stále zapotřebí a byl pouze nahrazen STT přímo od firmy Mama-AI, který provozují přímo oni. Tato verze však nebyla použita v tomto projektu, jelikož mi byla poskytnuta příliš pozdě.

2.2 Knihovna XR Interaction Toolkit

Jedná se o toolkit, který je vyvíjený přímo Unity. Do projektu se dá připojit pomocí Package manageru. Slouží jako abstrakce nad všemi možnými VR

platformami, pro které by se museli aplikace vytvářet samostatně, jelikož každá vyžaduje svůj vlastní plugin. XR Interaction Toolkit zahrnuje pluginy Oculus, Open XR, Windows Mixed Reality a MagicLeap. Můžeme tedy jeho pomocí vyvíjet aplikace pro všechny zmíněné platformy najednou. XR Interaction Toolkit zároveň obsahuje množství základních funkcí, které se ve VR využívají a které by se musely vždy znovu programovat, protože jsou používány skoro v každé aplikaci pro virtuální realitu. Jako příklad funkce můžeme uvést systém na chytání objektů nebo na interakci s objekty či menu na dálku pomocí paprsků [12].

2.3 Oculus XR plugin

Balíček Ocelus XR plugin je vytvořený společností Meta. Slouží k interakci s headsety jako jsou Meta Quest, Meta Quest 2 nebo Oculus Rift S. Balíček implementuje podobné funkce jako výše zmiňovaný XR interaction toolkit, ale je specificky zaměřený pouze na headsety společnosti Meta. Umožňuje však sledování rukou uživatele, což je něco, co XR interaction toolkit nenabízí. Na začátku roku 2023 byl uveřejněn balíček XR Hands, který umožňuje sledování rukou při použití OpenXR, který sjednocuje ovládání do jednoho balíčku, aby vznikl jednotný standard mezi výrobci headsetů pro virtuální realitu. Jeho používání zjednodušuje práci vývojářů, kteří mohou vyvíjet pouze jednu variantu aplikace využitelnou na všech platformách. Tento balíček však nebyl při implementaci mojí práce ještě zveřejněn, a proto jsem jej nemohl použít. Usnadnil by mi práci, jelikož bych mohl používat pouze OpenXR [13].

Kapitola 3

Návrh řešení

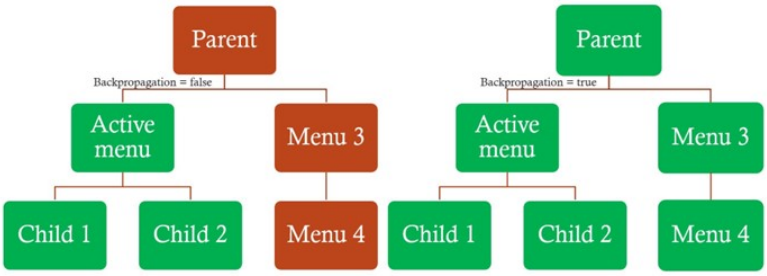
3.1 Návrh komponenty pro Unity

Mým cílem bylo vytvoření komponenty, která by měla sloužit k usnadnění vytváření hlasem ovládaného menu ve virtuální realitě. Jelikož každé menu se zabývá něčím jiným (například změna nastavení, výběr herního režimu aj.) a má tedy jinou doménu, je třeba, aby se komponenty byly schopné přizpůsobovat různým NLU modelům. Rozhodl jsem se řešit hlasem ovládané menu ve VR pomocí navržení 3 komponent. Mnou navržené komponenty mají zajišťovat a starat se o celý proces, který souvisí s fungováním hlasového menu ve VR. Uživatel těchto mých komponent tedy nebude muset vůbec pracovat přímo s knihovnou Mama-AI.

- **VoiceAction**
  Funkce této komponenty spočívá v substituci klasického tlačítka v tradičním menu. VoiceAction umožňuje nastavit název záměru, na který reaguje a umožňuje určení událostí, které probíhají po výběru VoiceAction VoiceManagerem. Pro správné fungování VoiceAction se musí specifikovaný záměr vyskytovat v příslušném využívaném NLU modelu. VoiceAction umožňuje nastavení seznamu entit, na které reaguje. Entity jsou ScriptableObject, které umožňují přiřadit jedné entitě synonyma. Jelikož jsou v Unity scriptableObject považovány za assety, můžeme jednu entitu používat ve více komponentách. To nám usnadňuje nastavování parametrů komponenty nebo případnou hromadnou změnu na všech místech, kde byla entita využita.

Jako příklad můžeme uvést následující situaci. Při vytvoření nového VoiceAction se záměrem CHANGECOLOR, přiřadíme jako entitu asset Green, který obsahuje slova "green"a "greener". Když dojde k vyhodnocování shody VoiceAction s výstupem z NLU modelu VoiceManagerem, který bude mít od NLU výstup CHANGECOLOR a entitu "greener", tak dojde ke shodě a vyvolají se události, které jsou k VoiceAction definované. Je důležité říct, že když žádná entita není přiřazena, tak postačí pouhá shoda záměru. Naopak, když je jich přiřazeno více, tak mezi nimi funguje OR operátor. Přirozený záměr může být dvojího typu, který reaguje nepřetržitě, anebo reaguje pouze když VoiceMenu, do kterého patří VoiceAction, je vybráno VoiceManagerem.

- **VoiceMenu**
  Funkcí této komponenty je sloužit jako jedna z obrazovek menu (nastavení je považováno za jednu obrazovku a výběr postavy je považováno za druhou obrazovku). Jednotlivé VoiceMenu vytváří strom, po kterém se dokáže VoiceManager přesouvat a prohledávat ho. To umožňuje, aby nemusely být všechny VoiceAction neustále dostupné v právě vybraném menu. Umožnuje to identifikaci a výběr VoiceAction, který je nejblíže k aktuálnímu vybranému VoiceMenu a odpovídá záměru. VoiceAction se k VoiceMenu připojí tak, že VoiceMenu bude přiřazeno do veřejné proměnné VoiceAction s názvem parentMenu. Ve VoiceMenu máme možnost povolit backpropagation. Když VoiceManager vyhledává ve stromu příslušné VoiceAction, které jsou shodné s výstupem NLU modelu, tak prohledává celý podstrom aktuálně zvoleného VoiceMenu. Povolením backpropagation však umožníme i hledání v menu, které je předkem vybraného menu. Na obrázku [3.1] jsou zeleně označená menu, která budou prohledávána. Je zde vidět, jak se zvětší prohledávaný prostor, když se povolí backpropagation. Jako příklad můžeme uvést, pokud nechceme, abychom z „Menu-nastavení“ mohli zapnout novou hru v hlavním menu, tak nastavíme „Menu-nastavení“ jako potomka hlavního menu a zakážeme backpropagation. To způsobí, že se budeme muset nejdříve vrátit zpět na hlavním menu a až potom si budeme moci spustit novou hru.



Obrázek 3.1: Diagram porovnávající prohledaná menu s a bez backpropagation

- **VoiceManager**
  Jde o hlavní komponentu, jejíž funkcí je starost o výběr jednotlivých

akcí, které zajišťuje. Tato komponenta jako jediná komunikuje přímo s komponentou Nexus z knihovny Mama-AI. Drží si instanci VoiceMenu, na které se ve stromě právě nacházíme. Od tohoto místa postupně prochází strom menu a vyhledává VoiceAction, který odpovídá zájmu, jenž dostal od NLU modelu. Prohledávání stromu probíhá prohledáváním do šířky [14].

To zajistí, že se jako první vybere shoda s VoiceAction, která je aktuálnímu VoiceMenu nejblíže. Když se totiž nacházíme v menu nastavení a NLU model vrátí záměr BACK, tak chceme, aby se použil VoiceAction, s tímto záměrem, který se nachází přímo v tomto menu a ne třeba VoiceAction, které je v menu nadřazeném menu nastavení. VoiceManager pro své správné fungování potřebuje referenci na Nexus a AudioManager.

Kapitola 4

Demonstrační aplikace

4.1 Návrh aplikace

Schopnosti mnou navržených komponent pro hlasové ovládání jsem demonstroval na aplikaci virtuálního konfigurátoru automobilů. Řešení pomocí mnou navržených komponent v konfigurátoru umožňuje uživateli výběr mezi třemi různými modely vozů různých značek (Volvo, Ferrari, Aston Martin) a dále mu dává možnost úpravy základních vzhledových vlastností a vybavení interiéru automobilů.

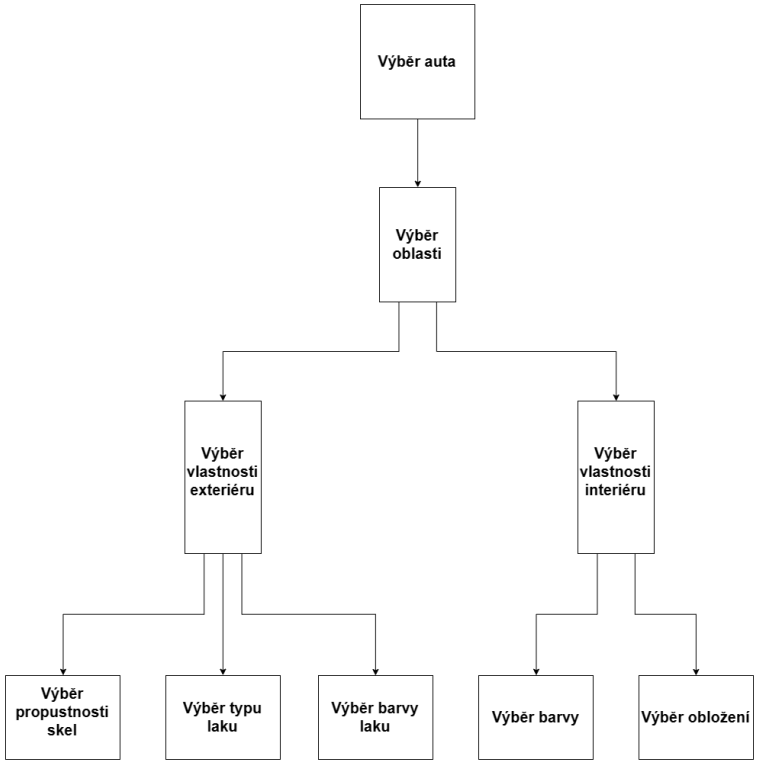
Po zapnutí aplikace se uživatel dostává do virtuálního studia, kde se může volně pohybovat pomocí ovladačů Oculus Touch nebo pomocí volného pohybu v herním prostoru. Uživatel je schopen si ovladačem vyvolat menu, které mu poskytne nabídku výběru jednotlivých možností. Menu je pak možné ovládat nejen hlasem, ale také pomocí druhého ovladače. Při použití přímé interakce (ruce) bude tabulka s menu stále připevněna na uživatelovy levé ruce a nebude ji možné skrýt. Pravou rukou je uživatel schopný si vybírat možnosti v menu jako by se jednalo o reálný tablet [4.4].

Pro vývoj aplikace jsem použil Unity verze 2021.1.23f. Vývoj jsem začal z ukázkové aplikace, která mi byla poskytnuta firmou Mama-AI. Ta již obsahovala jejich plugin pro rozpoznávání mluveného slova a záměru, a také Oculus XR plugin. Projekt jsem doplnil o XR Interaction Toolkit pomocí Unity Package Manageru. Tento toolkit mi usnadnil vývoj základních funkcí VR aplikace [15].

4.1.1 Struktura menu

Vlastní menu pro konfigurátor jsem vytvořil z několika menších samostatných menu 4.1. Všechny menu dohromady vytváří strom, který je možné procházet. Ovládání menu je umožněno pomocí ukazování ovladači, pomocí přímé interakce a také současně hlasem. Hlasové ovládání umožní přechod mezi jednotlivými menu nebo výběr jednotlivých položek.

Jako příklad mohu uvést situaci, kdy je uživatel na obrazovce s typem laku a požádá hlasem o přesunutí do menu pro změnu barvy laku. Nebude nucen se nejdříve vrátit zpět na menu exteriéru a až pak následně na menu výběru barvy laku. Předpokládám, že tato forma přechodu mezi menu je uživatelsky mnohem příjemnější. Proto chci tento předpoklad ověřit během uživatelského testování.



Obrázek 4.1: Diagram přechodů mezi jednotlivými menu.

4.1.2 Detailní popis struktury menu

Po zapnutí aplikace se uživatel dostává do studia, kde může probíhat výběr konfigurace aut. Při vstupu do studia se uživatelům zobrazí nabídka ve formě menu. V každém menu se nachází tlačítko „Back“, jenom v menu výběru modelu auta je toto tlačítko nahrazené tlačítkem „Quit“, které vypne aplikaci. V menu výběr modelu auta se také nachází tlačítko „Switch scene“, které přesune uživatele na scénu s jiným typem ovládání. Pokud je ve scéně s přímým ovládáním, tak ho přesune do scény s ukazováním a naopak.

1. **Výběr modelu auta** – Proběhne z menu. V této nabídce budou tři modely aut:
   - Aston Martine V12 Vantage
   - Ferrari Enzo
   - Volvo S90
2. **Výběr oblasti** - Zde dojde k volbě mezi exteriérem a interiérem.
3. **Výběr vlastnosti exteriéru** – Nabídka obsahuje možnosti volby:
   - barvy auta
   - typu laku
   - propustnosti skel
4. **Výběr barvy laku** – Nabízí se pět barev pro každé auto. Jde o tyto barvy:
   - černá
   - bílá
   - červená
   - zelená
   - růžová
5. **Výběr typu laku** – Pro výběr laku jsou dvě možnosti:
   - metalíza
   - matný lak
6. **Výběr propustnosti skel** – Nabídka má k volbě tři hodnoty:
   - dvaceti procentní propustnost světla
   - padesáti procentní propustnost světla

- osmdesáti procentní propustnost světla

Propustnost je možné ovládat také pomocí tlačítek plus a mínus, která posunují již vybranou hodnotu o jeden stupeň výše či níže.

7. **Výběr vlastnosti interiéru** – Tato volba umožňuje přechod mezi barvou interiéru a materiálem obložení.
8. **Výběr obložení** – Nabídka má k volbě dvě varianty:
   - plast
   - dřevo
9. **Výběr barvy interiéru** – V tomto výběru můžeme volit ze dvou variant:
   - černá
   - bílá

4.2 Implementace aplikace

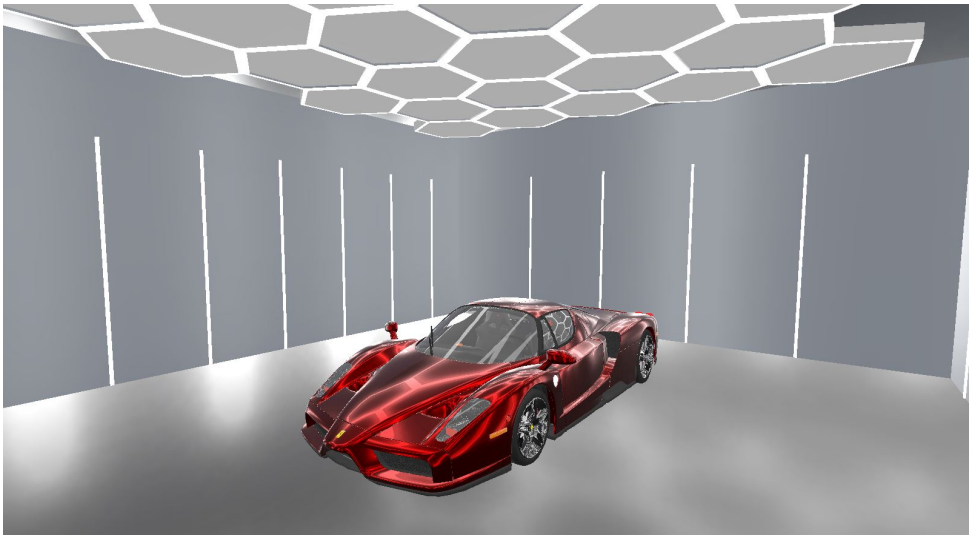
Implementace demonstrační aplikace jsem zahájil vytvořením prostředí, výběrem modelů aut a navržením ovládáním. Na základě navrženého ovládání jsem vytvořil jednotlivé menu aplikace.

Návrh prostředí

Tématem demonstrační aplikace je konfigurátor auta. Proto jsem vytvořil jednoduchý model místnosti, která má simulovat dojem fotografického studia 4.2. Model studia jsem vytvořil v programu Blenderu 3.3.0 [16].

Modely aut

Modely aut byly staženy z internetu pod vhodnou licencí. Následně jsem je optimalizoval v modelovacím softwaru Blender [16], aby je byl Meta Quest 2 schopný vykreslovat v přijatelné snímkové frekvenci. Jelikož modely nebyly vytvářeny mnou, tak ne všechny materiály byly rozloženy tak, jak bych si ideálně přestavoval. Některé části modelů mají materiály zapečené do textury, a tudíž je nejde lehce měnit. Tento problém se týká hlavně interiéru modelu Volva. Proto uživatel nemůže na tomto vozidle provádět změny tak markantní, jako je tomu u ostatních vozů.

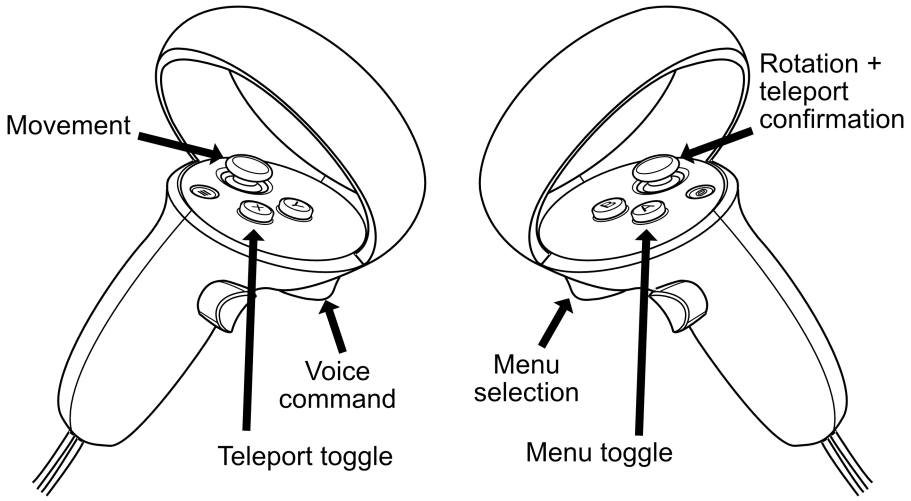


Obrázek 4.2: Ukázka jednoho z modelů aut ve studiu.

4.2.1 Ovládání

Základní ovládání aplikace je přizpůsobené dvěma ovladačům bez možnosti využití sledování rukou, které Meta Quest podporuje. [4.3].

- **Levý joystick** - Pohyb joysticku určuje směr a rychlost pohybu vzhledem k natočení hlavy uživatele.
- **Pravý joystick** - Pohyby joysticku doleva a doprava otáčí uživatele na místě o fixní úhel. Pokud však je zároveň drženo primární tlačítko (X) na levém ovladači, tak se vyšle křivka sloužící k označení cílového místa teleportace. Křivka se objeví, pouze pokud je s pravým joystickem pohnuto ze základní pozice. Směr joysticku určuje finální natočení uživatele po teleportaci. Teleportace je potvrzena stisknutím pravého joysticku.
- **Primární tlačítko pravého ovladače (A)** - Stisknutím tlačítka se vyvolá nebo skryje aktuální menu. Slouží to především k orientaci v jednotlivých vrstvách menu. Zobrazují se zde také všechny možnosti výběru a tlačítko zpět viz. Detailní popis struktury menu.
- **Levý trigger** - Během jeho stisknutí je zapnut hlasový vstup, který se okamžitě vyhodnocuje.



Obrázek 4.3: Schéma přiřazení funkcí k ovladačům.
Zdroj: Převzato z [17] a upraveno.

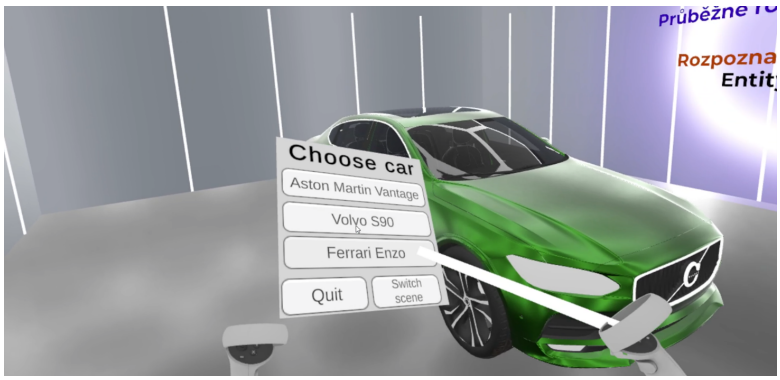
Přímá interakce (ruce)

Při ovládání rukou je menu s možnostmi připevněné k levé ruce a ukazováček na pravé ruce slouží k výběru jednotlivých možností v menu. Ovládání je podobné jako u tabletu či mobilního telefonu. Stačí zmáčknout tlačítko a provede se požadovaná změna. Jelikož se však nejedná o opravdový tablet, tak je možné ním prostrčit prst. To způsobovalo problémy, když se při zvedání prstu z tlačítka stisklo další nové tlačítku. Proto má uživatel po stisknutí tlačítka necelou sekundu na zvednutí prstu z tabletu. Uživatel nemá tlačítka držet, ale pouze do nich ťuknout. Tento časový úsek jsem zvolil z důvodu, že jsem nechtěl, aby byl zpomalován nutností čekat před dalším pokynem. To by mu mělo umožnit vyrovnat se v rychlosti výběru pomocí ukazování [18].

4.2.2 Vzhled menu

Pro každou volbu v aplikaci jsem vytvořil samostatné menu. Kvůli ovládání ukazováním jsem jednotlivé prvky menu udělal poměrně velké 4.5, aby byly snadno stisknutelné a příjemné pro testované uživatele. Protože jsou lidé zvyklí na používání tlačítek na dotykových obrazovkách, nemusel jsem brát žádné ohledy u ovládání přímou interakcí.

Obrázek 4.4: Ukázka menu s ovládáním přímou interakcí v menu výběru barvy interiéru



Obrázek 4.5: Ukázka menu s ovládáním ukazování v menu výběru auta

Kapitola 5

Testování

V rámci své práce jsem provedl testování aplikace hlasového ovládání ve VR. Účelem testování bylo porovnání tradičních technik ovládání VR (ruce, ovladače) s hlasovým ovládáním, případně popsat nedostatky hlasového ovládání. Uživatelské testování je analytická metoda reflektující interakci reálných uživatelů s aplikací. Pracujeme s reálnými osobami a srovnáváme naše předpoklady se skutečnými daty, které jsme získali prostřednictvím testu.

Vlastní testování jsem zaměřil na použitelnost demonstrační aplikace. Ověřoval jsem, zda testovaní uživatelé aplikace jsou schopni plnit zadané úlohy, které simulují její předpokládané použití. Zadání úloh se testuje v simulovaném prostředí, tak abychom se co nejvíce přiblížili skutečnému používání aplikace. Při testování aplikace jsem sledoval, jak úspěšní byli testovaní uživatelé při plnění zadaných úkolů a s čím měli problémy. Jednou z otázek v závěrečném dotazníku bylo ověření, zda mají uživatelé zájem o ovládání aplikace pomocí hlasového rozhraní. Nešlo ale o hlavní zájem testování.

K testování použitelnosti aplikace můžeme vybrat různé metody a přístupy. Nejdříve se musíme rozhodnout, jestli testování bude probíhat prezenčně nebo online. Protože testování online nebylo vhodné z důvodu nedostupnosti VR headsetů, rozhodl jsem se pro prezenční formu testování. Tato forma mi umožnila sledovat chování testovaných v reálném čase a zároveň mi umožnila simulaci ve ztížených podmínkách (časová tíseň).

5.1 Průběh testování

Testování probíhalo ve dvou fázích. První fáze byla nemoderovaná. Druhá fáze probíhala moderovanou formou.

1. Nemoderované testování bylo zahájeno seznámením testovaných uživatelů s aplikací. Poskytl jsem jim pouze informaci, že mohou používat ovládání pomocí přímé interakce (ruce), ukazováním (ovladače) nebo hlasovým ovládáním. Testování uživatelé nedostali žádné informace nebo návod, jak aplikaci ovládat. Bylo to proto, aby bylo možné vyhodnotit, který způsob ovládání pro uživatele bude intuitivní. Jedinou poskytnutou nápovědou bylo schéma ovládání ovladačů, které mohli nalézt ve scéně, ale nebyli o něm informování. Testovaným uživatelům bylo poskytnuto 3-5 minut na odzkoušení aplikace podle reakce uživatele.
2. Moderované testování probíhalo následovně:
   a. Uživatelům byly podrobně a názorně vysvětleny všechny způsoby ovládání aplikace – přímá interakce (ruce), ukazování (ovladače) a hlasové ovládání.
   b. Uživatelé byli seznámeni s průběhem testování, při kterém jim byly zadávány úkoly ke splnění. Šlo například o příkazy: - Přebarvi auto na matnou červenou barvu. - Vyber jako zobrazovaný model automobil Volvo. - Změň materiál interiéru na dřevěný apod.
   c. V další fázi testování došlo k omezení výběru ovládacích metod. Byly zvoleny různé kombinace dvou forem: - Hlasové ovládání a přímá interakce (zákaz používání ovladačů) - Hlasové ovládání a ukazování (zákaz používání rukou) - Přímá interakce a ukazování (zákaz používání hlasového ovládání)
   d. Následně byl výběr ovládání omezen jen na jednu formu (hlasové ovládání, přímá interakce, ukazování).
   e. V závěrečné fázi došlo k testování v časové tísni. Všichni testovaní uživatelé měli v časovém limitu 1 minuta splnit co nejvíce stejných standardizovaných úkolů podobného charakteru jako v bodě b).
   f. Postupně byli testováni uživatelé v následujících kombinacích ovládání:
      - Nejdříve jim byly dostupné všechny tři formy ovládání (přímá interakce, ukazování a hlasové ovládání).
      - Následně museli používat kombinaci dvou forem ovládání /hlasové ovládání a přímou interakci (zákaz používání ovladačů), hlasové ovládání a ukazování (zákaz používání rukou), přímá interakce a ukazování (zákaz používání hlasového ovládání).

- Na závěr jim byla dostupná jen jedna forma (hlasové ovládání, ruce, ovladače).

g. Zjištěné výsledky testování byly zaznamenávány do tabulky [5.1], která byla podkladem pro vyhodnocení preferovaného způsobu ovládání.

5.2 Vyhodnocení testování

Výsledky testování z tabulky [5.1] byly použity k formulování závěru testování použitelnosti hlasové aplikace ve VR

Při nemoderovaném testování bylo testujícím uživatelům umožněno nejdříve si projít aplikaci sám za využití jakéhokoliv ovládání. Nebyly dány žádné pokyny jen se pozorovalo k čemu se účastníci uchylují. Sledoval jsem reakce testovaných uživatelů a zajímalo mě, co začínali používat jako první volbu. Většinou využívali přímou interakci (ruce) a vyzkoušeli si hlasové ovládání. Lze předpokládat, že použití přímé interakce a hlasového ovládání souviselo se zájmem testovaných uživatelů o novou formu ovládání. Většina z nich v minulosti již používala k ovládání VR ukazování (ovladače).

Při moderovaném testování jsem poučeným testovaným uživatelům zadával jednotlivé příkazy, které se měli účastníci studie snažit splnit. Testovaní uživatelé nebyli žádným způsobem instruováni či motivováni k používání určitého typu ovládání. Sledoval jsem výběr použitého ovládání. Testovaní uživatelé se snažili využívat hlasové ovládání, ale efektivita byla nízká, někteří jej přestali používat.

Na základě údajů z testování jsem zjistil, že velký problém byl v převádění hlasu na text (speech to text), které využívalo Google STT API. Testovaní uživatelé měli problém s rozeznáním slov. Slova, která říkali, nebyla Google STT rozpoznána. Například: „Last item“ se převáděl na „lost item“ nebo z „car menu“ se stával „carneval“. Uživatelé museli mluvit hodně pomalu a snažit se výrazně artikulovat. Tento problém byl pravděpodobně způsoben nekvalitním mikrofonem umístěným na headsetu. Nepředpokládám, že by byl problém na straně Google STT, jelikož Google Assistent v telefonech s operačním systémem Android používá stejné STT a na telefonu k těmto vadám nedochází tak často jako při mém testování.

5.2.1 Porovnání vícemodálního ovládání

- **Ovládání ukazováním, hlasem a přímou interakcí**

  V prvním kole testování se testovaní uživatelé z nabízených tří variant ovládání nejčastěji uchylovali k používání přímé interakce, ale často zkoušeli i ovládání hlasem.

- **Ovládání ukazováním a hlasem**

  Při testování ukazováním a ovládáním hlasem, které probíhalo jako druhé kolo testování, se testovaní uživatelé primárně spoléhali na ukazování a ovládání hlasem použili pouze zřídka. Každý se však pokusil nejméně o jednu interakci pomocí hlasového ovládání.

- **Ovládání přímou interakcí a hlasem**

  Ve třetím kole testování při variantě přímé interakce a ovládání hlasem se většina testovaných uživatelů odklonila od použití ovládání hlasem a využívali přímou interakci. Pouze pět testovaných uživatelů použilo ovládání hlasem.

- **Ovládání ukazováním a přímou interakcí**

  Čtvrté kolo se nesoustředilo na ovládání hlasem. Testovaní uživatelé v něm využívali ovládání ukazováním a ovládání přímou interakcí v poměru 4 : 6. Tím je dán poměr mezi počty testovaných uživatelů, kteří primárně využívali dané ovládání. Zbylí dva uživatelé často přepínali mezi typy ovládání a nemohu vyhodnotit, které ovládání využívali více.

5.2.2 Porovnání jednomodálního ovládání

Při používání pouze jednoho druhu ovládání jsem se soustředil na porovnání kladů a záporů jednotlivých forem ovládání.

- **Ovládání ukazováním**

  Testovaní uživatelé se na ovládání ukazováním pomocí ovladačů adaptovali nejrychleji, protože již s touto formou ovládání měli předchozí zkušenost. Používání ovládání ukazováním i z těchto důvodů vykazovalo malou chybovost. Tato forma byla pro testované uživatele nejpohodlnější a hodnotili ji pozitivně, jako volbu číslo jedna.

- **Ovládání přímou interakcí**

Testovaní uživatelé ovládání přímou interakcí hodnotili kladně, ale nedostatkem zmiňovaným uživateli byla skutečnost, že při stisknutí tlačítka nedostali žádnou jinou zpětnou vazbu než vizuální. Mnoho uživatelů navrhovalo, aby při zmáčknutí tlačítka dostali haptickou odezvu. Jejich představa byla, že půjde třeba o vibraci. Tato interakce ale není u platformy Oculus Quest možná, protože k sledování rukou se využívají pouze čtyři kamery umístěné na headsetu. Takováto interakce by byla možná při použití doplňků ve formě VR rukavic (například [19]).

- **Ovládání hlasem**

  Hlasové ovládání testovaní uživatelé nehodnotili pozitivně vzhledem k problémům viz [5.1.2]. Tato forma se jim jevila zajímavou z důvodu možnosti ovládat i prvky, které se nenacházejí v aktuálním menu.

5.2.3 Porovnání typů ovládání

Po vyzkoušení všech typů ovládání VR byly testovaným uživatelům položeny otázky, které měly odpovědět na to, který způsob ovládání jim připadal nejvíce příjemný a proč. Dotazoval jsem se na to, jaký typ ovládání by upřednostnili k využívání, kdyby byly odstraněny jeho nedostatky. Osm uživatelů preferovalo ovládání ukazováním (ovladačem), čtyři preferovali přímou interakci (ruce). Žádný z testovaných uživatelů nepreferoval hlasové ovládání. Tento výsledek jsem předpokládal, jelikož ovládání hlasem není moc přesné. Kdyby byly odstraněny veškeré jeho nedostatky, tak se většina uživatelů shodla, že hlasové ovládání není moc vhodné pro procházení menu, ale určitě by ho využívali pro možnost ovládání prvků, které se nenacházejí v aktuálním menu. Například by hlasem změnili model auta, když jsou právě v menu pro výběr barvy laku, jelikož by se jim nechtělo vracet přes několik vrstev menu až na začátek a pak zpět.

5.2.4 Ovládání v časové tísni

Při testování v časové tísni se uživatel pokoušel vykonat co nejvíce příkazů za jednu minutu. Pokud měl na výběr z více forem interakce, tak si sám mohl vybrat, jak bude menu ovládat. Každý uživatel vyzkoušel všech 7 kombinací ovládání. Vždy se začínalo se všemi třemi formami ovládání a poté se postupovalo podle sloupečků v tabulce [5.1].

Jak je vidět v tabulce výsledků, tak nejhůře ovladatelná kombinace forem je ovládání pouze hlasem, se kterým uživatelé dokázali dosáhnout pouze poloviny

Číslo účastníka	Ovladač + ruce + hlas	Ovladač + hlas	Ruce + hlas	Ovladač + ruce	Ovladač	Ruce	Hlas	Průměr účastníka
1	7	7	6	8	8	7	2	6.43
2	10	11	9	9	10	9	4	8.86
3	8	7	5	6	7	6	3	6.00
4	3	2	3	4	6	5	4	3.86
5	9	8	9	8	8	9	3	7.71
6	4	3	5	5	6	6	3	4.57
7	9	8	7	8	9	7	2	7.14
8	4	5	4	5	5	6	5	4.86
9	8	9	10	10	8	9	4	8.29
10	3	4	4	5	6	5	2	4.14
11	6	7	7	8	8	6	2	6.29
12	7	6	5	6	7	6	4	5.86
Průměr formy	6.50	6.42	6.17	6.83	7.33	6.75	3.17	**Průměr celkově 6.17**

Tabulka 5.1: Tabulka zobrazující počet splněných úkonů jednotlivých účastníků studie s určitými možnostmi interakce za jednu minutu

celkového průměru. Také je vidět, že výsledky pro ovládání ovladačem a rukou se postupně zlepšují. To odpovídá tomu, že si uživatelé stále zvykají na ovládání a také tomu, že si zapamatovávají, kde se jednotlivá tlačítka v menu nacházejí a nemusí je tedy tak dlouho hledat. Další důvod je ten, že uživatelé přestávali požívat ovládání hlasem. Raději vše ovládali pomocí rukou nebo ovladačem.

- **Ovládání ovladačem a rukou**

  V tabulce [5.1] je vidět, že absence ovládání hlasem zvýšila průměrně dosažené skóre o více jak 0.5 bodu oproti kombinacím ovládání ukazováním + ovládání hlasem a ovládání rukou + ovládání hlasem. Přisuzuji to primárně faktu, že uživatelé se již nepokoušeli o interakci hlasem, která je zpomalovala.

- **Implementovaná vylepšení**

  Po testování s uživateli byla aplikace vylepšena o zpětnou vazbu při ovládání hlasem. Pokaždé, když se vyhodnotí výstup z NLU, ozve se specifický zvukový tón, podle toho jestli se našla shoda se zájmem a entitou nebo ne. Byla zkrácena doba, během které se neregistrují dotyky prstem po kliknutí na položku v menu. Uživatelé vnímali, že byli občas zpomalováni nutností čekat, než uplyne požadovaná doba.

Kapitola 6

Závěr

Ve své práci jsem se snažil prostudovat možnosti hlasové komunikace ve virtuální realitě. Na základě analýzy stavu využívání ovládání hlasem ve VR jsem se rozhodl navrhnout komponentu pro Unity s využitím analýzy hlasu pomocí knihovny firmy Mama-AI a ověřit její praktické využití. Podařilo se mi implementovat hlasové ovládání do aplikace pro konfiguraci automobilu. Ověřil jsem možnost hlasového ovládání, ale při testování jsem odhalil určité nedostatky, které podle testovaných uživatelů bránily plnému využití hlasového ovládání v aplikaci. Zjistil jsem, že existuje zájem uživatelů na používání ovládání hlasem ve VR, ale to musí být schopné dokonale odpovídat a reagovat na požadavky uživatele. V mém případě šlo zejména o neschopnost rozeznat pokyny uživatelů. To bylo zaviněno špatným převodem hlasu na text. Navrženou aplikaci by bylo možné vylepšit následujícími úpravami:

- Použitím nové verze knihovny, která používá STT přímo od Mama-AI, by se měla vyřešit interní chyba v kódu knihovny Mama-AI při aktivaci Nexusu. Podle erroru vypsaného v Unity se zdá, že je něco špatně s odpovědí od Google STT.
- Dalším jednoduchým vylepšení by mohlo být rozšíření entit o další synonyma. Je zapotřebí nasbírat více vstupů od uživatelů a pomocí nich rozšířit model. Toto vylepšení posílí NLU model, který je dosti slabý a reaguje pouze na specifické podněty.
- Vylepšení aplikace by přineslo podrobnější nastavení toho, jak se strom menu prohledává. Nyní jde zakázat prohledávání stromu menu směrem nahoru, ale jediná možnost, jak zakázat prohledávání směrem dolů je

nepřipojovat podmenu vůbec. Současný způsob neumožňuje povolit pouze prohledávání nahoru.

Využití ovládání hlasem ve VR nabízí rozšíření zážitků uživatelů a bude to důležité v dalším rozvoji používání VR, neboť to zvýší komfort uživatelů. Domnívám se, že ovládání hlasem ve VR může přinést i ekonomické výhody pro aplikace, které jej budou využívat (jde třeba o uživatele, kteří nejsou schopni používat ovladače).

Příloha A

Literatura

[1] N. Washani and S. Sharma, "Speech recognition system: A review," *International Journal of Computer Applications*, vol. 115, pp. 7–10, 04 2015.

[2] J. Zhang, "Gradient descent based optimization algorithms for deep learning models training," *CoRR*, vol. abs/1903.03614, 2019.

[3] P. Das, K. Acharjee, P. Das, and V. Prasad, "Voice recognition system: Speech-to-text," *Journal of Applied and Fundamental Sciences*, vol. 1, pp. 2395–5562, 11 2015.

[4] F. Ernawan, N. Abu, and N. Suryana, "Spectrum analysis of speech recognition via discrete tchebichef transform," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8285, pp. 82856L–82856L, 10 2011.

[5] F. Beaufays, "The neural networks behind google voice transcription (2015)," *URL: https://research. googleblog. com/2015/08/the-neural-networks-behind-google-voice. html (visited on 02/06/2018)*.

[6] D. Jurafsky and J. H. Martin, "Speech and language processing. vol. 3," 2014.

[7] B. Liu and I. R. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *CoRR*, vol. abs/1609.01454, 2016.

[8] Meta, "What you can say with voice commands on meta quest," *https://www.meta.com/help/quest/articles/in-vr-experiences/oculus-features/what-you-can-say-with-voice-commands/*, 2023.

[9] Google, "How google assistant works," *https://developers.google.com/assistant/howassistantworks*, 2023.

[10] google, "Build actions from web content," *https://developers.google.com/assistant/content/overview*, 2023.

[11] Rasa, "Overview," *https://rasa.com/*, 2023.

[12] Unity, *Dokumentace XR Interaction Toolkit*, 2023.

[13] Unity, *About the Oculus XR Plugin*, 2023.

[14] J. Holdsworth, "The nature of breadth-first search," 02 1999.

[15] Unity, *The Package Manager*, 2023.

[16] Blender, *Blender website*, 2023.

[17] Meta, "Map controllers," *https://developer.oculus.com/documentation/unity/unity-ovrinput/*, 2023.

[18] Oculus, *Set Up Hand Tracking*, 2023.

[19] Manus, "Virtual reality gloves," *https://www.manus-meta.com/vr-gloves*, 2023.

Příloha B

Instalační příručka

Tato příručka obsahuje návod na instalaci projektu s novou verzí knihovny od Mama-AI, která již nepoužívá Google Speech to text. Nová verze se používá za účelem zaručení funkčnosti projektu do budoucnosti.

B.1 Spuštění v editoru Unity

Pro spuštění zdrojového kódu přímo v editoru Unity je potřeba změnit konfigurační soubory, které obsahují přístupové údaje k STT a NLU od Mama-AI.

1. Rozbalte komprimované zip soubory do vámi určeného adresáře. Všechny zmíněné soubory se vyskytují ve vzniklé složce a instrukce jsou udávány relativně k ní.

2. Přejděte do složky gardovit_PROJ_VR/Assets/Resources/Text a v její podsložce Player se nachází soubory config.txt. V nich jsou definované adresy pro přístup k NLU modelu a také přístupové údaje. Ty není potřeba nijak měnit. Pokud by však bylo zapotřebí změnit NLU model nebo přístupové údaje, můžete to učinit v těchto souborech. Soubor ve složce Text se používá přímo v editoru Unity a soubor ve složce Player je použit při vytváření APK.

3. Nyní můžete otevřít Unity projekt (složka gardovit_PROJ_VR) v Unity verze 2021.23.f1. Mělo by být možné použít i novější verze, ale projekt byl vyzkoušen pouze na této verzi.
4. Připojte svůj Meta Quest headset a spusťte Oculus link, aby se headset propojil s počítačem.
5. Ve složce Scenes najdete dvě scény. Jednu z nich si vyberte a spusťte pomocí tlačítka play na horní liště editoru a aplikaci si vyzkoušejte.

B.2 Spuštění pomocí APK souboru

Pokud nepotřebujete spustit scénu v editoru, stačí nainstalovat přiložené APK do vašeho Meta Quest. Pro instalaci můžete použít aplikaci SideQuest (https://sidequestvr.com/) nebo oficiální aplikaci Meta Quest Developement Hub (https://developer.oculus.com/documentation/unity/ts-odh/).

Příloha C

Návod na použití navržených komponent v dalších projektech

Pokud máte zájem o využití mnou navržených komponent pro další projekty, zde je stručný návod, jak postupovat. Předpokládám, že komponenty máte nahrané do editoru a máte správně nakonfigurovaný přístup k NLU a STT od Mama-AI.

Nejprve je zapotřebí připravit si komponenty z knihovny Mama-AI. Je potřeba vytvořit Nexus, AudioManager a AudioSaveManger. Pro přesné nastavení se podívejte do demonstrační aplikace, kde jsou definované specifické konstanty, které fungují. Nyní, když je knihovna připravená k použití, můžeme vytvořit instanci komponenty VoiceManager. Komponenta vyžaduje referenci na instanci XRController, kterou používá, aby zjistila, kdy se má hlas rozpoznávat, když se nepoužívá nepřetržité poslouchání (alwayslisten = true). Také je potřeba přidat referenci na Nexus a AudioManager, aby mohla komunikovat s knihovnou Mama-AI. Teď se podíváme na vytváření jednotlivých menu. Struktura následujících objektů není přesně daná, ale pouze doporučená, jelikož se nijak neprohledává graf objektů ve scéně a vše je řešené pomocí referencí. K objektu, který bude reprezentovat menu, přidejte VoiceMenu komponent. Nastavte referenci na předchozí menu ve stromové struktuře. Pokud se jedná o kořen nebo nechcete používat procházení stromu, tak referenci vynechte. Nyní vytvořte další objekt, který bude potomkem objektu s VoiceMenu. Přidejte VoiceAction a nastavte referenci na VoiceMenu nadřazeného objektu. Poté stačí už jen definovat název záměru, entit, na které bude komponenta reagovat, a akce, které se mají provést při shodě. Entity jsou ScriptableObject a tudíž jdou využívat napříč komponentami nebo scénami. Entity jde vytvářet stejně jako třeba novou scénu. Jelikož skript

pro přidání entity tohoto menu způsoboval chybu při sestavování aplikace do APK, musel jsem kód zakomentovat. Otevřete si tedy skript Entity.cs a odstraňte komentář kolem funkce CreateAsset(). Nyní můžete vytvářet Entity pomocí Assets->Create->Voice->Entity.

Celá struktura je analogií k normálními UI. VoiceMenu nahrazuje Canvas a VoiceAction je ekvivalent k Button. Tudíž pro přehledné použití můžete mít jeden objekt s komponentou Button i VoiceAction.

Když máte strukturu menu vytvořenou, stačí nastavit v VoiceManager počáteční VoiceMenu.

Teď už máte funkční ovládání menu. Jednotlivé akce definované v VoiceAction se budou při shodě provádět. Pro přechod na jiné menu stačí zavolat VoiceManager.Instance().SetActiveMenu(new_menu), kde new_menu je reference na VoiceMenu, kam se má VoiceManager přesunout.

Když máte vše nastavené, jste jen krůček od fungující aplikace. Ještě je potřeba mít správně natrénovaný NLU model. Pro jeho vytvoření a hostování kontaktujte společnost Mama-AI.

Příloha D

Text souboru pro trénování NLU modelu

```
- intent: SELECT_ITEM
  examples: |
    - select [Volvo](menu_item)
    - choose item [Volvo](menu_item)
    - choose [Volvo](menu_item)
    - i want [Volvo](menu_item)
    - i want to choose [Volvo](menu_item)
    - i want to pick [Volvo](menu_item)
    - press [Volvo](menu_item)
    - pick [Volvo](menu_item)
    - [Volvo](menu_item)
    - how would it look with [Volvo](menu_item)
    - how [Volvo](menu_item) looks
    - how [Volvo](menu_item) look like
    - how would [Volvo](menu_item) look
    - show me [Volvo](menu_item)
    - show [Volvo](menu_item)

- lookup: menu_item
  examples: |
    - Volvo
    - Aston Martin
    - Aston
    - Astona
    - Ferrari Enzo
    - Ferrari
    - Enzo
    - interior
```

```
    - exterior
    - green
    - greenish
    - red
    - black
    - white
    - pink
    - metallic
    - matte
    - pearl
    - twenty percent
    - fifty percent
    - eighty percent
    - first
    - second
    - third
    - fourth
    - fifth
    - last
    - 20 percent
    - 50 percent
    - 80 percent
    - leather
    - plastic
    - wood
    - wooden


- intent: CHANGE_MENU
   examples: |
     - open [color](menu) menu
     - switch to [color](menu) menu
     - switch to [color](menu)
     - choose [color](menu) menu
     - choose [color](menu)
     - [color](menu) setting
     - [color](menu) settings
     - [color](menu)
     - alter [color](menu)
     - i want [color](menu)
     - i want [color](menu) menu


změna interitéru
výběr barvy
nastavení interiréru
```

```
chci změnit barvu auta

- lookup: menu
  examples: |
    - interior
    - exterior
    - car
    - paint
    - paint type
    - paint color
    - interior color
    - exterior color
    - interior material


- intent: CHANGE_COLOR
   examples: |
     - change color to [green](color)
     - make it [green](color)
     - color [green](color)
     - I want [green](color) car
     - paint it [green](color)
     - paint the car [green](color)
     - paint [green](color)

- lookup: color
  examples: |
    - green
    - red
    - black
    - white
    - pink


- intent: WINDOWS_VISIBILITY_PLUS
   examples: |
     - make windows [dark](darker)
     - make window [dark](darker)
     - [dark](darker) window
     - i want [dark](darker) window
     - i want to see less through window
     - less sunlight

- lookup: darker
  examples: |
    - dark
```

```
    - darker
    - more black
    - more dark


- intent: WINDOWS_VISIBILITY_MINUS
   examples: |
     - make windows [light](lighter)
     - make window [light](lighter)
     - [light](lighter) window
     - i want [light](lighter) window
     - i want to see more through window
     - more sunlight

- lookup: darker
  examples: |
    - light
    - lighter
    - more light
    - more transparent
```

Příloha E

Struktura odevzdaných zdrojových kódů

```
root
├── creds
│   └── stt.json - soubor s přístupem k Google Speech to text
├── gardovit_PROJ_VR - Unity projekt
│   ├── Assets
│   │   ├── Assets
│   │   │   ├── Materials
│   │   │   ├── Models
│   │   │   ├── Prefabs
│   │   │   ├── ScriptableObjects
│   │   │   │   └── Entity - obsahuje složky pro různé druhy entit
│   │   │   ├── Scripts
│   │   │   │   ├── CarControl - scripty pro jednotlivé modely aut
│   │   │   │   ├── Komponenta - scripty s navrženými komponentami
│   │   │   │   └── Assets
│   │   │   ├── Sounds
│   │   │   └── Textures
│   │   ├── Resources .5 Text - složka se soubory potřebnými k fungování
│   │   │   Mama-AI pluginu
│   │   └── Scenes - složka s soubory Unity scén
│   ├── Packages
│   └── ProjectSettings
├── Demonstration.mp4 - Videoukázka aplikace
├── gardovit-Bachelor-thesis.apk - instalační soubor pro Meta Quest
└── Readme.txt - Návod na spuštění aplikace v editoru
```