Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

DIPLOMOVÁ PRÁCE

Generovaná 3D hra v Unity

2019

Bc. Patrik Szkandera

Vedoucí práce:
Mgr. Martin Trnečka, Ph.D.

Studijní obor: Aplikovaná Informatika,
prezenční forma

Bibliografické údaje

Autor:	Bc. Patrik Szkandera
Název práce:	Generovaná 3D hra v Unity
Typ práce:	diplomová práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2019
Studijní obor:	Aplikovaná Informatika, prezenční forma
Vedoucí práce:	Mgr. Martin Trnečka, Ph.D.
Počet stran:	38
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliograhic info

Author:	Bc. Patrik Szkandera
Title:	Generated 3D game in Unity
Thesis type:	master thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2019
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Martin Trnečka, Ph.D.
Page count:	38
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Práce pojednává o vývoji 3D hry v herním enginu Unity 3D s generováním herních úrovní a za použití vlastních grafických prvků. V práci je popsán postup tvorby 3D modelů a implementace vybraných herních mechanizmů. Na závěr práce obsahuje krátkou herní příručku, která přináší dodatečné informace užitečné pro hráče.

Synopsis

This thesis is about development of the 3D game in Unity 3D game engine with customly created 3D models and level generation. The work describes the process of creating 3D models and implementation of selected game mechanisms. It also contains a short game guide that provides additional useful information for players.

Klíčová slova: Unity 3D; Blender; 3D grafika; generované prostředí

Keywords: Unity 3D; Blender; 3D graphics; generated environment

Děkuji Martinu Trnečkovi za odborné vedení diplomové práce.

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

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Na tvorbě her povětšinou spolupracují početné týmy lidí, které pokrývají jednotlivé části vývoje - od grafiky a tvorby vizuálních efektů, přes programování až po zvukové efekty, hudbu a testování. Dále se pro každou část vývoje často využívá specializovaný software, který zrychluje vývoj. Licence takových programů se ale běžně pohybují v řádech stovek až tisíců dolarů.

Záměrem mé práce bylo zjistit, zda podobnou práci je schopen zastat jeden člověk a to pouze se softwarem, který je nabízen zdarma. Rozhodl jsem se tedy vytvořit vlastní 3D hru. Pro tvorbu modelů jsem zvolil open-source 3D modelovací software Blender [1]. Jako herní engine jsem si zvolil Unity 3D [2].

1.1 Inspirace a styl hry

Před návrhem samotné práce jsem si nejprve musel projít spoustu indie her[1], abych zjistil, jaký rozsah projektu je reálný a co vše si můžu do hry naplánovat pro pozdější implementaci. Co se týče herních mechanik, tak mou největší inspirací byly hry:

- Magicka [3],
- The Binding of Isaac: Rebirth [4],
- Kingdom: New Lands [5].

Mým plánem tedy bylo vytvořit 3D hru s rogue-like[2] prvky a zaměřenou na procházení bludišť plných nepřátel. Hlavní zbraní hráče měla být magie a kouzla s rozličnými efekty.

Co se týče grafické stránky hry, tak jsem se hlavně nechal inspirovat hrami, které využívají styl grafiky s nízkým počtem polygonů:

- For The King [6],
- Epistory - Typing Chronicles [7],
- Superhot [8].

[1] za indie jsou označovány hry, které byly vytvořeny malými týmy či jednotlivci, zkratka vychází z anglického independent.

[2] jedná se o jeden z nejstarších herních žánrů, který se zaměřuje na role-playing a charakterizuje jej hlavně procedurálně generované prostředí, permanentní smrt a tahový styl hry. Tento žánr byl pojmenován po první hře, která tyto prvky implementovala s názvem Rogue.

1.2 Učební zdroje

Protože jsem se v rámci této diplomové práce pustil do pro mě úplně nového odvětví, hledal jsem co možná nejlepší zdroje informací. Na internetu je samozřejmě spousta návodů, které pojednávají o vývoji her, nicméně se mi nepodařilo najít žádný ucelený, který by mě provedl celým procesem. Nakonec jsem se rozhodl absolvovat dva online kurzy v rámci studijní platformy Udemy. Konkrétně se jednalo o kurzy:

1. Complete Blender Creator: Learn 3D Modelling for Beginners [9],
2. Complete C# Unity Developer 3D: Learn to Code Making Games [10].

2 Použité technologie

Zaměřil jsem se na výběr takových technologií, které jsou zdarma k použití. Ve většině případů jsem byl mile překvapen širokou nabídkou kvalitního softwaru (jediná výjimka potvrzující pravidlo je odvětví 3D modelování) a moje finální volba byla pak většinou ovlivněna dodatečným studiem jednotlivých programů.

2.1 Blender

Jak již bylo zmíněno výše, tak v oboru tvorby 3D modelů není moc softwaru dostupného zdarma. Většina profesionálních grafiků pracuje v placeném softwaru, kde se ceny licencí mohou vyšplhat až do výše desítek tisíc korun [11]. Proto mě velmi mile překvapil open-source software Blender. Když jsem navíc zjistil, že pro tento software existuje spousta výukových materiálů a stále se rozrůstající komunita, která dál sponzoruje celý vývoj [12], byla volba jasná.

Díky této komunitě a skutečnosti, že Blender je customizovatelný pomocí vlastní API a jazyka Python, vznikl velký počet užitečných rozšířený, které různě usnadňují proces tvorby 3D modelů [13].

Dalším významným plusem tohoto softwaru je fakt, že je naprosto univerzální a lze v něm projít celým procesem tvorby modelu: od samotného modelování, přes tvorbu textury až po animaci [14]. Není tedy potřeba učit se pracovat s dalšími nástroji. I když pravdou je, že nástroje specializované na daný krok tvorby dělají tuto práci efektivněji, ale opět je zde velkým problémem cenovka.

Hlavní nevýhodou Blenderu pro mě byla jednoznačně velmi strmá křivka učení a s tím související neintuitivní uživatelské rozhraní. Nicméně tento problém byl za mě jednoznačně odstraněn v nové verzi Blenderu 2.8.

2.2 Unity 3D

V oblasti herních enginů byl už výběr technologií mnohem bohatší [15]. Avšak spousta z nich byla specificky zaměřená a jejich využití neskýtalo až tolik budoucích možností. Chtěl jsem totiž použít takový software, jehož využití bych našel i nad rámec této diplomové práce. Po průzkumu trhu a možností jednotlivých enginů, jsem se nakonec rozhodoval mezi dvěma největšími hráči na trhu:

- Unity 3D,
- Unreal Engine.

Výběr mezi těmito technologiemi pro mě nebyl vůbec snadný, protože s oběma je možné docílit velmi impozantních výsledků a také jsou oba velmi využívané.

Konečná volba padla na Unity 3D a to hlavně z osobních důvodů. V C#, ve kterém se v Unity 3D pracuje, jsem byl mnohem zběhlejší než v C++, které používá Unreal Engine. Navíc mi přišlo, že v době začátku mé diplomové práce, bylo Unity 3D mnohem přívětivější k začátečníkům a nabízelo širší podporu komunity.

V průběhu diplomové práce jsem se s Unity 3D velmi sžil a až na pár výjimek, byla práce s tímto enginem velmi zábavná. Jedinou významnější výtkou pro mě byl systém pro správu ovládání a pathfinding. Tyto moduly jsou rozpracovány a bohužel jim chybí i některé základní funkce (například není možné zjistit, jakou klávesu si uživatel nastavil na jakou akci). Nicméně dobrou zprávou je, že Unity prochází neustálým vývojem a například zmiňovaný systém pro správu ovládání má dostat významný update v příští verzi 2019.2 [16].

I přes rozsáhlé změny, které každá nová iterace tohoto softwaru přináší, je aktualizace vesměs velmi jednoduchá. Při práci na projektu jsem sledoval vývojový cyklus Unity 3D a postupně jsem instaloval všechny nové hlavní verze enginu. Vývoj hry byl započat v Unity 5.4 a hra pak postupně procházela úpravami tak, aby odpovídala verzím: 5.5, 2017.1, 2017.2, 2017.3, 2017.4, 2018.1, 2018.2 a 2018.3. K aktualizaci herního enginu mě především lákaly nové možnosti, jak s enginem pracovat a pak také zvědavost, jak moc bude každá aktualizace složitá a zda je to vůbec možné.

2.3 Krita

Dalším příjemným objevem v oblasti open-source software byla Krita [17]. Tento nástroj slouží pro tvorbu a úpravu 2D obrázků a animací a dle mého dalece překonává dlouho zaběhnutého open-source konkurenta GIMP. I když jsem byl opět nováčkem co se týče tvorby 2D obrázků, tak jsem byl schopen poměrně rychle vytvořit některé základní prvky grafického rozhraní.

Jelikož můj plán vytvořit si i kompletní grafické rozhraní vzal za své z časových a jiných důvodů, využil jsem Kritu hlavně pro tvorbu textur jednotlivých kouzel a jejich ikonek, úpravu některých obrázků vyrenderovaných v Blenderu a úpravu zakoupených grafických balíčků pro uživatelské rozhraní.

2.4 Audacity

S hudbou a zvuky jsem v rámci mé diplomové práce setkal minimálně. Využil jsem zde povětšinou zakoupených balíčků obsahujících hudbu a zvukové efekty. Má práce s Audacity jakožto open-source editorem digitálního zvuku byla tudíž velice omezená a tento software zde uvádím spíše pro úplnost [18].

3 Koncept hry

Jak již bylo nastíněno v úvodní kapitole, snažil jsem se před zahájením vývoje co nejlépe promyslet celkový koncept hry, abych se mohl ubírat jedním směrem a co nejvíce omezil případné změny. V průběhu návrhu jsem se snažil připravit si co nejrealističtější hranice herních mechanismů tak, abych vše stihl dokončit v řádném termínu. Bohužel se mi to nepovedlo u každé části hry na sto procent a tak některé mechanismy byli implementovány jen z části.

3.1 Hlavní myšlenka

Hráč se ocitá ve hře jako postava kouzelníka, který má za úkol odstranit všechny zjevující se nestvůry. Hráč se musí postupně probojovat skrze několik náhodně generovaných úrovní a zabít vše, co se mu postaví do cesty. Má to však jeden háček, pokud hráč v průběhu své cesty umře, nestvůry opět zaplaví všechny již dobyté úrovně a hráč musí začít zase od začátku.

Hráči v jeho cestě pomáhá systém kouzel, které může na nepřátele sesílat, vylepšení, které může za peníze zakoupit ve své základně a také předměty, které může náhodně nalézt u zabitých nepřátel.

Hra by měla být v prvotní herní fázi poměrně složitá a smrt hráče by neměla být ojedinělá. Hráč musí nasbírat dostatek peněz, kouzel a vylepšení, aby se mu podařilo propracovat dál a v konečném důsledku i dokončit hru.

Nakonec by hra měla implementovat i některé další rogue-like prvky, takže hráči jsou vysvětleny pouze základní mechanizmy hry a je na něm, aby zjistil, jak funguje vše ostatní.

3.2 Soubojový systém a systém kouzel

Jedním z hlavních prvků této hry je samozřejmě soubojový systém a systém kouzel. Celý koncept jsem se snažil vymyslet tak, aby byl z pohledu hráče co možná nejzajímavější a nabízel mu různé možnosti.

Základem jsou čtyři různé barvy, které může hráč u jednotlivých kouzel měnit. Každá barva mění účinek daného kouzla na jednotlivé nepřátele a také na něj samotného.

Dále má hráč k dispozici celkem jedenáct kouzel, každé s vlastním vizuálním efektem a účinkem. Tyto kouzla hráč postupně odemyká v průběhu hry a dostává tak k dispozici stále silnější arzenál.

Kombinaci barvy a kouzla si pak hráč ukládá do své výbavy pro aktuální použití. Počet možných kouzel, které si hráč může vzít sebou, závisí na zakoupených vylepšeních. Jak správa kouzel ve hře vypadá ukazuje obrázek 1.

Popis všech dostupných kouzel a účinky barev kouzel jsou k dispozici v herní příručce na konci tohoto textu.

Obrázek 1: Správa kouzel ve hře

3.3 Generované úrovně a nepřátelé

Další důležitou částí je generování samotného herního světa, tak aby každý nový start přinesl hráči unikátní herní průběh. V průběhu návrhu konceptu hry jsem neměl moc velkou představu o tom, jak takového generování docílit, jen jsem věděl, že pro potřebu hry je to velmi nezbytný mechanismus.

Samotné nepřátele jsem rozdělil do třech různých skupin: kostlivci, roboti a démoni. Jak už bylo zmíněno výše, na každou skupinu nepřátel účinkuje každá barva trošku jinak. Může se tedy stát, že jednou barvou bude hráč jednu skupinu nepřátel léčit a druhé způsobí fatální zranění.

Každá tato skupina obsahuje celkem čtyři různé nepřátele, z nichž každý vykazuje rozličné chování. Dále každý takový nepřítel měl mít až tři různé úrovně vylepšení. Pro většinu úrovní jsou připraveny modely, nicméně, kvůli složitosti celkové implementace, se do hry dostala od každého nepřítele pouze jedna úroveň.

3.4 Smrt hráče

Jak již bylo zmíněno výše, smrt hráče by neměla být ojedinělá. Hráč může samozřejmě umřít zapříčiněním nepřátel, ale může také umřít, pokud se po vyčištění dané úrovně rozhodne dál nepokračovat a bude se chtít vrátit na svou základnu.

Každá smrt hráče mu přináší určité nevýhody, kterých se lze částečně zbavit pomocí vylepšení nebo předmětů. Ve hře tedy může dojít i k bezvýchodné situaci, kdy je pro hráče lepším řešením začít hru úplně od začátku než v ní dál pokračovat.

3.5 Hráčské vylepšení a předměty

Posledním herním mechanismem jsou hráčská vylepšení a předměty, které mu pomáhají v jeho nesnadném úkolu. Důležitou roli zde hrají zlaté mince, které lze získat ze zabitých nepřátel a po dokončení úrovně v časovém limitu.

Jednotlivá vylepšení může hráč zakoupit ve své základně. K dispozici jsou celkem čtyři různá vylepšení a každé obsahuje celkem tři různé úrovně. Zakoupená vylepšení jsou v základně reprezentována pomocí nových místností, které rozšiřují hráčskou základnu. Obrázek 2 ukazuje jak vypadá konkrétní zakoupené vylepšení.

Obrázek 2: Vylepšení místnosti s výherními automaty na maximální úroveň

Předměty hráč získává primárně z poražených nepřátel a může je také zakoupit v základně. Hráč smí mít v jednu chvíli aktivované maximálně dva různé předměty. Účinky předmětů nejsou hráči předem známy a měl by je sám objevit v průběhu hraní. Celkově se ve hře vyskytuje dvanáct různých předmětů a ty se dělí do tří základních kategorií:

Užitečné předměty hráči pomáhají a různě zlepšují jeho statistiky nebo pravděpodobnosti úspěchu jednotlivých akcí.

Kosmetické předměty pouze upravují hráčský vzhled a žádné výhody hráči nepřináší.

Škodící předměty jsou typem předmětů, které hráče znevýhodňují.

Poslední možností, jak může hráč získat výhody je pomocí změny barvy roucha. Ve hře se nachází celkem šest různých barevných variant.

Podrobný popis vylepšení lze nalézt v herní příručce na konci této práce.

3.6 Příběh

Příběh této hry by měl být minimalistický a jeho hlavním úkolem je dát hráči konkrétní cíl a tím i smysl celé hry. Původně jsem měl v plánu do hry zakomponovat větší počet dialogů, které by postupně odkrývali další podrobnosti příběhu. Bohužel, to se mi z časových důvodů nepodařilo dokončit a tak ve hře zůstal jen základní nástin herního příběhu.

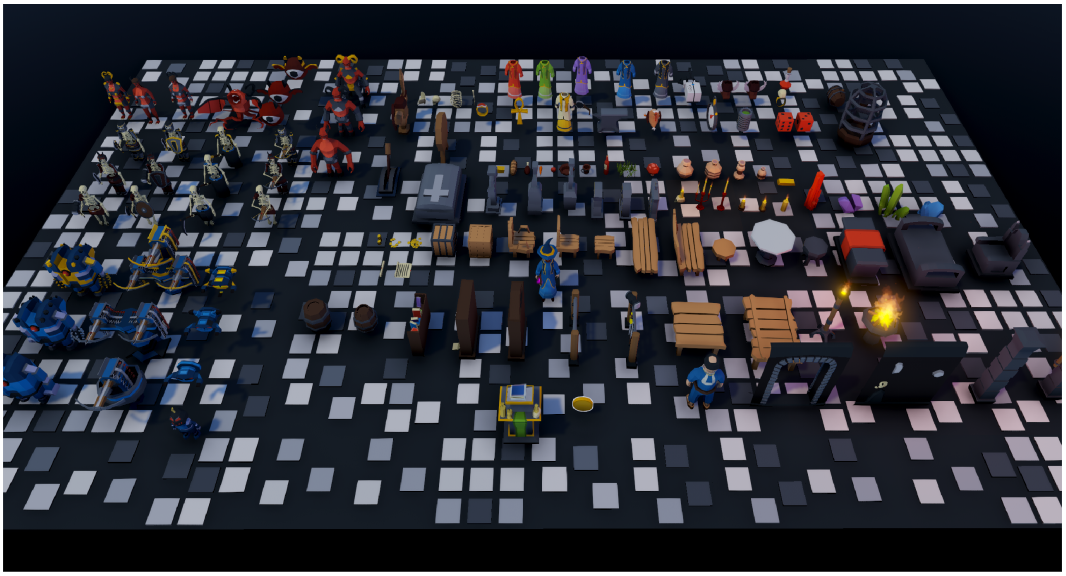
4 3D modely

V první fází jsem si vytvořil většinu potřebných 3D modelů. Díky tomu, že jsem měl před započetím této fáze celý koncept již z velké části připravený, podařilo se mi podchytit zhruba devadesát procent všech modelů. Proto jsem se mohl v pozdějších krocích už plně věnovat práci v Unity 3D a nebylo potřeba stále střídat pracovní prostředí.

Hru tvoří více než sto třicet modelů různých objektů. Z toho sedmdesát modelů má ještě svou zničenou verzi, která byla vždy vytvořena rozkouskováním originálního modelu. Takové modely se pak po interakci s hráčem či některými nepřáteli rozpadnou na kousky a tím vzniká iluze částečně zničitelného prostředí.

Dále hru tvoří dvanáct modelů různých nepřátel, včetně animací pohybu, útoků a zasažení hráčem. Pro většinu těchto nepřátel jsou dále připraveny další dvě varianty, které ve hře měli reprezentovat jejich silnější verze. Jak již bylo ale zmíněno v předchozí kapitole, zde jsem složitost implementace podcenil, takže se do hry dostala vždy jen jedna verze od každého nepřítele.

Ukázku jak část těchto modelů vypadá vykreslená v herním enginu ukazuje obrázek 3.



Obrázek 3: Ukázka části vytvořených modelů

Ve zbytku této kapitoly bych rád popsal postup, který je potřebný pro vytvoření modelu do hry.

4.1 Přípravy před modelováním

Samotnému modelování vždy předcházely drobné přípravy. Bylo potřeba si dobře ujasnit, jaký objekt bude modelován a přizpůsobit tomu začátek procesu (například výběr vhodného základního objektu - krychle, kužel, válec,...). Modelovaný

objekt si většinou nestačí jen představit v hlavě a je potřeba i jeho 2D reprezentace. V tomto ohledu jsem často hledal inspiraci na internetu, popř. mi pomohly drobné náčrtky.

Pro mě jako pro absolutního začátečníka bylo například velmi těžké vytvořit první humanoidní postavu. Po několika ne příliš úspěšných pokusech jsem si vyhledal anatomicky přesné náčrty lidské postavy a modeloval jsem podle nich, čímž jsem dosáhl pro mě už dostačující kvality.

4.2 Modelování

Jako grafický styl pro hru jsem si vybral tvorbu modelů s nízkým počtem polygonů. Proto jsem nejvíce využil metodu takzvaného boxového modelování. V této metodě se vždy začíná primitivním objektem s omezeným počtem polygonů. Tento objekt je pak pomocí různých operací formován do požadovaného tvaru. Název této metody vychází z faktu, že pro většinu objektů se jako výchozí objekt používá krychle, protože je nejvíce flexibilní.

Výběr tohoto grafického stylu měl hned několik důvodů:

Atmosféra
: U her zmíněných v úvodu mě překvapilo, jak výborně tento styl umí pracovat s atmosférou a jak také činí jednotlivé hry unikátní po grafické stránce.

Efektivita a jednoduchost
: Jelikož se pracuje pouze s omezeným počtem polygonů, je tvorba veškerých modelů značně zjednodušená. Což mi vyhovovalo, jelikož jsem měl v plánu si všechny modely připravit sám a zároveň to bylo moje první setkání s modelováním jako takovým.

Optimalizace
: Hry s použitím takových modelů není potřeba nijak výrazně optimalizovat.

Výsledný model by měl mít co možná nejmenší počet polygonů tak, aby stále na první pohled reprezentoval daný objekt. Žádná konkrétní hodnota počtu polygonů neexistuje a vždy závisí na složitosti daného modelu a požadované úrovni detailů.

Poté co je model hotový, je možné použít takzvaný sculpting, kterým se mohou modelu dodat další detaily. Nicméně tato technika při neopatrném použití může rapidně zvýšit počet polygonů. Sculpting jsem tedy při tvorbě využíval jen minimálně a uvádím jej zde spíše pro úplnost.

4.2.1 Základní postupy modelování

Při vytváření nových modelů v Blenderu máme k dispozici dva základní postupy pro budování výsledné struktury objektu:

Destruktivní postup se používá v rámci modelování nejčastěji. V podstatě pomocí různých nástrojů přímo měníme počty vrcholů a jejich pozice. Tyto operace jsou nezvratitelné[3] a přímo ovlivňují strukturu objektu, proto se tento postup nazývá destruktivním.

Nedestruktivní postup umožňuje provádět pokročilé modifikace objektu bez toho, aniž by byla změněná jeho struktura. V Blenderu se tyto nástroje jmenují modifikátory. Pokud tedy editujeme objekt, na kterém je umístěný některý z modifikátorů, umožňuje nám to stále pracovat s původní strukturou. Jakmile opustíme editaci objektu, vidíme změny, které modifikátor dopočítává. Blender dále v případě potřeby umožňuje modifikátor aplikovat a tím tak nevratně způsobit změny v struktuře objektu. Modifikátory jsou velmi užitečným nástrojem a významně urychlují tvorbu modelů.

4.3 UV mapování

Poté, co je model hotov, je potřeba připravit jej pro vytvoření textury. Tento proces se nazývá UV mapování a jde v něm o rozbalení 3D modelu na 2D plochu. Na tuto 2D plochu pak můžeme nanášet barvy a tím vytváříme základní texturu objektu.

Obtížnost UV mapování stoupá s počtem polygonů a komplexností rozbalovaného objektu. Hlavní metodou pro UV mapování je označování takzvaných švů. Tyto švy pak napovídají Blenderu, jak má daný objekt rozbalit.

Pokud pracujeme s jednoduchým modelem nebo nepotřebujeme příliš kvalitní UV mapování, je možné využití i automatický algoritmus rozbalování.

Jelikož jsem zvolil specifický způsob tvorby textur, tak jsem si ve většině případů vystačil právě s automatickým algoritmem UV mapování, který Blender nabízí.

4.4 Texturování

Jak již bylo zmíněno v předešlé kapitole, tak nanášením barev na 2D plochu tvoříme texturu objektu. Nicméně tato barevná textura není jediná, která se ve hrách využívá. Pro zvýšení detailů modelu a zachování nároků na výkonu se v moderních hrách běžně kombinuje více druhů textur, které napovídají vykreslovacímu enginu, jak má daný objekt zobrazit. Tyto textury pak běžně operují pouze s šedotónovou škálou barev. Ve hrách se pak můžeme setkat na příklad s následujícími texturami:

Normal (displacement) textura slouží pro vytvoření iluze dodatečné struktury objektu. Pomocí této textury tedy můžeme přidávat detaily samot-

[3] samozřejmě existuje v Blenderu jakási historie kroků, pomocí které můžeme zvrátit jednotlivé operace. Nicméně pokud bychom takto chtěli zvrátit některý z kroků, tak musíme zvrátit i všechny kroky po něm následující, nelze tedy selektivně zvrátit pouze jeden krok a následující nechat.

nému povrchu objektu, aniž bychom zvedali počet polygonů, kterými objekt disponuje.

Specular (gloss, roughness) textura umožňuje definovat jakým způsobem se má na objektu projevovat okolní světlo. Díky této textuře tedy můžeme definovat lesklé nebo matné povrchy.

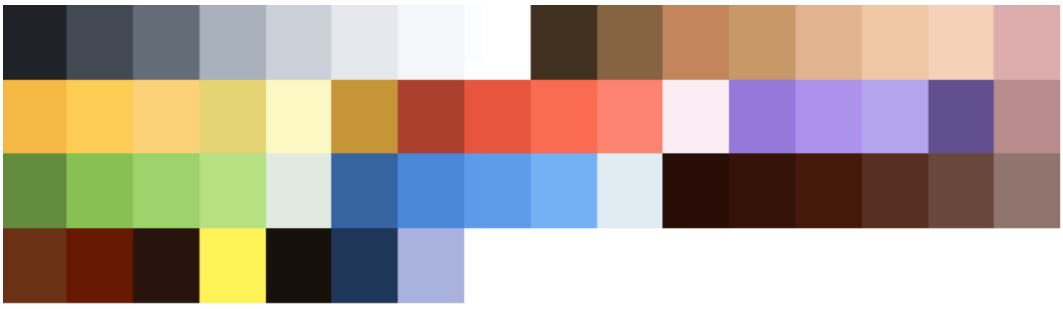
V mém případě jsem nepotřeboval pro své modely přidávat více realističnosti, takže jsem si vystačil pouze se základní barevnou texturou.

4.4.1 Barevná paleta

Pro textury jsem zvolil speciální metodu tvorby a to tvorbu pomocí barevné palety. Tato metoda se často využívá pro modely s nízkým počtem polygonů a spočívá ve vytvoření palety barev, které se pak nanášejí na jednotlivé polygony. Díky tomu vznikne jednotný systém barev, které mohou využívat všechny modely ve hře. Výhodou je i úspora potřebného výkonu. Jelikož všechny objekty využívají stejnou texturu, stačí tuto texturu držet v paměti pouze jednou pro všechny objekty.

Tato metoda má samozřejmě i své nevýhody. Tou hlavní je, že jelikož barvíme pouze celé polygony, dost výrazně omezujeme celý proces tvorby a textura nám není schopna přidat moc detailů. Další nevýhodou je, že při tvorbě textury jsme přímo závislí na struktuře objektu a jeho počtu polygonů. Není tak například možné nakreslit kruh na objekt, pokud takový objekt nemá na daném místě kruhové uskupení polygonů.

Pro potřeby této práce jsem vytvořil barevnou paletu o velikost 64x64 pixelů. Zde každá barva zabírá přesně 4x4 pixely. Tuto velikost jsem odvodil experimentálně od schopností Unity 3D. Při použití menší velikosti pro barvu, Unity nebylo schopno správně vykreslit texturu vznikaly grafické artefakty. Nicméně i taková textura byla více než dostačující a většina barevné palety zůstala nevyužita. Všechny modely napříč hrou využívají dohromady padesát osm barev, které lze vidět na obrázku 4.



Obrázek 4: Paleta barev, které využívají všechny modely

4.5 Tvorba kostry

Pokud model není čistě statický a je potřebné aby obsahoval animace, musíme nejdříve připravit takzvanou kostru modelu. Pomocí této kostry pak můžeme objekt ovládat a jeho struktura se bude přizpůsobovat změnám rotací, velikostí a pozic jednotlivých kostí. Sekvence takových změn nám pak tvoří výslednou animaci. Složitost tvorby kostry závisí na složitosti samotného modelu. U některých modelů si můžeme vystačit i s jedinou kostí a v jiných případe jich propojovat desítky.

Jak již bylo zmíněno, tak kostra modelu je tvořena z jednotlivých kostí. V kostře by měla vždy existovat jedna hlavní kost, na kterou se pak může napojit libovolná stromová struktura dalších kostí. Při tvorbě je tu jistá podobnost s reálnou kostrou, kdy jednoduše přidáváme a spojujeme kosti tam, kde očekáváme, že bude docházet k pohybu. Obrázek 5 ukazuje, jak tato kostra vypadá na postavě hráče.

Obrázek 5: Kostra modelu hráče

Blender nám zde opět ulehčuje práci a to hlavně pomocí symetrie kostry. Pokud tedy pracujeme na modelu, který je nějakým způsobem symetrický, můžeme pracovat pouze na jedné polovině kostry a druhou polovinu nechat na Blenderu.

4.5.1 Weight painting

Jakmile je kostra hotová, je potřeba ještě nadefinovat jak která kost ovlivňuje náš model. Tento proces nazýváme weight painting. Blender nám nabízí možnost

automatizace tohoto procesu a pro většinu modelů se jedná o dobrý startovní bod. Poté je většinou potřeba provést drobné úpravy a modifikace tak, aby se model deformoval podle našich představ. Pokud máme symetrický model, tak nám opět stačí pracovat pouze na jedné polovině modelu a kostry.

Po dokončení tohoto procesu, máme model připravený k animaci a veškeré manipulace s jednotlivými kostmi, by se měli projevit na samotném modelu.

4.6 Animování

Animace se tvoří pomocí takzvaných klíčových snímků. V těchto snímcích nedefinujeme požadovanou změnu a model se bude mezi těmito snímky v průběhu animace lineárně transformovat. Není tedy potřeba animovat každý snímek zvlášť, ale pouze nadefinovat správné klíčové snímky animace.

Postupů, jak vytvářet jednotlivé animace, je určitě mnoho. Mně osobně se osvědčilo nejprve vytvořit co nejméně klíčových snímků, pouze pro velmi charakteristické části animace. Následně jsem pak přidával klíčové snímky pouze tam, kde byl přechod až příliš lineární a nevzbuzoval příliš realistický dojem pohybu.

Model hráče a dále všechny ne humanoidní modely, mají veškeré animace vytvořeny ručně.

Humanoidní nepřátelé pak využívají animace z veřejně dostupné knihovny animací Mixamo.

4.7 Importování do Unity 3D

Posledním krokem v procesu tvorby modelů je jeho import do herního enginu. Unity 3D totiž používá jiný vykreslovací engine než Blender a tak se výsledky mohou různit. Dále samozřejmě je celkový výsledek ještě ovlivněn samotnou scénou, nastavenými světly a v neposlední řadě také post processingem. Bylo tedy celkem běžné, že jsem po importu musel ještě upravovat texturu modelu.

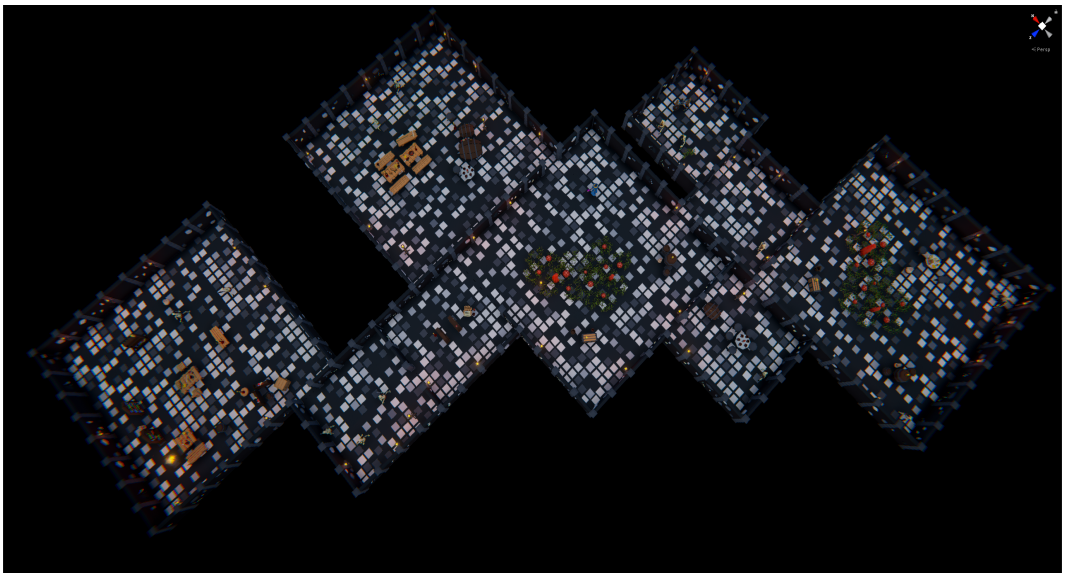
Unity 3D umí pracovat se soubory Blenderu a dokáže je připravit k použití. Ve většině případů nebylo tedy potřeba žádných dalších změn a pro import jsou použity přímo soubory Blenderu. U některých modelů Unity 3D import nezvládlo správně a bylo potřeba udělat manuální export modelu z Blenderu do formátu .fbx, který pak herní engine již zpracoval správně.

5 Herní mechanizmy

V této kapitole bych rád popsal implementaci konkrétních herních mechanizmů. Tato kapitola si v žádném případě neklade za úkol popsat vše, co bylo do hry implementováno. Kapitola tak bude spíše čtenáře informovat o nejzajímavějších částech z implementace.

5.1 Generování úrovní

Jedním z hlavních mechanizmů je samotné generování herních úrovní. Algoritmus generování je specifický pro tuto diplomovou práci a byl jednou z nejobtížnějších věcí pro implementaci. Hlavní koncept algoritmu spočívá ve vytvoření modulárních částí, které se spojují dohromady a tak vytváří výslednou herní úroveň. Výsledek algoritmu je možný vidět na obrázku 6.



Obrázek 6: Ukázka výsledku algoritmu pro generování úrovní

5.1.1 Příprava modelů a místností

Kvůli modularitě a co nejširším možnostem jsem tedy nejdříve připravil modely, které mi pomohly skládat jednotlivé místnosti dohromady. Tento systém budování místností pak využívám nejen v soubojové části hry, ale také v hráčské základně.

Takto vytvořené místnosti pak slouží generátoru jako hlavní vstup, coby vzory místností, které může pro generování použít. Ve hře jsou implementovány dvě základní čtvercové místnosti, které se liší pouze ve své velikosti. Nicméně celý algoritmus byl navržen tak, aby mohl pracovat s místnostmi různých rozměrů a tvarů. Každá místnost má dále svou kolekci různých dekorací a předmětů,

které se v místnosti při generování náhodně vytváří. Tím se algoritmus snaží docílit co největšího počtu různých variací.

Dále se musí u každé místnosti nastavit takzvané body zrodu. Každý takový bod musí mít definovanou svou orientaci (nahoře, vpravo, dole, vlevo) a množinu zdí, které mohou sloužit jako dveře v případě, že na tomto bodu dojde k propojení s jinou místností. Jak tedy z popisu vyplývá počet těchto bodů přímo ovlivňuje počet místností, které lze k dané místnosti připojit.

Posledním důležitým parametrem pro místnosti jsou výchozí body nepřátel a s nimi spojený poloměr. V tomto poloměru se pak můžou přidávat nepřátelé v poslední fázi procesu generování úrovní.

5.1.2 Cyklus přidávání místnosti

Před započetím samotného cyklu, který má na starosti přidávání jednotlivých místností, se vybere náhodná místnost z dostupných vzorů a zvolí se jako výchozí. Na tuto místnost se pak bude snažit napojit další vzorová místnost.

Cyklus pak probíhá následovně:

1. Vyber náhodnou místnost z předpřipravených vzorů a vytvoř její kopii na scéně. V tomto kroku se místnost skutečně zhmotní, aby byly k dispozici veškeré možnosti pro kontrolu kolizí.

2. Vyber náhodnou již vygenerovanou místnost, která má alespoň jeden volný bod zrození.

3. Vyber náhodný bod zrození této místnosti a snaž se jej napojit na vybraný vzor. V tomto kroku se zhmotněný vzor natočí tak, aby vybrané body zrodu byly k sobě čelem. Tato operace je možná díky tomu, že každý takový bod má definovanou svou orientaci.

4. Hledej pozici, která je volná. Místnosti postupně zkoušejí všechny možné vzájemné varianty zdí, které byly pro vybrané body zrodu definovány jako možné dveře.

   Pokud nedojde k žádné kolizi s jinou místností, tak je tento vzor považován za přidaný. Zdi, na kterých došlo k bezproblémovému přidání jsou odstraněny a cyklus opět pokračuje od prvního kroku. Cyklus se opakuje tak dlouho, dokud není vygenerován požadovaný počet místností.

   Pokud vzor nemohl být přidán, kvůli kolizím s již existujícími místnostmi, pak existuje několik možností, jak může algoritmus pokračovat:

   (a) Cyklus ještě nevybral všechny volné body zrodu vybrané místnosti, takže vybere další volný bod a znovu opakuje tento krok.

   (b) Cyklus ještě neprošel všechny vygenerované místnosti, v tomto případě tedy vybere další místnost a pokračuje třetím krokem.

(c) Cyklus ještě neprošel všechny předpřipravené vzory. Nejdříve tedy smaže aktuálně vytvořený vzor, vybere nový a pokračuje druhým krokem.

(d) Pokud cyklus vypotřeboval všechny možnosti, tak se celý proces resetuje a začíná se úplně od začátku. Což znamená, že se všechny již vytvořené místnosti smažou, vytvoří se výchozí místnost a pokračuje se od začátku cyklu.

5.1.3 Úklid a přidání dalších vchodů

Po skončení výše popsaného cyklu musí dojít ještě k úklidu a dalšímu vyhlazení výsledku. V popisu předchozího cyklu se mažou pouze zdi, které byly označeny jako dveře pro propojení jednotlivých místností. Tento fakt má za následek dva vážné nedostatky:

- Na většině míst budou zdi zdvojené, protože ne vždy dojde k propojení místností za pomocí rohových zdí. Jelikož zdi jsou uniformní, tak by to teoreticky nemuselo vadit. Problém je v tom, že v jednotlivých zdech se ještě náhodně vytvářejí různé dekorace, pro zpestření modelů. Dvojité zdi je tedy nutné odstranit.

- Do každé místnosti vede pouze jeden vchod, což má za následek velmi lineární průchod celou herní úrovní.

Tyto nedostatky se snaží vyřešit úklidový algoritmus. Ten v první fázi detekuje veškeré překrývající se zdi a postupně je maže. Při tomto mazání je nastavena určitá pravděpodobnost, kdy může dojít ke smazání obou takto překrývajících se zdí. Pokud k tomu dojde, tak jsme vytvořili nový vchod do místnosti. Tato pravděpodobnost byla pro algoritmus experimentálně nastavena tak, aby docházelo k občasnému vytvoření dalších vchodů a přitom ještě u většiny místností šlo stále rozlišit jejich hranice.

V druhé fázi ještě algoritmus ladí výsledek první fáze. Po první fázi se totiž občas vyskytovaly stěny, které stály uprostřed místnosti zcela osamostatněné. Druhá fáze tyto samostatně stojící stěny vyhledává a maže.

Po skončení úklidového algoritmu se hráč přesune do první vytvořené místnosti a začne se s přidáváním nepřátel do celé herní úrovně.

5.1.4 Přidávání nepřátel

Každá úroveň má předem definovanou obtížnost, kterou představuje sada koeficientů. Existuje právě jeden koeficient pro každou skupinu nepřátel (kostlivci, démoni, roboti) a sečtením těchto koeficientů získáme přibližnou obtížnost úrovně. Každý koeficient představuje kolik nepřátel daného typu bude do úrovně zhruba přidáno. Dále je možné i nastavit, jací konkrétní nepřátelé z dané skupiny budou při přidávání k dispozici. Tímto způsobem lze přesně ovládat, kterým nepřátelům bude muset hráč v dané úrovni čelit.

Dále každý nepřítel má nastaven svůj obtížnostní koeficient. Přičemž vyšší hodnota znamená silnějšího nepřítele. Takže pokud budou při přidávání vybírání pouze slabší jedinci, bude jich tam více, naopak pokud budou vybírání silnější, bude jich méně.

Přidávání jednotlivých nepřátel pak probíhá tak dlouho, dokud je suma koeficientů již přidaných jedinců menší, než nastavený koeficient úrovně. Přidávání probíhá pro každou skupinu nepřátel zvlášť a náhodně v rámci všech dostupných místností, krom té, ve které začíná hráč samotný.

5.2 Umělá inteligence nepřátel

Jelikož má hráč strávit většinu herního času právě soubojem s jednotlivými nepřáteli, snažil jsem se vytvořit rozmanité chování nepřátel a podařilo se mi vytvořit pro všech dvanáct dostupných nepřátel odlišné rysy chování. Bohužel jsem zde narazil na pár nedostatků komponent, které Unity 3D dodává pro pathfinding. Většinu problémů se mi naštěstí podařilo vyřešit stáhnutím vývojové verze těchto komponent ze stránek GitHub. Nicméně bylo pár okamžiků v průběhu vývoje, kdy jsem vážně zvažoval zahodit vše již hotové a přejít na nějaký jiný systém napsaný třetí stranou.

V následujících podkapitolách bych rád popsal některé z rysů chování nepřátel a také problémy, na které jsem při jejich vývoji narazil. V poslední podkapitole pak bude popsán příklad chování konkrétního nepřítele.

5.2.1 Společné rysy

Jak již bylo zmíněno, každý nepřítel je v něčem trošku unikátní, proto všichni nepřátele sdílejí pouze dva základní společné rysy:

- Nepřátele zůstávají neaktivní, dokud ve svém zorném poli nespatří hráče. Každý nepřítel má definovanou konstantní hranici viditelnosti. Nepřítele je možné aktivovat také zásahem některého z hráčských kouzel. Jakmile se nepřítel aktivuje, tak pak už následují akce, které jsou unikátní pro každý typ nepřítele.

- Jednotlivé akce jsou podmíněny krom okolních podmínek také časovači. V prvotní verzi jsem zkoušel systém, který místo časovačů používal pravděpodobnosti. Tento systém se ale neosvědčil, protože v některých případech se generovali situace, které byli příliš obtížné nebo naopak příliš snadné. Za použití časovačů vzniká více konstantní obtížnost, kdy na každou akci má nepřítel definovaný časový interval, za jak dlouho ji může použít znovu.

5.2.2 Pronásledování, útok, úprk

Většina nepřátel, po tom co byli zaktivovaní, se může nějakým způsobem pohybovat po herní úrovni a útočit na hráče. Mezi jednotlivými nepřáteli se ale liší,

jakým způsobem svůj pohyb využívají. Jednotlivé možnosti pohybu bych shrnul do tří kategorií v závislosti na typu nepřítele:

Statičtí nepřátelé
Jedná se o nepřátele, kteří nemají žádnou možnost pohybu. Pouze si kontrolují vzdálenost od hráče a pokud je na dosah, tak na něj zaútočí.

Nepřátele bojující zblízka
Tito nepřátelé se snaží za každou cenu přiblížit se na kontaktní dosah k hráči tak, aby na něj mohli zaútočit.

Nepřátele bojující pomocí projektilů
Nepřátelé pro boj na dálku se chovají nejvíce sofistikovaně. Každý má definovaný vlastní maximální vzdálenost pro střelbu na hráče a tuto vzdálenost se snaží nějakým způsobem udržovat. Tuto kategorii nepřátel lze dále rozdělit do dvou podkategorií:

Nepřátelé udržující si konstantní vzdálenost
Tito nepřátele se budou vždy, kdy je to možné, snažit dostat na konstantní vzdálenost k hráči a tuto vzdálenost si budou udržovat.

Nepřátele s nastavenými vzdálenostmi pro útok a úprk
Tito nepřátele se budou vždy snažit dostat na minimální vzdálenost pro možnost vystřelení projektilu na hráče. Pokud se hráč příliš přiblíží, začnou utíkat pryč. Algoritmus pro hledání únikové cesty nejprve zkouší hledat přímou cestu od hráče, takže nepříteli stačí pouze couvat. Pokud za sebou nemá dostatečný prostor, začne algoritmus zkoušet další únikové cesty pod širšími úhly. Tímto způsobem by měl nepřítel ve většině případů najít co nejideálnější cestu, jak vytvořit větší vzdálenost mezi sebou a hráčem. Algoritmus hledá pouze takové pozice, ze kterých bude mít nepřítel výhled na hráče a bude na něj moct hned zaútočit.

5.2.3 Ignorování překážek

V herních úrovních jsou generovány různé překážky, které brání ve volném pohybu hráči i nepřátelům. Do hry jsem proto přidal několik druhů nepřátel, kteří mají možnost tyto překážky ignorovat a procházet skrz ně nebo přes ně lítat.

Bohužel zde jsem při implementaci narazil na zásadní problém. Takový způsob chování nelze v rámci dostupných komponent nijak nastavit. Jednoduše řečeno, překážky existují pro všechny bez výjimek a není možné pro některý druh nepřítele tyto překážky vyjmout z algoritmu pro výpočet cesty mezi dvěma body.

Nakonec se mi to podařilo vyřešit způsobem, který není dle mého příliš optimální, nicméně to byl jediný způsob, na který jsem přišel. Do každé místnosti jsem přidal další skrytou podlahu, která je zvednuta do takové výšky, aby nepřátelé mohli ignorovat většinu překážek, ale přitom neignorovali samotné zdi

a tím i hranice jednotlivých místností. Jednotliví létající nepřátelé pak pro výpočet pohybu používají právě tuto zvednutou podlahu. Podobně pak funguje i robot, který chodí po zemi, ale jednotlivé překážky ignoruje a při pohybu skrz ně je rozbíjí. Výpočet jeho pohybu probíhá na zvednuté podlaze, ale samotný objekt robota je pak posunutý směrem k podlaze.

Tohle řešení pro mě bylo dostačující a dosáhl jsem kýženého výsledku, nicméně si dokáži představit situace, kdy by ani tohle řešení nepomohlo a po pravdě mě celkem překvapilo, že u takové základní věci jakou je pathfinding narazím na takové problémy.

5.2.4 Chování konkrétního nepřítele

Pro popis chování konkrétního nepřítele jsem si vybral jednoho z těch složitějších a tím je kostlivý mág. Jeho rozhodovací cyklus vypadá následovně:

1. Byl jsem aktivovaný? Pokud ne, tak aktivně vyhlížej hráče a čekej na aktivaci, jinak pokračuj dalším krokem.

2. Provádím zrovna nějakou akci (léčení, vyvolávání, útok,...)? Pokud ano, tak čekej na dokončení akce, jinak pokračuj dalším krokem.

3. Je k dispozici zraněný nepřítel a mám připravené léčení? Pokud ano, zahaj léčbu, jinak pokračuj dalším krokem.

4. Mohu vyvolat nepřítele? V tomto kroku se kontroluje, zda je vyvolání nabité a zda existuje pozice vhodná pro vyvolání nového nepřítele. Pokud ano, zahaj vyvolávání, jinak pokračuj dalším krokem.

5. Jsem ve vzdálenosti, ze které mohu zaútočit na hráče a mám nabitý útok? Pokud ano, zahaj útok, jinak pokračuj dalším krokem.

6. Je hráč příliš blízko a mám kam utéct? Pokud ano, mohou nastat dvě možnosti:

   (a) Teleport je nabitý a tak je možné zahájit teleportaci na dostupné místo úniku.

   (b) Jinak se začni postupně přesouvat na dostupné místo úniku.

   Pokud ne, tak pokračuj následujícím krokem.

7. Je hráč příliš daleko, nebo na něj nevidím? Pokud ano, přesuň se směrem k hráči, jinak proveď další průchod cyklem.

5.3 Fog of war

Pro hru bylo dále potřeba implementovat takzvaný fog of war. Tento mechanizmus skrývá před hráčem lokality, které by jeho herní postava neměla spatřit a omezuje tak jeho rozhled. Díky tomu tedy nelze vidět přes zdi do sousedních místností a stejný mechanizmus je použit i pro mapu, která se odkrývá jakmile hráč prochází herní úrovní.

Pro každou místnost je vytvořena speciální plocha, která je posunutá zhruba do výšky zdí v místnosti. Tato plocha je dále rozčleněná na polygony stejné velikosti. Jednotlivé vrcholy těchto polygonů nesou informaci o barvě, včetně alfa kanálu, a v kombinaci s ostatními tvoří barvu polygonů.

Hráč pak má nadefinovanou hranici viditelnosti, pro kterou se vypočítávají alfa hodnoty jednotlivých vrcholů. Čím je tedy hráč blíže k danému vrcholu, tím nižší alfa hodnotu bude mít (až do úplné průhlednosti). Vzdálenější vrcholy pak budou postupně přecházet do téměř maximální hodnoty alfa.

Výše popsaný postup by ale nefungoval úplně správně, protože stále nebere v potaz zdi jednotlivých místností. Proto se pro každý bod ještě kontroluje, zda mezi ním a hráčem nestojí zeď. Pokud ano, tak je přidána k celkovému výpočtu konstantní hodnota, zvedající finální hodnotu alfa kanálu.

5.4 Efekty kouzel

Důležitou součástí celé hry jsou samozřejmě i kouzla a jejich vizualizace. Právě tvorba vizualizace byl na celém systému kouzel zdaleka ten nejsložitější proces. Všechna kouzla ve hře jsou vytvořena pomocí částicových systémů přímo v Unity 3D.

Pro vytvoření požadovaného efektu je nejdříve potřeba vytvořit černo bílou texturu na průhledném pozadí. Tato textura je pak použita v částicovém systému jako jedna částice. Pomocí různých operací a kombinací více částicových systémů dohromady pak lze navodit 3D dojem celého efektu. Všechna kouzla využívají více různě nastavených částicových systémů s různými texturami.

Celý proces tvorby je velmi zdlouhavý. U každého částicového systému je totiž k dispozici velké množství různých vlastností, které lze nastavit. Každá tato vlastnost nějakým způsobem modifikuje chování jednotlivých částic. U efektů, které mají kolem desítky různých částicových systémů, je velmi složité, udržet je všechny v harmonii. Jedna zásadnější změna totiž znamená překonfigurování většiny ostatních.

Nicméně mě i přesto překvapilo, jak silným nástrojem dokážou tyto částicové systémy v Unity 3D být. A já jsem po několika zhlédnutých návodech byl schopen postupně začít tvořit ucházející herní efekty, které si myslím, že i zapadají do celkové grafické stránky hry.

5.5 Management stavu hry

V průběhu vývoje samozřejmě došlo k potřebě nějakým způsobem uchovávat aktuální stav hry. A to ať už kvůli testování nebo kvůli průchodu mezi jednotlivými herními úrovněmi. Tyto úrovně se v Unity 3D nazývají scény. Ve hře je tak několik scén, které je potřeba mezi sebou synchronizovat. Respektive je potřeba synchronizovat jen určitá data o aktuálním stavu hry: hráčovy životy, odemčená vylepšení, stav peněz,...

Za tímto účelem jsem vytvořil jeden objekt, který všechny tyto informace nese a je permanentní. Což znamená, že jej Unity 3D během přechodu mezi jednotlivými scénami nesmaže. Když pak má dojít k přechodu mezi scénami, tak se do něj uloží veškeré aktuální informace a až poté dojde k nahrání nové scény. Jakmile se nová scéna nahraje, tak jsou informace z tohoto objektu použity pro inicializaci této scény. V scéně základny to například znamená zobrazení zakoupených vylepšení nebo naplnění hrobů podle počtu hráčských smrtí, ale samozřejmě se díky tomu ukládá i nastavení hráčských kouzel, nebo aktuálně vlastněné předměty.

Tento způsob managementu hry mi umožnil i mnohem efektivnější testování. Stačilo totiž inicializovat tento objekt požadovanými hodnotami a mohl jsem začít simulovat různé varianty situací, které ve hře mohou nastat.

V neposlední řadě je tento způsob i velmi efektivní pro ukládání hráčského postupu hrou na disk. Musíme jen tento objekt serializovat a zapsat na úložiště zařízení. Při načítání pak stačí tento objekt vzít, deserializovat a nahradit jím aktuální objekt stavu hry, který je scéně k dispozici. O správnou inicializaci se pak už postará mechanizmus popsaný výše.

6 Herní příručka

Poslední kapitola této diplomové práce je věnována vysvětlením některých herních mechanizmů, které byly ponechány k objevení hráčem. Proto také slouží jako pokročilá hráčská příručka, kde se může hráč například dozvědět, jak přesně fungují jednotlivé předměty nebo vylepšení.

6.1 Barvy kouzel a jejich efekty na různé druhy nepřátel

Ve hře jsou k dispozici celkem čtyři barvy kouzel: purpurová, zelená, červená a žlutá. Ve hře reprezentují elementy magie, života, smrti a energie. Každá barva má pak na jednotlivé druhy nepřátel, ale i na hráče samotného, jiný účinek, jak uvádí tabulka 1.

Tabulka 1: Barvy kouzel a jejich účinky

	Purpurová	Zelená	Žlutá	Červená
Hráč	zranění	léčení	zranění	zranění
Kostlivci	zranění	extra zranění	snížené zranění	léčení
Roboti	extra zranění	snížené zranění	léčení	zranění
Démoni	snížené zranění	léčení	zranění	extra zranění

6.2 Postihy za smrti hráče

Za jednotlivé smrti je hráč penalizován pouze, pokud nemá již volné místo na hřbitově (viz níže). Za smrti, které jsou nad dostupnou kapacitou hřbitova, je hráč penalizován procentuálním snížením maxima životů. Tyto penalizace uvádí tabulka 2.

Tabulka 2: Procentuální snížení životů v závislosti na počtu smrtí

Počet smrtí nad limit hřbitova	Postih v %
1 až 2	10
3 až 4	20
5 až 6	30
7 až 9	40
10 a více	50

6.3 Dostupná kouzla

Ve hře je k dispozici celkem jedenáct různých kouzel. Tato kouzla jsou pak rámci hry rozdělená do třech různých úrovní. Kdy kouzla ve vyšší úrovni jsou silnější, než kouzla v úrovních nižších.

Většina kouzel automaticky působí i na hráče a tak, pokud má kouzlo například explozivní zranění, je možné, že se hráč zraní také.

6.3.1 Kouzla první úrovně

První dvě níže uvedená kouzla hráč odemkne hned v průběhu úvodního tutoriálu, zbytek musí nalézt poražením nepřítel nebo zakoupením v základně.

Simple arrow je základním kouzlem s nízkou dobou nabití a s nízkým poškozením.

Explosive ball je další ze základních kouzel, ale už má vyšší poškození a také po kontaktu exploduje a způsobuje v malém okruhu dodatečná zranění.

Tripple arrow a Tripple ball tvoří varianty výše zmíněných kouzel, kdy místo jednoho projektilu hráč vystřelí tři. Tyto projektily mají oproti svým protějškům mírně snížené účinky.

Orbitting orbs je kouzlo vyvolávající tři orby, které rotují kolem hráče. Orby mají omezenou životnost, ale jinak je nelze zničit. Při kontaktu s nepřáteli jim udělují zranění.

6.3.2 Kouzla druhé úrovně

Pro možnost odemčení těchto kouzel je potřeba mít odemčena všechna kouzla první úrovně.

Circle of arrows je další variantou základního kouzla, které vyvolá celkem deset základních projektilů kolem hráče.

Shield slouží jako dodatečná ochrana proti zranění. Pokud je tento štít aktivní, tak redukuje veškeré zranění které hráč obdrží. Pokud barva štítu odpovídá barvě, která dává danému nepříteli extra zranění (viz výše), pak je veškeré zranění od takového nepřítele redukováno ve větší míře.

Damage aura zvyšuje zranění všech kouzel stejné barvy, jakou má tato aura.

Mine umožňuje vykouzlit minu s omezenou životností. Pokud se nepřítel přiblíží na kolizní pozici, tak mina exploduje a způsobí vysoké zranění.

Projectiles throwing orb vytvoří velký orb, který bude automaticky zaměřovat blízké nepřátele a střílet po nich naváděné projektily.

Magical aura vytvoří na zemi auru s omezenou životností, která neustále zraňuje vše uvnitř.

6.3.3 Kouzla třetí úrovně

Pro odemčení těchto kouzel je potřeba nejdříve posbírat všechna kouzla první a druhé úrovně. Nicméně tato kouzla jsou nejsilnějším dostupným arzenálem a po jejich získání, by nemělo hráči dělat problémy hru dokončit.

Instant explosion vytvoří na požadovaném místě výbuch, který způsobí znatelné zranění všem v okolí.

Slash projectil je projektil, který se po kontaktu s nepřáteli nerozbije. Jednoduše putuje dál a zraňuje všechny nepřátelé, kteří se mu postaví do cesty. Pokud se střetne se zdí, nebo s jiným kouzlem, tak exploduje a způsobí dodatečná zranění.

Tornado vytvoří tornádo s omezenou životností, které neustále zraňuje a zpomaluje vše, co se s ním dostane do styku.

6.4 Dostupná vylepšení

Za peníze získané ve hře, si může hráč pořizovat jednotlivá vylepšení své základny, které mu dávají bonusy do hry. Hráč má k dispozici čtyři různá vylepšení a každé toto vylepšení má tři úrovně. Jednotlivé úrovně vylepšení stojí vždy stejně a to: 100 mincí za první úroveň, 250 mincí za úroveň druhou a 500 mincí za poslední třetí úroveň.

Library
Postaví knihovnu, ve které si může hráč měnit svá kouzla. Druhá a třetí úroveň tohoto vylepšení vždy dovolí hráči používat v boji o jedno kouzlo navíc. S maximální úrovní tohoto vylepšení, si tak hráč může nastavit čtyři různá kouzla.

Graveyard
Přidá hřbitov, který zvyšuje počet hráčských smrtí bez zavedení postihu. Každá další úroveň zvyšuje počet míst na hřbitově. První úroveň přináší čtyři volná místa, druhá úroveň jich má osm a poslední třetí umožňuje hráči šestnáctkrát zemřít zcela bez postihu.

Training ground
Dostaví k základně cvičiště, na kterém si může hráč zkoušet svá kouzla. Dále tato cvičiště zvyšují maximální počet životů a to o: 10, 25 nebo 50, v závislosti na zakoupené úrovni vylepšení. Poslední úroveň tohoto vylepšení také umožní trénovat proti některým skutečným nepřátelům.

Vending machine
Přidá na základnu místnost s výherním automatem. Výhra v automatu není jistá a pravděpodobnost výhry se zvyšuje s úrovní tohoto vylepšení. Od druhé úrovně vylepšení se do místnosti přidá druhý automat. První automat má vyšší šanci na výhru nějakého předmětu a druhý má vyšší šanci na získání nového kouzla. Poslední úroveň zároveň přidává šatník, který umožní hráči měnit róby.

6.5 Dostupné předměty

Ve hře je k dispozici celkem dvanáct různých předmětů, které různě ovlivňují hráče. Hráč je omezen možností mít v jednu chvíli pouze dva předměty. Jednotlivé předměty, které jsou ve hře a popis jejich vlastností:

Kovadlina je jedním ze dvou škodících předmětů, pouze mírně zpomaluje hráče v pohybu.

Stopky jsou druhý škodící předmět, který zvyšuje čas potřebný pro nabití kouzel.

Slepičí čepice je jeden ze dvou dekorativních předmětů, hráči neškodí ani mu nepomáhá, pouze částečně mění vzhled.

Brýle jsou druhým dekorativním předmětem.

Ankh zachrání hráče před smrtí. Hráč se objeví zpátky na základně, ale neztratí své mince, a také se tato smrt nebude započítávat do postihů. Ankh je nicméně po použití nenávratně ztracen.

Boty výrazně zrychlují hráče při kouzlení.

Prsten se srdcem zvyšuje maximum životů hráče o 50 bodů.

Kostky zlepšují pravděpodobnost ve prospěch hráče napříč hrou. Tento předmět tak například zvyšuje šance, že z nepřátel hráč něco získá, ale také snižuje pravděpodobnost smrti, při návratu do základny.

Portálová zbraň zaručuje hráči bezpečný návrat na základnu po dokončení úrovně.

Lektvar pokud se životy hráče dostanou na hranici nuly, tento předmět se automaticky použije a hráče vyléčí do maxima. Lektvar je po použití ztracen.

Prsten s lebkou ruší veškeré postihy plynoucí z hráčských smrtí.

Konzerva špenátu zvyšuje zranění všech kouzel o 15%.

6.5.1 Róby hráče

Speciálním druhem předmětů jsou hráčské róby. Tyto róby se nepočítají do limitu nošených předmětů, ale hráč můžeme mít vždy jen jednu aktivní róbu. Celkem je k dispozici šest barevných kombinací rób:

Modrá róba je výchozí róbou, která nepřináší hráči žádné výhody.

Purpurová, zelená, žlutá a červená róba jsou róby, které zvyšují zranění jednotlivých barev kouzel o 15%.

Černá róba je univerzální róbou, která zvyšuje zranění všech kouzel o 15%.

Závěr

Náplní diplomové práce bylo vytvořit 3D hru v herním enginu Unity 3D. Hra měla poskytovat generování herních úrovní, umělou inteligenci nepřátel a také obsahovat vlastní grafiku v podobě 3D modelů. Hlavní náplní hry měl být průchod několika vygenerovaných herních úrovní po sobě, bez smrti hráče a dále možnost zakupovat různá vylepšení. Všechny tyto části byly do hry implementovány a navíc zde byly dopracovány i dodatečné invence, které celou hru tvoří zajímavější a rozmanitější.

V rámci této práce jsem si osvojil tvorbu herních modelů v programu Blender, včetně pokročilých praktik a dále se mi podařilo nahlédnout do většiny částí herního enginu Unity 3D.

Text diplomové práce popisuje proces tvorby 3D modelů a také implementaci některých zajímavých herních mechanizmů. Text také může sloužit částečně jako hráčská příručka s užitečnými informacemi pro hráče.

V textu jsou také popsány využité technologie a motivace.

Conclusions

The goal of the thesis was to create a 3D game in the game engine Unity 3D. The game was supposed to have features like game level generation, artificial intelligence, and its own 3D models. The main task for the player should be completing several generated levels without death. Player should also be able to purchase various upgrades. All these elements are realized and furthermore additional inventions where implemented that make the whole game more interesting and diverse.

As part of this work, I have learned how to create game models in Blender, including advanced techniques. I have also learned how to use major parts of the Unity 3D game engine.

Text of diploma thesis is focused on creation of 3D models and describing implementation of some interesting game mechanisms. The text can also be used as a game guide with useful information for players.

The technology and motivation are also described in the text.

A Obsah přiloženého CD

Na přiloženém CD se nachází dvě složky a dva soubory.

`Dungeon.exe`
: Zkompilovaný soubor, který slouží pro spuštění hry

`UnityHubSetup.exe`
: Instalační soubor nástroje Unity Hub. Pomocí tohoto nástroje je potřeba nainstalovat Unity 3D ve verzi 2018.3, se kterou pak lze zkompilovat hru ze zdrojových souborů.

`dungeon/`
: Tato složka obsahuje zdrojové soubory vytvořené hry, včetně zdrojových souborů všech modelů.

`doc/`
: Složka obsahuje text diplomové práce ve formátu PDF. Ve složce se také nachází všechny soubory, které jsou potřebné pro vytvoření tohoto souboru.

Literatura

[1] BLENDER. Blender - Free & Open Source 3D creation software. Blender [online]. [cit. 2019-04-22]. Dostupné z: https://www.blender.org/

[2] UNITY TECHNOLOGIES. Unity for all. Unity [online]. [cit. 2019-04-22]. Dostupné z: https://unity3d.com/

[3] PARADOX INTERACTIVE. Magicka. Paradox Interactive [online]. [cit. 2019-04-22]. Dostupné z: https://www.paradoxplaza.com/magicka/MAMA01GSK-MASTER.html

[4] MCMILLEN Edmund. The Binding of Isaac: Rebirth. Steam store [online]. [cit. 2019-04-22]. Dostupné z: https://store.steampowered.com/app/250900/

[5] NOIO, LICORICE. Kingdom: New Lands. Kingdom the game [online]. [cit. 2019-04-22]. Dostupné z: https://www.kingdomthegame.com/kingdom-new-lands/

[6] IRONOAK GAMES INC.. For The King. IronOak Games [online]. [cit. 2019-04-22]. Dostupné z: https://www.ironoakgames.com/

[7] FISHING CACTUS. Epistory - Typing Chronicles. Epistory game [online]. [cit. 2019-04-22]. Dostupné z: http://www.epistorygame.com/

[8] SUPERHOT TEAM. Superhot. Superhot game [online]. [cit. 2019-04-22]. Dostupné z: https://superhotgame.com/

[9] GAMEDEV.TV. Complete Blender Creator: Learn 3D Modelling for Beginners. Udemy [online]. [cit. 2019-04-22]. Dostupné z: https://www.udemy.com/blendertutorial/

[10] GAMEDEV.TV. Complete C# Unity Developer 3D: Learn to Code Making Games. Udemy [online]. [cit. 2019-04-22]. Dostupné z: https://www.udemy.com/unitycourse2/

[11] CREATIVE BLOQ. The best 3D modelling software 2019. Creative Bloq [online]. [cit. 2019-04-22]. Dostupné z: https://www.creativebloq.com/features/best-3d-modelling-software

[12] BLENDER. About Blender. Blender [online]. [cit. 2019-04-22]. Dostupné z: https://www.blender.org/about/

[13] BLENDER. Blender Add-ons. Blender [online]. [cit. 2019-04-22]. Dostupné z: https://docs.blender.org/manual/en/latest/preferences/addons.html

[14] BLENDER. Blender Features. Blender [online]. [cit. 2019-04-22]. Dostupné z: https://www.blender.org/features/

[15] PETTY Josh. Top 12 Free Game Engines For Beginners & Experts Alike. Concept Art Empire [online]. [cit. 2019-04-22]. Dostupné z: https://conceptartempire.com/free-game-engines/

[16] UNITY TECHNOLOGIES. A new input system for Unity. GitHub [online]. [cit. 2019-04-22]. Dostupné z: https://github.com/Unity-Technologies/InputSystem#timeline

[17] KRITA. Krita - digital painting, creative freedom. Krita [online]. [cit. 2019-04-22]. Dostupné z: https://krita.org/en/

[18] AURACITY. Audacity - free, open source, cross-platform audio software. Audacity team [online]. [cit. 2019-04-22]. Dostupné z: https://www.audacityteam.org/