



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Zadání bakalářské práce

| | |
|-----------------------------|--|
| Název: | Interakce v muzeu historie ve virtuální realitě. |
| Student: | Adam Polívka |
| Vedoucí: | Ing. Jiří Chludil |
| Studijní program: | Informatika |
| Obor / specializace: | Webové a softwarové inženýrství, zaměření Počítačová grafika |
| Katedra: | Katedra softwarového inženýrství |
| Platnost zadání: | do konce letního semestru 2022/2023 |

Pokyny pro vypracování

Práce si klade za cíl vytvořit sadu interakcí ve virtuální realitě použitelných pro prohlídky v muzeích.

1. Projděte a analyzujte podobné projekty a aplikace, které se tématu věnují.
2. Analyzujte systémy virtuální reality a nástroje pro vývoj ve virtuální realitě.
3. Pomocí metod softwarového inženýrství navrhnete sady vhodných interakcí v historickém muzeu se zaměřením na uživatelskou skupinu široké veřejnosti.
4. Vypracujte návod a sadu doporučení pro provoz v prostoru muzea.
5. Implementujte prototyp aplikace s vybranými navrženými interakcemi v Unity3D ve virtuální realitě.
6. Podrobně a vyhodnoťte prototyp vhodným uživatelským testům.

Elektronicky schválil/a Ing. Radek Richtř, Ph.D. dne 23. února 2021 v Praze.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Interakce v historickém muzeu ve virtuální realitě

Adam Polívka

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Chludil

27. června 2021

Poděkování

Rád bych poděkoval vedoucímu mé práce Ing. Jiřímu Chludilovi za nápomocné nápady, rady, konzultace a vedení této práce. Poděkování také patří mé rodině za nepřetržitou podporu během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 27. června 2021

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Adam Polívka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Polívka, Adam. *Interakce v historickém muzeu ve virtuální realitě*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem a implementací interakcí v muzeu ve virtuální realitě. V práci analyzuji druhy dostupného hardwaru pro platformu virtuální reality. Podrobněji rozeberu zařízení HTC Vive, které je cílovým prostředím prototypu navržené aplikace. Dále se zaměřím na dostupný software určený pro realizaci aplikací virtuální reality a podrobněji popíšu Unity engine a Unity XR Interaction Toolkit, které použiji při implementaci výsledného prototypu. Ke konci analytické části práce prozkoumám aplikaci zabývající se virtuální prohlídkou společně s těmi, které obsahují zajímavé interakce. Na základě výstupů analýzy byla navrhována sada interakcí a pohybový model vhodných pro prostředí muzea.

Návrh obsahuje popis a vlastnosti chování jednotlivých interakcí, popsány pomocí různých metod softwarového návrhu. Celkem byly navrženy tři druhy interakcí (virtuální exponát, simulace střelby ze středověkých zbraní a skládací exponát) společně s modely pohybu po scéně pomocí teleportace a plynulého pohybu. Při návrhu i implementaci bylo dbáno na cílovou skupinu široké veřejnosti, jednoduchost a přístupnost pro nezkušené uživatele hrála velkou roli. Výstupní prototyp aplikace této práce implementuje navržené prvky. Prototyp slouží jako demonstrace interaktivního prostředí virtuální reality. V rámci implementace vznikne návod a sada doporučení pro provoz zařízení HTC Vive v prostoru muzea.

Klíčová slova virtální realita, interakce, interakce ve virtální realitě, Unity, XR interaction Toolkit, prohlídka muzea, historické muzeum, HTC Vive

Abstract

This bachelor thesis deals with the analysis, design, and implementation of interactions in a museum in virtual reality. In this thesis, I analyze the types of available hardware for the virtual reality platform. I will analyze the HTC Vive device in more detail as this device will be used to run the prototype application. I will also focus on available software designed for the implementation of virtual reality applications and describe in more detail the Unity engine and Unity XR Interaction Toolkit, which I will use for implementing the prototype. Towards the end of the analytical part, I will examine the applications dealing with the virtual tour and with those that contain interesting interactions. Based on the analysis output, a set of interactions and a movement model suitable for the museum environment will be designed.

The design contains a description and properties of the behavior of individual interactions described using various software design methods. Three types of interactions were designed for the prototype (virtual exhibit, simulation of shooting from medieval weapons, and puzzle exhibit) together with models of movement inside the interaction using teleportation and smooth movement. During the design and the implementation, attention was given to accommodate the target group of the general public, where simplicity and accessibility for inexperienced users play a big role. The output prototype consists of all three designed interactions. The prototype serves as a demonstration of an interactive virtual reality environment. As part of the implementation, a guide and set of recommendations for operating HTC Vive in the museum area.

Keywords virtual reality, interactions, interactions in virtual reality, Unity, XR Interaction Toolkit, museum tour, HTC Vive

Obsah

| | |
|---|-----------|
| Úvod | 1 |
| 1 Cíl práce | 3 |
| 2 Úvod do světa virtuálních realit | 5 |
| 3 Analýza | 7 |
| 3.1 Zařízení virtuální reality | 7 |
| 3.1.1 Výstupní zařízení | 7 |
| 3.1.2 Vstupní zařízení | 9 |
| 3.1.3 HTC Vive | 9 |
| 3.2 Nástroje dostupné pro vývoj ve VR | 11 |
| 3.2.1 Engine | 11 |
| 3.2.2 Software development kit | 13 |
| 3.3 Vybrané nástroje | 13 |
| 3.3.1 Unity | 14 |
| 3.3.2 Unity XR Interaction Toolkit | 15 |
| 3.4 Analýza stavajících aplikací pro VR | 19 |
| 3.4.1 Neos VR | 19 |
| 3.4.2 Remembering Pearl Harbor | 19 |
| 3.4.3 Masterworks: Journey Through History | 20 |
| 3.4.4 Kingdom Come: Deliverance | 21 |
| 4 Návrh | 23 |
| 4.1 Pohyb uživatele po scéně | 23 |
| 4.2 Sada navržených interakcí | 24 |
| 4.2.1 Virtuální exponát | 24 |
| 4.2.2 Simulace střelby ze středověkých zbraní | 25 |
| 4.2.3 Skládací exponát | 25 |
| 4.3 Model požadavků | 26 |

| | | |
|----------|--|-----------|
| 4.3.1 | Obecné požadavky prototypu | 26 |
| 4.3.2 | Interakce | 27 |
| 4.3.2.1 | Interaktivní exponát | 28 |
| 4.3.2.2 | Simulace střelby ze středověkých zbraní | 28 |
| 4.3.2.3 | Skládací exponát | 29 |
| 4.4 | Model domény a aktivit | 29 |
| 4.4.1 | Virtuální exponát | 29 |
| 4.4.2 | Simulace střelby ze středověkých zbraní | 32 |
| 4.4.3 | Skládací exponát | 33 |
| 4.5 | Použité technologie k vytvoření prototypu | 34 |
| 5 | Implementace | 37 |
| 5.1 | Pohyb ve virtuálním prostoru | 37 |
| 5.2 | Manipulace s objekty | 38 |
| 5.3 | Implementace navržených interakcí | 39 |
| 5.3.1 | Virtuální exponát | 39 |
| 5.3.2 | Simulace střelby ze středověkých zbraní | 41 |
| 5.3.2.1 | Společné části | 41 |
| 5.3.2.2 | Specifické části pro luk | 43 |
| 5.3.2.3 | Specifické části pro arbalest | 44 |
| 5.3.3 | Skládací exponát | 44 |
| 6 | Testování | 47 |
| 6.1 | Podkladové materiály pro testování | 47 |
| 6.2 | Průběh testování | 47 |
| 6.2.1 | Dotazník | 47 |
| 6.2.1.1 | Sada obecných otázek pro testery | 48 |
| 6.2.1.2 | Sada otázek pro testery pro jednotlivé interakce | 48 |
| 6.2.2 | Testovací scénáře | 48 |
| 6.2.2.1 | Pohyb a ovládání | 48 |
| 6.2.2.2 | Skládací exponát | 49 |
| 6.2.2.3 | Virtuální exponát | 50 |
| 6.2.2.4 | Simulace střelby ze středověkých zbraní | 50 |
| 6.3 | Výsledky testování | 51 |
| 6.3.1 | Shrnutí výsledků testování | 51 |
| | Závěr | 53 |
| | Literatura | 55 |
| | A Seznam použitých zkratk | 59 |
| | B Návod a sada doporučení pro provoz zařízení HTC Vive v prostoru muzea | 61 |

| | | |
|----------|--|-----------|
| B.1 | Sada doporučení pro provoz zařízení virtuální reality v prostoru muzea | 61 |
| B.1.1 | Vyhrazený prostor | 61 |
| B.1.2 | Zařízení s infračerveným zářením | 61 |
| B.1.3 | Zrcadla | 61 |
| B.1.4 | Hygienická opatření | 62 |
| B.2 | Zařízení HTC Vive a jeho komponenty | 62 |
| B.3 | Návod na zapojení HTC Vive | 63 |
| B.3.1 | Zapojení a rozmístění senzorů pohybu | 63 |
| B.3.2 | Zapojení rozhraní PC - VR | 65 |
| B.3.3 | Aplikace Steam VR a příprava místnosti | 68 |
| C | Obsah přiloženého CD | 71 |

Seznam obrázků

| | | |
|------|---|----|
| 2.1 | Virtuální, mixovaná a rozšířená realita [7] | 6 |
| 3.1 | Ovladače Oculus Touch, HTC Vive a Windows MR [16] | 9 |
| 3.2 | Sada pro virtuální realitu HTC Vive [17] | 10 |
| 3.3 | Detailní popis ovladače HTC Vive [18] | 11 |
| 3.4 | Struktura Unity XR frameworku a jeho práce s jednotlivými plat- formami [29] | 16 |
| 3.5 | Vývoj aplikace VR v Unity pro cílový HMD pomocí specifického plug-inu a alternativa pomocí Unity XR API. | 17 |
| 3.6 | Ilustrace vztahů mezi jednotlivými interactable skripty | 17 |
| 3.7 | Ilustrace vztahů mezi jednotlivými interactor skripty | 18 |
| 3.8 | Pohled vojáka na napadený přístav [32] | 20 |
| 3.9 | Náhled do dobového příbytku civilisty [32] | 20 |
| 3.10 | Detail exponátu s interaktivním informačním panelem [33] | 21 |
| 3.11 | Ilustrace pro Kingdom Come: Deliverance [35] | 22 |
| 4.1 | Ilustrace simulace střelby z luku | 25 |
| 4.2 | Obecné požadavky pro prototyp | 26 |
| 4.3 | Obecné požadavky pro prototyp | 27 |
| 4.4 | Diagram aktivit pro informační panel | 30 |
| 4.5 | Diagram aktivit pro informační panel | 31 |
| 4.6 | Doménový diagram střelby ze středověkých zbraní | 32 |
| 4.7 | Doménový diagram aktivit střelby ze středověkých zbraní | 33 |
| 4.8 | Doménový diagram skládačky | 34 |
| 4.9 | Aktivity diagram skládačky | 34 |
| 5.1 | Ilustrace implementované interakce virtuálního exponátu | 39 |
| 5.2 | Ilustrace implementované interakce simulace střelby ze středověkých zbraní | 41 |
| 5.3 | Ilustrace implementované interakce skládací exponát | 44 |

| | | |
|------|--|----|
| B.1 | Ilustrace vycházejícího infračerveného záření vycházejícího ze stanic [39] | 62 |
| B.2 | Popsané komponenty sady HTC Vive | 63 |
| B.3 | Zapojení napájecích kabelů do snímacích stanic | 64 |
| B.4 | Snímací stanice propojené synchronizačním kabele | 64 |
| B.5 | Konfigurace stanic pro bezdrátovou | 65 |
| B.6 | Konfigurace stanic pro kabelovou synchronizaci | 65 |
| B.7 | Zařízení pro propojení headsetu a počítače | 66 |
| B.8 | Zapojení kabelů headsetu do PC - VR rozhraní. | 66 |
| B.9 | Kompletace zapojení PC - VR rozhraní | 67 |
| B.10 | Zapínací tlačítko headsetu | 67 |
| B.11 | Tlačítko na úpravu vzdálenosti čoček | 68 |
| B.12 | Spuštění programu Steam VR přes aplikaci Steam | 68 |
| B.13 | Program <i>Steam VR</i> zobrazující aktuální status zapojení virtuálního headsetu. | 69 |
| B.14 | Program <i>Steam VR</i> zobrazující aktuální status zapojení virtuálního headsetu. | 69 |
| B.15 | Spuštění přípravy místnosti v aplikaci <i>Steam VR</i> | 70 |

Úvod

V poslední době se s pojmem virtuální reality setkáváme čím dál více. Díky rapidnímu zvýšení průměrného výkonu dostupných hardwarových komponentů se technologie virtuální reality rozšířila i do rukou běžných uživatelů.

Vývoj virtuální reality (dále VR) jde neustále kupředu a roširujícím se povědomím a znalostech o VR široké veřejnosti se objevují nové nápady a vize pro aplikaci této technologie v rámci různých odvětví. Jedním takovým je propojení virtuální reality a prohlídky muzea.

Motivací pro výběr tohoto tématu je příležitost prozkoumat hlouběji práci s technologií virtuální reality. Zapojení této technologie do prohlídek muzeí má dle mého názoru velkou budoucnost a potenciál úspěšně rozšířit klasický zážitek z návštěvy muzea. V průběhu studia jsem se podílel na práci na projektu *Věnná města českých královen*, který se zabývá propojením historického dědictví s virtuální realitou. Působení na tomto projektu dále prohloubilo motivaci k výběru tohoto tématu.

Cíl práce

Hlavním cílem této bakalářské práce je předvést interakce ve virtuální realitě. Zaměření těchto interakcí je na prostředí historického muzea s cílovou skupinou široké veřejnosti.

Cílovým zařízením je HTC Vive, kterému se budu v této práci podrobně věnovat. Jako vývojové prostředí bude použitý herní engine Unity, který bude prozkoumán společně s dostupnými prostředky pro vývoj aplikací virtuální reality v tomto enginu. Z těchto nástrojů bude vybrán jeden, který bude použit pro implementaci výsledného prototypu. Dále budou pospány základní SDK, které tento nástroj nabízí pro vývoj virtuální reality.

Návrh interakcí, které budou vycházet z analýzy aplikací zabývajících se problematikou virtuálních muzeí a zároveň těch, které implementují zajímavé prvky, které by se daly použít pro obohacení prohlídky muzea.

Na základě návrhu bude vytvořen prototyp implementující navržené interakce, který bude sloužit jako inspirace pro budoucí projekty zabývajících se podobnou problematikou. Aplikace prototypu bude implementována v enginu Unity za pomoci zvolených nástrojů pro vývoj pro virtuální realitu. Součástí výstupu bude návod a sada doporučení pro provoz zařízení HTC Vive v prostoru muzea.

Po implementační části v rámci testovací fáze dojde k vytvoření testovacích scénářů, které budou použity k uživatelskému testování a tedy získání zpětné vazby k implementovaným interakcím v prototypu

Úvod do světa virtuálních realit

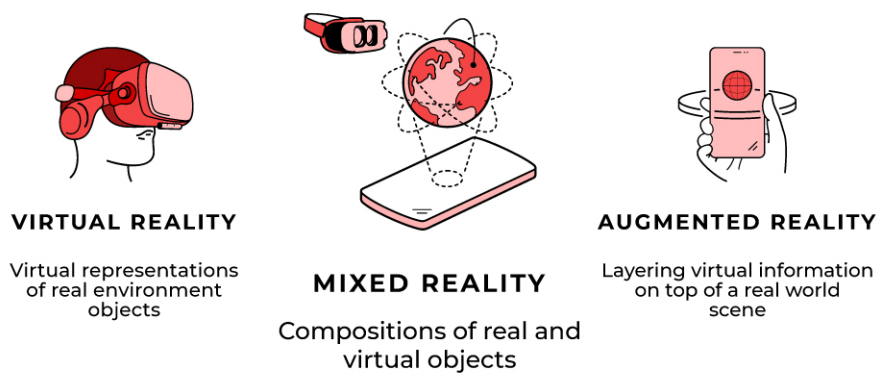
Extended reality (XR) [1] Je pojem, pro který zatím není ustanovený český název, zastřešující všechny počítačem vytvořené prostředí, kombinujících reálný svět s tím virtuálním. Pod tento pojem spadá například rozšířená realita (*Augmented reality*, neboli AR), mixovaná realita (*Mixed reality*, neboli MR) a virtuální realita (*Virtual reality*, neboli VR), které se od sebe navzájem liší mírou ponoření uživatele do virtuálního prostředí.

Smíšená realita (AR) (augmented reality) [2] je technologie založená na obohacení reálného světa prostřednictvím digitální kamery a displeje. V principu se jedná o vložení virtuálního objektu do reálného obrazu, který je zobrazený pomocí displeje (nejčastěji na chytré telefonu nebo tabletu). Mezi známé aplikace smíšené reality je například *Pokemon GO* [3].

Mixovaná realita (MR) (mixed reality) [4] je spojení reálného a virtuálního světa dohromady. Cílem je projekce prvků virtuálního světa do reálného života společně s následnou interakcí s těmito objekty. Interakce s virtuálními objekty probíhá pouze za pomoci očí a rukou uživatele. Z uvedených odvětví je mixovaná realita zatím rozvinuta nejméně. V této době je hlavním průkopníkem společnost Microsoft se svými *Microsoft HoloLens* a *Microsoft HoloLens 2* [5].

Virtuální realita (VR) (virtuální realita) [6] je technologie umožňující se naplno ponořit do počítačově simulovaného 3D světa. Uživatel vidí pouze promítaný obraz virtuálního světa, zbytek jeho zorného pole je zastíněna. V dnešní době se primárně používá v zábavním průmyslu, ale potenciál této technologie rozšířit se i do ostatních odvětví je obrovský. Tato bakalářská práce se soustředí na tento druh technologií.

2. ÚVOD DO SVĚTA VIRTUÁLNÍCH REALIT



Obrázek 2.1: Virtuální, mixovaná a rozšířená realita [7]

Analýza

V této kapitole se budu zabývat analýzou dostupných zařízení a vývojových platforem pro vývoj aplikací pro virtuální realitu. Jako první se zaměřím na hardwarové zařízení virtuální reality a výběr vhodného headsetu, který bude použit k vývoji prototypu. Dále budou rozebrány jednotlivé enginy a nástroje pro tvorbu aplikací virtuální reality. Nakonec se zaměřím na analýzu projektů a aplikací, které se věnují tématu mé práce.

3.1 Zařízení virtuální reality

Virtuální realita je velmi úzce spojena s Head-mounted displeji (HMD), které umožňují kompletní kontrolu nad tím, co uživatel virtuální reality vidí ve svém zorném poli. Tyto zařízení se dále rozdělují do podskupin, které budou rozebrány v následujících odstavcích.

Obecně lze hardware pro virtuální realitu rozdělit do dvou skupin. Podobně jako například u stolního PC se jedná o vstupní (klávesnice, počítačová myš) a výstupní zařízení (monitor). Stěžejním pilířem hardwaru pro virtuální realitu jsou již zmíněné HMD. Zbýlý hardware tvoří převážně doplňky zajišťující správné fungování headsetu, zařízení pro zachycení vstupů zadaných uživatelem a různé doplňky pro umocnění míry pohlcení uživatele virtuálním světem.

3.1.1 Výstupní zařízení

Typ zařízení zprostředkující uživateli prostředí virtuální reality.

Head-mounted displeje Nejdůležitějším výstupním zařízením jsou displeje pro zobrazení vizualizace virtuálního světa. V rámci virtuální reality se takřka výhradně setkáme s HMD. Ty můžeme rozdělit do následujících kategorií. [8]

3. ANALÝZA

- Počítačové HMD - Jsou headsety, které k provozu potřebují stále připojení k počítači. Mezi nejznámější headsety z této kategorie patří například HTC Vive, Oculus Rift nebo Valve Index. Vzhledem k tomu, že se hardware zajišťující rendering snímků a výpočet logiky aplikací nachází mimo headset jako takový, mohou se výrobci těchto druhů HMD při návrhu soustředit na rozšíření funkcionalit a výkonu zobrazovacího aparátu. Mezi parametry pomocí kterých tyto aspekty headsetů můžeme měřit, patří primárně obnovovací frekvence displeje, rozlišení displeje a vzdálenost mezi jednotlivými pixely.
- Standalone HMD - Zařízení, které jsou snadno vzhledově zaměnitelné s počítačovými HMD, ale na rozdíl od nich nepotřebují k provozu připojení k žádnému externímu zařízení. Není tedy třeba se při použití limitovat dostupným výkonem dodatečného zařízení. Do této kategorie spadá například *Oculus Quest 2*, který je zcela nezávislý, ale v případě nouze lze spravovat pomocí mobilního zařízení nebo případného připojení pomocí příslušného kabelu k počítači.
- Konzolové HMD - K provozu potřebují příslušnou herní konzoli do které se headset zapojí. Průkopníkem této kategorie je společnost Sony, která v 2016 představila *Playstation VR* jako doplněk dokoupitelný k herní konzoli Playstation 4. I přestože se jedná o poměrně starý headset na dnešní dobu, stále patří mezi jeden z lepší HMD dostupných na trhu. Pro rok 2021 je naplánováno představení druhé generace virtuální reality pro Playstation [9]. Vzhledem k tomu, že se jedná jako doplňkové zařízení k herní konzoli, hardwarová optimalizace je poměrně na vysoké úrovni.
- Mobilní HMD - Oproti ostatním kategoriím se nejedná přímo o zařízení, které by zobrazovalo obraz, ale pouze forma schránky, do kterého uživatel vloží svůj chytrý telefon. Toto mobilní zařízení nahradí vestavěný display, který najdeme v MHD z ostatních kategorií. Produkty z této kategorie slouží spíše jako zprostředkovatelé než přímo zařízení virtuální reality. Výkon je tedy plně závislý na modelu použitého chytrého mobilního zařízení. Mezi mobilní HMD patří například Google Cardboard [10].

Haptické zařízení Zařízení se zaměřením na stimulaci lidských vjemů vnímaných tělem zkrze lidskou kůži. V rámci virtuální reality se zatím jedná primárně o pocit dotyku a odporu.

Nejrozšířenějším druhem haptického zařízení jsou běžné ovladače k HMD, které jsou schopné vibracemi stimulovat hmatové smysly uživatele například při kontaktu rukou s objekty v aplikaci. Mezi další druhy haptických zařízení patří například specifické ovladače a různé obleky, které jsou daleko pokročilejší v rámci stimulace haptických vjemů a nabízí tím uživateli hlubší ponoření do virtuálního světa [11].

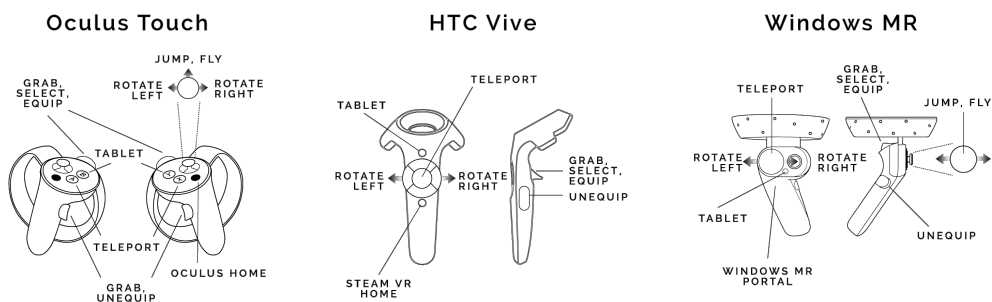
3.1.2 Vstupní zařízení

Typ zařízení, které snímají a předávají vstupy zadané uživatelem headsetu a aplikacím.

Ovladače Ovladače kompatibilní s použitým headsetem jsou primárním vstupním zařízením pro zařízení virtuální reality. Zpravidla jsou párové, pro každou ruku jeden.

Design jednotlivých ovladačů se výrazně liší mezi jednotlivými výrobci. Hlavním rozdílem v oblasti vstupů, které jsou uživateli k dispozici je zahrnutí *thumbsticku* (Oculus Touch)[12], *trackpadu* (HTC Vive) nebo dokonce kombinaci obou dohromady (Windows MR a Valve Index). Dále počet standardních tlačítek, kde Windows MR [13] nepoužívá žádné, HTC Vive [14] pouze jedno a Valve Index [15] společně s Oculus Touch 2. Na druhou stranu, všechny uvedené zařízení mají svým způsobem zahrnuté zadní *trigger*, postranní *grab* a *home* tlačítko, které se na ovladači nachází v okolí *thumbsticku* případně *trackpadu*.

Krom informací o stlačení tlačítek dále ovladače posílají detaily o pozici a svém natočení. Zmíněné ovladače s rozložením tlačítek je možné vidět na obrázku 3.1



Obrázek 3.1: Ovladače Oculus Touch, HTC Vive a Windows MR [16]

3.1.3 HTC Vive

Jako headset pro vývoj a posléze testování prototypu jsem zvolil sadu pro virtuální realitu HTC Vive. První generace HTC Vive byla představena v roce 2016 jako jeden z prvních HDM, který byl k dispozici na trhu pro širokou veřejnost.

Sada obsahuje headset s dvěma displeji, jeden pro každé oko, s rozlišením 1080 x 1200 pixelů, obnovovací frekvencí 90 Hz a technologií *Frenelových čoček*. Dále dva ovladače, které slouží ke snímání vstupů zadaných uživatelem. Pohyb ovladačů a headsetu v prostoru snímá pár statických stanic (*Base stati-*

3. ANALÝZA

ons), tento druh trackování se nazývá *lighthouse tracking*. Stanice emitují laserové paprsky o frekvenci 60 Hz, které headset s ovladači zachytávají pomocí světelných senzorů. Díky informacím zachyceným senzory, zařízení vypočítají svoji polohu a rotaci vůči stanicím. Jednotlivé stanice tedy nekomunikují s zařízením jako takovým, ale pouze poskytují projekci paprsků. Připojení stanic k PC přes USB kabel je nutné pouze v případě technických potíží. Aby tento model trackování mohl fungovat, je nutné, aby na sebe obě stanice navzájem viděly. Headset jako takový je k PC připojen pomocí speciálního propojovacího zařízení, který se skládá z USB, napájení a HDMI portu. Veškeré části HTC Vive jsou zobrazeny na obrázku 3.2 [14].

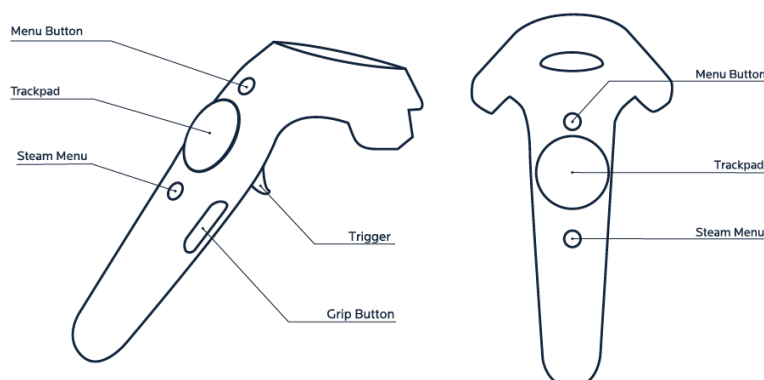


Obrázek 3.2: Sada pro virtuální realitu HTC Vive [17]

Ovladače HTC Vive V návrhu implementace jednotlivých interakcí se budu odvolávat na jednotlivé tlačítka na ovladačích. Proto se následující sekce věnuje podrobnému popisu jednotlivých vstupních periférií ovladače headsetu HTC Vive ilustrovaných na obrázku 3.3.

- Steam Menu - tlačítko jehož zmáčknutí zobrazí Steam menu, které slouží jako rozcestník mezi jednotlivými aplikacemi. V tomto menu můžeme přecházet mezi aplikacemi virtuální reality (vypnout jednu a zapnout druhou) bez nutnosti sundání headsetu z hlavy a spuštění aplikace přes počítač pomocí myši nebo klávesnice. Zároveň umožňuje si zobrazit počítačovou plochu a interagovat s ní pomocí kontrolerů virtuální reality.

- Trackpad - HTC narozdíl od velké většiny ostatních ovladačů virtuální reality nemá joysticky, který je ovšem nahrazen trackpadem. Ten umí nejen simulovat joystick pomocí kruhového povrchu snímající pohyb prstu, díky tomu je možné prstem pohybovat po kruhovém tlačítku jako při pohybu joystickem, ale zároveň nabízí možnost stisknutí jednotlivých oblastí. Nabízí tedy až o 4 možné vstupy navíc oproti běžnému joysticku.
- Menu Button - tlačítko sloužící ve většině případů k vyvolání menu spuštěné aplikace.
- Grip Button - párové tlačítko umístěné bocích spodní části ovladače. Vzhledem k tomu, že ovladače nemají určeno, který z nich bude uchopen pravou, případně levou rukou, nezáleží na tom, na které straně uživatel stiskne tlačítko.
- Trigger - tlačítko spouště, je schopné tedy zaznamenat nejen zda je stlačeno, ale i intenzitu stlačení. Pravděpodobně nejdůležitější tlačítko pro navigaci uživatelským rozhraním jednotlivých aplikací, kde hraje podobnou roli jako levé tlačítko na počítačové myši. Jedná se o tlačítko spouště.



Obrázek 3.3: Detailní popis ovladače HTC Vive [18]

3.2 Nástroje dostupné pro vývoj ve VR

3.2.1 Engine

Enginů pro tvorbu aplikací pro virtuální realitu existuje několik enginů. Každý z nich má své klady a zápory, liší se také v rámci jejich cenové dostupnosti. Pro účely bakalářské práce jsem zvažoval dva hlavní kandidáty, Unity a Unreal Engine. Tyto enginy jsou ideální pro mou práci z důvodu cenové dostupnosti, nabídkou funkcionalit, rozsáhlou podporou knihoven a širokému

množství dostupných edukativních materiálů. Při analýze vývojové platformy jsem vycházel převážně ze svých vlastních zkušeností se zmíněnými enginy a z jejich uživatelských recenzí publikovaných na internetu [19].

Unreal Engine Unreal Engine je platforma používaná mnoha velkými studii při tvorbě profesionálních aplikací. Nabízí obrosé grafické prostředky, pomocí kterých lze snadno vytvořit profesionálně vypadající aplikace. Vzhledem k tomu, že se Unreal Engine profiluje na větší komerční týmy profesionálních vývojářů než na nezávislé či začínající vývojáře, nemá za sebou tak rozsáhlou komunitu, která by tvořila takové množství tutorialů a doplňkových knihoven jako například komunita kolem enginu Unity. Vývoj v Unreal Enginu může být pro nováčky a středně pokročilé složitější oproti ostatním enginům díky své komplexnosti. Podporuje programovací jazyk C++ a skriptování v Pythonu, ale také nabízí možnost řízení chování aplikace pomocí grafického programování, tzv blueprints. Unreal Engine umožňuje práci na multiplatformních aplikacích, je tedy možné vyvíjet například pro PC, Android a Mac OS zároveň. Pro nekomerční vývoj je engine jako takový dostupný zdarma a zároveň nabízí podporu kolaborace na projektech pro menší týmy. Publikované aplikace vytvořené pomocí Unreal Enginu, které překročí tržbu 3 000 amerických dolarů za poslední čtvrtletí, si Epic Games, společnost za Unreal Enginem, účtuje poplatek 5% výdělky [20].

Unity Pravděpodobně nejrozšířenější platforma pro tvorbu aplikací a her mezi nezávislými vývojáři a menšími vývojovými studii je Unity. Velkou výhodou je její dostupnost a přívětivý workflow nejen, který lichoť nejen začínajícím vývojářům. Od roku 2005, kdy byla publikována první verze enginu Unity, si zvládl okolo sebe vybudovat rozsáhlou komunitu aktivních vývojářů. Tato komunita vytváří sady rozšiřujících balíčků, které jsou volně nebo za menší poplatek dostupné ke stažení v integrovaném obchodě. Tyto stáhnutelné balíčky mohou znatelně usnadnit a zároveň urychlit vývoj aplikací v Unity. Nativně podporuje skriptování v programovacím jazyce C#, ale oproti Unreal Enginu v této době ještě stále plně nepodporuje možnost plnohodnotného vývoje pomocí grafického programování. Na druhou stranu se zase může chlubit ohromným pokrytím podporovaných cílových platforem pro vývoj aplikací. Podporované jsou krom standartních desktovových operačních systémů jako jsou Windows a Mac OS tak zároveň i mobilní telefony až po herní konzole. Engine je zcela zdarma za podmínky, že příjmy z aplikace vytvořené pomocí Unity nepřesáhnou 100 000 amerických dolarů za rok. Dále poskytuje vlastní formu verzovacího systému, Unity collab, díky kterému může spolupracovat na projektu až 5 vývojářů zcela zdarma [21].

3.2.2 Software development kit

Sekce zabývající se analýzou SDK (*Software development kit*) dostupných pro vývoj aplikací VR.

Mixed Reality Toolkit Zkratkou MRTK [22] je sada vývojových nástrojů od pro virtuální a rozšířenou realitu od společnosti Microsoft. Jeho hlavním zaměřením je vývoj pro první a druhou generaci headsetu Microsoft HoloLens, které místo ovladačů používají jako vstup holé ruce uživatele. K tomu zároveň nabízí podporu pro VR headsety jako například HTC Vive a Oculus Rift.

MRTK se vyznačuje v porovnání s ostatními balíčky podporou hand-trackingu, která hraje vitální roli při vývoji pro již zmíněné HoloLens. Zároveň také podporuje možnost spuštění a ovládní aplikace v Unity bez připojeného HMD. Tento fakt může přijít velmi vhod v případě, kdy vyvojář pracuje na implementaci aplikace a zároveň nemá nepřetržitý přístup k zařízení HMD.

Steam VR Aplikace *Steam VR* [23] vydaná společností Valve je volně přístupná přes platformu Steam, která primárně slouží pro nákup počítačových her, aplikací a komunikaci s komunitou. Zajišťuje funkcionalitu komunikace aplikace a cílového headsetu pro virtuální realitu.

Stejně pojmenovaný balíček pro vývojáře je volně dostupný pro herní engine (Unreal Engine, Unity, atd.) a poskytující podporu vývoje pro široký výběr virtuálních headsetů. Nabízí implementaci základních forem pohybu (teleport a lokomotion) a manipulace s předměty.

Unity XR Interaction Toolkit *Unity XR Interaction Toolkit* [24] je systém umožňující v rámci Unity vývoj pro zařízení VR (*Virtual reality, virtuální realita*), AR (*Augmented reality, rozšířená realita*), MR (*Mixed reality, mixovaná realita*) pomocí skriptů základních interakcí. Poprvé byl představen jako součást Unity verze 2019.3 s cílem sjednotit vývoj a implementaci základních interakcí pro vývoj XR aplikací v Unity.

Hlavní výhodou *Unity XR Interaction Toolkit* je, krom velmi chytrého návrhu jednotlivých předprogramovaných interakcí, podpora velké většiny HMD vyskytujících se na trhu. Pro některé z nich, převážně pro ty běžící na systému Android, je ovšem nutné stáhnout dodatečný rozšiřující balíček.

3.3 Vybrané nástroje

V této části se budu zabývat analýzou použitých nástrojů a platforem pro vývoj této práce a implementaci výsledného prototypu.

Pro účely bakalářské práce jsem vybral pro vývoj prototypu Unity engine. Nabízí podporu většiny nejpoužívanějších frameworků pro vývoj aplikací pro virtuálních realitu jako například SteamVR, MRTK (Mixed reality tool kit) a

Unity XR, který jsem nakonec použil pro výslednou implementaci. Unity XR jsem si vybral z důvodu velmi chytře vyřešeného modelu interakcí s objekty ve virtuální realitě. Zároveň nabízí nezávislost aplikace na headsetu, na kterém aplikace poběží. Jako programovací jazyk pro psaní skriptů jsem zvolil C#, se kterým jsem se poměrně dobře seznámil během bakalářského studia. Pro psaní jednotlivých skriptů použiji Jet Brains Rider [25], který mám dostupná zdarma díky školní licenci. Unity má pro tento editor rozšiřující balíček, který velmi usnadní workflow při psaní jednotlivých skriptů.

3.3.1 Unity

Mezi hlavní důvody výběru engine Unity [21] patřila velmi dobře zpracovaná dokumentace a rozsáhlé komunitě. Díky velké popularitě a komunitě je na internetu k dispozici spousta naučných návodů, které budou použity jako jeden ze zdrojů informací při vývoji prototypu. Dalším faktorem byly také předešlé zkušenosti s vývojem v Unity engineu nasbírané v rámci bakalářského studia. Ten nabízí podporu většiny nejpoužívanějších frameworků pro vývoj aplikací pro virtuální realitu jako například SteamVR, MRTK (Mixed reality tool kit) a Unity XR, který jsem nakonec použil pro výslednou implementaci [19].

Komponenta v Unity Engine je postaven na komponentové architektuře. To znamená, že výsledné chování aplikace je složeno z jednotlivých komponent, které jsou velmi často zcela na sobě nezávislé. Komponenta je funkcionalita objektu, která je buďto předimplementována v rámci Unity nebo specifikována vlastnoručně napsanými skripty. Velkou výhodou této architektury je flexibilita a přehlednost nad parametry jednotlivých komponent přiřazeným k objektům. Díky tomu jdou hodnoty těchto parametrů velmi snadno pozměnit v Unity inspektoru i za běhu aplikace pomocí debug módu, který umožňuje velmi pohodlný způsob pro celkový debug a analýzu běhu aplikace. Příkladem komponenty je například *Box Collider*, která určuje kolizní oblast objektu a umožňuje ostatním objektům kolidovat s ním. S komponentami se nejčastěji setkáme v rámci tvorby a úpravy herních objektů obecně nazývaných *prefab*. Tento *prefab* je uložený herní objekt s na míru upravenými komponentami a nastavením [26].

Skriptování Primárním skriptovacím jazykem je C#. V dřívějších verzích bylo možné použití jazyků Boo a UnityScript, ale z důvodu malé popularity mezi vývojáři byla podpora těchto jazyků odstraněna. Skripty vytvořené v jazyce C# v rámci Unity z pravidla dědí z vestavěné třídy *MonoBehaviour*. Ta popisuje základní definice chování komponenty, to znamená, že třída dědicí z této třídy se automaticky stává komponentou. Nejpodstatnější vlastností této třídy jsou funkce události *events*, které jsou automaticky zavolány za běhu aplikace. Příkladem události je například:

- Metoda *Start()* je zavolána při prvním vytvoření herního objektu s touto komponentou. Používá se tedy pro inicializaci jednotlivých herních objektů při startu aplikace.
- Metoda *Awake()* je volána při každém vytvoření objektu, tedy ne pouze při prvním vytvoření objektu. Názorným použitím je za případu, že je potřeba objekt inicializovat při každém jeho vytvoření, například u vystřelených projektilů nastavení rychlosti.
- Metoda *Update()* se volá při každém snímku behu aplikace. Tedy pokud chceme provádět akci jako například detekce interakce s předmětem.
- Metoda *FixedUpdate()* má téměř identické chování jako metoda *Update()*. Rozdíl je v tom, že *FixedUpdate()* se volá v rámci interního cyklu enginu společně s kalkulací fyzických objektů ve scéně. Hodí se tedy pro implementaci náročnějších fyzikálních kalkulací.
- Metoda *LateUpdate()* je zavolána po doběhnutí metody *Update()* všech objektů ve scéně. Je tedy ideální pro provedení nějaké akce poté, co se objektem například pohne v rámci metody *Update()*.

Je samozřejmě možné vytvořit skripty, které z této třídy nedědí, ty ovšem nelze připnout k objektu jako komponentu [27].

Vizuální skriptování Je alternativou klasického skriptování, které umožňuje implementaci logiky aplikace bez nutnosti psaní klasického kódu. Programování pomocí vizuálního skriptování se skládá převážně ze spojování krabiček představujících jednotlivé předem nadefinované funkcionality. Vzhledem k tomu, že vizuálního skriptování nebylo původně vyvíjeno vývojáři přímo z Unity Technologies a jedná se o relativně nové rozšíření, se vývojáři z velké většiny přiklánějí k standardní formě skriptování v rámci enginu Unity. Tento způsob psaní kódu je velmi populární v enginu Unreal Engine nebo například 3D softwaru *Blender*, ve kterých naopak vizuální skriptování v rámci použití uživateli dominuje.

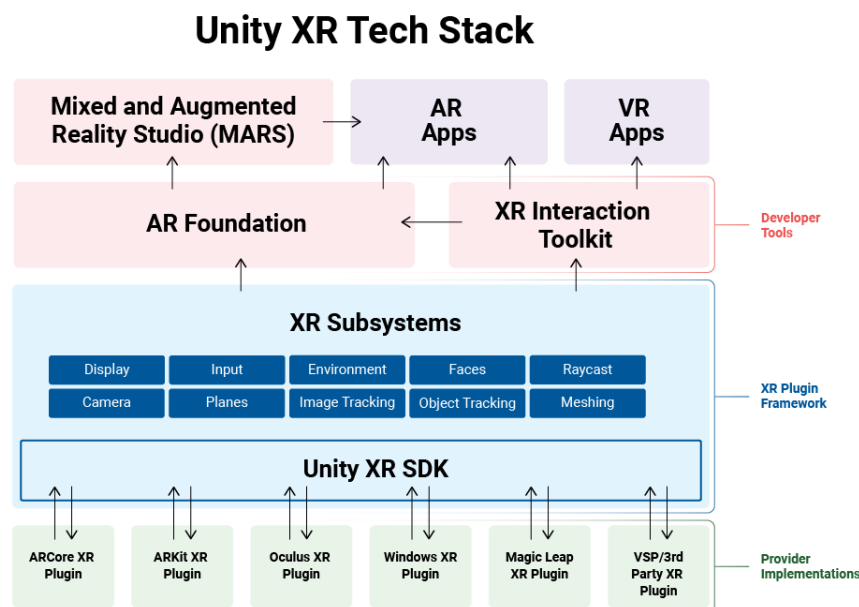
3.3.2 Unity XR Interaction Toolkit

Pro návrh a implementaci prototypu této práce jsem vybral *Unity XR Interaction Toolkit* z důvodu velmi chytrého návrhu základních interakcí, na kterých budu stavět při vývoji prototypu aplikace [24].

Jako platformu pro vývoj pomocí zmíněného Toolkitu se Unity snaží prosadit využití jejich vlastního systému *Unity XR* [28]. Ten je poměrně novým rozšířením pro vývoj XR aplikací v enginu Unity. Hlavním cílem je poskytnout robustní platformu s dlouhodobou podporou pro vývoj XR aplikací v Unity enginu. Snaha byla tedy postupem času odejmout podporu ostatních platform jako například *OpenVR*, kterou používá *SteamVR*. Oficiální podpora

3. ANALÝZA

pro zmíněného OpenVR byla přerušena od Unity verze 2019.3 a posléze byl nahrazen novým plug-inem *OpenXR*, který vyvinula společnost Valve pomocí Unity XR SDK [28] viz obrázek 3.4.



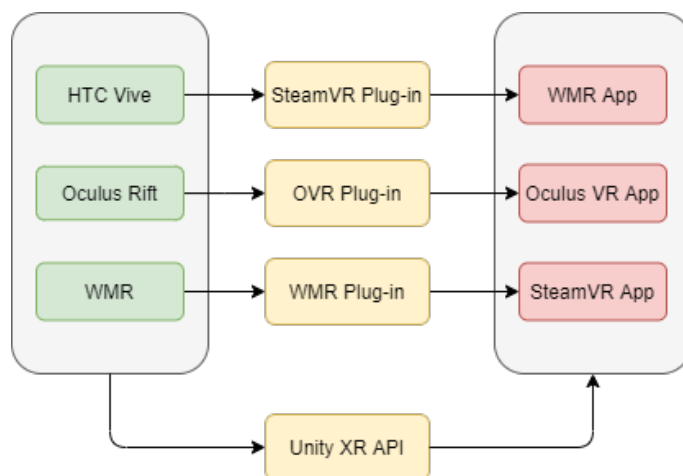
Obrázek 3.4: Struktura Unity XR frameworku a jeho práce s jednotlivými platformami [29]

Zároveň se mi také velmi zamlouvá myšlenka odproštění od nutnosti specificky přizpůsobovat implementaci kódu pro každé cílové zařízení HMD zvlášť. Ilustrace použití specifických SDK pro jednotlivé HMD a alternativní způsob pomocí Unity XR zobrazuje obrázek 3.5.

Základní předimplementované komponenty Balíček nabízí řadu předimplementovaných skriptů a komponent, pomocí kterých lze velmi rychle sestavit jednoduchou XR aplikaci [30]. Interakce mezi objekty jsou implementovány přes *Interactory* a *Interactables*, které mezi sebou komunikují pomocí *InteractionManager*. Mezi ty nejdůležitější patří následující skripty a komponenty:

Sledování headestu a ovladačů

- *XRRig* - je základní komponenta obstarávající logiku společně se sledováním VR kamery.
- *XRController* - dodá hernímu objektu základní implementaci pro sledování pohybu ovladačů.

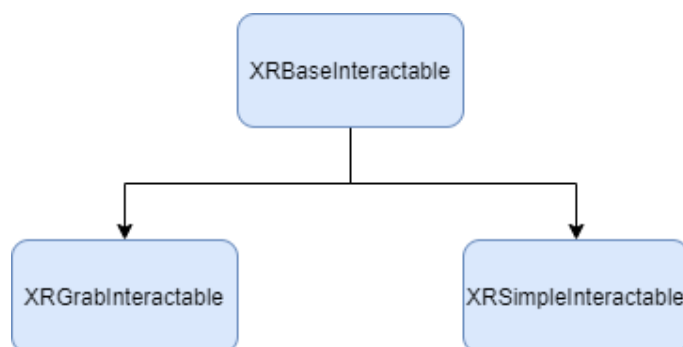


Obrázek 3.5: Vývoj aplikace VR v Unity pro cílový HMD pomocí specifického plug-inu a alternativa pomocí Unity XR API.

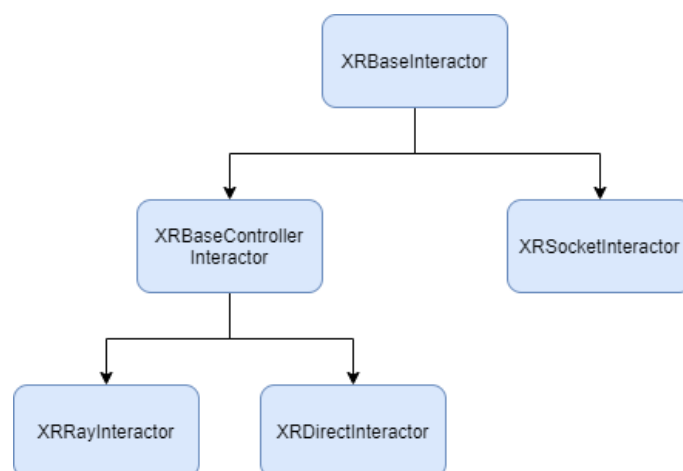
- *XRLocomotionSystem* - komponenta umožňující pohyb XR Rigu po scéně.

Interakce

- *XRBaseInteractable* - jeho připnutím nebo skriptu, který z něho dědí, se stane objekt interactable objektem 3.6. V praxi se nejčastěji setkáme s následujícími:
 - *XRSimpleInteractable* - dodává objektu schopnost jednoduché interakce (například selekce).
 - *XRGrabInteractable* - skript umožňující objektu možnost být sebrán pomocí *grab* akce. Tímto se připne na vybraný interactor (například ovladač) a následuje ho dokud není objekt puštěn.



Obrázek 3.6: Ilustrace vztahů mezi jednotlivými interactable skripty



Obrázek 3.7: Ilustrace vztahů mezi jednotlivými interactor skripty

- *XRBaseInteractor* - připnutím tohoto skriptu nebo skriptu, který z něho dědí, se stane objekt interactorem 3.6. V praxi se nejčastěji setkáme s následujícími:
 - *XRDirectInteractor* - skript používaný pro přímou interakci interaktoru s interactable objekty které se ho dotýkají.
 - *XRRayInteractor* - dodává objektu schopnost vzdálené interakce s interactable objekty pomocí laseru.
 - *XRSocketInteractor* - dodává objektu schopnost přichytit k sobě nějaký interactable objekt. Skript není navržen pro ovladač a nedědí ze třídy *XRBaseControllerInteractor*, která implementuje základní logiku pro interakci skrze ovladače.

Zpracování vstupů Pro sledování vstupů zadaných uživatelem balíček poskytuje dva způsoby, *Device-based* a *Action-based* systémy. Některé skripty v balíčku mají specifickou variantu pro každý ze zmíněných systémů a pro zaručení korektního fungování těchto skriptů, je třeba dávat pozor, kterou variantu používáme.

Starší *Device-based* systém získává informace o vstupu přímo ze vstupního zařízení. Nejprve tedy získám vstup specifický pro dané vstupní zařízení, uložím ho do proměnných a z těch pak vyhodnotím ručně. Jedná se o velmi přímočarou metodu implementace, která ovšem není flexibilní ani přehledná.

Alternativou je již zmíněný *Action-based*, který pracuje na bázi *Action* (akce) objektů. Na tyto akce se namapují vstupní tlačítka (například uchození objektu na tlačítko grab) a uloží se do setů mezi, mezi kterými poté může logika aplikace přepínat za běhu programu. Pro zpracování těchto akcí používá Unity svůj *Input system*, který snímá vstupy ze všech aktivních vstupních

zařízení. Velkou výhodou tohoto přístupu oproti Device-based systému je flexibilita a volnost, kterou vývojáři poskytuje při návrhu a implementaci vstupů pro svou aplikaci [30].

3.4 Analýza stavajících aplikací pro VR

V této sekci se budu zabývat analýzou aplikací a jejich interakcemi. Zaměřuji se na hry a aplikace pro virtuální realitu, které slouží jako virtuální prohlídka muzea, ale zároveň na ty s zajímavými interakcemi, které by mohly obohatit výsledný zážitek z návštěvy muzea. Z analýzy a vlastních nápadů bude sestaven návrh interakcí pro tuto práci.

3.4.1 Neos VR

Aplikace Neos VR je zaměřena na tvorbu virtuálních světů pro následné interakci více uživatelů mezi sebou. Vlastní světy můžou uživatelé tvořit pomocí umístění nabízených i vlastních naimportovaných objektů.

Aplikace obsahuje mnoho způsobů manipulace s objekty. Při interakci s předměty ve scéně si uživatel může vybrat mezi přímou interakcí, kde musí být rukou v přímém kontaktu, nebo interakcí na dálku pomocí ukazovátka.

Co se týče pohybu ve scéně Neos VR nabízí všechny klasické formy pohybu ve virtuální realitě. Je možné si vybrat mezi teleportem, kdy uživatel namíří na místo, kam se chce teleportovat a za předpokladu, že se jedná o validní lokaci, je jeho virtuální tělo přesunuto přesunuto na zvolenou pozici. Další implementovanou formou pohybu je *Continuous movement*, při kterém se kamera uživatele ve virtuálním světě pohybuje adekvátně ke vstupům, zadaným uživatelem pomocí *Joysticku* nebo případně *Touchpadu*. Oba výše zmíněné způsoby pohybu se mi velmi zamlouvají.

3.4.2 Remembering Pearl Harbor

Remembering Pearl Harbor [31] je aplikace slouží jako připomenutí 75. výročí napadení Spojených států amerických japonskými vojenskými silami. Uživateli se naskytne možnost prožít tuto historickou událost očima přeživšího vojáka 3.8. Druhý pohled je zprostředkovaný očima běžného civilisty a jeho dobového příbyku 3.9.

Aplikace zaujme především způsobem velmi detailním prostředím, které dokáže velmi snadno vtáhnout do scénérie, která se uživateli před očima. Halvní činností jakou může interagovat s okolím je prohlížení artefaktů a spouštění audionahrávek rozmístěných po scénách. Tyto artefakty je možné si podrobněji prohlédnout, případně si o nich přečíst informace navíc zobrazené na informačních tabulích.



Obrázek 3.8: Pohled vojáka na napadený přístav [32]



Obrázek 3.9: Náhled do dobového příbytku civilisty [32]

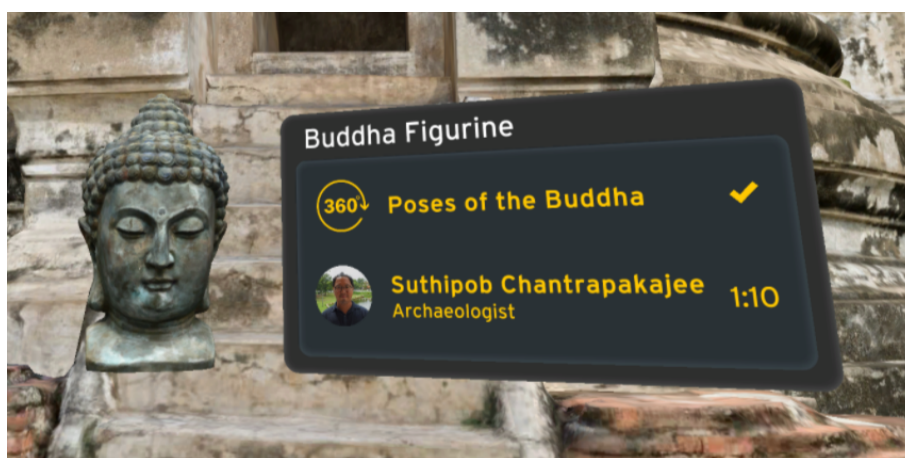
3.4.3 Masterworks: Journey Through History

Remembering Pearl Harbor [31] je volně ke stažení virtuální prohlídka čtyř historických lokací pod ochranou UNESCO. Aplikace je dostupná pro zařízení Oculus [12] a HTC Vive [14].

Pohyb uživatele v aplikaci je omezen pouze na teleportaci kombinovanou s *snap-turn* otačením. Teleportovat se lze pouze do předem vymezených ob-

lastí označených modrou vrstvou, která se objeví po zmáčknutí teleportačního tlačítka.

V rámci prohlídky narazí na zastávky, které se skládají z audionahrávek, 360 stupňových vyhlídek a artefktů. Pro tuto práci byly nejzajímavější artefakty, které po jejich rozkliknutí se zobrazí před uživatelem informační panel společně s detailním pohledem na artefkat. Tento informační panel obsahuje spustitelnou audionahrávku s iformacemi o artefaktu.



Obrázek 3.10: Detail exponátu s interaktivním informačním panelem [33]

3.4.4 Kingdom Come: Deliverance

Kingdom Come: Deliverance [34] je *role-playing* hra od české společnosti Warhorse Studios. Herní svět je posazen do historického prostředí, které reflektuje skutečné události, které se odehráli v roce 1403. Aplikace je cílena pro PC platformu, ale je možné si její části vyzkoušet i ve virtuální realitě, pomocí které se uživatel může naplno nechat pohltit vytvořeným světem. Hlavní náplní je dějová linka, která je obohacena vedlejšími aktivitami jak je u *role-playing* her zvykem.

Interakce v této hře byly navrženy, aby působily co nejrealisticky. Například aktivita lukostřelby nabízí na výběr několik druhů luků, které se od sebe navzájem liší vlastnostmi. Hlavním cílem lukostřelby v této hře je lov divokých zvířat a boj s nepřáteli. Při natažení tětiny luku se uživateli zobrazí zaměřovací indikátor.

3. ANALÝZA



Obrázek 3.11: Ilustrace pro Kingdom Come: Deliverance [35]

Návrh

Na základě analýzy bude navržen set interakcí vhodných pro cílové prostředí historického muzea. Navrhované interakce budou inspirovány interakcemi, které se objevovaly v analyzovaných aplikacích a vlastními nápady. Vybrané navržené interakce budou implementovány do prototypu, pomocí vybraných technologií pro vývoj aplikací virtuální reality. Zpracování návrhu jednotlivých interakcí bude uskotečněno pomocí modelů požadavků, domény a aktivit.

4.1 Pohyb uživatele po scéně

Při tvorbě aplikací pro virtuální realitu je důležité zajistit, aby se uživatel mohl snadno a intuitivně pohybovat po scéně. V případě, že by pohyb byl implementován nevhodným způsobem, razantně by se zvýšila možnost výskytu závratí společně s nevolností, která by udělala výslednou aplikaci či prototyp nepoužitelný. Za cílem vyhnout se těmto nežádaným efektům bylo navrženo několik standardní forem pohybu ve VR aplikacích, pomocí kterých lze minimalizovat výskyt zmíněných nežádaných efektů. Tyto způsoby pohybu jsou teleportace, continuous movement a fyzický pohyb po scéně. Kvůli rozdílným podmínkám (velikost místnosti, délka kabelu k headsetu), se kterými bude každý uživatel pracovat jsem se rozhodl pro implementaci kombinace všech tří způsobů, ze kterých si uživatel bude moci vybrat.

Teleportace Pro pohyb na velké vzdálenosti nebo čistě pro případ, že uživatel preferuje teleportační způsob pohybu je součástí návrhu prototypu teleportační systém. Teleport aktivuje uživatel stisknutím *Triggeru* na pravém kontroleru. Aby se uživatel neteleportoval omylem na místo, na které by neměl mít přístup, lze teleport uskutečnit pouze na vyhrazené plochy označené jako *Teleportation Area*.

Continuous movement Forma pohybu, kde uživatelova kamera ve scéně kopíruje vstupy, které zadá na *Touchpadu* levého kontroleru. Například v

případě, že zmáčkne na *Touchpadu* dopředu, jeho tělo ve scéně se také posune vpřed. Pro případ, kdy by se uživatel potřeboval vícekrát otočit dokola, je nutné předejít případnému zamotání uživatele do fyzických kabelů spojujících počítač s headsetem. Proto jsem přidal možnost otočení kamery ve scéně pomocí *Touchpadu* na pravém kontroleru.

Fyzický pohyb Pro precizní pohyb v rámci malého prostoru je ideální, aby pohyb uživatele ve scéně kopíroval pohyb v reálném životě. Při manipulaci s navrženými interakcemi bude tento zůsob pohybu vitální.

4.2 Sada navržených interakcí

Snaha navržených interakcí, které jsou rozebrány v této kapitole, je ukázat možnosti, jakými by virtuální realita mohla v budoucnu rozšířit prohlídku klasického muzea.

Jedním z hlavních faktorů při výběru typu navrhovaných interakcí byl fakt, který vyplynul z rozhovoru se správcem muzea, který aktivně projevil zájem o zapojení virtuální reality do svého muzea. Dle jeho názoru, by se nemelo jednat o virtuální prohlídku celého muzea, ale spíše pouze o prvek obohacující prohlídku reálného muzea.

Cílová skupina uživatelů pro tuto práci je široká veřejnost. Je tedy při návrhu třeba počítat s velmi nízkou úrovní zkušeností s virtuální realitou a vytvořit jednoduché interakce snadno přístupné nezkušeným uživatelům. Zároveň je důležité minimalizovat vlivy, které mohou způsobit nevolnost.

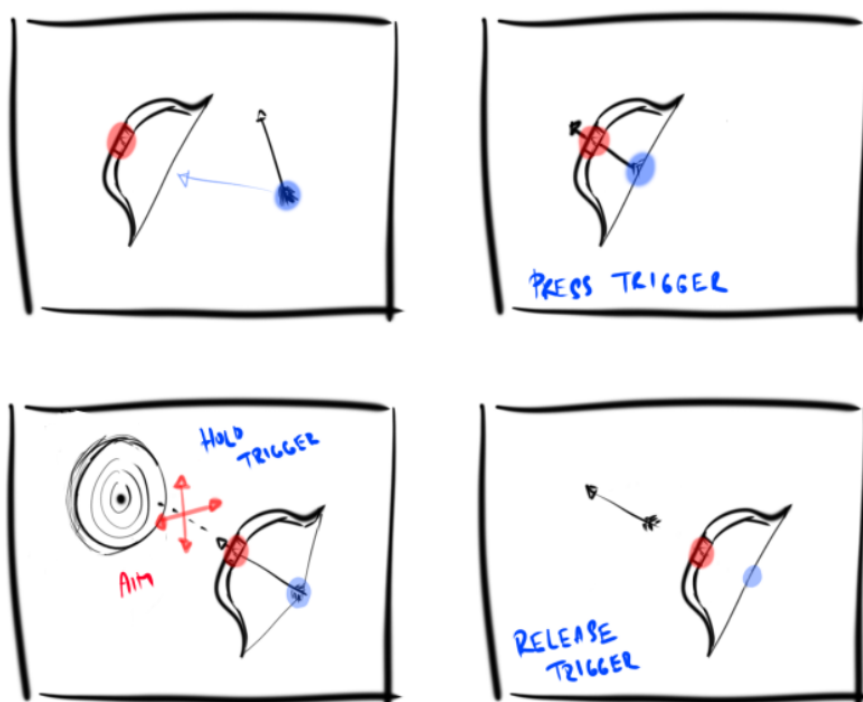
4.2.1 Virtuální exponát

I přestože není cílem práce vytvořit virtuální prohlídku muzea, rozhodl jsem se, že by jedna z navržených interakcí měla mít formu virtuálního exponátu. Tato interakce by umožnila doplnění výstav o exponáty, které nebylo možné získat či převést do prostoru budovy muzea. Virtuální exponát bude obsahovat to, co by bylo k nalezení u běžného modelu exponátu, písemný popis a exponát jako takový (v případě virtuálního exponátu nahrazený modelem), ale zároveň rozšíří tuto skupinu o možnost uchopit si exponát do ruky, libovolně si ho natočit a do určité míry zvětšovat a zmenšovat. Pro navození atmosféry bude při prohlídce exponátu hrát vhodně zvolená ambientní hudba, kterou bude možné vypnout pokud bude mít uživatel pocit, že ho hudba rozrušuje. V rámci virtuální reality občas bývá nekonfortní čtení textu, proto je do virtuálního exponátu zabudována nahrávka s namluveným informačním textem, kterou půjde přehrát.

4.2.2 Simulace střelby ze středověkých zbraní

Virtuální realita má velkou výhodu, že si za její pomoci může uživatel prožít nějakou situaci téměř na vlastní kůži. Pomocí simulace zprostředkované virtuální realitou uživatel může vyzkoušet obsluhu středověkých zbraní. Cílem interakce je přiblížit celkový postup obsluhy a mechanismus zbraně, ale zároveň demonstrovat jejich unikátní vlastnosti a získat perspektivu o tom jaké to, v rámci limitace simulace, bylo zacházet s těmito zbraněmi. Jedná se o autentický zážitek, vyzkoušet si něco, k čemu by za běžných podmínek bylo třeba obstarat repliky exponátů.

Tento druh interakce je v rámci této práce demontrován na simulaci střelby ze středověkých zbraní. To ovšem neznamená, že se myšlenka za touto interakcí omezuje pouze na obsluhu zbraní. Podobným způsobem by bylo možné vytvořit simulaci obsluhy různých nástrojů a minihry, které by provedli uživatelé různými řemesly.



Obrázek 4.1: Ilustrace simulace střelby z luku

4.2.3 Skládací exponát

Jednou z nevýhod některých exponátů je fakt, že se nedochovaly celé nebo byly nějakým způsobem značně poškozeny. Díky této interakci už nebude nutné si tyto nekompletní či poškozené exponáty pouze představovat, jak vy-

padaly v plné kráse. Model celého exponátu bude rozdělen do několika částí, například na dochovalé a nedochovalé části. Tyto části bude moci poskládat dohromady formou skládačky a po úspěšném složení si vzít do vlastních rukou a prohlédnout si ho v celé své kráse. Tato interakce usnadní uživatelům představivost, ale zároveň bude mít formu malé hry, která zabaví širokou veřejnost.

Potenciální možnosti jak tuto interakci rozšířit, které sahá rozsahem mimo tuto práci, by byla implementace archeologické minihry. V této minihře by se uživatel ocitl na archeologickém nalezišti, kde by jeho úkolem bylo nalézt všechny zakopaně části artefaktu a sestavilho.

4.3 Model požadavků

V této sekci bude zpracován model požadavků. Požadavky jsou kladeny obecně na celý prototyp a specificky na jednotlivé interakce. Všechny požadavky jsou rozděleny do dvou skupin funkčních a nefunkčních.

Funkční požadavky Definují nezbytné úkoly nebo akce, který musí prototyp vykonávat, v následujících sekcích jsou označeny *FP n* , kde n je číslo funkčního požadavku.

Nefunkční požadavky Definují požadavky na architekturu prototypu, uživatelské rozhraní a prostředí pro běh prototypu, v následujících sekcích jsou označeny *NP n* , kde n je číslo nefunkčního požadavku.

4.3.1 Obecné požadavky prototypu

Skupina požadavků, které jsou kladeny na celý prototyp nezávisle na konkrétních interakcích.

| Funkční požadavky | Nefunkční požadavky |
|----------------------|-----------------------------------|
| Obecné | Obecné |
| FP1 Pohyb | NP1 Virtuální realita |
| FP2 Uchopení objektů | NP2 Lokalizace v anglickém jazyce |
| | NP3 Unity |
| | NP4 UnityXR Interaction Toolkit |

Obrázek 4.2: Obecné požadavky pro prototyp

FP1 Pohyb Pohyb uživatele v prototypu je řízen pomocí continuous movementu a teleportu. Bude na uživateli, kterou metodu zvolí za vhodnou.

Obecně je continuous movement vhodnější pro přesnější pohyb na menší ploše, zato naopak teleport uživatel využije spíše pro pohyb na delší vzdálenosti.

FP2 Uchopení objektů V rámci prototypu uživatel může uchopit všechny objekty, se kterými může přímo interagovat. Uchopit bude moci předměty přímým kontaktem s nimi nebo za pomoci laserového ukazovátka, které přitáhne předmět do rukou uživatele.

NP1 Virtuální realita Prototyp aplikace obshující vybrané navržené interakce bude implementován pro platformu virtuální reality, konkrétně zařízení HTC Vive.

NP2 Lokalizce v anglickém jazyce Prototyp aplikace bude vyvíjen v anglickém jazyce. Veškeré popisky v rámci prototypu budou psány a dokumentace zdrojového kódu bude v angličtině.

NP3 Unity Prototyp aplikace obshující vybrané navržené interakce bude implementován pomocí Unity engineu.

NP4 UnityXR Interaction Toolkit Prototyp aplikace obshující vybrané navržené interakce bude implementován za pomoci vývojového balíčku UnityXR Interaction Toolkit.

4.3.2 Interakce

Požadavky na interakce jsou kladeny na každou interakci zvlášť.

| Funkční požadavky | Nefunkční požadavky |
|--|--|
| <p style="text-align: center;">Virtuální exponát</p> <p>FP1 Zobrazení infopanelu</p> <p>FP2 Spuštění audio nahrávky</p> <p>FP3 Manipulce s objektem</p> | <p style="text-align: center;">Virtuální exponát</p> <p>NP1 Audio nahrávka</p> <p>NP2 Transformace modelu exponátu</p> |
| <p style="text-align: center;">Simulace střelby ze středověkých zbraní</p> <p>FP1 Získání šípu</p> <p>FP2 Vystřelení z luku</p> <p>FP3 Vystřelení z arbalestu</p> | <p style="text-align: center;">Simulace střelby ze středověkých zbraní</p> <p>NP1 Fyzikální simulace projektilu</p> <p>NP2 Výpočet natažení tětiny luku</p> |
| <p style="text-align: center;">Skládací exponát</p> <p>FP1 Složení skládačky</p> | <p style="text-align: center;">Skládací exponát</p> <p>NP1 Rozdělení exponátů na menší díly</p> |

Obrázek 4.3: Obecné požadavky pro prototyp

4.3.2.1 Interaktivní exponát

FP1 Zobrazení infopanelu Pomocí namíření a rozkliknutí trigger tlačítkem na libovolném ovladači uživatel zobrazí infopanel s krátkým textem o daném objektu. Zobrazené informace budou obsahovat název objektu, historické informace a například v rámci exponátů místo jejich nálezu.

FP2 Spuštění audionahrávky Na infopanelu se bude nacházet tlačítko, které uživatel rozklikne trigger tlačítkem. Po rozkliknutí se spustí audionahrávka obsahující čtený text společně s vhodně vybranou hudbou a zvuky.

FP3 Manipulace s objektem Pomocí vhodně zpracovaného UI bude uživatel moci přiblížit, oddálit a libovolně natáčet vybrané objekty. V kontextu historického muzea převážně exponáty.

NP1 Audionahrávka Vytvoření nahrávky a její implementace do infopanelu umožňující přehrát danou nahrávku.

NP2 Transformace modelu exponátu Při uživatelské manipulaci s exponátem, který má přiděleným infopanel bude model exponátu transformován pomocí enginu Unity.

4.3.2.2 Simulace střelby ze středověkých zbraní

FP1 Získání šípu V rámci prototypu jsou šípy univerzální, tedy je možné použít šíp jako střelivo pro Luk i arbalest. Uživatel nasazuje šípy na zbraně implementované zbraně. Šípů může získat pomocí interakce s toulcem nebo sebráním libovolného šípu ve scéně, například již vystřelený šíp.

FP2 Vystřelení z luku Uživatel drží v jedné ruce luk a v druhé šíp. Položí šíp na vrub tětivy, na kterou se projektil automaticky nasadí. Uchopení tětivy a následná míra jejího natažení určuje kinematickou energii vystřeleného šípu. Natažený šíp uživatel vystřelí šíp puštěním tětivy.

FP3 Vystřelení z arbalestu Uživatel drží v jedné ruce arbalest a v druhé šíp, který umístí na tětivu. Tětivu natáhne pomocí klíčky umístěné na těle arbalestu. Natažený projektil uživatel vystřelí pomocí zmáčknutí spouště, která se nachází na zadní části. Projektilu je předána kinematická energie, která uvede šíp do pohybu.

NP1 Fyzikální simulace projektilu Trajektorie vystřeleného projektilu bude kalkulována pomocí vestavěných fyzikálních simulací, které jsou implementovány v Unity enginu. Při vypuštění je projektilu předána kinematická

energie tlačící ho kupředu. V průběhu letu na projektil působí gravitace. Projektil může být zastaven kolizí s hmotným objektem, který kinematickou energii absorbuje.

NP2 Výpočet natažení tětivy luku Výpočet pozice natahované tětivy od těla luku a následné odvození míry natažení společně s vizualizací tětivy.

4.3.2.3 Skládací exponát

FP1 Složení skládačky Exponát je rozdělen do několika částí, které jsou rozmístěny ve scéně. Finální podobu skládačky zobrazuje jeho silueta. Uživatel postupně vloží jednotlivé části skládačky na jejich určené místo. Po umístění všech částí je možné celý exponát lze uchopit a manipulovat s ním.

NP1 Socketový systém Části rozestavěné skládačky lze umístit do držáků (socketů), které jsou reprezentovány jako uzly. Při takovémto umístění socket aplikuje potřebnou transformaci na vloženou část. Každý socket kontroluje zda je do něj vložen správný kus skládačky. V případě, že jsou všechny uzly skládačky správně vyplněny, systém to identifikuje a vytvoří objekt představující složenou skládačku.

NP2 Rozdělení exponátů na menší díly Úprava modelů exponátů, který bude uživatel skládat. Nutnost rozdělit model na menší díly.

4.4 Model domény a aktivit

V následující sekci budou zpracovány doménové diagramy a diagramy aktivit pro jednotlivé navržené interakce.

Doménový diagram Zobrazuje jednotlivé třídy, ze kterých je interakce sestavena a vztahy mezi těmito třídami.

Diagram aktivit Ukazuje průběh jednotlivých interakcí. Vzhledem k tomu, že je cílová platforma virtuální realita, některé interakce vyžadují zapojení obou rukou (kontrolerů) během svého průběhu. Pro zpřehlednění diagramů budou ruce barevně rozlišeny. A to tedy primární ruka modrou barvou a sekundární barvou červenou.

4.4.1 Virtuální exponát

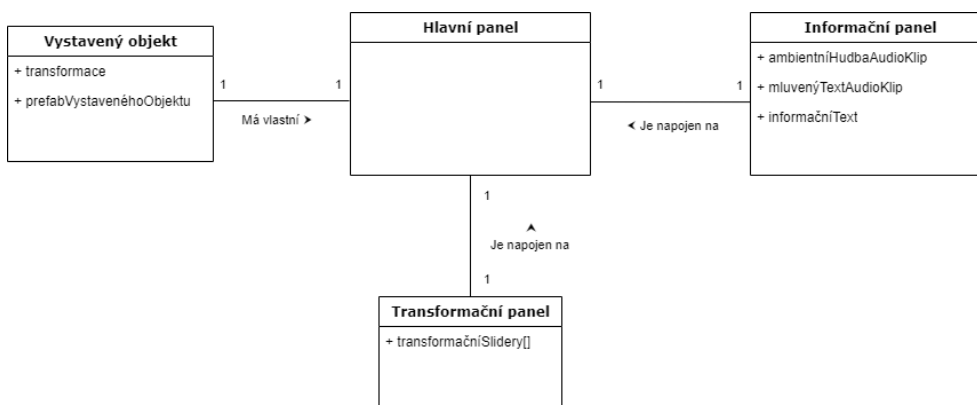
Vysvatený objekt Objekt reprezentující vizuální předlohu vystaveného modelu. Atribut `transformace` udává polohu a rotaci vystaveného modelu. Atribut `prefabVysvtenéhoObjektu` vzor vystaveného modelu.

4. NÁVRH

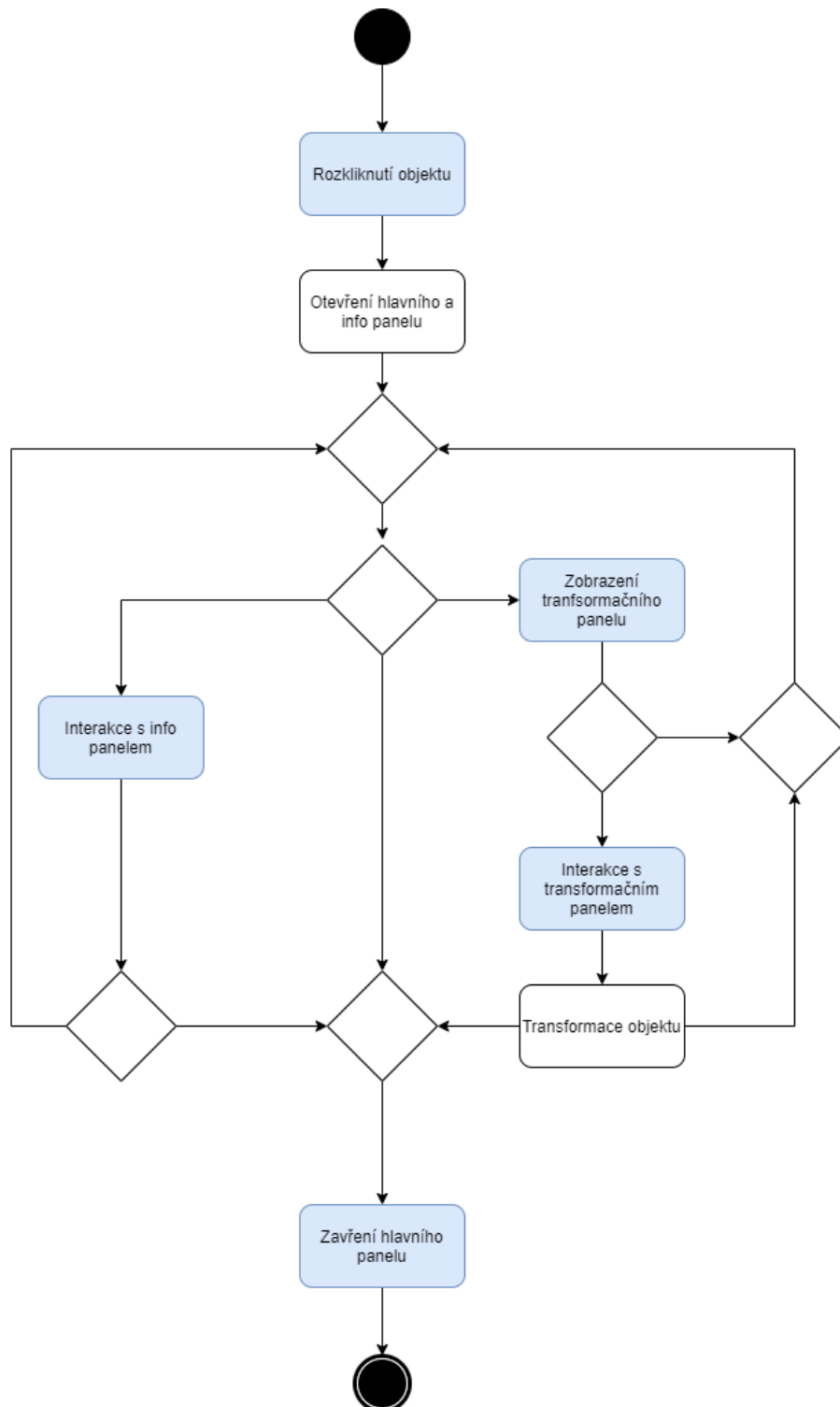
Hlavní panel Hlavní panel zastřešuje všechny ostatní panely. Jedná se o plochu na které se zobrazí transformační a informační panel.

Informační panel Panel obsahující hlavní text s informacemi o objektu. Atribut `informačníText` obsahuje info text, který se zobrazí na panelu. Atribut `ambientníHudbaAudioKlip` obsahuje audionahrávku s ambientní hudbou, která se spustí po otevření informačního panelu. Atribut `mluvenýTextAudioKlip` obsahuje audio nahrávku s namluveným info textem, která půjde přehrát pomocí interakcí s panelem.

Transformační panel Panel, pomocí kterého lze manipulovat s vystaveným objektem. Atribut `transformačníSlidery[]` obsahuje množinu sliderů pomocí, kterých se ovládá nastavení transformací aplikovaných na vystavený objekt.



Obrázek 4.4: Diagram aktivit pro informační panel



Obrázek 4.5: Diagram aktivit pro informační panel

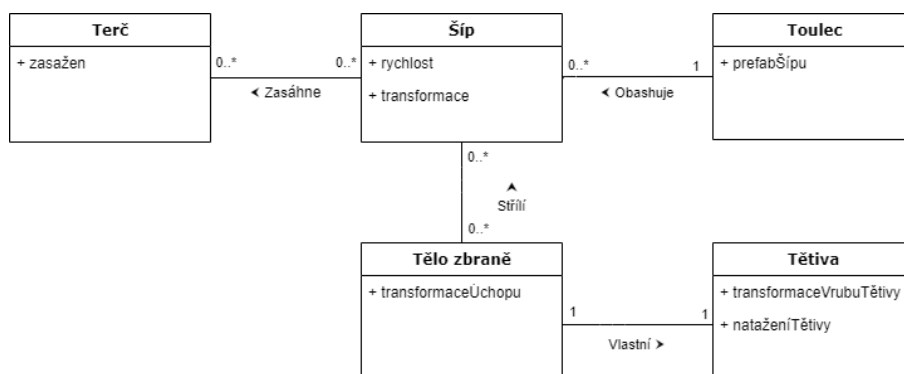
4.4.2 Simulace střelby ze středověkých zbraní

Tělo zbraně Představuje objekt s modelem těla zbraně. Atribut `transformaceÚchopu` určuje pozici úchopu zbraně.

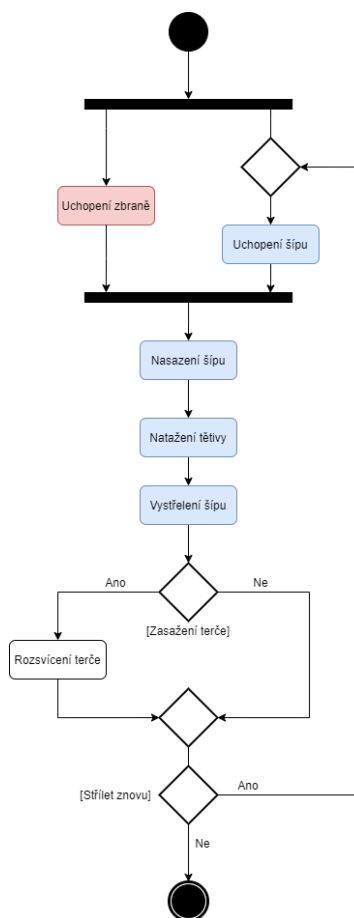
Tětiva Objekt předstávající tětivu zbraně. Atribut `natažení tětivy` určuje stav, ve kterém se momentálně tětiva nachází. Hodnoty, které může nabýt jsou omezeny od 0 do 1, kde 0 představuje tětivu v klidovém stavu a 1 maximální natažení tětivy. Atribut `transformaceVrubuTětivy` určuje pozici a rotaci při vložení šípu do vrubu. Tento atribut se mění v závislosti na změně míry natažení tětivy.

Toulec Objekt ze kterého uživatel bude získávat uživatel šípy. Atribut `prefabŠípu` představuje vzor šípu, které se budou tvořit při interakci s toulcem.

Šíp Objekt reprezentující projektil, který bude vystřelen ze zbraně. Atribut `rychlost` představuje jeho momentální rychlost poté co je vystřelen. Atribut `transformace` vlastní informace o natočení a pozici těla šípu.



Obrázek 4.6: Doménový diagram střelby ze středověkých zbraní



Obrázek 4.7: Doménový diagram aktivit střelby ze středověkých zbraní

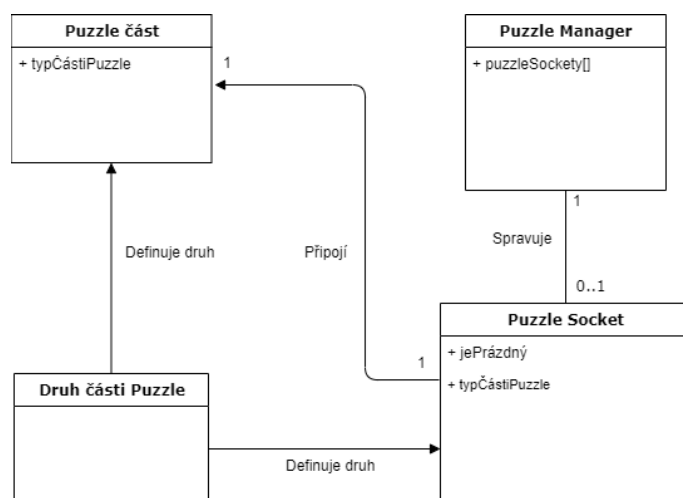
4.4.3 Skládací exponát

Puzzle Manager Má na starost správu všech socketů, které skládačka obsahuje. Kontroluje zda je skládačka celá složená. Po dokončení skládačky zničí všechny části vložené do socketů a vytvoří jeden objekt reprezentující poskládanou skládačku.

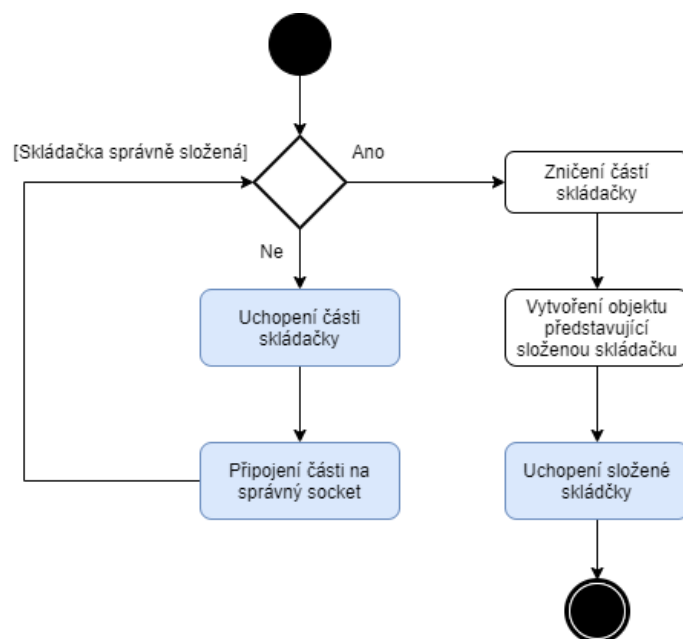
Puzzle část Představuje část nesložené skládačky. Atribut `typČástiPuzzle` definuje o jaký druh části skládačky patří se jedná.

Druh části Puzzle Určuje druh části skládačky.

Puzzle Socket Objekt do kterého se připojí část exponátu. Atribut `jePrázdný` určuje zda je socket zaplněný korektní částí skládačky. Atribut `typČástiPuzzle` definuje jaký druh části skládačky patří do tohoto socketu.



Obrázek 4.8: Doménový diagram skládačky



Obrázek 4.9: Aktivitní diagram skládačky

4.5 Použité technologie k vytvoření prototypu

Následující sekce obsahuje seznam softwaru a hardwaru, pomocí kterého bude výsledný prototyp implementován.

Unity [21] Navrhovaný prototyp bude implementován pomocí vývojového engine Unity, konkrétně verze 2019.4.f1. Jedná se o velmi rozšířený engine vhodný pro tvorbu aplikací pro jednotlivce a menší nezávislé studia. Zároveň poskytuje dobrou podporou pro vývoj VR aplikací.

UnityXR Interaction Toolkit [24] Prototyp aplikace bude implementován pomocí verze 1.0.0-pre.4 UnityXR Interaction Toolkitu.

JetBrains Rider [25] Je editor pro tvorbu aplikací převážně implementovaných v jazyce C# od společnosti JetBrains. Pro usnadnění implementace, vývojové prostředí nabídne dostatečné rozšíření pro vývoj v Unity.

Diagrams.net [36] Pro tvorbu návrhových diagramů bude použitý volně dostupný editor diagrams.net, také známý pod jménem Draw.io. Jedná se o nástroj, který umožňuje snadnou tvorbu diagramů.

HTC Vive [14] Pro vývoj a testování jsem vybral cílové zařízení HTC Vive. Důvodem byla dobrá kompatibilita s vývojovým prostředím, které jsem vybral společně s velmi dobrými hardwarovými parametry.

Blender [37] Je open-source software pro modelování a práci s 3D modely. Přestože je zcela zdarma, poskytuje uživateli možnosti na podobné úrovni jako profesionální placené editory. V rámci implementace prototypu bude potřeba provést jednoduché úpravy na některých modelech, případně vytvořit vlastní modely.

Implementace

5.1 Pohyb ve virtuálním prostoru

Do výsledného prototypu aplikace byly implementovány následující způsoby pohybu.

Teleportace Teleportaci uživatel spustí zmáčknutím *triggeru* na pravém ovladači a namířením mířícího paprsku na validní oblast. Zaměřovací paprsek jsem implementoval pomocí úpravy šablonového objektu *XRRayInteractor*, kterému jsem upravil vzhled a napojil na něj teleportační skript, který bude hýbat s virtuálním tělem uživatele. Vitální součástí teleportačního systému je skript *Teleportation Area*, který jsem připojil k objektů, které ve scéně slouží jako oblast vymezenou pro pohyb uživatele jako například podlaha.

Continuous movement Je metoda transformující vstupy zadané stiskem *Touchpadu* levého kontroleru na pohyb uživatele v prostoru. Při kalkulaci pohybu se bere v potaz úhel natočení kamery, aby docházelo ke korektnímu posunu v závislosti na směru pohledu uživatele. Zároveň se bere v potaz gravitace a zrychlení při vertikálním pohybu, například při skoku z výšky. Rychlost jakou se uživatel pohybuje ve virtuálním prostoru byla lehce snížena oproti základnímu nastavení, za účelem minimalizace potenciální možnosti výskytu nevolností při rychlém pohybu ve virtuálním prostředí zatímco je tělo uživatele ve statické póze.

Snap turn otáčení Pro otáčení do stran bez nutnosti se fyzicky otočit je použitý skript šablonový *Snap Turn Provider* s upravnými parametry. Tento způsob pohybu aktivuje stisknutím *Touchpadu* levého kontroleru.

Fyzický pohyb Fyzický pohyb uživatele v místnosti je snímán a posléze transformován do virtuální světa pomocí šablonového skriptu *XRRig*.

5.2 Manipulace s objekty

Veškeré interakce mezi *Interaktivními objekty* a *Interaktory* ve scéně jsou vyhodnoceny pomocí objektu *XRInteractionManager* a jeho skriptu *CustomInteractionManager*, který rozšiřuje šablonový *XRInteractionManager*. Do prototypu aplikace byly implementovány dva druhy interaktorů, které umožňují různé způsoby jak interagovat s interaktivním objektem.

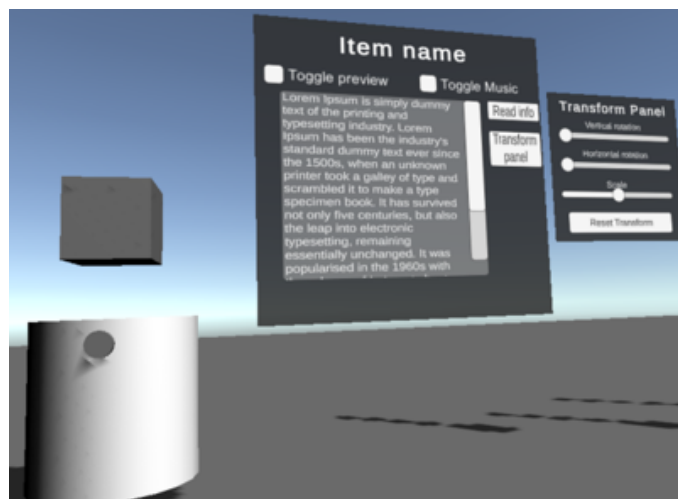
Interaktivní objekt Je objekt, se kterým lze interagovat pomocí interaktoru. Každý interaktivní objekt dědí ze skriptu *XRBaseInteractable*. Velká většina interaktivních objektů ve scéně obsahuje instanci třídy *XRGrabInteractable* nebo třídu, která z ní dědí. Tato třída rozšiřuje základní implementaci interaktivního objektu o možnost uchopit ho pomocí interaktoru. Varianta *interactable* skriptu, která stojí za zmínku je *XROffsetGrabInteractable*. Tato třída umožní u objektů, u kterých není definována specifická transformace úchopu, vytvořit dočasný bod úchopu v závislosti na pozici interaktoru na začátku interakce. Použitím *XROffsetGrabInteractable* vznikne u předmětů, které nemají předem určenou transformaci úchopného bodu, realističtější forma úchopu.

Interaktory Jsou objekty s připnutou komponentou *XRBaseInteractor*. V rámci prototypu je implementována interakce s objekty na dálku pomocí skriptu *XRRayInteractor*, který je připnutý na virtuální kontrolery. Tento skript při namíření na interaktivní objekt, zobrazí paprsek indikující možnost interakce s tímto objektem. Interakci na blízko obstarává skript *XRDirectInteractor* připnutý na kontrolery, který umožňuje interakci s objekty v přímém kontaktu s rukou uživatele.

XR Rig *XR Rig* je prefab virtuální těla uživatele ve scéně. Obashuje kameru a základní implementaci pro trackování rukou. Šablonový prefab *XR Rig* jsem rozšířil o modely simulující ruce, které přizpůsobují animaci podle toho, jaké tlačítka na kotroleru jsou zmáčknuty. Dále jsem obě ruce vybavil zmíněnými komponenty *XRDirectInteractor*, *XR RayInteractor* a pro pravý kontroler jsem přidal ještě *XR RayInteractor* modifikovaný pro účely vizualizaci míření při teleportu.

5.3 Implementace navržených interakcí

5.3.1 Virtuální exponát



Obrázek 5.1: Ilustrace implementované interakce virtuálního exponátu

Exponát Objekt vlastní fyzický model exponátu společně s komponentami *Audio Source*. Tyto komponenty obsahují jednotlivé audionahrávky jako ambientní hudba a namluvený text s informacemi o exponátu. *ItemController* obsahuje instanci *ItemTransformHandleru* a zároveň spouští nahrávky ze zmíněných *Audio Sourceů*.

```
// If ambient audio clip is not playing start play it
if (!isAmbientSoundPlaying)
{
    ambientSoundPlayer.loop = true;
    ambientSoundPlayer.volume = 0.3f;
    ambientSoundPlayer.Play();
}
else
{
    // Stop playing ambient audio clip
    ambientSoundPlayer.Stop();
}
```

Plátno Objekt, na který se vykreslují jednotlivé panely obsahující UI prvky. Obsahuje komponentu *Tracked Graphic Raycaster*, která umožní ovládat UI prvky obsažené na panelech pomocí raycastového interaktoru *XRRayInteractor*, který je připnutý k rukou uživatele. Kontrolu nad tím jaké panely budou

v daný moment aktivní je řízeno skriptem *ItemController*, který po zavolání určitých metod aktivuje a deaktivuje objekty jednotlivých panelů.

Hlavní panel Panel obsahující textové pole, které obsahuje informační text o exponátu. Na toto textové pole je napojený *Scrollbar*, kterým implementuje možnost vertikálně posouvat textem. Další součástí panelu je sada tlačítek. Například tlačítko pro spuštění audionahrávky s namluveným textem po zmáčknutí zavolá metodu *PlayTextInfo* skriptu *ItemController*.

Transformační panel Panel který obsahuje tři UI objekty typu *Slider* a tlačítko pro resetování modelu objektu do základního stavu. Pomocí těchto sliderů uživatel zadává parametry k jednotlivým transformacím, které se budou aplikovat na vystavený model. Zmíněné transformace jsou horizontální a vertikální rotace společně s zvětšením a zmenšením. Jakýkoliv zadaný vstup vytvoří event, na který čekají *EventListenery* ve skriptu *ItemTransformHandler*, která provede adekvátní transformaci na modelu exponátu.

Item Transform Handler Skript zastřešující provedení transformací na modelu exponátu. Obsahuje *EventListenery*, které naslouchají na event, který se zavolá při změně hodnoty v uživatelském rozhraní implementovaném v transformačním panelu. V ukázce kódu je ukázka implementace těchto *EventListenersů*.

```
public class DisplayItemTransformHandler : MonoBehaviour
{
    ...

    // Sliders
    public Slider sliderScale;
    public Slider sliderRotateX;
    public Slider sliderRotateY;
    public Slider sliderScale;

    ...

    /// <summary>
    /// Start is called before the first frame update
    /// </summary>
    void Start()
    {
        ...

        sliderScale.onValueChanged.AddListener(updateScale);
        sliderRotateX.onValueChanged.AddListener(updateRotationX);
    }
}
```

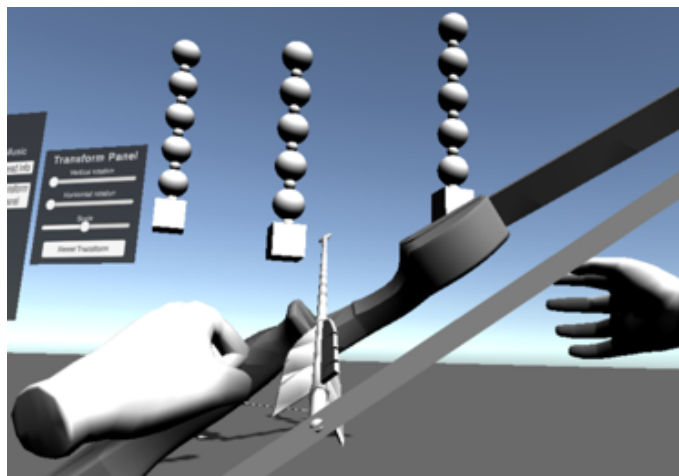
```

        sliderRotateY.onValueChanged.AddListener(updateRotationY);
    }
}

```

5.3.2 Simulace střelby ze středověkých zbraní

V rámci simulace střelby ze středověkých zbraní byla v prototypu implementována střelba z luku a arbalestu. Implementace této interakce byla inspirována projektem lukostřelby [38].



Obrázek 5.2: Ilustrace implementované interakce simulace střelby ze středověkých zbraní

5.3.2.1 Společné části

Šíp Každý šíp vlastní skript *Arrow* dědící z třídy *XRGrabInteractable* implementující základní chování uchopitelného objektu pomocí interaktoru. Přetížená metoda *OnSelectEntering* zkontroluje při pokusu o úchop šípu zda se jedná o interaktor třídy *XRDirectInteractor* a připraví vnitřní hodnoty potřebné pro nasazení a následné vystřelení šípu. Přetížená metoda *OnSelectExited* je zavolána když je šíp puštěn interaktorem. Metoda zkontroluje zda byl držen interaktorem typu *Notch* a v případě kladného vyhodnocení vystřelí šíp zavoláním metody *Launch*. Ta přidá tělu šípu kinematickou energii úměrné míře natažení tětiny.

Implementace kolize letícího šípu objekty je řešena následujícím způsobem. V přetížené metodě *ProcessInteractable*, za předpokladu, že je šíp vystřelen, je zkontrolována kolize šípu s libovolným objektem obsahujícím *Collider* a současně aktualizována poslední zaznamenaná pozice špičky šípu. Tuto pozici je použita v metodě *CheckCollision*, kde je zavolán *Physics.Linecast* s argu-

menty poslední známé pozice a aktuální pozice špičky šípu. Tento paprsek zaznamená případnou kolizi s objektem.

Toulec Obsahuje skript *Quiver*, který dědí ze třídy *XRBaseInteractable* implementující základní chování pro interakci s interaktory. Při uchopení toulce je vytvořena instance nového šíp objektu, který je automaticky zvednutý interaktorem, který toulec uchopil.

```
/// <summary>
/// When the interactor interacts with the quiver object, create
/// an arrow and force it into the hand
/// </summary>
/// <param name="args">SelectEnterEvent arguments describing the
/// select action</param>
private void CreateAndSelectArrow(SelectEnterEventArgs args)
{
    // Create arrow, force into interacting hand
    Arrow arrow = CreateArrow(args.interactor.transform);
    interactionManager.ForceSelect(args.interactor, arrow);
}

/// <summary>
/// Creates an instance of an arrow object in the scene
/// </summary>
/// <param name="orientation">Orientation of the spawned
/// arrow</param>
/// <returns>Newly created arrow object in the scene</returns>
private Arrow CreateArrow(Transform orientation)
{
    // Create arrow, and get arrow component
    GameObject arrowObject = Instantiate(arrowPrefab,
        orientation.position, orientation.rotation);
    return arrowObject.GetComponent<Arrow>();
}
```

Notch Je skript implementující základní funkcionalitu část tětivy, na kterou se nasadí šíp. Dědí ze třídy *XRSocketInteractor*. Přetížená metoda *CanSelect* zaručí, že do socketu, lze připojit pouze objekt šípu. V případě, že se šíp přiblíží k socketě je na šíp aplikována transformace, která ho nasedí na vrub tětivy. Při natažení tětivy se zavolá metoda *MoveAttach*, která posune nasazený šíp a upraví parametry pro vizualizaci tětivy. Vlastní instanci třídy, která počítá míru natažení *PullMeasurer*.

PullMeasurer Skript implementující výpočet a nastavení míry natažení tětivy. Pomocí metody *CalculatePosition* vypočítá míru natažení reprezentovanou hodnotou z intervalu 0 až 1, kde 0 je stav klidu a 1 maximální

natažení. Při změně míry natažení zavolá vytvoří event *Pulled* signalizující změnu stavu tětiny.

```
// Pull amount has changed
if (newPullAmount != pullAmount)
{
    // Set new pull value
    pullAmount = newPullAmount;
    // Invoke the pull event
    Pulled?.Invoke(newPullPosition, newPullAmount);
}
```

String renderer Objekt zajišťující vizuální reprezentaci tětiny. Vizualizace se provádí za pomoci přímkou, která je vytvořena přes komponentu *Line renderer*. Přímka je určena třemi body. *Start* začátek, *End* konec a *Middle* střed přímkou, který má stejnou pozici jako vrub tětiny. Změna vzhledu přímkou při natáhnutí se provede pomocí posunu bodu *Middle* a následného vykreslení v metodě *Update*.

5.3.2.2 Specifické části pro luk

Luk Hlavní část zbraně obsahující fyzický luku model. Na objekt tento je připojena komponenta se skriptem *Bow* dedící ze třídy *XRGrabInteractable*. Po zdvihnutí luku se ve skriptu *Bow* spustí *EventListenery*, pomocí kterých se aktivuje plná funkcionalita vrubu luku implementovaná v *BowNotch*.

Vrub luku Vrub luku je objekt umístěný na středu tětiny obsahující skript *BowNotch*. Ten dědí ze třídy *Notch* a rozšiřuje její funkcionalitu primárně o *EventListenery* naslouchající změnu stavu natažení tětiny a pokyn k vypstřelení ze zbraně. Zároveň také implementuje metodu *ReleaseArrow*, která se postará o vystřelení šípu.

```
// Only release if the target in the socket is an arrow and
// the pull amount is greater than set threshold
if (selectTarget is Arrow && PullMeasurer.PullAmount >
    releaseThreshold)
{
    // Release the arrow using custom deselect
    CustomManager.ForceDeselect(this);
}
```

PullMeasurerBow Skript implementující logiku výpočtu míry natažení tětiny luku. Dědí z mateřské třídy *PullMeasurer* a doplňuje jeho implementaci o výpočet míry natažení tětiny v závislosti na pozici natahující ruky.

5.3.2.3 Specifické části pro arbalest

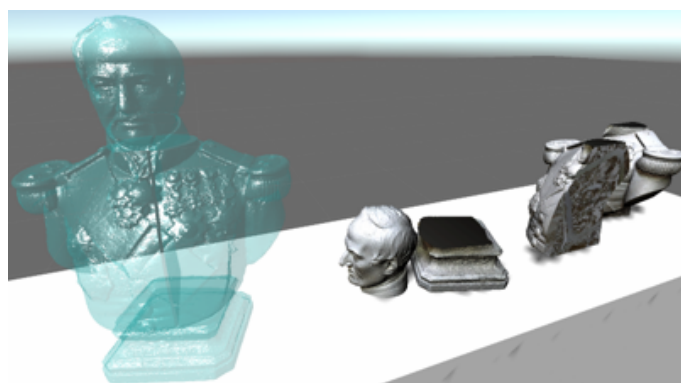
Arbalest Hlavní část zbraně obsahující fyzický arablestu model. Na objekt tento je připojená komponenta se skriptem *Crossbow* dedící ze třídy *XRGrabInteractable*. Po zdvihnutí těla se ve skriptu *Crossbow* spustí *EventListener*, pomocí kterých se plně aktivuje interaktor *CrossbowNotch* a umožní střelbu z arbalestu.

Vrub arbalestu Vrub arbalestu je objekt umístěný na středu tetivy obsahující skript *CrossbowNotch*. Ten který dědí ze třídy *Notch* a rozšiřuje její funkcionalitu primárně o *EventListener* naslouchající změnu stavu natažení tetivy arbalestu a pokyn k vystřelení ze zbraně. Zároveň také implementuje metodu *ReleaseCrossbowArrow*, která se postará o vystřelení šípu.

Natahovací mechanismus Objekt obsahující model nabíjecího mechanismu arbalestu. Jeho komponenta se skriptem *CrossbowPuller* implementuje přepočítání míry otočení nabíjecím mechanismem na hodnotu natažení tetivy. Aby s objektem mohl uživatel manipulovat, dědí zmíněný skript ze třídy *XRGrabInteractable*. Omezení pohybu na pouhou rotaci po ose Y je zajištěno pomocí komponenty *HingeJoint*.

Spoušť Je objekt, který při zmáčknutí vystřelí z nabitě zbraně. Je implementovaný pomocí rozšířené třídy *XRBaseInteractable*, která v metodě vyvolané *OnSelectEnteredIntekrací* se spouští zavolá metodu *ReleaseCrossbowArrow* z objektu.

5.3.3 Skládací exponát



Obrázek 5.3: Ilustrace implementované interakce skládací exponát

Kontroler skládačky Zastřešuje mechaniku skládačky a zajišťuje její úspěšné dokončení. Ve svých atributech vlastní všechny *PuzzleSocket*, ze kterých se

puzzle skláda. Metoda *FinishPuzzle* je zavolána v případě, že jsou všechny sockety skládačky korektně zaplněny a posléze zaručí zničení instancí objektů, které jednotlivé sockety drží a vytvoří instanci výsledného objektu představující model složené skládačky. Výsledný model má na sobě připnutou komponentu *XRGrabInteractable* zaručující možnost manipulovat s ním.

```
/// <summary>
/// Destroys separate parts of the puzzle currently placed in the
/// sockets and spawns the finished puzzle object into the scene
/// </summary>
private void FinishPuzzle()
{
    // Check all the sockets in the puzzle
    foreach (var socket in puzzleSockets)
    {
        // Destroy puzzle piece object inside the socket
        Destroy(socket.selectTarget.gameObject);
    }

    // Spawn an instance of finished puzzle object into the scene
    Instantiate(finishedPuzzlePrefab, finishedPuzzleTransform);
}
```

Část skládačky Každá část skládačky vlastní *PuzzlePiece* skript, který dědí z *XRGrabInteractable* implementující základní chování uchopitelného objektu. Tento skript definuje o jaký druh části skládačky se jedná a s jakým typem puzzle socket bude kompatibilní.

Puzzle Socket Socket do kterého uživatel vkládá části skládačky. Každý socket obsahuje skript *PuzzleSocket*, který dědí z třídy *XRSocketInteractor* definující základní chování socketu. Každý z socketů obsažených ve skládačce má definovaný svůj druh pomocí *PuzzlePieceType*. Do socketu určitého druhu lze připojit pouze část skládačky stejného druhu. V přetížených metodách *CanHover* a *CanSelect* se provádí kontrola zda je možné připojit část skládačky, kterou se uživatel snaží vložit do socketu.

```
/// <summary>
/// Checks whether the interactable can be selected into the socket
/// </summary>
/// <param name="interactable">Interactable object trying to
/// interact with the socket</param>
/// <returns>Bool value whether the interactable can be selected
/// into the socket</returns>
public override bool CanSelect(XRBaseInteractable interactable)
{
```

5. IMPLEMENTACE

```
        return base.CanSelect(interactable) &&
            isInteractableCorrectPieceType(interactable);
    }

    /// <summary>
    /// Checks whether the interactable can be hover in the socket
    /// </summary>
    /// <param name="interactable">Interactable object trying to
    /// interact with the socket</param>
    /// <returns>Bool value whether it can be hovered by
    /// interactable</returns>
    public override bool CanHover(XRBaseInteractable interactable)
    {
        return base.CanHover(interactable) &&
            isInteractableCorrectPieceType(interactable);
    }

    /// <summary>
    /// Checks if the interactable is trying to interact with socket
    /// is the same type as the socket
    /// </summary>
    /// <param name="interactable">Interactable object trying to
    /// interact with the socket</param>
    /// <returns>Bool value depending on whether they are the same
    /// type</returns>
    private bool isInteractableCorrectPieceType(XRBaseInteractable
        interactable)
    {
        if (interactable is PuzzlePiece interactablePiece)
        {
            if (pieceType == interactablePiece.type)
            {
                return true;
            }
        }
        return false;
    }
}
```

Testování

Testování implementovaného prototypu bude cíleno na jednotlivé interakce společně s obecnými prvky prototypu. Z výsledků provedených testů by mělo vyplynout, zda jsou jednotlivé interakce vhodně navržené.

6.1 Podkladové materiály pro testování

Účastníci testů obdrží k vyplnění dotazník společně s testovacími scénáři pro jednotlivé testy.

6.2 Průběh testování

Každý z účastníků obdrží dotazník a pokyny pro testera (viz Podkladové materiály pro testování). Testování proběhlo v soukromých prostorech. Vzhledem k omezené technické výbavě prostorů, kde testování proběhlo, průběh testování nebyl nahráván.

6.2.1 Dotazník

Každému účastníkovi testování bude přidělen výstupní dotazník. Výstupní dotazník poslouží jako zpětná vazba testera k návrhu a provedení prototypu. Obsahuje otázky s známkovým ohodnocením na škále 1 až 5 jako ve škole. Tedy 1 pro nejlepší a 5 pro nejhorší ohodnocení. Dotazník je rozdělen do dvou částí. První sekce obsahuje obecné otázky o virtuální realitě a jejím zapojení do klasických muzeí. Druhá sekce se skládá ze sady otázek vztahujících se k návrhu interakcí. Na každá interakci jsou tázány stejné dotazy.

Ke každé testované interakci bude zvlášť přidělen prostor pro nepovinný slovní komentář.

6. TESTOVÁNÍ

6.2.1.1 Sada obecných otázek pro testery

- Do jaké míry jste seznámeni s pojmem virtuální realita?
- Jak byste hodnotili Vaše osobní zkušenosti s virtuální realitou? (1 - Mám hodně zkušeností s použitím virtuální reality, 5 - Virtuální realitu jsem nikdy osobně nezkoušel)
- Je podle Vás vhodný nápad využít virtuální realitu v klasických muzeích.
- V případě, že byste byli na prohlídce v muzeu, kde by byla začleněná do prohlídky virtuální realita. Vyzkoušeli byste si ji?

6.2.1.2 Sada otázek pro testery pro jednotlivé interakce

- Je podle Vás tato interakce vhodná pro obohacení prohlídky muzea?
- Je podle Vás podobný druh interakce vhodný do prohlídky muzea?
- Jak hodnotíte uživatelské rozhraní interakce?
- Hodí se tato interakce pro virtuální realitu?
- Jaký máte celkový pocit z interakce?

6.2.2 Testovací scénáře

Testeři obdrží testovací scénáře. Každá interakce má vlastní testovací scénář.

6.2.2.1 Pohyb a ovládání

Odhadovaný čas

3 minuty.

Účel testování

Tento test má za úkol seznámit uživatele s implementovanými způsoby pohybu ve virtuální scéně a zhodnotit je.

Výchozí bod

Uživatel si spuštění aplikace nasadí headset a uchopí do rukou ovladače.

Koncový bod

Uživatel je seznámen s ovládáním pohybu a je schopen se pohybovat po scéně.

Instrukce pro testera

Nasaďte si headset virtuální reality a vezměte do rukou ovladače. Interakcí

s *touchpadem* levého kontroleru vyzkoušejte plynulý pohyb po scéně. Pomocí pravého *trigger* tlačítka vyzkoušejte teleportaci.

Očekávané kroky

- Nasazení headsetu a uchopení ovladačů.
- Dotek *touchpadu* na levém ovladači a pohyb v prostoru.
- Stisknutí *trigger* tlačítka na pravém ovladači a teleportování se po scéně.
- Plynulý pohyb testera po scéně.

6.2.2.2 Skládací exponát

Odhadovaný čas

3 minuty

Účel scénáře

Test interakce skládací exponát a schopnost uživatele dokončit skládačku.

Výchozí bod

Uživatel stojí ve scéně. Jednotlivé části exponátu rozmístěny po scéně.

Koncový bod

Úspěšně složené složení exponátu a jeho uhcopení.

Instrukce pro testera

Najděte jednotlivé části skládačky a uchopte je pomocí *grab* tlačítka na libovolné ruce a umístěte je na správné místo. Po dokončení skládačky uchopte objekt pomocí *grab* tlačítka na libovolné ruce.

Očekávané kroky

- Uživatel začne prozkoumávat scénu a hledat části skládačky.
- Uživatel najde a uchopí část skládačky pomocí *grab* tlačítka na libovolné ruce.
- Vložení uchopené části na správnou pozici.
- Opakování dvou předchozích bodů dokud úspěšně neumístí všechny části.
- Uchopení složené skládačky.

6.2.2.3 Virtuální exponát

Odhadovaný čas

3 minuty

Účel scénáře

Test interakce virtuální exponát. Rozkliknutí info panelu, spuštění audionahrávky.

Výchozí bod

Uživatel stojí ve scéně a před ním je umístěn interaktivní exponát.

Koncový bod

Úspěšné spuštění audionahrávky.

Instrukce pro testera

Rozkličte tlačítko na podstavci exponátu pomocí *grab* tlačítka na liboblné ruce. Spustěte audionahrávku pomocí rozkliknutím tlačítka na informačním panelu pomocí *trigger* tlačítka na pravé ruce.

Očekávané kroky

- Uživatel stiskne tlačítko na podstavci.
- Spuštění audionahrávky pomocí *trigger* tlačítka na pravé ruce.

6.2.2.4 Simulace střelby ze středověkých zbraní

Odhadovaný čas

3 minuty

Účel scénáře

Test simulace střelby ze středověkých zbraní. Střelba z luku.

Výchozí bod

Uživatel stojí ve scéně a před ním je umístěn luk a toulec.

Koncový bod

Uživatel úspěšně vystřelí šíp z luku.

Instrukce pro testera

Uchopte jednou rukou pomocí *grab* tlačítka luk a druhou stejným způsobem vyndejte šíp z toulce. Nasad'te šín na vrub luku. Natáhněte tětívu a vystřelte šíp.

Očekávané kroky

- Uživatel uchopí luk pomocí *grab* tlačítka na libovolném kontroleru.
- Druhou rukou šáhne do toulce pro šíp.
- Nasazení šípu na tětivu luku.
- Natáhní tětivu luku.
- Vystřelení šípu.

6.3 Výsledky testování

Implementovaný prototyp byl otestován za pomoci zařízení HTC Vive. Testování se zúčastnili dva testeři. Oba testeři neměli žádnou předešlou zkušenost s virtuální realitou.

I přes objevení několika menších chyb, nebyly během testování nebyly odhaleny žádné zásadní problémy, které by znemožnily splnit některý z testovacích scénářů. V testovacích dotaznících byly zaznamenány názory a připomínky obou testerů. Hodnocení jednotlivých interakcí bylo pozitivní a myšlenka rozšíření prohlídky muzea pomocí virtuální reality testery velmi zaujala .

6.3.1 Shrnutí výsledků testování

Problémy s ovládáním Tester musel být s ovládáním předem nebo za běhu test obeznámen. Vzhledem k cílové skupině tohoto prototypu by bylo vhodné přidat do scény návod s popisem ovládání.

Uchopení objektů Obecně neměli testeři s úchopem problém. Byla ovšem zaznamenána připomínka ohledně k nekomfortního způsobu úchopu objektu, kdy uživatel musí držet tlačítka úchopu po celou dobu manipulace s objektem. Tento problém by bylo možné vyřešit změnou logiky úchopu objektu, kde by při první zmáčknutí došlo pro zvednutí a opětovném zmáčknutí by došlo k jeho upuštění.

Pohyb po scéně Přestože účastníci testování neměli žádné předchozí zkušenosti s technologií virtuální reality, nedocházelo při pohybu po scéně k výskytu nevolností. Pro pohyb mezi jednotlivými interakcemi preferovali použití teleportu. Při testování kombinace metod pohybu byl odhalena chyba. Po použití teleportu a následnému pohybu po scéně pomocí *Continuous movement* zůstal objekt určený k vizualizaci teleportu ve scéně. V rámci této práce se nepodařilo odhalit příčinu této chyby.

Interakce - Skládací exponát Systém skládačky nedělal testerům problém. Při testování této interakce došlo k výskytu nevyžádaného chování úchopu částí skládačky. Aby uživatel udržel úchop na objektu, je nutné, aby nepřetržitě držel stisknuté tlačítko *grab*. Tento fakt zapříčinil neustálou aktivitu *RayInteraktoru* na ruce, ve které objekt drží. V okamžiku namíření tohoto interaktoru na jinou část skládačky došlo k nevyžádanému úchopu.

Interakce - Simulace střelby ze středověkých zbraní Celkový dojem z interakce byl pozitivní. Drobnější komplikace nastaly při úchopu šípu z toulce, kdy testerům chvíli trvalo, než byli schopni uchopit toulec za správnou část. Tuto komplikaci připisují použitému modelu pro toulec. Ostatní části střelby proběhly hladce vzhledem k intuitivnímu návrhu uživatelského rozhraní interakce. Problém s toulcem by vyřešila lepší volba jeho modelu.

Interakce - Interaktivní exponát Interakce s interaktivním exponátem proběhla také bez větších komplikací. Z uživatelských testů vyplynul návrh pro možné vylepšení funkcionality bylo přidání možnosti pozastavit audio-nahrávku s nahraným textem.

Závěr

Tato práce se zabývala interakcí ve virtuální realitě použitelných pro prohlídky v muzeích. Čtenář byl seznámen s technologií virtuální reality a jejími typy krátkým úvodem do virtuální reality.

Byla provedena analýza dostupných typů hardwarových zařízení určených pro zprostředkování virtuální reality. Podrobně bylo popsáno zařízení HTC Vive, které zároveň sloužilo pro vývoj a testování výstupní implementace této práce. Implementace prototypu byla zadána pomocí engine Unity, jehož základní vlastnosti byly popsány. V analýze byly dále prozkoumány dostupné nástroje pro tvorbu aplikací pro virtuální realitu. Z těchto nástrojů byl zvolen Unity XR Interaction Toolkit, jehož vlastnosti byly hlouběji popsány. Na základě analýzy aplikací zabývajících se problematikou a zvážením cílové skupiny byl sestaven návrh prototypu.

Návrh obsahuje různé metody pohybu a tři typy interakcí určených pro obohacení prohlídky muzea. Pro návrh interakcí byl využit model funkčních a nefunkčních požadavků. Na základě těchto požadavků byly vytvořeny diagramy aktivit a doménové diagramy, které sloužily jako podklad pro implementační část práce.

Pro implementaci byl použit engine Unity. Výstupem implementace je funkční prototyp demonstrující navržené interakce a metody pohybu. S ohledem na cílovou skupinu jsou interakce nekomplikované a způsob pohybu je přizpůsobený uživatelům bez jakékoliv zkušenosti s virtuální realitou. Proces vývoje aplikace pro virtuální realitu je velmi citlivý vzhledem k práci s velmi rychle se vyvíjející technologií. Během implementace jsem narazil na problémy s některými funkcionalitami napříč různými verzemi vývojového balíčku. Tyto komplikace společně s nedostatkem zkušeností s některými aspekty Unity, vývojovým balíčkem jako takovým a omezenými časovými prostředky zapříčinily relativní jednoduchost některých interakcí výsledné implementace.

Prototyp aplikace byl podroben uživatelským testům, které se skládaly z dotazníku a testovacími scénáři. Zpětná vazba testování ukázala, že interakce jsou vhodné pro obohacení prohlídky muzeí, ale zároveň poukázala na

ZÁVĚR

nedostatky v implementaci.

Stanovené cíle této práce považuji za splněné. Věřím, že tato práce může být cenným výchozím bodem pro práce zabývající virtuální realitu.

Literatura

- [1] Marr, B.: Extended reality. 08 2019, [cit. 21. června 2021]. Dostupné z: <https://www.forbes.com/sites/bernardmarr/2019/08/12/what-is-extended-reality-technology-a-simple-explanation-for-anyone/?sh=56424b817249>
- [2] Pauš, P.: [přednáška, online], 2019, [cit. 21. června 2021]. Dostupné z: https://courses.fit.cvut.cz/NI-PVR/files/10_ar.pdf
- [3] Niantic, I.: Pokémon Go. [software], 06 2016, [cit. 21. června 2021]. Dostupné z: <https://pokemongolive.com/en/>
- [4] BrandonBray; Coulter, D.: What is Mixed Reality. 08 2020, [cit. 21. června 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>
- [5] Microsoft: HoloLens 2 - features and technical specs. 2021, [cit. 21. června 2021]. Dostupné z: <https://www.microsoft.com/en-us/hololens/hardware>
- [6] Iberdrola: Virtual reality: another world within sight. 08 2020, [cit. 21. června 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>
- [7] Ecosystem, T.: Virtual, mixed and augmented reality. [online], 03 2019, [cit. 21. června 2021]. Dostupné z: <https://blog.ecosystem360.com/forget-ar-vr-its-game-mixed-reality-now/>
- [8] Anthes, C.; García Hernandez, R.; Wiedemann, M.; aj.: State of the Art of Virtual Reality Technologies. 03 2016, ISBN 978-1-4673-7676-1, doi:10.1109/AERO.2016.7500674, [cit. 21. června 2021]. Dostupné z: https://www.researchgate.net/publication/297760223_State_of_the_Art_of_Virtual_Reality_Technologies

- [9] Sutrich, N.: Playstation VR2: Everything we know so far. 03 2021, [cit. 21. června 2021]. Dostupné z: <https://www.androidcentral.com/psvr-2>
- [10] Delfino, D.: What is Google Cardboard? [online], 01 2019, [cit. 21. června 2021]. Dostupné z: <https://www.businessinsider.com/what-is-google-cardboard>
- [11] Alza.cz: Co jsou to haptická odezva? [online], 06 2021, [cit. 21. června 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality>
- [12] XinReality: Oculus Quest 2. 03 2021, [cit. 21. června 2021]. Dostupné z: https://xinreality.com/wiki/Oculus_Quest_2
- [13] Coulter, D.; wguyman: Windows MR Motion Controllers. 03 2018, [cit. 21. června 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/design/motion-controllers>
- [14] XinReality: HTC Vive. 03 2021, [cit. 21. června 2021]. Dostupné z: https://xinreality.com/wiki/HTC_Vive
- [15] Valve: ValveIndex. [online], 06 2021, [cit. 21. června 2021]. Dostupné z: <https://store.steampowered.com/sub/354231/>
- [16] NicePNG: Virtual reality controllers. [online], 04 2021, [cit. 21. června 2021]. Dostupné z: <https://www.nicepng.com/maxp/u2w7a9e6y3a9u2o0/>
- [17] EVETECH: HTC Vive. [online], 08 2016, [cit. 21. června 2021]. Dostupné z: <https://www.evetech.co.za/repository/ProductImages/htc-vive-vr-headset-1000px-v1-0004.jpg>
- [18] NicePNG: HTC Vive controller. [online], 09 2016, [cit. 21. června 2021]. Dostupné z: https://www.nicepng.com/ourpic/u2w7a9e6t4o0o0q8_htc-vive-controls-htc-vive-controller-guide/
- [19] Blog, C.: Unity vs Unreal Engine: which game engine is for you. 07 2020, [cit. 21. června 2021]. Dostupné z: <https://www.creativebloq.com/advice/unity-vs-unreal-engine-which-game-engine-is-for-you>
- [20] Technologies, U.: Unity. [online], 2021, [cit. 21. června 2021]. Dostupné z: <https://unity.com/>
- [21] Games, E.: Unreal Engine. [software], 2021, [cit. 21. června 2021]. Dostupné z: <https://www.unrealengine.com/en-US/>
- [22] Microsoft: Windows Mixed Reality Toolkit. [software], 2021, [cit. 21. června 2021]. Dostupné z: <https://github.com/microsoft/MixedRealityToolkit-Unity>

-
- [23] Corporation, V.: SteamVR. [software], 03 2021, [cit. 21. června 2021]. Dostupné z: <https://store.steampowered.com/app/250820/SteamVR/>
- [24] Technologies, U.: XR Interaction Toolkit. [software], 2021, [cit. 21. června 2021]. Dostupné z: `UnityPackageManager`
- [25] Brains, J.: Jet Brains Rider. [software], 2021, [cit. 21. června 2021]. Dostupné z: <https://www.jetbrains.com/rider/>
- [26] Technologies, U.: Introduction to components. [online], 2021, [cit. 21. června 2021]. Dostupné z: <https://docs.unity3d.com/Manual/Components.html>
- [27] Technologies, U.: Introduction to components. [online], 2021, [cit. 21. června 2021]. Dostupné z: <https://docs.unity3d.com/Manual/ScriptingSection.html>
- [28] Technologies, U.: Unity - Manual:XR. 2021, [cit. 21. června 2021]. Dostupné z: <https://docs.unity3d.com/Manual/XR.html>
- [29] Fuad, M.; Mahendran, A.: Unity XR Tech stack. [online], 01 2020, [cit. 21. června 2021]. Dostupné z: <https://blogs.unity3d.com/2020/01/24/unity-xr-platform-updates/>
- [30] Technologies, U.: XR Interaction Toolkit. [online], 2021, [cit. 21. června 2021]. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0/manual/index.html>
- [31] VR, L.: Remembering Pearl Harbor. [Software], 16 2017, [cit. 21. června 2021]. Dostupné z: <https://www.viveport.com/987f821a-346f-4fa3-b101-e44bb6b6ed2c>
- [32] VR, L.: Remembering Pearl Harbor. [online], 16 2017, [cit. 21. června 2021]. Dostupné z: <https://www.viveport.com/987f821a-346f-4fa3-b101-e44bb6b6ed2c>
- [33] FarBridge: Masterworks: Journey Through History. [online], 16 2018, [cit. 21. června 2021]. Dostupné z: <https://farbridge.medium.com/masterworks-journey-through-history-a-new-vr-experience-for-oculus-rift-and-samsung-gear-vr-52e61c784ba>
- [34] Studios, W.: Kingdom Come: Deliverance. [software], 13 2018, [cit. 21. června 2021]. Dostupné z: <https://www.kingdomcomerpg.com/>
- [35] RedBull.cz: Kingdom Come: Deliverance. [online], 12 2016, [cit. 21. června 2021]. Dostupné z: <https://www.redbull.com/cz-cs/kingdom-come-deliverance-dan-vavra-rozhovor-rpg-stredovek-cechy>

LITERATURA

- [36] Diagrams.net: Diagrams.net. [Software], 05 2021, [cit. 21. června 2021]. Dostupné z: <https://www.diagrams.net/>
- [37] Blender.org: Blender. [Software], 05 2021, [cit. 21. června 2021]. Dostupné z: <https://www.blender.org/>
- [38] with Andrew, V.: Bow and Arrow for Unity XR(Update). [Software], 03 2021, [cit. 21. června 2021]. Dostupné z: <https://www.youtube.com/watch?v=H0xTz4JtWiI>
- [39] wareable.com: Outside-in tracking. [online], 4 2017, [cit. 21. června 2021]. Dostupné z: <https://www.wareable.com/vr/inside-out-vs-outside-in-vr-tracking-343>
- [40] Valve: Steam. [Software], 05 2021, [cit. 21. června 2021]. Dostupné z: <https://store.steampowered.com/>

Seznam použitých zkratek

- XR** Extended reality
- AR** Augmented reality (rozšířená realita)
- MR** Mixed reality (mixovaná realita)
- VR** Virtual reality (virtuální realita)
- 3D** Three-dimensional (třírozměrný)
- HMD** Head-mounted display
- MRTK** Mixed Reality Toolkit
- OLED** Organic light-emitting diode
- SDK** Software development kit
- UE** Unreal Engine
- PC** Personal computer
- WMR** Windows Mixed Reality

Návod a sada doporučení pro provoz zařízení HTC Vive v prostoru muzea

B.1 Sada doporučení pro provoz zařízení virtuální reality v prostoru muzea

B.1.1 Vyhrazený prostor

Pro docílení co nejlepšího zážitku z virtuální reality, je nutné vyhradit pro provoz zařízení dostatečný prostor. Minimálními požadavky kladené na velikost prostoru jsou $2\text{ m} \times 1,5\text{ m}$, ale za ideálních podmínek by mělo mít stanoviště virtuální reality vyhrazený prostor $5\text{ m} \times 5\text{ m}$. Vyhrazený prostor by měl být výrazně ozařený a za provozu by do něj měl povolený vstup pouze jeden člověk, který bude mít nasazený.

B.1.2 Zařízení s infračerveným zářením

Zaměřovací stanice používají pro sledování pohybu headsetu s kontrolery po vyhrazeném prostoru infračervené záření, které vysílají do tohoto prostoru vysílají B.1. Vzhledem k charakteru jakým stanice vysílají toto infračervené záření, je nutné dát si pozor na ostatní zařízení v místnosti, kde bude virtuální realita umístěna. Běžný provoz některých zařízení (například bezpečnostní čidla) by mohl být tímto zářením narušen.

B.1.3 Zrcadla

Je důležité se vyvarovat umístění virtuální reality do místnosti, která obsahuje jedno nebo více zrcadel. Tyto zrcadla by totiž mohly narušit korektní fungování trackovacího systému zařízení HTC Vive.

B.1.4 Hygienická opatření

Krom klasických hygienických návyků, které by měly být v dnešní době normou, je důležité vyzdvihnout podstatu důkladného čištění jednotlivých částí headsetu. Nejdůležitější částí, která vyžaduje největší péči je headset, kde je potřeba pravidelně po každém použití čist vnitřní vystýlku společně s zobrazovacími čočkami.



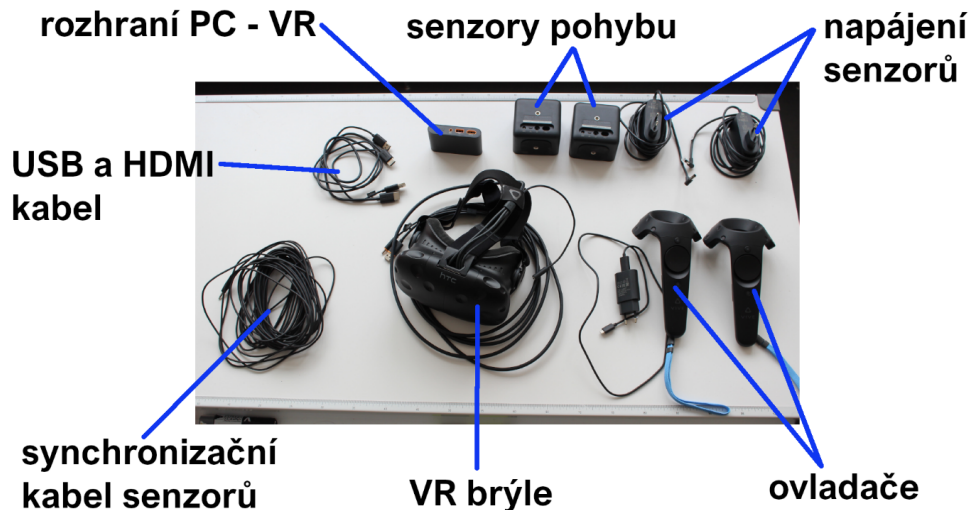
Obrázek B.1: Ilustrace vycházejícího infračerveného záření vycházejícího ze stanic [39]

B.2 Zařízení HTC Vive a jeho komponenty

Tato podsekcce se bude věnovat výčtu a popisu jednotlivých komponentů, ze kterých se skládá HTC Vive, cílové zařízení virtuální reality pro tuto práci (obrázek B.2).

- **VR brýle**, které si uživatel nasadí na hlavu. Jejich úkolem je sledování pohybu hlavy a promítání vizuálů uživateli.
- **Dva ovladače** pro záznam pohybu rukou a sledování vstupů zadaných uživatelem.
- **Senzory pohybu** snímající vymezenou plochu vyhrazenou pro umístění stanice virtuální reality.
- **Kabel pro napájení senzorů**
- **Synchronizační kabel senzorů** slouží pro komunikaci mezi stanicemi senzorů pohybu v nebezdrátové konfiguraci.

- **Rozhraní PC - VR** je zařízení, které působí jako most v komunikaci mezi headsetem a počítačem.
- **USB a HDMI kabel** na propojení počítače a headsetu přes rozhraní.



Obrázek B.2: Popsané komponenty sady HTC Vive

B.3 Návod na zapojení HTC Vive

Následující sekce obsahuje návod na zapojení zařízení pro virtuální realitu HTC Vive. Před začátkem prosím zkontrolujte zda Váš počítač splňuje minimální doporučené požadavky pro podporu zařízení HTC Vive. Dále je nutné si pořídit účet Steam a nainstalovat si aplikaci Steam [40].

B.3.1 Zapojení a rozmístění senzorů pohybu

- Zapojte napájecí kabely do obou stanic snímacích senzorů a oba kabely zapojte do elektřiny (obrázek B.3).
- Pro rozmístění senzorů jsou k dispozici dvě možnosti konfigurace. Pokud na sebe senzory neustále „uvidí“, tedy jsou například zavěšeny u stropu, spojí se bezdrátově a není třeba je propojovat synchronizačním kabelem. Pokud tuto možnost nemáte, tedy senzory nebudou v dostatečné výšce a mohlo by dojít k zastínění, je třeba je projít synchronizačním kabelem (obrázek B.4).

B. NÁVOD A SADA DOPORUČENÍ PRO PROVOZ ZAŘÍZENÍ HTC VIVE V PROSTORU MUZEA



Obrázek B.3: Zapojení napájecích kabelů do snímacích stanic



Obrázek B.4: Snímací stanice propojené synchronizačním kabele

- Pro zvolení zvoleného synchronizačního módu je nutné přepnout nastavení na synchronizačních stanicích. Stanice zobrazují aktuální nastavení pomocí zeleného písmene umístěného na přední straně. Zmáčknutím fyzického tlačítka na zadní straně změníte konfiguraci stanice.
 - Pro bezdrátovou konfiguraci musí na jedné stanici svítit *b* a na druhé *c* (obrázek B.5).

- V případě synchronizace přes kabel nastavte senzory na písmena *A* a *b* (obrázek B.6).



Obrázek B.5: Konfigurace stanic pro bezdrátovou



Obrázek B.6: Konfigurace stanic pro kabelovou synchronizaci

B.3.2 Zapojení rozhraní PC - VR

- Připravte si zařízení pro propojení headsetu virtuální reality a počítače (obrázek B.7).

B. NÁVOD A SADA DOPORUČENÍ PRO PROVOZ ZAŘÍZENÍ HTC VIVE V PROSTORU MUZEA



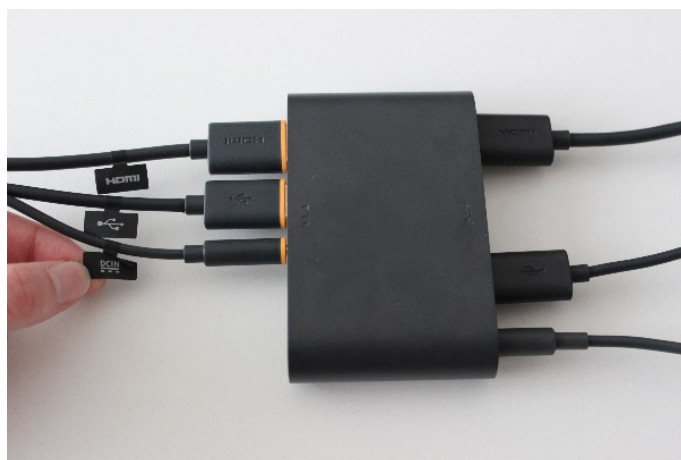
Obrázek B.7: Zařízení pro propojení headsetu a počítače

- Do oranžových portů zapojte kabely vedoucí z headsetu. Kabely vedoucí z headsetu (HDMI, USB a napájecí) jsou také označeny oranžovou barvou (obrázek B.8).



Obrázek B.8: Zapojení kabelů headsetu do PC - VR rozhraní.

- Z druhé strany rozhraní zapojte kabely s černými koncovkami. HDMI a USB kabel připojte k počítači, napájecí kabel zapojte do zásuvky (obrázek B.9).



Obrázek B.9: Kompletace zapojení PC - VR rozhraní

- Nyní zapněte headset virtuální reality pomocí tlačítka na levé straně (obrázek B.10).



Obrázek B.10: Zapínací tlačítko headsetu

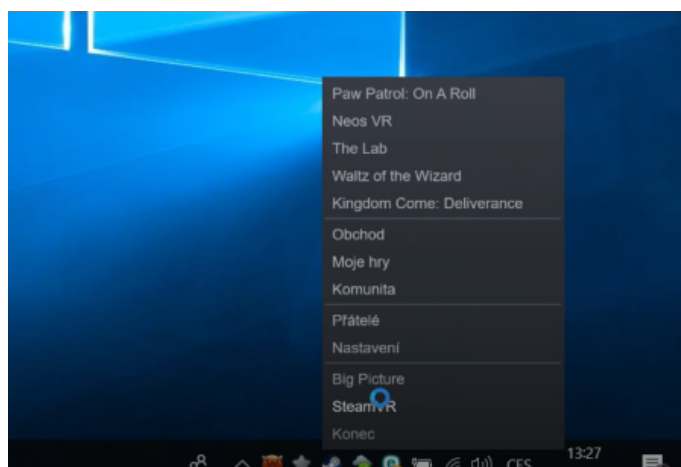
- Vzdálenost čoček upravíte pomocí kalibračního tlačítka na pravé spodní části brýlí (obrázek B.11). Bližší informace o nastavení čoček naleznete v oficiálním manuálu přibaleného k zařízení.



Obrázek B.11: Tlačítko na úpravu vzdálenosti čoček

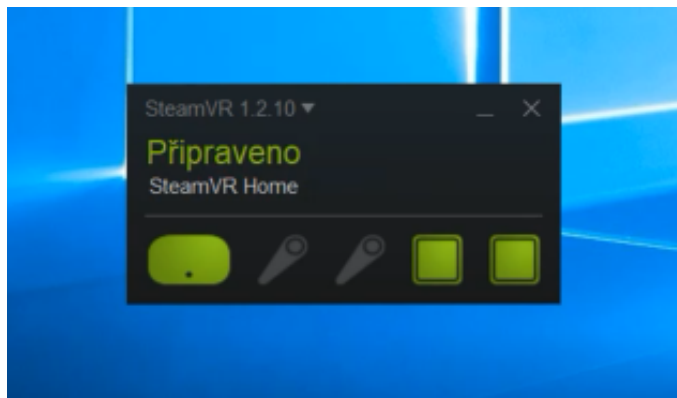
B.3.3 Aplikace Steam VR a příprava místnosti

- Otevřete aplikaci *Steam* [40] a spusťte *Steam VR* [23] (obrázek B.12).



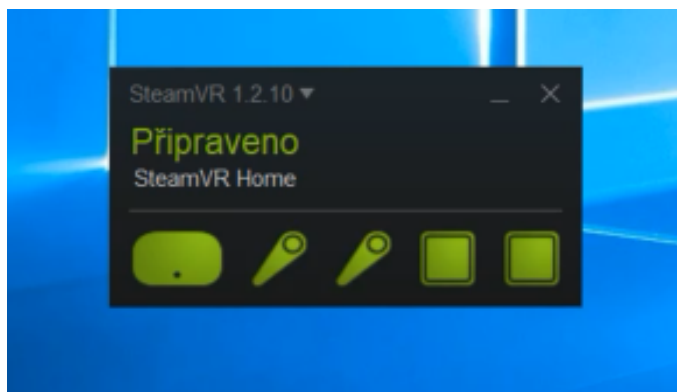
Obrázek B.12: Spuštění programu Steam VR přes aplikaci Steam

- Na obrazovce se objeví okno zobrazující status připojených komponent virtuální reality. Headset a snímací stanice by měli být již zapojeny a svítit v programu zeleně (obrázek B.13). V případě, že některé se zmíněných zařízení hlásí problém či nesvítí zeleně, zopakujte kroky z bodu *Zapojení a rozmístění senzorů pohybu*.



Obrázek B.13: Program *Steam VR* zobrazující aktuální status zapojení virtuálního headsetu.

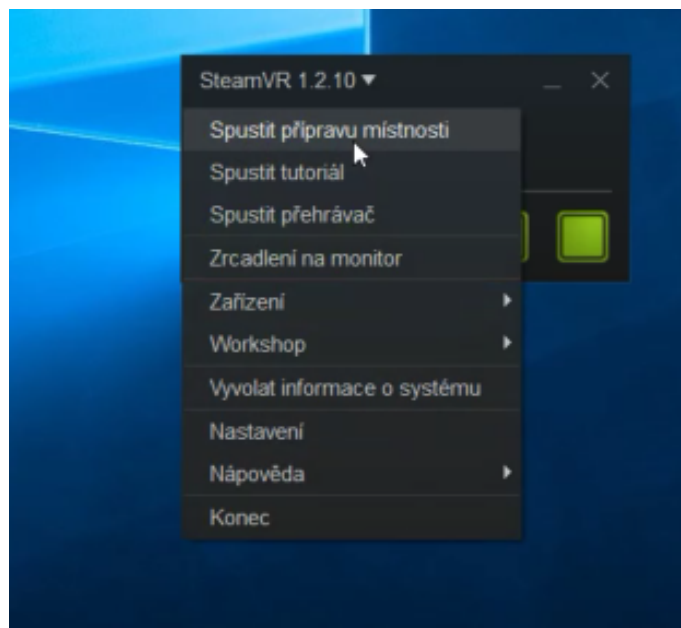
- Ovladače zapnete pomocí *Home* tlačítka u místěného na jejich horní části. Po úspěšném zapnutí se rozsvítí dioda na ovladači a v aplikaci *Steam VR* zezelánají jejich ikonky (obrázek B.14).



Obrázek B.14: Program *Steam VR* zobrazující aktuální status zapojení virtuálního headsetu.

- Nastavení místnosti se spustí pomocí funkce aplikace *Steam VR* „**Spustit přípravu místnosti**“ (obrázek B.15). Dalšími kroky budete provedeni návodem, který se objeví na obrazovce po spuštění přípravy místnosti. Tento návod vás provede vymezením prostoru a dalšími nezbytnými kalibracemi. Minimální požadavky na vymezený prostor je 2 m × 1,5 m. Pokud tímto prostorem nedisponujete, zvolte při kalibraci *Seated mode*.

B. NÁVOD A SADA DOPORUČENÍ PRO PROVOZ ZAŘÍZENÍ HTC VIVE V PROSTORU MUZEA



Obrázek B.15: Spuštění přípravy místnosti v aplikaci *Steam VR*

- Nyní je zařízení HTC Vive připraveno. Zapnutím aplikace pro virtuální realitu se obraz automaticky přesměruje do headsetu.
- V případě výskytu dalších problémů, které nepůjdou vyřešit zmíněným postupem, nahlédněte do oficiálního manuálu zařízení HTC Vive.

Obsah přiloženého CD

| | | |
|--|------------------|---|
| | readme.txt | stručný popis obsahu CD |
| | exe | adresář se spustitelnou formou implementace |
| | src | |
| | impl | zdrojové kódy implementace |
| | thesis | zdrojová forma práce ve formátu \LaTeX |
| | text | text práce |
| | thesis.pdf | text práce ve formátu PDF |