

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Framework pro vytváření tutoriálů v prostředí virtuální reality**

Místo této strany bude  
zadání práce.

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 28. června 2017

Graubner Martin

## **Abstract**

This thesis is dealing with the topic of developing applications for virtual reality and its possibilities. Within this work I provided a research of current options of technologies and devices used for virtual reality. Next there is a research of software usage options for VR development. The software consists of Unity 3D, Unreal Engine 4 and CryEngine game engines. All points are written up in the theoretical part of the document. The main part of the thesis is a framework application for developing tutorials for virtual reality itself. This app will be a base foundation for next planned development but it already gives an option for defining of tutorials for assembling of non-trivial objects.

## **Abstrakt**

Tato práce se zabývá širokými možnostmi virtuální reality a vývojem aplikací pro ni. V rámci této práce byl proveden průzkum aktuálních zařízení a technologií pro virtuální realitu. Dále byly prozkoumány možnosti softwaru související s vývojem pro virtuální realitu. Ten zahrnuje herní enginy Unity 3D, Unreal Engine 4 a CryEngine. Poznatky jsou sepsány v teoretické části práce. Hlavní částí této bakalářské práce byl samotný vývoj frameworkové aplikace pro vytváření tutoriálů v prostředí virtuální reality. Tato aplikace bude sloužit jako základ pro další vývoj, ale sama o sobě již poskytuje možnost definovat jednoduché tutoriály pro skládání netriviálních předmětů.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Řešení</b>	<b>8</b>
2.1	Zařízení pro VR . . . . .	8
2.1.1	Nástroje a komponenty . . . . .	8
2.1.2	Prozkoumané zařízení . . . . .	9
2.1.3	Ostatní . . . . .	11
2.1.4	Závěr - vybrané zařízení . . . . .	12
2.2	Porovnání game enginů . . . . .	12
2.2.1	Obecná charakteristika . . . . .	13
2.2.2	Unity 5 . . . . .	13
2.2.3	CryEngine . . . . .	16
2.2.4	Unreal Engine 4 . . . . .	18
2.2.5	Závěr - vybraný engine . . . . .	19
2.3	Aplikace pro VR . . . . .	19
2.3.1	Vývoj a kvalita . . . . .	19
2.3.2	VR Hry . . . . .	20
2.3.3	Ostatní VR aplikace . . . . .	22
<b>3</b>	<b>Praktická část</b>	<b>24</b>
3.1	Úvod . . . . .	24
3.2	Analýza . . . . .	24
3.2.1	Ovládání . . . . .	24
3.2.2	Pohyb . . . . .	24
3.2.3	Interakce . . . . .	25
3.2.4	Uživatelské rozhraní . . . . .	25
3.2.5	Generování scénáře . . . . .	25
3.2.6	Úkoly . . . . .	26
3.2.7	Události . . . . .	26
3.3	Implementace . . . . .	26
3.3.1	Lokalizace . . . . .	26
3.3.2	Struktura definice scény . . . . .	27
3.3.3	Vybrání scény . . . . .	30
3.3.4	Generování scény . . . . .	30
3.3.5	Uživatelská interakce . . . . .	31

4	Prostor pro budoucí vývoj	37
5	Závěr	38
	Literatura	41

# 1 Úvod

Tato práce vznikla na základě spolupráce Zápodočeské univerzity a firmy Aimtec a.s., které poskytly zázemí pro vývoj. Na univerzitě byla dostupná obě zařízení HTC Vive a Oculus Rift. Aplikace bude prezentována 20. září na konferenci TAL a bude dále upravována pro potřeby zákazníků společnosti Aimtec a.s.

Cílem této práce je prozkoumání aktuálních technologií, vývoje a možného využití virtuální reality. Dosavadní nejširší využití technologií pro virtuální realitu nalzáme v oblasti vývoje počítačových her a aplikací souvisejících se zábavním průmyslem. Druhé nejširší uplatnění těchto technologií najdeme ve vizualizaci, designu a prototypování, např. výrobních prostor, interiérů, často při návrhu prototypů vozidel a podobně konstrukčně složitých strojů. Pomocí technologií virtuální reality je možné vyzkoušet vlastnosti navrhovaných objektů v prvotní fázi vývoje - jejich ergonomii, vzhled, pohyb, a snížit tak náklady na vývoj fyzického prototypu.

Další teoretická část práce se bude věnovat seznámení s aplikacemi pro virtuální realitu a poskytne bližší pohled na manipulaci s objekty ve virtuální realitě. Se zvyšující se dostupností levnějšího a výkonnějšího hardwaru se i prostředky virtuální a rozšířené reality stanou dostupnější a použitelnější pro širší spektrum uživatelů. Se vznikem více aplikací je nutné sjednotit ovládání, aby bylo přirozené a stejné ve většině aplikací.

V praktické části práce se pak budu zabývat návrhem své frameworkové aplikace pro vytváření tutoriálů pro reálné výrobní procesy. Aplikace bude načítat připravené scénáře, které budou uživatele provádět krok za krokem výrobním procesem, uživatel bude dostávat pokyny k jednotlivým činnostem a zacházení s objekty.

Možnosti této aplikace poté budu demonstrovat na několika jednoduchých scénářích a jednom komplexním úkolu. Zpravidla se může jednat o skládání menších objektů do větších celků.

## 2 Řešení

### 2.1 Zařízení pro VR

SteamVR a jiné verze virtuální reality jsou si ve svých funkcích velmi podobné, ale najdeme mezi nimi i podstatné rozdíly. Předmětem této sekce bude jejich popis a srovnání.

#### 2.1.1 Nástroje a komponenty

Aby se uživatel mohl pohybovat ve virtuální realitě, jsou zapotřebí komponenty, které jsou pro všechna zařízení pro VR systémy podobné. Pro provoz virtuální reality potřebujeme příslušný software a hardware. Hardware zajistí počítač, který bude svým výkonem a parametry splňovat požadavky virtuální reality. Požadavky se týkají OS, verze procesoru, velikosti RAM a především parametrů grafické karty. Pro Oculus Rift i HTC Vive jsou k dispozici testy kompatibility, kterými můžeme porovnat naše vybavení s požadavky platformem a které budou popsány níže.

Kromě PC potřebujeme vstupní zařízení pro interakci s virtuálním prostředím, tím mohou být různé typy ovladačů, headsety a senzory, které jsou na nich umístěné. Dále je třeba systém pro snímání pohybu, u HTC Vive jsou to dvě základové stanice, které se umístí do rohů místnosti, odkud do senzorů vysílají laserové paprsky.

Pro vytvoření virtuálního prostředí je zapotřebí software pro vytváření a modelování objektů, např. Maya, Blender. Nakonec potřebujeme simulační software, který propojí funkci všech komponent dohromady. Přijímá data ze snímačů pohybu a vstupních zařízení, aplikuje je na vytvořené objekty a aktualizuje scénu. Pomocí něho nastavujeme chování objektů a pravidla virtuální scény.

Pro plnohodnotný provoz VR potřebujeme i herní prostor, ve kterém se budeme pohybovat, narozdíl od rozšířené reality vidíme pouze virtuální svět.



## Testy kompatibility

Test kompatibility pro Oculus Rift zkontroluje výkon a specifikace komponent, tj. CPU, RAM, grafické karty, také počet potřebných USB portů a jejich zapojení do základní desky. Některé starší verze mohou být s Oculus Rift nekompatibilní.

U HTC Vive je k dispozici benchmark SteamVR Performance Test s technologií, která měří okamžitou snímkovou frekvenci. Obtížnost scény je jí dynamicky přizpůsobována, aby se ani při nižší frekvenci u slabších karet nestalo, že uživatel ztratí dojem virtuální reality. Pokud technologie zaznamená větší prodlevu snímků, tvoří dočasně obraz v nižší vizuální kvalitě. Méně kvalitní snímky jsou generovány rychleji, takže umožní vrátit snímkovou frekvenci do optimálních hodnot.

### 2.1.2 Prozkoumané zařízení

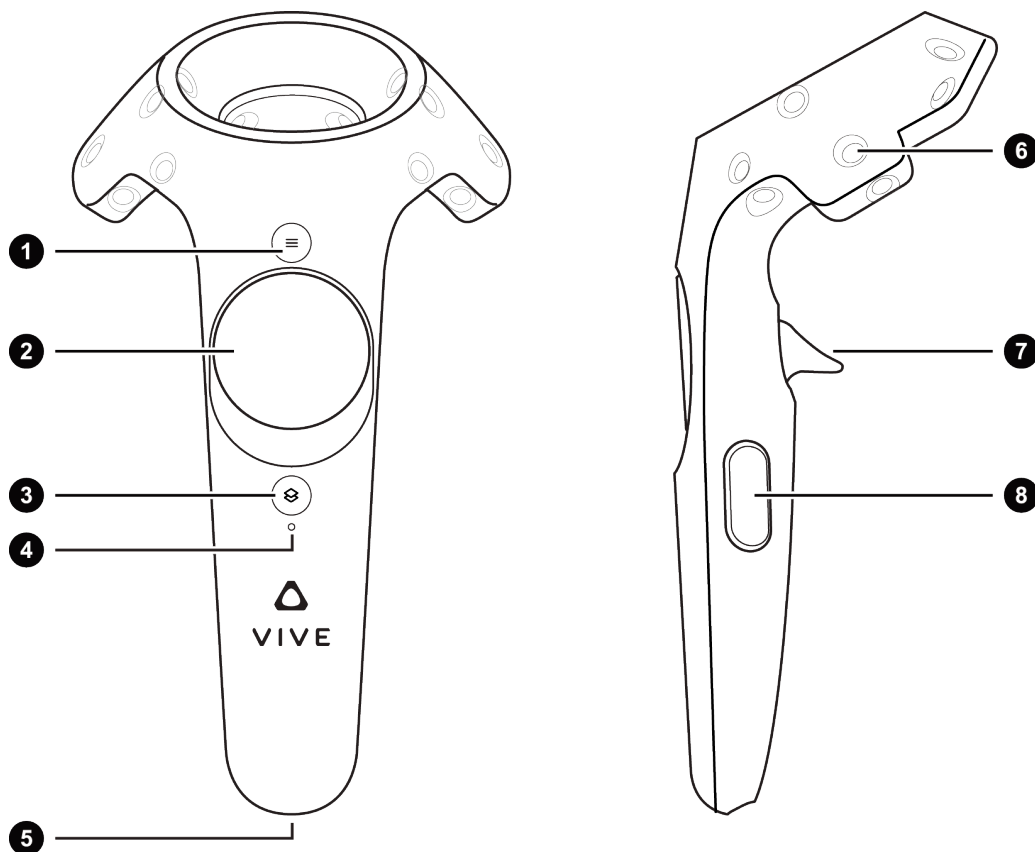
#### HTC Vive

Pro použití zařízení HTC Vive jsou k dispozici platformy SteamVR a VivePort, zatímco první je orientovaná na hry, druhá nabízí širší výběr zážitků ve virtuální realitě a podobá se více platformě pro Oculus Rift (Oculus Home).

Jedním z hlavních rozdílů mezi SteamVR (HTC Vive) a ostatními zařízeními pro VR systémy je způsob, jakým je sledován pohyb uživatele. Zatímco většina nástrojů pro virtuální realitu zaznamenává obraz polohy uživatele v realitě kamerou na zařízení a pak jej virtuálně zpracovává, sledovací systém SteamVR pro zachycení pohybu využívá dvě stanice pro mapování polohy pomocí laseru, který je zachycován headsetem a ovladači. To dělá sledovací systém pokročilejším a je tak umožněno sledovat pohyb a lokaci přesněji. Stanice jsou schopné lépe zachytit pozici a orientaci headsetu a ovladače, což optimalizuje jejich funkci a snižuje prodlevy při pohybu. Přesnost snímacích stanic umožňuje systematicky zpracovávat vstupy z více zařízení současně s více než milimetrovou přesností v řádech milisekund. Ovladače jsou navrženy tak, aby s nimi bylo jednoduché interagovat s objekty ve VR.

Podstatným rozdílem je možnost fyzicky pohybovat v prostoru se SteamVR zařízeními. Pohyb je přenesen do scény virtuální reality a uživatel má k dispozici nová místa ve scéně. Bez této možnosti u jiných zařízení pro virtuální realitu by jich nebyl schopen takto dosáhnout.

Pro kontrolu při pohybu si můžeme aktivovat tzv. Chaperone systém, kterým zviditelníme ve VR hranice našeho reálného pohybového prostoru, jehož velikost si přednastavíme. Na headsetu najdeme přední kameru a stiskem tlačítka se můžeme podívat kolem sebe v reálném prostoru, aniž bychom si sundávali headset, například když si nejsme jistí, kde přesně se fyzicky nacházíme.



Obrázek 2.1: Ovladače pro HTC Vive.

## Oculus Rift

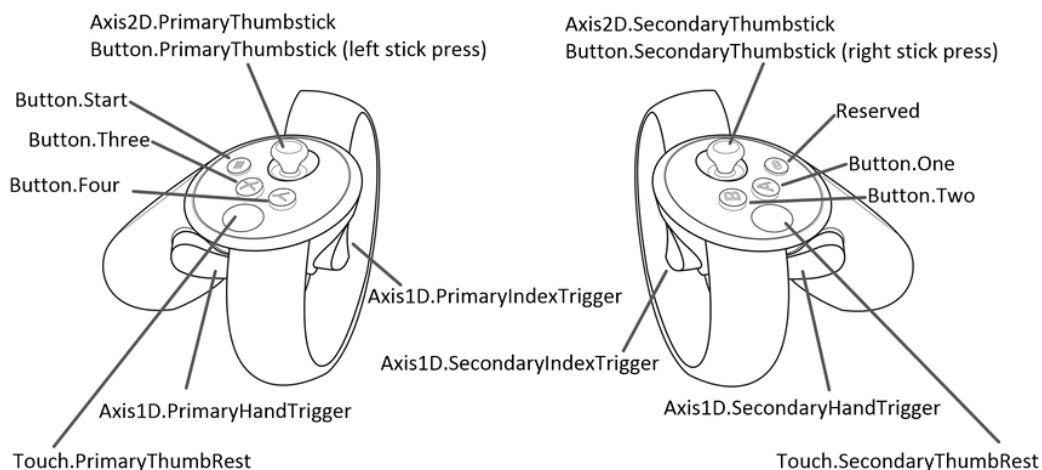
Oculus Rift (Development Kit 2) je provozován na platformě s názvem Oculus Home, podporuje hry přes Steam, nicméně katalog možností je podstatně menší než u HTC Vive.

Pro mapování pohybu je zde kamera, která používá infračervené záření. Je vhodný spíše pro použití v sedě a vestoje, pohyb v herním prostoru je možný, ale jeho mapování kamerou není tak všestranné jako u laserové technologie SteamVR.

Výhodou u tohoto produktu je v porovnání s jinými verzemi minimální screen-door efekt, který ve VR scéně pozorujeme jako mřížku drobných čar. Důvodem je prostor mezi pixely, náš obraz je zaostřený a displej v blízké vzdálenosti od oka. Tento efekt se nedá vyloučit a stává se se všemi aktuálními verzemi virtuální reality. V první vývojové verzi Oculus Rift DK1 byl velmi patrný.

Ve srovnání s HTC Vive má Oculus Rift shodné tyto parametry: OLED displej s rozlišením 2160 x 1200, snímkovou frekvencí 90 fps, rozsah pohledu 110 stupňů, vestavěné audio zařízení s mikrofonom, parametry zapojení (HDMI, USB 2.0, USB 3.0). Cenová dostupnost je u obou variant srovnatelná.

V souvislosti s tím, že Oculus Rift byl vydán dříve než HTC Vive, se tyto sety přirozeně liší v požadavcích na hardware, jako je verze jádra ap. HTC Vive má také poloviční nároky na velikost RAM. Rozdílem je i rozsah zachycení herního prostoru, snímací zařízení HTC Vive je schopné zmapovat větší prostor. Funkce přední kamery u Oculus Rift chybí. Aby uživatel neztratil svou pozici, je možné využít funkce Guardian, která je jakousi alternativou k Chaperone systému u HTC Vive. Postupným ztmavováním scény upozorňuje uživatele, že se blíží k hranici, kterou snímá kamera.



Obrázek 2.2: Ovladače pro Oculus RIFT a jejich mapování.

### 2.1.3 Ostatní

Následující produkty oproti předešlým poskytují velmi omezené funkce a jedná se spíše o široce dostupnou a nízkonákladovou alternativu virtuální

reality.

### **Samsung Gear VR**

Toto zařízení je alternativou od společnosti Samsung, která pro vytvoření efektu virtuální reality používá mobilní telefon, který je s aktuální verzí Gear VR kompatibilní. Skládá se z jednoho jednoduchého ovladače a headsetu, na který se mobilní telefon připevní a poskytne virtuální 3D obraz. Toto zařízení tedy nemapuje pohyb jako předchozí pokročilé VR systémy a svým konceptem je spíše vhodné ke srovnání s Google Cardboard VR. Oproti této verzi nabízí větší výběr aplikací, na druhou stranu omezuje uživatele požadavky na výrobce mobilního telefonu.

### **Google Cardboard VR**

Verze virtuální reality od společnosti Google funguje na podobném principu, je k ní třeba smartphone s operačním systémem Android nebo iOS, dostatečná velikost jeho obrazovky, funkce gyroskopu, která je potřebná k stabilizaci obrazu, a GPS. V tomto případě je 3D obraz tvořen dělením obrazovky smartphonu. Základní aplikace se jmenuje Cardboard. Uživatel se dívá skrze čočky v Cardboard brýlích na obrazovku svého telefonu, který vložil dovnitř.

#### **2.1.4 Závěr - vybrané zařízení**

Pro vývoj jsem vybral zařízení HTC Vive. Hlavním důvodem je dostupnost zařízení na univerzitě a ve firmě Aimtec, kde je možné aplikaci testovat a prezentovat. Zařízení mě také mnohem více oslovilo svými možnostmi využití prostoru a rychlou odezvou.

## **2.2 Porovnání game enginů**

Následující sekce se bude zabývat herními enginy, jejich obecnou charakteristikou, funkcemi a srovnáním. Game engine je sada softwarových nástrojů pro snazší vývoj her, poskytuje vývojáři nástroje tak, aby pro vývoj svého produktu měl určitý základ a nemusel všechny funkce, které bude používat, implementovat sám. Použitím enginu je docíleno snížení nákladů na vývoj herních aplikací a rychlejší vstup produktu na trh, což jsou faktory úspěchu v konkurenčním prostředí herního průmyslu. O výběru herního enginu rozhoduje, jaký typ aplikace vytváříme, jestli je projekt ve 2D nebo 3D a jaký typ licencování je pro nás vhodný.

### 2.2.1 Obecná charakteristika

Game engine je obecně framework, který se skládá z následujících komponent: hlavní program, renderovací engine, engine fyziky, audio engine a další, v závislosti na konkrétním enginu.

Hlavní program je implementací algoritmů, poskytuje vlastní logiku aplikace. Je oddělený od činností jako jsou renderování obrazu, práce se zvukem, tyto činnosti zajišťují ostatní specializované části enginu.

Grafikou se zabývá renderovací engine. Renderování nebo syntéza obrazu je proces tvorby obrazu scény. Pomocí některé z jejích metod, např. metody sledování paprsku (ray-tracing), je cílem dosáhnout realistického obrazu. Ten vychází z 3D modelu, který jsme si předtím vytvořili v některém modelovacím nástroji. Na tomto obojživelném modelu může být nadefinováno osvětlení, textura, stínování a další efekty viditelné na povrchu, zahrnutím těchto vlastností vznikne osvětlovací model. Různé renderovací techniky jsou založené na různých osvětlovacích modelech. Komponenty světelných modelů jsou kombinace různých typů osvětlení. Obecně se dá říci, že čím více efektivně popsaných komponent model má, tím realističtějšího obrazu můžeme dosáhnout.[2]

Fyzikální engine určuje způsob chování předmětů a vliv fyzikálních jevů na scénu. Typicky se jedná o gravitaci a dynamiku pohybu objektů včetně detekce kolizí.

Některé game enginey poskytují i nezávislost na platformě, aplikace může běžet na různých konzolích a počítačích bez nutnosti změn, nebo s minimálními změnami ve zdrojovém kódu. Obecný trend ve vývoji game engineů je zlepšování uživatelského prostředí, zvyšující se cenová dostupnost - většinou zdarma pro začínající vývojáře, stavění engineů na základě vyšších programovacích jazyků, jako jsou Java, C# atd.

### 2.2.2 Unity 5

Unity game engine nabízí velké množství funkcí a relativně snadno pochopitelné uživatelské rozhraní. Jeho výhodou je snadný přenos mezi platformami, to znamená, že aplikace jdou snadno převést na podporu pro Android, iOS, Windows Phone aj., což z něj dělá vhodný engine pro vývoj mobilních

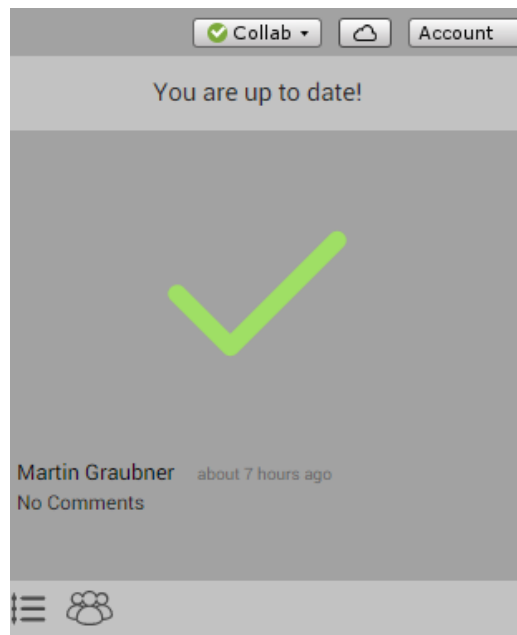
aplikací. Další možností je vývoj konzolových aplikací, zde je nutné mít přístup k jejich SDK. Tento engine navíc neomezuje typ souborového formátu z obecně používaných aplikací pro modelování, podporuje aplikace Maya, Blender, Softimage a další. V souvislosti s tím, že podporuje velké množství modelovacích nástrojů, nemá podobné funkce zabudované kromě některých základních tvarů. Velké množství assetů (zdroje využívané aplikací - grafický model, skript, zvuk, textura ap.) je volně dostupné nebo k zakoupení z Asset Storu. Unity pro jeho snadnou dostupnost a široký rozsah možností používá kromě herních studií zejména velké množství indie vývojářů. Je vhodný i pro vývoj 2D aplikací.

Vývoj je možný v C# , JavaScriptu, Cg, Boo, HLSL.

Unity game engine podporuje i zařízení pro virtuální realitu, mezi nimi jsou HTC Vive, Oculus Rift, Samsung Gear VR. Na Unity Asset Storu je zdarma dostupný SteamVR Plugin a další rozšíření editoru, mezi zajímavé patří VRTK (Virtual Reality Toolkit), Newton VR, VR Samples.

## **Collab**

Užitečnou funkcí je Unity Collaborate, která slouží pro ukládání, sdílení a synchronizaci projektu. Unity Collaborate je cloudové řešení, jednoduché pro uživatele. Je nejvhodnější pro jednočlenné až desetičlenné týmy např. různých specialistů - audio expertů, vývojářů apod., kterým umožní práci na společném projektu.



Obrázek 2.3: Okénko unity collab.

## Licence

Jsou poskytovány čtyři základní typy licencí s různými omezeními. Všechny licence obsahují všechny komponenty engine, neliší se v podpoře platform, zahrnují všechny průběžné aktualizace a přístup k beta verzím a neobsahují další licenční poplatky kromě stanovených poplatků za měsíc.

### 1. Personal

Licence vhodná pro začátečníky, studenty, použití k výzkumu, bez možnosti některých nástrojů pro pokročilejší vývoj, přístupu ke zdrojovému kódu a podpory. Je limitována výší ročního příjmu 100 000 \$, počtem 20 uživatelů pro multiplayer, před spuštěním aplikace není možné přizpůsobovat výchozí Unity uvítací obrazovku.

### 2. Plus

Licence pro vývojáře dostupná za 35 \$ měsíčně, je omezená ročním příjmem 200 000 \$, počet hráčů pro multiplayer je maximálně 50. Oproti předchozí licenci je tato rozšířená o možnost získat assety z

obchodu o 20 % levněji. Navíc má uživatel ještě možnost měsíčního přístupu na Unity Certified Developer Courseware, kde může prohloubit své dovednosti ve vývoji her. S touto licencí je možné přizpůsobit si uvítací obrazovku, nebo ji vynechat.

### 3. Pro

Zahrnuje funkce předchozí licence, vhodná pro profesionály s úmyslem profitovat z pokročilých funkcí a flexibility enginu. Cena je 125 \$ měsíčně, není omezena ziskem z vývoje, počet možných hráčů pro multiplayer je 200. Assety z obchodu dostupné s 40% slevou, licence zahrnuje tříměsíční přístup na Unity Certified Developer Courseware. Je zde zpoplatněná možnost přístupu ke zdrojovému kódu.

### 4. Enterprise

Tato licence je řešením pro konkrétní společnosti se zájmem o Unity game engine. Nástroje jako je např. počet multiplayer hráčů lze přizpůsobit.

## 2.2.3 CryEngine

CryEngine je navržen společností Crytek pro počítačové a konzolové platformy jako PlayStation 4 a Xbox One. Jeho grafické možnosti přesahují Unity a jsou srovnatelné s Unreal Engine 4, jedná se o osvětlení, realistickou fyziku objektů, pokročilé možnosti animací. Ve srovnání s Unity editor CryEnginu může být náročnější na pochopení a může chvíli trvat, než bude uživatel schopný využívat efektivně jeho funkce. Proto se hodí spíše pro uživatele s předchozími zkušenostmi s ostatními herními enginy. Pro své pokročilé grafické možnosti se hodí pro potřeby velmi reálných simulací a filmů. Možnosti licencí jsou zde odlišné a budou popsány níže.

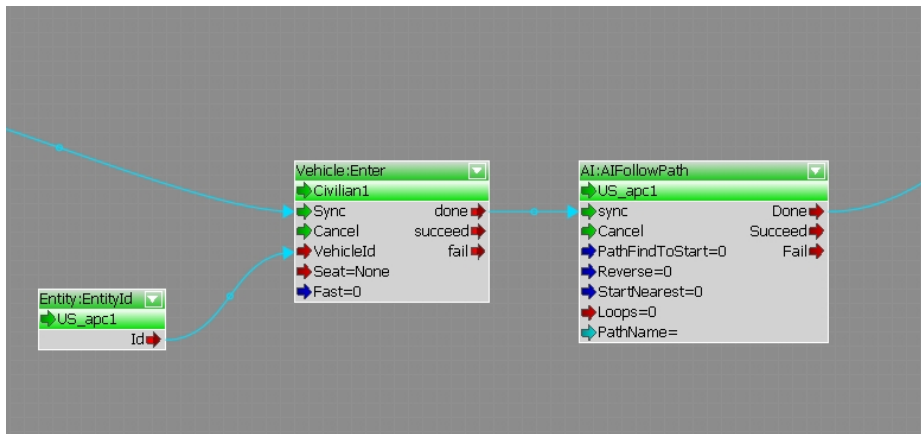
Implementaci logiky usnadňuje možnost využít Flow Graph Editor. Jeho schéma je znázorněno na obrázku č. 2.4. Flow Graph je vizuální skriptovací systém, jeho hlavní výhodou je, že uživatel při práci s ním nemusí přímo používat žádný programovací jazyk a znát skriptování. Je možné ho použít pro prototypování gameplay, efektů a zvukového designu, vytváření herní logiky.



## Flow Graph

Flow Graph se skládá z uzlů, které reprezentují entity nebo komponenty. Komponenta má různý počet vstupů a výstupů různého typu. Když obdrží hodnotu nebo událost na daný vstup, tento vstup se aktivuje a provede se specifická akce. Komponenty mohou mít cílové entity, se kterými spolupracují.

Uzly mají vstupní a výstupní porty, kterými jsou navzájem propojené orientovanými hranami. Datové typy vstupů a výstupů mohou být bool, float, int, string, any type a event type. Typ any předává jakoukoli hodnotu ze vstupu do výstupu v rámci komponenty. Typ event neobsahuje žádnou hodnotu, pouze aktivuje vstup a komponenta může zpracovat vstupní událost. Logika Flow Graph je uložena v xml formátu.



Obrázek 2.4: Ukázka flow graphu.

## Licence

Licence je poskytována zdarma pro všechny nástroje enginu, uživatelé mohou finančně přispět částkou v libovolné výši. Uživatel má přístup ke všem podporovaným platformám: Windows a Linux PC, PS4 a Xbox One. Neomezuje uživatele v komercializaci svých projektů vyvinutých s CryEngine. Od roku 2016 CryEngine volně zpřístupňuje i celý zdrojový kód v C++. Nově také zahrnuje i nástroje podporující virtuální realitu - Oculus Rift, HTC Vive, Open-Source VR a PlayStation VR.

Dodatečné výhody lze získat členstvím v komunitě, k dispozici je základní členství za 50 \$ měsíčně nebo prémiové za 150 \$ měsíčně. Prémiové oproti základnímu nabízí např. konzultační servis. Obě varianty zpřístupní uživateli Insider Answers sekci, kde může najít užitečné informace.

## 2.2.4 Unreal Engine 4

Unreal Engine 4 je v současnosti asi nejvýkonnější herní engine od Epic Games a je nástupcem předchozího UDK (Unreal Development Kit). Má společně s CryEnginem nejpokročilejší grafické možnosti, jsou zde nástroje pro animace a tvorbu videa. Pokročilý částicový systém, který zvládne práci s miliony částic ve scéně najednou, umožňuje v Unreal Enginu dosáhnout extrémně realistických efektů objektů, které jsou příliš členité nebo se neustále mění tak, že není možné nadefinovat jejich povrch. To se týká např. modelování dýmu, ohně, galaxií, vodopádů a podobných struktur.

Unreal Engine se snaží o uživatelsky přívětivé prostředí, vývojovým jazykem je C++. Hry napsané za pomoci tohoto enginu mohou být vydávány pro PC, Mac, iOS, Android, Xbox One a PlayStation 4. Je tedy použitelný jak pro vývoj jednoduchých mobilních aplikací ve 2D, tak zejména pro hry zaměřené na grafiku. Pro začínající vývojáře jsou nejběžnější mobilní a PC platformy, pro vývoj konzolových aplikací jsou z hlediska dostupnosti stejné podmínky jako u obou předchozích enginů, to znamená přístup k SDK a osvědčená verze vydání nutná pro získání příslušných nástrojů.

Unreal Engine 4 se od roku 2015 více zpřístupnil pro začínající vývojáře s bezplatnou licencí. Všechny funkce enginu jsou zdarma, pokud zisk uživatele nepřekročí 3000 \$ za čtvrtletí. Uživatel má také plný přístup ke zdrojovému kódu v C++. Pokud je překročena hranice výdělku, činí licenční poplatek 5 % ze zisku.

Implementaci a návrh usnadňuje nástroj Blueprints, podobně jako Flow Graph v CryEngine. Je to vizuální skriptovací systém a je pomocí něho možné rychlé prototypování hry bez psaní kódu. Je možné s ním vytvořit komplexní prvky do hry pomocí propojování uzlů, událostí a funkcí s vlákny.

Další užitečný nástroj je Source Control, je to zabudovaná podpora Unreal editoru pro systém správy projektu. Umožňuje přehledné sledování změn v kódu a datech a spolupráci na projektu vývojovým týmům. Uživatel si může aktivovat funkci automatického checkoutu, který zajistí aktualizace assetu, pokud jej někdo jiný modifikoval.



Obrázek 2.5: Ukázka bluepruntu z Unreal Engineu pro obsluhu teleportace pomocí levé ruky.

Unreal Engine editor podporuje vývoj přímo ve virtuální realitě, to znamená, že vývojář si může hned ověřit, jak jím vytvořené nové prvky ve VR fungují.

## 2.2.5 Závěr - vybraný engine

Pro účel BP jsem vybral engine Unity 3D. Rozhodnutí bylo těžké, protože se nedá jednoznačně říci, který z engineů je lepší. Hlavním aspektem výběru pro mě byl jazyk, ve kterém lze do engineu psát, v případě Unity C# nebo JavaScript. Dalším rozhodovacím aspektem byl verzovací systém, jaký určitý engine používá. V případě Unity je to Collab, který běží kompletně v cloudu a tedy odstiňuje uživatele od starostí s infraskruturou a vše je na jednom místě. Velkou pomocí je v Unity i již zmíněný Asset Store obsahující stovky zdrojů a knihoven, které jsou zcela zdarma a nebo za nízkou cenu, což se o Unreal Engineu říci nedá. Komunita kolem Unity je z velké části tvořena indie vývojáři, kteří si své know-how tolik neudržují jen v rámci svého studia, ale o poznatky se dělí na diskuzních forech a diskutují o problémech, které řeší. To může být velkou výhodou při řešení podobných problémů.

## 2.3 Aplikace pro VR

### 2.3.1 Vývoj a kvalita

Při vývoji aplikací pro VR je třeba zaměřit se na konfliktní informace, které může mozek získávat podobně jako při jízdě v autě. To může mít za následek nevolnost a nepohodlí při používání VR. Vývojáři proto musí věnovat pozornost např. rychlosti chůze, naklonění pohledu a kalibrování pohybu hlavy nebo stoupání. Současně s tím by výrobci zařízení měli dbát

na zlepšování technologií a hardwaru, zejména kvůli snižování prodlevy při odezvách zařízení a obecné kvalitě obrazu.

Kvalitativní parametry aplikací pro virtuální realitu jsou podobné jako u aplikací vytvářených pro PC nebo konzole. Důležitou roli hraje grafika, příběh hry, gameplay, zvuky a řada dalších vlastností, které uživatelé posuzují subjektivně. Důležitým kvalitativním parametrem je imerze uživatele. Jedná se o míru vtažení hráče nebo diváka do děje. Nevlnost může způsobit ale i imerze uživatele i při jinak kvalitních podmínkách, pokud provádí nereálné pohyby, které neodpovídají běžné reprodukci pohybu těla.

### **2.3.2 VR Hry**

#### **I Expect You To Die**

Hlavolamová hra, která se dá hrát na Oculus Rift, ve které je hráč elitním agentem a je vystaven hrozbám, kterým musí uniknout. Hráč se nepohybuje po scéně, z jednoho místa provádí interakci s okolními objekty. Používá kontrolu obou rukou a telekinezi na vzdálené předměty. Pokud spolu objekty nějak interagují, proběhně interakce samovolně, pokud jsou v dostatečné blízkosti u sebe, např. šroubovák a šrouby. V prvním úkolu je například hráč uvězněn v autě a snaží se dostat z nakládacího pásu letadla. Úkoly jsou celkem čtyři a bývá více možností řešení situace. Hra je nejvhodnější pro Oculus Rift s použitím Touch ovladačů, je dostupná i pro PlayStation VR a HTC Vive.

#### **The Lab**

The Lab tvoří série osmi miniher pro HTC Vive, které demonstrují, jak lze vytvářet tradiční herní žánry ve VR. Je možné velké množství různých interakcí s objekty. Hráč se pohybuje po scéně pomocí teleportace a fyzicky ve vymezeném prostoru. Uživatel se nejprve nachází v prostoru, kde si zvolí hru zvednutím koule s jejím jménem a přiblíží si ji k obličeji. Na výběr jsou:

##### **1. Longbow**

Střelba z luku, kdy hráč používá jeden ovladač jako luk, druhý jako šíp a snaží se napodobit přirozený pohyb. Cílem hry je ubránit bránu hradu, zatímco hráč stojí na zdech proti nepřátelům, kteří jsou znázorněni jako černé figurky. Hra demonstruje velmi dobrou kontrolu pohybu a přesnost ovladačů.

## 2. Slingshot

Střelba z praku, cílem je způsobit co největší poškození s počtem hořících koulí, které má hráč k dispozici. Je principem podobná hře Angry Birds.

## 3. Robot Repair

Tato minihra, ve které je opravován robot podle instrukcí, rozšiřuje funkce předchozího technického dema. Stejné prostředí je mj. využité v testu kompatibility se SteamVR. Touto hrou jsem se nechal inspirovat v praktické části práce, konkrétně v šabloně Workshop, viz dále.

## 4. Solar System

## 5. Xortex

## 6. Postcards

## 7. Human Medical Scan

## 8. Secret Shop

## **Out Of Ammo**

Out Of Ammo je akční a zároveň strategická hra. Hráč je na pozici velitele a čelí zesilujícím se vlnám nepřátelských vojsk. Staví obranné systémy, přesouvá jednotky, vydává rozkazy a pohybuje se po scéně přenesením do pohledu jednotlivých vojáků předem umístěných hráčem, pak může střídavě hrát z jejich pozice. Každý má jiné úkoly a vybavení. Klíčovým prvkem je manipulace se zbraněmi. Hráč provádí jednoduché činnosti jako střelbu a ruční přebíjení pistole nebo složitější přebíjení kulometu. Grafika této hry nemá být realistická, je velmi jednoduchá, hra se odehrává ve více prostředích - město, poušť atd. Je to hra vyvinutá pro HTC Vive a je možné ji hrát s Oculus Rift s Touch ovladači.

### 2.3.3 Ostatní VR aplikace

Aplikace pro VR dále mohou mít vzdělávací charakter nebo být použitelné při návrhu reálných objektů, vozidel apod.

#### Mecas ESI - IC.IDO

Během průzkumu ostatních aplikací jsem navštívil firmu Mecas ESI, která se zabývá převážně virtuální simulací a prototypováním v automobilovém a leteckém průmyslu. Jeden z jejich produktů jménem IC.IDO řeší právě téma virtuální reality jako takové. Následující citace popisuje aplikaci, která se zabývá simulací a prototypováním v oblasti průmyslu.

IC.IDO je výkonný softwarový produkt, který zobrazuje trojrozměrný model virtuálního prototypu v jeho reálných rozměrech a znázorňuje jeho chování v reálném čase. Tato věrná vizualizace přibližuje vnímání virtuálního produktu v realitě.

IC.IDO využívají významné společnosti z oblasti strojírenského průmyslu; uplatnění nachází také v leteckém a kosmickém průmyslu, v automobilovém průmyslu a ve výrobě průmyslových strojů. Virtuální realita je užívána pro posuzování konstrukčních návrhů, vyhodnocování a optimalizaci montážních a demontážních postupů, pro ověřování technických prostředků a nástrojů pro výrobu a údržbu. IC.IDO umožňuje během simulací montáže a údržby odhalovat kontakt a kolizní oblasti pevných součástí a okamžitě vyhodnocovat chování flexibilních komponent, jako jsou např. trubky a kabely. Představuje též podporu pro vyhotovení technické dokumentace a animací pro nácvik pracovních postupů. [1]

#### Destinations

Aplikace od společnosti Valve, hráč si v úvodu vybere destinaci, do které by se chtěl podívat, jedná se o prostředí reálného světa. Podporuje i multiplayer a virtuální cestování ve více lidech. Scény jsou renderovány pomocí fotogrammetrie (vědní obor zabývající se rekonstrukcí tvarů, velikostí a určení polohy objektů z fotografických snímků) tak, jako existují v životních velikostech. Bohužel nelze s objekty úplně vhodně interagovat. Detaily scény jsou dobře propracované a tedy velmi realistické i přes omezené rozlišení VR headsetů.

Pohyb po scéně je možný omezeně fyzicky a klasicky pomocí tlačítka teleportu, možné je tedy i procházení v sedě. Pohyb po scéně je samozřejmě omezený, např. na některé interiéry budov. Dodatečnou součástí je Destinations Workshop Tools pomocí kterého je možné vyvíjet vlastní scénu. Destinations je možné si zahrát na HTC Vive nebo Oculus Rift s Touch ovladači.

### **Google Earth VR**

Koncepčně podobný projekt pro HTC Vive s podporou i Oculus Rift Touch ovladačů. Jedná se o rozšíření Google Earth pro virtuální realitu. Nabízí větší množství destinací než Destinations a větší volnost pohybu, na druhou stranu renderuje méně fotorealistickou scénu. Uživatel pomocí ovladače může létat nad scénou. Má možnost vybrat si lokaci napsáním jejího názvu na virtuální klávesnici.

# 3 Praktická část

## 3.1 Úvod

Vývoj bude probíhat v enginu Unity 3D a pro zařízení HTC Vive. Unity editor má jednoduché a přehledné uživatelské rozhraní umožňující editování scény a objektů v ní. Všechny assety použitelné v editoru budou tříděny ve složce VR\_Trainer (pracovní název projektu) do podsložek se samopopisnými názvy, tedy například složka Scripts bude obsahovat všechny scripty, které budou vytvořeny v rámci projektu. Celý projekt bude na jednom z příložených DVD.

## 3.2 Analýza

### 3.2.1 Ovládání

První problém, který budu muset v rámci svého programu řešit, je, jak se uživatel bude pohybovat po scéně. K tomu nám pomůže již zmíněný plugin SteamVR a jeho nastavba VRTK, který dále budu rozšiřovat o vlastní skripty. Uživatel bude mít k dispozici dva ovladače. Tyto ovladače budou navrženy tak, aby je uživatel mohl prohazovat v rukou a jeho práci to nijak neovlivnilo. To znamená, že oba budou vykonávat identické funkce a žádná činnost nebude omezená tím, kterou rukou má být vykonána. Pokud bude program spuštěn na počítači, který nemá k dispozici zařízení Vive, bude přepnut do takzvaného debug modu, který umožní omezené ovládání (načítání scén a jejich procházení) pomocí klávesnice a myši.

### 3.2.2 Pohyb

Uživatel se po scéně bude pohybovat horizontálně pomocí teleportu na omezenou vzdálenost do oblastí, které by pro něj byly schůdné i v realitě, to znamená, že se nebude moci teleportovat například na zeď. Tlačítko teleportu se bude nacházet na touchpadu obou ovladačů. Vertikálně se bude pohybovat pomocí žebříků, které si může postavit tam, kam potřebuje (ve scéně bude vždy minimálně jeden žebřík). Bude také možné využít fyzického prostoru, který si uživatel vymežil.



### 3.2.3 Interakce

Hlavním viditelným prvkem aplikace bude interakce s předměty a prostředím kolem uživatele. Ovladače kromě funkce teleportu budou mít další funkce. Mezi ně bude patřit uchopení, laserové ukazovátko a přepínač do systemového menu. Bude kladen důraz na to, aby ovládání bylo co nejvíce podobné tomu, jak by člověk zacházel s předmětem v realitě. K uchopení předmětů bude využito tlačítko grip, aby se chycení co nejvíce podobalo reálnému uchopení. Laserové ukazovátko bude na tlačítku spouště a tím se bude podobat skeneru skladníků. Teleport bude namapován na touchpad, který bude moci mít i další funkce díky radialnímu menu. Systémové tlačítko bude mít standardní funkci VIVE. Každý předmět, se kterým lze nějak interagovat, bude zvýrazněn, pokud se ho uživatel dotkne ovladačem nebo na něj namíří ukazovátkem.

### 3.2.4 Uživatelské rozhraní

Uživatelské rozhraní (také UI) se bude skládat z uživatelské zprávy, která bude vždy přímo před pohledem hráče. Bude obsahovat aktuální instrukci pro hráče a zmizí po uplynutí nadefinovaného času. Pokud se uživatel podívá přímo na ovladač, ukáže se nápověda zobrazující jakou funkci má které tlačítko. V menu scéně bude k dispozici okno pro výběr tutoriálu ze souborového systému.

Všechny prvky uživatelského rozhraní se v Unity většinou řeší pomocí objektu zvaného canvas. Pro standardní aplikace, které nepodporují virtuální realitu, se všechny prvky UI připnou přímo na kameru hráče a následuje hráčův pohled. Ve VR toto řešení není možné a canvas musí být umístěn ve scéně a nenásleduje hráče. Pokud následuje hráče, může procházet zdmi, a to velice narušuje imerzi. Všechny prvky UI budou umístěny ve scéně kromě canvasu zobrazující instrukce, nebude nijak ošetřeno procházení zdí tohoto prvku.

### 3.2.5 Generování scénáře

Předtím, než uživatel bude moci projít tutoriálem, bude potřebovat zadefinovat XML soubor popisující scénu. Tato definice bude obsahovat informace o předmětech, které se budou ve scéně nacházet, název šablony, jazyk, seznam použitých uživatelských zpráv a jejich příslušný text a také popis příběhu, to znamená, co uživatel musí splnit za úkoly, aby postoupil dále.

Aplikace také bude obsahovat XML soubor se standardními texty, které mohou být použity v uživatelsky nadefinovaných zprávách. Tyto texty budou lokalizovány do třech jazyků: angličtina, čeština a němčina.

Po vybrání tutoriálu bude uživatel přenesen do jiné scény podle definice. Tato definice bude deserializována pomocí standardní knihovny C# System.Xml [3]. Abychom mohli XML deserializovat musíme mít připravené příslušné třídy v C# představující jednotlivé elementy XML. Po načtení a deserializování souboru můžeme podle této definice vygenerovat scénu v Unity. Pro každý vygenerovaný objekt poté uložíme referenci pro usnadnění pozdějšího přístupu, tím omezíme použití náročné metody Find(), která prochází celou hierarchii, dokud nenajde objekt splňující vyhledávací kritéria.

Pro umožnění generování objektů bude v aplikaci připravená sada prefabů a primitivních útvarů, které bude možné instancovat.

### 3.2.6 Úkoly

Definice bude moci obsahovat libovolný počet úkolů, které budou mít libovolné množství kritérií. Pokud uživatel splní aktuální úkol, vyvolá tím nadefinované události.

### 3.2.7 Události

Po splnění aktuálního úkolu se vyvolá libovolný počet událostí vztahených ke konkrétnímu úkolu. Události mohou být aktivace objektu, přehrání zvuku, změna uživatelské zprávy nebo ukončení aktuálního tutoriálu.

## 3.3 Implementace

Aplikace byla vyvíjena na verzi Unity 3D 5.5.2f1 (64-bit), novější verze obsahují bug zobrazení ovladačů, a proto projekt nebyl vylepšen na novější verzi enginu. Starší verze enginu nejsou otestovány a není známo, jak bude aplikace fungovat. Instalátor této verze je na přiloženém DVD.

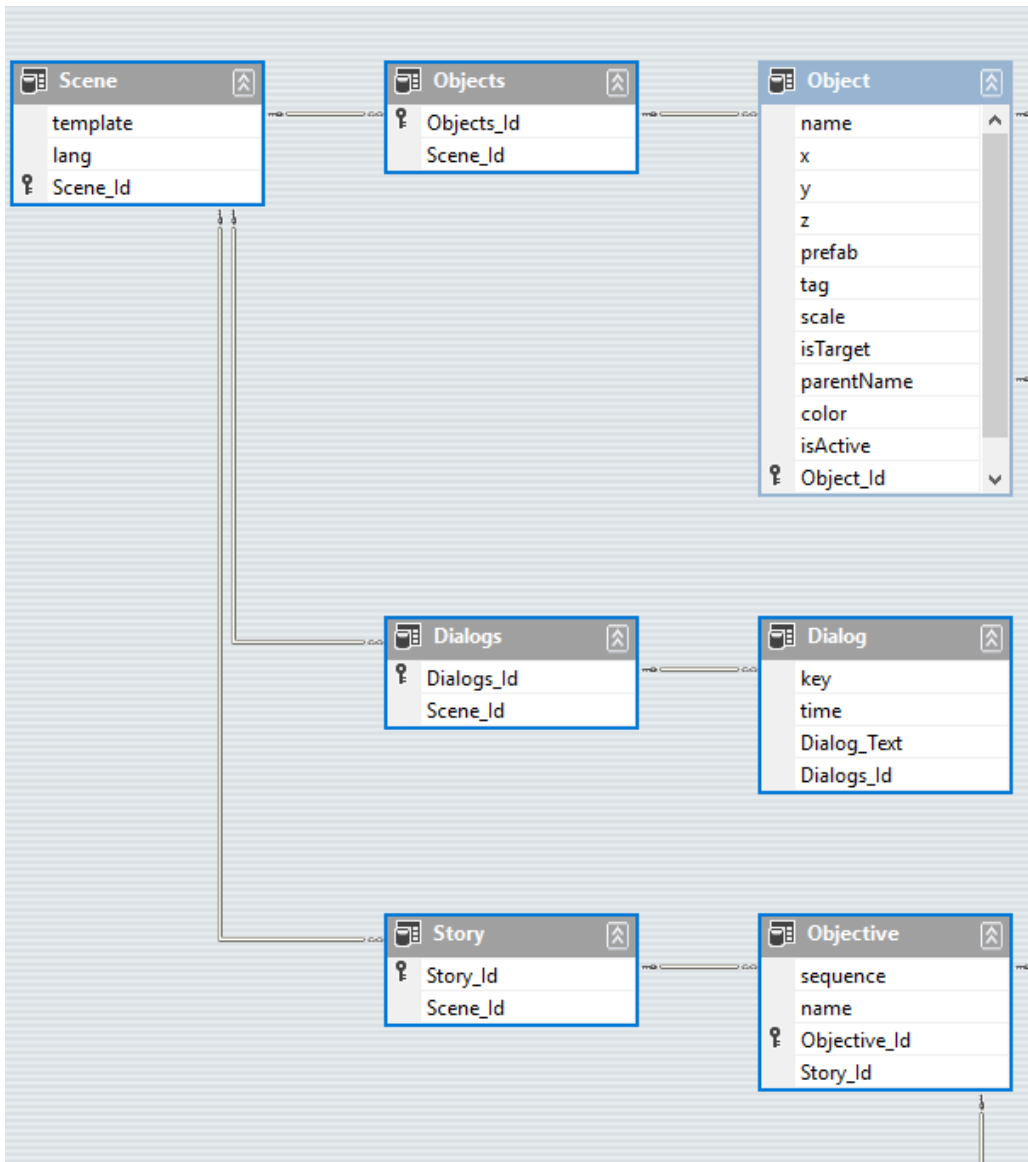
### 3.3.1 Lokalizace

Hlavním prvkem lokalizace je XML definice applicationResources a její příslušná C# třída ApplicationResources obsahující všechny standardní popisky přeložené do angličtiny, češtiny a němčiny. O překlad se pak stará třída

Localizer a především metoda localizeMessage, která pro zadaný klíč a jazyk najde v seznamu překlad.

### 3.3.2 Struktura definice scény

Scéna má atribut template, který představuje jaká šablona scény se má použít. Atribut lang představuje, v jakém jazyce scéna je, všechny příslušné standardní popisky se přeloží do definovaného jazyka. Scéna se skládá ze seznamu objektů, dialogů a příběhu. Struktura definice scény odpovídá diagramu 3.1.

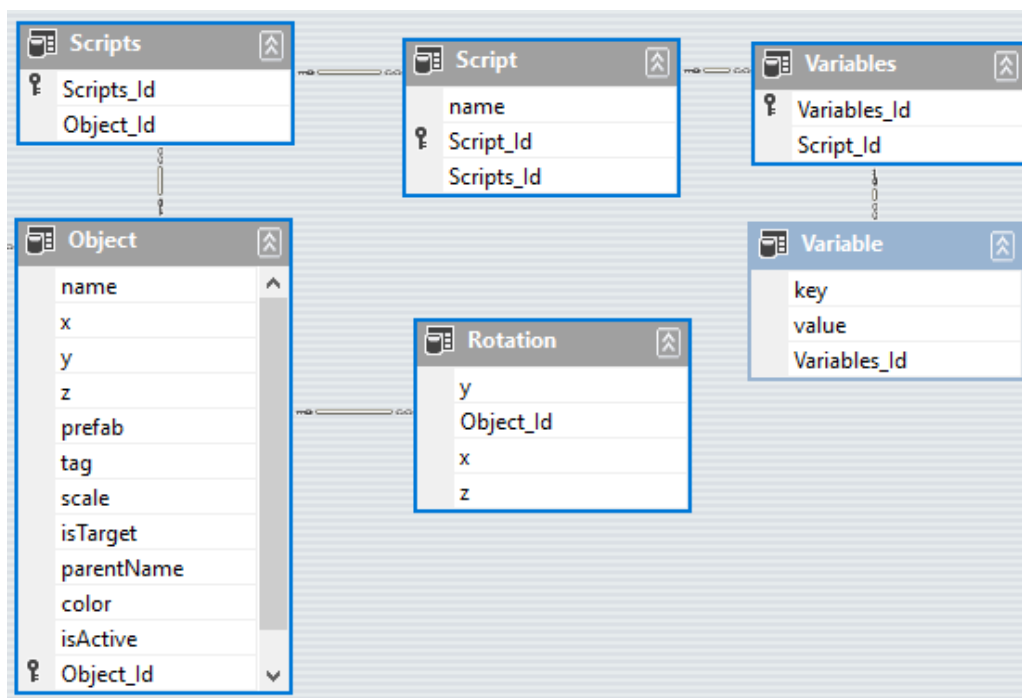


Obrázek 3.1: Struktura definice scény.

Objekt má následující atributy, všechny kromě name jsou nepovinné.

1. X, Y, Z - představují souřadnice
2. name - jméno objektu, mělo by být unikátní v definici
3. prefab - jméno prefabu, který se použije pro instanci objektu, pokud není specifikován, bude použita krychle rozměru 1\*1\*1 metrů
4. tag - značka, kterou bude objekt mít, může se využít například pro definování, jaký objekt se může přichytit k jinému, k dispozici jsou následující tagy: projectile, Box, Reso, Cubes, WoodenPin, UserTag01-05
5. scale - škála objektu, obsahuje číslo typu double, to znamená například formátu: 1.5
6. isTarget - booleanová hodnota vyjadřující, zda je objekt cílem jiného
7. parentName - název objektu, který bude aktuálnímu objektu přiřazen jako rodič
8. color - barva, která se přiřadí objektu, jsou jimi: black, blue, brown, green, grey, orange, pink, purple, red, teal, white, yellow
9. isActive - booleanová hodnota vyjadřující, zda je objekt aktivní při inicializaci scény

Objekt dále může obsahovat element rotation, který představuje, jak má být objekt otočen po osách XYZ (celočíselné hodnoty představující stupně) a seznam skriptů. Element script obsahuje seznam proměnných (variables) ty jsou definovány pomocí atributů key a value. Struktura objektu odpovídá diagramu 3.2

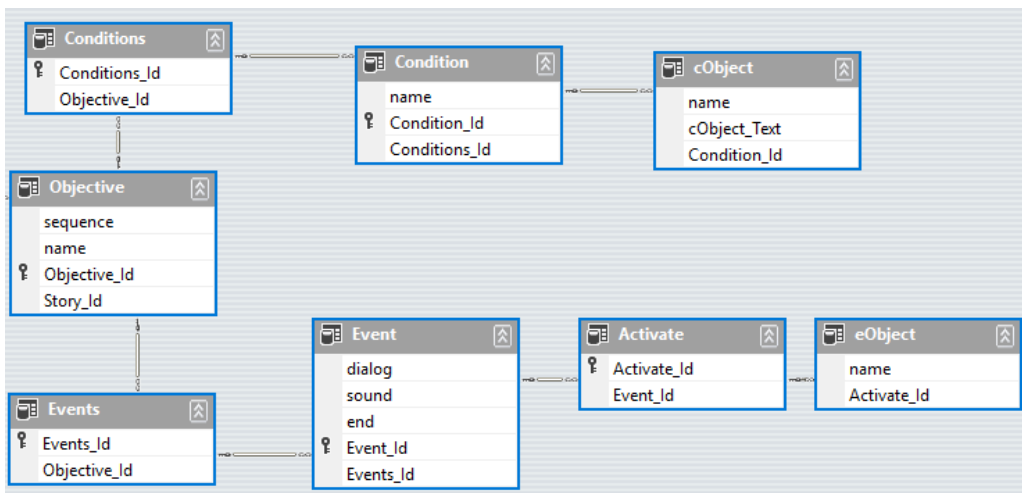


Obrázek 3.2: Struktura definice objektu.

Úkol, dále jen objective, je hlavní a jediný element elementu Story. Obsahuje seznam conditions (kritéria/podmínky) a events (události). Condition má atribut name a slouží jako kontejner pro element cObject, který je zjednodušeným druhem Objektu. CObjekt obsahuje atribut name představující jméno objektu, který byl dříve nedefinován, a text tohoto elementu představuje, co se s objektem má dít. Zatím je podporována pouze kondice full znamenající, že byl cílový objekt naplněn. Event má následující atributy:

1. dialog - klíč dialogu, který se má zobrazit
2. sound - jméno zvuku, který se má přehrát, ve stávající verzi je dostupný pouze zvuk succes
3. end - booleanová hodnota představující, zda se má ukončit aktuální scéna

Event dále obsahuje seznam eObjektů, které se mají aktivovat. EObject obsahuje pouze atribut name opět představující jméno objektu. Struktura příběhu odpovídá diagramu 3.3



Obrázek 3.3: Struktura definice úkolu.

### 3.3.3 Vybrání scény

Pro vybrání scény je použita placená knihovna OxOD. Tato knihovna byla následně upravena, aby se dala použít v prostředí virtuální reality a podporovala lokalizaci. Obsahuje třídy a assety umožňující načítání souborů ze systému souborů. Výběr souborů byl omezen pouze na soubory s .xml koncovkou. Po vybrání scény a potvrzení pomocí tlačítka load/nahrát proběhne načtení scény.

#### Načtení scény

Načítání scény obsluhuje třída SceneLoader. Obsahuje metody související s manipulací se scénami, tedy načtení load() a moveToScene(). Metoda load přijímá jako argument cestu k souboru, který načte pomocí metody load() třídy SceneDefinition. Ta cestu přijme, soubor otevře a deserializuje XML obsah souboru do příslušných objektů C#.

### 3.3.4 Generování scény

Pro generování scény slouží třída SceneGenerator, metody této třídy se volají po inicializaci načtením šablony zvolené scény z menu scény. Metoda Start() nejprve naplní všechny potřebné reference na instance ostatních tříd, které využívá, a plní jejich seznamy proměnných, poté volá metodu generateScene(). GenerateScene projde definici scény a vygeneruje jednotlivé objekty, podřadí objekty pod jejich rodiče a propojí reference cílových objektů.

## Generování objektu

Generování objektu nejprve zkusí vytvořit objekt podle jeho definovaného jména prefabu, pokud se toto nepodaří, přeruší se generování objektu a pokračuje se dalším. Po úspěšném vytvoření instance se objekt upraví podle dalších specifikovaných atributů. Nejprve se nastaví pozice podle souřadnic, pokud nejsou uvedeny, nastaví se hodnota 0, tedy střed scény. Následuje rotace, škála, přejmenování objektu, změna barvy, přidání scriptů, deaktivace, přidání tagu a na závěr je objekt přidán do seznamů pro usnadnění přístupu v pozdějším běhu aplikace.

### 3.3.5 Uživatelská interakce

Pro obsluhu uživatelského vstupu byla použita knihovna VRTK. Z této knihovny byly použity skripty na ovládání ovladačů, pohybu hráče po scéně, základní interakce s předměty a přichycení objektů. Pomocí těchto skriptů byla vytvořena šablona hráče, která je použita ve všech scénách. Obsahuje kromě modelů ovladačů také canvas s instrukcemi a všechny skripty související s hráčem.

### Systém dialogů

Pro zobrazení instrukcí a textů průvodce tutoriálem byl implementován jednoduchý systém dialogů. Obsluhu tohoto systému zajišťuje třída `InstructionSystem`. Zároveň se stará o překlad již existujících textů ve scéně. Všechny texty se zadávají podle klíčů. `InstructionSystem` při inicializaci všechny tyto texty najde a přeloží do příslušného jazyka z `applicationResources`.

### Plnění úkolů

Plnění úkolů obsluhují třídy `ObjectiveSystem` a `EventSystem`. `EventSystem` je třída, která se registruje u objektů, které mohou vyvolat nějakou událost, třída na tuto událost následně reaguje. `Event system` následně ověří, zda událost, která nastala, je předpokladem nějakého úkolu za pomoci `ObjectiveSystem`. Zatím je podporován pouze event `OnObjectSnappedToDropZone`, tedy pokud se objekt přichytil do cílové oblasti.

Třída `ObjectiveSystem` obsahuje seznam všech úkolů a metody pro jejich zpracování. Drží referenci na aktuální úkol a pomocí metody `checkConditions` kontroluje, zda vyvolaná událost nesplnila jednu z podmínek splnění úkolu.

## Vyvolání událostí

Pokud `ObjectiveSystem` zjistí, že všechny podmínky potřebné pro splnění úkolu nastaly, označí aktuální úkol za splněný a vyvolá přiřazené události. Mezi tyto události patří zobrazení nové instrukce pro uživatele, přehrání zvuku, který může být například signalizace úspěchu, a aktivace objektů potřebných k pokračování v příběhu. Dále může být vyvolána událost ukončení tutoriálu, která uživatele přenesení zpět do úvodní scény.

## Dostupné prefabry

Pro definování scény byla připravena velká sada prefabů a primitivních objektů, které může uživatel použít ve svém tutoriálu. Pokud chce uživatel prefab využít, musí použít v definici přesný název. Vytvořené cílové objekty obsahují skript definující list názvů nebo tagů, které označují, jaké objekty lze do nich přichytit, to se odráží v jejich názvu koncovkou `"_Name"` nebo `"_Tags"`. Pokud má prefab koncovku `"_Static"`, nelze s ním nijak pohybovat z místa. Pokud není uvedeno, že je prefab primitivní, obsahuje skript pro uživatelskou interakci a není potřeba ho přidávat. Následuje seznam a popis dostupných prefabů.

1. `Axe` - nástroj sekery, lze uchopit
2. `BedFrameFeet` - část postelového rámu na straně, kde by měly být nohy
3. `BedFrameFeetTarget_Name` - cíl předchozího prefabu
4. `BedFrameHead` - část postelového rámu na straně, kde by měla být hlava
5. `BedFrameHeadTarget_Name` - cíl předchozího prefabu
6. `BedFrameMiddleSupport` - prostřední příčka rámu postele
7. `BedFrameMiddleSupportTarget_Name` - cíl předchozího prefabu
8. `BedFrameSide` - venkovní příčka postele
9. `BedFrameSideTarget_Name/BedFrameSideTarget_Tags` - cíle předchozího prefabu
10. `BedFrameSupport` - podpěra rámu postele
11. `BedFrameSupportTarget_Name/BedFrameSupportTarget_Tags` - možné cíle předchozího prefabu



12. Board\_Static - tabule pro psaní křídou
13. Bolt - šroub
14. Box01 - dřevěná krabice
15. Box01\_Climbable - dřevěná krabice, která má žebříkovou příčku, to znamená, že si uživatel může vytvořit žebřík z beden
16. Box02 - černo-bílá krabice
17. BoxTarget\_Name/BoxTarget\_Tags - cíle pro krabice
18. BoxesStacked - blok naskládaných papírových beden
19. BreakableBox - rozbitelná bedna
20. Capsule01 - primitivní objekt kapsle (tobolky), tvarem připomíná tictac
21. Capsule01\_Scalable - objekt kapsle, který lze tvarovat ovladači
22. Case - otevíratelný kufřík
23. CaseWithWoodenPins - kufřík obsahující 36 dřevěných kolíčků
24. CeilingLight - stropní světlo, obsahuje zdroj
25. CeilingSupport/CeilingSupport\_Single - výztuhy stropů
26. ControlPannel\_Static - kontrolní panel
27. Cube01 - primitivní objekt kvádru
28. Cylinder01 - primitivní objekt válce
29. Door - otevíratelné dveře
30. Drawers - skříňka obsahující 4 funkční šuplíky
31. Drill - akumulátorová vrtačka
32. FireEscapeDoor - požární dveře
33. FireExtinguisher - hasicí přístroj
34. Forklift\_Static - vysokozdvizný vozík

35. Hammer01/Hammer02 - kladiva obsahují nezničitelnou vazbu na ovladač, to znamená, že narozdíl od jiných předmětů tyto uživatel neupustí při silném nárazu
36. HandLamp - ruční lampa, obsahuje světlo
37. Ladder - žebřík
38. MetalRail - ocelové zábradlí
39. OrangeBarrel - silniční kužel
40. PalletJack\_Static - ruční paletizér
41. Pipe\_Wrench - šoupací kleště
42. Plane01 - primitivní objekt plochy
43. Pliers - kombinační kleště
44. Quad01 - primitivní objekt menší plochy
45. Reflector - reflektor vyzařující červené světlo na 2 strany, nemá skript pro uživatelskou interakci
46. Reso01 - kostička podobná legu
47. Reso01\_snappable - kostička Reso s přednastaveným cílem nad sebou
48. Reso02 - menší kostička Reso
49. Reso01Target\_Name - cíl pro Reso01
50. RobotArmWelder - výrobní robot, nemá skript pro uživatelskou interakci
51. Saw - pila
52. ScrewCross - křížový šroub
53. ScrewSingle - šroub s drážkou
54. ScrewdriverCross - křížový šroubovák
55. ScrewdriverSingle - obyčejný šroubovák
56. ShelfGT01\_Static/ShelfGT02\_Static/ShelfGT03\_Static - velké skladové police, plné

57. ShelfLG\_Static - střední skladové police
58. ShelfSM\_Static - malá skladová police
59. Single-TaskArm - robotická animovaná ruka, neobsahuje uživatelskou interakci
60. SpawnedSound - objekt obsahující přehratelný zvuk
61. Sphere01 - primitivní objekt koule
62. TargetContainer\_Name - obecný cílový objekt tvaru krychle
63. ToolCartA01/ToolCartB01/ToolCartC01
64. woodenBoard1/woodenBoard2/woodenBoard3 - dřevěné desky, neobsahují uživatelskou interakci
65. WoodenCrate01 - velká dřevěná krabice
66. WoodenPallet - dřevěná paleta
67. WoodenPalletWithBoxes - dřevěná paleta naložená bednami
68. WoodenPin - dřevěný kolíček, pozor velice malý objekt
69. WoodenPinTarget\_Tags - cíl předchozího prefabu
70. WrenchCombination/WrenchOpen - klíče

### **Šablony scény**

K dispozici je několik šablon scén, mezi ně patří:

1. Workshop - dílna obsahující nástroje a podstavce
2. Greathall - velká hala vhodná pro výrobní linku nebo sklad
3. Forest - les
4. Plains - pláň
5. main\_scene\_1 - testovací scéna obsahující sadu Reso a roboty

### **Ukázkové tutoriály**

Aplikace obsahuje dva ukázkové scénáře a pět testovacích.

Prvním jednoduchým tutoriálem je mixování zboží na paletě podle předpisu. Uživatel má za úkol naskládat na paletu bedny tak, jak mu vyznačuje aplikace. V tomto tutoriálu lze dobře vidět, jak pracovat v definici s tagy, ty se tu využívají dva (UserTag01, UserTag02), protože jsou pouze dva druhy beden. Při načtení je uživatel uvítán a dostane první instrukci, jak má postupovat. Po naložení každé vrstvy dostává další instrukci a přehrává se zvuk představující úspěch. Na konci je uživatel přenesen do úvodní scény.

Druhý složitější tutoriál je složení rámu postele. Scénář se odehrává v menší scéně než předešlé, a to v dílně. Zde jsou vždy dvě skříňky s nářadím a podstavce různých rozměrů, na kterých by uživatel měl vykonávat svůj úkol. Při načtení je uživatel opět přivítán a je mu dána první instrukce. Složitost tohoto tutoriálu spočívá v obtížnosti manipulace s velmi malými objekty.

## 4 Prostor pro budoucí vývoj

Práce je opravdu pouze framework a pro plnohodnotnou aplikaci je potřeba pokračovat ve vývoji.

### Uživateleské rozhraní

Jedním z prvních bodů, na kterých bude potřeba zapracovat, je uživatelské rozhraní. V UI chybí možnost jakéhokoliv nastavení aplikace, přechodu do hlavního menu nebo ukončení celé aplikace. Dále chybí přehled úkolů, které uživatel již splnil a které mu ještě zbývají. Zobrazení času a relativního skóre.

### Interakce

Velký prostor pro rozšíření nabízí interakce s předměty. V aplikaci je k dispozici sortiment nástrojů, ale kromě pokládání, házení nebo mlácení s nimi nelze jinak pokročile zacházet, např. šroubovákem šroubovat šroubky. Nástroje a předměty by mohly být zničitelné nebo poškoditelné a za to by uživatel byl hodnocen, pokud by s nimi zacházel nešetrně. Aplikace také obsahuje několik přepravníků jako je vysokozdvíhový vozík nebo paletizér. Budoucí vývoj bude směřovat k tomu, aby uživatel mohl využívat i tyto nástroje pro svojí práci např. ve skladu. Možnost uživatelsky definovatelných tlačítek na radiálním menu touchpadu je samozřejmostí.

### Události

Další možná část vývoje je rozšíření možných událostí, které může uživatel definovat, ať jsou to události související s aktuálním úkolem nebo události, které se provedou, udělá-li uživatel něco špatně, např. vzdálí-li se příliš od svého pracoviště nebo poškodí nástroje.

### Objekty

Dále bude potřeba rozšířit dostupné prefaby o další objekty jakou jsou např. modely aut a jejich součástí. Reso se zatím skládá pouze ze dvou různých kusů, ale je možné vytvořit hotový balík součástek, ze kterých by uživatel mohl postavit opravdu cokoliv.

## 5 Závěr

V teoretické části práce bylo cílem seznámit se s dostupnými zařízeními pro virtuální realitu a aplikacemi pro vývoj. Dále prozkoumat již stávající aplikace a všechny poznatky sepsat. Tyto body byly splněny v plném rozsahu.

Pro účel práce jsem vybral engine Unity 3D. Rozhodnutí bylo těžké, protože se nedá jednoznačně říci, který z engineů je lepší. Hlavním aspektem výběru pro mě byl jazyk, ve kterém lze do engineu psát, v případě Unity C# nebo JavaScript. Dalším rozhodovacím aspektem byl verzovací systém, jaký určitý engine používá. V případě Unity je to Collab, který běží kompletně v cloudu a tedy odstiňuje uživatele od starostí s infraskruturou a vše je na jednom místě. Velkou pomocí je v Unity i již zmíněný Asset Store obsahující stovky zdrojů a knihoven, které jsou zcela zdarma a nebo za nízkou cenu, což se o Unreal Engineu říci nedá. Komunita kolem Unity je z velké části tvořena indie vývojáři, kteří si své know-how tolik neudržují jen v rámci svého studia, ale o poznatky se dělí na diskuzních fórech a diskutují o problémech, které řeší. To může být velkou výhodou při řešení podobných problémů.

Pro vývoj jsem vybral zařízení HTC Vive. Hlavním důvodem je dostupnost zařízení na univerzitě a ve firmě Aimtec, kde je možné aplikaci testovat a prezentovat. Zařízení mě také mnohem více oslovilo svými možnostmi využití prostoru a rychlou odezvou.

Poté se v praktické části vyvíjela frameworková aplikace pro tvorbu tutoriálů v prostředí virtuální reality. Aplikace splňuje zadání, ale prostor pro vylepšení je opravdu velký a nedá se říci, že by byla kompletní.

V průběhu práce jsem rozvíjel své schopnosti práce s Unity a programování v jazyce C#, který pro mě byl novinkou. Vývoj této aplikace pro mě byl velice zajímavý a v budoucnosti mám v plánu v něm pokračovat. Během vývoje se vyskytlo několik zdánlivě triviálních problémů, které stály více času, než by měly. Jako příklad uvedu rozdíl datových typů "Object" a "object", chyba v kódu nebyla na první pohled patrná a příčinu problému jsem tedy dlouho hledal jinde, než byla. Další z problému byla například nemožnost zjištění aktuální reference na objekt v Unity a následný přístup do paměti, kde se již objekt nenalézal, ale ostatní objekty na něj stále mohly udávat

referenci. Tím přistupovaly do špatné části paměti, což dělalo celou aplikaci nestabilní a často došlo k pádu Unity editoru.

Tato práce vznikla na základě spolupráce Zápodočeské univerzity a firmy Aimtec a.s., které poskytly zázemí pro vývoj. Na univerzitě byla dostupná obě zařízení HTC Vive a Oculus Rift. Aplikace bude prezentována 20. září na konferenci TAL a bude dále upravována pro potřeby zákazníků společnosti Aimtec a.s.

# Uživatelská příručka

Před spuštěním aplikace je nutné mít nainstalovaný a kalibrovaný headset HTC Vive. Aplikace se pustí pomocí VR\_Trainer.exe nebo VR\_Trainer\_debug.exe. Definování scény viz implementace.



# Literatura

- [1] *Mecas ESI IC.IDO* [online]. Mecas ESI, 2017. [cit. 2017/03/09]. Dostupné z:  
<https://www.esi-group.com/cz/softwarova-reseni/virtualni-realita>.
- [2] *Image synthesis* [online]. [cit. 2017/03/09]. Dostupné z: <http://www.cs.kent.edu/~farrell/cg02/lectures/rendering/render.htmlr>.
- [3] *Wiki.Unity3D.XmlSerializer* [online]. Wiki.Unity3D, 2013. [cit. 2017/03/09].  
Dostupné z:  
[http://wiki.unity3d.com/index.php?title=Saving\\_and>Loading\\_Data:\\_XmlSerializer](http://wiki.unity3d.com/index.php?title=Saving_and>Loading_Data:_XmlSerializer).