

# 3D vizualizace hradu Starý Světlov

David Falešník

---

Bakalářská práce  
2020



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav počítačových a komunikačních systémů

Akademický rok: 2019/2020

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **David Falešník**  
Osobní číslo: **A17013**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Informační technologie v administrativě**  
Forma studia: **Prezenční**  
Téma práce: **3D vizualizace hradu Starý Světlav**  
Téma práce anglicky: **A 3D Visualisation of Starý Světlav Castle**

**Zásady pro vypracování**

1. Seznamte se s historií hradu Starý Světlav a zpracujte literární rešerši.
2. Vyhledejte vhodné historické materiály, které bude možné použít pro 3D vizualizaci hradu Starý Světlav.
3. Ve vybraných programech vytvořte podrobný 3D model hradu Starý Světlav z 15. století.
4. K vytvořenému modelu přidejte vhodné textury a materiály. Využijte dostupných historických materiálů pro zajištění co největší autentičnosti.
5. Dosažené výsledky prezentujte formou vyrenderovaných obrazů, animací a aplikace virtuální reality.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. MICHÁLEK, Ladislav. Starý Světlav. Zříceniny hradu severně od Luhačovic. 1. Hukvaldy: PUTUJME, 2013. ISBN 978-80-87897-06-3.
2. RAZÍM, Vladislav. Řeč středověké fortifikační architektury: terminologie a její souvislosti. V Praze: Národní památkový ústav, 2018. ISBN 978-80-86516-96-7.
3. SHANNON, Tom. Unreal Engine 4 for design visualization: developing stunning interactive visualizations, animations, and renderings. Boston: Addison-Wesley, [2018]. ISBN 0134680707id.
4. VILLAR, Oliver. Learning Blender: A Hands-on Guide to Creating 3D Animated Characters. Second edition. Boston, MA: Addison-Wesley, [2017]. ISBN 0134663462id.
5. BLAIN, John M. The complete guide to Blender graphics: computer modeling & animation. Fifth edition. Boca Raton: Taylor & Francis, a CRC title, part of the Taylor & Francis imprint, a member of the Taylor & Francis Group, the academic division of T&F Informa, 2019. ISBN 9780367184742.

Vedoucí bakalářské práce:

**Ing. Pavel Pokorný, Ph.D.**

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: 19. prosince 2019

Termín odevzdání bakalářské práce: 27. května 2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Pracovní úkol	1. Úvodní kapitola
Pracovní úkol	2. Teoretická část
Pracovní úkol	3. Metodická část
Pracovní úkol	4. Praktická část
Pracovní úkol	5. Závěrečná část

### Podmínky vypracování

1. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
2. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
3. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
4. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
5. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.

Pracovní úkol

Pracovní úkol

Pracovní úkol

Pracovní úkol

1. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
2. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
3. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
4. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.
5. Práce musí být vypracována v souladu s podmínkami uvedenými v zadání.

doc. Mgr. Milan Adámek, Ph.D.  
děkan



doc. Ing. Martin Sysel, Ph.D.  
garant oboru

## **Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

## **Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 6.8.2020

David Falešník v.r.  
podpis diplomanta

## **ABSTRAKT**

Bakalářská práce na téma „3D vizualizace hradu Starý Světlov“ si dává za cíl ukázat postup práce při tvorbě rozsáhlé scény v herním engine a s 3D grafikou jako takovou. Začátek teoretické části pojednává o historii hradu Starý Světlov a o době jeho největší slávy. Zbylá část teoretické části pak pojednává o 3D grafice v širším měřítku a použitých programech. V praktické části je nejdříve popisován sběr informací, dále postup při modelování jednotlivých částí hradu v programu Blender a ve finální části se zaměřuje na přesunutí práce do Unreal Engine a vypracování jednotlivých výstupů, které zahrnují rendery, animaci a virtuální aplikaci.

Klíčová slova: 3D grafika, historie, hrad Starý Světlov, Blender, Unreal Engine, rendery, aplikace

## **ABSTRACT**

The bachelor thesis with the topic "3D visualization of the castle Starý Světlov" aims to show workflow tasks in creating large scenes in the game engine and 3D graphics as such. The beginning of the theoretical parts deals with the history of the Starý Světlov castle and the time of its greatest glory. The rest of the theoretical part deals with 3D graphics in wide areas and used programs. The practical part first describes the collection of information and then the procedure for modeling individual parts of the castle in Blender. The final part is focused on moving the work to the Unreal Engine and developing individual outputs, which include renders, animation and virtual application.

Keywords: 3D graphics, history, Starý Světlov castle, Blender, Unreal Engine, renders, applications

Chtěl bych touto formou poděkovat panu Radimu Vrlovi z Národního památkového ústavu v Kroměříži za jeho odborné rady a postřehy, které do značné míry formovali tuto práci. Dále bych chtěl poděkovat vedoucímu této práce, Ing. Pavlu Pokornému Ph.D. za jeho vedení a rady k vytvoření této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD .....</b>	<b>9</b>
<b>I TEORETICKÁ ČÁST .....</b>	<b>10</b>
<b>1 HISTORIE HRADU STARÝ SVĚTLOV .....</b>	<b>11</b>
1.1 PRAVĚKÉ OSÍDLENÍ .....	11
1.2 PRVNÍ ZMÍNKY O HRADU.....	11
1.3 REKONSTRUKCE HRADU .....	12
1.4 CHÁTRÁNÍ HRADU .....	13
1.5 ŠTERNBERKOVÉ .....	13
1.6 GOTIKA.....	13
<b>2 3D GRAFIKA.....</b>	<b>14</b>
2.1 MODELOVÁNÍ .....	14
2.1.1 Typy modelování.....	15
2.1.2 LOD.....	16
2.2 TEXTURA.....	17
2.2.1 UV Mapping.....	17
2.3 SHADER .....	18
2.3.1 Pixel shader .....	18
2.3.2 Vertex shader.....	18
2.3.3 PBR shader.....	19
2.4 RENDEROVÁNÍ .....	19
<b>3 POUŽITÉ PROGRAMY .....</b>	<b>21</b>
3.1 BLENDER .....	21
3.1.1 Renderovací engine .....	22
3.2 UNREAL ENGINE .....	23
3.2.1 Tvorba v Unreal Engine.....	24
3.2.2 Využití .....	24
3.3 QUIXEL.....	25
3.4 DAVINCI RESOLVE.....	26
<b>II PRAKTICKÁ ČÁST .....</b>	<b>27</b>
<b>4 KONCEPCE PRÁCE A ZÍSKÁNÍ PODKLADŮ.....</b>	<b>28</b>
4.1 PŘÍPRAVA MODELOVÁNÍ.....	29
<b>5 HRAD .....</b>	<b>30</b>
5.1 BRÁNA.....	30
5.2 HRADBY .....	35
5.3 OSTATNÍ BUDOVY HRADU.....	37
5.4 OBRANNÁ VĚŽ .....	37
<b>6 PŘEDHRADÍ A PODHRADÍ.....</b>	<b>38</b>
6.1 MODELOVÁNÍ BUDOV .....	38
<b>7 TEXTUROVÁNÍ.....</b>	<b>40</b>

7.1	APLIKACE TEXTUR .....	41
<b>8</b>	<b>TVORBA SCÉNY .....</b>	<b>43</b>
8.1	TERÉN .....	43
8.1.1	Získání terénních informací .....	44
8.1.2	Tvorba terénu .....	45
8.2	VKLÁDÁNÍ OBJEKTŮ .....	47
8.3	VEGETACE .....	47
8.4	MATERIÁLY .....	48
8.5	NASVÍCENÍ SCÉNY .....	50
<b>9</b>	<b>VÝSTUPY .....</b>	<b>51</b>
9.1	RENDERY .....	51
9.2	ANIMACE .....	52
9.3	APLIKACE .....	53
	<b>ZÁVĚR .....</b>	<b>55</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>57</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>60</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>61</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>62</b>



## ÚVOD

Smyslem této práce je umožnit lepší představu a obeznámení o již neexistujícím středověkém hradu Starý Světlov, který se v současné době nachází jako zřícenina nedaleko Luhačovic. Jedná se o hrad, který ve své době plnil významnou strategickou roli převážně v obraně proti Uhrům a neměl by tedy být zapomenut.

Toto téma jsem si vybral hlavně z důvodu mého osobního zájmu jak o historii, tak o 3D grafiku a tvorbu virtuálního světa jako takového. Zaujala mě také myšlenka nejen zobrazit široké veřejnosti již neexistující památku, ze které zbylo jen pár kamenů v obvodových zdích, ale také ukázat pomocí interaktivní aplikace, jak mohl daný hrad a jeho okolí v 15 stoletím vypadat z pohledu běžného člověka.

Před samotným modelováním bude nejdříve nutno se seznámit s historií daného hradu, k tomu značně pomůže kniha Starý Světlov od Ladislava Michálka a informace na webové stránce obce Podhradí, do jejíž správy hrad patří. V další řadě bude nutné získat co možná nejvíce informací o dané lokalitě a možné podobě hradu. Tyto informace poskytne pan Radim Vrla z Národního památkového ústavu v Kroměříži, bez jehož pomoci by vytvořit daný model nebylo možné a se kterým budou jednotlivé postupy v práci konzultovány.

Dále se bude nutné seznámit s principy 3D modelování a texturování v jednotlivých programech. Převážně u programů Blender, Quixel Mixer a Unreal Engine se bude nutné naučit ovládání na pokročilé úrovni, aby výsledkem byl co nejrealističtější výstup. Bude se jednat o teoretické i praktické znalosti v dané problematice.

Hlavním cílem této práce tedy bude vymodelovat za pomoci 3D programu Blender hrad a jeho okolí, následně budou dané modely texturovány v programu Quixel Mixer a přeneseny do Unreal Engine 4 pro finální obrazovou vizualizaci daného hradu společně s animací, která bude zobrazovat hrad a jeho součásti z různých stran a finální aplikace, ve které bude daný hrad možno procházet. Finální animace bude dále upravena pomocí editačního programu DaVinci Resolve 16, zde se bude jednat především o korekturu barev dané animace. Jednotlivé výstupy budou následně nabídnuty Národnímu památkovému ústavu v Kroměříži a obci Podhradí pro šíření lepšího povědomí o této památce.

## **I. TEORETICKÁ ČÁST**

## 1 HISTORIE HRADU STARÝ SVĚTLOV

Hrad Starý Světlov byl založen zřejmě v první polovině 14. století jako významné sídlo jedné z rodových větví Šternberků. V 15. století se stal důležitým vojenským a strategickým centrem, ale od přelomu 15. a 16. století zde život utichal a hrad nezadržitelně pustnul. Od druhé poloviny 18. století se zdivo navíc rozebíralo na stavební materiál. Z původně rozsáhlého hradu se dochovaly pouze impozantní terénní útvary a pozůstatky zděných konstrukcí soustředěné v prostoru hradního jádra, zdevastované ovšem působením přírodních vlivů i vandalstvím. [1]

### 1.1 Pravěké osídlení

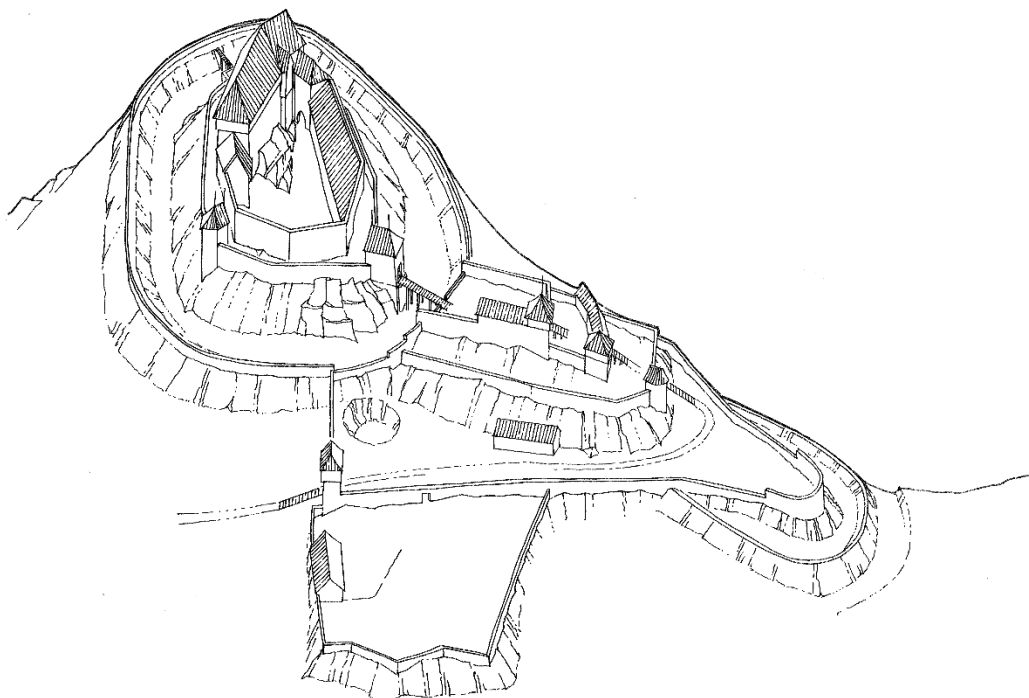
Okolí Starého Světlova bylo člověkem „objeveno“ už několik tisíciletí před vznikem samotného hradu. Z pozdní doby kamenné (4000 – 2000 př. n. l.) je známý nález sekeromlatu z Provodova. Rovněž v pozdní době bronzové (1000 – 750 př. n. l.) a ve starší době železné (750 – 400 př. n. l.) byl zdejší kraj silně zalidněn. V těchto časech vzniká mohutné hradisko na návrší Rýsov a další sídliště se nalézalo na katastru obce Podhradí, kam v současné době spadá i hrad Starý Světlov. V mladší době železné (400 př. n. l. až počátek letopočtu) obýval zdejší končiny takzvaný lid púchovské kultury. Ten byl silně ovlivněn vyspělým keltským osídlením v Pomoraví. Osídlení pak končí pravděpodobně na začátku našeho letopočtu a v následujících staletích zůstává zdejší kraj opuštěný. [2]

### 1.2 První zmínky o hradu

Během druhé poloviny 13. století dochází k osidlování moravsko-uherského pohraničí. V této době budují páni ze Šternberka nad obcí Provodov na jednom z návrší skalní hrádek Rýsov, který měl především obranný charakter před vpády Uhrů. Právě vpád uherského krále Matúše Čáka Trenčianského na Moravu se mu stal osudným a spolu s vypálením města Uherského Hradiště došlo k jeho zničení. Rýsov již nebyl nikdy obnoven. Šternberkové během let 1333–1335 ztrácejí také své sídelní místo hrad Lukov a jsou nuceni rychle hledat náhradu. Tak vzniká na jednom z výběžků vrcholového hřebene Vizovických vrchů nedaľeko vrchu Brda nový hrad Světlov. I když hrad Lukov vykoupil ze zástavy moravský markrabě Karel, pozdější český král a císař Karel IV., Šternberkové jsou opět jeho majiteli. První zmínka z roku 1360 již Světlovský hrad zmiňuje v držení pánů ze Šternberka. [2]

### 1.3 Rekonstrukce hradu

Mohutná rekonstrukce hradu proběhla zhruba v polovině 15. století. Neklidná doba husitských válek společně s následujícími válkami česko-uherskými přinesla pro hrad velké změny. Především zde bylo nutno ubytovat velké množství lidí. Jádro hradu bylo zesíleno novou kamennou parkánovou hradbou s cimbuřím a předsunutou bránou. Prstencový val byl na straně možného ohrožení nepřítelem zesílen a uzpůsoben pro postavení střelných zbraní. Na západě ho přerušila nově vyhloubená studna. Na nádvoří byla zvelebena budova obytného paláce. Zapuštěné přízemí bylo vystavěno celé z kamene. První a druhé patro roubené či hrázděné zakončeno vysokou sedlovitou střechou, převyšovalo obvodový hradební ochoz. Dá se předpokládat, že zároveň s velkou přestavbou hradního jádra změnilo svou podobu i předhradí uzpůsobené početné vojenské družině. Byla zesílena obrana přístupové cesty obrannými věžemi u bran horního předhradí. Plocha dolního předhradí byla zaplněna velkými, dřevěnými stavbami, ve kterých přebývali zbrojnoši a jejich rodiny. [2]



Obrázek 1 – Pravděpodobná podoba hradu

## 1.4 Chátrání hradu

Po skončení uherských válek se začala čím dál více ukazovat nevýhodná poloha hradu Světlova. Stával na samém okraji panství, hluboko v lesích, daleko od důležitých cest. Bojkovice, které byly povýšeny na město, se stávaly významným hospodářským centrem panství Landštejnů. Ti se rozhodli vystavět zde nové sídlo, Nový Světlov a roku 1490 přesídlili do Bojkovic. Hrad Starý Světlov byl pak ponechán svému osudu. Roku 1517 při prodeji panství Jitkou z Landštejna rytíři Burianovi z Vlčnova je zmiňován v příslušenství světlovského panství "hrad pustý Starej Světlov". Stavba hradu je zakreslena také v Mullerově mapě Moravy, což je dokladem, že hrad je stále zachován. Pozůstatky hradu pak byly součástí světlovského panství až do roku 1945.[2]

## 1.5 Šternberkové

Šternberkové patří mezi prastaré české panské rody a jsou také spjati s jihovýchodním pomezím Moravy, kde sídlili na hradě Lukově. Podle svého bájného prapředka Diviše, žijícího někdy na počátku 12. století, byli nazýváni Divišovci. První historicky doloženou postavou rodu je pak jiný Diviš z Divišova, jehož syn Zdeslav si někdy roku 1241 vybudoval na ostrohu nad řekou Sázavou hrad Šternberk, dnes Český Šternberk, který pojmenoval dle rodového znamení hvězdy (německy Štern). Podle hradu pak rod převzal své jméno. [3]

Už ve 13. století některé větve velmi početné rodiny přesídlily na Moravu. Rodina patřila vždy k úzkému okruhu lidí určující dějiny českého státu. Například Kašpar a František ze Šternberka se výraznou měrou zasloužili o vznik Národního muzea roku 1818. Potomkům toho rodu patří dodnes hrad Český Šternberk. [4]

## 1.6 Gotika

Gotika je zřejmě nejznámějším uměleckým slohem středověku. Navazuje na románský sloh a její počátky začínají ve Francii v druhé polovině 12. století. Oproti románskému slohu, který zaznamenáváme po celé Evropě, se gotika uchytila pouze ve střední a západní Evropě. Do Česka se gotika dostává až v polovině 13. století, ovšem přetrvává zde déle, než je tomu v jiných zemích. [3]

## 2 3D GRAFIKA

V současné době je trend 3D grafiky na čím dál větším vzestupu. Můžeme se s ní setkat od televizních reklam, přes celovečerní filmy, počítačové hry až po virtuální realitu či reálné předměty díky 3D tisku. Pro vytvoření takové 3D grafiky je ovšem potřeba více, než vytvořit virtuální model ve třech dimenzích, neboli v osách x,y,z jak název napovídá. [4]

### 2.1 Modelování

Modelování je z pohledu 3D grafiky často považováno za nejjednodušší odvětví a prakticky každý, kdo se 3D grafikou zabývá se s modelováním v menší, či větší míře setkal. Ovšem i u modelování je zde nutnost znát mnoho informací o tom, jak 3D prostor a vše s ním spojené funguje.

Jednou ze základních informací, kterou by měl znát každý kdo se zabývá 3D modelováním je rozdíl mezi takzvanými vertexy, edgi a polygony. [4]

**Vertex** – bod mezi minimálně třemi edgi, jedná se o nejmenší část modelu, se kterou můžeme pracovat a upravovat tak vzhled daného objektu

**Edge** – úsečka mezi dvěma vertexy, stejně jako u vertexu i zde můžeme s touto úsečkou pohybovat a upravovat tak vzhled objektu

**Polygon** – často také nazýván „face“ se skládá minimálně ze tří vertexů, potažmo tří edgů. Tvoří tak plochu daného objektu, na kterou již můžeme například nanést texturu. [5]

Dále je také nutné zvolit, zda daný model bude takzvaně „high poly“, nebo „low poly“. Hlavním rozdílem, jak název napovídá, je počet polygonů a tím pádem i detailnost daného modelu. High poly modely se používají převážně při filmové, či animační tvorbě, kde není nutností renderovat daný model v reálném čase a snímky se renderují při finálním vytvoření snímku. Low poly se používají převážně v herním průmyslu, a to z důvodu renderování v reálném čase přímo ve hrách, které tak potřebují co největší výkon, který by high poly značně omezoval. Často se také používá způsob takzvaného „bakingu“, neboli zapečení, Jedná se o způsob, kde se detaily z high poly modelu „zapečou“ do low poly modelu, který tak vypadá detailně i s menším počtem detailů. [6]

### 2.1.1 Typy modelování

Pokud se řekne 3D grafika, většině z nás se vybaví nějaký 3D model ať už nějaké virtuální postavy, prostředí, techniky nebo budovy. Než ovšem takový model spatří světlo světa, předchází tomu hodiny až dny titěrné práce. S postupem času bylo vytvořeno již mnoho různých typů modelování, kterými se dá dosáhnout požadovaného cíle. [4]

#### Primitive modeling

Pro značnou část neodborné veřejnosti může být překvapením, že jejich oblíbená herní, či filmová postavy, která byla vytvořena digitálně, byla na počátku modelování prostá krychle, či koule. Tímto postupem se zabývá takzvaný Primitive modeling. Jedná se pravděpodobně o nejrozšířenější způsob modelování. Jeho princip je prostý, ve vybraném modelovacím programu vložíme takzvaný prostý objekt. Například již zmíněnou krychli či kouli. Následně k ní různými nástroji v programu přidáme další a další polygony, těm upravíme velikost, směr, či jinou vlastnost tak, abychom ve výsledku dosáhli požadovaného tvaru. [4]

#### Spline modeling

Hlavním rozdílem mezi Spline modelingem a Primitive modelingem je v tom, že na rozdíl od Primitive, Spline modeling nepracuje s celým objektem, ale pouze s jednou plochou. Můžeme si to představit například jako destičku, která se může libovolně ohýbat, otáčet a upravovat do nejrůznějších tvarů. Tento modeling funguje na bázi křivek, které určují dané zahnutí povrchu. Jedná se se o obdobnou práci s křivkami jako například ve 2D grafice. Tento způsob se využívá hlavně u modelování aut, techniky či velmi zaoblených objektů, jako je například váza. [4]

#### 3D scan

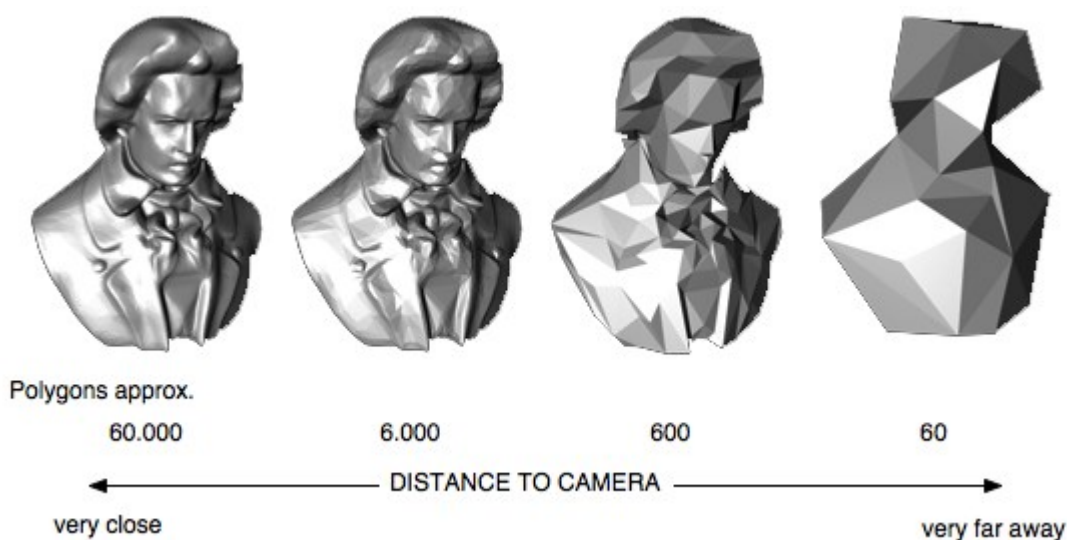
Skenování na poli 3D grafiky je v poslední době čím dál více se rozrůstající se trend, kterým se dají realné objekty přenést do počítače v trojrozměrné podobě bez nutnosti úplné modelace. Princip je takový, že herec, či objekt je nafocen několika desítkami kamer ze všech úhlů a za různých světelných podmínek. Tato data jsou následně přenesena do počítače kde z nich program udělá ucelený 3D model. [4]

## Sculpting

Proces sculptingu je asi běžnému člověku nejbližší, jedná se totiž prakticky o sochařství v digitální podobě. Pomocí sculptingu se nejvíce modelují postavy všeho druhu, od lidských po příšery či zvířata, nebo přírodní útvary jako například skály, či kameny. Nejedná se ovšem pouze o modelování postav či kamenů, ale o velmi detailní práci zahrnující modelaci i těch největších detailů na modelu. [4]

### 2.1.2 LOD

LOD, neboli Level of detail, určuje počet polygonů na daném objektu. Nejčastěji se používá ve 3D grafice pro počítačové hry, a to z důvodu čím dál většího důrazu na detailní modelování 3D modelů a čím dál více se rozrůstající oblasti, které počítač v reálném čase musí vykreslovat. Jedná se celkově o pět stupňů, takzvaných levelů, kde každý z těchto levelů určuje redukci daných polygonů. Tyto levely si daný program, nebo engine určuje podle výkonosti daného počítače. Například když si v počítačových hrách nastavujeme kvalitu grafiky, program si automaticky upraví i LOD jednotlivých modelů ve hře. Principem je počítání vzdálenosti objektu od kamery zobrazující daný render. Čím blíže je daný objekt, tím detailnější musí být, ovšem jak se vzdálenost od objektu zvětšuje, lidské si přestává všimnout jednotlivých detailů, a tak ani model nemusí být tak detailní. Jako příklad si můžeme představit lidskou postavu, která se od nás vzdaluje. Na prvních metrech rozpoznáme sebemenší detail, ovšem jak se daná osoba vzdaluje, jednotlivé detaily splývají dohromady. [7]



Obrázek 2 – Ukázka rozdílů mezi LOD levely [24]

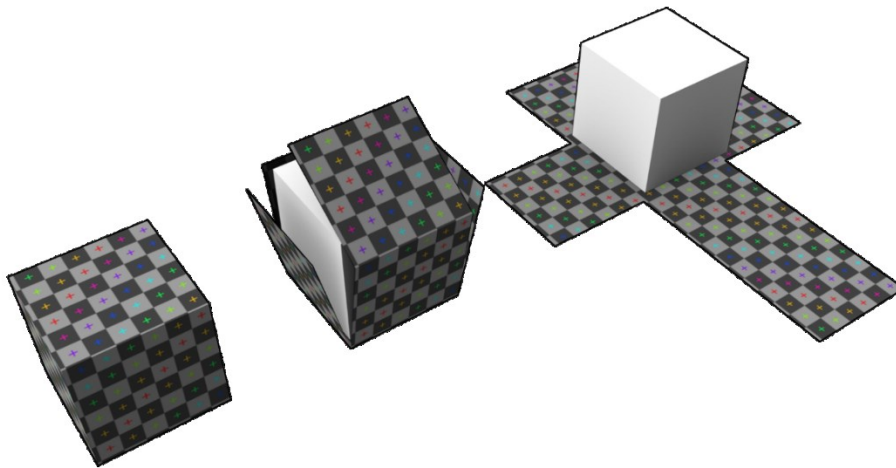


## 2.2 Textura

S pojmem textura se setkáváme i jiných uměleckých směrech, než je 3D grafika, případně i v běžném životě. Ve 3D grafice se texturou myslí pomyslný obal modelu, který je zpravidla fotografií, či obrázkem. Samotný obrázek se ovšem již v současné době nepoužívá, je doprovázen minimálně popisem, jak se daný povrch má lesknout, jak moc světla má odrážet či jak má být hrubý. V současné době je čím dál více vyhledávaný způsob tvoření textur za pomoci texture map neboli PBR shaderů popsaných výše. [8]

### 2.2.1 UV Mapping

UV mapování je nedílnou součástí každého texturování. Textury jako takové krom souřadnic  $x, y, z$  mají také souřadnice  $U$  a  $V$ . Tyto souřadnice jsou souřadnice v daném vrcholu a jejich rozsah je od 0 do 1. Pomocí UV souřadnic pak určujeme jak má být daný objekt texturován. Můžeme si to představit jako balení vánočního dárku, kdy balicí papír je naší texturou a dárek objektem, který chceme do této textury zabalit. [8]



Obrázek 3 – UV mapování krychle [25]

## 2.3 Shader

Jakýkoliv 3D model má daný soubor informací, které předává renderovacímu enginu. Tomuto souboru informací se říká shader. Takto daný engine pozná informace o tom, jak daný model, respektive jeho povrch reaguje na dopadající světlo. Tyto informace jsou nutné pro zobrazení daného modelu, v opačném případě by se jednalo jen o skupinu bodů a povrch modelu by byl neviditelný. Každý shader je tvořen tak, aby simuloval materiál, který známe z reálného světa. Například dřevo, železo, nebo voda. Každý z těchto materiálů je jinak průhledný, průsvitný, lesklý či hrubý. [4]

### 2.3.1 Pixel shader

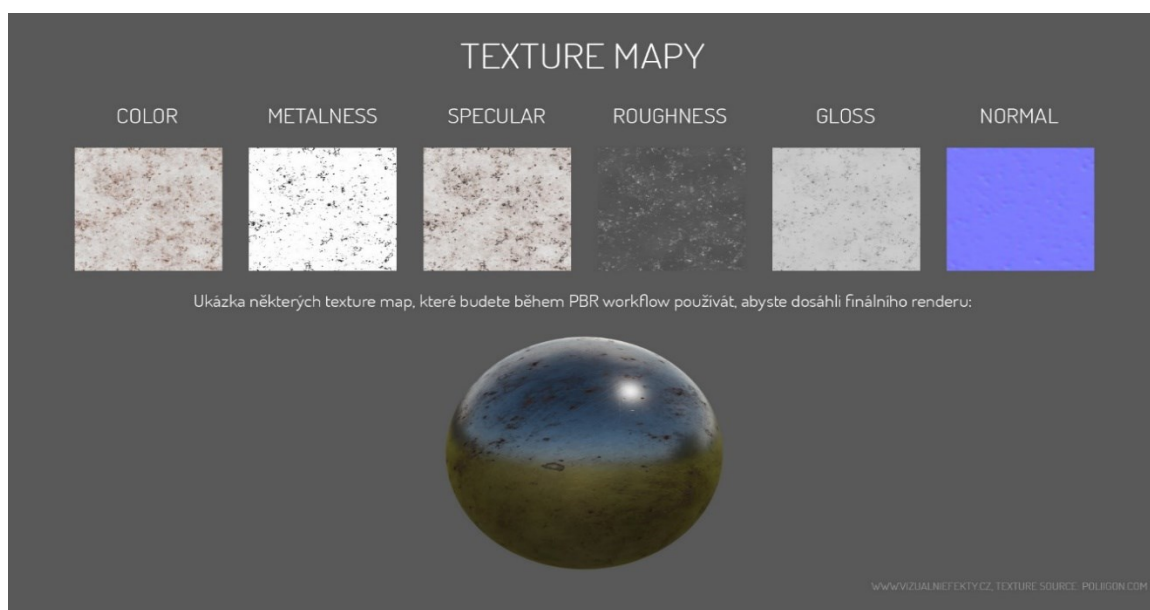
Pixel shadery jsou plně programovatelné programy, spadající do kategorie 2D shaderů, které běží na grafickém čipu. Ve zlomcích sekund dokáží o tom, jakým způsobem má objekt reagovat na nasvícení. Jinými slovy Pixel shader operuje s jednotlivými pixely, které se skládají z vertexů. Musí tak spolupracovat s Vertex shadery a v renderovací pipeline se nacházejí až za nimi. Na základě informací o vertexech pak provádí jednotlivé operace s celými pixely. Jeho příkladem může být zapojení do takzvaného bump mappingového slotu materiálu v modelovacím programu, který má pak za příčinu optické zvlnění povrchu objektu, i když model objektu je dokonale rovný. [4]

### 2.3.2 Vertex shader

Vertex shade nefungují oproti Pixel shaderům ve 2D prostoru, nýbrž ve 3D. Znamená to tedy že pracují s mnohem většími geometrickými tvary. Tento typ shadingu se následně stará o vzhled celých objektů a generování různých efektů jako je například mlha. Principem Vertex shaderu je práce s danými vertexy, které mohou být definovány několika proměnnými. Například souřadnicemi, barvou, texturou nebo světelnými charakteristikami. Vertex shader ovšem nemění typ dat, nýbrž změny hodnoty dat tak, aby se objevil vrchol například s jinou barvou, nebo texturou. [9]

### 2.3.3 PBR shader

PBR, neboli Physically Based Rendering shadery jsou v poslední době nejrozšířenější a nejmodernějším způsobem, jak dostat fotorealistických materiálů daného objektu. U předchozích shaderů jsme se snažili odhadnout, jakou odrazivost či hrubost může mít například kov, nebo dřevo. PBR ovšem simuluje jakým způsobem se světlo po takovýchto materiálech šíří v reálném světě. Fakt, že se jedná o fyzickou simulaci s reálnými hodnotami může mít sice svá negativa například pokud se snažíme o stylizované modely, pokud je ovšem cíle dosáhnout co nejreálnějších výsledků tak je PBR shader jasnou volbou. PBR materiály se skládají z takzvaných texture map (viz obrázek níže), které určují jednotlivé složky daného materiálu jako je odrazivost, lesklost, hrubost, výšku daného materiálu a další vlastnosti. [10]



Obrázek 4 – Mapy PBR materiálu [11]

## 2.4 Renderování

Renderování, nebo v českém překladu Vykreslování je proces, při kterém dochází ke generování 2D obrazu z 3D prostředí pomocí softwarového programu pro to určeného. Dané prostředí je popis trojrozměrných objektů v přesné jazykové, nebo datové struktuře. To obsahuje například geometrii, pohledy kamery, textury či osvětlení. Jedná se o poslední část při tvorbě 3D obrazu, či animace. [11]

Pro renderování se používají momentálně dva způsoby:

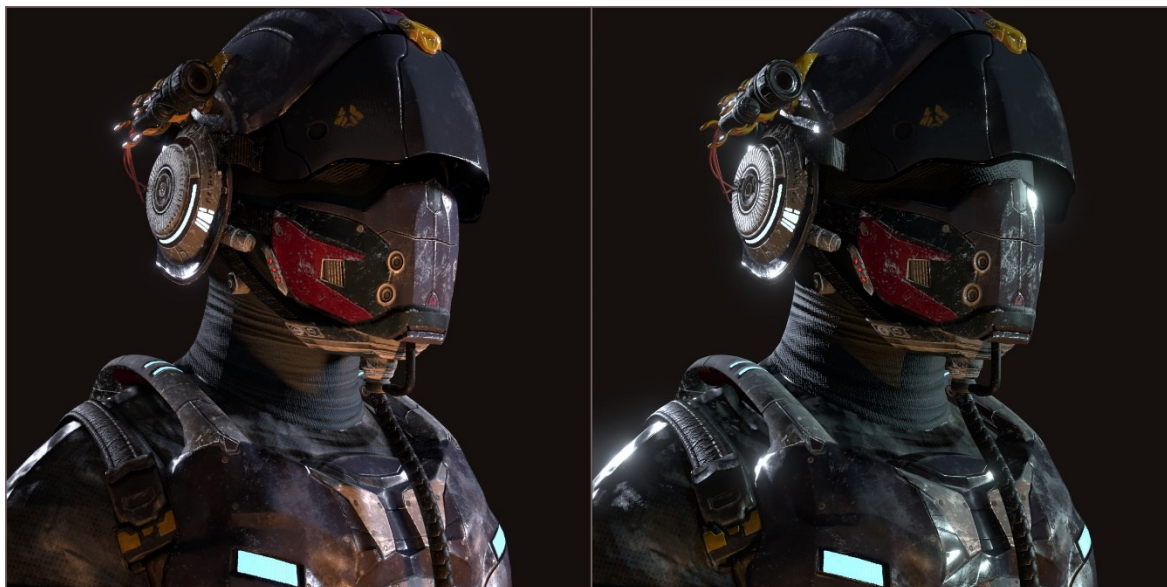
### **Real-Time**

Real-Time rendering, neboli vykreslování v reálném čase, se používá převážně v oblastech 3D grafiky, ve které je dbáno převážně na výkon. Jedná se o oblasti interaktivní grafiky či počítačových her. Vzhledem k tomu, že je v takovýchto oblastech interakce uživatele, potažmo hráče, na vysoké úrovni, je nutno vykreslit rozsáhlé prostředí. K tomuto napomáhá výkonný grafický hardware a předkompilace informací o scéně. [11]

### **Pre-Rendering**

Pre-Rendering, nazýván také offline rendering, by se do českého jazyka mohl přeložit jako plánovaný, či náročný render. Tato technika vykreslování se používá v prostředích, kde není vyžadováno rychlé vykreslování. Jedná se o techniku hojně využívanou například u animací, vizuálních efektů či fotorealistických renderů, kde je pevně dán úhel kamery a není tak nutné vykreslovat celou scénu. Na rozdíl od Real-Time vykreslování, kde se využívají převážně grafická jádra u offline renderu se jedná o jádra procesoru.

[12]



Obrázek 5 – Rozdíl mezi Pre-Rendering a Real-Time [26]

### 3 POUŽITÉ PROGRAMY

Pro tuto práci bylo použito vícero, převážně volně dostupných, programů. Jedná se o programy Blender, který byl zvolen k tvoření 3D modelů, Quixel Bridge a Mixer pro tvorbu textur, Unreal Engine 4 pro finální kompozici a tvorbu prostředí, nebo Da Vinci Resolve pro úpravu vytvořené animace. Popis konkrétní práce s uvedenými programy se dále nachází v praktické části této práce.

#### 3.1 Blender

Blender je bezplatný a otevřený grafický program pro počítačové modelování a animace. Obsahuje veškeré nástroje potřebné pro vytváření modelů a scén, které mohou být zpracovány do statických obrázků nebo animovaných filmů. [13]

Kromě modelování a animací zde lze tvořit další i řadu z dalších 3D i 2D prací. Jedná se o práce jako rigging, simulace, renderování, kompozice, motion tracking, nebo editování videa a tvorba her. V Blenderu lze dále také programovat, konkrétně se jedná o psaní scriptů v jazyce Python pro přizpůsobení celé aplikace či vytvoření nových funkcí, takzvaných *addonů*. Jak bylo zmíněno výše, jedná se o otevřený program, který funguje na všech platformách. Tedy na platformách Windows, iOS a Linux. Díky *General Public Licence* je veřejnosti umožněno zasahovat do kódu ať už s malými, či velkými úpravami. Tyto úpravy jsou často, i díky široké komunitě, následně implementovány do dalších verzí softwaru. Jedná se například o opravu chyb programu, vylepšení stávajících funkcí či převzetí nově vytvořených funkcí. Do Blenderu lze také investovat pomocí paušálních poplatků, které pomohou tvůrcům s přípravou dalších verzí. [14]

### 3.1.1 Renderovací engine

Program Blender používá celkem 3 renderovací systémy pro znázornění vytvářeného modelu ve scéně. Každý z těchto programů. Jedná se o systémy Cycles, Eevee a Workbench. Z tohoto výčtu je však umožněno vytvářet finální obrazy pouze Cycles a Eevee, Workbench je primárně určen pro renderování během tvorby modelů a slouží tak spíše jako kontrolní systém než finální renderer.

#### Cycles

Jedná se o fyzikálně renderovací engine vyvíjený společností Blender Foundation, která stojí i a samotným programem Blender. Velkou předností oproti Eevee, které je popsáno v kapitole , je reálnější zobrazování světla a stínů. Využívá se zde totiž sledování dopadu paprsků světla v závislosti na úhlu a nastavení kamery. Je schopen k renderování využívat i grafická chip dle výrobce. U značky Nvidia se jedná o CUDA, kdežto u značky AMD o pomoc skrz OpenCL a podporuje také render na více grafických kartách najednou. [15]

#### Eevee

Eevee je realtime renderovací systém, na rozdíl od Cycles, u kterých se jedná o pre-renderovací systém. Tento renderovací systém je postaven na užívání OpenGL a zaměřen na rychlost a možnost interakce v reálném čase což je velmi užitečné při použití v sekci Viewport i za předpokladu, že je pro finální render je užito Cycles a Eevee je použito například pro kontrolu PBR materiálů. Materiály se pro Eevee tvoří stejným způsobem, pomocí nodů. Na rozdíl od Cycles, Eevee nevyužívá systém sledování paprsků světla. Namísto výpočtu každého paprsku používá proces zvaný rasterizace. Rasterizace odhaduje, jak světlo interaguje s objekty a materiály pomocí řady algoritmů. [16]

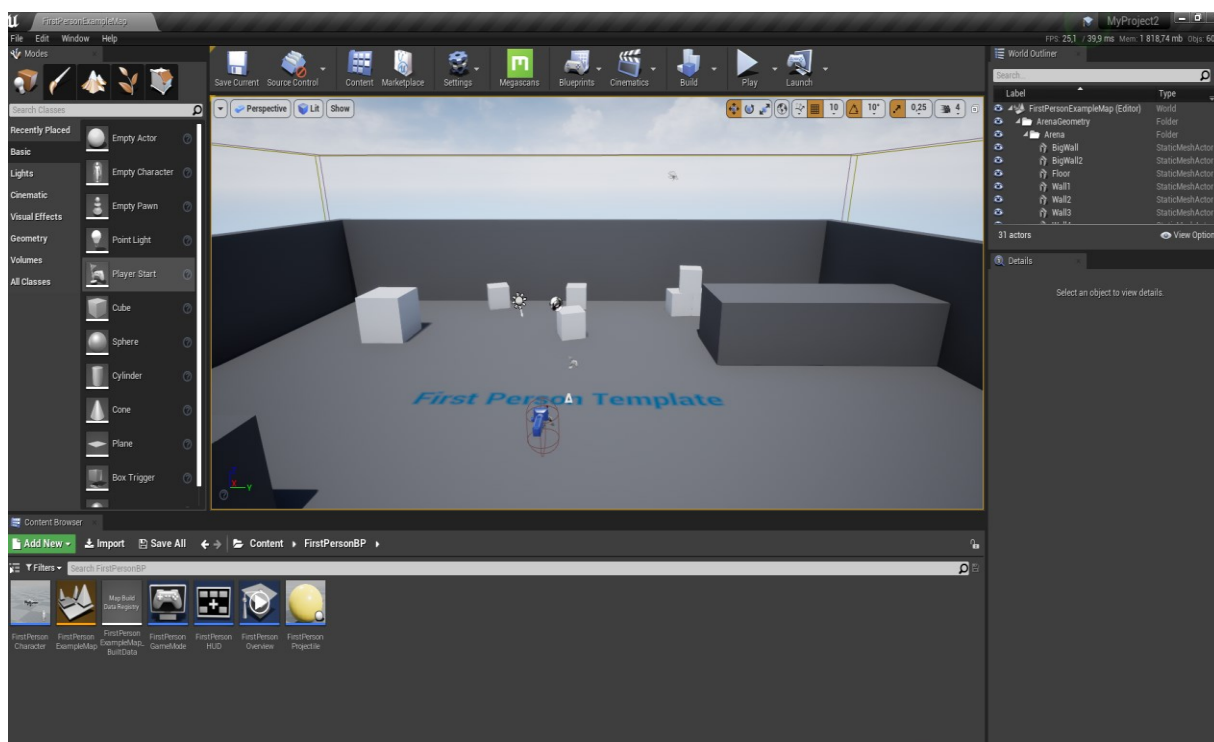
#### Workbench

Jedná se o engine, určený pro rychlé renderování během modelování a tvorby animací. Nejedná se tedy o engine, který by byl určen k tvorbě finálních obrázků. V základním nastavení se Workbench stará o nasvícení a stínování ve Viewportu. Dále také podporuje možnost přidělení náhodných barev objektům pro lepší přehlednost, toto rozdělení je možno také provést podle stejných materiálů, či vertexů. [17]

## 3.2 Unreal Engine

Unreal Engine je v současné době nejvyužívanějším herním enginem na světě, který vyvíjí společnost Epic Games od roku 1998. Využívá se především v herním průmyslu a proslavil se hrami jako je například Fortnite, kterou vydala právě společnost Epic Games. Dalším z možných využití Unreal Engine je tvorba architektonických vizualizací, nebo v současné době stále se rozvíjející tvorba vizuálních efektů do filmů či tvorba prostředí pro virtuální realitu. [18]

Unreal Engine umožňuje vývoj aplikací na všechny typy zařízení. Jedná se tedy o podporu všech operačních systémů pro počítače, herní konzole, jako je například Play Station od společnosti Sony, či již zmíněnou virtuální realitu. I díky tomuto vývoji je hlavní volbou pro mnoho vývojářů. K tvorbě aplikací pomocí Unreal Engine napomáhá také fakt, že společnost Epic Games podporuje studia a vývojáře, kteří se rozhodnou upřednostnit tento engine před konkurenčními, jako jsou například Unity či Cry Engine. [19]



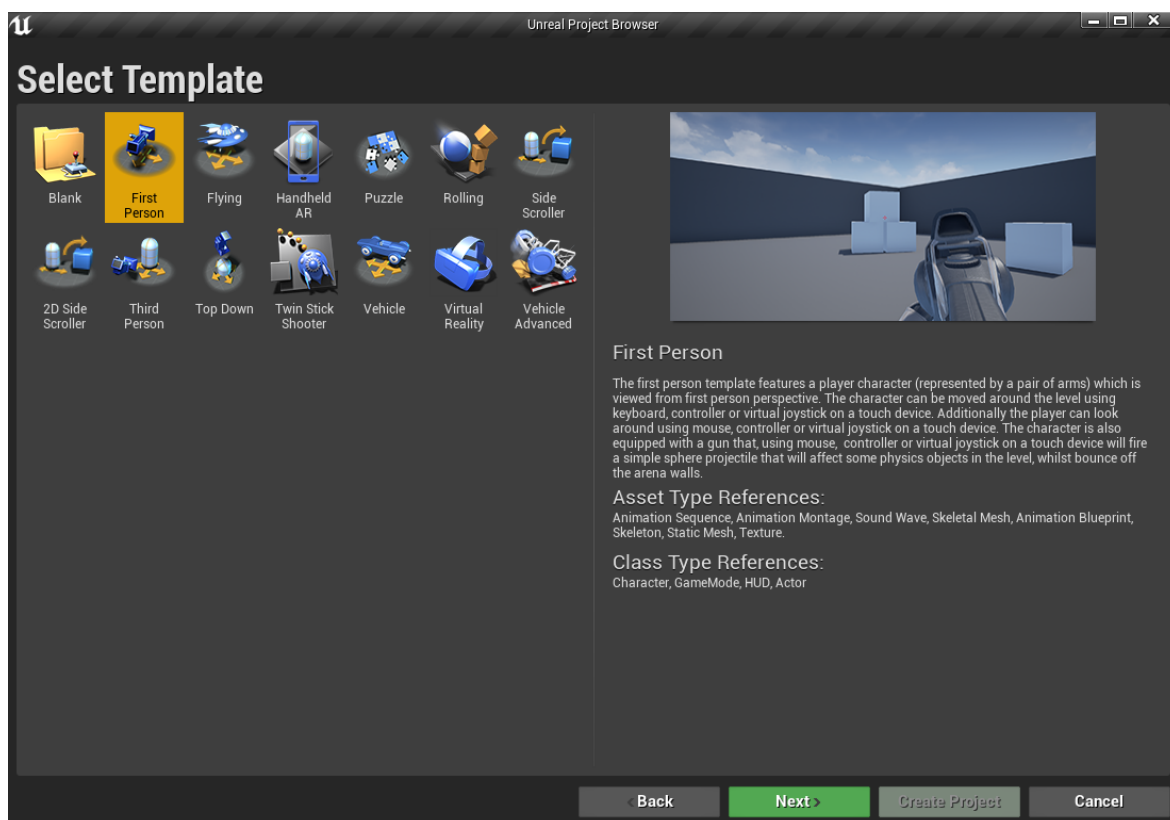
Obrázek 6 - Prostředí Unreal Engine

### 3.2.1 Tvorba v Unreal Engine

Tvorba aplikací a vizualizací je v Unreal Engine možná dvěma způsoby. Prvním způsobem je tvorba aplikací pomocí programovacího jazyka C++, který je v herních enginech značně rozsáhlý, a to i mezi odbornou veřejností. Druhým způsobem je pak využití takzvaných Blueprints. Jedná se o již předpřipravené bloky, které jsou mezi sebou spojovány pomocí nodů. Tyto bloky tak umožňují tvořit aplikace i amatérské veřejnosti, ovšem hojně je využívají i některá v herní studia. [19]

### 3.2.2 Využití

Prvotní určení Unreal Engine bylo tvoření výhradně her z první osoby. Postupem času i kvůli nárokům trhu byla zpřístupněna možnost tvorby i ostatních žánrů her. Pro tvorbu her Unreal Engine nabízí celou řadu šablon se základním prostředím, ve kterých je již předdefinováno, jakým směrem se daná aplikace bude vyvíjet. Zda půjde o akční hru z první, či třetí osoby, VR aplikaci či architektonickou vizualizaci. Tyto šablony mohou být následně rozšiřovány podle již zmíněných Blueprints a tím pádem danou aplikaci může vytvořit kdokoliv. [18]



Obrázek 7 – Nabídka šablon v Unreal Engine

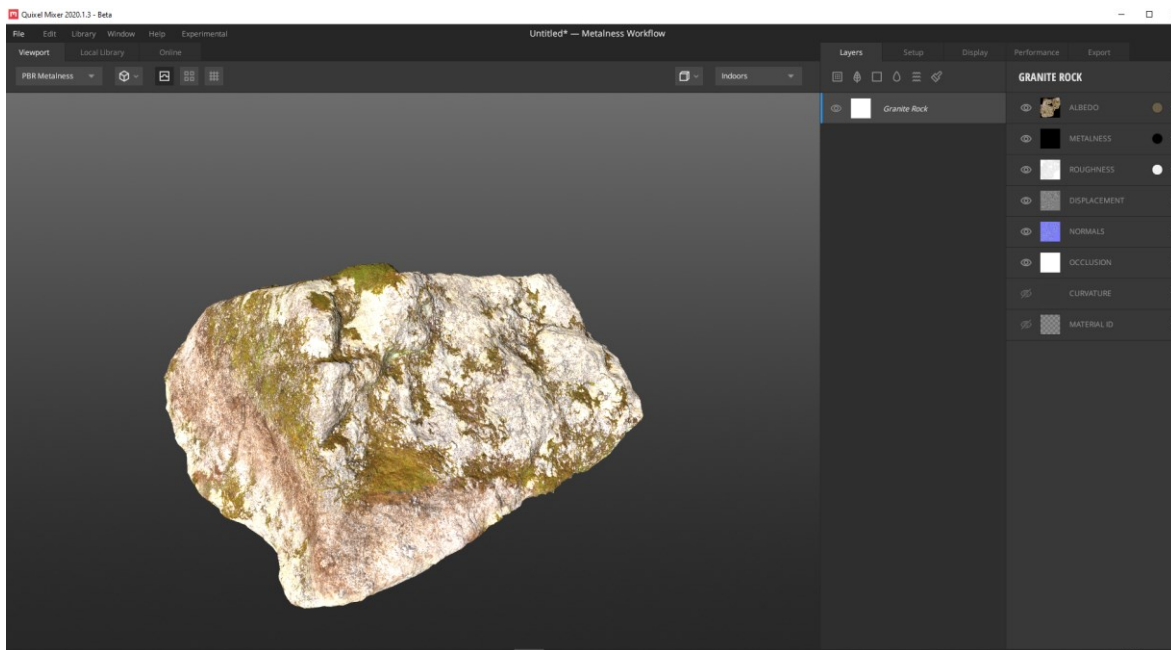


### 3.3 Quixel

Společnost Quixel byla založena v roce 2011 za účelem zjednodušení a zrychlení výtvarného procesu v digitální tvorbě. Napomáhá tomu díky stále se rozšiřující knihovně materiálů i modelů, které jsou získávány pomocí 3D skenování. Tato knihovna je díky spolupráci se společností Epic, stojící za Unreal Engine, zcela zdarma a dostupná tak pro všechny tvůrce. Quixel momentálně poskytuje dva programy, jedná se o programy Quixel Mixer a Quixel Bridge. [20]

Quixel Mixer je, dříve též nazýván Megascans Studio, je nástroj podporující kombinaci naskenovaných materiálů a možnosti přidávat do daných materiálů různé detaily. Napomáhá tak vytvářet unikátní materiály. Quixel Mixer je zdarma k použití, a to i bez aktivního předplatného a na platformách Windows a Mac. [20]

Quixel Bridge je aplikací pro online knihovnu s naskenovanými assety a materiály, které jsou ustavičně přidávány. Díky propojení s Quixel Mixer je také možnost jednoduchého exportu vlastních materiálů. V současné době obsahuje Quixel Bridge takzvaný live link do všech 3D programů a engineů. Tento link umožňuje během pár kliknutí exportovat vybraný model či materiál přímo do zvoleného programu. [21]

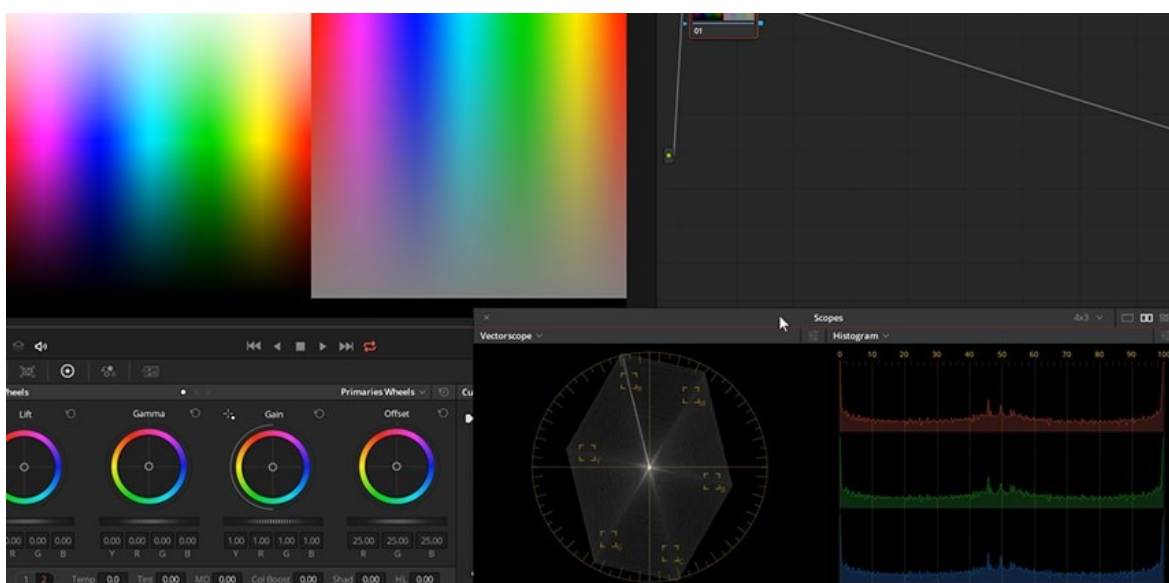


Obrázek 8 – Prostředí Quixel Mixer

### 3.4 DaVinci Resolve

Jedná se o editační systém, který v momentální době tvoří společnost Blackmagic Design. První verze tohoto programu vyšla v roce 2004 v té době pod záštitou společnosti Vinci Systems a jednalo se o poměrně převratný software. Převážně jeho editor pro korekce barev pro televizní a filmovou tvorbu byl hojně využíván. Roku 2009 byla práva prodána již zmiňované společnosti Blackmagic Design, která od té doby začala vyvíjet novodobý systém a propojovat editor se svým video a audio hardwarem. V současné době je nejaktuálnější šestnáctá verze programu. [22]

Další velkou předností DaVinci Resolve je volně dostupná verze tohoto programu v plně vybaveném provedení, které hravě konkuruje placené konkurenci. DaVinci Resolve nabízí i placenou a vylepšenou verzi s přídavkem Studio. Tato verze, jak název napovídá, je mířena hlavně na studiovou tvorbu a běžnému uživateli postačí volně dostupná verze. Jak bylo zmíněno výše, mezi hlavní přednosti tohoto softwaru se řadí korektura barev daného videa, která je dosud na vysoké úrovni a využívána i řadou profesionálů. Mezi další možnosti, jak využít tento software patří použití oken jako jsou Editace videa, Kompozice obrazu, nebo tvorba vizuálních efektů. [23]



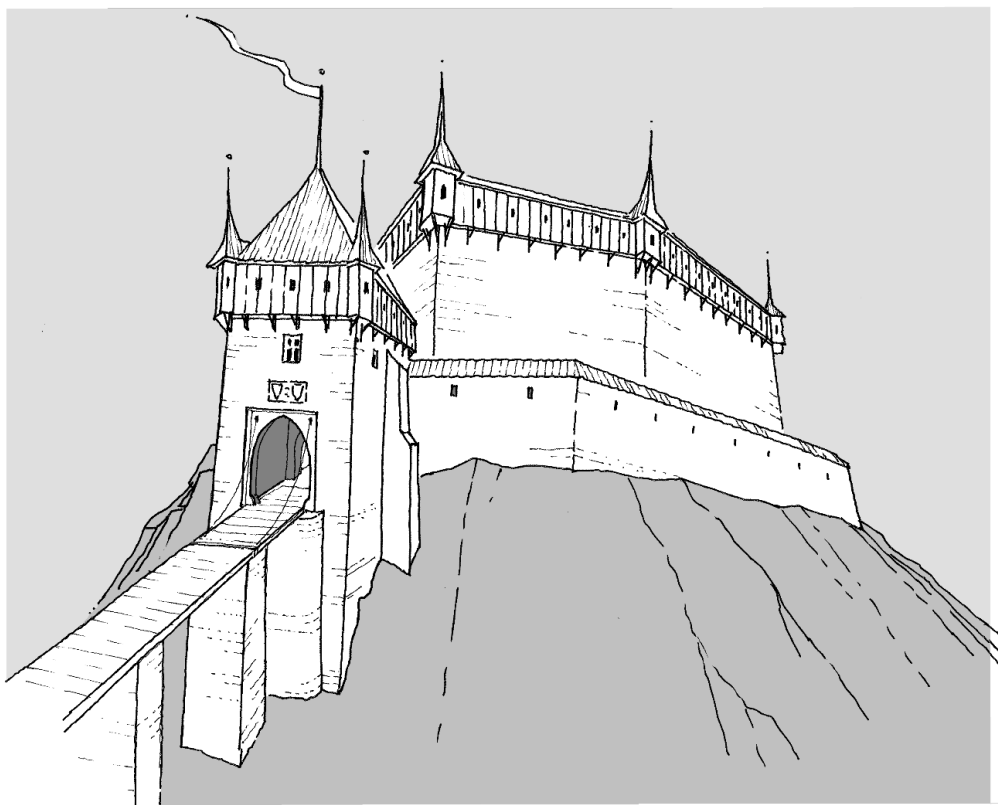
Obrázek 9 – Nástroje pro úpravu barev v DaVinci Resolve

## **II. PRAKTICKÁ ČÁST**

## 4 KONCEPCE PRÁCE A ZÍSKÁNÍ PODKLADŮ

Před samotným začátkem modelování hradu bylo nutné získat co nejvíce informací, referencí a podkladů pro jeho vyhotovení. Toto značně ztěžoval fakt, že z hradu krom terénních úprav, a ne příliš rozsáhlých zbytků zdiva mnoho nezbylo. Velkou zásluhu na podkladech, poznatcích a informacích o hradu má pan Radim Vrla z Národního památkového ústavu v Kroměříži. Díky několika ilustracím zaslaných panem Vrlou bylo možné modelovat základní části hradu, které byly následně s panem Vrlou konzultovány. Velmi také pomohla samotná návštěva místa, kde v současné době zřícenina hradu stojí a bylo možné tak nafotit jednotlivé terénní úpravy, zbytky zdiva, či krajinu v blízkosti hradu a udělat si tak lepší představu o daném místě a o možné podobě hradu.

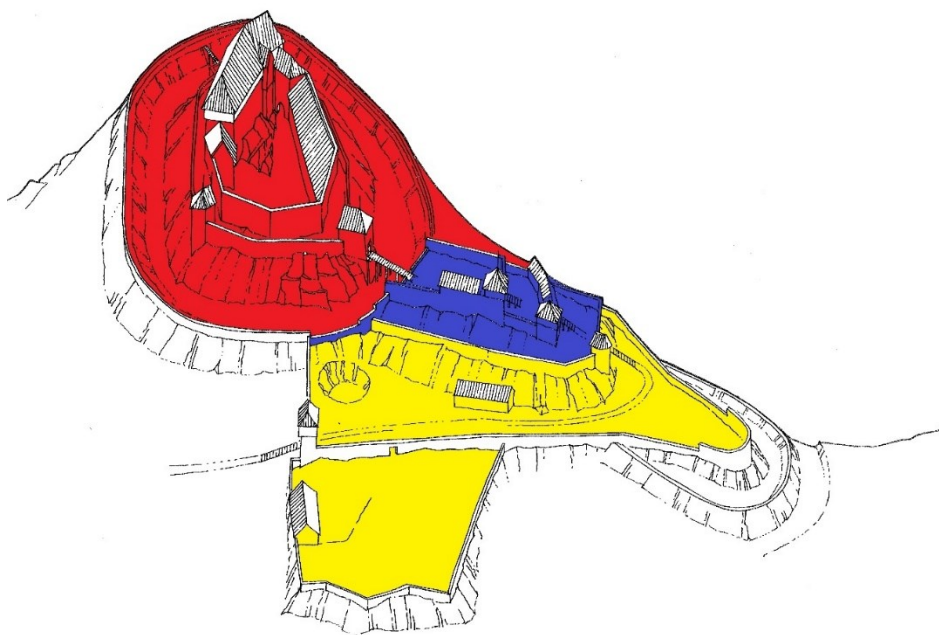
Dalším zdrojem referencí byl internet samotný, respektive webové stránky například různých skanzenů, dochovaných hradů či stránky zabývající Gotickou architekturou jako takovou. Z těchto stránek bylo možné získat reference například o typu obydlí v podhradí či nejruznější detaily.



*Obrázek 10 – Pravděpodobná podoba hlavní brány*

## 4.1 Příprava modelování

Po shromáždění dostatečného množství informací a referencí vhodných pro tuto práci se mohlo přejít k samotnému modelování. Celkový model byl rozdělen do tří částí podle rozložení hradu. Toto rozložení sloužilo pro lepší rozfázování práce. Rozložení je znázorněno v *obrázku 11*. Jednalo se o podhradí, předhradí a hrad samotný. Do těchto částí pak bylo možné modelovat a umisťovat objekty pro ně příznačné. Bylo tedy nutné vytvořit roubenky, hospodářská stavení a nástroje do podhradí, palisádové věže do předhradí, trámový most spojující hrad s předhradím i věže a hradby hradu. Takto vytvořené části celého komplexu pomocí programu *Blender* byly následně texturovány pomocí programu *Quixel Mixer*, kde byly vytvořeny unikátní materiály pro každý z modelů. Po vytvoření modelů a textur bylo možno vše exportovat do *Unreal Engine 4* pro finální kompozici. Zde bylo možné celý hradní komplex sestavit a následně mohlo dojít k vytvoření aplikace a animací.



Obrázek 11 – Rozdělení částí hradu

## 5 HRAD

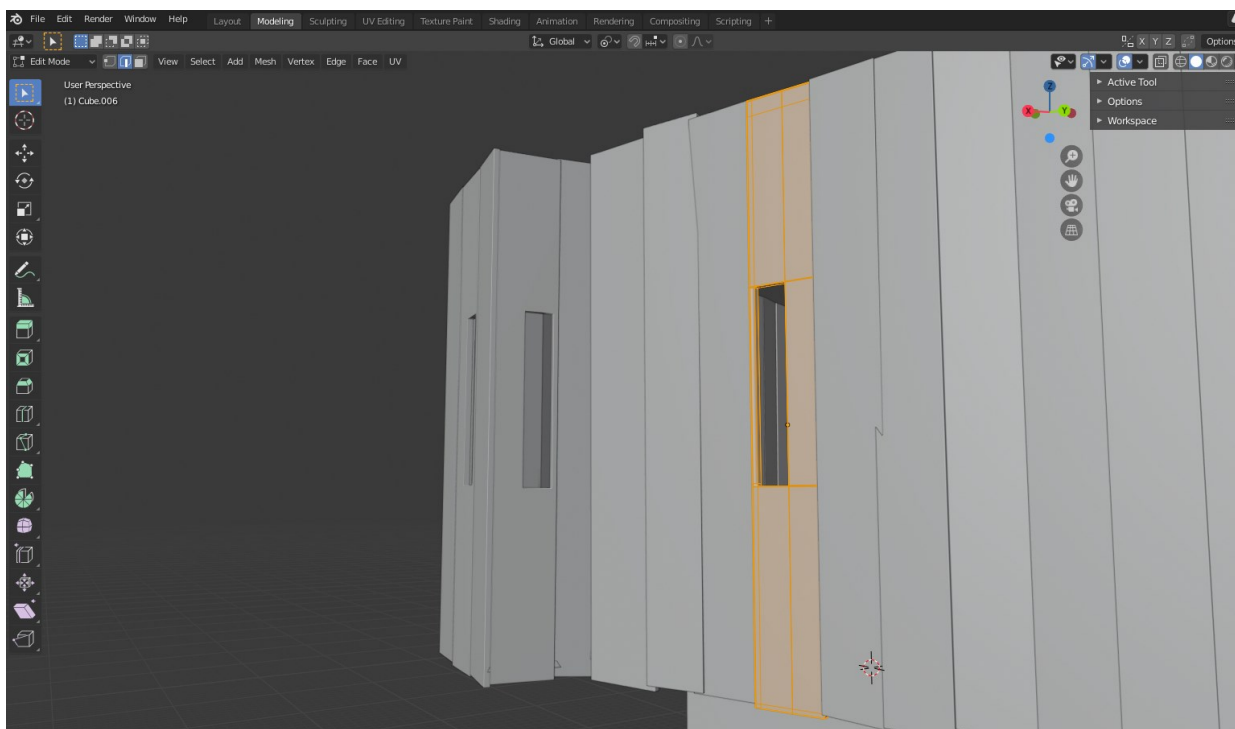
V současné době je hrad jedinou, alespoň z malé části, dochovanou stavbu dříve rozsáhlého celku. Do přítomnosti se dochovaly pouze obvodové zdi hradeb a hlavní brány. To ovšem stačilo pro vytvoření nákresů hradu a jeho nejbližšího okolí, ze kterých čerpá i tato práce.

### 5.1 Brána

Hlavní brána s padacím mostem je asi nejzachovalejší částí hradu, a to samé platí o nákresech, ve kterých je detailně znázorněna. Byla tedy zvolena jako první stavba této práce. Model jako takový se skládá z několika částí, které byly modelovány separátně. Jedná se o šindelovou střechu, dřevěné patro se střílnami a dvěma věžičkami, model kamenné věže, průchod a padací most.

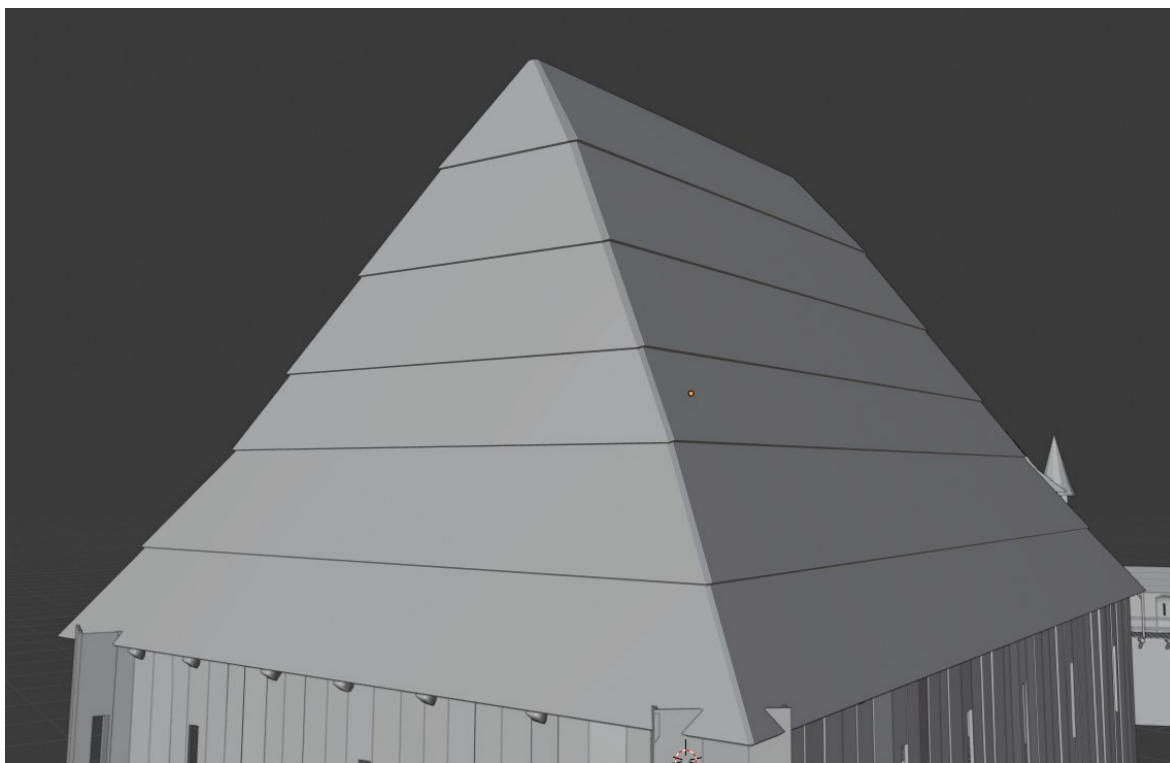
Pro jednodušší postup byla jako počáteční část vybrána kamenná věž. Díky této volbě se tak lépe dodržely proporce jednotlivých částí celého modelu, jednalo totiž o největší část modelu, která určuje i samotnou výšku věže. Postup tvorby byl prostý, pomocí zkratky *Shift + A* byla přidána jednoduchá krychle, která byla následně roztáhnuta pomocí funkce klávesy *S (Scale)* do rozměrů odpovídajících pravděpodobné velikosti věže. Při tomto rozšiřování bylo využíváno zamykání jednotlivých os pomocí kombinace kláves *S + X*, *Y*, nebo *Z* pro zamčení určité osy. Následně byla spodní strana této krychle mírně rozšířena. Toto rozšíření bylo provedeno v editačním režimu, kde byla nejdříve označena spodní část a následně pomocí funkce *Scale* rozšířena. Toto rozšíření bylo provedeno z důvodu vytvoření většího důrazu na realističnost, kde věže bývaly takto rozšířeny kvůli větší stability. Následně byl takto upravený model rozdělen na segmenty zkratkou *Ctrl + R (Loop-Cut)*. Takto rozdělený model lépe umožňuje vytvořit otvor daným modelem, v tomto případě otvor pro budoucí průchod celou věží. Otvor byl vytvořen v editačním módu vybráním jednotlivých segmentů modelu a následným odstraněním těchto segmentů. Obdobným postupem byl vytvořen prostor pro okna. Takto upravený model bylo možno opustit a začít pracovat na dřevěném patře.

Dřevěné patro se skládá z mnoha dřevěných prken, které byly vytvořeny obdobným způsobem jako kamenná věž. Za pomoci základní krychle a funkce *Scale* byla tato krychle přeměna na kvádr odpovídající rozměrům prken, ze kterých mohlo být vytvořeno dané patro. Následně daný model prkna byl duplikován pomocí zkratky *Shift + D* do potřebného počtu a pomocí funkce posunu, zvoleného klávesou *G* umístěn na vybrané místo. Tento postup byl možný udělat i modifikátorem *Array*, který může být pro podobnou práci vhodnějším nástrojem, ovšem pro větší detailnost byl zvolen postup popsany výše. Při takto zvoleném postupu lze lépe manipulovat s jednotlivými modely a lépe tak dosáhnout například větší nepravidelnosti v rozložení prken. Poté, co byly vytvořeny 4 strany tohoto patra bylo nutné vyhotovit střílny. Otvor střílen byl vytvořen pomocí modifikátoru *Boolean*. Ten vytvoří pomocí jednoho objektu otvor stejného tvaru do objektu druhého. Pro toto byla tedy vytvořena základní kostka, která byla pomocí funkce *Scale* a zamčení změny rozměrů na osách *Z* a *Y* změněna do tvaru odpovídajícího otvoru střílny. Následně bylo nutné vybrat tento objekt a pomocí zmíněného modifikátoru vytvořit otvor do objektu představující prkenní obložení dřevěného patra. Poté již stačilo stejný proces opakovat i u ostatních zvolených prken tak, aby bylo dosaženo pravidelné rozložení střílen po celém patře.



Obrázek 12 – Tvorba střílny v editačním módu

Pro model střechy byl zvolen typ šindelové střechy. Šindelová střecha byla ve středověku velmi rozšířená ať už pro běžné stavby prostého lidu, tak i pro šlechtické budovy. Pro vy-modelování doškové střechy pro věž brány byl zvolen defaultní objekt krychle. Ta byla nejdříve roztáhnuta do potřebné výšky, šířky a délky pomocí funkce *Scale* a zamčení na jednotlivých osách. Dále bylo nutno zmenšit její vrchní stranu na co nejmenší rozměry. Toho se docílilo v editačním módu výběrem horní strany krychle a použitím funkce *Scale* zamčené nejdříve v ose *Y* a poté *X*. Tímto zmenšením se dosáhlo požadovaného tvaru a velikosti střechy. Následovalo vytvoření schodovitěho dojmu střechy pro lepší uvěřitelnost, že se jedná o šindelový typ střechy. K tomu bylo nutné rozdělit střechu do několika částí za pomoci zkratky *Ctrl + R (Loop-Cut)*, která rozdělí model do určeného počtu segmentů. Následně byl vždy vybrán jeden segment po celém obvodu střechy a použita funkce *Extrude*, vyvolána klávesou *E*, která vytvoří další polygonovou stranu objektu.



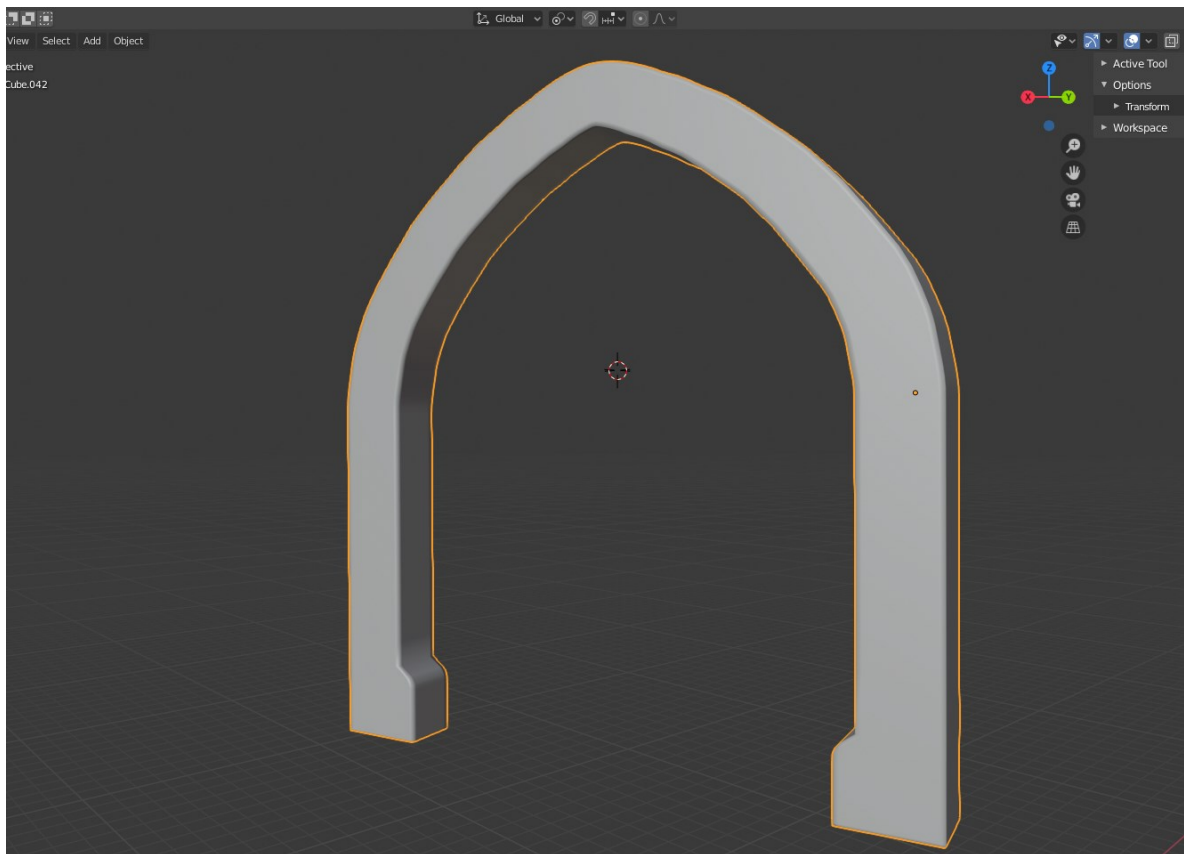
Obrázek 13 – Model šindelové střechy



Takto nově vytvořená strana byla pak pomocí funkce *Scale* zvětšena, po zadání velikosti v centimetrech, o určitou vzdálenost. Pro tento krok byla zvolena velikost 3 centimetry a model byl zvětšován od spodního segmentu vždy o 3 centimetry více než segment předchozí, tím se zajistil schodovitý tvar střechy a také přesná návaznost jednotlivých segmentů.

Průchod bránou a tím pádem i celou věží bylo nutné vymodelovat v dostačující kvalitě z důvodu následného použití ve výsledné aplikaci. Bylo nutné zachovat realistické rozměry jak průchodu, tak i vstupní brány samotné. To zajistil vstupní oblouk, *obrázek 14*, který je i typickým prvkem Gotické architektury. Před vytvořením tohoto oblouku bylo nejdříve nutné umístit do programu referenci, podle které bude následně modelováno. Tato reference byla přidána pomocí zkratky *Shift + A*, následně byl z nabídky vybrán *Image a Reference*. Po tomto postupu byla vybrána reference z uložště v počítači a program jí již sám umístit do scény. Po přidání reference byla do scény přidána také základní krychle, ze které byl oblouk modelován. Krychle byla zmenšena pomocí funkce *Scale* a umístěna na počátek oblouku u reference. Následně byla, v editačním režimu, vybrána horní plocha krychle a pomocí funkce *Extrude* byla tato krychle po malých krocích rozšiřována dle počáteční reference. V případě nutnosti byla použita také funkce rotace pro změnění směru rozšiřování.

Po dosažení vrcholu daného oblouku byl použit modifikátor *Mirror*. Ten daný objekt duplikuje a zrcadlově otočí podle určené osy. Po použití a aplikování tohoto modifikátoru byl oblouk hotov a funkcí posunutí umístěn na začátek průchodu. Oblouk byl dále také duplikován a umístěn na druhý konec průchodu, který je z terénních důvodů výše než oblouk předšlý. Tím byl vyčleněn rozsah průchodu. Pro vyplnění průchodu bylo nutné oba oblouky označit, přejít do editačního režimu a u obou z modelů vybrat zadní strany modelu. Označené strany byly mezi sebou propojeny pomocí funkce *Bridge*, která dané strany spojí do jedné. Tímto krokem tak vznikl průchod samotný.



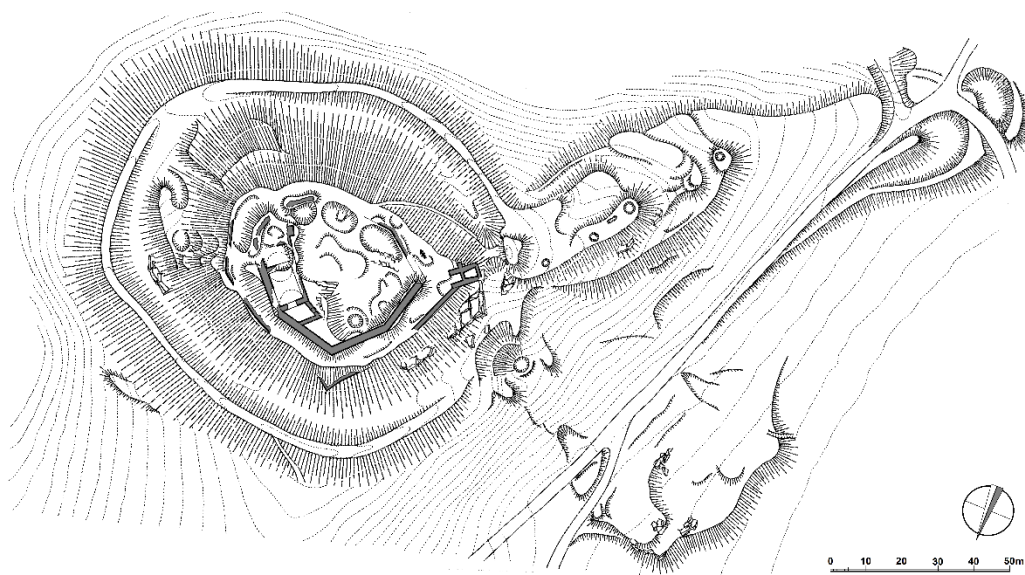
Obrázek 14 – Vstupní oblouk brány

Dle nákresů a archeologických nálezů je velmi pravděpodobné, že hlavní brána obsahovala i padací most. Ten byl vytvořen obdobným způsobem, jako bylo tvořeno dřevěné patro. Jedná se o most tvořený z velkých trámů a prken tvořených stejným postupem. Řetěz držící most byl vytvořen pomocí základního objektu Torus, který byl v editačním módu v polovině roztáhlý pomocí funkce *Extrude* do tvaru, připomínající oko řetězu. To bylo duplikováno a následně otočeno o 90 stupňů. Takto připravený pár modelů byl následně duplikován a pomocí rotace upravován do úhlu simulujícího visící řetěz a zapojen do trámu ležícího padacího mostu.

## 5.2 Hradby

Oproti Obranné věži jsou u hradeb výhodou dochované zbytky zdiva, které tak velkou měrou pomáhají ke správnému rozvržení hradu, ale také například k určení tloušťky hradeb. Hradby se ovšem nedochovaly celé, a tak k finálnímu modelu velkou měrou napomohl půdorys celého hradu a blízkého okolí dodaného panem Vrlou. Tento půdorys byl do programu naimportován pomocí klávesové zkratky *Shift + A*, vybrána možnost *Image a Reference*. Reference byla přidána z pohledu z vrchu pro lepší manipulaci a rozvržení následných hradeb. Jako poslední krok přípravy reference bylo nutné obrázek zvětšit natolik, aby odpovídal reálným proporcím. Po přípravě reference bylo možné začít s modelováním samostatných hradeb, jednalo se o dvě úrovně hradeb s odlišným tvarem a proporcemi. První, vnější hradba byla vedena podél většiny hradu, byla například vedena mezi bránou a obrannou věží a jednalo se o hradbu s cimbuřím

Byla modelována pomocí základní krychle, která byla vložena pomocí klávesové zkratky *Shift + A* a následně v sekci *Mesh* vybrána *Cube*. Tento základní objekt byl následně upraven pomocí funkcí *Scale* a *Rotate* tak, aby zapadal do předpřipraveného půdorysu. Následně byl v editačním módu rozšiřován pomocí funkce *Extrude* do tvaru kopírujícího daný půdorys. Po vytvoření základního modelu hradeb bylo nutné vytvořit cimbuří.



Obrázek 15 – Půdorys hradního komplexu

To bylo vytvořeno pomocí rozdělení modelu na dva segmenty, značící hradby jako takové a prostor pro cimbuří. Toto rozdělení bylo provedeno zkratkou *Ctrl + R (Loop-Cut)*. a následně část pro hradby vytažena funkcí *Extrude*. Takto vytažená strana byla následně rozdělena na další, pravidelné segmenty znovu zkratkou *Ctrl + R (Loop-Cut)*. Dále bylo nutné označit nově vzniklé hrany a pomocí zkratky *Ctrl + B (Bevel)* roztáhnout nově vzniklý segment do požadovaného rozměru. Tyto segmenty byly následně smazány, díky čemuž vznikl pravidelný ráz hradby. Následně byla přidána šindelová střecha, která byla tvořena stejně, jako u brány.

Vnitřní část hradeb byla modelována obdobně jako věž s bránou. Zde se ovšem nejednalo o základní kostku, ze které byla věž tvořena, ale o cylindr. Tomu bylo po přidání do programu nastaveno 9 stran, a to z důvodu, že z dochovaných pozůstatků a vykopávek víme o devítistranné hradbě vnitřního hradu. Na hradby následně bylo přidáno dřevěné cimbuří, tvořeno stejně jako dřevěné patro brány.



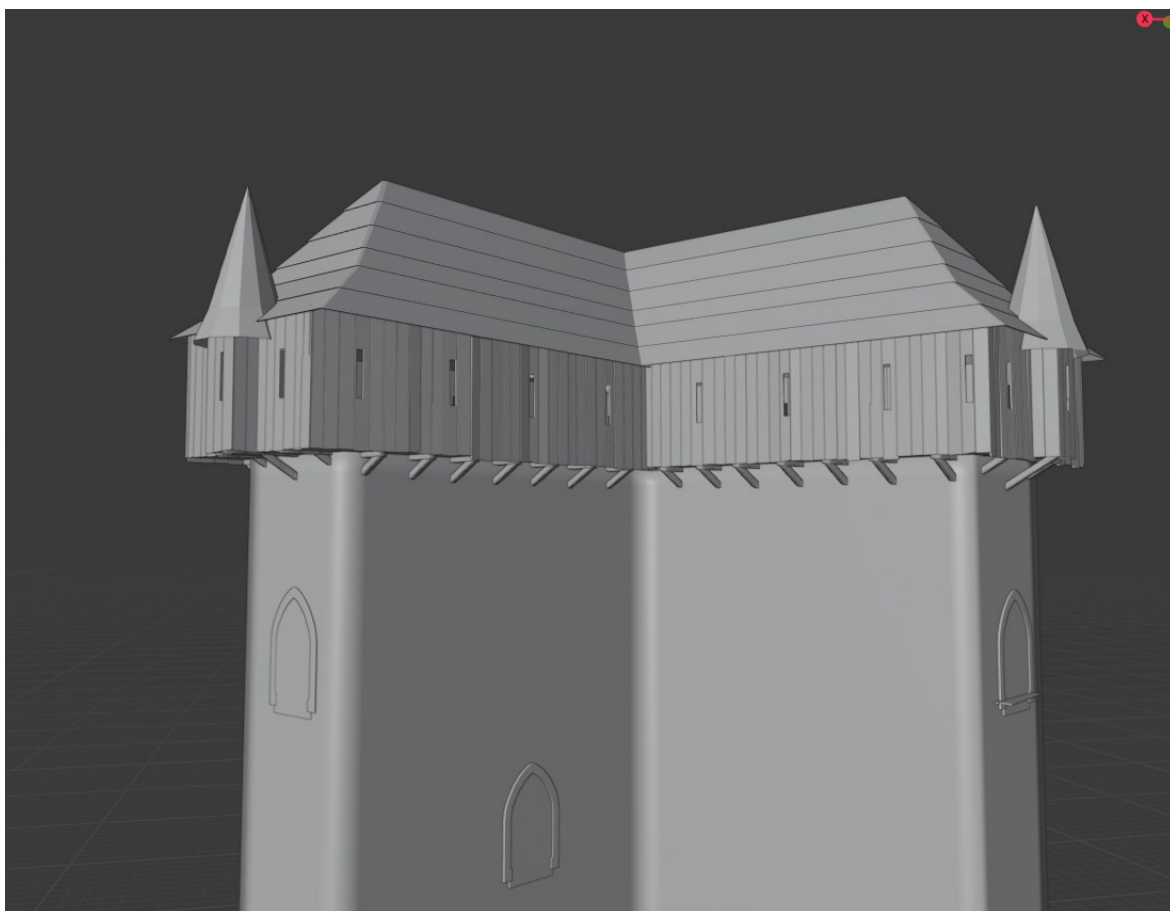
Obrázek 16 – Ukázka modelu vnější hradby ve Wireframe módu

### 5.3 Ostatní budovy hradu

U ostatních budov a částí vnitřního hradu se jednalo o totožný postup modelování. Byly modelovány menší stavení pro běžné obyvatelstvo i hlavní budova hradu. Podle dostupných nálezů a literatury lze usuzovat, že hlavní budova hradu měla jedno, až dvě patra nad přízemní částí budovy a vystupoval tedy až nad vnitřní část hradeb se svou šindelovou střechou.

### 5.4 Obranná věž

Druhou z dominantních věží byla věž obranná. K této stavbě byl dodán pouze jeden náčrt, a tak je její podoba do značné míry odhadována. Při tvorbě bylo nutné pomýšlet na fakt, že byla věž z obou stran spojena hradbami, které sahali do určité výšky. Věž samotná byla modelována obdobným stylem jako kamenná a dřevěná část brány. Jednalo se o upravování jednoduchých objektů pomocí funkcí *Scale*, *Extrude*, *Rotate* a modifikátorem *Boolean*.



Obrázek 17 – Model obranné věže před texturováním

## 6 PŘEDHRADÍ A PODHRADÍ

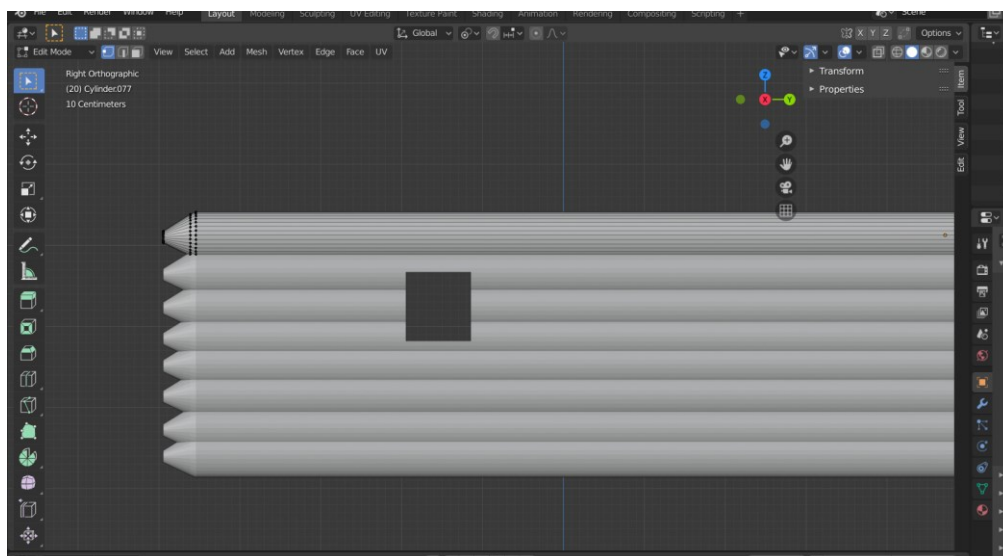
Díky dokumentům pana Vrly bylo možné určit přibližnou podobu předhradí, jednalo se o stupňovité, dřevěné opevnění obehnané palisádou. Toto opevnění bylo tvořeno pomocí jednoduchých, válcovitých objektů, které byly roztaženy pomocí funkce *Scale* do potřebného rozměru a duplikovány zkratkou *Shift + D*.

Z informací, které byly získány rešerší dokumentů o středověkém životě bylo dále zjištěno několik poznatků o životě a obydlí prostého lidu v podhradí. Obydlí bývalo zpravidla dřevěné a roubeného charakteru, obdobně jako ostatní stavení. Střechy bývaly doškového, nebo šindelové. V okolí domů se následně nacházely hospodářská stavení, jako například stodoly, malá pole či různé přístřešky. Hospodářská stavení, jako stodoly, přístřešky i drobné modely byly modelovány obdobným způsobem jako většina modelů v této práci

### 6.1 Modelování budov

Před samotným modelováním bylo příhodné najít plán, či náčrt roubené budovy pro lepší zachování proporcí a rozměrů daného objektu. Daný obrázek byl do Blenderu vložen pomocí klávesové zkratky *Shift + A* a z následné nabídky bylo vybráno *Image -> Reference*. Takto vložený obrázek byl následně upraven roztažením do správné velikosti a vše bylo připraveno k modelaci. Roubený vzor budovy byl docílen pomocí stejné klávesové zkratky jako u vložení obrázku, ovšem zde bylo nutné přejít do podsložky *Mesh* a vybrat *Cylinder*. Takto vložený *Cylinder* bylo následně nutné otočit pomocí klávesy *R* o devadesát stupňů podle osy *X* a následně roztáhnout klávesou *S(scale)* do požadované délky. Konce těchto válců byly následně v editačním módu zúženy a roztáhnuty kombinací kláves *E (extrude)* pro vytažení a *S (scale)* pro změnu velikosti. Takto upravený válec, který má připomínat jednu z klád roubené budovy byl následně rozkopírován modifikátorem *Array* a to na ose *Z*. Tento modifikátor slouží k rozkopírování daného objektu podle určité osy či křivky. Takto byla postavena jedna z postranních stran budovy. Přední strana byla vytvořena pomocí zkopírování dané postranní stěny a zmenšení pomocí klávesy *S*. Tato zmenšená strana budovy byla následně přemístěna do popředí klávesou *G* a otočena o devadesát stupňů klávesou *R*. Takto vytvořena přední a boční strana byla následně zkopírována a otočena o sto osmdesát stupňů, čímž se docílilo ucelení celé budovy. Okna a dveře byly do modelu přidány pomocí modifikátoru *Boolean*, který vytvoří do objektu otvor ve tvaru objektu vybraného. Zde byly vybrány krychle a kvádry o rozměrech určených pro okna a dveře.

Přidat obrázek reference s citací na zdroj.



Obrázek 18 – Tvorba roubené zdi

U doškové střechy se jednalo o postup, kdy byl vložen pomocí zkratky *Shift* + *A* mesh se jménem *Plane*. Jedná se pouze o plochu jednoho polygonu o 4 hranách. Tento objekt byl následně roztažen do požadované šířky klávesou *S*. Následně byla jeho vrchní hrana vytažena o určenou výšku pomocí *E* (*extrude*). Tento způsob se opakoval až do dosažení schodovitého tvaru střechy. Takto upravená střecha byla následně zkopírována a otočena o sto osmdesát stupňů. Následně bylo střechu nutno vytáhnout do výšky pomocí klávesy *E*. Dále byl použit modifikátor *Subdivision Surface*, který střechu zaoblil do požadovaného tvaru. Pravidelnost doškové střechy byla následně narušena v záložce *Sculpting*, kde se pomocí štětce *Draw* střecha uzpůsobila do nepravidelného tvaru.

Šindelová střecha byla tvořena do jisté míry obdobným způsobem, ovšem s tou změnou, kdy po vymodelování „schodovitého tvaru“ střechy byl daný model hotov.

## 7 TEXTUROVÁNÍ

Jakmile je samotný model hotov, prochází procesem texturování. Jedná se o postup, který je u všech modelů obdobný. Na dosud čistý model, či jeho část, se nanese materiál obsahující textury s různými druhy informací. Jedná se například o Albedo texturu, Roughness texturu či Normal mapu. Albedo určuje základní barvu materiálu, v podstatě jde o fotografii určitého reálného materiálu, například dřeva bez stínů. Roughness určuje odrazivost světla daného modelu neboli jak moc se bude daný materiál lesknout. Normal mapa pak určuje nerovnost povrchu, například různé výstupky či poškrábání daného materiálu.

Pro správné nanášení textur a materiálů s nimi spojených na model je nutné vytvořit takzvané UV mapy, které se vytvářejí přímo v programu pro 3D tvorbu, v tomto případě v programu Blender. UV mapy rozdělují jednotlivé části modelu na segmenty podle předem určeného mapování. Pro lepší představu si je možno představit rozloženou papírovou krabici při pohledu shora. Jednotlivé části jsou propojené, ovšem rozložené na ploše vedle sebe. Obdobně tomu funguje UV mapping. Jednotlivé segmenty je následně možné přesouvat či otáčet podle nutnosti a potřeby pro co nejlepší otexturování.

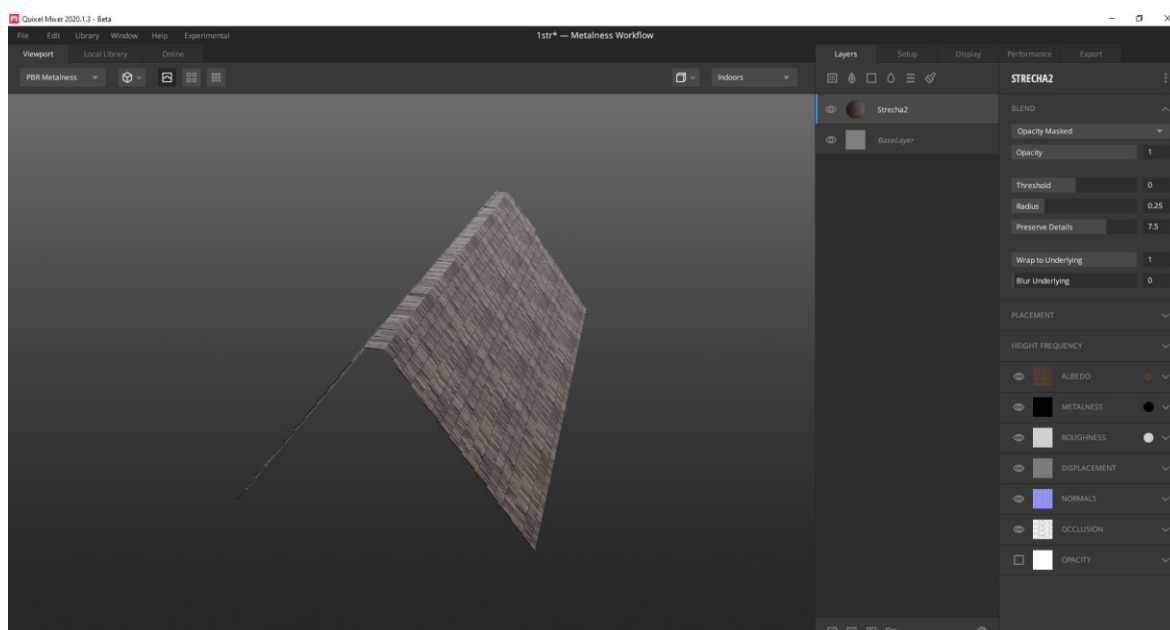
Textury samotné se dají získat mnoha způsoby. Lze si je pomocí kvalitního fotoaparátu a softwarů k tomu určeným vytvořit, nebo stáhnout z mnoha webových portálů ať už zdarma, či zakoupit za často paušální poplatek. V poslední době se ovšem v prostředí 3D grafiky a texturování obecně rozmohl nový fenomén. Jedná se o takzvaný 3D texture painting, neboli o způsob kreslení textur přímo na 3D model. Princip je takový, že se do softwaru pro tuto práci určeného, naimportuje 3D model s již vytvořenými UV mapami. V daném softwaru se zvolí různé materiály, které se mohou následně také kombinovat a pomocí různých štětců se nanášejí na daný model. Poté, co je „nabarvení“ modelu dokončeno, se materiál takzvaně zapeče, jedná se o proces, kdy se jednoduše řečeno ze všech použitých materiálů stane jeden materiál kombinující prvky materiálů ostatních. Pro tuto práci jsou všeobecně známy především dva softwary, jedná se o softwary Substance Painter a Quixel Mixer. Druhý z jmenovaných byl použit i pro tuto práci. Jedná se o software, pro který je, ve spolupráci s Unreal Enginem, veškerý obsah materiálů i naskenovaných 3D modelů zdarma.



## 7.1 Aplikace textur

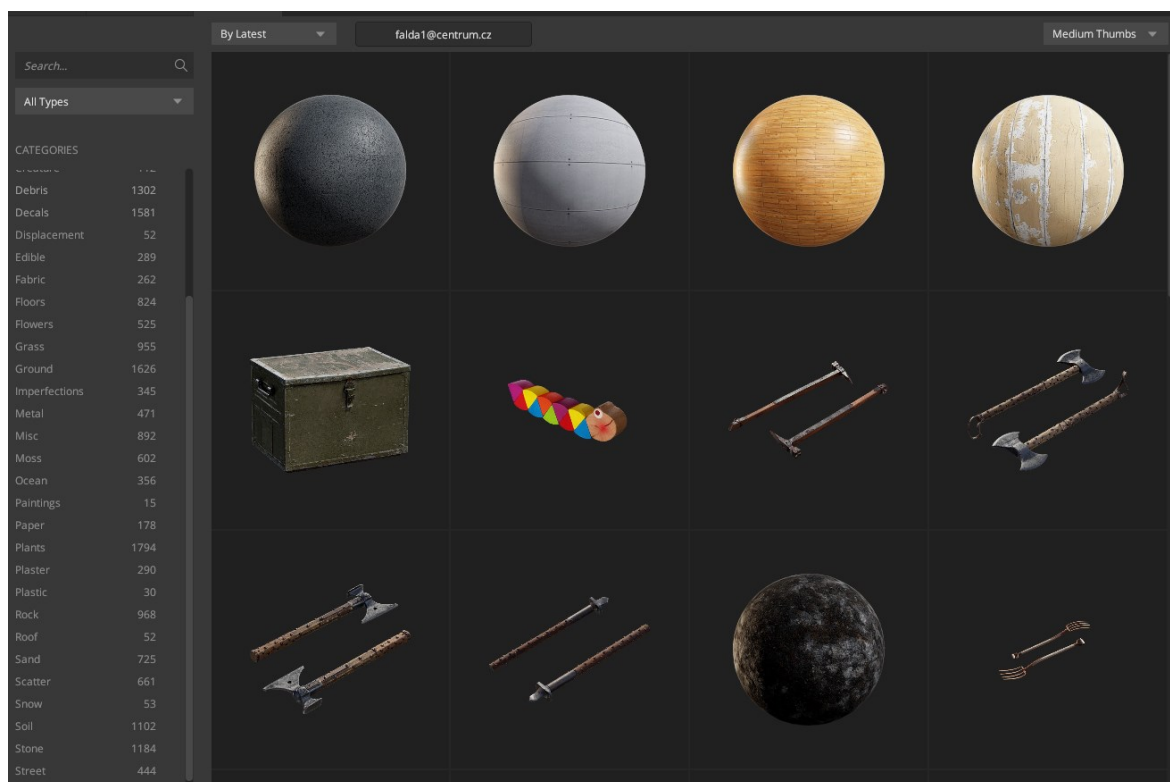
Po vytvoření celého modelu bylo nutné pro jednotlivé části vytvořit *UV mapy* pro následné nanesení textur. Toto se vytváří jednotlivě, na každém modelu zvláště. V editačním módu daného modelu jsou označeny veškeré plochy pomocí klávesy *A*. Následně je klávesou *U* vyvoláno *UV mappingové* menu, ve kterém se nachází vícero možností provedení tohoto mapování. Pro většinu primitivních a jednoduchých objektů byla použita funkce *Smart UV Project*. Jedná se o funkci, která automaticky vytvoří *UV mapování*, jedná se tak o nejrychlejší možnost, která je přesná převážně na jednoduchých objektech. Pro složitější objekty je nutné použít manuální mapování, které se provádí pomocí vybrání hrany objektu a použití funkce *Mark Seam* v mapovacím menu. Tato funkce rozdělí *UV mapu* v dané hraně nad dvě části.

Po dokončení *UV mapování* jsou modely připraveny k aplikaci textur. Toto aplikování lze dělat mnoha způsoby ať už v Blenderu či externích softwarech pro to určených. Pro tuto práci byla zvoleno texturování v externím programu Quixel Mixer. Jedná se o software, ve kterém lze nanášet různými štětci textury přímo na 3D model což umožňuje lepší představu o tom, jak finální model bude vypadat, urychluje čas texturování a také se zde lépe nanášejí detaily. Export objektu z Blenderu do Quixel Mixer by proveden pomocí výběru složky *File*, následně možnost *Export* a zde byl vybrán formát *FBX*, se kterým zmíněný Quixel Mixer umí pracovat.



Obrázek 19 – Prostředí Quixel Mixer

Po spuštění programu Quxiel Mixer bylo nutné vytvořit nový project pomocí tlačítka „new mix“, to následně otevřelo rozhraní po tvoření textur. Zde bylo následně nutné v *sekcí Model settings* vybrat v položce *Type* možnost *Custom Mesh*. Tato volba umožňuje nahrání vlastního modelu do programu. Stačilo tedy vybrat model z místa uložení. Textury byly nanášeny v záložce *Layers*, do kterých byly stahovány z online knihovny. Díky práci v Unreal Engineu byly veškeré textury dostupné zdarma. Každá z vrstev má mnoho funkcí, které se při nanášení textur mohou používat. Jedná se například o funkce jako prolínání, přidání barvicí masky či duplikování dané masky. Právě pomocí barvicí masky bylo možno nejlépe nanášet jednotlivé detaily na model. Po nanešení potřebných textur byly textury následně zapečeny a exportovány do předem určené složky pro uložení. Pro tento krok byla využita záložka *Export*, ve které lze nastavit i druhy map, které se mají exportovat. Takto exportované textury bylo dále možno použít jak v Blenderu, tak následně i v Unreal Engineu pro finální úpravy. Obdobný proces byl použit u všech modelů v této práci.



Obrázek 20 – Online knihovny s materiály a modely

## 8 TVORBA SCÉNY

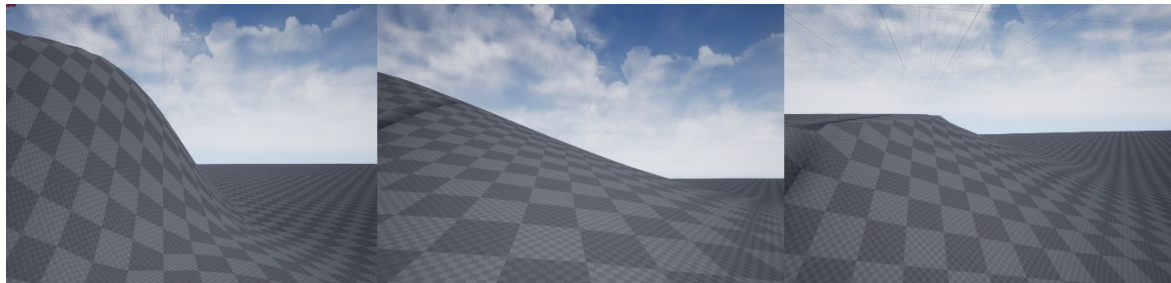
Po vytvoření všech modelů je bylo nutné zakomponovat do jedné, rozsáhlé scény. Ač je tato kompozice možná i v programu Blender, ve kterém byly modelovány veškeré objekty, existují softwary pro takovou práci určené. Takovéto softwary jsou nazývány *engine*, případně *herní engine*. Jedná se o komplexní program, zabezpečující plynulý chod zejména her, které jsou v nich tvořeny nejčastěji. Program samotný se stará o veškeré zázemí těchto her, ať už se jedná o simulaci fyzikálních jevů jako jsou poryvy větru, nasvícení scény, fyziky samotné, či vytvoření animací, *renderu* nebo aplikace samotné. Těchto enginů je celá řada, k nejznámějším patří Cry Engine, Unity a Unreal 4. Poslední z jmenovaných byl použit i v této práci, konkrétně verze 4.24.

### 8.1 Terén

Terén v Unreal Engine 4 lze vytvořit několika způsoby. Je zde možnost naimportování takzvaných výškových map, které jsou schopné vygenerovat výškový ráz krajiny z reálných dat. Tento postup byl testován i v této práci, ovšem z důvodu nekvalitní výškové mapy bylo od tohoto postupu upuštěno. Bylo tedy nutné krajinu vytvořit ručně. K tomuto postupu slouží v sekci *Modes* záložka *Landscape*. V této záložce bylo možno nastavit rozlohu následně generované krajiny, dále počet segmentů, ze kterých byl model krajiny tvořen, i poloha ve které bude následný model generován. K vygenerování krajiny došlo po stisknutí tlačítka *Create*. Engine následně vygeneroval plochu mapu dle předem zadaných parametrů. Tato plocha se následně musela vymodelovat do potřebných tvarů a nerovností v takové míře, aby co nejvíce odpovídala reálnému prostředí v okolí hradu

V Unreal Engine 4 je možné pomocí mnoha nástrojů editovat již vytvořenou plochu do potřebných tvarů a velikostí. Jedná se převážně o klasický *sculpting*, při kterém body v daném místě mění svou polohu dle zvoleného nástroje. Zde se jednalo nejčastěji o nástroje *Sculpt*, *Smooth* a *Flatten*. Nástroj *Sculpt* je zpravidla používán jako první, a to z důvodu vytvoření hrubého rázu upravované plochy. Jedná se o nejrychlejší a nejjednodušší způsob, jak vytvořit základ daného tvaru. Pro přidání reálnější podoby a plynulosti přechodu je zde pak nástroj *Smooth*, Tento nástroj zaručuje plynulý přechod mezi dvěma plochami. Jedná se tak o nástroj hojně využívaný v kombinaci s nástrojem *Sculpt* pro vytvoření například kopců, kdy je nejdříve nástin kopce vytvořen nástrojem *Sculpt* a následně použita funkce *Smooth* pro zjemnění rázu krajiny a přidání reálného rázu.

Třetí z používaných funkcí byla funkce *Flatten*. Ta vytváří z nerovných ploch plochy rovné. Používá se tak například pro vytvoření plochy pro budovy a jiné objekty, které musí stát na rovném povrchu. Pomocí těchto tří funkcí byla následně tvořena většina krajiny.



Obrázek 21 – Funkce *Sculpt*, *Smooth*, *Flatten*

### 8.1.1 Získání terénních informací

Pro získání informací o terénu v dané lokalitě by bylo nejrychlejším a pro následnou implementaci i nejjednodušším způsobem získat takzvanou výškovou mapu z reálných dat dané lokality. Z důvodu nekvalitních výškových map, které byly k dispozici, od tohoto posupu bylo upuštěno. Bylo tedy nutné přejít k manuálnímu sběru informací o dané lokalitě. Tento postup není tak přesným, jako jsou výškové mapy, ale za to dostupným řešením. Pro tuto práci byly zvoleny dva postupy manuálního sběru informací. Prvním postupem byla osobní návštěva dané lokality. Jedná se o kopcovitý terén u obce Podhradí, nedaleko Luhačovic. Už zprvu tak bylo patrné, že se nebude jednat o rovnou plochu, ale o kopcovitý terén.



Obrázek 22 – Dosud patrné terénní úpravy

Po vystoupení poměrně strmého kopce bylo zjištěno, že se do současné doby dochovaly dostatečně patrné terénní úpravy, které jsou zřejmě nejzachovalejší částí dříve rozsáhlého hradu. Zde bylo možné vytvořit desítky fotografií ať už terénních úprav, rostlin a kamenů v okolí, nebo celkového rázu krajiny. Z těchto nabraných referencí se následně při tvorbě terénu v Unreal Engine čerpalo v hojném počtu a krajina tak mohla nabrat reálnějšího rázu. Druhým z postupů manuálního zisku informací bylo využití Google Maps, konkrétně jejich 3D zobrazení. Díky tomuto zobrazení, které simuluje reálné, výškové informace dané lokality, bylo možné využívat, při manuálním modelování, pohledy z různých úhlů.



Obrázek 23 – Detail terénní úpravy

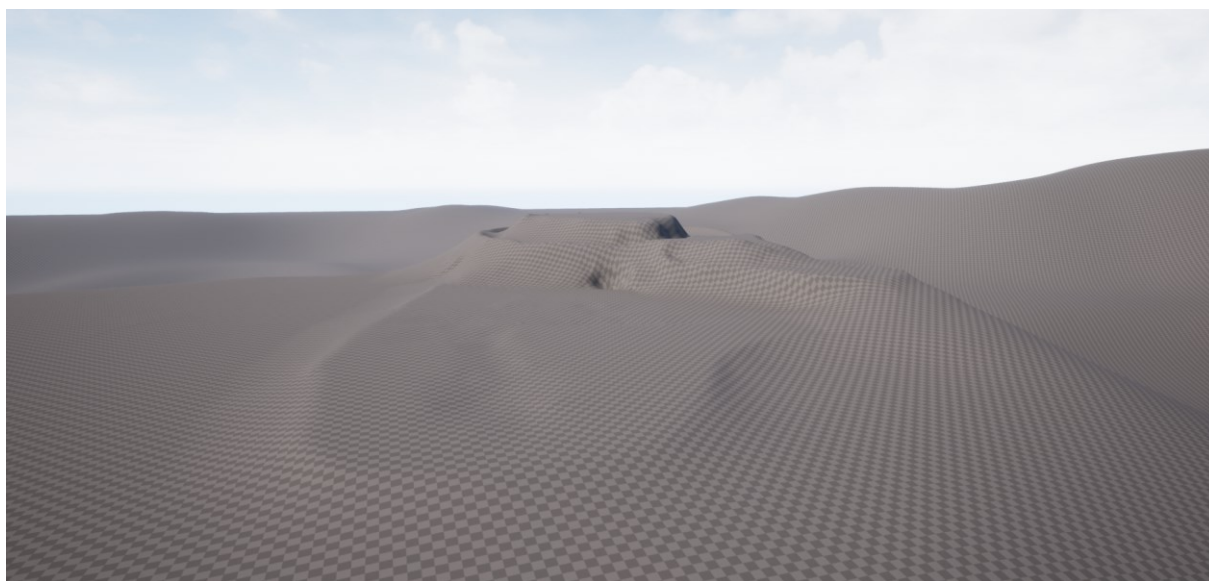
### 8.1.2 Tvorba terénu

Jak bylo zmíněno, v dostupných zdrojích nebylo možné nalézt kvalitně zpracovanou výškovou mapu, kterou by bylo možné použít pro generování mapy okolí hradu v Unreal Engine. Z tohoto důvodu bylo nutné prostředí vymodelovat manuálně. Manuální tvorba prostředí byla tvořena převážně pomocí nástrojů popsaných dříve, jedná se o nástroje v záložce *Landscape*, kde byla vybrána podzáložka *Sculpt*.

V této záložce se kromě zmíněných funkcí nachází i nastavení jednotlivých štětců. Jedná se převážně o nastavení velikosti, hrubosti a přechodu štětce. Před úpravou krajiny bylo nejdříve nutné vygenerovat plochu, na které bude následně tvořena krajina.

Pro toto generování bylo nutné v záložce *Landscape* nastavit základní parametry. Hodnoty pro *Location*, *Rotation* a *Scale* byly nechány v základních hodnotách. Změna proběhla až u položky *Section Size*, kde byla vybrána velikost 127x127. Toto rozložení zaručilo dostatečně velké prostředí pro danou krajinu, ostatní parametry byly také ponechány v základních hodnotách, a tak bylo možné mapu generovat. Po vygenerování dané mapy byl vytvořen, pomocí funkce *Sculpt*, základní nástin krajiny v prostorách hradu.

Jako první byl vytvořen prostor pro vnitřní hrad. Zde byla navolena síla štětce na 0,3 a velikost zhruba na 2000 bodů. Po vytvoření dostatečného prostoru pro hrad bylo se stejnými parametry vytvořeno také předhradí. Štětec byl následně zmenšen zhruba na 650 bodů a síla snížena na 0,2. Při stisknutí klávesy shift a použití se docílilo opačného efektu, tedy terén se nezvyšoval, ale snižoval. Tím bylo možno vytvořit snižující se ráz předhradí. S tímto nastavením byl dále vytvořen i val obíhající celý hrad. Ve chvíli, kdy byl terén takto upraven, mohlo se přejít k detailnějším úpravám pomocí funkce *Smooth*. Ta vytvořila reálnější ráz kopců a přechodů mezi terénními nerovnostmi. Po vyhlazení terénu byla použita funkce *Flatten*, která následně vytvořila rovné plochy pro umístění budov a ostatních modelů hradu.



Obrázek 24 – Vymodelované prostředí v Unreal Engine

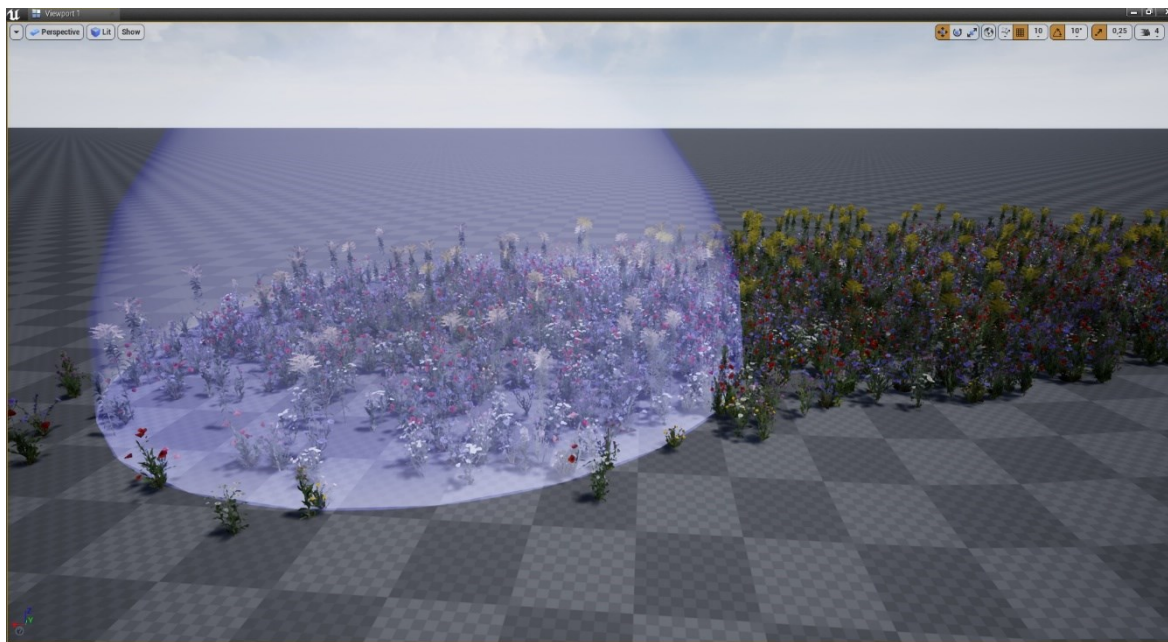
## 8.2 Vkládání objektů

Po vytvoření základního prostředí bylo možné začít s vkládáním jednotlivých modelů do scény. Toto vkládání objektů bylo prováděno v záložce *File* při výběru možnosti *Import into level*. Následně byl vybrán soubor formátu *fbx* s daným modelem z místa uložení. Po výběru souboru bylo nutné nastavit vstupní informace. Převážně v sekci *Import Options* vybrat možnost *Import as Dynamic*, což umožňuje pohybovat s jednotlivými částmi daného modelu ve scéně. Dále bylo nutno vybrat možnost *Preserve Smoothing Groups* v záložce *Skeletal Meshes*, toto zaručuje zachování hladkých hran modelu, které byly vytvořeny v programu Blender. Následně byl daný model přidán tlačítkem *Import* do scény, ve kterém s ním bylo možno pohybovat a umístit ho tak na vybrané místo. Pro lepší manipulaci byly modely hradu a předhradí skládány do větších skupin již v programu Blender a následně exportovány jako jednotný *fbx* soubor. Bylo tak možné rychleji nakládat s větší oblastí scény najednou. Po vložení všech základních částí hradu a předhradí bylo následně dopraveno prostředí okolí hradu tak, aby přesně odpovídalo rozložení modelů.

## 8.3 Vegetace

Vegetace je nedílnou součástí každé krajiny a do velké míry určuje její ráz a atmosféru. Stejně tak tomu bylo i v této práci, kde se hrad nachází v kopcovité krajině porostlé hustými lesy. Stejně tak i travnaté louky v nejbližším okolí hradu byly důležitou součástí atmosféry. Pro značné urychlení práce nebyly modely stromů, ani travnatého porosty modelovány, nýbrž získány z online knihovny Quixel Bridge či Marketplace pro Unreal Engine. V těchto knihovnách lze získat zdarma, či za poplatek různé typy vegetace s volnou licencí k užití. Výhodou získání modelů touto formou bylo jak ušetření velké dávky času, tak i získání kvalitních modelů již s nastavenými LOD verzemi. Jednalo se tedy o modely v rozmezí od detailních až po 2D plochy, které se v dálce natáčeli dle kamery a simulovali tak dojem prostorového zobrazení. Nanesení vegetace se do jisté míry podobá úpravám prostředí. Jedná se o práci v záložce *Foliage*, do které byly nejdříve naneseny modely trávy a stromů. Ty jsou nejdříve vybrány v sekci *Content Browser* a následně přetaženy do záložky sekce části *Drop foliage here* záložky *Foliage*. Po tomto přetažení bylo vytvořeno výběrové menu, ve kterém bylo možné pro jednotlivé typy vegetace nastavit v jaké míře, hustotě, lokaci či velikosti se má v daném štětku generovat

Bylo zde možno nastavit také rozmezí velikostí i rotací pro zachování co nejreálnějšího dojmu a náhodnosti. Štětec byl následně nastavován obdobně, kdy byla možnost nastavit velikost štětce či hustota generování vybraných modelů. Stejným způsobem lze do scény nanášet veškeré statické modely. V této práci tak byly umisťovány také jednotlivé kůly palisádové hradby.



Obrázek 25 – Nanášení vegetace pomocí štětce

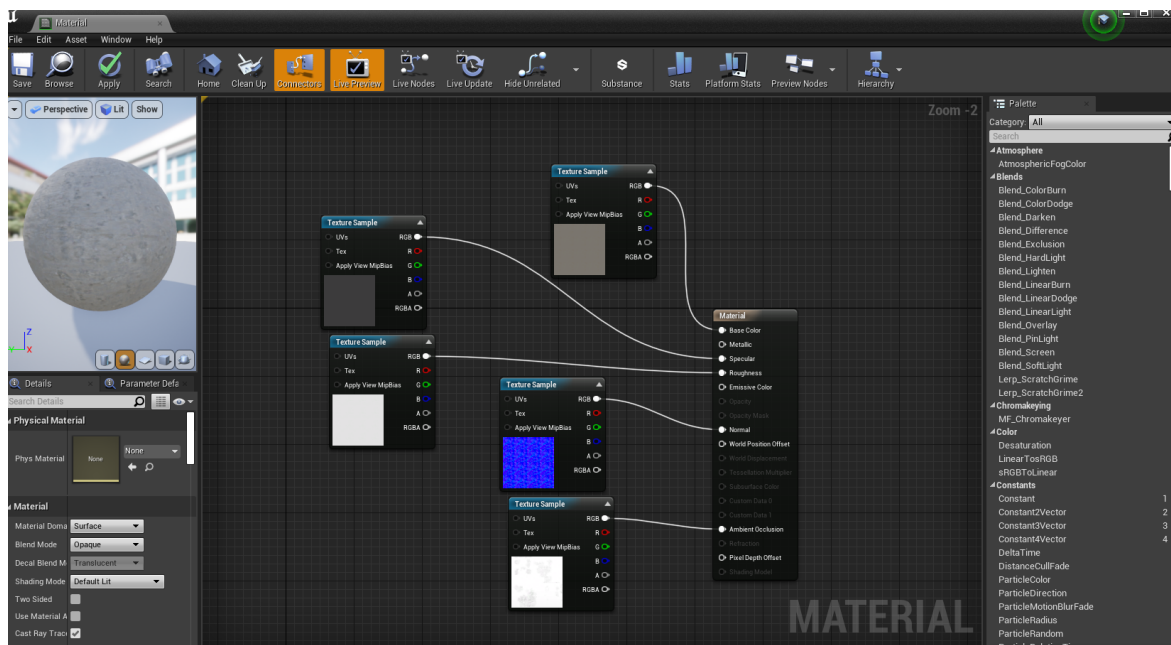
## 8.4 Materiály

Stejně jako vegetace rozhoduje o rázu a atmosféře krajiny, rozhodují materiály o finálním dojmu celé scény. Jelikož je lidské oko velmi citlivé a jevy a prvky, které zná z každodenního života, lehce rozpozná, zda je například materiál dřeva rozmázlý, moc lesklý či například moc roztáhlý. K potlačení takových jevů slouží PBR materiály. Tyto materiály jsou hojně využívány i v Unreal Engine, kde tak dodávají co nejrealističtější vzhled celé scény.

Materiály lze v Unreal Engine přidat několika způsoby. Základním způsobem je vytvoření materiálové instance v *Content Browser* pomocí pravého tlačítka myši a vybráním položky *Material*. Tímto se vytvoří prázdná materiálová složka. Po rozkliknutí této složky je otevřen materiálový editor, do kterého lze importovat z *Content Browser* jednotlivé mapy daného materiálu. Ty jsou následně propojeny pomocí takzvaného „nodového“ systému, který je obdobný jako například v programu Blender. Dokončení tvorby materiálu je následně zpečetěno uložením daného materiálu pomocí tlačítka *Save*.



Po dokončení tvorby materiálu lze daný materiál na vybraný objekt přiřadit tím způsobem, že se v sekci *Details* daného objektu vybere z rozbalovacího menu název požadovaného materiálu. Tímto se automaticky materiál nanese na objekt.



Obrázek 26 – Node systém materiálu

Dalším způsobem přidání materiálu, který byl také hojně využíván v této práci, je zachování materiálu při importu *fbx* souboru. Materiál z *de* není plně zachován s veškerými mapami, avšak je zde již vytvořena materiálová instance a přidělení materiálu na daný model, nebo jeho část. Po přidání map z *Content Browser* a následné aplikace změn materiálu, je tak daný materiál přímo nanesen.

Obdobným způsobem byl nanášen materiál i na povrch krajiny, zde se ovšem jednalo o složitější zapojení materiálu, a to z důvodu využití více textur v jednom materiálu. Tento materiál se nazývá chytrým materiálem, jedná se o materiál, který díky nodovému zapojení „rozpozná“ terénní nerovnosti a podle jejich výšky sám nanese danou texturu. Lze si tak představit rovnou plochu jako travnatou louku a po použití funkce *Sculpt* v úpravách terénu zde vznikne kopec s kamenitým materiálem. Tyto textury jsou následně, pomocí stejných nodů míchány, aby nedocházelo k takzvanému boxovému jevu. Jedná se o jev, kdy se nanáší stejná vrstva vedle sebe a vytváří tak repetitivní obraz. Takto vytvořené vrstvy je následně možné v podzáložce *paint*, záložky *Landscape*, nanášet pomocí štětce přímo na povrch daného prostředí. Na povrch tak lze malovat cesty a různé detaily.



Obrázek 27 – Ukázka materiálu prostředí již s vegetací

## 8.5 Nasvícení scény

Osvětlení je největším tvůrcem atmosféry dané scény. Lze jím natavit roční období, denní dobu i podnebí v dané lokaci. Nastavení správného světla může být také ale velká alchymie. Unreal Engine je schopen pracovat s několika zdroji světla najednou a nabízí několik druhů světelného záření, jako jsou například bodová a směrová světla, či svit oblohy. V této práci bylo použito světlo směrové společně se svitem oblohy. Tato kombinace byla zvolena pro co nejrealnější simulaci slunečního svitu. Svit oblohy byl přidán z toho důvodu, že ani v reálných podmínkách není slunce jediným světlem, ale jeho paprsky se odrážejí jak od předmětů na zemi, tak i atmosféry okolního světa. Směrové světlo bylo nejdříve přepnout v sekci *Transform* ze *Static* na *Movable*. Tím bylo možno s daným světlem libovolně pohybovat a rotovat pro lepší umístění a úhel „slunečního svitu“. Po najetí správného umístění bylo dále nutné v sekci *Light* zvolit správnou intenzitu světla a povolit „světelnou teplotu“ pro co nejrealnější simulace slunečního záření.

## 9 VÝSTUPY

Hlavními výstupy této práce jsou renderované snímky a animace 3D modelu hradu Starý Světlov v jeho možné podobě z 15. století. Tyto snímky budou nabídnuty Národnímu památkovému ústavu a obci Podhradí, do jejíž správy současná zřícenina patří, pro obeznámení veřejnosti s tímto již zaniklým hradem. Dalším výstupem je aplikace, ve které si lze daný hrad projít z pohledu, který se mohl naskytnout obyvatelům daného místa během 15. století.

### 9.1 Rendery

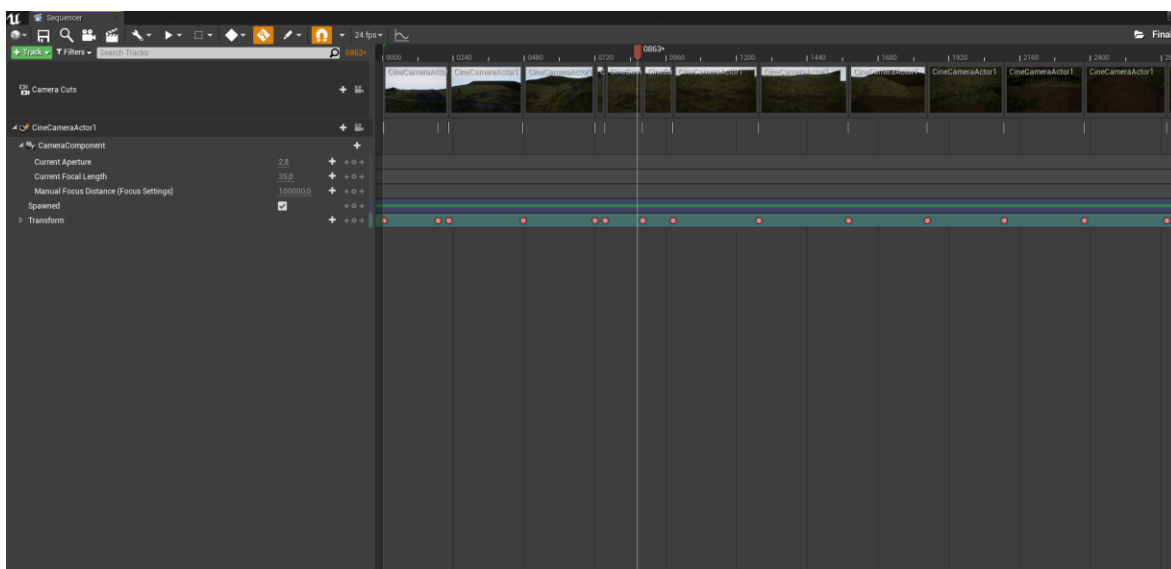
Jakmile byla scéna v Unreal Engine připravena, přišlo na řadu rozložení kamer do scény. Ty by měly zobrazovat jak detailní záběry na jednotlivé části hradu, předhradí, či stavení v podhradí tak i celkový pohled na celý hrad a jemu přidružené části. Rendery byly pořizovány přímo v Unreal Engine, který využívá realtime renderování a je tak možno kameru i prostředí nastavit bez jakýchkoliv zdržení. V enginu jsou dva typy kamer, *CameraActor* a *CineCameraActor*. Po nastavení hodnot, pozice a úhlu kamer bylo možné přidat funkci *Post Process Volume*, ve které lze nastavovat klasické, postprodukční hodnoty jako jsou například expozice, teplota světla, nebo barevné vyvážení. Po těchto všech úpravách bylo možno vytvořit samotný render. Ten se vytváří v okně viewport pomocí rozbalovacího menu, ve kterém je následně vyprána položka *High Resolution Screenshot*. Zde je možnost výběru velikosti výsledného obrázku a následně daný render vytvořit.



Obrázek 28 – Zakomponovaná obranná věž

## 9.2 Animace

Pro tvorbu animace bylo nutné do scény vložit kameru, díky které bylo možné animaci takzvaně rozklíčovat. Jedná se o postup, při kterém se na časové ose zaznamenávají klíčové hodnoty kamery. Jedná se především o pozici a rotaci kamery, dále však také o hloubku ostrosti či šířku záběru. Toto klíčování bylo tvořeno v záložce *Cinematics*, kde byla zvolena možnost *Create Sequence*. Po tomto kroku byl otevřen editor, podobný klasickým editačním programům. Dále bylo nutné přepnout vloženou kameru do takzvaného *Cinematic Mode*, kde byla zobrazena výsledná podoba záběru. Jakmile bylo vše nastaveno, stačilo již jen pohybovat s kamerou na místa, která by nejlépe vyhovovala znázornění hradu. Při dosažení takovéto pozice byl pomocí klávesy *S* vytvořen klíčový bod. Čím dále jsou od sebe jednotlivé klíčové body, tím pomaleji se mezi nimi kamera bude pohybovat. Po vyklíčování animace byl nastaven render. Zde bylo nastaveno *Full HD* rozlišení při 24 snímcích za sekundu. Jakmile byl render nastaven, mohlo se přejít pomocí tlačítka *Create* k jeho vytvoření. Po několika minutách byl render vytvořen do předem vybrané složky. Z této složky bylo možno vyrenderované video importovat do programu Da Vinci Resolve 16, kde byla upravena barevnost videa.

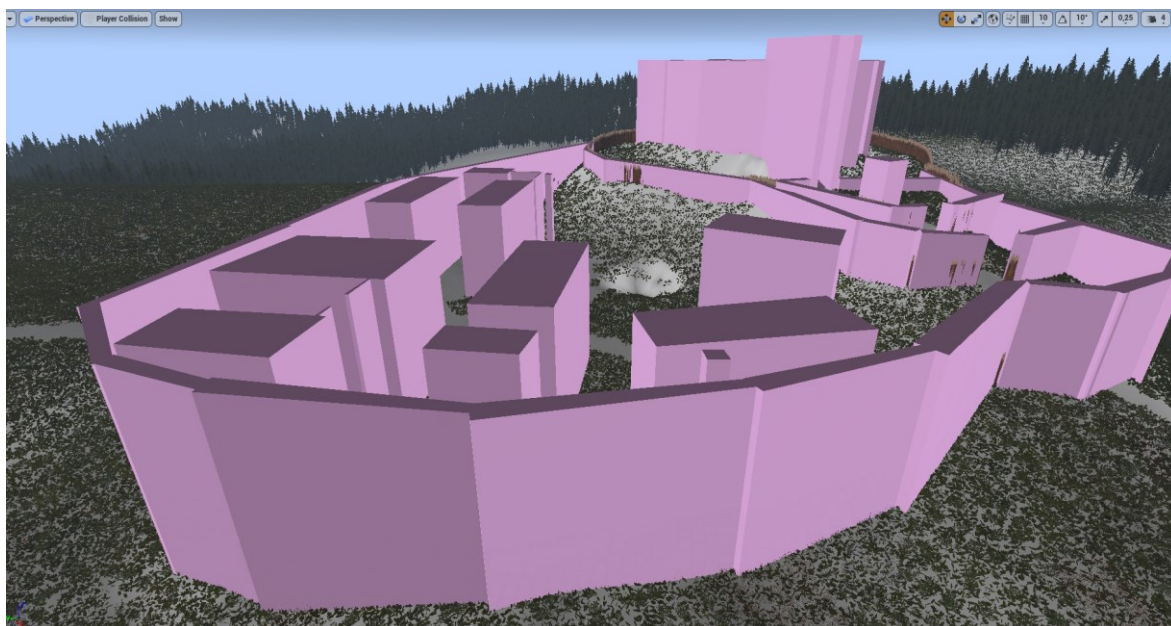


Obrázek 29 – Video editor v Unreal Engine

### 9.3 Aplikace

Po vytvoření celé scény, včetně vegetace a nasvícení, přišla na řadu tvorba aplikace. Nejdříve bylo nutné nastavit kameru. Jako šablona pro tento projekt byla zvolena takzvaná „třetí osoba“, jedná se o druh šablony, která obsahuje postavu s kamerou. Kamera je uzamčena zhruba tři metry za danou postavou a snímá tak jak jí, tak nejbližší okolí. Tento druh pohledu byl změněn na takzvanou „první osobu“. Zde je kamera umístěna a uzamčena na hlavě daného modelu, pohybuje se tedy s celým tělem a uživatel tak vidí okolí z výšky očí dané postavy.

Po nastavení kamery přišlo na řadu určit, kam daná postava bude smět, případně kam nebude smět chodit. Toto omezení je důležité, hlavně pro udržení „návštěvníka“, neboli dané postavy v lokaci pro to určené. Veškeré objekty, které jsou vloženy do Unreal Engine jsou zpravidla v základním nastavení průchozí. Po „pevnost“ se jim musí nastavit takzvaná kolize. Kolize zabraňuje buďto postavám, či kameře průchod danou oblastí, či daným objektem. Ten tak působí reálně. Dají se nastavit pomocí takzvaných kolizních bloků, které si lze představit jako průhledné, či neviditelné krychle, které určují, v jakém místě bude stát „neviditelná zeď“ a kudy se tedy nebude dát projít. Dále lze kolize nastavit přímo na objektu, ve kterém se vypočítají hodnoty hran daného objektu a ten je tak neprůchozí jen tam, kde zasahuje do scény. Jedná se tak o reálnější, ale z výkonnostního pohledu také náročnější provedení. Pro tuto práci byl vybrán postup s kolizními bloky.



Obrázek 30 – Rozložení kolizních boxů

Jakmile byly kolizní bloky nastaveny, bylo možné dané prostředí otestovat, zda se někde nenacházejí místa, která by umožňovala průchod objekty, zda někde daný objekt nechybí, nebo nemá špatně určený materiál. Tohoto testování se docílí po spuštění daného levelu ikonou Play v horní liště programu. Po kliknutí na tuto ikonu se načte daný level v takové podobě, v jaké bude ve finální aplikaci. Spouštění aplikace probíhá na předem určeném místě, zde se jedná o místo u vstupní brány do podhradí. Toto místo je vybráno proto, aby si návštěvník mohl prohlédnout celý komplex od začátku.

Pokud je vše v pořádku, je možné danou aplikaci „vydat“. Toto je provedeno v záložce *File*, ve které je vybrána možnost *Package Project*, následně je vybrána platforma, pro kterou daná aplikace vzniká a verze operačního systému, v tomto případě se jedná o Windows (64-bit). Následně je vygenerována aplikace, která lze spustit již bez nutnosti spuštění Unreal Engine.



Obrázek 31 – Pohled po spuštění aplikace

## ZÁVĚR

Tato práce byla vypracována ve spolupráci s panem Radimem Vrlou z Národního památkového ústavu v Kroměříži s následující perspektivou použití.

Vizualizace hradu Starý Světlov budou použity na webových stránkách Národního památkového ústavu, kde budou sloužit jako ukázkový materiál při prezentaci hradu Starý Světlov. Vizualizace budou nabídnuty i obci Podhradí, do jejíž správy hrad spadá, jako možnost rozšíření článku o Starém Světlovu na obecních webových stránkách či použití renderů přímo na zřícenině hradu. Jak obci Podhradí, tak Národnímu památkovému ústavu bude také poskytnuta interaktivní aplikace k možnému použití v praxi.

Při tvorbě bylo velmi náročné vytvořit co nejrealističtější model hradu a jeho okolí z minima informací, či obrazových ilustrací. Velkou měrou zde vypomoh postřehy a připomínkami pan Radim Vrla z Národního památkového ústavu v Kroměříži. Samotné modelování zabralo nejvíce času převážně z důvodu velké pestrosti a detailnosti vytvářených modelů. Další velkou překážkou byla následná optimalizace celého projektu v programu Unreal Engine 4. Model hradu a jeho okolí v životní velikosti si vyžádal velké nároky na výkon počítače a také jeho uložště, které by se v budoucnu mohly snížit dalším optimalizováním. Do budoucna je možno také vytvořit aplikaci pro brýle virtuální reality. O této možnosti bylo uvažováno i u této práce, ovšem z důvodu nutnosti velké optimalizace a s ní spojené časové náročnosti, bylo od tohoto kroku upuštěno. Z důvodu velikosti výstupní aplikace dále nebylo možné nahrát soubor s aplikací do školního portálu [stag.utb.cz/portal/](http://stag.utb.cz/portal/) a byla tedy odevzdána společně se zdrojovými soubory vedoucímu této práce Ing. Pavlu Pokornému Ph.D. na DVD a flash disku.

Velkým usnadněním byla možnost využití volně dostupných materiálů i modelů. Tyto materiály a modely jsou volně k použití bez nutnosti zmiňování autora či použití pouze pro nekomerční účely. Byly brány převážně z online knihoven Quixel Bridge a Unreal Engine Marketplace a jednalo se například o modely vegetace doplňující vytvořenou scénu.

V průběhu práce byla objevena řada problémů, vyplývajících však převážně z nedostatku zkušeností. Jednalo se například o špatné vyexportování souborů z programu Blender do programu Unreal Engine 4, které následně zapříčinili nemožnost spuštění aplikace a bylo tak nutné soubory z Blenderu znovu exportovat a scénu v Unreal Engine od začátku přetvořit. Při případném pokračování na této bakalářské práci by bylo také vhodnější využít textury

a materiály v rozlišení 2K a nikoliv 4K, jak tomu bylo v této práci doposud. Zmenší se tím velikost celého souboru.

Výsledkem této práce jsou renderované snímky a animace, které znázorňují hrad Starý Světlov v jeho nejrozsáhlejší podobě během poloviny 15. století. Dále také aplikace, která umožňuje detailnější pohled a prohlídku dnes již zříceného hradu. Tyto výstupy budou předány Národnímu památkovému ústavu a obci Podhradí pro uvedení do praxe.



**SEZNAM POUŽITÉ LITERATURY**

- [1] MICHÁLEK, Ladislav. Starý Světlov. Zříceniny hradu severně od Luhačovic. 1. Hukvaldy: PUTUJME, 2013. ISBN 978-80-87897-06-3.
- [2] Historie hradu Starý Světlov. Obec-podhradi.cz [online]. Podhradí: Obec Podhradí, 2020 [cit. 2020-03-08]. Dostupné z: <https://www.obec-podhradi.cz/historie-hradu-stary-svetlov/ds-1085>
- [3] RAZÍM, Vladislav. Řeč středověké fortifikační architektury: terminologie a její souvislosti. V Praze: Národní památkový ústav, 2018. ISBN 978-80-86516-96-7.
- [4] Jak začít s vizualními efekty [online]. Praha: vfxcz, 2015 [cit. 2020-04-12]. Dostupné z: <http://vizualniefekty.cz/jak-zacit-s-vizualnimi-triky/>
- [5] 3D Model Components: Vertices, Edges, Polygons & More [online]. online: lifewire, 2020 [cit. 2020-04-12]. Dostupné z: <https://www.lifewire.com/3d-model-components-1952>
- [6] What is the difference between Low Poly and High Poly models? [online]. Bitola: cad-laif, 2017 [cit. 2020-04-12]. Dostupné z: <https://www.cad-laif.com/digital-world/low-poly-high-poly-difference>
- [7] LEVELS OF DEVELOPMENT / LEVELS OF DETAIL / LOD EXPLAINED [online]. Redstack: a2kstore, 2020 [cit. 2020-04-12]. Dostupné z: <https://www.a2kstore.com/levels-of-development-levels-of-detail-lod-explained>
- [8] Jak na textury [online]. Praha: itnetwork, 2020 [cit. 2020-04-12]. Dostupné z: <https://www.itnetwork.cz/programovani/petr/petr-3d-grafika/petr-lite-tutorial-3d-grafika-jak-na-textury>
- [9] Vertex Shader [online]. Beaverton: khronos, 2017 [cit. 2020-04-12]. Dostupné z: [https://www.khronos.org/opengl/wiki/Vertex\\_Shader](https://www.khronos.org/opengl/wiki/Vertex_Shader)
- [10] ÚVOD DO PBR #1: ZA VŠECHNO MŮŽE FYZIKA [online]. Praha: vfxcz, 2019 [cit. 2020-04-12]. Dostupné z: <http://vizualniefekty.cz/uvod-do-pbr-1/>
- [11] Rendering: Computer Graphics [online]. online: Fandom, c2020 [cit. 2020-08-06]. Dostupné z: <https://graphics.fandom.com/wiki/Rendering>
- [12] What is Rendering? [online]. Edmonton: Techopedia, 2019 [cit. 2020-08-06]. Dostupné z: <https://www.techopedia.com/definition/9163/rendering>

- [13] BLAIN, John M. The complete guide to Blender graphics: computer modeling & animation. Fourth edition. Boca Raton: Taylor & Francis, a CRC title, part of the Taylor & Francis imprint, a member of the Taylor & Francis Group, the academic division of T&F Informa, 2018. ISBN 978-1138081918.
- [14] About [online]. Amsterdam: Blender, c2020 [cit. 2020-08-06]. Dostupné z: <https://www.blender.org/about/>
- [15] Cycles. Docs.blender.org [online]. Amsterdam: Blender, c2020 [cit. 2020-08-06]. Dostupné z: <https://docs.blender.org/manual/en/dev/render/cycles/>
- [16] Eevee. Docs.blender.org [online]. Amsterdam: Blender, c2020 [cit. 2020-08-06]. Dostupné z: <https://docs.blender.org/manual/en/dev/render/eevee/introduction.html>
- [17] Workbench. Docs.blender.org [online]. Amsterdam: Blender, c2020 [cit. 2020-08-06]. Dostupné z: <https://docs.blender.org/manual/en/dev/render/workbench/introduction.html>
- [18] SHANNON, Tom. Unreal Engine 4 for design visualization: developing stunning interactive visualizations, animations, and renderings. Boston: Addison-Wesley, [2018]. ISBN 0134680707id.
- [19] What is Unreal Engine. Conceptartempire.com [online]. online: Concept Art Empire, c2020 [cit. 2020-08-06]. Dostupné z: <https://conceptartempire.com/what-is-unreal-engine/>
- [20] What is Quixel Mixer? Help.quixel.com [online]. online: Quixel, 2020 [cit. 2020-08-06]. Dostupné z: <https://help.quixel.com/hc/en-us/articles/115000613005-What-is-Quixel-Mixer->
- [21] What is Quixel Bridge? Help.quixel.com [online]. online: Quixel, 2020 [cit. 2020-08-06]. Dostupné z: <https://help.quixel.com/hc/en-us/articles/115000613105-What-is-Quixel-Bridge->
- [22] Blackmagic DaVinci Resolve 15. Pixel.cz [online]. Praha: ATLANTIDA, 2018 [cit. 2020-08-06]. Dostupné z: <https://www.pixel.cz/recenze/blackmagic-davinci-resolve-15-t664.html>

- [23] DaVinci Resolve 16 [online]. Fremont: blackmagicdesign, c2020 [cit. 2020-08-06]. Dostupné z: <https://www.blackmagicdesign.com/products/davinciresolve/>
- [24] How change LOD in geometry? In: Game Development [online]. online: Game Development, 2012 [cit. 2020-08-06]. Dostupné z: <https://gamedev.stackexchange.com/questions/31883/how-change-lod-in-geometry>
- [25] File:Cube Representative ... In: Wikimedia [online]. online: Wikimedia, 2008 [cit. 2020-08-06]. Dostupné z: [https://commons.wikimedia.org/wiki/File:Cube\\_Representative\\_UV\\_Unwrapping.png](https://commons.wikimedia.org/wiki/File:Cube_Representative_UV_Unwrapping.png)
- [26] Get Ready For Eevee ... In: Cgcookie [online]. online: Wikimedia, 2018 [cit. 2020-08-06]. Dostupné z: <https://cgcookie.com/articles/get-ready-for-eevee-blender-s-new-real-time-rendering-engine>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

př.n.l.	Před naším letopočtem
3D	Trojrozměrný
2D	Dvojrozměrný
LOD	Level of detail
PBR	Physically Based Rendering
Ctrl	Control
FBX	Filmbox
2K	Rozlišení 2048×1080
4K	Rozlišení 3840x2160

**SEZNAM OBRÁZKŮ**

<i>Obrázek 1 – Pravděpodobná podoba hradu.....</i>	12
<i>Obrázek 2 – Ukázka rozdílů mezi LOD levely.....</i>	16
<i>Obrázek 3 – UV mapování krychle.....</i>	17
<i>Obrázek 4 – Mapy PBR materiálu.....</i>	19
<i>Obrázek 5 – Rozdíl mezi Pre-Rendering a Real-Time.....</i>	20
<i>Obrázek 6 - Prostředí Unreal Engine.....</i>	23
<i>Obrázek 7 – Nabídka šablon v Unreal Engine.....</i>	24
<i>Obrázek 8 – Prostředí Quixel Mixer.....</i>	25
<i>Obrázek 9 – Nástroje pro úpravu barev v DaVinci Resolve.....</i>	26
<i>Obrázek 10 – Pravděpodobná podoba hlavní brány.....</i>	28
<i>Obrázek 11 – Rozdělení částí hradu.....</i>	29
<i>Obrázek 12 – Tvorba střílny v editačním módu.....</i>	31
<i>Obrázek 13 – Model šindelové střechy.....</i>	32
<i>Obrázek 14 – Vstupní oblouk brány.....</i>	34
<i>Obrázek 15 – Půdorys hradního komplexu.....</i>	35
<i>Obrázek 16 – Ukázka modelu vnější hradby ve Wireframe módu.....</i>	36
<i>Obrázek 17 – Model obranné věže před texturováním.....</i>	37
<i>Obrázek 18 – Tvorba roubené zdi.....</i>	39
<i>Obrázek 19 – Prostředí Quixel Mixer.....</i>	41
<i>Obrázek 20 – Online knihovny s materiály a modely.....</i>	42
<i>Obrázek 21 – Funkce Sculpt, Smooth, Flatten.....</i>	44
<i>Obrázek 22 – Dosud patrné terénní úpravy.....</i>	44
<i>Obrázek 23 – Detail terénní úpravy.....</i>	45
<i>Obrázek 24 – Vymodelované prostředí v Unreal Engine.....</i>	46
<i>Obrázek 25 – Nanášení vegetace pomocí štětce.....</i>	48
<i>Obrázek 26 – Node systém materiálu.....</i>	49
<i>Obrázek 27 – Ukázka materiálu prostředí již s vegetací.....</i>	50
<i>Obrázek 28 – Zakomponovaná obranná věž.....</i>	51
<i>Obrázek 29 – Video editor v Unreal Engine.....</i>	52
<i>Obrázek 30 – Rozložení kolizních boxů.....</i>	53
<i>Obrázek 31 – Pohled po spuštění aplikace.....</i>	54

## SEZNAM PŘÍLOH

Příloha P I – struktura DVD

Příloha P II – pohled na hrad

Příloha P III – budovy v předhradí

## **PŘÍLOHA P I: STRUKTURA DVD**

### **Text**

- 3D vizualizace hradu Starý Světlov.docx – digitální forma práce
- 3D vizualizace hradu Starý Světlov.pdf – digitální forma práce

### **Podklady a reference**

- Sary Svetlov.zip – veškeré podklady k práci a reference pro jednotlivé modely

### **Rendery**

- Rendery.zip – vyrenderované obrázky z Unreal Engine

### **Animace**

- Animace.zip -vytvořená animace v Unreal Engine

## PŘÍLOHA P II: POHLED NA HRAD





## PŘÍLOHA P III: BUDOVY V PŘEDHRADÍ

