Bakalářská práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Vývoj počítačové hry s použitím dat ze zařízení pro záznam pohybu

Ondřej Okluský
Otevřená informatika
oklusond@fel.cvut.cz

Květen 2015
Vedoucí práce: Ing. David Sedláček, Ph.D.

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Ondřej Okluský**

Studijní program: Otevřená informatika
Obor: Softwarové systémy

Název tématu: **Vývoj počítačové hry s využitím dat ze zařízení pro záznam pohybu**

Pokyny pro vypracování:

Navrhněte a implementujte počítačovou hru využívající animace získané zařízením pro záznam pohybu. Hra bude mít minimálně jednu plně hratelnou úroveň a pokročilé chování NPU (2 nebo více reakčních mechanismů). Zvažte možnost procedurálního generování hudby. Proveďte beta-testing hry s potřebným počtem hráčů.
V rámci teoretické části práce analyzujte možnosti využití zařízení pro záznam pohybu (MOCAP), které je umístěno v laboratoři virtuální reality (VRlab), pro využití při vytváření animací pro hry. Popište způsob získání a aplikace dat z MOCAPu na vámi vytvořené 3D modely postav v herním enginu Unity.

Součástí přílohy k BP (na CD nebo jako tištěná příloha) bude kompletní Design Document hry.

Seznam odborné literatury:

Jesse Schell. The Art of Game Design: A book of lenses. CRC Press. 2008
Unity Game Development Essentials, W. Goldstone, Packt Publishing, 2009
Creating Games with Unity and Maya, Adam Watkins, Focal Press, 2011

Vedoucí: Ing. David Sedláček, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

V Praze dne 3. 12. 2014

Poděkování / Prohlášení

Děkuji rodině, přátelům a přítelkyni za podporu při studiu. Dále dlužím velké poděkování Miloslavu Mejzrovi za roli actora při natáčení pomocí motion capture. Poděkování patří také Jakubovi Černému, za pomoc se základy umělé inteligence. Také bych rád poděkoval všem participantům, kteří se účastnili testování. A v neposlední řadě děkuji Ing. Davidovi Sedláčkovi, Ph.D. za pomoc a vedení práce.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 20. 5. 2015

.....

Abstrakt / Abstract

Tato bakalářská práce se zabývá vytvořením počítačové hry za použití 3D modelů, na které budou aplikovány animace nahrané z motion capture. Prvním cílem této práce je návrh a vytvoření počítačové hry v jednom z dostupných 3D herních enginů, hra bude mít jednu plně hratelnou úroveň a pokročilé chování NPU. Druhým cílem je nahrát pohyby pomocí zařízení motion capture umístěného v laboratoři virtuální reality a aplikovat je na vytvořené 3D modely herních postav. Třetím cílem je vytvořit jednoduchý engine pro procedurální generování hudby.

Klíčová slova: Motion capture, počítačová animace, vývoj počítačových her, umělá inteligence.

This bachelors thesis is about to develop computer game using animations obtained from motion capture. First goal is to create concept and then create computer game using one of available 3D game engines, game must have one complete level and advanced behaviour of NPC. Second goal is to record animations using motion capture in VRlab and apply them to own 3D models. Third goal is to create simple sound engine able to generate music.

Keywords: Motion capture, animation, game development, artificial inteligence.

Obsah

1 Úvod 1
2 Rozbor zadání 2
3 Rešerše 3
3.1 Herní engine 3
3.2 Modelování 5
3.3 Počítačová animace 6
3.4 Motion capture 7
3.5 Akční realtimová strategie 9
3.6 Umělá inteligence 10
4 Návrh řešení 14
4.1 Volba nástrojů 14
4.2 Design hry 14
4.3 Návrh tříd 19
4.4 Umělá inteligence ve hře 20
4.5 Procedurální generování hudby 21
5 Průběh práce 23
5.1 Motion capture 23
5.2 Zpracování animací a přenos na 3D modely 26
5.3 Tvorba scény v Unity 28
6 Závěr 31
6.1 Beta testing 31
6.2 Zhodnocení splnění cílů 32
6.3 Diskuse dalšího postupu práce 32
Literatura 34
A Obsah přiloženého DVD 35
B 3D modely 36
C Tabulky 39
D Obrázky ze hry 46

Tabulky / Obrázky

4.1. Tabulka hodnot hlavní jednotky ... 16
4.2. Tabulka hodnot předmětů ... 16
4.3. Tabulka hodnot vylepšení ... 17
4.4. Tabulka hodnot mířeného ohnivého kouzla ... 18
B.1. Seznam obrázků 3D modelů ... 36

3.1. Graf scény ... 5
3.2. Mapa ARTS ... 9
3.3. Stavový automat ... 10
3.4. Rozhodovací strom ... 11
3.5. Behaviorální strom ... 12
3.6. Neuronové sítě ... 13
4.1. Grafický návrh GUI ... 18
4.2. Diagram tříd ... 19
4.3. Vztahy datových tříd k jednotce ... 19
5.1. Optitrack kalibrace ... 24
5.2. Rozmístění markerů ... 25
5.3. Zpracování ve formě signálu ... 25
5.4. Špatný skinning, dobrý skinning, kreslení vah ... 27
6.1. Obrázky ze hry ... 33
A.1. Obsah přiloženého DVD ... 35
B.2. Medvěd ... 36
B.3. Bonus ... 36
B.4. Drak ... 36
B.5. Plot ... 36
B.6. Vlajka ... 37
B.7. Hlavní hrdina ... 37
B.8. Dům ... 37
B.9. Velký dům ... 37
B.10. Malá jednotka ... 37
B.11. Houba ... 37
B.12. Jehličnatý strom ... 37
B.13. Kapradí ... 37
B.14. Terén ... 37
B.15. Věž ... 37
B.16. Listnatý strom ... 38
D.17. Obrázky ze hry ... 46

Kapitola 1
Úvod

Počítačové hry jsou v dnešní době významným artiklem zábavního průmyslu. Hraní počítačových her je zábavné, rozvíjí myšlení, zlepšuje koordianci a motoriku. Jejich vývoj zaměstnává specialisty z různých velmi odlišných odvětví lidské činnosti, tvůrce herních příběhů, programátory herní logiky a umělé inteligence, fyziky, počítačové grafiky, architekty herních enginů, umělce, kteří dávají počítačovým hrám tak krásné oslnivé pozlátko, hudebníky a zvukaře, kteří tvoří jedinečnou herní atmosféru a v neposlední řadě manažery a marketingové specialisty, snažící se získat co největší počet spokojených zákazníků.

Cílem této práce bylo vytvoření série modelů, na které budou aplikována data z motion capture. Tyto modely byly následně exportovány do herního enginu a s jejich pomocí byla vytvořena počítačová hra. Výsledný produkt byl zkompilován ve 32 bitové verzi do spustitelného souboru pro operační systém Windows. Pro co možná nejlepší výsledek byla hra otestována v herní komunitě.

Výsledky této práce budou zveřejněny na některém z volně dostupných datových serverů, stejně tak jako zdrojový kód, který bude možné využít pro inspiraci začínajících vývojářů počítačových her. Přínosem pro mou osobu budou cenné znalosti, obtížně dostupné zkušenosti a rozšíření mého portfolia.

Kapitola 2
Rozbor zadání

Hlavním cílem této práce je zaznamenat pohyby pomocí zařízení motion capture umístěného v laboratoři virtuální reality a s jeho pomocí vytvořit počítačovou hru. Přesto, že zadání vypadá na první pohled velmi komplexně, je možné ho rozdělit do poměrně přímočarých částí. Je nutné vybrat volně dostupný herní engine, ve kterém bude hra implementována a zjistit omezení enginu, která musí být dodržena, aby byl zvolen správný pracovní postup. Herní žánr vytyčí jaké modely, animace a herní logika budou potřeba k sestrojení hry. Dalším krokem je vybrat nástroje, které umožní práci s multimediálním obsahem. Bude potřeba vytvořit herní modely, na některé z nich budou aplikovány data z motion capture a zbylé budou sloužit k vytvoření herního prostředí. Poté je nutné získat data pomocí motion capture a zhodnotit jeho použitelnost při vytváření hry. Tato data se musí dále upravit a namapovat na připravené herní modely tak, aby vznikly animace použitelné ve hře. Nakonec bude možné tyto jednotlivé části spojit ve vybraném herním enginu k vytvoření počítačové hry. Závěrem projektu bude testování produktu s lidmi z hráčské komunity, aby se zjistilo, zda je hra funkční.

Další požadavek plynoucí ze zadání je návrh a implementace procedurálního generování hudby, což je způsob tvorby hudby za pomocí algoritmů.

Kapitola 3
Rešerše

V této kapitole se čtenář obecně seznámí s principy, které se využívají při tvorbě počítačových her a které byly použité k tvorbě této bakalářské práce. Nejprve se seznámí s některými herními enginy, které byly pro tuto práci zamýšlené. Poté se dozví některé pokročilé techniky tvorby virtuálních modelů a nakonec i jak funguje počítačová animace a co je motion capture. Bude popsán princip vybraného herního žánru, a jaké existují možnosti pro reprezentaci umělé inteligence počítačem řízených jednotek.

3.1 Herní engine

Herní engine je nástroj sloužící v dnešní době více než jen k vytváření počítačových her ale i filmů, předváděcích míst a virtuální reality. Herní engine je soubor funkcí umožňující práci s 2D a 3D grafikou, scénou a grafem scény, umělou inteligencí, vykreslováním, fyzikou, sítí, skripty, audiem, videem a dalšími. Vývoj vlastního herního enginu je náročná, dlouhotrvající záležitost, soubor algoritmů dlouhých miliony řádků kódu. A proto je výhodné využívat na trhu dostupné herní enginy. Výhodou současných herních enginů je možnost multiplatformního exportu, jsou dobře dokumentované, mají rozšiřitelnou funkčnost a uživatelské rozhraní. Jsou snadno použitelné, přehledné a mají širokou uživatelskou základnu ochotnou poradit. Nyní se blíže seznámíme s herními enginy, které byly zamýšlené pro využití v této práci. Hlavními kritérii pro výběr enginu jsou: jednoduché použití, dobrá dokumentace, animace kůže (skinned mesh animation)[1]), schopnost importovat běžně používané formáty 3D modelů a možnost volného užití.

Unreal Engine 4[2]) je velmi rozsáhlý volně dostupný herní engine. Při využití pouze pro studijní účely je zcela zdarma, při vydání hry pro komerční účely si společnost Epic Games nárokuje honorář 5 % z vydělané částky. Jedná se o 2D a 3D herní engine s bohatou historií v počítačových hrách. Editor herního enginu je přehledný a poskytuje širokou škálu prefabrikátů, které začátečníkům zjednodušují vytvoření funkční verze a profesionálům ušetří mnoho času s psaním kódu, který je v podstatě totožný. Unreal engine využívá UScript nadstavbu nad C++ jako skriptovací jazyk, je dobře zdokumentovaný a má širokou uživatelskou základnu.

Unity[3]) je rozsáhlý volně dostupný 2D a 3D herní engine. Unity je k dostání ve dvou provedeních Free a Pro, aktuální ceny Pro verze je možné zjistit na webových stránkách[4]). V Unity 4 byl mezi verzemi Pro a Free malý rozdíl v omezeních funkčnosti, která

[1]) `http://graphics.cs.cmu.edu/projects/sma/jamesTwigg_SMA.pdf`
[2]) `https://www.unrealengine.com/`
[3]) `https://www.unity3d.com/`
[4]) `https://store.unity3d.com/`

však ve tvorbě hry nebyla překážkou, spíše pozlátkem navíc. Unity 5, které vyšlo v průběhu této práce, již nemá rozdíl ve funkčnosti enginu, ale v Pro verzi nabízí nástroje navíc, využití cloudu a profesionálních služeb. Více informací je možné nalézt na webových stránkách Unity store[1]). Editor herního enginu je přehledný, nabízí malé množství prefabrikátů ulehčující začátečníkům rychlé vytvoření funkční verze. Unity engine využívá C# jako skriptovací jazyk, je dobře zdokumentovaný a má širokou uživatelskou základnu.

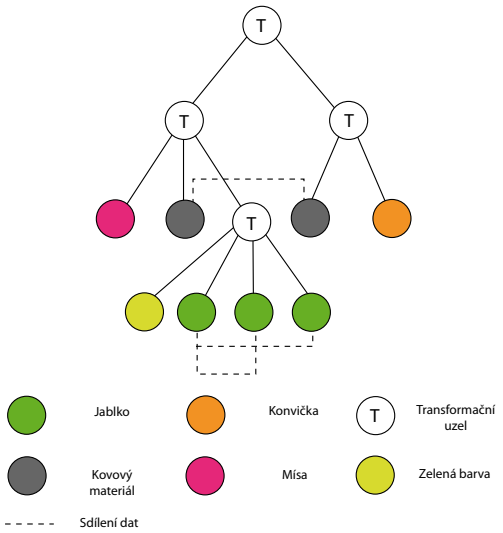
Paradox Engine[2]) je malý volně dostupný 2D a 3D herní engine, který je právě ve stádiu vývoje. Editor herního enginu není příliš přehledný a intuitivní, vykreslování není plynulé. Paradox engine využívá C# jako skriptovací jazyk, dokumentace stále ještě není kompletní a uživatelská základna není zatím velká.

Všechny vybrané herní enginy podporují animaci kůže, zajímavá je i možnost integrace s Microsoft Visual studiem [3]), což je veliký přínos i pro malé projekty. Paradox engine jsem nepoužil z důvodu jeho nekompletnosti. Vzhledem k předchozím zkušenostem a dobrým referencím jsem se pro tvorbu této práce rozhodl používat herní engine Unity.

Herní smyčka je nedílnou součástí herních enginů. Na rozdíl od ostatních programů, ve kterých program čeká na uživatelský vstup, jenž může vyvolat akci trvající až několik vteřin a dlouhé zpracování, herní smyčka neustále zpracovává data i bez vstupu od uživatele. Herní smyčka je rozdělena do dvou a více akcí, které se starají o obsluhu uživatelského vstupu, výpočet kolizí a fyziky, renderování grafiky, přehrávání audia, zpracování herní logiky a umělé inteligence, komunikace po síti a nahrávání dat. První akce se provede pouze poprvé při spuštění hry a slouží k inicializaci herních systémů. Další akce, zpracovávající chod herního enginu, se opakují v různě dlouhých časových úsecích. Se zvyšujícím se počtem zpracovávaných dat a prováděných operací se počet snímků, které je možné vykreslit za sekundu snižuje, což může vyústit v narušení plynulosti, které snižuje kvalitu herního zážitku a nakonec i hratelnosti.

Volně přejato z knihy Moderní počítačová grafika [1]. **Graf scény** je dynamická datová struktura používaná v počítačové grafice, která uchovává informace o 3D světě. Herní enginy mohou využívat různé principy k reprezentaci grafu scény. Uveďme si jako příklad reprezentaci pomocí stromu, dále již jen graf scény. Graf scény viz. obrázek 3.1 je n-nární strom, jeho uzly mohou obsahovat různé vlastnosti. Jedna ze základních je afinní transformace, která udává polohu, rotaci a velikost v lokálním souřadném systému, proto tyto uzly můžeme nazývat transformační. Jeho listy reprezentují geometrické objekty ve světě, materiály a další náležitosti, které se různí danou implementací. Uzly grafu scény mohou mít mezi sebou vazbu dědičnost (předek - potomek), která umožňuje jednoduše reprezentovat pozice objektů v hierarchickém uspořádání. Každý uzel může mít žádného, či více potomků, každý potomek má pouze jednoho předka, výjimkou je kořen stromu, který předka nemá.

[1]) `https://unity3d.com/get-unity`
[2]) `http://paradox3d.net/`
[3]) `https://www.visualstudio.com/products/visual-studio-community-vs`



Obrázek 3.1. zobrazující hierarchii grafu scény.

Graf scény je procházen při vykreslování virtuálního světa během každého snímku. Při vykreslování se graf scény prochází v pořadí pre-order, čili z kořene do listů, zleva doprava. Při průchodu každým uzlem se složí afinní transformace obsažená v uzlu s afinní transformací předka. Výsledná transforamce je aplikována na list stromu, tím dostaneme polohu geometrického objektu ve scéně. Při zpracování grafu scény můžeme narazit například na materiály a textury, ty se neskládají a je aplikována pouze poslední nalezená vlastnost. Představme si například situaci, kde máme jednoduchou scénu, ve které je mísa a na ní jablko, jablko je potomek mísy a mísa je předek jablka. Když hýbeme mísou, hýbeme i jablkem, když hýbeme jablkem, mísa zůstává na svém místě. Netriviální grafy scény využívají instancování. Všechny listy které obsahují stejná grafická data odkazují na místo, kde jsou tato data uložena, zjednodušeně řečeno jedna grafická data se stávají vzorem pro ostatní stejná data ve scéně. Tento přístup šetří paměť a zvyšuje rychlost. V případě použití instancování se jedná o takzvaný zhroucený strom.

3.2 Modelování

Nyní se čtenář dozví obecně o některých pokročilých technikách 3D modelování, které byly využity při tvorbě této práce. O základech 3D modelování je možné se dozvědět z knihy Moderní počítačová grafika[2].

Modelování pomocí dělených ploch je způsob umožňující řízení detailů nad povrchovou reprezentací pomocí plošek. Pro modelování pomocí dělených ploch se doporučuje použít co nejjednodušší geometrii vystihující podstatu modelovaného objektu, který je tvořen výhradně quady, ploškami tvořenými čtyřmi hranami a čtyřmi vrcholy. Po aplikování přerozdělení se vytvoří nová úroveň geometrie, která je tvořena hustší sítí polygonů. Pokud byl použit model tvořený quady, plošky se přerozdělí takovým způsobem, kdy z každého trojúhelníku se vytvoří tři quady, z každého quadu se vytvoří čtyři quady a tak dále nad každým dalším polygonem vyššího stupně.

Tento proces zvětšil počet plošek, kterými můžeme vyjádřit požadovaný detail, a také zjemnil přechody mezi sousedními polygony. Pokud není po prvním přerozdělení počet plošek dostačující k vyjádření požadovaného detailu, můžeme přerozdělovat vybrané plošky libovolně-krát, dokud nezískáme požadovanou úroveň detailu. Při modelování je možné vrátit se k libovolné úrovni přerozdělení a tuto úroveň dále upravovat. Provedené změny se následně propagují do všech úrovní geometrie. Vzhledem k vysokému počtu plošek, který je nutný k vyjádření vysokého detailu, jsou tyto modely nevhodné pro využití v počítačových hrách. Pro jejich využití je dobré vypéct normálovou mapu[3] z nejvyšší úrovně a aplikovat ji na nejnižší úroveň geometrie. Použití normálových map je v počítačových hrách běžná praxe, umožňující zobrazovat detailní modely s nízkým počtem plošek.

NURBS[4] jsou v 3D počítačové grafice mocným a využívaným nástrojem ať již při modelování nebo animaci. Jedná se o aproximační křivky s velmi vysokou variabilitou, určené složitými matematickými vztahy. Pro výpočet NURBS křivky je nutné znát posloupnost řídících bodů, které tuto křivku přesně definují. Jelikož se jedná o aproximační křivku, tato křivka může, avšak nutně nemusí procházet těmito body. Pomocí NURBS se dají vytvářet také povrchy, které jsou vhodné pro vytváření modelů v produkční kvalitě. Jelikož NURBS povrchy jsou reprezentovány matematickými vztahy na místo určitého počtu plošek, proto ani při velkém přiblížení nejsou vidět zlomy, které jsou znatelné na hranicích navazujících plošek. Jak již bylo řečeno, hlavní výhodou NURBS je jejich vysoká variabilita, kvalita obrazu, jednoduchost použití a převoditelnost do povrchové reprezentace pomocí plošek. Pro využití v herních enginech jsou NURBS povrchy zbytečně náročné, pokud je model vytvořený pomocí NURBS, je vhodné ho převést na reprezentaci pomocí plošek.

Vytvořené modely je pro použití v herních enginech někdy nutné exportovat do grafických datových formátů odlišných od proprietárních formátů modelářů. Po importování do herního enginu je již možné s 3D modelem pracovat, přidávat jeho instance do scény, vytvářet různé hierarchie v grafu scény, měnit jeho pozici, zvětšení a rotaci. Herní enginy umožňují pracovat i s materiály modelu, přiřazovat jim různé textury a měnit hodnoty daného materiálu.

3.3 Počítačová animace

Počítačová animace je postup, při kterém se snažíme z jednotlivých snímků vytvořit viditelný pohyb. Můžeme rozlišovat mezi 2D a 3D animací, v případě 2D animace vytváříme iluzi pohybu pomocí rychle se měnících snímků, které se od sebe liší malou změnou tvaru nebo posunem. Pokud jsme v průběhu animace schopni zobrazit dostatek snímků za sekundu, pohyb se díky vlastnostem našeho vizuálního systému zdá plynulým.

Při vytváření počítačové animace využíváme takzvaného klíčování, což znamená, že v určitém čase odpovídajícím některému snímku uložíme, „zaklíčujeme“ vlastnosti geometrického objektu jako je například jeho afinní transformace, barva či viditelnost. Mezi těmito zaklíčovanými snímky poté interpolujeme a tak získáváme hodnoty pro výsledný pohyb. Tento postup je velmi výhodný, pokud není zaklíčován každý přehrávaný snímek, pak se jeho použitím snižuje množství uložených dat. Dále můžeme rozlišovat mezi dopřednou kinematikou a inverzní kinematikou. Dopředná kinematika funguje tak, že

například u robotického ramene, musíme každému kloubu ramene nastavit určitou rotaci, tak aby rameno dosáhlo na určené místo. Úlohou inverzní kinematiky je dopočítat výsledné rotace kloubů ramene podle pozice efektoru umístěného na konci ramene.

Vytvořené animace je možné společně s 3D modelem importovat do herního enginu. V herním enginu můžeme s animacemi dále pracovat, můžeme určit, jak ostré budou přechody mezi dvěma animacemi, upravovat jejich rychlost, počet opakování, začátek a konec animace a tak dále. Animace mají velkou informativní hodnotu a také pomáhají graficky oživit hru. Namísto abychom uživatele zatěžovali výpisem do konzole, který ho informuje o sebrání předmětu ze země, přehrajeme animaci, jak se postava pro předmět ohýbá k zemi. Tím se uživatel dozví o provedené akci.

3.4 Motion capture

Motion capture, česky záznam pohybu, je technologie, jejímž cílem je zaznamenávat pohyby humanoida, pevného tělesa či zvířete. Zaznamenávaný objekt je označováno jako actor. Motion capture se v posledních letech hojně využívá v mnoha odvětvích, jako například k výrobě animací pro počítačové hry a filmy, v lékařství, kde jsou lékaři pomocí rozboru pohybu schopni určit některé vrozené vady pacienta. Své využití nalézá také ve sportu, kde je pohyb klíčovou součástí, trenéři mohou analýzou pohybu sportovce vylepšit jeho výkony. Významné uplatnění nalézá ve vojenství, kde se využívá k nejrůznějším simulacím. Motion capture je možné používat nejen k záznamu dat, ale také k jejich okamžitému náhledu. Ten usnadňuje práci režisérů a kameramanů na animovaných filmech.

Zásadní výhodou motion capture je poskytování reálných pohybů postav, které je velmi obtížné vyrobit ručně a vyžadují značně zkušeného animátora. Nevýhodou je jeho vysoká pořizovací cena a poměrně náročné zpracování dat, které však není v porovnání s ruční animací takovou překážkou. Během nahrávání jsou snímána data z actora s vysokou frekvencí, tato data jsou dále zpracovávána ve formě signálu. Zpracování dat z motion capture ve formě signálu, bude vystěleno později v kapitole 5.1. V posledních dvaceti letech se objevilo mnoho různých technologií pro záznam pohybu. Ve většině případů je actor oblečen do speciálního obleku, který umožňuje snímání pohybových dat. Přestože máme na vybranou z poměrně různorodých technických provedení, ujala se ve větší míře pouze ta optická. Ostatní technologie budou zmíněny pouze okrajově. Odlišné způsoby záznamu pohybu mají své výhody a nevýhody, někdy jsou omezeny prostorem, v jiných případech světelnými podmínkami. Další použitelné technologie motion capture jsou gyroskopické systémy, mechanické systémy, ultrazvukové systémy a další.

Motion capture také slouží k potřebám virtuální reality. Pomocí tohoto zařízení je možné snímat například pohyby rukou, na které je schopen program virtuální reality reagovat. Některá zařízení pro virtuální realitu dokonce umožňují na základě dat z motion capture poskytnout uživateli zpětnou vazbu, vzbuzující hmatové nebo tepelné stimuly. Nyní budou popsány obecné principy optického motion capture a zpracování získaných dat.

Optické systémy s markery se skládají ze tří částí: z kamer, ze speciálních světel a markerů. Kamery jsou rozestaveny v různých úhlech okolo scény tak, aby v co největším

prostoru zabíraly celou postavu actora. Po rozmístění kamer je nutné provést kalibraci. Speciální infračervená, nebo červená světla jsou umístěna okolo objektivu tak, aby bylo zajištěno, že směřují ve směru kamery. Poslední nedílnou součástí sestavy jsou markery, malé kuličky nebo půlkuličky potažené reflexním materiálem, aby co nejlépe odráželi světlo zpět do kamer. Markery se umísťují na elastický oblek ze suchého zipu, nebo na pásky ze suchého zipu rozmístěné po actorově těle. Rozmístění markerů je velmi variabilní a tudíž umožňuje snímání lidí, zvířat i předmětů. Pro snímání mimiky obličeje se používají malé reflexní nálepky nebo speciální barva.

Kalibrace kamer se provádí pomocí speciální soupravy, tím se určí vzájemná poloha a natočení kamer. Poté se definuje počátek souřadné soustavy souřadnic, ve kterém budou kamery natáčet. Pro výpočet pozice markeru v prostoru je nutné, aby alespoň dvě kamery zároveň snímaly stejný marker. Pokud marker není v nějakém okamžiku snímán alespoň dvěma kamerami, tento marker se ztrácí ze záznamu a po návratu do scény vzniká nekonzistence v natočených datech, jelikož systém neví zda marker, který opustil scénu, se vrátil, nebo se jedná o zcela jiný marker. Pro vyjádření rotace specifické části těla nebo předmětu je zapotřebí více než jeden marker.

Optické systémy bez markerů využívají dva různé principy ke zjištění polohy a rotace částí lidského těla. Jak již název napovídá, tento způsob nepotřebuje speciální markery pro určení pohybu actora. První princip, spojený s herními konzolemi jako je Microsoft Kinect[1]), využívá malého počtu zkalibrovaných kamer ke čtení světelného vzorce vysílaného na tělo actora. Skrz zaznamenaný obraz je proložena pomocná obecná kostra a zkoumáním deformací světelného vzorce na těle actora se vypočítá translace a rotace kostí. Druhý princip využívá jednu, nebo více zkalibrovaných kamer, a v hojné míře také rozpoznávání obrazu a rekonstrukci z fotografie. Nejvýznamnějším zástupcem tohoto principu je firma iPi Soft[2]).

Pořizování kvalitních dat z motion capture je náročný proces. Některé nahrávky se musejí kvůli neúplnosti zaznamenaných dat nebo nepřesnosti nahrát i vícekrát. To, která nahrávka se výsledně použije, je na úvaze obsluhy motion capture. Data se po natáčení zpracovávají - čistí, odstraňuje se z nich šum a záplatují se mezery vzniklé například opuštěním markeru ze scény. To vše je možné dělat již v proprietárním softwaru daného motion capture a nebo v programu schopném tato data zpracovávat. Po vyčištění se data exportují do jednoho z existujících formátů pro přenos animačních dat.

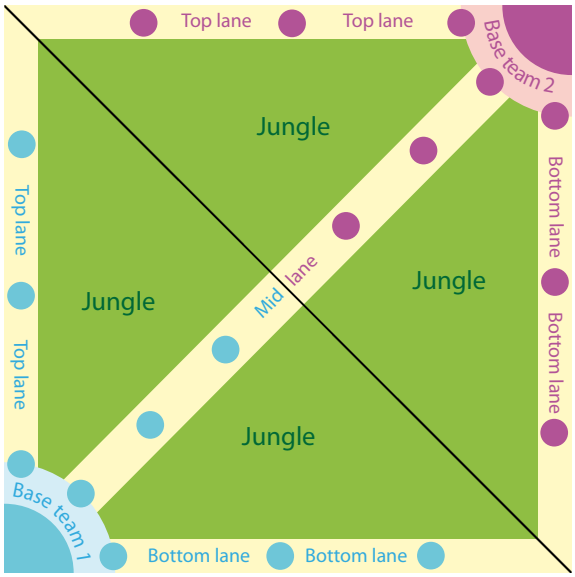
Retargeting je postup, při kterém se snažíme namapovat pohyby actora nebo animované postavy na jinou virtuální postavu. Při použití motion capture se nejčastěji řeší přenos pohybu actora na virtuální postavu, jehož cílem je získat pohyby v reálném prostředí a přenést je do toho virtuálního. Pokud se snažíme při retargetingu namapovat pohyby mezi nestejně velkými postavami, nebo dokonce postavami nepříslušícími stejnému živočišnému kmenu[3]), mohou vzniknout problémy. Například pokud by byl actor větší než 3D model postavy, pak je zřejmé, že ani jejich končetiny nebudou stejně dlouhé. Požadujeme-li, aby 3D model postavy dosáhl na místo kam dosáhl actor, pak v některých případech to nebude vypadat reálně, nebo to nebude vůbec možné. Proto mu-

[1]) https://www.microsoft.com/en-us/kinectforwindows/
[2]) http://ipisoft.com/
[3]) http://cs.wikipedia.org/wiki/Biologick%C3%A1_systematika

síme data následně upravit v 3D animačním nástroji nebo modeláři, který umí pracovat s animacemi.

3.5 Akční realtimová strategie

V této kapitole se čtenář seznámí s herním žánrem vybraným pro tuto práci, jeho vznikem a obecně používanými principy. Akční realtimová strategie[5] je druh herního žánru vycházející z realtimové strategie, ve kterém jsou hráči rozděleni do dvou soupeřících týmů. Hráč ovládá jednu hlavní postavu, která se v průběhu hry vyvíjí, zvyšují se její dovednosti, odolnosti, přibývají kouzla a předměty. Hlavním cílem tohoto herního žánru je zničit hlavní budovu nepřátelského týmu. Ke splnění hlavního cíle pomáhají hráči periodicky se objevující počítačem řízené malé jednotky. Hra se odehrává na jedné či více takzvaných linkách, cestách vedoucích od spřátelené hlavní budovy k nepřátelské hlavní budově. Na těchto linkách jsou rozmístěny obranné budovy a procházející malé jednotky.



Obrázek 3.2. Typická mapa pro akční realtimovou strategii.

Typická mapa pro herní žánr akční realtimové strategie vypadá jako na obrázku 3.2. Přesto, že hlavním cílem je zničit hlavní budovu nepřátelského týmu, ničení ostatních nepřátelských budov a jednotek přináší další benefity, jako jsou například peníze, předměty, zkušenosti, a další bonusy řízené konkrétní hrou. Během hry se může stát, že hráčova hlavní postava bude zabita. V jiných herních žánrech jako například Role Playing Game, česky Hra na hrdiny, by tento problém vedl k ukončení hry. V případě akční realtimové strategie hra nekončí, avšak hlavní postava hráče je penalizována ztrátou herního času, který stráví jako mrtvá, nebo s dočasným postižením. Postupem času se hlavní postavy stávají silnější a využívají různé strategie k získání převahy nad nepřítelem. Snaží se vydobýt si výhodu zabezpečováním svých budov, zabíjením nepřátelských hlavních postav a malých jednotek, tím získávají zdroje potřebné k ukončení hry.

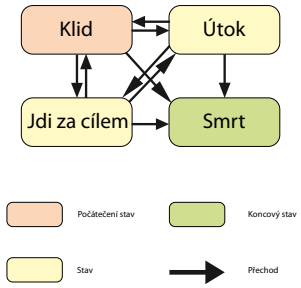
Výsledná podoba hry a herního systému se nemusí vždy stoprocentně shodovat se specifikací herního žánru akční realtimové strategie, avšak ve větší míře mají hry z toho

žánru velmi podobné rysy. První vážný zástupce tohoto žánru je Warcraft III - Defense of the Ancients zkráceně DotA.

3.6 Umělá inteligence

Snahou umělé inteligence je, aby mohl počítač přemýšlet nezávisle na lidech a v nejlepším případě na lidský vstup adekvátně reagovat. Při hraní počítačových her uživatel očekává stále složitější a složitější chování nepřátel blížící se až reálnému člověku na místo jednoduché mimiky. V oblasti umělé inteligence je nutné stálé zvyšování úrovně složitosti, neboť v oblasti počítačových her mají vysokou konkurenci a to skutečné hráče. Pevně věřím, že se této úrovně v budoucnu dočkáme. Programátoři umělé inteligence mají velmi náročný úkol, naprogramovat chování postav tak, aby bylo nepředvídatelné, zábavné, uvěřitelné z hlediska reálnosti prováděných akcí, ale aby přílišně nezatěžovalo chod CPU. Umělá inteligence postav nemusí být nikterak složitá, jedná se o hráčský zážitek. Pokud hráč nepozná rozdíl mezi složitým chováním a pouhou iluzí složitého chování, je účelné použít jednoduší a úspornější verzi.

Zprvopočátku byla umělá inteligence prováděna pevně zakódovanými povely, které postrádaly veškerou dynamiku rozhodování. Později se do tohoto procesu začaly začleňovat události řízené náhodou - změna počátečních a koncových časů, pravděpodobnost a další nepředvídatelné události využívající například generátoru náhodných čísel. Přidáním dynamiky do rozhodovacího procesu postav se podařilo hráče přesvědčit, že chování postav a prostředí není řízeno programem. Některé ze složitějších způsobů reprezentace jsou uvedeny v následujících podkapitolách. Podklady pro toto téma byly čerpány z knihy Game coding complete [6].



Obrázek 3.3. Ilustrace příkladu stavového automatu.

Konečný stavový automat je matematická konstrukce, která se může nalézat v jednom z konečně mnoha stavů. Mezi jednotlivými stavy existují vazby, přechody, které umožňují po splnění podmínky dostat se z jednoho stavu do druhého. V konečném stavovém automatu existuje právě jeden počáteční stav a několik stavů koncových. Uveďme příklad vojáka, jehož počátečním stavem je být v klidu. Pokud uvidí nepřítele, rozejde se tím směrem, pokud nepřítele nevidí, vrátí se do klidu. Další stav může být konečný, pokud našeho vojáka nepřátelská jednotka zastřelí, ocitne se ve stavu smrt, ze kterého se nemůže vrátit do jiného stavu. V případě, že je nepřítel na dostřel od vojáka, pře-

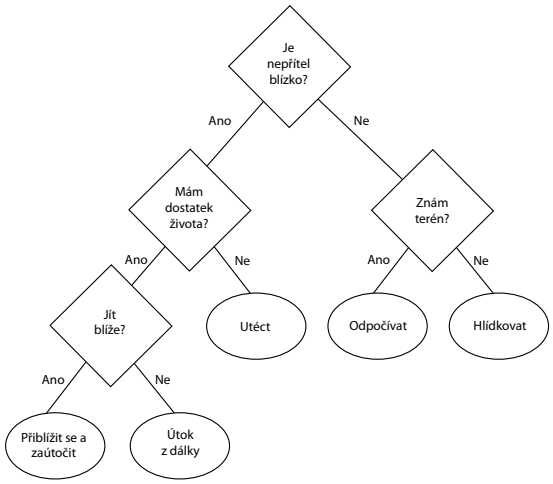
sune se do stavu útok, a jestliže nepřítele zabije, vrátí se do klidového stavu. Grafické znázornění uvedeného příkladu je na obrázku 3.3.

Stavové automaty se v počítačových hrách nepoužívají pouze k vytváření umělé inteligence, ale také například k řízení animací nebo reprezentaci stavu hry - hlavní menu, hraní hry a nastavení mohou být stavy stavového automatu, ve kterém se zrovna hra nachází. Dokonce i pro vykreslování pomocí OpenGL se používá stavový automat.

Rozhodovací stromy jsou jednoduchá stromová struktura. Každý uzel, který není listem, má dva potomky a vyjadřuje logickou podmínku s výsledkem ano či ne. Listy reprezentují atomické akce naprogramované příslušné umělé inteligenci. Pokud chceme zjistit akci z rozhodovacího stromu, která se má provést, zavoláme na kořen stromu funkci zodpovědnou za rozhodování. V případě že kořen není listem, získáme z podmínky v uzlu odpověď ano či ne a podle ní vybereme buď pravého nebo levého potomka. Takto postupujeme stromem hlouběji a hlouběji, dokud nenarazíme na list. Jakmile při prohledávání stromu narazíme na list, provede se akce v listu obsažená. Typická hierarchie rozhodovacího stromu je na obrázku 3.4.

Konkrétní implementace rozhodovacích stromů se mohou různit, v některých případech můžeme dokonce pomocí počtu volání regulovat čas potřebný k rozhodnutí. Každé volání stromu, aby se rozhodl, obsahuje jeden tik. V každém uzlu je uložen počet tiků, který je potřebný k jeho rozhodnutí. Teprve, když počet tiků z volání uzlu je roven počtu tiků potřebných k rozhodnutí, provede se akce v uzlu uložená.

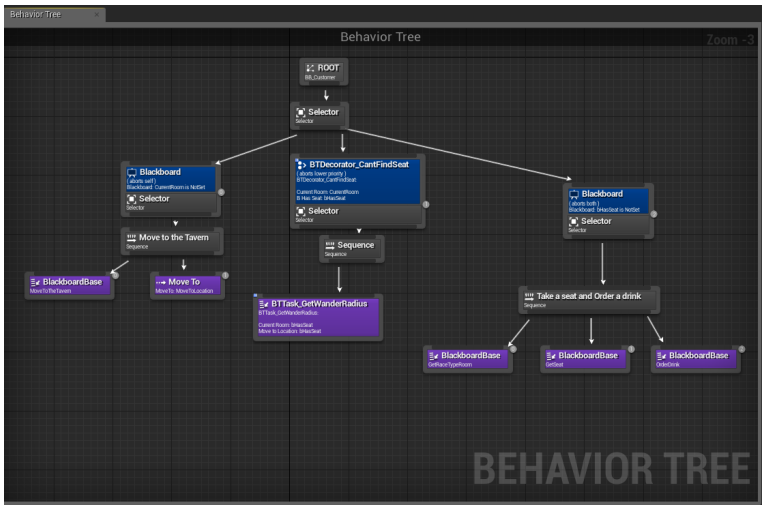
Výhodou stromové struktury je znovupoužití, což znamená, že uzly, které jsou identické, můžeme vyjádřit jedinou třídou, a namísto kopírování kusů kódu použijeme objekt této třídy. Statickou funkčnost stromu můžeme jednoduše rozšířit na dynamickou nahrazováním listů, podstromů, nebo dynamicky střídat různé stromy. Takto například můžeme měnit reakce umělé inteligence reprezentované rozhodovacím stromem na vývoj prostředí.



Obrázek 3.4. Ilustrace jednoduchého rozhodovacího stromu.

Behaviorální stromy [7] jsou do jisté míry rozšiřujícím následovníkem rozhodovacích stromů. Jsou velmi podobné co se týče způsobu používání, ale umožňují vyjádřit složitější vztahy za pomoci více rozličných uzlů, jak je možné vidět na obrázku 3.5. Každý

uzel může nabývat tří různých stavů - úspěch, běžící a neúspěch. Při rozhodování se zavolá kořen stromu, pak podobně jako u rozhodovacího stromu postupujeme po uzlech směrem dolů, až narazíme na list. Výsledná akce vykonaná v listu skončí jedním ze tří stavů. Pokud výsledkem akce byl neúspěch, akce se nezdařila a další volání bude opakováno opět na kořeni stromu. V případě že výsledkem byl stav běžící, akce potřebuje více volání k jejímu provedení a další volání bude uskutečněno na tomto listu. A pokud výsledkem byl úspěch, akce se zdařila a volání bude opět opakováno na kořeni stromu. Další typy uzlů rozšiřující behaviorální stromy jsou sequence, decorator, selector, while succes a modular tree.



Obrázek 3.5. Behaviorální stromy v Unreal Engine

Obrázek byl převzat z https://forums.unrealengine.com/showthread.php?45393AI-BehaviorisInconsistent

Sequence slouží k vyjádření návaznosti akcí, má nula a více potomků. Po zavolání uzlu jsou postupně zavoláni všichni potomci, tato akce může trvat i více tiků, například na každého potomka bude použit jeden tik.

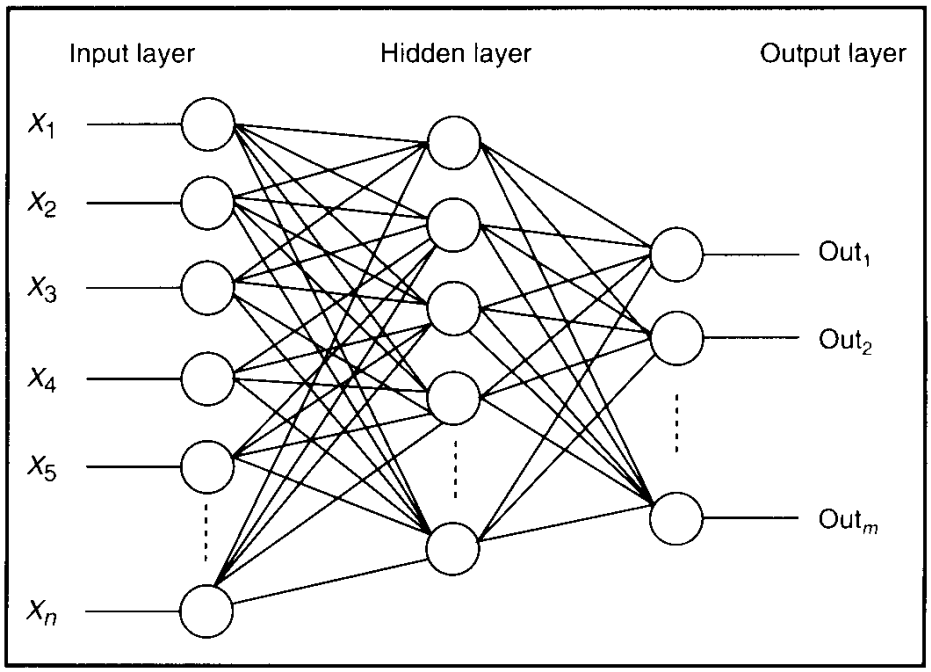
Decorator má pouze jednoho potomka, obsahuje logickou podmínku s výsledkem ano - ne. Účelem decoratoru je zjistit zda za dané podmínky má být volán podstrom tvořen potomky decoratoru.

Selector je uzel připomínající rozhodovací stromy. Selector má dva a více potomků, obsahuje logickou podmínku na základě které se vyhodnotí, který potomek má být zavolán.

While succes upravuje průchod behaviorálním stromem. Dokud podstrom toho uzlu vrací úspěch nebo stav běžící, namísto kořene stromu je volán tento uzel.

Modular tree [8] je uzel, který umožňuje napojení dalšího behaviorálního stromu. Rozdělení stromu na více na sobě nezávislých podstromů umožňuje snadněji vytvářet a pochopit jednotlivé akce a v neposlední řadě zvyšuje znovupoužitelnost.

Podobně jako u rozhodovacích stromů je možné statickou funkčnost rozšířit jednoduše na dynamickou, dokonce možná i snadněji. Behaviorální stromy mají uzel Modular tree, který je přímo určený k uchovávání podstromu, tento podstrom se dá pak jednoduše nahrazovat jiným a tím měnit celkové chování.



Obrázek 3.6. Neuronová síť

Obrázek byl převzat z
http://mechanicalforex.com/wp-content/uploads/2011/06/NN.png

Neuronové sítě [9] jsou biologicky inspirovaný výpočetní model používaný ve statistickém strojovém učení a umělé inteligenci. Typická struktura neuronové sítě je na obrázku 3.6. Pro chod neuronové sítě je zapotřebí stanovit tři základní parametry: model neuronu, který definuje, jak neuron převádí vstup na výstup, architekturu sítě, která definuje, jak jsou mezi sebou neurony propojeny a učící algoritmus, který mění parametry jednotlivých neuronů. Základní typy architektur používané v umělé inteligenci jsou dopředné a rekurentní. Výhodnější je použití rekurentních architektur, které využívají zpětnou vazbu a tedy přímo reagují na vliv prostředí, způsobený vlastní akcí. Moderní agenti používající neuronové sítě používají různé typy neuronů v jedné síti, například paměťové bloky.

Použití neuronové sítě není stejně jednoduché jako použití předchozích principů, ve kterých stačilo naimplementovat posloupnost kroků vedoucí k nějaké akci. Neuronové sítě potřebují učit jako by se učil mladý organismus dítěte, učení může probíhat dvěma způsoby. Známe výstup, kterého chceme dosáhnout a tak se rekurentně upravují váhy jednotlivých neuronů, aby se co nejvíce snížil rozdíl mezi požadovaným výstupem a výstupem z neuronové sítě. Tento druh učení se nazývá učení s učitelem. Druhý způsob je učení bez učitele, ve kterém není výstup znám a není se tedy čemu přibližovat. Proto si neuronové sítě vytváří skupiny vstupů a podle nich reagují, nebo si podle vstupu upraví vlastní topologii.

Kapitola 4
Návrh řešení

4.1 Volba nástrojů

V rámci zadání bylo nutné použít několik různých nástrojů pracujících s multimediálním obsahem. Existuje velké množství modelářů, ze kterých je možné vybírat. Ze sféry nekomerčních je pravděpodobně nejznámější Blender 3D, jeho ovládání není příliš intuitivní a proto mnoho začínajících 3D umělců odradí. V komerční sféře je opravdu široký výběr, například DAZ3D, Modo, Zbrush, Cinema 4D, Maya a další. K vytvoření modelů použitých v této práci byly použity Autodesk Maya a Autodesk Mudbox se studentskou licencí. Autodesk Maya je modelář spojující různé přístupy modelování viz. 3.2, byl použit k tvorbě většiny modelů a několika animací. Autodesk Mudbox je založen na principu virtuálního sochařství. Pomocí virtuálního sochařství nejsou vytvářeny 3D objekty, ale jsou pouze upravovány. Poté jako ve skutečném sochařství se použije šablona, do které se pak pomocí zařezávání, odebírání a přidávání materiálu začne tvarovat výsledná podoba modelu.

Přesto, že existují i jiné produkty pracující s daty z motion capture, k této práci byl využit Autodesk Motion Builder. Nástroje Autodesk jsou vytvořeny, aby byly používány společně, což přináší nesčetné výhody. Pro přenos dat mezi spuštěnými aplikacemi dokonce není nutný export do externího souboru, aplikace si mezi sebou vymění data a poté lze pracovat zároveň například v Maye a Motion Builderu.

A v neposlední řadě bylo důležité použít editor rasterové grafiky pro tvorbu textur a grafické uživatelského rozhraní. Přes velké množství různých editorů jako je GIMP a Corel Painter jsem se vzhledem k předchozím zkušenostem rozhodl použít Adobe Photoshop CS3.

4.2 Design hry

V této kapitole se čtenář seznámí s návrhem hry. Pro tvorbu počítačové hry podmíněné v zadání byl vybrán herní žánr akční realtimová strategie jak již bylo zmíněno v 3.5. Akční realtimová strategie je v dnešní době velmi populárním herním žánrem využívaným zejména v multiplayerové podobě viz. Dota2[1]) a League of Legends[2]), i přesto byla raději zvolena singleplayerová verze za použití umělé inteligence, i když rozšíření o multiplayer by nemuselo být náročné.

Hra se řídí podle herních pravidel popsaných v 3.5. Jednou z mála změn, která je však poměrně zásadní, je změna mapy. Běh herního děje se odehrává na jedné lince, která je symetrická okolo středu. Po úbočí linky jsou rozmístěny neutrální jednotky,

[1]) `http://blog.dota2.com/?l=english`
[2]) `http://leagueoflegends.com`

takzvaná džungle a teleporty. Tyto jednotky čekají na místě, dokud není jednotka z jiného týmu dostatečně blízko, aby na ni zaútočila. Teleporty slouží k přemístění ke své hlavní budově. Další změnou je reprezentace věží a hlavní budovy. V této implementaci bylo rozhodnuto, že pro ozvláštnění budou reprezentovány pomocí jednotek, které se při útoku hýbou po mapě. Jejich pohyb je omezen v určitém rozsahu a bez cíle, na který by mohly útočit, se vrací do výchozí pozice. Okolí věží a hlavní budovy poskytuje jednotkám procházejícím tímto okolím bonus. V okolí hlavní budovy se jednotka léčí, v okolí věží běží rychleji, nebo má silnější obranu. Hra zachovává princip hraní za jednu hlavní postavu, která se v průběhu hry zlepšuje, zvyšuje se její úroveň, nakupuje předměty a kouzla.

Hlavní postava začíná na úrovni 1, maximální úroveň je 16. Pokud hlavní postava zemře, délka jejího úmrtí je rovna čtyřnásobku její úrovně. Po uplynutí této doby se postava zrodí u hlavní budovy. Úroveň všech jednotek jednoho týmu je rovna úrovni hlavní postavy stejného týmu. Úroveň neutrálních jednotek je maximum z úrovní obou týmů.

Jak vyplývá z herního mechanismu, důležitou součástí hry jsou finance, které slouží k nákupu kouzel, předmětů a jejich vylepšování. Hráč začíná s 1 000 herních peněz určených na první nákup, přičemž každou vteřinou hráč získává 1 herní peníz. Další peníze navíc je možné získat zabíjením věží, malých jednotek, hlavní postavy nepřítele, nebo také tím, že hlavní postava stojí poblíž umírající nepřátelské jednotky. Při souboji jednotek také hraje velkou roli jejich vzájemné postavení. Pokud je na jednotku útočeno zezadu, může poškození jí udělené být až dvakrát vyšší nežli zepředu. I toho se dá využít k rychlejšímu získání peněz. Jen pro představu zakoupení plné výzbroje a některých kouzel může stát i okolo 30 000. To určuje náplň první části hry a to získat zabíjením co nejvíce peněz na vylepšení své postavy. S vylepšenou postavou je mnohem jednodušší získat potřebnou převahu.

Každá jednotka je specifikována několika vlastnostmi, jejichž hodnoty jsou součástí herních tabulek. Pro lepší pochopení je nejprve nutné seznámit čtenáře s vlastnostmi popsanými v tabulce, zkratkami, a až poté s tabulkami.

Úroveň (LVL) určuje úroveň jednotky v rozmezí 1 až 16. Životy (HP) určují zdraví jednotky. Pokud má jednotka 0 životů je zabita. Mana (MN) určuje speciální vlastnost, která umožňuje sesílat kouzla. Odolnost vůči kouzlům (SR) slouží k určení míry, jakou jednotka redukuje přijímané zranění od kouzel. Brnění (AR) slouží k určení míry, jakou jednotka redukuje přijímané zranění z útoků. Regenerace života (HR) určuje počet navrácených životů za vteřinu. Regenerace many (MR) určuje počet navrácené many za vteřinu. Zkušenosti za úmrtí (XP) je počet zkušeností, které cizí jednotka obdrží za zabití této jednotky. Útočné poškození (AD) je hodnota definující útok jednotky. Kouzelné poškození (SP) je hodnota definující sílu kouzel jednotky. Zkušenosti potřebné na další úroveň (LXP) je počet zkušeností potřebných k dosažení další úrovně jednotky. Peníze (G), které obdrží cizí jednotka za zabití této. Nyní je možné přejít k příkladu tabulky hlavní postavy 4.1, která udává všechny hodnoty postavy od 1. do 16. úrovně. Zbývající tabulky herních postav jsou umístěny v příloze C.

Základní vlastnosti hlavní postavy se dají vylepšovat nákupem předmětů. Po zakoupení předmětu je možné ho dále vylepšovat vsazením drahokamu. Každá jednotka může mít až pět předmětů, právě jednu zbraň, jedny boty, jedno brnění a až dva prsteny. Vy-

LVL	HP	MN	AR	SR	HR	MR	XP	AD	SP	LXP	G
1	550	226	10	5	1.8	1.9	300	40	40	1204	250
2	554	230	11	6	1.8	1.9	702	42	41	4294	250
3	563	240	12	7	1.9	2	1112	45	42	9632	250
4	582	257	12	8	1.9	2.1	1508	48	43	17474	250
5	612	280	12	9	2	2.3	1880	52	45	28013	250
6	658	313	13	10	2.2	2.6	2232	56	49	41409	250
7	721	354	13	12	2.4	3	2560	61	53	57797	250
8	806	406	13	14	2.7	3.4	2866	67	57	77293	250
9	914	468	14	16	3	3.9	3152	74	61	100000	250
10	1050	541	14	18	3.5	4.5	3422	83	65	126008	250
11	1215	626	14	20	4.1	5.2	3674	94	70	155402	250
12	1414	723	15	25	4.7	6	3910	114	75	188256	250
13	1648	834	15	30	5.5	7	4134	130	81	224641	250
14	1922	958	15	35	6.4	8	4344	148	87	264620	250
15	2237	1096	16	40	7.5	9.1	4544	168	94	308254	250
16	2598	1249	20	50	8.7	10.4	0	192	100	inf	250

Tabulka 4.1. Tabulka hodnot hlavní postavy.

bavení jednotky odpovídá reálnému modelu vybavení válečníka a eliminuje nesmyslné situace jako například používání pěti zbraní zároveň.

Každý předmět má definovaný počet, kolikrát je možné ho vylepšit. Vylepšení se provádí pomocí drahokamů, které zvyšují některé vlastnosti předmětů. Kombinací drahokamů je poměrně jednoduše možné reagovat na akce nepřítele. Vylepšováním specifických vlastností je možné podpořit hráčovu herní strategii. Například je jasné, že nákupem drahokamů se životy a brněním se hráč staví do defenzivní pozice, která má za následek větší výdrž proti základním útokům hlavního hrdiny nepřítele, malých jednotek a dalších. Naopak nákup drahokamů zlepšující kouzelné poškození, hráče staví spíše do útočného postavení, ve kterém musí být opatrný a bude útočit z dálky pomocí kouzel. Vylepšování předmětů tedy rozšiřuje možnost taktizování a reagování na nepřítele. Podobně jako u jednotek i hodnoty vlastností předmětů a drahokamů jsou definovány v tabulce.

Nyní se čtenář seznámí s vlastnostmi a použitými zkratkami v tabulce 4.2 a 4.3. Jelikož většina vlastností a zkratek jsou obdobné jako vlastnosti jednotky, nebudou zde uvedeny.

Počet vylepšení(UC) označuje kolikrát je možné předmět vylepšit, tedy vložit do něj drahokam. Cena(C) udává pořizovací cenu předmětu, nebo drahokamu.

Předmět	UC	AD	SP	SR	AR	MS	HP	MN	HR	MR	C
Meč	4	25	0	0	0	0	0	0	0	0	300
K. meč	4	0	25	0	0	0	0	100	0	0	300
Boty	2	0	0	0	0	5	0	0	0	0	250
Č. prsten	1	0	5	0	0	0	50	100	1.5	1.5	200
M. prsten	1	5	0	0	0	0	100	50	1.5	1.5	250
Brnění	6	25	0	0	0	0	150	150	2.5	2.5	500

Tabulka 4.2. Tabulka hodnot předmětů.

Drahokam	AD	SP	SR	AR	SP	HP	MN	HR	MR	C
Červený	15	0	0	0	0	0	0	0	250	
Černý	0	0	0	15	0	100	0	1	0	250
Žlutý	10	10	0	0	0	0	0	2	2	250
Čirý	0	0	20	0	0	0	150	0	1.5	250
Duhový	5	5	5	5	1	50	50	2.5	2.5	500
Modrý	0	15	0	15	0	0	75	0	0	250
Zelený	0	0	5	5	0	100	100	1	1	250

Tabulka 4.3. Tabulka hodnot vylepšení.

Další možností, jak může hráč vyjádřit svojí strategii, je nákup kouzel. Každá jednotka může mít až šest různých kouzel. Různé typy kouzel a efektů opravdu široce zpestřují průběh hry. Kouzla můžeme podle typu rozlišovat na plošná, cílená, mířená, zblízka a sesílaná na sebe. Plošná kouzla vytvářejí plochu, do které když vstoupí jednotka, je tímto kouzlem zasažena. Cílená kouzla jsou sesílaná přímo na jednotku, před těmito kouzly není možné utéct. Po dobu letu upravují svůj směr, dokud jednotku nezasáhnou. Mířená kouzla jsou sesílaná určitým směrem a letí, dokud nezasáhnou nějaký cíl, nebo nezmizí na konci doletové vzdálenosti. Kouzla sesílaná zblízka jsou speciální útoky, cílené na jednu jednotku. Kouzla sesílaná na sebe ovlivňují pouze postavu, která je sesílá. Po seslání kteréhokoli kouzla nastává prodleva, po kterou toto kouzlo není možné seslat.

Každé kouzlo může mít efekt, kterým ovlivní zasaženou jednotku na určitou dobu. Tyto efekty jsou zapálení, zmražení, otrávení, omráčení, zrychlení, štít, léčení, kopnutí, krvácení, teleportace, vícenásobný útok a zvýšení útoku. Nyní bude uvedeno co, který efekt dělá a k němu bude v závorkách přiřazená zkratka. Zapálení (F), krvácení (B), vícenásobný útok (C) a otrávení (P) po krátkou dobu zraňuje jednotku. Zmražení (I) a kopnutí (K) je efekt zpomalující jednotku. Omráčení (O) dočasně zcela znehybní jednotku. Zrychlení (Z) zvýší rychlost pohybu jednotky. Štít (Š) zvyšuje obrané vlastnosti jednotky. Léčení (L) navrací životy jednotce. Zvýšení útoku (Ú) přidává dočasně bonus k útočným vlastnostem jednotky. Teleportace (T) okamžitě přenese jednotku daným směrem.

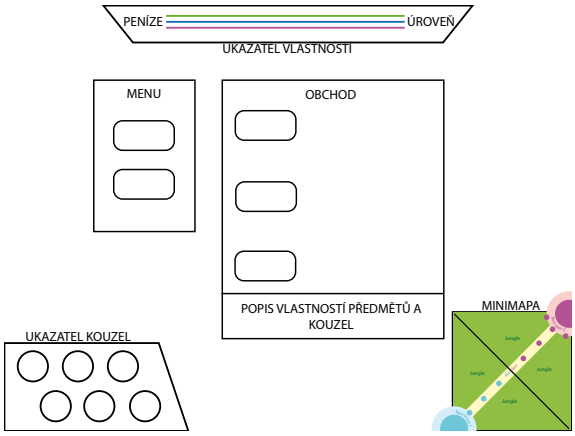
Tabulka reprezentující kouzla je uvedena za krátkým přehledem vlastností kouzel a použitých zkratek. Úroveň (LVL) určuje úroveň koupeného kouzla. Rychlost pohybu (MS) určuje jakou rychlostí se kouzlo hýbe. Cena (C) je celková cena kouzla. Vzdálenost (RNG) na kterou je možné kouzlo seslat. Hodnota kouzla (DMG) slouží k výpočtu zranění, léčení a dalších efektů. Trvání (DUR) určuje dobu, jak dlouho po vyvolání bude kouzlo existovat. Mana (MN) je mana potřebná k seslání kouzla. Prodleva (CD) po seslání kouzla je doba, po kterou musí jednotka počkat, než bude moci kouzlo seslat znovu. Trvání efektu (EDUR) určuje dobu, po kterou je jednotka ovlivněna efektem kouzla. Hodnota efektu (EV) slouží pro výpočet zranění, léčení a dalších hodnot po dobu trvání efektu. Příklad tabulky mířeného ohnivého kouzla viz. 4.4, všechna kouzla jsou popsána v příloze C.

Uživatelské rozhraní bylo navrženo z pěti základních bloků viz. grafický návrh 4.1. Ukazatel vlastností hlavní postavy, ukazuje v grafické i textové podobě aktuální počet životů, many, zkušeností, peněz a úroveň hráče. V textové podobě je hodnota vyjád-

Efekt LVL	F MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	70	500	160	100	0.5	5	50	5	75
2	71	750	165	225	0.75	23	40	5	90
3	72	1000	170	550	1	83	30	5	110
4	73	1250	175	900	1.25	180	18	5	135

Tabulka 4.4. Tabulka hodnot ohnivého kouzla.

řena aktuální stav/ maximální stav. Oproti tomu je v grafické podobě aktuální hodnota vyjádřena pomocí barevného panelu, který mění svou šířku v rozmezí 0 až 100 procent podle poměru aktuální hodnoty a maximálního stavu. Nedílnou součástí u her tohoto typu je minimapa. Minimapa je zmenšený obrázek mapy herního světa, který zobrazuje pozice jednotek ve hře. Každý tým je na mapě odlišen jinou barvou. Pravděpodobně nejdůležitější komponentou uživatelského rozhraní je obchod, který spojuje dříve popsané návrhy. V obchodě je možné nakupovat kouzla, předměty a drahokamy uvedené v herních tabulkách a vybavovat s nimi hlavní postavu. Obchod spojuje informativní a funkční část, pomocí obchodu je možné zjistit vlastnosti kouzel, předmětů a drahokamů a vybrat ten nejvhodnější v dané situaci. Funkční částí je myšleno právě ono vylepšení hlavní jednotky. Další částí uživatelského rozhraní, přímo navazující na obchod, je ukazatel kouzel hlavní postavy. Jak již bylo v návrhu popsáno, hlavní postava může mít až šest kouzel, po zakoupení kouzla v obchodě se kouzlo objeví v tomto ukazateli. Po seslání kouzla nastává prodleva, po kterou není možné kouzlo opětovně seslat, tento fakt je vyjádřen kruhovou výsečí, která se zmenšuje s uplynulým časem. Poslední součástí uživatelského rozhraní je herní menu, které umožňuje spustit hru znovu, nebo hru ukončit.



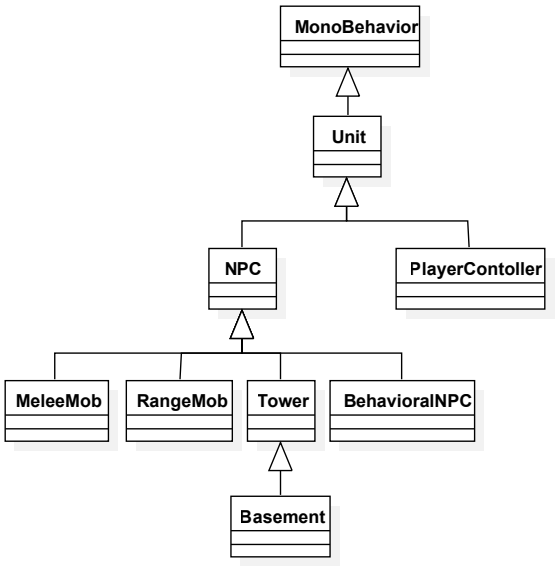
Obrázek 4.1. Grafický návrh GUI.

Zásadním bodem při návrhu bylo co nejvíce zjednodušit ovládání. Ve výsledné formě je možné hru ovládat pouze myší a několika málo klávesami. Pravým tlačítkem myši je určován cíl, kam má postava dojít, nebo na koho zaútočit. Levým tlačítkem myši je možné interagovat s uživatelským rozhraním. Při vyvolávání kouzla se pozicí myši určí cíl, na který bude kouzlo sesláno, poté je nutné stisknout příslušnou klávesu, kouzla 1 až 6 jsou namapována na klávesy 1,2,3,Q,W,E. Pomocí mezerníku je možné okamžité skočení pohledem na hlavní postavu. Již při návrhu bylo jasné, že některé animace nebudou mít ve hře praktické využití, ale přesto byly použity jako zábavná vsuvka,

jejich spuštění je namapováno na klávesy F1, F2. Pomocí klávesy ESC se vyvolá menu, které umožňuje začít novou hru, nebo stávající opustit.

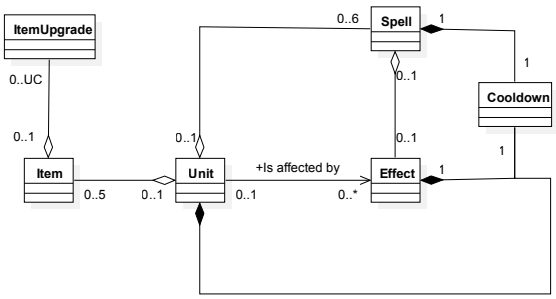
4.3 Návrh tříd

Design hry poměrně čistě vytyčil návrh tříd a vztahy mezi nimi. Jak je vidět na obrázku 4.2, k reprezentaci jednotek byla použita poměrně abstraktní struktura k vyjádření jejich funkčnosti. Všechny třídy dědí z MonoBehavior, což je třída v Unity, která slouží k vyjádření schopnosti konat akci ve hře. Abstraktní třída Unit definuje principy společné jednotkám jako je regenerace, inventář, kouzla, souboj, výpočet zranění a ohlídání smrti. Z této třídy dále dědí PlayerController, ta je zodpovědná za zpracování vstupů uživatele a NPC, která je základem pro všechny počítačem řízené jednotky.



Obrázek 4.2. Diagram tříd.

Dříve definované jednotky používají poměrně přehledný mechanismus souboje, datové třídy potřebné k jeho chodu jsou na obrázku 4.3. Poměrně důležitou a dříve zmiňovanou vlastností je prodleva, která je reprezentovaná třídou Cooldown. Jak je možné vidět, třída Cooldown je hodně využívaná, v jednotkách reprezentuje prodlevu mezi dvěma útoky a prodlevu pro seslání kouzla. Dale určuje jak dlouho ještě bude efekt působit. Ostatní vztahy přímo korespondují s designem hry.



Obrázek 4.3. Vztahy datových tříd k jednotce.

4.4 Umělá inteligence ve hře

Většina jednotek ve hře jsou pouze reakční mechanismy s prostým cílem a není od nich vyžadováno složité chování. Výjimkou je hlavní postava nepřátelského týmu, na kterou jsou kladeny vyšší nároky. Postava by měla být schopna adekvátně zhodnotit situaci, nepouštět se bezhlavě do rizikových akcí, snažit se využívat příležitostí a také by měla vědět, kdy je správný čas utéct. Právě tyto klíčové aspekty byly zohledněny při tvorbě pokročilých reakčních mechanismů pro počítačem řízenou hlavní postavu.

K tvorbě pokročilých reakčních mechanismů byla použita vlastní implementace behaviorálního stromu. K uzlům popsaným v 3.6 byl přidán uzel inverse, který převrací návratovou hodnotu z volání podstromu. Úspěch změní na neúspěch a obráceně, hodnota běžící zůstane netknuta.

Implementovaný reakční mechanismus je poměrně dobře znovupoužitelný behaviorální strom. Tento strom se skládá ze tří komplexních částí, které spolu vytvářejí celkové chování jednotky. První dvě části stromu jsou různé modulární stromy, které každý sám o sobě reprezentují určitou část chování jednotky. Poslední část, ze které se strom skládá, propojuje tato modulární chování a určuje defenzivní chování jednotky. Pro vytvoření nového reakčního mechanismu tedy stačí změnit jeden ze dvou modulárních stromů. Nahraditelné modulární stromy jsou odpovědné za nákup a útok.

Nyní bude popsáno chování a strategie, které je ve hře implementováno. Chování použité ve hře má pevně definovaný nákupní seznam. V průběhu implementace hry bylo vyzkoušeno několik různých pevně definovaných seznamů, které předměty, kouzla a vylepšení nakoupit. Pro nepřátelskou hlavní jednotku byla zvolena poměrně defenzivní strategie. Proto byly voleny pro nákup kouzla, předměty a drahokamy, které měly za následek co nejdéle prodloužit život této jednotky, což následně vede k vyšší úrovni a více penězům. Jakmile jednotka ztratí většinu života, pokusí se vyléčit kouzlem. Pokud se léčení nezdaří, jednotka se pokusí utéct k hlavní budově. Po doplnění zdraví se vrací zpět do boje. Útočná strategie jednotky je velmi opatrná. Do vyšší úrovně se jednotka zastavuje na hranici dosahu věže, aby neutrpěla zbytečné poškození. Jako hlavní cíl jsou vybírány malé jednotky, za které získává peníze. Na nepřátelskou hlavní postavu útočí pouze v případě, že má nepřátelská jednotka málo životů a je tedy v nevýhodě. Kouzla, které si jednotka zakoupila, používá pouze proti nepřítelátelské hlavní jednotce. Při kouzlení jednotka šetří manou tak, aby bylo možné vždy léčení kouzlem.

Z časových důvodů nebylo možné stihnout implementaci druhého reakčního mechanismu, přesto by bylo dobré zmínit některé aspekty z jeho návrhu. Druhý mechanismus pro řízení hlavní postavy měl být poněkud složitější. Jeho cílem bylo upravit rozhodování podle získaných dat v průběhu hry. Mechanismus se měl rozhodovat na základě tří různých statistik, které by se měnily při každé interakci s nepřítelem. První statistika by zpracovávala celkové zranění, které bylo této jednotce způsobeno a podle poměru zranění od kouzel a celkového zranění by měla určit, zda nakupovat obranu proti útokům, nebo proti kouzlům.

$$statistika_{zranění} = zranění_{kouzelné}/zranění_{celkové}$$

Výsledná hodnota se nachází v uzavřeném intervalu $[0, 1]$. Pro zjištění jakou obranu nakupovat, by bylo vygenerováno náhodné číslo $A \in [0, 1]$. Pokud by číslo

$A \in [0, statistika_{zranění})$, pak by jednotka nakupovala obranu proti kouzlům. V opačném případě, pokud by číslo $A \in (statistika_{zranění}, 1]$ jednotka bude nakupovat obranu proti útokům. Ve výjimečném případě kdy $A = statistika_{zranění}$ je možné rozhodnout náhodně.

Druhá statistka měla během konfliktu s nepřátelskou hlavní postavou zjistit, zda je výhodnější používat kouzla, nebo útoky.

$$statistika_{útok} = SR/(AR + SR)$$

Pak podobně jako u první statistiky by bylo vygenerováno náhodné číslo $A \in [0, 1]$. Pokud by číslo $A \in [0, statistika_{útok})$, pak by jednotka nakupovala předměty, drahokamy a kouzla zblízka. V opačném případě, pokud by číslo $A \in (statistika_{útok}, 1]$, jednotka bude nakupovat ostatní druhy kouzel, předměty a drahokamy podporující sílu kouzel. V případě kdy $A = statistika_{útok}$, což znamená, že nepřátelská hlavní postava má stejné brnění i odolnost vůči kouzlům, pak je možno rozhodnout náhodně.

Účelem třetí statistiky bylo zjistit, kdy je správný čas aby jednotka začala utíkat. V průběhu každého konfliktu se spočítala průměrná hodnota zdraví jednotky, pokud byla hodnota nižší než celková průměrná hodnota přes všechny konflikty, nastal správný čas kdy by měla jednotka začít utíkat.

Pomocí těchto tří statistik měl být rozšířen původní rozhodovací mechanismus, a tak vytvořit jednotku, která nereaguje pouze na akce během konfliktu, ale také na vývoj celé hry.

4.5 Procedurální generování hudby

Dalším požadavkem zadání bylo zvážit procedurální generování hudby. Procedurální generování hudby je kreativní činnost, která vyžaduje znalosti z oblasti hudební nauky, zpracování signálu a programování. Často je spojováno s různými multimediálními aplikacemi, mappingem a také počítačovými hrami. Jedná se o přístup, který namísto manuálního skládání používá počítačové algoritmy ke generování hudby. Existuje mnoho opravdu velmi odlišných přístupů jak, podle čeho a jakou hudbu skládat. Výhodou procedurálně generované hudby je nízká datová velikost, protože při jejím použití nemusí být nikde na disku uložen žádný hudební soubor, pouze kód hudbu provádějící. Další výhodou je možnost reagovat například na vstup uživatele, na počet životů hráče, na blížící se nebezpečí a mnoho dalších akcí, které se ve hrách a multimediálních aplikacích mohou vyskytnout.

Pro účely této práce byl vytvořen jednoduchý generátor hudby založený na generování akordů pomocí cellulárního automatu[1]). Generátor přehrává vygenerované akordy ze stupnice E dur. Pokaždé když přestane hrát jeden akord[2]), začne hrát nějaký jiný. Generátoru je možné nastavovat tři různé parametry, amplitudu výsledného signálu, kterou se dá regulovat hlasitost, délku ADSR obálky[3]), tím je možné měnit délku tónu a vzorkovací frekvenci.

[1]) http://cs.wikipedia.org/wiki/Celul%C3%A1rn%C3%AD_automat
[2]) http://cs.wikipedia.org/wiki/Akord
[3]) http://en.wikiaudio.org/ADSR_envelope

Cellulární automat je v této implementaci tvořen jedním stavovým řádkem o osmi hodnotách, který obsahuje informaci hrát či nehrát tón a osmi pravidly, které převádí současný stavový řádek do nového stavového řádku. Způsob jakým tento generátor funguje, vzdáleně připomíná Hammondovy varhany, protože pro generování zvuku používá osm různých sinusových signálů. Jejich frekvence je shodná s tóny ze stupnice E dur, ty se skládají dohromady v akordy pomocí stavového řádku vygenerovaného cellulárním automatem. Poté se takto vytvořené akordy namodulují ADSR obálkou a nahrají do bufferu, který je dopraví až na zvukovou kartu.

V současné době není generátor ovlivňován průběhem hry, ale je naprosto triviální rozšířit jeho funkčnost, aby reagoval například délkou tónu na počet životů hráče, nebo hlasitostí na vzdálenost od nepřítele.

Kapitola 5
Průběh práce

5.1 Motion capture

Jedním z hlavních bodů zadání bylo analyzovat možnosti zařízení motion capture v místnosti laboratoře virtuální reality. Zařízení zde umístěné pochází od firmy Optitrack[1]), jedná se o optický systém s markery viz. 3.4, který se skládá ze šesti kamer, switche a pracovní stanice, na které je umístěn Optitrack Motive[2]). Motive je nástroj umožňující záznam a čištění dat z motion capture. Kamery jsou pomocí síťových kabelů připojené ke switchi, ten je připojen k pracovní stanici. K motion capture je poskytnuto nutné vybavení v podobě jednoho elastického obleku se suchým zipem, brýlí s markery, velkého počtu markerů se suchým zipem, které se připevňují na oblek, hůlky s markery, která slouží ke kalibraci kamer a osovníku, který po zkalibrování slouží k vymezení souřadného systému souřadnic.

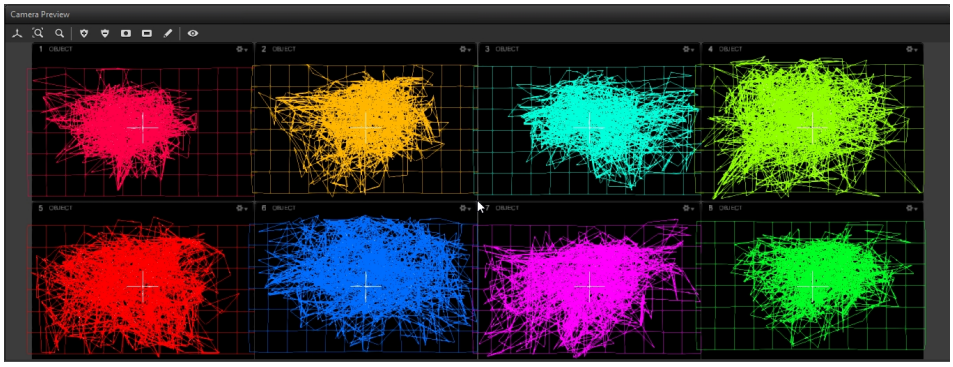
Vybraný herní žánr poměrně jednoznačně vytyčil pohyby, které je potřeba natočit. Jedná se o běh, chůzi, klidový stav, útoky se zbraní, tanec, a další. Jak je možné si představit, některé z těchto pohybů jsou rozsáhlé, co se prostoru týče. A právě prostor byl v laboratoři virtuální reality jedním z klíčových aspektů. Jedná se o poměrně malou místnost, ve které jsou kamery umístěné v obdélníku 3,5 m x 1,5 m a ve výšce přibližně 2,5 m, což pro rozsáhlé pohyby není velký prostor. Proto prvním cílem bylo pokud možno zvětšit snímaný prostor pomocí přemístění některé z kamer na stativ.

Po zapnutí Motive, se kamery připojí k aplikaci. Nyní bylo možné určit největší prostor, který mohly kamery zaznamenat. Je důležité pamatovat stále na to, že každý marker, který chceme zaznamenat, musí být po celou dobu viděn alespoň dvěma kamerami. Kamery je nutné natočit tak, aby actor procházející jejich záběrem byl zabírán co nejdéle všemi kamerami.

V okamžiku kdy jsou kamery správně nasměrovány, přichází na řadu jejich kalibrace, která slouží v první řadě k nalezení jejich vzájemné pozice. Čím lepší kalibrace, tím vyšší přesnost výsledně získáme na zaznamenaných datech. Před kalibrací je nutné odstranit nebo zakrýt všechny předměty, které by mohly být systémem rozpoznány jako marker. Předměty, zobrazující se v systému jako marker, které nelze ničím překrýt umožňuje Motive zablokovat a během pořizování dat budou ignorovány. Po vyčištění scény je možné postoupit k samotné kalibraci, která se provádí speciální hůlkou ve tvaru T s markery na konci. V Motive se zapne kalibrace a pomocí hůlky se snažíme pokrýt co největší objem v pohledu kamer. Máváním hůlkou v prostoru se zabarvují pohledy kamer, obrázek 5.1. Po nasbírání dostatečného počtu vzorků je nutné nechat Motive propočítat pozice kamer v prostoru. Výpočet probíhá iterativně, v každé iteraci je vidět

[1]) http://www.optitrack.com/
[2]) http://www.optitrack.com/products/motive/

vypočítaná přesnost a ohodnocení kvality. Výpočet je kdykoli možné zastavit, vhodné je počkat až se data ustálí a poté výpočet ukončit. Po skončení kalibrace položíme osovník na místo, kde chceme, aby byl počátek souřadné soustavy souřadnic a uložíme jeho pozici v Motive. Nyní je možné osovník odnést, do další kalibrace nebude potřeba. Po těchto úkonech nás motive vyzve k uložení kalibrace. Pokud se s kamerami mezi nahráváními nehýbalo, je možné používat uloženou kalibraci, v opačném případě se musí udělat kalibrace nová.

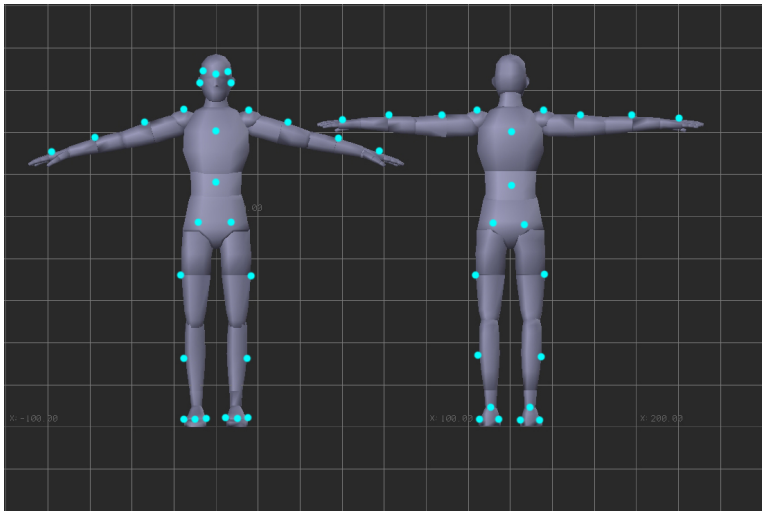


Obrázek 5.1. Pohled kamery během kalibrace.
Obrázek byl převzat z http://wiki.optitrack.com/index.php?title=Calibration

Před natáčením je možné v Motive zaregistrovat rigid bodies, česky pevná tělesa, jako jsou například brýle s markery. Zaregistrování probíhá velmi jednoduše, daný objekt umístíme do scény, označíme v Motive markery patřící do rigid body a necháme vytvořit rigid body. Takto vytvořené rigid body má několik výhod, lze jej pojmenovat a markery v něm obsažené se pojmenují automaticky. Motive tato rigid body po příchodu do scény umí rozeznat a vhodně je zvýrazňuje. A v neposlední řadě při zpracování rigid body je možné efektivně použít markery navzájem k opravě jejich trajektorie.

Nyní poprvé přichází na řadu actor, je nutné ho obléct do speciálního obleku a polepit markery. Oblek by měl co možná nejtěsněji přiléhat k tělu herce, kdyby se během natáčení oblek různě posouval, nasnímaná data by tuto chybu obsahovala také. Vzhledem k plánovanému využití Autodesk Motion Builder není rozumné k vyjádření pohybu a rotace jedné části těla použít více než pět markerů, pět je maximální počet markerů které v Autodesk Motion Builderu mohou být namapovány na jednu část těla. Dále pokud chceme vyjádřit rotaci části těla, je nutné použít nejméně dva markery, protože optické systémy nejsou schopné pomocí jednoho markeru určit rotace. Rozmístění markerů není náhodné, příklad jejich rozmístění je vidět na obrázku 5.2.

V průběhu natáčení byly použity dvě různá rozmístění markerů. První natáčení bylo provedeno v laboratoři virtuální reality. Zde bylo použito rozmístění markerů, které umožňovalo dokonce i snímání rotace zápěstí a ohybu prstů. Tento způsob rozmístění se neosvědčil, vzhledem k malému prostoru, ve kterém byly nahrávky pořizovány. Většina dat nesoucích informace o markerech na koncích rukou byla nekvalitní, v některých případech dokonce nepoužitelná. V případě druhého natáčení, které probíhalo ve větších prostorách v ČVUT IIM, bylo použito minimalistické rozmístění markerů jako na obrázku 5.2. Během druhého natáčení byly nahrány prostorově náročné pohyby a to běh a chůze.



Obrázek 5.2. Vhodné rozmístění markerů na actorovi.

Ve chvíli kdy je herec oblečen do obleku již nic nebrání natáčení. V Motive je nahrávání velice jednoduché, po celou dobu nahrávání je možné sledovat markery v náhledu scény. Pokud, jako v případě této práce, probíhá natáčení ve stísněném prostoru, je možné že některý marker opustí scénu a poté se vrátí zpět. Každé takovéto opuštění scény znamená přerušení návaznosti dat, které bude nutné nakonec opravit, proto může být dobré některé pohyby vyzkoušet nanečisto, případně upravit choreografii a až poté je nahrát. V průběhu natáčení je vhodné postupovat podle předem stanoveného scénáře a pro jistotu nahrát více stejných nahrávek, z nichž je možné při zpracování vybírat, která se výsledně upraví a použije. Hojně využívaná pozice je takzvaná T-pose, kdy actor stojí narovnaný ve stoji spatném, má roztažené ruce ve výši ramen a hledí vpřed. Na začátku nahrávky se actor posatví do T-pose, začne se nahrávat, poté předvede choreografii a nakonec se vrací opět do T-pose, tím nahrávka končí. Je nutné podotknout, že používání T-pose není naprosto nutné, ale v dalším zpracování dat z motion capture ulehčuje práci.

Obrázek 5.3. Reprezentace trajektorie v osách x, y, z pomocí signálu.
Obrázek byl převzat z http://wiki.optitrack.com/index.php?title=Timeline_Pane

Když jsou všechny nahrávky ze scénáře natočeny, nastává čas k jejich zpracování. Nejprve je nutné nahrávku, kterou chceme upravit trajektorizovat. Během trajektorizace se z natočených nahrávek vytvoří trajektorie jednotlivých markerů. Tyto trajektorie jsou v Motive reprezentovány v grafu pomocí signálu a je možné je tímto způsobem také upravovat. Reprezentací pomocí signálu se myslí fakt, že trajektorie každého markeru je vyjádřena v grafu odchylkou v osách x, y, z / snímek, v této chvíli bude názornější použít obrázek 5.3. Jak je vidět na obrázku, jedná se o trajektorii jednoho markeru v osách x, y, z. Také je možné vidět, že trajektorie markeru není spojitá a na některých místech jsou zaznamenané výchylky způsobené například otřesem kamery, prokluzováním obleku na actorovi a také v místě kde marker opustil nebo se vrátil do scény. Tato místa jsou na obrázku 5.3 vyznačena bledě modrými kroužky.

Samotné zpracování dat se dá poměrně jednoduše rozdělit na navazující akce. Jako první je nutné spojit trajektorie markerů, které opustily scénu. Pokud nějaký marker na actorovi opustil scénu Motive není schopen zjistit, že tento marker se vrátil zpět a proto vytvoří nový marker, jehož trajektorie bude začínat v čase, kdy se vrátil marker na actorovi do scény. Je nutné tedy vyhledat takto navazující trajektorie a spojit je dohromady. Tento postup se aplikuje na každý marker, který opustil během nahrávání scénu a může být aplikován i vícekrát, vzhledem k tomu kolikrát marker scénu opustil. Další akce je prováděná opět nad markery, které opustily scénu a to poté co byly pospojovány jejich trajektorie. V této fázi je důležité zazáplatovat díry v trajektoriích, pro které data nemáme, tato situace odpovídá obrázku 5.3. Před záplatováním je dobré se podívat na trajektorii okolo díry, jestli v nejbližším okolí není výrazný skok, takový skok je dobré vymazat, protože se do záznamu dostal opuštěním markeru ze scény. K záplatování trajektorií Motive nabízí mnoho přístupů a je jen otázkou, který je v dané situaci nejvýhodnější. První možností je dokreslení dat do grafu ručně, toto je však nejméně přesná metoda, která se používá pouze v krajních případech. Některé díry se dají zazáplatovat pomocí nahrazení konstantou, kdy poslední hodnota před dírou je zkopírována do všech snímků až ke konci díry. Ale hlavně se využívá záplatování pomocí interpolací, lineární a kubické, které se snaží dopočítat trajektorii o něco reálněji. V některých případech, je také dobré využít ostatních markerů s podobnou trajektorií a podle nich nechat dopočítat část chybějící trajektorie. Přes veškerou snahu o správné záplatování se může stát, hlavně pak u dlouhých úseků, které chyběly, že data nebudou zrekonstruována tak, aby vypadala uvěřitelně. V tomto případě je nutné doupravit pohyby výsledně, až budou namapovány na 3D model. Poslední akcí, kterou je vhodné udělat předtím, než se nahrávka považuje za hotovou je vyhladit data v místech, kde jsou vidět šumy a jednorázové špičky připomínající Diracovo delta.

Po zpracování nahrávky se musí ještě vymazat přebytečná data, která nepatří k žádné funkční trajektorii, jedná se o šum. Vyčištěná data se vyexportují ve formátu c3d a uloží se k dalšímu zpracování v Autodesk Motion Builderu.

5.2 Zpracování animací a přenos na 3D modely

V rámci této práce vzniklo množství 3D modelů, jejich podobu je možné nalézt v příloze B, kde jsou přehledně uvedeny v tabulce B.1. Některé modely obsahují kostru a control rig vytvořený pomocí Maya HumanIK. Control rig v tomto případě představuje

zástupnou kostru, využívající inverzní kinematiku a různé omezení pro zjednodušení práce při tvorbě animací a retargetingu. Na kostru je 3D model v Maye připevněn pomocí Smooth bind, tedy jako kůže která se transformací kostry deformuje [1]). Každá kost po připevnění kůže obsahuje informace, které vrcholy a do jaké míry ovlivňuje. Jelikož jsou animace zamýšlené pro herní enginy Unity, každý vrchol může být ovlivňován nanejvýš čtyřmi kostmi[2]). Jak již bylo řečeno, kosti obsahují informace o tom, do jaké míry ovlivňují vrcholy takzvanými váhami. Maya po připevnění kůže ke kostře vytvoří vlastní váhy, které nejsou nikterak přesné a pro další použití je vhodné tyto váhy upravit. Pro úpravu vah se používá jednoduchého principu, váhy v rozmezí 0 a 1 se kreslí v odstínech šedi na model ve 3D pohledu. Správné váhy jsou základním kamenem při používání animací kůže, bez správně nakreslených vah se kůže nedeformuje přirozeně a kazí herní zážitek. Ukázka špatně deformované kůže, kůže s opravenými váhami a kreslení vah je vidět na obrázku 5.4.

Obrázek 5.4. Špatný skinning, dobrý skinning a kreslení vah.

Když jsou natočena data z motion capture a správně připravené modely, nic nebrání přenosu animací. Nejprve je zapotřebí importovat do Motion Builderu všechny c3d soubory týkající se daného 3D modelu. Každý importovaný c3d soubor vytvoří v Motion Builderu vlastní nahrávku, která odděluje vložená data od ostatních. Nahrávky je možné mezi sebou přepínat a pracovat na každé odděleně. Do právě importované nahrávky vložíme postavu actora, která je využívána při retargetingu. V náhledu scény je možné vidět actora stojícího v T-pose a optická data získaná z motion capture. Nyní je nutné actora co nejpřesněji vměstnat do optických dat, přičemž k jeho polohování je důležité používat pouze rotaci a škálování, ponusutí způsobuje nežádoucí efekty. Posunutí je možné bez problémů provádět na pánvi, která hýbe s celým actorem a tak můžeme upravovat jeho pozici vůči optickým datům. Po vměstnání actora do optických dat, se musí vytvořit marker set, který umožní spojit data z motion capture s actorem a rozhýbat ho. Marker set je pouze relace, která spojuje určitý marker s určitou částí těla. Do vytvořeného marker setu je třeba přiřadit optická data, Motion Builder podle dat poupraví rotace částí actorova těla a poté už je možné přehrát animaci nahranou pomocí motion capture namapovanou na postavu actora.

Nyní je možné do scény importovat vytvořený model. Pokud byla kostra modelu vytvořena pomocí Maya HumanIK, není žádné další nastavení třeba, pokud ne, je nutné Motion Builder seznámit s kostrou modelu, aby mohla být animace retargetována z actora na tento model. Nyní se importovanému modelu určí zdroj animací, podle kterého

[1]) `http://graphics.cs.cmu.edu/projects/sma/jamesTwigg_SMA.pdf`
[2]) `http://docs.unity3d.com/Manual/Preparingacharacterfromscratch.html`

se bude řídit, v tomto případě to bude actor, ale mohl by být použit i jiný již animovaný model. Když se importovaný model hýbe podle actora, dalším krokem je vypečení animace do control rigu, nebo kostry. Vypečení animace znamená, že se uloží poloha, zvětšení a rotace každého koncového efektoru, nebo kosti, v každém snímku animace. V tomto okamžiku je na importovaném modelu přenesena animace z motion capture. Tento postup opakujeme pro každou nahrávku, až vytvoříme sérii oddělených animací.

Po vypečení animací do control rigu, nebo kostry bohužel zpracování neskončilo, každý problém, který vznikl při zaznamenávání dat, jako například přerušení trajektorie, šum na trajektorii markeru a podobně, se nyní viditelně projeví. Tyto artefakty je nutné opravit ručně, pro úpravu animací je vhodné vytvořit novou animační vrstvu, která umožňuje upravovat animace bez zásahu do původních animačních dat. Při vytváření výsledné podoby animace je dobré změnit začátek a konec animace tak, aby neobsahovala nechtěné pohyby jako například T-pose na začátku a na konci animace. Pokud je animace určená k běhu ve smyčce, je nezbytné, aby počáteční a koncová póza modelu byla stejná, čehož můžeme dosáhnout zkopírováním počátečního snímku na koncový, nebo v Motion Builderu pomocí Pose tool.

Posledním krokem je sloučení všech animací do jedné nahrávky, protože herní engine Unity tímto způsobem pracuje s animacemi. Motion builder má více způsobů jak je možné sloučit nahrávky do jedné, ale nejsrozumitelnější je použití Story tool. Story tool umožňuje poskládat jednotlivé animace za sebe a poté je nahrát do jedné nahrávky. V případě potřeby je možné mezi jednotlivými nahrávkami vytvořit přechody, které interpolují data mezi koncem předchozí a začátkem následující animace. Když jsou všechny animace vloženy do jedné nahrávky, model se vyexportuje zpět do Autodesk Maya. Unity umožňuje importovat proprietární soubory Autodesk Maya.

5.3 Tvorba scény v Unity

Tato kapitola popisuje způsob jakým byly modely, animace, zvuk a umělá inteligence použity k vytvoření hratelné úrovně v Unity. Jelikož zadáním práce bylo využití animací pro tvorbu počítačové hry, nebudou všechny vytvořené struktury diskutovány a bude popsáno pouze nejnutnější minimum. Nyní je nutné vysvětlit následující pojmy, které jsou potřebné k pochopení této kapitoly.

GameObject je základní objekt, reprezentující jednotky, rekvizity a jiné modely.

Transform je uzel grafu scény.

NavMesh je struktura sloužící NavMeshAgentům, k pohybu a hledání cesty.

NavMeshAgent je komponenta herního objektu, která v kombinaci s NavMesh umožňuje jednotce navigovat se v herním prostoru.

Prefab je GameObject rozšířený o nové komponenty, funkčnost a přednastavené hodnoty.

Rigid body je reprezentace pevného tělesa, která umožňuje pracovat s objekty na základě fyzikálního modelu.

Collision collider je komponenta která umožňuje fyzikálnímu enginu vypočítat, zda do sebe 2 collidery narazily, nebo collider narazil do Rigid body.

Sphere collider je obalující koule, která slouží jako Collision collider.

Character controller je speciální Collision collider zjednodušující práci s jednotkami.

Script je komponenta, která umožňuje přistupovat k vnitřním funkcím enginu a rozšiřovat jeho funkčnost.

Animator je komponenta sloužící k řízení animací modelů.

Animator controller je stavový automat používaný k uspořádání a řízení animací.

Audio filter je komponenta umožňující pracovat se zvukem, v tomto případě jej vytvářet.

Audio listener je komponenta přijímající zvuk z prostředí. Slouží k reprodukci zvuku uživateli.

V Unity byl založen projekt a s ním se vytvořila prázdná scéna, ve které se bude hra odehrávat. Při zakládání projektu se vytvoří na disku složka, ve které je projekt uložen. V této složce Unity hledá použitelné zdroje. Do této složky byly nakopírovány 3D modely a textury grafického uživatelské rozhraní, vytvořené pro účely této hry. Nyní je možné s těmito modely a texturami pracovat v Unity.

Prvním krokem bylo vytvoření statické scény, z modelů které se nepohybují. Nejdříve byl umístěn terén a pomocná geometrie sloužící jako NavMesh. Poté byly do scény rozmístěny zbývající modely, stromy, teleporty, voda, domky a další. Po vytvoření statické scény, je dobré uspořádat objekty v hierarchii tak aby jejich předkem byl terén.

Dalším krokem k vytvoření hry je definování animací na modelech, vytvoření animator controllerů a implementace skriptů. Když Unity importuje 3D model s animacemi, není jasné, kde na časové ose animace začíná a kde končí. Proto je nutné v Unity na časové ose vymezit jednotlivé animace a pojmenovat je. Každá animace je určena počátečním a koncovým snímkem. Unity poskytuje možnost vytvořit animace ve smyčce, pokud tak již nebylo učiněno dříve. Další možností je posunutí počátku ve smyčce, animace začne ve snímku posunutém o definovaný počet snímků.

Po rozdělení animací byl vytvořen animator controller. Každá animace je přiřazena jednomu stavu ve stavovém automatu. V animator controlleru je jeden stav vždy počáteční. Každý stav se dá určit jako počáteční, animace v tomto stavu je bez vnějšího zásahu přehrávána jako první. Stavy v animator controlleru jsou spojeny přechody, pomocí kterých je možné dostat se z jednoho stavu do druhého, přejít od jedné animace k druhé. Přechody mezi stavy mají logické podmínky, pomocí kterých se určí, do jakého stavu se bude přecházet. Některé podmínky mohou být založeny pouze na koncovém času animace, ale s jejich pomocí by nebylo možné vybírat ve hře konkrétní animace. Proto je možné vytvářet v animator controlleru logické, celočíselné a desetinné proměnné. Na základě těchto proměnných je možné vytvářet podmínky, nebo kombinace podmínek, které jednoduše řídí chod animací. Na úrovni přechodů je možné definovat, jak hladce přejde jedna animace v druhou. Změna jedné animace v druhou je vždy lineární, záleží pouze na definici, odkud, kam se mají animace prolínat. Čili úpravou této změny animací je možné vytvořit velmi plynulý, pozvolný, ale také až skokový přechod. Při použití proměnných v animator controlleru je nutné tyto proměnné nastavovat ze scriptu přidruženého k animovanému game objectu. Nejjednodušší možný příklad použití je vidět v uvedeném kódu.

```
private Animator animator;

void start(){
\\Uloží referenci na Animator
animator = this.getComponent<Animator>();
}

void update(){
\\Nastaví proměnnou Animation v AnimatorControlleru na hodnotu dance
\\Po nastavení hodnoty postava začne tancovat
animator.SetInteger ("Animation",(int)CharacterAnimations.dance);
}
```

Script nastaví celočíselnou proměnnou Animation v animator controlleru na hodnotu dance. Jak je možné domyslet, po této akci by správně postava měla začít tančit. Podobným způsobem je možné zajistit ve scriptu, aby při každé akci byla spuštěna příslušná animace.

Nyní je nutné všechny probrané komponenty spojit dohromady v jeden celek, který vytvoří animovanou postavu pohybující se ve hře. Do scény byl umístěn 3D model s pojmenovanými a rozdělenými animacemi. Dále na tento model ve scéně, v této době již game object, přidáme několik komponent. NavMeshAgent pro navigaci po NavMeshi, Character controller a Rigid body sloužící ke zjištění kolizí, Animator controller pro řízení animací a jako poslední skript, který celou postavu řídí. Takto vytvořenou postavu je možné uložit jako prefab a dále ji používat při instancování. Výhodou vytvoření prefabu je, že všechny hodnoty a komponenty jsou již nastavené, přiřazené a není nutné to dělat za běhu hry. Pokud nebude žádoucí vytvářet nové instance této postavy, není přímo nutné vytvářet z ní prefab. Takto je možné pomocí několika málo komponent vytvořit funkční postavu.

Ve hře je však více game objectů než jen postavy, jejichž použití je velmi podobné. Většina game objectů, které jsou ve hře použity, potřebují nějaký typ Collision collideru a script, který obstarává veškerou logiku a funkčnost. Pomocí několika prefabrikátů je možné vytvořit i rozsáhlou hru.

Posledním bodem, který je dobré zmínit, se týká procedurálního generování hudby. Na kameře snímající scénu je umístěn audio listener, který naslouchá zvukům scény. Na tomto objektu je možné také zvuky přehrávat. Pokud je přiřazen game objectu s audio listenerem script, je možné generovat na tomto game objectu hudbu. Ve scriptu se vytvoří nový audio filter a uvnitř něj je možné generovat data, která se přehrají na zvukové kartě.

Kapitola 6
Závěr

6.1 Beta testing

Cílem beta testu[10] je odhalit chyby, které se nepodařilo odhalit při implementaci. Při beta testu se uvolní takzvaná beta verze uzavřené skupině lidí, kteří používáním programu testují jeho funkčnost. Aby bylo možné zjistit většinu chyb v tak rozsáhlém programu jako je počítačová hra, je zapotřebí velké množství participantů a proto je někdy beta verze uvolněna pro širokou veřejnost. Výhodou testování s lidmi je jejich kreativita, která otestuje i krajní situace, na které tvůrci programu třeba ani nepomýšleli.

Pro účely beta testu byl vytvořen jednoduchý formulář na školním Google Drive[1]). Dotazník obsahuje úvod, který má za účel seznámit participanta s problematikou, jelikož nebylo možné se s participanty setkat osobně. V dotazníku je nutné uvést jméno a příjmení participanta, sloužící pouze ke kontrole, kolikrát participant odpověděl. Pokud v průběhu testování zaznamenal participant chybu, vyplní hlášení o chybě. Poslední otázkou je, zda je hra podle participanta hratelná, či není. Výsledky testu jsou umístěny v následující tabulce[2]). Testování se zúčastnilo celkem 12 participantů, většina pocházela z hráčské komunity, která má zkušenosti s podobným herním žánrem, někteří participanti neměli žádné zkušenosti s počítačovými hrami.

Odpovědi participantů byly různorodé a většinou neodpovídaly požadavkům beta testu, některé podněty z těchto odpovědí byly zahrnuty do diskuse o dalším postupu práce viz. 6.3. Během testování bylo zjištěno několik chyb, jejich popis a postup při odstranění chyby je popsán dále v této kapitole.

Chyby spojené s herními pravidly. Jeden z participantů zjistil zajímavou chybu. Pokud hra běží po nějakou dobu a poté spustí hru od začátku, hra vygeneruje velké množství jednotek. Chyba byla analyzována a odstraněna, ve výsledném buildu se nevyskytuje.

Další chyba, která se během testování vyskytla, způsobovala ztrátu životů, many i zkušeností. Chyba se ve hře objevovala po několikerém spuštění od začátku a uplynutí určité časové prodlevy. Chyba se objevila pouze v 64 bitové buildu pro Windows, 32 bitový build tuto chybu neobsahuje. Vzhledem k tomuto zjištění byl nadále využíván pouze 32 bitový build. Tuto chybu se nepodařilo analyzovat, tudíž ani opravit.

Některá kouzla dávají moc velké poškození. Zprvu se toto nezdálo jako chyba, po delší analýze bylo zjištěno, že kouzla s efektem trvajícím po určitý čas měla špatný výpočet

[1]) https://docs.google.com/a/fel.cvut.cz/forms/d/1CVxYZ7KCFlQBCi4ntILzPlQKBoqHFNOsqdPi35E4Q2Q/viewform

[2]) https://docs.google.com/a/fel.cvut.cz/spreadsheets/d/1Z2kKyhk_hQwCe6cbQdEn-4Cvm4tP2SAMSOk7V2NYjZo/edit?usp=sharing

průběžně aplikované hodnoty. Ve výsledku tato kouzla způsobovala větší poškození, nebo léčila více životů. Tato chyba byla odstraněna a výsledném buildu se nevyskytuje.

Poslední objevená chyba porušující herní pravidla se týkala cílených kouzel. V okamžiku kdy by měla narazit a způsobit poškození, se začla točit okolo cíle a v některých případech způsobila poškození, a v některých ne. Tuto chybu se nepodařilo replikovat a její odstranění proto nebylo možné.

Beta testing potvrdil, že je hra hratelná ve smyslu funkčnosti herního mechanismu a ovládání, také že je možné herní level dohrát až do konce. 11 z 12 participantů odpovědělo, že je hra hratelná. V jednom případě byla hra na pomezí hratelnosti, během tohoto testování byla objevena chyba v 64 bitové verzi. Žádný participant neodpověděl, že je hra nehratelná.

6.2 Zhodnocení splnění cílů

Požadavky plynoucí ze zadání byly splněny až na druhý rozhodovací mechanismus. Vzhledem k rozsahu práce a z časových důvodů zůstal pouze návrhem. Výsledek této práce je hratelná počítačová hra s množstvím 3D modelů a animací vytvořených pro tento účel. Obrázky 6.1 a D ukazují hru, která v průběhu této práce vznikla. Práce popisuje postup, jakým byla hra vytvořena, materiály ze kterých je dobré čerpat a potřebné znalosti pro úplný základ v odvětví počítačových her. Ve hře je implementovaný jednoduchý hudební generátor, hudba, kterou generuje, není nikterak líbivá, avšak splňuje základní pravidla hudební nauky.

Tvorba počítačových her a animací je náročná záležitost a bylo velmi poučné projít si osobně celým procesem jejího zrodu. V průběhu práce bylo nutné projít množství nejrůznějších grafických programů. Modelář Autodesk Maya a Mudbox pro tvorbu modelů. Adobe Photoshop pro tvorbu textur modelů a grafického uživatelského rozhraní. Při záznamu pohybů byl použit Optitrack motive a pro čištění a retargeting animací na modely byl použit Autodesk Motion builder. A v neposlední řadě Unity, ve kterém celá hra vznikla. Je až překvapující kolik různých odvětví lidské činnosti se snoubí při vytváření počítačových her a kolik práce stojí za každým prodaným titulem.

6.3 Diskuse dalšího postupu práce

V průběhu návrhu herních mechanismů a architektury byl kladen velký důraz na pozdější rozšiřitelnost a proto je možné do hry jednoduše doipmlementovat novou funkčnost využívající stávajícího systému. Nové typy kouzel, kombinace kouzel a rozšíření počtu efektů. Přidání rozdílných zbraní a předmětů a možnost prodávat staré předměty v obchodě. Dále by bylo zajímavé přidat speciální útoky s vizuálními efekty, jízda na koni, složitější a dynamičtější nepřátelé, další herní úrovně a tak dále.

Z odpovědí participantů bylo zřejmé, že by bylo vhodné rozšířit funkčnost uživatelského rozhraní. V první řadě šlo o interaktivitu minimapy a její vyjadřovací schopnost. Bylo požadováno, aby kliknutí do minimapy zobrazilo příslušnou část herního světa. Minimapa by graficky měla rozlišovat mezi hlavním hrdinou, věží a malými jednotkami. Poslední požadavky týkající se minimapy, požadovaly zobrazení obdélníku, který by na minimapě reprezentoval právě prohlíženou část herního světa. Dalším postupem by

Obrázek 6.1. Obrázky ze hry.

mohlo být přidání zvuků jednotlivým akcím nebo možnost personalizace a sdílení herních výsledků. V neposlední řadě by bylo možné nahradit umělou inteligenci a vytvořit multiplayerovou verzi.

Literatura

[1] "Jiří Žára, Petr Felkel, Jiří Sochor, Bedřich Beneš". *"Moderní počítačová grafika"*. In: "Computer Press", 2004 . "14".

[2] "Jiří Žára, Petr Felkel, Jiří Sochor, Bedřich Beneš". *"Moderní počítačová grafika"*. In: "Computer press", 2004 . "6".

[3] *"Normal map"*. "
http://wiki.polycount.com/wiki/Normal_map". 2015 .

[4] "Jiří Žára, Petr Felkel, Jiří Sochor, Bedřich Beneš". *"Moderní počítačová grafika"*. In: "Computer press", 2004 . "5.4".

[5] *"Multi player online battle arena - action realtime strategy"*. 2015.
http://en.wikipedia.org/wiki/Multiplayer_online_battle_arena.

[6] "Mike McShaffry, David Gragam". *"Game coding complete"*. "Course technology", 2012 .

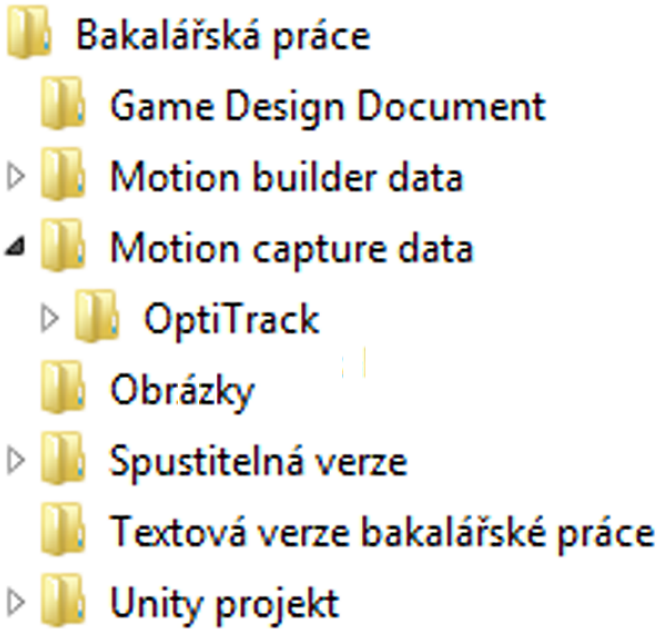
[7] "Chris Simpson". *"Behavior trees for AI: How they work"*. 2014.
http://www.gamasutra.com/blogs/ChrisSimpson/20140717/221339/Behavior_trees_for_AI_How_they_work.php.

[8] "Tomáš Plch, Matěj Marko, Petr Ondráček, Martin Černý, Jakub Gemrot, Cyril Brom". *"Modular Behavior Trees: Language for Fast AI in Open-World Video Games"*. 2014.
http://popelka.ms.mff.cuni.cz/~cerny/papers/MBT_ECAI_2014.pdf.

[9] *"Neuronová síť"*.
http://cs.wikipedia.org/wiki/Neuronov%C3%A1_s%C3%AD%C5%A5.

[10] *"Software testing"*. 2015.
http://en.wikipedia.org/wiki/Software_testing.

Příloha A
Obsah přiloženého DVD



Obrázek A.1. Obsah přiloženého DVD.

Příloha B
3D modely

Seznam obrázků	počet trojúhelníků	odkaz
Medvěd	1734	B.2
Bonus	300	B.3
Drak	12020	B.4
Plot	184	B.5
Vlajka	440	B.6
Hlavní hrdina	4552	B.7
Dům	702	B.8
Velký dům	670	B.9
Malá jednotka	1686	B.10
Houba	297	B.11
Jehličnatý strom	288	B.12
Kapradí	300	B.13
Terén	5144	B.14
Věž	1184	B.15
Listnatý strom	1062	B.16

Tabulka B.1. Seznam obrázků 3D modelů.

Obrázek B.2. Medvěd, ve hře reprezentuje věž.

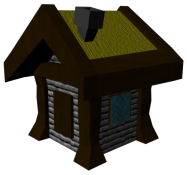
Obrázek B.3. Sebratelný bonus.

Obrázek B.4. Drak, ve hře reprezentuje hlavní budovu.

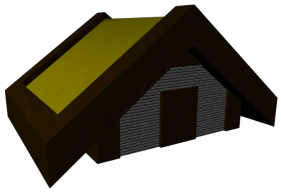
Obrázek B.5. Plot.

Obrázek B.6. Vlajka týmu.

Obrázek B.7. Hlavní hrdina.



Obrázek B.8. Dům.



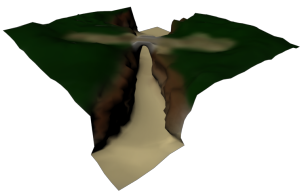
Obrázek B.9. Velký dům.

Obrázek B.10. Malá jednotka.

Obrázek B.11. Dekorativní houba.

Obrázek B.12. Jehličnatý strom.

Obrázek B.13. Kapradí.



Obrázek B.14. Terén herního prostoru.

Obrázek B.15. Pouze okrasná věž.

Obrázek B.16. Listnatý strom.

Příloha C
Tabulky

LVL	HP	MN	AR	SR	HR	MR	XP	AD	SP	LXP	G
1	550	226	10	5	1.8	1.9	300	40	40	1204	250
2	554	230	11	6	1.8	1.9	702	42	41	4294	250
3	563	240	12	7	1.9	2	1112	45	42	9632	250
4	582	257	12	8	1.9	2.1	1508	48	43	17474	250
5	612	280	12	9	2	2.3	1880	52	45	28013	250
6	658	313	13	10	2.2	2.6	2232	56	49	41409	250
7	721	354	13	12	2.4	3	2560	61	53	57797	250
8	806	406	13	14	2.7	3.4	2866	67	57	77293	250
9	914	468	14	16	3	3.9	3152	74	61	100000	250
10	1050	541	14	18	3.5	4.5	3422	83	65	126008	250
11	1215	626	14	20	4.1	5.2	3674	94	70	155402	250
12	1414	723	15	25	4.7	6	3910	114	75	188256	250
13	1648	834	15	30	5.5	7	4134	130	81	224641	250
14	1922	958	15	35	6.4	8	4344	148	87	264620	250
15	2237	1096	16	40	7.5	9.1	4544	168	94	308254	250
16	2598	1249	20	50	8.7	10.4	0	192	100	inf	250

Tabulka C.2. Tabulka hodnot hlavní postavy.

LVL	HP	MN	AR	SR	HR	MR	XP	AD	SP	LXP	G
1	135	0	5	2.5	0.5	0	150	34	0	0	25
2	135	0	5.5	3	0.5	0	351	34	0	0	25
3	137	0	6	3.5	0.5	0	556	35	0	0	25
4	140	0	6	4	0.5	0	754	35	0	0	25
5	145	0	6	4.5	0.5	0	940	35	0	0	25
6	152	0	6.5	5	0.5	0	1116	36	0	0	25
7	162	0	6.5	6	0.5	0	1280	37	0	0	25
8	174	0	6.5	7	0.5	0	1433	39	0	0	25
9	189	0	7	8	0.6	0	1576	42	0	0	25
10	206	0	7	9	0.6	0	1711	46	0	0	25
11	225	0	7	10	0.7	0	1837	50	0	0	25
12	250	0	7.5	12.5	0.8	0	1955	55	0	0	25
13	275	0	7.5	15	0.9	0	2067	61	0	0	25
14	301	0	7.5	17.5	1	0	2172	68	0	0	25
15	327	0	8	20	1.1	0	2272	75	0	0	25
16	356	0	10	25	1.2	0	2300	84	0	0	25

Tabulka C.3. Tabulka hodnot malé jednotky útočící zblízka.

LVL	HP	MN	AR	SR	HR	MR	XP	AD	SP	LXP	G
1	135	0	5	2.5	0.5	0	150	34	0	0	25
2	135	0	5.5	3	0.5	0	351	34	0	0	25
3	137	0	6	3.5	0.5	0	556	35	0	0	25
4	140	0	6	4	0.5	0	754	35	0	0	25
5	145	0	6	4.5	0.5	0	940	35	0	0	25
6	152	0	6.5	5	0.5	0	1116	36	0	0	25
7	162	0	6.5	6	0.5	0	1280	37	0	0	25
8	174	0	6.5	7	0.5	0	1433	39	0	0	25
9	189	0	7	8	0.6	0	1576	42	0	0	25
10	206	0	7	9	0.6	0	1711	46	0	0	25
11	225	0	7	10	0.7	0	1837	50	0	0	25
12	250	0	7.5	12.5	0.8	0	1955	55	0	0	25
13	275	0	7.5	15	0.9	0	2067	61	0	0	25
14	301	0	7.5	17.5	1	0	2172	68	0	0	25
15	327	0	8	20	1.1	0	2272	75	0	0	25
16	356	0	10	25	1.2	0	2300	84	0	0	25

Tabulka C.4. Tabulka hodnot malé jednotky útočící na dálku.

LVL	HP	MN	AR	SR	HR	MR	XP	AD	SP	LXP	G
1	10000	0	20	50	0	0	5000	250	0	0	250
2	10000	0	20	50	0	0	5000	250	0	0	250
3	10000	0	20	50	0	0	5000	250	0	0	250
4	10000	0	20	50	0	0	5000	250	0	0	250
5	10000	0	20	50	0	0	5000	250	0	0	250
6	10000	0	20	50	0	0	5000	250	0	0	250
7	10000	0	20	50	0	0	5000	250	0	0	250
8	10000	0	20	50	0	0	5000	250	0	0	250
9	10000	0	20	50	0	0	5000	250	0	0	250
10	10000	0	20	50	0	0	5000	250	0	0	250
11	10000	0	20	50	0	0	5000	250	0	0	250
12	10000	0	20	50	0	0	5000	250	0	0	250
13	10000	0	20	50	0	0	5000	250	0	0	250
14	10000	0	20	50	0	0	5000	250	0	0	250
15	10000	0	20	50	0	0	5000	250	0	0	250
16	10000	0	20	50	0	0	5000	250	0	0	250

Tabulka C.5. Tabulka hodnot věže.

LVL	HP	MN	AR	SR	HR	MR	XP	AD	SP	LXP	G
1	15000	0	20	50	10	0	0	300	0	0	0
2	15000	0	20	50	10	0	0	300	0	0	0
3	15000	0	20	50	10	0	0	300	0	0	0
4	15000	0	20	50	10	0	0	300	0	0	0
5	15000	0	20	50	10	0	0	300	0	0	0
6	15000	0	20	50	10	0	0	300	0	0	0
7	15000	0	20	50	10	0	0	300	0	0	0
8	15000	0	20	50	10	0	0	300	0	0	0
9	15000	0	20	50	10	0	0	300	0	0	0
10	15000	0	20	50	10	0	0	300	0	0	0
11	15000	0	20	50	10	0	0	300	0	0	0
12	15000	0	20	50	10	0	0	300	0	0	0
13	15000	0	20	50	10	0	0	300	0	0	0
14	15000	0	20	50	10	0	0	300	0	0	0
15	15000	0	20	50	10	0	0	300	0	0	0
16	15000	0	20	50	10	0	0	300	0	0	0

Tabulka C.6. Tabulka hodnot hlavní budovy.

Předmět	UC	AD	SP	SR	AR	MS	HP	MN	HR	MR	C
Meč	4	25	0	0	0	0	0	0	0	0	300
K. meč	4	0	25	0	0	0	0	100	0	0	300
Boty	2	0	0	0	0	5	0	0	0	0	250
Č. prsten	1	0	5	0	0	0	50	100	1.5	1.5	200
M. prsten	1	5	0	0	0	0	100	50	1.5	1.5	250
Brnění	6	25	0	0	0	0	150	150	2.5	2.5	500

Tabulka C.7. Tabulka hodnot předmětů.

Drahokam	AD	SP	SR	AR	SP	HP	MN	HR	MR	C
Červený	15	0	0	0	0	0	0	0	250	
Černý	0	0	0	15	0	100	0	1	0	250
Žlutý	10	10	0	0	0	0	0	2	2	250
Čirý	0	0	20	0	0	0	150	0	1.5	250
Duhový	5	5	5	5	1	50	50	2.5	2.5	500
Modrý	0	15	0	15	0	0	75	0	0	250
Zelený	0	0	5	5	0	100	100	1	1	250

Tabulka C.8. Tabulka hodnot vylepšení.

Efekt LVL	P MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	60	500	75	60	3	6	18	5	75
2	70	1000	80	125	4	25	16	5	90
3	80	1500	85	225	5	67.5	14	5	110
4	90	2000	90	350	6	140	12	5	135

Tabulka C.9. Tabulka hodnot cíleného otráveného kouzla.

Efekt LVL	P MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	1500	100	50	3	4	25	3	50
2	0	2000	105	100	4	10	20	3.33	75
3	0	2500	110	175	5	22	15	3.66	90
4	0	3000	115	250	6	38	10	4	12

Tabulka C.10. Tabulka hodnot plošného otráveného kouzla.

Efekt LVL	P MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	69	600	160	75	3	11	15	5	25
2	70	950	165	150	4	45	10	5	50
3	71	1450	170	275	5	124	5	3.66	75
4	72	2000	175	450	6	270	5	5	100

Tabulka C.11. Tabulka hodnot mířeného otráveného kouzla.

Efekt LVL	F MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	70	1000	85	50	1	3	20	5	40
2	80	2000	90	125	1.25	3.25	15	5	65
3	90	3000	95	225	1.5	3.5	11	5	85
4	100	4000	100	275	2.0	4	7.5	5	110

Tabulka C.12. Tabulka hodnot cíleného ohnivého kouzla.

Efekt LVL	F MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	1500	100	30	1	2	25	5	60
2	0	2000	105	100	1.5	2.5	20	5	75
3	0	2500	110	175	2	3	15	5	90
4	0	3000	115	250	2.5	4	10	5	120

Tabulka C.13. Tabulka hodnot plošného ohnivéo kouzla.

Efekt LVL	F MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	75	600	160	45	1	5	15	5	35
2	76	950	165	125	1.5	7	10	5	50
3	77	1450	170	225	2	9	7.5	5	75
4	78	2000	175	350	2.5	11	5	5	100

Tabulka C.14. Tabulka hodnot mířeného ohnivého kouzla.

Efekt LVL	I MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	100	1000	75	50	1	10	60	5	90
2	110	2000	80	175	1.25	31	50	5	125
3	120	3000	85	350	1.5	51	40	5	160
4	140	4000	90	700	2	88	25	5	195

Tabulka C.15. Tabulka hodnot cíleného mrazivého kouzla.

Efekt LVL	I MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	1500	100	50	1	3	100	2	100
2	0	2000	105	150	1.5	11	75	2.25	150
3	0	2500	110	250	2	25	50	2.5	200
4	0	3000	115	600	2.5	75	40	2.75	250

Tabulka C.16. Tabulka hodnot plošného mrazivého kouzla.

Efekt LVL	I MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	70	500	160	100	0.5	5	50	5	75
2	71	750	165	225	0.75	23	40	5	90
3	72	1000	170	550	1	83	30	5	110
4	73	1250	175	900	1.25	180	18	5	135

Tabulka C.17. Tabulka hodnot mířeného mrazivého kouzla.

Efekt LVL	O MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	70	1000	75	50	0.75	100	58	3	70
2	80	2000	80	115	1	100	42	4	100
3	90	3000	85	235	1.25	100	31	5	120
4	100	4000	90	400	1.5	100	22.5	6	135

Tabulka C.18. Tabulka hodnot cíleného omračujícího kouzla.

Efekt LVL	O MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	2000	100	30	0.5	100	80	2	80
2	0	2500	105	105	0.75	100	60	3	110
3	0	3000	110	215	1	100	45	4	135
4	0	3500	115	375	1.25	100	30	5	170

Tabulka C.19. Tabulka hodnot plošného omračujícího kouzla.

Efekt LVL	O MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	75	750	150	60	1.0	100	36	5	65
2	76	1000	155	125	1.25	100	24	5	85
3	77	1250	160	250	1.5	100	18	5	100
4	78	1500	165	425	2	100	15	5	115

Tabulka C.20. Tabulka hodnot mířeného omračujícího kouzla.

Efekt LVL	L MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	500	0	45	0	0	36	0	75
2	0	750	0	175	0	0	24	0	90
3	0	1000	0	400	0	0	18	0	130
4	0	1250	0	850	0	0	15	0	150

Tabulka C.21. Tabulka hodnot léčícího kouzla.

Efekt LVL	L MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	750	0	45	2	25	40	0	65
2	0	1000	0	150	2	100	30	0	85
3	0	1250	0	350	2	200	20	0	100
4	0	1500	0	600	2	300	12	0	115

Tabulka C.22. Tabulka hodnot regeneračního kouzla.

Efekt LVL	Š MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	500	0	0	2	10	30	0	60
2	0	750	0	0	3	20	24	0	80
3	0	1000	0	0	4	30	18	0	100
4	0	1250	0	0	5	50	14	0	120

Tabulka C.23. Tabulka hodnot kouzla štít.

Efekt LVL	Ú MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	400	0	0	5	10	20	0	50
2	0	600	0	0	6	20	15	0	60
3	0	800	0	0	7	30	12	0	70
4	0	1000	0	0	8	50	8	0	90

Tabulka C.24. Tabulka hodnot kouzla válečný řev.

Efekt LVL	Z MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	250	0	0	3	4	20	5	40
2	0	500	0	0	4	6	15	5	55
3	0	750	0	0	5	8	10	5	70
4	0	1000	0	0	6	10	5	5	85

Tabulka C.25. Tabulka hodnot kouzla zrychlení.

Efekt LVL	B MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	250	30	0	3	100	20	0	80
2	0	500	30	0	4	250	15	0	95
3	0	750	30	0	5	400	10	0	105
4	0	1000	30	0	6	800	5	0	120

Tabulka C.26. Tabulka hodnot útoku krvácení.

Efekt LVL	K MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	250	30	75	1	5	20	0	50
2	0	500	30	200	1	5	15	0	65
3	0	750	30	375	1	5	10	0	80
4	0	1000	30	650	1	5	5	0	95

Tabulka C.27. Tabulka hodnot útoku kopu.

Efekt LVL	C MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	250	30	50	3	50	20	0	60
2	0	500	30	175	4	60	15	0	75
3	0	750	30	350	5	75	10	0	90
4	0	1000	30	700	6	100	5	0	105

Tabulka C.28. Tabulka hodnot kombinačního útoku.

Efekt LVL	T MS	C	RNG	DMG	EDUR	EV	CD	DUR	MN
1	0	250	30	0	0	0	50	0	40
2	0	500	30	0	0	0	60	0	55
3	0	750	30	0	0	0	70	0	70
4	0	1000	30	0	0	0	100	0	85

Tabulka C.29. Tabulka hodnot teleportačního kouzla.

Příloha D
Obrázky ze hry

Obrázek D.17. Obrázky ze hry.