



## Zadání bakalářské práce

<b>Název:</b>	Věnná města českých královen - Interakce v historickém městě ve virtuální realitě
<b>Student:</b>	Klára Matoušková
<b>Vedoucí:</b>	Ing. Jiří Chludil
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Webové a softwarové inženýrství, zaměření Počítačová grafika
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	do konce letního semestru 2022/2023

### Pokyny pro vypracování

Věnná města českých královen je projekt zabývající se zobrazením historického prostředí ve virtuální realitě.

1. Analyzujte podobné projekty a aplikace, které se danému tématu věnují.
2. Analyzujte systémy virtuální reality a nástroje pro vývoj ve virtuální realitě v Unity3D.
3. Pomocí metod softwarových inženýrství navrhnete sady vhodných interakcí s historickými nástroji v historickém městě se zaměřením na uživatelskou skupinu široké veřejnosti.
4. Implementujte prototyp aplikace s navrženými interakcemi v Unity3D.
5. Podrobně a vyhodnoťte prototyp vhodným uživatelským testům.





**FAKULTA  
INFORMAČNÍCH  
TECHNOLÓGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Interakce v historickém městě ve virtuální realitě**

*Klára Matoušková*

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Chludil

11. května 2021



---

## Poděkování

Ráda bych poděkovala mému vedoucímu Ing. Jiřímu Chludilovi za odborné vedení a konzultace k této práci. Dále děkuji své rodině za podporu během mého studia.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) 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 11. května 2021

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2021 Klára Matoušková. 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**

Matoušková, Klára. *Interakce v historickém městě ve virtuální realitě*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.



---

# Abstrakt

Tato práce se zabývá analýzou a návrhem historických interakcí ve virtuální realitě. V práci analyzuji hardwarové zařízení virtuální reality. Zmíním různé typy výstupních a vstupních zařízení včetně podrobnější analýzy HTC Vive, které slouží jako vývojové prostředí této práce. V další části se zaměřím na balíčky pro vývoj virtuální reality v herním enginu Unity, neboť pomocí tohoto enginu bude implementován výstup této práce. Celkově bylo analyzováno pět balíčků. Tyto balíčky byly porovnány na základě jejich kompatibilit, funkcionalit, kladů a záporů. Shrnutí těchto informací pomohlo k výběru vhodného balíčku pro implementační část práce. Tímto balíčkem byl Unity XR Interaction Toolkit, který splňoval všechny podmínky a očekávání pro vývoj funkcionalit ve virtuální realitě. V další části se zaměřím na analýzu historických interakcí. Z aplikací byly vybrány interakce, které měly zajímavé provedení a potenciálně by sloužily jako podklad pro návrhovou část práce. Interakce byly detailně popsány a porovnány na základě jejich provedení, kladů a záporů. Toto porovnání pomohlo k tvorbě návrhové části práce, jelikož dokázalo poskytnout informace o několika variacích a problémech interakcí. Na základě této analýzy byl vytvořen návrh interakcí pro historické prostředí. Návrh obsahoval podrobný popis provedení interakcí včetně jejího průběhu a ovládání. Celkově bylo navrženo pět interakcí (lukostřelba, kování, broušení zbraně, oprava kola a pochodně). Tyto interakce jsou implementovány v prototypu, který je výstupem této práce. Interakce byly navrženy a implementovány tak, aby měly vhodné uživatelské rozhraní pro skupinu široké veřejnosti. Zároveň nabízí zajímavé a poučující provedení, které demonstruje interaktivní prostředí virtuální reality. Prototyp slouží jako základ a nabízí spoustu možností budoucího vývoje pro rozvoj těchto interakcí v historickém prostředí.

**Klíčová slova** virtuální realita, historické prostředí, interakce ve virtuální realitě, Unity, XR Interaction Toolkit

---

# Abstract

This thesis focuses on analysis and design of interactions in historical environment in virtual reality. The thesis contains the analysis of hardware of virtual reality devices which mentions various input and output devices. I will focus on device HTC Vive, which will be the part of the virtual reality development. Next section contains the analysis of frameworks for virtual reality development in game engine Unity. The result of this thesis will be implemented using this game engine. There were five frameworks analysed. These frameworks were compared based on their compatibility, functionality, pros and cons. Summary of this information helped with choosing the correct framework for development. Unity XR Interaction Toolkit was the choice of this framework since it fulfilled all expectations and requirements. In next section I will focus on the analysis of historical interactions. Interesting interactions were chosen from the applications. Interactions were described and compared based on their design, pros and cons. This comparison helped with the creation of the design part of this thesis as it provided useful information about the problems and design of interactions. The design of interactions in historical environment is based on this analysis and contains detailed description of interactions including the controls. The design contains five interactions (archery, blacksmithing, grindstone, repair of a wheel and torches). These interactions were implemented in a prototype, which is the result of this thesis. Interactions were designed and implemented in order that they provide an appropriate user interface for the general public. At the same time they provide an interesting and instructive design that demonstrates an interactive environment for virtual reality. The prototype serves as a basis and provides a lot of opportunities of future development.

**Keywords** virtual reality, historical environment, interaction in virtual reality, Unity, XR Interaction Toolkit

---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Úvod do virtuální reality</b>	<b>5</b>
2.1 Smíšená realita . . . . .	6
2.2 Mixovaná realita . . . . .	6
2.3 Rozšířená realita . . . . .	7
<b>3 Zařízení virtuální reality</b>	<b>9</b>
3.1 Hardware . . . . .	9
3.2 HTC Vive . . . . .	12
<b>4 Unity3D</b>	<b>15</b>
4.1 Komponenty . . . . .	15
4.2 Skriptování . . . . .	15
4.3 Nástroje pro vývoj virtuální reality . . . . .	16
<b>5 Základní metody interakcí ve virtuální realitě</b>	<b>23</b>
5.1 Selektce . . . . .	23
5.2 Manipulace . . . . .	24
5.3 Indikátory . . . . .	25
<b>6 Analýza aplikací</b>	<b>27</b>
6.1 Shadow Legend VR . . . . .	27
6.2 A Township Tale . . . . .	29
6.3 Blade and Sorcery . . . . .	31
6.4 Kingdom Come: Deliverance . . . . .	32
<b>7 Shrnutí analýzy</b>	<b>35</b>

<b>8</b>	<b>Návrh</b>	<b>37</b>
8.1	Model požadavků . . . . .	37
8.2	Model domény a aktivit . . . . .	42
8.3	Použité technologie k vytvoření prototypu . . . . .	49
<b>9</b>	<b>Implementace prototypu</b>	<b>51</b>
9.1	Projekt . . . . .	51
9.2	Scéna . . . . .	52
9.3	Integrace Unity XR ITK . . . . .	52
9.4	Interakce . . . . .	53
9.5	Implementační zajímavosti . . . . .	62
9.6	Ovládání . . . . .	64
<b>10</b>	<b>Uživatelské testování</b>	<b>65</b>
10.1	Popis testování . . . . .	65
10.2	Podkladové materiály . . . . .	65
10.3	Výsledky testování . . . . .	67
	<b>Závěr</b>	<b>71</b>
	<b>Literatura</b>	<b>73</b>
<b>A</b>	<b>Testovací scénáře</b>	<b>79</b>
<b>B</b>	<b>Seznam použitých zkratk</b>	<b>85</b>
<b>C</b>	<b>Obsah příloženého média</b>	<b>87</b>

---

## Seznam obrázků

2.1	Virtuální realita . . . . .	5
2.2	Směšená realita . . . . .	6
2.3	Mixovaná realita . . . . .	7
3.1	Ovladače . . . . .	10
3.2	Vive Tracker . . . . .	11
3.3	HTC Vive . . . . .	12
3.4	Chaperone systém . . . . .	13
4.1	Unity XR . . . . .	16
4.2	InteractionManager . . . . .	17
4.3	BaseInteractor . . . . .	17
4.4	BaseInteractable . . . . .	18
4.5	Mixed Reality Toolkit . . . . .	20
5.1	Selekce . . . . .	24
5.2	Škádování tahem ruky . . . . .	25
5.3	Škádování pomocí fyzického/virtuálního ovladače . . . . .	25
5.4	Indikátor objektu . . . . .	25
6.1	Shadow Legend VR . . . . .	28
6.2	A Township Tale . . . . .	30
6.3	Blade and Sorcery . . . . .	32
6.4	Kingdom Come: Deliverance . . . . .	34
8.1	Obecné požadavky na prototyp . . . . .	37
8.2	Požadavky pro jednotlivé interakce . . . . .	39
8.3	Lukostřelba - doménový diagram . . . . .	43
8.4	Lukostřelba - diagram aktivit . . . . .	43
8.5	Kování - doménový diagram . . . . .	44
8.6	Kování - diagram aktivit . . . . .	45

8.7	Broušení - doménový diagram . . . . .	46
8.8	Broušení - diagram aktivit . . . . .	46
8.9	Oprava kola u vozíku - doménový diagram . . . . .	47
8.10	Oprava kola u vozíku - diagram aktivit . . . . .	47
8.11	Pochodně - doménový diagram . . . . .	48
8.12	Pochodně - diagram aktivit . . . . .	48
9.1	Scéna prototypu . . . . .	52
9.2	Sockety . . . . .	54
9.3	Objekty lukostřelby . . . . .	54
9.4	Lukostřelba . . . . .	55
9.5	Objekty kování . . . . .	56
9.6	Kování . . . . .	57
9.7	Objekty broušení . . . . .	58
9.8	Broušení . . . . .	59
9.9	Objekty opravy kola . . . . .	59
9.10	Oprava kola u vozíku . . . . .	60
9.11	Objekty pochodně . . . . .	60
9.12	Pochodně . . . . .	61
9.13	Shader Graph . . . . .	62
9.14	Shader Graph materiál . . . . .	62
9.15	Blend Tree . . . . .	63
9.16	Animace luku . . . . .	63
9.17	Ovládání prototypu . . . . .	64

---

## Seznam tabulek

3.1	Variace HTC Vive . . . . .	14
7.1	Podporované platformy VR balíčků Unity . . . . .	35
7.2	Srovnání balíčků VR v Unity . . . . .	35
10.1	Odpovědi části dotazníku . . . . .	68





---

# Úvod

V dnešní době se setkáváme s pojmem *virtuální realita* (VR) zcela běžně a díky pokročilému hardwarovému zařízení je možné ponořit se do virtuálního světa. Pomocí této technologie se vyvíjí mnoho projektů nejen pro herní průmysl. *Věnná města českých královen* (VMČK) je spolupráce mezi pracovníky a studenty ČVUT FIT, UHK FF (*Univerzita Hradec Králové - Filozofická fakulta*), HU AV (*Historický ústav Akademie věd České republiky*) a dalších institucí. Cílem této spolupráce je zrekonstruovat historické okolí několika věnných měst, kde uživatel bude mít možnost interakce s historickými artefakty. Projekt je zaměřen především na práci s technologií virtuální reality a smíšené reality ve formě mobilního klienta.

Tato práce se bude věnovat interakcím s historickými nástroji v historickém prostředí ve virtuální realitě. Výstupy v rámci této spolupráce nejsou jednoznačně určeny pro historické pracovníky a studenty se zaměřením na tuto činnost, ale všeobecně pro širokou veřejnost.

Motivací k výběru tohoto tématu je práce s technologií virtuální reality, a předchozí spolupráce v rámci projektu VMČK. Práce s historickými artefakty za pomoci virtuální reality je zajímavé téma a nabízí spoustu možností budoucího vývoje.



---

## Cíl práce

Hlavním cílem této práce je demonstrovat interakce ve virtuální realitě. Součástí tohoto cíle je prozkoumání interakcí se zaměřením na historické prostředí a analyzování nástrojů pro vývoj virtuální reality. Zaměřím se na zařízení HTC Vive a herní engine Unity, kde porovnáím možnosti vývoje virtuální reality v tomto enginu.

Dalším cílem je návrh vybraných interakcí. Návrh bude vycházet z předchozí analýzy interakcí. Interakce budou navrženy pro historické prostředí ve virtuální realitě.

Na základě návrhu bude implementován prototyp aplikace, který bude sloužit jako základ pro budoucí projekty VMČK. Prototyp bude implementován v enginu Unity za pomoci vybraného nástroje a musí podporovat zařízení HTC Vive.

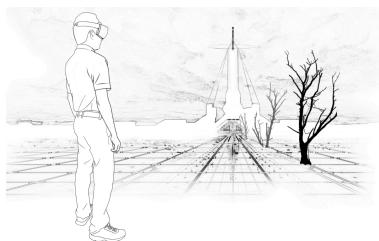
Posledním cílem je uživatelské testování, které bude sloužit jako zpětná vazba k návrhu a implementaci prototypu.



## Úvod do virtuální reality

Virtuální realita (*virtual reality*, nebo VR) je technologie, která umožňuje plně se ponořit do simulovaného virtuálního 3D světa. Celé odvětví této technologie se po své druhé vlně, která přišla v roce 2012 s revolučním zařízením *Oculus Rift*, vyvíjí rapidně a neustále hledá nové možnosti zlepšení kvality virtuálního zážitku [1]. Od této doby se do vývoje zapojilo značné množství známých značek (HTC, Sony, Oculus, Microsoft atd.), aby také probořili svými (i netradičními) technologiemi a hardwarem do světa virtuální reality.

Samostatný pojem *virtuální realita* byl poprvé popularizován roku 1987 *Jaronem Lanierem*, počítačovým informatikem vědcem a zakladatelem firmy VPL Research [2]. V dnešní době se virtuální realita využívá z velké části v zábavním průmyslu, ale existují i edukativní aplikace. Dle předpovídaných statistik má virtuální realita potenciál, že najde své využití i v odvětvích lékařství, vzdělání, sportu, kultury, cestování nebo i v rámci armádních zařízení [3]. Příkladem využití virtuální reality jsou herní aplikace (*Beat Saber* [4]), cestovatelské aplikace (*Google Earth VR* [5]), nebo virtuální prohlídky muzeí (*Věnná města českých královen* [6]). Vizualizace virtuální reality lze vidět na obrázku 2.1.

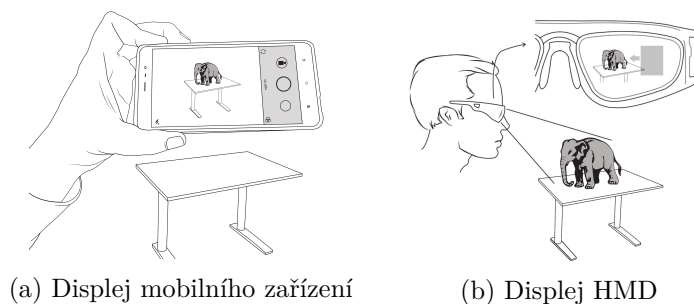


Obrázek 2.1: Virtuální realita [7]

## 2.1 Smíšená realita

Jedna z technologií blízké virtuální realitě je smíšená realita (*augmented reality*, neboli AR) [8]. Tato technologie obohacuje reálný svět, který je zobrazen prostřednictvím digitální kamery. Jedná se o kombinaci virtuálního a reálného obrazu. Základem je konstantní udržování pozice kamery a přepočítávání souřadnic objektu ve světě, neboli zkráceně - *trackování*.

Zobrazení smíšené reality je možné pomocí displeje (např. tablet, mobilní zařízení), které zobrazí digitální prvky do reálného obrazu v tomto zařízení (obrázek 2.2a). Druhá varianta je HMD zařízení, které promítá digitální prvek přímo na displej HMD (obrázek 2.2b). Smíšenou realitu lze vyzkoušet typicky pomocí mobilního zařízení nebo tabletu s digitální kamerou. Příkladem využití smíšené reality jsou herní aplikace (*Pokémon GO* [9]), interaktivní muzea, nebo aplikace zobrazující modely budov přímo v prostředí ulic města (*Věnná města českých královen* [6]).

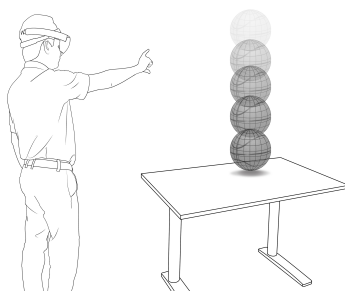


Obrázek 2.2: Smíšená realita [7]

## 2.2 Mixovaná realita

Mixovaná realita (*mixed reality*, nebo MR) je spojení reálného a virtuálního světa dohromady. Pojem označuje hybrid, nebo také střed mezi těmito světy [10]. Účelem je projekce virtuálních elementů do reálného světa. Další charakteristikou je možnost přímé manipulace a interakce s virtuálními objekty, které jsou umístěné uživatelem do reálného světa (obrázek 2.3).

Přestože je mixovaná realita stále v raném vývoji, tak v dnešní době existují technologie, který tento koncept uskutečnily a stále ho vyvíjí. *Windows Mixed Reality* (WMR) je součást systému Windows 10, která nabízí technologie VR, AR a MR [11]. Příkladem zařízení jsou *Microsoft HoloLens* a *Microsoft HoloLens 2* [12]. Interakce pomocí těchto zařízení je přímá a uskutečněna pouze pomocí rukou a očí. Microsoft představil toto zařízení společně s námětem *hologramů* v mixované realitě [13]. Jedná se o formu částečně průhledného displeje, který dokáže tyto hologramy integrovat do reálného světa.



Obrázek 2.3: Mixovaná realita [7]

### 2.3 Rozšířená realita

Všechny tyto zmíněné technologie (VR, AR a MR) jsou zahrnuty do jednoho společného pojmu rozšířená realita (*extended reality*, nebo XR) [14]. Písmeno X reprezentuje proměnou, od které se odvádí zkratky těchto technologií.





## Zařízení virtuální reality

V této sekci budu analyzovat různá hardwarová zařízení pro virtuální realitu. Zaměřím se na zařízení HTC Vive, jelikož se bude jednat o mé testovací prostředí implementovaného prototypu. Dále analyzuji další zařízení, které lze pro virtuální zážitky využít.

### 3.1 Hardware

Hardware je možné rozdělit do skupin. V zásadě se dělí na výstupní zařízení (*output devices*) a vstupní zařízení (*input devices*), zkráceně I/O zařízení. Existují i takové, které je možné zařadit do obou skupin zároveň, neboli *hybridní zařízení*. V podstatě se jedná o typické rozdělení všeobecného hardwarového zařízení, jako je např. počítačová myš - vstupní zařízení, počítačový monitor - výstupní zařízení. Bylo navrženo a realizováno již několik různých hardware zařízení pro VR, proto je také mnohé publikace srovnávají a dokonce systematicky rozdělují do grafických stromů a tabulek [15].

#### 3.1.1 Výstupní zařízení

**HMD displeje** Do výstupních zařízení se řadí displeje pro přenesení vizualizace. Co se týče zařízení virtuální reality, jedná se především o již zmiňované HMD. Zařízení HMD se dá rozdělit dle připojení na: mobilní, počítačové, samostatně fungující a konzolové [16].

*Mobilní HMD* je nejspíše finančně nejprínosnější variantou ze všech uvedených rozdělení. Tato varianta nabízí virtuální realitu bez připojení k PC, nebo k jakémukoliv dalšímu dodatečnému zařízení. Jedná se totiž o „držák“ pro chytré mobilní zařízení (smartphone), jehož obrazovka nahradí vestavěný displej. Příkladem je Google Cardboard nebo Samsung Gear VR.

*Počítačové HMD* je vyspělejší technologie než mobilní HMD. Tyto HMD vyžadují připojení k PC. Různé počítačové HMD se liší svými parametry. Z těchto parametrů nás nejčastěji zajímá zorné pole, stupeň volnosti, obnovovací

### 3. ZAŘÍZENÍ VIRTUÁLNÍ REALITY

frekvence, rozlišení displeje a typ displeje. Příkladem těchto HMD jsou již zmiňované zařízení Oculus Rift nebo HTC Vive.

*Samostatně fungující HMD* vypadají na první pohled jako počítačové HMD, avšak nevyžadují žádné dodatečné připojení k PC. Není tedy třeba řešit hardware připojovacího zařízení, n rozdíl od počítačových či mobilních HMD. Příkladem je *Oculus Quest 2*.

*Konzolové HMD* vyžadují herní konzoli, ke které se HMD připojí. S tímto typem HMD přišla pouze společnost Sony nabízející *PlayStation VR*. Díky tomu, že se jedná o konzolové připojení, není také třeba řešit hardwarové požadavky.

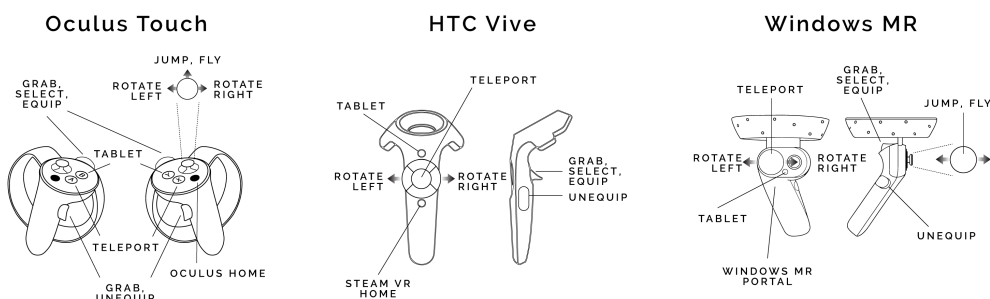
**Haptické displeje** Haptické technologie (*haptics*) nenabízí vizuální požitek, ale jsou pocíteny lidskou kůží. Může se jednat o pocit dotyku, teploty, síly, tahu atd. V dnešní době je nejčastěji ztvárněna prostřednictvím výstupního zařízení ve formě vibrací [17].

Většina ovladačů je vstupním zařízením, ale zároveň i výstupním zařízením (ačkoli to není jejich hlavní účel). Tyto ovladače mají schopnost vyvolávat haptické odezvy. Pomocí ovladače jsou přenášeny vibrace, nebo jiné pocity vyvolávající hmatový smysl, přímo do rukou.

Méně známé varianty jsou rukavice nebo obleky, které také dokážou vyvolávat haptickou odezvu (změna teploty, vibrace, pocit dotyku). Příkladem takového obleku je *TeslaSuit*, který má 64 haptických bodů se schopností vyvolat různé pocity na kůži nebo svalové kontrakce pomocí elektrické svalové stimulace. Zároveň slouží jako trackování celého těla [18].

#### 3.1.2 Vstupní zařízení

**Ovladače** Klasickým vstupním zařízením jsou ovladače. Ovladače jsou typicky dva, každý pro jednu ruku. Design ovladačů se od jednotlivých výrobců překvapivě hodně liší (obrázek 3.1). Ovladače neposílají jen informace o stlačení tlačítka, ale i o poloze a rotaci, protože jsou konstantně trackovány.

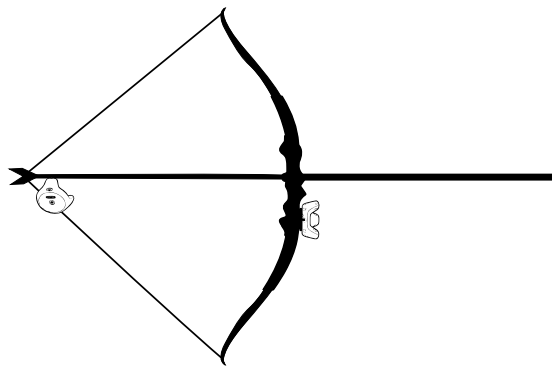


Obrázek 3.1: Ovladače Oculus Touch, HTC Vive a WMR [19]

**Eye-tracker** Eye-tracker slouží k sledování pohybu očí. V zásadě funguje na principu hledání vzdálenosti mezi středem zornicí a odrazem rohovky [20]. Technologie eye-trackingu je v dnešní době dokonce zabudována do některých počítačových HMD, ale pokud eye-tracker není zabudován, lze ho stále pořídit jako samostatný hardware a připojit ho k HMD. Eye-tracker byl poprvé zabudován do zařízení HMD prostřednictvím *HTC Vive Pro Eye* [21].

**Hand-tracker** Leap motion controller je technologie na podobné bázi jako eye-tracker, avšak místo očí plně trackuje ruce. K ovládání není zapotřebí ovladač, tedy interakce s digitálním světem jsou přirozené. Tato technologie plně trackuje obě ruce. Trackovací zařízení je možno připevnit například k zařízení Oculus Rift nebo HTC Vive [22].

**Vive Tracker** Vive Tracker je trackovací zařízení, které lze použít jako jednoduchý trackovací bod. Slouží například jako metoda trackování nohou. K tomu postačí upevnit jedno zařízení na každou nohu, aby se animace pohybu zobrazila i na virtuálním avataru. Tato forma trackování je univerzální, lze ji tedy použít víceméně kdekoliv. Skvělým příkladem využití zařízení Vive Trackerů je konstrukce luku, která umožňuje lepší proniknutí do virtuálního světa. Konstrukce je složená ze dvou trackovacích zařízení Vive Tracker, jeden upevněný pod úchytem luku a druhý na speciální konstrukci simulovaného šípu [23] (obrázek 3.2).



Obrázek 3.2: Konstrukce luku pro HTC Vive s využitím zařízení Vive Tracker [24]

## 3.2 HTC Vive

Původní *HTC Vive* vyšel v roce 2016 jako kolaborace mezi Valve a HTC a byl jeden z prvních počítačových HMD, které byly na trhu k dispozici [25]. Počítačová HMD zařízení vyžadují dobré minimální požadavky komponent PC, obzvláště CPU a GPU. Proto je dobré zkontrolovat minimální/doporučené požadavky, které zařízení HTC Vive vyžaduje [26].

HMD je připojeno přes rozhraní HDMI a USB k PC, a napájecím adaptérem do sítě. V základní verzi jsou k dispozici dva ovladače HTC Vive a dvě snímací stanice. Bezdrátové ovladače mají unikátní design a několik typů ovládacích prvků (obrázek 3.1). Pro lepší zážitek ve VR je možné použít i další trackovací zařízení, jako např. Vive trackery, jejich využití lze vidět na obrázku 3.2. HMD je vybaveno vlastním mikrofonom a jackem pro sluchátka. Základní sada HTC Vive je zobrazena na obrázku 3.3.



Obrázek 3.3: Základní HTC Vive sada [27]

### 3.2.1 Displej

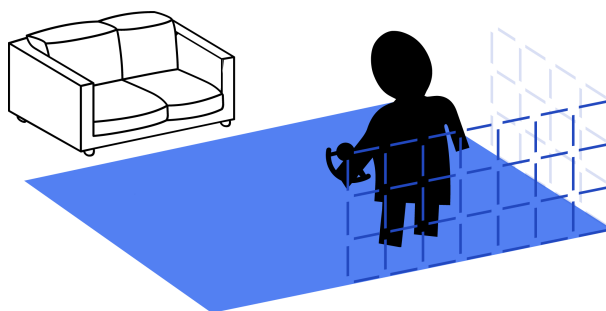
V dnešní době se rozlišují pro HMD VR zařízení dva hlavní typy displejů *OLED* a *LCD*. HTC Vive obsahuje dvojitý displej typu *OLED* s rozlišením  $1080 \times 1200$  pixelů s obnovovací frekvencí 90 Hz na oko a technologii Fresnelových čoček. Vzdálenost čoček by měla odpovídat hodnotě vzdálenosti zornic (*interpupilární distance*, neboli *IPD*). HMD umožňuje nastavení *IPD* pomocným otočným regulátorem na své pravé straně.

**Zorné pole** Dvojice čoček a displejů určuje zorné pole (*field of view*, nebo *FOV*). Lidské oko má zorný úhel zhruba  $150^\circ$  ve vertikální i horizontální rovině, avšak kombinovaný horizontální pohled odpovídá zornému úhlu přibližně  $120^\circ$  [17]. Tedy lidské zorné pole pro zdravé oči je přibližně  $150^\circ \times 120^\circ$ . Oficiálně je maximální *FOV*  $110^\circ$  horizontálně i vertikálně.

### 3.2.2 Trackovací metody

HTC Vive využívá metody *outside-in* trackingu. Neboli *lighthouse trackování* pomocí snímacích stanic (*Base Stations*) rozmístěných v místnosti [28]. Lighthouse trackování bylo vyvíjeno od Valve pro SteamVR a HTC Vive. Díky této metodě je možné docílit trackování plochy v místnosti až na  $5\text{ m} \times 5\text{ m}$  (čtyři Base Stations 2.0 až  $10\text{ m} \times 10\text{ m}$ ). Je nutné podotknout, že snímací stanice nesmí být zastíněné, aby trackovaly i sebe vzájemně. Proto se typicky rozmísťují dvě stanice do protějších rohů, aby bylo zajištěno snímání 3D prostoru. Pro lepší kvalitu trackování je možné využít i více stanic. SteamVR tímto docílil prvního trackingu, který nabízel tzv. *room-scale* (ve velikosti místnosti) zážitek.

Toto může tvořit problémy při pohybu, protože při nasazeném HMD nelze vidět na své reálné okolí. Proto byl implementován *Chaperone systém*, který dokáže indikovat hranice trackovacího okolí ve VR prostředí (obrázek 3.4) [29]. Všechna připojená hardwarová zařízení se senzory (ovladače, trackery, HMD) jsou trackována se stupněm volnosti 6DOF.



Obrázek 3.4: Chaperone systém - ve virtuálním světě jsou umístěny mřížky, které označují vyměřené hranice [29].

**Stupeň volnosti** Stupeň volnosti (*degrees of freedom*, nebo DOF) je dle definice termín popisující počet nezávislých způsobů, kterými se dynamický systém může hýbat, bez narušení jakéhokoliv omezení. V terminologii VR se jedná o jeden z parametrů HMD zařízení. Popisuje počet způsobů pohybu, kterým se trackuje HMD nebo ostatní trackovací zařízení [30].

V trojrozměrném světě, lze rozlišovat dohromady šest stupňů volnosti objektu. Tři pro posun (pohyb pomocí translace) ve světových osách  $x$ ,  $y$ ,  $z$ . Další tři jsou pro otočení (pohyb pomocí rotace) kolem světových os  $x$ ,  $y$ ,  $z$ .

V rámci VR rozlišujeme jako parametr dva druhy 3DOF a 6DOF. Většina HMD má schopnost šestého stupně volnosti, tedy 6DOF. Toto trackování je použito i na ovladače a ostatní zařízení. V případě 3DOF je odstraněno trackování translace, tedy nejsou zaznamenány (netrackuje) změny v tomto pohybu.

### 3.2.3 Verze

Nové verze HTC Vive se liší několika parametry svého zařízení. Všechny obsahují zorné pole  $110^\circ$  a trackování 6DOF, ale některé nabízejí nový způsob trackování nebo vylepšené technologie [31]. V tabulce 3.1 je porovnání všech verzí s původním HTC Vive v základních parametrech HMD.

	Rozlišení	Displej	Trackování	Konektivita	Další funkcionality
HTC Vive	$1080 \times 1200$	OLED	Outside-in	HDMI, USB	-
HTC Vive Pro	$1440 \times 1600$	OLED	Outside-in	DP 1.2, USB-C 3.0	Sluchátka
HTC Vive Pro Eye	$1440 \times 1600$	OLED	Outside-in	DP 1.2, USB-C 3.0	Sluchátka, eye-tracker
HTC Vive Cosmos	$1440 \times 1700$	LCD	Inside-out	DP 1.2, USB-C 3.0	Sluchátka
HTC Vive Cosmos Elite	$1440 \times 1700$	LCD	Outside-in	DP 1.2, USB-C 3.0	Sluchátka

Tabulka 3.1: Variace HTC Vive a porovnání mezi jednotlivými verzemi na základě základních parametrů.

---

# Unity3D

Unity je herní engine vyvíjen společností *Unity Technologies* [32]. Pojem herní engine (*game engine*) představuje vývojové prostředí v podobě softwaru, které umožňuje vyvíjet hry a aplikace pro různé platformy [33]. V dnešní době patří do první linie nejpoužívanějších enginů. Primárním důvodem je především jeho flexibilita a vhodné uživatelské rozhraní pro začínající vývojáře. Engine nabízí skriptovací API (*Application Programming Interface*) v jazyce C#. Unity umožňuje tvoření aplikací ve 2D i 3D rozhraní pro široké spektrum platform [34]. Podpora pro různé platformy je v dnešní době samozřejmostí a nutné kritérium pro vývoj mnoha her i aplikací.

## 4.1 Komponenty

Komponenta je vlastnost objektu, která charakterizuje dané chování pro daný objekt. Celý systém vlastních herních objektů je založen na připínání těchto komponent k daným objektům. Každá před implementovaná komponenta má své vlastní API rozhraní, které Unity nabízí a lze jej využít ve vlastním skriptu.

## 4.2 Skriptování

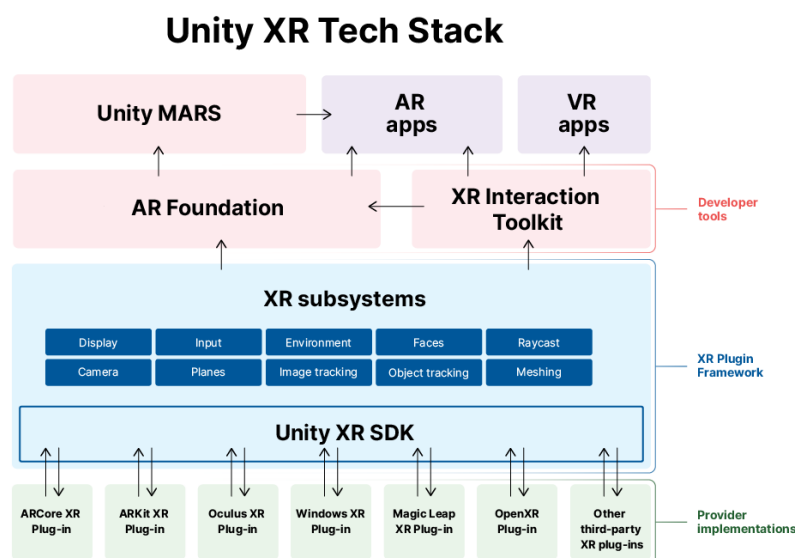
Pomocí vlastního skriptu je možné plně ovládat chování herních objektů, ale i scény a celkový běh aplikace. Unity nativně podporuje skriptování v jazyce C#. V dřívějších verzích bylo možné skriptování v jazycích Boo a UnityScript [35]. Všechny vytvořené C# skripty standardně dědí ze vestavěné třídy *MonoBehaviour*. Takový skript se stává komponentou. Nejdůležitější vlastnost této třídy jsou tzv. události (*events*). Události jsou zavolány automaticky za běhu aplikace v závislosti na typu události (např. funkce *Start* - inicializace objektu nebo *Update* - zavolána každý snímek).

### 4.3 Nástroje pro vývoj virtuální reality

Unity nabízí širokou podporu pro vývoj ve VR a nelimituje vývojáře zaměřením na jedno vývojové HMD. Způsoby implementace VR jsou realizovány pomocí balíčků SDK (*Software development kit*, sada vývojových nástrojů). V této sekci se zaměřím především na balíčky Unity XR Interaction Toolkit a SteamVR. Dále analyzuji Mixed Reality Toolkit, Virtual Reality Toolkit a Oculus Integration Package.

#### 4.3.1 Unity XR Interaction Toolkit

*Unity XR* [36] je systém v Unity pro vývoj a správu XR aplikací. *Unity XR Interaction Toolkit* [37] je systém interakcí v Unity umožňující univerzální vývoj pro zařízení VR, AR i MR. Účel balíčku je sjednocení vývoje XR v Unity a možnost tvorby základních interakcí bez nutnosti programování nebo psaní dlouhého kódu. Jednotlivé podporované HMD vyžadují vlastní plug-in, který je k dispozici přímo od vývojáře. Avšak všechny společně sdílí vývojovou platformu, což je díky Unity XR umožněno (obrázek 4.1).



Obrázek 4.1: Grafická reprezentace Unity XR v Unity 2020.2 [38]

**Funkcionality** Balíček obsahuje několik skriptů, které jsou přizpůsobitelné. Základem je dědění z hlavních tříd a vyvolávání událostí, které nabízí (např. selekce, aktivace, deaktivace, haptické odezvy, zvýraznění objektu aj.) [39]. Interakce se rozdělují na *interactory* a *interactable* objekty, které mezi sebou interagují. Je možná i interakce s UI elementy přes laserový interactor.



### Základ

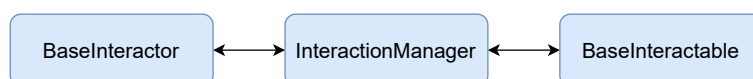
- *Rig* představuje základní komponentu a logiku VR kamery.
- *Controller* dodá objektu základní logiku trackování ovladače.
- *LocomotionSystem* umožňuje pohyb Rigu.

### Teleport

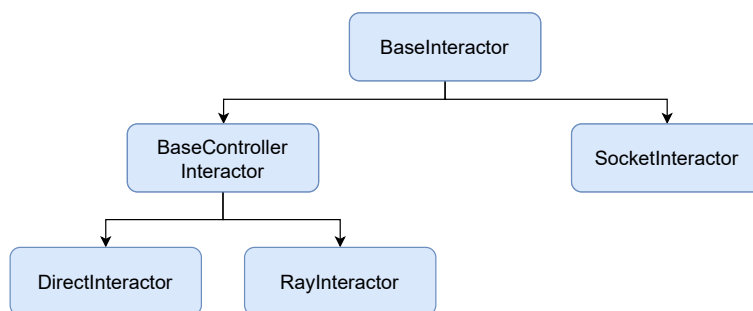
- *TeleportationProvider* dodá objektu schopnost se teleportovat.
- *TeleportationArea* dodá objektu schopnost teleportačního místa.
- *TeleportationAnchor* dodá objektu schopnost teleport. bodu.

### Interakce

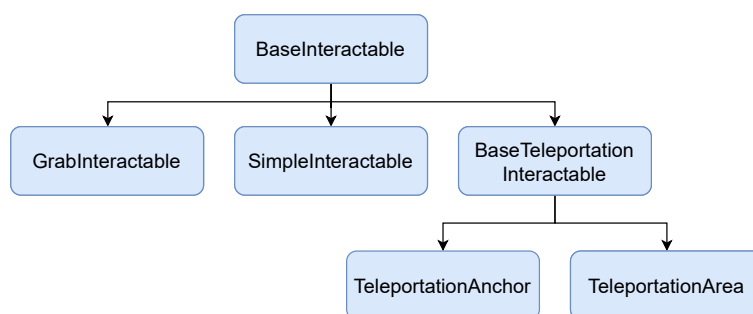
- *BaseInteractor* (4.3) dodá objektu schopnost interakce s *BaseInteractable*.
  - *DirectInteractor* dodá objektu schopnost přímé interakce na ovladači.
  - *RayInteractor* dodá objektu schopnost laserové interakce na ovladači.
  - *SocketInteractor* dodá objektu schopnost přichytit objekt.
- *BaseInteractable* (4.4) dodá objektu schopnost interakce.
  - *GrabInteractable* dodá objektu schopnost zvednutí.
  - *SimpleInteractable* dodá objektu schopnost jednoduché interakce.
- *InteractionManager* (4.2) komunikuje mezi interactory a interactable.



Obrázek 4.2: InteractionManager



Obrázek 4.3: BaseInteractor



Obrázek 4.4: BaseInteractable

**Získání vstupů** Existují dva způsoby pro získávání vstupů z ovladačů nebo dalších zařízení. Jedná se o systémy *Device-based* a *Action-based* (některé skripty v balíčku mají variantu Device-based nebo Action-based, je třeba využít správné skripty pro vybranou metodu zpracování vstupů).

Device-based systém získává vstupy přímo z daného zařízení. Tedy pomocí objektu konkrétního zařízení získáváme jeho vstupy do proměnné a pak vyhodnotíme vstup ručně. Tento systém je sice jednoznačný, ale zároveň není flexibilním ani přehledným řešením při implementaci interakcí pro XR aplikace.

Action-based způsob používá objekty *Actions* (akce) ke čtení nepřímého vstupu z jednoho, ale i více zařízení. K tomu využívá systém pro zpracování vstupů v Unity - *Input System* (zpracovává vstupy ze všech možných zařízení). Na akci se namapují vstupní tlačítka (např. aktivovat objekt - trigger). Jednotlivé akce jsou součástí setů akcí (např. levý a pravý ovladač). Výhodou tohoto systému je možnost namapování více vstupů na danou akci.

**Klady** Tvorba vlastních interakcí je silnou stránkou tohoto balíčku. Díky dědění z tříd interakcí je možné přehledně vytvořit komplexní interakce. Další výhodou je kompatibilita s většinou HMD, což byl jeden z účelů XR systému v Unity. Action-based systém je přehlednějším řešením pro rozvržení akcí na ovladači.

**Zápory** Balíček obsahuje pouze skripty a základní prefaby pro jednoduchou VR aplikaci, ale vývojář musí tyto funkcionality nastavit samostatně (např. teleport). Pro začínajícího vývojáře by toto mohlo tvořit komplikaci, jelikož systém nabízí značné množství vlastního přizpůsobení.

### 4.3.2 SteamVR

*SteamVR* [40] je aplikace od společnosti Valve dostupná přes Steam [41]. Využívá OpenVR API pro přístup k hardwaru a podporuje několik známých HMD včetně HTC Vive, Oculus Rift, Valve Index nebo WMR. Balíček pro vývojáře byl vydán pod jménem SteamVR Plugin [42]. Pro Unity je volně dostupný a podporuje všechny HMD, které podporuje samostatná aplikace SteamVR.

**Funkcionalita** Balíček nabízí základní funkcionality pro tvorbu jednoduché VR aplikace (kamera, trackování ovladače, uchopování předmětů) [43]. Také je obsaženo několik ukázkových komplexnějších interakcí (např. střílení z luku, model a animace ruky, virtuální joystick, virtuální tlačítko aj.).

#### Základ

- *CameraRig* prefab představuje hlavní kameru VR.
- *Player* prefab představuje hlavní kameru VR včetně logiky ovladačů.
- *Hand* dodá objektu Player základní logiku ovladače/ruky.

#### Teleport

- *Teleporting* prefab umožňuje schopnost teleportace pro prefab Player.
- *TeleportArea* dodá objektu schopnost teleportovat se na jeho povrch.
- *TeleportPoint* představuje ukotvený teleportační bod.

#### Interakce

- *Interactable* dodá objektu schopnost interagovat s ním.
- *Throwable* dodá objektu schopnost základní interakce (uchopení, házení).

**Získání vstupů** Verze 2.0 představila nový systém pro zpracování vstupu přes systém akcí *Actions*. Systém funguje velice podobně jako v Unity XR, tedy pomocí definovaných akcí s přiřazenými vstupy. SteamVR má své vlastní rozhraní pro mapování ovladačů pomocí *SteamVR Input*.

**Klady** SteamVR 2.0 obsahuje základní i komplexní interakce. Nabízí vhodné prostředky pro jednoduchou VR aplikaci. Aplikace vytvořená pomocí tohoto balíčku je kompatibilní se všemi HMD, které podporuje samostatné SteamVR. Je dostupný komplexní teleport, který vývojář nemusí implementovat.

**Zápory** Vlastní přizpůsobení může tvořit komplikaci a není vhodné pro tvorbu vlastních komplexních interakcí. Systém interakcí není flexibilní (narozdíl od Unity XR ITK).

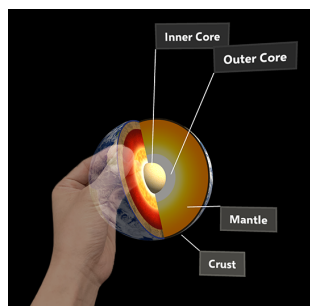
### 4.3.3 Mixed Reality Toolkit

*Mixed Reality Toolkit* (MRTK) [44] je projekt od společnosti Microsoft se zaměřením pro vývoj aplikací ve VR/MR. Hlavní využití je pro HoloLens, avšak i přesto MRTK podporuje značné množství ostatních platforem a díky tomu i značné množství HMD zařízení pro VR (všechna WMR HMD, HTC Vive, Oculus Rift, Oculus Quest aj.). V Unity je nutné nastavit platformu na *Universal Windows Platform*, aby bylo možné vytvořit spustitelnou verzi s tímto balíčkem.

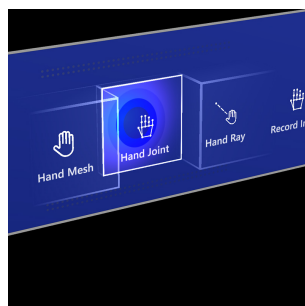
**Funkcionality** S porovnáním ostatních balíčků je tento balíček opravdu jedinečný a liší se od ostatních. Rozdíl je hlavně díky funkcionalitám navrženým s vývojem pro HoloLens. Tyto funkcionality lze využít i pro vývoj VR aplikace. Návrh funkcionalit balíčku je zaměřen především na interakci s UI a na základní interakce s objektem (uchopení, přenášení, škálování, ukazovátka, lasery, štítky, informační panely, UI elementy - obrázek 4.5 aj.), jelikož HoloLens je ovladatelné přímo pomocí rukou místo ovladačů [45]. V balíčku je přítomna i vlastní klávesnice a nechybí také základní teleport.

**Klady** Vývoj je možný pro VR/AR. Interakce je pro uživatele jednoduchá, jednostranná a robustní. Funkcionality jsou dobře navrženy a pro interakci a zkoumání objektu jsou ideální. Práce s UI elementy je silným kladem tohoto balíčku. Dokumentace pro balíček je přehledná a obsahuje několik návodů pro začínající vývojáře.

**Zápory** Počet skriptů je obsáhlý a struktura skriptů je komplikovanější, tedy pro začínajícího vývojáře by toto mohlo tvořit potenciální komplikaci. Pro většinu aplikací a her pro VR nejsou tyto funkcionality ideální, jelikož se zaměřují hlavně na interakci s UI a jednoduchou interakci s objektem.



(a) Manipulace s objektem [46]



(b) Stlačitelný tlačítko [47]

Obrázek 4.5: MRTK interakce pro VR/MR platformu

#### 4.3.4 Virtual Reality Toolkit

*Virtual Reality Toolkit* (VRTK) [48] je balíček s podporou platform vyžívající SteamVR nebo OVR. Jedná se spíše o rozšíření těchto balíčků ve formě third-party softwaru a je na jejich funkcionalitách plně závislý. Při importování balíčku VRTK v3 do Unity se importují i balíčky SteamVR 1.2.3 a OVR.

**Funkcionality** VRTK v3 nabízí nové funkcionality včetně laserových ukazatelů, virtuálních tlačítek, uchopování předmětů pomocí dvou rukou, interakce s UI pomocí laseru, nápověda pro ovladače, nové formy teleportu, střelení z luku, radiální menu a další komplexní interakce [49]. Vývojové prostředky jsou opravdu široké a kreativní, zároveň vhodné pro vývoj aplikací různých druhů. Tato verze VRTK již není podporována, ale dokumentace je stále přístupná.

**Klady** Toto rozšíření obsahuje mnoho zajímavých a komplexních interakcí, které lze využít bez náročné implementace. Z hlediska vývojáře je vhodný pro začátečníky a projekty, které nejsou tolik obsáhlé a nevyžadují vlastní systémy pro interakce.

**Zápory** VRTK v3 je plně závislý na SteamVR 1.2.3 a OVR, které jsou importovány do projektu společně s tímto balíčkem. To může tvořit značné komplikace při neustálých aktualizacích. Příkladem tohoto problému je vydání nové verze SteamVR 2.0. VRTK v4 beta je aktuálně stále ve vývoji a neobsahuje přehlednou dokumentaci. Struktura beta verze se neustále mění a nelze momentálně spoléhat na její funkcionalitu nebo dokumentaci.

#### 4.3.5 Oculus Integration Package

*Oculus Integration Package* (OVR) [50] je balíček obsahující základní komponenty a skripty usnadňující vývoj pro VR v Unity. Od předchozích balíčků se liší tím, že podporuje pouze zařízení Oculus.

**Funkcionality** Tento balíček je velice jednoduchý a neobsahuje komplexní interakce, nebo flexibilní skripty. Základ tvoří uchopování objektů a čtení vstupů z ovladačů (manuálně, device-based).

**Klady** Základ je jednoduchý a vhodný pro vlastní přizpůsobení.

**Zápory** Balíček podporuje pouze zařízení Oculus a jeho funkcionality nejsou komplexní, či jednoduché pro implementaci vlastního přizpůsobení. Vývojář musí implementovat funkcionality bez možnosti rozšířit již dostupné interakce (např. teleport).



## Základní metody interakcí ve virtuální realitě

Pomocná hardwarová zařízení (ovladače, eye-tracker, hand-tracker) umožňují větší rozmanitost při interakci s objekty ve virtuálním prostředí, avšak samotnou interakci v aplikaci nezmění. Například pomocí technologie trackování očí je možné docílit lepší a originálnější interakce s objekty, ale základy zůstávají stejné (stejného paprsku lze docílit i tlačítkem na ovladači). Důležité je vytvořit interakce, které jsou pro uživatele pochopitelné a srozumitelné. Ideální návrh je originální, efektivní a srozumitelný. V této sekci se budu věnovat základním návrhům a interakcím ve virtuálním prostředí.

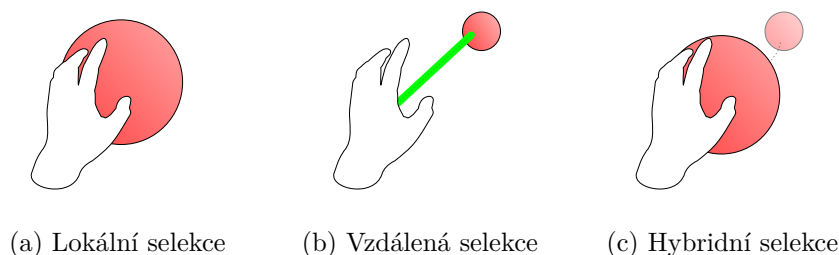
### 5.1 Selekcce

Pomocí selekcce (výběru) objektu uživatel určí, s kterým objektem chce interagovat a manipulovat. Výběru lze docílit pomocí několika metod, avšak mezi základní patří lokální a vzdálená selekcce [51].

*Lokální selekcce* (obrázek 5.1a) vyžaduje přímou interakci ruky (ovladače, či jiného trackovacího zařízení) s objektem. Neboli přítomnost ruky v selektivním rozsahu objektu (*bounding-box*). Tato metoda je užitečná pro manipulaci s objektem zblízka (např. práce s objektem do velikosti lidské postavy, nebo simulace reálné interakce s objektem).

*Vzdálená selekcce* (obrázek 5.1b) pracuje s prvkem paprsku (laseru). Selekcce objektu je provedena namířením paprsku na objekt. První objekt, který tvoří s paprskem od svého zdroje průsečík je vybrán. Tato metoda je užitečná pro manipulaci s objektem ze značné dálky (např. práce s objektem velkého rozměru, nebo přesouvání předmětů z dálky).

*Hybridní selekcí* (obrázek 5.1c) se vybere objekt vzdáleně (průsečík paprsku určí objekt) a poté se přitáhne automaticky k ruce, kde s ním lze pracovat lokálně.



Obrázek 5.1: Vizualizace základních selekcí

### 5.1.1 Uchopení

Při selekci je transformace objektu plně ovlivněna transformací ruky. Podobně jako tomu je v reálné realitě, i ve VR lze ovlivnit transformaci objektu jednou nebo dvěma rukama.

*Jednoruční* uchopení je jednoduché a typické pro většinu interakcí s objekty. Místo úchytu má identickou pozici s rukou, popř. je upravena vlastní rotací nebo posunem.

*Obouruční* uchopení je druhá a komplexnější metoda pro uchopení. Základem je správné natočení objektu při úchytu a manipulaci druhé ruky. Toho lze docílit např. pomocí tzv. *LookAt* transformace, která natočí objekt ve zvoleném směru (využívá se při namíření kamery na pohledový bod) [52].

## 5.2 Manipulace

Manipulací s objekty se myslí samostatná interakce a vliv na objekty ve virtuálním prostředí. Interagovat lze několika způsoby, realisticky i nerealisticky v závislosti na charakteru interakce [51].

*Tahem ruky* (obrázek 5.2) lze ovládat objekty přirozeně a direktně. Při přemísťování objektu lze uchopit předmět rukou a přímo ho přemístit tahem dané ruky. Mezi méně přirozené interakce patří např. škálování pomocí rukou, které funguje na principu uchopení objektu dvěma rukama a následně tahem měnit škálu. Otáčení lze provést metodou virtuálního trackballu (*virtual trackball*), kde tahem ruky otáčíme objekt v daném směru.

Ovládání *fyzickým ovladačem* (obrázek 5.3a) je mechanika ovládání přes prvky ovladače. Typické je namapování selekce objektu na jedno tlačítko na ovladači, který uživatel vyvolá v blízkosti předmětu. Otáčení, škálování a translaci objektu lze provést např. pomocí joysticku nebo trackpadu.

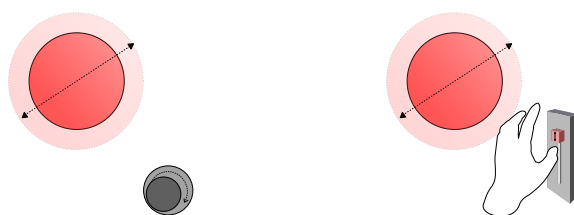
*Virtuální ovládání* (obrázek 5.3b) funguje na podobném principu jako fyzické, avšak fyzické ovladače nahradily virtuální. Pomocí virtuálních rukou lze ovládat přímo ve virtuálním světě. Hlavní je interakce s objektem za pomoci jiného virtuálního objektu. Příkladem jsou 3D tlačítka.





(a) Objekt před škálováním (b) Objekt po škálování

Obrázek 5.2: Vizualizace škálování pomocí tahem ruky



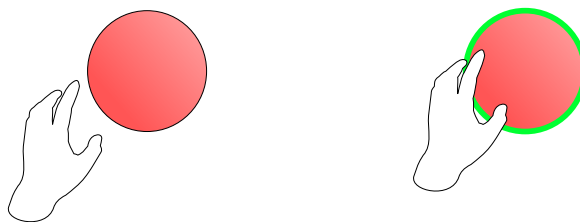
(a) Fyzický ovladač (joystick) (b) Virtuální ovladač (slider)

Obrázek 5.3: Vizualizace škálování pomocí fyzického/virtuálního ovladače

## 5.3 Indikátory

Virtuální svět je komplexní a nabízí několik metod pro manipulaci a interakci s objekty. Avšak pro uživatele může být složité pochopit základní metody ovládání, nebo vůbec rozeznat, že s daným objektem lze interagovat [53]. Nedostatek navigačních prostředků ve virtuálním světě může vést k dezorientaci uživatele v prostoru, nebo nedokončení úlohy.

Důležité je upozornění na objekty zvláštního charakteru a navigovat při interakci. Příkladem je barevné zvýraznění objektu, zvýraznění obrysu objektu (obrázek 5.4), nebo zvýraznění navigační trasy.



(a) Objekt bez zvýraznění (b) Zvýrazněný obrys objektu

Obrázek 5.4: Vizuální indikátor objektu, s kterým lze interagovat.



---

## Analýza aplikací

V této sekci budu analyzovat aplikace a jejich interakce. Zaměřím se především na VR aplikace v historickém prostředí, ale i na desktopové varianty jiných zajímavých aplikací. Interakce analyzuji a případně provedu porovnání podobných interakcí. Z analyzovaných interakcí bude sestaven návrh interakcí pro prototyp této práce. Vybrané aplikace obsahují komplexnější interakce.

### 6.1 Shadow Legend VR

*Shadow Legend VR* [54] je RPG hra navržena pro VR platformu. Svět se nachází ve středověkém prostředí, kde je možné provádět mnoho zajímavých interakcí. Vizualizační prvky hry jsou realistické a odpovídají středověkému prostředí.

**Interakce** Hra nabízí několik interakcí s objekty, např. zjednodušené kování meče, střelbu z luku a kuše, broušení meče, krmení koní, šplhání po zdi, souborový systém, lupa, minihry atd. Objekty jsou uchopeny pomocí lokální selekce, tedy přímo pomocí rukou.

#### 6.1.1 Kování

- Kovárna (obrázek 6.1a) obsahuje základní potřeby pro simulaci kování včetně uhelné pece, kalící vody, kovacího kladiva, kovadliny a ocele.
- Ocele na kování jsou především zbraně, ale i podkovy. Nekovaný meč je indikován svou „zakřiveností“ a má již připevněnou rukojeť.
- Ocel se před kovááním zahřeje v uhelné peci. Po pár sekundách začne postupně měnit barvu. Míra zahřátí je indikována změnou barvy ocele.
- Jakmile kovaný předmět se již více nezahřívá (nemění barvu), tak ho lze přemístit ke kovadlině. Meč se ková kovářským kladivem.
- Progres kování je indikován dle zakřivenosti meče. Ocel je třeba několikrát zahřát, aby se dokovala. Dokovaný meč má plochý tvar.



(a) Kování meče kladivem



(b) Broušení meče na brusce

Obrázek 6.1: Shadow Legend VR

### 6.1.2 Meče

- Meče lze získat kováním. Obsahují vlastní statistiky (hlavní je útok), které se mění při aktivitách s mečem.
- Pokud hráč trénuje soubojové mechaniky, tak meč ztrácí hodnotu útoku. Naopak zvýšit hodnotu útoku lze broušením na brusce.
- Meč lze připnout k opasku a tím ho má uživatel stále při sobě (kdykoliv jde zaměnit za jiný).

### 6.1.3 Broušení

- Bruska je ovládána pomocí točení rukojetí (tedy uchopení rukojeti a tahem ruky). Tah rukojetí není plynulý. Po přiložení meče na brusku je meč obroušen.
- Zpětné nabrání útoku meče je možné při nabroušení meče na brusce. Hodnota útoku je přidána po několika sekundách broušení fixně daným číslem (obrázek 6.1b).

### 6.1.4 Lukostřelba

- Uchopení luku je „nepřirozeně“ otočeno o devadesát stupňů dopředu.
- Pokud je luk uchopen, lze z něj získat šípy z toulce (nelze pokud není uchopen). Toulece je připevněn za hráčem (na zádech) a má neomezenou kapacitu.
- Šíp se automaticky upevní jakmile se přiblíží k upevňovacímu místu na těživě luku. Uchopení těivy, natáhnutí a puštění vystřelí šíp.
- Některé objekty reagují na zasáhnutí šípem (aplikuje se na ně kinematická energie šípu, např. pomalé posouvání objektu).
- V rámci příběhu je minihra, ve které musí hráč získat magické runy (kamene). Některé runy jsou umístěny v nedosažitelné vzdálenosti, tedy jediný způsob je sestřelení kamene pomocí šípů.

## 6.2 A Township Tale

*A Township Tale* [55] je online multiplayer RPG svět navržen pro VR platformu. Jelikož se jedná o online multiplayer, tak je k dispozici několik serverů. Po přihlášení je hráč přenesen do místnosti, kde si vybere a vytvoří vlastní avatar. Poté si vybere svět, do kterého se přihlásí. Hráč má vlastní statistiky (výdrž, zdraví aj.), které si musí neustále doplňovat. Hráči mohou mezi sebou plně komunikovat přes mikrofon, obchodovat se surovinami či společně plnit úkoly.

**Interakce** Hlavní náplní této herní aplikace je získávání surovin a práce v různých řemeslech (kovář, zemědělec, horník, bojovník, dřevorubec aj.). Pomocí těchto rolí mohou hráči získat potřebné suroviny pro tvorbu různých objektů a zpřístupní tím další interakce (souboje, lukostřelba, dřevorubectví, hornictví, rozšíření inventáře, tvorba mincí aj.). Interakce jsou tvořeny přes virtuální ovládání, které dobře simulují realitu (např. otáčení kulatého stolu pomocí otočné páčky). Dokonce i záleží na tom, co inventář obsahuje a v jakém množství. Těžší inventář znamená větší zátěž a pomalejší chůzi.

### 6.2.1 Spojování materiálů

- Základní suroviny lze spojit a vytvořit tím unikátní náradí či zbraně (např. připojení rukojeti k meči po dokování).
- Kompatibilita surovin je indikována barevným proudem (obrázek 6.2a).
- Některé suroviny mají více přípojných míst. Příkladem náradí je sekera, kladivo, pochodeň, meč, šíp aj.

### 6.2.2 Výroba z materiálů

- Komplexní objekty lze vytvořit z více surovin pomocí návodu (např. záchytný koš, neboli rozšířený inventář).
- Návod přesně určuje, které suroviny hráč potřebuje. Hráč je doprovázen barevnou indikací, která určuje, kam surovinu umístit.
- Aby suroviny držely pohromadě, je nutné použít kladivo a připevnit je dohromady (obrázek 6.2b).

### 6.2.3 Dřevorubectví

- Kácení stromů vyžaduje, aby hráč aktivně dělal správné pohyby sekerou z několika úhlů po dobu několika sekund, někdy i minut.
- Sekání je dokončeno po pádu stromu a dalšího sekání dle potřeby surovin hráče.



(a) Dvojice materiálů



(b) Komplexní výroba

Obrázek 6.2: A Township Tale

### 6.2.4 Kování

- Hlavní mechanika je uhelná pec, která slouží jako zdroj materiálů a zároveň vyžaduje materiály ke zpracování. V uhelné peci je třeba dodat uhlí a rozpálit oheň.
- Nejdříve je potřeba vytvořit ocelové cihly. Do pece je umístěno několik kusů rudy. Rozpálí se oheň a po určité době jsou vytvořeny ocelové cihly.
- Následně je potřeba umístit tyto cihly do pece znovu a vybrat z nabídky tzv. „blueprint“. Blueprint je tabulka, která určuje, co se z daného materiálu má vytvořit (např. dlouhý meč, špička šípu atd.).
- Výsledný objekt je část ocele připravená na kování. Ocel je třeba zahřát v jiné peci, míra zahřátí je indikována svou barvou.
- Následně je třeba mít vlastní kovací kladivo (lze vytvořit ze spojování materiálů) a kovat zahřátou ocel.
- Opakované zahřátí ocele je potřeba v závislosti na tvaru ocele (ocel pro dlouhý meč je potřeba zahřát několikrát, pro krátkou čepel jednou). Dokončení kování je indikováno grafickým indikátorem.
- Dokovaná ocel se stává materiálem, který lze napojit na ostatní materiály. Například meč je napojen na rukojeť.

### 6.2.5 Lukostřelba

- Základ luku je možné vytvořit postupným osekáváním dřeva na speciálním přístroji a bluepintu (podobný princip jako u kování). Dále je třeba spojit základ luku s dalším materiálem k vytvoření tětiny na luku.
- Šípy je možné vytvořit z dřeva pomocí dřevorubectví. Tenké dřevo se spojí s špičkou šípu a s peřím.
- Materiál špičky a peří ovlivní dynamiku šípu. Šíp bez peří míří náhodně.
- Natažení tětiny na luku a vystřelení šípu funguje podobným principem jako v předchozích aplikacích.

## 6.3 Blade and Sorcery

*Blade and Sorcery* [56] je akční soubojová herní aplikace navržená pro VR platformu. Souboje využívají středověké zbraně a odehrávají se ve středověkém prostředí. V této simulaci je kladen důraz na realistickou fyziku a interakce se zbraněmi v souboji.

**Interakce** Hlavní interakcí je soubojový systém. Hráč má schopnost uchytit zbraně oběma rukama na jakémkoliv místě, uchopit zbraně protivníka aj. Hráč má také možnost využít magických schopností (což je v rozporu s realistickým soubojem ve hře). Způsob selekce objektů je pomocí hybridní selekce. Ta probíhá postupně namířením na objekt, stisknutím tlačítka (objekt je vybrán a „levituje“ ve směru pohybu ruky, lze vidět na obrázku 6.3b) a následné stisknutí jiného tlačítka (objekt se přiblíží a je připevněn k ruce).

### 6.3.1 Soubojový systém

- Soubojový systém je založen na realistické fyzice a plně záleží na každém detailu (místo udeření, síla udeření, váha zbraně aj.).
- Pokud je úder přímý a silný, tak zbraň dokonce může vypadnout z ruky. Hlavní tedy je přesnost v manipulaci zbraně za pomoci ovladačů.
- Ve hře existuje několik módů, které definují obtížnosti soubojů (počet protivníků, obtížnější mód aj.).

### 6.3.2 Zbraně

- Výběr zbraní je široký a neomezený (např. dlouhé a krátké meče, dýky, obouruční zbraně, luky, toulce, štíty aj.).
- Uchopení zbraní je jednoruční i obouruční (obrázek 6.3a).
- Při jednoručním uchopení je možné být vyzbrojen dvěma zbraněmi (každá ruka jednu). Pro vylepšení obrany je lze také ozbrojit se štítem.
- Uchopit zbraň (jednoručně i obouručně) je možné víceméně na jakémkoliv místě (na úsečce mezi počátečním a konečným bodem úchytu).

### 6.3.3 Pouzdra

- Zbraně lze u hráče umístit do několika pouzder. Hráč má pouzdro po své levé ruce u pasu a za zády další dvě.
- Systém pouzder je flexibilní a funguje jako realistický inventář pro minimum objektů.
- Pokud se ruka dostane k pouzdru, tak je zobrazen informační panel, který obsahuje informaci o stavu pouzdra (prázdný či jaký objekt se v něm vyskytuje).



(a) Indikátor uchopení



(b) Hybridní selekce

Obrázek 6.3: Blade and Sorcery

### 6.3.4 Lukostřelba

- Uchopení luku je nerealistické a nepřírozené (otočeno o devadesát stupňů).
- Luk i toulec mohou být připnuty k pouzdrům na zádech. Hráč nasadí šíp z připevněného toulce.
- Narozdíl od předchozích aplikací je nutné fyzicky nasadit toulec (tedy luk negarantuje automatické nasazení toulce).
- Šíp se nasadí na tětivu automaticky. Natažen a puštění tětivy vystřelí šíp. Šípy ubírají život protivníkovi, jinak nereagují na prostředí.

## 6.4 Kingdom Come: Deliverance

*Kingdom Come: Deliverance* [57] je RPG hra vyvíjená českou společností Warhorse Studios. Děj hry se odehrává v historickém prostředí roku 1403 a je založená na skutečných událostech. Hlavní náplní je dějová linka společně s několika aktivitami, který hráč může ve světě vyzkoušet.

**Interakce** Hra není kompatibilní s VR zařízením, tedy interakce a ovládání je z tohoto pohledu jiná. Hráč má na výběr několik historických interakcí. Interakce byly navrženy, aby byly co nejvíce realistické (např. broušení meče na brusce, lukostřelba). Nejdůležitější je souborový systém a interakce se zbraněmi, avšak existují i ostatní unikátní aktivity (např. odemykání zámků, jezdeckví, alchymie). Sbíráání a tvorba předmětů přes tyto aktivity je důležitá část RPG. Všechny předměty mají vlastní statistiky, které jsou ovlivněny dalšími aktivitami (např. obroušení meče). Hráč si zlepšuje schopnosti, které vylepší průběh danou aktivitou.



### 6.4.1 Broušení

- Bruska se ovládá přes pohyb nohy. Je nutné neustále hýbat nohou, aby se kolo brusky točilo. Toto je umožněno přes fyzické ovládání (opakované stisknutí tlačítka).
- Meč je uchopen obouručně, jednou rukou za rukojeť a druhou za opačný konec meče (obrázek 6.4a).
- Meč je rotován a posunován dle jedné osy, která vede podél těla meče. Rotace i translace je umožněna přes fyzické ovládání (joystick - osa y pro rotaci, osa x pro translaci).
- Dále lze zvýšit tlak zbraně na brusku přes fyzické ovládání (stisknutí tlačítka).
- Pro správné obroušení meče je nutné natáčet a posunovat meč v dané ose. Zároveň záleží na tlaku zbraně na brusku. Správné obroušení je indikováno jiskrama a zároveň se zvyšuje statistika ostrosti zbraně.

### 6.4.2 Odemykání zámků

- Odemykání zámků je 2D minihra, která umožní hráči otvírat nepřístupné oblasti.
- Interakce lze vyvolat na daných místech a za pomoci speciálního předmětu paklíče (hráč ho musí mít v inventáři).
- Náplň minihry je udržování paklíče ve správné pozici při točení zámku (obrázek 6.4b).
- Točení zámku je uskutečněno fyzickým ovladačem (první joystick) a zároveň točení paklíče v opačném směru (druhý joystick). Tím se udržuje správná pozice paklíče.
- Po otočení zhruba o 180° a udržování správné pozice je zámek odemčen.
- Pokud je paklíč v nesprávné pozici (v jakémkoliv momentě), tak se poškodí nebo dokonce rozbije.

### 6.4.3 Lukostřelba

- Hra nabízí několik luků, které se liší statistikou síly.
- Použitelnost luku je ovlivněna statistikami hráče (minimální síla, hbitost, stamina aj.). Tedy každý luk vyžaduje tyto statistiky na určitou úroveň.
- Šípky obsahují statistiku útoku, jelikož se jedná o formu zbraně.
- Při uchopení luku a nenatažené tětivy je hráči poskytnut indikátor cíle, který ukazuje na zamířené místo. Po natažení tětivy tento indikátor zmizí.
- Hlavní účel lukostřelby je lovení a souboje. V souboji je možné s lukem pohybovat, avšak automaticky se sníží přesnost zamíření cílu.

### 6.4.4 Soubojový systém

- Hra obsahuje široký výběr zbraní (meče, dlouhé meče, sekery, kladiva, štíty, luky), které se liší svou statistikou (útok, váha, kondice, durabilita aj.).
- Soubojový systém byl navržen tak, aby se hráč zaměřil na jednoho či menší počet protivníků.
- Statistika hráče obsahuje jeho život a staminu. Stamina je ovlivněna tíhou brnění. Při zasažení protivníkem je snížena stamina i samostatný život hráče.
- Protivník je označen pomocnou „hvězdou“, která určuje možné směry zasažení zbraní.
- Uprostřed hvězdy se objeví ikona blokování, pokud v ten moment je vyžadován blok hráče jako reakce na protivníkov útok (stisknutí tlačítka). Bez blokování je hráč zasažen a je mu snížěn život i stamina.
- Po úspěšném blokování protivníka nastane vhodný moment pro hráčův protiútok. Útok je možný několika směry podle pomocné hvězdy. Zranitelný směr je vyznačen červenou barvou. V zásadě se dělí na útok švihem (pět směrů hvězdy) a bodnutím (prostřední část hvězdy).
- Základní útok je jednoduše obráněn protivníkem (záleží na schopnosti protivníka obránit se). Neočekávaný útok je možný zaměřením útoku na daný směr a změnou směru na opačný v poslední moment.



(a) Broušení meče [58]



(b) Odemykání zámku [59]

Obrázek 6.4: Kingdom Come: Deliverance

## Shrnutí analýzy

Jelikož existuje několik balíčků, projektů, plug-inů pro vývoj aplikací ve VR, tak je třeba shrnout jejich vlastnosti a porovnat je mezi sebou (tabulka 7.2). Vzhledem k tomu, že má práce je omezená na hardware HTC Vive, tak primárním faktorem při výběru je podpora tohoto HMD (tabulka 7.1).

	Platformy
UnityXR ITK	Oculus, WMR, OpenXR/OpenVR
SteamVR 2.0	OpenVR
OVR	Oculus
MRTK	Oculus, WMR, OpenXR, OpenVR
VRTK v3	OVR, SteamVR 1.2.3

Tabulka 7.1: Podporované platformy VR balíčků Unity

	Aktivní podpora	Vhodné pro začátečníky	Vlastní přizpůsobení	Komplexní interakce
UnityXR ITK	✓	✗	✓	✓
SteamVR 2.0	✓	✓	✗	✓
OVR	✓	✗	✗	✗
MRTK	✓	✗	✗	✓
VRTK v3	✗	✓	✓	✓

Tabulka 7.2: Srovnání analyzovaných balíčků vývoje VR v Unity

Na základě analýzy jsem se rozhodla použít ve své práci balíček UnityXR Interaction Toolkit (sekce 4.3.1). Tento balíček je definitivně nejlepší řešení co se týče podpory a nastávajícího vývoje XR. Systém může být však složitý pro začínající vývojáře, naopak vhodný pro projekty s vlastním systémem interakcí.

Většina analyzovaných aplikací byla ve VR provedení, tedy nabízí určitou formu způsobu, kterým lze danou interakci navrhnout a implementovat. Aplikace obsahovaly lukostřelbu, kování a broušení jako formu interakce v historickém prostředí. Pro VR platformu se jedná o vhodnou interakci, neboť dobře využívají spolupráci obou rukou a ovladačů. Tedy budou součástí návrhové i implementační části této práce. Aplikace také obsahují souborové systémy jako formu interakce, avšak nejsou vhodné pro náplň této práce (Kingdom Come: Deliverance, Blade and Sorcery). Podobný případ tvoří 2D minihry, které nejsou vhodné pro VR provedení (Kingdom Come: Deliverance).

Držení luku při lukostřelbě bylo nerealistické (Blade and Sorcery, Shadow Legends VR) a celkově netvoří příjemný zážitek. Jeden z cílů mé implementace bude toto napravit. Luky i šípy se liší různými statistikami, avšak jedná se spíše o variaci a zdokonalení. Luky se liší tvarem a svým účelem (Kingdom Come: Deliverance), kde lepší luk vyžaduje lepší statistiky hráče. Toto není pro účely návrhu vyžadováno. Šípy se liší svou přesností nebo rychlostí (Township Tale). Zdroj šípů je primárně z toulce, s kterým lze manipulovat (Blade and Sorcery). Avšak toulce by měly mít kapacitu a možnost jí doplnit.

Kování nezahrnuje jako nutnou součást procesu kalení (A Township Tale, Shadow Legends VR), proto se pokusím zahrnout tuto aktivitu jako součást procesu kování. Indikátory dokovanosti ocele jsou spíše vizuální a nelze jej exaktně určit. Při zahřátí je změněn materiál kovu, avšak nelze určit teplotu kovu. Dalším cílem návrhu bude zobrazení teplot kovu. Proces kování nelze žádným způsobem zakončit s neúspěchem, naopak realisticky by tomu tak být nemělo.

Broušení ve verzi VR aplikace (Shadow Legends VR) a Desktop aplikace velmi odlišené (Kingdom Come: Deliverance). Ve VR provedení využívá přítomnosti obou rukou a rukojetí je točeno kolo brusky. V druhé ruce je držen broušený objekt. Broušený objekt je jednoduše přiložen na brusku a postupem času zbraň nabírá útok. Desktop verze využívá správné natočení a pozici zbraně, avšak využívá při tom obouruční uchopení zbraně, které lépe manipuluje se zbraní. Broušením se zlepšil parametr obroušenosti, který také ubývá při manipulaci a souboje se zbraní. Točení kola brusky je umožněno nohou. Ve VR provedení by toto nebylo možné nebo realistické, pokud není k dispozici další zařízení (např. Vive Tracker).

Tvorba vlastních předmětů a nástrojů (např. sekera, pochodeň, šíp atd.) je flexibilním systémem s možností variací (A Township Tale). Vyžaduje, aby hráč vynaložil práci i kreativitu pro tvorbu těchto nástrojů a zbraní. Následně slouží pro další využití v interakcích. Systém je založen na skládání a připojování základních materiálů, neboli forma velmi jednoduché skládačky. Další cíl mého návrhu bude zahrnout interakci na principu skládačky.

# Návrh

Na základě analýzy bude navržen a implementován prototyp aplikace. Prototyp aplikace bude obsahovat navržené interakce v jednoduché scéně. Navržené interakce jsou inspirovány z analyzovaných aplikací. Implementace prototypu bude uskutečněna prostřednictvím jednoho z navržených balíčků pro vývoj VR v Unity. Pro návrh bude využito několik forem modelů (model požadavků, model aktivit a doménový model).

## 8.1 Model požadavků

Tato sekce obsahuje model požadavků. Požadavky jsou rozděleny na funkční a nefunkční požadavky. Požadavky jsou kladeny obecně na prototyp (obrázek 8.1) a na samostatné interakce (obrázek 8.2).

### 8.1.1 Obecné na prototyp

Obecné požadavky na prototyp jsou požadavky, které jsou kladeny na celý prototyp. Tedy nejsou vázaný na konkrétní interakce.

Funkční požadavky		Nefunkční požadavky	
Obecné		Obecné	
+ F1	Pohyb	+ N1	Implementace v Unity3D
+ F2	Uchopování předmětů	+ N2	Implementace v UnityXR Interaction Toolkit
		+ N3	Implementace pro virtuální realitu
		+ N4	Lokalizace v angličtině

Obrázek 8.1: Obecné požadavky na prototyp

### **F1 Pohyb**

Veškerý pohyb pro uživatele je v aplikaci řízen teleportační metodou. Teleport je možný stisknutím touchpadu na pravém ovladači směrem nahoru, nasměrování místa a následně puštění tlačítka.

### **F2 Uchopování předmětů**

Uchopení předmětů je možné držením tlačítka selekce. Lze uchopit jen předměty s kterými lze interagovat.

### **N1 Implementace v Unity3D**

Prototyp aplikace bude implementován v enginu Unity 3D.

### **N2 Implementace v UnityXR Interaction Toolkit**

Prototyp aplikace bude implementován pomocí balíčku UnityXR Interaction Toolkit.

### **N3 Implementace pro virtuální realitu**

Prototyp aplikace bude vyvíjen a implementován pro platformu virtuální reality, konkrétně zařízení HTC Vive.

### **N4 Lokalizace v angličtině**

Prototyp aplikace bude zobrazovat text v angličtině, a také obsahovat anglickou dokumentaci pro zdrojový kód.

## **8.1.2 Interakce**

Požadavky na interakce jsou kladeny na každou interakci zvlášť. Funkční požadavky popisují principy interakcí a nefunkční použité metody.

### **8.1.2.1 Lukostřelba**

#### **F1 Získání šípu**

Uživatel nasazuje na luk šípy. Nové šípy získá z toulce nebo lze použít již vystřelené. Toulec má danou kapacitu šípů. Pokud je toulec prázdný, lze ho doplnit.

#### **F2 Vystřelení šípu**

Získaný šíp je držen v jedné ruce. V druhé ruce je držen luk. Položením šípu na vrub tětivy se šíp automaticky nasadí na luk. Natažením tětivy určuje kinematickou sílu vystřeleného šípu. Puštěním tětivy se šíp vystřelí.

Funkční požadavky	Interakce	Nefunkční požadavky	Interakce
<b>Lukostřelba</b>		<b>Lukostřelba</b>	
+ F1	Získání šípu	+ N1	Fyzická simulace
+ F2	Vystřelení šípu	<b>Kování</b>	
+ F3	Terč	+ N1	Deformace modelu
<b>Kování</b>		+ N2	Dynamický materiál
+ F1	Informační panel	<b>Broušení</b>	
+ F2	Nasazení kovu na kleště	+ N1	Fyzická simulace
+ F3	Zahřívání kovu	<b>Oprava kola u vozíku</b>	
+ F4	Kování a kalení	+ N1	Socketový systém
+ F5	Překování do zbraně		
<b>Broušení</b>			
+ F1	Točení rukojetí		
+ F2	Broušení zbraně		
<b>Oprava kola u vozíku</b>			
+ F1	Získání částí		
+ F2	Skládání částí		
+ F3	Nasazení kola		
<b>Pochodně</b>			
+ F1	Rozsvícení pochodně		
+ F2	Zhasnutí pochodně		
+ F3	Zesílení intenzity pochodně		

Obrázek 8.2: Požadavky pro jednotlivé interakce

**F3 Terč**

Vystřelený šíp se zastaví a uchytí při kolizi. Pokud kolizí byl terč, tak se zobrazí a připíše přiměřené skóre.

**N1 Fyzická simulace**

Trajektorie šípu po vystřelení bude počítána pomocí fyzických simulací, které daný engine nabízí s kombinací parametrů ze vstupu ovladačů a vektorové matematiky. Dle nataženosti tětiny je šípu dodána kinematičká energie, kterou šíp poletí, přičemž na něj působí zákony gravitace. Šíp je zastaven kolizí s jiným objektem. Na terč také působí kinematičká energie šípu a je po kolizi s ním rozpohybován.

### 8.1.2.2 Kování

#### F1 Informační panel

Informační panel zobrazuje stav kovaného kovu. Obsahuje typ kovu, typ zbraně po dokování, aktuální teplotu kovu v °C, ideální teplotu kování, ideální teplotu kalení, stav obroušení v % a stav kování v %. Informační panel je možné zobrazit držením tlačítka aktivace pokud je uchopen.

#### F2 Nasazení kovu na kleště

Uživatel uchopí kov do jedné ruky a kovářské kleště do druhé ruky. Připnutí kovu ke kleštím je možné přetažením kovu na vyznačené místo a upuštěním kovu.

#### F3 Zahřívání kovu

Umístěním kleští s kovem do ohně se zahájí zahřívání kovu. Kov je zahříván určitou rychlostí, která se liší dle charakteru kovu. Pokud je kov odložen od místa zahřívání, tak postupně začne chladnout.

#### F4 Kování a kalení

Zahřátý kov je možný kovat a kalit. Kování je možné pouze speciálním nástrojem (kovacím kladivem). Pokud je kov dokován, tak se ponoří do kalící kapaliny (vody) a kalí se (rychle sníží teplotu).

#### F5 Překování do zbraně

Kalením je kov převeden do zbraně, ale pouze pokud splňuje všechny podmínky (kov byl plně dokovaný před kalením, kov byl kován a kalen při správné teplotě). Zbraň již není možné kovat, zahřívát ani zchladit.

#### N1 Deformace objektu

Kovaný objekt bude deformován pomocí změny vertexů meshe za běhu aplikace. Deformace bude ovlivněna kinematickou energií i hmotností působeného předmětu, ale také poměrem zahřátí kovaného objektu a jeho tvrdosti.

#### N2 Dynamický materiál

Zahřátí kovaného objektu je vizualizováno pomocí jednoduchého dynamického materiálu s vlastním vytvořeným shaderem.



### 8.1.2.3 Broušení

#### F1 Točení rukojeti

Uchopení rukojeti a točením pomocí pohybů ruky se kolo brusky začne točit v daném směru a rychlostí.

#### F2 Broušení zbraně

Přiložením uchopené zbraně na točící kolo brusky se zbraň začne brousit. Obroušenost zbraně bude indikována na informačním panelu zbraně.

#### N1 Fyzická simulace

Kolo brusky je otáčeno v závislosti na tahu rukojeti. Rychlost točení brusky je také ovlivněna rychlostí točení rukojeti. Obroušenost zbraně závisí na kinematické energii točící brusky, tvrdosti zbraně a tlaku zbraně na kolo brusky.

### 8.1.2.4 Oprava kola u vozíku

#### F1 Získání částí

Kolo je možné složit z několika částí. Části jsou rozděleny na kategorie podle tvaru. Části je možné získat z nekonečného „balíku“. Pro lepší manipulaci nepůsobí na části zákony gravitace (levitují).

#### F2 Skládání částí

Části lze na sebe připnout pomocí vyznačených připínacích míst (uzlů). Kořen spojených částí je místo vyznačené na pomocné siluetě kola (vizualizuje finální podobu skládačky). Na toto místo se připojí první část, následně se připojují části na sebe. Správně složené kolo lze proměnit na spojené kolo a nelze již rozebrat. Kolo se promění uchopením kořenové části a provedení akce. Počet kol, které lze složit je neomezený.

#### F3 Nasazení kola

Spojené kolo lze nasadit na správné místo vozu.

#### N1 Socketový systém

Části kola lze spojit pomocí socketového systému, který obsahuje uzly, na které se části napojují. Pokud je část umístěna do socketu a upuštěna je automaticky napojena na socket (včetně správné transformace). Systém bude kontrolovat napojené části, zda jsou správně složené.

### 8.1.2.5 Pochodně

#### F1 Rozsvícení pochodně

Pochodeň lze rozsvítit zapálením pomocí jakéhokoliv zdroje ohně. Pomocí uchopení pochodně a vložením do ohně se pochoděň zapálí. Ostatní pochodně lze zapalovat z předešlých zapálených pochodní.

#### F2 Zhasnutí pochodně

Pochodeň lze zhasnout ponořením do jakéhokoliv zdroje vody.

#### F3 Zesílení intenzity pochodně

Pokud je pochoděň rozsvícená a uchopená, lze zesílit její intenzitu světla pomocí tlačítka akce.

## 8.2 Model domény a aktivit

Následující diagramy (doménový diagram a diagram aktivit) jsou rozděleny podle typu interakce. Doménový diagram zobrazuje jednotlivé třídy a vztah mezi nimi. Diagram aktivit zobrazuje průběh interakcí. Jelikož se jedná o interakce ve VR, je vhodné rozlišit a upřesnit aktivity na danou ruku. Barevné označení u diagramu aktivit představuje primární (modrá barva) a sekundární (červená barva) ruku.

### 8.2.1 Lukostřelba

**Luk** Luk představuje objekt s úchytným místem pro šíp a natahující tětivu. Atribut `transformaceVrubu` představuje pozici a rotaci, do kterého se přichytí šíp. Atribut `aktualniNataženíTětivy` představuje stav tětivy (0 - minimální natažení, 1 - maximální natažení).

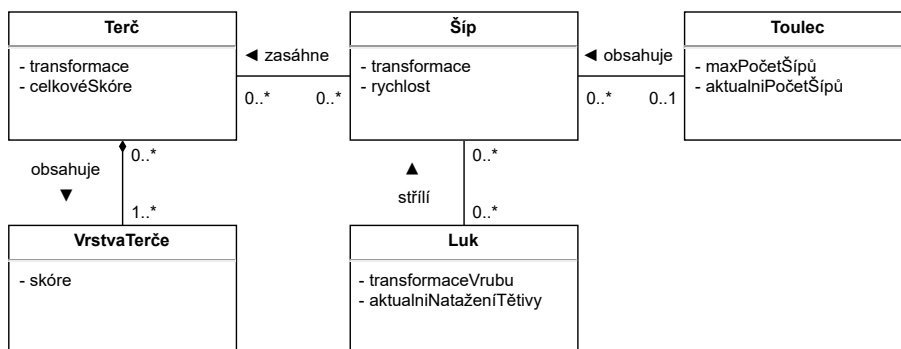
**Šíp** Šíp představuje objekt šípu. Atribut `transformace` představuje aktuální pozici a rotaci šípu. Atribut `rychlost` definuje rychlost šípu po vystřelení (zkombinována s natažeností tětivy).

**Toulec** Toulec představuje objekt se zdrojem pro šípy. Atribut `maxPočetŠípů` představuje maximální kapacitu toulce, při odebrání šípu se sníží. Atribut `aktualniPočetŠípů` představuje aktuální kapacitu.

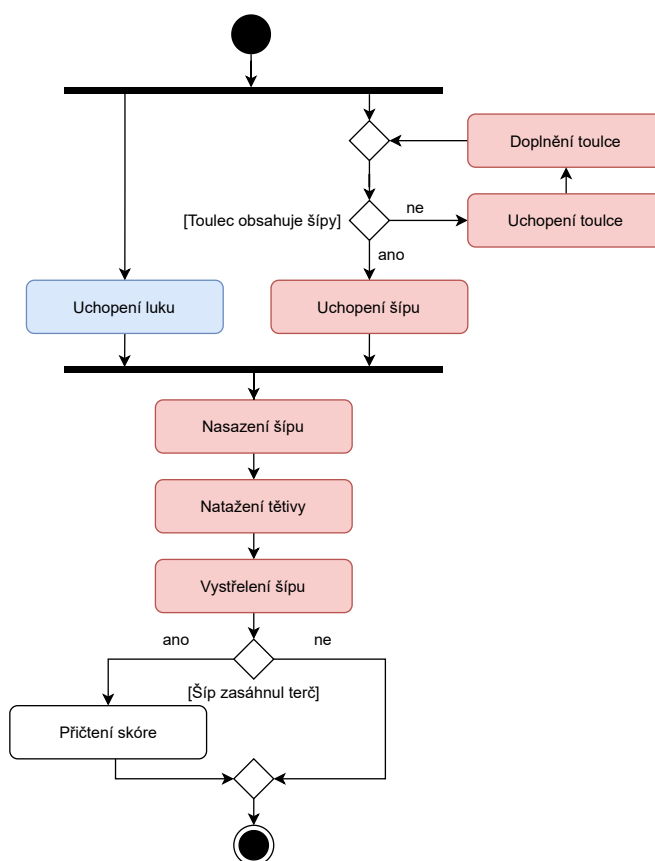
**Terč** Terč představuje zasáhnutelný objekt se speciální odezvou na šíp. Atribut `transformace` představuje pozici terče. Atribut `celkovéSkóre` představuje aktuální celkové skóre.

**VrstvaTerče** VrstvaTerče představuje vrstvu terče v terči se skórem. Atribut `skóre` představuje skóre vrstvy.

## 8.2. Model domény a aktivit



Obrázek 8.3: Lukostřelba - doménový diagram



Obrázek 8.4: Lukostřelba - diagram aktivit

### 8.2.2 Kování

**Kov** Kov představuje kovový objekt ke kování. Atribut **zahřátí** představuje míru zahřátí kovu (0 - minimální zahřátí, 1 - maximální zahřátí). Atribut **dokovanost** představuje míru dokovanosti kovu (0 - minimální dokovanost, 1 - maximální dokovanost).

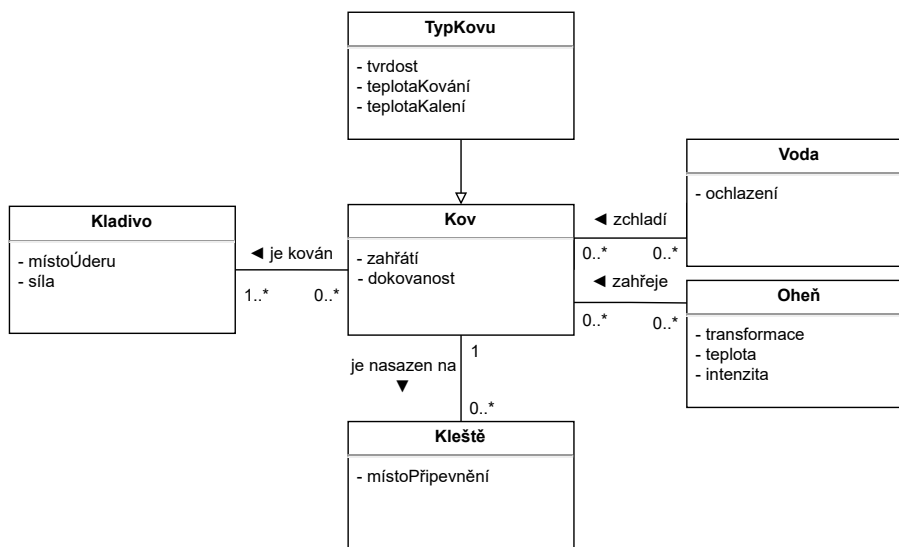
**TypKovu** TypKovu představuje typ kovu. Atribut **tvrdost** představuje tvrdost materiálu. Atribut **teplotaKování** představuje ideální teplotu zahřátí pro kování. Atribut **teplotaKalení** představuje ideální teplotu zahřátí pro kalení.

**Kladivo** Kladivo představuje objekt s deformační silou, který dokáže kovat a deformovat kov. Atribut **místoÚderu** představuje místo objektu, které reaguje deformační silou. Atribut **síla** představuje sílu úderu.

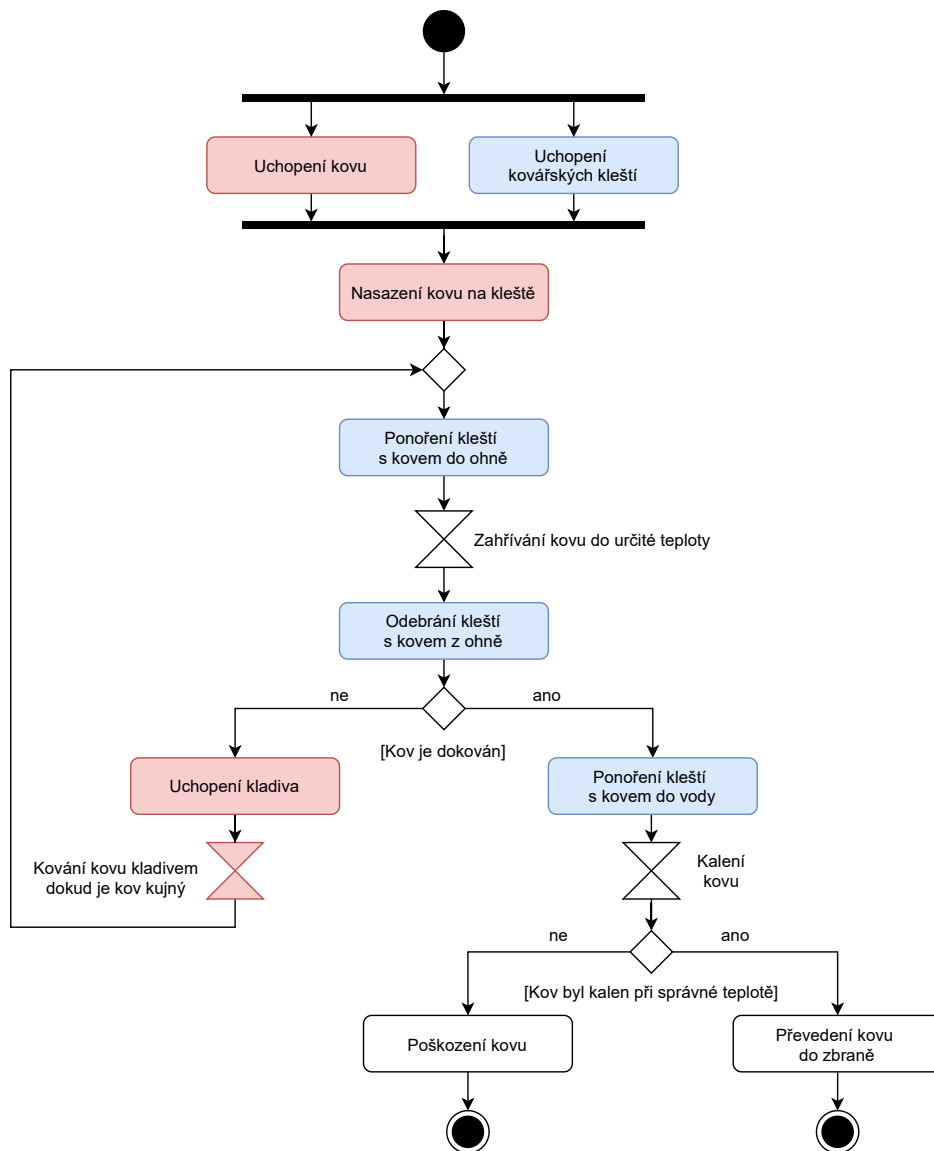
**Kleště** Kleště představují objekt, který připevní kov. Atribut **místoPřipevnění** představuje místo připevnění kovu.

**Voda** Voda představuje kalící prostředí pro kov. Atribut **ochlazení** představuje míru ochlazení kalícího prostředí.

**Oheň** Oheň představuje místo pro ohřátí kovu. Atribut **transformace** určuje pozici a rotaci ohně. Atribut **teplota** představuje teplotu ohně. Atribut **intenzita** představuje intenzitu světla ohně.



Obrázek 8.5: Kování - doménový diagram



Obrázek 8.6: Kování - diagram aktivit

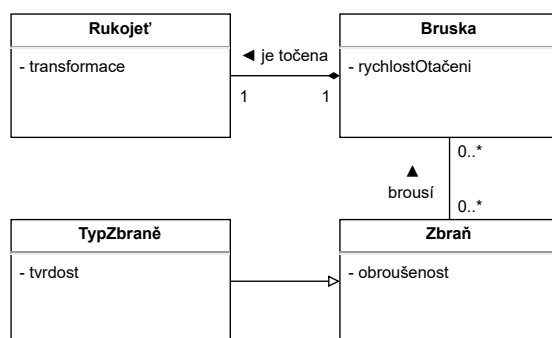
### 8.2.3 Broušení

**Bruska** Bruska představuje otáčecí kolo brusky. Atribut `rychlostOtáčení` představuje rychlost otáčení kola brusky.

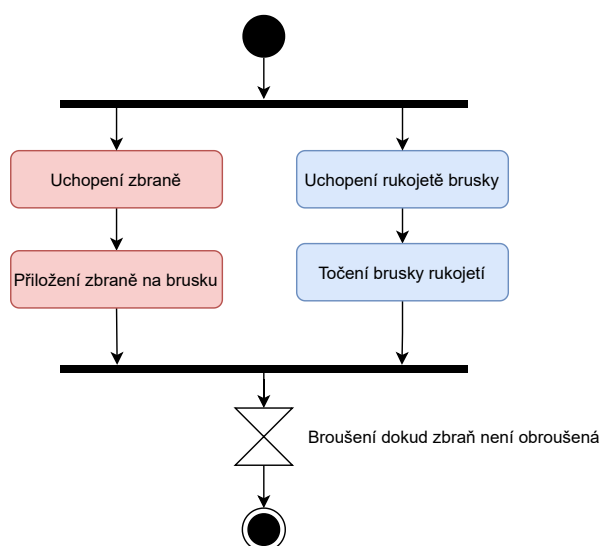
**Rukojeť** Rukojeť představuje objekt, pomocí kterého lze točit kolo brusky. Atribut `transformace` představuje pozici a rotaci rukojeti.

**Zbraň** Zbraň představuje objekt zbraně. Atribut `obroušenost` představuje míru obroušenosti zbraně (0 - minimální obroušenost, 1 - maximální obroušenost).

**TypZbraně** TypZbraně představuje typ zbraně. Atribut `tvrdost` představuje míru tvrdosti materiálu (převzato z Kovu).



Obrázek 8.7: Broušení - doménový diagram



Obrázek 8.8: Broušení - diagram aktivit

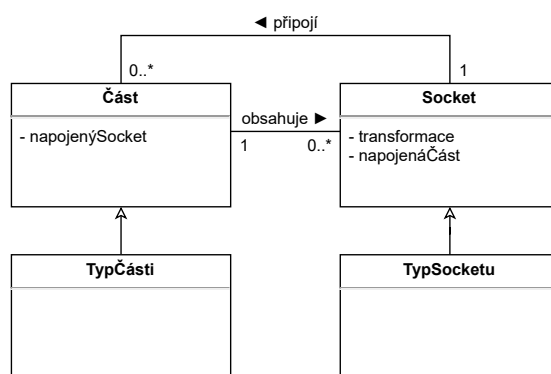
### 8.2.4 Oprava kola u vozíku

**Část** Část představuje část nesloženého kola. Atribut `napojenýSocket` určuje, který socket má aktuálně napojenou tuto část.

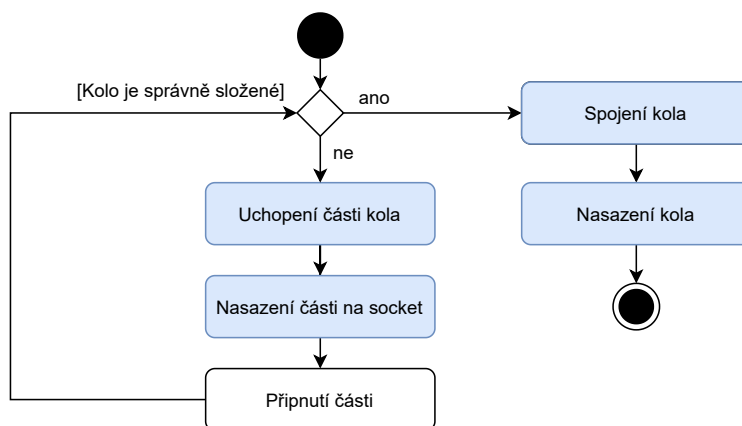
**TypČásti** TypČásti určuje typ části ke složení.

**Socket** Socket představuje napojovací objekt (místo) pro části kola. Atribut `transformace` určuje pozici a rotaci socketu. Atribut `napojenáČást` určuje, která část je aktuálně napojená na tento socket.

**TypSocketu** TypSocketu určuje typ socketu, na který lze připojit určité typy částí.



Obrázek 8.9: Oprava kola u vozíku - doménový diagram



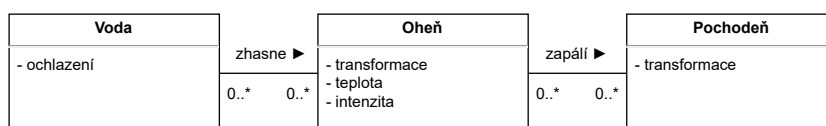
Obrázek 8.10: Oprava kola u vozíku - diagram aktivit

### 8.2.5 Pochodně

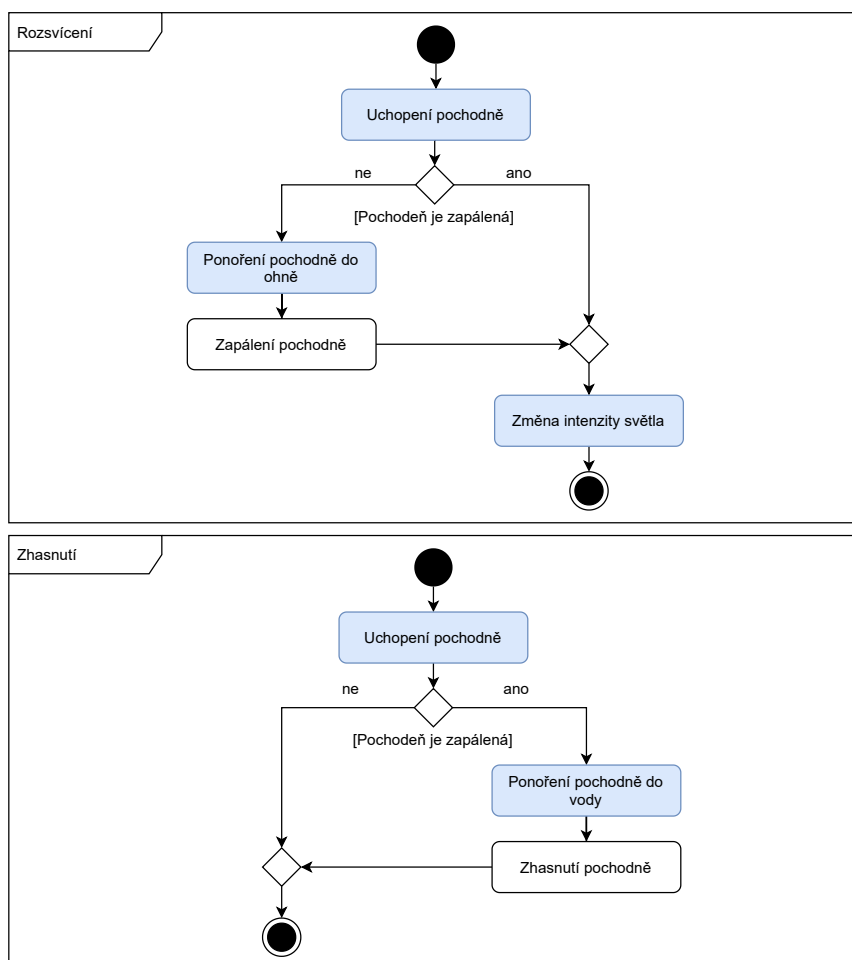
**Pochodeň** Pochodeň představuje objekt pochodně. Atribut **transformace** představuje pozici a rotaci pochodně.

**Oheň** Oheň představuje objekt, který zapálí pochodně (atributy v 8.2.2).

**Voda** Voda představuje prostředí pro uhašení ohně (atributy v 8.2.2).



Obrázek 8.11: Pochodně - doménový diagram



Obrázek 8.12: Pochodně - diagram aktivit



## 8.3 Použité technologie k vytvoření prototypu

Zde jsou vypsány použité technologie (software, balíčky aj.), které budou použity k vytvoření prototypu (a jeho návrhu).

**HTC Vive** [31] Pro vývoj a testování ve virtuální realitě jsem využila technologie HTC Vive s klasickými ovladači. Tlačítka na ovladači mají přiřazené akce definované pomocí Unity XR Interaction Toolkit.

**Unity** [32] Pro tento projekt jsem použila verzi Unity 2019.4.11f1. Jedná se o mojí druhou zkušenost v Unity, celkově třetí v odvětví těchto engineů, a proto je skvělou volbou pro implementaci prototypu mé práce.

**UnityXR Interaction Toolkit** [37] Tento projekt budu implementovat pomocí verze 1.0.0 tohoto balíčku.

**Blender** [60] Blender je open-source software pro modelování 3D modelů. Pro tento projekt jsem vytvořila pár jednoduchých modelů pro implementaci a testování interakcí.

**Visual Studio Code** [61] Visual Studio Code je volně dostupný editor vyvíjený společností Microsoft. Pro lepší implementaci jsem použila balíčky Unity Tools, Debugger for Unity a Unity Code Snippets. Propojují editor s projektem v Unity3D (např. automatickou kompletací nabízených funkcí, proměnných a tříd).

**Diagrams.net** [62] Pro tvorbu diagramů jsem využila volně dostupný online software diagrams.net (dříve známé pod doménou draw.io). Software nabízí jednoduchou a elegantní tvorbu diagramů. Proto je skvělou volbou pro tvorbu UML diagramů, které jsou obsaženy v návrhové části mé práce.



---

## Implementace prototypu

V této sekci se budu věnovat implementaci prototypu. Prototyp aplikace byl implementován pomocí prostředků uvedené v sekci 8.3. Skripty v Unity jsou napsány v jazyce C#. Hlavní účel prototypu je demonstrovat navržené interakce ve virtuální realitě pomocí základních principů.

### 9.1 Projekt

V projektu bylo využito několik volně dostupných balíčků a prostředků k vylepšení implementace. Pro implementaci ve VR byl využit balíček Unity XR Interaction Toolkit. Dále byly využity balíčky:

**URP** [63] Projekt aplikace byl vytvořen pomocí šablony pro URP (*Universal Render Pipeline*). Pomocí URP lze vytvořit aplikaci s optimalizovanými grafickými prostředky pro různé platformy.

**OpenVR Desktop** [64] OpenVR je SDK a API společnosti Valve, které umožňuje přístup k hardwaru VR. Ztráta oficiální podpory OpenVR v Unity je zaměřena na verze od 2019.3. Jelikož projekt využívá verze 2019.4, tak je tento balíček využit se zastaralým nastavením. OpenVR nahrazuje nový balíček OpenXR, který má podporu od verze 2020.2.

**XR Plugin Management** [65] Tento balíček slouží pro správu jednotlivých XR plug-inů (obrázek 4.1).

**TextMesh Pro** [66] Pro lepší kvalitu písma v UI byl využit tento balíček, který obsahuje pokročilé prostředky ke tvorbě písma v UI.

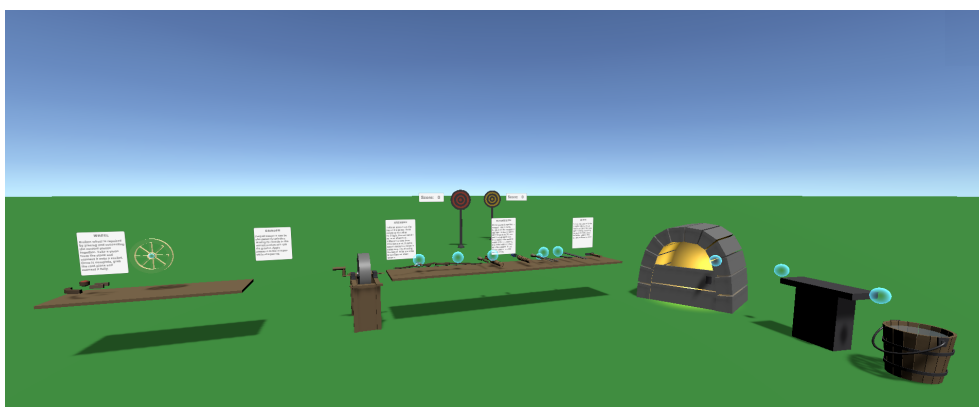
**Driven Decals** [67] Jelikož Unity zatím nepodporuje obtisky/potisky (*decals*) pro URP projekty, tak byl využit tento balíček.

**Shader Graph** [68] Tento balíček slouží pro tvorbu shaderů pomocí vizuálního grafu.

### 9.2 Scéna

Prototyp obsahuje jednu scénu, ve které je možné vyzkoušet všechny interakce. Scéna je velice jednoduchá a byla vytvořena ze základních modelů. Většina objektů byla vytvořena v Blenderu nebo ve samotném Unity. Další modely byly získány ze zdroje *Quaternius*, které jsou volně k dispozici [69]. Konkrétně balíčky *Survival* a *Modular Weapons*. Modely pro virtuální ruce byly využity z balíčku *OVR*. Model luku byl využit z balíčku *SteamVR*.

Objekty jsou dostupné na virtuálních stolech. Každá interakce obsahuje krátký informační panel, který stručně popisuje určitou interakci (obrázek 9.1).



Obrázek 9.1: Scéna prototypu s objekty interakcí

### 9.3 Integrace Unity XR ITK

V projektu je využit Action-based systém vstupů. Při importu balíčku do Unity je vhodné importovat i výchozí sety akcí (*Default Input Actions*) s namapovanými tlačítky na ovladače XR zařízení.

#### 9.3.1 XR Rig

Ve scéně je umístěn základní prefab kamery *Room-Scale XR Rig (Action-based)*. Prefab v základu obsahuje *XRRayInteractor* a *XRController (Action-based)* na oba ovladače. V projektu byl *XRRayInteractor* odstraněn a nahrazen interactorem *XRDirectInteractor* (přímá interakce, ruka) pro základní interakce s objekty.

### 9.3.2 XR Interaction Manager

XR Interaction Manager spravuje interakci mezi objekty *XRBaseInteractable* a *XRBaseInteractor* (obrázek 4.2). Objekt obsahuje komponenty *InputActionManager* a *XRInteractionManager*.

### 9.3.3 Teleport

Teleport byl vytvořen jako nový objekt *TeleportationController*. Podobně jako je struktura v XR Rigu, tak tento objekt obsahuje *XRRayInteractor* a *XRController* (Action-based). V komponentě *XRController* byla přenastavena akce *Select* na *TeleportSelect*. Objekt je umístěn do pravého ovladače a je aktivován/deaktivován v závislosti na stisku teleportační akce.

Objekt XR Rig obsahuje komponentu *Locomotion System* a *Teleportation Provider*, aby byl označen jako teleportační objekt ve scéně. Pro otáčení do stran obsahuje komponentu *Continous Turn Provider* (Action-based). Plocha, na kterou lze teleportovat obsahuje komponentu *TeleportationArea*.

## 9.4 Interakce

Implementace prototypu obsahuje všechny navržené interakce. V implementaci jsou navíc další minoritní interakce, které nebyly v původním návrhu jako součást interakcí, jen obohacují danou interakci a netvoří její hlavní účel (pouzdra, otevírací dveře, zapálení dalších objektů).

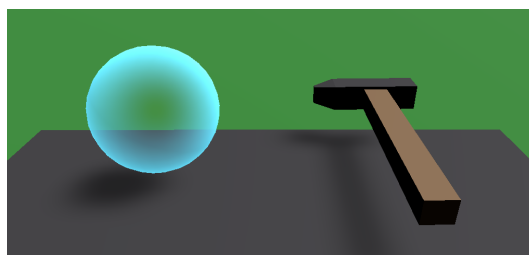
Objekty, s kterými lze interagovat dědí z komponenty *XRBaseInteractable*. Nejčastější forma této třídy je *XRGrabInteractable*, která umožňuje uchopení předmětu jakýmkoliv interaktorem. Ruce obsahují komponentu *XRDirectInteractor* a sockety obsahují *XRSocketInteractor*.

### 9.4.1 Sockety

Scéna obsahuje tzv. *sockety*, které umožňují automaticky přichytit volný objekt do určité pozice a rotace. Výhoda socketů je fixní upevnění objektu, tedy objekt se jednoduše „neztrácí“ ve scéně při manipulaci s rukama.

Sockety jsou vyznačeny modrou bublinou pokud neobsahují momentálně žádný objekt. Při přiblížení předmětu k socketu je barva socketu změněna na zelenou.

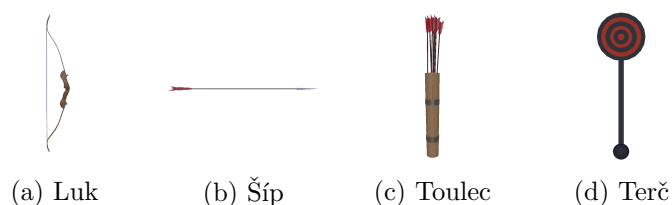
**Pouzdra** Uživatel má na „zádech“ připnutý dva tyto sockety (levý a pravý), které slouží jako pouzdra pro vybrané objekty (např. toulec nebo zbraň - obrázek 9.2).



Obrázek 9.2: Prázdný socket a socket obsaující objekt

### 9.4.2 Lukostřelba

Interakce lukostřelby obsahuje objekty luku, šípu, toulce a terče (obrázek 9.3, dle návrhu v sekci 8.2.1). Implementace této interakce byla inspirována projektem lukostřelby optimalizovaného pro novou verzi Unity XR Interaction Toolkit [70].



Obrázek 9.3: Objekty lukostřelby

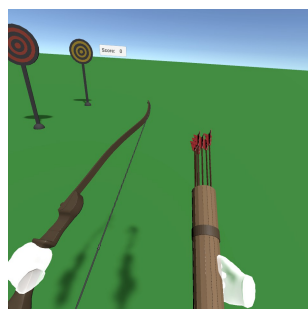
**Luk** Hlavní část luku tvoří tělo, za který se luk uchopí. Tento objekt obsahuje skript *Bow*, který dědí ze třídy *XRGrabInteractable*. Při uchopení luku je zavolána přetížená metoda *OnSelectEntered*, umožní připevnit šíp do interactoru *Notch*.

**Vrub** Vrub je část tětiny luku, na kterou se připevní šíp. Tento objekt obsahuje skript *Notch*, který dědí ze třídy *XRSocketInteractor*. Interactor automaticky připevní šíp do socketu v dané pozici a rotaci při jeho přiblížení. Toto obstarává přetížená metoda *CanSelect*. Objekt dále obsahuje skript *Pull*, který dědí ze třídy *XRBaseInteractable*. V přetížené metodě *ProcessInteractable* se aktualizuje pozice šípu a animace tětiny v závislosti na natažení tětiny. Animace tětiny je řízena animátorem se stavem *Blend Tree*, který mixuje animaci tětiny mezi 0 (žádné natažení) a 1 (plné natažení). Tato hodnota je počítána v metodě *CalculatePull*. Natažení tětiny vyvolá haptiky na daném ovladači.

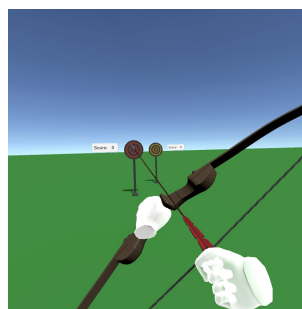
**Šíp** Šíp obsahuje skript *Arrow*, který dědí ze třídy *XRGrabInteractable*. Při puštění šípu je zavolána přetížená metoda *OnSelectExited*, která šíp vypustí ze socketu a zároveň je mu dodána síla pomocí metody *Rigidbody.AddForce*. V přetížené metodě *ProcessInteractable* je kontrolována kolize šípu s jakýmkoliv *Colliderem* a aktualizována poslední pozice špičky šípu. Tato pozice je využita v metodě *CheckCollision*, kde je použit *Physics.Linecast*. Paprsek mezi aktuální a poslední pozicí špičky zaznamená případnou kolizi. Při kolizi je zavolána metoda *Hit* rozhraní *IHittable* (pokud daný objekt toto rozhraní implementuje).

**Toulec** Toulec obsaue skript *Quiver*, který dědí ze třídy *XRGrabInteractable*. Objekt obsahuje metodu *Fill*, která doplní toulec. Tato metoda je zavolána při vyvolání akce na tomto objektu. Součástí toulce je prázdný objekt umístěný na pozici, v které je brán šíp z toulce. Tento objekt obsahuje skript *ArrowSpawner*, který dědí ze třídy *XRBaseInteractable*. Při uchopení tohoto objektu je vytvořen nový objekt šípu. Šíp je automaticky připevněn na interagující ruku.

**Terč** Terč obsahuje skript *Target*, který implementuje rozhraní *IHittable*. Rozhraní obsahuje metodu *Hit*, která je vyvolána při zasažení. Terč se skládá z vrstev (pro každý skóre), každá vrstva obsahuje skript *TargetLayer*. Při kolizi se šípem je přičteno dané skóre zasažených vrstev pomocí metody *Score* v rozhraní *IHittable*. Dolní část terče je objekt tyče, který obsahuje skript *XRGrabInteractable*. Při aktivaci tohoto objektu je vymazáno skóre tohoto terče, které je zobrazeno na vedlejším panelu. Objekt obsahuje komponentu *HingeJoint*, která obsahuje závislé *Rigidbody* objektu kulatého terče. Tato komponenta zajišťuje charakter pružiny v terči.



(a) Uchopení luku a toulce

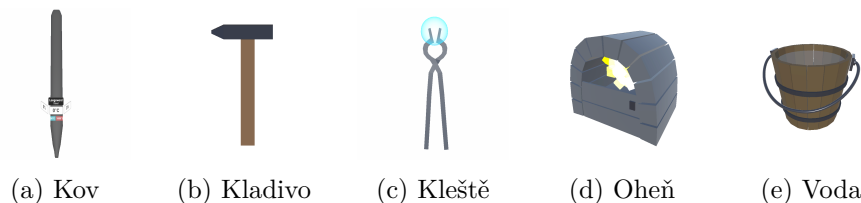


(b) Nasazení šípu

Obrázek 9.4: Lukostřelba

### 9.4.3 Kování

Interakce kování obsahuje objekty kovu, pece s ohněm, vědra s vodou, kleščí a kladiva (obrázek 9.5, dle návrhu v sekci 8.2.2).



Obrázek 9.5: Objekty kování

**Kov** Kov obsahuje skript *Metal*, který implementuje rozhraní *IDeformable*, *IHeatable* a *ICoolable*. Tyto rozhraní implementují dané metody, které se vyvolají při interakci s tímto objektem. Při kolizi s kladivem je kov deformován (*IDeformable*), při kolizi s ohněm je kov zahříván (*IHeatable*), při kolizi s vodou je kov kalen (*ICoolable*). Zahřívání, chladnutí a kalení je prováděno pomocí korutin (*Coroutine*).

Charakter kovu je popsán skriptem *MetalCharacter*, který dědí ze třídy *ScriptableObject*. Pomocí této šablony lze v editoru vytvořit několik objektů s různými parametry kovu (druh zbraně, druh kovu, obroušenost, zahřátí, dokovanost, tvrdost, rychlost ohřívání, průměrná teplota kování a kalení). Součástí kovu je informační panel, který obsahuje informace o aktuální teplotě (uprostřed), průměrnou teplotu kování (červený panel), průměrnou teplotu kalení (modrý panel), dokování (levá část) a obroušení (pravá část) v %. Informační panel je zobrazen při vyvolání akce na tomto objektu.

Další komponentou objektu je *MeshDeform*, který obstarává jednoduchou deformaci meshu při běhu aplikace pomocí metody *AddDeformForce*. Deformace je určena opačným směrem normály vertexu, na které zapůsobila deformační síla kladiva. Poměr dokovanosti je určen vynaloženou silou kladiva a je postupně sčítán.

Komponenta *MaterialDynamic* obstarává aktuální stav materiálu kovu. Materiál kovu je vytvořen pomocí speciálního shaderu, který mění barvu materiálu v závislosti na zahřátí. Tento shader byl vytvořen pomocí nástroje *ShaderGraph*. Materiál obsahuje parametry *Power* (0 - nejnižší zahřátí, 1 - nejvyšší zahřátí) a *Rim color* (barva Fresnelového efektu).

**Kladivo** Kladivo obsahuje skript *Hammer*, který dědí ze třídy *XRGrabInteractable*. Tento objekt obsahuje druhou komponentu *Collider* v místě úderu kladiva. Tento *Collider* určuje a zaznamenává kolize s objekty charakteru *IDeformable* a zavolá příslušnou metodu *Deform*. Deformace je závislá na rychlosti komponenty *Rigidbody* a je provedena na nejbližším vertexu zasaženého ob-



jektu charakteru *IDeformable*. Tento bod je nalezen pomocí kontaktních bodů třídy *ContactPoint*, které jsou k dispozici v objektu *Collision* při kolizi. Při kolizi jsou vyvolány haptiky na příslušném ovladači.

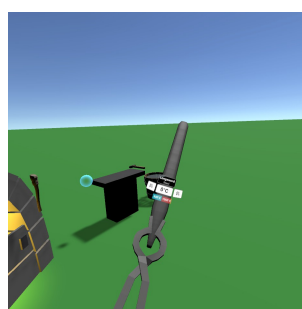
**Kleště** Kleště obsahují skript *XRGrabInteractable*. Jako součást objektu je socket, který umožňuje připevnění kovu. Aby nedocházelo ke kolizím mezi objektem v socketu a kleštěma, jsou oba objekty umístěny do odlišných vrstev (*Layer*). Následné vypnutí fyziky a kolize mezi těmito maskami je možné v nastavení projektu v záložce *Physics* (matice kolizí vrstev).

**Oheň** Oheň v peci obsahuje skript *Fire*, který implementuje rozhraní *ICoolable*. Oheň má danou maximální teplotu, do které je možné kov zahřát. Při kolizi s objektem, který implementuje rozhraní *IHeatable* je na daném objektu zavolána metoda tohoto rozhraní *Heat*.

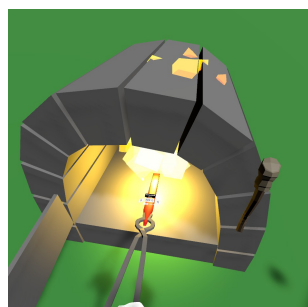
**Voda** Voda obsahuje skript *Water*. Objekt obsahuje komponentu *Collider*, který je charakteru *Trigger*. V metodě *OnTriggerEnter* je kontrolována kolize s objekty, které implementují rozhraní *ICoolable*. Při kolizi je zavolána metoda *Cool* tohoto objektu. Při kolizi s kovem je provedeno kalení.



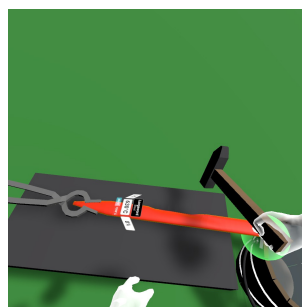
(a) Informační panel



(b) Kov v kleštích



(c) Ohřívání kovu

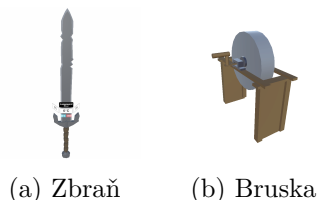


(d) Kování kovu

Obrázek 9.6: Kování

#### 9.4.4 Broušení

Interakce broušení obsahuje objekty zbraně, kola brusky a rukojeti (obrázek 9.7, dle návrhu v sekci 8.2.3).



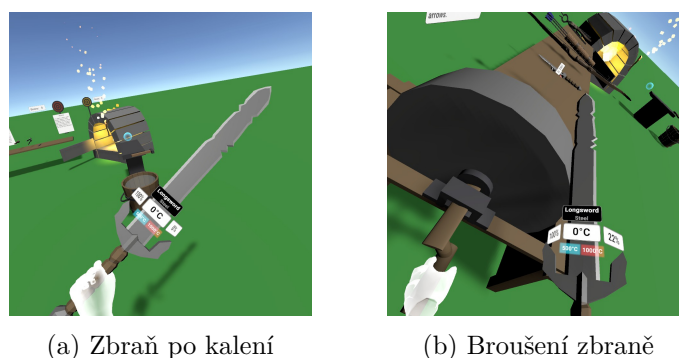
Obrázek 9.7: Objekty broušení

**Kolo brusky** Kolo brusky obsahuje skript *Grinder*. Tento skript zaznamenává kolizi s objekty, které implementují rozhraní *IGrindable* a následně vyvolají metody *Grind* na těchto objektech. Toto je uskutečněno pomocí metody *OnTriggerStay*. Kolo brusky obsahuje komponentu *Collider*, který je charakteru *Trigger*. Důvodem je plynulejší kolize objektu s kolem brusky.

Komponenta *Rigidbody* má zapnutou gravitaci a vypnutou kinematiku. Další komponentou tohoto objektu je komponenta *Hinge Joint*, která má nastavenou vlastní osu otáčení.

**Rukojeť** Rukojeť obsahuje skript *XRGrabInteractable*. Důležitá část je nastavení parametru *Movement Type* na *Velocity Tracking*. Toto indikuje pohyb objektu pomocí rychlosti komponenty *Rigidbody*. Komponenta *Rigidbody* má zapnutou gravitaci i kinematiku. Další komponentou objektu je komponenta *Fixed Joint*. Parametr *Connected Body* je nastaven na objekt kola brusky (jeho komponenta *Rigidbody*). Díky této komponentě a nastavení parametru *Movement Type* v *XRGrabInteractable* je uskutečněno točení kola brusky pomocí rukojeti.

**Zbraň** Zbraň obsahuje skript *Weapon*, který dědí ze třídy *XRGrabInteractable* a implementuje rozhraní *IGrindable*. Zbraň je vytvořena správným překováním kovu a tímto procesem zdědí všechny parametry kovu. Metoda *Grind* rozhraní *IGrindable* brousí zbraň zvýšením ostrosti zbraně v závislosti na rychlosti točící brusky. Dále závisí na parametru tvrdosti kovu. Broušení zbraně vyvolá haptiky na daném ovladači. Síla haptik je dána rychlostí točící brusky.



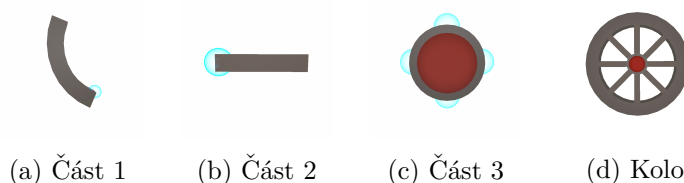
(a) Zbraň po kalení

(b) Broušení zbraně

Obrázek 9.8: Broušení

### 9.4.5 Oprava kola u vozíku

Interakce opravy kola obsahuje objekty částí kola (čtyři různé části) a nekonečného balíku těchto částí (obrázek 9.9, dle návrhu v sekci 8.2.4).



(a) Část 1

(b) Část 2

(c) Část 3

(d) Kolo

Obrázek 9.9: Objekty opravy kola

**Balík částí** Část kola lze získat z nekonečného balíku částí. Pro každou část je každý balík. Balík je vizualizován samostatnou částí. Tento objekt obsahuje skript *SimpleSpawner*, který dědí ze třídy *XRBaseInteractable*. Při uchopení tohoto předmětu je vytvořen nový objekt dané části a automaticky uchopen do interagující ruky (podobně jako u skriptu *ArrowSpawner*).

**Část kola** Část kola se skládá ze samostatné části a přípeňovacího socketu. Samostatná část je charakterizována skriptem *WheelPart*, který dědí ze třídy *XRGrabInteractable*. Pro kolo jsou k dispozici čtyři různé druhy částí, každá z nich obsahuje vlastní skript dědící ze třídy *WheelPart*. Důvodem těchto samostatných skriptů je rozpoznání typu třídy při napojování částí. Části nemají zapnutou gravitaci pro lepší manipulaci se skládačkou. Správné natočení části je umožněno pomocí parametru *Attach Transform* v komponentě *XRGrabInteractable*.

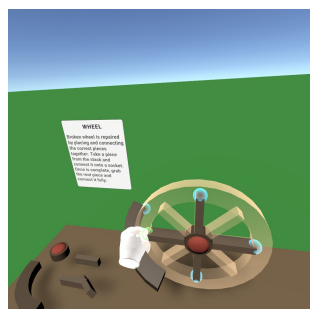
Druhou částí je přípeňovací socket. Socket obsahuje skript *WheelSocket*, který dědí ze třídy *XRSocketInteractor*. V přetížené metodě *CanSelect* je omezeno, jakou komponentu interactable může socket přípevnit. Podobně je tomu v metodě *CanHover*. Jednotlivé části a sockety obsahují vlastní skripty

## 9. IMPLEMENTACE PROTOTYPU

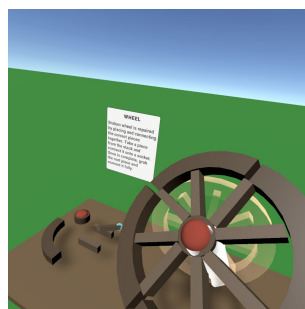
---

dědicí ze třídy *WheelSocket*. Každý typ může na sebe napojit určitý typ části skládačky.

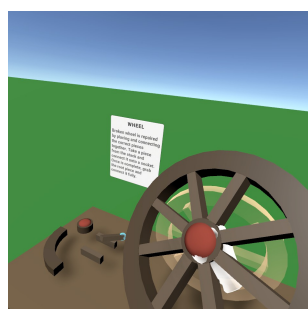
*WheelRootPart* navíc obsahuje metodu *TryConnect*, která je zavolána při aktivaci objektu. Tato metoda rekurzivně kontroluje připojené části kola, zda jsou správně napojeny a jejich počet. Pokud je kolo správně složeno, tak se promění do složeného modelu a všechny části jsou zničeny.



(a) Skládání kola



(b) Složené kolo



(c) Přeměněné kolo

Obrázek 9.10: Oprava kola u vozíku

### 9.4.6 Pochodně

Interakce pochodní obsahují objekty pochodně, ohně, vody a dalších jednoduchých zapalitelných objektů (obrázek 9.11, dle návrhu v sekci 8.2.5).



(a) Pochodeň



(b) Oheň



(c) Voda

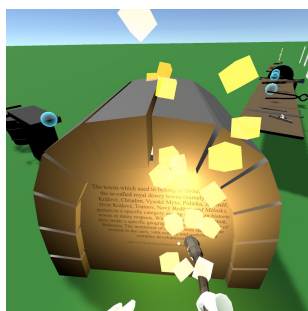
Obrázek 9.11: Objekty pochodně

**Pochodeň** Pochodeň obsahuje skript *Torch*, který dědí ze třídy *XRGrabInteractable* a implementuje rozhraní *IFireable*. Metoda *Ignite* tohoto rozhraní vytvoří nový objekt ohně na objektu pochodně. Aktivace pochodně zavolá metodu *IncreaseLight*, která zvýší intenzitu světla objektu ohně. Podobně deaktivace pochodně zavolá metodu *DecreaseLight*.

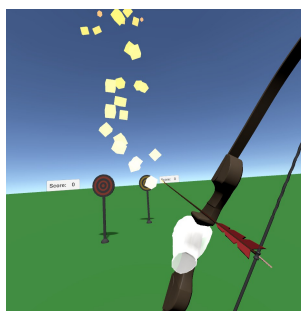
**Oheň** Oheň je prázdný objekt bez modelu. Objekt obsahuje komponentu *Collider*, který je charakteru *Trigger*. Obsahuje skript *Fire*, který implementuje rozhraní *ICoolable*. Metoda *Cool* tohoto rozhraní zničí tento objekt při kolizi s daným objektem (voda). Při kolizi s objektem, který implementuje rozhraní *IFireable* je na tomto objektu zavolána metoda *Ignite*. Tato metoda vytvoří nový objekt ohně a zkopíruje jeho parametry (teplota). Další komponentou objektu je *ParticleSystem*, který vytváří jednoduchou vizualizaci a efekt ohně. Komponenta světla *Light* je typu *Point* (bodové světlo).

**Voda** Voda obsahuje skript *Water*. Objekt obsahuje komponentu *Collider*, který je charakteru *Trigger*. V události *OnTriggerEnter* je kontrolována kolize s objekty, které implementují rozhraní *ICoolable*. Při kolizi je zavolána metoda *Cool* tohoto objektu. Při kolizi s ohněm je provedeno uhašení ohně.

**Zapalitelný objekt** Jednoduché zapalitelné objekty obsahují skript *SimpleFireableObject*, který implementuje rozhraní *IFireable*. Při kolizi s objektem, který implementuje rozhraní *IFireable* je na tomto objektu zavolána metoda *Ignite*. Tato metoda vytvoří nový objekt ohně a zkopíruje jeho parametry (teplota). Tento skript může být komponentou jakéhokoliv objektu (např. dřevěné šípky).



(a) Rozsvícená pochodeň



(b) Zapalitelný objekt

Obrázek 9.12: Pochodně

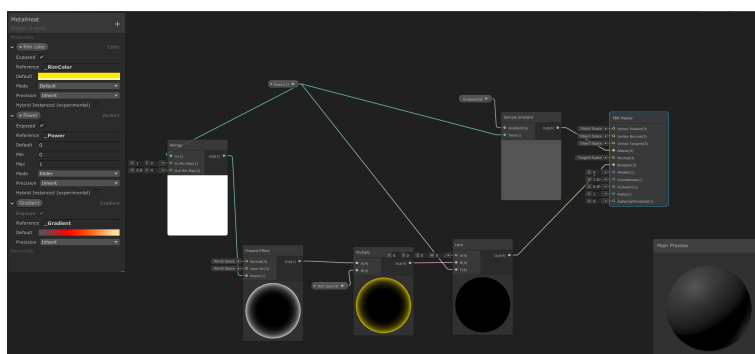
## 9.5 Implementační zajímavosti

V této sekci zmíním implementační zajímavosti a dodatky k interakcím.

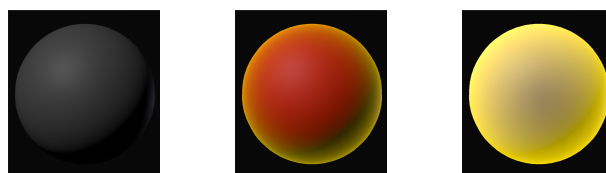
### 9.5.1 Materiál kovu

Materiál kovu využívá vlastní shader vytvořený pomocí nástroje *Shader Graph*. Tento nástroj umožňuje tvorbu shaderů pomocí vizuálního programování. Tento shader je jednoduchý, avšak lze vytvořit i komplexnější shadery. Hlavní účel bylo vytvoření dynamického shaderu, který bude závislý na parametrech. Tyto parametry lze měnit přímo v kódu.

Vytvořený shader obsahuje záři *Fresnelového efektu* jako emisní složku. Barva záře je řízena parametrem *Rim color*. Barva jednoduchého materiálu je řízena parametrem *Gradient*. Síla gradientu a záře je řízena parametrem *Power* (obrázek 9.13). Při využití vytvořeného shaderu na více objektech je nutné vytvořit a přiřadit vlastní instanci materiálu v kódu (ne v inspektoru). Vytvořený materiál lze vidět na obrázku 9.14.



Obrázek 9.13: Vizuální programování shaderu pomocí nástroje Shader Graph.

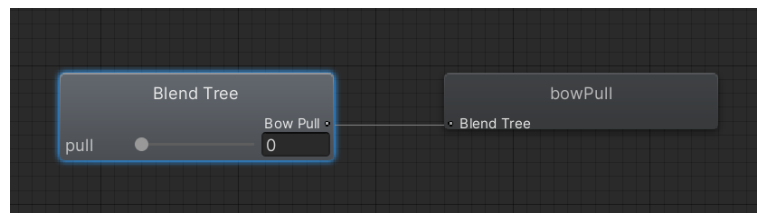


(a) Zahřátí 0%      (b) Zahřátí 50%      (c) Zahřátí 100%

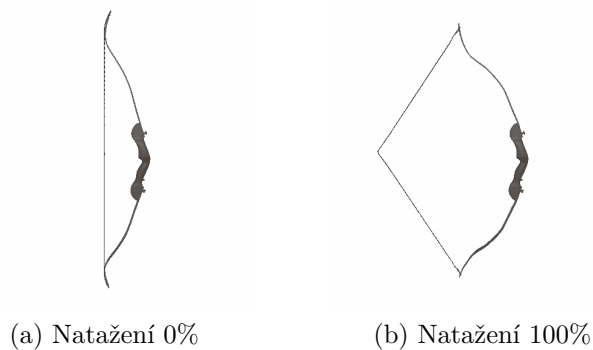
Obrázek 9.14: Materiál vytvořený pomocí nástroje Shader Graph pro zahřívání kovu.

### 9.5.2 Animace luku

Model a samostatná animace luku je z balíčku SteamVR. Objekt obsahuje komponentu *Skinned Mesh Renderer*. Mesh modelu je transformován v animaci dle skeletonu, který je tvořen z kostí (*bones*) meshe. Objekt obsahuje komponentu *Animator*, která má animator obsahující jeden stav charakteru *Blend Tree* (obrázek 9.15). Parametr animatoru *pull* určuje průběh animace luku. Tento parametr lze měnit přímo v kódu. Animace luku lze vidět na obrázku 9.16.



Obrázek 9.15: Stav Blend Tree v animatoru luku s animací

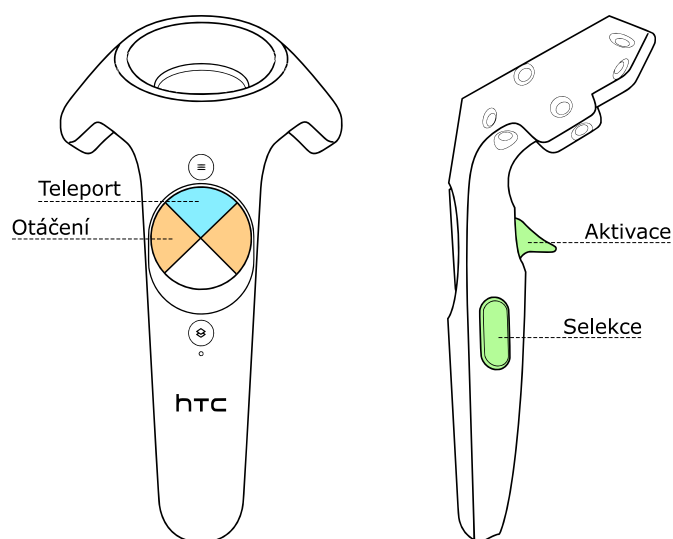


Obrázek 9.16: Animace pomocí komponent *Skinned Mesh Renderer* a *Animator*

## 9.6 Ovládání

Díky balíčku Unity XR Interaction Toolkit lze jednoduše vytvořit spustitelné verze pro většinu VR platforem (Oculus, OpenVR, WMR). Jednotlivé ovladače pro HMD se sice liší, ale akce jsou namapovány pomocí výchozích setů akcí automaticky. Jelikož aplikace byla implementována a testována pomocí HTC Vive, tak uvedu rozložení akcí pro klasické ovladače tohoto HMD (obrázek 9.17).

Akce *aktivace* je dostupná pouze pokud s daným objektem *XRGrabInteractable* interaguje *XRDirectInteractor* nebo *XRRayInteractor*. Každý objekt definuje vlastní aktivační a deaktivaci akci a určuje tím, jaká funkce se v daný moment má vyvolat. Ovládání prototypu je zobrazeno na obrázku [71].



Obrázek 9.17: Ovládání prototypu [71]



---

# Uživatelské testování

Testování prototypu bude zaměřeno na testování implementovaných interakcí. Z testů by se mělo zjistit, zda jsou interakce vhodně navržené pro danou skupinu uživatelů.

## 10.1 Popis testování

Testování bude nahráváno a probíhat v prostoru fakulty (*Usability Lab*). Účastníci obdrží podkladové materiály (dotazníky, instrukce pro testera). Moderátor testování bude navigovat účastníky podle testovacích scénářů.

## 10.2 Podkladové materiály

Součástí testování budou testovací dotazníky a testovací scénáře.

### 10.2.1 Dotazník

Účastníci testování obdrží výstupní dotazník. Výstupní dotazník bude sloužit jako zpětná vazba testovaného uživatele k návrhu interakcí. Dotazník obsahuje otázky s bodovým ohodnocením na škále od 1 do 5 (1 - nejméně bodů, 5 - nejvíce bodů) a slovní hodnocení pro každou interakci.

#### Otázky

- Je podle Vás interakce vhodná do historického města/prostředí?
- Je podle Vás interakce vhodná pro provedení ve VR?
- Má podle Vás interakce vhodně navržené uživatelské rozhraní ve VR?
- Je podle Vás interakce zábavná pro uživatele široké veřejnosti?
- Slovní hodnocení/vylepšení interakce

### 10.2.2 Testovací scénáře

Účastníci testování obdrží testovací scénáře. Každá interakce bude mít vlastní testovací scénář. Testovací scénář obsahuje informace testované interakce (čas, popis, instrukce pro testera, očekávané kroky).

Zde je uveden testovací scénář pro interakci kování. Ostatní scénáře mají stejnou strukturu a jsou obsaženy jako příloha této práce.

#### 10.2.2.1 Kování

##### **Odhadovaný čas**

5 min.

##### **Účel testování**

Testujeme interakci kování a schopnost uživatele úspěšně dokovat kovový objekt do zbraně.

##### **Počáteční bod**

Uživatel má prázdné ruce. Kov je připravený ke kování a nebyl poškozen.

##### **Koncový bod**

Kov byl překován do zbraně.

##### **Instrukce pro testera**

Zobrazte informační panel na kovu. Připevněte kovový objekt na kovářské kleště se zapnutým informačním panelem. Kovejte kov do příslušné teploty. Po úplném dokování proveďte kalení kovu. V případě neúspěchu zkuste znovu s jiným kovovým objektem.

##### **Očekávané kroky**

1. Uchopení kovu do jedné ruky.
2. Zobrazení informačního panelu tlačítkem aktivovat.
3. Uchopení kleští do druhé ruky.
4. Připevnění kovu na kleště při držení tlačítka aktivace na kovu.
5. Kování kovu dokud není okován na 100%.
  - a) Ponoření kleští s kovem do středu ohně a ohřívání do kovací teploty.
  - b) Uchopení kladiva do druhé ruky.
  - c) Připevnění/odložení kleští s kovem do socketu.
  - d) Kování kovu kladivem.

6. Kalení kovu.

- a) Ponoření kleští s kovem do středu ohně a ohřívání do kalící teploty.
- b) Ponoření kovu do kalící kapaliny (vody).

7. Odebrání dokované zbraně z kleští.

## 10.3 Výsledky testování

Prototyp byl otestován pomocí zařízení HTC Vive Pro s klasickými ovladači. Testování se účastnili dva testeři. Komunikace mezi testery a moderátorem byla uskutečněna vzdáleně. Testeři neměli žádnou či minimální zkušenost s technologií virtuální reality.

### 10.3.1 Pozorování

Testování bylo nahráváno, lze tedy vyhodnotit postupy testerů. Následně budou porovnány výstupy z dotazníků. Pomocí těchto informací lze vyhodnotit navržené interakce a následně zhodnotit případné úpravy.

#### 10.3.1.1 Student 1

##### **Lukostřelba**

Nepodařilo se brání šípu z toulce. Nepodařilo se uchopení toulce a zpětné nasazení. Podařilo se střelení na terč.

##### **Kování**

Nepodařilo zahřívání v ohni, pomocí instrukcí se podařilo. Podařilo se připevnění kovu na kleště s panelem. Podařilo se zahřívání a kování kovu.

##### **Broušení**

Pochopil interakci broušení.

##### **Oprava kola u vozíku**

Pochopil systém skládání kola.

##### **Pochodně**

Pochopil interakci pochodní.

#### 10.3.1.2 Student 2

##### **Lukostřelba**

Podařilo se brání šípu z toulce. Podařilo se uchopení toulce a zpětné nasazení. Podařilo se střelení na terč.

### **Kování**

Podářilo se připevnění kovu na kleště s panelem. Podářilo se zahřívát a kovat kov.

### **Broušení**

Pochopila interakci broušení.

### **Oprava kola u vozíku**

Nedařilo se pochopit systém skládání kola, s pomocí instrukcí se podařilo.

### **Pochodeň**

Pochopila interakci pochodní.

### **10.3.2 Dotazník**

Dotazník obsahoval navržené otázky (sekce 10.2.1). Výsledky byly celkově pozitivní, nejčastější problémy se vyskytovaly v oblasti uživatelského rozhraní ve virtuální realitě (tabulka 10.1). Slovní hodnocení nabízelo potenciální vylepšení do budoucích implementací nebo kritizovalo nalezené problémy (sekce 10.3.3). Ostatní otázky byly ohodnoceny na plný počet bodů.

	Lukostřelba	Kování	Broušení	Oprava kola	Pochodně
Student 1	3	5	5	3	4
Student 2	5	3	5	4	5

Tabulka 10.1: Odpovědi dotazníku na otázku vhodně navrženého uživatelského rozhraní

### **10.3.3 Problémy**

Během testování se objevila problematika, která způsobovala občasné propadání objektů do země po upuštění objektů. Terč občasně nezaznamenává vystřelený šíp a tím nezapočte body. Z testování vyplynuly obecné pozorování, které by jednoznačně prototyp vylepšily:

#### **Ovládání**

Tester musí být s ovládáním předem dobře obeznámen. Bylo by tedy vhodné přidat návod přímo do aplikace (aktuálně není žádný návod ovládání v aplikaci).

#### **Teleport**

Samostatné teleportování proběhlo v pořádku. Avšak bylo by vhodné umožnit ovládání teleportu na obou ovladačích (aktuálně je teleport umožněn pouze na pravém).

### **Uchopení předmětu**

S uchopováním předmětů nebyl problém. Možné komfortní vylepšení je umožnit přichycení objektu a při opětovném stisknutí upuštění (aktuálně držení tlačítka indikuje držení předmětu).

### **Lukostřelba**

Tester musí být obeznámen, že na jeho „zádech“ se nachází toulec. Braní šípu z toulce je složitější aktivita, neboť tester nevidí na tuto akci. Také záleží na schopnosti testera tento objekt uchopit z této oblasti. Bylo by vhodné, aby uživatel nebyl přímo závislý na toulci v zádech a tím umožnit i alternativní variantu (aktuálně je možné braní šípu pouze z toulce na zádech).

### **Kování**

Oblast zahřívání je malá a její nalezení je tím komplikované. Kov začíná rychle chladnout a tester často nestihnul tento kov kovat. Tester s procesem kování není obeznámen a musí se mu vysvětlit proces kování a kalení. Tato realita nesouvisí se samotným návrhem mechanik, celkově spíše se znalostí kování. Tato interakce by tedy vyžadovala detailnější vysvětlení přímo v aplikaci (aktuálně je aktivita vysvětlena bez detailů).

### **Broušení**

Interakce proběhla během testování v pořádku. Potenciální vylepšení se týká správného úhlu při broušení zbraně na brusce (aktuálně nezávisí na úhlu zbraně).

### **Oprava kola u vozíku**

Systém skládání kola není jednoznačný na první pohled. Tester byl zmaten výběru dvou skoro identických částí, avšak při pomoci a vysvětlení pochopil jejich rozdíl. Tato skutečnost by z pohledu uživatele mohla tvořit problémy. Bylo by vhodné lépe indikovat systém spojování (aktuální systém tvoří problémy z pohledu uživatelského rozhraní).

### **Pochodně**

Tato interakce není v prototypu rozvinutá a je velice jednoduchá. Pro budoucí návrhy by bylo vhodné propojit tuto interakci s více objekty a vytvořit tím aktivitu, která bude mít více účelů (aktuálně slouží jako zdroj světla a přirozeně reaguje na prostředí ohně a vody).

## **10.3.4 Implementované vylepšení**

Kvůli nedostatku časových prostředků nebyly všechny problémy vyřešeny v rámci implementace. Základní problémy byly upraveny a vylepšeny. Jedná se o tato řešení:

### **Ovládání**

Byl přidán návod k ovládání, návod se zobrazí při puštění aplikace před uživatelem.

### **Teleport**

Teleport byl implementován na obou ovladačích.

### **Uchopení předmětu**

Uchopitelné předměty je nyní možné uchopit pomocí tlačítka selekce. Upuštění je umožněno opětovným stisknutím tohoto tlačítka. Tato změna nebyla aplikována na objekty se speciální interakcí (tětiva, rukojeť, dveře od pece).

### **Lukostřelba**

Toulec byl umístěn na vhodné místo odkud uživatel střílí na terč. Toulec na zádech je stále aktivní a je možné ho použít.

### **Kování**

Byl dodán informační panel detailně popisující kov. Oblast zahřívání je nyní rozsáhlejší a tím zjednodušuje proces zahřívání kovu. Chladnutí kovu je nyní více opožděno, aby uživatel měl více času přemístit se ke stolu.

### **Oprava kola u vozíku**

System skládání není závislý na předchozích částích a je více flexibilní pro budoucí vývoj či skládání jiných objektů. Byla odstraněna část, která byla téměř identická s jiným typem částí. Závěrečné spojení je provedeno automaticky.

### **10.3.5 Navržené vylepšení**

Tato sekce popisuje potenciální rozvoj daných interakcí v budoucích pracích. Funkcionality byly vylepšeny na základě zpětné vazby testerů.

### **Broušení**

Broušení zbraně bude záviset na jejím natočení. Pro správné obroušení zbraně je nutné nalézt správný úhel broušení.

### **Oprava kola u vozíku**

System skládačky bude obsahovat více konstrukcí, které uživatel musí složit. Uživatel bude mít možnost výběru konstrukce z nabídky.

### **Pochodně**

Možnost zapálení objektů bude tvořit logické minihry. Zapálená pochodně bude mít schopnost „vykreslit“ cestu. Při aktivaci pochodně se tato cesta začne vykreslovat a časem ustoupí. Jedná se o princip 3D malování štětcem.

---

## Závěr

Tato práce se zabývala interakcí ve virtuální realitě se zaměřením na historické prostředí. Technologie byla čtenáři přiblížena krátkým úvodem do virtuální reality a příbuzných technologií.

Byla provedena analýza hardwarového zařízení virtuální reality se zaměřením na HTC Vive, které také sloužilo jako vývojové a testovací prostředí výstupu této práce. Dalším úkolem byla implementace pomocí Unity, proto byla provedena analýza pěti vývojových balíčků pro vývoj virtuální reality v Unity. Jejich klady, zápory a funkcionality byly zhodnoceny a porovnány. V analýze podobných aplikací byly popsány a porovnány vybrané interakce. Na základě analyzovaných aplikací a analýzy základních interakcí ve virtuální realitě vzniknul návrh aplikace.

Návrh obsahoval celkově pět interakcí, které byly navrženy pro historické prostředí. Součástí návrhu byly diagramy aktivit a doménové diagramy, které sloužily jako podklad pro implementační část práce.

Implementace byla uskutečněna prostřednictvím enginu Unity. Dle analýzy byl vybrán balíček Unity XR Interaction Toolkit, který sloužil jako vývojové prostředí pro virtuální realitu v tomto enginu. Výstupem implementace byl prototyp, který obsahuje navržené interakce. Interakce byly jednoduché či střední náročnosti, neboť byly zaměřeny pro širokou veřejnost. Byly ozkoušeny metody vybraného balíčku, avšak kvůli nedostatku zkušeností s Unity, tímto balíčkem a časových prostředků byly interakce implementovány s jednoduchostí.

V závěru byly navrženy vhodné uživatelské testy, na kterých byl prototyp otestován. Podklady pro testování obsahovaly testovací scénáře a krátký dotazník. Výsledky testování ukázaly, že navržené interakce jsou vhodné do historického prostředí a pro provedení ve virtuální realitě. Některé interakce měly komplikované uživatelské rozhraní nebo obsahovaly určité problémy. Část těchto problémů byla vyřešena a upravena v prototypu. Mezi nejprospěšnější interakce patřila lukostřelba, kování a broušení. V budoucích pracích je možné tyto interakce rozvíjet a vytvořit pro ně komplexnější systém.

## ZÁVĚR

---

Jelikož vývojový hardware byl omezen na dané zařízení, tak nebyly pro implementaci využity další pokročilá zařízení virtuální reality. Při možnosti vývoje tohoto zařízení (eye-tracker, Vive Tracker atd.) je pro interakce možné navrhnout vylepšené metody a způsoby provedení.



---

## Literatura

- [1] Inc., V. M.: Oculus Rift: From \$ 2.4 million Kickstarter to \$ 2 billion sale. 03 2014, [cit. 18. dubna 2021]. Dostupné z: <https://www.engadget.com/2014-03-28-oculus-rift-from-2-4-million-kickstarter-to-2-billion-sale.html>
- [2] Machover, C.; Tice, S. E.: Virtual reality. [online], 1994, doi: 10.1109/38.250913. Dostupné z: <https://ieeexplore.ieee.org/abstract/document/250913>
- [3] Inc., S.: The Diverse Potential of VR and AR Applications. 04 2016, [cit. 18. dubna 2021]. Dostupné z: <https://www.statista.com/chart/4602/virtual-and-augmented-reality-software-revenue/>
- [4] Games, B.: Beat Saber. [software], 05 2019, [přístup 7. března 2021]. Dostupné z: [https://store.steampowered.com/app/620980/Beat\\_Saber/](https://store.steampowered.com/app/620980/Beat_Saber/)
- [5] Google: Google Earth VR. [software], 11 2016, [přístup 7. března 2021]. Dostupné z: [https://store.steampowered.com/app/348250/Google\\_Earth\\_VR/](https://store.steampowered.com/app/348250/Google_Earth_VR/)
- [6] Králové, U. H.: Věnná města českých královen. [online], 2021, [cit. 18. dubna 2021]. Dostupné z: <https://www.kralovskavennamesta.cz/index.html/>
- [7] Jana, A.; Sharma, M.; Rao, M.: HoloLens Blueprints. [online], 2017. Dostupné z: <https://www.packtpub.com/product/hololens-blueprints/9781787281943>
- [8] Pauš, P.: Smíšená realita (AR). [přednáška, online], 2019, [cit. 21. dubna 2021]. Dostupné z: [https://courses.fit.cvut.cz/NI-PVR/files/10\\_ar.pdf](https://courses.fit.cvut.cz/NI-PVR/files/10_ar.pdf)

- [9] Niantic, I.: Pokémon GO. [software], 2021, [cit. 18. dubna 2021]. Dostupné z: <https://pokemongolive.com/en/>
- [10] Milgram, P.; Takemura, H.; Utsumi, A.; aj.: Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, ročník 2351, 01 1994, doi: 10.1117/12.197321. Dostupné z: [https://www.researchgate.net/publication/228537162\\_Augmented\\_reality\\_A\\_class\\_of\\_displays\\_on\\_the\\_reality-virtuality\\_continuum](https://www.researchgate.net/publication/228537162_Augmented_reality_A_class_of_displays_on_the_reality-virtuality_continuum)
- [11] Microsoft: Virtual Reality - Experiences and devices. [online], 2021, [cit. 18. dubna 2021]. Dostupné z: <https://www.microsoft.com/en-us/mixed-reality/windows-mixed-reality>
- [12] Microsoft: HoloLens 2 - přehled, funkce a specifikace. [online], 2021, [cit. 18. dubna 2021]. Dostupné z: <https://www.microsoft.com/cs-cz/hololens/hardware>
- [13] Doc-Ok.org: What is holographic, and what isn't? 01 2015, [cit. 18. dubna 2021]. Dostupné z: <http://doc-ok.org/?p=1172>
- [14] Arrow Electronics, I.: What is Extended Reality (XR)? 02 2020, [cit. 18. dubna 2021]. Dostupné z: <https://www.arrow.com/en/research-and-events/articles/extended-reality-explained-ar-vr-and-mixed-reality-technology>
- [15] Anthes, C.; García Hernandez, R.; Wiedemann, M.; aj.: State of the Art of Virtual Reality Technologies. [online], 03 2016, doi:10.1109/AERO.2016.7500674. Dostupné z: [https://www.researchgate.net/publication/297760223\\_State\\_of\\_the\\_Art\\_of\\_Virtual\\_Reality\\_Technologies](https://www.researchgate.net/publication/297760223_State_of_the_Art_of_Virtual_Reality_Technologies)
- [16] Alza.cz: Jak vybrat brýle pro virtuální realitu. 2021, [cit. 18. dubna 2021]. Dostupné z: <https://www.alza.cz/jak-vybrat-bryle-pro-virtualni-realitu>
- [17] Mazuryk, T.; Gervautz, M.: Virtual Reality - History, Applications, Technology and Future. 12 1999. Dostupné z: [https://www.researchgate.net/publication/2617390\\_Virtual\\_Reality\\_-\\_History\\_Applications\\_Technology\\_and\\_Future](https://www.researchgate.net/publication/2617390_Virtual_Reality_-_History_Applications_Technology_and_Future)
- [18] CNET, A. R. V. C.: Wearing the Teslasuit is a shocking experience. 11 2018, [cit. 19. dubna 2021]. Dostupné z: <https://www.cnet.com/news/wearing-the-teslasuit-is-a-shocking-experience/>
- [19] NicePNG: Oculus Touch, Htc Vive, And Microsoft Mr Controllers - Diagram. [online], 2021, [cit. 17. dubna 2021]. Dostupné z:

- [https://www.nicepng.com/ourpic/u2w7a9e6y3a9u2o0\\_oculus-touch-htc-vive-and-microsoft-mr-controllers/](https://www.nicepng.com/ourpic/u2w7a9e6y3a9u2o0_oculus-touch-htc-vive-and-microsoft-mr-controllers/)
- [20] iMotions A/S: What is Eye Tracking and How Does it Work? 04 2019, [cit. 19. dubna 2021]. Dostupné z: <https://imotions.com/blog/eye-tracking-work/>
- [21] XinReality: HTC Vive Pro. 03 2021, [cit. 18. dubna 2021]. Dostupné z: [https://xinreality.com/wiki/HTC\\_Vive\\_Pro](https://xinreality.com/wiki/HTC_Vive_Pro)
- [22] Ultraleap: Leap Motion Controller. 2021, [cit. 19. dubna 2021]. Dostupné z: <https://www.ultraleap.com/product/leap-motion-controller/>
- [23] Lamora, L. G.: VR archery using a real bow — The last target felt so satisfying! [online], 2019, [cit. 18. dubna 2021]. Dostupné z: <https://www.youtube.com/watch?v=HbFln8r21XI>
- [24] Pk0001: bows type. [online], 2018, [cit. 18. dubna 2021]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Bow-type001.svg>
- [25] XinReality: HTC Vive. 03 2021, [cit. 18. dubna 2021]. Dostupné z: [https://xinreality.com/wiki/HTC\\_Vive](https://xinreality.com/wiki/HTC_Vive)
- [26] Corporation, H.: What are the system requirements? 2021, [cit. 19. dubna 2021]. Dostupné z: [https://www.vive.com/eu/support/vive/category\\_howto/what-are-the-system-requirements.html](https://www.vive.com/eu/support/vive/category_howto/what-are-the-system-requirements.html)
- [27] Tukwila, V. S.: HTC. [online], 2020, [cit. 18. dubna 2021]. Dostupné z: <https://virtualsports.net/virtual-reality/>
- [28] XinReality: Lighthouse. 03 2021, [cit. 18. dubna 2021]. Dostupné z: <https://xinreality.com/wiki/Lighthouse>
- [29] Corporation, V.: Chaperone. [online], 2017, [cit. 18. dubna 2021]. Dostupné z: [https://www.nicepng.com/ourpic/u2w7a9e6y3a9u2o0\\_oculus-touch-htc-vive-and-microsoft-mr-controllers/](https://www.nicepng.com/ourpic/u2w7a9e6y3a9u2o0_oculus-touch-htc-vive-and-microsoft-mr-controllers/)
- [30] Ltd., V.: Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection. 05 2019, [cit. 19. dubna 2021]. Dostupné z: <https://virtualspeech.com/blog/degrees-of-freedom-vr>
- [31] Corporation, H.: VIVE United States. [online], 2021, [cit. 18. dubna 2021]. Dostupné z: <https://www.vive.com/us/>
- [32] Technologies, U.: Unity. [software], 2005, [přístup 1. března 2021]. Dostupné z: <https://unity.com/>

- [33] Christopoulou, E.; Xinogalos, S.: Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices. *International Journal of Serious Games*, ročník 4, 12 2017: s. 21–36, doi: 10.17083/ijsg.v4i4.194. Dostupné z: [https://www.researchgate.net/publication/322027338\\_Overview\\_and\\_Comparative\\_Analysis\\_of\\_Game\\_Engines\\_for\\_Desktop\\_and\\_Mobile\\_Devices](https://www.researchgate.net/publication/322027338_Overview_and_Comparative_Analysis_of_Game_Engines_for_Desktop_and_Mobile_Devices)
- [34] Technologies, U.: Multiplatform. 2021, [cit. 17. dubna 2021]. Dostupné z: <https://unity.com/features/multiplatform>
- [35] Technologies, U.: Documentation, Unity scripting languages and you. 2014, [cit. 17. dubna 2021]. Dostupné z: <https://blogs.unity3d.com/2014/09/03/documentation-unity-scripting-languages-and-you/>
- [36] Technologies, U.: Unity - Manual:XR. 2021, [cit. 17. dubna 2021]. Dostupné z: <https://docs.unity3d.com/Manual/XR.html>
- [37] Technologies, U.: XR Interaction Toolkit. [software], 2021, [cit. 17. dubna 2021]. Dostupné z: `UnityPackageManager`
- [38] Technologies, U.: XR Tech Stack. [online], 12 2020, [cit. 17. dubna 2021]. Dostupné z: <https://forum.unity.com/threads/unity-support-for-openxr-in-preview.1023613/>
- [39] Technologies, U.: XR Interaction Toolkit. 2021, [cit. 17. dubna 2021]. Dostupné z: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0/manual/index.html>
- [40] Corporation, V.: SteamVR. [software], 03 2021, [přístup 15. března 2021]. Dostupné z: <https://store.steampowered.com/app/250820/SteamVR/>
- [41] Corporation, V.: Steam. [software], 09 2003, [přístup 16. září 2020]. Dostupné z: <https://store.steampowered.com/app/250820/SteamVR/>
- [42] Corporation, V.: SteamVR Plugin. [software], 2018, [přístup 15. března 2021]. Dostupné z: [https://github.com/ValveSoftware/steamvr\\_unity\\_plugin/releases](https://github.com/ValveSoftware/steamvr_unity_plugin/releases)
- [43] Corporation, V.: SteamVR Unity Plugin. 2018, [cit. 17. dubna 2021]. Dostupné z: [https://valvesoftware.github.io/steamvr\\_unity\\_plugin/articles/intro.html](https://valvesoftware.github.io/steamvr_unity_plugin/articles/intro.html)
- [44] Microsoft: Mixed Reality Toolkit - Unity. [software], 2016, [přístup 17. dubna 2021]. Dostupné z: <https://github.com/microsoft/MixedRealityToolkit-Unity>
- [45] Microsoft: What is the Mixed Reality Toolkit. 03 2021, [cit. 17. dubna 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/>

- 
- [46] Microsoft: Manipulation. [online], 12 2021, [cit. 18. dubna 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/button>
- [47] Microsoft: Pulse on trigger. [online], 12 2021, [cit. 18. dubna 2021]. Dostupné z: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/ux-building-blocks/button>
- [48] Ltd, S. S.: VRTK - Virtual Reality Toolkit - [ VR Toolkit ]. [software], 2019, [přístup 17. dubna 2021]. Dostupné z: <https://assetstore.unity.com/packages/tools/integration/vrtk-virtual-reality-toolkit-vr-toolkit-64131>
- [49] ExtendRealityLtd: VRTK - Virtual Reality Toolkit. [cit 17. dubna 2021]. Dostupné z: <https://vrtoolkit.readme.io/docs>
- [50] Oculus: Oculus Integration. [software], 2021, [přístup 17. dubna 2021]. Dostupné z: <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022>
- [51] Mine, M. R.: Virtual Environment Interaction Techniques. [online], 1995. Dostupné z: <https://dl.acm.org/doi/book/10.5555/897820>
- [52] Jiří Žára, J. S. P. F., Bedřich Beneš: *Moderní počítačová grafika*. Computer Press, 2004, ISBN 9788025104545, [cit. 25. dubna 2021]. Dostupné z: <https://books.google.cz/books?id=USQnAAAACAAJ>
- [53] Kaur, K.; Sutcliffe, A.; Maiden, N.: Improving Interaction with Virtual Environments. [online], 06 1998, doi:10.1049/ic:19980653. Dostupné z: [https://www.researchgate.net/publication/3766257\\_Improving\\_Interaction\\_with\\_Virtual\\_Environments](https://www.researchgate.net/publication/3766257_Improving_Interaction_with_Virtual_Environments)
- [54] VitruviusVR: Shadow Legend VR. [software], 03 2019, [přístup 13. března 2021]. Dostupné z: [https://store.steampowered.com/app/987230/Shadow\\_Legend\\_VR/](https://store.steampowered.com/app/987230/Shadow_Legend_VR/)
- [55] Ltd, A. R. P.: A Township Tale. [software], 2021, [přístup 14. března 2021]. Dostupné z: <https://townshiptale.com/>
- [56] WarpFrog: Blade and Sorcery. [software], 12 2018, [přístup 7. března 2021]. Dostupné z: [https://store.steampowered.com/app/629730/Blade\\_and\\_Sorcery/](https://store.steampowered.com/app/629730/Blade_and_Sorcery/)
- [57] Studios, W.: Kingdom Come: Deliverance. [software], 02 2018, [přístup 7. dubna 2021]. Dostupné z: [https://store.steampowered.com/app/379430/Kingdom\\_Come\\_Deliverance/](https://store.steampowered.com/app/379430/Kingdom_Come_Deliverance/)

- [58] Jahoda, B.: Zbraně. [online], 2015, [cit. 19. dubna 2021]. Dostupné z: <https://kingdomcome.cz/zbrane/>
- [59] Media, V.: Kingdom Come: Deliverance lockpicking guide. [online], 02 2018, [cit. 19. dubna 2021]. Dostupné z: <https://www.polygon.com/kingdom-come-deliverance-guide/2018/2/16/17020602/lockpick-guide-lock-pick-help-video-instructions>
- [60] Foundation, B.: Blender. [software], 1995, [přístup 20. dubna 2021]. Dostupné z: <https://www.blender.org>
- [61] Microsoft: Visual Studio Code. [software], 2021, [přístup 1. března 2021]. Dostupné z: <https://code.visualstudio.com/>
- [62] diagrams.net: diagrams.net. [software], 2015, [přístup 1. března 2021]. Dostupné z: <https://www.diagrams.net/>
- [63] Technologies, U.: Universal Render Pipeline. [software], 2021, [cit. 17. dubna 2021]. Dostupné z: `UnityPackageManager`
- [64] Technologies, U.: OpenVR Desktop. [software], 2021, [cit. 17. dubna 2021]. Dostupné z: `UnityPackageManager`
- [65] Technologies, U.: XR Plugin Management. [software], 2021, [cit. 17. dubna 2021]. Dostupné z: `UnityPackageManager`
- [66] Technologies, U.: TextMesh Pro. [software], 2021, [cit. 17. dubna 2021]. Dostupné z: `UnityPackageManager`
- [67] Driver, S.: Driven Decals. [software], 2021, [přístup 20. dubna 2021]. Dostupné z: <https://github.com/Anatta336/driven-decals>
- [68] Technologies, U.: Shader Graph. [software], 2021, [cit. 17. dubna 2021]. Dostupné z: `UnityPackageManager`
- [69] Quaternius: Quaternius - Free Game Assets. [online], 2021, [přístup 18. dubna 2021]. Dostupné z: <https://quaternius.com/?i=1>
- [70] with Andrew, V.: Bow and Arrow for Unity XR (Update). [online], 2021, [cit. 21. dubna 2021]. Dostupné z: <https://www.youtube.com/watch?v=H0xTz4JtWiI>
- [71] SeekPNG: Vive Controllers Technical Drawing White Blank - How-to. [online], 2021, [cit. 20. dubna 2021]. Dostupné z: <https://www.seekpng.com/ima/u2w7a9e6t4i1o0e6/>

---

## Testovací scénáře

V této sekci uvedu testovací scénáře pro každou interakci. Tyto scénáře byly použity při uživatelském testování prototypu.

### A.0.1 Lukostřelba

#### **Odhadovaný čas**

5 min.

#### **Účel testování**

Testujeme interakci lukostřelby a schopnost uživatele správně interagovat s lukem, šípy, terčem a toulcem.

#### **Počáteční bod**

Toulec s červenými šípy je připevněn v socketu na „zádech“ uživatele. Toulec s červenými i modrými šípy obsahuje plný počet šípů.

#### **Koncový bod**

Žlutý terč obsahuje zasáhnuté šípy, červený terč je prázdný.

#### **Instrukce pro testera**

Vystřelte 15 červených šípů z toulce do červeného terče (toulec s červenými šípy se nachází na zádech po startu aplikace). Následně vyměňte toulec s červenými šípy za toulec s modrými šípy. Vystřelte 10 modrých šípů do žlutého terče. Nakonec vymažte všechny šípy z červeného terče úplně (aktivujte terč).

#### **Očekávané kroky**

1. Uchopení luku do levé ruky.
2. Vystřelení červeného šípu z toulce na červený terč 10x.

## A. TESTOVACÍ SCÉNÁŘE

---

- a) Uchopení daného šípu z toulce na zádech.
- b) Připevnění šípu na luk a vystřelení šípu na daný terč.
3. Uchopení toulce, který je na zádech.
4. Doplnění toulce tlačítkem aktivovat.
5. Připevnění toulce na záda.
6. Vystřelení červeného šípu z toulce na červený terč 5x.
  - a) Uchopení daného šípu z toulce na zádech.
  - b) Připevnění šípu na luk a vystřelení šípu na daný terč.
7. Uchopení toulce, který je na zádech.
8. Odložení toulce.
9. Uchopení toulce s modrými šípy.
10. Připevnění toulce na záda.
11. Vystřelení modrého šípu z toulce na žlutý terč 10x.
  - a) Uchopení daného šípu z toulce na zádech.
  - b) Připevnění šípu na luk a vystřelení šípu na daný terč.
12. Uchopení tyče červeného terče.
13. Zničení šípů tlačítkem aktivovat.

### A.0.2 Kování

#### **Odhadovaný čas**

5 min.

#### **Účel testování**

Testujeme interakci kování a schopnost uživatele úspěšně dokovat kovový objekt do zbraně.

#### **Počáteční bod**

Uživatel má prázdné ruce. Kov je připravený ke kování a nebyl poškozen.

#### **Koncový bod**

Kov byl překován do zbraně.



---

### **Instrukce pro testera**

Zobrazte informační panel na kovu. Připevněte kovový objekt na kovářské kleště se zapnutým informačním panelem. Kovejte kov do příslušné teploty. Po úplném dokování proveďte kalení kovu. V případě neúspěchu zkuste znovu s jiným kovovým objektem.

### **Očekávané kroky**

1. Uchopení kovu do jedné ruky.
2. Zobrazení informačního panelu tlačítkem aktivovat.
3. Uchopení kleští do druhé ruky.
4. Připevnění kovu na kleště při držení tlačítka aktivace na kovu.
5. Kování kovu dokud není okován na 100%.
  - a) Ponoření kleští s kovem do středu ohně a ohřívání do kovací teploty.
  - b) Uchopení kladiva do druhé ruky.
  - c) Připevnění/odložení kleští s kovem do socketu.
  - d) Kování kovu kladivem.
6. Kalení kovu.
  - a) Ponoření kleští s kovem do středu ohně a ohřívání do kalící teploty.
  - b) Ponoření kovu do kalící kapaliny (vody).
7. Odebrání dokované zbraně z kleští.

### **A.0.3 Broušení**

#### **Odhadovaný čas**

2 min.

#### **Účel testování**

Testujeme interakci broušení a schopnost uživatele úspěšně obrousit dokovanou zbraň.

#### **Počáteční bod**

Uživatel má prázdné ruce. Kov z předchozího bodu byl dokován do zbraně, nebo použit již dokované (testovací) zbraně ve scéně.

#### **Koncový bod**

Zbraň byla obroušena.

## A. TESTOVACÍ SCÉNÁŘE

---

### **Instrukce pro testera**

Obrousíte dokovanou zbraň na brusce. V případě neúspěchu při kování použijte testovací zbraň na virtuálním stole (malý nůž).

### **Očekávané kroky**

1. Uchopení dokované zbraně do pravé ruky.
2. Uchopení rukojeti brusky.
3. Točení bruskou správnými pohyby rukojetí.
4. Přiložení zbraně na povrch točící brusky.
5. Obroušení zbraně dokud nebude plně obroušená na 100%.

### **A.0.4 Oprava kola u vozíku**

#### **Odhadovaný čas**

2 min.

#### **Účel testování**

Testujeme interakci opravy kola u vozíku a schopnost uživatele správně složit kolo z jednotlivých částí.

#### **Počáteční bod**

Uživatel má prázdné ruce. Silueta (a socket) pro složení kola je prázdná. Scéna neobsahuje žádné složené kolo.

#### **Koncový bod**

Silueta (a socket) pro složení kola je prázdná. Scéna obsahuje objekt složeného kola.

### **Instrukce pro testera**

Poskládejte kolo. Poté co si myslíte, že kolo je správně složené, uchopte kořenovou část kola a spojte kolo dohromady do jednoho objektu.

### **Očekávané kroky**

1. Uchopení kořenové (první část) kola z balíku.
2. Připojení kořenové části na socket siluety kola.
3. Uchopení správné části kola z balíku a připojte na ostatní části.
4. Opakování předchozího kroku dokud kolo není složeno.

- 
5. Uchopení kořenové části kola.
  6. Spojení kola dohromady tlačítkem aktivace.

### **A.0.5 Pochodně**

#### **Odhadovaný čas**

2 min.

#### **Účel testování**

Testujeme interakci pochodní a schopnost uživatele zapálit, zhasnout a zvýšit intenzitu pochodně.

#### **Počáteční bod**

Nezapálené pochodně a alespoň jeden zdroj ohně a vody ve scéně.

#### **Koncový bod**

První pochoděň je zapálená a umístěná v socketu u pece, druhá pochoděň se nachází ve vodě.

#### **Instrukce pro testera**

Zapalte jednu pochoděň a umístěte jí do socketu u pece. Zapalte druhou pochoděň pomocí první pochodně a zvýšte její intenzitu. S touto pochodní prozkoumejte scénu, obzvlášť temné místa. Poté pochoděň zhasněte.

#### **Očekávané kroky**

1. Uchopení první zhaslé pochodně.
2. Ponoření pochodně do pece.
3. Umístění první pochodně do socketu u pece.
4. Uchopení druhé zhaslé pochodně.
5. Ponoření pochodně do zdroje ohně jiné pochodně.
6. Zvýšení intenzity pochodně tlačítkem aktivace.
7. Prozkoumání scény a nalezení skrytého textu.
8. Ponoření pochodně do vody.



## Seznam použitých zkratk

- 2D** Two-dimensional (dvojměrný)  
**3D** Three-dimensional (trojměrný)  
**API** Application Programming Interface  
**AR** Augmented reality (smíšená realita)  
**CPU** Central processing unit  
**DOF** Degree of freedom  
**DP** DisplayPort  
**FOV** Field of view  
**GPU** Graphic processing unit  
**HDMI** High-Definition Multimedia Interface  
**HMD** Head-mounted display  
**I/O** Input/output  
**IPD** Interpupilární distance  
**ITK** Interaction Toolkit  
**LCD** Liquid Crystal Display  
**MR** Mixed reality (mixovaná realita)  
**MRTK** Mixed Reality Toolkit  
**OLED** Organic light-emitting diode  
**OVR** Oculus Integration Package

## B. SEZNAM POUŽITÝCH ZKRATEK

---

**PC** Personal computer

**SDK** Software development kit

**UE** Unreal Engine

**UE4** Unreal Engine 4

**UI** User interface

**URP** Universal Render Pipeline

**USB** Universal Serial Bus

**VR** Virtual reality (virtuální realita)

**VRTK** Virtual Reality Toolkit

**WMR** Windows Mixed Reality

**XR** Extended reality (rozšířená realita)

---

## Obsah přiloženého média

readme.txt .....	stručný popis obsahu média
build.zip .....	soubor se spustitelnou formou implementace
src	
├─ impl .....	zdrojové kódy implementace
└─ thesis .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text .....	text práce
└─ thesis.pdf .....	text práce ve formátu PDF
preview .....	snímkové ukázky z výsledné implementace