



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROGUELIKE HRA ZE TŘETÍ OSOBY V UNITY

THIRD PERSON ROGUELIKE GAME IN UNITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL MOUDRÝ

VEDOUcí PRÁCE

SUPERVISOR

Ing. TOMÁŠ MILET, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Moudrý Daniel**
Program: Informační technologie
Název: **Roguelike hra z třetí osoby v Unity**
Third Person Roguelike Game in Unity
Kategorie: Počítačová grafika

Zadání:

1. Nastudujte techniky vývoje her a herní engine Unity. Nastudujte procedurální generování využitelné v roguelike hrách.
2. Navrhněte hru z třetí osoby využívající procedurální generování.
3. Implementujte navrženou aplikaci.
4. Aplikaci uživatelsky otestujte, zhodnoťte a navrhněte rozšíření.
5. Vytvořte demonstrační video.

Literatura:

- Gregory, Jason. *Game engine architecture*. crc Press, 2018. ISBN 1351974289, 9781351974288
- Bishop, Lars, et al. "Designing a PC game engine." *IEEE Computer Graphics and Applications* 18.1 (1998): 46-53.
- Adams, Ernest, and Joris Dormans. *Game mechanics: advanced game design*. New Riders, 2012. ISBN 0321820274, 9780321820273

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2 a kostra aplikace.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Milet Tomáš, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

Abstrakt

Práce obsahuje informace o tvorbě roguelike počítačové hry ve třetí osobě v herním enginu Unity. Ze začátku je popsáno, o čem hra je, poté práce popisuje princip fungování této hry a dále samotnou implementaci v herním enginu Unity. Konkrétně v programovacím jazyce C#.

Abstract

The thesis comprises information about the creation of a third-person roguelike computer game in Unity game engine. At the beginning the work explains what the game is about, then the work describes the principles of the games functioning and then the implementation in Unity game engine itself in the C# programming language.

Klíčová slova

Unity, herní engine, počítačové hry, grafika, herní design, C#, hra na hrdiny, roguelike

Keywords

Unity, game engine, computer games, graphics, game design, C#, role-playing game, roguelike

Citace

MOUDRÝ, Daniel. *Roguelike hra ze třetí osoby v Unity*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Milet, Ph.D.

Roguelike hra ze třetí osoby v Unity

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáše Mileta, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Daniel Moudrý
11. května 2022

Poděkování

Děkuji panu Ing. Tomáši Miletovi, Ph.D. za vedení, podporu a konzultace při tvorbě této práce.

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 2 |
| 2 | Představení hry | 3 |
| 3 | Teorie | 6 |
| 3.1 | Herní engine | 6 |
| 3.2 | Proces tvorby počítačových her | 6 |
| 3.3 | Průzkum trhu | 7 |
| 4 | Herní mechaniky | 11 |
| 4.1 | Návrh | 11 |
| 4.2 | Programový koncept | 12 |
| 5 | Implementace | 18 |
| 5.1 | Prostředí Unity | 18 |
| 5.2 | Provedení v prostředí programovacího jazyka | 31 |
| 6 | Závěr | 39 |
| | Literatura | 40 |

Kapitola 1

Úvod

Tato bakalářská práce se věnuje návrhu a vývoji počítačové hry v herním enginu Unity. Konkrétně se jedná o roguelike hru ze třetí osoby.

Roguelike hry nemají jednoznačnou definici, obecně je ale podstatou roguelike her to, že obsahují procedurálně generované úrovně, které hráč hraje stále znovu a po každém průchodu úrovní se stále zlepšuje (ať už vybavení a statistiky své herní postavu nebo svoji schopnost zvládnout úroveň odehrát lépe). Díky jejich procedurálnímu generování jsou úrovně pokaždé v něčem jiné. [4]

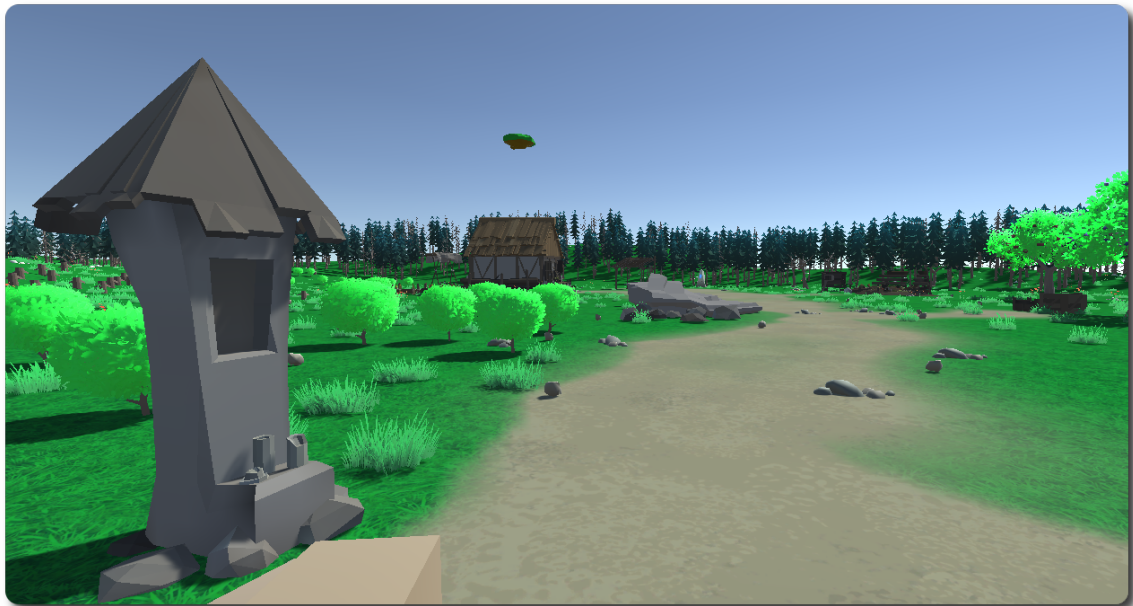
V tomto projektu spočívá průběh hraní v tom, že má hráč svoji postavu, se kterou prochází náhodně generované dungeony a snaží se ji stále vylepšovat. V herním prostředí je pojem dungeon označován pasáž hry, kdy je prostředí situováno do podzemních katakomb, labyrintů, jeskyní nebo jiných podobných kulis [1].

Dungeony se skládají z různých místností propojených chodbami. V těchto místnostech číhá všemožné nebezpečí a nepřátelé, ale také zde hráč najde zasloužené odměny v podobě lepšího vybavení a dalších věcí, které mu pomohou úspěšně projít až k východu a zdárnému konci dungeonu. Na konci dungeonu bude mít možnost odejít, nebo se podívat do dalších místností, které ještě neprozkoumal. Pokud se některá z místností ukáže být pro hráče nad jeho síly a připraví jeho postavu o život, tak ztratí všechny věci, které má v inventáři, a průchod skončí. Nalezené věci si hráč nechá pouze, pokud opustí dungeon.

Kapitola 2

Představení hry

Hráč vstupuje do malebného světa, který mu poskytuje útočiště a potřebné zázemí pro jeho výpravy. V tomto světě se ale dlouho nezdrží, protože je třeba se připravit, vybavit svoji postavu nutnou výbrojí a pustit se do dobrodružství, které ho čeká. Jakmile je hráč připraven, vydává se ke vstupu do dungeonu, který ho přenesení do nebezpečného a zrádného prostředí podzemních kobek a síní. Zde se snaží nalézt lepší výbavu a magické lektvary, které mu pomohou s jeho cílem - porazit hlavního nepřítele, který se v dungeonu nachází. To se mu ale nepovede bez získání vyšší úrovně (která jeho postavě přidá více bodů zdraví), lepších zbraní a dobré zbroje. Než bude připraven, musí stále dokola bloudit spletitou sítí místností a v nich zápolit s množstvím nepřátel, aby našel východ, kterým se dostane zpět do mírumilovného světa, ze kterého přišel. Zde může načerpat síly, vylepšit si zbraně, které našel, a odložit si do své truhly předměty, o které nechce přijít. Neuspěje-li totiž při jedné z jeho výprav a jeho postava zemře, ztrácí všechny předměty, jež nese s sebou, kromě těch, které má jeho postava vybaveny. Postava se ožíví, ale hráč tím ztrácí drahocenný čas a předměty, které by mu mohli pomoci v boji s jeho hlavním nepřítelem.

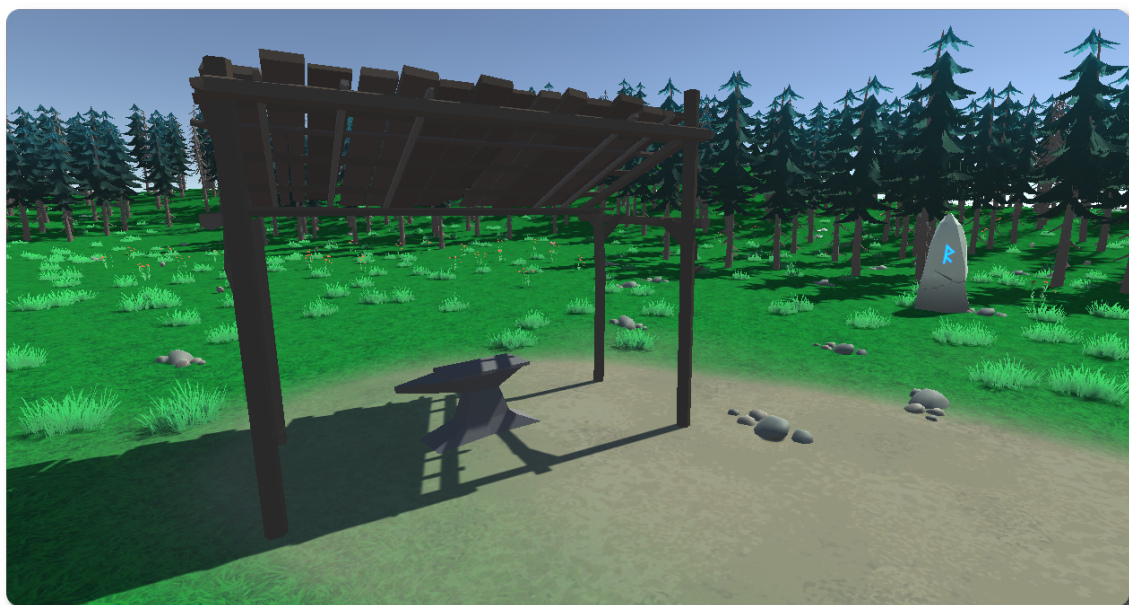


Obrázek 2.1: Pohled na svět, ze kterého hráč podniká své výpravy do dungeonu. Může si zde uložit předměty v truhle nebo si vylepšit zbraně u kovadliny.

Vzhled světa, který poskytuje hráči útočiště, je vyobrazen na snímcích 2.1, 2.2 a 2.3. Prostředí nebezpečného dungeonu a některé momenty soubojů s nepřáteli lze vidět na obrázcích 2.4 a 2.5.



Obrázek 2.2: Jedním z modelů nacházejících se ve světě je domek, před kterým se nachází truhla pro ukládání předmětů. V dále je na obloze vidět jeden z pohyblivých vznášejících se ostrovů, které působí jako dekorace a prvek pro oživení prostředí.



Obrázek 2.3: Na tomto obrázku lze vidět kovářinu pro vylepšování předmětů a dále vpravo je obelisk, kterým se hráč přesouvá do dungeonu.



Obrázek 2.4: Zde je vyobrazen souboj mezi hráčovou postavou a jedním z nepřátel v dungeonu.



Obrázek 2.5: Pohled na největší místnost v dungeonu, která je jako jediná z místností dvoupatrová a má více východů.

Kapitola 3

Teorie

Tato kapitola se věnuje teorii herních enginů a vysvětluje, co herní engine vlastně je a nahlíží na některé existující herní enginy. Další částí je také teorie tvorby počítačových her od počátečního návrhu až po vydání dokončené hry. A poslední částí této kapitoly je průzkum trhu, kde jsou zmíněny hry, které svými prvky a herními mechanikami inspirovaly tuto práci.

3.1 Herní engine

Herní engine, jak jej popisují autor článku Co je to herní engine [6] a autor článku Herní engine [9], je softwarový nástroj sloužící k vývoji počítačových her. Díky obsahu znovupoužitelných prvků, zařizujících, mimo jiné, například simulace fyzikálních zákonů a vykreslení scén, dokáže herní engine velmi urychlit a zlevnit vývoj počítačových her. Míra urychlení vývoje závisí na typu vybraného enginu a typu hry, která má být vytvořena. Pro různé typy her jsou vhodné různé herní enginy. Některé enginy poskytují velké množství pokročilých funkcí a prvků na úkor toho, že jsou zaměřeny na specifický typ her. Ty enginy, které jsou univerzálnější, většinou neposkytují tolik funkcí jako enginy pro užší rozsah zaměření.

Některá vývojářská studia si herní engine vytvoří sama a některá použijí již existující engine. Důvodem vytvoření nového enginu může být například právě velká specifická hra a potřeb pro její tvorbu.

Pro detailní náhled do teorie tvorby a architektury herních enginů lze nahlédnout do knihy Game Engine Architecture [5].

3.2 Proces tvorby počítačových her

Tvorba počítačových her, kterou popisuje autor článku Produkce počítačové hry od A do Z [7], je časově náročný a nákladný proces, na kterém se u větších her, jako jsou AAA hry, podílí velké množství lidí.

AAA hry nebo také Triple-A hry jsou velkorozpočtové videohry, jejichž vývoj zahrnuje obrovské týmy zaměstnané významným vývojářským studiem, které pracují měsíce až roky na vývoji jednoho herního titulu. Vývoj obecně následuje velká marketingová kampaň. Tyto hry poté distribuuje velký vydavatel, jako jsou například Activision, Electronic Arts, Ubisoft, Nintendo a další. [2]

Nejdříve je nutné sestavit návrh samotné hry, který obsahuje, mimo jiné, design levelů, popis používaných zbraní a nástrojů, popis postavy, popis světa, popřípadě i příběh, který

je nedílnou součástí mnoha her. Tomuto návrhu se říká design document. Ačkoliv jsou různé hry různě komplexní, tento návrh zabere vždy velké množství času, i když někdy více a někdy méně, jelikož v něm musí být obsažen kompletní návrh všech aspektů celé hry. Tento postup návrhu počítačových her je pouze stručnou verzí. Podrobnější postup lze najít v knize *Game mechanics: advanced game design* [3].

Dalším krokem vývoje počítačových her je předání design documentu vydavatelské společnosti, která se na základě návrhu rozhodne, bude-li projekt financovat a vydá jej. U menších titulů, jako jsou například indie hry, se ale stává i to, že hru zafinancují a vydají samotní vývojáři.

Indie hra (z anglického independent video game) nebo také nezávislá hra, je videohra vytvářená jednotlivcem nebo malým týmem bez finanční podpory vydavatele videoher. Proto je základem takových her často nějaký originální nápad týkající se například zajímavého a nešedního příběhu nebo neotřelých a inovativních herních mechanik. Indie hry jsou často distribuovány elektronicky přes internet, nikoli fyzicky. Příkladem velmi známé indie hry je *Minecraft*, který byl vyvinut Markusem Perssonem v roce 2009. Je nutno podotknout, že od té doby prošel *Minecraft* a společnost Mojang, kterou Persson se hrou založil, mnoha změnami a také byl v roce 2014 odkoupen společností Microsoft a dnes už se díky těmto okolnostem neoznačuje jako indie hra. [12, 11]

Po získání finančních prostředků může nastat samotný vývoj hry. Vývoj většinou začíná u herního enginu, který si společnost vytvoří, nebo použije některý z již existujících engineů (například Unity nebo Unreal Engine).

Následuje tvorba textur, animace, zvuky, hudba, modely, skripty a další částí. Tato fáze vývoje je velmi časově náročná a běžně trvá i roky. V průběhu vývoje se jednotlivé části hry musí testovat, aby byly nalezeny buggy (chyby), které je nutno opravit. Výsledkem by měla být hra připravená k vydání.

3.3 Průzkum trhu

Tato sekce se zaměřuje na některé již existující hry, které byly nějakým způsobem inspirací pro tuto práci.

3.3.1 Rogue Legacy¹

Rogue Legacy je platformová roguelike hra vyvinuta firmou Cellar Door Games a vydaná v roce 2013. Úrovně jsou procedurálně generovány a cílem hry je porazit čtyři bosse a poté jednoho finálního bosse. Grafické zpracování hry lze vidět na obrázku 3.1.

¹https://en.wikipedia.org/wiki/Rogue_Legacy



Obrázek 3.1: Začátek hry a úvod hry Rogue Legacy²

„Boss se používá v terminologii PC her jako označení pro postavu opoziční strany, která se obvykle nachází na konci herního kola / úrovně a je mnohem silnější a lépe vybavena k boji než předešlí nepřátelé. Souboj s ní, označovaný též jako bossfight, často probíhá v místnosti, která je k tomuto účelu vytvořena a postupem hry se do ní hráč dostane.“ [8]

Každý ze zmíněných čtyř bossů je v jedné ze čtyř vzhledově odlišených sekcí dungeonu, do kterých jednotliví bossové patří. Po porážce těchto bossů je na řadě souboj s finálním bossem. Pokud hráč neuspěje a jeho postava ztratí všechny body zdraví, postava zemře a hráč si musí zvolit dědice, kterého si vybere ze tří postav, které mu hra nabídne. Vlastnosti dědice jsou náhodně generovány a mohou být kladné i záporné. Příklady těchto vlastností jsou třeba ADHD (rychlejší pohyb postavy), barvoslepost (hráč vše vidí černobíle) a tak dále (obrázek 3.2). Po tom, co si hráč vybere dědice, se přesune do svého sídla, kde si za měnu, kterou v průběhu jeho hraní získává, může koupit různá vylepšení, jež mu usnadní další průchody úrovněmi.

²https://store.steampowered.com/app/241600/Rogue_Legacy/



Obrázek 3.2: Přehled statistik a vlastností postavy v Rogue Legacy³

3.3.2 Escape from Tarkov⁴

Jedná se o hru vyvinutou společností Battlestate Games a následně vydanou v roce 2017. Vývojáři této hry je označována jako náročná a realistická střílečka z první osoby s prvky her pro více hráčů.

V této hře hráč ovládá postavu (vojáka/žoldáka), kterou může vybavit různými předměty (zbraně, obvazy apod.), pokud je má. S postavou vstupuje na herní mapu a jeho cílem je probojovat se skrze další hráče a NPC. „NPC (*non-playable character*; česky *nehratelná postava*) je postava v počítačové hře (nejčastěji ve hře na hrdiny), kterou neovládá hráč, ale počítač.“ [13] Hlavně chce ale hráč získat nové vybavení, dostat se do extrakčního bodu na mapě a uniknout. Pokud se hráči nepovede uniknout a jeho postava zemře, ztratí většinu předmětů, které u sebe jeho postava měla.

Ve hře je více herních módů, ale ten, který je výše popsán, je nejzásadnější ve spojitosti s touto prací.

3.3.3 Enter the Gungeon⁵

Enter the Gungeon je bullet hell roguelike hra se specifickým designem. Pojem bullet hell označuje podžánr stříleček, kde se hráč musí neustále vyhýbat spoustě nepřátelských projektilů [14]. Jak už název hry, tak i příběh a vzhled celé hry se točí kolem střelných zbraní. Nepřátelé jsou často v podobě nábojů, teleporty vypadají jako válec revolveru a tak dále. Grafické zpracování lze vidět na obrázku 3.3. Co se týče příběhu, tak je velmi zkráceně o tom, že hráč se snaží najít zbraň, která dokáže zničit minulost jejich postavy. Místnosti

³https://store.steampowered.com/app/241600/Rogue_Legacy/

⁴https://cs.wikipedia.org/wiki/Escape_from_Tarkov

⁵https://en.wikipedia.org/wiki/Enter_the_Gungeon

jsou předdefinovány, ale jejich rozložení v dungeonu, poklady v nich, nepřátelé a jiné prvky jsou procedurálně generovány.



Obrázek 3.3: Ukázka na které lze vidět grafický styl hry Enter the Gungeon.⁶

Průběh hry spočívá ve vybrání jednoho ze čtyř (později i více) dobrodruhů, poté se hráč vydává do dungeonu. Ten se skládá z několika pater, kde na každém patře je několik místností, ve kterých se nacházejí nepřátelé. Aby se hráč dostal do dalšího patra musí porazit bosse na daném patře a poté může jít do nižšího patra. V průběhu hraní získává hráč různé zbraně a vylepšení, které jeho postavu činí silnější. Pokud hráčova postava zemře, ztratí všechny zbraně a vylepšení, objeví se před vstupem do dungeonu a za měnu, kterou v průběhu hraní hráč získává si může odemknout nové předměty, které se poté mohou objevit v dungeonu při dalších průchodech.

⁶https://store.steampowered.com/app/311690/Enter_the_Gungeon/?l=czech

Kapitola 4

Herní mechaniky

Kapitola herní mechaniky popisuje návrh různých důležitých herních mechanik, které se ve hře objevují. Návrh je nejprve obecný a poté přejde v konkrétnější podobu popisu mechanik tak, aby je bylo možno následně dle tohoto návrhu programovat.

4.1 Návrh

Hráč má svoji postavu, kterou může vybavit různými předměty. Věci, jimiž může hráč vybavit svoji postavu jsou brnění a zbraně. Další předměty, které u sebe postava může mít, jsou například lektvary zdraví.

Postava má statistiky, od kterých se odvíjí její síla. Statistiky jsou útok, obrana, šance na kritický zásah, body zdraví a mana. „*Ve fantasy, především ve hrách na hrdiny a počítačových hrách, je mana častým označením pro magickou energii umožňující sesílat kouzla.*“ [10] Vybavení má také statistiky, nebo lépe řečeno bonusy, které, když je těmito předměty postava vybavena, se promítnou do statistik postavy.

Kromě statistik má postava také úroveň, která se zvyšuje získáváním zkušeností za porážení nepřátel. Zvýšení úrovně znamená zvýšení bodů zdraví a many, možnost nalézt lepší vybavení, ale také zvýšení bodů zdraví a útočného poškození nepřátel.

Hráč se svoji postavou vstupuje do náhodně vygenerovaného dungeonu a objevuje se ve vstupní místnosti, ze které vedou cesty do dalších částí dungeonu. Dungeon se skládá z místností propojených koridory. Při vstupu do místnosti se uzavřou všechny východy z této místnosti, objeví se nepřátelé a hráč je musí porazit, aby mohl pokračovat. Potom, co nepřátele porazí, se východy otevřou a hráč může jít dále.

Cílem hráče je najít místnost s bossem a porazit ho. To se mu ale nepodaří bez dobrého vybavení, takže musí procházet dungeon pořád dokola a získávat stále lepší vybavení, aby byl nakonec schopen bosse porazit. Pokud se hráč dostane do místnosti s východem z dungeonu, může odejít a tím pádem si nechat věci, které v daném průchodu dungeonem získal. Odejít ale nemusí a může pokračovat v prozkoumávání. Pokud v dungeonu jeho postava zemře, ztrácí všechny věci, které doposud našel (kromě věcí, jimiž hráč postavu přímo vybavil) a objevuje se před vstupem do dungeonu. Jediné, co hráči zůstane i po smrti jeho postavy, je nasbíraná měna a zkušenosti.

Po smrti nebo odchodu z dungeonu se hráč objeví na místě, ze kterého do dungeonu vstupuje. Zde má možnost odložit své věci do truhly, aby uvolnil místo v inventáři a ochránil předměty před ztrátou. Další možnost, kterou zde hráč má, je vylepšení zbraní za nasbíranou měnu.

4.2 Programový koncept

V této sekci je popsán koncept přenesení herních mechanik z obecné podoby do návrhu v programovém prostředí. Tato část se nezabývá přímo implementací herních mechanik v herním engine a programovacím jazyku, ale spíše přepisem těchto mechanik do podoby, kterou je poté možno naprogramovat.

4.2.1 Kamera

Pohled hráče na jeho postavu je tzv. ze třetí osoby, což znamená že se hráč dívá své postavě v podstatě „přes rameno“. Kamera je zaměřena na hráčovu postavu a pokud narazí na překážku, tak se k postavě přiblíží natolik, aby mezi kamerou a postavou žádná překážka nebyla. Díky tomu hráč na svoji postavu stále vidí a nic mu nebrání sledovat, co se ve hře odehrává.

4.2.2 Animace

Spuštění, opakování nebo zastavení animací se řídí podle stavu, ve kterém se animovaný objekt nachází a podle vstupů jak od hráče, tak z okolí herního světa. Animované objekty nejsou jen hráčova postava a nepřátelé, ale také například dveře, truhly a jiné objekty.

Pohyb

To, jestli postava stojí, jde nebo běží, se řídí podle rychlosti, jakou se pohybuje. (Pokud se jedná o uzemněný pohyb.) Části animace běhu lze vidět na obrázku 4.1. Animace skoku je aktivována, když hráč dá pokyn pro skok a je vyhodnoceno, že skok může nastat (je-li postava uzemněna). Animace pádu nastane bezprostředně po skoku (je-li to žádoucí) nebo pokud postava není uzemněna například kvůli tomu, že hráč seběhl z nějaké překážky, kterou není možno sejít, kvůli příliš velkému náklonu této překážky.



Obrázek 4.1: Snímky animace běhu hráčovy postavy, které byly pořízeny z testovací scény.

U nepřátel jsou animace méně obsáhlé, protože v této hře protivníci neumí skákat a z povrchů seběhnout nemohou, tudíž nemohou ani padat. U nepřátel je tedy součástí souboru animací pohybu pouze běh a stání.

Souboj

Hráč zadává pokyn pro útok a pokud je postava schopna útočit vzhledem k tomu, jestli je uzemněna a od posledního útoku uplynula dostatečně dlouhá doba, postava započne animaci útoku. Tato animace je různá podle toho, jestli má postava vybavenou nějakou zbraň v hlavním slotu (takzvaném main hand slotu), nebo bojuje beze zbraně, či má vybavený předmět pouze ve vedlejším slotu (takzvaném off-hand slotu). Útok s předmětem vybaveným ve vedlejším slotu je možno vidět na obrázku 4.2.



Obrázek 4.2: Hráčova postavu útočící, zatímco drží vybavenou dýku.

Animace nepřátelských útoků se řídí podle typu nepřátel, kteří mohou mít útok na dálku i zblízka. Pokud se nepřítel dostane dostatečně blízko k hráčově postavě a od jeho posledního útoku uplynula dostatečně dlouhá doba, nastane animace útoku. Jedná-li se o nepřítel, který útočí také na dálku, tak dává útokům na dálku přednost před útoky zblízka. To znamená, že se aktivně nesnaží přiblížit k hráčově postavě, ale přiblíží se pouze natolik, aby byl v potřebné vzdálenosti pro provedení útoku na dálku. Tento nepřítel útočí zblízka pouze, když k němu hráč sám přijde.

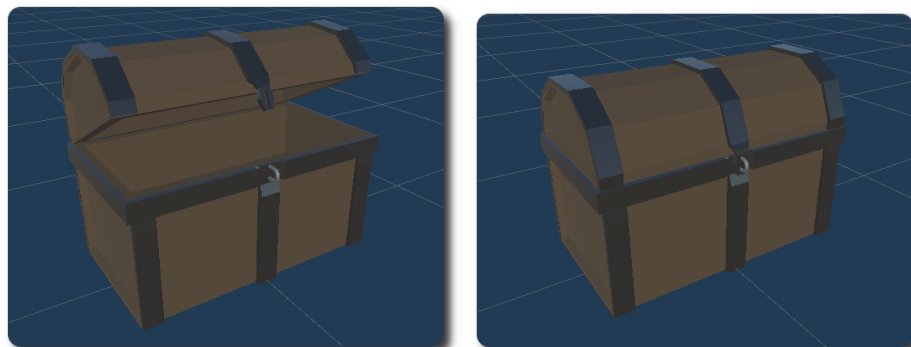
Pokud v průběhu nějaké animace utrhne nepřítel nebo hráč (vyjma animace skoku a pádu) poškození, animace se přeruší a spustí se animace utržení poškození.

Herní svět

Ve světě této hry jsou i předměty, kterým byly přidány animace. Jedním typem těchto předmětů jsou létající ostrovy nacházející se na místě, ze kterého hráč podniká výpravy do dungeonu. Působí výlučně jako dekorace herního světa; konkrétně se zde vznášejí na obloze. Tato animace se opakuje stále dokola a je spuštěna po načtení herního světa automaticky bez jakýchkoliv vstupů.

Dalším takovým předmětem jsou dveře. Ty mají animaci zavření a otevření, které se vybírají podle toho, v jakém stavu dveře zrovna jsou (zavřeny/otevřeny). Animace je spuštěna interakcí hráče s těmito dveřmi.

Posledním animovaným předmětem jsou truhly. Ty mají, stejně jako dveře, animaci otevření a zavření (příklad na obrázku 4.3), která je vybrána podle aktuálního stavu truhly a podnětem spuštění je interakce hráče.



Obrázek 4.3: Otevřená a zavřená truhla zobrazující krajní body animace.

4.2.3 Inventář

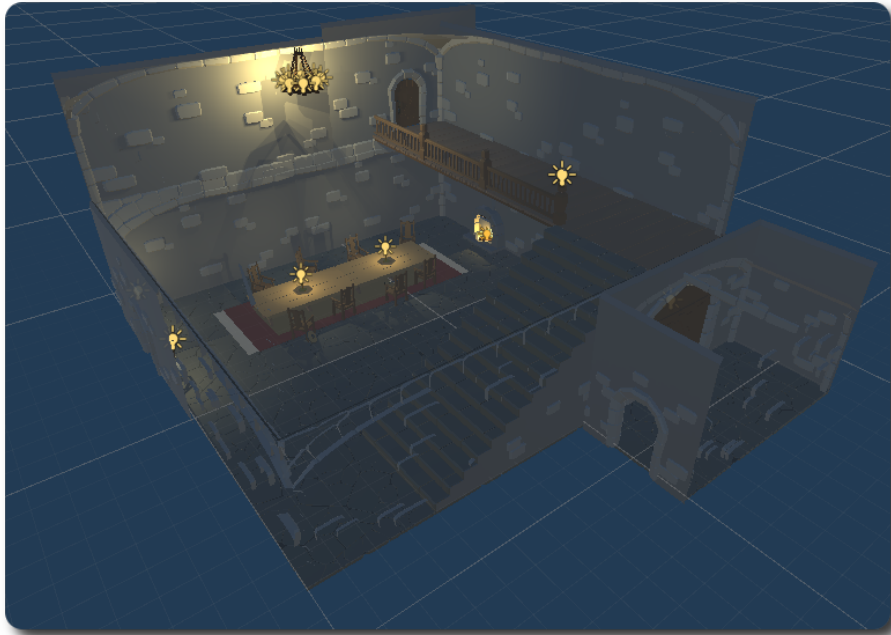
Inventář je rozdělen na dvě části – část pro předměty (zbroj, zbraně, lektvar apod.) a část pro aktuální vybavení postavy.

V části inventáře pro předměty jsou uloženy všechny věci, které hráč posbírá při průchodu dungeonem a předměty, které si s sebou do dungeonu vzal. Takové předměty mohou být například lektvary na doplnění bodů zdraví. Sekce předmětů také zobrazuje množství měny, které má hráč k dispozici.

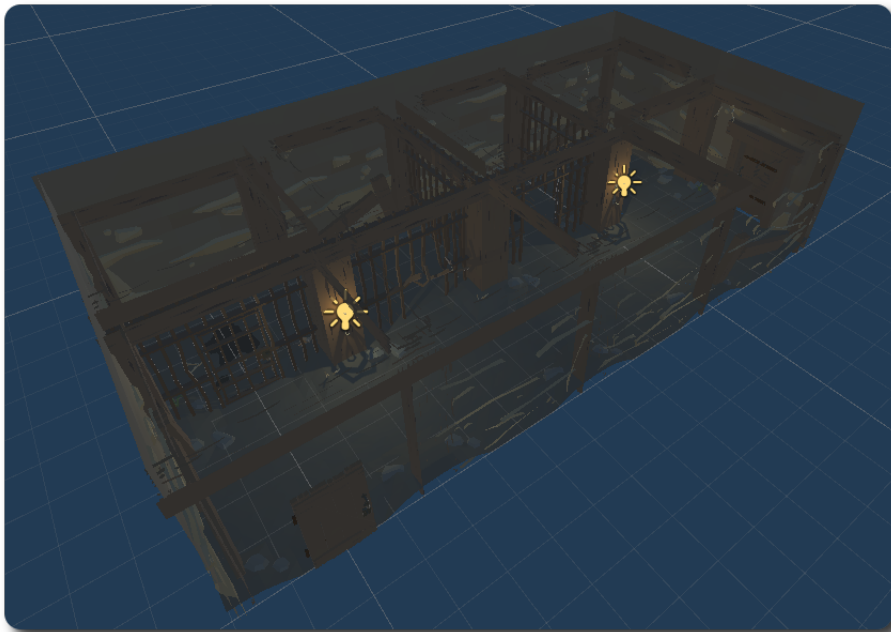
Sekce vybavení zobrazuje to, čím je postava zrovna vybavena. Je zde několik slotů – helma, hrud, nohy, boty a zbraně. Dále tato část inventáře zobrazuje statistiky postavy - body zdraví, mana, útok, šance na kritický zásah a obrana.

4.2.4 Dungeon

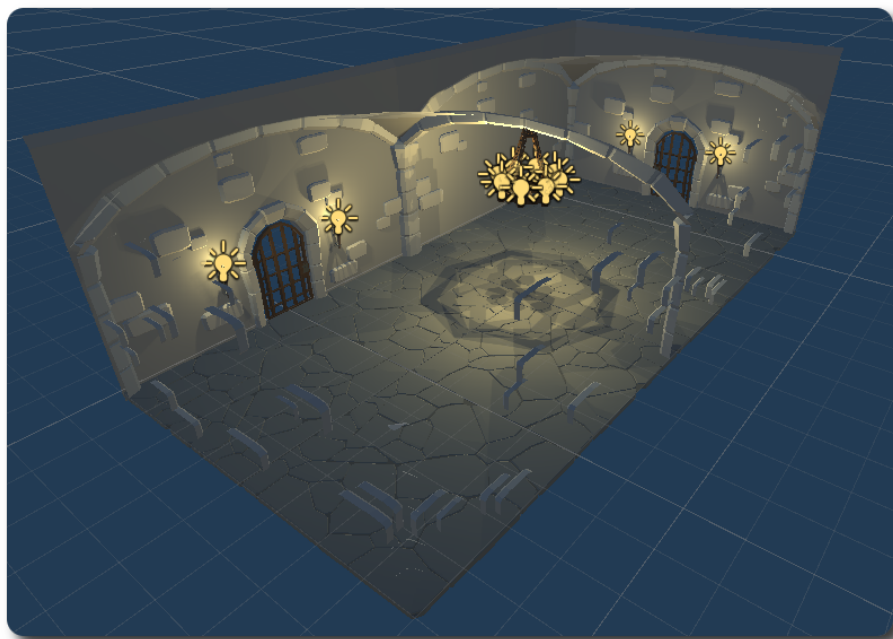
Dungeon se skládá z místností (obrázky 4.4, 4.5 a 4.6), koridorů a dekorací. Je náhodně generován podle předem daných pravidel (maximální počet místností, maximální počet koridorů na místnost). Generování probíhá na bázi připojování předdefinovaných místností a koridorů na východy z již vygenerovaných místností a koridorů. Generování začíná první, neboli vstupní, místností (se čtyřmi, třemi, nebo dvěma východy, kterými se hráč může dát), takže se dungeon rozrůstá část po části od svého středu.



Obrázek 4.4: Místnost typu s \ddot{a} n, je \acute{z} je největší místností dungeonu a jediná, která má více pater. Jsou zde některé dekorace, které jsou neměnné, například stoly, \acute{z} idle, krb a další. Ostatní dekorace se náhodně generuj \acute{i} .



Obrázek 4.5: \acute{Z} alář je nejmenší místností v dungeonu. Je zde velmi málo místa na uhýbání nepřátelským útokům, proto může hráči připadat těžší místnost zdolat.



Obrázek 4.6: Velká místnost s dostatečným místem pro pohyb a uhýbání nepřátelským útokům.

Po započítání generování se náhodně vybírá, který předdefinovaný objekt se použije. Na koridor může navazovat místnost nebo koridor, ale na místnost může navazovat pouze koridor. Pokud je vybrán následující objekt a zjistí se, že se na dané místo nevejde, zkusí se jiný objekt, dokud stále existuje nějaký, který nebyl vyzkoušen. Nastane-li situace, že se žádná z předdefinovaných místností nebo koridorů na požadované místo nevejde, východ se uzavře stěnou.

Je-li splněna některá z ukončujících podmínek, generování rozložení místností se ukončí a zkontroluje se, jestli se vygeneroval zadaný počet místností. Pokud je počet místností menší, než je požadovaný počet, generování se opakuje. Je-li počet místností správný, vygenerují se dekorace.

Dekorace jsou také různé předdefinované objekty, které se vygenerují na předem určené body v místnostech. Aby místnosti nevypadaly stále stejně, tak je zde také přidán prvek náhody. Na místa určená k vygenerování dekorací se dekorace generují náhodně ze seznamu potenciálních dekorací.

Mimo dekorace se také v místnostech na určených místech vygenerují truhly, ve kterých může hráč nalézt vybavení a měnu. Ne v každé místnosti se truhla vygeneruje. Je zde pouze šance, že se tak stane, a její umístění se vybírá náhodně z předem určených míst.

Po vygenerování rozložení místností se místnost, která se vygenerovala jako poslední, nastaví jako místnost s bossem. Je také náhodně vybrána jedna místnost a ta bude místností, ze které může hráč odejít z dungeonu. Touto místností nikdy není právě místnost s bossem.

4.2.5 Nepřátelé

Když hráč vstoupí do nějaké místnosti, zavřou se za ním dveře a uzavřou se všechny ostatní východy z této místnosti. Také se objeví nepřátelé, kteří se pokusí zabít hráčovu postavu.

Pokud hráč porazí všechny nepřátele v místnosti, východy se otevřou a hráč může pokračovat dungeonem dále.

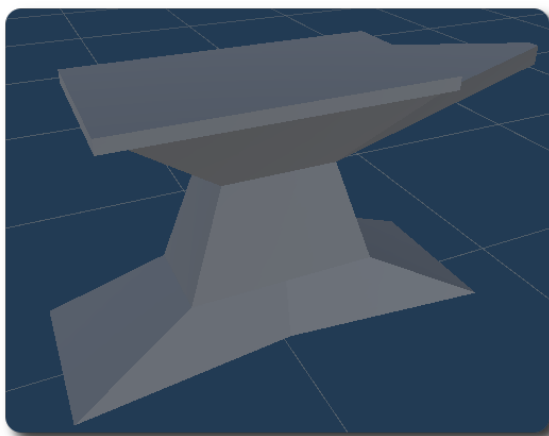
Existují zde dva druhy nepřátel - útočící zblízka a útočící (primárně) na dálku. Tyto dva typy jsou zastoupeny kostlivcem a golemem (útočí zblízka) a morovým doktorem (útočí na dálku). Golem se ve hře nachází pouze jako boss. Herní modely nepřátel lze vidět na obrázku 4.7.



Obrázek 4.7: Lze zde vidět nepřátele, které je možné ve hře potkat. Těmi jsou (zleva) kostlivec, morový doktor a golem

4.2.6 Vylepšování předmětů

Některé předměty, konkrétně některé zbraně, lze vylepšovat a tím zvyšovat bonusy, které přidávají. Vylepšení je možno provést u kovadliny (jejíž model lze vidět na obrázku 4.8) za nasbíranou měnu.



Obrázek 4.8: Kovadlina používaná pro vylepšování předmětů za nasbíranou měnu.

Kapitola 5

Implementace

Kapitola se zabývá implementací herních mechanik v herním enginu Unity a přenesením programového konceptu do podoby kódu v programovacím jazyce C#. Mechaniky a chování objektů jsou v Unity implementovány pomocí skriptů využívajících API, které Unity poskytuje. Samozřejmě je možné využít vše, co nabízí jazyk C# sám o sobě.

Nejdříve je rozebráno, jak je vše implementováno v prostředí Unity a jak toto prostředí pracuje. Druhou částí bude pohled přímo na to, jak spolu komunikují jednotlivé skripty a objekty.

5.1 Prostředí Unity

Projekt v Unity je rozdělen do jednotlivých scén, které se podle potřeby načítají a přepínají. Do těchto scén se vkládají (ručně nebo pomocí skriptů) různé herní objekty. Objekty mají hierarchickou strukturu a mohou obsahovat mnohé komponenty a skripty, které mohou být také považovány za komponenty. Komponenty lze k hernímu objektu opět přiřadit ručně nebo za pomoci skriptu. Ke komponentům a samotným objektům se přistupuje skrze skripty. Důležité je, že komponenty nemohou existovat sami o sobě, ale musí být přiřazeny k nějakému hernímu objektu.

Herní objekty mohou mít určen tag, podle kterého je možno objekty vyhledávat, a také mají určenou vrstvu, do které patří. Každý herní objekt má také vždy komponent **Transform** (**RectTransform** u uživatelského rozhraní), který určuje pozici, rotaci a měřítko na jednotlivých osách souřadného systému. Pokud je objekt kořenovým objektem, jedná se o pozici, měřítko a rotaci globální, tedy vzhledem k souřadnému systému celé scény. Jedná-li se ale o potomka nějakého jiného herního objektu, mluvíme o pozici, měřítku a rotaci lokální, tedy vzhledem k rodičovskému objektu. Každá transformace u rodičovského objektu ovlivňuje i potomky tohoto objektu.

Jak nepřátelé, místnosti, dekorace, tak i hráčova postava a další prvky scény jsou herními objekty. Objekty, které jsou předem vytvořené a do scén se poté vkládají podle potřeby, se nazývají prefaby.

Tento projekt je rozdělen do tří scén: hlavní menu, svět se vstupem do dungeonu a samotný dungeon.

5.1.1 Uživatelské rozhraní

Komponentem umožňujícím zobrazení uživatelského rozhraní je `Canvas`. Vždy když chceme zobrazit nějaký objekt s prvky uživatelského rozhraní (text, obrázek, posuvník, tlačítko a tak podobně), musí být tento objekt potomkem objektu s komponentem `Canvas`.

Největší vliv na vzhled uživatelského rozhraní má použití komponentu `Image` nebo `RawImage`, u kterých je možno nastavit obrázek, který budou zobrazovat a také upravit jeho barvu.

Hlavní menu

Hlavní menu je scéna, která obsahuje pouze uživatelské rozhraní. Na její vzhled lze nahlédnout na obrázku 5.1. Jsou zde dva hlavní objekty, kterými jsou hlavní okno a okno s nastavením. V hlavním okně je tlačítko pro započítí nové hry, tlačítko pro načtení uložené hry a tlačítko pro zobrazení okna s nastavením. Při stisknutí tlačítka pro novou hru nebo tlačítka pro načtení uložené hry se načte nová scéna a scéna s hlavním menu se opustí.



Obrázek 5.1: Zde lze vidět vzhled scény hlavního menu. Hlavní menu slouží k započítí nové hry, načtení uložené hry a změny nastavení zvuků.

Dále je v pravém dolním rohu tlačítko pro zavření aplikace (ukončení hry) a v levém dolním rohu jsou dvě další tlačítka pro ovládání zvuku. Prvním z těchto dvou tlačítek lze vypnout/zapnout zvuk prostředí a hudba, druhým tlačítkem lze vypnout/zapnout ostatní zvuky (zvuk dupaní, boje a tak podobně). Všechny tlačítka obsahují komponent `Button`, který slouží právě k přijetí a zpracování události kliknutí na příslušný herní objekt.

Při kliknutí na tlačítko nastavení se herní objekt obsahující prvky hlavního okna deaktivuje a aktivuje se herní objekt obsahující prvky okna s nastavením. V tomto okně se nalézají posuvníky sloužící k určení hlasitosti audia. Funkčnost posuvníků zařizuje primárně komponent `Slider`. Pod posuvníky se nachází stejná dvě tlačítka pro vypnutí/zapnutí zvuků, jako jsou v hlavním okně. Posledním tlačítkem v tomto okně je tlačítko, které deaktivuje

herní objekt s oknem nastavení a aktivuje herní objekt s hlavním oknem, tudíž přesune hráče zpět do hlavního okna.

Rozhraní používané při samotné hře

Jako první je třeba se zmínit o modálních okně, které je využíváno při potvrzení, nebo zamítnutí nějaké akce. Použito je u inventáře, u vstupu do dungeonu, východu z dungeonu a u vstupu do místnosti s bossem. Důležité části okna jsou: hlavní text, jenž popisuje akci, kterou se hráč chystá vykonat, dále tlačítko k potvrzení akce a tlačítko pro zavření ona, tedy pro odmítnutí akce. Vzhled modálního okna je představen na obrázku 5.2.



Obrázek 5.2: Modální okno využívvané pro potvrzení, nebo odmítnutí příslušné akce.

Kromě hlavního menu obsahuje hra mnohé další prvky uživatelského rozhraní. Jedněmi z těch prvků, kterých si hráč může všimnout hned, když začne hrát, jsou ukazatele bodů zdraví a many, vedle kterých je ukazatel aktuální úrovně hráčovy postavy (obrázek 5.3). Dalším takovým prvkem je ukazatel zkušeností u spodního okraje obrazovky.



Obrázek 5.3: Rozhraní zobrazující body zdraví, many a úroveň hráčovy postavy. Množství bodů zdraví a many zobrazuje nejen posuvníkem, ale také čísly. Konkrétně lze vidět aktuální stav následovaný stavem maximálním.

Ukazatelé bodů zdraví, many a zkušeností jsou opět primárně tvořeny pomocí komponentu `Slider`, tedy posuvníku, a komponentu `TextMeshProUGUI` z balíčku `TMPPro`. (Všechny textové prvky v tomto projektu jsou tvořeny pomocí tohoto komponentu.) Textový komponent se zde využívá pro zobrazení čísel ukazujících aktuální a maximální hodnotu posuvníku.

Tyto hodnoty reprezentují aktuální a maximální hodnotu bodů zdraví, many a zkušeností. Ukazatel úrovně je pouze text na pozadí, které je reprezentováno obrázkem vloženým pomocí komponentu `RawImage`.

Stiskne-li hráč klávesu *Escape*, hra se pozastaví a otevře se menu, které lze vidět na obrázku 5.4. Toto menu je skoro stejné jako hlavní menu s tím rozdílem, že toto menu nemá tlačítka pro novou hru a načtení hry, ale místo nich jsou zde tlačítka pro zavření menu a odchod do hlavního menu.



Obrázek 5.4: Menu do kterého se hráč ve hře dostane stiskem klávesy *Escape*. Je velmi podobné hlavnímu menu s tím rozdílem, že toto menu nemá tlačítka pro novou hru a načtení hry, ale místo nich jsou zde tlačítka pro zavření menu a odchod do hlavního menu.

Dalším prvkem uživatelského rozhraní je inventář postavy (obrázek 5.5), který se zobrazí stisknutím příslušné klávesy (klávesa *I*). Část inventáře pro předměty obsahuje osm míst pro předměty (takzvaných slotů), ve kterých se předměty zobrazují. Pokud je předmět takový, který může tvořit svazky (jeden slot pojme více kusů tohoto předmětu), slot zobrazuje aktuální počet kusů. Vizualní část předmětu v inventáři se skládá z ikony předmětu a rámečku předmětu. Barva rámečku je odvozena od vzácnosti předmětu.



Obrázek 5.5: Inventář, kde lze vidět předměty a množství nasbírané měny, a část pro zobrazení výbavy a statistik, kde lze vidět i model hráčovy postavy.

V inventáři je možno, kromě předmětů, najít tlačítko sloužící k přepnutí módu inventáře na mód, ve kterém se kliknutím na slot s předmětem příslušný předmět označí (jeho rámeček zčervená, jak je vidět na obrázku 5.6) a po vybrání všech požadovaných předmětů a opětovném kliknutí na přepínač je hráči zobrazeno modální okno upozorňující na to, že se chystá předměty z inventáře odstranit. Pokud hráč akci potvrdí, předměty jsou odstraněny a inventář se přepne do původního módu. Odmítne-li hráč akci, rámečkům se změní barva na barvu podle vzácnosti příslušného předmětu a mód inventáře se opět přepne do původního módu.



Obrázek 5.6: Mód označení předmětů v inventáři. Předměty zvýrazněny červeným rámečkem jsou určeny pro odstranění z inventáře.

Posledním prvkem části předmětů v inventáři je zobrazení měny, kterou hráč vlastní. Zobrazení se skládá z obrázku zlaté mince, vedle kterého je číslo reprezentující aktuální množství hráčem vlastněné měny.

Druhou částí inventáře je část s vybavením a statistikami postavy. Tato část obsahuje pohled na hráčovu postavu, sloty pro výbavu (zbroj, zbraně) a statistiky zobrazující aktuální hodnoty bodů zdraví, many, síly útoku a obrany a procentuální šanci na kritický zásah.

Dalším prvkem uživatelského rozhraní ve hře je rozhraní pro vylepšování předmětů zobrazené na obrázku 5.7. Zde se zobrazují předměty, tedy alespoň ty jež lze vylepšit, jak z inventáře, tak z vybavených předmětů. U každého takového předmětu lze vidět jeho ikonu s rámečkem, bonusy, které poskytuje, a zelenými čísly je vedle těchto bonusů napsáno, o kolik se dané bonusy zvýší, vylepší-li hráč tento předmět. Úplně napravo u každého předmětu je ikona kladiva, která je tlačítko, jehož událost kliknutí vylepší příslušný předmět. Předměty, které se dají vylepšit, lze vylepšit pouze dvakrát (z první úrovně na druhou a z druhé úrovně na třetí, tedy nejvyšší). Je-li předmět vylepšen na třetí úroveň, je v okně pro vylepšování předmětů zobrazen jinak, a to zobrazením jeho ikony a nápisu, který hráče informuje o tom, že je předmět vylepšen na maximální úroveň.



Obrázek 5.7: Vzhled rozhraní pro vylepšování předmětů, kde lze vidět jak předměty, které lze ještě vylepšit, tak předmět, který je vylepšen na maximální úroveň.

Posledním prvkem okna je posuvná lišta umožňující zobrazit předměty, které se nevejdou do části pro předměty určené. Tuto lištu je poměrně složitější implementovat než posuvníky zmíněné výše. Je tvořena pomocí komponentů `Scrollbar`, `ScrollRect` a `Mask`. Komponent `ScrollRect` slouží k určení viditelného místa (pomocí nastavení jeho velikosti v komponentu `RectTransform` téhož herního objektu), ve kterém se budou posouvané elementy zobrazovat. Tomuto komponentu je přiřazen herní objekt s komponentem `Scrollbar` a herní objekt, který je v hierarchii objektů rodičovským objektem pro objekty zobrazující jednotlivé předměty, tedy je objektem s obsahem, který je posouván. Velikost rodičovského objektu herních objektů s předměty musí být dostatečně velký, aby se do něj všichni potomci vešly. (Jeho `RectTransform` komponent musí mít takové rozměry, aby pojal velikost komponentů `RectTransform` všech objektů s předměty.) Pokud je v objektu s obsahem tolik předmětů, že přesahuje rozměr tohoto objektu rozměr objektu s komponentem `ScrollRect`, tak obsah přesahuje rozměry místa vymezeného pro zobrazení předmětů. K tomu, aby obsah nepřesahoval, je použit komponent `Mask`, který zajistí, že je vidět pouze ten obsah, který se právě nachází v místě vymezeném rozměry objektu s komponentem `ScrollRect`. Posuvná lišta je ovládána kliknutím a podržením levého tlačítka myši a následným tažením myši ve směru, kterým hráč chce obsah posunout, nebo použitím kolečka myši.

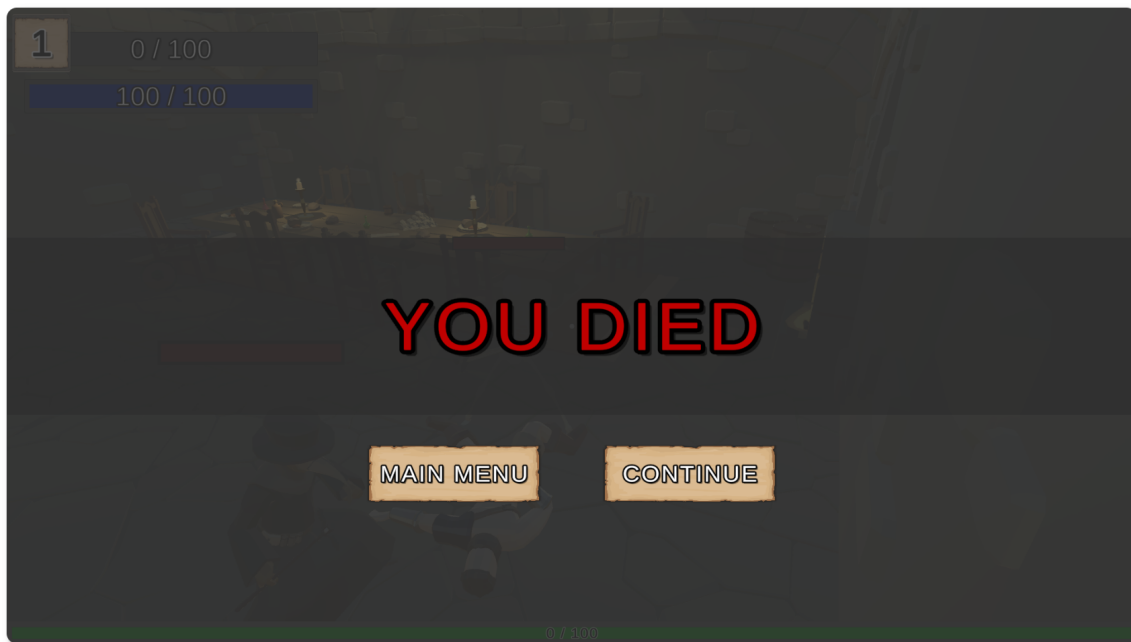
Hráč si může své předměty uložit v truhle. Pro ukládání do truhly a zobrazení uložených předmětů slouží uživatelské rozhraní obsahující dvě části, které lze vidět na obrázku 5.8. V první, hlavní, části je vidět třicet slotů pro uložené předměty. Pokud je uložených před-

mětů více než třicet, automaticky se přidá více slotů; úložný prostor je tedy neomezený. Pokud je slotů více než třicet, k jejich zobrazení je opět třeba posuvná lišta, která je implementována stejně jako lišta zmíněna u předchozího uživatelského rozhraní. Ve druhé části se zobrazují předměty, které má hráčova postava zrovna v inventáři. Klikne-li hráč na jeden z předmětů, přesune se tento předmět do truhly. Totéž platí, pokud chce hráč přesunout předmět z úložného prostoru do inventáře.



Obrázek 5.8: Rozhraní hráčovy truhly, kam si může ukládat předměty.

Pokud hráč v dungeonu selže a jeho postava zemře, zobrazí se uživatelské rozhraní, které informuje o smrti postavy a poskytuje možnost pokračovat dále (obrázek 5.9). Hlavní prvky zmíněného rozhraní jsou: text informující o smrti postavy, který je animovaný a po chvíli se zvětší do čitelné podoby, a dvě tlačítka, kde jedno přesune hráče zpět do hlavního menu a druhé ukončí průchod a následně vrátí hráče s jeho postavou před vstup do dungeonu.

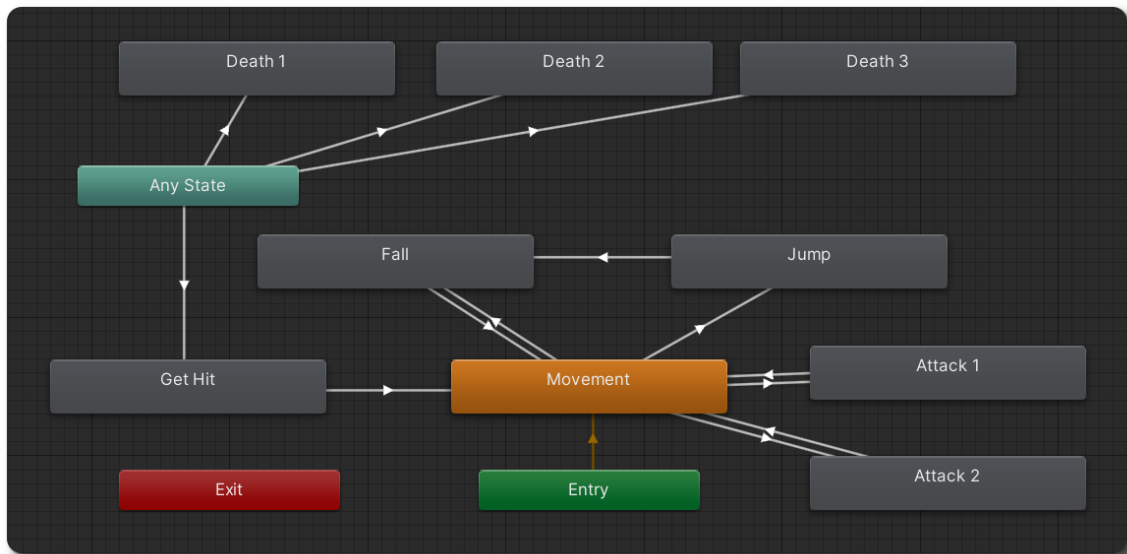


Obrázek 5.9: Na tomto obrázku lze vidět vzhled okna, které se zobrazí po smrti hráčovy postavy. Okno informuje o selhání v dungeonu a poskytuje možnost pokračovat dále, nebo odejít do hlavního menu.

Poslední věcí, kterou je třeba zmínit v této části, je to, že u většiny rozhraní (obsahujících nějaké sloty, potažmo předměty) pokud přesune hráč kurzor myši nad slot obsahující nějaký předmět, zobrazí se informace o daném předmětu - název, vzácnost, úroveň, úroveň vylepšení (je-li předmět možné vylepšit) a bonusy, které poskytuje. Z barvy textu lze také rychle rozeznat, jakou má předmět vzácnost bez toho, aby bylo nutno tuto informaci přečíst.

5.1.2 Animace

V Unity je k použití animací třeba, aby měl herní objekt komponent `Animator`, který využívá prvek `Animator Controller`. Na `Animator Controller` pro animace hráčovy postavy lze nahlédnout na obrázku 5.10, na němž je vidět, že je to vlastně stavový automat, kde stavy jsou jednotlivé animace a mezi nimi jsou přechody. Přechody mohou být podmíněné, nebo bez podmínky. Podmíněné přechody se řídí hodnotami různých proměnných, které jsou jim nastaveny. Nepodmíněné přechody se provedou po dokončení animace. U jednotlivých stavů je také možné vytvořit skript, který ve své třídě dědí z třídy `StateMachineBehaviour`. Takové skripty mohou obsahovat metody, které se provedou v určitém bodu stavu, ke kterému je skript přidán; například při vstupu do stavu, odchodu ze stavu a tak podobně.



Obrázek 5.10: Animator Controller (v podobě stavového automatu) ovládající animace hráčovy postavy.

5.1.3 Kamera

Kamera v tomto projektu zobrazuje pohled ze třetí osoby, tedy hráč nevidí očima svojí postavy, ale vidí ji z větší dálky a kouká jí tzv. „přes rameno“. Kamera je ovládána pohybem myši, ke kterému se přistupuje přes osy Unity. Hodnota osy *Mouse X* vyjadřuje pohyb myši vlevo a vpravo, hodnota osy *Mouse Y* vyjadřuje pohyb myši dopředu a dozadu. Pomocí vstupů z těchto os je určen pohyb kamery, kdy pohyb myši vlevo a vpravo otáčí i se samotnou hráčovou postavou (podle osy *Y* souřadného systému Unity) a pohyb myši dopředu a dozadu rotuje pouze se samotnou kamerou (podle osy *X*). Rotace podle osy *X* má takové omezení, že nelze dostat kameru dále než přímo nad, nebo přímo pod hráčovu postavu. Kdyby takové omezení neměla, bylo by možné kameru přetočit tak, že by se dostala před hráčovu postavu a byla takzvaně „vzhůru nohama“, což je samozřejmě nežádoucí.

V módu pohledu ze třetí osoby je třeba zařídit, aby při vstupu nějaké překážky mezi kameru (pohled hráče) a hráčovu postavu hráč stále na svoji postavu viděl. Toho je docíleno přiblížením kamery před onu překážku. Až se kamera opět dostane do bodu, kde mezi ní a hráčovou postavou není žádná překážka, posune se zpět na danou vzdálenost od postavy.

K implementaci kýžených mechanik popsaných výše je použita Cinemachine camera z balíku Cinemachine.

5.1.4 Pohyb

K pohybu hráče je také využíván vstup z os - horizontální a vertikální. Horizontální osa reprezentuje vstup kláves *A* a *D* a je využita pro pohyb vlevo a vpravo. Vertikální osa reprezentuje klávesy *W* a *S* a ovládá pohyb dopředu a dozadu.

Na herním objektu hráčovy postavy je použit komponent `CharacterController`, který poskytuje funkce a nastavení pro pohyb (rychlost pohybu, maximální náklon terénu, který je postava schopna vyjít, a tak podobně).

Hráčova postava je nastavena tak, že se může pohybovat pouze po objektech, které jsou ve vrstvách `Walkable` a `Interactable`, tedy po objektech, které jsou ve vrstvě určené

pro objekty, po kterých lze chodit, a po objektech ve vrstvě pro objekty, se kterými lze interagovat (například truhly).

Pohyb nepřátel je řízen, mimo jiné, pomocí komponentu `NavMeshAgent`, který opět poskytuje potřebné funkce a umožňuje nastavení různých hodnot (rychlost pohybu, zrychlení a tak dále). Je nutno podotknout, že nepřátelé se mohou pohybovat pouze tam, kde je vytvořena navigační síť (`NavMesh`), protože právě díky té komponent `NavMeshAgent` funguje tak, jak má.

5.1.5 Soubojový systém

Boj je založen na uhýbání nepřátelským útokům a následném odvetném útoku, když jsou bezbranní. Zaútočí-li totiž hráč nebo nepřítel, může znovu zaútočit až po určité době. V průběhu útoku se nelze pohybovat (s výjimkou otáčení), proto v tuto chvíli může hráč nepřátelskému útoku uhnout. To, jestli je hráč nebo nepřítel útokem zasažen, se rozhodne pomocí vytvoření neviditelné koule v místě útoku, která zaznamená objekty, které jsou uvnitř. Poté se tyto objekty prochází a je-li jedním z těchto objektů hráč, potažmo nepřítel, vyšle se k tomuto objektu paprsek od útočnicka. Trefí-li paprsek daný objekt, znamená to, že mezi útočnickem a objektem není žádná překážka (například zeď) a útok tedy uspěl.

V případě nepřátelských útoků na dálku souboj probíhá skoro stejně s tím rozdílem, že útočník nemusí přijít až k hráči, ale stačí, aby byl v dostatečné vzdálenosti pro útok na dálku. Útok na dálku má určenou delší dobu před tím, než může být proveden znovu. Útok je proveden na hráčovu pozici. (Většinou je seslán na pozici, kde se hráč nacházel před chvílí, aby se útokům dalo vyhnout.) Na daném místě je opět vytvořena koule, díky které se kontroluje, jestli byl útok úspěšný. Zde se ale paprsek vysílá ještě před samotným útokem a to proto, aby byl útok proveden až potom, co je ustanoveno, že mezi útočnickem a hráčem není žádná překážka. I nepřátelé útočící na dálku mají možnost provádět útoky zblízka. Tyto útoky provádí pouze, pokud se hráč dostane do bezprostřední blízkosti k nim.

Samotný útok probíhá tak, že se spustí animace útoku a v určitém momentě této animace se spustí událost útoku a až právě ta vyvolá zmíněnou neviditelnou kouli, potažmo udělí poškození.

Ať už se jedná o útoky na dálku nebo zblízka, pokud je útok úspěšný, tak u toho, který byl útokem zasažen, je přehrána animace, která vyobrazuje daného nepřítele, nebo hráčovu postavu, jako trefeného útokem. Pokud trefený v danou chvíli zrovna útočí (tedy je přehrávána animace útoku) a útok ještě nebyl plně proveden (nenastala událost útoku), tak se animace útoku přeruší společně se samotným útokem, tudíž se událost útoku neprovede. V tomto případě ten, kdo utřžil zranění, nemusí čekat danou dobu před dalším útokem, ale může útočit hned znovu. U hráčovy postavy se animace utržení poškození nevyvolá, pokud není postava uzemněna (postava skáče nebo padá). Míru utrženého zranění a množství zbývajících bodů zdraví zobrazují posuvníky, které u hráče lze nalézt v levém horním rohu obrazovky a u nepřátel je lze vidět přímo nad jejich herními modely. Dostane-li se množství bodů zdraví na nulu, přehraje se animace smrti a hráčova postava nebo příslušný nepřítel umírá. Jestliže zemře nepřítel, hráč získá body zkušeností, které ho posunou blíže k získání další úrovně.

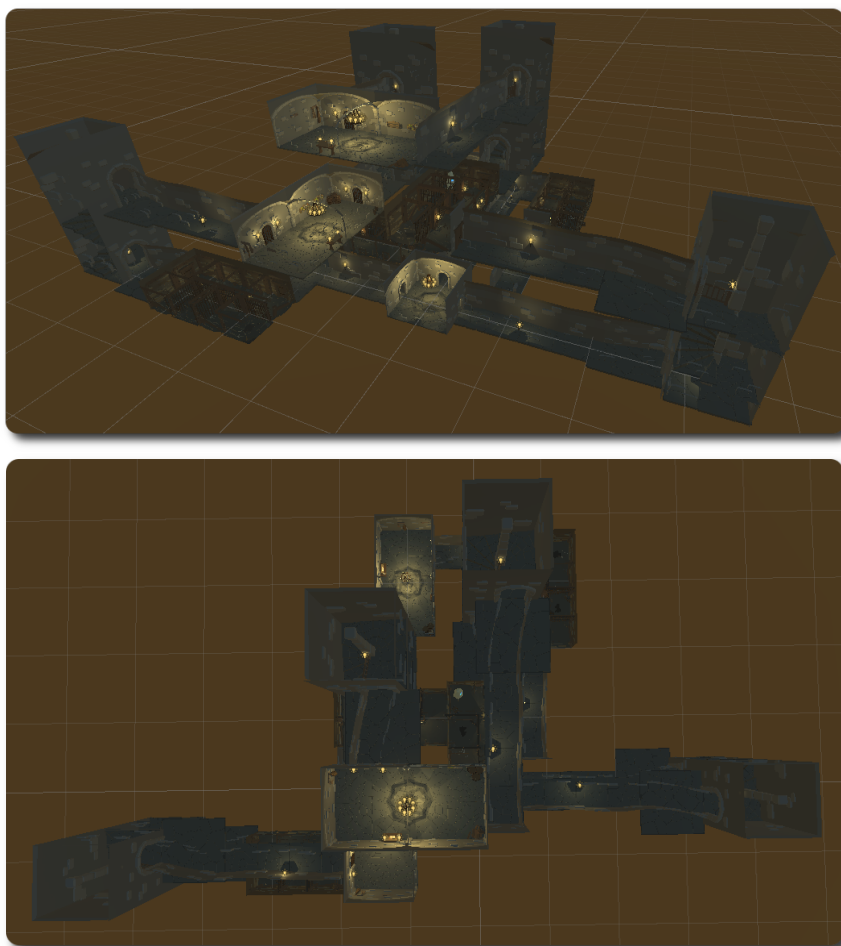
Pokud hráč použije lektvar zdraví (kliknutím na jeho ikonu v inventáři), doplní tak své postavě body zdraví. Množství doplněných bodů se odvíjí od síly a typu lektvaru. Chce-li ale hráč použít lektvar v průběhu boje, musí tak učinit velmi rychle, protože při otevření inventáře není zastaven čas a nepřátelé útočí dále, zatímco hráč má otevřeno rozhraní inventáře a nemůže se pohybovat.

5.1.6 Dungeon

Jak bylo popsáno, dungeon se skládá z místností propojených koridory. Ty jsou předem vytvořeny a následně se náhodně skládají k sobě, čímž vytvoří celou herní mapu (obrázek 5.11). Místnosti mají pevně danou velikost, tvar a osvětlení (lustry, louče apod.), ale ostatní prvky jsou náhodně generovány.

Místnosti (a koridory) samotné mají pevně dané body u východů, na které se může připojit další místnost nebo koridor. Ty se připojují k danému bodu jiným bodem, tentokrát určeným jako bod, kterým se místnost nebo koridor připojuje. Tento bod je u vstupu do místnosti.

Pokud se nějaká místnost vygeneruje a překrývá se s jinou místností, je místnost, která se vygenerovala poději, odstraněna. Tyto překryvy jsou detekovány pomocí komponentu `Collider` nastaveném v módu `trigger` a komponentu `RigidBody`. Tyto komponenty jsou vloženy v hierarchii objektů jednotlivých prefabů místností. Pokud se tedy místnost s komponentem `RigidBody` ocitne v prostoru vymezeném komponentem `Collider` jiné místnosti, tato skutečnost je zaregistrována a rozhoduje se, která z místností zůstane a která bude odstraněna. Aby byla ve správném čase zachycena a zpracována kolize, jsou pro generování použity korutiny, které průběžně vrací kontrolu Unity a tím pádem je možné, aby byla kolize zpracována před vygenerováním další místnosti.



Obrázek 5.11: Pohled na vygenerovaný dungeon v prostředí Unity editoru.

Dekorace v místnostech mají určená místa, kde se generují, a na těchto místech jsou nastaveny ty dekorace, které se na daném místě mohou nacházet a z těch se náhodně generuje jedna, která bude na dané místo umístěna. Nepřátelé mají také určená místa, kde se mohou objevit. Oproti dekoracím se ale náhodně vybírá nejen typ nepřítele na daném místě (pokud je na daném místě možnost více druhů nepřátel), ale také jestli se na daném místě nepřítel vůbec objeví. V místnosti se tedy nemusí objevit nepřátelé na všech možných místech, vždy se ale musí objevit alespoň jeden. Dalším rozdílem vzhledem k dekoracím je to, že nepřátelé se objeví až po vstupu hráče do místnosti, zatímco dekorace se generují při počátečním generování celého dungeonu. V místnostech je také určeno místo, na kterém se objeví boss, je-li místnost určena jako místnost s bossem, a také místo pro objekt používaný pro odchod z dungeonu, pokud je daná místnost pro východ určena. Předmět určený pro opuštění dungeonu je obelisk, který lze vidět na obrázku 5.12. Příklad všech bodů určených pro generování jednotlivých herních objektů je představen na obrázku 5.13.

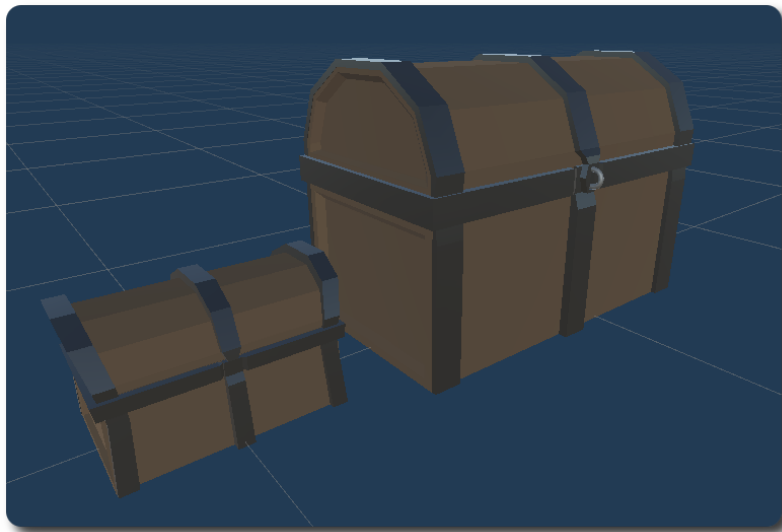


Obrázek 5.12: Vzhled obelisku, který se využívá pro přesun do dungeonu a zpět.



Obrázek 5.13: Velká místnost se zvýrazněnými body pro generování dekorací (žluté), truhel (fialové), nepřátel (zelené), obelisku (modrý) a bosse (červený).

Dalším prvkem v místnosti, který má určená místa, kde se může objevit, je truhla. Truhla se neobjeví vždy (je zde pouze šance) a pokud se objeví, tak náhodně na jednom z míst k tomu určených. Může nabývat dvou podob; malé truhly a velké truhly (obrázek 5.14). Vzhled je pouze kosmetický a nemá vliv na množství ani vzácnost předmětů, které lze v truhle najít.



Obrázek 5.14: Ukázka variant truhel, které se v dungeonu mohou objevit.

5.2 Provedení v prostředí programovacího jazyka

Unity API poskytuje mnoho důležitých tříd a funkcí pro práci s herními objekty, komponenty a dalšími prvky prostředí Unity. Většina potřebných funkcí se nachází v balíku `UnityEngine`. Díky tomuto API lze tvořit za pomoci jazyka C# skripty. Skripty jsou vlastně komponenty, takže je lze umisťovat na herní objekty. Tedy alespoň skripty s třídami, které dědí z `MonoBehaviour`, což je základní třída, ze které dědí všechny komponenty. Použit lze i skripty, které nedědí z `MonoBehaviour`, ty ale nelze vložit jako komponenty na herní objekty.

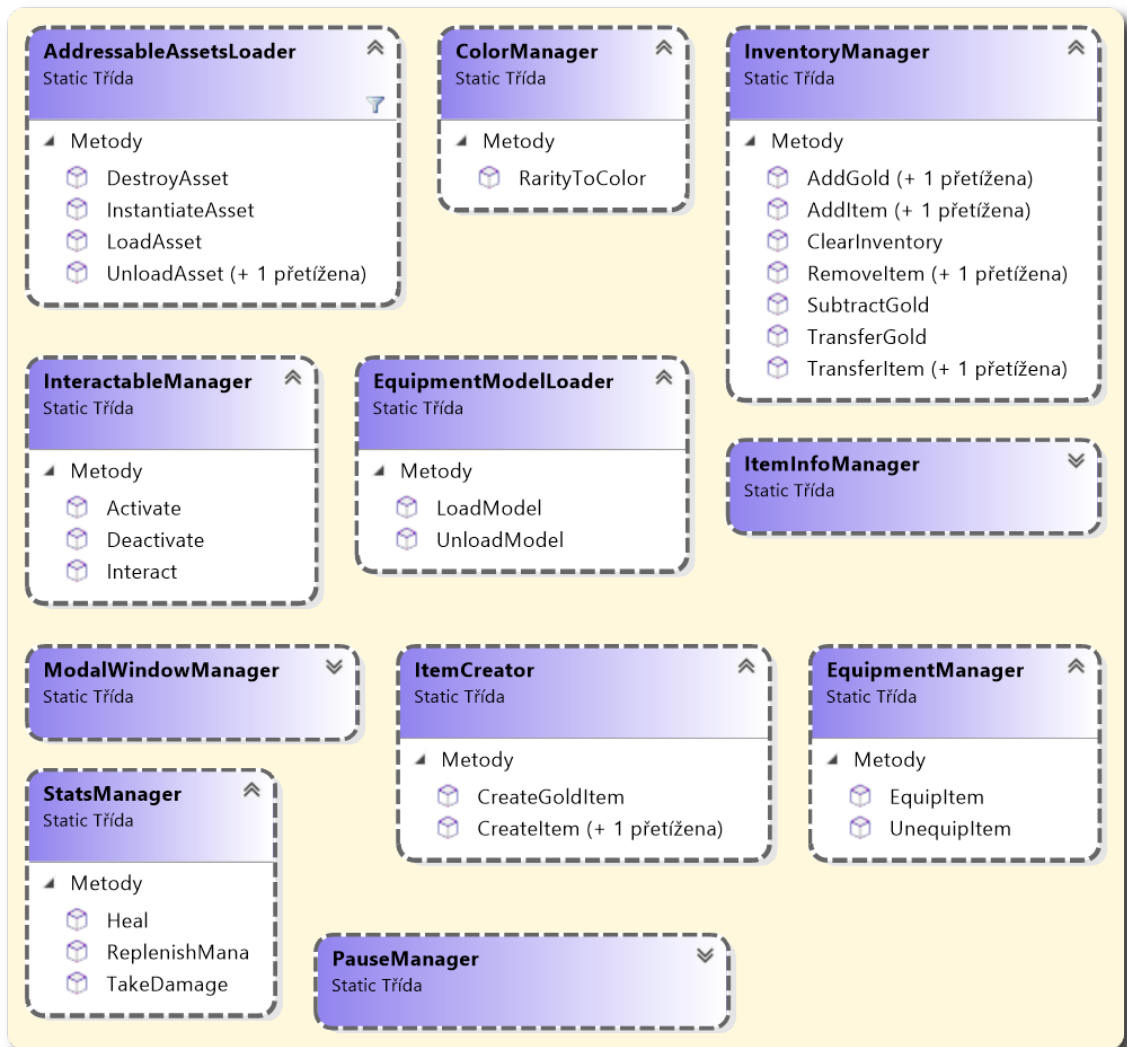
Následující popisy skriptů pouze definují jejich hlavní účel, je ale třeba mít na paměti, že tyto skripty jsou navzájem propojeny a komunikují spolu. Toto platí nejen pro řídicí skripty, ale také pro samotné komponenty.

5.2.1 Řídicí skripty

V tomto projektu je použito mnoho řídicích skriptů, které jsou tvořeny statickou třídou, jejíž metody jsou volány z mnoha jiných skriptů, nebo třídou, která dědí z `MonoBehaviour` a místo statické třídy je tato třída jedináček (návrhový vzor singleton). Takový řídicí skript je nutno vložit jako komponent hernímu objektu ve scéně, aby se k němu dalo přistupovat. Tento postup je třeba proto, že některé funkce (například instancování herních objektů) nelze použít bez dědění z třídy `MonoBehaviour`.

Na obrázku 5.15 lze vidět tyto řídicí skripty:

- **AddressableAssetsLoader** - Využívá se pro řízení načítání, instancování, uvolňování a mazání tzv. Addressable Assets, což mohou být prefabry, textury a tak podobně, z jakéhokoliv skriptu.
- **ColorManager** - Slouží k získání příslušné barvy z předané vzácnosti předmětu.
- **InventoryManager** - Metody tohoto skriptu se volají, když je třeba přidat/odebrat předmět do, respektive z, inventáře. (Inventář má hráč, ale také například truhla.) Taktéž slouží k přidání nebo odebrání měny, vymazání předmětů z inventáře a převod měny a předmětů mezi inventáři.
- **InteractableManager** - Řídí aktivaci a deaktivaci indikace možnosti interagovat s předměty, tedy řídí aktivaci nebo deaktivaci skriptu, který zvýrazní obrys modelu daného předmětu. Také se pomocí tohoto skriptu iniciuje samotná interakce.
- **EquipmentModelLoader** - Používán k zobrazování modelů právě vybavených zbraní.
- **ItemInfoManager** - Stará se o aktivaci, změnu informací a deaktivaci informačního okna předmětů.
- **ModalWindowManager** - Nastavuje a aktivuje modální okno.
- **PauseManager** - Pozastavuje a poté znovu aktivuje hru. Také obsahuje indikaci stavu pozastavení hry a stavu různých oken uživatelského rozhraní, kterou používají ostatní skripty.
- **EquipmentManager** - Slouží k změně vybavení.
- **StatsManager** - Řídí doplňování bodů zdraví, doplňování many a zařizuje udělování poškození.
- **ItemCreator** - Ze skriptovatelných objektů vytváří instance příslušných předmětů.



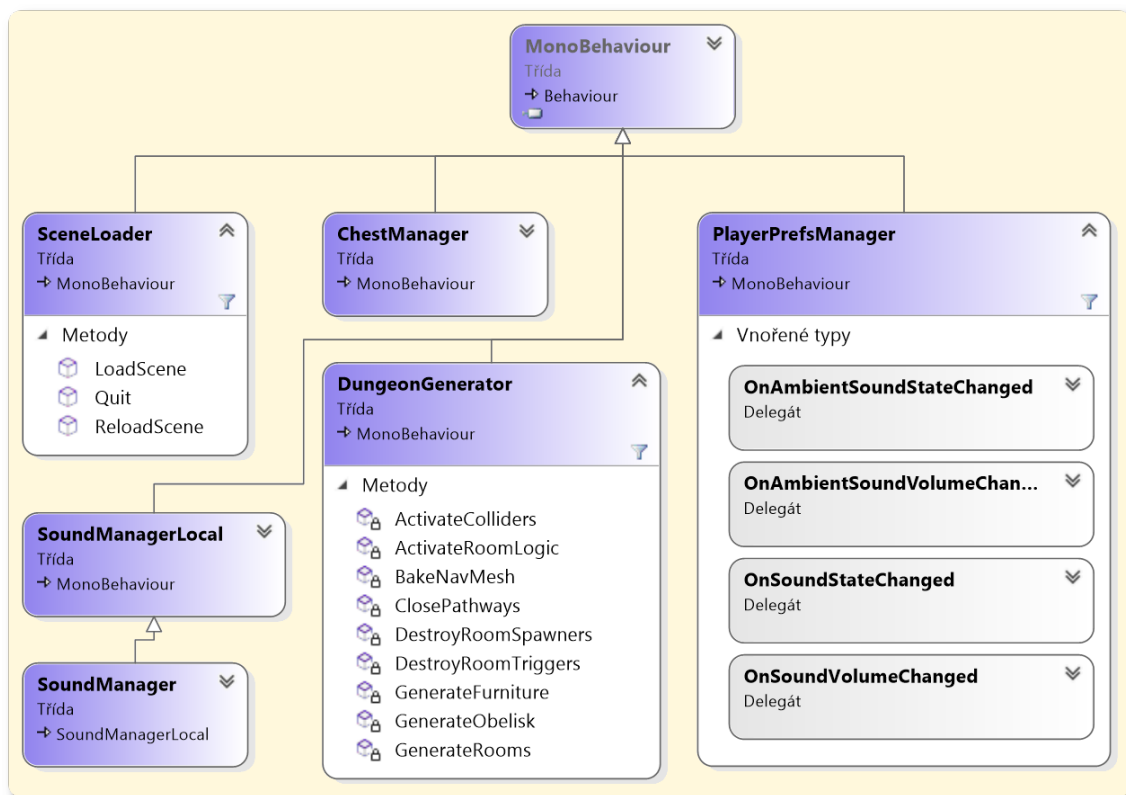
Obrázek 5.15: Řídící skripty - statické třídy

Řídící skripty druhého typu (jedináčci) lze vidět na obrázku 5.16 a jsou to:

- **SceneLoader** - Používán pro načítání scén a ukončení aplikace.
- **ChestManager** - Slouží k náhodnému generování obsahu truhel v dungeonu.
- **PlayerPrefsManager** - Pomocí třídy `PlayerPrefs` z balíku `UnityEngine` ukládá a načítá nastavení hlasitosti zvuků a nastavení toho, zda jsou zvuky zapnuty, nebo vypnuty.
- **DungeonGenerator** - Generuje rozložení místností dungeonu, vytváří navigační síť, vkládá dekorace, umísťuje obelisk a provádí další funkce spojené s generováním dungeonu.
- **SoundManager** - Pomocí tohoto skriptu se spouští a přerušují zvukové efekty a hudba.

Skript `SoundManagerLocal` jako jediný ze skriptů na obrázku 5.16 není jedináčkem, ale je vložen jako komponent na herní objekty, které vydávají zvuky a je třeba, aby byly zvuky

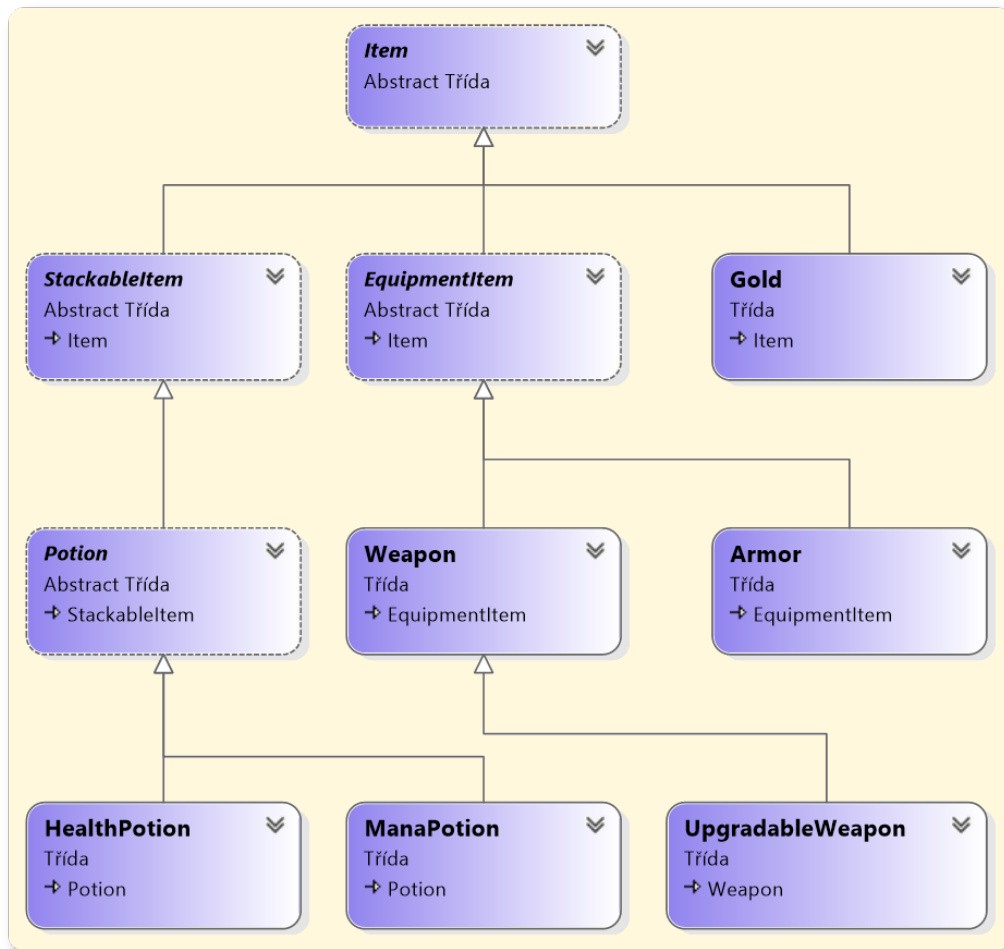
slýšet z pozice daného herního objektu. Je použit například u nepřátel, u kterých je důležité, aby jejich zvuky byly slyšet přímo od nich, na rozdíl od hudby a ambientních zvuků, které vychází z okolí.



Obrázek 5.16: Řídící skripty - nestatické (tedy komponenty)

5.2.2 Předměty

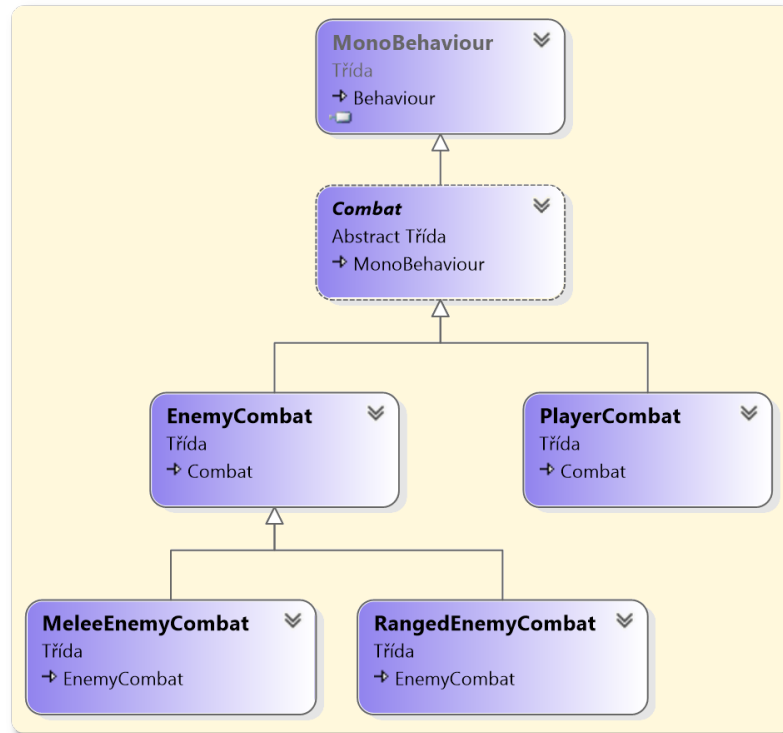
Pomocí skriptovatelných objektů v Unity (třída skriptu dědí z třídy `ScriptableObject`) jsou definovány vlastnosti předmětů. Skriptovatelné objekty je totiž možné ručně vytvořit v Unity editoru a poté upravit hodnoty proměnných. Pro každý předmět ve hře je vytvořen skriptovatelný objekt, u kterého jsou definovány krajní hodnoty jednotlivých bonusů (pokud nějaké poskytují), typ, jméno a ikona tohoto předmětu, model (jedná-li se o zbraň) a tak podobně. Z takto vytvořených a nadefinovaných skriptovatelných objektů se pomocí `ItemCreator` řídicího skriptu vytváří instance samotných předmětů, se kterými se potom dále pracuje (třídy lze vidět na obrázku 5.17). Z krajních hodnot bonusů vybírá řídicí skript výslednou hodnotu náhodně.



Obrázek 5.17: Třídy předmětů, jejichž instance vytváří řídicí skript `ItemCreator` ze skriptovatelných objektů.

5.2.3 Soubojové skripty

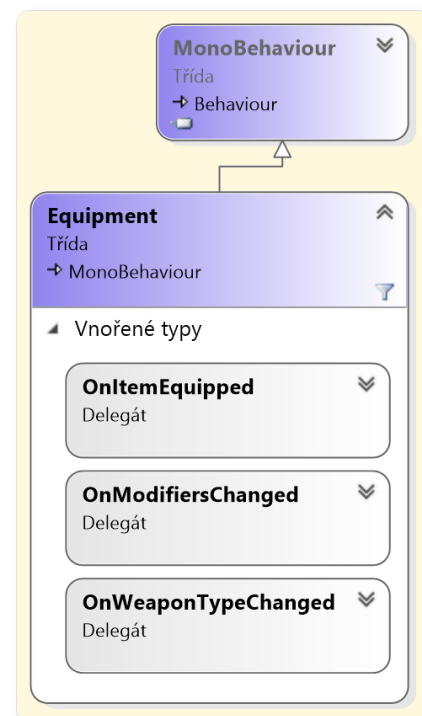
Pro schopnost souboje musí herní objekt obsahovat soubojový skript. Základním skriptem je skript `Combat`, ze kterého se odvíjí jiné skripty podle potřeb dané entity, která má být schopna boje. Konečnými skripty, které využívají jednotlivé herní objekty jako komponenty jsou `PlayerCombat`, `RangedEnemyCombat` a `MeleeEnemyCombat`. Jejich diagram tříd lze vidět na obrázku 5.18.



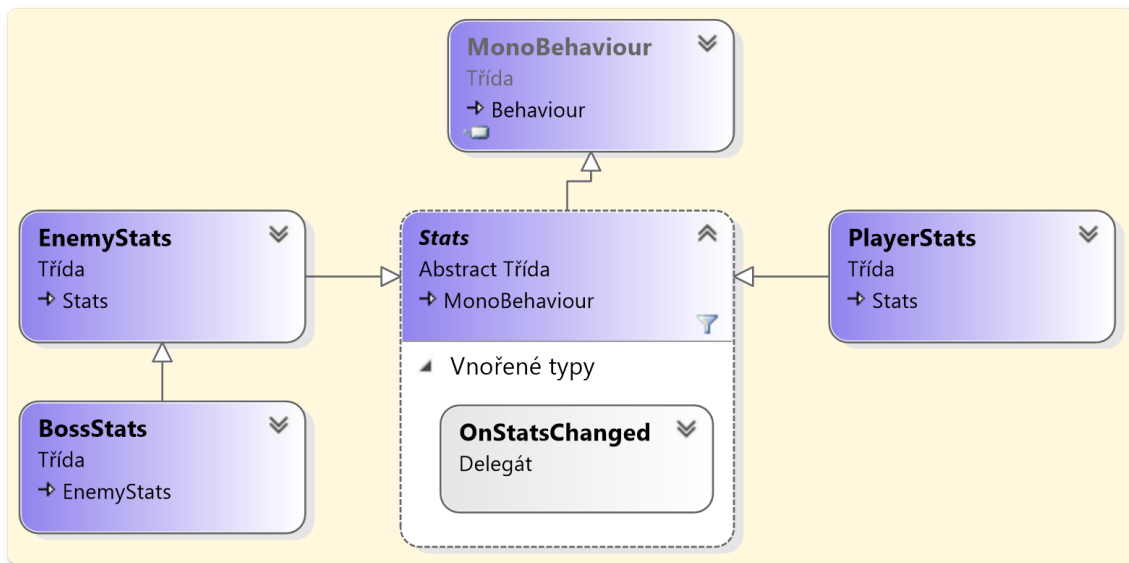
Obrázek 5.18: Skripty umožňující schopnost boje. Základním skriptem je skript `Combat`, ze kterého se odvíjí jiné skripty podle potřeb dané entity, která má být schopna boje.

5.2.4 Vybavení a statistiky

Možnost měnit vybavení a tím i statistiky své postavy má pouze hráč. K tomu slouží skript (komponent) `Equipment` z obrázku 5.19 a komponent `PlayerStats` z obrázku 5.20, které využívají řídicí skripty `EquipmentManager` a `StatsManager`. Statistika jako takové musí mít i nepřátele, pro které je používán komponent `EnemyStats`, potažmo `BossStats` pro bosse. U nepřátel jsou statistiky neměnné (kromě množství bodů zdraví), kdežto statistiky hráčovy postavy se mění v závislosti na vybavených předmětech. Množství bodů zdraví jak u hráčovy postavy, tak u nepřátel se odvíjí od úrovně hráče. Pouze boss se tímto pravidlem neřídí a má naprosto všechny statistiky pevně dány a to včetně bodů zdraví.



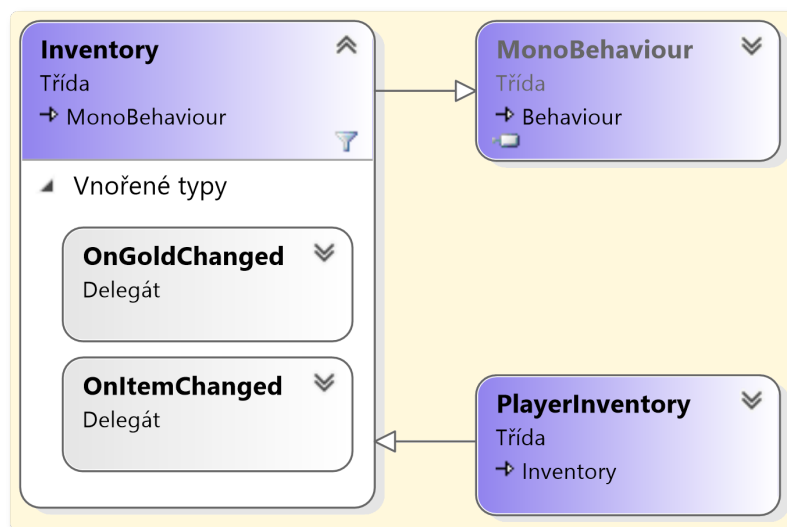
Obrázek 5.19: Skripty umožňující mít vybavení. Tuto možnost má ve hře pouze hráč.



Obrázek 5.20: Skripty pro uchovávání statistik a práci s nimi.

5.2.5 Inventář

Komponent pro inventář (`Inventory`) je třeba nejen u hráčovy postavy, ale také pro truhly. Přesněji je komponent `Inventory` využit u truhel a odvozený komponent `PlayerInventory` je použit pro inventář hráčovy postavy. Na diagram tříd lze nahlédnout na obrázku 5.21.

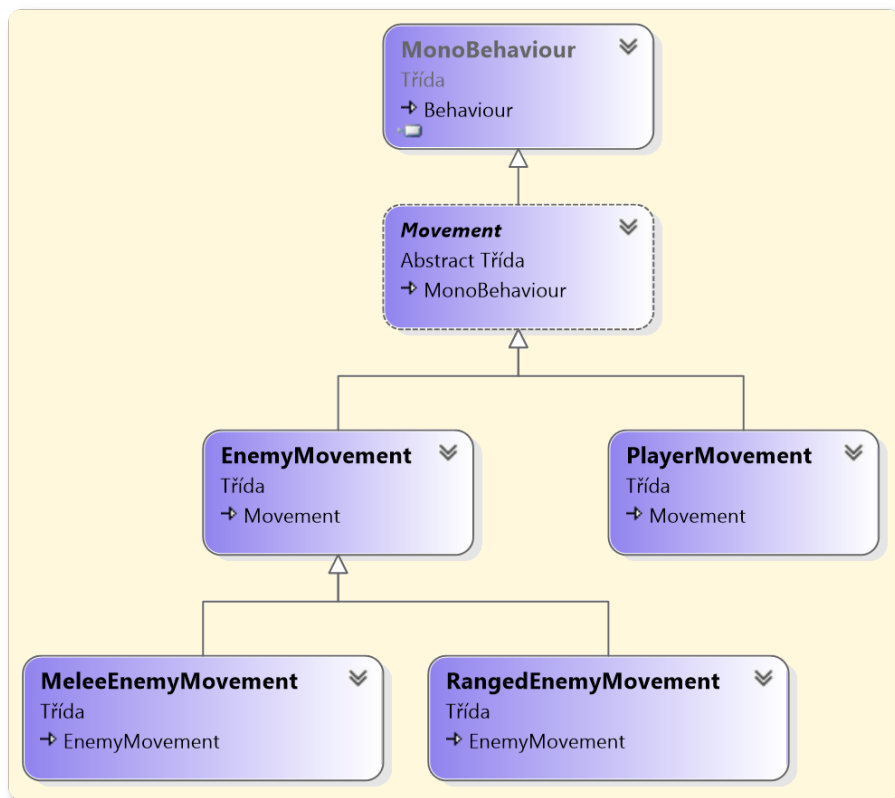


Obrázek 5.21: Skripty inventáře, které jsou potřeba pro hráčovu postavu, ale také truhly v dungeonu a hráčovu truhlu sloužící jako úložný prostor pro předměty.

5.2.6 Pohybové skripty

Základním skriptem je zde `Movement`, jak lze vidět na obrázku 5.22, ze kterého se odvíjí rozšířené komponenty pro pohyb hráče (`PlayerMovement`), jenž pracuje s Unity komponentem `CharacterController`, a komponenty `MeleeEnemyMovement` a `RangedEnemyMovement`,

kteře pracují s komponentem `NavMeshAgent` a řídí pohyb nepřátel. Důvodem rozdělení pohybového komponentu nepřátel do dvou různých komponentů je, že tyto skripty řídí kromě samotného pohybu i to, kdy nepřítel na hráčovu postavu zaútočí. A jelikož jsou pro útok na dálku a útok zblízka rozdílné podmínky, je i pohyb různých typů nepřátel rozdílný, proto je třeba mít dva komponenty pro pohyb nepřátel.



Obrázek 5.22: Skripty umožňující pohyb. U hráčovy postavy pracují s Unity komponentem `CharacterController` a u nepřátel pracují s Unity komponentem `NavMeshAgent`.

5.2.7 Ukládání a načítání

U toho typu her je potřeba řešit, jak uložit a následně načíst postup, předměty, úroveň a tak dále. Zde je to řešeno serializací statistik, vybavení a inventáře hráčovy postavy a také obsahu úložného prostoru hráčovy truhly. Serializovaná data jsou poté uložena do souboru, jehož cesta prochází přes `Application.persistentDataPath`, což je v Unity cesta přímo určená pro ukládání persistentních dat.

Pokud je při spouštění scény nastaveno, že se mají data načíst (nejedná se o novou hru), načtou se ze zmíněného souboru, deserializují se, vytvoří se z nich datový objekt v podobě komponentu a ten se vloží na herní objekt ve scéně. Z tohoto objektu si poté jiné skripty potřebná data převezmou a uloží.

Kapitola 6

Závěr

Cílem práce bylo vytvořit plně funkční počítačovou hru v herním enginu Unity. Důležitými a fixními body zadání bylo také to, aby hra spadala do žánru roguelike a hrála se z pohledu třetí osoby. Výsledek práce tyto stanovené cíle splňuje. Pro tvorbu byl použit programovací jazyk C#, pracující se skriptovacím API Unity. Kód byl napsán v nástroji Visual Studio. Samozřejmým nástrojem, který nelze opomenout, byl také Unity Editor, díky kterému je možno pracovat v samotném prostředí Unity.

Hru tvoří zajímavou právě prvek náhody, který je zaveden procedurálním generováním herní mapy a předmětů. Výrazným prvkem je také zvolené grafické zpracování, tedy low poly styl, který dodává hře nevšední vzhled.

Hra byla otestována a ač byl počet testovacích subjektů nevelký, lze ze zpětné vazby vyvodit některé závěry a skutečnosti. Například to, že je hra poměrně těžká (pro hráče, kteří nejsou s takovýmto typem hry obeznámeni, nebo nehrají počítačové hry často, až příliš těžká) a chvíli trvá, než si hráč zvykne, jakým způsobem se ve hře chovat, aby byl schopen dostat se dále. Na různé chyby je možno narazit častěji, s čímž se ale dá počítat, protože je hra stále ve velmi rané fázi vývoje.

Co se týče takzvaných assetů (zvuků, modelů a tak podobně), většina byla získána z Unity Asset Store¹. Animace, až na pár výjimek, byly převzaty z webové stránky Mixamo². Některé části kódu jsou převzaty, či převzaty a následně upraveny podle potřeby. Určité části realizace funkcí aplikace nebo části kódu jsou inspirovány jinými projekty a pracemi. Tyto skutečnosti jsou zdokumentovány ve zdrojových souborech aplikace. Práce s prostředím Unity v editoru byla příležitostně také inspirována jinými projekty.

Ač je projekt hotov, díky povaze této práce, tedy toho, že se jedná o počítačovou hru, bude vždy místo pro různá vylepšení a rozšíření. To se může týkat rozšíření obsahu a vylepšení hrátelnosti hry jako takové nebo například zlepšení optimalizace a běhu aplikace.

¹<https://assetstore.unity.com/>

²<https://www.mixamo.com/#/>

Literatura

- [1] *Co je to Dungeon?* [online]. [cit. 2022-01-10]. Dostupné z: <https://it-slovník.cz/pojem/dungeon>.
- [2] *Co jsou AAA (Triple-A) videohry?* [online]. [cit. 2022-01-24]. Dostupné z: <https://www.thefastcode.com/cs-czk/article/what-are-aaa-triple-a-video-games>.
- [3] ADAMS, E. a DORMANS, J. *Game mechanics: advanced game design*. New Riders Publishing, 2012. ISBN 9780321820273.
- [4] CAL. *Co je to Roguelike? Průvodce pro začátečníky* [online]. [cit. 2022-01-10]. Dostupné z: <https://soubory.info/info/co-je-to-roguelike-pruvodce-pro-zacatecniky/>.
- [5] GREGORY, J. *Game engine architecture*. CRC Press, 2018. ISBN 9781351974288.
- [6] KUMA. *Co je to herní engine* [online]. [cit. 2022-01-24]. Dostupné z: <https://www.ceskemody.cz/clanky.php?clanek=56>.
- [7] MAKAWIEL. *Produkce počítačové hry od A do Z* [online]. [cit. 2022-01-24]. Dostupné z: <https://tvorbaher.cz/produkce-pocitacove-hry-od-dil-i/>.
- [8] WIKIPEDIA. *Boss (počítačová hra)* [online]. [cit. 2022-01-25]. Dostupné z: [https://cs.wikipedia.org/wiki/Boss_\(po%C4%8D%C3%ADta%C4%8Dov%C3%A1_hra\)](https://cs.wikipedia.org/wiki/Boss_(po%C4%8D%C3%ADta%C4%8Dov%C3%A1_hra)).
- [9] WIKIPEDIA. *Herní engine* [online]. [cit. 2022-01-24]. Dostupné z: https://cs.wikipedia.org/wiki/Hern%C3%AD_engine.
- [10] WIKIPEDIA. *Mana (energie)* [online]. [cit. 2022-01-10]. Dostupné z: [https://cs.wikipedia.org/wiki/Mana_\(energie\)](https://cs.wikipedia.org/wiki/Mana_(energie)).
- [11] WIKIPEDIA. *Minecraft* [online]. [cit. 2022-01-24]. Dostupné z: <https://cs.wikipedia.org/wiki/Minecraft>.
- [12] WIKIPEDIA. *Nezávislá videohra* [online]. [cit. 2022-01-24]. Dostupné z: https://cs.wikipedia.org/wiki/Nez%C3%A1visl%C3%A1_videohra.
- [13] WIKIPEDIA. *NPC* [online]. [cit. 2022-01-25]. Dostupné z: <https://cs.wikipedia.org/wiki/NPC>.
- [14] WIKIPEDIA. *Shoot 'em up* [online]. [cit. 2022-01-25]. Dostupné z: https://en.wikipedia.org/wiki/Shoot_%27em_up#Bullet_hell.