

**Univerzita Palackého v Olomouci**  
**Přírodovědecká fakulta**  
**Katedra geoinformatiky**

**GAMIFIKOVANÝ VIRTUÁLNÍ PRŮVODCE  
HISTORICKÝM OBJEKTEM**

**Bakalářská práce**

**Radim HOLUB**

**Vedoucí práce RNDr. Jan BRUS, Ph.D.**

**Olomouc 2022**  
**Geoinformatika a geografie**

## **ANOTACE**

Bakalářská práce se věnuje vytvoření gamifikovaného průvodce vybraným historickým objektem. Výstupem práce je aplikace pro PC vytvořená v softwaru Unity. Aplikace obsahuje vymodelované detailní 3D modely vybraného historického objektu a zohledňuje uživatelské vstupy a možnosti volby příběhu. Uživatel si může volit mezi hlavním herním příběhem a režimem průlet, který umožňuje neomezený pohyb ve vytvořeném 3D prostředí. Ve výsledné aplikaci je zakomponovaná dynamická změna počasí. Uživatel si může libovolně vybírat v jakých atmosférických podmínkách bude průlet realizován: sluneční den, deštivý den, nebo zimní den. V samotné aplikaci je pak zakomponováno dynamické střídání dne a noci. Gamifikovaný průvodce je situován do Václavského návrší v Olomouci a jeho děj je zasazen do 12. století. Hlavní příběh disponuje umělou inteligencí AI (artificial intelligence) pro dynamický pohyb NPC (non-player character). Je zde vytvořena interakce právě mezi NPC a hráčem ve formě dialogů a cut-scén. Samotný průvodce je založen na hlavním příběhu, kde je uživatel obohacen nejen o historický materiál, ale také plní nejrůznější úkoly. Gamifikovaný průvodce je vytvořen z pohledu třetí osoby s možností změny kamery. Aplikace obsahuje základní fyziku ovlivňující hráče, ale i herní objekty s možností interakce. Výsledná aplikace je optimalizovaná a splňuje tak WYSIWYP (what you see is what you get). Uživatel si libovolně může nastavit grafickou náročnost.

## **KLÍČOVÁ SLOVA**

Herní engine; 3D modelování; gamifikace; GIS

Počet stran práce: 55

Počet příloh: 2 (z toho 2 volné)

## **ANOTATION**

The bachelor thesis is dedicated to the creating a gamified guide to a selected historical object. The output of the thesis is a PC application created in Unity. The application contains detailed 3D models of the selected historical object. It also takes into account user inputs and story choices. The user chooses between the main story and the fly-through mode. The resulting application incorporates dynamic weather changes. The user freely chooses which atmospheric conditions he wants to play in (alternating day and night, rain and snow). The gamified guide takes place on Václavské návrší in Olomouc and is set in the 12th century. The main story uses AI (artificial intelligence) for dynamic NPC (non-player character) movement. There is the interaction between the NPCs and the player program in addition. It takes place in the form of dialogues and cutscenes. The user in the main story is not only enriched with historical material, but also performs various quests and riddles. The gamified guide is created from a third-person perspective with the possibility of changing the camera. The application contains basic physics affecting the player and the game objects. The resulting application is optimized to meet WYSIWYP requirements (what you see is what you get). The user can set any graphical difficulty.

## **KEYWORDS**

Game engine; 3D modeling; gamification; GIS

Number of pages: 55

Number of appendixes: 2

**Prohlašuji, že**

- bakalářskou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

- jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne

Radim Holub

Děkuji vedoucímu práce RNDr. Janu Brusovi, Ph.D a Mgr. Radku Barvířovi, Ph.D za připomínky, náměty a cenné rady při vypracování bakalářské práce. Dále děkuji firmě Tasty Air za poskytnutí graficky upraveného modelu Zdikova paláce a panu Mgr. Miroslavu Kindlovi, Ph.D. za možnost využití napsaného scénáře. Také děkuji všem, co se podíleli na testování aplikace a to zejména týmu GeoLab Technology a Rie-G. V neposlední řadě děkuji svojí mamince, tatínkovi, sourozencům, nejbližší rodině, přítelkyni a přátelům za podporu během studia.

# UNIVERZITA PALACKÉHO V OLOMOUCI

Přírodovědecká fakulta

Akademický rok: 2020/2021

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Radim HOLUB  
Osobní číslo: R19462  
Studijní program: B1301 Geografie  
Studijní obor: Geoinformatika a geografie  
Téma práce: Gamifikovaný virtuální průvodce historickým objektem  
Zadávající katedra: Katedra geoinformatiky

### Zásady pro vypracování

Cílem bakalářské práce je vytvoření gamifikovaného průvodce vybraným historickým objektem. Student vytvoří detailní 3D model daného objektu a vhodně zakomponuje vybrané gamifikační prvky. Práce bude využívat možnosti herního engínu Unity 3D a implementovat různé módy scény. Aplikace bude zohledňovat uživatelské vstupy a podle potřeby zobrazit model v různých atmosférických podmínkách. Dílčím cílem práce je testování přenosu vytvořeného modelu do prostředí virtuální reality. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data) se odevzdá v digitální podobě na paměťovém nosiči (CD, DVD, SD karta, flash disk). Text práce s vybranými přílohami bude odevzdán ve dvou svázaných výtiscích na sekretariát katedry. O diplomové práci student vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Práce bude zpracována podle zásad dle Voženílek (2002) a závazné šablony pro diplomové práce na KGI. Povinnou přílohou práce bude poster formátu A2.

Rozsah pracovní zprávy: max. 50 stran  
Rozsah grafických prací: dle potřeby  
Forma zpracování bakalářské práce: tištěná

### Seznam doporučené literatury:

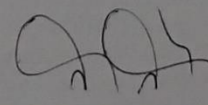
1. BUYUKSALIH, Ismail, et al. 3D MODELLING AND VISUALIZATION BASED ON THE UNITY GAME ENGINE-ADVANTAGES AND CHALLENGES. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2017, 4.
2. GANG, Su-Myung, et al. Unity3D-Based Flood Simulation Visualization Web System for Efficient Disaster Management. *Journal of the Korean Association of Geographic Information Studies*, 2017, 20.1: 98-112.
3. HARRAP, Rob, et al. Our GIS is a Game Engine: Bringing Unity to Spatial Simulation of Rockfalls. *GeoComputation 2019*. 2019.
4. LAKSONO, Dany; ADITYA, Trias. Utilizing a game engine for interactive 3D topographic data visualization. *ISPRS International Journal of Geo-Information*, 2019, 8.8: 361.
5. MA, Yu-Pin. Extending 3D-GIS District Models and BIM-Based Building Models into Computer Gaming Environment for Better Workflow of Cultural Heritage Conservation. *Applied Sciences*, 2021, 11.5: 2101.
6. MAT, Ruzinoor Che, et al. Using game engine for 3D terrain visualisation of GIS data: A review. In: *IOP Conference Series: Earth and Environmental Science*. IOP Publishing, 2014. p. 012037.
7. TEMPLIN, Tomasz; BRZEZINSKI, Grzegorz; RAWA, Marcelina. Visualization of spatio-temporal building changes using 3D web GIS. In: *IOP conference series: earth and environmental science*. IOP Publishing, 2019. p. 012084.
8. VOŽENÍLEK, Vit. Diplomové práce z geoinformatiky. Olomouc, Univerzita Palackého v Olomouci, 2002, 31.
9. ZHAO, Yankun, et al. The research and development of 3D urban geographic information system with Unity3D. In: *2013 21st International Conference on Geoinformatics*. IEEE, 2013. p. 1-4.

Vedoucí bakalářské práce: RNDr. Jan Brus, Ph.D.  
Katedra geoinformatiky

Datum zadání bakalářské práce: 4. května 2021  
Termín odevzdání bakalářské práce: 4. května 2022

UNIVERZITA PALACKÉHO V OLOMOUCI  
PŘÍRODOVĚDECKÁ FAKULTA  
KATEDRA GEOINFORMATIKY  
17. listopadu 50, 771 46 Olomouc  
-1-

LS.



---

doc. RNDr. Martin Kubala, Ph.D.  
děkan

---

prof. RNDr. Vít Voženílek, CSc.  
vedoucí katedry

V Olomouci dne 13. září 2021

# OBSAH

ÚVOD .....	9
<b>1 CÍLE PRÁCE.....</b>	<b>10</b>
<b>2 METODY A POSTUP ZPRACOVÁNÍ.....</b>	<b>11</b>
2.1 Použité metody .....	11
2.2 Použitá data .....	11
2.3 Použité programy .....	11
2.4 Postup zpracování.....	12
<b>3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY .....</b>	<b>13</b>
3.1 Gamifikace .....	14
3.2 Vizualizace a přenos DMR do herních enginů .....	18
3.2.1 ArcGIS maps SDK for Unity.....	18
3.2.2 GIS software .....	19
3.2.3 Specializované 3D software.....	20
<b>4 VLASTNÍ ŘEŠENÍ .....</b>	<b>21</b>
4.1 Tvorba uživatelského rozhraní.....	21
4.1.1 Hlavní menu .....	21
4.1.2 Prohlídka .....	26
4.1.3 Další položky menu .....	28
4.2 Dynamická změna prostředí.....	30
4.3 Modelování objektů.....	34
4.3.1 Modelování interiéru a okolí .....	34
4.3.2 Modelování minimap .....	35
4.4 Hlavní příběh.....	38
4.4.1 Cut-scény a dialogy .....	39
4.4.2 Gamifikace.....	42
4.5 Průlet .....	49
<b>5 TESTOVÁNÍ PŘENOSU MODELU DO VR.....</b>	<b>50</b>
<b>6 VÝSLEDKY .....</b>	<b>51</b>
<b>7 DISKUZE .....</b>	<b>53</b>
<b>8 ZÁVĚR .....</b>	<b>55</b>
<b>POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE</b>	
<b>PŘÍLOHY</b>	



## ÚVOD

S rozvojem gamifikace se čím dál častěji setkáváme s aplikacemi, které mohou sloužit k lepšímu uchopení či vnímání jisté problematiky. V dnešní době, kdy rozsáhlá míra populace je zvyklá na digitální svět, nachází gamifikace široké uplatnění nejen v marketingu či školství. S jistotou se dá říct, že implementace gamifikace se stává silným a nepostradatelným nástrojem pro atraktivnější prezentaci.

Díky velké a zejména rychlé adopci technologií, je jednoduché v současné době pomocí softwarů pro 3D modelování, GIS, animace, herní enginy a jim podobným, ztvárnit libovolný jev z minulosti, nebo naopak z budoucnosti. Příkladem toho můžou být nejrůznější simulace, vizualizace či samotné aplikace integrující tyto možnosti. V posledních letech také není výjimkou ztvárňovat významné minulé epochy v podobě filmů, knih, či v podobě her.

Pro odlišení a získání většího ohlasu začínají galerie a muzea přenášet část svých prohlídek do digitálního světa. Nabízí se tedy možnost propojení a přiblížení vybrané události či jevu současné generaci způsobem, který jim je blízký. Spojením vybrané události nebo jevu a gamifikace s nejnovějšími technologiemi umožňuje vznik gamifikovaného průvodce. Tedy průvodce, který využívá prvky pro zachování historických informačních hodnot, edukativní stránky, ale i gamifikačních či herních prvků, které slouží pro zpestření a uspokojení náročnějších uživatelů.

# **1 CÍLE PRÁCE**

Cílem bakalářské práce je vytvoření gamifikovaného průvodce vybraným historickým objektem. Vytvoří se detailní 3D modely daného objektu a vhodně se zakomponují vybrané gamifikační prvky. Práce bude využívat možnosti herního enginu Unity 3D a implementovat různé módy scény. Aplikace bude zohledňovat uživatelské vstupy a podle potřeby zobrazovat model v různých atmosférických podmínkách. Dílčím cílem práce je testování přenosu vytvořeného modelu do prostředí virtuální reality.

## 2 METODY A POSTUP ZPRACOVÁNÍ

### 2.1 Použité metody

Druhy metod použité v této práci se lišily na základě postupu práce. Pro urbanistické modelování byla zvolena metoda *Hard-Surface*. Zvolená metoda využívá pro tvorbu 3D modelů tzv. tvrdé hrany. Tedy hrany s přesnou a často uspořádanou geometrií, která se pouze minimálně ohýbá. Současně počet celkové geometrie v poměru k velikosti 3D modelu je oproti metodě *Sculpting* zanedbatelný. Pro organické modelování byla použita již zmíněná metoda *Sculpting*. Tato metoda slouží pro objekty, kde dochází k nadměrnému ohýbání geometrie z důvodu zachycení jemnějších detailů. Pro umělou inteligenci, jež využívají NPC, byla použita metoda *NavMesh*, na základě níž se mohou jednotlivé NPC pohybovat. Její použití se liší podle nastavení rychlosti, akceleraci a úhlové rychlosti NPC. Nebo také podle výšky a šířky NPC a mnoha dalších parametrů, které například pracují se sklonem terénu digitálního modelu reliéfu nebo výškou jednotlivých objektů digitálního modelu povrchu. Pro zlepšení výkonu a plynulosti aplikace byla při optimalizaci použita metoda *Occlusion*, jež renderuje pouze objekty, na které se uživatel dívá. Funguje na principu, že objekt typu kamera posílá virtuální paprsky směrem k herním objektům, pokud splňují předem nastavené podmínky, daný herní objekt se vykreslí. Zbylé herní objekty zůstanou nevykresleny a šetří tak výkon uživatele. Pro dynamickou změnu denní doby byla použita metoda, která nepožaduje žádné vstupní textury, na místo toho generuje *Skybox* čistě na základě nastavených grafických vlastností daného materiálu. Naopak pro výběr atmosférických podmínek byla využita metoda, která požaduje vstupní texturu v podobě *HDR maps* (high dynamic range image maps). Pro detekci 3D modelu byla použita metoda *Physics.Raycast*. Dále zde byly použity metody pro selekci herních objektů v seznamu a mezi nejčastější používané metody se řadí metody, které volají neboli aktivují jednotlivé části skriptu.

### 2.2 Použitá data

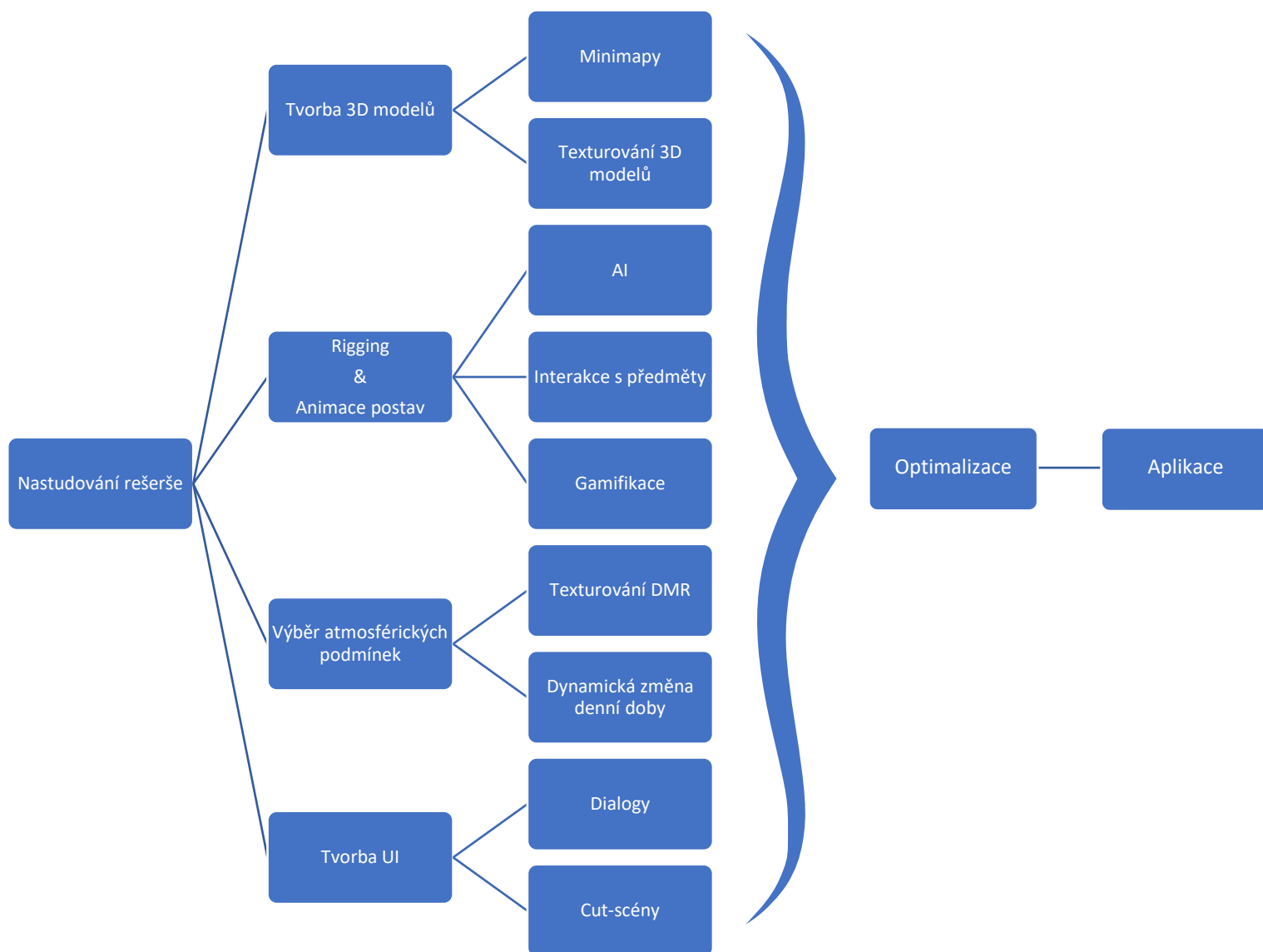
Bakalářská práce využívá data převzatá od historiků z Muzea umění v Olomouci. Data byla využita pro tvorbu 3D modelů a byla převzata v podobě náčrtků, mapek, kreseb a podobně. Digitální model reliéfu byl získán od katedry geoinformatiky v Olomouci, kdy vznikl na základě interpolace vrstevnic dochovaných z jednoho plánu. Práce dále využívá již vymodelovaný 3D model Zdikova paláce, jehož vytvoření obdobně probíhalo na katedře geoinformatiky v Olomouci. Na jeho grafickém zpracování se podílela ostravská firma Tasty Air. Dále byly použity nejrůznější herní assety, zvukové efekty, nebo části skriptu či shaderu a to od: reddification, Adam Pawlak, Johana-PS, VanyaMazalevich, LBobor, Michael, Chris Nolit, Andy woodhead, iceboxX708, Goldman Studio, Taylor, einar, AIV, AzTiZ, deniska40k, Invector, Game Dev Guide, Poneti free, ALP8310, Alexander Nakarada, TomasKiniulis, BinaryImpact, Omni studio, DENISNK, HDRIMAPS, brvasia, Worl of Zero, RBG\_illustrations a Joao Paulo.

### 2.3 Použité programy

V bakalářské práci bylo použito hned několik softwarů. Většina práce probíhala v softwaru Unity 3D, verze 2019.4.11.f. V softwaru unity 3D byl pro tvorbu cut-scén využit balíček *Timeline*, pro tvorbu VFX (visual effects) efektů byl použit balíček *Visual Effect Graph*. Dále to byly *TextMeshPro* a *Postprocessing*. Pro psaní skriptu byl používán Microsoft Visual Studio verze 16.8.4. Pro modelování 3D objektů byly zvoleny dva

software. Pro urbanistické modelování byl zvolen software Sketchup verze 2021.1.2. Pro organické modelování byl zvolen software Zbrush verze 2020.0.1. Vytváření chybějících map pro textury se provádělo v softwaru CrazyBump verze 1.22. Pro tvorbu rigging systému a animací byla použita webová služba Mixamo. Pro 2D práce byly použity grafické editory Inkscape verze 0.9.2.4 a Gimp verze 2.10.12. Pro vyzkoušení modelování a export digitálního modelu reliéfu do herního engine byly použity software ArcGISPro verze 2.8.6, QGIS verze 3.10.4, Blender verze 1.60 a 3DS MAX verze 2021.

## 2.4 Postup zpracování



### 3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Historie průvodce se datuje s prvními cestovateli a spadá do starověku. Za jedny z nejstarších cestovatelů jsou považováni Řekové, Římané a Egypťané. Charakter průvodcovské činnosti se stále více upevňoval a vymezoval ve středověku díky zvyku evropské šlechty cestovat do odlehlých cizokrajných zemí. Během renesance a novověku došlo k rozšíření cestování z nejvyšších vrstev šlechty směrem k měšťanstvu.

V té době průvodce plnil nejen funkci informativní, ale i funkci tlumočnickou. K dalšímu rozvoji došlo v období průmyslové revoluce. S vynálezem knihtisku vyvstala možnost doplnit činnost průvodce i tištěnými texty – tzv. bedekry. Bedekry se neomezovaly pouze na popisy pamětihodností, ale zároveň informovaly o aktuální situaci v destinaci, zajímavostech a podobně (Klímová a Fišerová 2014).

V dnešní době má průvodce mnoho tváří. Již to není pouze vyškolený pracovník, provádějící tuzemské či zahraniční návštěvníky, dávající jim odborný výklad o kulturním a přírodním dědictví oblasti pro kterou je specializován (Ministerstvo práce a sociálních věcí 2017). Můžeme se setkat s analogovými mapami, jako jsou cestovní mapy, topografické mapy pro pěší turistiku či cykloturistiku, které jsou doplněny o komentáře vybraných zajímavých lokalit. Nebo to mohou být cestopisy, přibližující lidem odlehle kouty země, zvyky a tradice tamních obyvatel.

S čím dál větším rozvojem digitalizace se můžeme setkat s aplikacemi, jako je SmartGuide, která má za cíl nahradit knižní či fyzické průvodce. Jedná se o aplikaci, kdy vám do sluchátek bude na základě vaší polohy přehráván odborný text o dané památce. Na displeji současně běží doplňující informace, jako jsou obrázky, videa, či obohacující dokumenty (Kůžel 2020).

Vznik herního průmyslu přispěl k rozvoji digitálních průvodců. Vývojáři pro širokou veřejnost začali zakomponovávat IRL (in real life) prostředí a taktéž se snažili napodobit co největší autenticitu prostředí. Díky tomu dokázali věrně vyobrazovat události a jevy jež nastaly v minulosti. Příkladem toho jsou historické hry, které mohou sloužit i jako dobový průvodce. Jedná se např. o KCD (Kingdom Come Deliverance) nebo sérii Age of Empires, kdy hráči mají možnost poznat původní zbraně, šatstvo, stavby, tradice a celkově historický život. V případě KCD jsou hráči přeneseni do českých zemí v období pozdního středověku – 15. století. Díky velkolepé oblibě herního průmyslu a jeho obrovské adopci, se centrály cestovního ruchu prostřednictvím vzniklých her snaží lákat turisty do vybraných lokalit, odehrávajících se ve hře. Jako tomu bylo v případě KCD, kdy v Středočeském kraji na základě úspěchu KCD vznikla oficiální turistická příručka (Polesný 2018). Průvodce najdeme nejen v hrách s historickou tematikou, ale obdobným příkladem, kde se objevuje průvodce, může být tentokrát nehistorická hra FIFA (Fédération Internationale de Football Association), načež autenticita z IRL byla co nejvíce zachována. Hráči nejenže soupeří mezi sebou, ale ve hře jsou zakomponované prvky sloužící jako průvodce. Ať už se jedná o historii klubů, stadionů či samostatných hráčů.

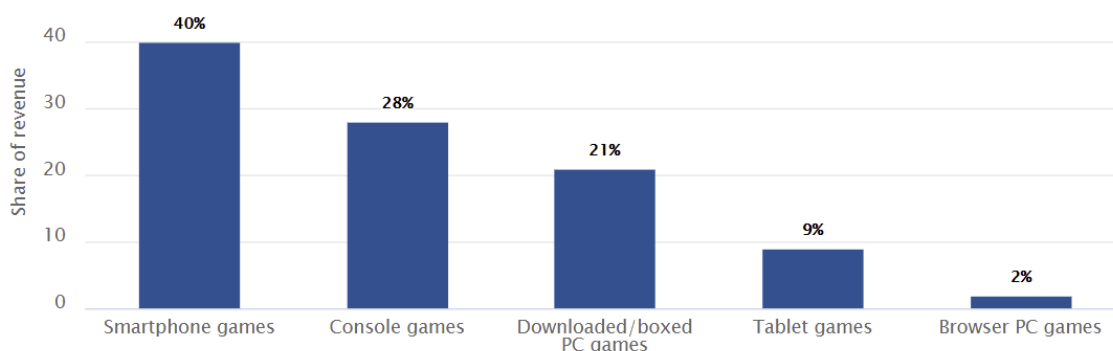
V neposlední řadě VR (virtual reality) a AR (augmented reality) podpořila tzv. gamifikaci ve firmách, kde pomocí nich probíhá školení. Toho se využívá pro velmi rizikové povolání, jako je řízení letadla, či simulování provozu nových strojů (Vinukonda 2018). Gamifikaci označujeme implementací herních principů do ne-herní sféry. Cílem není pouze využití hry, jako metody, ale vytáhnout nejužitečnější psychologické mechanismy a ty použít pro budoucí zaujetí potenciálních zákazníků (Jiránek 2020). Obecně hranice mezi hrou a gamifikací je velmi malá, co se týče vývoje, či obsahu. Je ale důležité si uvědomit, že v případě gamifikace, by měly převládat edukativní prvky, oproti herním a naopak.

### 3.1 Gamifikace

Jak již bylo nastíněno, pojem gamifikace označuje implementace herních principů do neherní sféry (Jiránek 2020). Často se s gamifikací setkáváme ve firmách, marketingu nebo ve vzdělání. Pojem gamifikace poprvé použil herní vývojář Nick Pelling v roce 2002, nicméně již v roce 1896 společnost Sperry & Hutchinson zavedla věrnostní program, ve kterém jejich zákazníci sbírali známky, které následně mohli směňovat za odměny – což je označováno jako počátek gamifikace v marketingu (Fiala 2019).

Gamifikace nachází uplatnění především ve školství nebo v rámci různých vzdělávacích programech. Hlavním problémem ve vzdělávání je, že studenti se odmítají aktivně zapojovat v hodinách a samostudium považují za zbytečné. Informace jsou často nesrozumitelné, zdlouhavé a jedná se spíše o výčet faktů a pravidel, které si student musí naučit nazpaměť. Obdobným problémem je, že učitelé od svých žáků nedostávají zpětnou vazbu, nemají tedy přehled o hloubce porozumění dané látce. U studentů tím pádem dochází k nepozornosti, uvolnění či pocitu nudy nebo ztráty času. Tradiční výuka je studenty vnímaná, jako neefektivní a dochází k tomu, že školy nemají nástroje a schopnosti, jak žáky správně motivovat (Bajer 2021).

Podle dat Entertainment Software Association z roku 2020 vyplývá, že ve Spojených státech až 70 % mladistvých hraje počítačové hry. Zatímco ve věku 18 a více let pravidelně hraje až 64 % dospělých. Studie z roku 2020 zjistila, že s tím, jak se videohry stávají typickou součástí života, stoupá i průměrný věk hráče (Essential 2020). V současnosti má gamifikace obrovský potenciál ve školství, kdy se stala rozšířenou výukovou metodou. Příkladem toho je škola Quest to learn, která pro výuku využívá různé typy her. Veškeré procesy jsou gamifikované, hlavními principy je vést studenty k schopnosti samostatně uvažovat a tvořit, implementovat a pracovat s komplexitou systémů (EDUin 2015). Dalším příkladem, že dochází k celosvětové adopci a masovému rozvoji herního průmyslu, tedy i gamifikace ukazují data od Newzoo. Na základě dat lze vyčíst, že na konci roku 2020 bylo po celém světě přes 2,7 miliardy hráčů. Meziroční růst, co se týče počtu hráčů, se zvětšuje průměrně o 5,6 %. Dle dat od Newzoo vychází, že dva z pěti lidí jsou hráči. V čele světového trhu co do počtu hráčů je Čína, těsně následovaná Spojenými státy. Mobilní hry tvoří téměř 49 % celkového trhu, přičemž příjmy z her na chytrých telefonech a tabletech v roce 2020 dosáhly 86,3 miliardy dolarů. Všechny segmenty dosáhly pozitivního růstu s výjimkou webových her, kde došlo k meziročnímu poklesu o -8,7 % (Gilbert 2021).



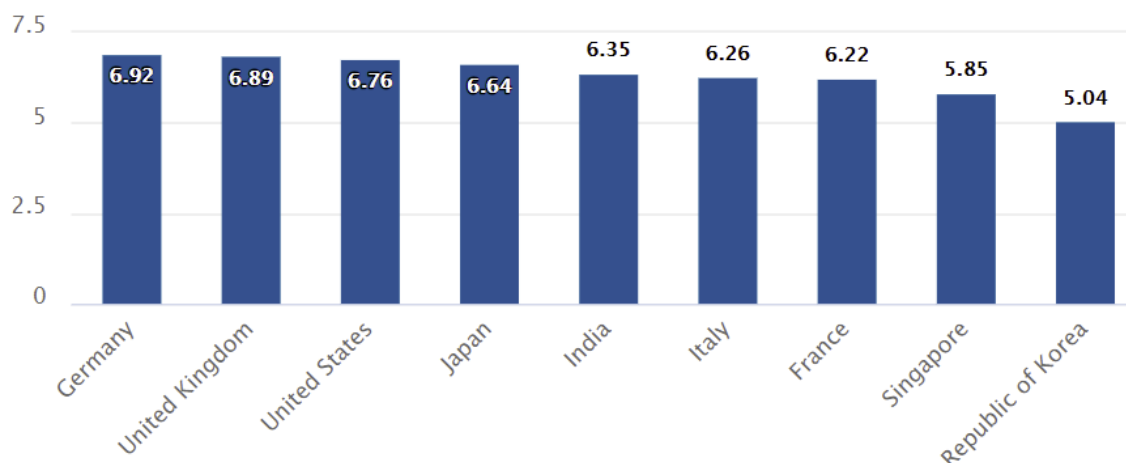
Source: Newzoo (2020)

Designed by  FinancesOnline

Obr. 3.1.1 Podíl tržeb na celosvětovém trhu s hrami v roce 2020  
(zdroj: <https://financesonline.com/number-of-gamers-worldwide/>).

Jednou ze studií, zaměřující se na průměrnou týdenní dobu strávenou u videoher, se zabývala společnost Limelight. Podle dat z roku 2020 vévodí na první příčce Německo, co by s nejvyšší průměrnou týdenní dobou strávenou hraní videoher. Německo má již dva roky po sobě nejvyšší průměrnou týdenní herní dobu, a to téměř 7 hodin hraní týdně, což je ale o 13% pokles oproti roku 2019 (Gilbert 2021). Americký psycholog (Malone 1980) vypracoval v roce 1980 studii, zabývající se herními prvky ve školství v souvislosti s motivací žáků, ve které tvrdí: „Jestliže žáci získají motivaci k učení, stráví nad určitými problémy mnohem více času, danému problému věnují více svého času a své snahy o jejich vyřešení, na konci budou mít lepší pocit ze svých výsledků a pravděpodobně využijí nabyté znalosti i v budoucnu.“

Malone definuje z pohledu motivace pět základních charakteristik optimálního vzdělávacího prostředí: „Žák se může prostředím volně pohybovat a taktéž svobodně zkoumat. Žák dostává okamžitou zpětnou vazbu o svém výkonu a je informován o svých důsledcích a postupech, na základě nich sám vyhodnocuje, jak dobře se mu daří. Žák postupuje vlastním tempem, kdykoliv se může vrátit na začátek, obtížnost zvyšuje, snižuje na základě svých schopností. Žák není při objevování limitován prostředím. Žák je nabádán k objevování souvislostí, daná aktivita by měla mít širokou škálu dílčích úkolů.“ Takové prostředí lze najít právě u počítačových her a pro školní účely jej lze dosáhnout pomocí gamifikace. Právě tyto fakta si uvědomila již zmiňující škola Quest to learn, která zařadila gamifikaci, jako nepostradatelný prvek do své výuky.



Source: Limelight (2020)

Designed by  FinancesOnline

Obr. 3.1.2 Průměrná týdenní doba strávená hraním her  
(zdroj: <https://financesonline.com/number-of-gamers-worldwide/>).

Klíčovým cílem gamifikace je zvyšovat motivaci a výkonnost daných uživatelů. Toho se dosahuje díky aplikaci herního systému, jako je soupeření o nejlepší umístění v žebříčku, sbírání omezených věcí, plnění úkolů a podobně (Ohio University 2022).

V České republice se můžeme setkat s gamifikovaným průvodcem - hrou Kód Salomon, kdy technologický postup vyřešili vývojáři firmy Intelis a obsah dodělala firma Gamelon. Aplikace uživatele zavede do historie města Ostravy. Aplikace obsahuje šifry a nejrůznější trasy, kdy je na samostatném uživateli, kterou trasu si zvolí. Po splnění dílčích úkolů, je hráč odměněn (E-Newspeak 2014).

Ve světě se můžeme setkat s podobnými aplikacemi ať už je to německá aplikace Actionbound, která umožňuje učitelům vytvořit interaktivní scénář. Na základě scénáře

studenti s chytrým telefonem či tabletem prochází předem definovaná místa, plní úkoly a vzdělávají se. V Belgii vznikl vzdělávací program Centre informatieve spelen, kdy za úkolem tamních vývojářů je tvorba edukativních her (EDUin 2015). Dále pro učitele existují aplikace jako je Socrative, jedná se o online nástroj, který pomáhá vytvářet učitelům herní prostředí, jež umožňuje vzdělávat žáky pomocí zábavní formy.

Socrative byla vytvořena z důvodu vylepšení digitální komunikace mezi žáky a učiteli. Učitelé mohou používat Socrative pro kvízy pro celou třídu nebo rozdělit třídu do skupin. Jsou zde také individuální kvízy, které učitelům umožňují pracovat podle potřeby pro daný předmět. Učitelé mohou vytvářet kvízy s výběrem odpovědi z několika možností, s pravdivými nebo nepravdivými odpověďmi nebo s odpověďmi jednou větou, přičemž všechny tyto odpovědi mohou být hodnoceny se zpětnou vazbou pro každého studenta (Edwards 2021).

Obdobnou aplikací je Toglic či aplikace Kahoot. Aplikace jako Duolingo, Memrise a Babbel slouží k zdokonalování cizích jazyků a to vše zábavnou formou prostřednictvím telefonu. Ačkoli jsou si v konceptu podobné, v konkrétních věcech se hodně liší. Duolingo nabízí poslechová cvičení, flashkarty a otázky na procvičování nových slovíček, frází a vět. Většina otázek má vlákno s komentáři, kde mohou uživatelé o konkrétní otázce podrobně diskutovat. Memrise nabízí podobné funkce jako Duolingo, seznamuje s novými slovíčky a frázemi pomocí kartiček a poslechových cvičení. Aplikace také nabízí funkci nazvanou Learn With Locals, která páruje slovíčka s videi rodilých mluvčích, kteří frázi říkají nahlas a předvádějí ji. Babbel se od ostatních dvou liší. Používá sice podobná cvičení s výběrem odpovědi nebo poslechová cvičení jako ostatní, ale na příkladech z konverzace ukazuje, jak nová slova nebo fráze používat při rozhovoru s jinou osobou. (nytimems 2019).

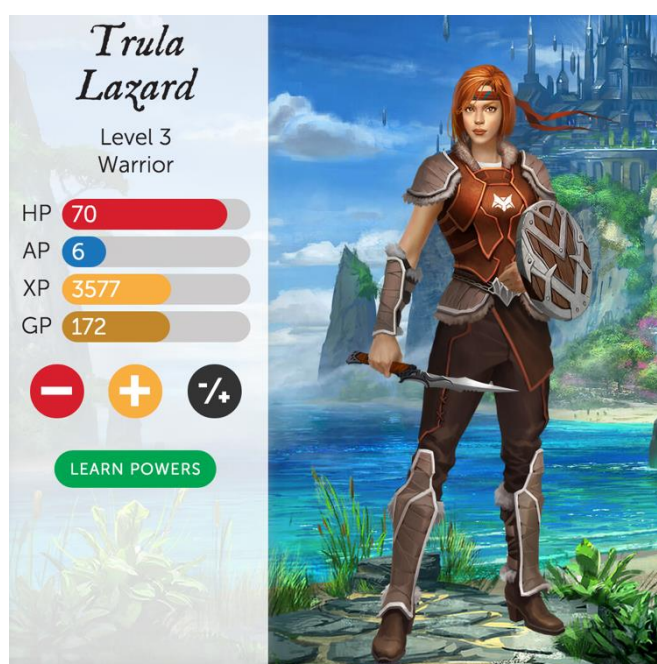
Aplikace jako Classcraft nebo Class-Dojo dovolují vyučujícímu zčásti přenést učební proces do formy hry, ve které jsou žáci motivováni k plnění úkolů. Za aktivitu, vzájemnou spolupráci a správné řešení získávají body, které mohou uplatnit pro svého avatara, kterého si vybrali na začátku. Svě úspěchy mohou žáci sdílet se svými rodiči a vzájemně se mezi sebou porovnávat. Class-Dojo je kompatibilní jak s desktopovou verzí tak i se systémem iOS či Android. Poté, co si žáci vyberou svého avatara, který je charakterizuje, dostávají od svých učitelů body. Tyhle body mohou obdržet za správné řešení domácích úkolů, za pomoc jiným spolužákům, za aktivitu v hodině, za pořádek na svém místě apod. Za věci jako vyrušování, nedodržování pokynů, mluvení během hodiny se bod ztrácí. Žáky to motivuje k větší aktivitě a správnému chování. Jejich profily jsou transparentní a umožňují tak lepší komunikaci mezi rodiči a učiteli (Fiala 2019).

Americký edukátor Karlin se na svém webu podrobně zabývá aplikací Classcraft. Jedná se tedy podle něj o online platformu, která je vytvořena tak, aby pomáhala učitelům zvýšit zapojení žáků v hodinách. Cílem Classcraftu je přeměnit běžné vyučovací hodiny na příběhové úkoly, v nichž má každý žák důležitou roli. Classcraft funguje na principu spolupráce. Studenti při procházení lekcí a úkolů si musí navzájem pomáhat. Za podobnou aktivitu, jako to tomu bylo v Class-Dojo, studenti získávají zkušenostní body, které jim pak dále mohou sloužit, jako výhody. Kromě zkušenostních bodů mají studenti také body zdraví (HP). Ty studentům umožňují zůstat ve hře "aktivní". Pokud v důsledku neplnění úkolů, neaktivity či vyrušování, ztratí všechny HP, jejich postavy čeká za to určité potrestání, které zvolí učitel. Studenti mají také akční body (AP) a právě ty určují, jak často mohou používat získané schopnosti. Každý nový den se AP trochu obnovují, takže si je studenti mohou znovu nasbírat a používat tak své schopnosti. Nakonec jsou tu zlaťáky. Studenti získávají zlaťáky automaticky při zvyšování levelů u svých avatarů. Jestliže učitel má přístup k prémiové verzi, může také rozdávat zlaťáky studentům za vhodné chování a aktivitu v hodinách a podobně. Za zlaťáky si mohou žáci



koupit například nové vybavení pro svého avatara. Studenti si vybírají jeden ze tří typů postav - Mág, Strážce nebo Léčitel. Jak již bylo zmíněno výše, Classcraft je koncipován jako týmový a v každém týmu musíte mít jednoho z každého typu postavy. To pomáhá zajistit, aby každý student měl v týmu důležitou roli.

Aby bylo pro učitele používání úkolů ještě snazší, Classcraft nedávno spustil režim Story Mode. Jedná se o epizodické dobrodružství, které pomáhá učitelům zapojit a motivovat žáky prostřednictvím vyprávění příběhů. Režim Story Mode nabízí učitelům připravené příběhy, které jim pomohou zlepšit zapojení žáků a podpořit pozitivní atmosféru ve třídách. Zatímco žáci pracují na úkolech, jsou k dispozici také "třídní nástroje", které může učitel použít. Jedná se o věci jako náhodný výběr studentů, časovač nebo bitvu s bossem, která funguje jako formativní hodnocení. Classcraft poskytuje širokou škálu statistik a analýz. Tyto analýzy zahrnují věci, které nabízí většina platform - docházku, údaje o přihlášení, přehledy o používání atd. Zahrnují však také analýzy, jako je aktivita žáků v jednotlivých předmětech (Karin 2019).



Obr. 3.1.3 Avatar v gamifikační aplikaci Classcraft (zdroj: <http://www.edtechroundup.org/>).

Pokud se instituce rozhodne oživit svůj program a nechá si vytvořit kvalitní gamifikovanou aplikaci na míru, musí počítat s velkými náklady, ale i s tím, že ne vždy se podaří efektivně implementovat aplikaci k daným uživatelům. Stejně jako za vývojem RPG (role-playing game) či RTS (real-time strategy) her, tak i za vývojem gamifikované hry, stojí programátoři, ať už se jedná o UI (user interface) nebo AI, dále jsou zde 3D enviromentální designéři, 3D modeláři postav či jiných charakterů, animátoři, herní architekti, scénáristi a mnoho dalších profesí. Často se stává, že se herní studia specializují na odlišné technologie, je tedy na místě mít dopředu rozmyšlené zda se bude jednat o aplikace využívající: AR, VR, Mobile 3D, Mobile 2D, 2D nebo 3D herní prostředí a podle toho se rozhodovat.

## 3.2 Vizualizace a přenos DMR do herních enginů

V posledních letech došlo k rozvoji GIS a to zejména v oblasti vizualizace 3D dat. Díky tomu lze aplikovat 3D data nejen do herních enginů, ale naskytuje se možnost volby vizualizovat 3D data v různě zaměřených softwarech. Široká nabídka softwarů umožňující vizualizaci 3D data pomáhá nejen samostatným vývojářům her dosáhnout reálnějšího světa, ale současně to podporuje cestovní ruch, nebo např. predikci katastrof. Z realizovaného výzkumu je patrné, že 3D terén uživatel mnohem lépe čte a vnímá, než běžnou 2D mapu, která vyžaduje po uživateli určitou míru porozumění (Ruzinoor a kol 2014). Ruzinoor a kolektiv provedli průzkum softwarů pro 3D vizualizaci terénu s využitím geoinformatických dat. Zároveň zkoumali různé druhy technik pro jednotlivé vizualizace 3D terénu. Jazyk VRML (Virtual Reality Modeling Language) využili Ruzinoor a kolektiv pro online vizualizaci terénu. Jazyk VRML je standardizován obdobně jako HTML (HyperText Markup Language), avšak VRML umožňuje nejen vytvořit model, ale současně disponuje interakčními prvky, které pomáhají uživateli s komunikací. Dalším příkladem je 3DEM, kdy autor (Horne 2009) vytvořil nejdříve software pro komerční trh, avšak po několika letech vývoje projekt zastavil a nabídl plnou verzi softwaru pro veřejné použití. Software má schopnost importovat a generovat 3D terén z dat digitálního modelu reliéfu (DMR). Tento software také poskytuje zobrazení povrchu Marsu. Zuse Institut Berlin & Lenné3D vytvořili program BioSphere3D. Tento software má schopnost zobrazit terén planety Země pomocí dat, jako jsou DMR, satelitní a letecké snímky (Ruzinoor a kol 2014). Dalším příkladem je open source CESIUM, jedná se o 3D vizualizér terénu, který využívá koncepci zobrazení zeměkoule, jež má schopnost zvětšovat a zkoumat terén celého světa. Používá knihovny WebGL (Web Graphics Library) pro načítání shapefile při vytváření terénu (Cesium 2012). Gunia vyvinul aplikaci Earth3D v jazycích Java a C++. Data pro tento software jsou generována z NASA (National Aeronautics and Space Administration), USGS (United States Geological Survey), CIA (Central Intelligence Agency) a dalších zdrojů (Gunia 2014). GenesisIV je software vytvořený společností Geomatics. Výhody tohoto softwaru je schopnost generovat fotorealistické terény (Ruzinoor a kol 2014). G3DGMV vyvinul Yostn pro vizualizaci terénu, který jako hlavní zdroj dat používá DMR a DLG (Digital Line Graph), umožňuje vytváření průvodce na základě uživatelských preferencí (Yostn 2001). Hftools představil Stock, což je prohlížeč terénu založený na Unixu, jehož výstupem je terénní mapový podklad, který se z tohoto softwaru dá převést do různých typů formátů souborů (Hftools 1998). Posledním příkladem je integrovaný prohlížeč dat. Integrovaný prohlížeč byl vytvořen společností Unidata a je založen na programování v jazyce Java. Tento software je schopen například vytvořit animaci satelitních snímků (Unidata 2007).

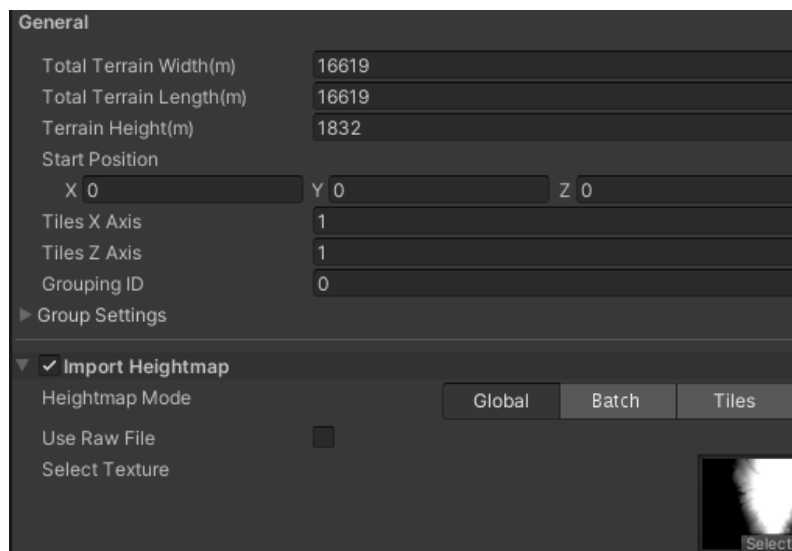
### 3.2.1 ArcGIS maps SDK for Unity

Jednou z možností, kde vytvářet a vizualizovat 3D data je herní engine Unity. Pro vizualizaci 3D dat v herním enginu Unity, může být využit SDK vyvinutý Esri, ArcGIS Maps SDK for Unity. ArcGIS Maps SDK for Unity umožňuje přístup k online i offline datům z platformy ArcGIS. I když se stále jedná o beta verzi, a chybí zde základní vlastnosti, jako je práce s objekty v neherním módu a tedy není možnost, jak aplikovat nebo uložit polohu pro jakýkoliv typ objektu bez geografických dat, nebo zde nefunguje provádění operací nad 3D daty, tak přesto ArcGIS Maps SDK for Unity umožňuje uživatelům volby lokání nebo globální scény. Dále disponuje možností výběrem lokace, *base map*, nebo přidáním vlastních dat. V budoucnosti vývojáři slibují velký potenciál v rychlosti renderingu detailních dat (ArcGIS 2021).

### 3.2.2 GIS software

Při použití podrobných dat, které nejsou volně dostupné, anebo jimi nedisponují vizuální prohlížeče, či s nimi je potřeba jakkoliv manipulovat, je na místě využít GIS softwaru. Existuje více způsobů, jak vizualizovat a přenést DMR v herních enginech s použitím GIS softwarů. Nicméně pro jejich budoucí použití je důležité rozdělovat DMR pouze pro vizualizaci, nebo pro případnou budoucí práci, která umožňuje zásah do samotného terénu. Pro budoucí práci s DMR je nutné změnit daný objekt na typ *Terrain*. Rozdíl mezi 3D modelem DMR a *Terrain* DMR je v tom, že *Terrain* je v Unity objekt, který lze dále modelovat, lze na něj nanášet assety stromů či jiné vegetace a obsahuje větrné zóny (Unity 2005).

Software ArcGIS PRO v současnosti nedokáže exportovat 3D formáty. Existují ovšem možnosti, jak přenést DMR do herního engine Unity i bez vygenerování samostatného 3D modelu za použití GIS softwarů. První příklad vychází z toho, že rastrová data budou převedena na PNG / JPG formát. V exportu rastru je nutné nastavit stejný počet řádků a sloupců. V herním engine Unity je využít Package Manager, kde je stáhnut *Terrain Toolbox*. Do vybrané složky bude naimportována mapa výšek (heightmap), jako textura. Nakonec na základě počtu řádků, sloupců a velikosti pixelů je potřebné doplnit informace o délce a šířce terénu v metrech. Výška terénu zde funguje, jako převýšení ( $z$  – factor).



Obr. 3.2.2.1 Nástroj Terrain v herním engine Unity (zdroj: autor).

Další možností je, že heightmapa v ArcGIS PRO bude vyexportována ve formátu TIF či jiném nevektorovém formátu. Následně tato data budou přenesena do softwaru QGIS, kde je nezbytné si stáhnout plugin DEMto3D. V nástroji DEMto3D budou nastaveny požadované parametry, jako měřítko, zkreslení a podobně. Export dat proběhne v jediném dostupném 3D formátu – STL. Problémem je, že Unity nepodporuje formát STL, je teda potřeba vyexportovaný 3D model v jakémkoliv 3D softwaru změnit na formát OBJ nebo FBX. V herním engine Unity je nejen importován vytvořený 3D model, ale je nutné vytvořit skript, který mění vlastnosti herního objektu na objekt typu *Terrain* (Shakeel 2017).

```

void CreateTerrain() //Ukázka kódu pro vytvoření z herního objektu, objekt typu terrain
{
    TerrainData terrain = new TerrainData();
    terrain.heightmapResolution = resolution;
    GameObject terrainObject = Terrain.CreateTerrainGameObject(terrain);
    Undo.RegisterCreatedObjectUndo(terrainObject, "Object to Terrain");
    MeshCollider collider = Selection.activeGameObject.GetComponent<MeshCollider>();
    Cleanup cleanup = null;
    if (!collider)
    {
        collider = Selection.activeGameObject.AddComponent<MeshCollider>();
        cleanup = () => DestroyImmediate(collider);
    }
}

```

### 3.2.3 Specializované 3D software

Většina 3D softwarů dokáže stejně jako GIS softwary pracovat s mapou výšek a navíc 3D softwary narozdíl od většiny GIS softwarů vytvoří z požadované oblasti *mesh*. Veškeré postupy vychází z předešlé podkapitoly 3.2.2. Ve vybraných 3D softwarech, ať už s pomocí pluginů nebo s pomocí mapou výšek, je získán 3D model DMR a ten je následně vyexportován ve formátu OBJ nebo FBX. Na konci jsou podle potřeby dané modely v herním enginu Unity převedeny na objekt typu *Terrain* pomocí skriptu (Shakeel 2017).

Sketchup – je software pro tvorbu 3D modelů, vyvíjený od roku 2000 společností Trimble. Slouží nejen pro profesionální architekty, filmové tvůrce ale taky pro designéry počítačových her. Součástí softwaru je knihovna 3D Warehouse (Trimble 2000). Prvním příkladem je použití funkce *Add location*. Funkce *Add location* otevírá nové pop-up okno, kde si uživatel z celého světa vybere požadovanou oblast a přes nástroj *Toggle Terrain* přidá vybrané oblasti informace o výšce. Nevýhodou je, že tahle funkce bývá zpoplatňována od velikosti vybraného území nebo podle jejich podrobnosti. Textura terénu je automaticky přiřepčena k modelu, po přidání převýšení. Druhým příkladem je plugin *Mesh From Heightmap*. Plugin vytvořil Thomassen a funguje pouze na jednoduchém principu zadávání největší možné reálné nadmořské výšky v dané oblasti (Thomassen 2010). Jelikož Sketchup pracuje s trojúhelníkovou sítí, existuje další způsob, jak vizualizovat DMR pomocí mračna bodů. Vývojář Fredo6, naprogramoval plugin *TopoShaper*, který díky triangulaci vytvoří z mračna bodů DMR (Fredo6 nedatováno).

3DS MAX – je software pro tvorbu 3D modelů, vyvíjený od roku 1990 společností Autodesk. Často bývá používán v postprodukci při výrobě reklam nebo ve filmařském průmyslu, snadno se zde vytvářejí animace a vizualizace (Autodesk 1990). Jedním z příkladů v softwaru 3DS MAX jak vytvořit 3D model DMR, je díky pluginu *Terrain* od vývojáře (West 2010). Uživatelé stačí zadat souřadnice požadované oblasti a navíc díky importu KML přímo importuje na svůj model ortofotografické textury z Google Earth pro.

Blender – je free and open source software. Jedná se o všestranný 3D software. Jedním z nejpoužívanějších bezplatných nástrojů pro tvorbu DMR je plugin *BlenderGIS*, který funguje obdobně, jako předešlé zmiňované pluginy. Uživatel si zvolí oblast, vybere si podkladovou mapu a na základě toho se mu vytvoří 3D model DMR. Výhodou je, že obsahuje importy pro Shapefile (SHP), rastry (TIF, JPG, JP2, PNF), Open street mapx (XML, OSM) a Esri asc grid (ASC) (Blender 1995).

## 4 VLASTNÍ ŘEŠENÍ

Cílem vlastního řešení bylo vytvořit desktopovou aplikaci s motivem gamifikovaného průvodce. Gamifikovaný průvodce se odehrává ve 12. století na Václavském návrší v Olomouci. Z toho důvodu byly vytvořeny detailní 3D modely spadající do daného období. Navíc zde byly zakomponované interaktivní prvky, volba atmosférických podmínek, nebo dynamická změna prostředí. K samotné tvorbě byl využit scénář napsaný panem Miroslavem Kindlem.

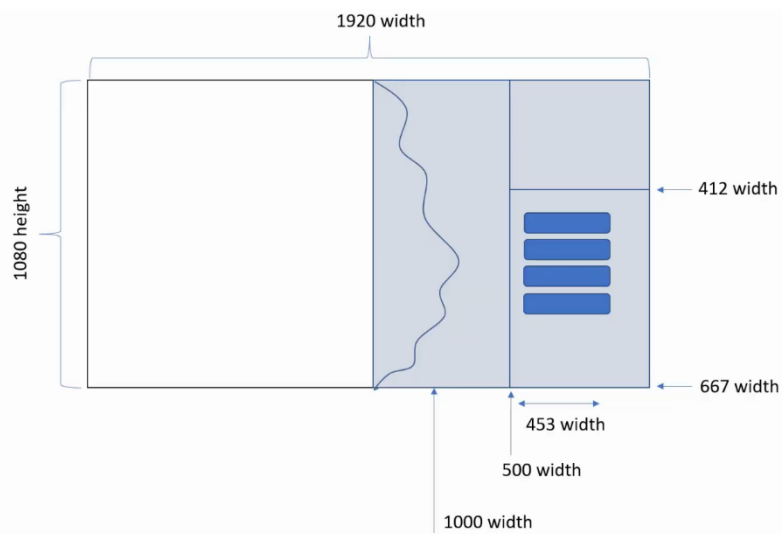
### 4.1 Tvorba uživatelského rozhraní

Prvním úkolem v praktické části bylo navrhnout a naprogramovat UI, tedy uživatelské rozhraní. Jelikož výsledná aplikace je vytvářena pro PC, používá se termín GUI – grafické uživatelské rozhraní. Grafická uživatelská rozhraní se stala nepostradatelným standardem při vývoji jakékoliv aplikace. Umožňuje totiž přímou manipulaci s grafickými ikonami, jako jsou tlačítka, posuvníky, okna, karty, nabídky apod. Většina práce se provedla v softwaru Unity 3D, verze 2019.4.11.f1, jako grafický editor byl použit Inkscape verze 0.9.2.4 a Gimp verze 2.10.12. Pro psaní skriptu byl použit Microsoft Visual Studio verze 16.8.4.

#### 4.1.1 Hlavní menu

Prvotní dojem o celkové aplikaci uživatel získá již v defaultní scéně, která se zobrazí po zpuštění aplikace. Tou scénou je *Menu*. Menu obsahuje základní panely, jako je přístup do hry. Přístup do hry umožní uživateli dále pokračovat, proto byly vytvořeny následující scény, kde se odehrává zbytek celého příběhu. Plány, jež obsahují zjednodušené grafické zpracování Zdikova paláce ve 2D podobě. Dále jsou zde panely tiráž, obsahující základní informace o aplikaci a autorovi. A panel ovládání, který pomocí 2D grafiky seznámí uživatele s ovládáním aplikace. Na závěr zde byl vytvořen panel ukončení.

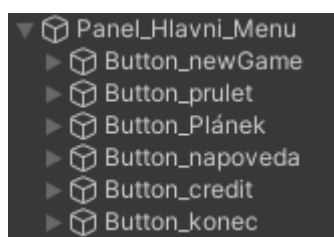
Jelikož hlavní menu slouží jako prvotní kontakt s uživatelem. Bylo hlavní menu vytvořeno s ohledem na atraktivnost, jako zajímavé a nevšední prostředí, které ihned od začátku uživatele zaujme. Celá aplikace byla optimalizovaná na FULL HD rozlišení tedy 1920×1080 pixelů, prvotní scéna byla navržena podle (Obr. 4.1.1.1).



Obr. 4.1.1.1 Rozložení scény Menu (zdroj: autor).

Unity engine poskytuje nástroje pro tvorbu *UI*. Díky tomu umožňuje jednoduchou manipulaci s *UI elementy*, jako jsou tlačítka, posuvníky, zaškrtačací pole, textová pole apod. Veškeré *UI elementy* musí být potomky objektu *Canvas*. *Canvas* vymezuje plochu, ve kterém se budou *UI elementy* nacházet, dále také disponuje třemi základními módy, které slouží pro vykreslování. Společně s *UI* je zde *Event System*, sloužící ke spuštění události na základě uživatelského vstupu. Aby *Event System* správně fungoval, musí být ve scéně právě jednou. *Event System* obsahuje komponenty *Touch Input Module* a *Standalone Input Module*. *Standalone Input Module* slouží ke zpřístupnění vstupu z klávesnice, myši a ovladačů (Starý 2015).

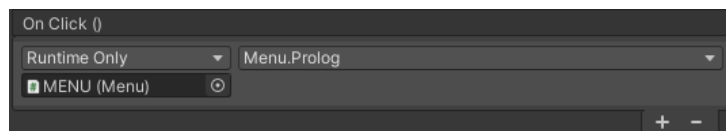
Prvním krokem bylo vytvoření objektu *Canvas* o rozlišení 1920×1080 pixelů. Do vytvořeného objektu byl přidán *UI element Panel* a komponent *Vertical Layout Group*, který umožňuje pracovat s jednotlivými prvky, jako s celkem. Tedy vypočítává vzdálenosti v daném *Panelu* pro všechny komponenty dohromady, a ne pro každý zvlášť. Na závěr byl vytvořen prázdný objekt, na který byl připojen skript *Menu*. Jednotlivými prvky v *UI elementu Panel*, byly objekty typu *Button*. Celkově bylo v defaultní scéně *Menu*, vytvořeno šest objektů typu *Button*.



Obr. 4.1.1.2 Objekty typu *Button* v *UI elementu Panel* (zdroj: autor).

Automaticky jim byla přiřazena funkce *OnClick()*. Funkce *OnClick()* funguje na principu toho, že se při kliknutí na *UI element* vykoná metoda připojeného skriptu. Důležité je, aby každý objekt typu *Button* správně odkazoval na část daného skriptu. Podmínkou bylo, aby dané metody byly *public*, díky tomu budou dané části skriptu viditelné ve funkci *OnClick()*. Avšak skript byl připojen pouze u vybraných *UI elementů* (*Button\_newGame*, *Button\_prulet* a *Button\_konec*). U *UI elementu Button\_newGame* byla přiřazena část skriptu, která po vykonání funkce *OnClick()* načte první cutscénu viz podkapitola 4.1.1.

```
public void Prolog()
{
    DefaultScene.SetActive(false);
    PrologCutScene.SetActive(true);
    CameraTimeline.SetActive(true);
    Default.enabled = false;
    StartProlog();
}
```



Obr. 4.1.1.3 Ukázka nastavení funkce *OnClick()* napojení na skript (zdroj: autor).

Pro zbývající *UI elementy* (*Button\_prulet* a *Button\_konec*) byly obdobně připojeny zbývající části skriptu.

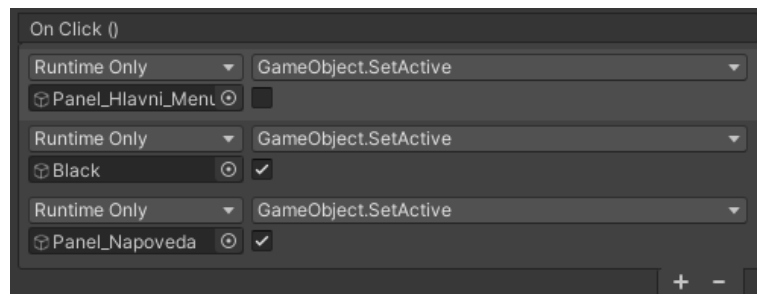
```

public void ActivationPrulet()
{
    SceneManager.LoadScene("Prulet"); // Načtení Scény s volbou prohlídky
}

public void QuitGame()
{
    Application.Quit(); // Vypnutí aplikace
}

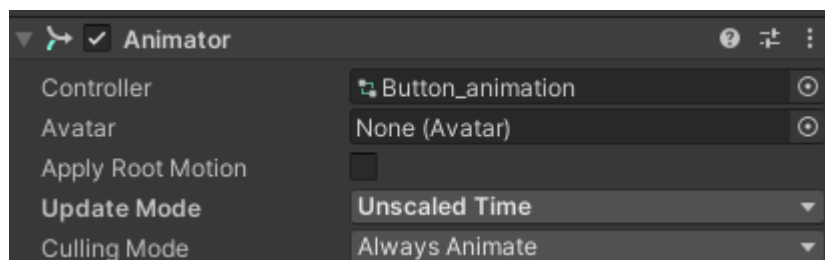
```

Dále byly vytvořeny tři *UI elementy Panel* (*Panel\_Planky*, *Panel\_Napoveda* a *Panel\_Credit*), které byly deaktivovány a to z důvodu, aby při spuštění aplikace se nepřekrývaly s defaultním panelem *Panel\_Hlavni\_Menu*. Pro objekty, kterým nebyla přiřazena část skriptu tedy (*Button\_napoveda*, *Button\_Plánek* a *Button\_credit*) byla již funkce *OnClick()* taktéž přidána, avšak její nastavení bylo odlišné. Byly zde přidány vybrané objekty. První objekt odkazuje na nově vytvořený *Panel\_Planky*, *Panel\_Napoveda* nebo *Panel\_Credit*, druhý objekt odkazuje na *Panel\_Hlavni\_Menu*, tedy *Panel*, který se automaticky zobrazí po spuštění aplikace. Je nutné v okně *Inspector* daný objekt nastavit jako aktivní. Pro zjednodušení práce na pozadí nebyly vytvořeny nové scény, pouze po provedení funkce *OnClick()* se aktivuje námi vybraný panel podle toho, jaký objekt typu *Button* se myší potvrdil a deaktivuje se defaultní *Panel\_Hlavni\_Menu*.



Obr. 4.1.1.4 Ukázka ovládání funkce *OnClick()* napojení na objekty (zdroj: autor).

Pro větší atraktivnost a orientaci, byly *UI Elementům* přidány animace: *Normal*, *Highlighted*, *Pressed*, *Selected* a *Disabled*. Aby animace správně fungovaly, musí každý *UI element* obsahovat komponent *animator* (Obr. 4.1.1.5). Do komponentu *animator* byl připojen komponent *Controller* (Obr. 4.1.1.6), který obsahuje námi vytvořené animace. Tyhle komponenty lze přidat v okně *Inspector*, komponent *Controller*, lze připojit tehdy, pokud byl námi již dříve vytvořen.

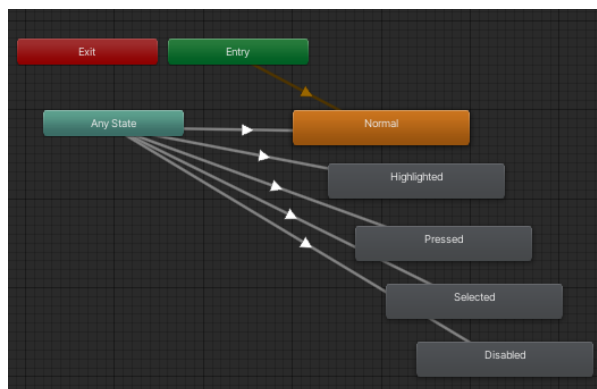


Obr. 4.1.1.5 Komponent *Animator* (zdroj: autor).

Animace *Normal*, zobrazuje *UI element* v základním stavu. Nastává tehdy jestliže se kurzor myši nedotýká daného *UI elementu*. Následující animace *Highlighted*, *Pressed*, *Selected* a *Disabled* zpustí specifické efekty, obdobné jejich názvům. Tedy animace

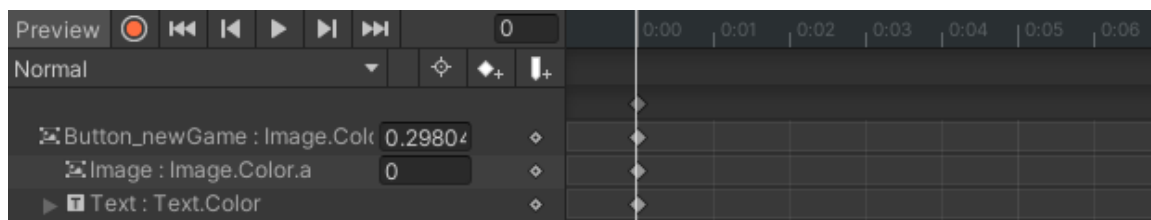


*Highlighted* spouští po přejetí kursoru myši po *UI element* efekt *highlights* – zvýrazňuje daný *UI element* nad ostatními. Po stisknutí levého tlačítka myši nad vybraným *UI elementem* proběhne efekt *Pressed* – odliší tak vybraný *UI element*, pokud neproběhne animace *Pressed*, vykoná se po odjetí kursoru myši z daného *UI elementu* animace *Disabled*, která vrátí *UI element* do základního stavu. V případě, kdy se vykoná animace *Pressed*, automaticky proběhne animace *Selected*, která zvýrazňuje již námi dříve vybraný *UI element*. Pro *UI elementy* typu *Button* byly v scéně *Menu* přidány pouze aplikace *Normal* a *Highlighted*.



Obr. 4.1.1.6 Controller pro UI elementy (zdroj: autor).

Samotné animace se vytvořily v okně *Animation* (Obr. 4.1.1.7), kde se nastavují parametry pro *UI elementy*. Jako je přemístění *UI elementu*, u *Image* a *Textu* se nastavuje barva, velikost, viditelnost a mnoho dalších parametrů. Aby se animace zaznamenávala, musí být aktivovaný *keyframe mode*. Rodina písma byla nastavena, jako *Vinque*. Jestliže by byl použit *Text Mesh Pro*, je potřeba vytvořit v projektu jeho vlastní *Font Asset*.

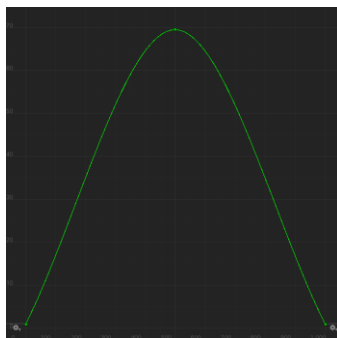


Obr. 4.1.1.7 Okno Animation pro UI elementy (zdroj: autor).

Jedním z grafických prvků jenž je dominantní na defaultní scéně, je nadpis. Nejdříve byl vytvořen objekt typu *Text*. Nadpis (Obr. 4.1.1.9) byl zakřiven pomocí skriptu od (Michael 2016). Tento skript avšak musel být upraven z důvodu jeho zastaralého zápisu a chyb. Část skriptu byla proto deaktivována. Následně byla v grafickém editoru vytvořena křivka, kterou text napodobuje a to tak, že z defaultní linie byla vytvořena konkávní křivka (Obr. 4.1.1.8).

```
//protected override void OnRectTransformDimensionsChange()
//{
//    var tmpRect = curveForText[curveForText.Length - 1];
//    tmpRect.time = rectTrans.rect.width;
//    curveForText.MoveKey(curveForText.Length - 1, tmpRect);
//}
```





Obr. 4.1.1.8 Konkávní křivka sloužící, jako vodítka pro text (zdroj: autor).

Jedním z viditelných efektů je *Blur effect* (Obr. 4.1.1.9). Pro jeho vytvoření byl využit *shader* od autora Denisnk. Nejdříve byl vytvořen v projektu nový *shader* přesněji *Standart Surface Shader*, do kterého byl importován skript od Denisnk. Následovalo vytvoření nového materiálu, jehož *Surface Shader* byl změněn podle skriptu na *BLUR*. Dále byl v dotčeném *Canvasu* přidán další *Panel* o rozlišení 1920 x 1080 pixelů tak, aby pokrýval celou scénu. V hierarchii byl umístěn tak, aby zakrýval pouze vybrané objekty. Samotný materiál se potom nastavil podle potřeby, kde byly změněny velikosti a transformace pixelů.

Veškeré 2D grafické prvky nacházející se v hlavním menu byly vytvořeny v grafickém editoru Gimp a Inkscape. Po exportu jako PNG / JPG formát do software Unity musí být jejich typ upraven jako: *Sprite (2D and UI)*. Následně již do vytvořených *UI elementů* byl přidán typ *Image*, kde byly tyto *UI* napojeny.



Obr. 4.1.1.9 Finální verze scény defaultní scéna – Menu (zdroj: autor).

## 4.1.2 Prohlídka

Ve scéně prohlídka má uživatel možnost volby scény ve vybraných atmosférických podmínkách. Hráč po aktivaci *UI Button Prohlídka* se ocitne v scéně *Průlet*. Zde má možnost si zvolit, v jakých atmosférických podmínkách si chce Zdikův Palác prohlížet. Scéna *Průlet* obsahuje *UI elementy* které spouští funkce, jež dovolují pokračovat dále, či se vrátit zpět do defaultní scény *Menu*. Podrobnější funkcionalita scény *Prohlídka* popisuje podkapitola 4.5, vytvoření výběru atmosférických podmínek se zabývá podkapitola 4.2.

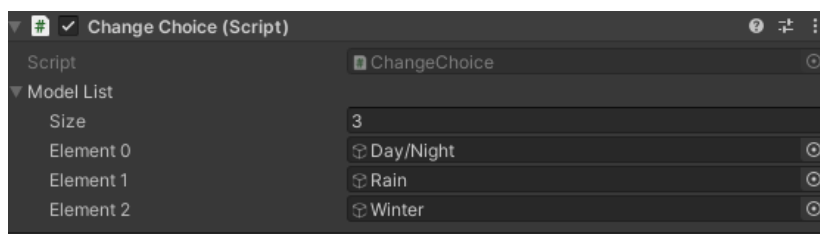
Pro plnou funkcionalitu bylo nutné vyřešit problém vztahující se k seznamu modelů. Aby se jednotlivé modely Zdikova paláce zobrazovaly ve správném pořadí, byl každému modelu přiřazen jedinečný index. Zároveň zde byla vytvořena podmínka, která aktivovala po spuštění scény první model v seznamu.

```
void Start()
{
    NextScene = false;
    index = PlayerPrefs.GetInt("ModelSelected");
    ModelList = new GameObject[transform.childCount];
    for (int i = 0; i < transform.childCount; i++)
        ModelList[i] = transform.GetChild(i).gameObject;

    foreach (GameObject Scene in ModelList)
        Scene.SetActive(false);

    if (ModelList[index])
        ModelList[index].SetActive(true);
}
```

Tomu to předcházelo vytvoření seznamu *ModelList*. Zde se přidaly objekty, které evokovaly, jakou ze tří atmosférických podmínek si uživatel vybere.



Obr. 4.1.2.1 Seznam již vytvořených atmosférických podmínek (zdroj: autor).

Na základě těchto indexů se doprogramovaly metody, které aktivovaly /deaktivovaly jednotlivé objekty v seznamu. Objekty v seznamu měly pouze informativní hodnotu sloužící pro předávání informací metodám, které umožňovaly procházet seznam zleva doprava a naopak. Podmínkou bylo, aby dané metody byly *public* a návratovým typem pro metody byl zvolen typ *void*.

```
public void ToggleLeft()
{
    CharacterList[index].SetActive(false);

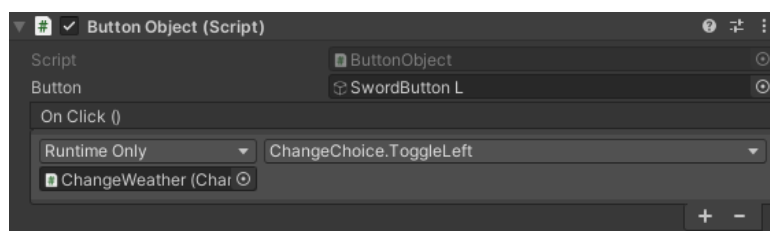
    index -= 1;
    if (index < 0)
        index = CharacterList.Length - 1;
    CharacterList[index].SetActive(true);
}
```

Dále bylo nezbytné tyto metody přiřadit k jednotlivým objektům typu *Button*. Za objekty typu *Button* byly vybrány 3D modely mečů od autora iceboxX708. Meče byly vybrány pro lepší atraktivitu a nejedná se tak o *UI elementy*, které mají předdefinované chování, jako je např. funkce *OnClick()*. Bylo zapotřebí pomocí scriptu těmto 3D modelům přidat funkci *OnClick()*. Toho se docílilo na základě *Physics.Raycast*, jež nám detekuje 3D model.

```
void Update()
{
    var ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    RaycastHit Hit;

    if (Input.GetMouseButtonDown(0))
    {
        if (Physics.Raycast(ray, out Hit) && Hit.collider.gameObject == gameObject)
        {
            OnClick.Invoke();
        }
    }
}
```

Aby script správně fungoval, byly přidány jednotlivým 3D modelům tzv. *Collider komponenty*. Na základě nich software pozná, zda se kurzor myši nachází v požadované oblasti a dovolí tak provést funkci *OnClick()*. Díky tomu, že jednotlivé metody byly *public*, staly se tak viditelnými v okně *Inspector*. V okně *Inspector* byly přidány jednotlivým 3D modelům požadované metody. Také k nim byl připojen skript *Outline* od Chrise Nolita, jež zvýrazňuje a odlišuje vybrané objekty od ostatních pomocí linií.



Obr. 4.1.2.2 Připojení skriptu na 3D model meče, aktivace funkce *OnClick()* (zdroj: autor).

Následně byl vytvořen skript, který kontroloval aktivaci / deaktivaci objektů v seznamu a na základě toho aktivoval / deaktivoval jednotlivé objekty ve hře tak, aby se správně zobrazovaly atmosférické podmínky.

```
// Obdobně byly přidány podmínky i pro objekty Day / Night a Winter
else if (ChangeRain.active)
{
    rain.SetActive(true);
    day.SetActive(false);
    winter.SetActive(false);
    RenderSettings.skybox = RainingMat;
    WinterTerrain.SetActive(false);
    SummerTerrain.SetActive(true);
    PartRain.SetActive(true);
    VFXSnow.SetActive(false);
}
```

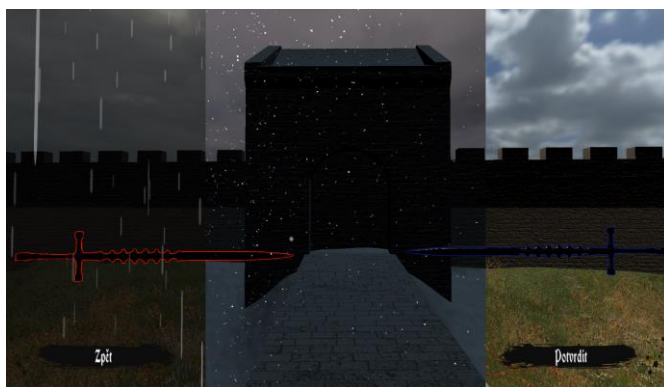
Na závěr byly přidány další dva *UI elementy* *Potvrdit* a *Zpět*. Oběma *UI elementům* byla přiřazena funkce *OnClick()*, každá z nich odkazovala na jednotlivou část scriptu. V případě *UI elementu* *Potvrdit* funkce odkazovala na metodu *ConfirmButton()*, jež po její aktivaci

aktivovala kameru s volným pohybem. V případě *UI elementu Zpět* funkce odkazovala na metodu *ZpetMenu()*, jenž po její aktivaci vrátila uživatele o krok dozadu, tedy do defaultní scény *Menu*.

```
public void ConfirmButton()
{
    PlayerPrefs.SetInt("CharacterSelected", index);
    SceneManager.LoadScene("Portal");
}

public void ZpetMenu()
{
    SceneManager.LoadScene("Menu");
}
```

Stejně, jako tomu bylo v předešlé podkapitole, bylo *UI elementům* přidány totožné animace: *Normal* a *Highlighted*.



Obr. 4.1.2.3 Výběr atmosférických podmínek (zdroj: autor).

### 4.1.3 Další položky menu

V defaultní scéně *Menu* byly vytvořeny již zmiňující tři *UI elementy* typu *Panel*. Po rozkliknutí *Panel\_Planky* se uživateli zobrazí dvě patra paláce Jindřicha Zdíka s podrobným popisem jednotlivých místností (Obr. 4.1.3.1). Jejich vytvoření se věnuje podkapitola 4.3.2.



Obr. 4.1.3.1 Přizemí Jindřicha Zdíka znázorněné pomocí plánku (zdroj: autor).

Byly zde přidány dva objekty typu *Button* evokující změnu patra paláce Jindřicha Zdíka. Nejdříve byl vytvořen seznam pro dvě patra a každému patru byl stejně jako v minulé podkapitole přiřazen jedinečný index. Objektům typu *Button* byla následně přiřazena odpovídající metoda, jež umožňuje listovat seznam doprava nebo doleva. Všechny zmíněné objekty typu *Panel* měly přidáné objekty typu *Button* s metodou, která vracela uživatele do defaultního menu.

Panel *Napoveda* zobrazuje pomocí grafiky (Obr. 4.1.3.2) ovládání pro hlavní a vedlejší příběh. Grafika byla vytvořena v softwaru Inkscape, po převedení do Unity byla grafika změněna na typ *Sprite (2D and UI)*. *Panel\_Napoveda* je tvořen objekty typu *Image*, *Text* a *Button*. *Panel\_Credit* obsahuje tiráž.



Obr. 4.1.3.2 Nápoveda ovládání pro hlavní příběh (zdroj: autor).



Obr. 4.1.3.3 Tiráž (zdroj: autor).

## 4.2 Dynamická změna prostředí

V hlavní scéně, kde se odehrává příběh, se dynamicky střídá den a noc. Cyklus dne trvá v reálném čase necelých šest minut a stejně dlouho potom trvá i cyklus noci. Prvním krokem k tomu bylo nutné v projektu vytvořit dva nové materiály pro noc a den. Jejich *shader* byl změněn na *Skybox/Procedural*. V dalších krocích se tyto materiály využijí pro grafické nastavení a vykreslení atmosféry.

Dynamická změna počasí běží nezávisle na hráči v pozadí. Správné fungování daných materiálů, které evokují buď den nebo noc, je závislé na poloze dvou objektů *Moon* a *Sun* typu *Light* o rozdílu  $180^\circ$ . Tyto objekty neustále rotují kolem dané osy X, každý má své vlastní grafické nastavení, avšak vždy pouze jeden z nich je aktivován v intervalu  $0^\circ - 180^\circ$ . Je právě důležité, aby vždy byl aktivní pouze jeden z důvodu rozdílných nastavení, ať už se to týká stínů, barvy, intenzity a podobně.

```
// Den trvá 24 jednotek, neboli čas, za který se objekt typu Light otočí o 180°
void Update()
{
    timeDay += Time.deltaTime * orbitSpeed;
    if (timeDay > 24)
    {
        timeDay = 0;
    }
    UpdateTime();
}

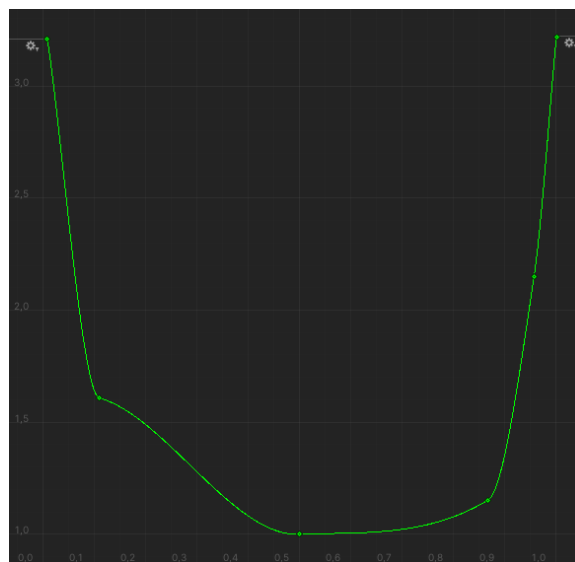
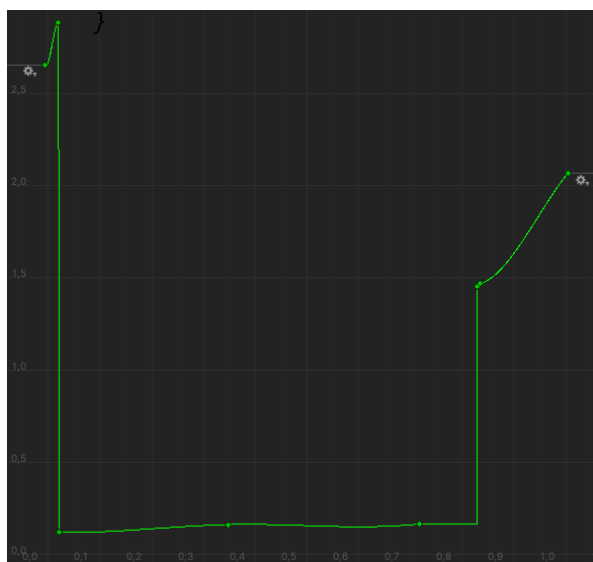
// Aby se objekt neustále otáčel kolem osy x, byl vytvořen navazující část skriptu
private void UpdateTime()
{
    float newCycle = timeDay / 24.0f;
    float sunRotation = Mathf.Lerp(-90, 270, newCycle);
    float moonRotation = sunRotation + 180;
    sun.transform.rotation = Quaternion.Euler(sunRotation, -150.0f, 0);
    moon.transform.rotation = Quaternion.Euler(moonRotation, -150.0f, 0);
    CheckMaterial();
    CheckDayNight();
}

// Skript také kontroloval, aby vždy byl aktivní pouze jeden objekt typu Light a aby
// správný materiál byl přiřazen v okně Lightning

if (isNight == true)
{
    moon.GetComponent<Light>().enabled = true;
    sun.GetComponent<Light>().enabled = false;
    if (moon.transform.rotation.eulerAngles.x > 180)
    {
        StartDay();
    }
}
else
{
    moon.GetComponent<Light>().enabled = false;
    sun.GetComponent<Light>().enabled = true;
    if (sun.transform.rotation.eulerAngles.x > 180)
    {
        StartNight();
    }
}
```

Pro co nejvíce realistické grafické nastavení atmosféry bylo pracováno s nastavením intenzity jednotlivých objektů typu *Light* a barvou atmosféry tedy s daným materiálem. Zbarvení atmosféry se nastavovalo pomocí hodnot *Atmosphere Thickness*, jedná se o defaultní barevnou stupnici. Avšak aby se dané hodnoty dynamicky měnily, bylo zapotřebí vytvořit skript, který sleduje pohyb jednoho z objektů typu *Light*. A současně byla vytvořena matematická / animační křivka (Obr. 4.2.1), podle které se dané hodnoty nastavení dynamicky mění v závislosti na poloze objektu. Osa X znázorňuje tedy polohu objektu, osa Y znázorňuje hodnoty daného nastavení materiálu či objektu typu *Light*. Počátek soustavy souřadnic značí to, že dané objekty typu *Light* dosáhly na ose X 0°, tedy započaly svůj nový cyklus.

```
void Update()
{
    RenderSettings.skybox.SetFloat("_AtmosphereThickness", atmosphereThickness);
    var newAtmosphereThicknessDay =
        AtmosphereIntensity.Evaluate(transform.rotation.eulerAngles.x / 180f);
    atmosphereThickness = newAtmosphereThicknessDay;
}
```

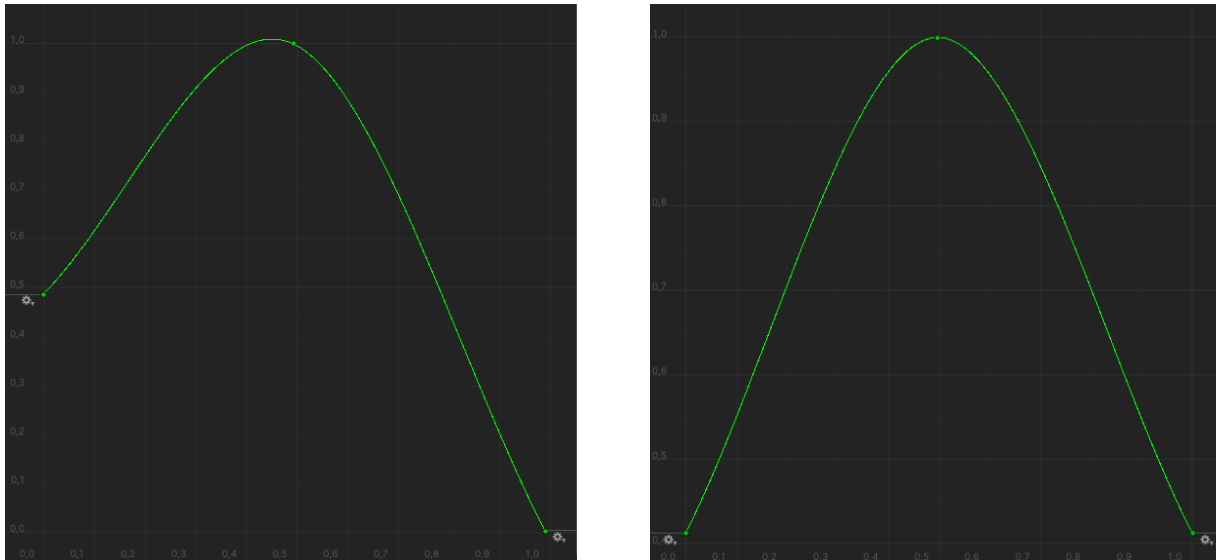


Obr. 4.2.1 Hodnoty pro změnu barvy atmosféry v závislosti na poloze objektu (Moon) a (Sun) (zdroj: autor).

Obdobně se postupovalo při nastavení intenzity objektu typu *Light*. Tedy s ubíhajícím časem nebo jejich polohou se měnila jejich intenzita podle předem definované matematické / animační křivky. Rozptyl hodnot byl u objektu Moon mnohonásobně menší než u objektu Sun z důvodu, že noc ve hrách nikdy nebývá úplně tmavá.

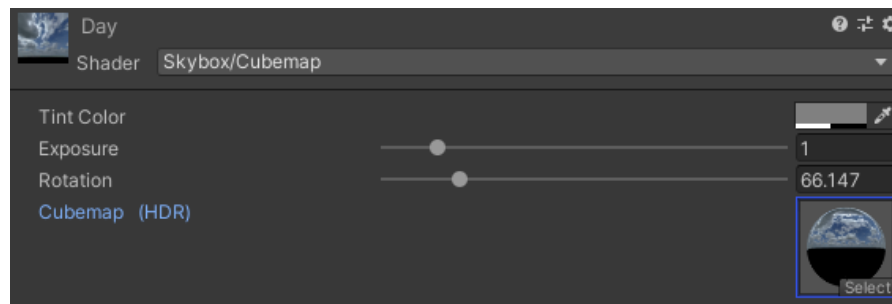
```
void Update()
{
    var newIntensity =
        CurveIntensity.Evaluate(transform.rotation.eulerAngles.x / 180f);
    light.intensity = newIntensity;
}
```





Obr. 4.2.2 Hodnoty pro změnu intenzity světla v závislosti na poloze objektu (Moon) a (Sun) (zdroj: autor).

Ve scéně *Průlet* byl vytvořen výběr atmosférických podmínek pro sluneční den, déšť a sníh. Nejdříve byly obdobně, jako v předešlé části vytvořeny tři nové materiály evokující grafické nastavení pro déšť, sníh a běžný slunečný den. Jejich *shader* byl nastaven na *Skybox/Cubemap*. Pro každý materiál byl vytvořen jedinečný objekt typu *Light* s vlastním nastavením. Na rozdíl od prvního způsobu, kdy se měnily defaultní hodnoty *Skybox* nastavené samotným softwarem Unity, se zde se využilo metody, do které vstupují *HDR maps*. *HDR maps* byly stáhnuty od (HDRMAPS, nedatováno) a upraveny v grafickém editoru Gimp. Následně byly jejich *texture shape* změněny v Unity z typu 2D na typ *cube*. Díky tomu se dané *HDR maps* dají přiřadit jako *Cubemap* do předem vytvořených materiálů.



Obr. 4.2.3 Hodnoty nastavení materiálu – Cubemap (zdroj: autor).

U třech objektů typu *Light* byl vytvořen skript, který rotuje s *Cubemap* a simuluje tak pohyb mraků v atmosféře. Pro objekt zaměřený na vizualizaci deště, byl ještě dopsán část skriptu, která vytváří dojem blesku. A to tak, že je v náhodných úsecích přidáván či snižován jas. U objektů simulující scénu v dešti a scénu v zimě byly vytvořeny podle Binary Impact a Goldman Studio pomocí *VFX* a *Particle system* efekty modelující padající sníh či déšť (Obr. 4.1.2.3).

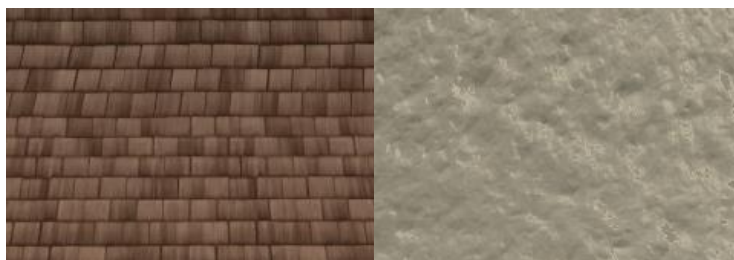


```

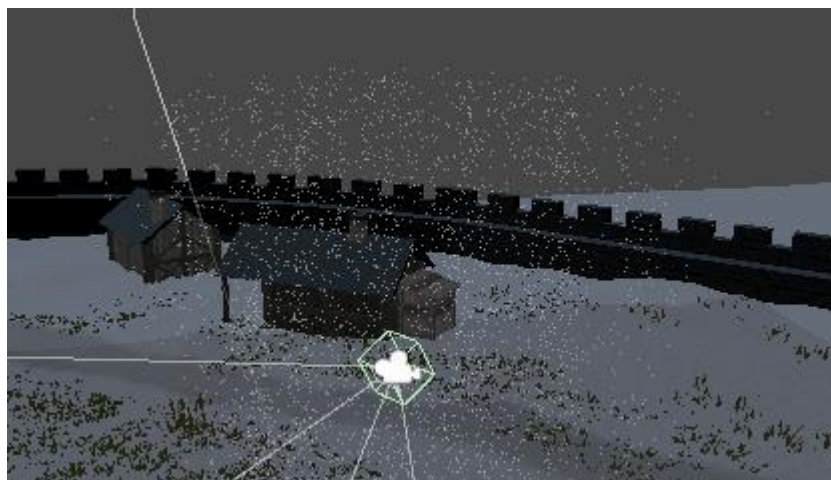
IEnumerator Thunder() //Vykreslení efektu blesku
{
    number = Random.Range(3, 8);
    for (int i = 0; i < number; i++)
    {
        yield return new WaitForSeconds(0.1f);
        RenderSettings.skybox.SetFloat("_Exposure", exposure + randValue);
        yield return new WaitForSeconds(0.1f);
        exposure = 1;
        RenderSettings.skybox.SetFloat("_Exposure", exposure);
    }
    exposure = 1;
    RenderSettings.skybox.SetFloat("_Exposure", exposure);
}

```

Jak již bylo zmíněno v podkapitole 4.1.2, uživatel si může libovolně vybírat atmosférické podmínky. Pro simulaci slunečního dne a deště byl zachován terén s veškerou vegetací, jako v hlavním příběhu. Pro simulaci sněhu byl vytvořen nový terén digitálního modelu reliéfu, který byl otexturován sněhem a zůstala zachována pouze vybraná vegetace. Pro větší důvěryhodnost byl využit a pozměněn *shader* od World of Zero. Nově vytvořený *shader* se aplikoval na vybrané objekty. *Shader* funguje na principu průniku *normal maps* vybraných objektů s danou texturou barvy evokující sních. Díky tomu vznikne efekt zasněžených objektů (Obr. 4.2.4). Padající sních byl vytvořen pomocí stáhnutého balíčku *VFX graph*, kde je možnost pro dosažení lepšího výkonu nastavit vykreslování pro zvolenou vzdálenost (Obr. 4.2.5). Pro synchronizaci pohybu kamery a *VFX efektu* bylo potřeba *VFX efekt* připojit ke kameře pomocí skriptu tak, aby uživatel při pohybu měl pocit, že padá sních po celém území. Stejného principu se využilo pro efekt padajícího deště, avšak rozdíl byl v tom, že déšť byl vytvořen pomocí *Particle systému*. Efekt sněhu a deště se deaktivuje na základě skriptu, jakmile uživatel vstoupí do paláce a zase se aktivuje po jeho opuštění viz podkapitola 4.3.2.



Obr. 4.2.4 Změna textur pro objekty v zimě s využitím normal map (zdroj: autor).



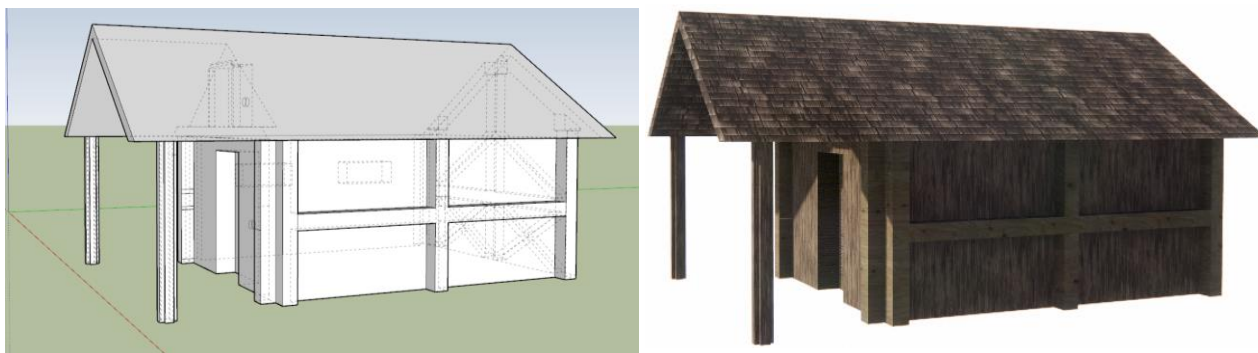
Obr. 4.2.5 Efekt padajícího sněhu kolem kamery (zdroj: autor).

## 4.3 Modelování objektů

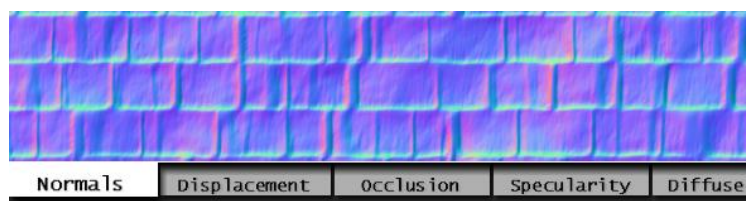
V práci byl využit otexturovaný model Zdíkovy paláce od firmy Tasty Air, který avšak neobsahoval předměty v interiéru a žádné okolní objekty. Proto samotné modelování bylo zaměřeno na interiér a tvorbu objektů v jeho okolí. 3D modelování probíhalo ve dvou softwarech. Pro urbanistický objekty byl použit Sketchup, pro organické objekty byl využíván software Zbrush. Pro úpravu textur byl využit software CrazyBump. Veškeré použité textury byly od autora Joao Paulo. V určitých případech se 3D modely převáděly do 2D podoby, a k tomu byly použity grafické editory Gimp a Inkscape.

### 4.3.1 Modelování interiéru a okolí

Při 3D modelování byl kladen důraz na čistotu geometrie a to z důvodu texturování. Každý model byl modelován podle zásad, aby byl výsledný model použitelný i k 3D tisku. Vymodelovány byly čtyři typy chalup, které se nacházejí v okolí Zdíkovy paláce a také nejrůznější assety pro interiér. Následně byly modely otexturovány a po importu do herního engine pomocí softwaru CrazyBumb byly z jednotlivých textur vytvořeny chybějící mapy (Obr. 4.3.1.2). Jednotlivé 3D modely byly poté rozmístěny na dané místa v herním engine Unity (Obr. 4.3.1.3) a byl jim přidán *Mesh collider* tak, aby jimi hráč nemohl procházet. Taktéž v softwaru Sketchup byla vytvořena a zneviditelněna umělá hradba, která omezuje pohyb hráče na území Václavského návrší ve scéně *Průlet*.



Obr. 4.3.1.1 3D modelu v softwaru Sketchup a 3D otexturovaný 3D model (zdroj: autor).



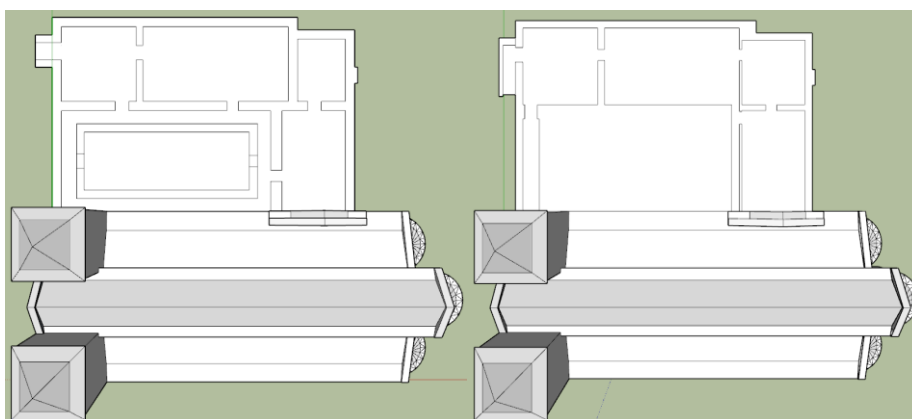
Obr. 4.3.1.2 Vytvoření chybějících map v softwaru CrazyBumb pro materiál v Unity (zdroj: autor).



Obr. 4.3.1.3 Rozmístění vytvořených 3D modelů včetně 3D modelů od LBobor (zdroj: autor).

### 4.3.2 Modelování minimap

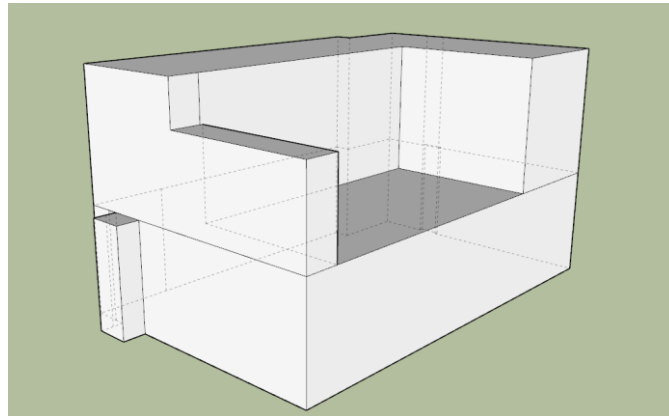
Dalším krokem bylo vytvoření minimap z důvodu lepší orientaci ve hře. V hlavním příběhu hráč využívá celkem tři minimapy, které se mění na základě pozice hráče. První minimapa zobrazuje celé okolí včetně Zdikova paláce, zbývající minimapy znázorňují jeho jednotlivá patra. Jelikož Sketchup slouží pro profesionální architekty, bylo vhodné dané minimapy vytvořit, jako architektonické výkresy v podobě půdorysu, aby byla co nejvíce zachována přesnost. Pro vymodelování minimap, bylo zapotřebí palác Jindřicha Zdíka rozdělit do dvou pater. V programu Sketchup byly vytvořeny dva modely evokující přízemí a první patro Zdikova paláce. Přízemí a první patro (Obr 4.3.2.1) bylo vytvořeno z hlavního modelu nacházející se v aplikaci. Vytvořené modely byly také převedeny do grafických editorů, z jejichž podkladů byly vytvořené plány viz podkapitola 4.1.3.



Obr. 4.3.2.1 3D model přízemí a 1. patra pro minimapu (zdroj: autor).

Současně byly vymodelovány objekty, které detekují hráče a na základě toho skript vykreslí v minimapě přízemí nebo 1. patro Zdikova paláce. Tyhle objekty byly sloučeny v jeden (Obr 4.3.2.2), čehož bylo využito i pro aktivaci a deaktivaci efektů sněhu a deště.

```
if (Patro1Trigger.GetComponent<TriggerMinimap1>().InPatro1 == true)
{
    Patro1UI.SetActive(true);
    Default.SetActive(false);
    Patro2UI.SetActive(false);
    MinimapCamera.orthographicSize = 15f;
    MinimapCamera.cullingMask = (1 << LayerMask.NameToLayer("Minimap"));
}
else if (Patro2Trigger.GetComponent<TriggerMinimap2>().InPatro2 == true)
{
    Patro1UI.SetActive(false);
    Default.SetActive(false);
    Patro2UI.SetActive(true);
    MinimapCamera.orthographicSize = 15f;
    MinimapCamera.cullingMask = (1 << LayerMask.NameToLayer("Minimap"));
}
else
{
    Patro1UI.SetActive(false);
    Patro2UI.SetActive(false);
    Default.SetActive(true);
}
```



Obr. 4.3.2.2 3D model objektu pro detekce hráče (zdroj: autor).

V softwaru Inkscape bylo vytvořeno velmi zgeneralizované okolí a půdorys samotného Zdikova paláce. Byly zde také graficky upraveny jednotlivé modely pater Zdikova paláce do podoby 2D.

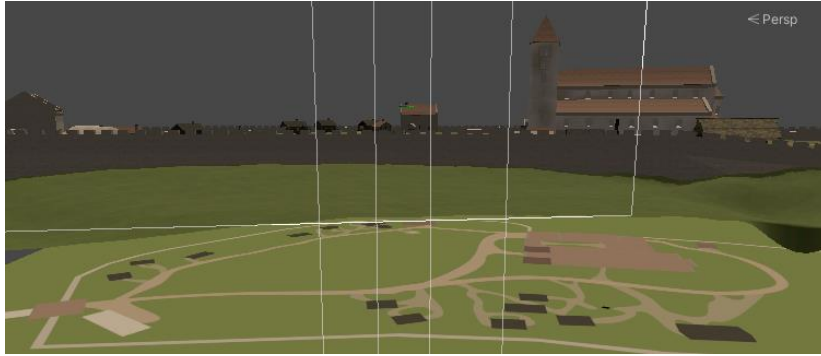


Obr. 4.3.2.3 2D model zgeneralizovaného okolí kolem Zdikova paláce (zdroj: autor).

Importované 2D objekty byly změněny na: *Sprite (2D and UI)*, jejich *Layer* byl změněn z defaultního na nově vytvořený. Objekty se umístily pod hlavní model. Samotná minimapa se vykresluje pomocí ortografické kamery, ta byla zvolena proto, aby nedocházelo ke zkreslení. Zpočátku byl v Unity vytvořen objekt typu *Render Texture*, kdy objekt byl přiřazen vybrané kameře (Obr. 4.3.2.4). Také zde byl zvolen nově vytvořený *Layer* a byl přiřazen jako *Culling Mask*, díky tomu kamera pouze vykresluje 2D objekty minimap (Obr. 4.3.2.5). Na vytvořené modely pro minimapy nepůsobí žádné světelné nastavení.



Obr. 4.3.2.4 Nastavení kamery (zdroj: autor).



Obr. 4.3.2.5 Umístění vytvořených modelů pro minimapy pod hlavním modelem (zdroj: autor).

Byl také vytvořen kurzor, jež sleduje pohyb hráče (Obr. 4.3.2.6) a který je vykreslený pouze kamerou pro minimapu. Pro dynamicky se měnící prostředí byl vytvořen skript, který sleduje pohyb hráče a na základě toho se mění poloha kamery.

```
void LateUpdate()
{
    Vector3 newPosition = player.position;
    newPosition.y = transform.position.y;
    transform.position = newPosition;
    transform.rotation = Quaternion.Euler(90f, player.eulerAngles.y, 0f);
}
```



Obr. 4.3.2.6 Kurzor sledující hráče (zdroj: autor).



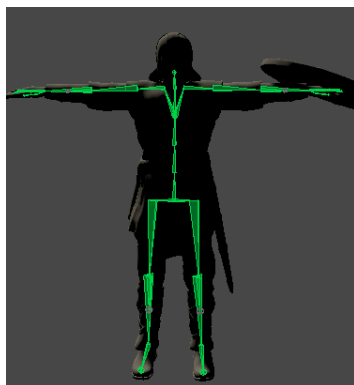
Obr. 4.3.2.7 Ukázka hotové minimapy (zdroj: autor).



## 4.4 Hlavní příběh

Tvorbou hlavního příběhu došlo k samotné realizaci gamifikovaného průvodce. Ze začátku je hráč pomocí cut-scén vnořen do příběhu, kde mu jsou vysvětleny okolnosti. Je seznámen, kdo je a co ho v příběhu čeká. Následně ho průvodce, využívající umělou inteligenci (AI), provází po paláci. Ve formě titulků, jsou uživateli prezentovány historické, ale i další zajímavé informace o daných místnostech. Pomocí vytvořených dialogů může hráč rozvíjet konverzaci s průvodcem a doptávat se na nejrůznější věci. Současně hráč může libovolně interagovat s předměty. Byla zde vytvořena metoda, která omezuje fyzickou zdatnost hráče. V hlavním příběhu dochází k dynamické změně denní doby. Po prozkoumání paláce se spouští samotná hra, kdy hráči začne ubývat čas, do kterého musí splnit nejrůznější úkoly. Přitom využívá nabyté informace, které mu byly sděleny během prohlídky. Po splnění všech úkolů v daném čase hra končí.

Pro základní pohyb postav bylo nutné získat a zpracovat vybrané animace. Animace byly stáhnuty z Mixamo, kde byl vytvořen i automatický rigging systém pro postavy (Obr. 4.4.1). Jednotlivé postavy bylo potřeba upravit v programu Zbrush do T pozice.



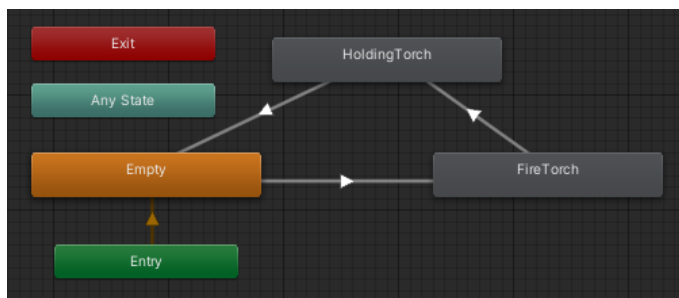
Obr. 4.4.1 Výsledek auto rigging charakteru v T pozici, 3D model od Andy woodhead (zdroj: autor).

Po importu do herního engine Unity, byl jednotlivým postavám *Animation* typ změněn na *Humanoid*. Dále byl vytvořen *Animator Controller*, který slouží jako grafické rozhraní pro práci s animacemi. U hlavní postavy bylo v *Animator Controller* vytvořeno pět *Layers* sloužící pro základní pohyb, zdvih předmětů ze země, zdvih předmětů z výšky, zahození předmětů a zapálení louče. *Layers* slouží pro rozdělení animací jako celku při využití dvou a více rozdílných úkonů. Platí podmínka, že musí být vždy jeden aktivní *Layers*. Tím byl zvolen defaultní pro pohyb, který byl vytvořený podle *Animator Controller*, jako dynamický *locomotion animsets*. Zde je potřeba rozdělit jednotlivé úkony pro postavu a to když postava běží, kráčí nebo stojí na místě. Pro tenhle účel byl využit *blend tree*. Základní *blend tree* se dělí na další. Tohoto bylo využito pro hladký přechod mezi chůzí, během a stáním.

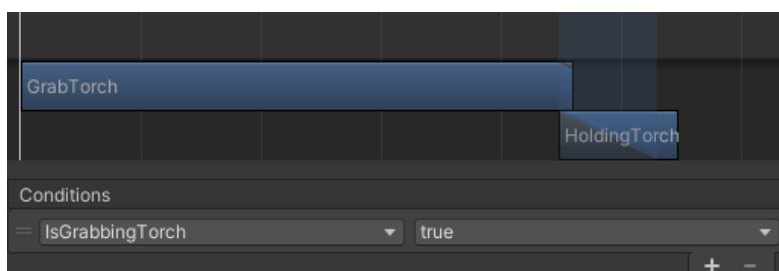
Pro zbylé *Layers* evokující (zdvih předmětu z výšky, zdvih předmětu ze země, odhození předmětu a zapálení louče) bylo potřeba vytvořit objekt typu *Avatar*. V nově vytvořeném objektu typu *Avatar* se aktivují / deaktivují části, které požadujeme pro přepsání defaultních dat. Poté byl do *Animator Controller* přiřazen nově vytvořený objekt typu *Avatar* do *Layers* a byl parametr *Weight* nastaven na rovno 1.

Po přidání animací do *Animator Controller* (Obr. 4.4.2) pomocí námi vytvořených parametrů (Obr. 4.4.3) můžeme přidat podmínky, podle kterých se jednotlivé animace mohou řídit. Tyto podmínky jsou i ovladatelné pomocí skriptu. Na podobném principu byly vytvořeny zbývající *Layers*, pro každý *Layers* byl vytvořený objekt typu *Avatar*.

Jelikož jsou využívány tři typy rozdílných postav – hlavní postava, strážci a průvodce. Pro každý z nich byl vytvořen rozdílný *Animator Controller* s vlastním nastavením animací. Na závěr byl u každé postavy přidán komponent *Animator*.



Obr. 4.4.2 Layers FireTorch, přidání a nastavení animací (zdroj: autor).



Obr. 4.4.3 Přidání vytvořené podmínky do transition mezi jednotlivé animace (zdroj: autor).

#### 4.4.1 Cut-scény a dialogy

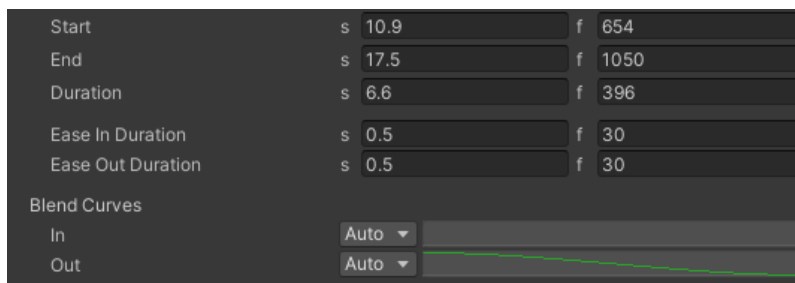
Pro tvorbu cut-scén byl využit balíček *Timeline* od Unity verze 1.2.13. Cut-scény využívají po sobě navazující titulky, které defaultně nejdou v *Timeline* vytvořit. Prvním úkolem bylo doprogramování editace titulků. Jako inspirace byl využit návod od Game Dev Guide. Skript pracuje s *Text Mesh Pro*, pro jeho správnou funkcionalitu musel být vytvořen jeho vlastní *Font Asset*.

```

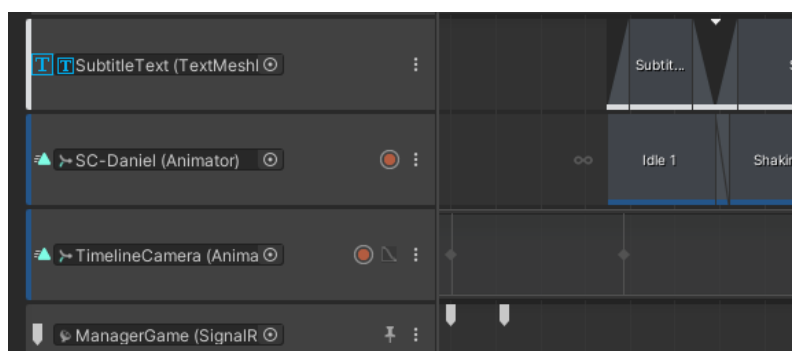
if (!text) { return; }
int inputCount = playable.GetInputCount();
for (int i = 0; i < inputCount; i++)
{
    float inputWeight = playable.GetInputWeight(i);
    if (inputWeight > 0f)
    {
        ScriptPlayable<SubtitleBehaviour> inputPlayable =
        (ScriptPlayable<SubtitleBehaviour>)playable.GetInput(i);
        SubtitleBehaviour input = inputPlayable.GetBehaviour();
        currentText = input.subtitleText;
        currentAlpha = inputWeight;
    }
}
text.text = currentText;
text.color = new Color(1, 1, 1, info.weight);
  
```

Pomocí upraveného skriptu od Game Dev Guide, bylo možné v *Timeline* titulky editovat. Celkem bylo vytvořeno šest cut-scén. V okně *Hierarchy* bylo vytvořeno šest objektů, odkazující na jednotlivé *Timeline*, a každý objekt obsahoval objekt typu *Canvas*, ve kterém byly objekty, kterým byly přidány komponenty typu *Text Mesh Pro*. Až na první cut-scénu, kde se dynamicky pohybuje pouze kamera, obsahují cut-scény vždy *Animator* pro hlavní postavu a průvodce. Ten pomáhá postupně čist animace, tak jak byly poskládané v okně

*Timeline*. Dále obsahují kameru, do které byl přidán komponent *Animator* z důvodu tvorby animací a jejich zaznamenání v okně *Timeline*. Také je zde přidána na sebe navazující sekvence titulků s vlastní stopáží (Obr. 4.4.1.1). A na závěr zde byly vytvořeny značky (Obr. 4.4.1.2), které slouží pro aktivaci vybraných metod ve skriptu.



Obr. 4.4.1.1 Ukázka nastavení jedné sekvence titulků (zdroj: autor).

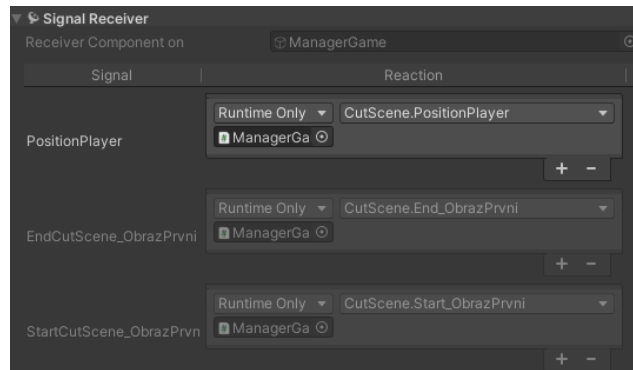


Obr. 4.4.1.2 Timeline (zdroj: autor).

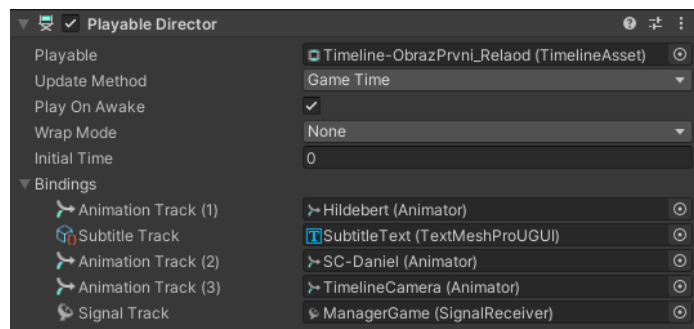
Jednotlivé značky (*Signal*) obsahují (Obr. 4.4.1.3) odkazy na metody ve skriptu, které se vždy provedou po jejich aktivaci. Ty musely být přiřazeny, jak v objektu typu *Signal*, tak v objektu, který obsahoval skript s metodami a v jednotlivých objektech volající *Timeline*. Tytéž objekty obsahují komponent *Playable Director* a *Animator* (Obr. 4.4.1.4). Pro nerušené zážitky ze hry je po celou dobu deaktivovaný kurzor, který se aktivuje pomocí již zmiňovaných značek v případě, že ho hráč potřebuje k jednotlivým úkonům ve hře.

```
public void PositionPlayer()
{
    players.gameObject.transform.position = new Vector3(73.385f, 36.60f, 48.051f);
    players.transform.eulerAngles = new Vector3(0f, -63f, 0f);
}
public void Start_ObrazPrvni()
{
    Cursor.visible = false;
    CameraMinimap.SetActive(false);
    Stats.SetActive(false);
}
```





Obr. 4.4.1.3 Vytvoření značek odkazující na jednotlivé metody ve skriptu (zdroj: autor).



Obr. 4.4.1.4 Ukázka okna Inspector pro objekt odkazující na vybraný *Timeline* (zdroj: autor).

Aby hráč měl možnost rozvíjet konverzaci s průvodcem, byly pomocí metod aktivovány na konci cut-scén tři otázky, na které se hráč mohl průvodce zeptat (Obr. 4.4.1.5). Pro jejich vytvoření bylo nutné mít objekt typu *Canvas*, který obsahoval tři objekty typu *Button*. Jednotlivým objektům typu *Button* byla přiřazena funkce *OnClick()* odkazující na danou metodu ve skriptu a také jim byly obdobně přidány efekty jako v kapitole 4.1.1. Při aktivaci cut-scén byly herní prvky jako minimapa a statistiky hráče deaktivované.

```
public void End_ObrazPrvni()
{
    Cursor.visible = true;
    Cut_SceneDefault.SetActive(false);
    CameraQuestion.SetActive(true);
    Options.SetActive(true);
}
```



Obr. 4.4.1.5 Ukázka aktivace otázek (zdroj: autor).

Při probíhající cut-scéně se hráč nemohl pohybovat, nicméně pro prohlídku Zdikova paláce bylo vhodné, aby tuhle možnost hráč měl. Proto jsou cut-scény využívány pouze na začátku. Při prohlížení paláce, i když jsou hráči sdělovány zajímavosti a informace o dané místnosti, se může libovolně pohybovat. Pro tento účel byly vytvořeny objekty na detekci průvodce. Po jejich aktivaci se spustí titulky vytvořené v samotném skriptu, a ne v grafickém rozhraní Unity.

```
void OnTriggerEnter(Collider NPC)
{
    if (NPC.gameObject.tag == "PLace")
    {
        if (point == 0)
        {
            point = point + 1;
            Anim.SetBool("walking", false);
            StartCoroutine("Dialogue");
            Look = true;
        }
    }
}

IEnumerator Dialogue()
{
    yield return new WaitForSeconds(0.1f);
    SubText.GetComponent<Text>().text = "";
    SubText.GetComponent<Text>().text = "Prosím vstupte první.";
    yield return new WaitForSeconds(3f);
    SubText.GetComponent<Text>().text = "";
}
}
```



Obr. 4.4.1.6 Aktivace titulků v místnosti ve Zdikově paláci (zdroj: autor).

## 4.4.2 Gamifikace

Pro celistvost gamifikovaného průvodce bylo potřeba vymyslet a vytvořit řadu úkolů, které hráč plní na základě předchozí edukaci o Zdikově paláci. Pro větší atraktivnost byla vytvořena metoda, která omezuje běhání hráče. Aby metoda správně fungovala, byl vytvořen skript, který hlídá stav energie hráče.

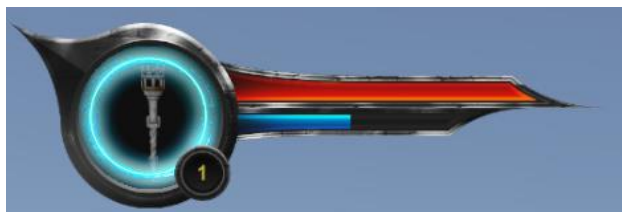
```
if (ManaSprint.fillAmount < 0.1)
{
    CanJump = false;
    CanRun = false;
    StartCoroutine("Wait");
}
}
```

Uživatel energii postavy ovlivňuje na základě svého rozhodnutí. Jestliže bude dlouho běhat či skákat, postava se vyčerpá. Po uplynutí deseti sekund bude uživatel moci znova využívat skok či běh.

```
{
  yield return new WaitForSeconds(10f);
  CanRun = true;
  CanJump = true;
}
```

Uživatel vidí stav energie v herním okně znázorněnou pomocí modrého slideru (Obr. 4.4.2.1). Energie dynamicky ubývá či přibývá podle toho, jak se daný uživatel chová.

```
if ((GameObject.FindWithTag("Manager").GetComponent<CheckSprint>().checkSprint == true)
    && (Input.GetKey(KeyCode.LeftShift)) && CanRun == true)
{
  ManaSprint.fillAmount -= 1.0f / waitTime * Time.deltaTime;
}
else if ((GameObject.FindWithTag("Manager").GetComponent<CheckSprint>().checkSprint ==
false))
{
  ManaSprint.fillAmount -= -1.0f / waitTime * Time.deltaTime;
}
else
{
  ManaSprint.fillAmount -= -1.0f / waitTime * Time.deltaTime;
}
```



Obr. 4.4.2.1 Statistika (zdroj: autor).

Dále byl vytvořen systém pro sbírání a odhazování věcí. Aby postava dokázala vzít předmět, bylo potřeba vytvořit objekt, který bude detektovat jednotlivé předměty. Daný objekt, sloužící pro detekci předmětů, obsahuje prázdné objekty s předem danou pozicí jednotlivých předmětů. Vytvořený skript pracuje právě s prázdnými objekty. Na základě toho ví, jaký konkrétní předmět přetransformovat k hráči. Aby mohl být předmět zdvižen, musí být vypnuta jeho „fyzika“.

```
if (GameObject.FindWithTag("OnGroundTorch"))
{
  Destroy(this.gameObject.GetComponent<Rigidbody>());
  this.gameObject.transform.SetParent(myParentObject.transform);
  this.gameObject.transform.position = NewTorch.transform.position;
  this.gameObject.transform.rotation = NewTorch.transform.rotation;
  this.gameObject.GetComponent<Collider>().isTrigger = false;
}
```

Jednotlivé předměty jsou rozděleny do skupin pomocí *Tagu* a to zda se nacházejí u hráče, na zemi nebo na původním místě. Pro odhazování předmětů je nutné fyzikální vlastnosti u daného předmětu aktivovat.

```

if (GameObject.FindWithTag("InHandTorch"))
{
    Anim.GetComponent<Animator>().SetBool("EquipTorch", false);
    GameObject.FindWithTag("InHandTorch").transform.parent = null;
    GameObject.FindWithTag("InHandTorch").AddComponent<Rigidbody>();
}

```

Jestliže hráč bude mít prázdné ruce, bude moci zdvihnout předměty ze země nebo z výšky. Podle zvoleného input systému se přehrají dané animace.

```

GameObject.FindWithTag("LeftHand").GetComponent<BoxCollider>().enabled = true;
Anim.SetBool("IsGrabbingTorch", true);
Anim.Play("GrabTorch");
Anim.SetBool("IsGrabbingTorch", false);

```



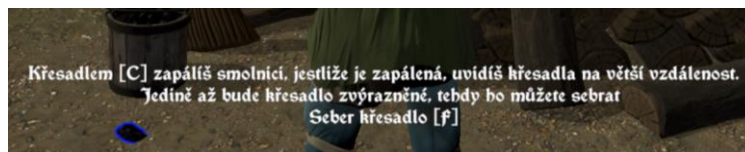
Obr. 4.4.2.2 Zdvih předmětu ze země (zdroj: autor).

U předmětu louče byl naprogramovaný časovač, který je znázorněn na (Obr. 4.4.2.1). Časovač znázorňuje dobu, po kterou louč hoří. Hráč pomocí křesadel, jejichž stav je také znázorněn ve statistikách hráče, dokáže louč znova zapálit. Pro sběr křesadel byl napsán skript, který pomáhá hráči dané křesadlo najít. Jestliže se hráč nachází v blízkosti křesadel (Obr. 4.4.2.3) jsou dané křesadla zvýrazněny a mohou být sebrány pomocí klávesnice F. Dále byla vytvořena metoda, jež znásobuje efekt zvýraznění, pokud louč hoří.

```

else if (dist < 2)
{
    InZoneFlint = true;
    this.Parents.GetComponent<Outline>().enabled = true;
    this.Parents.gameObject.tag = "FlintZone";
}

```



Obr. 4.4.2.3 Sběr křesadel (zdroj: autor).

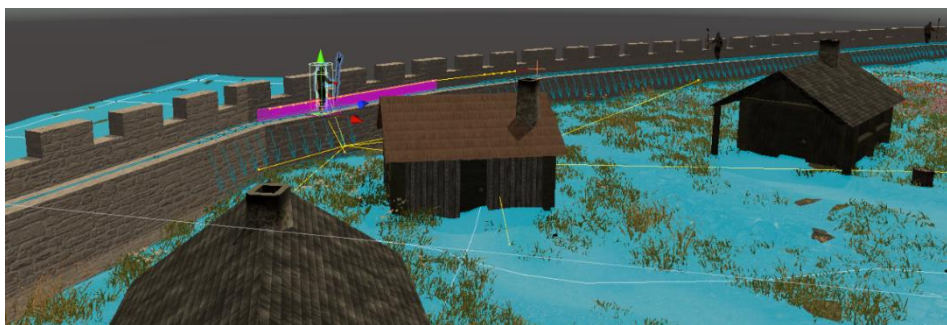
Rozmístění křesadel je znázorněno pomocí plánek, jenž byl vytvořen v grafickém editoru Gimp a Inkscape z 3D modelů minimap. Pro lepší čitelnost byla vytvořena metoda, která změní pohled kamery.

```

if ((Input.GetKeyDown(KeyCode.LeftControl)) && (FP_Camera == false))
{
    rightOffset = -0.74f;
    defaultDistance = 0.6f;
    height = 1.8f;
}

```

Téměř všechny NPC v hlavním příběhu využívají umělou inteligenci (AI), na základě které se mohou pohybovat. Pro správné fungování je potřeba daným postavám přiřadit komponent *Nav Mesh Agent* a *Rigidbody*. Dalším krokem je vybrání objektů, po kterých se dané postavy mohou pohybovat. Zejména to je digitální model reliéfu a vybrané části digitálního modelu povrchu (hradby, schody atd...). Nad zvolenými objekty se vytvořila analýza sklonu povrchu (Obr. 4.4.2.4). Současně objekty digitálního modelu povrchu slouží pro počítání samotné výšky nad digitálním modelem reliéfu. Jelikož pohyb umělé inteligence funguje na principu hledání nejbližší možné trasy z bodu A do bodu B, byla přidána možnost seskočení. Pokud je splněna podmínka a modely digitálního povrchu objektu jsou do určité výšky nad terénem, poté je možnost si zkrátit cestu seskočením. Tato podmínka je evokována modrými šipkami (Obr. 4.4.2.4).



Obr. 4.4.2.4 Analýza sklonu povrchu pro NPC nad DMR (zdroj: autor).

V každých vchodových dveřích byly vytvořeny objekty, které detekují, zda hráč neukradl věci z paláce. V momentě, kdy je podmínka splněna, spustí se vytvořená metoda, která navede strážce na hradbách pomocí umělé inteligence za hráčem. Jestliže je hráč chycen, aktivuje se vytvořený objekt typu *Canvas*, jenž demonstruje vyskakovací pop-up okno a hra se ukončí. Pomocí funkce *StartCoroutine()*, je za určitý čas vrácen do defaultního menu.

```

theAgent.SetDestination(player.position);
Anim.SetBool("RunToDaniel", true);
float dist = Vector3.Distance(transform.position, player.position);
if (dist < 1)
{
    StartCoroutine("Wait");
}

```



Obr. 4.4.2.5 Pop-up okno (zdroj: autor).



Pro větší dynamiku gamifikace byl vytvořen a naprogramován časovač, který je znázorněn červeným objektem typu slider (Obr. 4.4.2.1). Časovač slouží jako maximální doba, do kdy hráč musí splnit úkoly. Časovač byl nastaven na deset minut a každou sekundu dochází k jeho grafickému úbytku.

```
_time += Time.deltaTime;
while (_time >= waitTime)
{
    TorchLife();
    _time -= waitTime;
}
```

Celkem byly vytvořeny tři úkoly. Pro jejich správné vyřešení hráč využívá znalosti získané studiem plánek a komunikaci s průvodcem. Hráč musí donést caltu (středověký koláč), zapálenou smolnici a deset křesadel. Zadávatel všech úkolů je postava kováře (Obr. 4.4.2.6). Pro aktivaci a odevzdávání úkolů musí být hráč v určité vzdálenosti od kováře. K vzájemné komunikaci se používá vytvořený systém dialogů viz podkapitola 4.4.1. Současně aplikace byla naprogramována tak, že pokud hráč nesplní první úkol, nemůže dále pokračovat v příběhu.

```
Float dist = Vector3.Distance(transform.position, players.position);
if (dist < 7)
{
    BLText.SetActive(true);
    if (PromLuvSKovarem == true)
    {
        SubText.GetComponent<Text>().text = „Promluvit si s kovářem\n“ +
        „[F]“;
        if (Input.GetKeyDown(KeyCode.F))
        {
            PromLuvSKovarem = false;
            StartCoroutine(„První“);
        }
    }
}
```



Obr. 4.4.2.6 Zadání úkolů od kováře (zdroj: autor).

Pokud hráč úkoly nesplní správně nebo v daném časovém rámci, byly vytvořeny metody, které hlavní příběh ukončí a vrátí uživatele do hlavního menu. Současně na uživatele vyskočí pop-up okno, které bylo vytvořeno pomocí objektu typu *Canvas*. Obdobně na uživatele vyskočí pop-up okno po dokončení všech správných úkolů. Zároveň byl vytvořen postup, který využívá časovač a hráči se tak zobrazí za jak dlouho celou hru dokončil.

```

yield return new WaitForSeconds(7f);
SceneManager.LoadScene("Menu");

//Po splnění všech úkolů se aktivuje objekt, který slouží pro ukončení gamifikovaného
průvodce

SubText.GetComponent<Text>().text = "Zde máte klíč, ale že to bylo o fous!\n" +
    "Vzít si klíč\n" +
    "[F]";
if (Input.GetKeyDown(KeyCode.F))
{
    SubText.GetComponent<Text>().text = "";
    Klic.SetActive(true);
}

//Metoda pro zaznamenávání času
else if ((yes == false) && (OO.GetComponent<Hildebert_AI>().StartTime == true))
{
    _time += Time.deltaTime;
    while (_time >= waitTime)
    {
        TimeKonecHry();
        _time -= waitTime;
    }
}

```

Pro kontrolu aktivních úkolů byla vytvořena metoda, jež aktivuje dané objekty typu *Text*, které vypisují hráči aktuálně rozdělaný úkol.

```

if ((Input.GetKeyDown(KeyCode.LeftAlt)) &&
(BLaaack.GetComponent<BlackSmith>().firstcontact == true) && (hide == false))
{
    hide = true;
    Calta1.SetActive(true);
}

```

Na závěr bylo vytvořeno uživatelské rozhraní. Pro tvorbu uživatelského rozhraní byl použit stejný postup, jako v podkapitole 4.1.1. Byla zde přidána možnost výběru grafického nastavení v podobě rozlišení textur, vykreslování stínů a podobně. Vytvořené metody vždy odkazují na předpřipravené grafické nastavení, které se předem nastavilo v herním enginu v okně *Quality*.

```

public void Ultra()
{
    QualitySettings.SetQualityLevel(4);
}

```

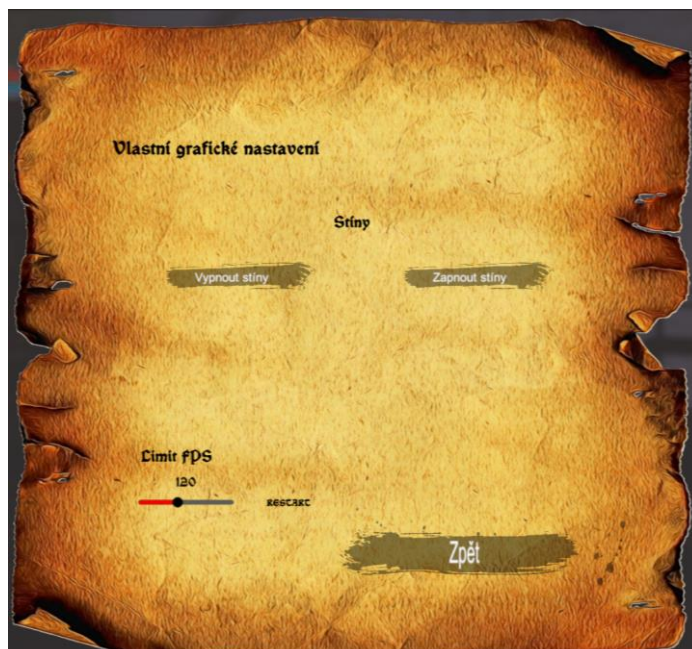
Také zde byly naprogramované metody pro vypnutí / zapnutí stínů nebo zamknutí FPS (Frames per second) (Obr. 4.4.2.7). Pro tvorbu zamknutí FPS byl vytvořen objekt typu *Slider*, na který byla připojena funkce *OnValueChanged()*. Nakonec byla celá hra optimalizována pomocí systému WYSIWYP. Optimalizace se vytvořila v okně *Occlusion*, kde se nastavuje například velikost nejmenšího objektu, který dokáže zakrýt objekty za sebou nebo nejmenší mezera mezi jednotlivými geometriemi objektu.

```

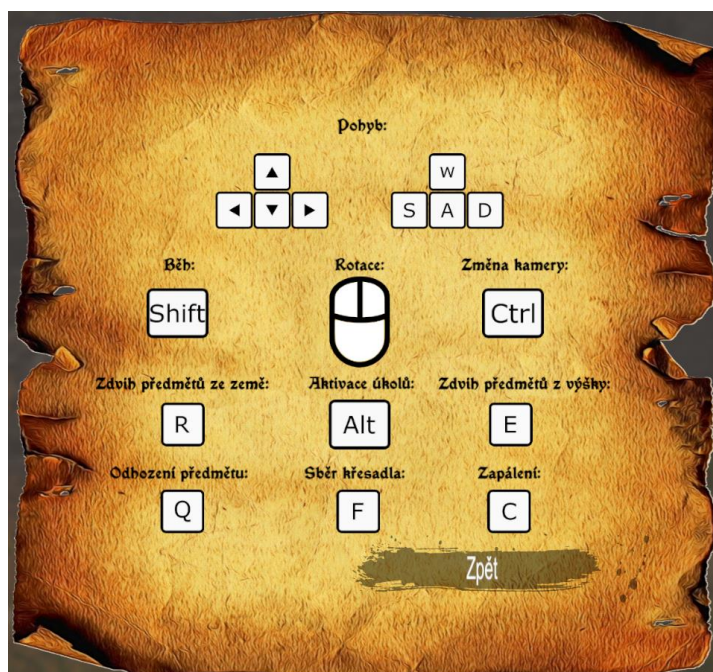
Application.targetFrameRate = 120;
FPSSlider.onValueChanged.AddListener((v) =>
{
    SliderText.text = v.ToString("0");
});

```

Uživatelské rozhraní neboli menu se aktivuje na základě napsaného skriptu. Jak již bylo zmíněno, byl vytvořen objekt, který detekuje aktivaci uživatelského rozhraní a na základě toho aktivuje / deaktivuje kurzor myši. Hlavní příběh v uživatelském menu také obsahuje vytvořenou nápovědu pro ovládání (Obr. 4.4.2.8). Aplikace obsahuje hudbu, jež se přehrává ve smyčce pro její správnou funkcionalitu musely být přidány komponenty *Audio Listener* a *Audio Source*.



Obr. 4.4.2.7 Vlastní grafické ovládání (zdroj: autor).



Obr. 4.4.2.8. Nápověda pro ovládání (zdroj: autor).



## 4.5 Průlet

Pro možnost získání prostorového vjemu a nastudování exteriéru a interiéru před samotnou hrou byl vytvořen režim průlet, který slouží pro volný průlet nebo prohlídku Zdikova Paláce z prvního pohledu osoby (Obr. 4.5.1). Scéna byla vytvořena také z důvodu větší rozmanitosti a získání lepší představivosti o Zdikově paláci. Uživatele neprovází žádné dialogy, cut-scény ani scénář. Uživatel se pohybuje volně po celé délce území, jedinému omezení se věnuje podkapitola 4.3.2. Uživatel má navíc možnost si vybrat atmosférické prohlídky viz podkapitola 4.2, ve kterých si bude chtít prohlédnout Zdikův palác. Nejdříve bylo nutné vytvořit skript pro ovládání kamery, který je jedním z hlavních prvků v celé scéně.

```
mouseDelta = Vector2.Scale(mouseDelta, new Vector2(sensitivity.x * smooth.x,  
    sensitivity.y * smooth.y));  
MouseMotor.x = Mathf.Lerp(MouseMotor.x, mouseDelta.x, 1f / smooth.x);  
MouseMotor.y = Mathf.Lerp(MouseMotor.y, mouseDelta.y, 1f / smooth.y);
```



Obr. 4.5.1 Pohled v Prohlídce na Zdikův palác (zdroj: autor).

Dále zde bylo pro dynamičtější ovládání kamery připsána metoda, která pohyb zrychluje nebo naopak zpomaluje.

```
if (Input.GetKey(KeyCode.LeftShift) || Input.GetKey(KeyCode.RightShift))  
{  
    rb.velocity = Vector3.zero;  
    rb.velocity += transform.forward * (DefaultSpeed * UpMotor) *  
        Input.GetAxis("Vertical") * Time.deltaTime;  
    rb.velocity += transform.right * (DefaultSpeed * UpMotor) *  
        Input.GetAxis("Horizontal") * Time.deltaTime;  
}
```

Obdobně, jako v hlavním v příběhu, i zde je kurzor skrytý pomocí skriptu a aktivuje se po vyvolání stiskem klávesy Escape. Menu je totožné až na jednu změnu, kdy si uživatel může měnit hodnoty nastavení kamery. V nastavení kamery bylo přidáno možnost si libovolně upravit sensitivitu, zrychlení nebo zpomalení. Pro jejich grafické znázornění byly vytvořeny tři objekty typu *Slider*, na které byla připojena funkce *OnValueChanged()*, která ze skriptu pro ovládání kamery vytahuje jednotlivé metody pro změnu sensitivity, zrychlení, nebo zpomalení. Taktéž byly vytvořeny metody, které vrátí změněné hodnoty do původního stavu. Na závěr, obdobně jako v minulé podkapitole byl v softwaru Inkscape vytvořený návod pro ovládání. Opět zde hraje v pozadí hudba, která je nastavená ve smyčce a spustí se po změně defaultního menu do scény *Průlet*.

## 5 TESTOVÁNÍ PŘENOSU MODELU DO VR

Závěrem práce a dílčím úkolem bylo testování přenosu modelu do virtuální reality. Pro testování přenosu modelu do virtuální reality bylo nutné vytvořit nový projekt v softwaru Unity. Zde byl stáhnuta a vybrán takový *Template*, který v projektu defaultně obsahoval stáhnuté a nakonfigurované balíčky potřebné pro virtuální realitu. Přesto v okně *Project Settings* bylo manuálně nastaveno *IL2CPP* (Intermediate Language to C-plus-plus). Dané nastavení podporuje chod aplikace na širší škále platformem. Pro co nejvíce realistické zachování barev bylo v okně *Rendering* nastaveno *Color Space* na hodnotu *Linear*. K tomu bylo nutné nastavit v okně *Quality* *4x multisample anti-aliasing* (MSAA), jenž kompenzuje stereo vykreslování, které snižuje horizontální rozlišení o 50 %. Následně se software Unity musel připojit a spárovat s brýlemi Oculus Quest. Pro správnou funkcionalitu byl vytvořen objekt typu *camera*, který se umístil doprostřed scény. Tenhle objekt sloužil jako prostředník mezi brýlemi a softwarem Unity. Díky objektu typu *camera* bylo možné přenést obraz ze softwaru Unity do brýlí Oculus Quest.

Výsledkem testování bylo, že kompozice modelu Zdikova paláce se podařilo bez problému přenést do virtuální reality. Díky testování přenosu vybrané scény do virtuální reality bylo zjištěno a potvrzeno, že virtuální realita oproti desktopovému řešení zaostává po grafické stránce. Daná scéna byla ve virtuální realitě vykreslena v nižším rozlišení a bez viditelných grafických detailů.

Nicméně původní aplikace byla vytvořena pro desktopovou platformu využívající kvalitnější hardware pro vykreslování grafické stránky i pro plynulejší chod aplikace. Výslednou aplikaci by bylo možné převést do virtuální reality. Avšak bylo by nutné, velmi snížit grafické nároky a to zejména u dynamického střídání dne a noci či u jiných graficky náročných prvků, jako je vykreslování stínů či pohyb více postav v jednom záběru. Zároveň převod desktopové aplikace do virtuální reality by byl časově náročný, z důvodu nutnosti vytvoření nového *input* systému a přenesení celého příběhu do první osoby.



Obr. 5.1 Testování přenosu modelu do VR (zdroj: autor).

## 6 VÝSLEDKY

Hlavním výsledkem práce je desktopová aplikace, která slouží jako gamifikovaný průvodce Václavským návrším v Olomouci ve 12. století. Její konečná velikost je 480 MB, aktivně využívá přes 5000 herních objektů a obsahuje dva herní módy.

Po zapnutí aplikace je uživatel přenesen do hlavního menu, jehož postup zpracování je naznačen v kapitole 4.1.1. Hlavní menu slouží jako výchozí bod, ze kterého si uživatel volí následující cesty dle svého rozhodnutí. Pro lepší orientaci a představivost byly vytvořeny plány Zdikova paláce. Ještě před tím, než si uživatel vybere, jaký příběh chce hrát, má tedy možnost si prohlédnout Zdikův palác pomocí plánů. Plány zobrazují jednotlivá patra s podrobnými popisy místností. Plány vznikly z tvorby jednotlivých minimap v softwaru Sketchup a následným importem do grafického editoru Inkscape. V hlavním menu taktéž nechybí tiráž a možnost ukončit hru.

Desktopová aplikace obsahuje dva příběhy a to hlavní a vedlejší příběh. Vedlejší příběh byl vytvořen pro možnosti prohlídky Zdikova paláce, nezávisle na napsaném scénáři. Vedlejší příběh obsahuje volbu atmosférických podmínek. Uživatel má možnost volby zobrazení Zdikova paláce ve slunečném, deštivém nebo zimním dni. Pro jednotlivé změny atmosférických podmínek bylo nutné vytvořit odlišné materiály evokující vždy nastavení daného dne, navíc u zimního dne bylo nutné nově otexturovat digitální model reliéfu, zredukovat na něm vegetaci a změnit pomocí upraveného shaderu od World of Zero textur na jednotlivých stavebách. Podrobnější postup tvorby změny atmosférických podmínkách je vysvětlen v podkapitole 4.2 a postup selekce jednotlivých dnů v podkapitole 4.1.2. Vedlejší příběh také obsahuje objekty pro detekci hráče, nacházející se v Zdikově paláci tak, aby při vstupu do místností se deaktivovaly efekty evokující padající déšť či sníh viz podkapitola 4.3.2. Vedlejší příběh disponuje volným pohybem, tedy pohledem z první osoby či kamery viz podkapitola 4.5. Uživatel si může díky vytvořenému uživatelskému rozhraní nastavit sensitivitu myši, zrychlení či zpomalené kamery. Dále po aktivaci menu klávesnicí Esc, má možnost se vrátit do defaultního menu či úplně ukončit hru.

Hlavní příběh obsahuje prvky gamifikace i průvodce na rozdíl od vedlejšího příběhu, který slouží jen jako prohlídka. Současně hráč hraje z pohledu třetí osoby s možností změny kamery pomocí klávesnice Ctrl. Pro větší atraktivnost byla přidána všem postavám základní animace a fyzika, jež se provedla v softwarech Zbrush, Mixamo a Unity. Hned na začátku je uživatel vtáhnut do příběhu pomocí cut-scén obsahující dynamickou kameru. Na základě cut-scén je uživateli vysvětleno za koho hraje, kde se nachází a co ho čeká. Díky vytvořeným dialogům jež se věnuje podkapitola 4.4.1, hráč ještě před vstupem do paláce může rozvíjet konverzaci s průvodcem.

Celkem byly vytvořeny tři možnosti pro rozšíření konverzace s průvodcem. První dvě slouží čistě pro rozvinutí konverzace, kdy vždy po jejich dokončení se uživateli opětovně zobrazí tři možnosti volby. Po zvolení a dokončení poslední volby se aktivuje umělá inteligence u průvodce, zatímco u strážných je aktivní po celou dobu. Průvodce využívá umělou inteligenci pro pohyb v paláci po předem definovaných značkách. Na základě ní se průvodce postaví ke vstupu do paláce, vybidne hráče aby vstoupil první a celou dobu se dynamicky otáčí podle toho, kde se zrovna hráč nachází. Jestliže hráč vstoupí do paláce, průvodce začne postupně vyprávět pomocí titulků v monologu různé zajímavosti a informace o dané místnosti v níž se právě nacházejí. Po dokončení prohlídky Zdikova paláce je nabídnuta volba spuštění gamifikace viz podkapitola 4.4.2.

Pokud se hráč nerozhodne aktivovat gamifikaci, má možnost dále procházet Zdikův palác a Václavské návrší avšak v určitém omezení. Nicméně kdykoliv je uživateli

umožněno opětovně aktivovat gamifikaci, jež zaručuje úplnou volnost po celém Václavském návrší. Po její zpětné nebo prvotní aktivaci je hráči vysvětleno v podobě titulků, co se od něj očekává. Následuje aktivace odpočítávání času, který evokuje dobu do které hráč musí splnit úkoly. Zbývající čas je znázorněn ve statistikách v podobě červeného slideru vlevo nahoře. Současně byla naprogramována metoda, která omezuje hráči v kuse běhat. Stav energie hráče je obdobně zobrazena pomocí slideru ve statistikách tmavě modrou barvou. Celkem byly vytvořeny tři úkoly, pro jejich správné vyřešení je nutné si odnést základní znalosti z výkladu průvodce či mít předešlou znalost Zdíkovy paláce a tehdejší doby. Právě probíhající úkol je možné si zkontrolovat pomocí klávesnice Alt. První úkol je zaměřen na dobovou kuchyni, druhý úkol na nástroje z 12. století a poslední úkol má za cíl uživatele provést veškerými místnostmi Zdíkovy paláce. Aplikace byla naprogramována tak, že postup ve hře je uživateli umožněn pouze tehdy, pokud plní jednotlivé úkoly za sebou. Není možné aktivovat zbývající úkoly, dokud hráč nemá splněný některý z předešlých úkolů.

Pro plnění úkolů byl vytvořen systém pro interakci mezi hráčem a definovanými předměty. Byly přidány animace, které evokují odhození předmětu, sběr předmětu z výšky, sběr předmětu ze země či zapálení louče. Dále byl vytvořený systém pro sběr křesadel, jež se využívá v úkolu dva a tři. Pro jejich lehčí nalezení byly vytvořeny dva plánky s jejich přibližnou polohou, nacházející se ve slavnostní síni. Také byl vytvořen časovač pro každou louč, který se aktivuje po zapálení louče křesadlem a udává tak její životnost, do kdy bude hořet. Zapálená louč má navíc tu funkci, že uživatel vidí ztracená křesadla na větší vzdálenost. Uživatele celou dobu provází titulky, které ho navádí, co má dělat. Pro špatný postup či nedostatek času je hráči zastavena hra. Uživateli vyskočí upozorňující pop-up okno, kdy se mu vypíše důvod, proč byl jeho postup ve hře zastaven a proč musí hrát od začátku. Jestliže uživatel plní správně všechny kroky, je odměněn v podobě klíče, který mu vykoval kovář. Po odemknutí truhly, což je cílem celé gamifikace, se uživateli zastaví čas a vypíše se pomocí pop-up okna jeho celkový čas od aktivace gamifikace. Následně je v obou případech přenesen do defaultního menu.

Aplikace dále obsahuje vytvořenou minimapu, která se dynamicky mění v závislosti na poloze hráče. Celkem byly vytvořeny tři minimapy a to pro dvě patra Zdíkovy paláce a pro celé Václavské návrší. Vybrané objekty vykresluje kamera s ortografickým nastavením z důvodu zkrácení. Jejich postup tvorby je popsán v podkapitole 4.3.2. Dále byly vytvořeny v softwaru Sketchup urbanické modely a v softwaru Zbrush organické modely jak pro okolí, tak interiér Zdíkovy paláce viz podkapitola 4.3.1. Následně byly jednotlivé 3D modely rozmístěny v herním enginu Unity. Na závěr pomocí softwaru CrazyBump byly vytvořeny chybějící mapy dané textury a v herním enginu jim byly nastaveny příslušné parametry. Hlavní příběh disponuje dynamickou změnou denní doby, kdy se neustále střídá den a noc. Pro jejich správné fungování byly vytvořeny dva objekty typu Light, které neustále rotují kolem osy X v rozdílu 180°. Každý objekt evokuje grafické nastavení, kdy na základě polohy či času bylo toto nastavení proloženo vytvořenou matematickou křivkou, jež slouží jako vstupní data pro skript. Aplikace také obsahuje základní uživatelské rozhraní v podobě nastavení. Hráč podle svého výkonu počítače si může nastavit přednastavenou grafickou náročnost (v podobě rozlišení textur, vzdálenosti jednotlivých LOD (level of detail), vykreslování stínů a podobně), nebo si může samostatně zapnout / vypnout stíny či uzamknout FPS. Celá aplikace je optimalizována podle systému WYSIWYP. Uživateli se tedy vykresluje pouze to, na co se právě dívá. Obdobně, jako u vedlejšího příběhu, doprovází uživatele po celou dobu na pozadí hudba a má možnost se kdykoliv vrátit do defaultního menu či úplně ukončit hru. Na závěr byl úspěšně přenesen model do virtuální reality a to s využitím brýlí Oculus Quest

## 7 DISKUZE

Z důvodu obří komplexnosti a náročnosti, jež obsahovalo vytvoření aplikace, existuje stále mnoho dílčích úkolů, které je možné upravit, doladit, vylepšit, anebo i vytvořit. Problémy či úkoly, jež budou dále v textu jmenované, je možné při zájmu široké veřejnosti vyřešit či zdokonalit. Jedná se zejména o *save & load* systém. Tedy systém na základě něhož by hráč mohl uložit rozdělaný postup ve hře a aplikaci ukončit s klidným vědomím, že po jejím znovu zapnutí se vrátí do stavu, v jakém hra byla, než ji vypnul. K tomuto problému se váže také ukládání mezivýsledku. Současná verze aplikace nedisponuje možností si zálohovat jednotlivé pokroky ve hře. Jakmile uživatel plní kroky nevhodným způsobem, je vrácen zpět do hlavního menu a není mu dovoleno pokračovat od momentu, kdy správně vyřešil poslední úkol, tak jak to bývá u běžných her. Zároveň by bylo příhodné vytvořit funkcionalitu pro zaznamenávání a zapisování časů do databáze. Převedení desktopové aplikace do virtuální reality by taktéž mohlo splnit i požadavky náročnějších uživatelů. Testování přenosu modelu do virtuální reality proběhlo úspěšně, nicméně pro celistvost a funkčnost by bylo potřeba naprogramovat a změnit celý *input systém*, jelikož aplikace byla programována jako desktopová aplikace. Taktéž by bylo nutné z grafické náročnosti omezit nejrůznější efekty a jiné grafické prvky, které by mohly způsobovat problémy.

V rámci zatraktivnění aplikace by bylo vhodné nadabovat jednotlivé texty, přidat možnost výběru cizího jazyka a současně vymyslet způsob, kterým by se dané cut-scény a titulky daly přeskakovat. U jednotlivých postav by bylo vhodné dodělat *lip-sync* systém. Funkcionalita *lip-sync* systému slouží pro sladění pohybů rtů mluvicí postavy. Tedy evokuje u uživatelů pocit, že jednotlivé postavy doopravdy mezi sebou vedou dialogy. V rámci cut-scén by bylo na místě dodělat *fade out* efekty mezi jednotlivými střihy tak, aby když dojde k náhlému přeskocení z jedné kamery na druhou, stihla se vygenerovat a vyrenderovat ta část scény z pohledu změněné kamery. Pro lepší uživatelské rozhraní by bylo potřeba vytvořit mezi jednotlivými scénami informativní hodnotu, z jak velké části je daná scéna načtená. Jednou z dalších možností, kam posunout vývoj a upgrade aplikace by bylo směrem k vytvoření většího obsahu jednotlivých příběhů. V hlavním příběhu by mohl průvodce déle hovořit o zajímavostech a historii nejen v kontextu k dané místnosti. Také by gamifikovaný průvodce mohl obsahovat více propracovanějších úkolů, které by současně byly herně zajímavé (proplížit se někam, odposlouchávat někoho, vyřešit hádanky, rébusy a podobně), avšak zároveň by byly i více zaměřené na poznávání tehdejší doby. Taktéž by bylo vhodné v jednotlivých scénách dodělat patřičné zvukové efekty, jež by evokovaly daný jev (zvuk chůze po trávě, po dřevě, po dlažbě, zpěv kanovníků, šumění větru, padání kapek, hřmění hromu a podobně). Hlavní příběh, by podobně jako vedlejší, mohl obsahovat seznam s více postavami, které by po jejich zvolení vždy měly odlišné postavení v aplikaci, co se týče postupu řešení příběhu. Ve vedlejším příběhu by mohly být vytvořeny archivy, popisující způsob život městských, venkovských či urozenějších lidí ve 12. století. Prohlídka by mohla obsahovat naskenované 3D modely dochovaných artefaktů, listin, či jiných předmětů s historickou hodnotou, které bývají zpřístupněny pouze ve vybraných expozicích po určitý čas. Dále by bylo vhodné vymodelovat interiéry jednotlivých chalup, které se nachází v okolí Zdikova paláce a zakomponovat je do příběhů, stejně tak vytvořit více postav, které by vždy měly jinou úlohu a vliv na průchod hlavního příběhu. Pro vylepšení grafické části by bylo vhodné zasadit Zdikův palác přímo na Václavské návrší s dobovými částmi města Olomouce. Tedy nahradit jednotlivé *HDRI maps*, které mají pouze enviromentální hodnotu, buď nově vytvořenou *HDRI map*, jež by zachycovala tehdejší pohledy na město Olomouc od Zdikova

paláce, nebo přímo vytvořené a otexturované 3D modely okolních budov a terénu. Také by bylo zajímavé pomocí *REST API* získávat reálná data o počasí ve městě Olomouce. Po zapnutí aplikace by se automaticky a dynamicky vykreslovalo počasí ve hře v závislosti na počasí v reálném čase v Olomouci. Grafická změna by zasáhla i optimalizaci jednotlivých kamer, které jsou využívány v podobě třetí a první osoby. V hlavním příběhu se stává z důvodu *Occlusion*, že v určitých místech kamera špatně vykresluje dané 3D objekty. Tedy vykresluje oblohu či terén a ignoruje 3D objekty, které je překrývají. V scéně *Průlet* by bylo možné optimalizovat způsob, kterým je evokována atmosférická změna. Při vstupu do Zdikova paláce se může stát, že efekt deště či sněhu propadá skrz 3D modely, zároveň jakmile je uživatel uvnitř a podívá se z okna ven, tak jmenované efekty nejsou aktivní, což nepůsobí kladným dojmem. Dynamická změna denní doby působí hůře, než statická obloha ve vedlejšímu příběhu. Je to z toho důvodu, že vedlejší příběh využívá statické *HDRI map*, které se pomocí metody otáčejí a evokují tak pohyb atmosféry. Avšak v hlavním příběhu je pro dynamickou změnu denní doby využívána metoda *Procedural Skybox*. Tedy metoda, která pracuje s defaultním grafickým nastavením, které je možné měnit pomocí napsaných metod. Tato metoda nevyžaduje a ani neumožňuje práci s jednotlivými texturami evokující atmosféru. Řešením by bylo spojit jednotlivé metody dohromady. Unity 3D neumožňuje v defaultním nastavení kombinovat *Procedural Skybox* a *HDRI map*. Není tedy možné spojit vytvořenou *HDRI map* s objektem typu *Light*. Dané objekty typu *Light* by byly překryty *HDRI map*. Jedním z dalších řešení by bylo nutné jít cestou, kdy se vytvoří 2D objekty evokující Slunce a Měsíc a sjednotit jejich pohyb na objekty typu *Light*. Avšak zároveň by byla potřeba vymyslet systém, který vizuálně bude přímo dané 2D objekty dynamicky měnit. Dalším výrazným grafickým nedostatkem je vykreslování stínů. Pouze vybrané objekty z důvodu plynulejšího chodu aplikace vytváří svůj stín. Menším nebo méně významným herním objektům bylo přiřazeno takové nastavení, které znemožňuje vykreslovat stíny.

Z již zmíněné velké komplexnosti a náročnosti tvorby samotné aplikace, nebylo možné doladit jednotlivé systémy, jež jsou implementovány ve hře. Většina chyb, nedostatků či možných rozšíření byla již zmíněna. Avšak existují nedostatky, které mají přímý vliv na vizuální dojem hry. Jedním z příkladů je systém interakce mezi hlavní postavou a předměty, kdy animace nepůsobí mnohdy realisticky. Často během aktivace animací dochází ke kolizím mezi hráčem a jednotnými objekty. Systém pro sbírání věcí také není zcela optimalizován. Jedná se o situaci, kdy se uživatel musí přímo dotknout předmětu, který potřebuje vzít. Tento problém by šel vyřešit stejným postupem, jako byl vyřešen sběr křesadel. Uživatel by se musel nacházet v požadované vzdálenosti od předmětu, aby ho mohl sebrat bez přehrání animace. Problém nastává v momentě, kdy by hráč chtěl sebrat z dvou různých zvýrazněných předmětů jeden konkrétní. Zmiňovaný problém se dá zlepšit zvětšením zóny, která slouží pro detekci jednotlivých předmětů. V tomto případě by přehrávané animace nedávaly smysl. Hráč by sebral předmět bez toho, aniž by se ho dotkl. Při řešení bakalářské práce bylo naraženo i na nevyzpytatelné chování NPC, využívající umělou inteligenci. Strážci mají občas tendenci seskakovat z hradeb dříve, než hráč udělá chybu a dojde k situaci, že ho strážci zatknou a znemožní mu další postup ve hře. Dalším problémem je, že pokud se hráč nachází v blízkosti dvou a více křesadel, daná křesadla jsou tedy zvýrazněna naráz, vždy se po jejich sesbírání připočte pouze jedno. Zároveň se v aplikaci nachází mnoho částí, které jsou mnohdy obtížné či úplně nevyřešitelné z důvodu omezení samostatného softwaru. Příkladem je toho již zmiňované vykreslování stínů, nebo slabší grafické zpracování oproti ostatním hernímu enginu.

## 8 ZÁVĚR

Cílem bakalářské práce bylo vytvořit gamifikovaného průvodce vybraným historickým objektem pomocí desktopové aplikace. Pro celistvost aplikace bylo nutné vytvořit přívětivé uživatelské rozhraní, zakomponovat gamifikační prvky a ztvárnit změnu atmosférických podmínek. Dílčím úkolem bylo přenést daný historický objekt do virtuální reality. Jako historický objekt bylo vybráno Václavské návrší v Olomouci ve 12. století.

Pro naplnění cílů bylo nejdříve nutné si nastudovat řešenou problematiku, pochopit principy gamifikace a správně je implementovat. Většina práce probíhala v softwaru Unity 3D, kde se vytvářel samotný gamifikovaný průvodce. Gamifikovaný průvodce obsahuje dva příběhy a to hlavní a vedlejší. V hlavním příběhu je uživatel stylizován do postavy otce Daniela. Hra je zobrazena z pozice třetí osoby s možností změny kamery. Uživatele celou dobu provází předem napsaný scénář. Pro větší realističnost a atraktivnost byly jednotlivé postavy animovány. Toho bylo docíleno pomocí webové služby Mixamo, kde se vytvořil rigging systém, který slouží pro čtení vytvořených animací. Aplikace využívá scénáře, který byl napsán panem Miroslavem Kindlem vedoucím odboru starého umění a vedoucím oddělení Arcidiecézní muzea Olomouce. Pro ztvárnění napsaného scénáře bylo potřeba vytvořit cut-scény a naprogramovat systém pro dialogy či samostatné titulky. V hlavním příběhu byl vytvořen průvodce s výkladem, který provází uživatele po jednotlivých místnostech Zdikova paláce. Hlavní příběh disponuje třemi minimapami, které slouží pro lepší orientaci uživatele na Václavském návrší a dynamickým střídáním dne a noci.

Vedlejší příběh byl vytvořen z důvodu získání prostorového vjemu a nastudování exteriéru a interiéru před samotnou hrou. Byl zde vytvořen výběr pro znázornění atmosférických podmínek. A to pro slunečný, deštivý či zimní den. Pro správné fungování bylo potřeba naprogramovat metodu, která aktivuje vždy vybranou simulaci atmosféry. Pro znázornění zimního dne navíc došlo k redukci vegetace a přetexturování vybraných objektů. Uživatelské rozhraní obsahuje v defaultním menu nejen výběr hlavního či vedlejšího příběhu, ale také Zdikův palác znázorněn pomocí plánů, či samotnou tiráž. Hráč si dále může manuálně vypnout / zapnout stíny, či zamknout FPS. Ve vedlejším příběhu byla vytvořena možnost pro nastavení sensitivity myši, zrychlení, či zpomalení kamery. Při 3D modelování byl kladen důraz na okolí Zdikova paláce a jeho interiér. Na závěr byla celá aplikace optimalizována tak, aby se uživateli šetřil výkon počítače. V aplikaci se renderují pouze objekty, na které se uživatel přímo dívá.

Vytvořená desktopová aplikace bude volně zpřístupněna široké veřejnosti na webových stránkách Katedry geoinformatiky v Olomouci a na webu Muzea umění Olomouce. Aplikace bude sloužit jako dobový gamifikovaný průvodce. Uživatelé si budou moci prohlédnout Zdikův palác za dob vlády Vladislava II. Výsledná aplikace může mít potenciál přiblížení tamějšího života ve 12. století a přivedení nových turistů do města Olomouce. Aplikace také může posloužit jako podklad pro tvorbu nových aplikací s tematikou gamifikovaného průvodce vybraným historickým objektem.

S masovým rozvojem turismu a rozvojem nových technologií se čím dál častěji budeme setkávat v galeriích a muzeích s podobnými aplikacemi. Velký potenciál slibují již dnes technologie, jako je virtuální realita či rozšířená realita. Pro galerie a muzea bude čím dál tím atraktivnější a smysluplnější přenášet část svých výstav do digitální podoby. Nejenže se tím vyhnou riziku poškození daného předmětu, ale současně dokážou tímhle způsobem ztvárnit nejrůznější epochy z minulosti. Navíc také vyjdou vstříc i široké veřejnosti či novodobějším turistům, jenž jsou zvyknuti na digitální dobu v podobě moderních technologií.

## POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

AIV. Dostupné z: <https://sketchfab.com/3d-models/medieval-chest037b03a3e0274279be4b93b7c7cedf01>

Alexander Nakarada. Dostupné z: <https://soundcloud.com/serpentsoundstudios>

ALP8310. Dostupné z: <https://assetstore.unity.com/publishers/37465>

Andy woodhead. Dostupné z: <https://sketchfab.com/Andywoodhead>

ARCGIS, 2021. *ArcGIS Maps SDK for Unity* [online] [vid. 2021-10-04]. Dostupné z: <https://developers.arcgis.com/unity-sdk/>

AUTODESK, 1990. 3DSMAX [online] [vid. 2021-10-01]. Dostupné z: <https://www.autodesk.cz/products/3ds-max/overview?term=1-YEAR&tab=subscription>

AzTiZ. Dostupné z: <https://sketchfab.com/aztiz>

BAJER, Daniel, 2021. Gamifikace a její využití ve vzdělání [online] [vid. 2021-10-11]. Dostupné z: <https://medium.com/edtech-kisk/gamifikace-a-její-využití-ve-vzdělávání-2c1b5f3864f4>

BLENDER, 1995. *Blender* [online] [vid. 2021-10-04]. Dostupné z: <https://www.blender.org/>

CESIUM, 2012. *3D geospatial visualization for the web* [online] [vid. 2021-11-15]. Dostupné z: <https://cesium.com/platform/cesiumjs/>

Chris Nolit. Dostupné z: <https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488>

deniska40k. Dostupné z: <https://sketchfab.com/3d-models/candle-c408586d8c1547448c4ff47abb4fb981>

DENISNK. Dostupné z: <https://pastebin.com/wVxpxpvt>

E-NEWSPEAK, 2014. *Kód Salomon útočí! – Nová turistická hra o tablet* [online] [vid. 2021-10-21]. Dostupné z: <http://www.e-newspeak.eu/kod-salomon-utoci-nova-turisticka-hra-o-tablet/>

EDUIN, 2015. *Co může využívání herních principů přinést vzdělávání?* [online] [vid. 2021-10-21]. Dostupné z: <https://www.eduin.cz/clanky/co-muze-vyuzivani-hernich-principu-prinest-do-vzdelavani/>



EDWARDS, Luke, 2021. *What is Socratic and How Does It Work? Best Tips and Tricks* [online] [vid. 2021-10-19]. Dostupné z: <https://www.techlearning.com/how-to/what-is-socratic-and-how-does-it-work-best-tips-and-tricks>  
einar Dostupné z: <https://sketchfab.com/3d-models/einar-einar-viking-4c443475435f40859ba1030e21c384a6>

ESSENTIAL, 2020. *Facts About the Video Game Industry* [online] [vid. 2021-10-18]. Dostupné z: <https://www.theesa.com/resource/2020-essential-facts/>

FIALA, Jan, 2019. *Gamifikační ve výuce* [online] [vid. 2021-10-19]. Dostupné z: <https://spomocnik.rvp.cz/clanek/21961/GAMIFIKACE-VE-VYUCE.html>

FREDO6, nedatováno. *TopoShaper* [online] [vid. 2021-10-05]. Dostupné z: <https://sketchucation.com/pluginstore?pln=TopoShaper>

GILBERT, Nestor, 2021. *Number of Gamers Worldwide 2022/2023: Demographics, Statistics, and Predictions* [online] [vid. 2021-10-18]. Dostupné z: <https://financesonline.com/number-of-gamers-worldwide/>

GUNIA, Dominique Andrea, 2014. *Earth3D* [online] [vid. 2021-11-15]. Dostupné z: <https://www.earth3d.org/>

Goldman Studio. Dostupné z: <https://assetstore.unity.com/packages/3d/magic-particle-systems-lite-95868>

HDRIMAPS Dostupné z: <https://hdrmaps.com/freebies/>

HFTOOLS, 1998. *Landscape Utilities for Unix* [online] [vid. 2021-11-15]. Dostupné z: <http://markjstock.org/pages/hftools.html>

HORNE, Richard, 2009. *3DEM* [online] [vid. 2021-11-16]. Dostupné z: <https://freegeographytools.com/2009/3dem-website-is-gone-but-3dem-still-available-here>

iceboxX708. Dostupné z: <https://sketchfab.com/3d-models/saw-cut-sword-9734b19c0e1d422f9048c5222665afea>

Invector. Dostupné z: <https://assetstore.unity.com/publishers/13943>

Joao Paulo. Dostupné z: <https://3dtextures.me/>

JIRÁNEK, Petr, 2020. *Co je to gamifikace?* [online] [vid. 2021-10-21]. Dostupné z: <https://www.petrjiranek.cz/pojmy/gamifikace/>

KARLIN, Mike, 2019. *Classcraft: Engagement Management System for K-12* [online] [vid. 2021-10-22]. Dostupné z: <http://www.edtechroundup.org/reviews/classcraft-engagement-management-system-for-k-12-classrooms>

KLÍMOVÁ, Monika a Věra FIŠEROVÁ, 2014. *Průvodcovské služby a animace (studijní opora pro kombinovanou formu studia)* [online] [vid. 2021-10-04]. Praha: VŠH. ISBN 978-80-87411-55-1. Dostupné z: [https://is.vsh.cz/do/vsh/studijni\\_opory\\_pro\\_nk/Pruvodcovske\\_sluzby\\_a\\_animace\\_978-80-87411-55-1.pdf](https://is.vsh.cz/do/vsh/studijni_opory_pro_nk/Pruvodcovske_sluzby_a_animace_978-80-87411-55-1.pdf)

KŮŽEL, Filip, 2020. *VYZKOUŠENO: SmartGuide je průvodce v mobilu. Mapu a fotky ukáže na displeji, komentář pustí do sluchátek* [online] [vid. 2021-10-06]. Dostupné z: <https://www.zive.cz/clanky/vyzkouseno-smartguide-je-pruvodce-v-mobilu-mapu-a-fotky-ukaze-na-displeji-komentar-pusti-do-sluchatek/sc-3-a-204492/default.aspx>

LBobor. Dostupné z: <https://assetstore.unity.com/publishers/12140>

MALONE, Thomams W., 1980. What Mmakes Things Fun to Learn? [online] [vid. 2021-11-16]. 94. Dostupné z: [https://hcs64.com/files/tm\\_study\\_144.pdf](https://hcs64.com/files/tm_study_144.pdf)

MAT, Ruzinoor Che, Abdul Rashid Mohammed SHARIFF, Abdul Nasir ZULKIFLI, Mohd Shafry Mohd RAHIM a and Mohd Hafiz MAHAYUDIN, 2014. Using game engine for 3D terrain visualisation of GIS data: [online] [vid. 2021-11-15]. Dostupné z: [https://www.academia.edu/26974529/Using\\_game\\_engine\\_for\\_3D\\_terrain\\_visualisation\\_of\\_GIS\\_data\\_A\\_review](https://www.academia.edu/26974529/Using_game_engine_for_3D_terrain_visualisation_of_GIS_data_A_review)

Michael, 2016. Dostupné z: <https://gist.github.com/saint-angels/ef4cb0612730fd516058e40d37b2229a>

MINISTERSTVO PRÁCE A SOCIÁLNÍCH VĚCÍ, 2017. *Průvodce turistů* [online] [vid. 2021-10-06]. Dostupné z: <https://nsp.cz/jednotka-prace/pruvodce-cestovniho-ruchu>

NYTIMEMS, 2019. *500 days of duolingo what you can and cant learn from a language app* [online] [vid. 2021-10-09]. Dostupné z: <https://www.nytimes.com/2019/05/04/smarter-living/500-days-of-duolingo-what-you-can-and-cant-learn-from-a-language-app.html>

OHIO UNIVERSITY, 2022. *Social Marketing Ultimate Guide: Tips and Resources for Modern Marketers* [online] [vid. 2021-11-20]. Dostupné z: <https://onlinemasters.ohio.edu/blog/social-marketing-guide/>

Omni studio. Dostupné z: <https://assetstore.unity.com/publishers/38064>

POLESNÝ, David, 2018. *Podle Kingdom Come: Deliverance vznikla oficiální turistická příručka* [online] [vid. 2021-10-06]. Dostupné z: <https://doupe.zive.cz/clanek/podle-kingdom-come-deliverance-vznikla-oficialni-turisticka-prirucka>

Poneti free. Dostupné z: <https://assetstore.unity.com/publishers/41110>

reddification. Dostupné z: <https://sketchfab.com/3d-models/medieval-torch-worn-16e59c78217d42c195492089d7f398c5>

Starý, Jiří, 2015. *Interaktivní kniha pro mobilní zařízení v herním enginu Unity* [online] [vid. 2021-12-10]. Dostupné z: [https://is.muni.cz/th/yqgxy/stary-text\\_prace.pdf](https://is.muni.cz/th/yqgxy/stary-text_prace.pdf)

SHAKEEL, Abdullah, 2017. *Object2Terrain* [online] [vid. 2021-06-20]. Dostupné z: <http://www.mediafire.com/file/3oyvdjcjix2d3iu/Object2Terrain.cs/file>

Taylor. Dostupné z: <https://sketchfab.com/3d-models/medieval-guard-ca548172f0984660bc4213b9404284fa>

THOMASSEN, Thomas, 2010. *Bitmap To Mesh* [online] [vid. 2021-10-12]. Dostupné z: <https://sketchucation.com/forums/viewtopic.php?f=323&t=31339>

TomasKiniulis. Dostupné z: <https://www.artstation.com/tomaskiniulis>

TRIMBLE, 2000. *Sketchup* [online] [vid. 2021-10-09]. Dostupné z: <https://www.sketchup.com/>

UNIDATA, 2007. Unidata Integrated Data Viewer (IDV) [online] [vid. 2021-11-15]. Dostupné z: [https://www.researchgate.net/publication/235793841\\_Unidata\\_Integrated\\_Data\\_Viewer\\_IDV](https://www.researchgate.net/publication/235793841_Unidata_Integrated_Data_Viewer_IDV)

UNITY, 2005. *Unity* [online] [vid. 2021-10-09]. Dostupné z: <https://unity.com/>  
VanyaMazalevich. Dostupné z: <https://sketchfab.com/3d-models/candle-holder-38c0f8b4df1b41e092e8cdc623b3b6ff>

VINUKONDA, Parthasarathy, 2018. *Virtual Reality and Gamification: A Blend for Better Learning Outcomes and Performance* [online] [vid. 2021-10-18]. Dostupné z: <https://trainingindustry.com/blog/learning-technologies/virtual-reality-and-gamification-a-blend-for-better-learning-outcomes-and-performance/>

WEST, Chris, 2010. *Terrain* [online] [vid. 2021-11-15]. Dostupné z: [http://www.west-racing.com/mf/?page\\_id=2983](http://www.west-racing.com/mf/?page_id=2983)

YOSTN, 2001. *3D Graphical Map Viewer* [online] [vid. 2021-11-17]. Dostupné z: <https://sourceforge.net/projects/g3dgmw/>

## **PŘÍLOHY**

# SEZNAM PŘÍLOH

## **Volné přílohy**

Příloha 1 DVD

Příloha 2 Poster

## **Popis struktury DVD**

Adresáře:

- aplikace
- web