

Sem vložte zadání Vaší práce.





**FAKULTA  
INFORMAČNÍCH  
TECHNologiÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Virtuální historický průvodce - modul virtuální reality**

***Helena Pavlíková***

Katedra softwarového inženýrství

Vedoucí práce: Ing. Jiří Chludil

7. května 2018



---

## Poděkování

Děkuji vedoucímu práce.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické 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. § 46 odst. 6 tohoto zákona 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 7. května 2018

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2018 Helena Pavlíková. 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**

Pavlíková, Helena. *Virtuální historický průvodce - modul virtuální reality*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.



---

## Abstrakt

Tato bakalářská práce je součástí týmového projektu Virtuální historický průvodce. Cílem práce je návrh modulu aplikace pro virtuální realitu. Tento modul zobrazí městskou scénu rozmístěním 3D modelů budov. Jednotlivé modely jsou ukládány v databázi a práce řeší jejich efektivní zobrazování za různých podmínek. Aplikace dokáže zohlednit uživatelské vstupy a podle potřeby zobrazí modely v různých atmosférických podmínkách, například v noci, ráno, ve sněhu či za deště. Díky své interaktivitě může aplikace sloužit nejen jako výukový materiál vzdělávání v oblasti historie, ale může být přínosná i pro obory památkářství či tvorby her pro technologie VR.

**Klíčová slova** VR aplikace historický průvodce, vizualizace architektury, databáze 3D modelů domů, památková péče, Unity3D, virtuální realita

---

## Abstract

This bachelor thesis is a part of a team project called Virtual Historical Guide. The goal of this thesis is to design a prototype of a virtual reality module for this application. This module shall visualize a city scene by putting 3D models of buildings in space. The individual models are saved in a database and this thesis deals with their effective visualisation under different settings.

The application can take user input and according to it visualize the city under different atmospherical conditions, such as in night, morning, snowy, rainy and so on. The application is interactive and therefore can become a valuable education material in history lessons. It can also prove useful for cultural heritage architects or even for creating games for VR.

**Keywords** VR application historical guide, visualization of architecture, database of 3D models of houses, historical building preservation, Unity 3D, virtual reality

---

# Obsah

Odkaz na tuto práci . . . . .	vi
<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 Analýza podobných řešení . . . . .	5
2.1.1 3DCityDB . . . . .	5
2.1.1.1 Softwarové požadavky . . . . .	5
2.1.1.2 Vlastnosti . . . . .	6
2.1.1.3 Rozdíly ve funkcionalitě . . . . .	7
2.1.2 Virtuální stará Praha . . . . .	7
2.1.2.1 Softwarové požadavky . . . . .	7
2.1.2.2 Vlastnosti . . . . .	7
2.1.2.3 Rozdíly ve funkcionalitě . . . . .	7
2.1.3 Etruské hrobky . . . . .	8
2.1.3.1 Vlastnosti . . . . .	8
2.1.3.2 Rozdíly ve funkcionalitě . . . . .	9
2.1.4 Shrnutí . . . . .	9
2.2 Analýza použitých technologií . . . . .	11
2.2.1 Unity . . . . .	11
2.2.1.1 XR v Unity . . . . .	11
2.2.1.2 Nastavení podpory pro VR v Unity . . . . .	11
2.2.2 Virtual Reality Toolkit . . . . .	12
2.2.2.1 Simulátor VRTK . . . . .	12
2.2.2.2 Ovladače pohybu . . . . .	12
2.2.2.3 VR ukazatel . . . . .	12
2.2.3 C#, .NET, MONO . . . . .	13
2.2.3.1 C# . . . . .	13

2.2.3.2	.NET Framework . . . . .	13
2.2.3.3	Mono . . . . .	13
2.2.4	Swagger . . . . .	15
<b>3</b>	<b>Návrh</b>	<b>17</b>
3.1	Návrh architektury . . . . .	17
3.1.1	Prezentační vrstva . . . . .	17
3.1.2	Business vrstva . . . . .	17
3.1.3	Datová vrstva . . . . .	18
3.2	Funkční a nefunkční požadavky . . . . .	18
3.2.1	Funkční požadavky . . . . .	18
3.2.2	Nefunkční požadavky . . . . .	20
3.3	Případy užití . . . . .	20
3.4	Návrh použitého API . . . . .	22
3.4.1	Položky API . . . . .	22
<b>4</b>	<b>Implementace</b>	<b>27</b>
4.1	Napojení na datové úložiště . . . . .	27
4.1.0.1	Přímé propojení s databází . . . . .	27
4.1.0.2	JSON v Unity . . . . .	28
4.1.0.3	Mono a HTTPS . . . . .	29
4.2	Vykreslení v Unity . . . . .	30
4.2.1	Textury . . . . .	30
4.2.2	StreamingAssets . . . . .	31
4.3	Grafické uživatelské rozhraní . . . . .	32
4.3.1	Použitelnost GUI pro VR . . . . .	32
4.3.2	Prvotní návrh . . . . .	32
4.3.2.1	Celkový koncept . . . . .	32
4.3.2.2	Výběr lokality . . . . .	32
4.3.2.3	Výběr počasí . . . . .	33
4.3.2.4	Administrátorský režim . . . . .	33
4.3.3	Uživatelské testování . . . . .	34
4.4	Uživatelská příručka . . . . .	34
4.4.1	Instalace na PC . . . . .	34
4.4.2	Ovládání simulátoru . . . . .	35
4.4.2.1	Mód ovládání kamery . . . . .	35
4.4.2.2	Mód pohybu rukou . . . . .	36
	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>39</b>
	<b>A Seznam použitých zkratk</b>	<b>41</b>
	<b>B Obsah přiloženého USB</b>	<b>43</b>

C Scénář uživatelského testování	45
D Dotazník pro uživatelské testování	47



---

## Seznam obrázků

2.1	Ukázka běhu aplikace 3DCityDB, model města Berlín . . . . .	6
2.2	Virtuální stará Praha, na levé části jsou vidět modely bez textur .	8
2.3	Ukázka běhu aplikace Virtuální stará Praha . . . . .	9
2.4	Aplikace vizualizující archeologické naleziště etruské hrobky v Unity 3D [3] . . . . .	10
2.5	Nastavení VRTK . . . . .	14
3.1	Návrh architektury celé aplikace . . . . .	19
3.2	Diagram případů užití . . . . .	22
3.3	API v YAML . . . . .	25
3.4	API v JSON . . . . .	26
4.1	Wireframe pro vybírání místa . . . . .	33
4.2	Wireframe pro vybírání počasí . . . . .	34
4.3	Wireframe pro režim architekta . . . . .	35





---

# Úvod

Technologie zabývající se vizualizacemi pro virtuální realitu jsou jedny z nejprogresivnějších odvětví dnešního IT sektoru. Mají široké spektrum využití, interaktivními hrami počínaje a architektonickými návrhy konče. Tyto technologie mají navíc veliký potenciál do budoucna, kdy by se mohly například stát součástí výuky na školách či běžnou domácí zábavou.

Projekt historického průvodce v sobě nese přínos jak pro odbornou, tak pro širokou veřejnost. Vizualizace středověkých měst názorně přiblíží lidem naši historii a umožní jim procházet ulice měst tak, jak vypadaly před stovkami let. Zároveň ale tato práce nabízí využití i pro obory architektury nebo památkové péče, kde může napomoci vizualizovat staré či budoucí městské oblasti. Téma také úzce souvisí s počítačovou grafikou a s vývojem počítačových her, takže jistě osloví odborníky z této oblasti. Právě toto mnohostranné využití bylo motivací autorky k výběru daného tématu.

Tato práce je součástí velkého projektu, který bude zpracovávat více studentů několik let. Další jeho části, jakými jsou například databázové jádro, navigace či serverová optimalizace jsou pokryty v jiných semestrálních a bakalářských pracích.



---

## Cíl práce

Tato bakalářská práce si klade za cíl využít technologie VR (virtuální realita) k zobrazení částí historického města. Uživatel výsledné aplikace bude s její pomocí přenesen na konkrétní lokaci v daném městě a kolem něj budou vygenerovány nejbližší domy s ohledem na dané počasí a denní dobu. K samotné implementaci bude použito vývojové prostředí Unity 3D ve spolupráci s databází modelů budov. Komunikace mezi nimi bude zajištěna přes REST API (Representational State Transfer Representational State Transfer, neboli architektura rozhraní).

Téma přináší v dané oblasti důležité inovace jak po stránce jeho využití, tak po stránce technologického řešení. Protože ve virtuální realitě má uživatel omezený prostor, obvykle se hry a vizualizace pro ně vytvořené nesnaží zachytit větší okruh než několik metrů čtverečních. Výsledná aplikace ovšem bude mít k dispozici mnohem větší množství dat, které si bude ukládat v databázi a z ní bude vždy zobrazovat jen konkrétní požadovanou oblast. Takto bude zajištěna co největší efektivita programu i pro tak velké množství dat, jakým je databáze modelů domů v celém městě.



## Analýza

### 2.1 Analýza podobných řešení

V této kapitole budou rozebrány především aplikace, které se také zabývají problematikou vizualizace městských scén. Kromě měst je ovšem možné používat vizualizaci pro VR také pro jednotlivé objekty či přímo v muzeích. Jeden z takovýchto projektů bude blíže nastíněn na konci této sekce.

Rozsáhlé 3D modely měst bývají používány pro účely rozvoje urbanismu a infrastruktury a proto jsou často vytvářeny tak, aby poskytovaly pokud možno co nejúplnější pohled na město. Proto se od tématu této bakalářské práce odchyľují hlavně snahou zobrazit co nejvíce budov najednou převážně z pohledu ptačí perspektivy bez snahy pro optimalizaci a vyšší LoD (level of detail, neboli úroveň detailu jednotlivých modelů). Příklady takovýchto řešení jsou mimo jiné projekty Vizicities, WRLD3D či 3DCityDB. Podrobněji zde bude rozebrána poslední jmenovaná z těchto aplikací.

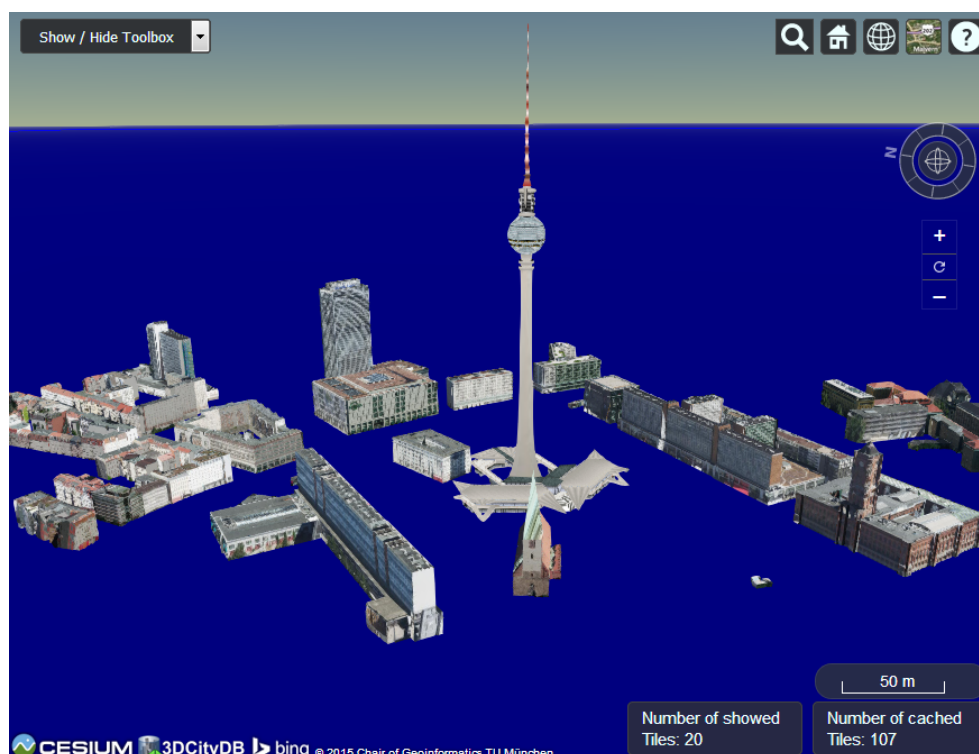
#### 2.1.1 3DCityDB

Jedním z obsahově nejbližších existujících řešení problému, se kterým se potýká projekt Virtuální průvodce, je aplikace 3DCityDB. Jedná se o projekt zaštiťovaný Technickou univerzitou v Mnichově, konkrétně na katedře geoinformatiky. 3DCityDB je, jak již název napovídá, velká databáze modelů budov, které dohromady tvoří konkrétní městskou scénérii. Tato aplikace se ujala v praxi hlavně v situacích, kdy měla samotná města zájem na uchování své stávající podoby ve 3D modelu. Zároveň se také používá v různých výzkumných projektech. Tuto aplikaci používají mnohá města převážně v Německu či v Nizozemí. [1]

##### 2.1.1.1 Softwarové požadavky

3DCityDB bylo vyvinuto v souladu se standardem CityGML (GML je zkratka pro geografický značkovací jazyk), což je speciální podmnožina XML (neboli

## 2. ANALÝZA



Obrázek 2.1: Ukázka běhu aplikace 3DCityDB, model města Berlín

rozšiřitelný značkovací jazyk) vhodná pro práci v oboru geografie. Zároveň pak pro svůj back-end aplikace využívá RDBMS (systém pro správu relačních databází) typu Oracle, případně také speciální verzi PostgreSQL, tzv. PostGIS, který podporuje prostorové a lokační dotazy. Jakožto front-endový nástroj zde bylo zvoleno WebGL, vizualizace jednotlivých měst je tedy možné zobrazovat ve webovém prohlížeči. Aplikace podtřebuje ke svému fungování JRE (Java Runtime Environment), je Freeware a Opensource. [1]

### 2.1.1.2 Vlastnosti

Projekt 3DCityDB si klade za cíl vizualizovat města v co nejucelenější podobě. Z toho také vyplývají jeho vlastnosti. Nepokládá přílišné nároky na LoD, přestože podporuje i modely s vyšší úrovní detailu (např. berlínská televizní věž má detailní model s velkým množstvím trojúhelníků, ovšem obytné domy v jejím okolí jsou převážně v podobě kvádrů s texturami, viz obr. 2.1 ). Tato aplikace pojme velké množství modelů (údajně až několik milionů), které zobrazuje postupně po částech (tzv. tiling strategy). Za následek to ovšem má zobrazení budov trochu chaotickým způsobem, kdy se při přibližování a odalování některé modely nově načítají a některé stále zůstávají.[1]

### 2.1.1.3 Rozdíly ve funkcionalitě

Tato aplikace má sice blízko k projektu Virtuální průvodce, ovšem zaměřuje se hlavně na celkové vizualizace měst, zejména pro potřeby městské správy, rozvoje infrastruktury apod. Tato bakalářská práce se naproti tomu bude zabývat zobrazením pouze menšího záběru z velké databáze, díky čemuž by mělo být možné použít modely s vyšším LOD. Zároveň si také tato práce klade za cíl větší obecnost (budeme pracovat s klasickými 3D formáty, jako je .obj a ne specifickými geografickými formáty typu GML) a vyšší uživatelskou přívětivost.

### 2.1.2 Virtuální stará Praha

Projekt Virtuální Stará Praha vznikl v roce 1999 pod vedením profesora J. Žáry jakožto spolupráce matematicko-fyzikální fakulty Univerzity Karlovy a katedry počítačové grafiky na FEL, ČVUT. Jedná se o jednoduchou vizualizaci scény v nejbližším okolí pražského Malostranského náměstí. [2]

#### 2.1.2.1 Softwarové požadavky

Aplikace je určená pro webové prohlížeče, je ovšem potřeba nainstalovat plugin VRML (Virtual Reality Modeling Language, jazyk pro modelování VR) prohlížeč, například Cortona VRML nebo též Cosmo Player VRML. Mezi další technické požadavky patří internetový prohlížeč s funkčním JavaScriptem a Java Applets. [2]

#### 2.1.2.2 Vlastnosti

Projekt VSP nabízí možnost procházení městské scény, která se dynamicky přizpůsobuje pozici uživatele. Z tohoto důvodu je načítání jednotlivých modelů prováděno v několika fázích podle sektoru, ve kterém se uživatel nachází. Tyto sektory se stále znovu vyhodnocují při pohybu městem pomocí algoritmu, který nad sektory provádí množinové operace a podle toho zobrazuje modely. Nejvzdálenější budovy tedy nebudou načteny vůbec, bližší jsou vyobrazeny pouze tvarově bez textur (viz obr. 2.2) a teprve ty nejbližší budovy jsou načteny úplně (obráz. 2.3). [2]

#### 2.1.2.3 Rozdíly ve funkcionalitě

Oproti tomu tato bakalářská práce nebude pracovat s dynamickým načítáním scény vyvolaným pohybem uživatele. Díky svému zaměření na VR není nutné importovat z databáze víc domů, než těch v bezprostředním okolí uživatele, neboť dále technologie VR uživateli nedovolí dojít. Bude sice možné provést posun do jiného sektoru města, ovšem tím budou také opuštěny všechny dosud načtené modely a namísto toho se uživatel přenesení do úplně nově vygenerované scény.



Obrázek 2.2: Virtuální stará Praha, na levé části jsou vidět modely bez textur

### 2.1.3 Etruské hrobky

Tématikou vizualizace se taktéž zabývají archeologové. Možnosti virtuální reality jim umožňují rekonstruovat velká archeologická naleziště a následně je zblízka procházet. Takto byly například zmapovány v roce 2016 etruské hrobky Bartoccini a Bettini. Hrobky byly naskenovány a tím byly získány jejich 3D modely.

#### 2.1.3.1 Vlastnosti

Naskenované modely měly ovšem příliš velké množství polygonů na to, aby bylo možné s nimi efektivně pracovat, a proto byly v programu 3Ds Max následně zjednodušeny. Dále jim byly přiděleny textury a byly naimportovány do Unity. Zde byla pak vytvořena jednoduchá VR aplikace pro jejich vizualizaci, která byla ovládána pomocí technologií Oculus a Kinect. Také zde byla implementována interaktivní místa, která uživateli zobrazila malé okno s několika zajímavostmi (viz obr. 2.4 ). [3]





Obrázek 2.3: Ukázka běhu aplikace Virtuální stará Praha

### 2.1.3.2 Rozdíly ve funkcionalitě

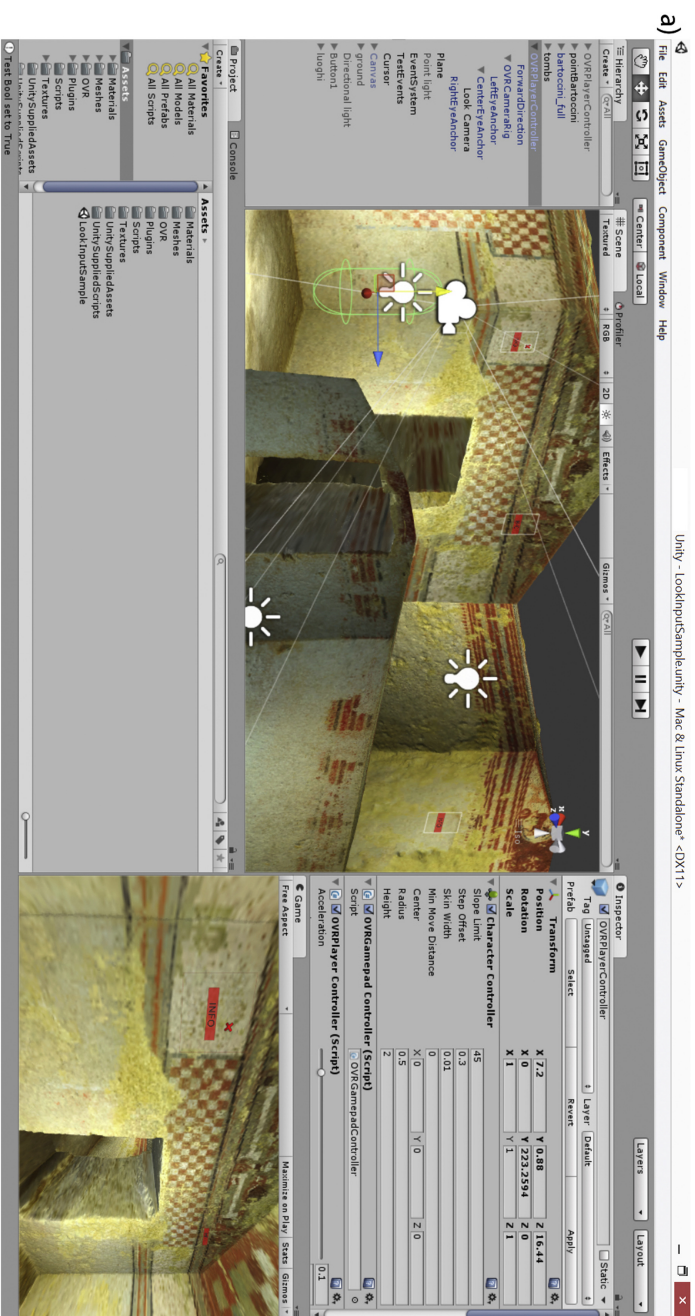
Od této bakalářské práce se tento typ vizualizací liší hlavně použitím jediného statického modelu (hrobky, archeologického naleziště apod.). Aplikace Virtuální historický průvodce bude naproti tomu zobrazovat mnoho modelů, které bude dynamicky získávat z databáze.

### 2.1.4 Shrnutí

Všechny výše zmíněné aplikace mají blízko k tématu této bakalářské práce, každý z nich se ovšem trochu odlišuje funkcionalitou a hlavně cílovou skupinou.

Z hlediska cílové skupiny je této práci nejbližší projekt vizualizace etruských hrobek, protože se taktéž jedná o aplikaci, která je určena jednak široké veřejnosti za účelem ponaučení a objevování starých světů, jednak odborníkům v oblasti historie, kterým zjednodušuje bádání. Proto se od tohoto projektu tato bakalářská práce inspiruje v možnosti interakce s okolím a zobrazování modálních oken s různými zajímavostmi o daném objektu.

Projekty 3DCityDB a Virtuální Stará Praha mají blíže k této bakalářské práci tématikou, ovšem jsou zaměřeny na jiné cílové skupiny. 3DCityDB slouží hlavně městům jako takovým, naproti Virtuální Stará Praha je vizualizace pro každého. Problematické je zde u obou těchto projektů ovšem ovládání



## 2. ANALÝZA

Obrázek 2.4: Aplikace vizualizující archeologické naleziště etruské hrobky v Unity 3D [3]

a přístupnost. Oba tyto projekty vyžadují dosti složité nastavení webového prohlížeče, což může pro určité uživatele představovat jistou překážku. Navíc pak ani po úspěšném spuštění ani jedné z těchto aplikací není úplně intuitivní jejich ovládání.

Analýza těchto řešení tak vnáší nejen vzhled do funkcionality podobných aplikací, ale taktéž ukazuje, že při návrhu uživatelského rozhraní je u podobných aplikací nutné vydat se jiným směrem. V kapitole 4.3 bude blíže rozebráno zvolené řešení návrhu uživatelského rozhraní, které se bude snažit být uživatelsky přijatelnější a co nejvíce intuitivní.

## 2.2 Analýza použitých technologií

V této sekci budou rozebrány jednotlivé technologie, které byly zvoleny pro implementaci projektu Virtuální historický průvodce.

### 2.2.1 Unity

Unity 3D je robustní software, který slouží především k vývoji počítačových her. Nabízí široké využití jak pro programátory, tak pro grafiky nebo i game designéry. Orientuje se převážně na 3D aplikace (odtud název), ovšem umožňuje i vývoj ve 2D. V dnešní době patří mezi nejzákladnější výbavu vývojářů aplikací pro VR.

#### 2.2.1.1 XR v Unity

Unity podporuje všechny technologie sdružené s VR, které se souhrnně označují "XR". Řadíme sem jednak klasické VR technologie, jednak technologie rozšířené reality (augmented reality - AR) či smíšené reality (mixed reality - MR). Mezi konkrétní technologie, které je možné sem zařadit, jmenujme například Oculus, HTC VIVE, Microsoft Hololens, Playstation VR, Cardboard a další.[5]

#### 2.2.1.2 Nastavení podpory pro VR v Unity

Aplikace v Unity může být v nastavení upravena pro VR mód. Pokud je v Player settings povolena podpora VR, je možné nastavit seznam všech technologií, které bude aplikace podporovat v tom pořadí, jak jsou v tomto seznamu uvedeny. Toto se dá nastavit jednak v GUI (grafickém uživatelské rozhraní) Unity, jednak v kódu pomocí proměnných `XRSettings.supporteddevices`. Není možné přidávat nová zařízení dynamicky, seznam technologií musí být úplný již při kompilaci aplikace.

Tímto nastavením se změní pozice kamery tak, aby podporovala HMD (head mounted display, neboli VR brýle), což mimo jiné zajistí, že pohledová matice a matice projekce se přizpůsobí pohybům hlavy. Všechny potřebné

## 2. ANALÝZA

---

informace o pozici hlavy se zajistí ještě před samotným renderováním obrazu, což zabrání různým negativním vedlejším efektům, mezi které patří například nevolnost či ztráta rovnováhy.

Vývojáři jednotlivých VR technologií vytvořili pro uživatele Unity speciální balíčky, které je možno stáhnout zdarma v Unity Asset store. Tyto balíčky obsahují nastavení pro vývoj ve VR.[5]

### 2.2.2 Virtual Reality Toolkit

VRTK neboli Virtual Reality Toolkit je sada skriptů, které usnadňují vývoj aplikací pro VR v Unity 3D. VRTK momentálně plně podporuje technologie SteamVR a Oculus, do budoucna bude zavádět i podporu technologií Daydream a XimmerseVR. Mezi jeden z nejpraktičtějších aspektů jejich využití patří testování VR módu aplikace bez nutnosti zapojení VR headsetu díky vestavěnému VR simulátoru. VRTK je uvedeno na trh pod licenci MIT [11], která umožňuje volné šíření i modifikaci tohoto systému.

#### 2.2.2.1 Simulátor VRTK

VRTK SDKManager je sada skriptů, která obsahuje základní ovládací prvky pro simulátor VR v Unity. Jedná se o přenositelný samostatný balíček, který je možno importovat z Unity Asset Store do jakéhokoli projektu v Unity. Tím získáme balíček VRTK, ve kterém jsou předpřipravené ukázkové scény se simulátorem. Objekt simulátoru zvaný VRTK můžeme z těchto scén překopírovat do vlastních scén, je ovšem nutné vymazat nyní již nadbytečnou hlavní kameru. Vestavěné uživatelské rozhraní SDKManageru umožňuje přepínat libovolně mezi ovládáním přes simulátor a ovládáním přes nastavené VR zařízení. [6] Toto nastavení je vidět na obrázku 2.5.

#### 2.2.2.2 Ovladače pohybu

Velká výhoda VRTK je v tom, že nabízí předpřipravené objekty LeftController a RightController, které zaznamenávají pohyby rukou. Je tedy jedno, zda bude výsledná aplikace ovládána tím nebo oním VR zařízením, nebo zda jenom simulátorem, všechny události vyvolané pohybem jsou zpracovávány stejně. K tomuto účelu nabízí VRTK rozhraní VRTK\_ControllerEvents, které obsluhuje všechny události vyvolané uživatelem a dovolí programátorovi vytvořit pro ně specifické handlers.

#### 2.2.2.3 VR ukazatel

Pro interakce s okolím za pomoci takzvaného pointeru, neboli ukazatele v podobě barevného laseru, který umožňuje interakci s objekty na dálku, má VRTK také vestavěné funkce. Stačí k ovladačům pohybu připojit knihovny skriptů

VRTK\_Pointer (pro základní funkcionalitu ukazatele), VRTK\_Straight Pointer Renderer (pro zobrazení ukazatele za pomoci barevných čar) a ideálně taktéž VRTK\_UI Pointer (pro interakci s uživatelským rozhraním).

### 2.2.3 C#, .NET, MONO

Unity 3D umožňuje vývojářům na výběr ze dvou programovacích jazyků: JavaScript a C#. Protože C# je v Unity už více zavedený (a tedy nabízí větší množství zdrojů a návodů spojených s Unity), je i kód této bakalářské práce sepsán v C#. Unity ovšem nepoužívá základní C# framework .NET, nýbrž jeho open source verzi Mono. Níže jsou blíže rozepsané vlastnosti a odlišnosti těchto technologií.

#### 2.2.3.1 C#

C# je objektově orientovaný programovací jazyk vyvíjený firmou Microsoft. Podobně jako Java nepředává své instrukce přímo hardwaru, ale stojí za ním virtuální stroj, tzv. CLR (Common Language Runtime). Jedná se o jazyk velice blízký jak jazyku C++, tak jazyku Java.

#### 2.2.3.2 .NET Framework

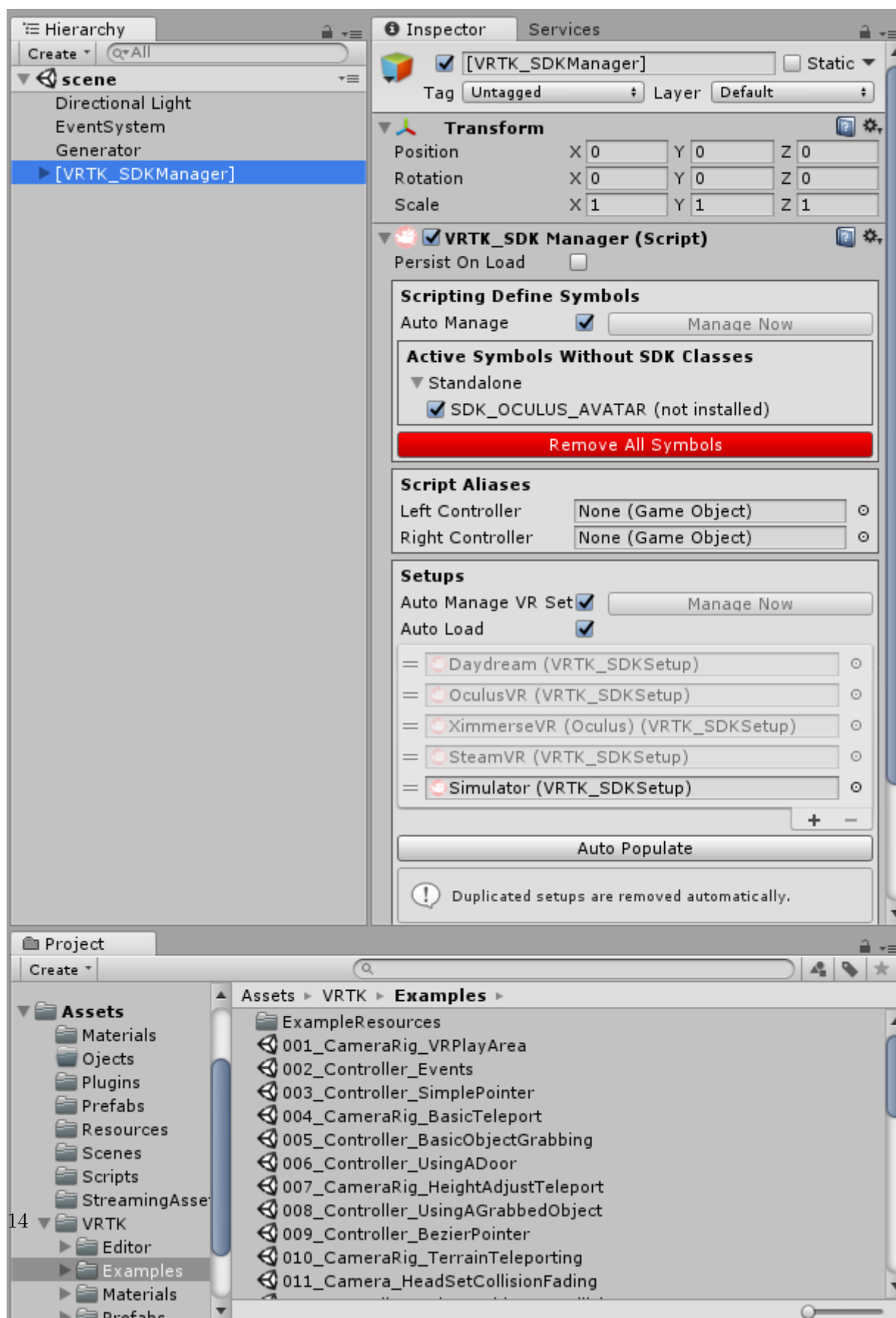
Jedná se o framework, který velice úzce navazuje na výše zmíněný jazyk C#. Jazyk C# je s .NETem tak úzce provázaný, že je takřka nemožné jej používat samostatně bez volání alespoň některých knihovných funkcí z .NETu. .NET bohužel není multiplatformní, je spustitelný pouze pod MS Windows.

#### 2.2.3.3 Mono

Mono je open source verze .NET frameworku. Jedná se o nástroj, který umožňuje vývoj aplikací ve stylu C# a .NET i pro jiné platformy, než je MS Windows. Aplikace vyvíjené v Mono jsou plně přenositelné jak na Linux, Mac, Windows, tak i na mnoho dalších platforem. Mono v sobě spojuje následující komponenty:

- C# kompilátor
- Mono Runtime
- .NET knihovna tříd
- Mono knihovna tříd [9]

## 2. ANALÝZA



Obrázek 2.5: Nastavení VRTK

### 2.2.4 Swagger

Pro návrh API byla zvolena technologie Swagger, která umožňuje vazby zaznamenat ve formátu YAML a JSON.

Technologie Swagger umožňuje rychlý vývoj REST API. Řídí se primárně heslem "neprve návrh, kód až poté". Jedná se tedy o několik technologií (Swagger Editor, Swagger UI a Swagger Codegen) umožňující vývoj prezentační vrstvy aplikací bez nutnosti mít již plně naimplementovanou business vrstvu. Naopak, business vrstva může být vygenerována ze samotného návrhu rozhraní. K tomuto účelu slouží konkrétně Swagger Codegen, který podle nadefinovaného API dokáže vygenerovat kódy pro server, které jsou potřeba. [4]





---

# Návrh

## 3.1 Návrh architektury

Architektura celého projektu je navržena jako třívrstvá (viz obrázek 3.1). Tato bakalářská práce se zabývá návrhem a implementací dvou svrchních vrstev, tedy prezentační a business vrstvy. Funkcionalita jednotlivých vrstev je popsána níže podrobněji.

### 3.1.1 Prezentační vrstva

Prezentační vrstva zajišťuje zobrazení celé scény a veškerou komunikaci přímo s uživatelem. Je implementována v souboru Generator.cs a dalších. Konkrétně se jedná o:

- Vykreslení jednotlivých modelů
- Zobrazení GUI
- Interaktivní prvky

### 3.1.2 Business vrstva

Naproti tomu business vrstva se zabývá logickým vyhodnocením dat získaných z datové vrstvy. Sem mimo jiné spadá:

- Logika rozmisťování modelů
- Vyhodnocování dotazů zadaných uživatelem
- Nasvětlení scény
- Přiřazení textur načteným modelům

#### 3.1.3 Datová vrstva

Nejspodnější datová vrstva (data layer DL) obsahuje datové úložiště, které bude v PostgreSQL databázi obsahovat všechny modely, textury, jejich jednotlivé verze a další. Jedná se o robustní databázi, která bude zpracována nezávisle na ostatních vrstvách. Tato datová vrstva bude jádrem celého projektu, který bude kromě modulu VR (zpracovávaného v této práci) přenositelný na mnohá další zařízení. Proto je databázová vrstva zpracována nezávisle (v rámci jiné bakalářské práce) a tudíž bylo pro komunikaci s ní navrženo API. Jeho podrobná specifikace je uvedena v kapitolách níže.

## 3.2 Funkční a nefunkční požadavky

Tato kapitola se zabývá rozbořem funkčních a nefunkčních požadavků, které bude výsledná aplikace podporovat či nikoliv. Tyto požadavky pomáhají vymezit rámec aplikace a její celkovou finální podobu.

### 3.2.1 Funkční požadavky

Tato sekce obsahuje popis funkčních požadavků, které jsou na systém kladeny.

#### F1 - vizualizace městské scény

Systém přenesení uživatele na konkrétní místo ve městě, jinými slovy vygeneruje modely v nejbližší blízkosti podle konkrétní specifikace požadované uživatelem. Zohlední také světelné a atmosférické podmínky, podle konfigurace scény tedy zobrazí například ve tmě, ráno, v dešti, za sněhu apod. K tomuto bude potřeba načíst modely z databáze, rozmístit je v dané konfiguraci, nastavit textury a materiály podle počasí a přidat skybox.

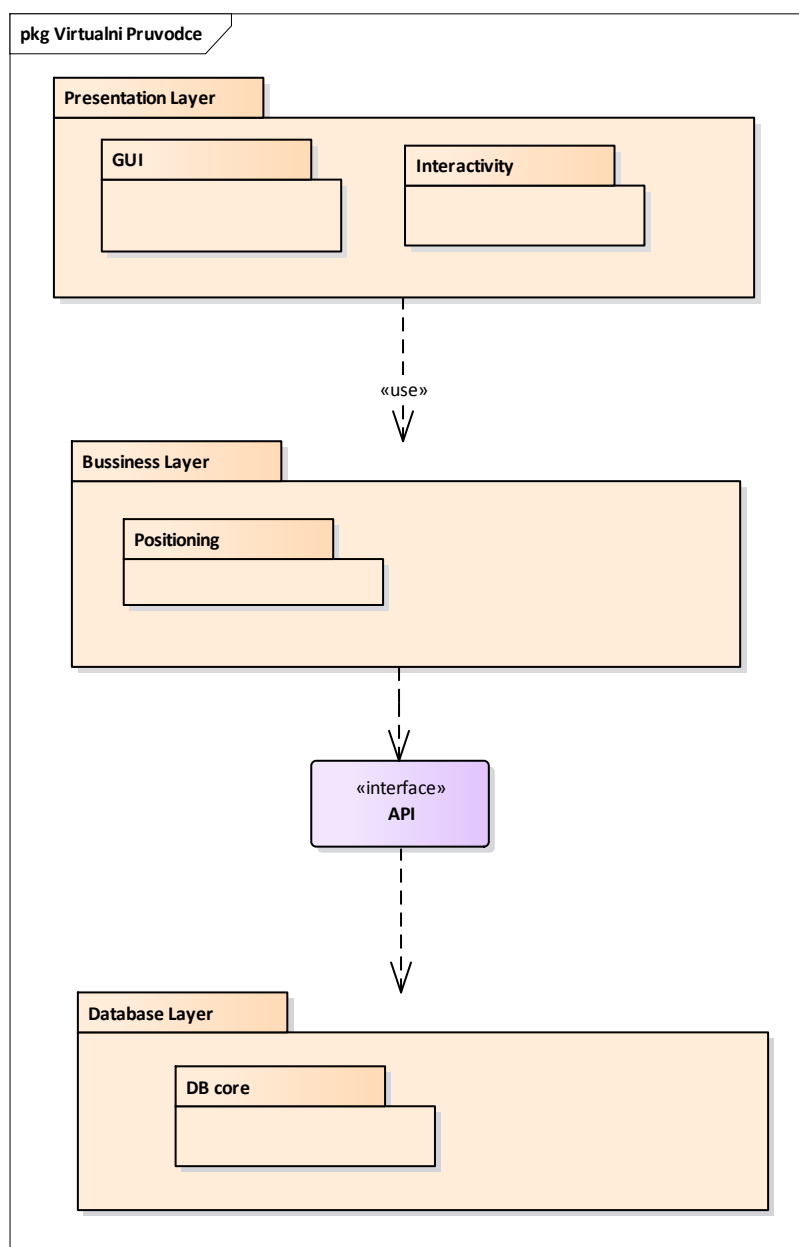
#### F1.1 - interaktivita

Systém umožní uživateli se svým okolím interagovat, a tím je lépe prozkoumávat. Například při ukázání na dům se zobrazí jednoduché informace o tomto domě, dále by bylo například možné zhasnout či rozsvítit lampu a tím změnit nasvícení modelů atd.

#### F3 - modul pro VR

Během vývoje bude aplikace vytvářena pro PC, celkový výsledek bude ovšem funkční pro technologii VR brýlí. Jelikož Unity3D engine podporuje široké množství VR technologií, bude výsledná aplikace přenositelná na téměř jakékoli technologie virtuální reality. Protože základní myšlenka aplikace je vizualizace města pouze v nejbližším okruhu uživatele, skvěle tak koresponduje s omezeným prostorem, které technologie VR v dnešní době podporují.

### 3.2. Funkční a nefunkční požadavky



#### **F4 - administrační interface**

Aplikace bude uživatelům také umožňovat vlastní konfiguraci. Systém bude možné přenastavit dle vlastních preferencí, bude tedy možné nahrát do města nové domy, nově je rozmístit a vložit je do databáze. Tato funkcionality je určena pouze pro specifické uživatele, proto bude tato umístěna stranou ve vedlejším menu.

#### **3.2.2 Nefunkční požadavky**

Tato sekce obsahuje popis nefunkčních požadavků, které nesouvisí přímo s funkcí systému, ale přesto jsou pro správný provoz systému důležité.

##### **N1 - komunikace s datovým úložištěm**

Aplikace bude modely získávat z vnější databáze, ke které bude přistupovat pomocí RestAPI. Tento přístup by měl zajistit efektivní zobrazování scény v závislosti na konkrétní lokaci, kdy se zobrazí pouze ty modely, které jsou v nejbližší blízkosti daných souřadnic. V databázi budou mimo samotných modelů uloženy také jejich transformační matice pro správné umístění v prostoru a zároveň jejich různé materiály a textury pro různá počasí.

##### **N2 - simulátor pro desktop PC**

Aplikace bude mít pro potřeby testování i vestavěný simulátor, který bude napodobovat ovládání přes VR konzoli. Díky tomu nebude nutné k vývoji ani k prvotnímu testování vlastnit VR headset.

### **3.3 Případy užití**

Tato kapitola se zabývá rozбором aplikace z pohledu uživatele. Přehled případů užití níže a obrázek 3.2 zachycují všechny základní scénáře, které při používání aplikace mohou nastat.

#### **UC1 - Procházení města ve VR**

Systém umístí uživatele na požadovanou pozici ve městě.

1. Případ užití začíná ve chvíli, kdy si uživatel nasadí VR headset a objeví se na připravené startovní pozici.
2. Systém uživateli nabídne dialog, který umožní uživateli přesun na jinou lokalitu.
3. Uživatel zvolí ulici (nebo GPS souřadnice?) denní dobu, počasí a roční období, ve kterém se chce ocitnout (výchozí nastavení je léto, odpoledne, slunečno).

4. Systém uživatele "přenes" na danou lokalitu tím, že kolem něj vygeneruje potřebné 3D modely domů ve správné konfiguraci (pozice, natočení, textury, skybox)
5. Uživatel může v omezeném prostoru definovaném VR headsetem procházet své okolí.

**UC2 - Interakce s městem** Basic Path: Uživatel při procházení města interaguje s budovami.

1. Uživatel klikne (v PC verzi) či gestem ukáže (ve VR verzi) na konkrétní budovu.
2. Systém zobrazí okénko s několika informacemi o daném objektu, jmenovitě název, účel, datace, zajímavosti (např. Faustův dům, 1378, popis legendy o Faustovi, dnes součástí Všeobecné fakultní nemocnice).

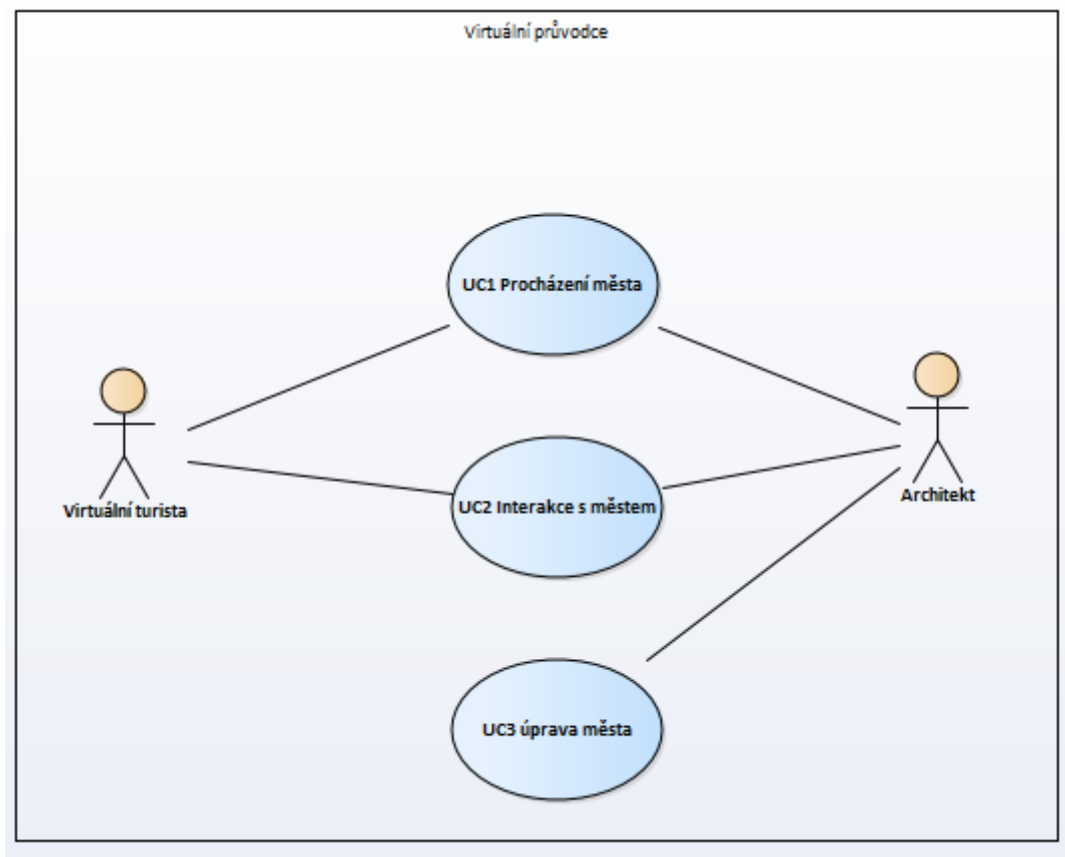
Alternate Path: Uživatel interaguje s jednotlivými objekty.

1. Uživatel zjistí, že je možné interagovat s objektem díky upozornění.
2. Uživatel klikne (v PC verzi) či gestem ukáže (ve VR verzi) na konkrétní objekt.
3. Systém změní nějaké vlastnosti objektu, například zhasne lampu apod.

**UC3 - Úprava města**

Pokročilý uživatel využívá administrátorské funkce.

1. Uživatel poklepá na ikonu vedlejšího menu v pravém horním rohu a zvolí položku "Architect mode".
2. Systém uživateli zobrazí město z ptáčí perspektivy a zobrazí administrátorské menu.
3. Uživatel v administrátorském menu vybere položku "vložit budovu".
4. Systém uživateli zobrazí okno s pohledem do FS, kde si uživatel může najít soubor 3D modelu nově vkládané budovy.
5. Uživatel pomocí myši umístí budovu na požadované místo.
6. Systém otevře potvrzovací okno, zda si je uživatel jistý pozicí nové budovy.
7. Uživatel potvrdí své rozhodnutí.
8. Systém uloží budovu do databáze.



Obrázek 3.2: Diagram případů užití

## 3.4 Návrh použitého API

Jelikož na projektu Virtuální Průvodce bude spolupracovat více lidí, je potřeba navrhnout API pro vzájemnou komunikaci jednotlivých částí aplikace. Konkrétně se jedná o propojení databáze 3D modelů se samotným zobrazovacím modulem.

### 3.4.1 Položky API

API bylo navrženo co nejjednodušeji, ale zároveň tak, aby podporovalo všechny požadavky, které nad databází mohou být vzneseny. Konkrétně se jedná o:

- název modelu
- informace o modelu
- ID modelu

- souřadnice
- počasí
- denní doba
- roční období
- link na 3D model
- link na texturu
- samotný 3D model

Databáze tedy vrátí na konkrétní dotaz nad touto množinou specifickou odpověď, s pomocí které bude model umístěn do scény v potřebné konfiguraci. Podrobné specifikace parametrů jsou popsány níže.

**Název modelu** Jedná se o popis modelu, který pomáhá k jeho identifikaci. Oproti ID slouží spíše ke zpřehlednění, případně jakožto nadpis k dodatečným informacím o modelu.

**Informace o modelu** Stručný popis konkrétní stavby, který bude možno ve finální aplikaci zobrazit při interakci s daným modelem.

**ID modelu** Jednoznačné ID sloužící k jeho identifikaci v databázi.

**Souřadnice** Tento parametr bude rozdělen na 3 podsložky, a to na rotaci, zvětšení (scale) a pozici (transpozici). Toto zjednoduší dotazování nad databází, například pro zobrazování všech modelů v určité vzdálenosti bude potřeba pouze hodnota parametru určující pozici.

**Roční období** Čtyři klasické roční období uložené pomocí typu enum.

**Link na texturu** Odkaz do FS na bitmapu dané textury.

**Link na 3D model** Odkaz do FS na link 3D modelu v daném formátu (např. obj).

**Samotný 3D model** Do budoucna budeme experimentovat i s možností uchovávání samotného 3D souboru v databázi namísto pouze informací o jejím umístění na FS.

**Počasí** Rozlišujeme jen několik základních typů počasí (pomocí enum) a to:

- slunečno (sunny)
- deštivo (rainy)
- zataženo (cloudy)

### 3. NÁVRH

---

- sníh (snow)

**Denní doba** Taktéž charakterizována jednotlivými typy za pomoci datového typu enum:

- ráno
- odpoledne
- večer
- noc



```
weather:
  type: string
  enum: [sunny, rainy, cloudy, snow]
time:
  type: string
  enum: [morning, afternoon, evening,
        night]
season:
  type: string
  enum: [spring, summer, autumn, winter]
texture_link:
  type: string
position_coords:
  type: object
  properties:
    x:
      type: number
    y:
      type: number
    z:
      type: number
rotation_coords:
  type: object
  properties:
    x:
      type: number
    y:
      type: number
    z:
      type: number
scale_coords:
  type: object
  properties:
    x:
      type: number
    y:
      type: number
    z:
      type: number
```

Obrázek 3.3: API v YAML

### 3. NÁVRH

---

Example Value	Model
<pre>[   {     "model_name": "string",     "model_ID": "string",     "info": "string",     "file_link": "string",     "file": "string",     "weather": "sunny",     "time": "morning",     "season": "spring",     "texture_link": "string",     "position_coords": {       "x": 0,       "y": 0,       "z": 0     },     "rotation_coords": {       "x": 0,       "y": 0,       "z": 0     },     "scale_coords": {       "x": 0,       "y": 0,       "z": 0     }   } ]</pre>	

Obrázek 3.4: API v JSON

# Implementace

## 4.1 Napojení na datové úložiště

Tato kapitola se zabývá způsobem propojení aplikace Virtuální průvodce s externím datovým úložištěm. V první části je zmíněna varianta přímého napojení na databázi, která sice není použita ve výsledné aplikaci, ovšem kvůli jednoduššímu přístupu k datům sloužila jako prvotní prototyp celé aplikace. Pro finální produkt je toto řešení ovšem nevhodné, kvůli velkému rozsahu celého projektu bude použito REST API. Popis komunikace přes toto API s pomocí souborů JSON je popsán v dalších částech této sekce.

### 4.1.0.1 Přímé propojení s databází

Toto téma bylo hlavní náplní týmového projektu HK (Hradec Králové) v rámci předmětu SP2. Kromě autorky této bakalářské práce se na dané semestrální práci podíleli také studenti FITu Josef Struž a Iuliia Ostrokomorets.

K tomuto účelu byla použita knihovna NpSQL. V souboru Sqlconnector.cs ve třídě Sqlconnector je uložena proměnná “connecting”, která v sobě obsahuje informace o připojení k databázi (Server, Port atd.). Při inicializaci této třídy se vytvoří propojení s databází. Tato třída v sobě uchovává list Modelů (“Model” je struktura, která obsahuje všechny informace o daném modelu, které se vytáhly z databáze). K přístupu k Modelům se používá funkce GetModel(int i). Nejdůležitější části tohoto kódu jsou níže:

```
public Sqlconnector()
{
    connection = new NpgsqlConnection(connstring);
    connection.Open();
    listItems();
}

void listItems()
{
```

## 4. IMPLEMENTACE

---

```
NpgsqlCommand command = new NpgsqlCommand("SELECT * FROM
models", connection);
NpgsqlDataReader dataReader = command.ExecuteReader();

}
```

### 4.1.0.2 JSON v Unity

Finální aplikace ovšem nekomunikuje s databází přímo, jako to bylo implementováno v projektu pro předmět SP2, nýbrž přes REST API (viz návrh použitého API v kapitole 3.4). Navržené API bude komunikovat s databázovým serverem za pomoci souboru JSON. JSON je zavedený standard, takže jej Unity podporuje a nabízí i knihovny, které umožní jeho deserializaci. Jedná se konkrétně o vestavěnou třídu `JSONUtility`, která nabízí rozhraní pro serializaci i deserializaci JSON souboru. Je tedy možné snadno namapovat obsah JSON souboru do vytvořené třídy s příslušnými třídními proměnnými, na které je možné JSON namapovat. [7]

V tuto chvíli ovšem ještě není zavedena podpora polí objektů v rámci jednoho JSONu. [7] Vzhledem k tomu, že API pro projekt Virtuální Průvodce očekává JSON soubory právě v tomto formátu, neboli pole modelů budov v jediném souboru, je nutné tuto funkcionalitu dodatečně zajistit za pomoci dodatečných obalovacích tříd. Můžeme si tedy nadefinovat ještě jednu další třídu, která bude sloužit jako obal, který v sobě bude uchovávat pole modelů. V implementaci byla tato třída pojmenována `Root`, neboť je jakýmsi kořenem všech podtříd v rámci JSON souboru. Toto pole modelů má ovšem název, který by nebylo možno na nic v souboru JSON namapovat. Toto se dá jednoduše spravit tím, že před přijaté pole modelů od API po přečtení připojíme řetězec obsahující název proměnné, do které se uloží pole modelů s pomocí následující funkce.

```
public string fixJson(string value)
{
    value = "{\\"models\\": [" + value + "] }";
    return value;
}
```

Aby byla serializace možná, je nutné nad deklaraci mapovací třídy `Model` přidat Unity direktivu `[System.Serializable]`. Všechny definice tříd, na které se JSON namapuje, jsou k vidění níže.

```
public class Root
{
    public Model[] models;
}
```

```

public class Coords
{
    public float x;
    public float y;
    public float z;
}

[System.Serializable]
public class Model
{
    public string model_name;
    public int model_ID;
    public string info;
    public string file_link;
    public string file;
    public string weather;
    public string time;
    public string season;
    public string texture_link;
    public Coords position_coords;
    public Coords rotation_coords;
    public Coords scale_coords;
}

```

#### 4.1.0.3 Mono a HTTPS

Výchozí nastavení Mono (na kterém C# v rámci Unity běží) je v rámci bezpečnosti vysoce selektivní, slovy vývojářů "Mono nevěří nikomu!" [9]. Pokud je tedy potřeba pomocí HTTPS protokolu přistupovat k serveru, je pravděpodobné, že tomuto přístupu bude zamezeno. Řešení tohoto problému je funkce, která projde všechny chyby, které se v žádosti o certifikáty naskytly, a manuálně vytvoří nový certifikát X509, který znovu sama zkontroluje. Zavoláním této funkce těsně před přístupem k API s pomocí GET zajistí správný průběh komunikace mezi serverem a aplikací. Níže je ukázka funkce, kterou za tímto účelem nadefinovali uživatelé Unity [10].

```

public bool MyRemoteCertificateValidationCallback(System.Object
    sender,
    X509Certificate certificate, X509Chain chain, SslPolicyErrors
    sslPolicyErrors)
{
    bool isOk = true;
    // If there are errors in the certificate chain,
    // look at each error to determine the cause.
    if (sslPolicyErrors != SslPolicyErrors.None)
    {
        for (int i = 0; i < chain.ChainStatus.Length; i++)

```

```
        {
            if (chain.ChainStatus[i].Status ==
                X509ChainStatusFlags.RevocationStatusUnknown)
            {
                continue;
            }
            chain.ChainPolicy.RevocationFlag =
                X509RevocationFlag.EntireChain;
            chain.ChainPolicy.RevocationMode =
                X509RevocationMode.Online;
            chain.ChainPolicy.UrlRetrievalTimeout = new
                TimeSpan(0, 1, 0);
            chain.ChainPolicy.VerificationFlags =
                X509VerificationFlags.AllFlags;
            bool chainIsValid =
                chain.Build((X509Certificate2)certificate);
            if (!chainIsValid)
            {
                isOk = false;
                break;
            }
        }
    }
    return isOk;
}
```

## 4.2 Vykreslení v Unity

Pro vykreslování se využívá `GameObject Generator` ze souboru `Generator.cs`. Generátor má v sobě uložený vzor budovy “Building”. Při spuštění se vytvoří nový `GameObject` “Buildings”, který je rodičem všech ostatních budov (pokud by byla potřeba smazat nebo měnit velikost aktuální scény, stačí upravit pouze tohoto rodiče) a také se vygenerují všechny budovy. Každá budova obsahuje funkci `Set(Model)`, která načte mesh (pomocí třídy `FastObjImporter`) a pomocí zobrazovací matice nastaví pozici, rotaci a velikost.

### 4.2.1 Textury

Odkaz na textury je nadefinovaný v proměnné, jejíž obsah byl získán z databáze přes API. Textury jsou načítány ze souboru ve FS, který je v této proměnné uložen. Soubor je otevřen, načten do pole bajtů a následně je toto pole bajtů vloženo do speciální třídy `Texture2D`, kterou interně definuje Unity. Kód tohoto načítání je převzat z (cit) ukázka níže.

```
public static Texture2D LoadTexture(string filePath)
{
```

```
Texture2D tex = null;
byte[] fileData;
Debug.Log("filepath = " + filePath);

if (File.Exists(filePath))
{
    fileData = File.ReadAllBytes(filePath);
    tex = new Texture2D(2, 2);
    tex.LoadImage(fileData);
}
return tex;
}
```

#### 4.2.2 StreamingAssets

Tento prototyp načítá modely i textury z FS. Je pravděpodobné, že v budoucnu bude naimplementováno i řešení, které si bude uchovávat binární soubory textur a 3D modelů přímo v databázi, ovšem pro tuto bakalářskou práci bylo prozatím zvoleno toto řešení, neboť je praktičtější na testování a zároveň efektivnější z hlediska fungování databáze.

To ovšem znamená, že i hotová zkompileovaná aplikace musí mít tyto odkazy na textury a modely k dispozici. To by mohlo způsobovat problémy z hlediska odkazů do FS, které by ve finálním nasazení aplikace už nemusely být na stejném místě, na kterém byly při vývoji.

Tento problém řeší Unity za pomoci složky StreamingAssets. Jedná se o složku umístěnou v adresáři Assets, ke které je možno přistupovat dynamicky. Pokud budou tedy všechny soubory, které aplikace za běhu dynamicky načítá uloženy v této složce, budou k dispozici bez problémů i po kompilaci, neboť složka StreamingAssets je vždy nově vytvořena a přenesena i při kompilaci aplikace do nového projektu vedle spustitelného souboru. Ukázka dynamického volání souborů z této složky a jejich následné využití je k vidění níže:

```
objPath = Path.Combine(Application.streamingAssetsPath,
    model.file_link);
texturePath = Path.Combine(Application.streamingAssetsPath,
    model.texture_link);

GetComponent<MeshFilter>().mesh =
    FastObjImporter.Instance.ImportFile(objPath);
Renderer rend = GetComponent<Renderer>();
rend.material.mainTexture = LoadTexture(texturePath);
```

### 4.3 Grafické uživatelské rozhraní

Tato kapitola se bude věnovat všem záležitostem, které jsou spojeny s tvorbou GUI. V první části budou rozebrány specifika GUI pro technologie VR, v druhé pak samotné návrhy GUI pro aplikaci Virtuální průvodce.

#### 4.3.1 Použitelnost GUI pro VR

VR headset je velice odlišné prostředí, než na jaké je většina uživatelů z osobních počítačů či mobilních telefonů zvyklá. Navíc vzhledem k dosavadním cenám VR ještě nejsou tyto technologie tak rozšířené, takže se s nimi uživatelé setkávají spíše nahodile na různých akcích, ve VR hernách apod. Proto je nutné předpokládat, že uživatel, který naši aplikaci spustil, nikdy předtím VR headset na hlavě neměl. Tomuto předpokladu s musí návrh uživatelského prostředí podřídít a být tak jednoduchý a intuitivní, jak je to jen možné. Zcela jednoznačně bude dáván klad na minimalismus a přehlednost, aby uživatel, který poprvé drží v ruce ovladač VR, nemusel složitě zjišťovat, jak správně kliknout na malé tlačítko schované mezi mnoha dalšími.

#### 4.3.2 Prvotní návrh

V této kapitole budou rozebrány prvotní návrhy (neboli wireframy) grafického uživatelského rozhraní. Finální verze se může nakonec lišit podle výsledků uživatelských testů.

##### 4.3.2.1 Celkový koncept

V rámci GUI bývá v dnešní době standardní základní menu obrazovka, která se skládá z několika rovin, které uživatel vidí rozmístěné kolem sebe. Do této obrazovky je možné se vracet stiskem příslušného tlačítka na VR ovladačích. Je to první obrazovka, kterou uživatelé při spuštění uvidí, a zároveň se k ní mohou vrátit kdykoli během samotného běhu aplikace. Otáčením hlavy dokola se pak uživatel rozhlíží po jednotlivých obrazovkách a interaguje s nimi za použití ovladačů. Stejný přístup byl zvolen i pro tuto aplikaci.

##### 4.3.2.2 Výběr lokality

Na této obrazovce si uživatel může zvolit, kde ve městě se chce v danou chvíli nacházet. Na levé polovině mohou uživatelé buď vybírat libovolné body na mapě, nebo si v levé polovině mohou vybrat konkrétní ulici ve městě z nabídky. Svůj výběr pak potvrdí tlačítkem OK, které je umístěné uprostřed pod oběma možnostmi výběru. Viz obr. 4.1.





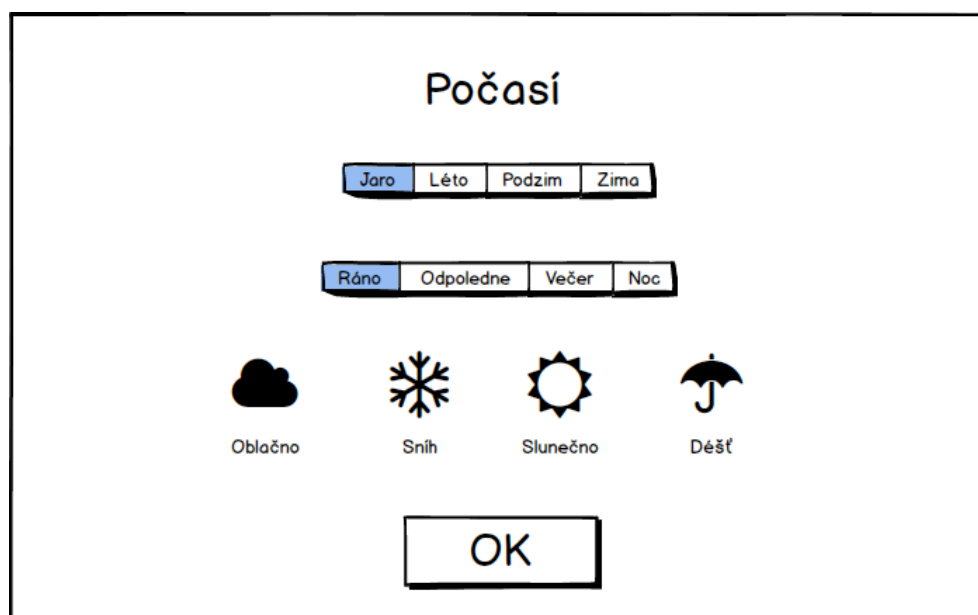
Obrázek 4.1: Wireframe pro vybírání místa

#### 4.3.2.3 Výběr počasí

Druhá obrazovka, která bude umístěna v hlavním menu, bude výběr počasí. Zde si uživatelé mohou vybrat roční období, denní dobu a počasí v dané scéně. Pro jednoduchost bude vždy jedna možnost předem vybraná, aby uživatelé nebyli nuceni proklikávat všechny varianty, i když chtějí změnit například jednu jedinou. Výběr potvrdí rovněž tlačítkem OK umístěným ve spodní části obrazovky. Viz obr. 4.2.

#### 4.3.2.4 Administrátorský režim

Administrátorský režim, neboli režim architekta (Architect Mode), bude umožňovat vkládání nových budov, případně úprava budov stávajících. Předpokládáme, že tento mód bude využíván pouze v PC verzi aplikace, protože je potřeba mít přístup a FS (pro vybrání souboru nově vkládané budovy) a ke klávesnici (pro vložení jejího popisu). Proto se při návrhu tohoto UI nemusí brát v potaz specifika návrhu pro VR headset, a wireframe na obr. 4.3 připomíná více klasické desktopové aplikace.



Obrázek 4.2: Wireframe pro vybírání počasí

### 4.3.3 Uživatelské testování

Testování aplikace Virtuální Historický průvodce proběhne v rámci akce Muzejní noc 2018. Tato akce se bude konat až v červnu letošního roku, proto budou výsledky testování známy později. Uživatelé budou vedeni aplikací za pomoci moderátora, který bude postupovat podle přiloženého scénáře (příloha C). Po vyzkoušení aplikace uživatelé vyplní dotazník (příloha D) ohledně použitelnosti aplikace.

## 4.4 Uživatelská příručka

V této sekci budou popsány základní principy fungování aplikace Virtuální historický průvodce, včetně instalace a ovládání tohoto softwaru. Přestože výsledná aplikace bude přenositelná na různé VR zařízení, základní prototyp byl vyvíjen převážně pro stolní PC. Proto se zde věnujeme převážně instalaci a spuštění na PC.

### 4.4.1 Instalace na PC

Pokud aplikaci získáme ve formě souboru .zip, pak stačí tento soubor rozbalit do libovolného adresáře. Aplikace se sama nainstaluje a spustí poklepáním na soubor pruvodce.exe.

The wireframe shows a window titled "Architect Mode" with the subtitle "Vložení nové budovy". It contains four input fields: "Vyberte soubor .obj" with a "Procházet" button, "Vyberte texturu" with a "Procházet" button, "Zadejte název budovy" with a text box, and "Zadejte popis budovy" with a larger text box containing the placeholder "Zadejte informace o této budově...". A "Vložit" button is at the bottom center.

Obrázek 4.3: Wireframe pro režim architekta

#### 4.4.2 Ovládání simulátoru

Pokud aplikace není spuštěna přes VR headset, ale na stolním PC, je možné ji ovládat za pomoci simulátoru. Následuje seznam základních ovládacích prvků tohoto simulátoru.[6]

##### 4.4.2.1 Mód ovládání kamery

V tomto módu uživatel automaticky začíná. Je možné se v něm rozhlížet po okolí (ovládat pohled kamery simulující pohyby hlavy), ovšem není možné ovládat pohyby rukou. Klávesou levý Alt se pak může přepnout do módu ovládání pohybu rukou a zpět.

##### Otáčení hlavy

Pohybem myši.

### **Dálkový ukazatel**

Tlačítko Q.

### **Interakce**

Tlačítka myši.

### **Chůze**

Tlačítka W, A, S, D.

### **Hlavní menu**

Tlačítko F.

#### **4.4.2.2 Mód pohybů rukou**

Simulátor zobrazuje ovladače v rukou uživatele jako červený a zelený bod, které jsou ve výchozím nastavení umístěny pod hlavní kamerou (z lidské anatomie pod úrovní hlavy), proto je dobré před prvním přepnutím do tohoto módu sklopit pohled kamery dolů. Přepínání mezi tímto módem a ovládáním kamery lze docílit stiskem klávesy levý Alt.

#### **Pohyb rukou**

- V osách x/z pohybem myši
- V ose y stiskem Ctrl a následným pohybem myši

#### **Změna ovládané ruky**

Stisknutím klávesy levý Tab.

---

## Závěr

Tato práce se zabývala vizualizací městské scenérie a je součástí týmového projektu Virtuální Historický Průvodce. Jejím výstupem je aplikace, která umožní uživatelům prozkoumávat středověká města za pomoci technologií virtuální reality. Jedná se o řešení, které umožní ukládat logickou strukturu města v databázi a díky tomu pak zobrazovat pouze ty budovy, které se nacházejí k uživateli nejbližší, čímž se jednoznačně zvýší její efektivita. Tohoto bylo úspěšně dosaženo, jak je možné zjistit z přiložených zdrojových kódů a spustitelných souborů v příloze. V budoucnu bude pravděpodobně tato aplikace sloužit jako prototyp pro robustnější systém, který mimo jiné umožní také podporu smíšené reality, nasazení na Android a další možnosti.



---

## Literatura

- [1] CHAIR OF GEOINFORMATICS, TECHNISCHE UNIVERSITÄT MÜNCHEN, 3D City Database for CityGML Documentation version 3.3 [online], in <https://3dcitydb.net>, 2016, [cit. 18. 4. 2018], Dostupné z [https://github.com/3dcitydb/3dcitydb/blob/master/Documentation/3DCityDB\\_Documentation\\_v3.3.pdf](https://github.com/3dcitydb/3dcitydb/blob/master/Documentation/3DCityDB_Documentation_v3.3.pdf)
- [2] CHROMÝ P., ČÍŽEK J., GHAIK K., HOLUB M., MIKEŠ S., RAJNOCH J., Virtual Old Prague Project [online], in <http://dcgi.fel.cvut.cz/cgg/vsp/>, 2011, [cit. 18. 4. 2018], Dostupné z <http://dcgi.fel.cvut.cz/cgg/vsp/about/doc.html>
- [3] FERNÁNDEZ-PALACIOS B. J., MORABITO D., REMONDINO F., Access to complex reality-based 3D models using virtual reality solutions [online], Journal of Cultural Heritage, Volume 23, January–February 2017, Pages 40-48, [vid. 6. 5. 2018], ISSN: 1296-2074 Dostupné z <https://doi.org/10.1016/j.culher.2016.09.003>
- [4] SMARTBEAR SOFTWARE, Getting Started With SwaggerHub [online], in [app.swaggerhub.com](http://app.swaggerhub.com), 2018, [cit. 18. 4. 2018], Dostupné z <https://app.swaggerhub.com/help/tutorials/getting-started>
- [5] UNITY TECHNOLOGIES, VR overview [online], in <https://docs.unity3d.com>, 2018, [cit. 29. 4. 2018], Dostupné z <https://docs.unity3d.com/Manual/VROverview.html>
- [6] VIRTUAL REALITY TOOLKIT, [Basics] Using the VR Simulator Prefab, in <https://www.youtube.com> [video], květen 2017, [cit. 29. 4. 2018], Dostupné z <https://www.youtube.com/watch?v=bxxGaJg75g4>
- [7] UNITY TECHNOLOGIES, JSON Serialization [online], in <https://docs.unity3d.com>, 2018, [cit. 29. 4. 2018], Dostupné z <https://docs.unity3d.com/Manual/JSONSerialization.html>

- [8] PROGRAMMER, Serialize and Deserialize Json and Json Array in Unity [online], in <https://stackoverflow.com>, 2018, [cit. 29. 4. 2018], Dostupné z <https://stackoverflow.com/questions/36239705/serialize-and-deserialize-json-and-json-array-in-unity>
- [9] MONO PROJECT, FAQ: Security [online], in <http://www.mono-project.com>, 2018, [cit. 3. 5. 2018], Dostupné z <http://www.mono-project.com/docs/faq/security/>
- [10] NIG, How to validate SSL certificates when using HttpWebRequest [online], in <https://answers.unity.com>, 2018, [cit. 3. 5. 2018], Dostupné z <https://answers.unity.com/questions/792342/how-to-validate-ssl-certificates-when-using-httpwe.html>
- [11] THESTONEFOX, The MIT License (MIT) [online], in <https://github.com>, 2018, [cit. 5. 5. 2018], Dostupné z <https://github.com/thestonefox/VRTK/blob/master/LICENSE.md>



---

## Seznam použitých zkratk

**2D** Two dimensional, dvojrozměrný

**3D** Three dimensional, trojrozměrný

**API** Application programming interface, rozhraní pro programování aplikací

**AR** Augmented reality, rozšířená realita

**CLR** Common Language Runtime, runtime pro C#

**CityGML** Aplikační schéma pro reprezentaci GML

**COLLADA** Formát pro interaktivní 3D scény, XML schéma

**FS** File System, souborový systém

**GML** Geography Markup Language, geografický značkovací jazyk

**GUI** Graphical user interface, grafické uživatelské rozhraní

**HMD** Head-mounted display, neboli headset, VR brýle

**JRE** Java runtime environment, rozhraní pro Java aplikace

**LoD** Level of Detail, úroveň detailu grafických modelů

**KML** Keyhole Markup Language, XML vyvinuté pro google earth

**.NET** Knihovna spojená s jazykem C#

**NpSQL** Knihovna pro navázání spojení s databází v .NET

**.OBJ** Frekventovaný formát 3D

**PostgreSQL** Open source RDBMS

**PostGIS** Geografický PostgreSQL

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**REST** Representational state transfer, architektura rozhraní

**RDBMS** Systém pro správu relačních databází

**SDK** Software development kit, sada vývojových nástrojů

**VR** Virtual reality, virtuální realita

**WebGL** Javascriptové API pro zobrazování 3D grafiky

**UI** Uživatelské rozhraní

**VRML** Formát pro popis 3D scén a virtuální reality

**VRTK** Sada skriptů pro vývoj VR aplikací v Unity 3D

**XML** Extensible markup language, rozšiřitelný značkovací jazyk

**XR** Zastřešující pojem pro technologie virtuální, rozšířené a smíšené reality

## Obsah přiloženého USB

	readme.txt .....	stručný popis obsahu USB
	exe .....	adresář 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
	thesis.ps .....	text práce ve formátu PS



## **Scénář uživatelského testování**

Přiložený scénář testování bude použit při prvním zkušebním nasazení aplikace, které by mělo proběhnout na pražské muzejní noci 2018. Tento scénář je provázen dotazníkem, který je taktéž k vidění v příloze D.

## Scénář k testování uživatelského rozhraní aplikace virtuální průvodce

1. Moderátor uživateli nasadí VR headset. Ujistí se, že uživateli dobře sedí a vysvětlí mu ovládání ovladačů.
2. Moderátor se ujistí, že uživatel vidí základní menu a vyzve ho, ať si vše pečlivě prohlédne.
3. Dále moderátor uživatele vyzve, aby zkusil vybrat lokalitu z dané nabídky. Pokud se uživatel zrovna dívá na obrazovku s výběrem počasí, vyzve ho, ať otočí hlavou a najde obrazovku s výběrem lokality.
4. Moderátor sleduje, jak uživatel interaguje s nabídkou. V ideálním případě uživatel sám ovladačem vybere nějakou lokalitu z menu vpravo. Pokud se mu to dlouho nedaří, moderátor mu připomene, které tlačítko na ovladači stisknout. Moderátor také sleduje, zda uživatele samotného napadne potvrdit svou volbu tlačítkem OK, nebo zda očekává další instrukce.
5. Uživatel by nyní měl být přenesen na danou lokalitu ve městě. Moderátor jej chvíli nechá, ať se uživatel sám porozhlédne. Přitom moderátor sleduje, zda se nevyskytují nějaké problémy (dezorientace apod.).
6. Moderátor vysvětlí uživateli, že může s domy interagovat a vyzve jej, aby to zkusil. Namířením ovladače na dům a následný stisk tlačítka by mělo vyvolat modální okno s informacemi. Moderátor nechá uživatele si popis domu přečíst a znovu sleduje, zda vše probíhá v pořádku. Pak uživatele vyzve, aby okno s popisem znovu zavřel.
7. Moderátor vysvětlí uživateli, že v aplikaci je možné také měnit počasí a denní dobu. Poprosí tedy uživatele, aby stiskl na ovladači příslušné tlačítko, které jej vrátí do hlavního menu a zkusí změnit nastavení scény na noc.
8. Moderátor sleduje, zad se uživatel příliš neztrácí v ostatních možnostech, které aplikace nabízí a jak dlouho mu trvá změnit nastavení. Poté, co se uživateli povede změnit nastavení na noční scénu, poděkuje mu za spolupráci a poprosí jej o vyplnění přiloženého dotazníku.



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## **Dotazník pro uživatelské testování**

Tento dotazník bude nasazen při uživatelském testování aplikace. Uživatelé jej vyplní poté, co si aplikaci vyzkouší podle scénáře, který je taktéž přiložen k této práci v příloze C.

# Dotazník o použitelnosti aplikace Virtuální historický průvodce

Prosíme o vyplnění tohoto krátkého dotazníku pro účely testování použitelnosti prototypu aplikace Virtuální historický průvodce. Dotazník je anonymní.

## Demografie

Váš věk	méně než 15	15 - 20	20 - 30	30 - 40	40 - 50	50 a více
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Zájmy a technická zdatnost

	Ano	Spíš ano	Spíš ne	Ne
Už jste někdy používali technologie VR (virtuální realitu)?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vlastníte nějaké VR (SteamVR, Oculus, HTC Vive atd.)?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pracujete pravidelně na PC?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hrajete počítačové hry?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zajímá vás historie?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zajímá vás stará architektura?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cestujete rádi po historických místech?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Hodnocení aplikace

	1	2	3	4	5	
Aplikace se celkově snadno ovládá.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aplikace se celkově špatně ovládá.
Aplikace je celkově přehledná.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aplikace je celkově nepřehledná.
Interakce s budovami byla snadná.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Interakce s budovami byla obtížná.
Změna lokality byla snadná.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Změna lokality byla obtížná.
Texty v aplikaci byly snadno čitelné.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Texty v aplikaci byly obtížně čitelné.
Ovládání menu bylo intuitivní.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Ovládání menu nebylo intuitivní.
Při používání aplikace jsem se cítil(a) dezorientován(a).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Při používání aplikace jsem se necítil(a) nedezorientován(a).
Během používání aplikace mi nebylo dobře (pocit závratě, nevolnost apod.).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Při používání aplikace jsem necítil(a) žádné problémy.



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**