

# ZÁVĚREČNÁ STUDIJNÍ PRÁCE

## dokumentace

# Vývoj videohry v Unity: Wood Game

## Ondřej Honus



**Autor:** Ondřej Honus  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
**Zaměření:** se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2025/2026

## **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 dne 6. 1. 2026

.....  
Podpis autora

# **Abstrakt**

Tato závěrečná práce se zabývá vývojem 3D videohry Wood Game v herním enginu Unity. Cílem práce je vytvoření hratelné videohry založené na těžbě dřeva. Hlavními funkcemi jsou ekonomika, vozidla, ukládání postupu, inventář a hlavní menu. Herní funkce jsou realizované v jazyce C#. Data se ukládají do formátu JSON. 2D grafika a uživatelské rozhraní jsou vytvořeny v programu Aseprite. Hudba byla vytvořena v Bitwig Studiu a zvukové efekty v Audacity. Výsledkem je hratelný prototyp hry, ve které hráč těží dřevo, prodává ho a vylepšuje své vybavení.

## **Klíčová slova**

Videohra, Unity, C#, vývoj herního backendu, simulátor těžby dřeva, 2D grafika, tvorba hudby a zvuků.

# **Abstract**

This final thesis focuses on the development of a 3D video game called Wood Game using the Unity game engine. The objective of the project is to create a playable video game centered around wood harvesting. The main features include an economy system, vehicles, save/load functionality, inventory, and a main menu. The game mechanics are implemented in C# programming language, and data is stored in JSON format. The 2D graphics and user interface are created using Aseprite, while the music is composed in Bitwig Studio and sound effects are produced in Audacity. The result is a playable game prototype where players can harvest wood, sell it, and upgrade their equipment.

## **Keywords**

Video game, Unity, C#, game backend development, wood harvesting simulator, 2D graphics, music and sound creation.

# Obsah

<b>Úvod</b>	<b>5</b>
<b>1 TEORETICKÁ VÝCHODISKA</b>	<b>6</b>
1.1 Herní design . . . . .	6
1.1.1 Proces vývoje her . . . . .	6
1.2 UX a UI design . . . . .	6
1.2.1 UI . . . . .	6
1.2.2 UX . . . . .	7
1.3 Herní mechaniky . . . . .	8
1.4 Herní engine . . . . .	8
1.4.1 Výběr herního enginu . . . . .	8
1.5 Tvorba hudby a zvukových efektů . . . . .	8
1.5.1 Tvorba hudby . . . . .	9
1.5.2 Tvorba zvukových efektů . . . . .	9
<b>2 VYUŽITÉ TECHNOLOGIE</b>	<b>10</b>
2.1 Herní engine Unity . . . . .	10
2.2 Programovací jazyk C# . . . . .	10
2.3 2D grafika . . . . .	10
2.3.1 Aseprite . . . . .	11
2.4 Zvukové efekty a hudba . . . . .	11
2.4.1 Bitwig Studio . . . . .	11
2.4.2 Audacity . . . . .	11
<b>3 HERNÍ KONCEPT</b>	<b>12</b>
3.1 Gameplay . . . . .	12
3.2 Herní svět . . . . .	13
3.3 Herní funkce . . . . .	14
3.3.1 Kácení stromů . . . . .	14
3.3.2 Ukládání a načítání hry . . . . .	15
3.3.3 Obchod . . . . .	17
3.3.4 Prodej dřeva . . . . .	18
3.3.5 Hlavní menu . . . . .	19
3.3.6 Zvuky a hudba . . . . .	20
<b>Závěr</b>	<b>21</b>
<b>Seznam použitých informačních zdrojů</b>	<b>22</b>

# Úvod

Cílem této práce bylo vytvořit 3D videohru v Unity, ve které může uživatel ve 3D prostředí kácer stromy, získávat za ně peníze a vylepšovat své vybavení. Stromy, které se ve hře vyskytují na různých místech a mají různou hodnotu, lze pokácet sekerou a prodat. Za získanou měnu lze nakupovat lepší sekery či vozidlo. Uživatel má také možnost uložit svůj postup a načíst jej později.

Téma tvorby videohry jsem si zvolil, protože je mi svět herního vývoje blízký. Přijal jsem nabídku spolupráce od spolužáka, který se ujal vizuální a grafické stránky projektu. Zároveň jsem však chtěl rozšířit své znalosti v oblasti objektového programování a práce s herním enginem. Byla to pro mě příležitost vyzkoušet si komplexní proces vývoje od návrhu architektury až po finální ladění.

Na projektu jsem spolupracoval se spolužákem. Měli jsme jasné rozdělené role. Já jsem byl v roli hlavního programátora a technického designéra. Měl jsem na starosti veškerý backend, herní logiku (skripty), fyziku, systém ukládání dat (save/load system), část tvorby 2D grafiky (UI) a také celou stránku audia (hudba a efekty). Spolužák se zaměřil na vizuální stránku 3D prostředí, konkrétně na modelování 3D objektů a texturování.

Jedním z plánů byl i režim pro více hráčů (multiplayer), ale kvůli časovému omezení a složitosti tohoto úkolu jsem se rozhodl funkci vynechat a prozatím se zaměřit na vytvoření kvalitního singleplayerového zážitku.

# 1 TEORETICKÁ VÝCHODISKA

## 1.1 HERNÍ DESIGN

Herní design je proces vytváření pravidel, mechanik a struktur, které tvoří základ herního zážitku. Důležité je zaměřit se na interakci hráče s herním světem a na to, jakým způsobem budou herní prvky fungovat.

### 1.1.1 Proces vývoje her

Cesta od abstraktního nápadu k plně vyvinuté hře zahrnuje složité plánování a realizaci. Začíná konceptualizací – fází, kde je pečlivě vytvořen počáteční koncept, včetně žánru hry, estetiky a interaktivních prvků. Tato fáze je klíčová, určuje směr pro všechny následné vývojové aktivity. Designéři musí zohlednit demografii hráčů, cíle hraní a vizuální styl, aby tyto komponenty spojili do koherentního plánu. [1]

## 1.2 UX A UI DESIGN

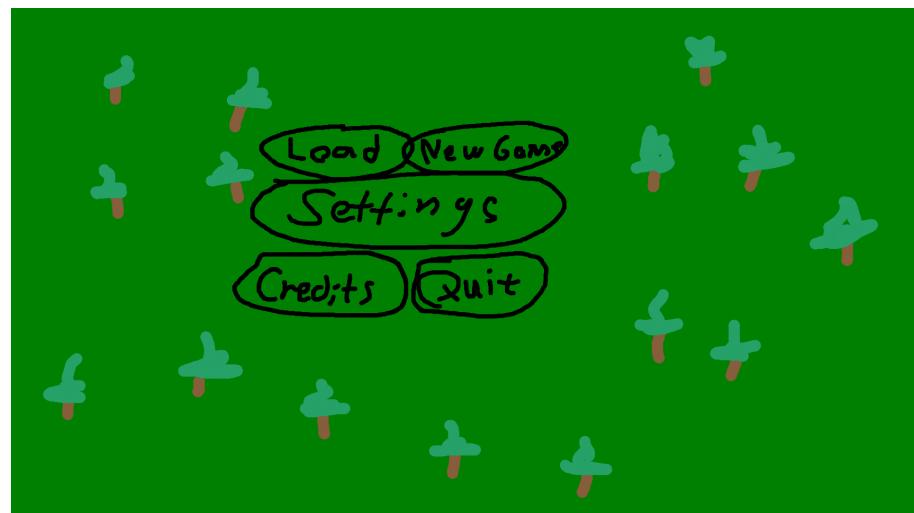
Uživatelské rozhraní (UI) a uživatelská zkušenosť (UX) jsou klíčové aspekty herního designu, které ovlivňují, jak hráči interagují s hrou a jak ji vnímají.

### 1.2.1 UI

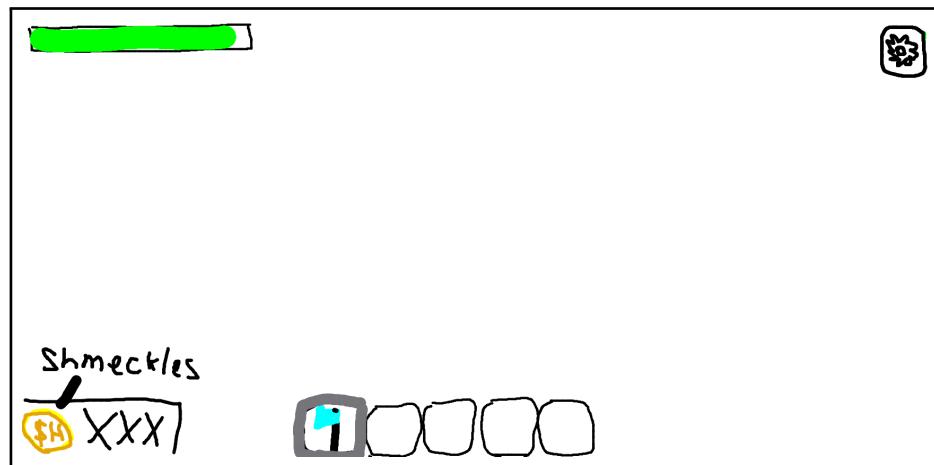
UI zahrnuje všechny vizuální prvky, jako jsou menu, tlačítka a ikony, které hráči používají k navigaci ve hře. Důležité je, aby bylo UI přehledné, intuitivní a esteticky příjemné, což zajišťuje pozitivní uživatelskou zkušenosť. Dobrou praktikou je minimalizovat počet kroků potřebných k dosažení cíle a zajistit, aby byly všechny prvky snadno dostupné.

## 1.2.2 UX

UX se zaměřuje na celkový zážitek hráče, včetně intuitivnosti ovládání, plynulosť interakcí a celkové spokojenosti s hrou. Do UX patří i zvukové efekty a hudba, které mohou výrazně ovlivnit atmosféru hry.



Obrázek 1.1: Příklad návrhu menu pro hru Wood Game.



Obrázek 1.2: Příklad návrhu herního rozhraní pro hru Wood Game.

## 1.3 HERNÍ MECHANIKY

Herní mechaniky jsou základními stavebními kameny, které definují interakce hráče s herním světem. Tyto mechaniky zahrnují pravidla, systémy a procesy, které určují, jak hráči dosahují cílů ve hře.

Mezi běžné herní mechaniky patří sbírání předmětů, řešení problémů, ekonomický systém a přežívání. V dobře navržené hře jsou herní mechaniky intuitivní a zábavné, což přispívá k celkovému zážitku hráče.

## 1.4 HERNÍ ENGINE

Herní engine (někdy také herní motor) je softwarový framework, který soustředí uje obecné funkce používané v počítačových hrách, díky čemuž dovoluje zrychlit a zlevnit vývoj nových her. Rozsah funkcí se u různých enginů liší a lze tak nalézt jak jednoduché knihovny omezující se na vykreslování, tak rozsáhlé enginy i s vlastní sadou vývojových nástrojů. [2]

Mezi nejznámější herní enginy patří například:

- Unity
- Unreal Engine
- Godot
- CryEngine
- GameMaker Studio

### 1.4.1 Výběr herního enginu

Výběr herního enginu závisí na požadavcích projektu, dostupných zdrojích a zkušenostech vývojového týmu. Každý engine má své výhody a nevýhody, které je třeba zvážit při plánování vývoje hry. Například Unity je oblíbený pro svou uživatelskou přívětivost a širokou komunitu, zatímco Unreal Engine je známý svou pokročilou grafikou a výkonem.

## 1.5 TVORBA HUDBY A ZVUKOVÝCH EFEKTŮ

Hudba a zvukové efekty hrají klíčovou roli ve vytváření atmosféry a zvyšování imerze ve videohrách. Správně zvolená hudba může podpořit emocionální odezvu hráče, zatímco zvukové efekty mohou zlepšit pochopení a vnímání herního světa a interakcí.

### **1.5.1 Tvorba hudby**

Při tvorbě hudby do pozadí je důležité zvážit styl hry, tempo a náladu, kterou chcete vyvolat. Například pro akční hru může být vhodná rychlá a energická hudba, zatímco pro dobrodružnou hru může být lepší klidnější a atmosférická hudba.

### **1.5.2 Tvorba zvukových efektů**

Zvukové efekty by měly být realistické a odpovídat akcím ve hře, jako jsou kroky postavy, zvuky zbraní nebo interakce s objekty. Důležité je také zajistit, aby zvukové efekty nebyly příliš rušivé a nepřehlušovaly hudbu.

## 2 VYUŽITÉ TECHNOLOGIE

### 2.1 HERNÍ ENGINE UNITY

Unity jsem zvolil jako herní engine pro vývoj hry Wood Game z několika důvodů. Především je to velmi populární engine s velkou komunitou, což znamená, že je k dispozici mnoho zdrojů, tutoriálů a assetů (např. mechaniky pohybu, fyziky), které mohou usnadnit vývoj a implementaci herních prvků. Unity také podporuje širokou škálu platform, což umožňuje snadné nasazení hry na různé operační systémy, jako jsou Windows, macOS, Android a iOS.

**Unity** je multiplatformní herní engine vyvinutý společností Unity Technologies. Byl použit pro vývoj her pro PC, konzole, mobily a web. [3]

### 2.2 PROGRAMOVACÍ JAZYK C#

Pro vývoj her v Unity se používá programovací jazyk C#.

- C# je moderní objektově orientovaný jazyk vytvořený společností Microsoft. Je nedílnou součástí platformy .NET a je široce používaný v oboru.
- Nabízí všeobecnost pro desktopové, webové i mobilní aplikace, široký ekosystém knihoven a aktivní komunitu, která usnadňuje učení.
- Podporuje objektově orientované programování, robustní zpracování výjimek, statické typy a jmenné prostory, které podporují modulární a udržitelný kód.
- Díky neustálému vývoji společnosti Microsoft a použití v .NET a Xamarin si C# udržuje vysokou poptávku po platformě a neustálý vývoj. [4]

C# v Unity umožňuje vývojářům vytvářet skripty pro herní logiku, ovládání postav, interakce s objekty a další herní funkce.

### 2.3 2D GRAFIKA

Pro tvorbu 2D grafiky jsem použil program **Aseprite**, který je specializovaný na pixel art (druh počítačové grafiky, která je vytvářena po jednotlivých pixelech [5]). Tento program nabízí uživatelsky přívětivé rozhraní a nástroje, které usnadňují tvorbu pixel art grafiky. Pro mě byl Aseprite ideální volbou díky své jednoduchosti a intuitivnosti.

### 2.3.1 Aseprite

Aseprite (dříve ASE nebo Allegro Sprite Editor) je proprietární bitmapový editor od společnosti Igara Studio napsaný v Qt (C++). Je zaměřený na tvorbu animovaných spritů a pixel artu pro videohry a využívá se v oblasti digitálního umění, animace a vývoji videoher. [6]

## 2.4 ZVUKOVÉ EFEKTY A HUDBA

Pro tvorbu zvukových efektů a hudby jsem použil programy **Bitwig Studio** a **Audacity**. Bitwig Studio jsem využil, protože ho již znám díky své zkušenosti s tvorbou hudby a také proto, že nativně podporuje operační systém Linux. Audacity jsem použil pro úpravu a nahrávání zvukových efektů, protože je to jednoduchý a efektivní nástroj pro základní úpravy zvuku.

### 2.4.1 Bitwig Studio

Bitwig Studio je moderní software pro tvorbu hudby (DAW), který slouží k nahrávání, skládání, zvukovému designu i živému vystupování. Na rozdíl od tradičních programů se zaměřuje na extrémní flexibilitu a modulární přístup, což z něj dělá oblíbený nástroj pro producenty elektronické hudby. [7]

### 2.4.2 Audacity

Audacity je volně dostupný, multiplatformní zvukový editor. Začali ho vytvářet Dominic Mazzoni a Roger Dannenberg na Carnegie Mellon University na podzim roku 1999. Funguje na operačním systému Mac OS X, Microsoft Windows, GNU/Linux, ale i v dalších operačních systémech. [8]

## 3 HERNÍ KONCEPT

### 3.1 GAMEPLAY

Hráč začíná u sebe doma s prázdnou kapsou a obnosem 50 Shmecklů (herní měna). Jeho prvním úkolem je zajít do obchodu a koupit si sekifu, aby mohl začít káct stromy. Po zakoupení sekery může hráč vyrazit do lesa, kde najde různé druhy stromů.

Každý strom má jinou hodnotu a počet úderů potřebných pro jeho pokácení. Po pokácení stromu z něj opadnou listy, které zmizí a hráči zůstane dřevo, které může dále zpracovat a prodat v obchodě za Shmeckly.

S penězi může hráč nakupovat lepší nástroje, které mu umožní káct stromy rychleji, nebo vozidlo, které mu umožní přepravit více dřeva najednou.

Hráč může také ukládat svůj postup tím, že si v menu hry vybere možnost „Uložit hru“. Uloží se vše, co má hráč na svém pozemku (auto, stromy), jeho peníze, životy a předměty v inventáři. Pokud hráč má auto, tak se uloží i jeho pozice na mapě včetně nákladu dřeva v autě.



Obrázek 3.1: Ukázka domova od hráče.

## 3.2 HERNÍ SVĚT

Herní svět je tvořen 3D prostředím s různými lokacemi, jako jsou různé typy lesů, obchod a domov hráče. Prostředí je navrženo tak, aby hráč ze začátku měl nejlehčí přístup k nejméně hodnotným stromům, a postupně se dostával k hodnotnějším stromům, které jsou umístěny dále od jeho domova.

Pro zjednodušení navigace hráče v lese je vytvořena hliněná cesta, kterou může hráč následovat, aby se v lese neztratil.

Mapa je rozdělena na tři různé typy lesů:

- Dubový les – nejméně hodnotné stromy
- Kaktusový les – středně hodnotné stromy
- Zamrzlý les – nejvíce hodnotné stromy



Obrázek 3.2: Ukázka dubového lesa.

## 3.3 HERNÍ FUNKCE

### 3.3.1 Kácení stromů

Hráč může káct stromy, pokud drží sekuru v ruce a je dostatečně blízko ke stromu. Po kliknutí levým tlačítkem myši na strom se podle pozice kurzoru při pohledu z první osoby, nebo pozice myši při pohledu třetí osoby, vypočítá pozice rozdělení kmene. Podle ní vzniknou 2 nové objekty, jeden z nich je kmen (má kolizi se zemí) a druhý je horní část stromu, která padne k zemi.

#### Implementace kácení stromů

Každý strom je reprezentován jako objekt s připojeným skriptem, který kontroluje, zda se ho hráč snaží přeseknout.

Jako indikátor průběhu kácení stromu slouží černý kvádr, který se postupně zvětšuje skrz kládu v bodě, kde hráč strom přesekává. Těchto indikátorů může být více, pokud hráč začne klikat na jiné místo, než na které původně kliknul.

Listing 3.1: Zjednodušená ukázka skriptu pro kalkulaci přeseknutí stromu v daném bodě

```

1 // Přečíst pozici myši
2 Vector2 mousePosition = Mouse.current.position.ReadValue();
3 // Vytvořit "paprsek" z kamery na pozici myši
4 Ray ray = Camera.main.ScreenPointToRay(mousePosition);
5 // Zasáhne paprsek objekt ve vrstvě stromů?
6 if (Physics.Raycast(ray, out RaycastHit hit, 100f, chopLayer))
7 {
8     // Je zasáhnutý objekt (strom) dostatečně blízko hráče?
9     if (hit.collider.gameObject == gameObject && Vector3.
10         Distance(playerInventory.transform.position, hit.point)
11         <= chopRange) {
12         // Spustit funkci na seknutí do stromu
13         HandleChop(hit.point);
14     }
15 }
```

Počet seknutí potřebných k pokácení stromu závisí na druhu sekery, kterou má hráč zrovna v ruce. Dále také závisí na typu stromu a jeho tloušťce.

Při přeseknutí stromu se také tzv. odemkne fyzika listů, které začnou padat na zem a po čase zmizí.



Obrázek 3.3: Ukázka indikace průběhu sekání stromu.



Obrázek 3.4: Ukázka pokáceného stromu.

### 3.3.2 Ukládání a načítání hry

Hráč může kdykoliv během hry uložit svůj postup pomocí tlačítka „Save Game“ v menu. Při uložení se vytvoří soubor ve formátu JSON, který obsahuje údaje o objektech. Ukládá se pozice hráče, jeho peníze, životy, předměty v inventáři, a pokud má hráč auto, tak se uloží i jeho pozice na mapě včetně nákladu dřeva v autě. Také se ukládají všechny objekty na pozemku hráče (pokácené stromy).

#### Implementace ukládání

Pro ukládání herních dat jsem vytvořil třídu **GameData**, která obsahuje všechny potřebné informace o stavu hry. Tato třída je poté použita pro samotné ukládání a načítání dat.

Listing 3.2: Ukázka třídy pro herní data

```

1  public class GameData {
2      // Peníze a životy hráče
3      public int savedMoney;
4      public float savedHP;
5
6      // Věci v inventáři hráče
7      public List<string> InventoryItems = new List<string>();
8
9      // Pozice a rotace auta
10     public float truckX, truckY, truckZ;
11     public float truckRotX, truckRotY, truckRotZ;
12
13     // Pozice a rotace hráče
14     public float playerX, playerY, playerZ;
15     public float playerRotX, playerRotY, playerRotZ;
16
17     // Seznam všech uložených pokácených stromů
18     public List<LogData> allLogs = new List<LogData>();
19 }
```

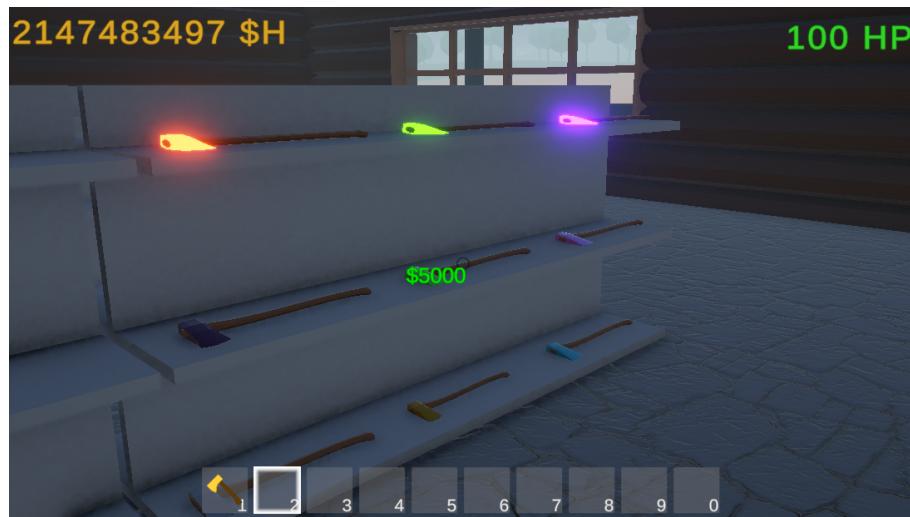
## Implementace načítání

Při načítání hry se načtou data ze souboru JSON a přiřadí se do daných herních objektů. Poté se nastaví souřadnice hráče a popřípadě jeho auta na pozici, ve které hru uložil. Všechny objekty (pokácené stromy) na jeho pozemku jsou poté obnoveny do stejné pozice, ve které je hráč nechal. Pokáceným stromům jsou také zpátky přiřazeny skripty a jejich parametry, které umožňují je stále sekat a nosit.

Pro načítání inventáře hráče jsem vytvořil funkci, která podle názvu uloženého předmětu najde odpovídající prefab (předpřipravený objekt) a přidá ho do inventáře hráče.

### 3.3.3 Obchod

V obchodě může hráč nakupovat lepší sekery, nebo také automobil pro přepravu dřeva. Každý předmět má svou cenu, která je zobrazena při najetí myši na daný předmět. Pokud hráč nemá dostatek peněz, tak se mu zobrazí cenovka červeně. Pokud má dostatek peněz, tak se zobrazí zeleně.



Obrázek 3.5: Ukázka regálu se sekerymi a cenovkou.

Pokladna funguje tak, že když hráč přinese objekt s označením „Sellable“ (prodejně) na pokladnu, dojde k odečtení částky z jeho peněz. Každý prodejný objekt má připojený skript, který obsahuje cenu daného objektu. Po prodeji objektu zmizí označení „Sellable“, aby hráč nemohl objekt koupit znova. V případě koupě sekery se objektu přidá skript, který umožní hráči sebrat sekuru do inventáře.

Pokud se hráč pokusí prodejný předmět ukrást, tak se mu zobrazí červená varovná hláška a objekt se vrátí zpět do regálu. Podobně funguje i obnovení objektu po koupi. Spustí se stejná funkce, která vrátí objekt zpět do regálu.

### 3.3.4 Prodej dřeva

Hned vedle obchodu se nachází místo, kde může hráč prodat nasbírané dřevo. Dřevo je možné prodat vhodným do bedny. Automaticky se poté přičtu peníze podle hodnoty dřeva.

Hodnota dřeva závisí na druhu stromu, ze kterého bylo získáno, a také na jeho velikosti. Cena dřeva se vypočítá podle následujícího vzorce:

$$\text{cena dřeva} = \text{index\_typu\_dřeva} \times \text{velikost.x} \times \text{velikost.y}$$

kde *index\_typer\_dřeva* je číslo reprezentující hodnotu daného druhu stromu, které jsem určil na základě jeho vzácnosti.

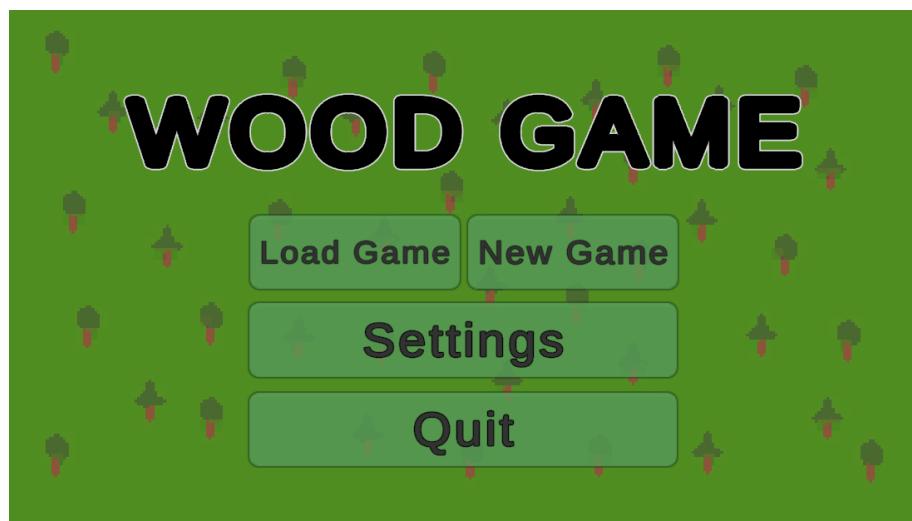


Obrázek 3.6: Ukázka prodeje dřeva ve hře Wood Game.

### 3.3.5 Hlavní menu

Hlavní menu obsahuje několik tlačítek, která umožňují hráči začít novou hru, načíst uloženou hru, upravit nastavení hry nebo hru ukončit. Menu je navrženo tak, aby bylo přehledné a intuitivní, s jednoduchou navigací. Pozadí menu bylo vytvořeno pomocí 2D grafiky v programu Aseprite.

Menší podoba herního menu je také dostupná během hry, kdy hráč může pomocí klávesy ESC menu otevřít a upravit nastavení nebo uložit svůj postup.



Obrázek 3.7: Ukázka hlavního menu hry Wood Game.

### Nastavení

V nastavení je hráči umožněno upravit kvalitu grafiky, kterou jsem rozdělil na dvě úrovně: *Nízká* a *Vysoká*. Nízká úroveň grafiky je optimalizována pro starší zařízení a zajišťuje plynulý chod hry i na méně výkonných počítačích. Vysoká úroveň grafiky nabízí detailnější textury, které zlepšují vizuální zážitek, ale vyžadují výkonnější hardware.

Dále je zde možnost upravit rozlišení obrazovky a zapnout nebo vypnout režim celé obrazovky. Hráč může také upravit celkovou hlasitost hry, hlasitost hudby a zvukových efektů.

### 3.3.6 Zvuky a hudba

Všechny zvuky a hudba ve hře byly vytvořeny mnou pomocí programů Bitwig Studio a Audacity. Hudba je navržena tak, aby doplňovala atmosféru hry a nebyla příliš rušivá během hraní. Zvukové efekty jsou realistické a odpovídají akcím ve hře, jako jsou například kroky postavy či zvuky kácení stromů.

#### Soundtrack

Soundtrack (hudba na pozadí) hry byl vytvořen speciálně pro tuto hru a při hraní se na pozadí přehrává ve smyčce. Cílem při tvorbě soundtracku bylo vytvořit chytlavou a příjemnou melodii, která hráče uklidní a nebude mu po celou dobu hraní vadit.

Hlavní část soundtracku je tvořena opakujícími se čtyřmi akordy hranými na elektrickou kytaru. Přes tuto smyčku jsem poté přidal zvuk bicích, který jsem získal pomocí MIDI nástroje EZdrummer 3, který jsem připojil do Bitwig Studia jako VST plugin. Efektu pro elektrickou kytaru jsem dosáhl pomocí Phaseru, reverbu a odstranění nízkých a vysokých frekvencí pomocí ekvalizéru. Soundtrack je dostupný na YouTube pod názvem „Wood Game OST | Fair Sapling“ na adrese: <https://www.youtube.com/watch?v=x9g16xBjP1Y>.

Podobná hudba byla použita i v menu hry, ale s mírně klidnější a jednodušší melodíí, aby se odlišila od hudby během hraní.

#### Zvukové efekty

Zvukové efekty byly vytvořeny pomocí nahrávání reálných zvuků a jejich následnou úpravou v programu Audacity. Například zvuk kácení stromu byl vytvořen nahráním úderu do zadní části akustické kytary, nebo zvuk kroků byl vytvořen nahráním zvuku hlíny v květináči.

# Závěr

Cílem této závěrečné práce bylo vytvořit funkční 3D videohru v prostředí Unity, která bude obsahovat mechaniky těžby dřeva, ekonomický systém a možnost ukládání postupu. Tento cíl byl naplněn. Vytvořená aplikace demonstruje funkční zhodnocení všech klíčových systémů: od interakce s objekty, kácení stromů, přes správu inventáře a ekonomiky, až po ukládání dat.

Ačkoliv se jedná o prototyp, hra obsahuje všechny důležité prvky a je plně hratelná. Výsledný produkt slouží jako solidní základ, který je možné v budoucnu dále rozšiřovat o nové mechaniky, jako je multiplayer, detailnější vizuální zpracování stromů nebo pokročilejší fyzika přenášení předmětů.

Kompletní zdrojový kód projektu a spustitelná verze hry jsou dostupné v repozitáři na adrese: <https://github.com/ondrejhonus/wood-game>

# **Seznam použitých informačních zdrojů**

- [1] Meshy – Základy návrhu her: Průvodce pro začátečníky [online]. [cit. 2026-01-04]. Dostupné z: <https://www.meshy.ai/cz/blog/game-design#proces-hern%C3%ADho-designu>
- [2] Wikipedia: Herní engine [online]. [cit. 2026-01-04]. Dostupné z: [https://cs.wikipedia.org/wiki/Hern%C3%AD\\_engine](https://cs.wikipedia.org/wiki/Hern%C3%AD_engine)
- [3] Wikipedia: Unity (herní engine) [online]. [cit. 2026-01-04]. Dostupné z: [https://cs.wikipedia.org/wiki/Unity\\_\(hern%C3%AD\\_engine\)](https://cs.wikipedia.org/wiki/Unity_(hern%C3%AD_engine))
- [4] ITD: Úvod do programovacího jazyka C# [online]. [cit. 2026-01-04]. Dostupné z: <https://informatecdigital.com/cs/%C3%9Avod-do-programovac%C3%ADho-jazyka-C-SHARP/>
- [5] Wikipedia: Pixel art [online]. [cit. 2026-01-04]. Dostupné z: [https://cs.wikipedia.org/wiki/Pixel\\_art](https://cs.wikipedia.org/wiki/Pixel_art)
- [6] Wikipedia: Aseprite [online]. [cit. 2026-01-04]. Dostupné z: <https://cs.wikipedia.org/wiki/Aseprite>
- [7] Bitwig: Bitwig Studio [online]. [cit. 2026-01-05]. Dostupné z: <https://www.bitwig.com/>
- [8] WikiKnihovna: Audacity [online]. [cit. 2026-01-05]. Dostupné z: <https://wiki.knihovna.cz/index.php/Audacity>