

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

DIPLOMOVÁ PRÁCE

3D plošinová hra vytvořená pomocí Unity Engine

Hirok



2022

Vedoucí práce: Mgr. Petr Osička,
Ph.D.

Bc. Albert Kaufman

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Bc. Albert Kaufman
Název práce: 3D plošinová hra vytvořená pomocí Unity Enginu (Hirok)
Typ práce: diplomová práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2022
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Petr Osička, Ph.D.
Počet stran: 36
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Bc. Albert Kaufman
Title: 3D platformer in Unity Engine (Hirok)
Thesis type: master thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2022
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Petr Osička, Ph.D.
Page count: 36
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Práce se zabývá vytvořením 3D plošinové hry v Unity Enginu. Hra je podobná hře Crash Bandicoot nebo Spyro ze staré éry herní konzole PlayStation 1. V šesti levelech bude hráč čelit jak nepřítelům, tak skákání po plošinkách. Nad rámec zadání byla složena i vlastní hudba.

Synopsis

This thesis deals with the creation of a 3D platform game in Unity Engine. The game is similar to Crash Bandicoot or Spyro from the old era of PlayStation 1. In six levels, the player will face both enemies and jumping on platforms. Custom music was also composed beyond the brief.

Klíčová slova: videohra, 3D plošinovka; unity engine

Keywords: videogame; 3D platformer; unity engine

Děkuji vedoucímu práce za cenné rady, které mi pomohly při vývoji hry a rodině za veškerou podporu, kterou mi poskytla během studia.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1 Úvod	8
1.1 Cíle a členění	8
1.2 Motivace	8
1.3 Název a logo	8
1.4 Co je to 3D plošinová hra	9
1.5 Existující 3D plošinovky	9
1.5.1 Crash Bandicoot	10
1.5.2 Spyro	10
1.5.3 A Hat in Time	11
1.5.4 Funk Unplugged	12
2 Použité technologie	13
2.1 Unity Engine	13
2.2 C#	13
2.3 Blender	14
2.4 Photoshop	14
2.5 Bosca Ceoil	14
2.6 Audacity	14
3 Základy Unity Enginu	15
3.1 Co je to Asset	15
3.2 Uživatelské rozhraní	15
3.2.1 Panel nástrojů	15
3.2.2 Scéna	15
3.2.3 Okno hierarchie	16
3.2.4 Okno inspektoru	16
3.2.5 Okno projektu	16
3.3 Scény	16
3.4 GameObject	16
3.5 Prefab	16
3.6 Skripty	17
3.7 Ukládání dat	17
3.8 Animation Controller	18
3.9 Audio	18
3.10 Materiály, Textury a Shadery	18
4 Programátorská příručka	19
4.1 Skripty	19
4.1.1 Hráčovy skripty	19
4.1.2 Nepřátelské skripty	19
4.1.3 Level a menu skripty	20
4.1.4 Sběratelské předměty skripty	20
4.1.5 Ostatní skripty	21

4.2	Animační kontrolery	21
4.2.1	HirokAnimatorController	21
4.2.2	Controllery nepřátel	21
4.2.3	Ostatní animační kontrolery	21
4.3	Scény	23
4.4	Zvuky	23
4.5	Hudba	23
4.6	Dabing	23
4.7	Shadery	23
4.8	Možná vylepšení	24
5	Uživatelská příručka	25
5.1	Požadavky	25
5.2	Instalace a odinstalace	25
5.3	Ovládání	25
5.4	Hlavní menu	26
5.4.1	Hrát	26
5.4.2	Nastavení	27
5.4.3	Statistiky	27
5.4.4	Nápověda	27
5.4.5	Dětský režim	28
5.4.6	Pauzovací menu	28
5.5	Příběh	28
5.6	Uživatelské rozhraní	28
5.7	Pohybový systém	29
5.8	Soubojový systém	29
5.9	Sběratelské předměty	29
5.10	Sytém vylepšování	30
5.11	Typy nepřátel	30
5.12	Levely	30
	Závěr	32
	Conclusions	33
	A Obsah přiloženého CD/DVD	34
	Literatura	35

Seznam obrázků

1	Logo	9
2	Původní Crash Bandicoot z roku 1996	10
3	První díl Spyra z roku 1998	11
4	A Hat in Time	11
5	Funk Unplugged	12
6	Unity Engine - Uživatelské rozhraní	15
7	Automat pro animace těla hlavní postavy	22
8	Automat pro animace rukou hlavní postavy	22
9	Automat pro animace nepřítele Roháče	23
10	Hlavní menu	27
11	Uživatelské rozhraní	28
12	Disketa a zlatka	30
13	Výbušňák a roháč	31
14	Level 3	31

Seznam zdrojových kódů

1	Kód po vytvoření skriptu	17
---	------------------------------------	----

1 Úvod

1.1 Cíle a členění

Cílem této diplomové práce bylo vytvořit 3D plošinovou hru v Unity Enginu. Hra by se měla inspirovat tituly jako jsou například Crash Bandicoot nebo Spyro z herní konzole Playstation 1. Hra by měla mít hráčem ovládanou postavu, která se umí pohybovat, skákat, dvojitě skákat a mít další podobné pohybové schopnosti. Dále by měla hra mít jednoduchý soubojový systém pro boj s nepřáteli a systém pro postup ve hře. Tím se myslí například levely nebo sběratelské předměty, které vedou k odměnám.

Tato práce je rozdělena na pět kapitol. V úvodní kapitole se kromě motivace a názvu hry hlavně zaměřuji, co je to 3D plošinová videohra a příklady takovýchto her. Ve druhé kapitole shrnuji použité technologie. Ve třetí kapitole ve stručnosti popisuji základy Unity Enginu. Ve čtvrté kapitole je programátorská příručka, kde popisuji hlavně skripty hry a v poslední páté kapitole se nachází uživatelská příručka ve které je popsáno, jak hru hrát.

1.2 Motivace

Mým dětským snem vždy bylo vytvářet hry. Během studia jsem si řekl, že by 3D plošinová hra v Unity Enginu nemusela být vůbec špatný nápad. Chtěl jsem si totiž zkusit, jaké by to bylo vytvořit trojrozměrnou hru úplně sám. To znamená včetně vlastních modelů, animací, textur i hudby. Dříve jsem už pár her vytvořil, ale byly to pouze takové dvojrozměrné hříčky na pár minut. Během studia jsem měl i předmět, jehož výstupem byla hra na mobilní zařízení s operačním systémem Android. Konkrétně jsem vytvořil variaci na hru Sudoku. Víím, že vytvářet hry úplně sám je velká výzva, protože zastávat pozice programátora, umělce i skladatele je velmi složité. Věřím ale, že pokud mě něco hodně baví, tak je to nutné dělat i přes tuto náročnost.

1.3 Název a logo

Hru jsem pojmenoval Hirok, což je odvozenina od slova anglického slova *Hero*, aneb hrdina s písmenem *k* na konci. Hirok je jméno hlavní postavy. U loga jsem se inspiroval mým již dříve vytvořeným pro jinou hru s názvem Warp Back, kterou jsem vytvořil v roce 2016. Logo se mi líbilo, a tak jsem ho přepracoval aby sedělo i do této hry. Logo se objevuje při spuštění hry a jako ikona ve spouštěcím souboru. Ikona jako taková je hlava hlavního hrdiny vystupující z šedého pozadí. Celý obrázek je pak filtrem zaskicován aby vypadal jako rozmazaná kresba. Logo jsem vytvořil v programu Photoshop CS5.



Obrázek 1: Logo

1.4 Co je to 3D plošinová hra

3D plošinová hra [1], v češtině též často označována jako 3D plošinovka, skákačka nebo hopsačka, je videoherní žánr, který vychází z 2D plošinovky. Většinou hráme za postavu, která má dojít z bodu A do bodu B. Level design je rozmanitý, plný nástrah a překážek jako jsou například různé propasti, bodláky, padající plošiny atd. K jejím překonání má hráčova postava různé schopnosti. Základem je skákání, dvojité skákání, plachtění ve vzduchu nebo odrážení od zdí. Drtivá většina plošinovek obsahuje i jednoduchý souborový systém, kdy většinou pomocí jednoho tlačítka provádí postava útok, aby zničila své nepřátele.

Jak jsem již zmínil, tak první byly dvojrozměrné plošinovky, kde se první hra z tohoto žánru datuje do roku 1981 a je jím Space Panic. Osmdesátá léta minulého století je také období, kdy tento žánr zažil největší popularitu. Vznikly zde takové legendární hry jako je Donkey Kong nebo Mario Bros. Trojrozměrné plošinovky vznikaly až o dekádu později. V osmdesátých letech vnikaly jen takové experimenty, kdy 2D hry simulovaly 3D. První plně 3D plošinovka vnikla v roce 1990 a byla jím hra Alpha Waves. Opravdový rozmach ale nastal až v roce 1996, kdy na konzoli Playstation 1 vyšla hra Crash Bandicoot a na konzoli Nintendo 64 Super Mario 64. Od tohoto momentu se stal žánr velmi oblíbeným. Vycházely hry jako Croc, Sonic Adventure nebo Psychonauts. Bohužel, stejně jako u 2D plošinovek, trvala jejich oblíbenost přibližně dekádu, kdy je nahradil dodnes oblíbený žánr, a to akční adventury. Ty mají na rozdíl od plošinovek realističtější pojetí, propracovanější souborový systém a silný příběh. Dnes vychází plošinovky hlavně od nezávislých herních studií. Velké korporáty sázejí spíš na akční adventury, a tak vysoko rozpočtové plošinovky vychází velmi málo.

1.5 Existující 3D plošinovky

Do této podkapitoly jsem vybral dvě starší a dvě novější 3D plošinovky s tím, že dvě jsou vysoko rozpočtové a dvě od nezávislých tvůrců.

1.5.1 Crash Bandicoot

Crash Bandicoot [2] je jedna z prvních 3D plošinovek, co kdy vnikla. Vyšla v roce 1996 pro herní konzoli Playstion 1 a dlouho byla hlavní postava jmenující se, jako název hry neoficiálním maskotem společnosti Sony. Hru vytvořilo studio Naughty Dog, jež později vytvořilo i plošinovku Jak and Daxter. Ve hře hrajete za tasamánského čerta, který sbírá jablka, ničí bedny, sbírá různá vylepšení, jako jsou například různé masky a zabíjí nepřátele. Hra funguje tak, že hráč může jít pouze vpřed a pohyb do stran je silně omezen. Občas se objevují i levely, které jsou svou perspektivou ve 2D, ale s 3D objekty. Levely se odemykají postupně za sebou. Díky svému velkému úspěchu vznikly v devadesátých letech další 2 pokračování a závodní spin-off. Pak značku přebrala jiná studia a vytvořily další díly, které už ale nebyly tak oblíbené, jako ty první z devadesátých let. Kvůli své oblíbenosti a kultovnímu statusu, vnikla v roce 2016 předělávka prvních třech dílů a v roce 2020 vznikl díl úplně nový, který navazuje na odkaz dílů původních.



Obrázek 2: Původní Crash Bandicoot z roku 1996

1.5.2 Spyro

Spyro [3] je 3D plošinová hra z roku 1998 od vývojářského studia Insomniac Games. Hrajete v ní za malého dráčka Spyra, který musí během své cesty sbírat diamanty a krystaly. Během své cesty musí vysvobodit ze zajetí uvězněné draky, kteří mu pomohou na cestě dál. Hra je spíš akčněji laděná a soubojový systém je tedy trochu rozsáhlejší. Můžeme plivat oheň, nabrat někoho na rohy a nebo z výšky dupnout na zem. Z pohybových schopností tu máme sprintování, skákání nebo plachtění. Jsou zde i levely kdy létáme. Celkově vznikla trilogie a v roce 2019 vznikla předělávka, kterou poháněl Unreal Engine 4. Hra byla stejně jako Crash Bandicoot vytvořena znovu od píky, a to včetně nového dabingu i hudby.



Obrázek 3: První díl Spyra z roku 1998

1.5.3 A Hat in Time

A Hat in Time [4] je relativně nová plošinová hra, která vznikla v roce 2017 a stojí za ní studio Gears for Breakfast. Hra vnikala bez velkého vydavatele, a tak by se dala nazvat nezávislou hrou. Hrajeme v ní za malou holčičku, která se musí ve 4 světech proskákat a probojovat na konec. Navštívíme město místní mafie, filmové studio, temný les nebo nebesa. Ze všech zmíněných her je zde pohybový systém nejdotaženější. Můžeme zde skákat, dvojitě skákat, uskakovat ve vzduchu, zachytávat se o kraje a mnoho dalšího. To vše je podpořeno velmi plynulými animacemi. Cestou bude hrdinka získávat nové klobouky díky kterým získá nové schopnosti ulehčující postup ve hře.



Obrázek 4: A Hat in Time

1.5.4 Funk Unplugged

Funk Unplugged [5] je z tohoto seznamu asi nejméně známá, ale zároveň nejnovější hra. Je z roku 2018 a stálo za ní malé herní studio Vamped Games, které je stejně jako autoři hry A Hat in Time nezávislé. Tato hra je zářným příkladem moderní 3D plošinové hry. Sbíráme zde předměty, kterými si vylepšujeme postavu. Máme zde sadu pohybů, které od plošinovky očekáváme. Můžu zde uvést dvojitý skok, odrážení od zdí, plachtění, jednoduchý souborový systém a jednoduchou grafiku. Hrajeme zde za chodící gramofon, který musí najít všechny gramofonové desky, abychom mohli porazit hlavního nepřítele hry, Kloboučníka. Spojovníkem celé hry je centrální hub, ze kterého si postupně odemykáme jednotlivé levely. Do levelů se můžeme pak následně vracet, abychom dosbírali všechny předměty, které jsou nutné pro postup ve hře.



Obrázek 5: Funk Unplugged

2 Použité technologie

2.1 Unity Engine

Unity Engine [6] je multiplatformní herní engine vytvořen firmou Unity Technologies. Původně byl vydán v roce 2005 exkluzivně pro operační systém Mac OS X. Postupně se ale rozšířil na všechny současné platformy jako například Windows, Linux nebo Android. V enginu lze vytvářet jak 2D, tak 3D hry všech žánrů. Unity má čtyři verze předplatného. Verze Personal, která je zadarmo, je pro vývojáře, jejichž příjmy z vytvořených her nepřesáhnou 100 000 dolarů za rok. Pokud je výdělek větší, je si nutné Unity ročně předplácet. V placeném předplatném jsou navíc nadstandardní služby, jako je třeba pokročilá cloudová diagnostika nebo rozšířená technická podpora. Unity podporuje programovací jazyk C# a do srpna roku 2017 podporoval i javascript. Do roku 2015 se verzoval engine celočíselně až do verze 5, a od roku 2016 se verzuje číslem roku ve kterém vyšel. Navíc každého čtvrt roku přicházejí podverze s číslem 1-4. Nejnovější verzí je 2022.1. Nové verze vždy přinášejí nějaký technologický posun. Například verze 2018 přinesla novou formu renderování, která posunula grafickou stránku her o kus dál. Diplomovou práci jsem začal vyvíjet na verzi 2019.2 a u ní jsem zůstal, abych přešel případným problémům s kompatibilitou při přechodu na novější verzi. Unity je díky jednoduchému uživatelskému rozhraní velice jednoduchý k naučení a velmi často je doporučován jako engine, na kterém začít vytvářet hry.

Druhým enginem, který se doporučuje, je Unreal Engine [7], který je obohacen o modernější technologie a hry vytvořené na něm vypadají lépe. Má také delší historii, jelikož první verze vyšla už v roce 1998. Oproti Unity má ale o trochu nepřehlednější UI a pro plnou kontrolu vyžaduje znalost jazyka C++, který je oproti C# složitější, ale na druhou stranu jsou pak výsledné hry optimalizovanější. Pro začátečníky naštěstí Unreal Engine nabízí i vizuální skriptování přes takzvané Blueprints.

Další zástupce herních enginů může být relativně nový Godot engine [8], který je zcela zdarma. Lze v něm programovat přes GDScript, vlastní jazyk, který je podobný Pythonu, nebo nově má i plnou podporu C#. Přes svou mladost Godot postrádá přehlednější uživatelské rozhraní a moderní technologie, díky kterým vypadají hry na pokročilejších enginech tak dobře.

2.2 C#

C# [9] je vysokoúrovňový objektově orientovaný programovací jazyk vyvinut společností Microsoft v roce 2000. Vznikl zároveň s platformou .NET. Je založen na jazyce C++ a Java. Byl schválen standardizačními komisemi ISO a ECMA. Využívá se hlavně k tvorbě konzolových, webových a formulářových aplikací.

Jako vývojové prostředí pro tento jazyk jsem využil Microsoft Visual Studio 2019 [19].

2.3 Blender

Blender [11] je svobodný software pro modelování a vykreslování trojrozměrné počítačové grafiky a animací. Program je multiplatformní hlavně díky knihovně OpenGL, kterou používá na vykreslení svého rozhraní. Blender je zdarma i pro komerční využití pod licencí GNU GPL. V Blenderu se nejčastěji modelují objekty, následně se texturují a nakonec animují. Do verze 2.80 dokonce obsahoval vestavěný herní engine s podporou skriptování pomocí programovacího jazyka Python.

Dalším zástupcem modelovacího softwaru, nad kterým jsem přemýšlel, byl program Milkshape 3D [12], který má jednodušší grafické rozhraní i způsob modelování. Bohužel poslední verze vyšla v roce 2011 a Unity Engine má problém s modely a animacemi. Blender je neustále aktualizován a podpora je stoprocentní.

2.4 Photoshop

Photoshop [13] je bitmapový grafický editor pro tvorbu a úpravu bitmapové grafiky. Software vyvinula společnost Adobe Systems v roce 1990. Photoshop jsem použil na tvorbu pozadí v menu a ikony hry.

Navíc jsem využíval program Gimp [14] pro tvorbu paletové textury a program Krita [15] pro kreslení textur povrchů.

2.5 Bosca Ceoil

Bosca Ceoil [16] je program pro tvorbu vlastní hudby. Vytvořil ji Terry Cavanagh v roce 2013. V roce 2015 vydal zatím nejnovější verzi V2, která má hlavně aktualizované uživatelské rozhraní a webovou verzi. Jeho cílem bylo vytvořit program pro tvorbu hudby, který by dokázal ovládat každý a byl přitom úplně zadarmo. Samotný program obsahuje přes 100 nástrojů, ze kterých lze vytvářet hudbu. Nevýhodou je, že se program neptá na ukončení. Zmáčknutí klávesy escape nebo křížku tak znamená ztrátu rozdělané práce. Proto je nutné často projekt ukládat. Boscu Ceoil jsem si vybral kvůli své jednoduchosti.

2.6 Audacity

Audacity [17] je svobodný multiplatformní editor zvuku. Program vytvořil Dominik Mazzoni v květnu roku 2000. Software slouží pro přímé nahrávání zvuku, editaci zvukových souborů a efektování zvuku. Audacity jsem použil hlavně na normalizaci hlasitostí zvuků a převod zvuku ze stera do mona.

Navíc jsem využil i pokročilejší program Vegas Pro 17 [18] pro dabování a zvukovou úpravu replik.

3 Základy Unity Engine

3.1 Co je to Asset

Asset [20] je reprezentace jakékoliv položky, kterou můžeme použít ve hře. Assety jsou nejčastěji vytvořeny mimo Unity Engine. Může se jednat o 3D model, zvukový soubor, obrázek nebo jiné typy které Unity podporuje. Objekty ve scéně jsou také assety. Například pro import modelů z Blenderu podporuje Unity formát FBX.

3.2 Uživatelské rozhraní

Po vytvoření projektu se nám zapne Unity Engine s následujícím uživatelským rozhraním [21].



Obrázek 6: Unity Engine - Uživatelské rozhraní

3.2.1 Panel nástrojů

Ve vrchní části se nachází panel nástrojů, kde zleva jsou základní transformační nástroje jako je posun, zvětšení nebo rotace. Uprostřed je tlačítko pro spuštění hry v editoru a napravo jsou online věci jako je např. přihlášení k účtu.

3.2.2 Scéna

Je interaktivní okno do světa, který tvoříme. Po kliknutí na objekt ve scéně ho označíme a poté s ním můžeme manipulovat a měnit ho.

3.2.3 Okno hierarchie

Okno hierarchie je textová reprezentace všech objektů ve scéně. Každý objekt ve scéně má v hierarchii záznam, a proto jsou tyto dvě okna propojena. Oproti scéně se zde ale objevuje kompletní struktura objektů. To znamená, že si lze ve stromovém pohledu rozkliknout jejich potomky a upravovat například jen ty.

3.2.4 Okno inspektoru

Toto okno nám umožňuje upravovat vlastnosti právě vybraných objektů. Pro každý objekt je okno rozdílné, jelikož každý objekt obsahuje jiné skripty, zvuky a jiné komponenty.

3.2.5 Okno projektu

Okno projektu zobrazuje knihovnu assetů, které jsou dostupné pro náš projekt. Když do projektu importujeme nový asset, tak se zobrazí zde.

3.3 Scény

Scény [22] obsahují prostředí a menu naší hry. Každá scéna se dá chápat jako samostatný level. V každé scéně můžeme umisťovat objekty, zvuky, kamery nebo světla.

3.4 GameObject

GameObject [23], neboli herní objekt, je základním konceptem v Unity Enginu. Každý objekt ve hře je GameObject. Od kamer, zvuků, modelů až po speciální efekty. Samotné GameObjecty toho ale samy moc neumí. Jejich reálnou funkcionalitu zajišťují komponenty, které k nim přiřazujeme přes Okno inspektoru. Chovají se tedy jako nějaké kontejnery do kterých sypeme funkce. Například, aby byl GameObject pevný a dalo se po něm chodit, stačí, když mu přiřadíme komponentu BoxCollider, která nám tuto funkcionalitu zajistí. Komponenty nám tedy obstarávají chování GameObjectu. Komponentů je mnoho druhů. Tím nejdůležitějším jsou naše skripty. Dalšími podstatnými je transformační komponenta, AudioSource komponenta pro vložení zvuku nebo komponenty pro řešení kolizí. Na samotném vrchu inspektoru můžeme odškrtnout Game Object vypnout. Tím tak zmizí ze scény a v okně hierarchie se nám zprůhlední.

3.5 Prefab

Prefab [24] nám umožňuje uložit si GameObject se všemi komponentami, hodnotami a potomkami. Prefab se pak chová jako šablona, ze které můžeme ve scéně vytvořit jeho novou instanci. Prefaby můžeme do sebe vnořovat. Když Prefab změním, změním i všechny jeho instance ve scéně. Pokud chceme nějaký Prefab

ve scéně osamostatnit, stačí na něj v oknu hierarchie kliknout pravým tlačítkem a vybrat *Unpack Prefab Completely*.

3.6 Skripty

Skripty [26] zajišťují chování GameObjectů. Po vytvoření skriptu a otevřením v editoru kódu se nám ukáže tento předpis.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class NovySkript : MonoBehaviour
6 {
7     void Start ()
8     {
9
10    }
11
12    void Update ()
13    {
14
15    }
16 }
```

Zdrojový kód 1: Kód po vytvoření skriptu

MonoBehaviour je základní třída z které je odvozen každý skript. Metoda Start se spustí při prvním snímku hry a metoda Update se používá každý snímek hry. To znamená, že jestli nám hra jede na 60 snímků za vteřinu, tak se Update spustí 60x. Proto si zde musíme dávat pozor, abychom neprováděli nějaký složitý výpočet, jinak bychom tím rapidně snížili výkon.

3.7 Ukládání dat

Ukládání dat lze učinit dvěma způsoby.

- První možností je PlayerPrefs [25], což je třída, díky které můžeme ukládat a načítat data. Data se ukládají do systémového registru. Tato metoda je doporučena pouze pro menší data, jelikož můžeme ukládat pouze proměnné typu string, float a int.
- Druhou možností je načítat externí XML nebo binární soubor. Zde můžeme ukládat cokoliv a v zásadě u binárního souboru můžeme mít i jakoukoliv strukturu.

V Hirokovi jsem použil první možnost.

3.8 Animation Controller

Animation Controller [27] je asset, který se přiřazuje GameObjectu pro fungování animací. Kontroler je zobrazen jako automat, který přechází mezi stavy pomocí různých proměnných, které můžeme nastavovat ve skriptech. Automat konstruujeme z jednotlivých animací, kterým následně určujeme přechody. Kontroler obsahuje i vrstvy, kdy můžeme animace aplikovat na různé části těla. Můžeme tak mít například vrstvu pro nohy a zároveň vrstvu pro ruce, každou s jiným automatem.

3.9 Audio

Pro fungování zvuku musíme nějakému GameObjectu přiřadit komponentu AudioSource [28] a následně s ním spárovat samotný zvuk. AudioSource je možno následně upravovat. Jsou zde možnosti jako přehrát zvuk ve smyčce, vzdálenost slyšitelnosti nebo aplikace různých audio efektů. Zvuky je pak možné spouštět přes skript.

3.10 Materiály, Textury a Shadery

Shadery jsou skripty, které obsahují matematické výpočty a algoritmy pro výpočet barvy každého vykresleného pixelu založené na světle a typu materiálu. Shadery se v Unity Enginu zapisují pomocí varianty jazyku HLSL nebo od verze Unity Enginu 2019.3 přes Shader Graph, což je mnohem jednodušší způsob.

Díky materiálům [29] definujeme, jak se mají vykreslit povrchy včetně přiložené textury, barevných odstínů a dalších. Možnosti materiálu závisí na shaderu.

Textura je bitmapový obrázek. Materiál může obsahovat referenci na texturu, a stejně tak shader materiálu může používat texturu, zatímco počítá barvu GameObjectu.

Na model se umísťuje textura tak, že vytvoříme nový materiál, do kolonky *albedo* v inspektoru umístíme texturu a případně zvolíme jiný shader. Výsledný materiál přetáhneme z okna projektu na náš model.

4 Programátorská příručka

4.1 Scripty

V této podkapitole popíši většinu skriptů mé hry.

4.1.1 Hráčovy skripty

Základním skriptem pro fungování hráče je **PlayerController**, který díky *Update* metodě zajišťuje pohyb a na něj závislých zapínání animací hratelné postavy. Dále se zde nachází metoda *KnocBack*, která postavu odhodí dozadu, pokud dojde k poškození a metodu *Jumper*, která vymrští hráče do vzduchu pokud hráč skočí na *Jumper*.

Dalším důležitým skriptem je **CombatController**, který zajišťuje souborový systém. V metodě *Update* se postupně střídají animace třech útoků pomocí triggerů: *1utok*, *2utok* a *3utok*. Pokud je mezi kliky pro útok více, než sekundová prodleva a nejsme na začátku kombi, tak se kombo zresetuje pomocí animačního triggeru *resetUtok*. Metoda čeká i na klik druhého tlačítka na střílení kouzla, kde po jeho vystřelení následuje časovač *castReset* nastavený na 4 sekundy, kdy nemůžeme střílet. To zajišťuje, aby hráč nemohl střílet neustále.

Dalším skriptem je **GameManager**, který zajišťuje obecné fungování hry. Metody *AddDuhovka*, *AddZlatka*, *AddSrdicko*, *AddOziveni* započítávají nově nabitě předměty a také je zapisují do uživatelského rozhraní. Metoda *minusZivoty* na druhou stranu odečítá počet životů hráče a zároveň pokud hráčovy životy klesnou pod jeden, tak spustí metodu *Respawn*, která hráče přesune k poslednímu kontrolnímu bodu. Dále jsou v tomto skriptu už jen metody na zobrazení textu v uživatelské rozhraní po sesbírání bonusů nebo disketky.

Skript **PlayerAnimatorController** slouží pro spouštění zvuků hráčových kroků a útoku.

4.1.2 Nepřátelské skripty

Každý nepřítel má svůj AI skript. Výbušňák má **VybusnakAI**, kde v metodě *Update* pomocí *NavMeshAgent* vypočítává vzdálenost a směr od hráče pomocí metody *SetDestination*. Když hráč dojde do vzdálenosti 50, tak Výbušňák po něm půjde a díky *NavMeshAgentovi* bude směřem k hráčovi obcházet překážky. Pokud je vzdálenost od hráče větší než 100, tak Výbušňák hráče přestane pronásledovat a vrátí na svou původní pozici. V metodě *OnTriggerEnter* jsou pak reakce na hráčovy útoky a hráče samotného. U Výbušňáka je to jednoduché, protože při jakémkoli kontaktu s hráčem vybuchne a tím poškodí jak jeho, tak ostatní nepřátele. K tomu slouží skript **ZasahVybuchem**. Ostatní nepřátele mají podobně fungující skript. **RohacAI** pro změnu reaguje na útoky hráče, kdy po dotyku s hráčovou rukou se spustí metoda *ubraní*, která ubere Roháčovi životy a zároveň spustí animaci pomocí triggeru *ubrani*. Po ubrání všech životů Roháč spustí efekt pro smrt. Poslední běžný nepřítel má skript **StrilecAI**, který má rozdíl v tom,

že nikdy sám nejde až k hráčovi. Drží si od něj odstup a střílí na něj, když je pod určitým úhlem. Umělá inteligence hlavního nepřítele hry **BossAI** je nejjednodušší, protože za hráčem nejde. V metodě *Update* pouze stojí na místě a střílí na něj. Nepřátelé, kteří umí chodit mají navíc skript **StrilecAnimatorController**, **RohacAnimatorController** nebo **VybusnakAnimatorController**, který pomocí metody *StrilecFootstepSound*, *RohacFootstepSound*, *VybusnakFootstepSound* spustí zvuk kroku, jehož hlasitost se odvíjí od vzdálenosti od hráče. Roháč má i skript **RohacSpicka**, který je umístěn na jeho rohu. Po kontaktu rohu s hráčem se hráčovi ubere jeden život.

4.1.3 Level a menu skripty

Skript **LevelEnd** uloží na disk statistiky nyní dokončeného levelu. Následně skript **LevelEndScreen** tyto uložené informace porovná se statistikami skutečnými a uloží do nich lepší výsledek. Na obrazovku se však vypíší statistiky z právě dokončeného levelu. Skript **LevelLoader** zajišťuje načítání levelů a **LevelLocker** nás nepustí do levelů, které ještě nemáme odemčené. **LoadLevel** zajišťuje, aby po načtení levelu nebyl vidět kurzor myši a hra nebyla zapauzována. Skript **MenuPauza** spouští pauzovací menu a zároveň obsahuje metody pro jednotlivé položky menu. Díky skriptu **MenuZpet** můžeme stiskem klávesy *escape* nebo *B* na ovladači se vracet zpět v menu. Největší skript **SettingApply** obsluhuje všechna nastavení i tlačítka v menu. V metodě *Start* se při zapnutí hry načtou všechny nastavení i herní postup. Pokud dojde k nějaké změně v nastavení, tak se spustí metody *Set*. Například metoda *SetResolution* uloží nově zvolené rozlišení a použije ho. Pomocí metod *BuyUpgrade* se ukládají vylepšení a metoda *KonecHry* ukončí hru.

Z konkrétních levelů by stál za zmínku skript **SachyLogika**, který kontroluje kompletní logiku třetího levelu se šachy. To znamená objevování šachů a nepřátel, kteří se náhodně objevují na jedné ze 4 pozic pomocí metody *nahodnySpawn*. Jednotlivé fáze jsou postupně pouštěny podle proměnných faze. S předchozím skriptem souvisí **SachyPozice**, který kontroluje, jestli byly šachy dotlačeny na správnou pozici. Tuto informaci pak dají vědět skriptu **SachyLogika**, který podle tohoto koná dál.

4.1.4 Sběratelské předměty skripty

Skripty **DuhovkaPickup**, **ZlatkaPickup**, **SrdickoPickpu** a **OziveniPickup** zajišťují sebrání předmětu, vytvoření efektu a zároveň připsání skóre a to přes metodu *OnTriggerEnter*.

Se sbíráním předmětů souvisí i ničení beden a sudů, ve kterých jsou ukryty. Ničení podporuje skript **DestroyBox**, který po kolizi s hráčovým útokem vyvolá efekt zničení krabice.

Posledním sběratelským předmětem je ukládací disketa, která obsahuje skript **Checkpoint**, jež zajišťuje ukládání a zároveň animaci otáčení.

4.1.5 Ostatní skripty

Z ostatních skriptů by stál za zmínku **SpinAround**, který rotuje předmět nebo **PortToStart**, který hráče teleportuje na začátek levelu. Dále **PlatformMove** pomocí metody *FixedUpdate* pohybuje v cyklu platformami tam a zpátky. Naopak skript **PadajiciPodlaha** po kontaktu s hráčem nechá platformu spadnout. Skript **Napoveda** zobrazí nápovědu a spustí dabovanou repliku. **MoveTo** posune platformu k danému objektu a **Chrlic** hráčovi ubere život při jeho kontaktu. Díky **BoxPush** může hráč posouvat dané objekty. **ButtonPush** spustí světlo nad bránou a následně skript **GateOpen** bránu otevře, pokud jsou všechna světla rozsvícena.

4.2 Animační kontrolery

V této podkapitole se podíváme na animační kontrolery a jejich automaty.

4.2.1 HirokAnimatorController

Tento animační kontroler se skládá ze dvou vrstev. První vrstva se stará o animaci celého těla a druhá vrstva o ruce. Je to z toho důvodu, že když postava zaútočí a zároveň chodí, tak udělá dvě animace zároveň.

Pro vrstvu celého těla je výchozí stav *Idle*, kdy postava v klidu stojí. Pokud se pohneme přejde do stavu *Run*. Při skákání se přejde do stavu *Jump* a pokud ve vzduchu skočíme podruhé, tak *Double Jump*. Po zbytek doby ve vzduchu zůstává automat ve stavu *Jump*. Pokud postava chodí dozadu, přepne se automat na stav *Backward* a pokud se otáčí na místě, tak na stav *Přešlapování*.

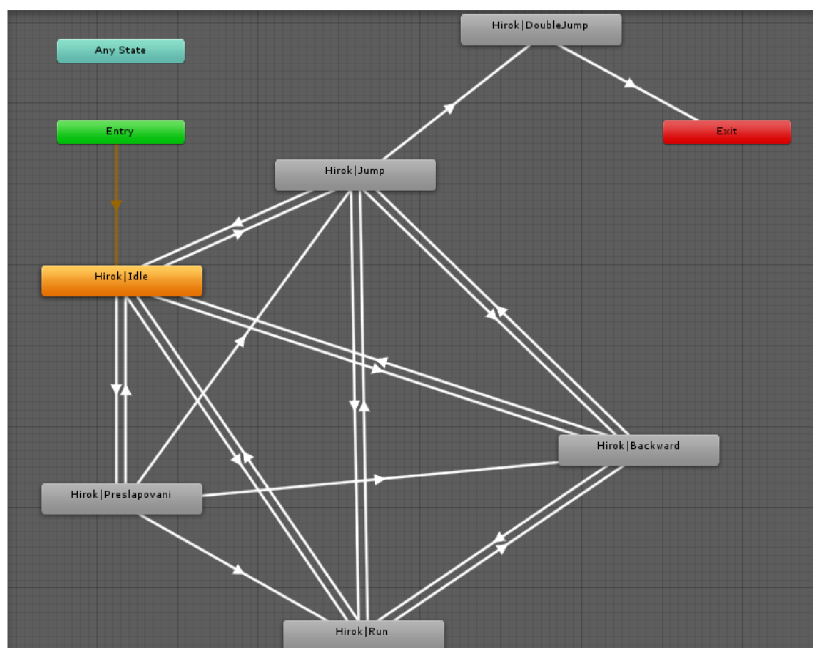
Ve vrstvě pro útok je základní stav *ten*, kdy není zapnuta žádná animace, a používá se ta ze základní vrstvy. Při útoku se střídají tři stavy *1hit*, *2hit* a *3hit*, které zajišťují trojitě kombo v závislosti na tom, jestli pořád klikáme na útok. Dalším možným stavem je *Shoot*, kdy postava vyvolá kouzlo.

4.2.2 Controllery nepřátel

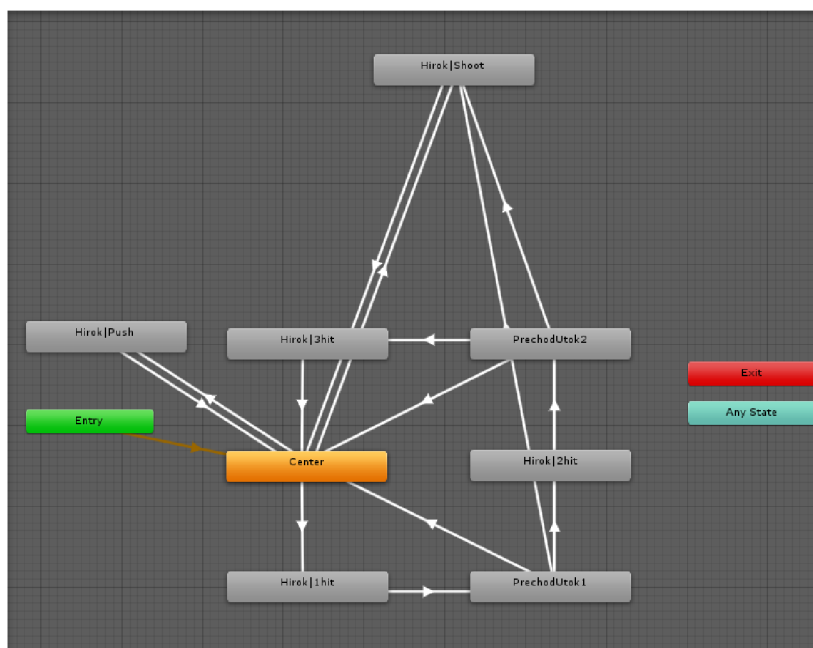
Animační kontrolery nepřátel jsou mnohem jednodušší. Výchozí stav je *Idle*, což je stav kdy stojí nepřítel v klidu. Po spatření hráče, když je ve vzdálenosti 50, se přesune do stavu *Run*. Pokud dostane poškození od hráče, přejde do stavu *Ubrani* a při útoku do stavu *Utok*. Tyto stavy mají nepřátelé Roháč a Stříleč. Výbušňák má jen stavy *Idle* a *Run*, protože při kontaktu s hráčem vybuchne, takže jiné stavy nepotřebuje. Boss má naopak jen animaci pro *Idle* a *Utok*, protože celou dobu stojí na místě.

4.2.3 Ostatní animační kontrolery

Ostatní animační kontrolery jako například **JumperAnimatorController** nebo **DuhovkaAnimator** mají jen jeden stav, který se ve smyčce opakuje. U Jumperu jeto neustálá změna velikosti a u duhovky je to otáčení. Později v vývoji jsem

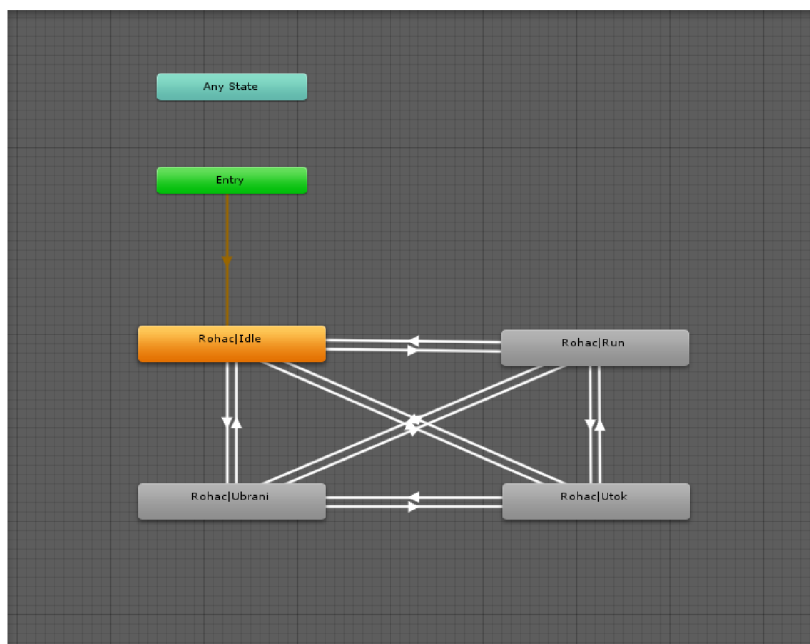


Obrázek 7: Automat pro animace těla hlavní postavy



Obrázek 8: Automat pro animace rukou hlavní postavy

přišel na to, že tyto jednoduché operace jako je otáčení, posouvání a změnu velikosti není nutné animovat ručně, ale je to možno udělat přes skript programově. Proto jsou tyto kontrolery spíše takovým pohrobkem mé předchozí nezkušenosti.



Obrázek 9: Automat pro animace nepřítele Roháče

4.3 Scény

Scén se ve hře nachází celkově 7. Šest pro levely a jedna pro hlavní menu.

4.4 Zvuky

Všechny zvuky ve hře jsem stáhl z webové stránky <https://freesound.org/>. Na tuto stránku umisťují uživatelé své nahrané zvuky zdarma ke stažení.

4.5 Hudba

Ve hře se nachází 7 skladeb. Hudbu jsem složil v programu Bosca Ceoil. Žánrem jde většinou o ambient, který je podle mě na skládání nejjednodušší.

4.6 Dabing

Ve hře se nachází dabing při popisu levelu před jeho začátkem a během nápovědy ve hře. Všechny repliky byly nadabovány na kondenzátorový USB mikrofon.

4.7 Shadery

Shaderů moc ve hře nepoužívám. Jediné shadery jsou pro vodu, lávu a opakující se texturu. V základu totiž Unity Engine textury roztahuje.

4.8 Možná vylepšení

Hra rozhodně není dokonalá a dala by se určitě o nějaké prvky obohatit. Nejvíce chybí lokalizace do cizích jazyků, hlavně do angličtiny. U tohoto prvku by bylo ale nutné překopat všechny řetězce ve hře a přiřadit je do externího souboru, ze kterého by hra tyto textové řetězce četla.

Dalším možným vylepšením by mohlo být nastavitelné ovládání, protože je teď pevně nastavené.

Posledním vylepšením, které je ale ryze subjektivní, je celkové zkrášlení hry, protože grafický styl má rozhodně prostor k vylepšení.

5 Uživatelská příručka

5.1 Požadavky

Minimální hardwarové a systémové požadavky [30] pro spuštění hry jsou následující.

- Operační systém Windows 7 SP1 a novější
- Procesor podporující SSE2 instrukční sadu
- Grafická karta kompatibilní s DX10 (shader model 4.0)

Tyto hardwarové nároky jsou z oficiální dokumentace. Testování však probíhalo na níže uvedené sestavě a hra si držela v rozlišení 1920x1080 na nejvyšší grafické nastavení se zapnutou vertikální synchronizací stabilních 60 snímků za sekundu.

- Operační systém - Windows 10 64 bit
- Procesor - Intel I5-3570 3.40 GHz
- Operační paměť - 16 GB DDR3 1333 MHz
- Grafická karta - Nvidia RTX 3060 12 GB
- Pevný disk - SSD 512 GB

5.2 Instalace a odinstalace

Instalace je velmi jednoduchá. Stačí otevřít archív Hirok.zip, který se nachází ve složce /bin a následně jeho obsah rozbalit kamkoliv na disk v PC.

Pro odinstalaci stačí smazat složku, ve které se nachází nainstalovaná hra. Pokud chceme smazat současný postup ve hře, tak stačí zvolit možnost RESET v nastavení hry. Jestli ale navíc chceme smazat i veškeré další nastavení, jako je rozlišení nebo nastavená hlasitost, tak je nutné smazat složku v registrech na tomto umístění:

HKEY_CURRENT_USER\Software\AlbKauSoft\Hirok

5.3 Ovládání

Ovládat se hra dá jak na klávesnici, tak na ovladači. V obou případech jsou pak dvě možnosti, jak hru ovládat. A to buď základní schéma nebo alternativní. Ovladač byl do hry přidán na poslední chvíli, proto jsou animace u pohybu trochu neodladěné. Jediný problém u ovladače nastane při funkci resetu hry, kdy je nutné napsat RESET do textového pole. Tuto funkci musí zastat klávesnice. Testovány byly ovladače pro konzoli Xbox 360 a Xbox Series. Níže uvádím primární ovládací schéma pro klávesnici a ovladač. Ovládání ve hře nejde přenastavit, ale pro některé akce existuje i alternativní klávesa.

- Pro pohyb postavy se používá standardní kombinace kláves WASD a u ovladače levá analogová páčka
- Kamera se ovládá buď myší nebo pravou analogovou páčkou
- Pro skok se využívá klávesa Mezerník u klávesnice nebo tlačítko A u ovladače
- Pro úskok nebo plachtění klávesa Levý Shift nebo tlačítkem RB, či LB u ovladače
- Útočí se levým tlačítkem myši u klávesnice nebo tlačítkem RT u ovladače
- Kouzlo se sesílá pravým tlačítkem myši nebo tlačítkem LT u ovladače
- V menu se dá pohybovat pomocí šipek nebo WASD, potvrzuje se tlačítkem Enter, pro návrat zpět pak klávesa Escape. U ovladače je to pak levá analogová páčka nebo DPAD. Tlačítko A pro potvrzení a tlačítko B pro návrat.

5.4 Hlavní menu

Obrazovka hlavního menu se skládá z položek:

- Hrát - Obsahuje seznam levelů, které lze spustit. Navíc je zde i podmenu s vylepšeními.
- Nastavení - Zahrnuje nastavení pro ovládání, video, audio nebo reset herního postupu
- Statistiky - Zobrazí celkové statistiky počtu získaných zlatek, duhovek a nejlepší časy pokoření jednotlivých levelů
- Autoři - Zobrazí autora hry
- Konec - Ukončí hru

5.4.1 Hrát

V tomto menu se nachází jednotlivé odemčené levely. Po kliknutí na určitý level se otevře stránka s názvem a čteným popisem levelu. Tento popis je nadabován mnou samotným. Z tohoto menu je možné level spustit nebo se vrátit zpět na výběr levelů. Ve výběru levelů je ještě položka vylepšení, kde si může hráč koupit za nasbírané předměty až tři vylepšení usnadňující hratelnost.



Obrázek 10: Hlavní menu

5.4.2 Nastavení

Nastavení obsahuje následující položky:

- Ovládání - Zobrazí ovládání hry, jak na klávesnici, tak na ovladači. Navíc je možné nastavit citlivost kamery, invertování kamery, zobrazení nápovědy a zapnutí dětského režimu.
- Video - Zde lze nastavit rozlišení hry, zobrazení hry, vertikální synchronizaci, kvalitu grafiky, anisotropní filtr textur a vyhlazování hran pomocí metody MSAA (Multisample anti-aliasing).
- Audio - Zde lze nastavit celkovou hlasitost hry nebo zapnout dabing. Dabing je ve výchozím nastavení zapnutý.
- Reset - Po napsání slova RESET a potvrzení se resetují všechny odemčené levely, vylepšení a statistiky. Tedy veškerý pokořený postup ve hře.

5.4.3 Statistiky

Ve statistikách lze sledovat současný postup hrou. Zobrazuje počet sesbíraných předmětů v jednotlivých levelech a časy jejich dokončení. Je zde i karta s celkovými statistikami, které obsahují i počet pokořených levelů a odemčených vylepšení.

5.4.4 Nápověda

Nápověda se ve hře zobrazí jako trojrozměrný točící se otazník. Po vstoupení do něj se nám zobrazí nápověda související se současným segmentem hry. Pokud je zapnutý dabing, tak je nápověda dabovaná. Ve výchozím nastavení je nápověda zapnutá.

5.4.5 Dětský režim

V dětském režimu se zvýší celkový počet oživení o 90. Tento režim je mířen hlavně na nehráče, jelikož s takovým počtem oživení je téměř nemožné nedokončit hru. Ve výchozím nastavení je dětský režim vypnutý.

5.4.6 Pauzovací menu

Do pauzovacího menu se lze dostat ze hry stiskem klávesy Escape nebo Start na ovladači. Pauzovací menu pozastaví hru. Oproti plnohodnotnému hlavnímu menu obsahuje ořezanější možnosti a nelze nic nastavovat. Obsahuje následující položky:

- Pokračovat - Pokračuje v rozehrané hře.
- Restart - Restartuje současně rozehraný level.
- Ovládání - Zobrazí ovládání hry.
- Menu - Navrátí se do hlavního menu.

5.5 Příběh

Příběh pojednává o hlavní postavě Hirokovi, který po světě ztratil svých šest katan. Chodí proto po světě a chce je získat zpět. Je to ale sběratel a tak je nikdy v boji nepoužije.

5.6 Uživatelské rozhraní

Uživatelské rozhraní je prosté. V levém horním rohu je počet získaných zlatek a duhovek a na pravé horní straně obrazovky se nachází počet srdíček, tedy životů a pod ním počet oživení.



Obrázek 11: Uživatelské rozhraní

5.7 Pohybový systém

Pohyb funguje na klávesách WASD. Kamera se otáčí myší. Na rozdíl od většiny současných her se postava otáčí ve směru kamery, takže nikdy nejde vidět zepředu. To má tu výhodu, že je pak pohyb přesnější, než když se kamera pohybuje nezávisle na směru pohybu postavy. Ve hře funguje jak jednoduché, tak dvojitě skákání při dvojitém kliku na klávesu Mezerník. Dále je zde systém uhýbání, kdy se postava doslova teleportuje o pár metrů do směru ve který vede pohyb. Aktivuje se klávesou Levý Shift. U tohoto pohybu je čtyř sekundové obnovení, aby se tohoto pohybu nedalo zneužívat. Zejména se hodí při boji s nepřáteli, kdy lze jejím útoků či náběhům uhnout. Posledním pohybem je plachtění, kdy hráč doletí dále. Aktivuje se stejně jako úhyb klávesou Levý Shift, ale funguje pouze ve vzduchu, kdy se hráč přemění na modrý oblak prachu. Další z pohybů, který zde je, je posouvání předmětů. Jedná se o kovové bedny a šachy v Levelu 3. Stačí jít naproti a postava automaticky zvedne ruce a začne tlačit.

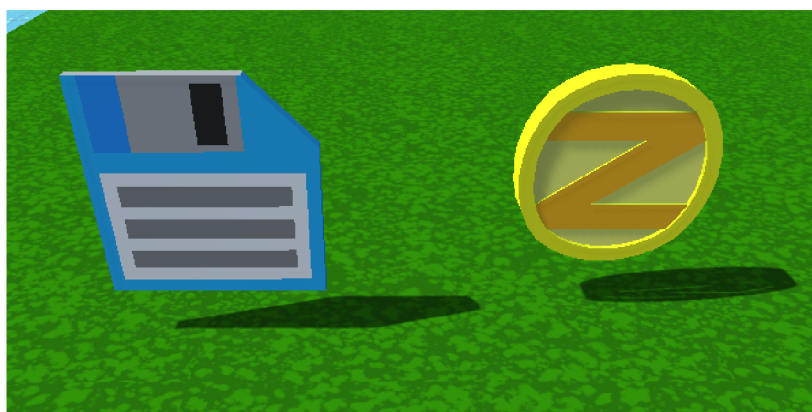
5.8 Soubojový systém

Soubojový systém je tvořen dvěma věcmi. Za prvé může postava bojovat rukama, kde funguje tzv. trojitě kombo. Po klikání může mít až 3 animace. Druhým útokem je kouzlo které hráč může vystřelit stisknutím pravého tlačítka myši. Kouzlo trvá obnovit 4 vteřiny.

5.9 Sběratelské předměty

Sběratelských předmětů jsou čtyři druhy:

- Zlatky - Zlatky se vyskytují v dřevěných bednách nebo ve volném prostranství. Za 6 zlatek se hráčovi připíše bonusový život. Za zlatky lze také nakupovat vylepšení z menu hry. Tento předmět je nejběžnější.
- Duhovky - Duhovky se vyskytují v sudech nebo ve volném prostranství. Za 4 duhovky se hráčovi připíše bonusové oživení. Za duhovky lze také nakupovat vylepšení z menu hry.
- Srdíčka - Srdíčko připíše hráčovi život navíc.
- Oživení - Oživení připíše hráčovi oživení navíc. Tento předmět je nejvzácnější.
- Disketa - Uloží hru na současné pozici. Po smrti se postava objeví na místě diskety.



Obrázek 12: Disketa a zlatka

5.10 Systém vylepšování

V menu vylepšení si lze vylepšit za 50 zlatek a 15 duhovek každé z vylepšení.

- Silnější kouzlo - Silnější kouzlo je a účinnější než kouzlo původní.
- Dva životy navíc - Hráč bude každý level začínat s 5 místo se 3 srdíčkama.
- Dva oživení navíc - Hráč bude začínat každý level s 5 místo se 3 oživeními.

5.11 Typy nepřátel

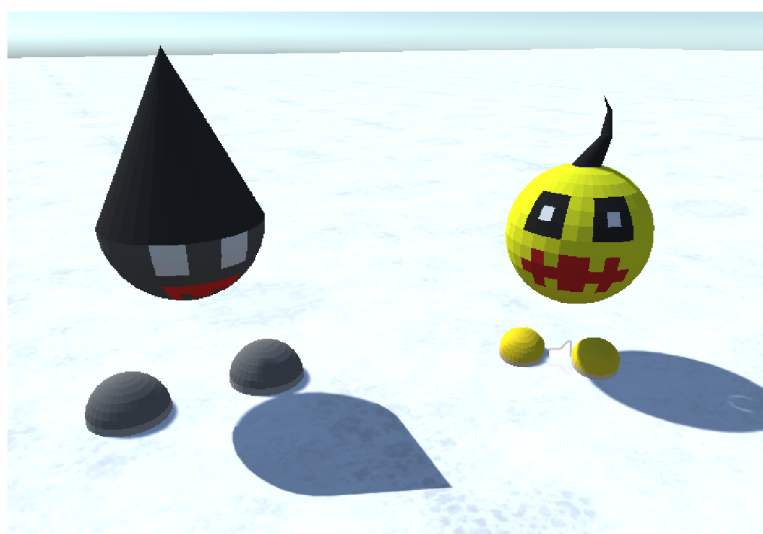
- Výbušňák - Je nepřítel který při kontaktu s hráčem vybuchne a tím mu ubere jeden život. Jedinou účinnou obranou je ho střelit kouzlem z dostatečné vzdálenosti.
- Roháč - Roháč je výdržný nepřítel útočící na blízko svým rohem.
- Střileč - Střileč je středně výdržný nepřítel střílející na hráče svá kouzla.
- Boss - Je finální nepřítel hry, na kterého je nutné použít speciální taktiku.

5.12 Levely

Levely se odemykají postupně.

- Level 1 - První level se odehrává na moři s ostrůvky.
- Level 2 - Druhý level se odehrává také na moři ale v zasněženém prostředí. Na začátku je brána, za kterou se skrývá katana. K otevření brány je třeba se vydat všemi čtyřmi směry, kde na konci je nášlapné tlačítko. Po nášlapnutí před branou zezelená indikátor. Po nášlapnutí na všechny 4 se brána otevře.

- Level 3 - Tento level se odehrává v prázdnosti na šachovém poli. Zde je nutné řídit se instrukcemi napsanými v prostoru. Ve stručnosti je třeba posouvat šachy na správné pozice a pak eliminovat nepřítele.
- Level 4 - Tento level se odehrává v nebi, kdy se stanou plošinkami karty.
- Level 5 - Tento level se odehrává v lávové jeskyni. Hráč musí z této jeskyně vyskákat nahoru.
- Level 6 - Poslední level je souboj s bossem, který se nachází uprostřed místnosti a podobně jako v druhém levelu, tak i zde je nutné skočit na čtyři spínače, aby byl boss zranitelný.



Obrázek 13: Výbušňák a roháč



Obrázek 14: Level 3

Závěr

V diplomové práci jsem si vyzkoušel kompletní proces tvorby 3D počítačové hry v jedné osobě. Zjistil jsem, že je to velmi náročný proces a jeden člověk musí zastávat funkce jak programátora, tak umělce. Bylo třeba programovat, modelovat herní objekty, kreslit textury, vytvářet hudbu a nakonec to dát celé dohromady.

Na hře je možné toho zlepšit mnoho. Určitě by se hodil komplexnější soubojový systém nebo více nepřátel. Do budoucna by se dala hra rozšířit o více lokalizací, přidat nastavitelné ovládání nebo ji kompletně zkrášlit.

Při tvorbě hry jsem získal obrovské množství zkušenosti v Unity Enginu, modelovacím programu Blender i hudebním editoru Bosca Ceoil. Tvorba mě velmi bavila a doufám, že si v budoucnu zase nějakou menší hru vytvořím. Výsledkem je tedy počítačová hra, která může aspoň na chvíli dost zabavit.

Conclusions

In my master thesis I tried the complete process of creating a 3D computer game in one person. I found that it is a very demanding process and one person has to act as both programmer and artist. It involved programming, modelling game objects, drawing textures, creating music and finally putting it all together.

There's a lot that can be improved on a game. A more complex combat system or more enemies would definitely be useful. In the future, the game could be expanded to include more localizations, add customizable controls, or beautify it completely.

I gained a huge amount of experience in Unity Engine, Blender modeling program and Bosca Ceoil music editor while creating the game. I had a lot of fun making it and hope to make a smaller game again in the future. The result is a computer game that can entertain at least for a while.

A Obsah příloženého CD/DVD

bin/

Spustitelná verze hry HIROK, která se nachází v archívu Hirok.zip.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archívu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové kódy hry HIROK se všemi potřebnými knihovnamy a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí hry.

readme.txt

Instrukce pro spuštění hry HIROK, včetně všech požadavků pro jeho bezproblémový provoz.

Literatura

- [1] Wikipedia: *Platform game* [online] [cit. 2022-04-30]. Dostupné z: https://en.wikipedia.org/wiki/Platform_game
- [2] Wikipedia: *Crash Bandicoot* [online] [cit. 2022-04-30]. Dostupné z: https://en.wikipedia.org/wiki/Crash_Bandicoot
- [3] Wikipedia: *Spyro* [online] [cit. 2022-04-30]. Dostupné z: <https://en.wikipedia.org/wiki/Spyro>
- [4] Wikipedia: *A Hat in Time*. [online] [cit. 2022-04-30]. Dostupné z: https://en.wikipedia.org/wiki/A_Hat_in_Time
- [5] Wikipedia: *Funk Unplugged* [online] [cit. 2022-04-30]. Dostupné z: https://store.steampowered.com/app/517100/Funk_Unplugged/
- [6] Wikipedia: *Unity (game engine)* [online] [cit. 2022-04-30]. Dostupné z: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [7] Wikipedia: *Unreal Engine* [online] [cit. 2022-04-30]. Dostupné z: https://en.wikipedia.org/wiki/Unreal_Engine
- [8] Wikipedia: *Godot (game engine)* [online] [cit. 2022-04-30]. Dostupné z: [https://en.wikipedia.org/wiki/Godot_\(game_engine\)](https://en.wikipedia.org/wiki/Godot_(game_engine))
- [9] Wikipedia: *C Sharp* [online] [cit. 2022-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/C_Sharp
- [10] Wikipedia: *Microsoft Visual Studio* [online] [cit. 2022-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Microsoft_Visual_Studio
- [11] Wikipedia: *Blender* [online] [cit. 2022-04-30]. Dostupné z: <https://cs.wikipedia.org/wiki/Blender>
- [12] Wikipedia: *Milkshape 3D* [online] [cit. 2022-04-30]. Dostupné z: https://en.wikipedia.org/wiki/Milkshape_3D
- [13] Wikipedia: *Adobe Photoshop* [online] [cit. 2022-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Adobe_Photoshop
- [14] Wikipedia: *GIMP* [online] [cit. 2022-04-30]. Dostupné z: <https://cs.wikipedia.org/wiki/GIMP>
- [15] Wikipedia: *Krita* [online] [cit. 2022-04-30]. Dostupné z: <https://cs.wikipedia.org/wiki/Krita>
- [16] CAVANAGH Terry: *Bosca Ceoil* [online] [cit. 2022-04-30]. Dostupné z: <https://bosceoil.net/>

- [17] Wikipedia: *Audacity* [online] [cit. 2022-04-30]. Dostupné z: <https://cs.wikipedia.org/wiki/Audacity>
- [18] Wikipedia: *Vegas Pro* [online] [cit. 2022-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Vegas_Pro
- [19] Wikipedia: *Visual Studio 2017* [online] [cit. 2022-04-30]. Dostupné z: https://en.wikipedia.org/wiki/Milkshape_3D
- [20] UNITY: Asset Workflow [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/AssetWorkflow.html>
- [21] UNITY: The Main Windows [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/UsingTheEditor.html>
- [22] UNITY: Scenes [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/CreatingScenes.html>
- [23] UNITY: GameObjects [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/GameObjects.html>
- [24] UNITY: Prefabs [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/Prefabs.html>
- [25] UNITY: PlayerPrefs [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/ScriptReference/PlayerPrefs.html>
- [26] UNITY: Scripting [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/ScriptingSection.html>
- [27] UNITY: Animator Controller [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/class-AnimatorController.html>
- [28] UNITY: Audio [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/class-AudioSource.html>
- [29] UNITY: Materials, Shaders & Textures [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/Shaders.html>
- [30] UNITY System requirements for Unity 2019.2 [online] [cit. 2021-07-31]. Dostupné z: <https://docs.unity3d.com/2019.2/Documentation/Manual/system-requirements.html>