

Modovací aplikace REDkit

Jiří Myška

Bakalářská práce | 2026

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2025/2026

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Jiří Myška
Osobní číslo: A23614
Studijní program: B0613A140020 Softwarové inženýrství
Forma studia: Kombinovaná
Téma práce: Modovací aplikace REDkit
Téma práce anglicky: REDkit Modding Application

Zásady pro vypracování

1. Seznamte se s módovacím nástrojem Redkit.
2. Popište aplikace použité v praktické části práce.
3. Navrhněte rozsáhlejší modifikaci hry Zaklínač 3.
4. S využitím nástroje Redkit tuto modifikaci implementujte do hry.
5. Modifikaci otestujte a odstraňte případné chyby.
6. Celý proces podrobně zdokumentujte.

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

Seznam doporučené literatury:

1. CD PROJEKT. The Witcher 3 REDkit [online]. 2024. [cit. 2025-10-28]. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/overview?homepagelid=2850903>
2. BLENDER FOUNDATION. Blender 4.5 LTS Manual [online]. 2025. [cit. 2025-10-28]. Dostupné z: <https://docs.blender.org/manual/en/latest/>
3. FAZZITO, Guido. The Witcher 3 REDkit & Wwise: A Setup Guide [online]. 2025. [cit. 2025-10-28]. Dostupné z: <https://www.audiokinetic.com/en/blog/the-witcher-3-redkit-tutorial/>
4. RINDO, Abigail. Game Narrative Design and UX Fundamentals: Tools for Player-centric Storytelling [online – ebook]. CRC Press, 2024. ISBN 9781040254219.
5. VINICIUS MACHADO VENÂNCIO. Blender 3D Asset Creation for the Metaverse [online – ebook]. De Gruyter GmbH, Walter, 2023. ISBN 9781801813013.

Vedoucí bakalářské práce: **Ing. Pavel Pokorný, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **19. října 2025**
Termín odevzdání bakalářské práce: **26. května 2026**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 2. března 2026

Prohlášení autora závěrečné kvalifikační práce

Beru na vědomí, že

- odevzdáním závěrečné práce souhlasím se zpřístupněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- závěrečná práce bude uložena v elektronické podobě v univerzitním informačním systému;
- jedno vyhotovení závěrečné práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji závěrečnou 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;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – závěrečnou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování závěrečné práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky závěrečné práce využít ke komerčním účelům;
- pokud je výstupem závěrečné 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 závěrečné práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze závěrečné práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

Abstrakt

Bakalářská práce se zabývá modovací aplikací REDkit určenou pro tvorbu modifikací počítačové hry Zaklínač 3: Divoký hon. Teoretická část shrnuje problematiku modování počítačových her, charakteristiku hry jako platformy pro modifikace, možnosti nástroje REDkit a využití externích aplikací Blender a Wwise. Praktická část je zaměřena na návrh a realizaci vlastní herní modifikace ve formě samostatné úkolové linie inspirované motivem příběhu blanických rytířů od spisovatele Aloise Jiráska. Modifikace rozšiřuje hru o nové lokace, dialogové scény, zvukový obsah a odměny pro hráče. Součástí práce je také popis tvorby venkovního a jeskynního prostředí, úprav 3D modelů a procesu publikace modu. Výsledkem práce je funkční ukázka rozšíření hry demonstrující praktické možnosti aplikace REDkit.

Klíčová slova

REDkit, Zaklínač 3, modování, mod, modifikace, Blender, Wwise

Abstract

This bachelor's thesis deals with the REDkit modding application intended for creating modifications for the video game The Witcher 3: Wild Hunt. The theoretical part summarizes the issue of video game modding, characteristics of the game as a platform for modifications, capabilities of the REDkit tool and the use of external applications such as Blender and Wwise. The practical part focuses on the design and implementation of an original game modification in the form of a standalone quest line inspired by the theme of the Knights of Blaník story written by Alois Jirásek. This modification extends the game with new locations, dialogue scenes, audio content, and player reward. The thesis also includes a description of the creation of outdoor and cave environments, 3D models, and the process of publishing the mod. The result of the thesis is a functional demonstration of a game extension presenting the practical possibilities of the REDkit application.

Keywords

REDkit, The Witcher 3, modding, mod, modification, Blender, Wwise

Poděkování

Tímto bych rád poděkoval vedoucímu své bakalářské práce, Ing. Pavlu Pokornému Ph.D., za návrh tématu práce, odborné vedení, vstřícný přístup a efektivní i motivační komunikaci v průběhu celého zpracování práce. Současně děkuji své přítelkyni za morální podporu, motivaci a pomoc při namluvení jedné z postav modifikace.

„ ... Hle, Blaník hora v hávu tmavého lesa, jenž splývá jí od temene po všech bocích. Vážně, až zachmuřeně shlíží na krajinu od světa odlehlou ... Ale pod hradbami, v hoře samé dřímají ozbrojení rytíři, „svatováclavské vojsko“, dřímají a čekají, až nastane den, kdy bude potřeba jejich pomoci, až budou do boje povoláni ... “

Alois Jirásek: Staré pověsti české – Blaničtí rytíři

Obsah

Seznam obrázků	17
Seznam použitých symbolů a zkratk	21
Seznam příloh	22
Úvod	23
A TEORETICKÁ ČÁST	24
1 Modování počítačových her	24
1.1 Definice a charakteristika modování	24
1.2 Pohled starší literatury na pojem modování	24
1.2.1 Úprava uživatelského rozhraní	25
1.2.2 Herní konverze.....	25
1.2.3 Machinima a umělecké modifikace	25
1.2.4 Modifikace herního počítače	25
1.2.5 Hackování herní konzole	26
1.3 Moderní pojetí modování.....	26
1.3.1 Hlavní perspektivy modování.....	26
1.3.2 Typy modů v moderní době	26
1.3.3 Proč jsou modifikace důležité pro moderní hry i herní průmysl	27
2 Zaklínač 3: Divoký hon jako platforma pro modifikace	28
2.1 Základní charakteristika hry	28
2.1.1 Struktura herního obsahu	28
2.1.2 Distribuce modifikací pro Zaklínače 3	29
2.2 REDengine a technická architektura.....	29
2.2.1 Skriptovací vrstva a vazba na engine.....	29
2.2.2 Limity REDkitu a modifikací pro Zaklínače 3	30
2.3 Oficiální podpora modifikací.....	30
2.4 Komunitní moderská scéna.....	30
3 Nástroj REDkit	31
3.1 Co je to REDkit.....	31
3.2 Vymezení pojmů	31

3.3	Požadavky a zavedení REDkitu do systému.....	32
3.3.1	Stažení editoru REDkit	32
3.3.2	Prvotní spuštění REDkitu.....	32
3.3.3	Vytvoření projektu.....	33
3.4	Základní práce s REDkitem	34
3.4.1	Vytvoření nového světa.....	34
3.4.2	Načtení existujícího světa	34
3.4.3	Hlavní obrazovka editoru.....	36
3.4.4	Ovládání kamery	38
3.4.5	Spuštění hry v editoru	38
3.4.6	Referenční svět.....	40
3.5	Nástroje REDkitu.....	41
3.5.1	Nástroje pro editaci terénu	41
3.5.2	Nástroj pro editaci vegetace	46
3.5.3	Marauder's Map	47
3.5.4	Editace Vertexů	48
3.5.5	Job Tree Editor	49
3.5.6	Editory animací	51
3.5.7	Quest Debugger a Quest Editor	58
3.5.8	Editor Entit.....	58
3.5.9	Attitude Editor.....	60
3.5.10	Loot Editor	61
3.5.11	Ruler.....	62
3.5.12	Editor Lokalizovaných Řetězců	62
3.5.13	Scene Editor	64
3.6	Witcher Script	65
3.6.1	Příprava prostředí pro WitcherScript	65
3.6.2	Základy WitcherScriptu	65
3.6.3	Kompilační chyby	72
3.6.4	Script Merger	72
3.6.5	Anotace ve skriptu	73

3.6.6	Základy nástroje Witcher Script Studio	74
4	Implementace externích aplikací Blender a Wwise.....	76
4.1	Blender.....	76
4.1.1	Použití s REDkitem	76
4.1.2	Integrace do REDkitu přes Blender plugin.....	76
4.2	Audiokinetic Wwise.....	76
4.2.1	Použití Wwise s REDkitem	76
4.2.2	Licence.....	77
4.2.3	Základní použití a vazba na REDkit	77
4.2.4	Příprava prostředí Wwise v rámci REDkitu	77
B	Praktická část.....	79
5	Návrh modifikace hry Zaklínač 3.....	79
5.1	Cíle návrhu a požadavky na modifikaci	79
5.2	Narativní východiska a tematika.....	80
5.3	Aktivace úkolu.....	80
5.4	Návrh lokací.....	80
5.4.1	Horská lokace	80
5.4.2	Jeskyně.....	81
5.5	Struktura úkolové linie.....	81
5.6	Finální volba a následky	81
6	Vytvoření venkovního světa	82
6.1	Vytvoření nového světa pro horskou oblast.....	82
6.2	Tvorba struktury horské oblasti	82
6.3	Tvorba terénu horské oblasti.....	83
6.4	Tvorba dekorací a důležitých lokací.....	85
6.4.1	Vstup do jeskyně na východním svahu hory	85
6.4.2	Kovářova oblast	85
6.4.3	Malé jezero jako demonstrace použití vodní plochy	87
6.5	Vytvoření vegetace v oblasti	89
6.6	Nastavení definice prostředí v horské oblasti	91
7	Vytvoření jeskyně.....	92

7.1	Vytvoření nového světa a modelace terénu	92
7.2	Tvorba struktury jeskyně	93
7.2.1	Vstupní hala	93
7.2.2	Skladiště	96
7.2.3	Hlavní chodba jeskyně	97
7.2.4	Rytířská místnost.....	99
7.2.5	Kořist a interaktivní objekty	100
7.3	Nastavení prostředí světa	101
7.3.1	Nastavení definice prostředí.....	101
7.3.2	Environmentální sondy	104
7.4	Využití Blenderu pro tvorbu rytířů.....	105
7.5	Využití Blenderu pro tvorbu meče	108
8	Tvorba zvuků pro modifikaci	110
8.1	Přidání ambientní hudby pro horský svět a jeskyni	110
8.2	Tvorba hlasového projevu postav a lipsync animace.....	112
8.2.1	Tvorba hlasového projevu.....	112
8.2.2	Tvorba lipsync animací.....	113
9	Tvorba dějové linie a skriptování	115
9.1	Realizace aktivace úkolu.....	115
9.2	Nalezení kováře a interakce s ním	121
9.3	Vyřešení příběhu v jeskyni.....	123
9.4	Uzavření příběhu a odměna	127
10	Publikace a problémy	128
10.1	Proces publikace modu na platformu Steam.....	128
10.2	Zpětná vazba z publikovaného modu.....	128
10.3	Problémy	129
Závěr	131
Seznam použité literatury	133

Seznam obrázků

Obr. 1: Snímek ze hry Zaklínač 3: Divoký hon zobrazující postavu Geralta s mečem, který obdržel jako odměnu za splnění úkolu vytvořeného v rámci této práce.....	28
Obr. 2: Snímek úvodního okna po prvotním spuštění REDkitu [18].....	32
Obr. 3: Snímek okna pro vytvoření nového projektu.....	33
Obr. 4: Snímek nabídky pro vytvoření nového světa.....	34
Obr. 5: Snímek nabídky pro otevření existujícího světa.....	34
Obr. 6: Snímek obrazovky s Asset Browserem zobrazující svět novigrad.w2w	35
Obr. 7: Ukázka rozdílu mezi nenačtenými objekty (horní část) a načtenými objekty (spodní část).....	35
Obr. 8: Popis částí hlavní obrazovky editoru	36
Obr. 9: Snímek zobrazující tlačítka pro spuštění hry (zvýrazněno červeným obdelníkem)	38
Obr. 10: Snímek s označením vyhledávacího pole Asset Browseru.....	39
Obr. 11: Snímek s označením tlačítka pro zachování pohledu kamery	40
Obr. 12: Snímek ukázky referenčního světa	40
Obr. 13: Snímek okna Terrain Edit Tools zobrazující záložku Setup	41
Obr. 14: Snímek okna Terrain Edit Tools zobrazující záložku Shape Brush	42
Obr. 15: Snímek okna Terrain Edit Tools zobrazující záložku Material.....	43
Obr. 16: Snímek okna Terrain Edit Tools zobrazující záložku Colors.....	44
Obr. 17: Snímek okna Terrain Edit Tools zobrazující záložku Import / Export	45
Obr. 18: Snímek okna Terrain Edit Tools zobrazující záložku Collision.....	45
Obr. 19: Snímek okna Vegetation Edit Tool	46
Obr. 20: Snímek zobrazující nástroj Marauder's Map při identifikaci nepřátelské postavy.....	47
Obr. 21: Snímek zobrazující kontextové menu v Marauder's Map nástroji.....	48
Obr. 22: Snímek zobrazující označenou oblast CTriggerAreaComponent0 před editací vertexů	48
Obr. 23: Snímek zobrazuje oblast CTriggerAreaComponent0 po editaci vertexů	49
Obr. 24: Snímek okna Job Tree Editor s otevřeným kontextovým menu pro výběr modelu postavy	50
Obr. 25: Snímek okna Job Tree Editor s přehrávanou animací sit_loudly_dining_jt_loop_01_rh.....	50
Obr. 26: Snímek zobrazuje animační soubor rat_animation v Asset Browseru.....	51
Obr. 27: Snímek zobrazující volbu exportování animace z AnimBrowseru.....	52
Obr. 28: Snímek zobrazující HDFView část pro re_animation s xformframes daty pro levou ruku	53
Obr. 29: Snímek zobrazující HDFView část pro re_skeleton s transformační tabulkou pro levou nohu.....	53
Obr. 30: Snímek zobrazující nabídku instalace Pluginů (Add-ons) z Disku	54
Obr. 31: Snímek aktivovaného pluginu pro zpracování animací do Blenderu	55
Obr. 32: Snímek zobrazující Blender software s importovanou animací krysy ve scéně....	55
Obr. 33: Snímek zobrazující možnost exportování animace z Blenderu	56

Obr. 34: Snímek zobrazující možnosti importování animací do REDkitu pomocí AnimBrowseru	57
Obr. 35: Snímek zobrazující rozdíl originální a editované animace	57
Obr. 36: Snímek zobrazuje okno Quest Debugger s načteným úkolem ACHIEVEMENT – monster hunts	58
Obr. 37: Snímek zobrazující Asset Browser se zobrazenými šablonami entit hlavních postav	59
Obr. 38: Snímek zobrazující Entity Editor s načtenou šablonou entity caranthil.w2ent	59
Obr. 39: Snímek zobrazující provedení změny v Attitude editoru	60
Obr. 40: Snímek zobrazující okno Loot Editor	61
Obr. 41: Snímek zobrazující použití nástroje Ruler	62
Obr. 42: Snímek zobrazující možnosti nastavení údajů o projektu	63
Obr. 43: Snímek zobrazující Localized Strings Editor	64
Obr. 44: Snímek zobrazující možnou kompilační chybu skriptu při spuštění hry [34]	72
Obr. 45: Snímek zobrazující okno Witcher Script Studio	74
Obr. 46: Snímek zobrazující pokus o editaci souboru ve Witcher Script Studio	75
Obr. 47: Snímek zobrazující umístění Wwise projektu uvnitř REDkit složky	77
Obr. 48: Nastavení světa horské oblasti	82
Obr. 49: Pohled na uskupení kopců vytvořených štětcem Rise	83
Obr. 50: Ukázka postupu zvrátnění terénu hor pomocí štětce Rise a nízké pravděpodobnosti	84
Obr. 51: Ukázka aplikovaného horského materiálu na vrcholky hor	84
Obr. 52: Pohled na vstup do jeskyně umístěn na východním svahu hory	85
Obr. 53: Zvolený materiál pro půdu na pozemku kováře	85
Obr. 54: Pohled na lokalitu obydlí kováře	86
Obr. 55: Pohled na exteriérovou a interiérovou část domu kováře	86
Obr. 56: Pohled do hotové oblasti kováře z publikované verze modifikace	87
Obr. 57: Vyznačené nastavení pro umožnění zobrazení vodních ploch v editoru (disableWaterShaders)	87
Obr. 58: Pohled před přidáním vodní plochy (nahore), umístění vodní plochy (uprostřed), po přidání vegetace (dole)	88
Obr. 59: Pohled z okraje horské oblasti na stezku vedoucí k ruině na hoře skrze zalesněnou oblast	89
Obr. 60: Ukázka úpravy masky pro generování trávy	90
Obr. 61: Nastavení štětce trávy pro generování podle masky	90
Obr. 62: Pohled na pole před horou	91
Obr. 63: Detail nastavení druhů počasí horské oblasti a jejich pravděpodobností	91
Obr. 64: Nastavení světa pro jeskyni	92
Obr. 65: Náhled modelace terénu pro referenci půdorysu jeskyně	93
Obr. 66: Pohled na tvorbu podlahy vstupní haly jeskyně	94
Obr. 67: Pohled na podlahu a stěny vstupní haly jeskyně	94
Obr. 68: Pohled na vstupní halu po dokončení podlahy, stěn a stropu	95
Obr. 69: Pohled na hotovou část vstupní haly jeskyně	96
Obr. 70: Pohled na ukázkou dekorací ve skladišti	96
Obr. 71: Pohled na spojovací chodbu kolem skladiště jeskyně z průběhu modelování	97

Obr. 72: Pohled na půdorys hlavní chodby vedoucí k místnosti s rytíři.....	98
Obr. 73: Ukázka hlavní chodby s dekoracemi a osvětlením.....	98
Obr. 74: Pohled z ptáčích perspektivy na půdorys rytířské místnosti z průběhu tvorby	99
Obr. 75: Pohled od vstupu do rytířské místnosti.....	100
Obr. 76: Pohled na umístěný interaktivní objekt s kořistí ve skladišti (stojan na meče) ...	101
Obr. 77: Pohled na nerealistický stín sochy a odlesk na podlaze způsobený umělým sluncem, před jeho úpravou.....	102
Obr. 78: Snímek zobrazující nastavení globálního světla v prostředí jeskyně	102
Obr. 79: Pohled na výsledek po úpravě hodnot umělého slunce, již není vidět nerealistický stín sochy na podlaze	103
Obr. 80: Porovnání před a po přidání mlhy do oblasti jeskyně.....	103
Obr. 81: Nastavení parametrů komponenty globální sondy pro jeskyni.....	104
Obr. 82: Ukázka rozdílu bez použití environmentální sondy (vlevo) a s použitím globální environmentální sondy k získání tmavě červeného odlesku (vpravo).....	105
Obr. 83: Ukázka tvorby šablony entity kostlivce a jeho vzhledů (anglicky <i>Appearances</i>)	106
Obr. 84: Ukázka použití Pose Mode v Blenderu pro úpravu pozice armatury modelu kostlivce	106
Obr. 85: Ukázka slučování částí modelu lebky dohromady (zeleně vybarvená je sloučená lebka modelu kostlivce).....	107
Obr. 86: Model meče jako odměna hráči za opravu artefaktu (Blender nahoře, REDkit dole).....	108
Obr. 87: Ikona meče vytvořená pro zobrazení v inventáři hráče	109
Obr. 88: Event Property Editor ze softwaru Wwise zobrazující audio soubory pro událost spuštění hudby na pozadí v oblasti jeskyně.....	110
Obr. 89: Ukázka s vloženými audio soubory do SoundBank Editoru	111
Obr. 90: Kontextové menu v softwaru Wwise pro generování zvukové banky	111
Obr. 91: Ukázka umístění oblasti pro zvukovou událost (žlutá barva) v horské oblasti ...	112
Obr. 92: Scene Editor zobrazující záložku scénáře (vlevo nahoře), časovou osu (vlevo dole) a vlastnosti konkrétní věty scénáře (vpravo)	113
Obr. 93: Blender projekt zobrazující připravený projekt pro automatické generování lipsync animací ze zvukového WAV souboru.....	114
Obr. 94: Localized Strings Editor zobrazující řetězce použité v úvodní části úkolu modifikace	115
Obr. 95: Šablona entity modifikovaného dřevěného sloupu otevřená v Entity Editoru	116
Obr. 96: Detail nastavení vlastností dřevěného sloupu (zakázaná interakce).....	116
Obr. 97: Pohled na umístěný modifikovaný dřevěný sloup poblíž informační tabule sloužící jako teleport	117
Obr. 98: Ukázka struktury úkolu, jeho popisků a mappin položek v nástroji Journal Editor	117
Obr. 99: Quest Editor nástroj zobrazující úkolový soubor tohoto modu rozdělený do jednotlivých fází (uzly s názvem <i>Phase</i>)	118
Obr. 100: Quest Editor nástroj zobrazující graf v místě začátku modifikace (uzel Script nalevo)	118
Obr. 101: Quest Editor zobrazující graf řešící logiku splnění části úkolu přečtení dopisu a zahájení části úkolu nalezení dřevěného sloupu pro transport do horské oblasti	119

Obr. 102: Quest Editor zobrazující část grafu řešící splnění části úkolu nalezení sloupu, spuštění skriptu pro umožnění interakce dřevěného sloupu a aktualizace údajů k úkolu..	119
Obr. 103: Detail nastavení parametrů skriptového uzlu řešící umožnění interakce dřevěného sloupu	120
Obr. 104: Quest Editor zobrazující část grafu řešící interakci se sloupem a teleport do horské oblasti	120
Obr. 105: Quest Editor zobrazující začátek druhé fáze řešící úvodní nastavení počasí a času	121
Obr. 106: Quest Editor druhé fáze řešící změnu světa a kontrolu, zda hráč našel kováře	121
Obr. 107: Ukázka grafu scény řešící dialog kováře s hráčem	122
Obr. 108: Ukázka z testování scény u vstupu do jeskyně s nabídkou vstoupit do jeskyně, nebo odejít.....	122
Obr. 109: Připravené druhé setkání s nepřáteli v hlavní chodbě.....	123
Obr. 110: Quest Editor graf řešící paralelní děj: spodní větev sledující událost nalezení knihy, horní větev pokračuje v ději	124
Obr. 111: Pohled na umístěné kostlivce rytířů a poznámky na stole	124
Obr. 112: Ukázka větvení v grafu scény nalezení poznámek rytíře.....	125
Obr. 113: Pohled na krystal důležitý pro úkol umístěný do dekorace vázy	125
Obr. 114: Pohled na opravený artefakt osvětlující okolí svou září	126
Obr. 115: Pohled na postavu Geralta používající meč, který obdržel jako odměnu za úkol z tohoto modu	127
Obr. 116: Ukázka špatné animace pro koně v publikované verzi modu	129
Obr. 117: Špatně zrozený nepřítel pod terénem (bílé kruhy uprostřed označující jeho polohu)	130

Seznam použitých symbolů a zkratek

3D	trojrozměrný / trojrozměrná grafika
CSV	Comma-Separated Values – souborový formát, kde jsou hodnoty odděleny čárkou
DDS	DirectDraw Surface – formát souboru pro ukládání zejména textur
DLC	Downloadable Content – stahovatelný dodatečný obsah
DIY	do it yourself – svépomocná úprava / vlastní činnost
FBX	Filmbox – formát trojrozměrných modelů a animací
HDF	Hierarchical Data Format – hierarchický datový formát
ID	Identifier – identifikátor, často se jedná o číselnou hodnotu
LOD	Level of Detail – úroveň detailu, často spojená s různou úrovní LOD daného objektu pro různou vzdálenost od objektu
OOP	Object oriented Programming – objektově orientované programování
PNG	Portable Network Graphics – rastrový obrazový formát
RPG	Role-playing game – hra na hrdiny / role
UI	User interface – uživatelské rozhraní
UV	souřadnice pro mapování textury na trojrozměrný model
WAV	Waveform Audio File Format – nekomprimovaný a bezeztrátový formát zvukového souboru
XML	Extensible Markup Language – rozšiřitelný značkovací jazyk
ZIP	komprimovaný archivní formát

Seznam příloh

Přílohy vložené v závěrečné práci

Příloha A: Zdrojový kód a prompty

Příloha B: Obrazová příloha

Ostatní přílohy

Příloha C: Projekt Příloha C – projekt.zip

Úvod

Modování počítačových her představuje významnou součást současné herní kultury i praktické tvůrčí činnosti, protože umožňuje rozšiřovat původní herní obsah o nové prvky, mechaniky, lokace nebo celé příběhové linie. Vedle významu pro komunitu hráčů a moderů má modování také rozměr technický a vývojový, jelikož propojuje práci s herním světem, skriptováním, audiovizuálními prostředky a externími nástroji pro tvorbu obsahu.

Jedním z nástrojů pro modování je REDkit [1], určený pro modifikaci počítačové hry Zaklínač 3: Divoký hon [2]. Tento nástroj umožňuje upravovat a rozšiřovat hru pomocí prostředků, které byly využity při jejím vývoji. Uživatel tak může pracovat s herními světy, úkoly, scénami, objekty, skriptovou logikou i zvukovou složkou. Tato širší možností činí z REDkitu vhodný nástroj pro zkoumání praktických možností tvorby herních modifikací.

Cílem této bakalářské práce je představit modovací aplikaci REDkit, popsat její možnosti a ověřit její praktické využití na návrhu a realizaci vlastní herní modifikace pro hru Zaklínač 3: Divoký hon. Práce je rozdělena na teoretickou a praktickou část.

Teoretická část se věnuje problematice modování počítačových her, charakteristice hry Zaklínač 3: Divoký hon jako platformy pro modifikace, představení nástroje REDkit a roli externích aplikací při tvorbě obsahu pro REDkit.

Praktická část navazuje návrhem vlastní modifikace a dokumentací její realizace od návrhu úkolové linie a tvorby nových prostředí až po skriptování, zvukové zpracování, publikaci, testování a odstranění zjištěných chyb výsledného modu.

A TEORETICKÁ ČÁST

1 Modování počítačových her

1.1 Definice a charakteristika modování

Modování (anglicky *modding*) označuje činnost vytváření vlastního obsahu pro videohry [6]. Nejčastěji se jedná o úpravy herního světa, objektů, postav, herních pravidel nebo příběhu. Modování lze chápat jako uživatelsky řízený proces, jehož cílem je rozšíření nebo změna původního herního zážitku bez přímého zásahu původního vývojářského studia.

Počátky takových úprav jsou pozorovány už od počátku 90. let a jsou stále populárnější. S rozvojem internetu, komunitních fór a specializovaných platforem se stal běžnou součástí herní kultury. Populární webové stránky zabývající se mody je například NexusMods [4] nebo MODDB [5].

Někteří vydavatelé her tuto činnost aktivně podporují a poskytují své oficiální nástroje pro modifikace jejich hry, příkladem je právě REDkit od společnosti CD PROJECT RED určený pro modifikaci hry Zaklínač 3: Divoký hon. Díky těmto nástrojům mohou uživatelé vytvářet nové lokace, postavy, úkoly, skripty i audiovizuální obsah.

1.2 Pohled starší literatury na pojem modování

Modování podle Scacchiho [3][1] nelze jednoznačně definovat, protože význam se liší podle toho, kdo jej interpretuje. V době vydání jeho článku [3] v roce 2010, Scacchi rozděluje herní modifikace do pěti kategorií:

- Úprava uživatelského rozhraní
- Herní konverze
- Machinima a umělecké modifikace
- Modifikace herního počítače
- Hackování herní konzole

Toto rozdělení pojímá modování komplexně, zahrnuje nejen tvorbu nového herního obsahu, ale také technické, vizuální, umělecké i hardwarové zásahy.

1.2.1 Úprava uživatelského rozhraní

Uživatelské rozhraní (anglicky *User Interface – UI*) funguje jako prostředník mezi hráčem a herním světem a určuje jakým způsobem hráč získává informace o dění ve hře.

Cílem těchto modifikací bývá například: zvýšení přehlednosti herního rozhraní, zpřehlednění informací o stavu postavy, nepřátel atd., případné získání kompetitivní výhody.

Scacchi [3] rozlišuje tři základní typy úprav uživatelského rozhraní hráčem:

1. možnost změnit, upravit vzhled nebo přidat doplňky herní postavě
2. úprava barevnosti a vizuálních prvků rozhraní
3. přídatné informační prvky (anglicky *add-ons*) zobrazující údaje o nepřátelích atd.

1.2.2 Herní konverze

Podle Scacchiho [3] herní konverze představují nejrozšířenější herní modifikaci. Herní konverze dále rozděluje do dvou základních podob: částečné a celkové.

Částečné herní konverze zahrnují:

- Úpravy vzhledu postav a jejich schopností
- Úpravy herních předmětů (zbraně, lektvary, kouzla, ...)
- Změny herních lokací, terénu a úrovní
- Úpravy herních pravidel a herních mechanik.

Celkové herní konverze vytvářejí z původní hry zcela novou hru z již existujících her. Znamé příklady takové celkové herní konverze je úspěšná hra Counter-Strike mod pro počítačovou hru Half-Life [7]. Dalším příkladem je Garry's mod vycházející ze hry Half-Life 2 [7], který se vyvinul v celý toolkit, s jehož pomocí vnikla spousta dalších herních konverzí. Tento typ modování bývá technicky i organizačně velmi náročný.

1.2.3 Machinima a umělecké modifikace

Machinima se soustředí na tvorbu média v kinematické podobě. Proces je takový, že hráč při hraní hry pořídí záznam videa, nebo i zvuku, a následně je pomocí editačního software upraví k tvorbě filmových sekvencí, krátkých filmů nebo příběhových scén.

Umělecké modifikace pak slouží k experimentálním zásahům do vizuální podoby hry. Primárním cílem není změna herní mechaniky, ale vyjádření uměleckého záměru, kritiky nebo společenského sdělení.

1.2.4 Modifikace herního počítače

Scacchi [3] zařazuje jako jeden z typů modování rovněž modifikaci herního počítače jako takového. Nejedná se o modování samotného herního software, ale hardwarové platformy. Patří zde například modování počítačové skříně (anglicky *case-modding*), který nemá vliv na herní funkcionalitu nebo výkon, ale zaměřuje se na vizuální kvality počítače. Modifikace

upravující výkon hry jsou například přetaktování procesoru na vyšší frekvenci s úpravou chlazení procesoru.

1.2.5 Hackování herní konzole

Poslední kategorií je hackování herních konzolí. Jedná se o zásahy do zabezpečení konzolových systémů s cílem umožnit spouštění neoficiálního softwaru, vlastních aplikací nebo modifikovaných her. Takové zásahy často porušují licenční podmínky výrobců a mohou vést ke ztrátě záruky zařízení. Hackování konzolí je často spojeno s reverzním inženýrstvím nebo tzv. „DIY“ modifikacemi.

Z uvedených kategorií je pro tuto bakalářskou práci relevantní především úprava uživatelského rozhraní a herní konverze.

1.3 Moderní pojetí modování

V současné době je herní komunitou výraz mod (modifikace) používáno pro označení jakékoli uživatelské úpravy, nebo rozšíření původní hry. Může se jednat o malé vizuální, nebo funkční změny. Stejně jako o rozsáhlé zásahy přetvářející hru v novou zkušenost.

Podle Dyera [7] jakákoli úprava originální hry, ať už malá změna nebo celková konverze, může být legitimně považována za mod.

1.3.1 Hlavní perspektivy modování

Thiel a Lyle ve své práci [15] analyzují literaturu o modování a navrhují rozdělení modů podle perspektivy a funkcí, nikoli podle technické, nebo hardwarové definice.

- Perspektiva zvenku (anglicky *outside perspective*) – modování může porušit copyright, jelikož modeři používají zdroje vydavatelů ke změně produktu, ke kterému pak cítí morální vlastnictví. Vydavatelé mohou, pokud je mod vytvořen bez jejich souhlasu, pak provést právní kroky k ukončení činnosti moderů. Postupem času se přístup herního průmyslu stal tolerantnější k moderské scéně, ale stále dochází k občasným porušením copyrightu.
- Perspektiva zevnitř (anglicky *inside perspective*) – zaměřuje se na interní charakteristiky modování a moderské komunity. Snaží se zodpovědět jaké jsou motivy, pravidla, vztahy a každodenní praktiky moderů. Motivací moderů pro úpravy herních titulů může být například získání nových dovedností, prodloužení životnosti hry, zábava a kariérní postup.
- Meta-level perspektiva (anglicky *meta-level perspective*) – kombinuje perspektivy zvenku i zevnitř. Zkoumá důsledky, vliv a vnímání vytvářených modů.

1.3.2 Typy modů v moderní době

Aktuálnější rozdělení diskutuje Dyer ve svém článku [7]. Poukazuje na úspěch modování počítačových her, který se stále zlepšuje a zvětšuje, ať už profesionály nebo amatéry.

V rámci modování počítačových her v moderní době jsou markantní tyto typy modů:

- Kosmetické, nebo drobné úpravy – změny vzhledu, textur, barev, grafických nastavení UI.
- Rozšíření obsahu – přidání nových zbraní, předmětů, postav, úkolů apod. Díky tomu se prodlužuje životnost hry a expanduje její svět.
- Celkové konverze – kompletní přetvoření hry, z původního herního titulu zůstává pouze herní engine.

1.3.3 Proč jsou modifikace důležité pro moderní hry i herní průmysl

Přesto, že v některých případech jsou modifikace her považovány za porušení copyrightu a jsou na tuto problematiku zaměřené jiné studie, lze v modifikacích identifikovat benefity.

- Prodloužení životnosti hry – díky modům si hráči mohou hru užívat i dlouho po vydání, což zvyšuje hodnotu a návratnost investice do zakoupení hry.
- Podpora kreativity a komunity – modování dává hráčům možnost stát se tvůrci, sdílet svou tvorbu a spolupracovat.
- Inovace bez oficiálního vývoje – modéři často experimentují s novými mechanikami, příběhy, které by možná oficiální vývoj nikdy nezkusil. Mody tak mohou být testovacím prostředím pro inovace.
- Ekonomický přínos a udržitelnost – pro vývojáře je podpora modů často benefiční: komunitní rozšíření, pozitivní publicita, prodloužená životnost produktu.

2 Zaklínač 3: Divoký hon jako platforma pro modifikace



Obr. 1: Snímek ze hry Zaklínač 3: Divoký hon zobrazující postavu Geralta s mečem, který obdržel jako odměnu za splnění úkolu vytvořeného v rámci této práce
(zdroj: vlastní)

Zaklínač 3: Divoký hon (anglicky *The Witcher 3: Wild Hunt*) patří mezi hry s dlouhodobě aktivní moderskou komunitou. V této kapitole je stručně popsána charakteristika hry, struktura jejího obsahu z pohledu modifikací, technické aspekty enginu REDengine a dostupná oficiální i komunitní infrastruktura pro tvorbu a distribuci modů.

2.1 Základní charakteristika hry

Zaklínač 3 je příběhové akční RPG (anglicky *Role Playing Game*) zasazené do otevřeného dark-fantasy světa, kde hráč ovládá postavu Geralta z Rivie a postupuje prostřednictvím hlavní dějové linie i velkého množství vedlejších úkolů a aktivit [8]. Pro modování je podstatné, že hra kombinuje otevřený svět, který umožňuje hráči volnou exploraci a výrazně narativní obsah (úkoly, dialogy, filmové sekvence / příběhové scény – anglicky *cutscenes*), které lze rozšiřovat, nebo upravovat v různých vrstvách herních dat.

2.1.1 Struktura herního obsahu

Z pohledu modifikací je vhodné chápat herní obsah Zaklínače jako několik “vrstev”, které se často modují samostatně nebo v kombinaci:

- svět a lokace – počasí, terén, objekty, navmesh (navigační data pro pohyb a chování postav), rozmístění entit
- obsah úkolů – graf úkolu (anglicky *quest graph*), podmínky a větvení úkolů, spouštěče (anglicky *triggers*), odměny
- scény a dialogy – příběhové scény, volby v dialogu, animace, práce s kamerou
- entity a herní systémy – postavy, předměty, inventář, UI prvky, chování
- audiovizuální vrstva – modely, textury, animace, zvuk a hudba
- skriptová logika – skripty a jejich vazby na výše uvedené vrstvy

Modovací nástroj REDkit při prvním spuštění pracuje s lokální instalací hry Zaklínač 3 a připraví tzv. *depot*, tj. složku obsahující extrahovaná pracovní data. Zároveň odděleně vytváří a spravuje *workspace*, tedy projektový prostor, ve kterém autor modifikace provádí vlastní změny a přidává nový obsah [9]. Tímto postupem se jasně odděluje „referenční“ obsah původní hry od souborů, které modifikace reálně mění nebo doplňuje.

Toto členění pomáhá odlišit, zda mod upravuje datovou vrstvu (například prostředky – anglicky *assets* – a definice entit), nebo logickou vrstvu (úkoly / scény / systémy), což má dopad na kompatibilitu i náročnost ladění. Prakticky to znamená, že mod této hry může být jak drobná úprava (například změna parametrů nějakého objektu), tak rozsáhlé rozšíření (nová úkolová linie, lokace, postavy). Oficiální REDkit cílí právě na tvorbu a editaci těchto složek pomocí nástrojů používaných vývojáři.

2.1.2 Distribuce modifikací pro Zaklínače 3

Z hlediska distribuce modifikace pro hru je klíčovým krokem publikování (anglicky *publishing*). REDkit poskytuje proces, který z mod projektu vytvoří výstup určený pro spuštění ve hře a případně připravený k publikaci online. Výsledkem publikování je balíček obsahující pouze ty soubory a zdroje, které byly v rámci *workspace* změněny, nebo nově vytvořeny, což usnadňuje instalaci modu do hry i jeho sdílení.

2.2 REDengine a technická architektura

Zaklínač 3 běží na technologii REDengine [10], konkrétně verze REDengine 3. Pro potřeby otevřeného světa engine využívá průběžné načítání (anglický pojem *streaming*) a rozsáhlý systém nástrojů pro tvorbu a správu obsahu. Z pohledu modifikací je důležité, že herní obsah je rozdělen do samostatných typů zdrojů (světová data, entity, scény, úkoly a další), které lze upravovat odděleně.

2.2.1 Skriptovací vrstva a vazba na engine

Hra obsahuje skriptovací vrstvu používanou pro chování systémů, logiky úkolů, interakce a další. Skripty jsou psány v jazyce vyvinutém pro hru Zaklínače nazvaný Witcher Script. Soubory s tímto skriptem jsou uloženy s příponou *.ws*. O tomto skriptu je pojednáno v kapitole 3.6.

2.2.2 Limity REDkitu a modifikací pro Zaklínače 3

Nástroj REDkit nabízí spoustu nástrojů pro modifikace různých vrstev hry, nicméně i podle oficiální dokumentace může vývoj modifikací vyžadovat externí nástroje, jako je například Blender [23], HDFView [22] a další.

Při zásahu do stejné části obsahu v rámci více modifikací může nastat problém s kompatibilitou a může být potřeba událost vyřešit sloučením modifikací (obzvláště u skriptu), o kterém je hovořeno v kapitole 3.6.4.

Náročnější grafické modifikace pro hru mohou způsobit problémy se stabilitou.

2.3 Oficiální podpora modifikací

Developeři (CD PROJECT RED) hry Zaklínač 3: Divoký hon mají s modováním svých her zkušenosti z minulosti. Pro zaklínače 2 byl v roce 2012 oznámen a následně uveden editor REDkit jako nástroj pro tvorbu obsahu a úkolů a na svých stránkách modovací komunitu developeři podporovali k provádění modifikací [11].

Pro REDkit byla také v roce 2024 přidána integrace do systému Steam Workshop, kdy si uživatelé mohou mody do Zaklínače 3 instalovat přímo přes platformu Steam pomocí Subscribe tlačítka [12].

CD PROJECT RED v roce 2025 oznámilo “cross-platform” podporu modů [13], tedy podpora modu nejen na PC, ale i na jiných zařízeních, jako je například Playstation nebo XBOX.

2.4 Komunitní moderská scéna

Komunita moderů je kolem Zaklínače 3 dlouhodobě velmi aktivní a zahrnuje široké spektrum modů. Od úprav uživatelského rozhraní, “quality of life” změn, přes grafická vylepšení, celkové herní přeměny (anglicky *gameplay overhaul*), až po komplexnější zásahy do úkolů, či přidávání obsahu.

V praxi je nejvýznamnějším distribučním uzlem pro modování počítačových her dlouhodobě Nexus Mods [4]. V závěru roku 2025 zde bylo možné nalézt přibližně 8 200 modifikací pro hru Zaklínač 3: Divoký hon.

Další variantou distribuce je pomocí Steam Workshop [14], kde je počet položek výrazně nižší (řádově stovky). Obdobně je tomu na dalších platformách pro sdílení modifikací jako je ModDB [5] a podobné.

3 Nástroj REDkit

Tato kapitola popisuje REDkit – modovací nástroj pro počítačovou hru Zaklínač 3: Divoký hon (anglicky *The Witcher 3: Wild Hunt*) – a shrnuje schopnosti tohoto nástroje, jeho funkce a postupy podle oficiální dokumentace. Cílem je objasnit, jak se REDkit používá, co umožňuje a jaké kroky musí uživatel (moder) následovat, aby vytvořil funkční modifikace.

3.1 Co je to REDkit

The Witcher 3 REDkit je modovací nástroj, který uživatelům umožňuje upravovat a rozšiřovat počítačovou hru Zaklínač 3: Divoký hon za využití stejných nástrojů používaných developery, kteří tuto hru vytvořili [16].

Podle vydavatelů REDkit umožní uživatelům vytvořit (mimo jiné):

- Nové úkoly (anglicky *quests*)
- Nové světy (anglicky *worlds*)
- Nové příběhové scény
- Nové postavy, monstra
- Nové záznamy v deníku, dialogy a další...

Uživatelé mají taktéž přístup k veškerým herním souborům, které mohou upravovat pomocí plně podpořených nástrojů. Tímto způsobem by uživatel měl být schopen vytvořit nové DLC pro hru a zabalit mody standardním způsobem například přes platformu Steam Workshop [14], nebo NexusMods [4].

3.2 Vymezení pojmů

V rámci této práce je použit pojem mesh, pod kterým je myšlen trojrozměrný model objektu používaný ve scéně REDkitu. Jedná se např. o budovu, dekoraci, část vybavení nebo jiný prostorový prvek. Mesh určuje především geometrickou podobu objektu. Jeho výsledný vzhled je dále ovlivněn přiřazenými materiály a texturami.

Pod pojmem materiál je v kontextu REDkitu myšlen prostředek určující vizuální vlastnosti povrchu objektu nebo terénu – např. jeho barevnost, strukturu, odlesky nebo způsob vykreslení povrchu. U terénu byly využívány zejména existující terénní materiály dostupné v REDkitu. U některých vlastních, nebo upravených modelů bylo nutné materiály znovu nastavit po importování do editoru.

Šablona entit je další používaný pojem představující objektovou definici, která sdružuje typicky mesh, materiály, případně další komponenty a vlastnosti potřebné pro použití konkrétního objektu, nebo postavy ve světě hry. V případě postav (jako jsou v této práci vytvoření kostlivci nebo postava kováře) je použito například komponenty pro vzhled, která umožňuje takové šabloně přidat libovolné vzhledy (anglicky *Appearances*), jak má po použití této šablony vypadat konkrétní instance této postavy.

Označení environment definition (přeloženo jako „definice prostředí“), nebo také soubory typu `.env` jsou v této práci chápány jako data určující vizuální a atmosférické vlastnosti světa – např. intenzita světla (slunce / měsíce), mlha, barevné ladění prostředí atd.

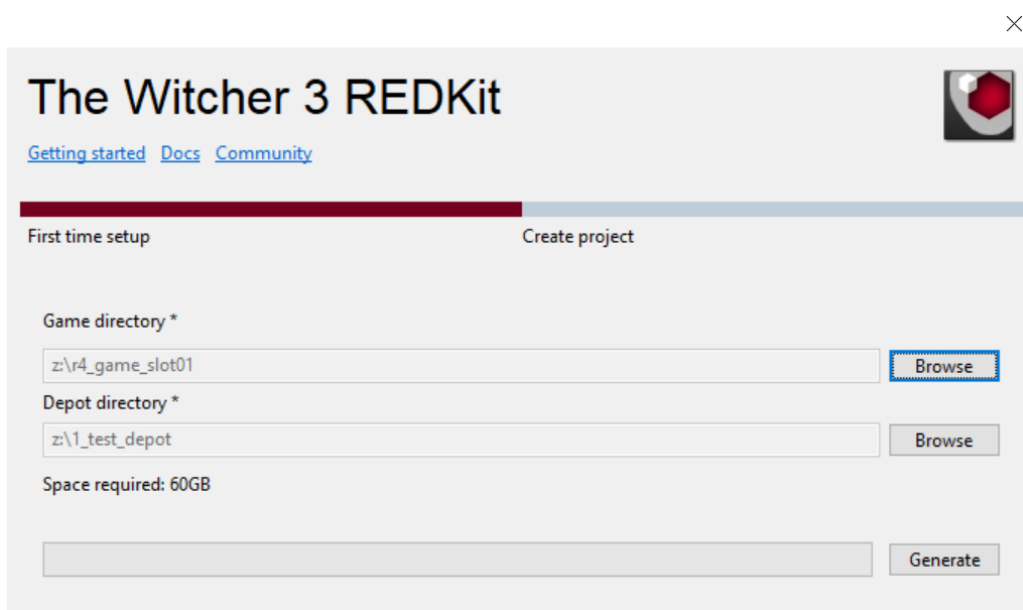
Zvuková složka modifikace byla realizována pomocí prostředků REDkitu a nástroje Wwise. V práci se proto objevují také odkazy na zvukové události, zvukové banky a audio soubory, které slouží k přehrání ambientní hudby, nebo jiných zvukových efektů v konkrétních částech herního světa.

3.3 Požadavky a zavedení REDkitu do systému

Uživatel může získat REDkit zdarma, ale použitelný je pro uživatele jen pokud je vlastníkem počítačové hry Zaklínač 3: Divoký hon na jedné z platform: Steam, Epic Games Store nebo GOG. V rámci této bakalářské práce byla zakoupena oficiální kopie hry z platformy Steam a následující postupy pro zavedení REDkitu byly aplikovány z platformy Steam. Dále je popsán postup, jak zprovoznit editor REDkit.

3.3.1 Stažení editoru REDkit

Lze provést z aplikace Steam na počítači, nebo využít jejich oficiálních stránek [17].



Obr. 2: Snímek úvodního okna po prvotním spuštění REDkitu [18]

3.3.2 Prvotní spuštění REDkitu

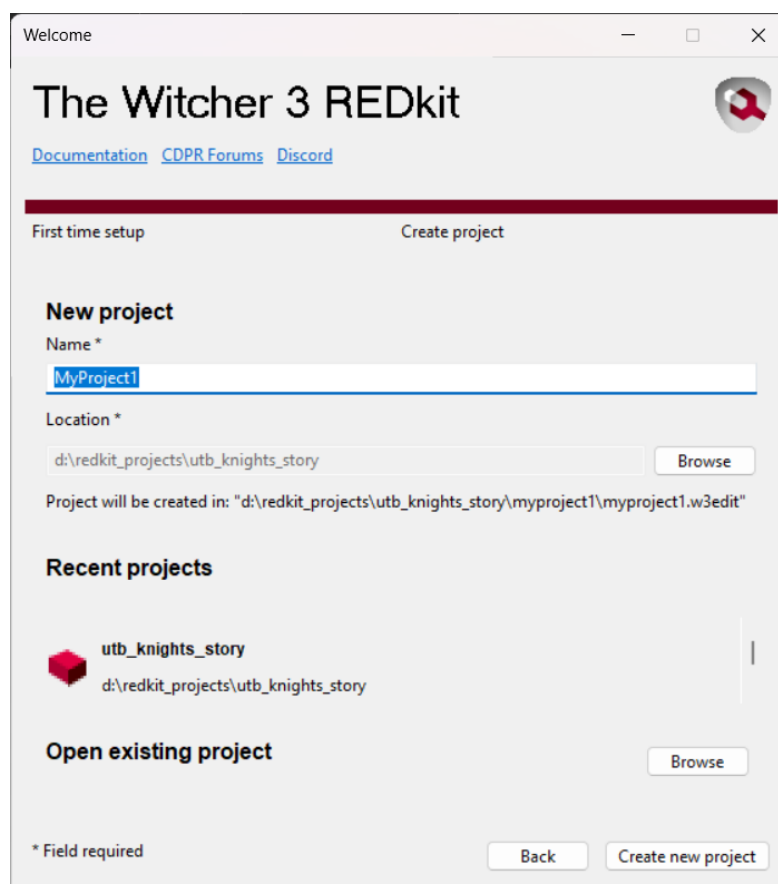
Při prvním spuštění REDkitu je nutné vygenerovat (tento proces popisuje dokumentace anglickým výrazem *uncook*) určité soubory ze hry Zaklínač 3, kterou musí mít uživatel již nainstalovanou v systému. Úvodní obrazovka viz Obr. 2 po spuštění obsahuje tato pole:

- Game directory (překlad: herní složka)
- Depot directory (překlad: složka depotu)

Cestu k herní složce musí uživatel nastavit podle toho, kde má nainstalovanou hru Zaklínač 3. Validní cesta pro nastavení cesty ke složce s hrou může vypadat například takto: “C:\SteamLibrary\steamapps\common\The Witcher 3\”. Dále uživatel musí nastavit cestu, kam bude zaveden tzv. depot, tedy místo, kde budou rozbaleny veškeré potřebné soubory, které REDkit potřebuje ke svému fungování. Tuto cestu si uživatel může zvolit libovolně, doporučuje se konvence: “C:\REDkit\depot\” a podobné. Na závěr stačí na úvodní obrazovce stisknout tlačítko “Generate”, které zahájí rozbalování souborů pro REDkit.

3.3.3 Vytvoření projektu

Po úspěšném prvním spuštění bude každé další spuštění nabízet již založení nového projektu viz Obr. 3, případně otevření existujících projektů, nebo nedávno otevřených. Uživatel musí zvolit název nového projektu a umístění, kde se vytvoří struktura projektu.



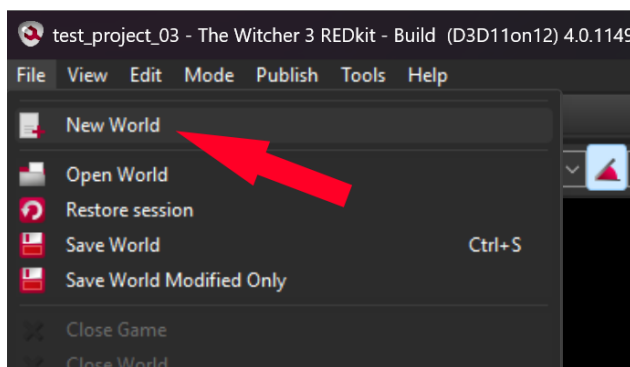
Obr. 3: Snímek okna pro vytvoření nového projektu
(zdroj: vlastní)

3.4 Základní práce s REDkitem

3.4.1 Vytvoření nového světa

Nový svět lze vytvořit, pokud uživatel chce tvořit novou lokaci. Svět (anglicky *World*) lze považovat za úroveň (anglicky *level*). Pod pojmem “svět” si může uživatel představit celou oblast, například celou mapu Novigrad, včetně všeho, co se v této lokaci děje.

V REDkitu po načtení projektu se nový svět vytvoří kliknutím na File – New World. Následuje nabídka pro zadání jména světa a případně názvu DLC. Svět je po tomto kroku vytvořený, zcela prázdný a nachystaný pro další práci s ním. V prvé řadě se obvykle pokračuje vytvořením terénu a jeho textur, tyto postupy jsou popsány v dalších kapitolách.

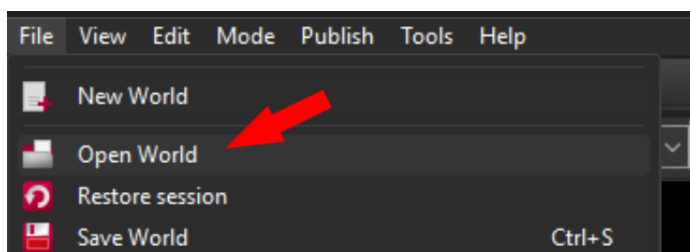


Obr. 4: Snímek nabídky pro vytvoření nového světa

(zdroj: vlastní)

3.4.2 Načtení existujícího světa

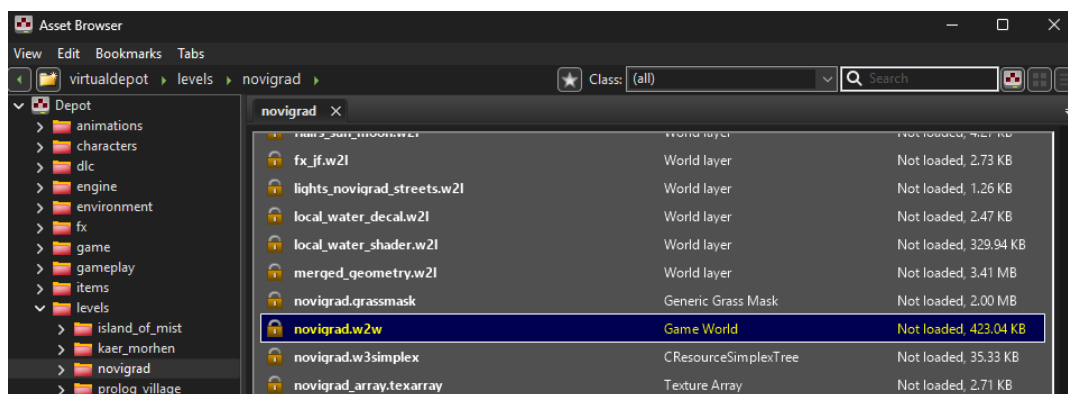
Uživatel může do projektu načíst mimo svůj svět (anglicky *world*), který vytváří, také předvytvořené světy, které se vyskytují v originální hře, například kompletní svět Novigrad. Postup je v obou případech stejný: v Editoru je potřeba kliknout na File – Open World, poté se otevře Asset Browser, který funguje jako průzkumník souborů – jsou zde umístěny veškeré prostředky pro tvorbu světa, ať už originální od tvůrců hry, nebo vlastní vytvořené uživatelem.



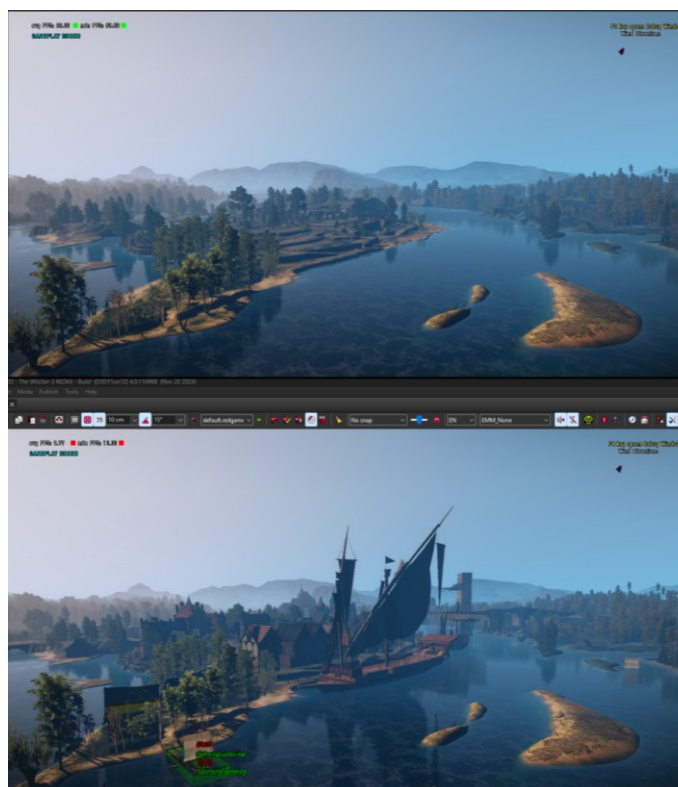
Obr. 5: Snímek nabídky pro otevření existujícího světa

(zdroj: vlastní)

V Asset Browseru je vše uloženo v kořenové složce virtualdepot (také nazvané Depot), ve složce levels jsou pak podle konvence uloženy jednotlivé levely včetně vytvořených uživatelem. Lze si tak vybrat a načíst libovolný level, ze kterého pak uživatel může čerpat zdroje, modifikovat atd. Soubor obsahující level má vždy formát souboru “.w2w” a v Asset Browseru je označen jako typ “Game World”.



Obr. 6: Snímek obrazovky s Asset Browserem zobrazující svět novigrad.w2w (zdroj: vlastní)

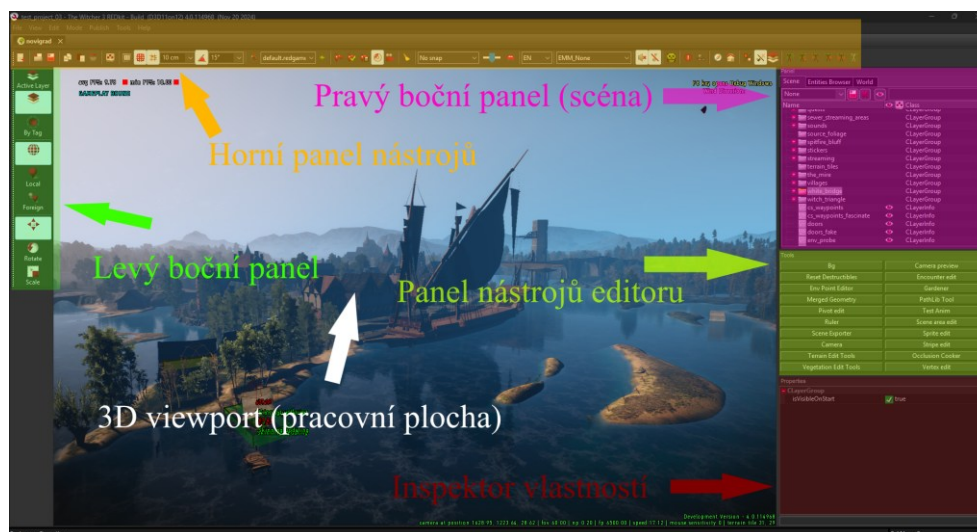


Obr. 7: Ukázka rozdílu mezi nenačtenými objekty (horní část) a načtenými objekty (spodní část) (zdroj: vlastní)

Po otevření vybraného levelu – v ukázce novigrad.w2w – se provede načtení, v editoru je viditelný terén, flóra a vodní plochy. Nejsou viditelné objekty, které lze očekávat v hotovém levelu. Je zapotřebí v pravé části editoru v oblasti Panel načíst to, s čím chce uživatel pracovat – klidně i kompletní mapu – a to lze provést stisknutím pravého tlačítka na požadovanou složku, což vyvolá kontextové menu, kde pomocí “Load all in group” načte všechno pro danou oblast. V ukázce viz Obr. 7 je viditelný rozdíl mezi nenačtenými objekty a načtenou oblastí “white_bridge”.

3.4.3 Hlavní obrazovka editoru

Hlavní obrazovka editoru REDkit se zobrazí po načtení projektu. Lze ji rozdělit do několika funkčních částí.



Obr. 8: Popis částí hlavní obrazovky editoru

(zdroj: vlastní)

3.4.3.1 Pracovní plocha (3D Viewport)

Pracovní plocha, dále viewport, je hlavní plocha pro vizuální práci se scénou. K pohybu po scéně uživatel ovládá kameru, kterou se lze pohybovat v prostoru. Uživatel může také označit a vybrat viditelné objekty ve scéně, popřípadě s nimi pohybovat, nebo měnit jejich velikost a změny vidí v reálném čase. V nastavení View – Settings lze vybrat konkrétní filtry, které uživateli zviditelní (nebo zneviditelní) zadané vlastnosti, například renderování světla, zobrazení oblastí (anglicky *Areas*), spouštěčů atd.

3.4.3.2 Horní panel nástrojů

Slouží jako hlavní řídicí centrum editoru. Obsahuje jak rychlé akce a přepínače, které ovlivňují práci napříč projektem a scénou, tak možnosti nastavení editoru samotného, publikování modu, spuštění právě modifikovaného levelu přímo v REDkitu apod.

3.4.3.3 Levý boční panel

Obsahuje rychlé akce, které jsou přednastavené developery REDkitu. Lze si zvolit způsob výběru buď z aktivní vrstvy, nebo Multi-layer režim. Přítomen je i nástroj pro hledání entit podle jejich tagu (označení v rámci REDkitu). Další rychlé akce umožňují přepínat Transform (World, Local, Foreign) a poslední rychlé akce umožní zvolit, který manipulační nástroj objektu má REDkit použít na editovaný objekt (Move pro posunutí, Rotate pro rotaci a Scale pro škálování).

3.4.3.4 Pravý boční panel (scéna)

Panel pro správu obsahu scény a organizaci projektu, zobrazení údajů o přítomných entitách nebo aktuálním světě. Panel po prvotním otevření má v sobě záložky:

- Scene (česky *Scéna* – myšleno to, co uživatel vidí zobrazené ve viewportu)
- World (česky *Svět*)
- Tools (česky *Nástroje*)
- Properties (česky *Vlastnosti*)
- Entities Browser (česky *Prohlížeč Entit*)

Záložky lze libovolně vytahovat z panelu a popřípadě usadit přímo v pravém bočním panelu a vytvořit více oddílů panelu. V rámci této bakalářské práce je použito rozložení viz Obr. 8. Resetování všech záložek do původního stavu – všechny záložky vedle sebe v pravém bočním panelu – lze provést kdykoliv kliknutím na horní panel nástrojů: View – Panel – All to tab. Dále následuje popis s aplikovaným rozložením pro tuto práci.

Typicky si uživatel může zorganizovat při vytváření nového světa souborovou strukturu v záložce Scény (Scene), aktivovat a načíst pouze vybrané oblasti, se kterými pracuje, aby zbytečnými objekty nezatěžoval své zařízení. Pokud je označena entita v tomto panelu, uživateli se zobrazí údaje v dalším panelu (Inspektor vlastností) a také vidí změnu ve Viewportu, kde se mu zobrazí u daného objektu ovládací prvek (v anglické literatuře často používaný výraz *gizmo*) umožňující manipulaci s objektem.

Další záložky obsahují informace o aktuálním světě (záložka World), nebo také prohlížeč entit (Entity Browser) umožňující uživateli vyhledat instance entit použité v aktuálním světě.

3.4.3.5 Panel nástrojů editoru

Tento panel slouží jako “spouštěč” specializovaných editorů a pracovních modulů, které jsou popsány dále v této práci v podkapitole *Nástroje REDkitu*.

3.4.3.6 Inspektor vlastností

Tento panel je kontextový panel zobrazující parametry právě vybrané entity / objektu. Podle aktuálního kontextu tedy zobrazí například přepínače, číselné hodnoty, popřípadě reference na další objekt a ostatní údaje pro nastavení vlastností objektu. Často je klíčový pro jemné doladění chování, viditelnosti, vazeb a konfigurace objektů.

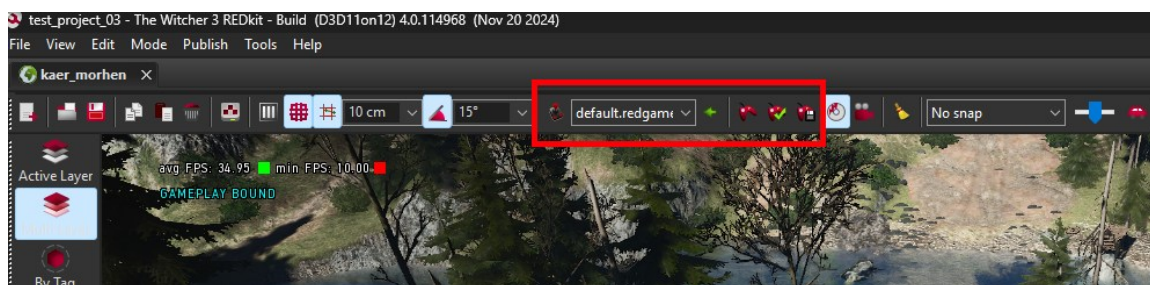
3.4.4 Ovládání kamery

Pro pohyb ve scéně je potřeba mít schopnost ovládat kameru viewportu. REDkit umožňuje resetovat, popř. vycentrovat kameru skrze horní panel nástrojů (záložka Edit) a zkratky pro běžné ovládání kamery pomocí klávesnice a myši následujícím způsobem:

- W, S, A, D – pohyb dopředu, dozadu, doleva, doprava
- Q, E – pohyb nahoru a dolů
- SHIFT – držení klávesy pro rychlejší pohyb + směrová klávesa (WSAD)
- Levé tlačítko myši – držením tlačítka a pohybem myši lze také pohybovat kamerou
- ALT + Pravé tlačítko myši – držením tlačítek a pohybu myši lze měnit rotaci kamery

3.4.5 Spuštění hry v editoru

REDkit nabízí možnost testovat spuštěním právě modifikovaného světa přímo uvnitř editoru. Uživatel může spustit hru v editoru pomocí tlačítek rychlých akcí nacházející se na horním panelu nástrojů. Spuštění hry, jinými slovy „zrození“ postavy Geralta do herního světa, lze v REDkitu provést více způsoby, každý má své použití.



Obr. 9: Snímek zobrazující tlačítka pro spuštění hry (zvýrazněno červeným obdelníkem)
(zdroj: vlastní)

3.4.5.1 Zrození postavy do herního světa

Postavu Geralta lze zrodit (anglicky *spawn*) do herního světa pomocí tlačítek viz Obr. 9 zobrazených jako červený herní ovladač.

První (pouze ovladač) postavu Geralta zrodí v herním světě na pozici 0, 0, 0 (osy X, Y, Z herního světa) a načte vrstvy světa.

Druhé tlačítko (ovladač se zelenou fajfkou) načte pouze vrstvy herního světa, které má uživatel při editaci světa zrovna načteny a Geralta zrodí v přesném místě, kde má uživatel v době stisknutí tlačítka umístěnou kameru – pokud bude kamera vysoko ve vzduchu, Geralt se objeví také ve vzduchu a po dopadu postava zemře.

Třetí variantou spuštění je načtení uložené hry – ikona červeného herního ovladače s obrázkem diskety (viz Obr. 9 ikona poslední vpravo uvnitř ohraničené části) – který načte hru ze souboru obsahující uložený stav hry. Po načtení uložené hry se veškeré neuložené změny v editovaném světě vymažou, je tedy důležité před načtením hry změny uložit. Tento způsob nenačítá uložené hry (pozice / stavy), ale načítá svět podle kontextu, který je v datech uložené pozice uveden – ne svět, který je aktuálně otevřen v editoru. Oficiální dokumentace

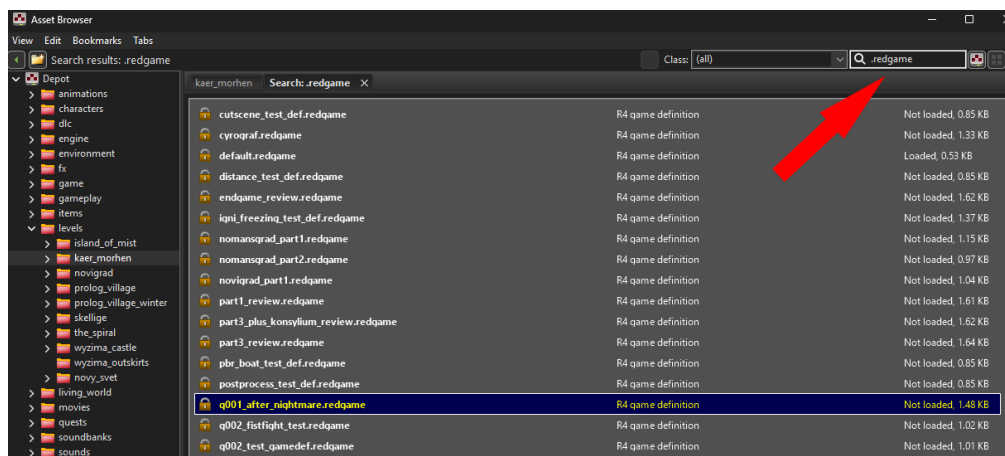
REDkitu [19] doporučuje mít soubory s uloženými pozicemi ve složce na počítači: Documents – Witcher 3 – gamesaves.

3.4.5.2 Spuštění hry z herní definice

Pro spuštění hry z vybraného bodu, úkolu atd. je zapotřebí použít soubor herní definice (anglicky *game definition file*). K výběru souboru herní definice se používá rozbalovací seznam, defaultní herní definice je „default.redgame“.

Další soubory herní definice lze najít v Asset Browseru pomocí vyhledávacího pole zadáním „.redgame“ viz Obr. 10. Ze seznamu nalezených souborů lze označit požadovaný soubor herní definice a kliknutím na zelenou šipku viz Obr. 9 - uprostřed červeně ohraničené zóny – se přidá do rozbalovacího seznamu vybraná herní definice, kterou poté lze spustit tlačítkem „joysticku“.

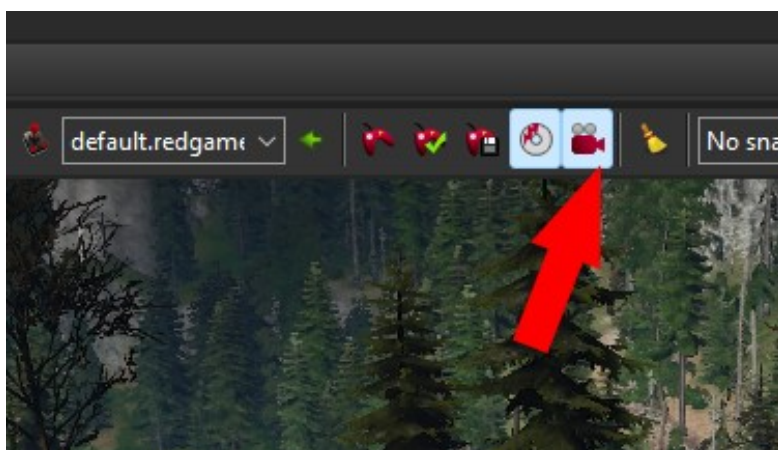
Po spuštění se načte herní svět spojený s herní definicí a hra se v editoru spustí v místě, které definuje soubor herní definice. Svět, který měl uživatel načtený v editoru předtím, je nahrazen světem z herní definice.



Obr. 10: Snímek s označením vyhledávacího pole Asset Browseru

(zdroj: vlastní)

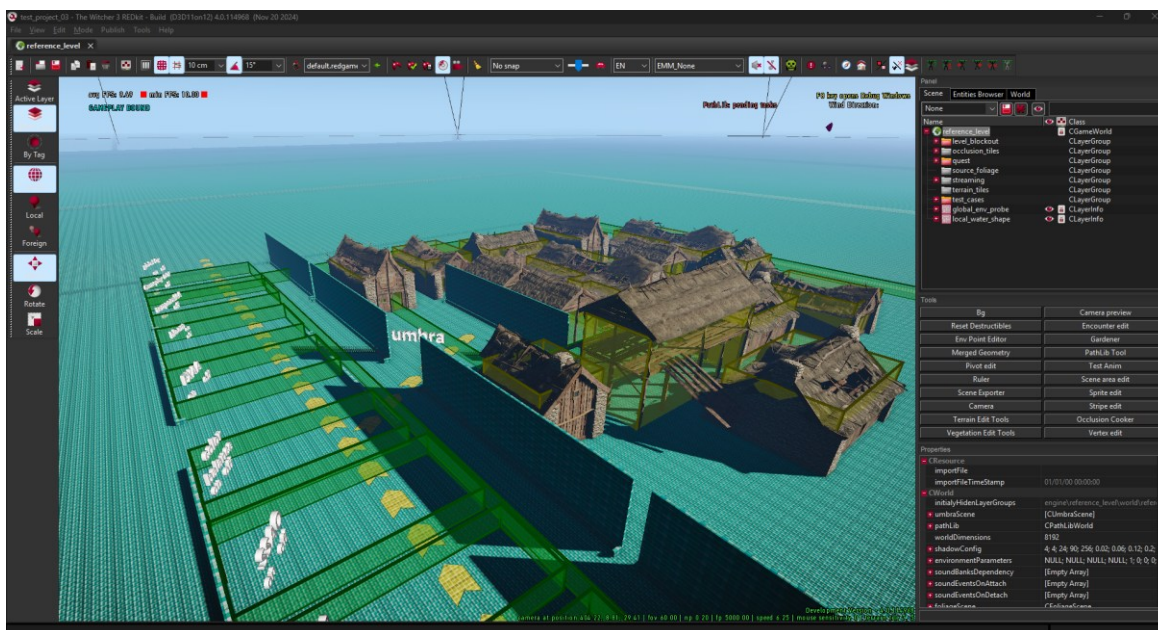
Pro návrat do editoru po spuštění hry se používá klávesa F10. Pokud uživatel chce zachovat po návratu pozici kamery, kde během testovacího hraní v editoru skončil, je potřeba mít aktivované tlačítko “Leave game camera position” umístěné na horním panelu nástrojů (viz Obr. 10) buď před spuštěním hry, nebo během hraní v editoru.



Obr. 11: Snímek s označením tlačítka pro zachování pohledu kamery
(zdroj: vlastní)

3.4.6 Referenční svět

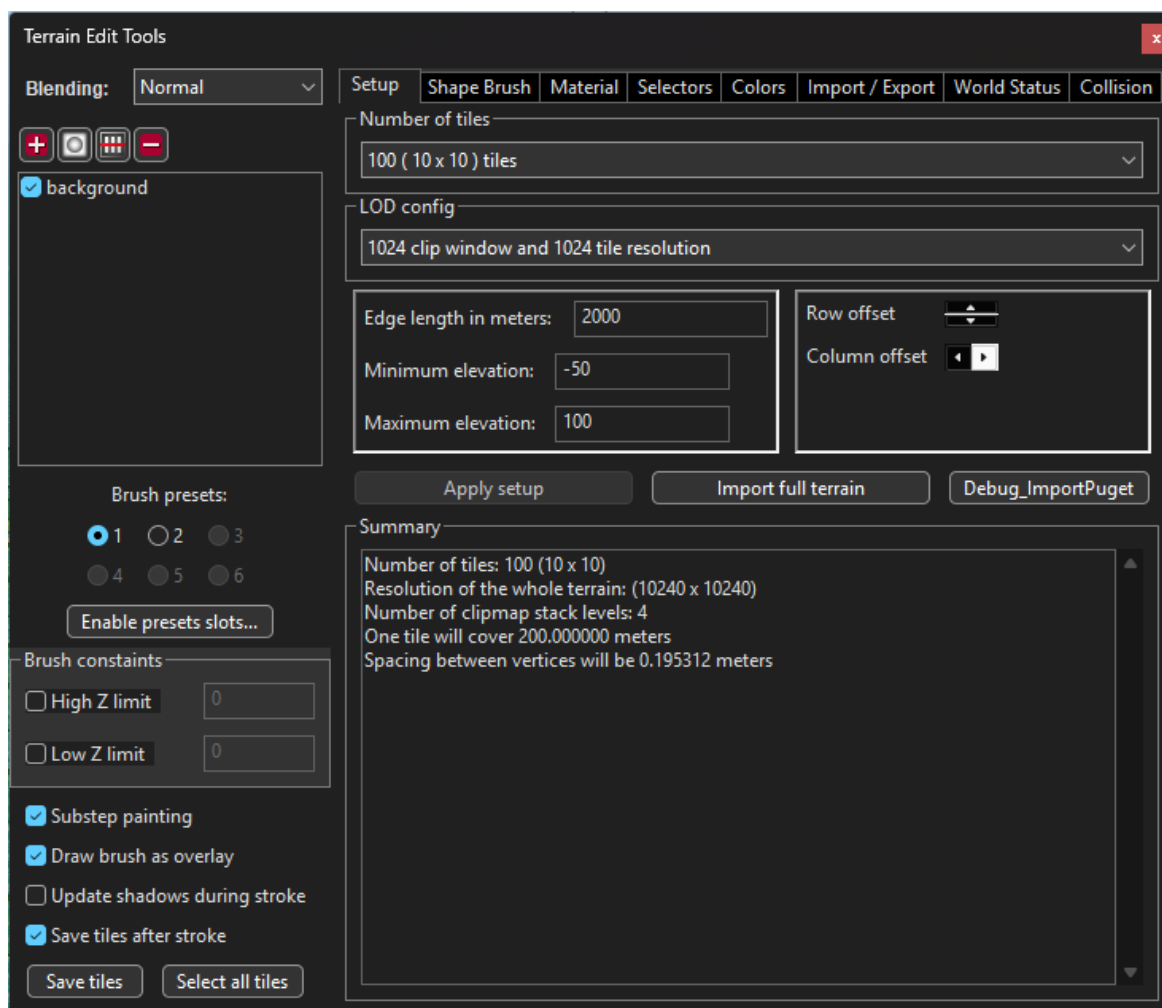
REDkit nabízí tzv. referenční svět (anglicky *Reference level*), kde může uživatel prozkoumat, studovat a testovat funkcionality, které REDkit nabízí. Referenční svět lze spustit z Asset Browseru, pomocí vyhledávacího pole, kde je potřeba zadat “reference_level.w2w”, nebo ho lze najít v Depot ve složce: engine\reference_level\world\.



Obr. 12: Snímek ukázky referenčního světa
(zdroj: vlastní)

3.5 Nástroje REDkitu

Tato podkapitola shrnuje specializované nástroje dostupné v REDKitu, které slouží k tvorbě a úpravám herního světa, entit, úkolů, animací a dalších herních systémů. REDkit obsahuje sadu integrovaných editorů, které se spouští z panelu nástrojů editoru a typicky pracují nad aktivním světem nebo entitou.



Obr. 13: Snímek okna Terrain Edit Tools zobrazující záložku Setup
(zdroj: vlastní)

3.5.1 Nástroje pro editaci terénu

Tyto nástroje jsou důležité pro vytvoření herní mapy. Jsou seskupeny v rámci okna “Terrain Edit Tools” do několika záložek:

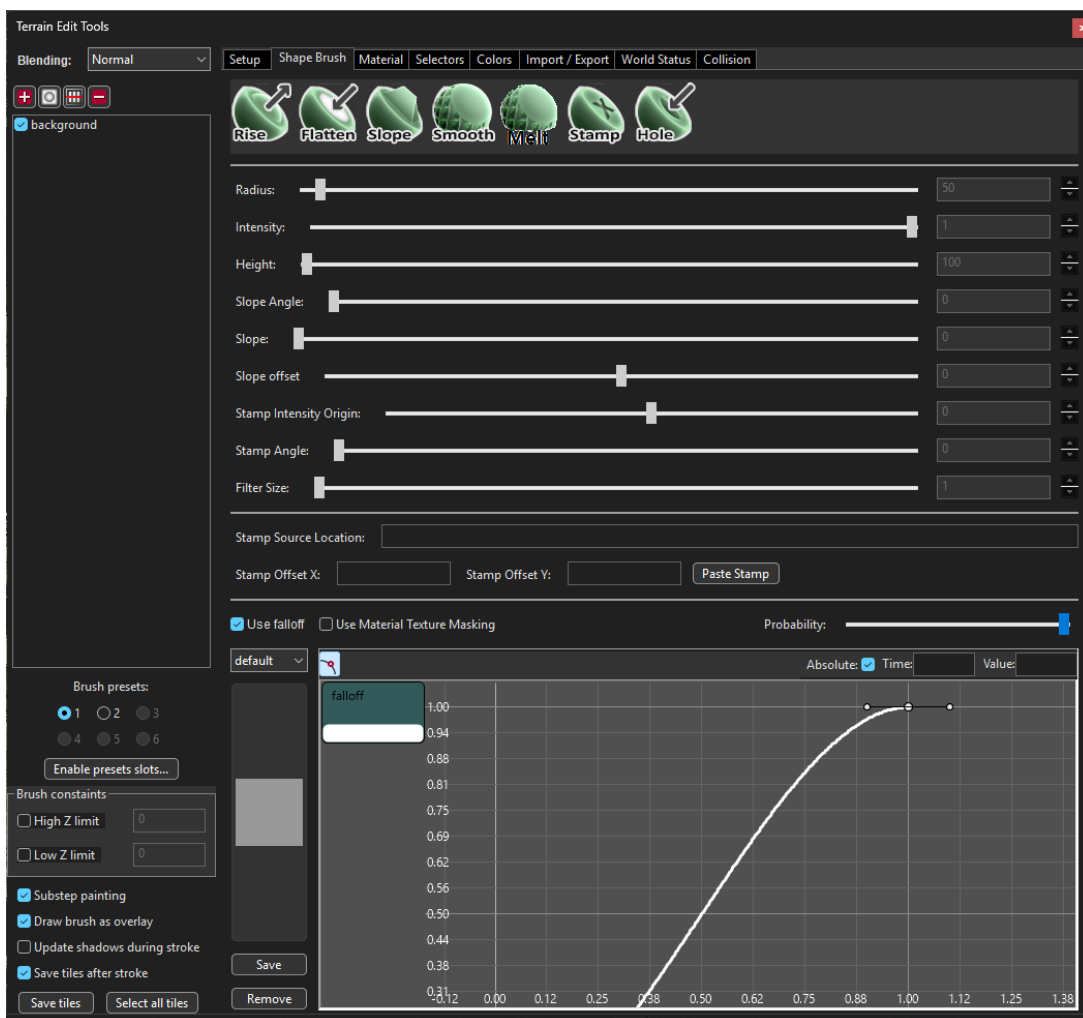
- Setup
- Shape Brush
- Material

- Selectors
- Colors
- Import / Export
- World Status
- Collision

3.5.1.1 Setup

Tento nástroj viz Obr. 13 umožňuje uživateli vytvořit terén s různou konfigurací. Lze například nastavit počet dlaždic (anglicky *tiles*), konfigurovat úroveň detailu terénu (anglicky *Level of Detail* – často používáno jako zkratka *LOD* i v jiném kontextu než terén), rozměr okrajů, minimální a maximální výška terénu.

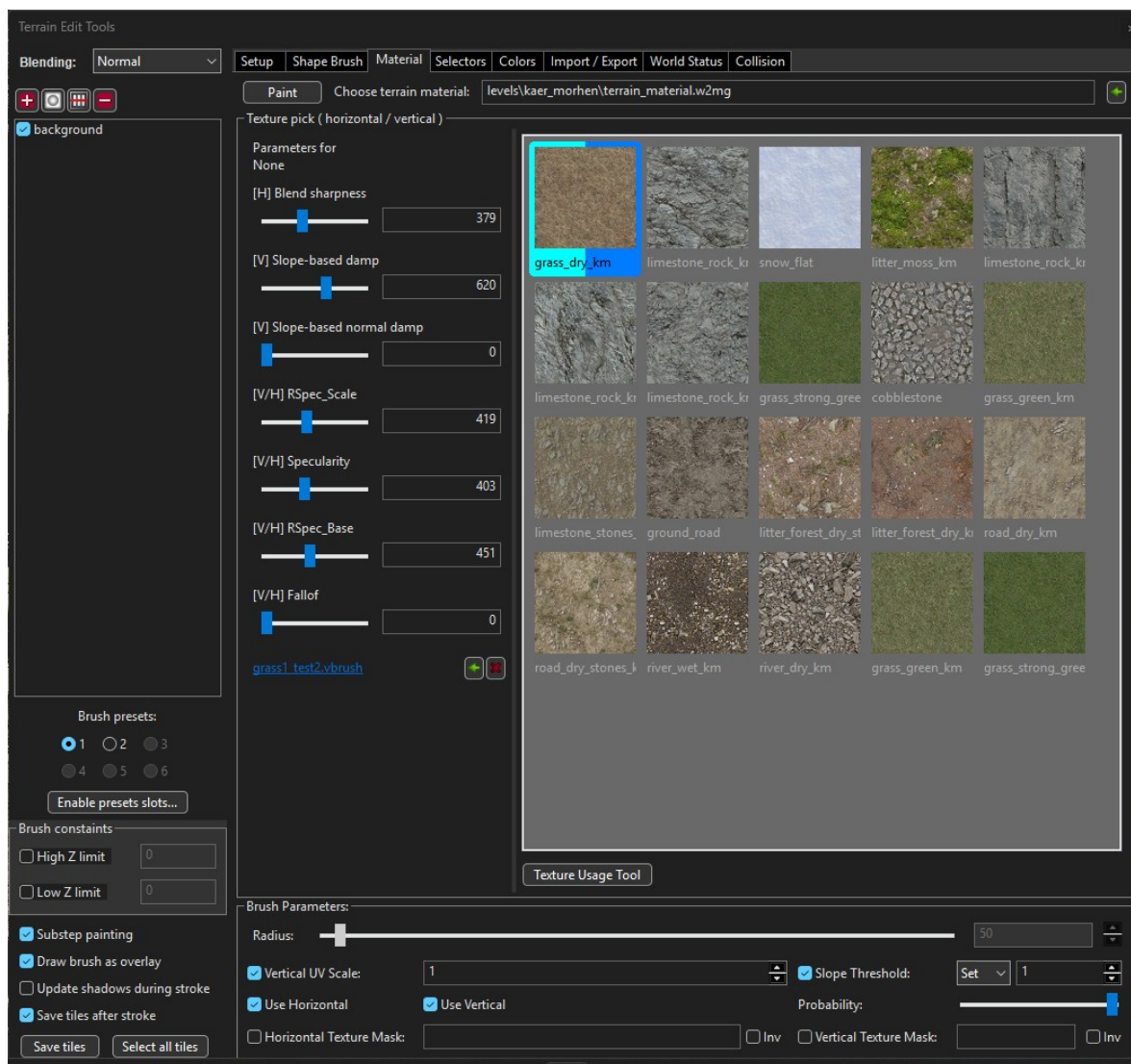
3.5.1.2 Shape Brush



Obr. 14: Snímek okna Terrain Edit Tools zobrazující záložku Shape Brush (zdroj: vlastní)

Zpřístupňuje uživateli řadu štětců (anglicky *Brush*), které při aplikaci tvarují terén požadovaným způsobem. Každý z nástrojů má specifické možnosti nastavení.

3.5.1.3 Material

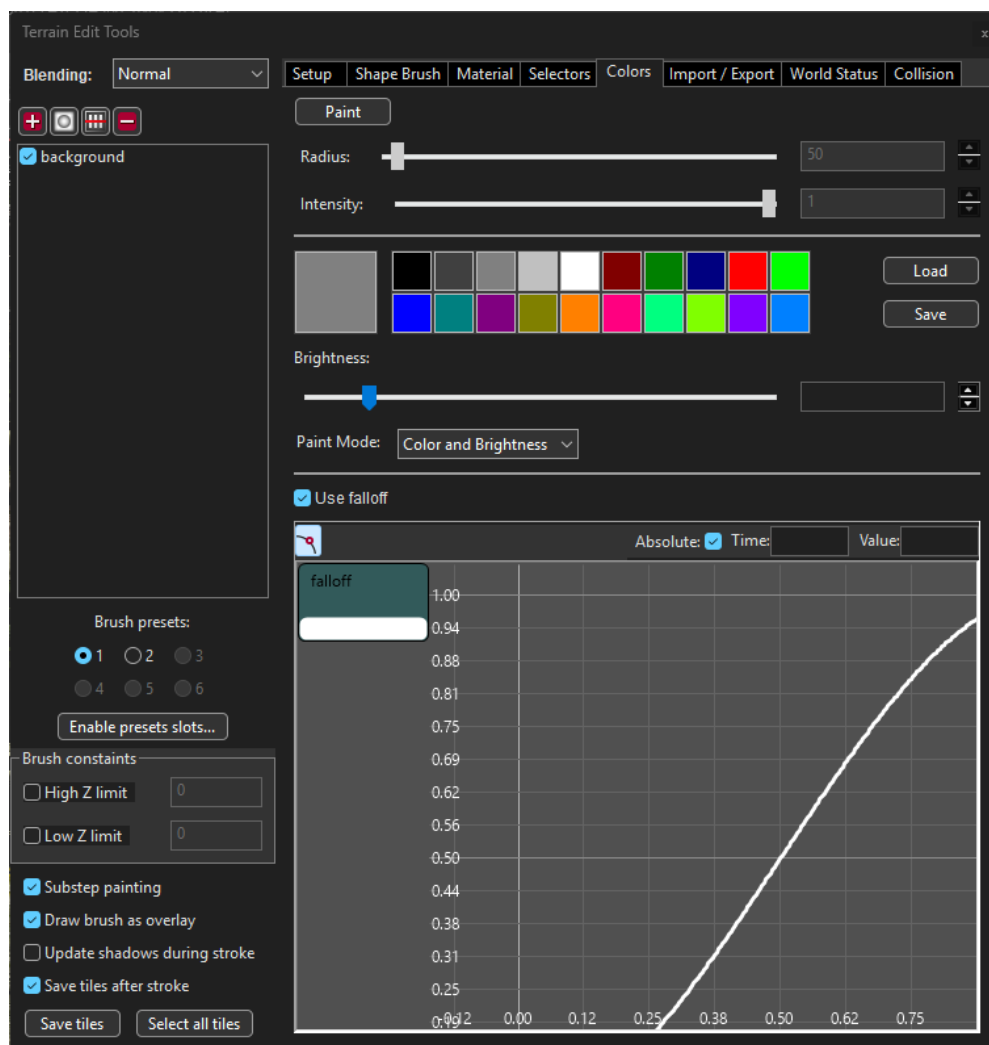


Obr. 15: Snímek okna Terrain Edit Tools zobrazující záložku Material (zdroj: vlastní)

Pomocí této záložky lze aplikovat na terén světa textury. Pro zvolení textur musí mít uživatel nejdříve nastavený materiál terénu, který musí vybrat z Asset Browseru označením souboru třídy CMaterialGraph (nebo soubor s koncovkou .w2mg) a následně kliknutím na zelenou šipku v okně Terrain Edit Tools v záložce Material viz Obr. 15 vedle pole pro výběr materiálu (pole "Choose terrain material") - tímto se vloží odkaz na cestu k označenému souboru obsahující data k texturám, ze kterých si následně může uživatel vybrat textury, které chce

použit pro kreslení. Uživatel může vybrat 1-2 textury pomocí levého a pravého tlačítka myši. Po označení textur, které chce uživatel použít na terén, může editovat nastavení štětce pomocí posuvníků, které mimo jiné nastavují způsob prolínání dvou vybraných textur. Po nastavení štětce může uživatel do 3D Viewportu přímo “kreslit” textury na terén. Pro přepínání mezi kreslícím modem ve viewportu a standardním modem slouží tlačítko Paint v záložce Material.

3.5.1.4 Colors

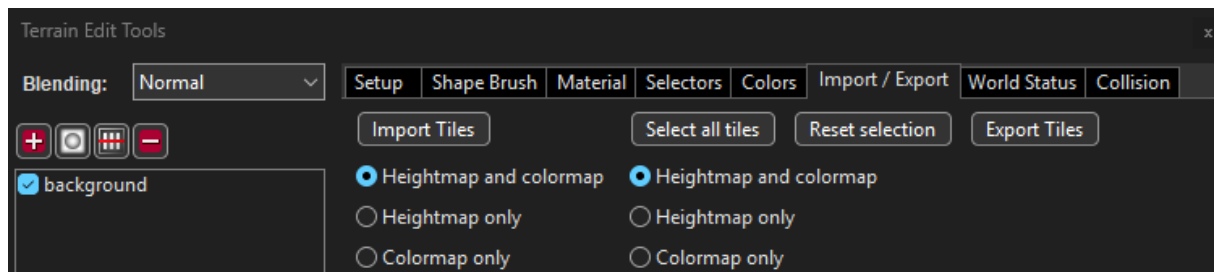


Obr. 16: Snímek okna Terrain Edit Tools zobrazující záložku Colors
(zdroj: vlastní)

Uvnitř záložky Colors může uživatel použít konkrétní barvy ke kreslení textury do terénu, podobně jako v záložce Material, s tím rozdílem, že kreslí přímo barvy místo textur.

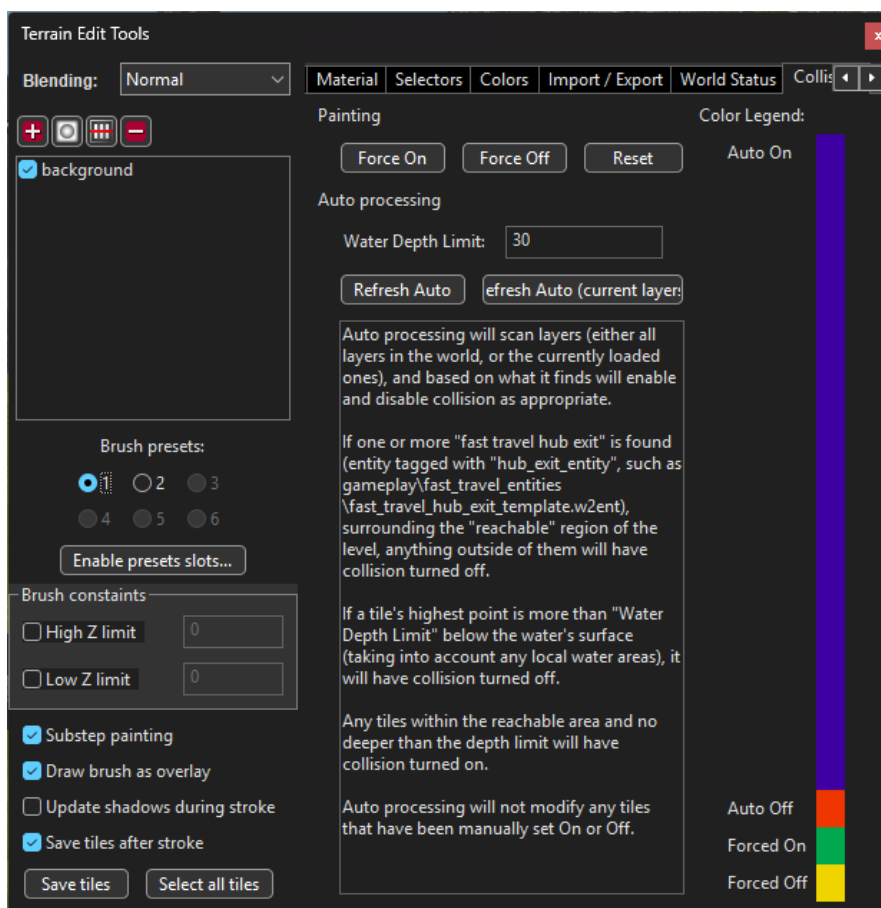
3.5.1.5 Import / Export

Pro možnost importování a exportování herní mapy REDkit nabízí nástroj v záložce “Import / Export“. Lze si zvolit mezi importováním / exportováním, výškové mapy (anglicky *Height map*), mapy barev (anglicky *Color map*), nebo obojí.



Obr. 17: Snímek okna Terrain Edit Tools zobrazující záložku Import / Export (zdroj: vlastní)

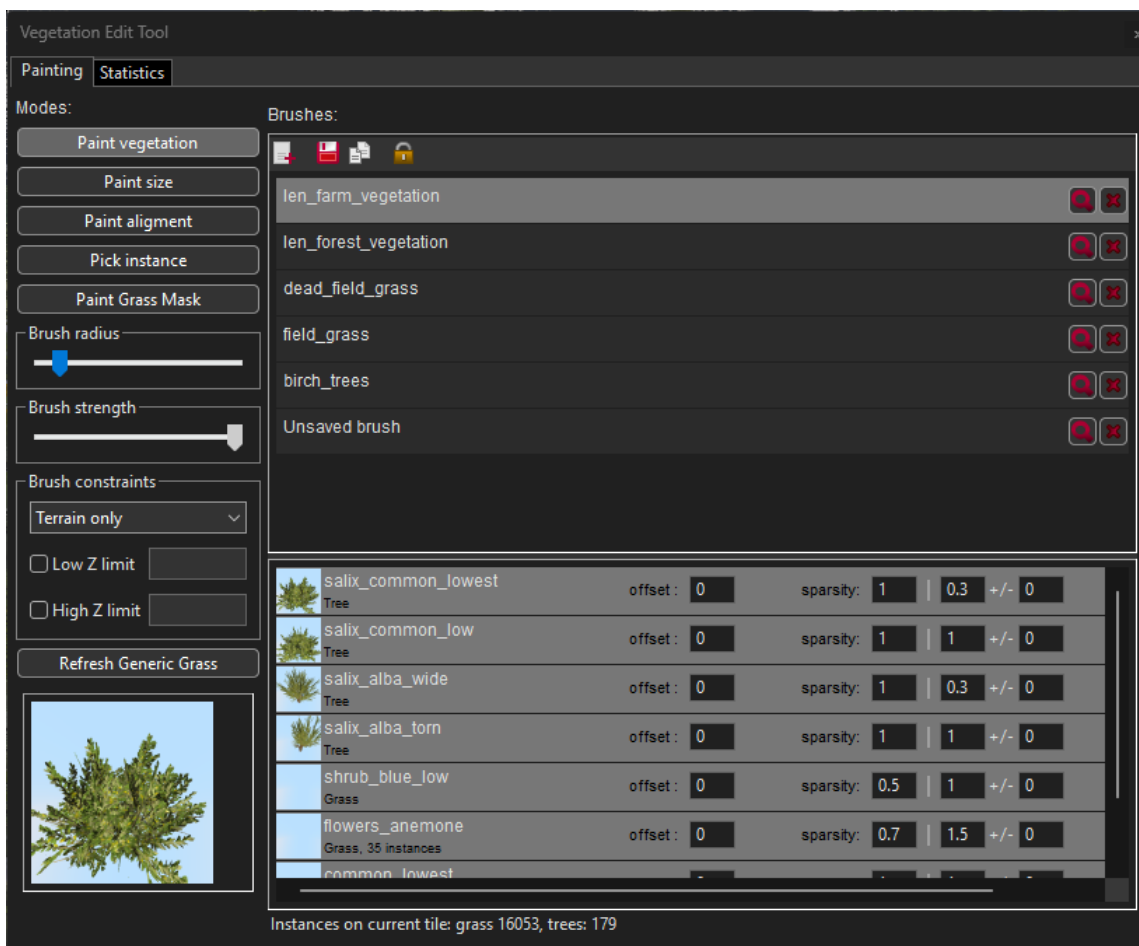
3.5.1.6 Collision



Obr. 18: Snímek okna Terrain Edit Tools zobrazující záložku Collision (zdroj: vlastní)

Při otevření záložky Collision se viditelně zbarví terén do zón modré, červené, zelené a žluté barvy viz Obr. 18 naznačující, kde je zapnutá, nebo vypnutá kolize s hráčem. V místech, kde je kolize vypnutá, se hráčova postava propadne terémem.

3.5.2 Nástroj pro editaci vegetace



Obr. 19: Snímek okna Vegetation Edit Tool

(zdroj: vlastní)

Nástroj pro upravování vegetace (anglicky *Vegetation Edit Tool*) se používá ke “kreslení” trávy, stromů, křoví atd. štětcem (anglicky *Brush*). Většinou je používán na terén nebo na statický mesh. Uvnitř nastavení nástroje viz Obr. 19 lze importovat více štětců, mezi kterými může uživatel přepínat. Další štětce lze importovat z Asset Browseru přetažením do okna Vegetation Edit Tool mezi ostatní štětce (anglicky *brushes*). Originální štětce jsou uloženy ve složce virtuálního depa: `virtualdepot/environment/vegetation/brushes`.

Po zvolení štětce se zobrazí po řádcích jednotlivé prostředky (anglicky *assets*), dále assety, které aktuálně zvolený štětec obsahuje s možnostmi nastavení pro offset a četnost výskytu. Uživatel má možnost buď kreslit všechny assety zároveň s nastavenou četností

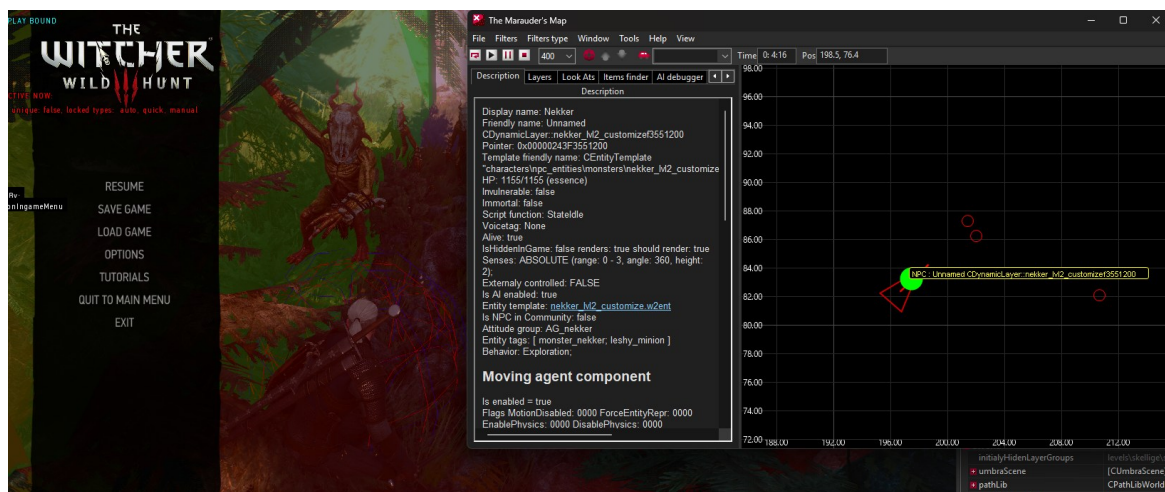
výskytu pro každý asset, nebo si zvolit libovolné assety označením a kreslit pouze je. Uživatel si rovněž může upravit nastavení štětce samotného (jeho poloměr, síla atd. podle potřeby).

Další možností nástroje pro editaci vegetace je kreslení masky pro trávu (anglicky *Grass Mask*). Po kliknutí na tlačítko “Paint Grass Mask” se označí zelenou barvou celý terén mapy a uživatel může pomocí štětce zakreslovat místa, kde označením do červené barvy tyto plochy vyjme z prostoru pro generování trávy.

Záložka “Statistics” uživateli dává informace o počtu instancí vegetace na dlaždicích, které ve viewportu uživatel kliknutím označí. Tímto by mělo být umožněno odstranit duplikáty instancí, pokud je jich někde příliš mnoho.

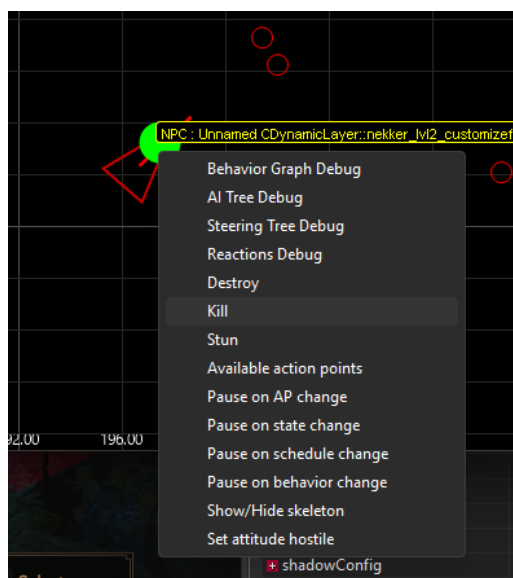
3.5.3 Marauder’s Map

Umožňuje uživateli získávat informace o objektech, příšerách a ostatních assetech okolo postavy hráče. Lze ji použít například k okamžitému zabítí příšery, pokud v rámci vývoje modifikace uživatel nechce ztrácet čas bojem.



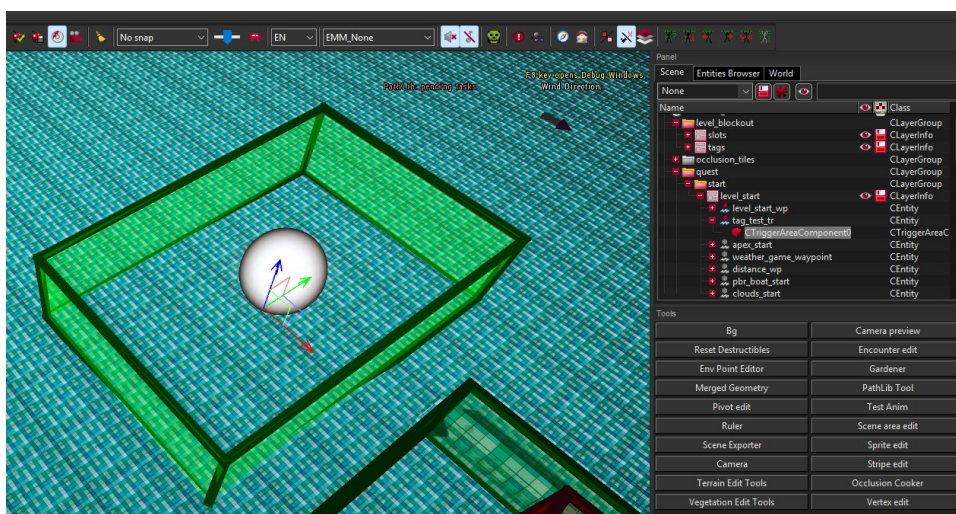
Obr. 20: Snímek zobrazující nástroj Marauder’s Map při identifikaci nepřátelské postavy (zdroj: vlastní)

Ukázka použití viz Obr. 20. Uživatel během testovacího hraní může kliknutím levého tlačítka myši získat informace o objektu, v případě ukázky (kolečko zelené barvy) se jedná o entitu se jménem “Nekker”, včetně spousty dalších dat. Entita označeného Nekker je viditelná v ukázce ve výskoku nad herní postavou Geralta, a zároveň je pozice entity naznačená ve viewportu červeno modrou sítí (v případě entity ve výskoku je síť o něco níž, než její vizuální pozice). V mapě jsou viditelné další kruhy (červené barvy). Pokud se uživatel nechce zabývat bojem s těmito entitami, může kliknutím pravého tlačítka myši v Marauderově mapě na libovolnou entitu vyvolat kontextové menu viz Obr. 21 a zvolit možnost “Kill”, která instantně zabije označenou entitu.



Obr. 21: Snímek zobrazující kontextové menu v Marrauder's Map nástroji (zdroj: vlastní)

3.5.4 Editace Vertexů

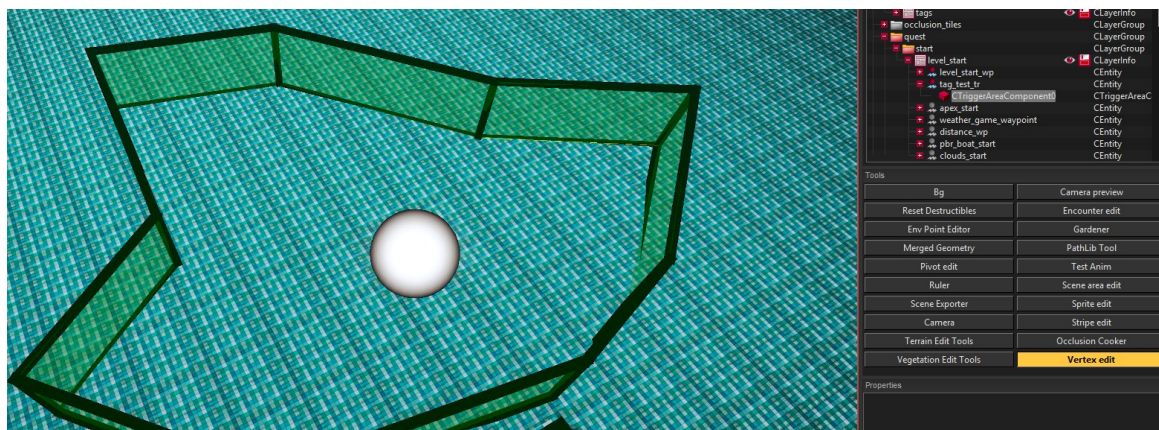


Obr. 22: Snímek zobrazující označenou oblast CTriggerAreaComponent0 před editací vertexů (zdroj: vlastní)

Spousta objektů jako jsou spouštěče (anglicky *Triggers*) a cesty (anglicky *Paths*) jsou sestaveny z vertexů. REDkit nabízí editační nástroj těchto vertexů pomocí „Vertex edit“ nástroje. Postup editace je následující:

- Uživatel musí mít označený objekt, který chce editovat v panelu scény (nebo jej najít a označit ve viewportu).

- V panelu nástrojů uživatel klikne na tlačítko „Vertex edit“, které změní barvu, čímž dává najevo, že je aktivní.
- Přidávání nebo odstraňování vertexů lze provést pomocí klávesy ALT + levé tlačítko myši.
- U editace cest (anglicky *paths*) lze nové vertexy na konec cest přidat pomocí CTRL + levé tlačítko myši.



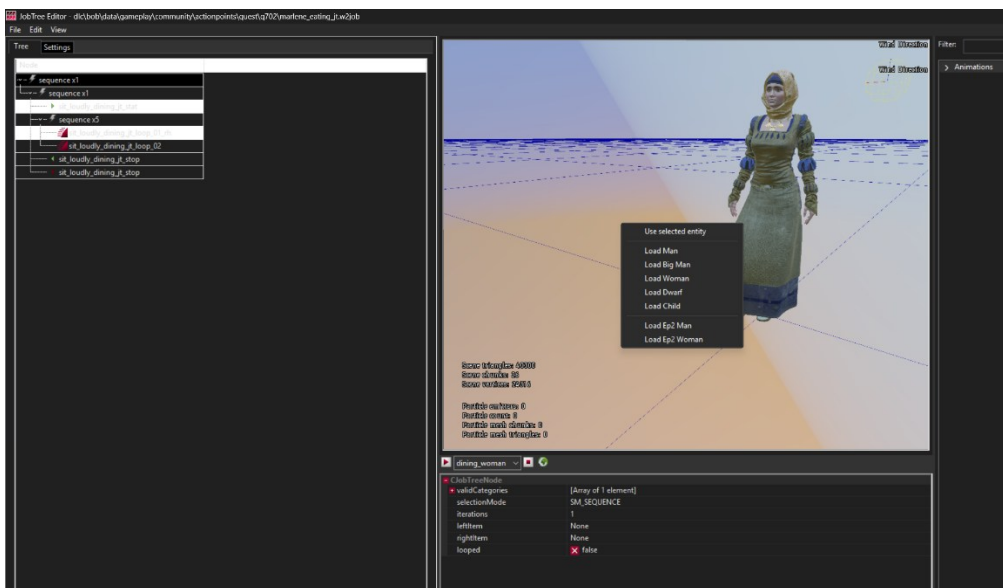
Obr. 23: Snímek zobrazuje oblast CTriggerAreaComponent0 po editaci vertexů (zdroj: vlastní)

Dokumentace REDkitu [20] varuje před křížením hran při upravování vertexů oblastí (anglicky *areas*) - mohlo by údajně dojít k pádu editoru.

3.5.5 Job Tree Editor

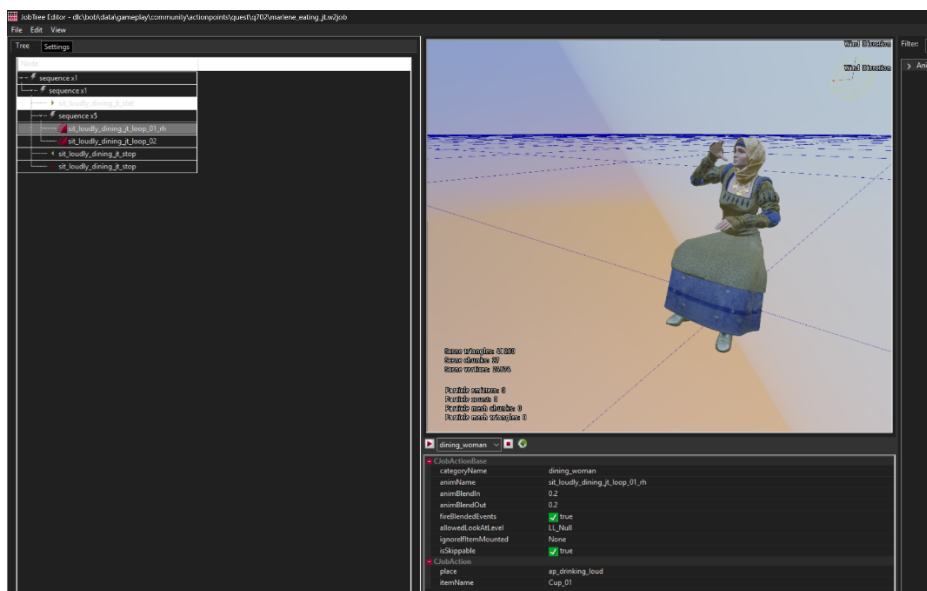
Pro nastavení chování postav ve hře a jejich propojení s animacemi slouží „Job Tree Editor“. Jednotlivé Jobs (česky *práce*) jsou sekvence akcí s jejich odpovídajícími animacemi. Příkladem by mohla být akce „Sednout si“ spojená s animací potřebnou pro postavu, aby si sedla. Každý „Job Tree“ je k nalezení v Asset Browseru, jedná se o soubory s koncovkou „w2job“, soubor je typu Job Tree. Lze jej filtrovat při hledání pomocí třídy CJobTree.

Editor je rozdělený do několika částí viz Obr. 24. Uprostřed se nachází viewport zobrazující aktuálně spuštěnou animaci. Je potřeba nejdříve načíst model entity, která má animaci ukazovat, to lze provést kliknutím pravého tlačítka myši do viewportu editoru. Zobrazí se kontextové menu s nabídkou několika možných postav k načtení. V ukázce viz Obr. 24 byl zvolen model ženy kliknutím na „Load Woman“.



Obr. 24: Snímek okna Job Tree Editor s otevřeným kontextovým menu pro výběr modelu postavy (zdroj: vlastní)

Pod viewportem editoru se zobrazí data k aktuálně označené akci, která se volí z levého panelu, kde jsou rozkreslena data k celému stromu prací (Job tree). V pravém panelu si uživatel může zobrazit dostupné animace, které se po kliknutí začnou přehrávat na aktuálně zvoleném modelu entity ve viewportu editoru. V ukázce viz Obr. 25 byla v levém panelu označena animace “sit_loudly_dining_jt_loop_01_rh”, která se po označení začala přehrávat ve viewportu. Okamžik animace byl zachycen.



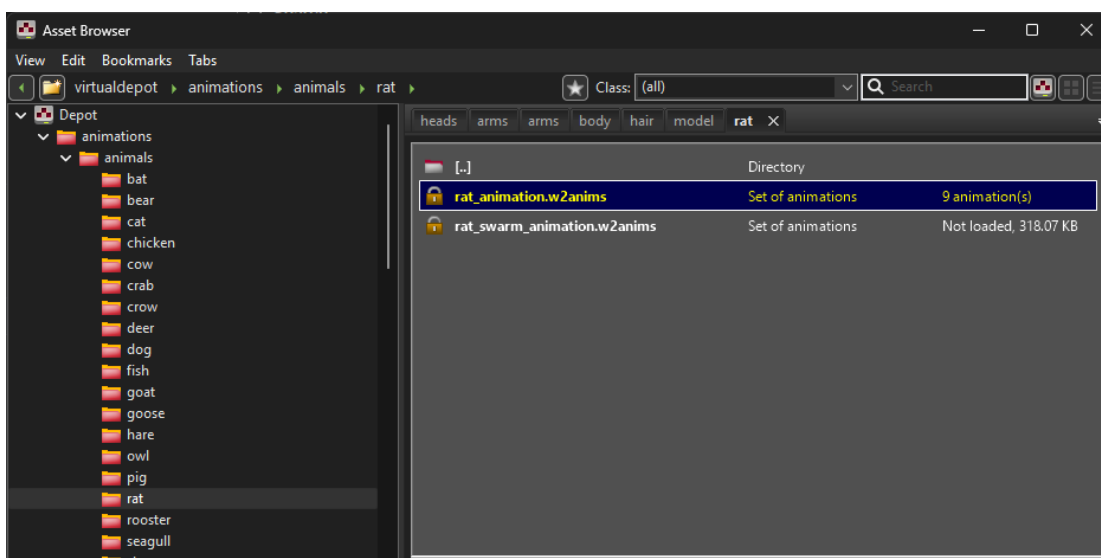
Obr. 25: Snímek okna Job Tree Editor s přehrávanou animací sit_loudly_dining_jt_loop_01_rh (zdroj: vlastní)

3.5.6 Editory animací

Vzhledem k tomu, že animační soubory jsou z REDkitu exportovány ve formátu HDF, který je open source, oficiální dokumentace [21] navrhuje použití nástroje HDFView [22] ke zkoumání a editaci animačních souborů pomocí skriptů. Druhou variantou je exportování a editace v software Blender [23]. Více animací najednou je seskupováno do souboru typu „Set of animations“. V Asset Browseru jsou animační soubory dohledatelné s koncovkou „w2anims“.

3.5.6.1 Příprava animačního souboru

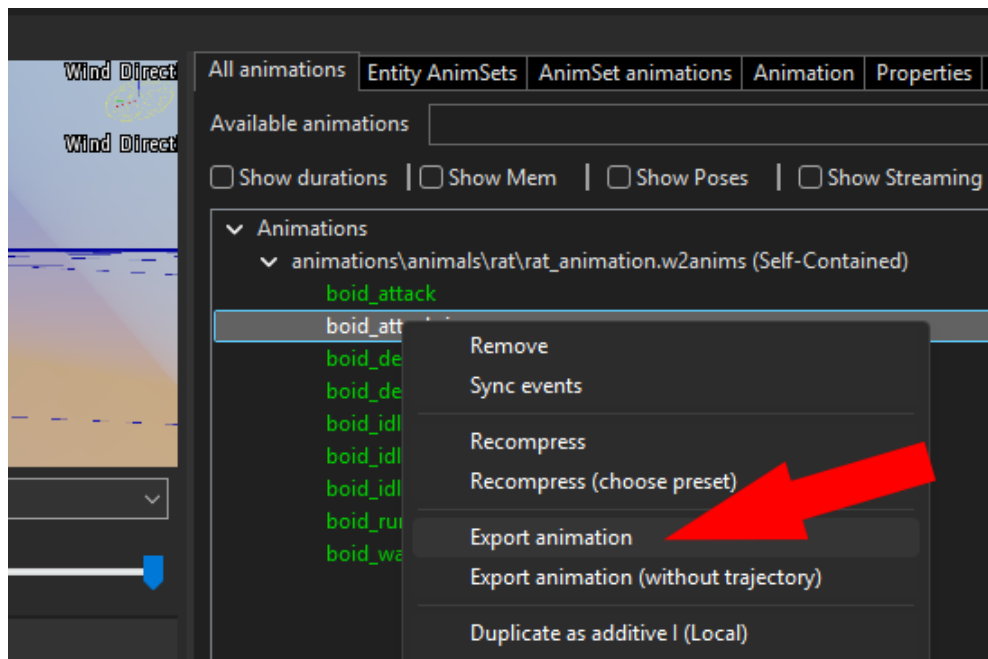
Pro přípravu editace animace musí být nalezen odpovídající soubor v Asset Browseru, například pokud bude uživatel chtít editovat animaci pro entitu krysy, lze ji nalézt ve složce „Depot/animations/animals/rat“, v ukázce Obr. 26 soubor „rat_animation.w2anims“ je soubor typu „Set of animations“ a obsahuje 9 animací – tento soubor lze otevřít a načte se do nástroje REDkitu nazvaný AnimBrowser.



Obr. 26: Snímek zobrazuje animační soubor rat_animation v Asset Browseru (zdroj: vlastní)

V otevřeném AnimBrowseru uživatel uvidí viewport pro zobrazování průběhu animace, seznam jednotlivých animací obsažených v otevřeném animačním souboru a časovou osu s průběhem animací. Nejprve je potřeba vybrat šablonu entity (anglicky *Entity template*), která bude vhodná pro animaci – jedná se o soubor třídy *CEntityTemplate* s koncovkou „w2ent“. V Asset Browseru lze pomocí filtrování Class použít *CEntityTemplate* a v hledacím poli zadat výraz „rat_“, a nalézt tak šablonu entity pro krysu „rat_aggressive.w2ent“. Tento soubor je potřeba označit kliknutím levého tlačítka myši a poté lze v AnimBrowseru kliknout pravým tlačítkem myši do viewportu, objeví se kontextové menu a možností „Use selected entity“ se načte označená entita jako model pro přehrávání animací. V ukázce Obr. 26 je

viditelná animace “boid_attack_jump” ve svém průběhu pro model vybrané krysy. V časové ose ve spodní části okna AnimBrowseru lze přidat nové stopy (anglicky *tracks*) kliknutím pravého tlačítka myši a volbou “Add new track”.

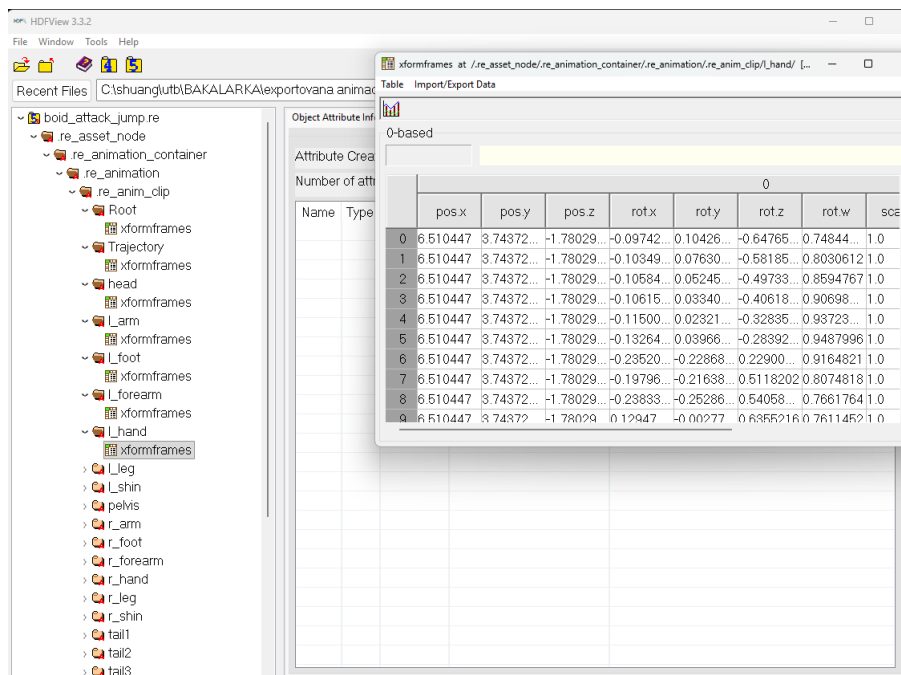


Obr. 27: Snímek zobrazující volbu exportování animace z AnimBrowseru
(zdroj: vlastní)

3.5.6.2 Exportování animace

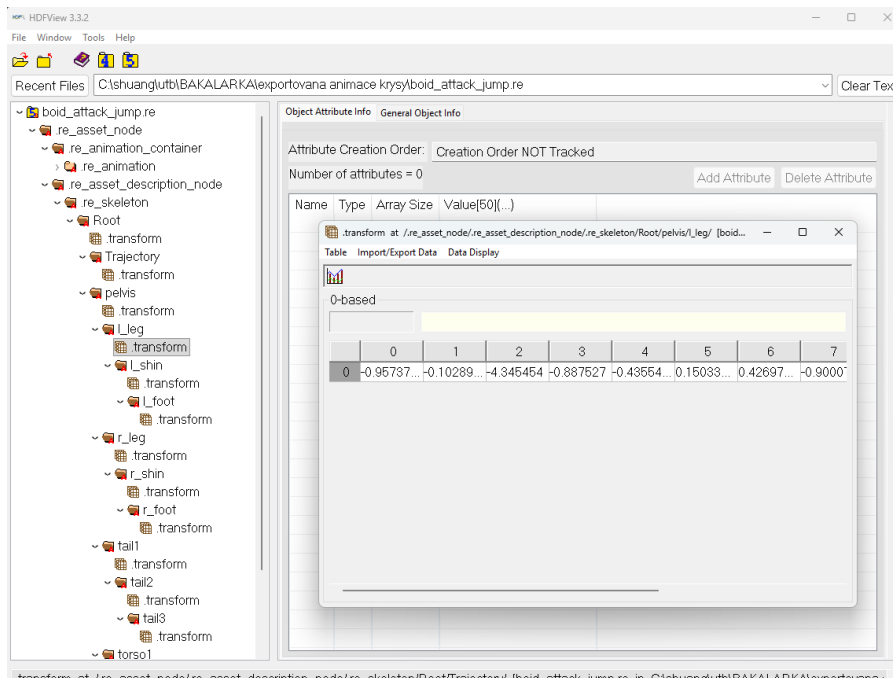
Exportování animací se provádí z pravé části okna AnimBrowseru, kde je zobrazen výpis animací (záložka *All animations*). Pomocí pravého tlačítka myši se kliknutím na požadovanou animaci musí zvolit “Export animation” viz Obr. 27 a následně výběrem místa, kam má být animace uložena. Exportovaný soubor má koncovku “re”. V případě ukázky je to tedy “boid_attack_jump.re”. Každý takový animační soubor se skládá ze dvou hlavních částí:

- re_anim_clip – obsahuje xformframes soubory zobrazující data k transformacím animačních rámců pro všechny kosti viz Obr. 28



Obr. 28: Snímek zobrazující HDFView část pro re_animation s xformframes daty pro levou ruku (zdroj: vlastní)

- re_skeleton – obsahuje hierarchii kostí kostry (anglicky *skeleton's bone hierarchy*) a transformace postroje (anglicky *rig*) pro všechny kosti

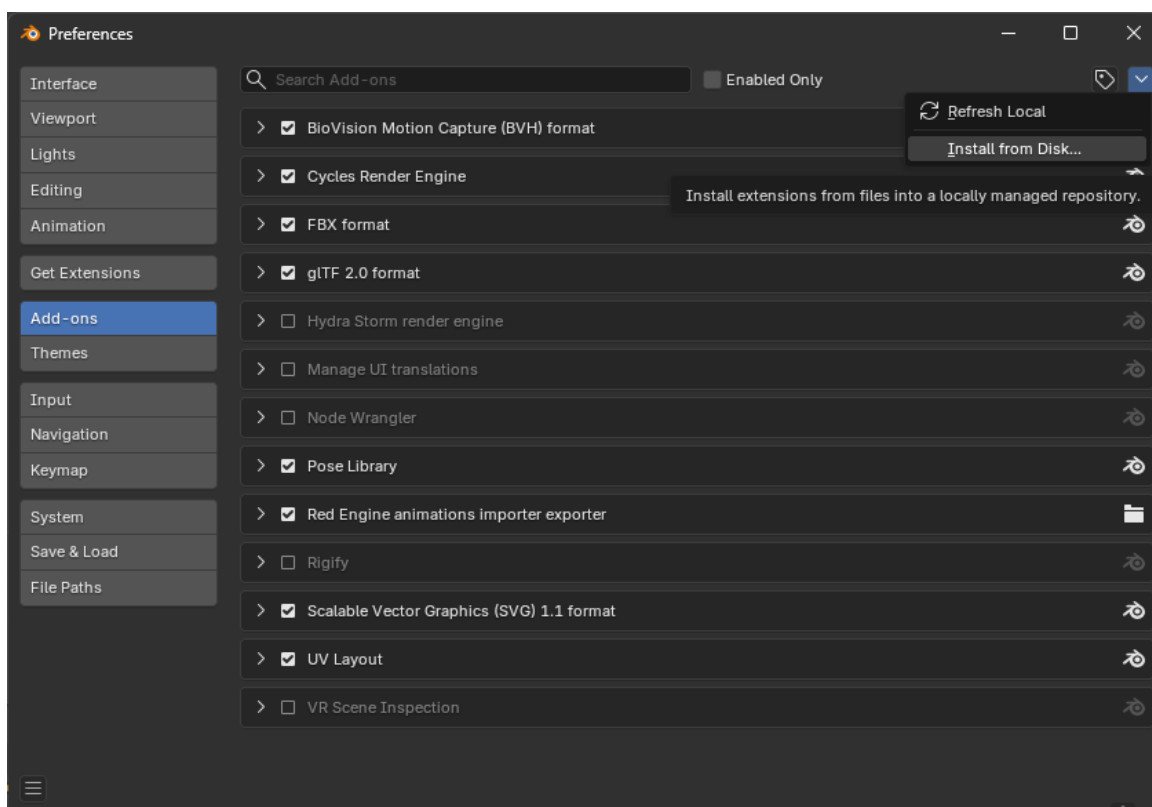


Obr. 29: Snímek zobrazující HDFView část pro re_skeleton s transformační tabulkou pro levou nohu (zdroj: vlastní)

3.5.6.3 Příprava prostředí HDFView a Blender pro import animačního souboru

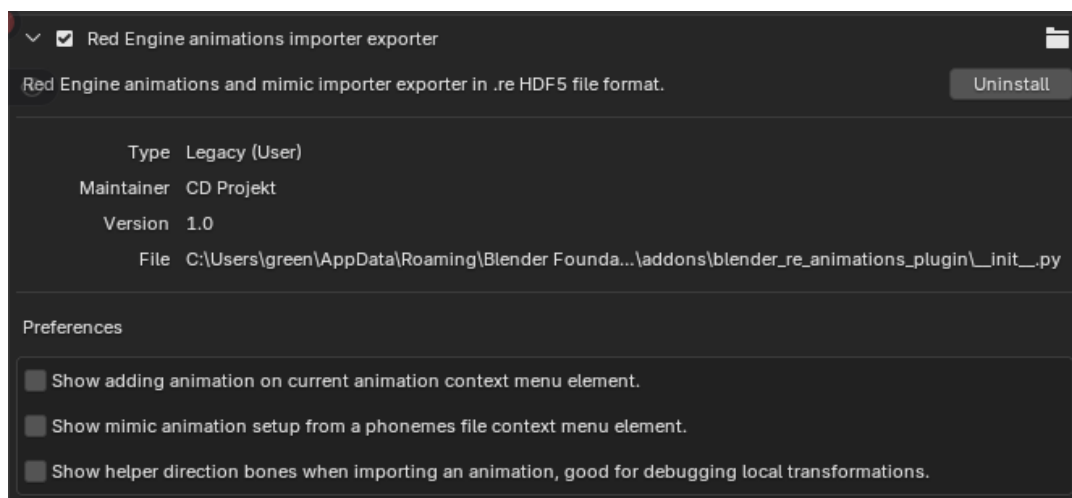
Soubor, který je exportován, lze následně otevřít v HDFView [22] bez nutnosti instalace dalších pluginů. Uživatel tak může zkoumat data pro případnou editaci skriptováním.

Pro editaci v Blenderu je podle oficiální dokumentace [24] REDkitu nutné nejdříve nainstalovat do Blenderu plugin, který by měl být přítomný v instalační složce REDkitu: “SteamLibrary\steamapps\common\The Witcher 3 REDkit\bin\tools\BlenderPlugin”. Postup instalace podle dokumentace je následující:



Obr. 30: Snímek zobrazující nabídku instalace Pluginů (Add-ons) z Disku (zdroj: vlastní)

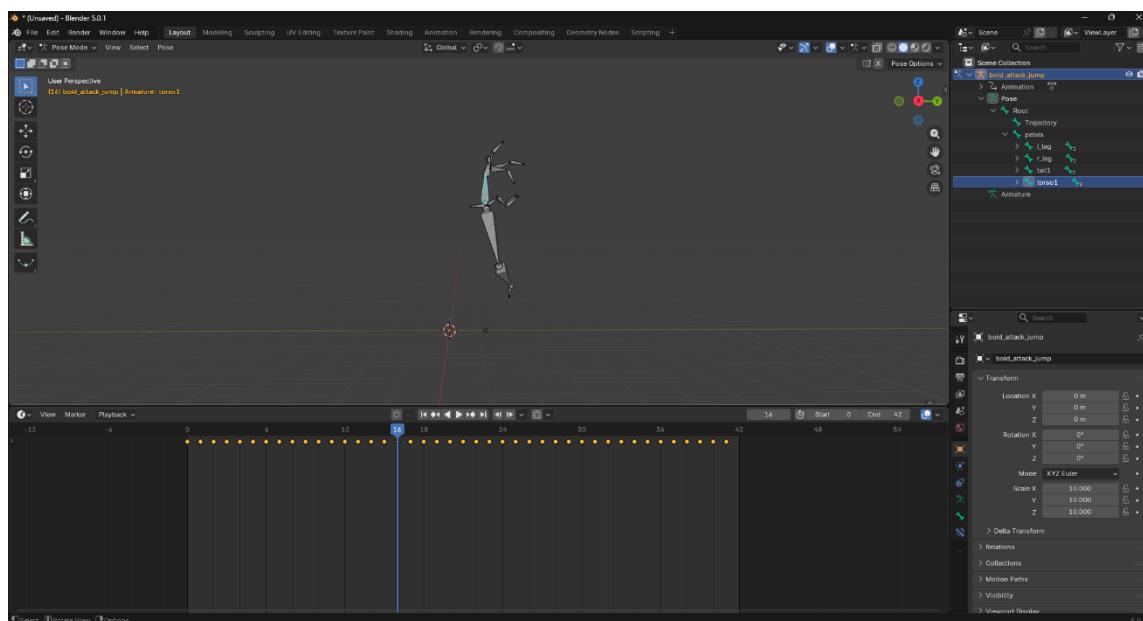
- V Blenderu kliknout na záložku Edit – Preferences...
- Zvolit položku Add-ons a v ní podle verze Blenderu kliknout na tlačítko “Install from Disk...” viz Obr. 30
- Dohledat soubor “blender_re_animations_plugin.zip” v instalační složce REDkitu uvedené výše a vybrat jej k instalaci
- Po instalaci pluginu je potřeba jej mít aktivovaný v Blenderu, případně Blender restartovat, v položce Add-ons má tento plugin název “Red Engine animations importer exporter” viz Obr. 31



Obr. 31: Snímek aktivovaného pluginu pro zpracování animací do Blenderu
(zdroj: vlastní)

3.5.6.4 Použití Blenderu pro editaci animačního souboru

V Blenderu po instalaci a aktivaci pluginu pro zpracování animací může uživatel v panelu Outliner, kde se mu zobrazuje kolekce scény, použít plugin k importování animačního souboru z REDkitu do své scény v Blenderu.



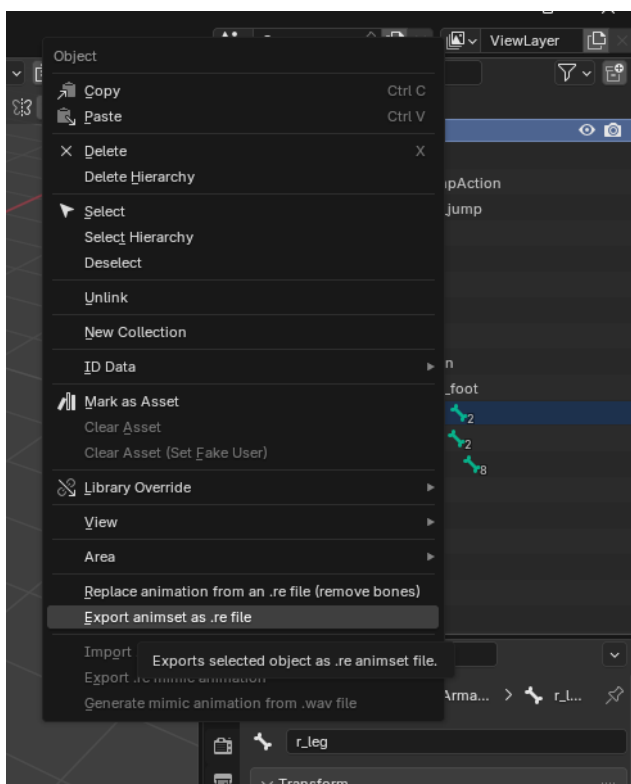
Obr. 32: Snímek zobrazující Blender software s importovanou animací krysy ve scéně
(zdroj: vlastní)

Kliknutím pravého tlačítka myši do panelu Outliner v Blenderu se uživateli zobrazí kontextové menu a výběrem možnosti “Import .re animset file” může uživatel vybrat

exportovaný animační soubor z REDkitu viz podkapitola 3.5.6.2. Po výběru souboru se uživateli do viewportu Blenderu nahraje armatura (anglicky *armature*) včetně celé animace viz Obr. 32. Uživatel nyní může editovat podle potřeby animaci.

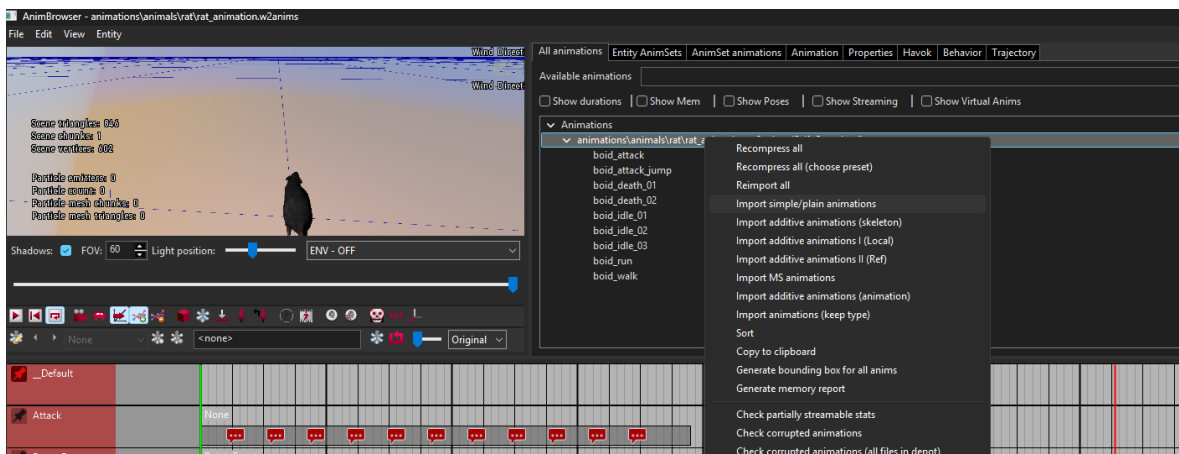
3.5.6.5 Exportování z Blenderu do REDkitu

Po editaci animace může uživatel hotovou animaci exportovat z Blenderu kliknutím pravého tlačítka myši do panelu Outliner na požadovanou animaci – zobrazí se kontextové menu a výběrem možnosti “Export animset as .re file” se soubor exportuje.



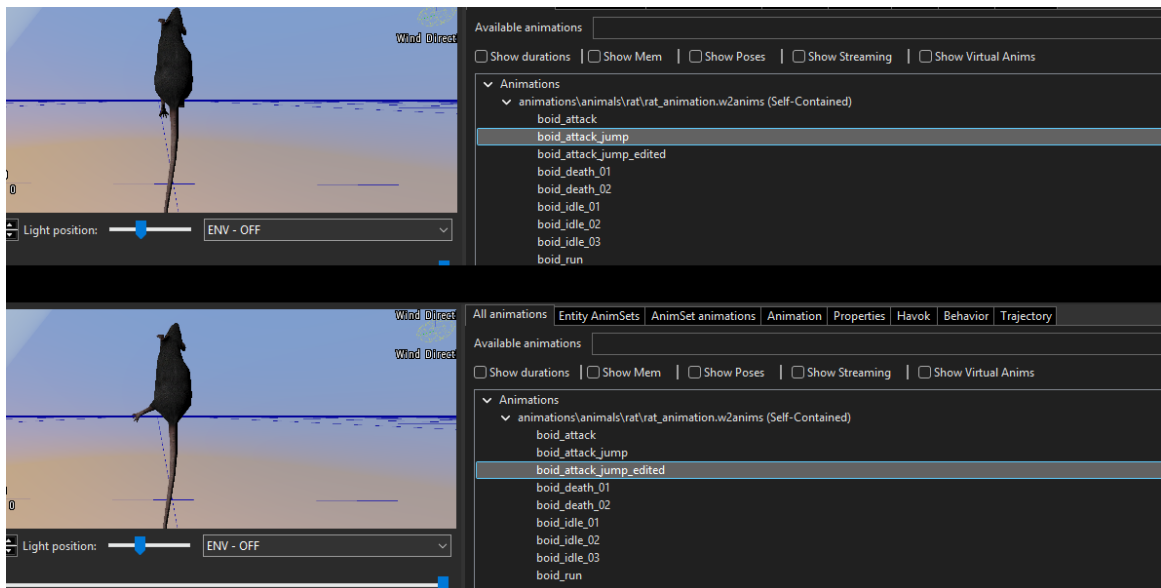
Obr. 33: Snímek zobrazující možnost exportování animace z Blenderu
(zdroj: vlastní)

Editovanou animaci může poté uživatel importovat do REDkitu v AnimBrowseru kliknutím pravého tlačítka myši do panelu zobrazující seznam animací. Zobrazí se kontextové menu, kde výběrem možnosti “Import simple/plain animations” může uživatel vybrat soubor, který byl exportován z Blenderu jako animace.



Obr. 34: Snímek zobrazující možnosti importování animací do REDkitu pomocí AnimBrowseru (zdroj: vlastní)

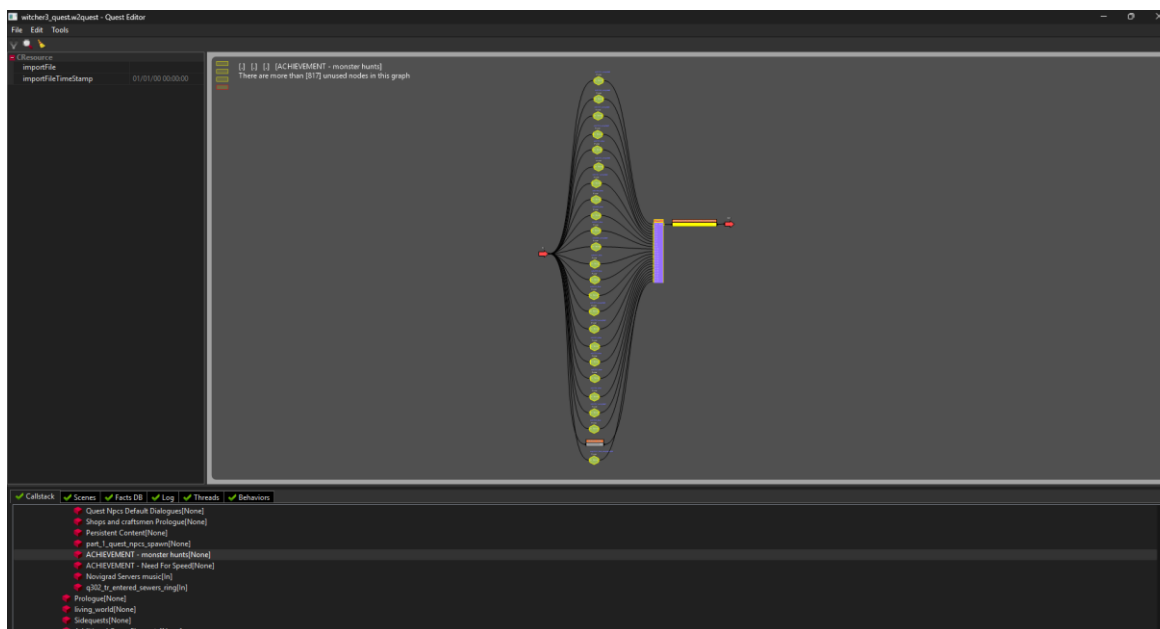
Po importování animace do REDkitu ji lze použít jako ostatní animace viz ukázka rozdílu originální animace “boid_attack_jump” (horní část ukázky Obr. 35) a editované animace “boid_attack_jump_edited” (spodní část ukázky Obr. 35).



Obr. 35: Snímek zobrazující rozdíl originální a editované animace (zdroj: vlastní)

3.5.7 Quest Debugger a Quest Editor

Debugger úkolů [25] zobrazuje stromový pohled (anglicky *tree view*) na úkolové linie (anglicky *questlines*) spojené s aktuálně načteným stavem hry do editoru REDkit. Tento nástroj zobrazí okno „Quest Editor“, ale včetně debugovacích možností. Lze jej otevřít z horního panelu nástrojů: Tools – Debuggers – Quest Debugger...



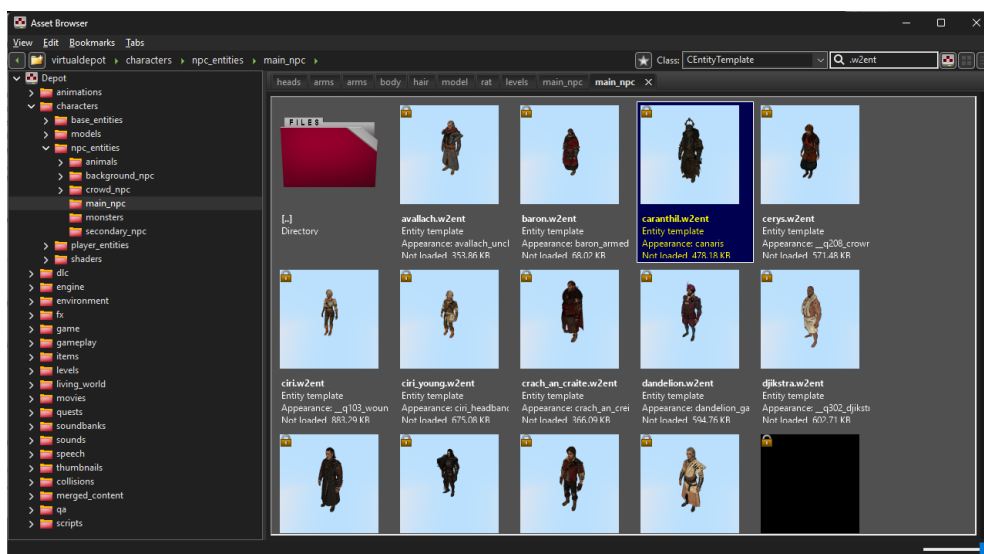
Obr. 36: Snímek zobrazuje okno Quest Debugger s načteným úkolem ACHIEVEMENT – monster hunts (zdroj: vlastní)

Uživatel si může zobrazit graf uzlů (anglicky *node graph*) dvojklikem levého tlačítka myši na libovolný úkol (koncová položka stromu úkolů s ikonou červené krychle). Tímto se ve viewportu Quest Editoru načte a zobrazí graf uzlů k danému úkolu viz Obr. 36. Ve viewportu může uživatel získávat informace o libovolné bloku v grafu, případně jej editovat.

Uvnitř Quest Debugger okna se nachází mimo jiné také záložka „Facts DB“, která zobrazuje tzv. databázi faktů, která udržuje bázi proměnných s přiřazenými hodnotami, na kterých závisí spousta úkolů a dalších systémů hry. Lze je modifikovat skripty nebo skrze úkoly (blok „FactsDB Change“ v grafu uzlů pro úkoly). Hodnoty těchto faktů lze poté kontrolovat v průběhu úkolů, scén a skriptů.

3.5.8 Editor Entit

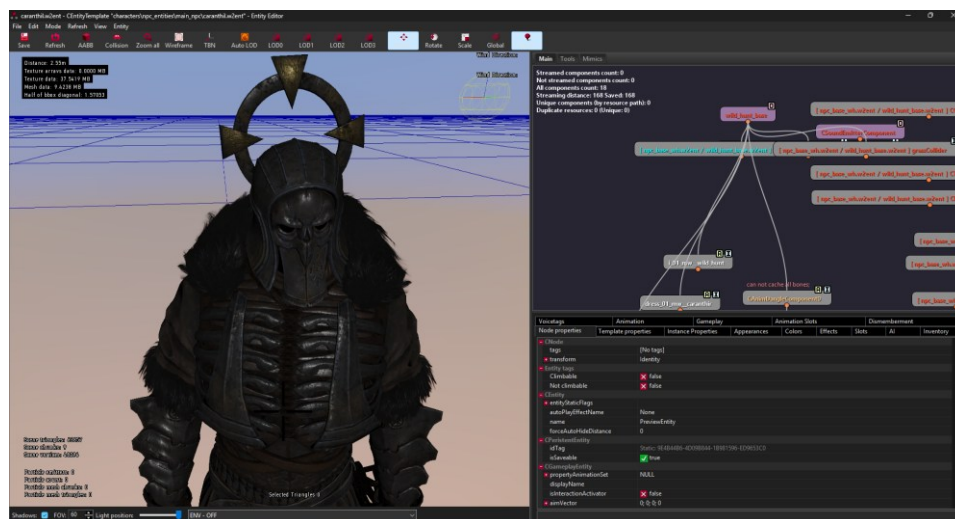
Jedná se o editor umožňující provádět změny vlastností entit, respektive šablon entit (anglicky *entity template*). Ze šablon entit se následně mohou vytvořit instance dané entity ve vytvářeném světě. K nalezení šablon entit lze použít Asset Browser. Šablony entit jsou uloženy v souboru třídy CEntityTemplate s koncovkou „w2ent“.



Obr. 37: Snímek zobrazující Asset Browser se zobrazenými šablonami entit hlavních postav (zdroj: vlastní)

Editor Entit lze spustit dvojklikem levého tlačítka myši na vybranou šablonu entity, v ukázce viz Obr. 37 lze například takto rozkliknout šablonu entity Caranthal (soubor caranthal.w2ent) a spustí se “Entity Editor” viz Obr. 38. Editor je defaultně rozdělen do tří částí:

- Levá část – viewport zobrazující náhled na aktuální entitu
- Pravá horní část – graph editor zobrazující všechny komponenty entity
- Pravá spodní část – záložky zobrazující data k entitě, popř. označené komponentě

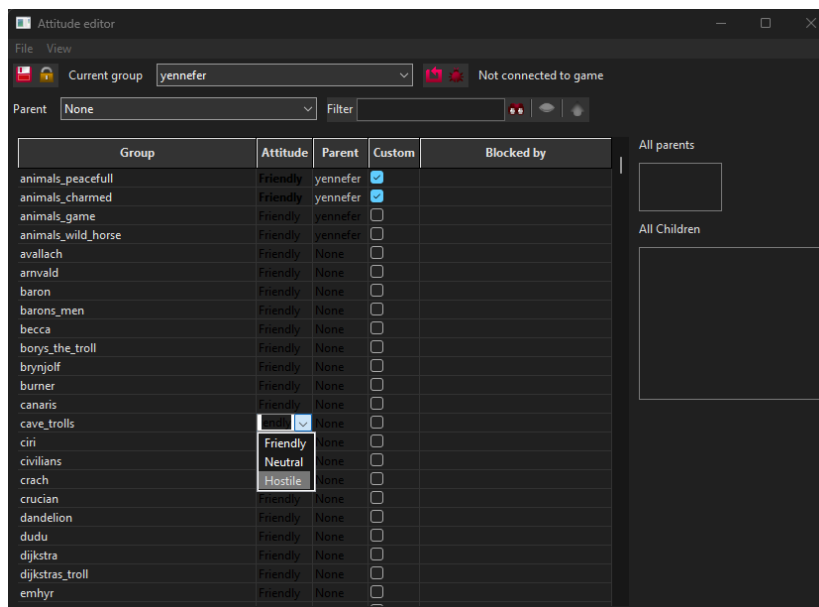


Obr. 38: Snímek zobrazující Entity Editor s načtenou šablonou entity caranthil.w2ent (zdroj: vlastní)

Pro samotnou editaci slouží hlavně pravá spodní část, kde jednotlivé záložky nabízí různé možnosti modifikací. V následujících podkapitolách jsou popsány ty záložky, které oficiální dokumentace [26] popisuje a jsou také často používané při modifikacích. Editor také nabízí v horní části okna panel nástrojů, kde uživatel může například zobrazit kolize pro entitu, přepínat Wireframe režim zobrazení, přepínat jednotlivé úrovně detailů (anglicky *Level of Detail – LOD*).

- Node properties – kliknutím na pozadí graph editoru se zobrazí generické vlastnosti entity. Označením jednotlivých komponent, nebo i více komponent zároveň (položky v graph editoru) se zde zobrazí jejich odpovídající vlastnosti.
- Template properties – zobrazuje vlastnosti aktuálně načtené šablony entity. Umožňuje zvolit třídu entity (anglicky *Entity class*), nastavit základní šablonu entity, ze které tato dědí, případně přidat další.
- Appearances – dává uživateli informace o jednotlivých částech oděvu, těla, účesů atd. Jedná se opět o soubory šablon entit (anglicky *Entity Templates*) s koncovkou “w2ent”. Lze přidávat další šablony, nebo dokonce vytvářet více sad “vzhledů” pro danou entitu.
- Effects – umožňuje editovat jaké efekty nebo animace mají být použité pro danou entitu, případně efekty a animace otevřít do jejich editorů a tam je editovat přímo.

3.5.9 Attitude Editor



Obr. 39: Snímek zobrazující provedení změny v Attitude editoru (zdroj: vlastní)

Attitude Editor může uživatel použít pro modifikování vztahů skupin (anglicky *Groups*) mezi sebou. Jako skupina může být považovaná i samostatná postava (například postava

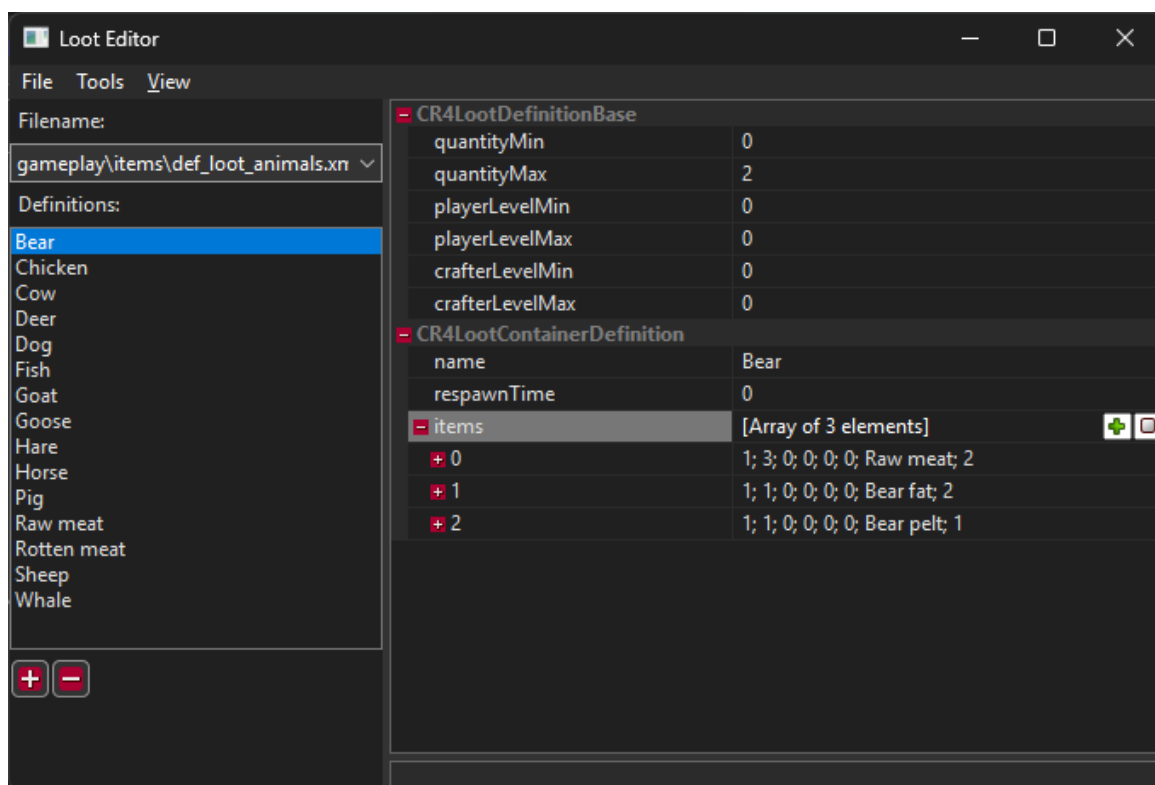
Vesemir), nebo také celý typ kreatury (například všichni divocí koně). Tento editor umožňuje také testovat, jak se různé skupiny entit chovají, když se jejich vztahy změjí.

Oficiální dokumentace REDkitu [27] dáva za příklad nastavení vztahu postavy Yennefer ke skupině jeskynních trolů (anglicky *Cave Trolls*) jako nenávistný. V důsledku této změny by postava Yennefer měla být automaticky nepřátelská, pokud spatří jakoukoliv instanci entity jeskynního trola. Po nastavení změny z Friendly na Hostile u položky cave_trolls pro skupinu yennefer viz Obr. 39 by se měl automaticky označit stav ve sloupci Custom jako zaškrknutý.

Attitude editor navíc zobrazuje údaje o rodiči zvolené skupiny. Pokud je vytvořena nová entita a je nastaven rodič na již existující entitu, která má nastaveny vztahy k ostatním skupinám, automaticky dědí i její vztahy. Pro většinu případů stačí změnit vztah jednostranně a efekt nastane pro obě skupiny. V příkladu Yennefer a trolů není tedy potřeba dohledávat skupinu “Cave Trolls” a měnit jim Attitude ke skupině “yennefer” na nepřátelský.

3.5.10 Loot Editor

Tento editor umožňuje uživateli definovat nové úkolové předměty, nebo kořist (anglicky *loot*). Loot Editor lze spustit v hlavním okně REDkitu, na horním panelu nástrojů pomocí “Tools – Loot editor...”.



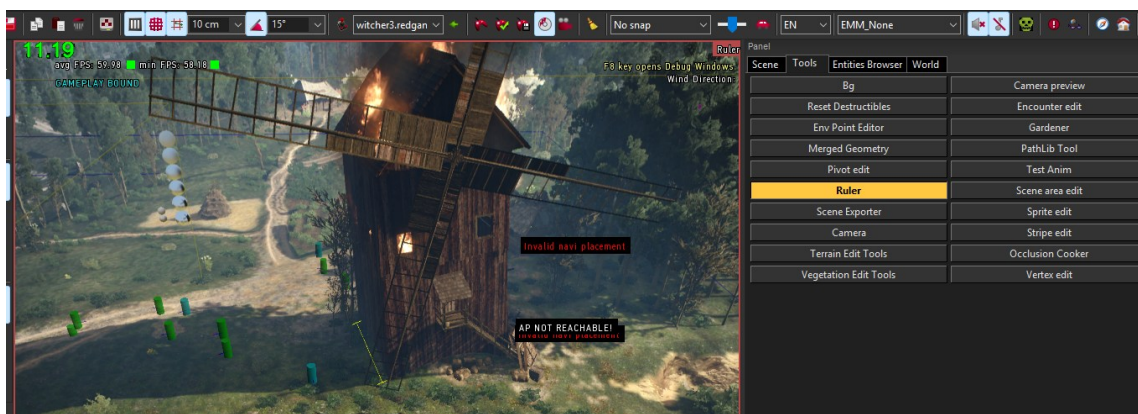
Obr. 40: Snímek zobrazující okno Loot Editor

(zdroj: vlastní)

Loot editor zobrazuje seznam definicí a jejich konfigurace. Definicí se v tomto kontextu rozumí objekt (například medvěd, nebo mrtvola vesničana atd...). V každé definici lze mimo jiné editovat: rozsah množství předmětu, rozsah požadované úrovně hráče, jednotlivé předměty uvnitř kontejneru v definici (například pro definici medvěda to mohou být: syrové maso, medvědí tuk, medvědí kůže) a jejich množství.

3.5.11 Ruler

Ruler (česky *pravítka*) je nástroj určený k měření vzdálenosti mezi dvěma body ve viewportu hlavního okna REDkitu. V načteném světě lze v pravém panelu nástrojů v záložce “Tools” kliknutím na možnost “Ruler” aktivovat tento nástroj. Následně může uživatel kliknutím levého tlačítka myši (v závislosti na verzi REDkitu a nastavení je možné, že bude potřeba při měření stisknout i klávesu SHIFT) začít měřit a pohybem myši do požadovaného bodu kliknutím znovu ukončit měření. V levé části viewportu se během měření objeví zelené číslo s naměřenou hodnotou viz Obr. 41, kde je v ukázce použit nástroj Ruler pro změření délky jedné zdi základny budovy mlýna (z pohledu kamery v ukázce se jedná o levou zeď), v levém horním rohu viewportu je naměřená hodnota (číslo zelené barvy).



Obr. 41: Snímek zobrazující použití nástroje Ruler
(zdroj: vlastní)

Pro ukončení používání nástroje Ruler (platí i pro jiné nástroje, které se aktivují obdobným způsobem) lze kliknout pravým tlačítkem do viewportu a zvolit možnost “Close active tool”, nebo kliknout na tlačítko Ruler, které nástroj aktivovalo.

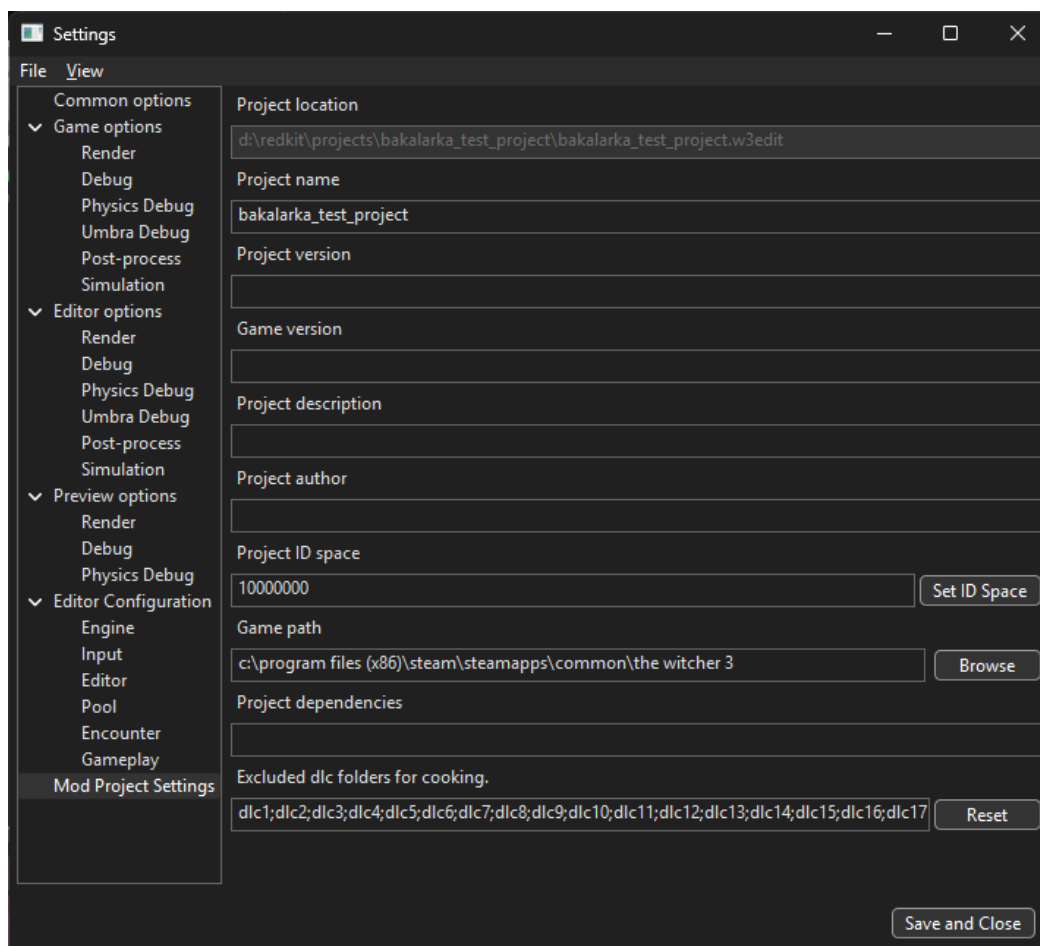
3.5.12 Editor Lokalizovaných Řetězců

Editor [28] spustitelný z nabídky horního panelu nástrojů volbou možností “Tools – Localized Strings Editor...”. Lokalizace (anglicky *Localization*) je v kontextu REDkitu překládání textu a hlasových záznamů do jiných jazyků a kultur. String je považován za text zobrazený ve hře jako řádek dialogu, nápověda nebo jméno. String může mít více jazykových variant (angličtina, čeština, němčina atd...). Hra vždy zvolí jazykovou variantu řetězce, podle

zvoleného jazyka v nastavení hry. String může mít krátký identifikátor nazvaný klíč (anglicky *key*).

Řetězce (dále stringy) s klíčem mívají spíše generické použití, například jména, klíčová slova, zprávy ve vyskakovacím okně (anglicky *pop-up messages*). Stringy bez klíčů bývají spojené se systémem hry, například dialogové řádky ve scénách. Stringy lze používat ve skriptech pomocí funkcí “getLockStringByID” a “getLockStringByKey”.

REDkit editor má funkci “ID Spaces”, která by měla pomoci vyhnout se kolizím identifikátorů stringů (anglicky *String ID collisions*) mezi mody nebo modery pracujícími na stejném modu. ID prostoru (anglicky *ID space*) je číslo, od kterého se následující stringy budou inkrementálně navyšovat. Nastavení ID prostoru lze provést otevřením menu nastavení z hlavního okna REDkitu na horním panelu nástrojů možnosti “View – Settings”, položka “Mod Project Settings”. V okně nastavení zobrazuje data k projektu a v položce “Project ID space” viz Obr. 42 lze nastavit tento ID prostor, který by měl být unikátní pro každý mod, jelikož v případě, že se bude číslo tohoto prostoru shodovat s jiným modem, mohou si pak navzájem přepsat stringy, což je nežádoucí.

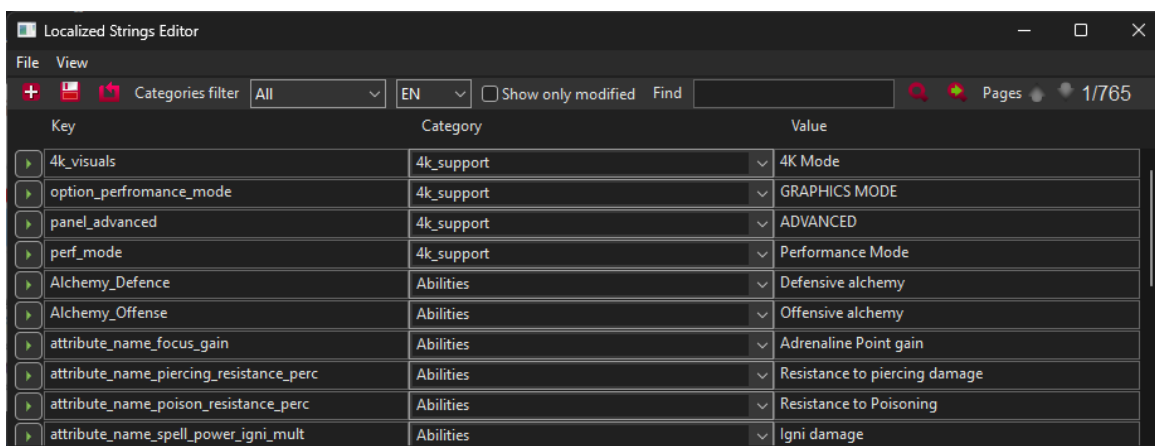


Obr. 42: Snímek zobrazující možnosti nastavení údajů o projektu
(zdroj: vlastní)

Nástroj Localized Strings Editor zobrazí uživateli data rozdělena do tří sloupců:

- Key – unikátní identifikátor, kterým se lze odkázat na danou hodnotu stringu později
- Category – pomáhá rozdělit stringy do skupin
- Value – skutečná hodnota textu, která má být použita ve hře

Uživatel může také exportovat .csv soubor s provedenými modifikacemi na své lokální databázi stringů. Tento soubor si pak může někdo další importovat do své databáze stringů. Díky tomu je umožněna kolaborace na modu. Obě varianty lze provést z nabídky “File – Import” nebo “File – Export”. Další funkcí tohoto editoru je vytváření nových stringů v databázi.



Obr. 43: Snímek zobrazující Localized Strings Editor

(zdroj: vlastní)

3.5.13 Scene Editor

Editor scén (anglicky *scene editor*) představuje nástroj REDkitu určený pro tvorbu a úpravu dialogových a příběhových scén ve hře. Umožňuje připravovat sekvence rozhovorů mezi postavami, příběhovými scénami, nebo také scény bez narušení hry tzv. *gameplay scene*, kde scéna je využita například pro reakci hráčovy postavy na spouštěcí událost. Scény se také ve velké míře používají pro získání vstupu od uživatele ohledně rozhodnutí například jako dialogová volba. Scény se typicky spouští po nějaké události z Quest Editoru a po získání výstupu, nebo ukončení scény děj úkolu pokračuje dále.

Scéna je uložena ve vlastním souboru. Po otevření souboru se spouští editor scén. Editor nabízí nastavení herců ve scéně, případně rekvizit použitých pro scénu, jejich polohy ve scéně a počáteční nastavení. Rozmístění herců je provedeno pomocí záložky *Dialogset Settings*, kde uživatel nastaví rozmístění herců do případné scény, může také nastavit značku (anglicky *tag*), která odpovídá vytvořené značce přímo ve tvořeném světě v REDkitu. Pro scénář, který se má ve scéně odehrát, obsahuje editor scén vlastní záložku (anglicky *screenplay*), kde uživatel může specifikovat scénář viz Obr. 92 pro daný soubor scény. Děj ve scénáři může být rozdělen do jednotlivých bloků. Pro manipulaci s bloky slouží záložka

grafu scény, kde má každý blok scénáře svůj uzel, příklad grafu scény viz Obr. 107. V rámci uzlů lze využít i jiné uzly, než je blok scénáře, například skriptovací funkce, uzly pro větvení podle sledovaných faktů v databázi faktů, které mohou spustit jiný blok scény a dále určovat pořadí bloků a konečný výstup ze souboru scény.

Další důležitá záložka v editoru scén je časová osa (anglicky *timeline*), kde uživatel přímo přidává jednotlivé kamery do časové osy, udává každé kameře nastavení, animace apod., také každému z použitých herců scény přidává animace, pozici apod. Podle aktivního bloku scény (scénáře) se automaticky odhadne čas běhu každé věty. Uživatel má možnost pomocí panelu vlastností (anglicky *properties*) v pravé části okna editoru nastavit pro každou konkrétní větu zvukový soubor, kde je záznam přečtené věty, který se má při scéně přehrát. Ve stejné části vlastností může uživatel rovněž nastavit animační soubor obličeje, který odpovídá aktuálně mluvené větě (anglický pojem *lipsync*).

3.6 Witcher Script

WitcherScript [29] je primární skriptovací jazyk hry Zaklínač 3: Divoký hon. Umožňuje uživateli mimo jiné “oživit” jejich mod, debugovat ho, nebo vytvářet debugovací zprávy přímo ve hře. Oficiální dokumentace uvádí, že herní skripty jsou umístěny ve složce: “content\content0\scripts”. Každý WitcherScript soubor má koncovku “ws” za názvem souboru.

3.6.1 Příprava prostředí pro WitcherScript

Uživatel si může povolit ve hře vývojářskou debug konzoli (anglicky *developer debug console*), kterou může volat určité funkce definované ve WitcherScript (.ws) souborech. Podle oficiální dokumentace [30] je postup k povolení konzole následující:

- otevřít soubor “general.ini” umístěný ve složce “The Witcher 3/bin/config/base”
- do souboru přidat řádek “DBGConsoleOn=true” a uložit

Po povolení konzole může uživatel kdykoli během hraní pomocí klávesy pro tildu (~) spustit konzoli, kde může ve hře zadávat příkazy, volat funkce WitcherScript atd.

Doporučená [31] struktura složek uvnitř modu je následující:

- mod_složka
 - content
 - scripts
 - local

Nový WitcherScript soubor lze vytvořit jako textový soubor a přejmenovat jej na “.ws” soubor. K editaci lze použít textový editor, externí vývojové prostředí jako je například Microsoft Visual Studio Code [32], anebo integrované Script Studio přímo uvnitř REDkitu.

3.6.2 Základy WitcherScriptu

Tato podkapitola shrnuje základní používání skriptovacího jazyka WitcherScript.

3.6.2.1 Komentáře

Do souboru s WitcherScriptem lze zapisovat komentáře, které slouží obdobně jako v jiných jazycích k okomentování daného bloku kódu, případně vysvětlení, co daný kód má dělat. Následující ukázka kódu ukazuje jednořádkový a poté víceřádkový komentář.

```
// Toto je jednořádkový komentář

/*
Toto je
víceřádkový komentář
*/
```

3.6.2.2 Datové typy

WitcherScript podle oficiální dokumentace [33] nabízí sadu základních datových typů, mezi nejčastější patří:

- `int` – standardní 32bitové celé číslo, rozsah: -2.147.483.648 až 2.147.483.647.
- `String` – standardní string, řetězec znaků ohraničený uvozovkami: “Toto je string”.
- `name` – jméno, název, nebo tag, řetězec znaků ohraničený apostrofy: ‘Toto je name’.
- `float` – standardní float, reálné číslo s plovoucí řádovou čárkou, například: 1.0f.
- `Vector` – čtyř složkový vektor (kvaternion), uspořádané do XYZW, například (100, 100, 100, 1).
- `EulerAngles` – rotace tří proměnných seřazené jako Pitch (klopení), Yaw (stáčení), Roll (náklon/rotace), například: (0, 180, 0).
- `bool` – standardní boolean, tedy proměnná nabývající hodnoty `true`, nebo `false`. Přijímá také číselné hodnoty 0 (`false`) a 1 (`true`).
- `Matrix` – indikuje typ maticové proměnné.
- `array<X>` - indikuje pole o datovém typu X, například `array<int>` bude pole celých čísel.

3.6.2.3 Globální objekty

Uživateli je zpřístupněna řada globálních objektů, které jsou vždy přístupné a odhalují spoustu užitečných funkcí. Tyto objekty lze modifikovat v souboru `redscripts.ini`, ale není to doporučeno, jelikož takový mod nebude kompatibilní s ostatními mody, které mají defaultní nastavení. Výchť těchto objektů podle dokumentace [33]:

```
theGame=CR4Game
theServer=CServerInterface
thePlayer=CR4Player
theCamera=CCamera
theUI=CGuiWitcher
theSound=CScriptSoundSystem
theDebug=CDebugAttributesManager
theTimer=CTimerScriptKeyword
theInput=CinputManager
```

Lze tedy použít například objekt “thePlayer” k zobrazení zprávy hráči:

```
thePlayer.DisplayHudMessage('Hello')
```

Odpadá tak nutnost získávat instanci hráče pomocí funkce “GetWitcherPlayer()”, jelikož hra díky použití globálního objektu bude vědět, že se jedná o hráče.

3.6.2.4 Proměnné

Proměnné ve WitcherScriptu slouží k ukládání dat, se kterými skript pracuje (například mezivýsledky výpočtů, odkazy na objekty apod.). Deklarují se pomocí klíčového slova “var”, po něm následuje název proměnné a její datový typ.

```
// toto je deklarace promenne typu int:  
var promenna : int;
```

Pro použití proměnných pouze uvnitř funkce (lokální proměnné) je lze deklarovat uvnitř samotné funkce. Obecně platí, že proměnné musí být deklarovány předtím, než jsou použity.

```
exec function udelejNeco(nazevNeceho : name)  
{  
  var i : int;  
  var skills : String;  
  
  // Zde již mohou být použity proměnné i, skills  
}
```

Další variantou je použít proměnné jako členské proměnné tříd (anglicky *class member variables*). Jsou spojeny se třídou, ve které jsou definovány. Lze je použít k uložení dat.

```
class Test  
{  
  var SomeText : String;  
}
```

3.6.2.5 Parametry funkcí

Umožňují předat do funkce vstupní hodnoty, tím ji tzv. parametrizovat. K definování parametru ve funkci se používá syntaxe: *nazev* : *typ*, kde *nazev* je název proměnné, pomocí kterého lze získat hodnotu proměnné a *typ*, který určuje datový typ proměnné.

```
Exec function changeWeather(weatherName : name)  
{  
  // použití parametru weatherName k volání jiné funkce  
  RequestWeatherChangeTo(weatherName, 1, false);  
}
```

3.6.2.6 Funkce

WitcherScript nabízí několik druhů funkcí, které může uživatel nastavit. Liší se způsobem volání a místem, odkud je může uživatel zavolat (konzole, scény, quest graf atd.), zda se vykonají okamžitě, nebo ve specifickém čase.

1. `exec function` – spustitelné funkce z herní debug console, lze je použít ke spuštění dalších funkcí. Následující příklad ukazuje volatelnou funkci z debug konzole, která zastaví déšť ve hře:

```
exec function stopRain()
{
    RequestWeatherChangeTo('WT_Clear', 1.0, false);
}
```

2. `Latent function` – jsou podobné koprogramům (anglicky *coroutines*), umožňují běh v čase, případně zpomalit exekuci. V ukázce níže je příkladná latentní funkce, která využívá `Sleep()` funkci pro uspaní funkce na zadané množství vteřin:

```
latent storyscene function ShaveGeralt( player : CStoryScenePlayer )
{
    var acs : array< CComponent >;

    acs = thePlayer.GetComponentsByClassName( 'CHeadManagerComponent' );
    ( ( CHeadManagerComponent ) acs[0] ).Shave();
    Sleep(1.0f);
}
```

3. `Timer function` – funkce spouštěné časovačem (anglicky *timer*) navázaným na entitu. Typicky přijímá parametr *dt* (delta time, tj. uplynulý čas). Některé tyto funkce jsou ve skriptech hry, konkrétně v souboru *entity.ws*:

```
// přidává pojmenovaný časovač entitě založený na reálném čase
// vrací unikátní ID časovače
import final function AddTimer(
timerName : name,
period : float,
optional repeats : bool,
optional scatter : bool,
optional group : ETickGroup,
optional saveable : bool,
optional overrideExisting : bool
) : int;

// přidává pojmenovaný časovač entitě založený na herním čase
// vrací unikátní ID časovače
import final function AddGameTimeTimer(
timerName : name,
period : GameTime,
optional repeats : bool,
```

```

optional scatter : bool,
optional group : ETickGroup,
optional saveable : bool,
optional overrideExisting : bool
) : int;

// Odstraní všechny časovače se stejným zadaným jménem entity
import final function RemoveTimer(
timerName : name,
optional group : ETickGroup
);

// Odstraní všechny časovače se stejným zadaným ID
import final function RemoveTimerById(
id : int,
optional group : ETickGroup
);

// Odstraní všechny časovače z entity
import final function RemoveTimers();

```

4. **Storyscene function** – speciální funkce, které jsou použitelné ve scénách, například pro ovládání herních systémů během scén, nebo skriptovaných sekvencí. Příklad vytvoření takové funkce:

```

storyscene function NecoSpust()
{
    // provedení něčeho pro scénu
}

```

5. **Quest function** – speciální funkce, použitelné v uzlech úkolů (quest graph), například spouštění událostí, menu, změny stavu úkolů a podobné. Příkladná funkce:

```

quest function LaunchCredits()
{
    theGame.GetGuiManager().RequestCreditsMenu(CreditsIndex_Witcher3)
}

```

6. **Reward function** – tyto funkce mohou být připojeny k odměně uvnitř editoru. Například funkce, která odmění postavu zaklánače zkušenostními body:

```

reward function TutorialLevelUp()
{
    var witcher : W3PlayerWitcher;

    witcher = GetWitcherPlayer();
    witcher.AddPoints(EExperiencePoint, 50, false);
    if(witcher.GetLevel() == FactsQuerySum('tutorial_starting_level'))
    {

```

```

    witcher.AddPoints(EEExperiencePoint,
    witcher.GetMissingExpForNextLevel(), true);
}
}

```

7. Cleanup function – z funkce se nevrací nic a nemají žádné parametry. Slouží pro “čištění” po ukončení nějakého děje. Například funkce pro zničení instance vyhozené entity:

```

cleanup function ThrowProjectileCleanup()
{
    if( thrownEntity )
    {
        thrownEntity.StopAiming( false );

        if ( !parent.wasBombReleased )
        {
            rider.RaiseEvent( 'actionShootEnd' );
            parent.wasBombReleased = true;
            thrownEntity.Destroy();
            thrownEntity = NULL;
        }
    }
}

```

8. Entry function – podle dokumentace [33] se jedná o výjimečně používané stavové funkce. Například:

```

entry function DrawEvent()
{
    parent.ownerPlayer.SetBehaviorVariable( 'failSafeDraw', 1.0 );
    virtual_parent.RaiseOwnerGraphEvents( 'Crossbow_Draw', true );
    parent.performedDraw = true;
}

```

3.6.2.7 Třídy

Třída v rámci WitcherScript jazyka vychází z definice třídy z objektově orientovaného programování (anglicky *Object Oriented Programming – OOP*). Používá se jako šablona pro vytváření objektů. Třída sdružuje data pro daný objekt a metody pro manipulování s nimi pohromadě. Umožňuje organizovat kód do znovupoužitelných a modulárních komponent. Třída může dědit i z jiné třídy.

1. Regular class – běžná třída deklaruje členské proměnné, metody/funkce. Příklad definice takové třídy:

```

class InterpCurve
{
    var something: float;
    function doSomething( inVal : float, outVal : float ) : int
    {

```

```

    return 10;
}
}

```

2. Native class – jsou to třídy definované v programovacím jazyce C++, tzv. nativní třídy, je zapotřebí je importovat do WitcherScript souboru. Ukázka importování nativní třídy nacházející se ve skriptu 2DArray.ws:

```

import class C2dArray extends CResource
{
    import final function GetValueAt( column : int, row : int ) :
    string;

    import final function GetValue( header : string, row : int ) :
    string;

    import final function GetValueAtAsName( column : int, row : int ) :
    name;

    import final function GetValueAsName( header : string, row : int ) :
    name;

    import final function GetNumRows() : int;

    import final function GetRowIndexAt( column : int, value : string )
    : int;

    import final function GetRowIndex( header : string, value : string )
    : int;
}

```

3. Statemachine – tyto třídy definují defaultní stav a uživatel může definovat další stavy, do kterých se třída může přesunout. Níže je uveden příklad z oficiální dokumentace [33] pro statemachine řešící “Witches cage”, která může být ve stavu zapnuto nebo vypnuto:

```

// definování statemachine
statemachine class W3WitchesCage extends CEntity
{
    default autoState = 'TurnedOff';
}

// definování vypnutého stavu (TurnedOff)
state TurnedOff in W3WitchesCage
{
    event OnEnterState( prevStateName : name )
    {
        super.OnEnterState( prevStateName );
        parent.ApplyAppearance("roots_off");
    }
}

```

```

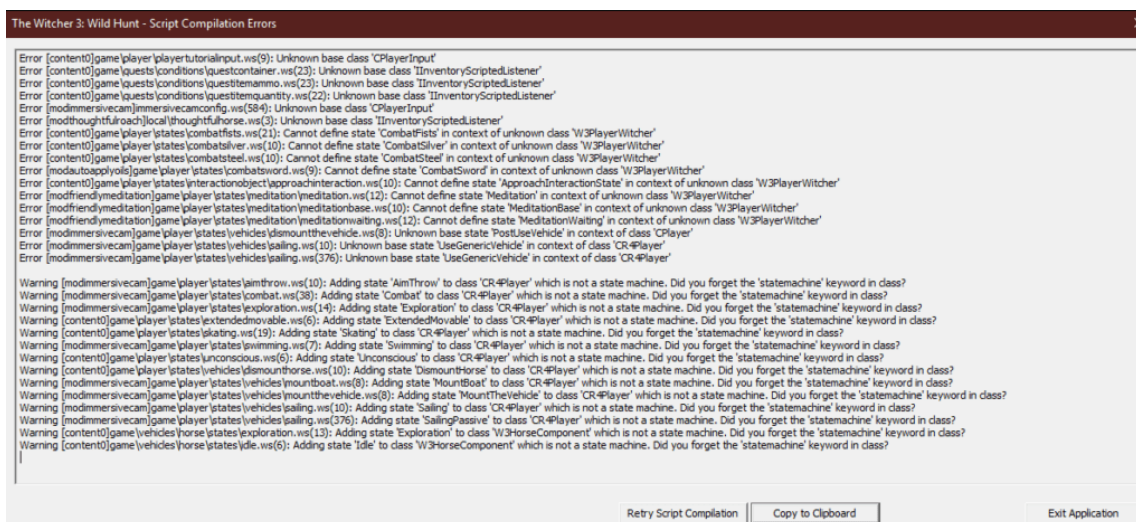
}

// definování zapnutého stavu
state TurnedOn in W3WitchesCage
{
    event OnEnterState( prevStateName : name )
    {
        super.OnEnterState( prevStateName );
        parent.ApplyAppearance("roots_on");
    }
}

```

3.6.3 Kompilační chyby

Může nastat situace, pokud je na jednom zařízení nainstalováno více modů, které modifikují stejné soubory. Ke kompilaci skriptů dochází při zapnutí hry Zaklínač 3: Divoký hon a v tom okamžiku může být uživatel / hráč notifikován ohledně konfliktu v modifikovaných souborech skriptů viz Obr. 44.



Obr. 44: Snímek zobrazující možnou kompilační chybu skriptu při spuštění hry [34]

3.6.4 Script Merger

Pokud dva, nebo více modů editují stejný soubor, je podle oficiální dokumentace [34] nutné sloučit skript soubory pomocí nástroje Script Merger [35].

Pro vyhnutí se problému s editací stejných souborů více modifikacemi a jejich nutností sloučení pomocí nástroje Script Merger, oficiální dokumentace [34] doporučuje vytvářet mody s použitím tzv. “script overrides”, které představil herní patch [36].

3.6.5 Anotace ve skriptu

WitcherScript soubory podporují několik anotací skriptu. Jejich hlavním účelem je přidat / modifikovat funkcionalitu v existujících herních funkcích a třídách.

- `@wrapMethod(class)` – slouží k “obalení” existující metody. Moder musí definovat funkci se stejnou signaturou (název, parametry, návratový typ) jako má cílová metoda ve hře. Výsledkem je, že při volání původní metody se vykoná obal (anglicky *wrapper*), který může provést vlastní logiku před / po volání původní metody. Pokud neexistuje cílová třída (v příkladu dále je to `CPlayer`), nebo *wrapper* nemá stejnou signaturu, tak kompilace selže. Příklad použití této anotace:

```
@wrapMethod( CPlayer ) function CheatGod2(on : bool)
{
    wrappedMethod(on); // tento řádek volá originální metodu CheatGod2
    thePlayer.AddMoney(300);
}
```

- `@addMethod(class)` – přidá nově definovanou metodu (`scriptextAdd`) do specifikované třídy (`CPlayer`). Pokud cílová třída neexistuje, dojde k selhání při kompilaci. Příklad použití:

```
@addMethod(CPlayer) public function scriptextAdd()
{
    dummy = 12345;
}
```

- `@replaceMethod(class)` – zamění existující funkci (`CheatGod2`) ve specifikované třídě (`CPlayer`). Pokud ve specifikované třídě neexistuje zadaná funkce se stejným jménem, dojde k selhání při kompilaci. Příklad použití:

```
@replaceMethod(CPlayer) function CheatGod2(on : bool)
{
    thePlayer.AddMoney(2000);
}
```

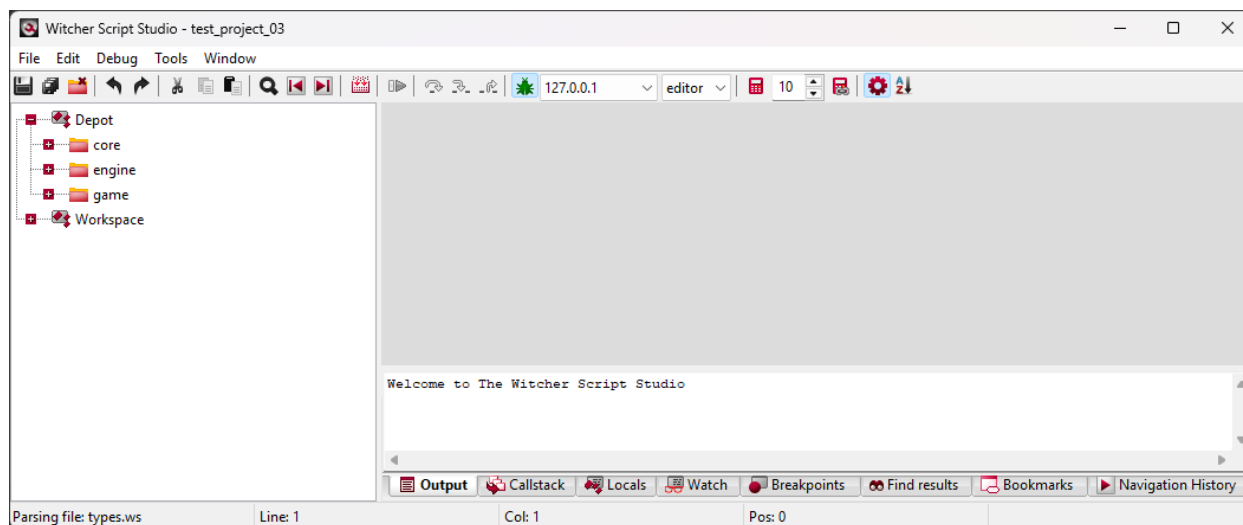
- `@replaceMethod` – zamění globální funkci, nspecifikuje se tedy žádná třída, pouze název globální metody. Pokud specifikovaná metoda neexistuje, dojde k selhání při kompilaci. Příklad použití:

```
@replaceMethod function klapaucius()
{
    thePlayer.AddMoney(1000);
}
```

- `@addField(class)` – přidá členskou proměnnou do specifikované třídy. Pokud specifikovaná třída neexistuje, dojde k selhání při kompilaci. Příklad použití:

```
@addField(CPlayer)
public var dummy: int;
```

3.6.6 Základy nástroje Witcher Script Studio



Obr. 45: Snímek zobrazující okno Witcher Script Studio

(zdroj: vlastní)

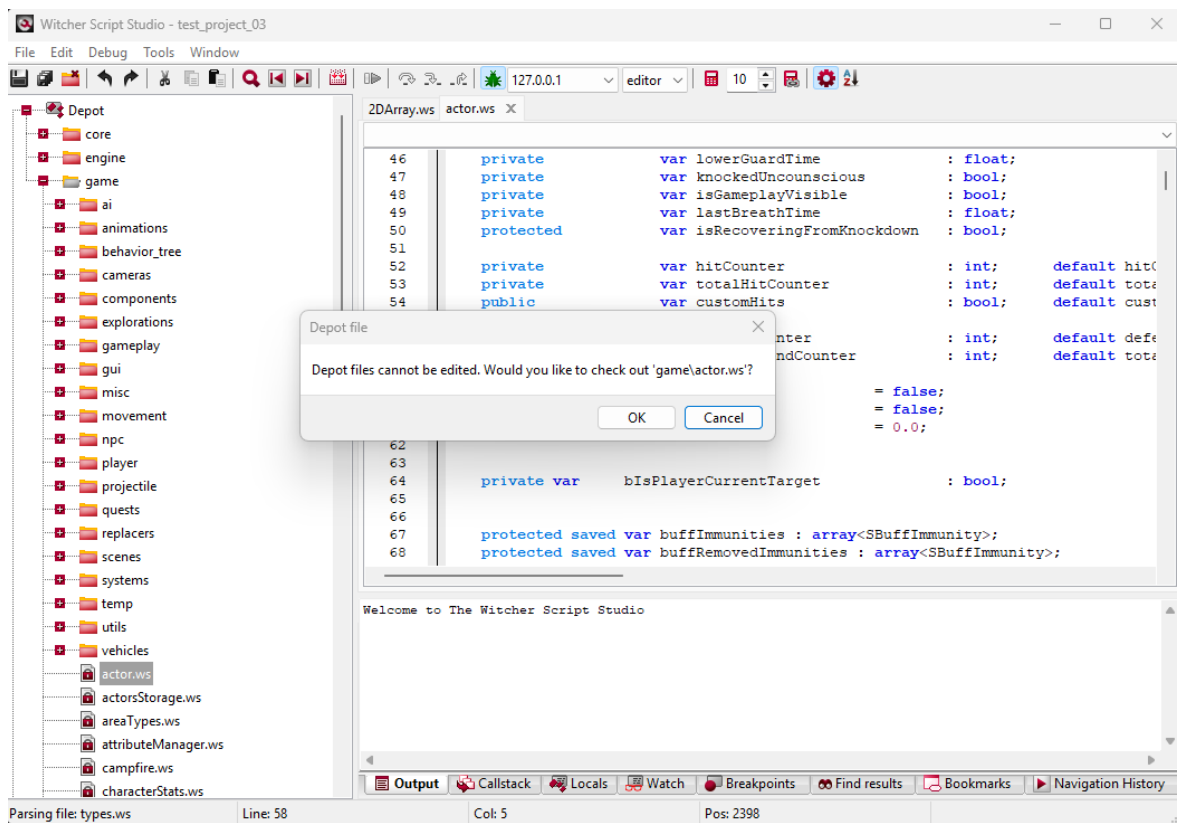
Podle oficiální dokumentace [37] se jedná se o nástroj obsažený v REDkitu, který lze použít pro editaci, vytváření a debugování WitcherScript funkcí. Studio lze spustit v REDkitu pomocí horního panelu nástrojů volbou možnosti “Tools – Script Studio...”. Studio se spustí v novém okně viz Obr. 45. Po spuštění projektu v REDkitu se studio stará o to, aby byly skripty synchronizované s otevřeným projektem.

3.6.6.1 Checkout souborů a Workspace

Při editování souborů je nutné provést tzv. *Checkout* modifikovaných souborů, tím se označí jako modifikované, budou se zobrazovat ve složce s označením *Workspace*, která označuje modifikované soubory, a díky tomu REDkit bude vědět, že je má při publikaci modu přibalit také.

Při pokusu o editaci souboru se Studio samo zeptá, zda chce uživatel “check out” daný soubor. Po potvrzení (možnost *OK*) viz Obr. 46 jí REDkit označí jako *workspace* soubor sám.

Witcher Script Studio umožňuje používat *breakpoints*, tedy body ve skriptu, na kterých se debugger pozastaví. Lze tak debugovat a upravovat kód během hraní v REDkit editoru.



Obr. 46: Snímek zobrazující pokus o editaci souboru ve Witcher Script Studio
(zdroj: vlastní)

4 Implementace externích aplikací Blender a Wwise

Tato kapitola popisuje, jakým způsobem lze implementovat externí software pro REDkit použitý v praktické části této práce. Při tvorbě modifikace pomocí nástroje REDkit je běžné kombinovat samotný editor s těmito externími nástroji. REDkit se používá k integraci výsledných dat do hry, zatímco externí aplikace slouží k vytvoření nebo úpravě zdrojového obsahu (3D modely, animace, zvuky). V této práci jsou klíčové zejména dvě externí aplikace: Blender pro 3D obsah a animace a Audiokinetic Wwise pro tvorbu a zabalení audia do formy použitelné herním engine.

4.1 Blender

Blender je open-source 3D software, který si uživatel může pořídit zdarma. Pokrývá řadu možností tvorby 3D obsahu jako je modelování, UV mapování, materiály, rigging a animaci, simulace a rendering. V kontextu modifikací umožňuje nedestruktivní úpravy (modifikátory), kontrolu topologie a zejména práci se skeletovou animací (armature), což je požadovaný základ pro úpravu animací postav a objektů.

4.1.1 Použití s REDkitem

Uživatel může použít Blender k vytvoření / editování obsahu (mesh, skeleton, animace), který se následně importuje do REDkitu. Také může sloužit jako prostředí, kde lze data z REDkitu otevřít, provést kontrolu (měřítko, orientace, hierarchie kostí atd.) a exportovat data do formátu, který REDkit očekává.

4.1.2 Integrace do REDkitu přes Blender plugin

CD PROJEKT RED poskytuje oficiální Blender plugin, který rozšiřuje Blender o import/export funkce pro datové formáty použité v REDkitu. Postup instalace add-onu do Blenderu je popsán v kapitole 3.5.6.3. Exportování z Blenderu a import do REDkitu je popsán v kapitole 3.5.6.5.

4.2 Audiokinetic Wwise

Wwise je software od společnosti Audiokinetic pro interaktivní média a video hry. Obsahuje audio authoring nástroj a cross-platform zvukový engine [38]. Wwise je dostupný zdarma pro nekomerční účely a s licencí pro herní developery.

4.2.1 Použití Wwise s REDkitem

REDkit používá Wwise pro téměř všechny audio-úlohy, od běžných zvukových efektů přes počasí, až po audio ve scénách a animacích. REDkit dokumentace zároveň přímo uvádí, že cílem je naučit se základní práci s aplikací a následně vytvořit zvuky pro různé účely (například počasí, scény, animace) [39].

4.2.2 Licence

V kontextu REDkitu je důležité, že pro použití Wwise nad určitý rozsah je potřeba licence. REDkit dokumentace [39] uvádí hranici 250 zvukových souborů přenesených do hry/modu, po jejímž překročení je nutné licenci získat. Zároveň dokumentace REDkitu popisuje postup, jak získat licenci pro potřeby REDkitu.

4.2.3 Základní použití a vazba na REDkit

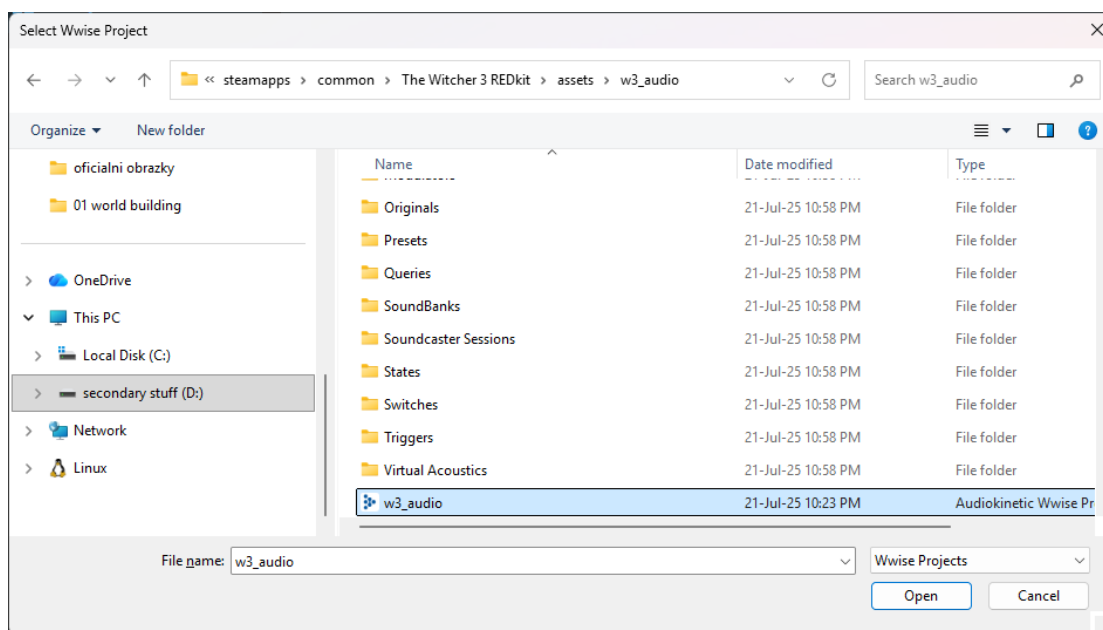
REDkit dokumentace [39] také popisuje návod k implementaci zvuků a vysvětluje, že práce ve Wwise se typicky soustředí na tři logické části (záložky), které na sebe navazují:

- Audio – obsahuje importované zvuky a jejich vlastnosti
- Events – události, které lze následně referencovat v REDkitu a vytvořit z nich tzv. soundEvents (tzv. dodání zvuku do modu [39])
- SoundBanks – seznam zvuků a eventů, které se mají načíst do hry/modu. SoundBank nejsou přímo zvuky, ale struktura odkazů na zvuky a jejich události

Z pohledu integrace Wwise do REDkitu je vhodné chápat Wwise jako “autoritu” nad zvuky, tedy pomocí Wwise se definují zvuky a jeho eventy, zatímco v REDkitu se tyto eventy používají v konkrétních herních kontextech (úkol, scéna, počasí, animace).

4.2.4 Příprava prostředí Wwise v rámci REDkitu

4.2.4.1 Instalace



Obr. 47: Snímek zobrazující umístění Wwise projektu uvnitř REDkit složky
(zdroj: vlastní)

Nástroj Audiokinetic Wwise Launcher může uživatel stáhnout zdarma z oficiálních stránek [38] a nainstalovat na své zařízení. Po instalaci Wwise Launcheru je potřeba doinstalovat verzi Wwise, to lze provést v Launcheru v záložce Wwise.

Po instalaci je software Wwise připraven na použití a uživatel v záložce Projects může otevřít Wwise projekt, nebo vytvořit nový projekt. Každý Wwise projektový soubor má příponu *.wproj*.

4.2.4.2 Získání licence

Oficiální dokumentace [40] REDkitu a online dokumentace Wwise softwaru [42] přímo popisují návod, jak získat licenci pro práci se softwarem Wwise určenou pro modifikace zvuků Zaklínače 3. Pokud uživatel očekává větší množství zvukových assetů, musí získat licenci.

Prvním krokem je vytvoření účtu na oficiálních stránkách Audiokinetic [38]. K vytvoření účtu je potřeba mít aktivní email.

Druhým krokem je na stránce Wwise software požádat o přístup u položky Witcher 3 – REDkit Modding [41]. Zde se rozchází tvrzení dokumentace REDkitu [39] s vydavatelem nástroje Wwise. REDkit uvádí 250 assetů bez nutnosti licence, zatímco Audiokinetic pouze 200 assetů.

Po požádání musí uživatel počkat na potvrzení notifikace o získání licence. Dokumentace REDkitu [40] uvádí dobu platnosti licence tři měsíce.

4.2.4.3 Otevření projektu a ověření propojení s REDkitem

Po instalaci Wwise a případném získání licence může uživatel otevřít soubor umístěný přímo ve složce s REDkitem: “The Witcher 3 REDkit – assets – w3_audio” viz Obr. 47 soubor *w3_audio.wproj*. Po vybrání tohoto projektu se spustí samotný nástroj Wwise ve stažené verzi a načte projektová data. Po načtení může uživatel již začít tvořit své úpravy tímto nástrojem.

B Praktická část

5 Návrh modifikace hry Zaklínač 3

Praktická část práce je realizována formou samostatné úkolové linie (anglicky *questline*), která rozšiřuje narativní obsah hry o novou lokaci, nové úkolové uzly, dialogové scény a odměny. Návrh je koncipován tak, aby byl realizovatelný v nástroji REDkit a současně demonstrativně využil typické “vrstvy” herního obsahu jako je svět/lokace, quest graph, příběhové scény, skriptové vazby a audio.

Příběhová linie se zabývá motivem pověsti Blaničtí rytíři [44], zejména pasáž související s kovářem a jeho interakcí s rytíři. Inspirace slouží jako výchozí rámec (motiv kováře, rytířů, magického jevu a zvuku bubnu a polnice na poli). Zatímco konkrétní zasazení, rekvizity a událost jsou přizpůsobeny světu Zaklínače tak, aby působily uvěřitelně v jeho pravidlech (monstra, magie, artefakty, zaklínačské vyšetřování, volby s následkem).

V rámci praktické části této práce jsou používány dva základní typy prostředků. První skupinu tvoří již existující prostředky, které jsou volně dostupné v prostředí REDkit. Jedná se tedy o assety pocházející z dat původní hry Zaklínač 3: Divoký hon, které jsou po vygenerování depotu přístupné v nástroji Asset Browser. Druhá skupina prostředků (assetů) jsou nově vytvořené, nebo upravené autorem této práce, například upravené 3D modely, textury, zvuky, nové úkolové a skriptové soubory.

5.1 Cíle návrhu a požadavky na modifikaci

Návrh modifikace pokrývá následující:

- Definici vstupního bodu modu pro hráče – získání herního úkolu
- Zavedení hráče do nových oblastí vytvořené v modifikaci
- Využití typických mechanik Zaklínače 3 – zaklínačské smysly, dialog, hledání stop
- Vytvoření volby s významným dopadem na průběh finále úkolu a odměny

5.2 Narativní východiska a tematika

Navrhnutá úkolová linie kombinuje tyto motivy:

- Záhada místa – v horské oblasti dochází k neobvyklým jevům, zejména podivná atmosféra v okolí hory (zvuk bubnů a polnice), zprávy o lidech, kteří se “ztratí” na rok.
- Kovář jako “quest giver” – poskytuje postavě Zaklínače příležitost dalšího obsahu do hry, zasvěťte jej do situace v nové lokalitě modu, a dává mu nový úkol v této lokalitě. Po vyřešení úkolu dává hráči odměnu. Lze jej interpretovat jako typ postavy zvaný “The Quest Giver” podle Rindo [43].
- Rozpad řádu rytířů – místo očekávaných rytířů hráč nachází pouze jejich ostatky a poškozený artefakt, který naznačuje magický původ a vysvětluje časové anomálie, jeho poškození implikuje selhání ochrany rytířů a jejich aktuální stav.

Tematicky je úkol postaven na kontrastu mezi osobním ziskem a stabilizací situace. Hráč je postaven před volbu, zda využít artefakt pro vlastní prospěch, nebo se pokusit napravit jeho stav a předejít možným následkům, případně se do situace nijak nezapojit.

5.3 Aktivace úkolu

Návrh počítá se dvěma variantami aktivace:

- Aktivace přes informační tabuli v obci – hráč získá quest přečtením lístku napsaného kovářem (kontrakt). Text zmiňuje podivné jevy v nové oblasti (horská oblast), mizící lidi, případně časovou anomálii. Tato varianta odpovídá běžnému způsobu získávání úkolu/kontraktů ve hře.
- Automatická aktivace po instalaci modu – úkol se založí hráčovi po splnění jednoduché podmínky, například vstup do určité oblasti, hráčova úroveň apod. Tato varianta slouží jako alternativní vstup.

V obou případech je prvním cílem hráče zjistit, jak se do horské oblasti dostat, například instrukce v lístku, přesun (teleport) z určeného místa.

5.4 Návrh lokací

Úkolová linie využívá dvě oblasti: horské údolí a jeskyně uvnitř hory.

5.4.1 Horská lokace

Venkovní svět s dominantou v podobě hory, nebo skalnatého kopce reprezentující horu Bláník z povídky Blaničtí rytíři [44]. Tato dominanta podobně jako píše Rindo [43] ve své kapitole o “Guiding the Player Experience”, slouží nejen jako pozadí nové lokace, ale úmyslně vede svou viditelností hráče a navádí ho jednoznačně k sobě, kde následně vede úkolová linie. V lokalitě se nachází zázemí kováře (chata). Prostor zázemí kováře slouží jako bezpečný bod pro zahájení a ukončení úkolu.

Před horou se podle povídky nachází pole, kde lze slyšet zvuky připomínající utlumené bubny a polnici. Zvuková stopa plní současně narativní (odkaz na povídku) a user-experience funkci (informování hráče, že se blíží k hledanému místu, o kterém slyšel od kováře).

Vstup do hory se jeví jako jeskyně pod vrcholem hory.

5.4.2 Jeskyně

Soustava chodeb a hal. Uvnitř hráč postupně zjišťuje stopy o přítomnosti rytířů pomocí zaklínačského smyslu. Závěrečná velká hala obsahuje kostlivce rytířů, centrální magický artefakt se zjevným poškozením. Jeskyně obsahuje také boční východ umožňující hráči rychlou cestu ven, pro případ únikové sekvence, nebo jako prevence uvíznutí hráče.

5.5 Struktura úkolové linie

Úkol je navržen jako sled kroků, které hráč musí splnit, aby úkol dokončil:

- Získání informace o úkolu (kontraktu) – tabule, nebo automatická aktivace
- Přesun do horské oblasti dle instrukcí
- Rozhovor s kovářem – expozice: rytíři, hora, časový posun, strach kováře se tam vrátit
- Cesta k hoře a průjezd polem se zvukem bubnů a polnice
- Nalezení vstupu do jeskyně – zaklínačské smysly, možný nález obsahu měšce podle povídky [44] (smetí a mince)
- Průchod jeskyní a odhalování stop – atmosféra, případný konflikt s příšerou (důkaz, že něco není v pořádku)
- Finále ve velké síni – artefakt a volba
- Návrat ke kováři a vyhodnocení – dialog a odměna dle volby

5.6 Finální volba a následky

Vrchol úkolu tvoří rozhodnutí u poškozeného artefaktu. Hráč dostane na výběr mezi konstruktivní/dobráckou volbou a destruktivní/sobeckou volbou.

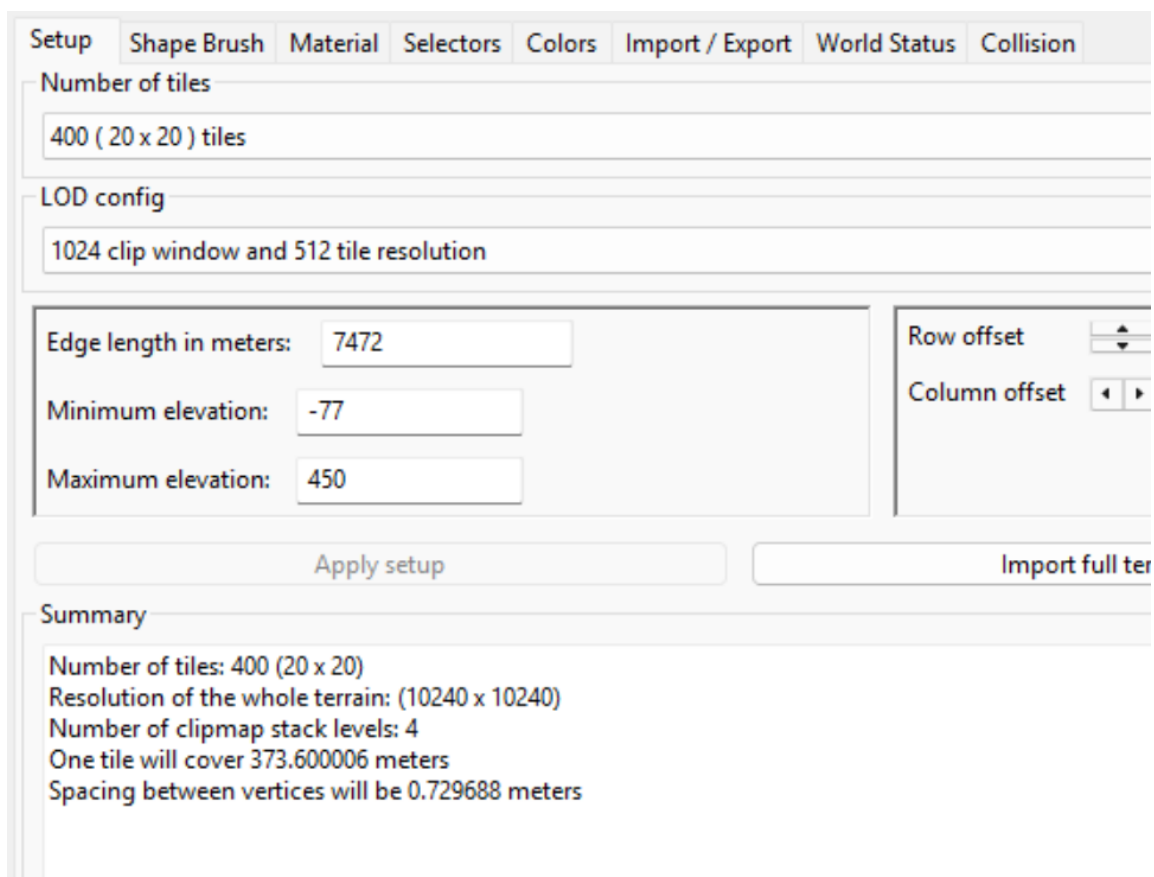
- Destruktivní volba – hráč odebere cennou část magického artefaktu. Následkem je kolaps jeskyně a úniková sekvence, případně znamení, o kterém se dozví kovář. Odměna je materiální (cenný předmět).
- Konstruktivní volba – hráč se snaží artefakt stabilizovat/opravit, případně ponechat tak. Jeskyně se nebortí a úkol je zakončen klidnějším vyústěním. Odměna je alternativní (zkušenostní body, recept, specifická odměna u kováře atd.).

Návrh předpokládá, že obě volby budou hodnotově srovnatelné, ale liší se charakterem. Hráč má možnost následně některé varianty odměny použít dále v klasické hře (motivace si tento mod zahrát).

6 Vytvoření venkovního světa

6.1 Vytvoření nového světa pro horskou oblast

Pro svět horské oblasti, ve které se musí nacházet zadavatel úkolu (kovář), hora samotná se vstupem do jeskyně a okolní tematické pozadí, byl zvolen svět o velikosti 400 dlaždic (20 x 20) s rozměrem přibližně 373 herních metrů na dlaždici viz Obr. 48.



Obr. 48: Nastavení světa horské oblasti
(zdroj: vlastní)

6.2 Tvorba struktury horské oblasti

V rámci tvorby terénu byla vytvořena struktura skupin a vrstev následovně:

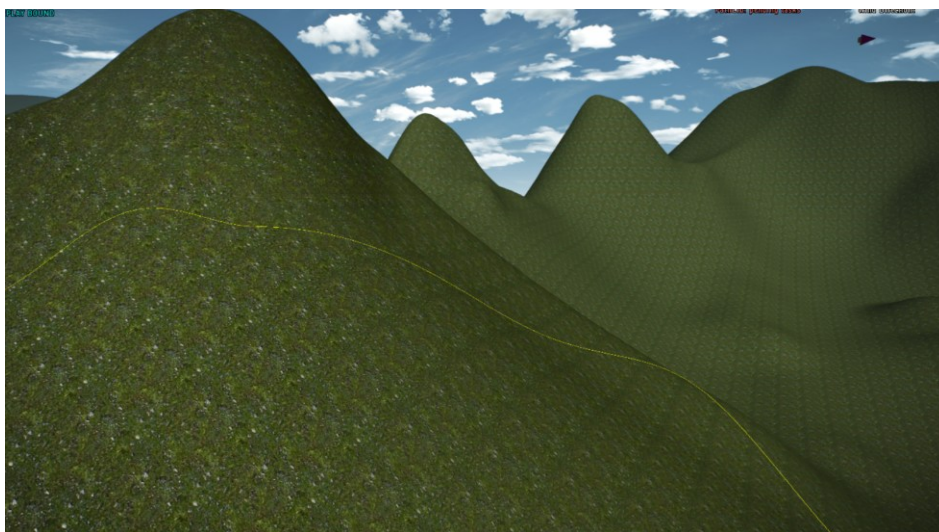
- Skupina *mountain_area*
 - *buildings* – vrstva pro umístění meshů ruin hradu a vstupu do jeskyně v okolí hory
 - *decorations* – dekorativní předměty v okolí hory
- Skupina *quest_giver_area*

- *buildings* – vrstva pro budovy, zdi a podobně v oblasti kováře
- *decorations_outside* – venkovní dekorace pro stáj, kovárnu atd.
- *decorative_inside* – dekorace uvnitř kovářova domu
- *interactive_objects* – objekty, se kterými může hráč interagovat
- *lights* – vrstva pro umístění zdrojů světla, jako jsou pochodně atd.
- *lootables* – určeno pro objekty, ze kterých může hráč získat kořist

6.3 Tvorba terénu horské oblasti

Horská oblast je otevřený kopcovitý / hornatý terén. Prvním krokem bylo aplikovat materiál v nástroji Terrain Edit Tools. Pro horskou oblast byl použit existující materiál dostupný v REDkitu, který poskytl vhodně barevný základ pro další modelaci a texturování povrchu viz Obr. 49, kde je viditelný jako zelený povrch terénu.

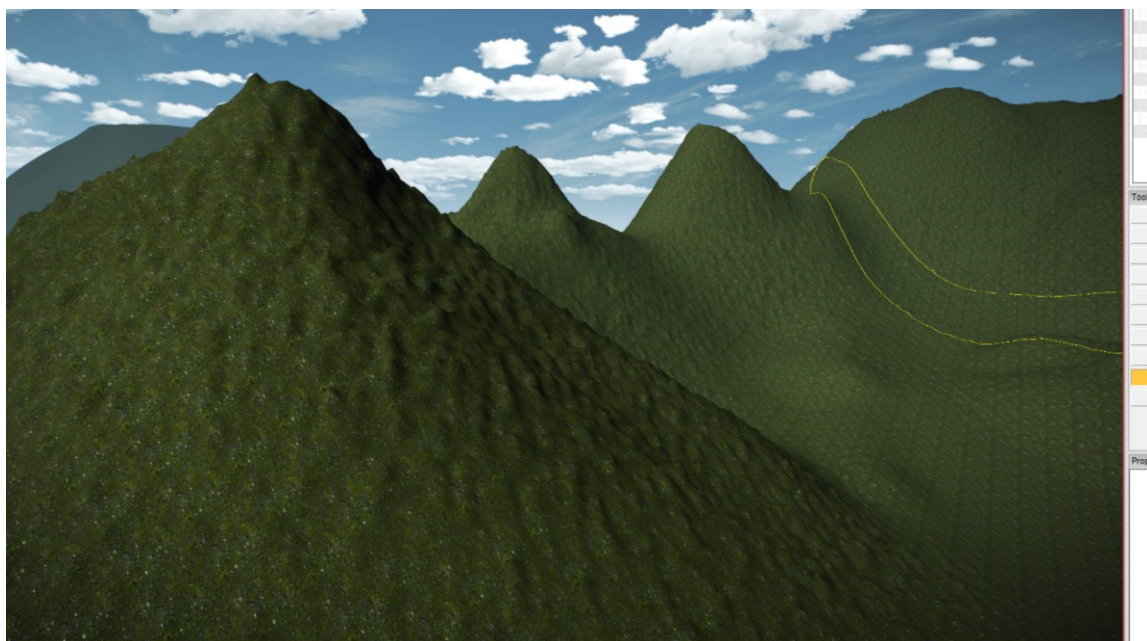
Dalším krokem byla modelace terénu, zejména hor a kopců. Pro tento účel byla použita sada štětců (anglicky *Shape Brush*) v nástroji Terrain Edit Tools. Pro vytvoření kopců a hor byl nejefektivnější štětec Rise, kterým šlo terén modelovat vertikálně vzhůru, ale také tvořit prohlubně. Ukázka části vymodelovaných hor viz Obr. 49. Tímto způsobem byl vytvořen terén pro celou mapu, včetně pozadí (hory v dálce, malé jezero atd.).



Obr. 49: Pohled na uskupení kopců vytvořených štětcem Rise

(zdroj: vlastní)

V rámci modelace terénu byly použity štětce Rise a Flatten s nastavením pravděpodobnosti (anglicky *probability*) na nízkou hodnotu ke zvrásnění terénu, zejména vrcholků hor, případně štětec Smooth, nebo Melt ke zjemnění. Ukázka aplikování tohoto postupu viz Obr. 50.



Obr. 50: Ukázka postupu zvrásnění terénu hor pomocí štětce Rise a nízké pravděpodobnosti
(zdroj: vlastní)

Po požadovaných deformacích terénu byly v nástroji Terrain Edit Tools v záložce Material použity různé materiály k obarvení odpovídajících ploch terénu. Ukázka aplikování skalnatého materiálu viz Obr. 51. Tímto způsobem byl “obarven” celý terén mapy.



Obr. 51: Ukázka aplikovaného horského materiálu na vrcholky hor
(zdroj: vlastní)

6.4 Tvorba dekorací a důležitých lokací

6.4.1 Vstup do jeskyně na východním svahu hory

Podle povídky má být vstup do hory k rytířům na východním svahu kopce, byl tedy štětcem Flatten připraven menší otvor ve zvoleném místě a připraven mesh vypadající jako vstup do krypty / jeskyně viz Obr. 52.



Obr. 52: Pohled na vstup do jeskyně umístěn na východním svahu hory
(zdroj: vlastní)

Pro reprezentaci kamenných hradeb na vrcholu hory byly zvoleny meshe odpovídající ruinám a nastavena jejich renderovací vzdálenost na „dostatečnou“ k zobrazení z okraje mapy viz Obr. 59 - slouží tak jako naváděcí prvek při orientaci v mapě.

6.4.2 Kovářova oblast



Obr. 53: Zvolený materiál pro půdu na pozemku kováře
(zdroj: vlastní)

Další částí bylo vytvořit oblast, kde sídlí kovář. Pro potřeby této lokality byly na terén aplikovány materiály odpovídající blátivé, kamenité a travnaté ploše, aby okolí obydlí působilo přirozeně a odpovídalo charakteru venkovského pracovního prostoru viz Obr. 53. Pro úpravu byl použit nástroj Terrain Edit Tools.



Obr. 54: Pohled na lokalitu obydlí kováře
(zdroj: vlastní)

Pozemek byl dále vymezen kombinací existujících meshů dostupných z REDkitu, které jsou vhodné jako stavební materiál pro reprezentování kamenné zdi a dřevěné konstrukční prvky. Tímto vznikl uzavřenější prostor odpovídající charakteru skromného obydlí venkovského kováře viz Obr. 54.



Obr. 55: Pohled na exteriérovou a interiérovou část domu kováře
(zdroj: vlastní)

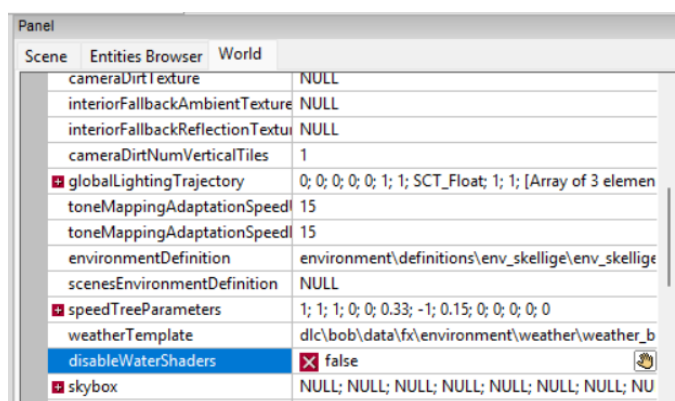
Samotná chata kováře byla vytvořena kombinací dvou existujících meshů, jeden tvořil vnější část budovy a druhý její interiér s několika dekoracemi. Oba objekty bylo nutné prostorově zarovnat a vzájemně sladit tak, aby výsledná stavba působila jako jeden funkční celek viz Obr. 55 s viditelným rozložením před zarovnáním obou meshů. Do kovářovy oblasti byly postupně během tvorby modifikace přidány další objekty do svých vrstev, vegetace atd. pro důvěryhodnější dojem viz Obr. 56.



Obr. 56: Pohled do hotové oblasti kováře z publikované verze modifikace (zdroj: vlastní)

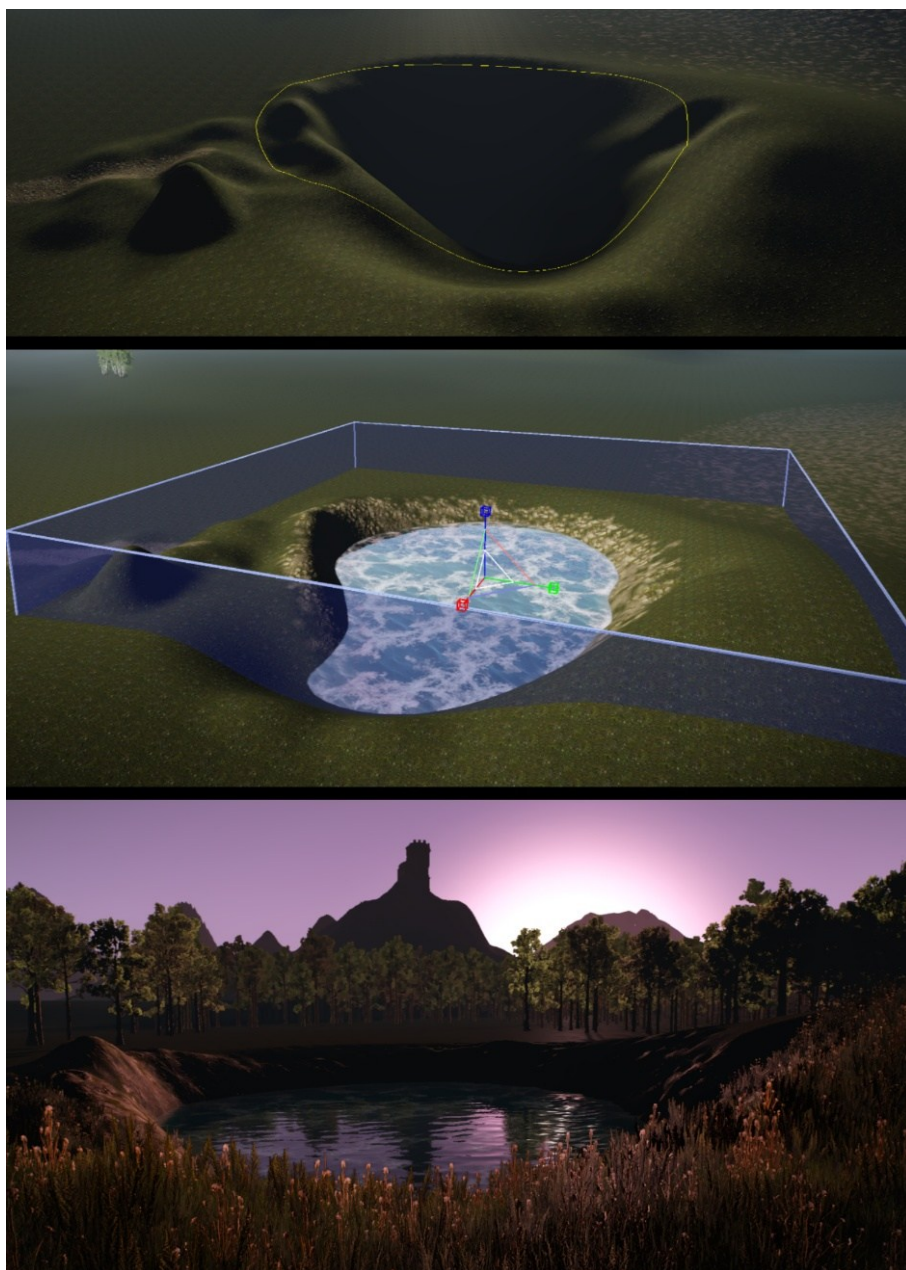
6.4.3 Malé jezero jako demonstrace použití vodní plochy

V terénu bylo také připraveno menší dekorativní jezero vytvořené pomocí předchozích postupů.



Obr. 57: Vyznačené nastavení pro umožnění zobrazení vodních ploch v editoru (disableWaterShaders) (zdroj: vlastní)

Nově bylo zapotřebí vytvořit novou vrstvu. Po aktivaci vrstvy dvojklikem levého tlačítka myši byla vložena vodní plocha kliknutím pravého tlačítka myši na požadované místo ve scéně a zvolením možnosti *Graphics – Add Local Water Area*. Pro možnost vyrenderování vodní plochy při editaci bylo zapotřebí v nastavení světa zvolit u *disableWaterShaders* možnost *false* viz Obr. 57. Výsledek po usazení vodní plochy do vyhloubené jámy je viditelný na Obr. 58 níže.



Obr. 58: Pohled před přidáním vodní plochy (nahore), umístění vodní plochy (uprostřed), po přidání vegetace (dole)
(zdroj: vlastní)

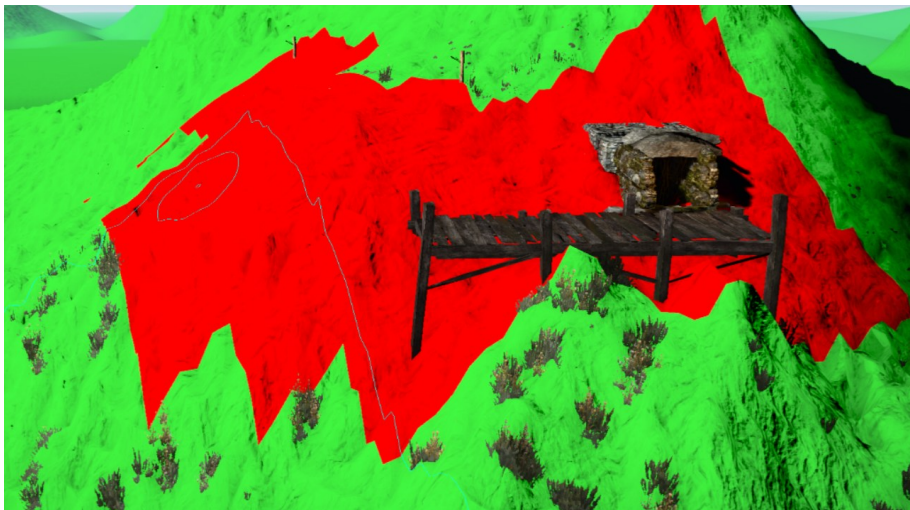
6.5 Vytvoření vegetace v oblasti

Dalším krokem bylo pokrytí terénu vegetací. K tomuto účelu byl použit nástroj Vegetation Edit Tools. Pro zalesněné oblasti byly z Asset Browseru přetažením myši do okna nástroje Vegetation Edit Tool vybrány štětce (anglicky *Brushes*). Podle potřeby byly “nakresleny” stromy po celé mapě viz Obr. 59 a to způsobem, aby byly vizuálně vyblokovány nechtěné oblasti, případně skrze koruny stromů viditelná významná místa mapy. Vegetace byla také umístěna na dno dekorativního jezera.



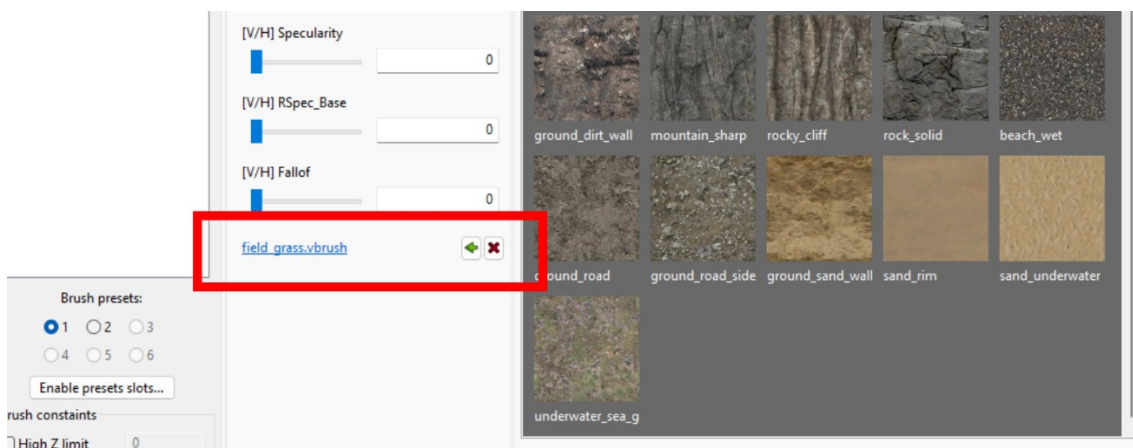
Obr. 59: Pohled z okraje horské oblasti na stezku vedoucí k ruině na hoře skrze zalesněnou oblast
(zdroj: vlastní)

Následujícím krokem bylo nastavení masky pro kresbu generické trávy. Nejprve bylo potřeba v nástroji Vegetation Edit Tool zvolit možnost Paint Grass Mask, která celou mapu obarví zelenou barvou a klikáním levého tlačítka myši byly obarveny červeně zóny, kde se tráva nemá generovat viz Obr. 60 - ukázka zakázání generování trávy v okolí vstupu do jeskyně.



Obr. 60: Ukázka úpravy masky pro generování trávy
(zdroj: vlastní)

Po vyřešení masky byl v Asset Browseru vyhledán štětec obsahující vhodně vypadající vegetaci pro trávu po jeho označení v Asset Browseru jej bylo možné zvolit pro generování trávy v nástroji Terrain Edit Tools v záložce Material a kliknutím na zelenou šipku viz Obr. 61 se vložil tento štětec a automaticky se vykreslila tráva do scény. Tento krok byl však později zjištěn jako nefunkční, respektive při testování v REDkitu se tráva vykreslila do hry v pořádku, v publikované verzi zcela chyběla. Bylo tedy nutné manuálně kreslit štětcem přímo do scény. Tento problém je také zadokumentován v kapitole 10.3.



Obr. 61: Nastavení štětce trávy pro generování podle masky
(zdroj: vlastní)

Další částí vegetace bylo vytvoření pole před horou podle povídky. K tomuto účelu byl použit vhodný štětec obsahující vegetaci vypadající jako obilí. Ukázka viz Obr. 62.



Obr. 62: Pohled na pole před horou
(zdroj: vlastní)

6.6 Nastavení definice prostředí v horské oblasti

Pro prostředí v horské oblasti byla zvolena definice prostředí *env_skellige_brown.env*, která byla nastavena pomocí parametru *environmentDefinition* v záložce World a v nastavení pravděpodobností pro každý typ počasí viz Obr. 63 v souboru *weather_bob.csv* byla nastavena pravděpodobnost 0.8 pro počasí *WT_Clear*, které pro danou lokalitu působí dojmem polojasné a klidné oblohy viz Obr. 62 – díky této pravděpodobnosti zůstává prioritním počasím.

weather_bob.csv - 2d Array Editor

File View

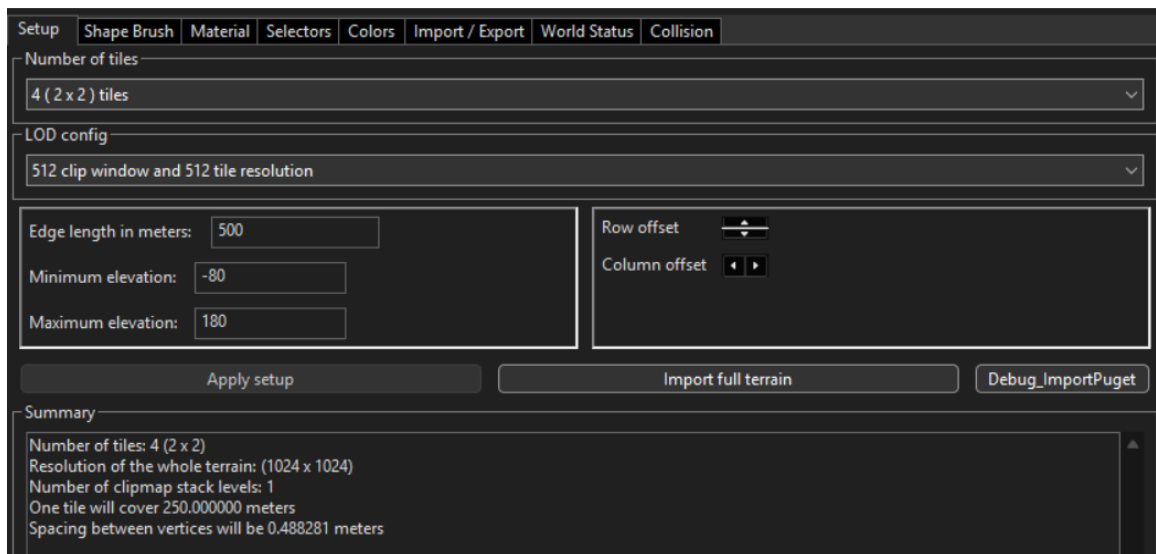
Line	name	probability
1	WT_Clear	0.8
2	WT_Rain_Storm	0.1
3	WT_Light_Clouds	0.3
4	WT_Mid_Clouds	0.4
5	WT_Fog	0.1
6	WT_Heavy_Distant	0.0
7	WT_Snow	0.0
8	q704_Cloud	0.0
9	q704_Detlaff_arena	0.0
10	q704_main	0.0
11	q704_Cloud_Rain	0.0
12	q704_Cloud_Sunny	0.0

Obr. 63: Detail nastavení druhů počasí horské oblasti a jejich pravděpodobností
(zdroj: vlastní)

7 Vytvoření jeskyně

7.1 Vytvoření nového světa a modelace terénu

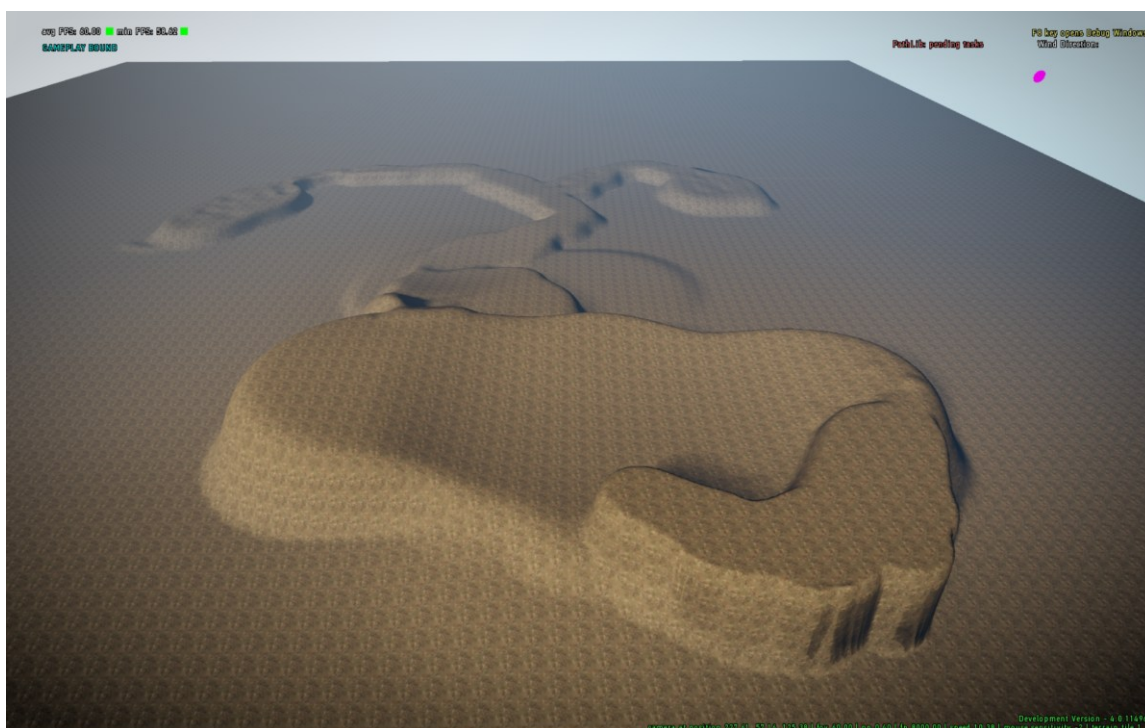
Tento svět, ve kterém se odehrává hlavní děj úkolu této modifikace, spočívá v uzavřeném interiéru. Pro tyto účely byl vytvořen malý svět s nastavením viz Obr. 64, obsahující pouze 4 dlaždice (2 x 2) s rozměrem 250 metrů na dlaždici.



Obr. 64: Nastavení světa pro jeskyni

(zdroj: vlastní)

Terén pro uzavřený svět skládající se pouze z chodeb (podlaha, stěny a strop) není nutný, ale může sloužit jako reference, na které lze tvořit samotný interiér. Pro referenční účel byl zvolen materiál (není podstatný, slouží pouze jako zabarvení terénu, aby nebyl neviditelný) a pomocí Terrain Edit Tools (nástroje určené k editaci terénu) byl terén z rovne plochy deformován do podoby viz Obr. 65, kde je viditelná vyvýšená plocha pro usazení vstupu do jeskyně, kde se hráč při vstupu do lokace objeví. Z této vyvýšené plochy by měl dále interiér jeskyně pokračovat chodbou směrem dolů, kde jedna z navrhnutých chodeb byla plánována jako hlavní místnost úkolu s rytíři.



Obr. 65: Náhled modelace terénu pro referenci půdorysu jeskyně
(zdroj: vlastní)

7.2 Tvorba struktury jeskyně

Pro účel modifikace byly vytvořeny tři místnosti: vstupní hala, skladiště a rytířská místnost. Vstupní hala přímo navazuje krátkou chodbou na skladiště, ze kterého se hráč dostane do hlavní chodby, kde dochází ke změně stylu dekoru a končí v rytířské místnosti.

Základní postup tvorby jednotlivých místností byl stejný. Nejdříve byla vytvořena skupina (anglicky *group*), poté uvnitř skupiny vytvořeny jednotlivé vrstvy (anglicky *layers*), tak, aby každá místnost nebo chodba měla 4 vrstvy: dekorace, podlaha, stěny, strop.

Pro usnadnění vyhledání vhodných meshů bylo využito schopnosti Asset Browseru vyhledat zadaný výraz, typicky třídu (anglicky *Class*) CMesh a poté relevantní hledané slovo, například *cave_wall* atd. Každý mesh bylo potřeba pomocí nástrojů pro posun, rotaci a případně škálování upravit do vhodné pozice.

7.2.1 Vstupní hala

Tvorba začala vždy podlahou, nejdříve byla aktivována odpovídající vrstva dvojklikem myši. Pro podlahu bylo v Asset Browseru zvoleno několik meshů odpovídajících stylisticky jeskyni viz Obr. 66 a byly postupně vkládány přetažením do scény v REDKitu. Přidané mesh se automaticky vložily do aktivní vrstvy.



Obr. 66: Pohled na tvorbu podlahy vstupní haly jeskyně
(zdroj: vlastní)

Po podlaze byly vytvořeny stěny vstupní haly viz Obr. 67 obdobným způsobem, tedy byla aktivována odpovídající vrstva, pomocí Asset Browseru nalezeny vhodné meshe, které byly následně přetaženy do scény.



Obr. 67: Pohled na podlahu a stěny vstupní haly jeskyně
(zdroj: vlastní)

Třetím krokem vstupní haly bylo vytvoření stropu viz Obr. 68, který uzavírá interiér. Opět byla aktivována odpovídající vrstva, vyhledány vhodné meshe a vloženy do scény.



Obr. 68: Pohled na vstupní halu po dokončení podlahy, stěn a stropu
(zdroj: vlastní)

Další částí vstupní haly bylo vytvoření dekorací a prvků pro zamaskování přechodů (anglicky *seams*) mezi grafickými prvky. Pro tento účel byly v Asset Browseru vyhledány odpovídající meshe nebo šablony entit, které vizuálně zapadly do prostoru jeskyně a působili tak přirozeně. Ukázka použitých dekorací jako jsou kameny s mechem apod. viditelné na Obr. 69.

Pro každou místnost bylo důležité i osvětlení, pro které byla vytvořena další vrstva. Do této vrstvy byly vloženy objekty vydávající světlo pomocí šablon entit (například pochodně) a nastavení jejich vlastností (barva světla, které pochodeň generuje, intenzita apod). Výsledek tvorby osvětlení vstupní haly ukázán na Obr. 69.



Obr. 69: Pohled na hotovou část vstupní haly jeskyně
(zdroj: vlastní)

7.2.2 Skladiště



Obr. 70: Pohled na ukázkou dekorací ve skladišti
(zdroj: vlastní)

Místnost skladiště viz Obr. 70 je určena jako místo, kde je pro hráče poznatelné, že jeskyně není pouze přírodního charakteru, ale také se stopami po delším obývání někým neznámým.

Postup tvorby byl obdobný jako u vstupní haly, nejdříve byly vytvořeny odpovídající vrstvy pro strukturu skladiště a přilehlé chodby (podlaha, stěny, strop a dekorace). Postupně byly dohledány a použity vhodné vypadající meshe a šablony entit tak, aby reprezentovaly místnost skladiště. Místnost byla také doplněna o osvětlení pomocí šablon entit krbů, pochodní a svícňů. V rámci vrstev byla zakomponována i menší spojovací chodba kolem skladiště viz Obr. 71.



Obr. 71: Pohled na spojovací chodbu kolem skladiště jeskyně z průběhu modelování
(zdroj: vlastní)

7.2.3 Hlavní chodba jeskyně

Hlavní chodba viz Obr. 72 navazuje za skladištěm na spojovací chodbu. Chodby jsou vytvořeny tak, aby byl zřejmý rozdíl v kvalitě materiálů a stylu. Chodba vede do cílové místnosti rytířů. Základní vrstvy (podlaha, stěny, strop a dekorace) byly vytvořeny opět obdobným způsobem jako předchozí místnosti.



Obr. 72: Pohled na půdorys hlavní chodby vedoucí k místnosti s rytíři
(zdroj: vlastní)

Hlavní chodba byla doplněna o osvětlení tak, jako předchozí místnosti. Byly také přidány dekorativní prvky (sochy, pavučiny) viz Obr. 73 do vlastní vrstvy.



Obr. 73: Ukázka hlavní chodby s dekoracemi a osvětlením
(zdroj: vlastní)

7.2.4 Rytířská místnost

Rytířská místnost je finální místnost jeskyně. Tvorba probíhala stejně jako předchozí oblasti jeskyně. Nejdříve byla vytvořena struktura místnosti (podlaha, stěny, strop, dekorace). Výjimkou (oproti předchozím oblastem) je v rytířské místnosti použití meshe pro vyplnění mezer ve stropu, nebo skulinách mezi modely, byl zvolen mesh vypadající jako hlína s kamenem viditelný v horních částech Obr. 74. Mesh byl vložen do vrstvy stropu této místnosti.



Obr. 74: Pohled z ptačí perspektivy na půdorys rytířské místnosti z průběhu tvorby (zdroj: vlastní)

Osvětlení místnosti bylo provedeno obdobně jako předchozí místnosti, s tím rozdílem, že byly vybrány šablony entit kvalitněji vypadajících zdrojů světla, jako je lustr, svícný, ohniště. Ukázka hotové struktury rytířské místnosti s osvětlením viz Obr. 75.



Obr. 75: Pohled od vstupu do rytířské místnosti
(zdroj: vlastní)

7.2.5 Kořist a interaktivní objekty

Pro kořist a případné interaktivní objekty byla přidána vrstva do místností, kde tyto objekty byly umístěny. Objekty s kořistí, jako jsou například bedny, krabice, police atd. byly vyhledány v Asset Browseru jako šablony entit s vhodným vzhledem a vloženy na svá místa. Hráč s nimi tak může při hraní interagovat a získat případnou kořist. Zároveň tyto předměty ve scéně slouží jako tematický předmět zasazen do daných místností – příklad viz Obr. 76.



Obr. 76: Pohled na umístěný interaktivní objekt s kořistí ve skladišti (stojan na meče)
(zdroj: vlastní)

7.3 Nastavení prostředí světa

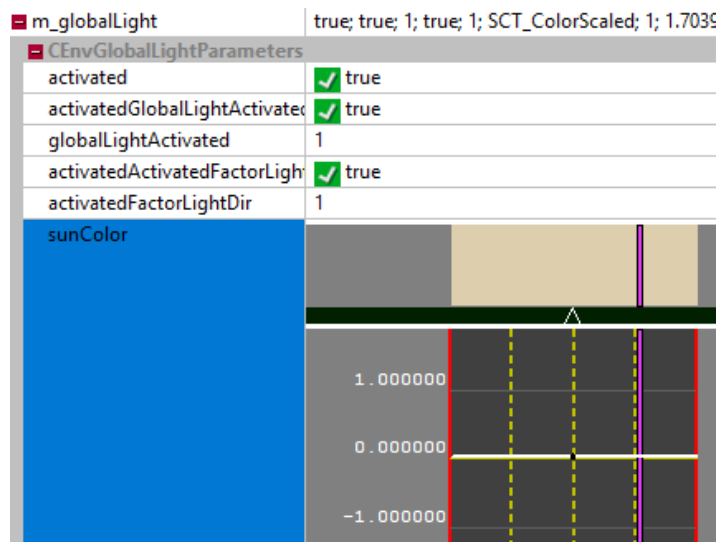
7.3.1 Nastavení definice prostředí

V rámci prostředí světa jeskyně byly nastaveny parametry prostředí (anglicky *environment parameters*) v záložce *World* tak, aby atmosféra odpovídala tmavé uzavřené jeskyni. Pro tento účel byl v záložce *World* a položce *environmentParameters* nastaven soubor definice prostředí (anglicky *environment definition*) na *env_global_clear_interior.env*. Tento soubor byl následně uložen do projektového adresáře pomocí *Checkout* možnosti v *Asset Browseru*. Tímto byla umožněna jeho persistentní editace v nástroji editor prostředí (anglicky *Environment Editor*) pro tento projekt. V tomto editoru byl nastaven preset na *full_color*. Pro odstranění nerealistických stínů a odlesků vycházejících z umělého slunce, viditelné i v uzavřeném interiéru, viz Obr. 77, byly upraveny přednastavené parametry.



Obr. 77: Pohled na nerealistický stín sochy a odlesk na podlaze způsobený umělým sluncem, před jeho úpravou
(zdroj: vlastní)

Úpravou parametru *m_globalLight* a mimo jiné jeho hodnoty *sunColor* na hranici kolem 0 - viz snímek upravených hodnot na Obr. 78., byly vypnuty stíny a odlesky umělého slunce.



Obr. 78: Snímek zobrazující nastavení globálního světla v prostředí jesyně
(zdroj: vlastní)

Viditelný rozdíl po úpravě viz Obr. 79, stín sochy a odlesky na podlaze způsobené sluncem již nejsou, zůstaly pouze stíny a odlesky z místních zdrojů světla.



Obr. 79: Pohled na výsledek po úpravě hodnot umělého slunce, již není vidět nerealistický stín sochy na podlaze
(zdroj: vlastní)

Další úpravou bylo nastavení mlhy pomocí parametru `m_globalFog`. Úpravou jeho parametrů jako jsou `fogDensity`, `fogVertDensity`, `fogAppearRange` a `fogAppearDistance` byla nastavena jemně hustá mlha na koncích chodeb. Ukázka porovnání přidání mlhy viz Obr. 80.

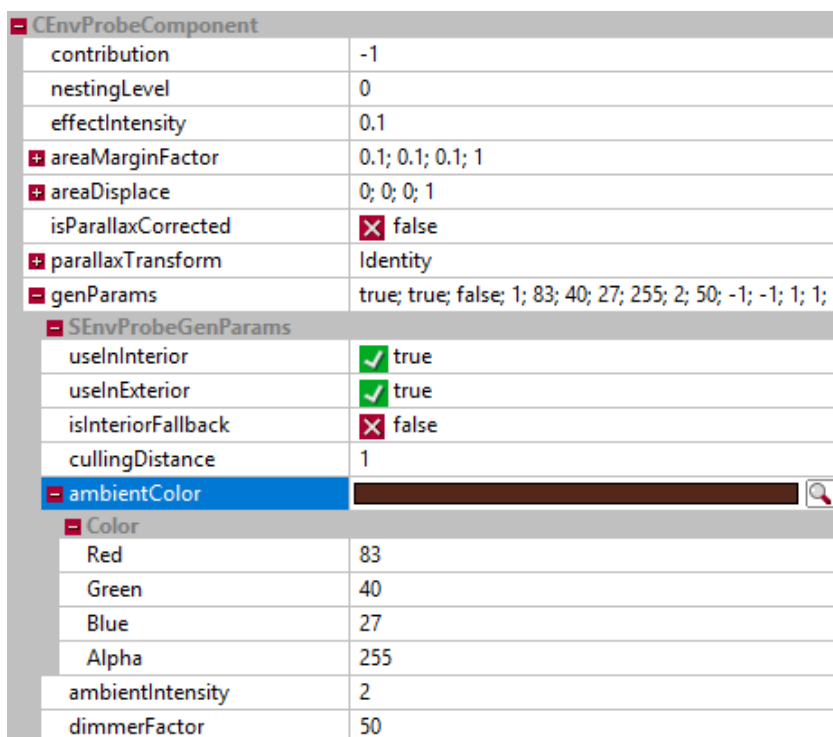


Obr. 80: Porovnání před a po přidání mlhy do oblasti jeskyně
(zdroj: vlastní)

Šablona počasí (anglicky *weather template*) nebyla nastavena, jelikož pro svět obsahující pouze uzavřený jeskynní prostor není potřeba, odpovídající parametr v záložce *World* zůstal nastaven na *NULL*.

7.3.2 Environmentální sondy

Environmentální sondy (anglicky *environmental probes*) byly umístěny do vlastní vrstvy. Pro globální odlesky byla vytvořena jedna globální sonda a to tak, že byla aktivována vrstva pro sondy a do scény pomocí pravého tlačítka myši v kontextovém menu vybrána možnost Graphics – Add Environment Probe. Ve vlastnostech této sondy byly nastaveny parametry tak, aby celá oblast měla v odlescích lehce tmavě červený nádech. Nastavení globální sondy viz Obr. 81, parametrem *contribution* nastaveným na hodnotu -1 se ze sondy stala globální sonda s nejnižší prioritou.



CEnvProbeComponent	
contribution	-1
nestingLevel	0
effectIntensity	0.1
areaMarginFactor	0.1; 0.1; 0.1; 1
areaDisplace	0; 0; 0; 1
isParallaxCorrected	<input checked="" type="checkbox"/> false
parallaxTransform	Identity
genParams	true; true; false; 1; 83; 40; 27; 255; 2; 50; -1; -1; 1; 1;
SEnvProbeGenParams	
useInInterior	<input checked="" type="checkbox"/> true
useInExterior	<input checked="" type="checkbox"/> true
isInteriorFallback	<input checked="" type="checkbox"/> false
cullingDistance	1
ambientColor	<input type="color" value="#8B4513"/>
Color	
Red	83
Green	40
Blue	27
Alpha	255
ambientIntensity	2
dimmerFactor	50

Obr. 81: Nastavení parametrů komponenty globální sondy pro jeskyni

(zdroj: vlastní)

Obdobným způsobem byla vytvořena druhá environmentální sonda pouze pro horní část jeskyně, kde je umístěn vstup s odleskem modré barvy. Aby mohla získat prioritu nad globální sondou byl nastaven její parametr *contribution* na hodnotu 1. Ukázka rozdílu před a po aplikování sondy viz Obr. 82.

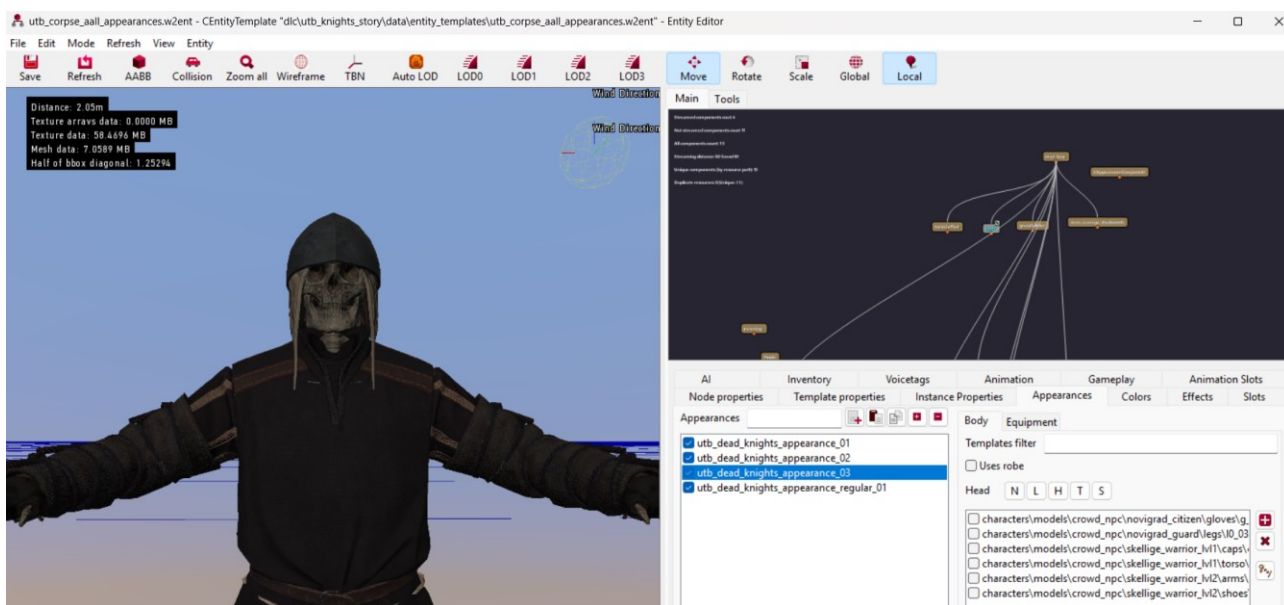


Obr. 82: Ukázka rozdílu bez použití environmentální sondy (vlevo) a s použitím globální environmentální sondy k získání tmavě červeného odlesku (vpravo)
(zdroj: vlastní)

7.4 Využití Blenderu pro tvorbu rytířů

Pro rytíře byly vyhledány v Asset Browseru vhodné meshe a šablony entit odpovídající vzhledem nábytku. Nábytek byl umístěn do míst, kde by se postavy rytířů mohly nacházet. Zejména postel a lavice se stoly. Pro postel byl vytvořen model ležícího kostlivce v civilním oděvu ve variantě na zádech a na břiše. Pro lavice byly vytvořeny čtyři varianty sedících rytířů v různém brnění. Zajímavější a složitější byla tvorba rytířů v brnění, proto je popsána pouze tato část. Během editace v Blenderu bylo využito rad a zásad uvedených v literatuře Blender 3D Asset Creation for the Metaverse [46]. Všechny kroky tvorby sedících rytířů jsou obdobné jako tvorba ležících rytířů.

Prvním krokem bylo vytvoření šablony entity, kde bylo využito znalostí z kapitoly 3.5.8 popisující Editor Entit, zejména část o *Appearances*. Vytvořené šabloně entity z prostředků nalezených volně v Asset Browseru jako je tělo kostlivce, brnění, kusy oděvu atd. bylo sestaveno několik vzhledů (anglicky *Appearances*). Ukázka nově vytvořené šablony entity kostlivce viz Obr. 83.

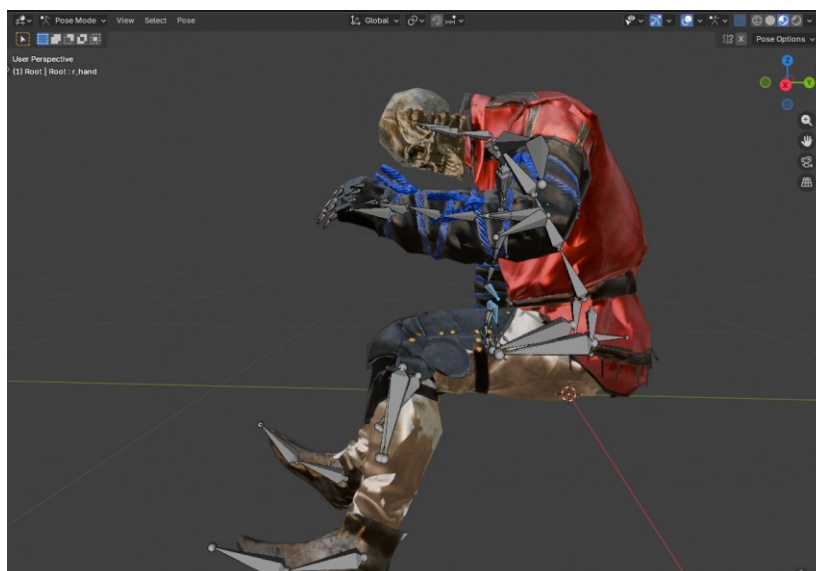


Obr. 83: Ukázka tvorby šablony entity kostlivce a jeho vzhledů (anglicky *Appearances*)

(zdroj: vlastní)

Pro nastavení správné pózy kostlivce bylo nutné tento soubor exportovat z REDkitu skrze Asset Browser jako soubor. FBX, který byl importován do Blenderu jako *FBX (.fbx)* (*Legacy*) soubor. V nastavení importu v Blenderu byla změněna pouze možnost v položce *Armature* takto:

- Primary Bone Axis: X Axis
- Secondary Bone Axis: Y Axis



Obr. 84: Ukázka použití Pose Mode v Blenderu pro úpravu pozice armatury modelu kostlivce

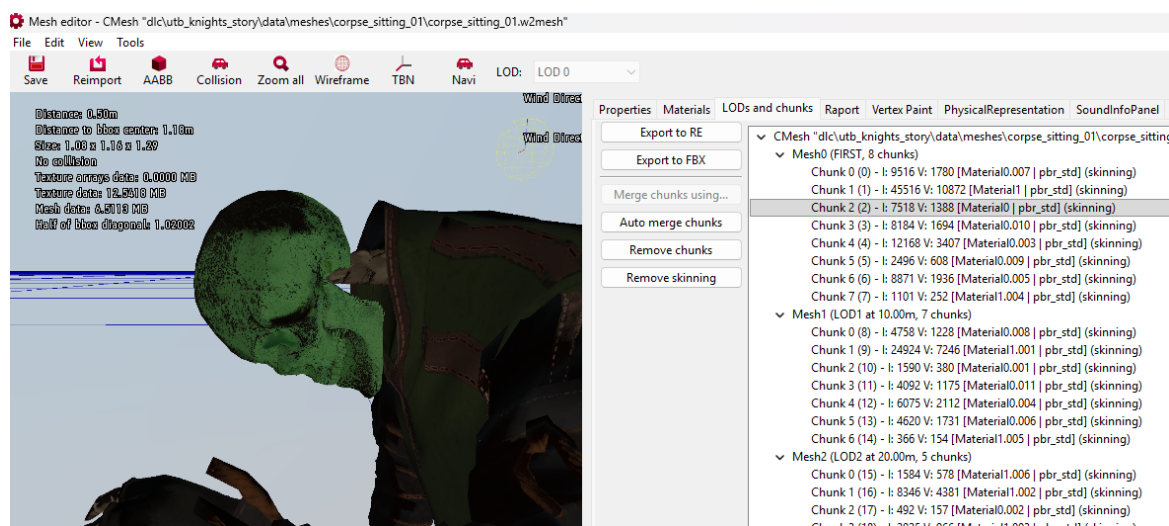
(zdroj: vlastní)

Importovaný model byl v Blenderu pomocí jeho *Pose Mode* režimu upraven model tak, aby vypadal, že sedí – viz Obr. 84. Poloha modelu byla také umístěna co nejvíce do středu koordinačního systému Blenderu. Hotový model byl následně exportován z Blenderu jako soubor. FBX s aktivní možností *Only Deform Bones* a nastavením *Armature* takto:

- Primary Bone Axis: X Axis
- Secondary Bone Axis: -Y Axis

Exportovaný soubor byl následně importován do REDkitu, ale tentokrát jako *Mesh* pomocí kontextového menu Asset Browseru. Celý tento proces byl opakován pro každou variantu vzhledu modelu kostlivce.

Dalším důležitým krokem bylo po otevření importovaných meshů v Mesh editoru sloučení částí modelu v jednotlivých úrovních detailů (anglicky *LODs and chunks*), jelikož u některých modelů se například část modelu lebky rozdělila klidně na tři individuální části. Ukázka sloučené lebky viz Obr. 85.



Obr. 85: Ukázka slučování částí modelu lebky dohromady (zeleně vybarvená je sloučená lebka modelu kostlivce)

(zdroj: vlastní)

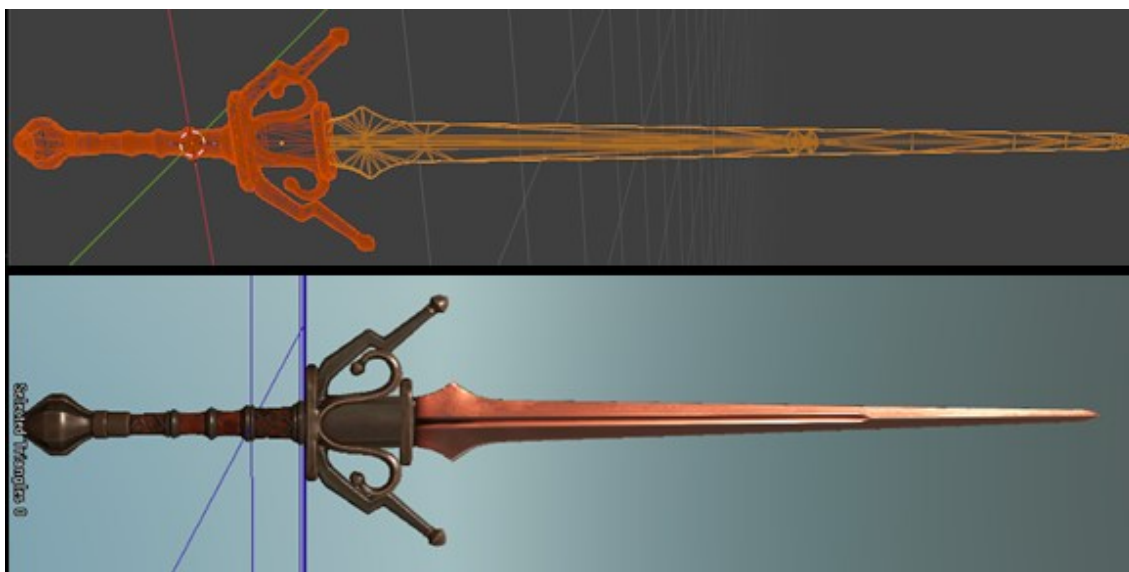
Importovaný mesh neměl nastaven materiál, ani shader. Bylo zapotřebí manuálně každé části modelu nastavit odpovídající materiál a shader ze záložky *Materials*. Pro demonstrování možnosti modifikovat části oděvu a textur obecně byl pro každou variantu vzhledu kostlivce exportován jeho soubor(y) typu .xbm obsahující 2D textury oděvu jako soubory typu .dds a dále upraveny ve fotoeditačním software paint.NET [45] do odstínů zelené a černé barvy. V REDkitu byly zpátky naimportovány na stejná místa, aby přepsaly předchozí neupravené soubory. Tyto soubory byly nastaveny jednotlivým částem modelu v Mesh editoru v záložce *Materials* v parametru *Material parameters* a jeho položkách *Diffuse* a *Normal*. Zeleno-černě upravená textura modelu (resp. jeho oděvu) je viditelná například na Obr.

83 a Obr. 85. Proces manuálního sloučení částí modelu a nastavení materiálů a shaderu byl opakován pro každý importovaný mesh.

Z hotových mesh souborů byly vytvořeny šablony entit, kde byl použit jeden mesh s požadovanou variantou sedu a oděvu. Pro variaci byly použity jiné přilby stejně sedícím kostlivcům. Vznikla tak jedna šablona entity pro každý celkový vzhled kostlivce.

Na závěr pro sloučení a snazší použití byla vytvořena jediná šablona entit pro sdružení všech vzhledů sedícího kostlivce. Tato šablona již byla připravena k použití do světa.

7.5 Využití Blenderu pro tvorbu meče



Obr. 86: Model meče jako odměna hráči za opravu artefaktu (Blender nahoře, REDkit dole)
(zdroj: vlastní)

Jedna z možných odměn pro hráče je nová zbraň (meč). K vytvoření bylo využito již existujících modelů jiných mečů, které byly z REDkitu exportovány do Blenderu obdobným způsobem jako byli vyřešeni kostlivci. Pro potřeby této modifikace z nich byl v Blenderu vytvořen jeden unikátní meč – viz Obr. 115 - pomocí standardních úprav v Blenderu. Dále byl model exportován zpět do REDkitu obdobně jako kostlivci.

Po importu zpět do REDkitu jako Mesh soubor bylo potřeba opět nastavit materiály pro jednotlivé části meče. Materiály byly pro meč nastaveny tak, aby působil unikátním dojmem. Je specifický svým načervenalým ostrím, obdobně jako záře opraveného artefaktu – viz Obr. 114. K meči byla také přiřazena odpovídající pochva na meč.



Obr. 87: Ikona meče vytvořená pro zobrazení v inventáři hráče
(zdroj: vlastní)

Pro správné zobrazení meče v inventáři hráče byla také vytvořena ikona jako *png* obrázek – viz Obr. 87. Dále bylo nutné přidat tento předmět do souboru obsahujícího předměty vytvořené touto modifikací, kde byly nastaveny údaje pro meč, jako je například jeho cena, poškození dávané nepřítelům, ikona v inventáři, pochva a podobné.

Část kódu řešící nastavení meče je popsána v příloze A.1 viz Výpis 1.

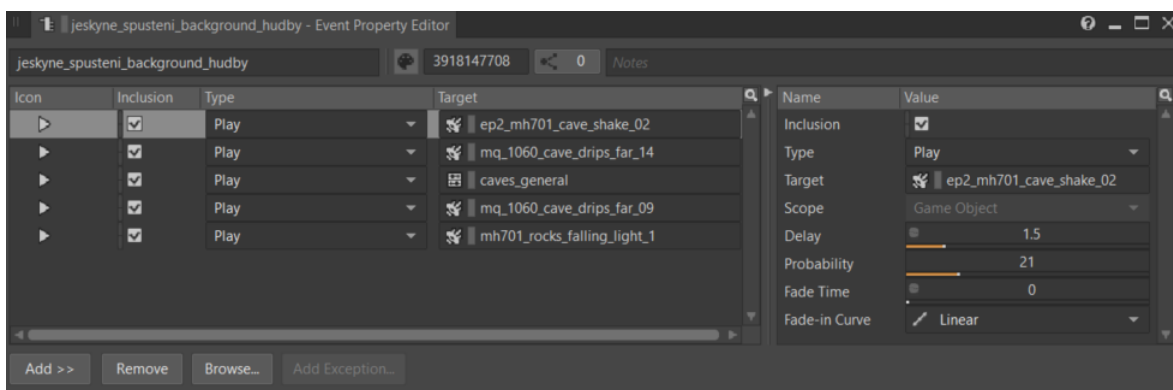
8 Tvorba zvuků pro modifikaci

K tomuto účelu byl použit software Wwise ve verzi 2021.1.14. Pro vytvoření vlastní zvukové banky (anglicky *sound bank*), a také událostí, které spustí požadované zvuky ve hře, bylo zapotřebí nad rámec oficiální dokumentace spustit soubor přiložený v adresáři s nainstalovaným REDkitem, konkrétně soubor `..\The Witcher 3 REDkit\assets\w3_audio\run_extract_original_audio.bat`. Tento soubor umožnil pracovat softwaru Wwise s audiosoubory REDkitu konverzí z formátu `.opus` na `.wav`. Pro jeho funkčnost bylo zapotřebí stáhnout z odkazu: <https://www.gyan.dev/ffmpeg/builds/ffmpeg-git-essentials.7z>, program `ffmpeg.exe`, který bylo nutné před spuštěním souboru `run_extract_original_audio.bat` umístit do adresáře `..\The Witcher 3 REDkit\assets\w3_audio\Originals`.

8.1 Přidání ambientní hudby pro horský svět a jeskyni

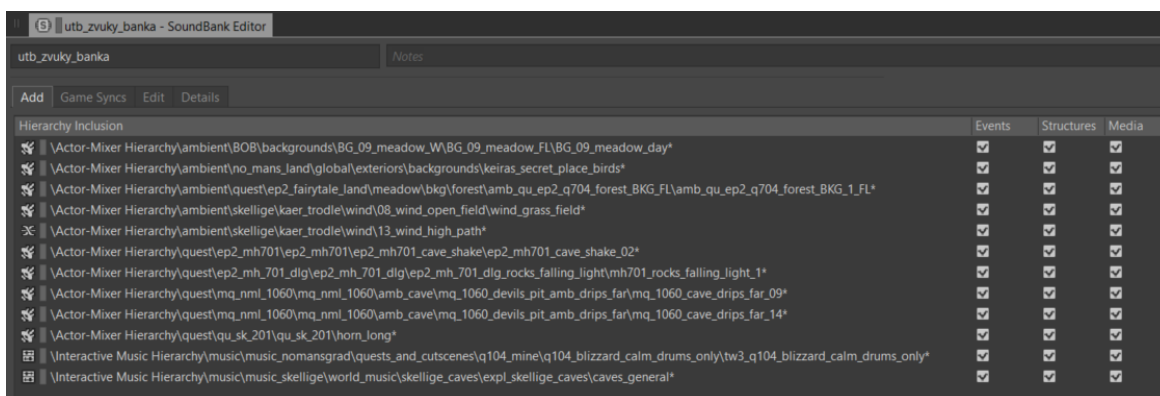
V softwaru Wwise byla v záložce Events vytvořena pro zvuky prostředí obou světů nová pracovní jednotka (anglicky *worker unit*), do které byly přidány nové události deskriptivně nazvané tak, aby název odpovídal místu a typu události. Například název události pro spuštění ambientní hudby na pozadí jeskyně byl zvolen jako „jeskyne_spusteni_background_hudby“.

Do každé události byly ze záložky Audio vloženy zvuky a melodie pro odpovídající oblast pomocí *Event Property Editoru* viz Obr. 88 s typem *Play*.



Obr. 88: Event Property Editor ze softwaru Wwise zobrazující audio soubory pro událost spuštění hudby na pozadí v oblasti jeskyně
(zdroj: vlastní)

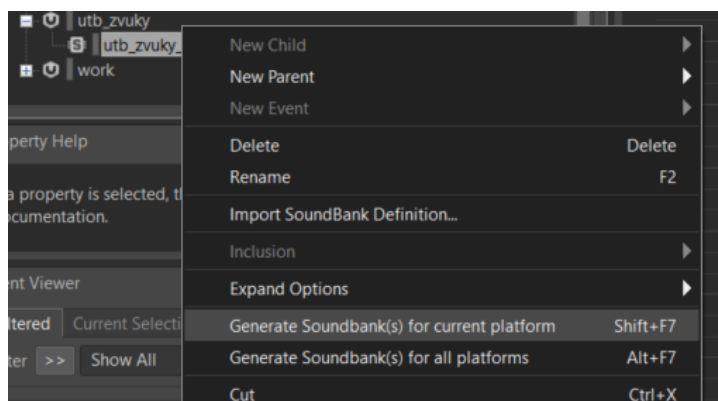
Vytvoření zvukové banky bylo provedeno v záložce *SoundBanks* vytvořením nové pracovní jednotky a následně vytvořením nové zvukové banky. Do banky byly přidány použité audio soubory přetažením z okna Event Property Editoru viz Obr. 88 do okna SoundBank Editoru viz Obr. 89.



Obr. 89: Ukázka s vloženými audio soubory do SoundBank Editoru

(zdroj: vlastní)

Po vložení použitých audio souborů do zvukové banky bylo zapotřebí v kontextovém menu viz Obr. 90 zvukové banky vygenerovat tuto banku, aby tak byla použitelná v REDkitu.

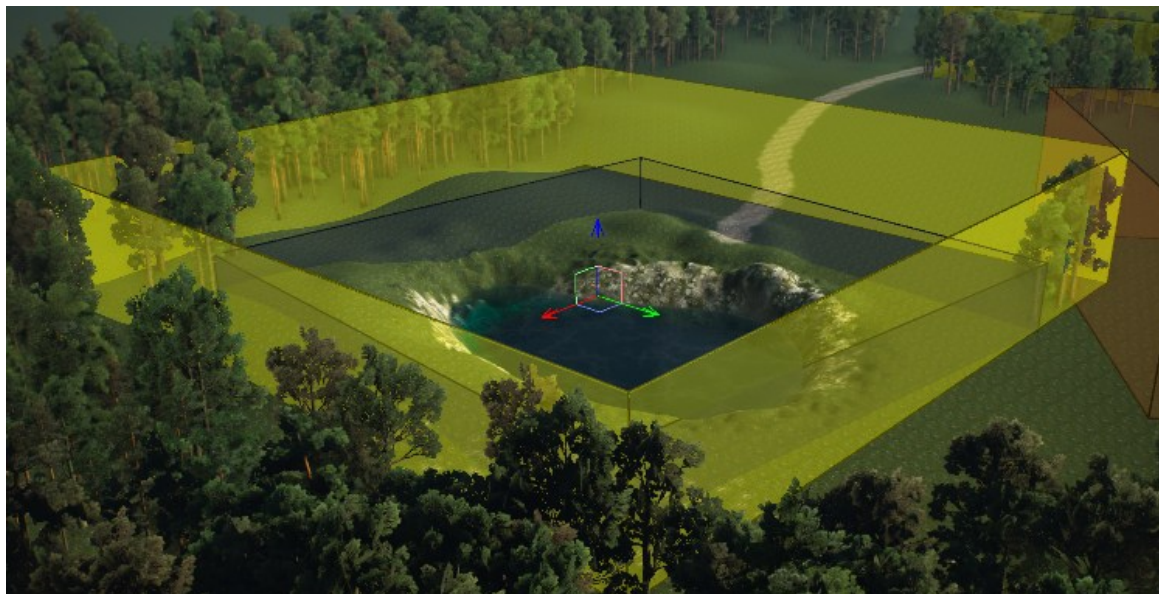


Obr. 90: Kontextové menu v softwaru Wwise pro generování zvukové banky

(zdroj: vlastní)

V REDkitu byl doplněn v záložce *World* parametr *soundBanksDependency* o vytvořenou zvukovou banku z Wwise. Pro použití takto vytvořených zvukových událostí byla vždy přidána nová vrstva s označením *Audio* a typem *AutoStatic*. Pro definování oblastí, ve kterých se spustí zvukové události a začnou přehrávat specifické zvuky a melodie, které byly vytvořeny pomocí Wwise, byla aktivována nově vytvořená vrstva pro zvuky a do scény vloženy instance objektů *CSoundAmbientAreaComponent* kliknutím pravého tlačítka myši a ve vyvolaném kontextovém menu zvolena možnost *Sound – Add Sound Ambient Area*. Tyto oblasti byly následně upraveny velikostně, a také jejich pozice, do vhodných míst světa (například zvuková oblast se zvuky jezera umístěna poblíž jezera v horské oblasti viz Obr. 91). V obou světech (horská oblast a jeskyně) byla zvolena jedna oblast jako ambientní hudba hrající na většině území na pozadí. Ve vlastnostech těchto vrstev byl nastaven jejich parametr *priorityParameterMusic* na *true*. U ostatních oblastí tento parametr zůstal *false*.

Nastavení pro to, která zvuková událost se má spustit a začít přehrávat zvuky, se provedlo pomocí parametru *SoundEvents* a zvolením odpovídající události, která byla vytvořena ve Wwise.



Obr. 91: Ukázka umístění oblasti pro zvukovou událost (žlutá barva) v horské oblasti (zdroj: vlastní)

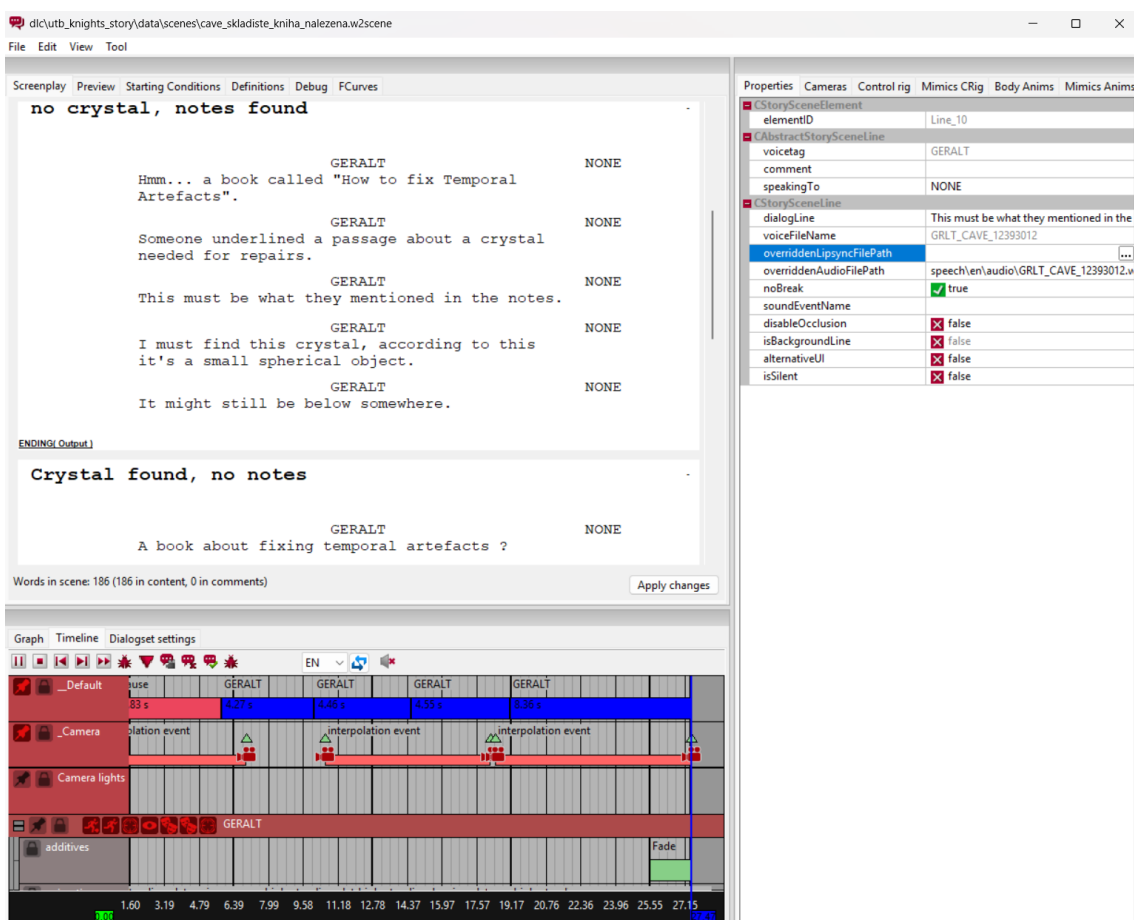
8.2 Tvorba hlasového projevu postav a lipsync animace

8.2.1 Tvorba hlasového projevu

S ohledem na příběhové scény a dialogy vytvořené v rámci této práce vznikla potřeba doplnit modifikaci také o odpovídající hlasový obsah. Pro každou větu bylo nutné připravit samostatný zvukový soubor, který byl následně přiřazen ke konkrétní větě scénáře v editoru scén. Podle doporučení tvůrců REDkitu byl pro tento účel použit formát souboru WAV.

Pro zkvalitnění výsledného hlasu postavy kováře a Geralta, a také prozkoumání možnosti využití umělé inteligence v souvislosti s dabingem herních scén, bylo využito nástroje Voice Changer od ElevenLabs [47]. Tento nástroj umožňuje uživateli vytvořit hlasové profily podle zadaného promptu. Pro účel této práce byly vytvořeny dva hlasové profily, konkrétně jeden pro postavu kováře a jeden pro postavu Geralta. Znění promptů pro vygenerování obou hlasových profilů bylo uvedeno v příloze A.3.

Pracovní postup spočíval v pořízení zvukového záznamu konkrétní věty ze scénáře každé scény, jejím následném zpracování v nástroji Voice Changer s využitím odpovídajícího hlasového profilu a exportu výsledného souboru ve formátu WAV. Takto vytvořený zvukový soubor byl následně přiřazen k odpovídající větě v editoru scén.

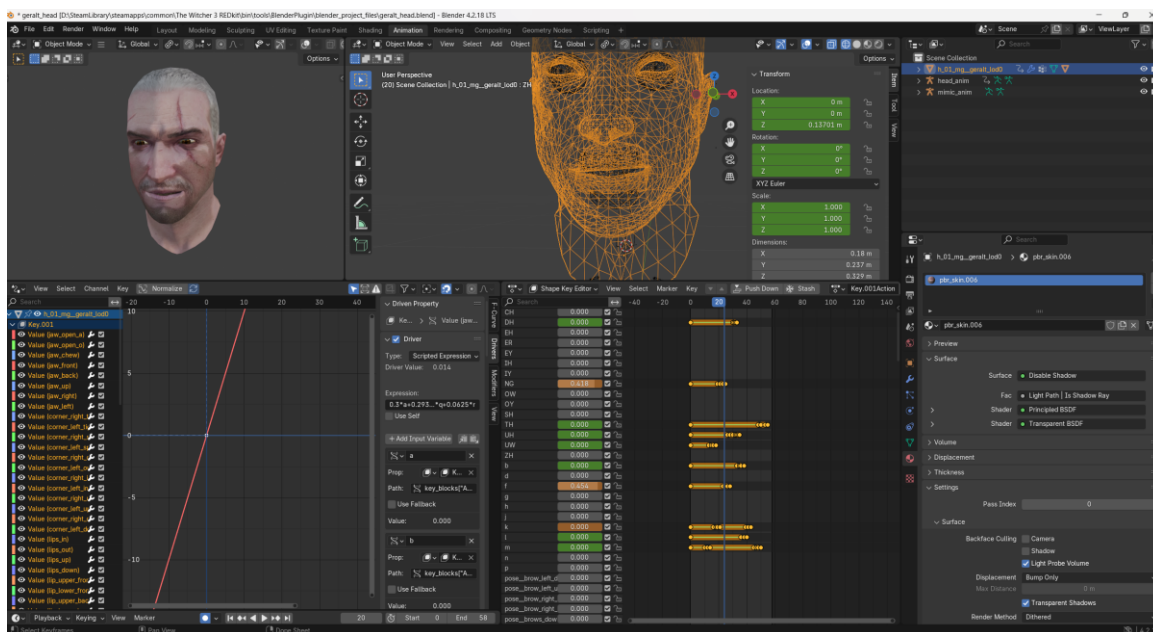


Obr. 92: Scene Editor zobrazující záložku scénáře (vlevo nahoře), časovou osu (vlevo dole) a vlastnosti konkrétní věty scénáře (vpravo)
(zdroj: vlastní)

Tímto způsobem byl postupně vytvořen hlasový obsah pro jednotlivé dialogové věty obou hlavních postav vystupujících ve scénách této modifikace. Zvolený postup umožnil zachovat jednotný charakter hlasového projevu obou postav napříč celou modifikací. Současně bylo ověřeno praktické využití nástroje založeného na umělé inteligenci při přípravě dabingu pro herní obsah.

8.2.2 Tvorba lipsync animací

Pro vytvoření hlasového obsahu bylo nutné zajistit také odpovídající animaci obličeje postav, aby pohyb úst a mimika odpovídaly přehrávaným dialogovým scénám. Tvůrci REDkitu poskytují připravený Blender projekt viz Obr. 93, který je nachystaný pro automatické generování lipsync animace po načtení zvukového souboru WAV.



Obr. 93: Blender projekt zobrazující připravený projekt pro automatické generování lipsync animací ze zvukového WAV souboru (zdroj: vlastní)

Prvním krokem pro úspěšné použití tohoto projektu bylo mít aktivovaný add-on *Red Engine animations importer exporter*, který je poskytnut přímo s REDkitem. Po úspěšné aktivaci byl spuštěn projekt *geralt_head.blend*, který je také poskytnut přímo v adresáři REDkitu. Po načtení projektu byla pro každý zvukový soubor – již změněný nástrojem Voice Changer – vygenerována lipsync animace pomocí funkce *Generate mimic animation from .wav file*, která v Blender projektu viz Obr. 93 byla nalezena v kontextovém menu Scene Collection. Po vygenerování lipsync animace byly exportovány z Blender jako .re animační soubory.

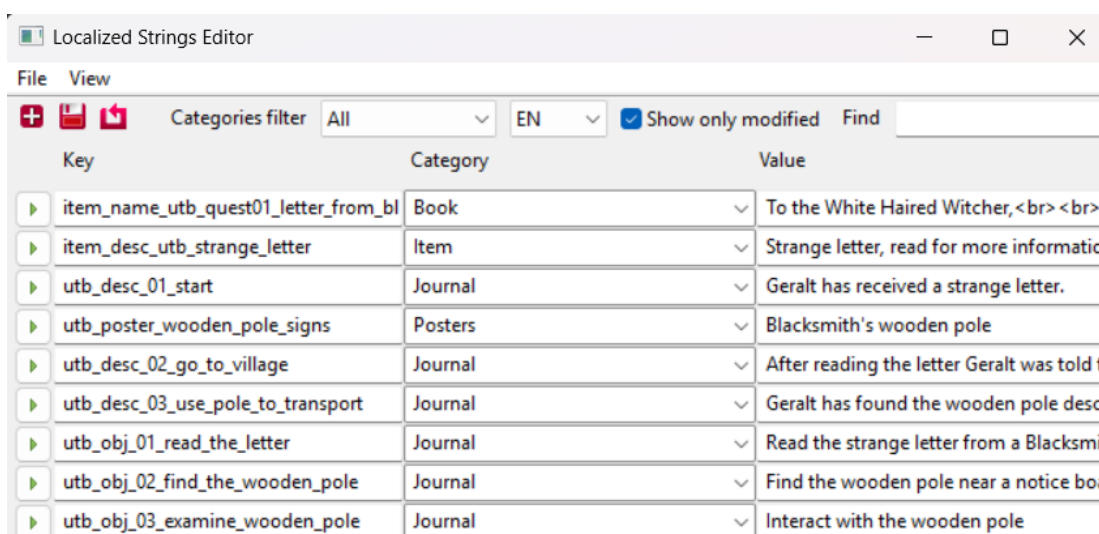
Takto exportované animační soubory byly následně přiřazeny obdobně jako zvukové soubory pro každou větu scénáře.

Celkem bylo vytvořeno 120 hlasových záznamů a stejná počet lipsync animačních souborů. Tento postup umožnil doplnit vytvořené dialogové scény o synchronizovaný hlasový a mimický projev postav.

9 Tvorba dějové linie a skriptování

9.1 Realizace aktivace úkolu

Aktivace úkolu byla realizovaná tak, že po stažení a instalaci modu má hráč po spuštění hry v inventáři automaticky dopis od kováře s dalšími instrukcemi, a zároveň se aktualizuje seznam úkolů o tento nový úkol. Bylo nutné vytvořit nový předmět (dopis od kováře) přidáním záznamu do xml souboru `utb_items`, který byl vytvořen pro účely tohoto modu. Díky tomu se soubor objeví hráči v inventáři automaticky po aktivaci modu. Záznam tomuto dopisu přiděluje jeho název, popis a ikonu v inventáři hráče. V souboru byl přidána část kódu pro item s názvem „`utb_quest01_letter_from_blacksmith`“ viz Výpis 1.



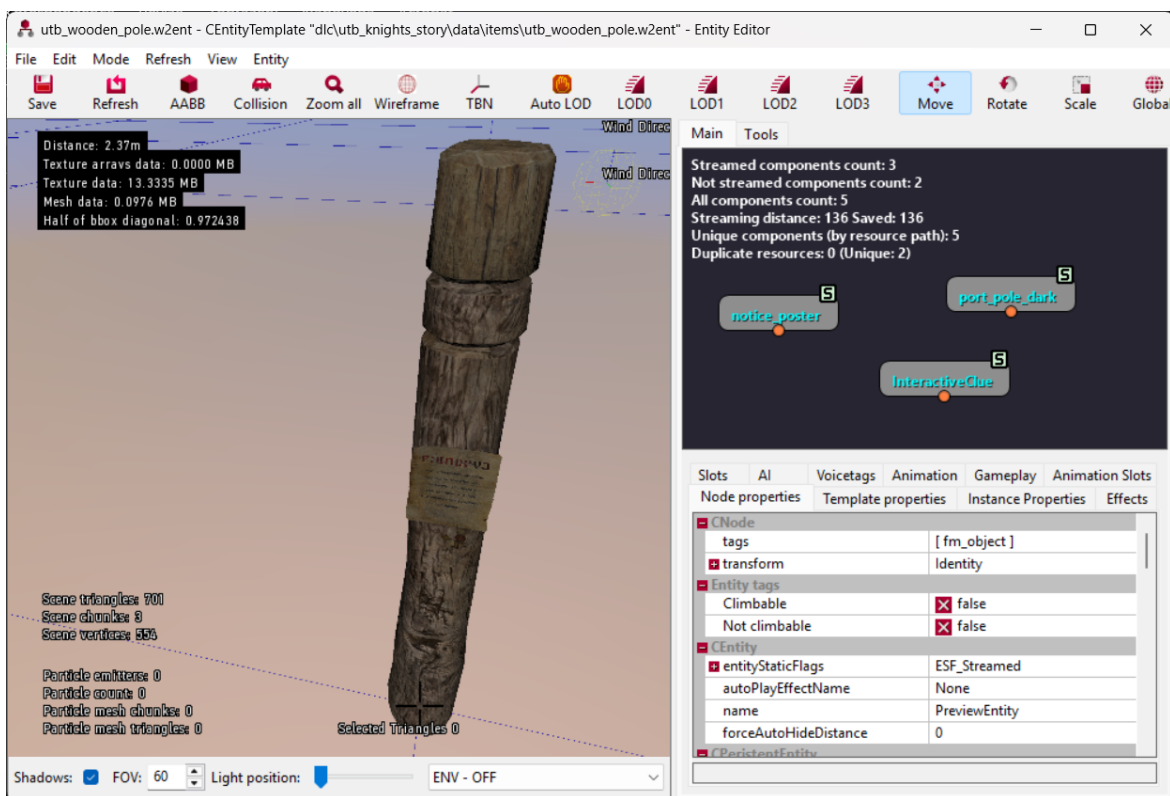
The screenshot shows the 'Localized Strings Editor' window. It has a menu bar with 'File' and 'View'. Below the menu bar are icons for adding, deleting, and undoing, followed by a 'Categories filter' dropdown set to 'All', a language dropdown set to 'EN', a 'Show only modified' checkbox which is checked, and a 'Find' search box. The main area contains a table with three columns: 'Key', 'Category', and 'Value'. The table lists several localization entries for the game mod.

Key	Category	Value
▶ item_name_utb_quest01_letter_from_bl	Book	To the White Haired Witcher,
▶ item_desc_utb_strange_letter	Item	Strange letter, read for more informatio
▶ utb_desc_01_start	Journal	Geralt has received a strange letter.
▶ utb_poster_wooden_pole_signs	Posters	Blacksmith's wooden pole
▶ utb_desc_02_go_to_village	Journal	After reading the letter Geralt was told t
▶ utb_desc_03_use_pole_to_transport	Journal	Geralt has found the wooden pole desc
▶ utb_obj_01_read_the_letter	Journal	Read the strange letter from a Blacksmi
▶ utb_obj_02_find_the_wooden_pole	Journal	Find the wooden pole near a notice boz
▶ utb_obj_03_examine_wooden_pole	Journal	Interact with the wooden pole

Obr. 94: Localized Strings Editor zobrazující řetězce použité v úvodní části úkolu modifikace (zdroj: vlastní)

Pomocí nástroje *Localized Strings Editor* byly vloženy řetězce pro popis dopisu od kováře a jeho text v inventáři a řetězce použité pro tuto část úkolu viz Obr. 94. V textu dopisu kovář hráči (postavě Geralta) nabízí práci, a poskytuje instrukce jako postupovat dále, respektive nabádá hráče, aby našel dřevěný sloup v nedaleké vesnici v oblasti White Orchard poblíž informační tabule.

V Asset Browseru byl na cestě: *virtualdepot – dlc* vytvořen nový adresář pro uchování specifických souborů a dat pro tento mod. V tomto adresáři byl následně vytvořen nový soubor pro šablonu entity reprezentující dřevěný sloup viz Obr. 95, uvedený v dopise od kováře. Instance této entity byla použita jako interaktivní entita pro hráče sloužící k teleportaci postavy hráče z originální hry do světa vytvořeného tímto modem.



Obr. 95: Šablona entity modifikovaného dřevěného sloupu otevřená v Entity Editoru
(zdroj: vlastní)

Instance sloupu byla umístěna v originální oblasti White Orchard do vesnice poblíž informační tabule viz Obr. 97. Ve vlastnostech u místěného sloupu byly nastaveny atributy sloupu umožňující interakci na *false* viz Obr. 96 tak, aby hráč se sloupem nemohl interagovat, dokud nepostoupí v úkolu.

Kolem sloupu byla přidána spouštěcí oblast (anglicky *Trigger Area*) pro využití ve skriptu úkolu v rámci podčásti úkolu objevit tento sloup.

W3MonsterClue	
isAvailable	<input type="checkbox"/> false
isInteractive	<input type="checkbox"/> false
isReusable	<input type="checkbox"/> false
isVisible	<input type="checkbox"/> false
isIgnoringFM	<input checked="" type="checkbox"/> true
playerVoiceset	EPVT_None

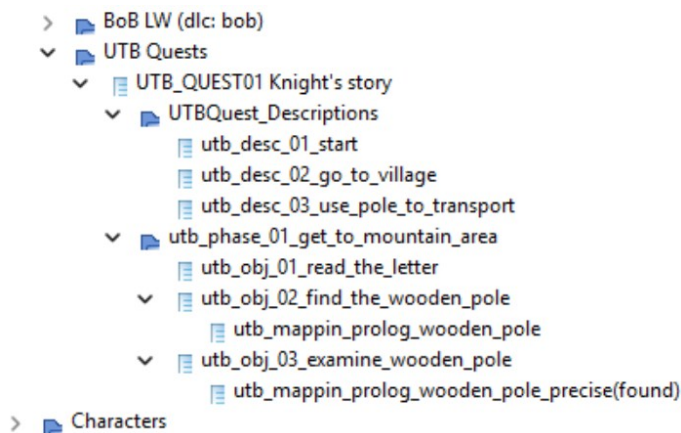
Obr. 96: Detail nastavení vlastností dřevěného sloupu (zakázaná interakce)
(zdroj: vlastní)



Obr. 97: Pohled na umístěný modifikovaný dřevěný sloup poblíž informační tabule sloužící jako teleport

(zdroj: vlastní)

Pro využití v grafu a skriptech byly vytvořeny nové záznamy pomocí *Journal Editoru* využívající řetězců vytvořených pomocí nástroje *Localized Strings Editor*. V hlavní složce *Quests* byla vytvořena nová skupina úkolů viz Obr. 98 (možnost *Create new quest group*). U vhodných položek, kde je užitečná navigace hráče k určitému bodu, nebo oblasti v mapě, byly přidány *mappin* položky odkazující se na konkrétní objekty vytvořené přímo ve světě.



Obr. 98: Ukázka struktury úkolu, jeho popisků a mappin položek v nástroji Journal Editor

(zdroj: vlastní)

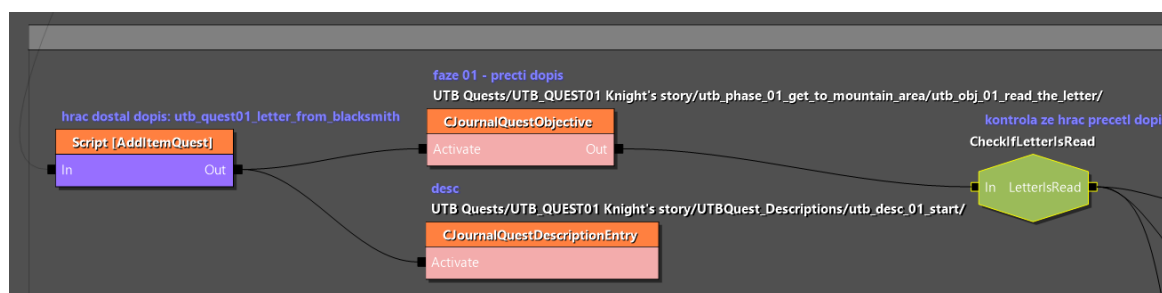
Spouštěcí okamžik pro modifikovaný obsah, a tedy zahájení úkolu, nastane automaticky po instalaci modifikace a spuštění nové hry. Pro tento účel byl vytvořen nový úkolový soubor obsahující veškeré části úkolu rozdělené do jednotlivých fází.



Obr. 99: Quest Editor nástroj zobrazující úkolový soubor tohoto modu rozdělený do jednotlivých fází (uzly s názvem *Phase*)
(zdroj: vlastní)

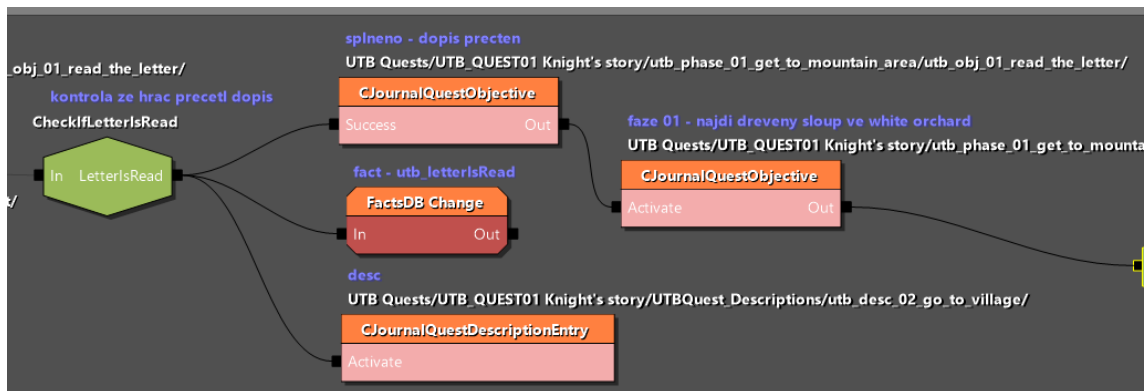
V místě počátečního uzlu (červená šipka Start vlevo viz Obr. 99) byl napojen nový uzel typu *CJournalQuest*, který je v rámci modu vytvořeného touto prací použit k aktivaci úkolu a jeho podčásti, přidání popisu do deníku, nebo splnění části úkolu. Další uzly byly použity pro rozdělení logiky do jednotlivých fází pomocí uzlů typu *Phase*. První fáze byla použita k řešení získání úkolového dopisu od kováře hráčem, jeho přečtení a nalezení dřevěného sloupu. Po interakci se sloupem dojde k teleportu do horské oblasti. Získání dopisu bylo vyřešeno uzlem typu *Script* využívající funkci *AddItemQuest* viz Obr. 100 s nastaveným parametrem *itemName* na název dopisu vytvořeného touto modifikací, tímto se přidal do inventáře hráče předmět dopisu od kováře.

Za uzel přidávající Geraltovi do inventáře dopis byly následně přidány uzly typu *Journal – Quest* pro zahájení úkolu a jeho aktualizaci popisu v deníku viz Obr. 98 – tedy zobrazení informací postavě Geralta, aby si přečetl dopis. Pomocí jejich parametru *questEntry* byly vybrány odpovídající hodnoty vytvořené v nástroji *Journal Editor*.



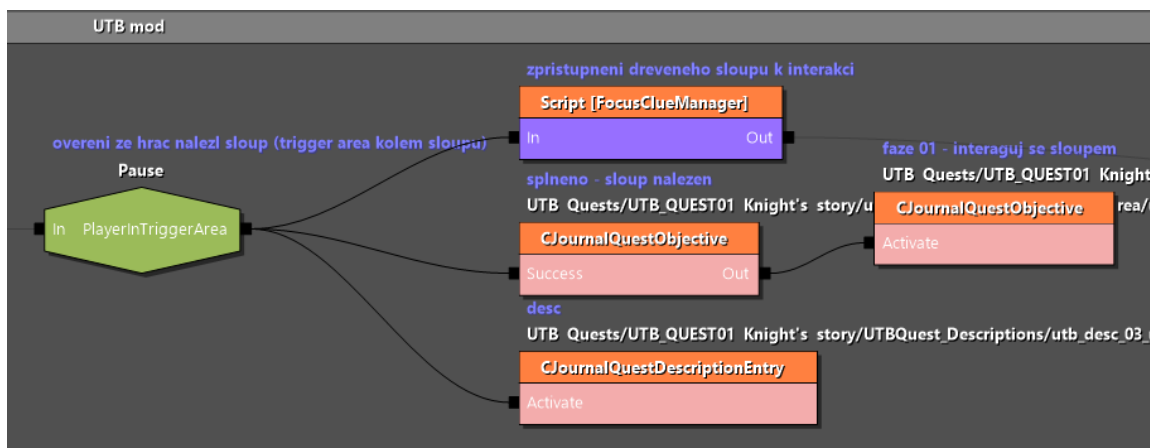
Obr. 100: Quest Editor nástroj zobrazující graf v místě začátku modifikace (uzel *Script* nalevo)
(zdroj: vlastní)

Po přečtení dopisu dojde k události díky uzlu viz Obr. 100 (zelený uzel vpravo), kdy tento uzel typu *Pause* využívá podmínky *W3QuestCond_BookHasBeenRead* s hodnotou parametru nastavenou opět na dopis od kováře, která umožní průchod uzlem pouze po události přečtení tohoto dopisu.



Obr. 101: Quest Editor zobrazující graf řešící logiku splnění části úkolu přečtení dopisu a zahájení části úkolu nalezení dřevěného sloupu pro transport do horské oblasti (zdroj: vlastní)

Za uzel kontrolující přečtení dopisu byly napojeny opět uzly pro aktualizaci úkolu a popisu v deníku, tak, aby byl hráč informován o další části úkolu – nalézt dřevěný sloup. V rámci uzlů byl nejprve splněn úkol přečtení dopisu napojením z uzlu kontrolující přečtení dopisu na vstupní slot *Success* uzlu typu *Journal* a z tohoto uzlu následně napojen na další uzel typu *Journal* na slot *Activate* zahajující další část úkolu (nalezení dřevěného sloupu) viz Obr. 101.



Obr. 102: Quest Editor zobrazující část grafu řešící splnění části úkolu nalezení sloupu, spuštění skriptu pro umožnění interakce dřevěného sloupu a aktualizace údajů k úkolu (zdroj: vlastní)

Pro kontrolu, zda hráč objevil sloup, byl vložen další uzel typu *Pause* využívající podmínky *CQuestInsideTriggerCondition* s nastavenou hodnotou parametru *triggerTag* na název spouštěcí oblasti vytvořené kolem dřevěného sloupu. Po průchodu tímto uzlem byly opět přidány uzly k aktualizaci úkolu a jeho popisku v deníku. Dále byl přidán uzel typu *Script* používající funkci *FocusClueManager*, které byl nastaven *tag* objektu dřevěného sloupu. Funkce byla použita ke změně parametrů tak, aby byla umožněna interakce hráče se sloupem (nutná pro transport hráče do horské oblasti) viz Obr. 103.

CQuestGraphBlock	
name	Script
comment	zprístupneni dreveneho sloupu k interakci
forceKeepLoadingScreen	<input checked="" type="checkbox"/> false
guid	7A092E0E-4950CAF7-90FEABA0-2139D063
CQuestScriptBlock	
functionName	FocusClueManager
saveMode	QSCSM_SaveBlocker
Dynamic Properties	
tag	utb_wooden_pole_white_orchard
actionType	FCAA_ForceSet
isAvailable	<input checked="" type="checkbox"/> true
isInteractive	<input checked="" type="checkbox"/> true
isReusable	<input checked="" type="checkbox"/> true
isVisible	<input checked="" type="checkbox"/> true
wasDetected	<input checked="" type="checkbox"/> false
isIgnoringFM	<input checked="" type="checkbox"/> true

Obr. 103: Detail nastavení parametrů skriptového uzlu řešící umožnění interakce dřevěného sloupu (zdroj: vlastní)

Pro kontrolu interakce s dřevěným sloupem byl přidán uzel *Pause* kontrolující, zda existuje v databázi faktů (anglicky *FactsDB*) fakt označující, že hráč objevil dřevěný sloup. Pro tento účel byl použit fakt, který se v databázi objeví pouze při interakci hráče se sloupem.



Obr. 104: Quest Editor zobrazující část grafu řešící interakci se sloupem a teleport do horské oblasti (zdroj: vlastní)

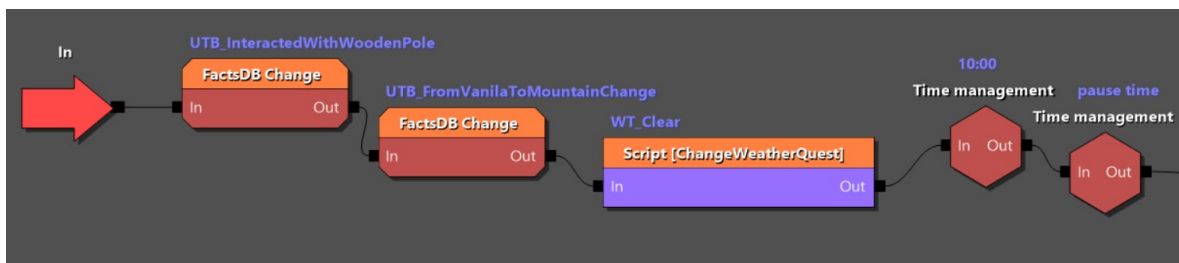
Další uzly typu *Quest* byly nastaveny pro aktualizaci úkolů:

- splnění úkolu interakce se sloupem
- přidání popisu do deníku
- aktivace úkolu nalezení chaty s kovářem.

Po aktualizaci byla tato fáze ukončena uzlem *Out*.

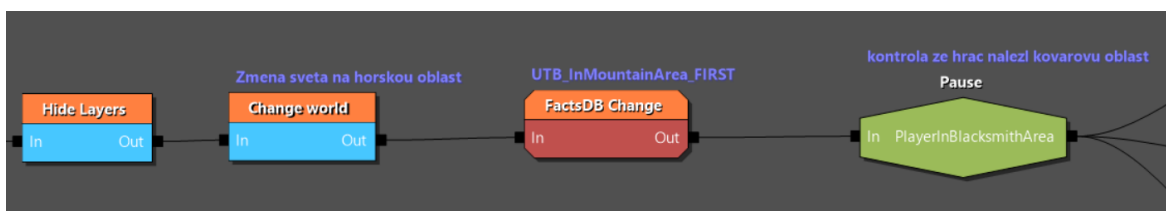
9.2 Nalezení kováře a interakce s ním

Na začátku další fáze viz Obr. 105 byly přidány uzly typu *FactsDB Change* pro uložení faktů o proběhnuté interakci se sloupem a cestě z originální hry do horské oblasti. Další uzly byly nastaveny na změnu počasí a času pomocí uzlů typu *Script* a *Time Management*.



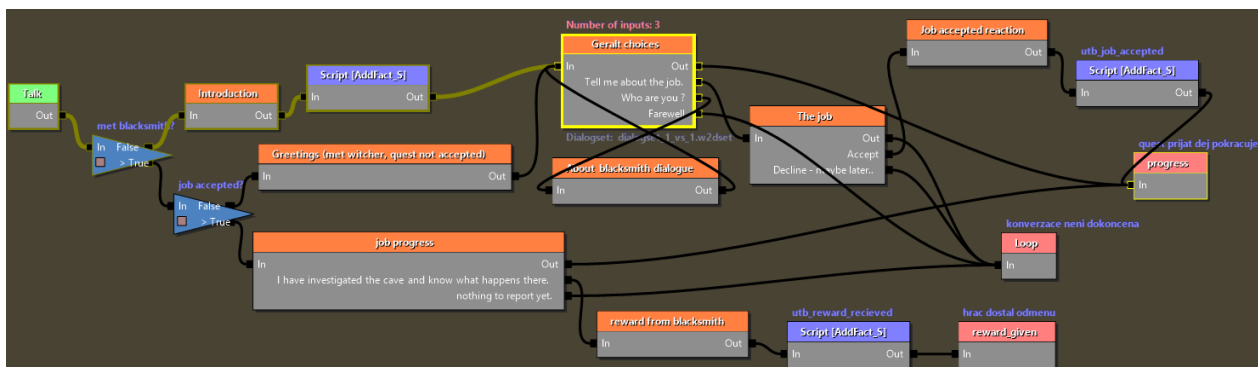
Obr. 105: Quest Editor zobrazující začátek druhé fáze řešící úvodní nastavení počasí a času (zdroj: vlastní)

Důležitý uzel pro změnu světa na horskou oblast je uzel typu *Change world* s nastaveným světem horské oblasti a určením místa, kde se má hráč objevit. Další uzel typu *Pause* byl použit pro kontrolu, zda se hráč nachází v oblasti kolem kovářovy chaty (tedy podmínka splnění této části úkolu) viz Obr. 106.



Obr. 106: Quest Editor druhé fáze řešící změnu světa a kontrolu, zda hráč našel kováře (zdroj: vlastní)

Nalezením oblasti kováře byla umožněna hráči interakce s postavou kováře. Interakce byla realizována spuštěním scény s dialogem postavy Geralta a kováře. Během dialogu kovář podá potřebnou expozici příběhu a nabídku k přijetí jeho úkolu. Pro scénu byl vytvořen soubor obsahující nastavení scény, kamery, volby úkol od kováře přijmout, nebo prozatím děj pozastavit s možností se k dialogu vrátit. Ukázka grafu viz Obr. 107.



Obr. 107: Ukázka grafu scény řešící dialog kováře s hráčem
(zdroj: vlastní)

Logika scény byla nastavena tak, aby se po přijetí úkolu od kováře pomocí skriptu uložil do databáze faktů údaj o přijetí úkolu. V nástroji Quest Editor je tento fakt sledován a po jeho existenci bylo umožněno pokračovat v ději a aktualizaci úkolů na nalezení vstupu do jeskyně. Další scény bylo využito při vstupu do jeskyně, kde je scéna využita pouze s postavou hráče a je mu umožněno buď vstoupit dovnitř, čímž dojde k teleportu do další lokace – Jeskyně, nebo zůstat před jeskyní a hráč může pokračovat v prozkoumání horské oblasti. Ukázka z této scény viz Obr. 108. Pokud se hráč rozhodne vstoupit do jeskyně, byla sekvence uzlů a skriptů nastavena obdobně jako v případě přechodu mezi světem originální hry a horskou oblastí.



Obr. 108: Ukázka z testování scény u vstupu do jeskyně s nabídkou vstoupit do jeskyně, nebo odejít
(zdroj: vlastní)

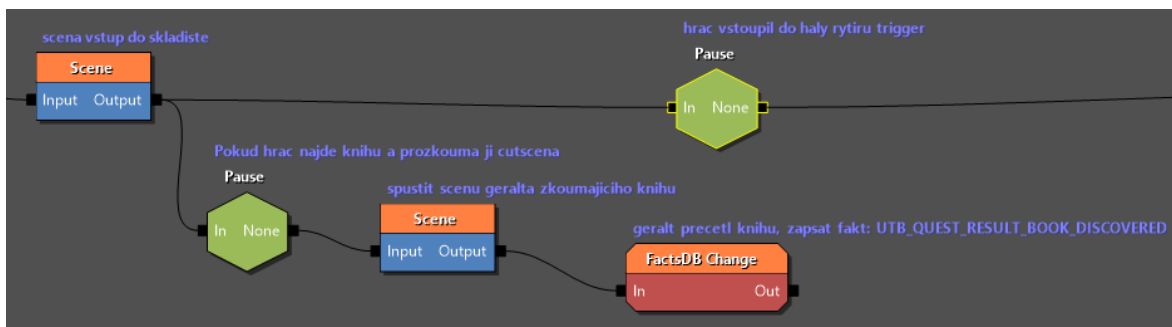
9.3 Vyřešení příběhu v jeskyni



Obr. 109: Připravené druhé setkání s nepřáteli v hlavní chodbě
(zdroj: vlastní)

Pro řešení příběhu v jeskyni byla vytvořena další fáze pro hlavní úkol. Hráč se ocitne v horní části jeskynní oblasti, která vypadá jako prostá přírodní jeskyně. Poblíž vstupu bylo hráči připraveno setkání (anglicky *encounter*) pomocí spouštěcí oblasti, která je sledována úkolovým grafem v nástroji Quest Editor. V okamžiku, kdy hráč vstoupí do oblasti, bylo setkáním nastaveno zrození nepřátel rozmístěných po této části jeskyně a hráč je dostane za úkol zabít. Tímto byla hráči předána dějová informace, že v jeskyni, a tím pádem i s rytíři, pravděpodobně není něco v pořádku – po splnění zabití nepřátel Geralt ve hře podotkne tuto skutečnost. Obdobné setkání s nepřáteli bylo vytvořeno v hlavní chodbě.

Důležitá událost pro dějovou linku byla vytvořena v místnosti skladiště, kde byl na stůl umístěn interaktivní předmět (dále kniha), která byla vytvořena pomocí Entity Editoru. Kniha je volitelná k nalezení – pokud knihu hráč nalezne a interaguje s ní pomocí zaklínačských smyslů, dojde ke scéně, při které postava Geralta tuto knihu zkoumá, dozvídá se informace o opravě neznámého artefaktu a potřebném krystalu na jeho opravu. Do databáze faktů je uložen fakt, že hráč našel tuto knihu, která může dále ovlivnit výsledek příběhu. Logika řešící tento případ byla vytvořena v nástroji Quest Editor pomocí dvou paralelních uzlů viz Obr. 110, kde spodní větev bude spuštěna pouze pokud hráč interaguje s knihou a horní větev pokračuje v ději dále. V rámci scény bylo vytvořeno více variant scény, podle toho, zda hráč před interakcí již nevyřešil situaci ve scéně. Geralt podle těchto okolností reaguje na nález knihy jinými způsoby.



Obr. 110: Quest Editor graf řešící paralelní děj: spodní větev sledující událost nalezení knihy, horní větev pokračuje v ději
(zdroj: vlastní)

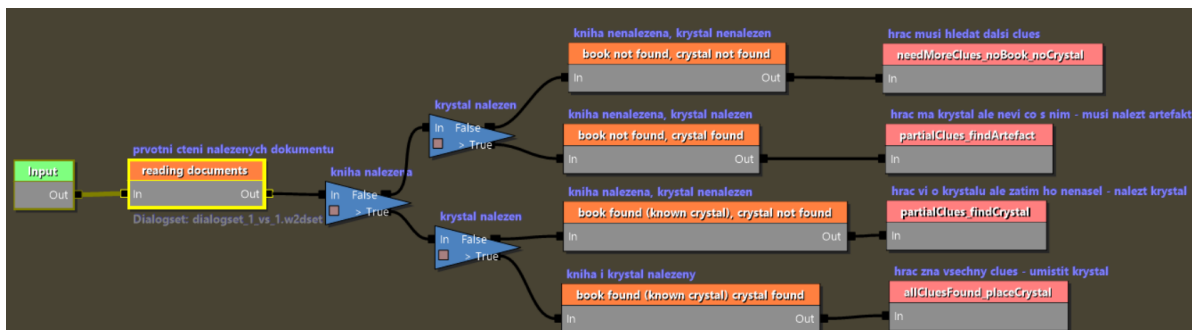
Hráč postupným propátráním jeskyně nalezne poslední hlavní místnost s rytíři. Pro tuto místnost byli vytvořeni rytíři, respektive kostlivci v různém oděvu. Postup tvorby je popsán v kapitole 7.4. Několik instancí entit těchto rytířů bylo vloženo do místnosti, aby působili dojmem, že ve svém čekání u stolů (případně spící v posteli) se z nich stali kostlivci.



Obr. 111: Pohled na umístěné kostlivce rytířů a poznámky na stole
(zdroj: vlastní)

Na stůl k postavě jednoho z kostlivců byly umístěny interaktivní poznámky viz Obr. 111 (kostlivec vpravo). Tyto poznámky byly vytvořeny jako volitelný předmět pro úkol, obdobně jako kniha v místnosti skladiště. Pokud hráč interaguje s poznámkami, byla vytvořena další scéna, která se v tento okamžik spustí. Postava Geralta z poznámek zjistí, že rytíři měli problémy s časovým polem (anglicky *temporal field*) kvůli rozbité části artefaktu v zadní místnosti haly. Pro tuto scénu bylo vytvořeno více variant viz Obr. 112 (graf scény

nalezení poznámek), které se spustí v závislosti na určitých faktech uložených v databázi faktů. Pokud již hráč našel krystal na opravu, knihu, nebo obojí zároveň, bude Geraltova reakce na poznámky pokaždé jiná.



Obr. 112: Ukázka větvení v grafu scény nalezení poznámek rytíře

(zdroj: vlastní)

Krystal, který hráč potřebuje k vyřešení části úkolu v jeskyni, byl schován v místnosti do dekorace vázy viz Obr. 113. Krystal byl vytvořen obdobně jako kniha a poznámky. Při interakci s krystalem Geralt poznamená, bez narušení hry, že našel krystal.



Obr. 113: Pohled na krystal důležitý pro úkol umístěný do dekorace vázy

(zdroj: vlastní)

Poslední důležitý objekt je artefakt samotný, který je hráči zmíněn v poznámkách a v knize. Artefakt byl vytvořen podobně jako krystal a umístěn do zadní části místnosti. Při interakci se podle toho, co hráč již našel, spustí odpovídající scéna.

Cílem hráče je nalézt krystal, alespoň jedno z vodítek (kniha, nebo poznámky) a interagovat s artefaktem. Pokud splní tyto požadavky, bude mu při interakci nabídnuta volba rozhodnutí. Buď krystal použije, aby jej umístil do artefaktu a tím jej opravil (dobrácká konstruktivní volba), nebo se rozhodne si jej ponechat (sobecká volba). Pro obě volby se spustí odpovídající scéna. Pro enviromentální výklad příběhu bylo využito kostlivců rozmístěných v prostorách rytířské haly – pokud hráč zvolí opravu artefaktu (umístění krystalu), tak z místnosti zmizí kostlivci rytířů a obdobně díky této volbě při průjezdu v horské oblasti polem u hory již hráč neuslyší zvuky bubnů a polnice (trubky). Pokud si Geralt krystal ponechá, v prostředí se nic nezmění.

Ven z jeskyně se hráč po volbě u artefaktu dostane stejnou cestou jako přišel. U východu z jeskyně pak byla nachystána interakce, kterou se hráč teleportuje ven z jeskyně zpátky na úpatí hory.



Obr. 114: Pohled na opravený artefakt osvětlující okolí svou září
(zdroj: vlastní)

9.4 Uzavření příběhu a odměna

Podle rozhodnutí hráče u artefaktu byla nastavena dějová linie tak, aby se větvila – buď na úkol donést nalezený krystal kovářovi, nebo mu říct, co Geralt zjistil a co s krystalem udělal. Podle aktivního úkolu se při interakci s postavou kováře spustí odpovídající scéna, ve které Geralt předává informace a případně krystal.

Odměna pro hráče je závislá na volbě hráče u artefaktu – pokud hráč použil krystal na opravu artefaktu, obdrží odměnu v podobě zkušeností a nové zbraně viz Obr. 115, která byla vytvořena pro účely této modifikace. Postup vytvoření je popsán v kapitole 7.5. Pokud hráč provedl rozhodnutí si ponechat krystal a předat ho kovářovi, obdrží pouze zkušenosti a finanční obnos herní měny.



Obr. 115: Pohled na postavu Geralta používající meč, který obdržel jako odměnu za úkol z tohoto modu

(zdroj: vlastní)

Po obdržení odměny se spustí závěrečná scéna ukazující Geralta odjíždějícího na koni a hráč je teleportován zpět do originálního světa hry k dřevěnému sloupu, který hráče přenesl do světa této modifikace.

Pro definici odměn bylo potřeba vytvořit nový soubor `utb_rewards.xml`, kód definující obě odměny je přiložen v příloze A.2 viz Výpis 2.

10 Publikace a problémy

10.1 Proces publikace modu na platformu Steam

Posledním krokem modu s využitím nástroje REDkit byla publikace na vybranou platformu. Pro účely této práce byla vybrána platforma Steam. Během procesu publikace bylo zapotřebí použít několikrát proces přípravy dat (anglicky *cook*) pro publikaci, pro který REDkit nabízí jednotlivé funkce, které byly použity následovně:

- V horské oblasti i jeskyni:
 - Cook Occlusion for All Tiles – předzpracování dat viditelnosti pro jednotlivé části světa, aby hra správně rozhodovala, které objekty mají být během běhu vykresleny.
 - Generate Navigation Data – generování navigačních dat (navmesh) pro pohyb postav, hledání tras, které mohou použít, (anglicky *pathfinding*) a správné chování postav ve světě.
- Nezávisle na světě byly poté provedeny jednorázově pro celý mod:
 - Cook Mod Strings – příprava textových a lokalizačních řetězců modu pro výslednou publikovanou verzi.
 - Cook Mod Sounds – příprava zvukových dat modu pro publikaci. Zde bylo umožněno připravit zvuky vytvořené pro tento mod pomocí Wwise.

S takto připravenými soubory bylo využito možnosti *Save and publish mod project*, která otevřela okno s průvodcem publikace modu. V první záložce tohoto okna byl zvolen název modu pro platformu, jeho verze a popis. V druhé záložce byla provedena další příprava dat, nyní už celého modu pro publikaci (proces *cooking*). Třetí záložka byla použita pro vytvoření ZIP souboru celého modu a jeho konečné zveřejnění na platformu Steam.

Celý tento proces se iterativně opakoval pro aktualizaci postupně přidávaných změn a jejich testování v publikované verzi. Během tvorby této práce a modu byl aktualizován celkem 36krát.

10.2 Zpětná vazba z publikovaného modu

Mod byl zpřístupněn veřejnosti na platformě Steam ke stažení dne 9.4.2026 a od tohoto data byl sledován pro zpětnou vazbu po dobu přibližně tří týdnů. Po uplynutí této doby byly z dostupných dat na platformě Steam k datu 29.4.2026 zjištěny následující skutečnosti:

- 2514 osob si prohlédlo mod
- 45 osob má mod stažený a nainstalovaný
- 12 osob má mod přidán jako oblíbený
- 5 pozitivních komentářů
- 80 hodnocení (78 pozitivních, 2 negativní)
- Celkové hodnocení 4 z 5 hvězd

10.3 Problémy

Zatímco při testování v REDkitu před publikací bylo vše v pořádku a podle očekávání, v publikované verzi, tj. ve spuštěné originální hře Zaklínač 3 s aktivovaným modem, byly zpozorovány problémy.



Obr. 116: Ukázka špatné animace pro koně v publikované verzi modu (zdroj: vlastní)

Jedním z problémů byla chybná animace a kolize pro koně i postavu Geralta. Kůň se po zavolání hráčem objevil ve světě poblíž postavy hráče, tak, jak by správně měl, ale ihned bylo zřejmé, že jeho animace je špatná a při jeho pohybu docházelo k problémům viz Obr. 116.

Dalším problémem bylo „zasekávání“ se postavy Geralta (pěšky / na koni) v objektech rozmístěných po světě, nebo v terénu. Oba tyto problémy (animace koně a zasekávání) byly vyřešeny zvětšením mapy horské oblasti přibližně na dvojnásobek a znovu vygenerováním navigačního meshe. Pravděpodobným problémem byl malý svět pro správný výpočet navigačního meshe.

V jeskynní oblasti v publikované verzi došlo ke špatné pozici zrození nepřátel v hlavní chodbě viz Obr. 117. Nepřátelé, zrození ve vstupní části jeskyně, byli v pořádku. Pro nápravu správného zrození nepřátel v hlavní chodbě bylo zapotřebí přidat novou vrstvu do jeskynního světa a přidat objekt pomocí kontextového menu ve viewportu scény *Navigation – Add Navmesh Instance*. Pro tuto instanci navigačního meshe bylo nutné nastavit rozměr na potřebnou plochu tak, aby obsáhla problémovou oblast a pomocí kontextového menu v panelu scény zvolit pro tuto instanci možnost *Generate navmesh*. Tímto se vytvořil lokální navigační mesh – při následné opětovné publikaci se problém odstranil.



Obr. 117: Špatně zrozený nepřítel pod terénem (bílé kruhy uprostřed označující jeho polohu)
(zdroj: vlastní)

Dalším problémem byly chybějící (průhledné) některé materiály, textury atd. Tento problém je popsán jako jediný v oficiální dokumentaci REDkitu [48], kde se uvádí, že řešení by mohlo být provedení tzv. *Checkout* na všechny instance použitých materiálů (soubory s koncovkou *w2mi*) pro problémový mesh. Toto řešení na tento mod nefungovalo. Problémové meshe byly nahrazeny jinými, které fungovaly bez problému. Stejný problém nastal s vegetací, respektive trávou, která byla vygenerovaná pomocí masky pro trávu popsané v kapitole 3.5.2. Tráva se při testování v REDkitu vykreslovala normálně, v publikované verzi zcela chyběla. Byla proto použita metoda manuálního kreslení popsaná v téže kapitole (*Paint*). Po opětovném publikování se tento problém již nevyskytl.

Další problém publikované verze byla absence všech zvuků vytvořených touto modifikací. Problém byl vyřešen přidáním mezikroku do procesu publikace. Bylo nutné zavřít publikování po provedení druhé záložky procesu publikace (*Cook*), a zopakovat možnost *Cook mod sounds* pomocí nabídky z horního panelu *Publish*. Tento krok generuje soubor *sound-spc.cache* ve složce *packed*, ale z nějakého důvodu se během procesu publikování smaže a je potřeba jej takto znovu vygenerovat. Po opětovném vygenerování stačí znovu spustit proces publikace – zde je nutné přeskočit krok na druhé záložce pomocí tlačítka *Skip Cooking*. Na třetí záložce již lze pokračovat vytvořením ZIP souboru a publikací jako obvykle. Tímto se přidaly zvuky do publikovaného modu a fungují.

Závěr

Cílem této bakalářské práce bylo představit modovací aplikaci REDkit jako nástroj pro tvorbu modifikací počítačové hry Zaklínač 3: Divoký hon a současně ověřit její praktické možnosti na vlastní navrhnuté rozsáhlejší modifikaci. Tento cíl byl naplněn jak v teoretické, tak v praktické rovině.

Teoretická část shrnula problematiku modování počítačových her, popsala hru Zaklínač jako platformu pro modifikace a představila možnosti nástroje REDkit tak, jak je popisuje oficiální dokumentace jejími tvůrci. V souvislosti s popisem použití REDkitu byla představena také role externích aplikací Blender a Wwise při tvorbě obsahu pro hru.

V praktické části byla navržena rozsáhlejší modifikace hry Zaklínač 3 pomocí zcela nové úkolové linie inspirované motivem příběhu blanických rytířů z povídky spisovatele Aloise Jiráska a zasazené do světa hry Zaklínač 3. V rámci modifikace byly vytvořeny dvě nové lokace v podobě horské a jeskynní oblasti, dialogové scény spjaté s navrženým příběhem, skriptovací logika úkolu, zvuková složka a odměna pro hráče. Součástí práce bylo rovněž využití externích nástrojů, zejména Blenderu pro úpravy vybraných 3D modelů a Wwise pro práci se zvukem. Praktická část tak ukázala, že REDkit umožňuje nejen úpravy jednotlivých herních prvků, ale i tvorbu komplexnějšího rozšíření kombinující práci se světem, příběhem, objekty, skripty i zvukem.

Při publikaci modu a jeho testování se zároveň ukázalo, že práce s REDkitem je spojena s problémy. Přestože se modifikace během testování v REDkitu chovala podle očekávání, v publikované verzi se objevily rozdíly, například v oblasti animací, kolizí nebo zvuků. Nalezené problémy bylo po analýze problému a hledání řešení možné odstranit, i přes absenci oficiální dokumentace pro tyto situace. Pro efektivní analýzu možných problémů se ukázalo důležité trpělivé testování a logický návrh struktury vrstev obsahující jednotlivé kategorie herního obsahu při vytváření modifikace a orientace v nich.

Výsledkem práce je funkční ukázka herního rozšíření, která demonstruje praktické možnosti aplikace REDkit při tvorbě vlastního obsahu pro Zaklínače 3 a jeho dokumentace. Práce zároveň potvrzuje, že oficiální modovací nástroj může sloužit nejen komunitní tvorbě, ale i jako vhodný prostředek pro studijní a tvůrčí experimentování v oblasti návrhu herního obsahu. Práce je sama o sobě také návodem v českém jazyce pro použití REDkitu.

Do budoucna by bylo možné modifikaci dále rozšířit například o propracovanější větvení úkolu, další herní lokace, více vlastních modelů, nebo důkladnější optimalizaci publikované verze.

Na základě zkušeností získaných při tvorbě modifikace během této práce lze doporučit postupovat při návrhu nových herních světů iterativně, nejprve vytvořit základní terén a prostorové uspořádání lokace, následně doplnit vegetaci a další prvky prostředí, u důležitých míst dočasně využít zástupné objekty a s každou iterací přidávat detaily. U každé vývojové etapy funkčnost průběžně ověřovat testováním.

Seznam použité literatury

- [1] CD PROJEKT. The Witcher 3 REDkit. Online. 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/overview?homepageId=2850903>. [cit. 2025-10-28].
- [2] CD PROJECT RED. The Witcher 3: Wild Hunt. Online. C2025. Dostupné z: <https://www.thewitcher.com/us/en/witcher3>. [cit. 2025-12-29].
- [3] SCACCHI, Walt. Computer Game Mods, Modders, Modding, and the Mod Scene. Online. First Monday. C1995-2025, vol. 2010, no. 15, article 5, s. 1. ISSN 1396-0466. Dostupné z: <https://firstmonday.org/ojs/index.php/fm/article/view/2965>. [cit. 2025-12-04].
- [4] BLACK TREE GAMING LTD. NEXUSMODS. Online. C2025. Dostupné z: <https://www.nexusmods.com/>. [cit. 2025-12-28].
- [5] DBOLICAL PTY LTD. MODDB. Online. C2002-2025. Dostupné z: <https://www.moddb.com/>. [cit. 2025-12-28].
- [6] TANCRED, Nicoletta Chiara; TÜRKAY, Selen; VICKERY, Nicole E M a WYETH, Peta. Understanding Modding Devotion, Practices and Motivations using Serious Leisure Perspective. Online. Proceedings of the 35th Australian Computer-Human Interaction Conference. 2023, vol. 2024, s. 11-23. ISBN 9798400717079. Dostupné z: <https://doi.org/10.1145/3638380.3638386>. [cit. 2025-12-06].
- [7] DYER, Andy. History of PC game mods: PC Game Beyond! Smurfs to Counter-Strike and Beyond!. Online. In: NVIDIA CORPORATION. GeForce Articles, Guides, Gaming News, Featured Stories | NVIDIA. 2016. Dostupné z: <https://www.nvidia.com/en-us/geforce/news/history-of-pc-game-mods/>. [cit. 2025-12-07].
- [8] CD PROJECT RED. The Witcher 3: Wild Hunt. Online. VALVE CORPORATION. Steam. 18 May, 2015. Dostupné z: https://store.steampowered.com/app/292030/The_Witcher_3_Wild_Hunt/. [cit. 2025-12-29].
- [9] FULNECZEK, Adrian. Collaborating on mods. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Apr 16, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6328326/Collaborating+on+mods>. [cit. 2025-12-29].
- [10] SAQIB, M. RedENGINE Game Development Engine. Online. MYCPLUS. C2024. Dostupné z: <https://www.mycplus.com/featured-articles/redengine/>. [cit. 2026-01-24].
- [11] CD PROJECT RED. CD Projekt RED & GOG.com Special Event Summary. Online. CD PROJECT. The Witcher. 2012. Dostupné z: <https://www.thewitcher.com/us/en/news/447/cd-projekt-red-gog-com-special-event-summary>. [cit. 2025-12-29].
- [12] CD PROJECT. Modding support update. Online. CD PROJECT. The Witcher. 2024. Dostupné z: <https://www.thewitcher.com/ba/en/news/50443/modding-support-update>. [cit. 2025-12-29].
- [13] CD PROJECT RED. The Witcher 3: Wild Hunt – Cross-Platform Mod Support. Online. CD PROJECT. The Witcher. 2025, September 25, 2025. Dostupné

- z: <https://www.thewitcher.com/ch/en/news/51479/the-witcher-3-wild-hunt-cross-platform-mod-support>. [cit. 2025-12-29].
- [14] VALVE CORPORATION. Steam Workshop. Online. VALVE CORPORATION. Steam. C2025. Dostupné z: <https://steamcommunity.com/workshop/>. [cit. 2025-12-28].
- [15] THIEL, Sarah-Kristin a LYLE, Peter. Malleable Games – A Literature Review on Communities of Game Modders. Online. In: Proceedings of the 9th International Conference on Communities. New York, NY, USA: ACM, 2019, s. 198-209. ISBN 978-1-4503-7162-9. Dostupné z: <https://doi.org/10.1145/3328320.3328393>. [cit. 2025-12-09].
- [16] CD PROJECT RED. The Witcher 3 REDkit User Guide. Online. 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/overview?mode=global>. [cit. 2025-12-10].
- [17] CD PROJECT RED. The Witcher 3 REDkit. Online. VALVE. Steam. 21 May, 2024. Dostupné z: https://store.steampowered.com/app/2684660/The_Witcher_3_REDkit/. [cit. 2025-12-12].
- [18] CD PROJECT RED. Starting the Editor. Online. CD PROJECT RED. The Witcher 3 REDkit. 2024, Jun 05, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6324341/Starting+the+Editor>. [cit. 2025-12-12].
- [19] CD PROJECT RED. Start the game from the editor. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Apr 16, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6324858/Start+the+game+from+the+editor>. [cit. 2025-12-21].
- [20] CD PROJECT RED. TOOL: Vertex edit. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, May 21, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6328393/TOOL+Vertex+edit>. [cit. 2025-12-22].
- [21] CD PROJECT RED. TOOL: HDFview. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, May 21, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6328581/TOOL+HDFview>. [cit. 2025-12-23].
- [22] THE HDF GROUP. HDFView Software. Online. THE HDF GROUP. The HDF Group. C2006. Dostupné z: <https://www.hdfgroup.org/download-hdfview/>. [cit. 2025-12-24].
- [23] BLENDER FOUNDATION. Blender 4.5 LTS Manual. Online. 2025. Dostupné z: <https://docs.blender.org/manual/en/latest/>. [cit. 2025-10-28].
- [24] CD PROJECT RED. HOW-TO: Install the Blender plugin. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, May 15, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6326607/HOW-TO+Install+the+Blender+plugin>. [cit. 2025-12-24].
- [25] CD PROJECT RED. TOOL: Quests Debugger. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, May 21, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6328642/TOOL+Quests+Debugger>. [cit. 2025-12-24].

- [26] CD PROJECT RED. HOW-TO: Create a new NPC. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Apr 16, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6326348/HOW-TO+Create+a+new+NPC>. [cit. 2025-12-25].
- [27] CD PROJECT RED. TOOL: Attitude Editor. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6328884/TOOL+Attitude+Editor>. [cit. 2025-12-26].
- [28] FULNECZEK, Adrian. TOOL: Localized Strings Editor. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, May 14, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6329166/TOOL+Localized+Strings+Editor>. [cit. 2025-12-26].
- [29] FULNECZEK, Adrian. WitcherScript. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Jun 10, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/36306960/WitcherScript>. [cit. 2025-12-26].
- [30] FULNECZEK, Adrian. WS: Enable debug console in-game. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Jun 06, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/36208717/WS+Enable+debug+console+in-game>. [cit. 2025-12-26].
- [31] CD PROJECT RED. WS: Create your first script mod. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Jun 05, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/36241465/WS+Create+your+first+script+mod>. [cit. 2025-12-26].
- [32] MICROSOFT CORPORATION. Visual Studio Code. Software. Verze 1.107. Online. December 10, 2025. Dostupné z: <https://code.visualstudio.com/Download>. [cit. 2025-12-26].
- [33] FULNECZEK, Adrian. WS: Language Guide. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Jun 06, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/36307090/WS+Language+Guide>. [cit. 2025-12-26].
- [34] FULNECZEK, Adrian. WS: Script Compilation Errors overrides. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Jun 05, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/36241598/WS+Script+Compilation+Errors+overrides>. [cit. 2025-12-28].
- [35] ANOTHERSYMBIOTE. Script Merger. Online. BLACK TREE GAMING LTD. NEXUSMODS. 23 August 2015, 03 July 2020. Dostupné z: <https://www.nexusmods.com/witcher3/mods/484>. [cit. 2025-12-28].
- [36] CD PROJECT RED. Hotfix on PC is now available!. Online. CD PROJECT. The Witcher. 2024. Dostupné z: <https://www.thewitcher.com/us/en/news/50533/hotfix-on-pc-is-now-available>. [cit. 2025-12-28].
- [37] FULNECZEK, Adrian. WS: Script Studio basics. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Jun 06, 2024. Dostupné

- z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/36864027/WS+Script+Studio+basics>. [cit. 2025-12-28].
- [38] AUDIOKINETIC. Wwise. Online. C2025. Dostupné z: <https://www.audiokinetic.com/en/wwise/overview>. [cit. 2025-12-30].
- [39] FULNECZEK, Adrian. HOW-TO: Implement sounds and use Wwise. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6327994/HOW-TO+Implement+sounds+and+use+Wwise>. [cit. 2025-12-30].
- [40] FULNECZEK, Adrian. HOW-TO: Get a Wwise license. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024, Nov 12, 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/6327958/HOW-TO+Get+a+Wwise+license>. [cit. 2025-12-30].
- [41] AUDIOKINETIC. Wwise Modding Projects. Online. AUDIOKINETIC. Audiokinetic. 9/1/2024. Dostupné z: <https://www.audiokinetic.com/en/profile/school/6808/>. [cit. 2025-12-30].
- [42] FAZZITO, Guido. The Witcher 3 REDkit & Wwise: A Setup Guide. Online. 2025. Dostupné z: <https://www.audiokinetic.com/en/blog/the-witcher-3-redkit-tutorial/> [cit. 2025-10-28].
- [43] RINDO, Abigail. Game Narrative Design and UX Fundamentals Tools for Player-Centric Storytelling. Online – eBook. CRC Press, 2024. ISBN 9781040254219. Dostupné z: https://www.vitalsource.com/products/game-narrative-design-and-ux-fundamentals-abigail-rindo-v9781040254219?srsId=AfmBOopPtbq_i0FHjG0CR_5m8QGksbCOy2BEfyyq6kcwW0FSElejrids. [cit. 2025-12-30].
- [44] JIRÁSEK, Alois. Staré pověsti české. 1400. Praha: Dobrovský, 2021. ISBN 978-80-277-0315-9. [cit. 2025-12-30].
- [45] DOTPDN LLC. Paint.NET. Online. 9/9/2017, 11/13/2025. Dostupné z: <https://www.getpaint.net/index.html>. [cit. 2026-02-28].
- [46] VINICIUS MACHADO VENÂNCIO. Blender 3D Asset Creation for the Metaverse. Online – ebook. De Gruyter GmbH, Walter, 2023. ISBN 9781801813013. [cit. 2026-03-27].
- [47] ELEVENLABS INC. *ElevenLabs*. Online. [bez data]. Dostupné z: <https://elevenlabs.io>. [cit. 2026-04-21].
- [48] FULNECZEK, Adrian. Level-only assets: issue and fix. Online. CD PROJECT RED. The Witcher 3 REDkit User Guide. 2024. Dostupné z: <https://cdprojektred.atlassian.net/wiki/spaces/W3REDkit/pages/39256100/Level-only+assets+issue+and+fix>. [cit. 2026-04-04].

Příloha A: Zdrojový kód a prompty

A.1: Soubor utb_items.xml

```
<?xml version="1.0" encoding="UTF-16"?>
<redxml>
<definitions>

<!-- ### ABILITY DEFINITIONS ### -->
<abilities>

<!-- Silver Swords -->
<ability name="UTB reward sword _Stats">
  <weight type="base" min="4.2"/>
  <quality type="add" min="4"/>
  <SlashingDamage type="base" min="1"/>
  <item_level_min type="add" min="1"/>
  <item_level_max type="add" min="100"/>
  <critical_hit_chance type="add" min="0.10"/>
  <critical_hit_damage_bonus type="add" min="0.50"/>
  <armor_reduction type="base" min="150"/>
  <PoisonEffect is_ability="true"/>
  <buff_apply_chance type="add" min="0.10"/>
  <desc_poinsonchance_mult type="add" min="0.10"/>
  <spell_power_igni type="mult" min="0.20"/>
</ability>

</abilities>

<!-- ### ITEMS DEFINITIONS ### -->
<items>

<!-- Minor Quests-->
  <item
    name="utb_quest01_letter_from_blacksmith"
    category="misc"
    price="1"
    stackable="100"
    localisation_key_name
      ="item_name_utb_quest01_letter_from_blacksmith"
    localisation_key_description="item_desc_utb_strange_letter"
    icon_path="icons/inventory/scrolls/scroll15.dds"
  >
    <tags>ReadableItem, Quest, mod_book </tags>
    <recycling_parts>
      <parts count="1">Note</parts>
    </recycling_parts>
  </item>
</item>
```

```

name="utb_quest01_crystal_inactive"
category="misc"
price="1"
stackable="100"
localisation_key_name
    ="item_name_utb_quest01_crystal_inactive"
localisation_key_description
    ="item_desc_utb_crystal_inactive"
icon_path
    ="icons/inventory/ingredients/junk/gem_amber_64x64.png"
>
<tags>Quest, mod_junk</tags>
<recycling_parts>
    <parts count="1">Ashes</parts>
</recycling_parts>
</item>

<!-- swords -->
<item
    name                ="utb_knights_sword"
    category             ="silversword"
    price               ="1500"
    initial_durability  ="100"
    max_durability      ="100"
    enhancement_slots   ="3"
    stackable           ="1"
    grid_size           ="2"
    equip_template      ="reward_sword"
    equip_slot          ="silver_sword_back_slot"
    hold_slot           ="r_weapon"
    weapon              ="true"
    lethal              ="true"
    ability_mode        ="OnHold"
    hand                ="right"
    sound_identification="long steel"
    draw_event          ="DrawWeapon"
    draw_act            ="draw_steel_sword_back_act"
    draw_deact         ="draw_steel_sword_back_deact"
    holster_event       ="HolsterWeapon"
    holster_act         ="holster_steel_sword_back_act"
    holster_deact      ="holster_steel_sword_back_deact"
    localisation_key_name="item_name_utb_quest01_reward_sword"
    localisation_key_description
        ="item_desc_utb_desc_reward_sword"
    icon_path           ="icons/inventory/weapons/utb_sword_64x128.png"

```

```
>
<tags>
    Autogen, AutogenUseLevelRange, AutogenForceLevel,
    PlayerSilverWeapon, Weapon, swordlh, mod_weapon, EP1, EP2
</tags>
<base_abilities>
    <a>Default weapon _Stats</a>
    <a>UTB reward sword _Stats</a>
</base_abilities>
<recycling_parts>
    <parts count="2">Leather squares</parts>
    <parts count="4">Leather straps</parts>
    <parts count="4">Green gold ore</parts>
    <parts count="2">Emerald dust</parts>
    <parts count="6">Amber dust</parts>
</recycling_parts>
<anim_actions>
    <action
        name="draw"
        event="DrawWeapon"
        act="draw_steel_sword_back_act"
        deact="draw_steel_sword_back_deact">
    </action>
    <action
        name="holster"
        event="HolsterWeapon"
        act="holster_steel_sword_back_act"
        deact="holster_steel_sword_back_deact">
    </action>
    <action
        name="attack"
        event="attack_silver_sword_back"
        act="attack_silver_sword_back_act"
        deact="attack_silver_sword_back_deact">
    </action>
</anim_actions>
<anim_switches>
    <anim_switch
        category="steelsword"
        equip_slot="steel_sword_back_slot"
        event="steel_to_silver"
        switch_act="steel_to_silver_act"
        switch_deact="steel_to_silver_deact">
    </anim_switch>
    <anim_switch
        category="steelsword"
        equip_slot="axe_back_slot">
```

```
        event="steel_to_silver"
        switch_act="steel_to_silver_act"
        switch_deact="steel_to_silver_deact">
    </anim_switch>
</anim_switches>
<player_override>
    <bound_items>
        <item>utb_sword_scabbard</item>
    </bound_items>
</player_override>
</item>

<item
    name="utb_sword_scabbard"
    category="silver_scabbards"
    equip_template="utb_sword_scabbard"
    attachment_type="skinning"
>
    <tags>NoShow,NoDrop,EncumbranceOff</tags></item>
</items>

</definitions>
</redxml>
```

Výpis 1: xml kód definující pro REDkit jednotlivé předměty vytvořené touto prací

A.2: Soubor utb_rewards.xml

```
<?xml version="1.0" encoding="UTF-16"?>
<redxml>
<!-- ### REWARDS DEFINITION ### -->
  <reward
    name          ="utb_quest_reward_crystal"
    experience     ="3000"
    gold           ="5000"
    achievement    ="0"
    script         =""
    comment       =""
  />
  <reward
    name          ="utb_quest_reward_repaired"
    experience     ="2000"
    gold           ="0"
    achievement    ="0"
    script         =""
    comment       =""
  />
</redxml>
```

Výpis 2: kód definující odměny za úkol vytvořený touto prací

A.3: Prompty pro umělou inteligenci

Níže jsou uvedeny prompty použité pro vygenerování hlasových profilů postav pro nástroj Voice Changer od ElevenLabs:

- Postava kováře:

„An extremely old male village blacksmith voice, about 80 years old. Very gravelly, cracked, smoky, hoarse, and rough, with a deep but fading strength underneath. Worn out by decades at the forge, breathing smoke and working iron. Slight Slavic or Eastern European flavor in pronunciation. Speaks slowly, with weight, age, and fatigue in every sentence. Gruff, blunt, practical, and hardened, but still human and believable. He sounds like an old northern peasant craftsman with soot in his lungs, pain in his bones, and iron in his soul. Avoid sounding polished, theatrical, noble, magical, or cinematic.“

- Postava Geralta:

„A deep, low-pitched, gravelly male voice in his 40s, from a harsh northern medieval fantasy land. Slight Eastern European or Slavic influence in pronunciation, but clear and natural in English. Extremely stoic, emotionally restrained, laconic, and battle-worn. Dry, rough, weathered vocal texture with a husky rasp and dark chest resonance. Speaks slowly and deliberately, with short phrases, subtle pauses, and a tired, controlled tone. He sounds like a hardened professional monster hunter who has survived war, loss, and decades of violence. Cold, intimidating, experienced, cynical, and haunted, but never theatrical. Avoid sounding cheerful, youthful, polished, noble, dramatic, or like a narrator. The voice should feel realistic, masculine, dangerous, worn down, and quietly authoritative.“

Příloha B: Obrazová příloha

Tato příloha obsahuje vlastní snímky z publikovaného modu, který byl vytvořen touto prací.
Seznam obrázku v pořadí, jak jsou seřazeny:

- Pohled ze vstupního místa horské oblasti
- Snímek zachycující okamžik scény dialogu hráče s kovářem
- Pohled na pole před horou
- Pohled do rytířské místnosti







