

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Vývoj videohry v Unity: Wood Game



Autor: Matyáš Bezděk
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2025/26

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 6. 1. 2024

.....
Podpis autora

Abstrakt

Tato maturitní práce se zabývá vývojem videohry *Wood Game* v herním enginu Unity. Jádrem praktické části je tvorba herního prostředí a doplňků s využitím grafických nástrojů Blender a Gimp, přičemž důraz byl kladen na modelování 3D objektů, tvorbu 2D grafiky a textur. Součástí realizace byla rovněž implementace optimalizačních skriptů pro rendering a vizuální stránku projektu (backendová logika byla řešena v rámci týmové spolupráce). Cílem projektu bylo vytvořit simulátor těžby dřeva s ekonomickými prvky, který využívá fyziku herního prostředí, zejména při kácení stromů. Výsledkem práce je funkční hratelný prototyp, který demonstruje navržené mechaniky a umožňuje uživateli ovládat herní postavu v pohledu první osoby, používat nástroje a interagovat s vytvořenými herními prvky.

Klíčová slova

Unity, Gimp, Blender, VS Code, 3D grafika, 2D grafika, tvorba textur, tvorba objektů, vývoj herního prostředí, simulátor těžby dřeva, C#.

Abstract

This graduation thesis focuses on the development of the video game *Wood Game* using the Unity game engine. The core of the practical part lies in the creation of game assets and environments using Blender and Gimp tools, with a primary emphasis on 3D modeling, 2D graphics, and texture creation. The realization also included the implementation of optimization scripts for rendering and visual aspects of the project (backend logic was handled within team collaboration). The aim of the project was to create a wood logging simulator with economic features, utilizing game environment physics, particularly for tree felling. The outcome of the thesis is a functional playable prototype that demonstrates the designed mechanics and allows the user to control a character, utilize tools, and interact with the created game elements.

Keywords

Unity, Gimp, Blender, VS Code, 3D graphics, 2D graphics, texture creation, object creation, environment development, wood logging simulator, C#.

Obsah

Úvod	2
1 Teoretická východiska vývoje her	3
1.1 Herní design a vizuální styl	3
1.2 Proces vývoje her	4
1.3 Teorie UX a UI ve hrách	5
1.4 Herní enginey a technologie	6
1.5 Herní mechaniky	6
1.6 Grafická pipeline a rendering	7
2 Herní koncept a svět	8
2.1 Gameplay a herní smyčka	9
2.2 Herní svět a biomy	9
3 Tvorba grafických aktiv (Assets)	11
3.1 Využité nástroje	11
3.2 Tvorba vegetace	11
3.3 Hranice biomů (Kameny)	13
3.4 Tvorba budov a architektury	13
3.5 Tvorba nástrojů (Hard-surface modeling)	15
3.6 2D Grafika a UI	15
4 Implementace v Unity a Level Design	16
4.1 Import modelů a technické výzvy	16
4.2 Tvorba herního světa (Level Design)	16
5 Programová optimalizace scény	18
5.1 Návrh optimalizačního systému	18
5.2 Implementace kódu	18
5.3 Zhodnocení optimalizace	20
Závěr	21

ÚVOD

Téma maturitní práce jsem si zvolil na základě svého dlouhodobého vztahu k videohram a ze zájmu vyzkoušet si proces jejich vývoje v praxi. Společně s kolegou jsme se rozhodli vytvořit vlastní projekt, přičemž jsme chtěli postavit hratelnost na mechanice, která není zcela běžná – na fyzikálně věrném kácení a zpracování stromů.

Cílem projektu bylo vytvořit funkční prototyp hry s názvem *Wood Game*, primárně zaměřený na aplikaci fyziky dřeva v herním prostředí a osvojení si práce v herním enginu Unity. Hra je koncipována jako simulátor z pohledu první osoby (FPS) s jednoduchou, stylizovanou grafikou.

Hráč začíná s počátečním kapitálem, který mu umožňuje zakoupit základní sekeru. Herní náplní je těžba tří typů dřeva, které jsou ve světě rozmístěny s rostoucí obtížností dostupnosti – od dubového dřeva v blízkosti startovní pozice, přes cestu pouští, až po vzácnější zmrzlé dřevo. Vytěženou surovinu lze s pomocí vozidla transportovat, prodávat a získané finance investovat do lepšího vybavení. Hra rovněž obsahuje systém ukládání postupu.

Zatímco herní logika, backend a zmíněný systém ukládání byly řešeny v kooperaci s dalším vývojářem, tato práce dokumentuje mou část projektu. Zaměřuji se zde primárně na roli *Technical & Environment Artista*. Hlavní náplní mé práce byla tvorba herního prostředí (Level Design), modelování 3D objektů, tvorba textur a následná technická integrace a optimalizace těchto prvků v Unity pomocí skriptů v jazyce C#.

Práce je rozdělena na teoretickou a praktickou část. Teoretická část vymezuje základní pojmy herního designu, grafiky a používaných technologií. Praktická část pak detailně popisuje workflow tvorby assetů – od 2D nákresu přes 3D modelování v Blenderu až po finální post-processing ve hře.

1 TEORETICKÁ VÝCHODISKA VÝVOJE HER

Tato kapitola shrnuje základní teoretické poznatky z oblasti herního vývoje, které byly aplikovány při tvorbě projektu *Wood Game*. Zaměřuje se na proces designu, vizuální stránku, uživatelské rozhraní a technické pozadí používaných nástrojů.

1.1 HERNÍ DESIGN A VIZUÁLNÍ STYL

Herní design (Game Design) je proces navrhování obsahu a pravidel hry. Z vizuálního hlediska je klíčové zvolit takový grafický styl, který podporuje atmosféru hry a zároveň je realizovatelný v rámci dostupných časových a technických možností.

Častým přístupem u nezávislých (indie) vývojářů je využití tzv. **stylizované grafiky** nebo **low-poly** stylu. Tento směr se nesnaží o fotorealismus, ale využívá menšího počtu polygonů a zjednodušených textur k dosažení specifické estetiky. Výhodou je nižší náročnost na výpočetní výkon a rychlejší tvorba assetů.

1.1.1 Vizuální pojetí Wood Game

Pro projekt *Wood Game* byla zvolena stylizovaná grafika s důrazem na čistotu a přehlednost. Tento přístup mi umožnil soustředit se na tvorbu koherentního herního světa bez nutnosti vytvářet časově náročné fotorealistické textury. Jednoduchost modelů zároveň zajišťuje, že hra běží plynule i při větším počtu objektů (stromů) ve scéně.

1.2 PROCES VÝVOJE HER

Vývoj videohry je komplexní disciplína, kterou lze rozdělit do tří základních fází:

1. **Pre-produkce:** Fáze nápadu, tvorby konceptu, sepisování GDD (Game Design Document) a volby technologií.
2. **Produkce:** Samotná realizace. Zahrnuje programování, tvorbu grafiky (modelování, texturování), level design a zvukovou produkci.
3. **Post-produkce:** Testování, ladění chyb (bugfixing) a vydání.



Obrázek 1.1: Původní návrh chaty.

V profesionálních studiích jsou role striktně rozděleny (Game Designer, Programmer, Environment Artist, Level Designer). V menších týmech dochází ke kumulaci rolí.

1.2.1 Organizace vývoje projektu

Na projektu jsme pracovali ve dvoučlenném týmu. Mé zaměření pokrývalo vizuální stránku hry, tedy roli *Environment Artista* a *Level Designera*, a částečně roli *Technical Artista* při řešení importů a optimalizačních skriptů. Kolega zajišťoval backendovou logiku a systémy ukládání dat.

1.3 TEORIE UX A UI VE HRÁCH

Zkratka **UX** (User Experience) označuje celkový zážitek hráče a to, jak intuitivně se hra ovládá. **UI** (User Interface) představuje konkrétní prvky na obrazovce, jako jsou menu, ukazatele zdraví nebo inventář (HUD - Head-Up Display). Kvalitní UI musí být čitelné a konzistentní s vizuálním stylem hry.

1.3.1 Implementace UI

Pro uživatelské rozhraní, konkrétně pro inventář a ikony nástrojů, byl zvolen styl **Pixel Art**. Tento retro styl založený na viditelných pixelech přináší do hry žádaný kontrast vůči 3D prostředí a zajišťuje jasnou čitelnost ikon. Pro jejich tvorbu byl využit rastrový editor GIMP, se kterým mám předchozí zkušenosti.

1.4 HERNÍ ENGINEY A TECHNOLOGIE

Herní engine je softwarový framework určený pro tvorbu videoher. Poskytuje vývojářům sadu nástrojů pro rendering (vykreslování) grafiky, fyzikální simulace, detekci kolizí a správu zvuku. Mezi nejznámější současné enginy patří Unity, Unreal Engine a Godot.

1.4.1 Unity Engine

Pro realizaci projektu byl zvolen engine **Unity**. Důvody pro tuto volbu byly následující:

- **Podpora jazyka C#:** Jazyk C# je standardem v oblasti vývoje aplikací na platformě .NET a jeho znalost je součástí mého studia.
- **Dostupnost a komunita:** Unity má rozsáhlou dokumentaci a aktivní komunitu, což usnadňuje řešení technických problémů.
- **Licence:** Pro studentské a nekomerční projekty je engine zdarma (Unity Personal).

1.4.2 Grafické nástroje (Blender a GIMP)

Pro tvorbu 3D modelů byl využit open-source software **Blender**. Volba padla na tento nástroj díky předchozím zkušenostem nabytým během studia, což urychlilo proces modelování. Pro úpravu textur a 2D grafiky byl využit editor **GIMP**, který představuje efektivní bezplatnou alternativu k nástrojům jako Adobe Photoshop.

1.5 HERNÍ MECHANIKY

Herní mechaniky definují pravidla a způsoby interakce hráče s herním světem. V žánru simulátorů je kladen důraz na uvěřitelnost těchto interakcí.

1.5.1 Aplikované mechaniky

Jádrem hry je mechanika těžby dřeva. Na rozdíl od jednoduchého "kliknutí a zmizení stromu" využívá náš projekt fyzikální simulace pádu stromu a jeho následného zpracování. Další klíčovou mechanikou je ekonomický cyklus: *Těžba* → *Transport* → *Prodej* → *Nákup lepšího vybavení*. Interakce s objekty je technicky řešena pomocí metody *Raycasting* (vysílání paprsku z kamery hráče).

1.6 GRAFICKÁ PIPELINE A RENDERING

Grafická pipeline (vykreslovací řetězec) je sekvence kroků, které grafická karta provádí, aby převedla 3D data (vertexy, textury) na 2D obraz na monitoru. Klíčovými pojmy jsou zde **Shader** (program určující vzhled povrchu) a **Draw Calls** (požadavky na vykreslení).

1.6.1 Workflow v projektu

Proces tvorby grafiky probíhal v krocích:

1. Modelování geometrie a UV mapování v Blenderu.
2. Export do formátu FBX.
3. Import do Unity a nastavení materiálů (PBR workflow).
4. Optimalizace vykreslování pomocí skriptů (řešení viditelnosti objektů pro snížení zátěže GPU).

2 HERNÍ KONCEPT A SVĚT

Před samotnou technickou realizací bylo nutné definovat pravidla hry, strukturu herního světa a způsob, jakým bude hráč postupovat hrou. Tato kapitola popisuje návrh gameplay smyčky a rozvržení herního prostředí.



Obrázek 2.1: Ukázka začáteční zóny.

2.1 GAMEPLAY A HERNÍ SMYČKA

Hra je koncipována jako simulátor těžby dřeva z pohledu první osoby (FPS). Hráč se ocitá v roli dřevorubce v otevřeném světě. Cílem není jen bezduché kácení, ale efektivní management zdrojů a času.

Základní herní smyčka (Game Loop) funguje na následujícím principu:

1. **Příprava:** Hráč začíná s určitým finančním obnosem, za který si v obchodě zakoupí základní vybavení (sekeru).
2. **Těžba:** Vyhledání vhodných stromů v herním světě a jejich pokácení. Zde se uplatňuje fyzikální model – strom padá reálně podle směru seku a gravitace.
3. **Transport:** Vytěžené dřevo je nutné naložit a dopravit pomocí vozidla do výkupny. Vozidlo hraje klíčovou roli při přesunu větších objemů suroviny na delší vzdálenosti.
4. **Ekonomika:** Prodejem dřeva hráč získává herní měnu.
5. **Progrese:** Získané finance lze investovat do lepšího vybavení (efektivnější sekery, vylepšení), což umožňuje přístup k tvrdším a vzácnějším typům dřeva.

Celý postup lze kdykoliv uložit pomocí implementovaného Save systému, což hráči umožňuje budovat své zázemí dlouhodobě.

2.2 HERNÍ SVĚT A BIOMY

Herní mapa je koncipována jako jeden spojitý celek. Hráč začíná v bezpečném středu mapy a postupně expanduje do vzdálenějších oblastí.

2.2.1 Začáteční zóna

Startovní lokace hráče. Nachází se zde dům (spawn point), garáž pro vozidlo a obchodní místo. Tato zóna slouží jako ekonomické zázemí hry – zde probíhá veškerý prodej natěžených surovin a nákup nového vybavení.

2.2.2 Dubový les

Oblast navazující na začáteční zónu. Je tvořena jednoduchým travnatým terénem a duby. Tato surovina je nejlevnější, ale pro začínajícího hráče nejdostupnější.

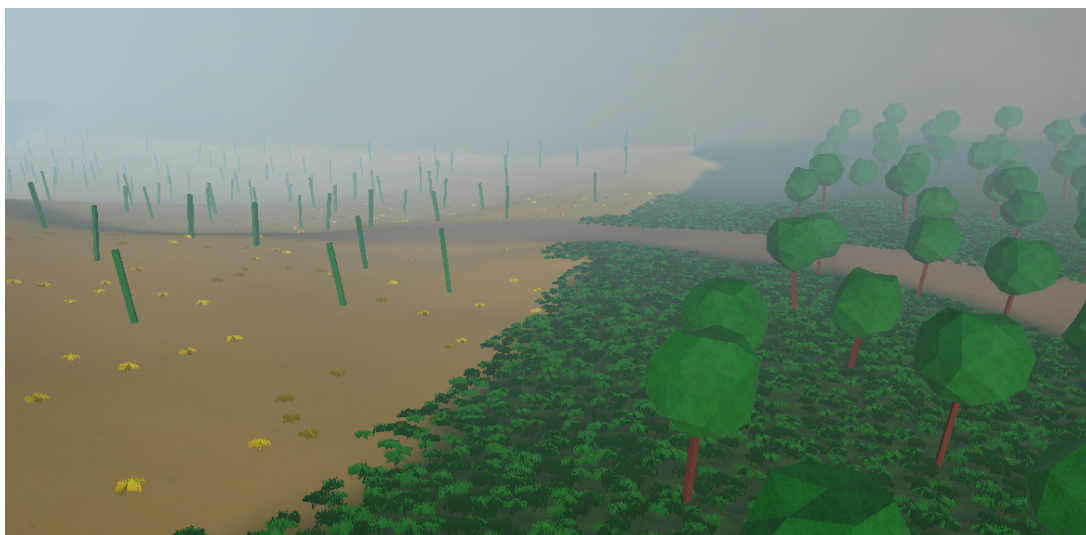
2.2.3 Poušť

Písečná oblast, která slouží jako mezistupeň mezi lesem a finální lokací. Hráč zde může těžit kaktusy, které mají vyšší hodnotu, ale jejich svoz je kvůli vzdálenosti časově náročnější.

2.2.4 Zamrzlá krajina

Nejvzdálenější část herního světa pokrytá sněhem. Roste zde zmrzlé dřevo – nejtvrdší a nejdražší surovina ve hře.

Cílem tohoto rozvržení bylo vytvořit uvěřitelné, i když stylizované prostředí, které hráče vizuálně odměňuje za prozkoumávání světa.



Obrázek 2.2: Přejít krajiny z dubového lesa na poušť.

3 TVORBA GRAFICKÝCH AKTIV (ASSETS)

Tato kapitola popisuje praktický postup tvorby herních modelů a textur. Vzhledem k tomu, že hra běží v reálném čase a obsahuje velké množství vegetace, byl při tvorbě kladen důraz na optimalizaci geometrie (Low-poly) a efektivní využití textur.

3.1 VYUŽITÉ NÁSTROJE

Pro tvorbu vizuální stránky hry byly zvoleny volně dostupné nástroje, které jsou standardem v nezávislém herním vývoji:

- **Blender:** Open-source program pro 3D modelování a UV mapování.
- **GIMP:** Rastrový grafický editor využitý pro tvorbu textur a Pixel Art grafiky.

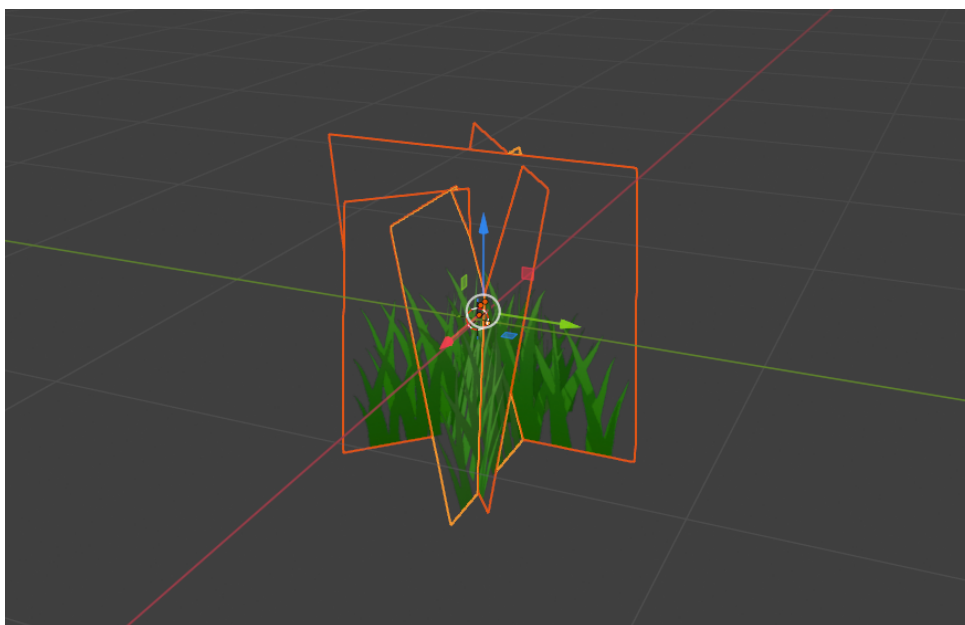
3.2 TVORBA VEGETACE

Vegetace tvoří drtivou většinu objektů ve scéně. Bylo nutné najít rovnováhu mezi vizuální hustotou lesa a výkonem počítače.

3.2.1 Modelování trávy

Pro trávu jsem se rozhodl nepoužívat techniku *billboard* (2D obrázek, který se natáčí vždy směrem ke kameře), protože v pohledu první osoby působí ploše. Místo toho jsem v Blenderu vytvořil vlastní 3D modely trsů.

- **Konstrukce:** Každý trs trávy je tvořen několika křížícími se plochami (planes), na které je aplikována textura.
- **Optimalizace:** Tento přístup využívá velmi nízký počet polygonů, ale díky textuře s průhledností (Alpha Channel) působí tráva hustě a prostorově.



Obrázek 3.1: Ukázka modelu trávy v Blenderu.

3.2.2 Stylizované stromy

Stromy byly modelovány s cílem zachovat stylizovaný vzhled. Koruna stromu není tvořena jednotlivými listy, ale vychází z geometrického primitiva koule (IcoSphere), která byla deformována a randomizována, aby působila přirozeně nepravidelně. Na tento tvar byla následně aplikována textura listů. Kmen je tvořen jednoduchým kvádrem, který je ve hře dynamicky upravován skriptem podle délky kmene.

Pro zvýšení plasticity byla materiálům stromů přidána tzv. **Normal Map** (mapa normál), která i na plochém modelu simuluje nerovnosti kůry a listů pomocí hry světla a stínu.

3.3 HRANICE BIOMŮ (KAMENY)

Pro vizuální i fyzické oddělení jednotlivých biomů (hranice mapy nebo přechody oblastí) jsem vytvořil sadu čtyř různých kamenů. Postup tvorby byl následující:

1. Vytvoření základní krychle (Cube).
2. Rozdělení geometrie (Subdivide) pro získání více vrcholů.
3. Aplikace funkce *Randomize Vertices* pro náhodné zdeformování tvaru.
4. Manuální úprava vrcholů do finálního "skalnatého" tvaru a aplikace materiálu.

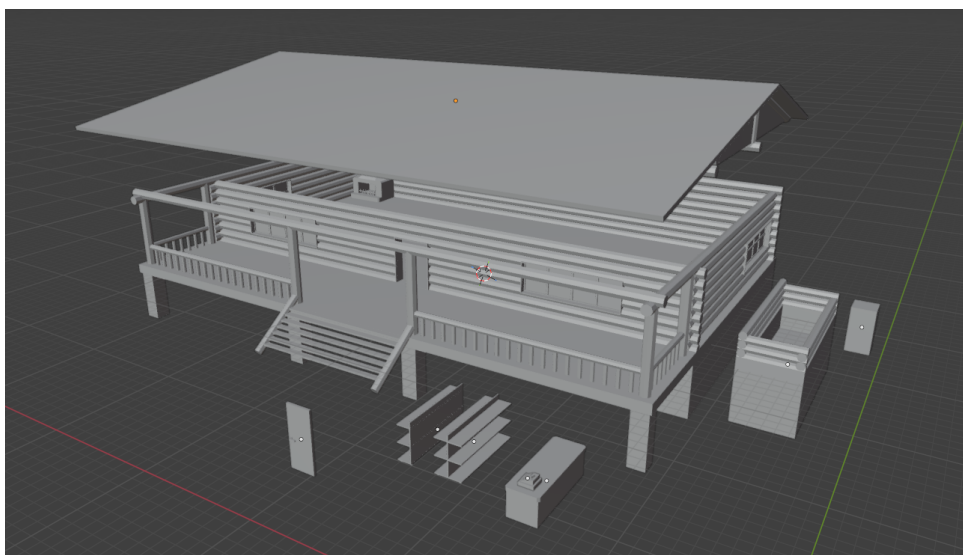
3.4 TVORBA BUDOV A ARCHITEKTURY

Centrální zóna hry obsahuje dvě klíčové budovy, které slouží jako zázemí pro hráče a obchodní místo. U obou staveb byl zachován jednotný stylizovaný vizuál, přičemž každá využívá odlišný přístup k modelování.

3.4.1 Obchod a Sell Point

Obchodní zóna je tvořena hlavním srubem a přilehlým menším stánkem pro prodej dřeva (*Sell Point*). Obě stavby sdílejí stejnou konstrukci z horizontálně ložených klád.

- **Modifikátor Array:** Abych nemusel klády skládat ručně, vymodeloval jsem jednu základní kládu (válec) a pomocí modifikátoru *Array* ji nechal automaticky namnožit do výšky, což zajistilo přesné rozestupy a urychlilo práci.
- **Výřezy (Boolean):** Pro vytvoření otvorů pro okna a dveří byl použit modifikátor *Boolean*. Ten pomocí operace odečítání (Difference) „vykrojil“ do stěn přesné otvory podle tvaru pomocného objektu.



Obrázek 3.2: Ukázka kompletního modelu chaty s doplňky.

3.4.2 Dům hráče s garáží

Startovní dům poskytuje hráči zázemí a funguje ve hře jako **Save Point** (místo pro uložení postupu).

- **Konstrukce:** Objekt byl postaven metodou skládání a škálování základních primitiv (Cubes). Stěny jsou tvořeny jednotlivými kvádry, které byly roztaženy (Scale) do požadovaných rozměrů.
- **Otvor pro dveře a okna:** Podobně jako u srubu, i zde byly průchozí otvory pro dveře a okna vytvořeny pomocí boolovských operací, což zajistilo čisté průřezy zdí.

3.5 TVORBA NÁSTROJŮ (HARD-SURFACE MODELING)

Pro interakci se světem (kácení) hráč využívá sekeru. Při jejím modelování byla využita technika *Box modeling*.

- **Hlava sekery:** Vychází ze základní krychle (Cube), která byla postupně dělena řezy (Loop Cuts) a vytahována (Extrude) do tvaru ostří.
- **Topůrko:** Vzniklo z válce, který byl deformován a modelován do ergonomického tvaru.

U těchto modelů jsem ponechal automatické rozbalení UV map (Auto UV), což u stylizované grafiky postačovalo a výrazně urychlilo práci.

3.6 2D GRAFIKA A UI

Veškeré 2D prvky, včetně textur povrchů a uživatelského rozhraní, vznikaly v editoru GIMP.

- **Seamless textury:** Pro kůru stromů a terén bylo nutné vytvořit bezešvé (seamless) textury, které na sebe plynule navazují ve všech směrech.
- **Pixel Art UI:** Inventář a ikony předmětů byly nakresleny stylem Pixel Art. Tento styl využívá přesnou mřížku pixelů a omezenou paletu barev, což zajišťuje dobrou čitelnost na obrazovce.



Obrázek 3.3: Ukázka ikony železné sekery v editoru GIMP.

4 IMPLEMENTACE V UNITY A LEVEL DESIGN

Samotné modely by bez importu do herního enginu nefungovaly. Tato kapitola popisuje technickou integraci vytvořených aktiv a proces tvorby herního světa (Level Design).

4.1 IMPORT MODELŮ A TECHNICKÉ VÝZVY

Kritickou částí práce byl přenos modelů z Blenderu do Unity (formát FBX). Během tohoto procesu bylo nutné vyřešit několik technických problémů:

4.1.1 Rozdílné souřadné systémy

Blender využívá jako vertikální osu **Z** (Z-up), zatímco Unity využívá osu **Y** (Y-up). Při importu často docházelo k nechtěné rotaci modelů o -90 stupňů na ose X, což bylo nutné manuálně korigovat v nastavení importu nebo aplikovat rotaci a měřítko přímo v Blenderu před exportem.

4.1.2 Nastavení materiálů

Materiály z Blenderu nejsou plně kompatibilní s Unity shadery, proto musely být vytvořeny znovu přímo v enginu.

- U trávy a listů bylo nutné nastavit parametr **Transparency** na režim *Alpha Cutout*, aby průhledné části textury nebyly vykresleny jako černé plochy.
- U nástrojů byly upraveny parametry **Smoothness** a **Metallic**, aby kovová část sekery odrážela světlo jinak než dřevěné topůrko.

4.2 TVORBA HERNÍHO SVĚTA (LEVEL DESIGN)

Tvorba mapy neprobíhala náhodně, ale s cílem vytvořit uvěřitelné biomy a jasné hranice mezi nimi.

4.2.1 Modelování terénu

Základ terénu byl vytvořen pomocí vestavěných nástrojů Unity Terrain. Pro dosažení přirozenějšího vzhledu eroze a nerovností jsem si doinstaloval rozšířenou sadu štětců (Brushes). Tím jsem se vyhnul nepřirozeně hladkým a uniformním kopcům, které vznikají při použití pouze základních nástrojů.

4.2.2 Procedurální generování vegetace

Vzhledem k rozloze mapy nebylo možné umisťovat každý trs trávy ručně. Pro tento účel byl využit nástroj *Vegetation Generator*. V něm jsem definoval pravidla pro jednotlivé biomy:

- **Dubový les:** Generuje se hustá zelená tráva pokrývající většinu plochy.
- **Poušť:** Generace je řídká, používá se varianta suché trávy.
- **Zimní krajina:** Střední hustota porostu, využívá se textura zmrzlé trávy.

4.2.3 Manuální dotváření scény

Zatímco vegetace byla generována procedurálně, důležité prvky a detaily byly umisťovány manuálně (Hand-placed), aby měl level designér plnou kontrolu nad hratelností.

- **Hranice biomů:** Pro jasné vizuální oddělení jednotlivých zón (např. les a poušť) byly použity modely kamenů, které vytvářejí přirozené bariéry a přechody.

5 PROGRAMOVÁ OPTIMALIZACE SCÉNY

Jedním z největších úskalí otevřených světů v Unity je náročnost na vykreslování (Rendering). Vzhledem k tomu, že scéna hry *Wood Game* obsahuje tisíce stromů a hustou vegetaci, narážel vývoj na limity grafického výkonu. Při prvních testech docházelo k výraznému poklesu snímkovací frekvence (FPS), protože grafická karta musela zpracovávat geometrii i těch stromů, které byly pro hráče v dálce neviditelné.

Standardní řešení Unity (Occlusion Culling) vyžaduje předpočítávání dat a pro dynamicky se měnící svět (kácení stromů) nebylo ideální. Proto jsem přistoupil k vytvoření vlastního řešení.

5.1 NÁVRH OPTIMALIZAČNÍHO SYSTÉMU

Místo toho, aby každý strom ve scéně sám kontroloval svou vzdálenost od hráče (což by znamenalo tisíce volání metody `Update` v každém snímku), jsem zvolil architekturu **centrálního manažera**.

Skript `TreeRenderManager` je umístěn na objektu kamery a spravuje seznam všech stromů ve scéně. Optimalizace je dosaženo pomocí tří klíčových technik:

1. **Vypínání Rendererů:** Skript neničí objekty, pouze vypíná jejich komponentu `Renderer`. Strom tam fyzicky stále "je" (má kolize, data), ale grafická karta ho ignoruje.
2. **Interval kontroly (Throttling):** Vzdálenost se nepřepočítává v každém snímku, ale pomocí `Coroutine` pouze jednou za 0,2 sekundy. Lidské oko tuto prodlevu při pohybu v dálce nepozná, ale procesoru to ušetří až 80 % práce.
3. **Optimalizace matematiky (SqrMagnitude):** Místo klasické funkce `Vector3.Distance`, která využívá náročnou odmocninu, porovnává skript tzv. "čtverec vzdálenosti" (`sqrMagnitude`). Matematicky je výsledek porovnání stejný, ale pro procesor je to mnohem rychlejší operace.

5.2 IMPLEMENTACE KÓDU

Níže je uveden kompletní zdrojový kód manažera, který se stará o řízení viditelnosti vegetace.

```

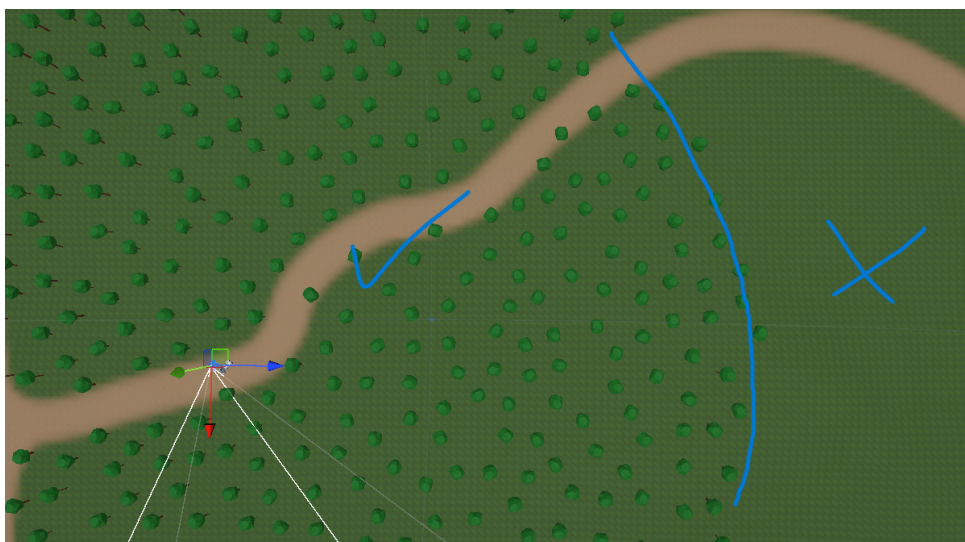
1 // TreeRenderManager - hlavní kontrolní smyčka (vynatek)
2 IEnumerator CheckVisibilityLoop()
3 {
4     WaitForSeconds wait = new WaitForSeconds(checkInterval);
5     while (true)
6     {
7         if (player == null)
8         {
9             yield return wait;
10            continue;
11        }
12
13        Vector3 playerPos = player.position;
14        float distSq = renderDistance * renderDistance;
15
16        for (int i = registeredTrees.Count - 1; i >= 0; i--)
17        {
18            TreeData tree = registeredTrees[i];
19            if (tree.transform == null)
20            {
21                registeredTrees.RemoveAt(i);
22                continue;
23            }
24
25            // Vypocet vzdalenosti
26            float currentDistSq = (tree.transform.position - playerPos).
                sqrMagnitude;
27            bool shouldBeVisible = currentDistSq < distSq;
28
29            if (tree.isVisible != shouldBeVisible)
30            {
31                tree.isVisible = shouldBeVisible;
32                foreach (var r in tree.renderers)
33                    if (r != null) r.enabled = shouldBeVisible;
34            }
35        }
36        yield return wait;
37    }
38 }

```

Kód 5.1: TreeRenderManager.cs - Hlavní optimalizační smyčka (zkráceno)

5.3 ZHODNOCENÍ OPTIMALIZACE

Po nasazení tohoto skriptu došlo ke stabilizaci snímkovací frekvence. Zatímco bez optimalizace klesalo FPS v hustém lese pod hranici 30 snímků za sekundu, s aktivním TreeRenderManagerem se hra drží stabilně nad 60 FPS, jelikož v každém okamžiku se reálně vykresluje pouze vegetace v okruhu 150 metrů kolem hráče.



Obrázek 5.1: Výsledek renderingu v okruhu 150 metrů.

ZÁVĚR

Cílem této maturitní práce bylo vytvoření vizuální stránky a technická optimalizace herního světa pro projekt *Wood Game*. Závěrem mohu konstatovat, že stanovené cíle byly splněny. Podařilo se nám vytvořit funkční prototyp hry, který demonstruje navržené mechaniky v kompletním 3D prostředí.

Během realizace jsem vytvořil ucelenou sadu grafických aktiv a navrhl tři odlišné biomy. Největší technickou výzvou byla integrace modelů z Blenderu do Unity a následná optimalizace výkonu pomocí vlastního C# skriptu. Ověřil jsem si, že pro plynulý běh hry v otevřeném světě je nutné aktivně řídit vykreslování objektů, nikoliv se spoléhat pouze na automatické procesy enginu.

Práce na projektu mi přinesla cenné zkušenosti, které přesahují rámec samotného programování. Poprvé jsem měl možnost pracovat na dlouhodobějším projektu v týmu, což vyžadovalo pravidelnou komunikaci a koordinaci. Osvojil jsem si základy práce s verzovacím systémem Git a nástrojem GitHub Desktop, což považuji za klíčovou dovednost pro svou budoucí praxi v IT.

Zároveň mi tento projekt posloužil jako důležitý experiment v oblasti herní grafiky. Zjistil jsem, že tvorba vizuálně atraktivního a konzistentního stylu (Game Art) je časově i technicky náročná disciplína.

Pokud by byl projekt dále rozvíjen, zaměřil bych se na zvýšení variability prostředí. Hra by profitovala z přidání drobných detailů, jako jsou různé druhy květin, menší kameny, variace travin a více druhů stromů, které by rozbily vizuální sterilitu některých pasáží a dodaly světu větší hloubku.

LITERATURA

- [1] Unity Technologies. *Unity User Manual 2022 LTS* [online]. [cit. 2026-01-05]. Dostupné z: <https://docs.unity3d.com/Manual/index.html>
- [2] Unity Technologies. *Unity Scripting API* [online]. [cit. 2026-01-05]. Dostupné z: <https://docs.unity3d.com/ScriptReference/>
- [3] Microsoft. *C# Guide - .NET Documentation* [online]. [cit. 2026-01-05]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/>
- [4] Blender Foundation. *Blender 4.0 Reference Manual* [online]. [cit. 2026-01-05]. Dostupné z: <https://docs.blender.org/manual/en/latest/>
- [5] The GIMP Team. *GNU Image Manipulation Program - User Manual* [online]. [cit. 2026-01-05]. Dostupné z: <https://docs.gimp.org/2.10/en/>
- [6] *How to Make Beautiful Terrain in Unity 2020* [cit. 2026-01-05]. Dostupné z: <https://www.youtube.com/watch?v=ddy12WHqt-M>
- [7] *Seamless Textures in Gimp* [cit. 2026-01-05]. Dostupné z: <https://www.youtube.com/watch?v=qAXRDfDaJMk>
- [8] *The best way to make billboard grass in your game* [cit. 2026-01-05]. Dostupné z: <https://www.youtube.com/watch?v=iAovvZTeL8I>
- [9] *Quickly Scatter Grass and Trees on Terrain* [cit. 2026-01-05]. Dostupné z: <https://www.youtube.com/watch?v=qV9YaRXupK8>
- [10] *Vegetation Spawner* [Unity Asset]. Unity Asset Store. [cit. 2026-01-05]. Dostupné z: <https://assetstore.unity.com/packages/tools/terrain/vegetation-spawner-free-automatic-tree-grass-placement-177192>
- [11] Google. *Google Images* [online]. Použito pro vyhledávání referenčních snímků a textur. [cit. 2026-01-05]. Dostupné z: <https://images.google.com/>
- [12] Google. *Gemini (Large Language Model)* [AI model]. Verze 2024-2025. Použito pro konzultace gramatiky a stylistickou úpravu textu. Dostupné z: <https://gemini.google.com/>