STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Dungeoneer – Videohra

Tomáš Vik

Pardubický Kraj **Pardubice 2024**

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Dungeoneer – Videohra

Dungeoneer – Videogame

Autoři: Tomáš Vik
Škola: DELTA - Střední škola informatiky a ekonomie, s.r.o.
Kraj: Pardubický Kraj
Konzultant: Jan Mottl

Pardubice

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 03.25.2024 Tomáš Vik

Poděkování

Rád bych poděkoval Mgr. Janu Mottlovi za celkovou pomoci s projektem. Lukášovi Adámkovi za inspiraci a spolupráci a Štěpánovi Pozděnovi, který našel spoustou bugů při hraní.

Anotace

Ve své práci SOČ jsem se zabýval vývojem hry jménem „Dungeoneer“, která spadá do žánru dungeon-crawler a bere si inspiraci z her jako Minecraft Dungeons, Hypixel Skyblock a Shakes & Fidget. V práci jsem se zabýval nejen návrhem hry jako takové, ale i celkovým provedením. Tento projekt mě naučil spoustu věcí a pomohl mi rozšířit si zkušenosti. Ve hře byla implementována procedurální generace dungeonů (žalářů). Dále byl kladen důraz na tvorbu algoritmem řízených nepřátel, přátelských nehratelných postav. V projektu pracuji s komplexní sadou hráčských schopností a jednoduchým systém ukládání a načítání dat. Hra je ve testovací fázi a hratelném stavu beta verze.

Klíčová slova

Hra, Unity, Video hra, Dungeon, Procedurální generace, Pixel art, RPG, Dungeon-Crawler, 2D

Annotation

In my SOČ work, I developed a game called "Dungeoneer", which falls into the dungeon-crawler genre and takes inspiration from games such as Minecraft Dungeons, Hypixel Skyblock and Shakes & Fidget. In my work, I dealt not only with the design of the game, but also with the execution itself. This project taught me a lot and helped me expand my experience. Procedural dungeon generation, enemies, friendly non-playable characters, player abilities, and save and load data were implemented in the game. The game is in beta testing and playable state.

Keywords

Hra, Unity, Video game, Dungeon, Procedural generation, Pixel art, RPG, Dungeon-Crawler, 2D

1 OBSAH

Obsah 5
1 Úvod 7
2 Teoretická část 8
2.1 Pojmy 8
2.1.1 Herní engine 8
2.1.2 JSON 8
2.1.3 Procedurální generace ve videohrách 8
2.1.4 XOR šifra 8
2.1.5 Prefab 8
2.1.6 Instance 8
2.1.7 Rigidbody2D 9
2.1.8 Dungeon 9
2.1.9 Enum 9
2.2 Použité technologie 10
2.2.1 Aseprite 10
2.2.2 Unity Engine 10
2.2.3 JetBrains Rider 10
2.2.4 C# 10
2.3 Herní design 11
3 Praktická část 12
3.1 Mechaniky hráče 12
3.1.1 Pohyb hráče 12
3.1.2 Útočení hráče 12
3.1.3 Výpočty útočení 13
3.1.4 Statistiky hráčské postavy 14
3.1.5 Inventář Hráče 14
3.1.6 Diagram Hráče 15
3.2 Uživatelské rozhraní 16
3.2.1 Inventář 16
3.2.2 Přátelské nehratelné postavy 16
3.2.3 Překryvný panel 16
3.3 Přátelské nehratelné postavy 18
3.3.1 Prodavač 18

3.3.2 Vylepšovač 18
3.3.3 Úkolovník 18
3.4 Nepřátelé 20
3.4.1 Nepřítel na blízko 20
3.4.2 Nepřítel na dálku 21
3.4.3 Boss 21
3.5 Procedurální generace dungeonu 22
3.5.1 Příprava místonstí 22
3.5.2 Generace cest 23
3.5.3 Vstup do místnosti 24
3.5.4 Porovnání různých provedení 24
3.6 Use case diagram hry 26
4 Závěr 27
5 Použitá literatura 28
6 Zakoupené Assety 29
7 Seznam obrázků a tabulek 30

2 ÚVOD

Video hry mě vždy zaujímaly obzvlášť hry s procedurální generací a širokým výběrem výbavy. Kombinace těchto dvou prvků mě přivedla k vytvoření hry Dungeoneer. Dungeoneer tedy převzal oba tyto principy. Poskytnout hráčům nezapomenutelný zážitek z prozkoumávání temných a nebezpečných procedurálně generovaných dungeonů a hráčskou přizpůsobitelnost herní stylům. Tato hra slouží jako beta verze pro budoucí plnou verzi, avšak i v této rané fázi přináší hodiny zábavy a napětí.

Inspirací pro Dungeoneer byla populární minihra Skyblock na Minecraft serveru Hypixel, která představuje široké spektrum aktivit pro hráče, avšak jejím nezapomenutelným prvkem je náhodně generovaný dungeon plný silných nepřátel a nebezpečí. Tato konceptuální základna poskytla důležitý impulz pro vytvoření Dungeoneer, který se zaměřuje na podobný zážitek, avšak v 2D prostředí.

Cílem hry Dungeoneer je nejen přinést hráčům adrenalinové dobrodružství plné nástrah a výzev, ale také demonstrovat moje praktické znalosti vývoje her a schopnost přenést tyto dovednosti do tvorby zábavného a poutavého herního prostředí. Každý aspekt hry byl navržen s ohledem na poskytnutí hráčům příjemného a vyváženého zážitku.

Tato dokumentace poskytuje detailní pohled na fungování hry, včetně herních mechanik, prostředí, nepřátel a dalších klíčových prvků. Doufám, že Dungeoneer přinese hráčům nejen spoustu zábavy, ale také příležitost ponořit se do fascinujícího světa dungeonů, plného nebezpečí a tajemství. Připravte se na dobrodružství, které vás uchvátí a nepustí až do samotného konce!

3 TEORETICKÁ ČÁST

3.1 Pojmy

3.1.1 Herní engine

Herní engine je softwarový framework, který soustřeďuje obecné funkce používané v počítačových hrách, díky čemuž dovoluje zrychlit a zlevnit vývoj her. [1]

3.1.2 JSON

JavaScript Object Notation (JavaScriptový objektový zápis, JSON) je způsob zápisu dat (datový formát) nezávislý na počítačové platformě, určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Vstupem je libovolná datová struktura (číslo, řetězec, boolean, objekt nebo z nich složené pole), výstupem je vždy řetězec. Složitost hierarchie vstupní proměnné není teoreticky nijak omezena. [2]

3.1.3 Procedurální generace ve videohrách

Procedurální generace je proces vytváření herního obsahu pomocí algoritmů. Generátor obsahu může být řízen omezenou hodnotou nebo nepřímým uživatelským vstupem. Generování může být randomizováno poskytnutím náhodné počáteční hodnoty, aby se dosáhly více nepředvídatelné výsledky. Procedurální generace si našla své místo v mnoha oblastech. Hry, které implementují procedurální generací mohou vytvořit jedinečné, zajímavé a téměř nekonečné světy a také může ušetřit čas, pokud se to udělá správně. [3]

3.1.4 XOR šifra

Exkluzivní disjunkce je logická operace, jejíž hodnota je pravda, právě když každá vstupní hodnota nabývá, v porovnání s ostatními vstupy, unikátní hodnotu. **Error! Reference source not found.**

3.1.5 Prefab

Systém Prefab společnosti Unity vám umožňuje vytvářet, konfigurovat a ukládat herní objekt(Game Object) kompletní se všemi jeho komponentami, hodnotami vlastností a podřízenými objekty herní objekt(Game Objects) jako opakovaně použitelné aktivum. Prefab Asset funguje jako šablona, ze které můžete vytvářet nové prefabrikované instance ve Scéně. [5]

3.1.6 Instance

Instance je v informatice samostatná existence jakéhokoli objektu – objektu v objektově orientovaných jazycích, okna nebo jiného prvku grafického uživatelského rozhraní, procesu jako instance běžícího programu, virtuálního stroje, apod. [6]

3.1.7 Rigidbody2D

Přidáním komponenty Rigidbody2D ke objektu se dostane pod kontrolu fyzikálního enginu. To samo o sobě znamená, že na objekt bude působit gravitace a lze jej ovládat ze skriptů pomocí sil. Toto chování pochází výhradně z fyzikálního systému Unity; k získání působivého a autentického fyzického chování je zapotřebí velmi málo kódu. [7]

3.1.8 Dungeon

Hladomorna se dá charakterizovat jako hluboká jáma sloužící k potrestání provinilců, kde dotyčný pomalu umíral hladem a žízní. [8]

Ve video hrách se ale slovo dungeon používá k popsání obrovských podzemních místností a cest, které je propojují. Tyto dungeony jsou většinou schované pod nějakou starou budovou jako třeba zámkem nebo kostelem. Hodně často se tu také objevují nepřátelé jako kostlivci, zombie a pavouci.

3.1.9 Enum

Výčtový typ (v angličtině označovaný jako „enumerated type“ nebo zkráceně „enum“) patří mezi návrhové vzory. Je to datový typ tvořený konečnou omezenou množinou pojmenovaných hodnot. Každý člen výčtového typu je tvořen identifikátorem a hodnotou. Hodnotou zpravidla bývá celočíselná konstanta. [9]

3.2 Použité technologie

3.2.1 Aseprite

Asperite je software pro tvorbu 2D pixel artu vyvinutý studiem „Igara Studio S.A.“. Patří mezi jedny z nejvíce využívaných programů pro vytváření pixel artu, spritů a dlaždic, nabízející spoustu nástrojů umělcům, vývojářům her a nadšencům. Díky intuitivnímu rozhraní a silné kolekci funkcí usnadňuje Asperite proces tvorby a umožňuje uživatelům snadno realizovat své umělecké vize. [10]

3.2.2 Unity Engine

Unity je moderní a výkonný herní engine vyvinutý společností Unity Technologies. Patří mezi nejpopulárnější platformy pro vývoj her a interaktivních 3D a 2D her. Díky své široké podpoře pro různé platformy, včetně PC, mobilních zařízení a herních konzolí, a snadnému použití je Unity volbou mnoha vývojářů po celém světě. [11]

3.2.3 JetBrains Rider

Rider je moderní a výkonné integrované vývojové prostředí (IDE) vyvinuté společností JetBrains. Patří mezi nejvíce využívané nástroje pro vývoj softwaru v jazyce C#, ale také nabízí podporu pro další jazyky, jako je například Java. Rider je navržen s důrazem na produktivitu a komfort vývojářů, poskytující širokou škálu funkcí a nástrojů pro rychlé a efektivní programování. [12]

3.2.4 C#

C# (vyslovováno "See Sharp") je moderní, objektově orientovaný a typově bezpečný programovací jazyk. Umí objektově orientované a komponentově orientované programování a umožňuje vývojářům vytvářet mnoho typů bezpečných a robustních aplikací, které běží v .NET. C# má své kořeny v rodině jazyků C a programátoři v jazycích C, C++, Java a JavaScript jej budou okamžitě znát. [13]

3.3 Herní design

Herní design je proces tvorby her, který zahrnuje vytváření pravidel, herních mechanik, herních prostředí, uměleckého stylu a dalších prvků, které tvoří celkový zážitek z hraní hry. Klíčové prvky při tvorbě herního designu jsou motivace hráče k dokončení hry, získání nových dovedností, nebo k zábavě. Hratelnost se týká pravidel a interakce mezi hráčem a hrou. Hry RPG se často soustředí na vývoj postav a příběhové linie, zatímco hry akčního žánru se zaměřují na bojový systém a pohyblivost postav. Gameplay-loop (herní smyčka) by měl hráče udržovat zapojeného tím, že poskytne něco, co mohou dělat stále dokola. Občas může nastat u hráče stav – ponoření se do hry, kdy hráč není ochoten hru pozastavit nebo vypnout. Celkově lze říci, že herní design je klíčovým prvkem při tvorbě her a ovlivňuje jak úspěch, tak i kvalitu hry. [15]

4 PRAKTICKÁ ČÁST

4.1 Mechaniky hráče

V této kapitole budu vysvětlovat základní mechaniky hráčské postavy. Tyto mechaniky jsou základy každé hry a je velice důležité, aby hráč porozuměl jejich fungování a uměl je efektivně využívat k dosažení svých cílů.

4.1.1 Pohyb hráče

Hráčský pohyb je zajištěn pomocí stisknutí kláves WSAD, které jsou typicky využívané i v jiných podobných hrách. Při stisknutí klávesy, Input funkce zachytí stisk, roznásobí rychlostí a pomocí Rigidbody2D přidá hráči pohyb ve směru inputu hráče. Hráčova rychlost se může měnit záležíc na jeho rychlostní statistice.

Hráč má také možnost použít kotoul pomocí tlačítka shift. Kotoul hráče vymrští v směr, ve kterém se v momentu pohyboval a ubere mu to konkrétní přednastavené množství z výdrže.

Jiný způsob pohybu je také Unity Ovládač Postav (Character Controller) který umožňuje jednoduchý systém pro ovládání hráčské postavy v Unity, nebo i simulovat vlastní posun použitím matematických výpočtů a postavu posouvat. Ale z osobních zkušeností vím že ovládač postav může produkovat zasekaný a nepředvídatelný. Použití vlastních výpočtů také není nejefektivnější, protože musíme počítat s tím, že postava musí mít kolize. Vlastní posun by mohl v některých situacích kolize úplně ignorovat.

4.1.2 Útočení hráče

Hráčský útok je napevno nastaveno na levé tlačítko myši, přičemž zbraně nabízejí tři různé typy útoků: okamžitý útok na blízko, nabíjecí útok na blízko a okamžitý útok na dálku. Když hráč spustí útok, vytvoří se instance zbraně, která je přiřazena s konkrétní animací podle aktuálního komba hráče. Kombo hráče určuje, jaká animace z listu možných animací se má přehrát. V rámci skriptu AttackCollider, který je přidán k této zbrani. Poté se přidělí hodnota poškození vypočítaná na základě útoku, vrstva pro kolize a identifikace majitele dané instance zbraně.

4.1.3 Výpočty útočení

Útok hráče závisí na velkém množství statistik hráče a zbraně. Základní myšlenka je že každá zbraň má 5 hodnot které ovlivňují celkové poškození. Hlavní z nich je základní poškození zbraně, které hráčská postava bude vždy způsobovat i bez ostatních statistik. Poté má zbraň tíhu a ostrost které jsou jiné pro každou zbraň záležíc na jejím typu. Tyto statistiky negarantují hráči poškození ale umožňují mu ho zvětšit pomoci zvětšení svých statistik, které jsou síla a obratnost. Hráčská postava vlastně vyplňuje ostrost zbraně svojí obratností a tíhu zbraně svojí silou, přes toto se poté přidává víc poškození záležíc na základním poškození zbraně:

- D = poškození
- D_{crit} = kritické poškození
- D_{charge} = napřahující se poškození
- D_{base} = základní poškození zbraně
- D_{taken} = zranění na životech
- S = síla hráče
- O = obratnost hráče
- T = tíha zbraně
- R = ostrost zbraně
- C = kritický násobitel hráče
- CH = doba napřažení
- CH_{max} = maximální doba napřažení
- A = odolnost

Normální útok:

- $D = D_{base} + (T \times Clamp(\frac{S}{T}, 0{,}1) + (R \times Clamp(\frac{O}{R}, 0{,}1))$

Napřahující útok:

- $D_{charge} = D \times \frac{CH+1}{CH_{max}}$

Kritický útok:

- $D_{crit} = D \times C$

Zranění na životech:

- $D_{taken} = D \times \frac{A}{100+A}$

4.1.4 Statistiky hráčské postavy

Hráčská postava má několik statistik, které mají vliv celkové jeho chování a hratelnost hry. Tyto statistiky ovlivňují všechno od jeho rychlosti až po jeho sílu a životy.

Tyto statistiky jsou:

- Jméno
- Štěstí
- Síla
- Obratnost
- Bonusová rychlost útoku
- Bonusová kritická šance
- Bonusový kritický násobitel
- Maximum životů
- Regenerace životů
- Maximum výdrže
- Regenerace výdrže
- Odolnost
- Bonusová rychlost pohybu

Tyto statistiky se aktualizují pokaždé když se hráčovo vybavení změní. Protože jsou na vybavení závislé. Každé vybavení má své vlastní statistiky, které jsou stejné jako hráče až na zbraně, které mají sílu vyměněnou za tíhu a obratnost za ostrost. Také mají přidanou statistiku poškození. Tyto statistiky při každé aktualizaci sečtou a nastaví se hráči.

Hráči si mohou vybrat jakékoli vybavení podle toho, jak chtějí hráči hru hrát. Například jestli hráč chce moci být rychlí pořídí si výbavu která mu dává bonusovou rychlost pohybu

4.1.5 Inventář Hráče

Hráčův inventář uchovává veškeré jeho vybavení. K tomuto účelu využíváme slovník v jazyce C#, který funguje podobně jako list, ale umožňuje přiřadit každé položce dva údaje v jednom záznamu – klíč a hodnotu. Klíčem je umístění vybavení v inventáři, zatímco hodnota obsahuje samotnou položku. Tento přístup je efektivnější než použití pole, které by vyžadovalo rezervování několika prázdných míst. Další vybavení v inventáři je vybavení, které je kurentní nošeno hráčem.

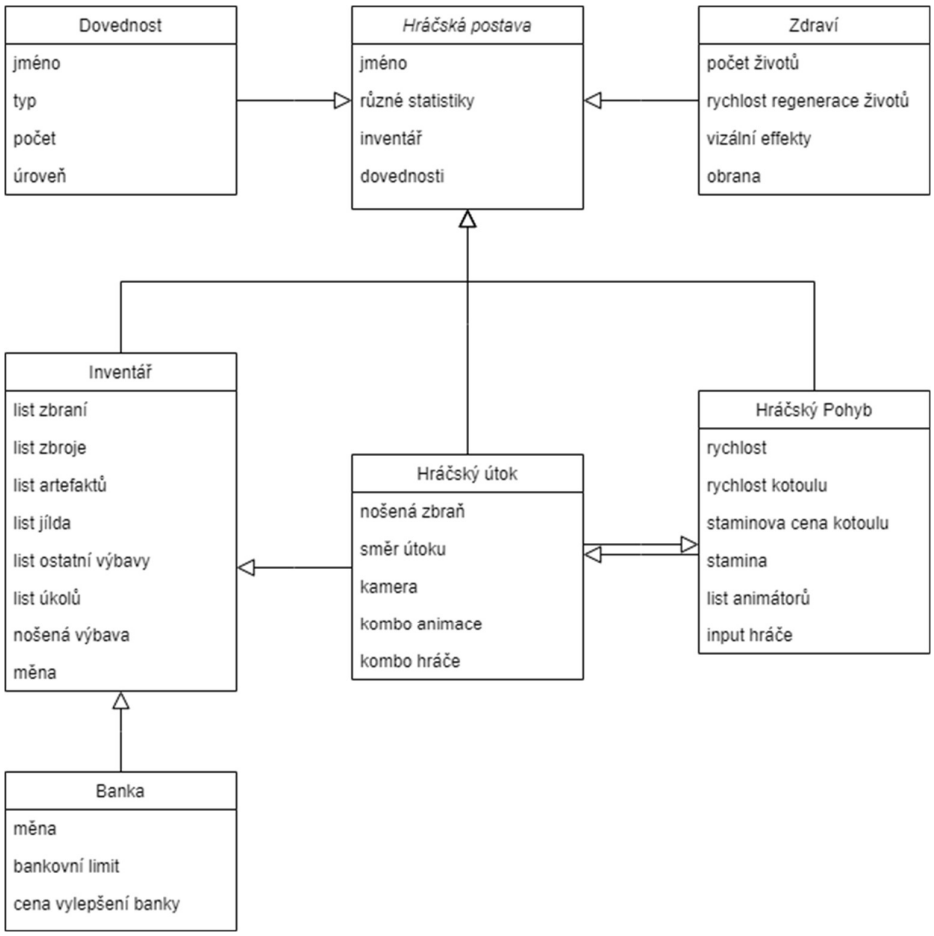
Pod vybavení spadá zbroj, zbraně, artefakty a jídlo. Zbroje a zbraně ovlivňují hráčskou postavu pouze když ho má nasazené na sobě, ale artefakty nepotřebují být nasazené na to aby ovlivňovali hráče a jeho statistiky. Jídlo musí být snědeno, aby ovlivnilo hráče.

V inventáři hráče nalezneme nejen jeho vybavení, ale také jeho měnu, kterou je zlato. Zlato hráč získává zejména prostřednictvím několika způsobů, jako je zabíjení nepřátel, prodávání výbavy obchodníkům a splňování úkolů. Tato měna slouží hráči k nakupování nového vybavení, vylepšení existujícího, nebo k dalším účelům, které jsou součástí herního prostředí.

Zlato tak představuje důležitý prvek herní ekonomiky, který hráči umožňuje postupovat ve hře a vylepšovat svoji postavu.

4.1.6 Diagram Hráče

Zde je ukázka diagramu hráče tato ukázka je velice zjednodušena ale i tak dobře reprezentuje skripty které hráčská postava využívá, jejich data a jejich propojení.



Obr. 1: Class diagram hráčské postavy

4.2 Uživatelské rozhraní

4.2.1 Inventář

Hráč má možnost interagovat s inventářem prostřednictvím uživatelského rozhraní (UI), které mu umožňuje pohodlně přidávat a odebírat vybavení z inventáře, nasazovat a sundávat vybavení. Tato funkcionalita je klíčová pro efektivní správu vybavení a optimalizaci herního prostředí podle individuálních potřeb hráče.

Inventář používá takzvaný slot-based(oknový) inventářový systém. To co umožňuje, aby byl inventář považován za slotový, závisí na způsobu, jakým je prostor a počet slotů předem nastavený. Jakmile jsou všechny obsazeny, pokud se hráč pokusí sebrat předmět. Jestli vyžaduje nový slot k obsazení, hráč ho nemůže sebrat. [16]

Díky uživatelskému rozhraní může hráč rychle a snadno procházet své položky, vybírat je a vybavovat svou postavu podle aktuální situace. Zobrazení inventáře pomocí UI přináší hráči intuitivní a přístupný způsob správy jeho vlastnictví, což zlepšuje celkový herní zážitek.

Inventář navíc zobrazuje hráči jeho hráčské statistiky. To zahrnuje důležité informace o postavě, jako jsou životy, úroveň zkušeností, síla, obratnost, odolnost nebo jakékoli další relevantní atributy postavy. Tato data umožňují hráči lépe porozumět schopnostem své postavy a přizpůsobit svou strategii a vybavení podle aktuálních potřeb a výzev ve hře.

Zobrazení hráčských statistik v inventáři umožňuje hráči rychle a snadno monitorovat svůj pokrok a efektivitu během hraní. Díky této funkcionalitě může hráč lépe plánovat své akce, vylepšovat svou postavu a optimalizovat své schopnosti a vybavení podle aktuální situace ve hře.

Tímto způsobem poskytuje inventář hráči důležitý nástroj pro sledování a vylepšování jeho herního výkonu a zároveň ho udržuje informovaného o klíčových aspektech jeho postavy a herního prostředí.

4.2.2 Přátelské nehratelné postavy

Když hráč přistoupí k přátelskému NPC, zobrazí se mu indikace, že může s NPC interagovat prostřednictvím stisknutí tlačítka "F". Po stisknutí tohoto tlačítka se objeví menu NPC obsahující jméno postavy, vítací text a tlačítko na zavření menu. Kromě toho se v menu zobrazí další tlačítko, které hráči umožní otevřít specifické menu, které je spojené s funkcemi, jež NPC poskytuje. Tímto tlačítkem hráč aktivuje další interakční možnosti s NPC, které mohou zahrnovat obchodování, zadávání úkolů nebo jakékoli další specifické akce, které NPC poskytuje.

4.2.3 Překryvný panel

V neposlední řadě uživatelské rozhraní, které hráč vidí vždy toto zahrnuje minimapu, která zobrazuje okolí hráče, panel s aktuálním zbraní, textové pole s počtem zlatých mincí a dvě čáry které se vyplňují podle toho jak moc životů a výdrže hráč má. Tato informační pole poskytují hráči potřebné informace o jeho postavě a okolním prostředí bez toho, aby musel otevřít

jakékoli menu. Toto menu se aktualizuje velice rychle tím se zajišťuje že hráč dostává aktuální informace o své postavě.

4.3 Přátelské nehratelné postavy

V rámci hry lze potkat větší množství nehratelných postav (non-playble characters, zkráceně NPC). Ty se dělí v základu na tři typy:

- Prodavač
- Vylepšovač
- Úkolovník

Přátelské NPC jsou jako duše města, rozptýlené po různých místech, připravené pomoci a podpořit hráče na jeho dobrodružství. Každý z nich má své místo, které reprezentuje jejich roli a zájmy. Například prodavač zbraní se nachází ve své malé zbrojnici, kde na zdi visí meče a luky, zatímco vylepšovač zbraní najdeme v dílně, obklopený kovadlinou a pecí, kde ladí a zdokonaluje zbraně.

Když hráčská postava přistoupí k jednomu z těchto přátelských NPC a zmáčkne klávesu „F“, otevře se jim vstřícné menu. Objeví se obrázek NPC, společně s vítací hláškou. Tlačítka v menu se liší v závislosti na typu NPC. U prodavačů máme tlačítko "Obchod", zatímco u vylepšovače je tlačítko "Vylepšit" a u úkolovníků tlačítko „Úkol“, ale u všech najdeme tlačítko „Vrátit“, které vypne menu.

Každé interakce s těmito NPC může hráči poskytnout nejen potřebné vybavení, vylepšení a úkoly, ale mohou i prohloubit příběh a svět hry.

4.3.1 Prodavač

Každý obchodník má svou specializaci, což znamená, že se zaměřuje na konkrétní typ vybavení, které nabízí hráčům. Tato specializace zahrnuje kategorie, jako jsou zbraně, zbroj, jídlo, artefakty nebo jinou kořist z nepřátel.

Hráč může taky svoje vybavení prodat jakémukoli obchodníkovi za stejnou cenu, takže nemusí běhat tam a zpět aby dostal nejlepší možnou cenu.

4.3.2 Vylepšovač

Každý vylepšovač má svou specializaci, což znamená, že se zaměřuje na konkrétní typ vybavení, které může vylepšit. Tato specializace zahrnuje kategorie zbraně nebo zbrojí. Vylepšovači vylepšují specifické statistiky výbavy. Pro zbraně se vylepší poškození a pro zbroj se vylepší odolnost o dané procento specifické statistiky. Výbava může být vylepšena celkově desetkrát.

4.3.3 Úkolovník

Úkolovníci dávají hráčům cíle a motivaci k průzkumu herního světa, plnění úkolů a dosahování herního pokroku. Tyto úkoly mohou být součástí hlavního příběhu, vedlejších příběhů nebo speciálních událostí a událostí v herním světě.

4.4 Nepřátelé

Nepřátele, jako nepostradatelná součást herního prostředí, představují klíčový prvek ve hře, a proto je zásadní, aby jejich design byl velmi flexibilní a snadno upravitelný.

Výhodou modulárního přístupu k designu nepřátel je možnost snadno aktualizovat a vylepšovat herní zážitek v průběhu času. Modularita dovoluje reagovat na zpětnou vazbu hráčů, přidávat nový obsah, opravovat chyby a udržovat herní prostředí svěží a zajímavé i dlouho po vydání hry.

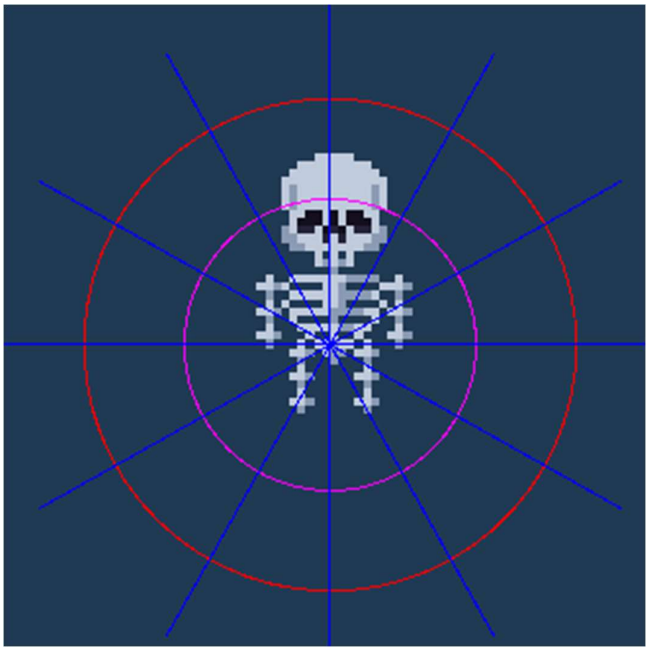
Celkově je klíčové, aby design nepřátel reflektoval potřeby a očekávání hráčů a umožnil hráčovy zábavný ale ne lehký průchod hrou. Modulární a jednoduše změnitelné nepřátele poskytují prostředky pro dosažení tohoto cíle a zajistí zábavný a vyvážený herní zážitek pro hráče všech úrovní dovedností a preferencí.

4.4.1 Nepřítel na blízko

Jedním z variantních typů nepřátel je nepřítel na blízko, který slouží jako výchozí forma a z ní jsou odvozeny všechny ostatní typy nepřátel. Tato modulární struktura umožňuje snadnou úpravu vlastností a chování nepřátel podle potřeby.

Pohyb nepřítele je řízen pomocí 12 směrů kolem něj, přičemž každý z těchto směrů odpovídá jedné hodině na ciferníku. Při pohybu zvažuje přítomnost hráče, zdí a dalších nepřátel v okolí a na základě těchto informací vypočítává optimální směr pohybu. Tato dynamická reakce umožňuje nepřátelům efektivněji navigovat prostředím a reagovat na změny ve hře.

V případě útoku nepřítel nečeká na příchod hráče zcela pasivně. Namísto toho sleduje vzdálenost mezi sebou a hráčem a aktivuje svůj útočný mechanismus, jakmile se hráč přiblíží na určitou vzdálenost. Následně začne krátký odpočet, jehož délka závisí na síle útoku. Tímto způsobem má hráč trochu času na reakci a vytváří se pravidelnější a interaktivnější herní prostředí. Tato schopnost umožňuje nepřátelům útočit efektivně a reagovat na pohyb hráče v jejich blízkosti.



Obr. 2: Ukázka nepřítele a vizualizace jeho vize

4.4.2 Nepřítel na dálku

Nepřítel na dálku je v podstatě variací nepřítele na blízko, přičemž dědí jeho způsob pohybu. Stejně jako jeho blízký protějšek, i tento nepřítel se pohybuje podle 12 směrů kolem sebe, přičemž každý směr odpovídá jedné hodině na ciferníku.

Útok tohoto nepřítele funguje na podobném principu jako u nepřítele na blízko. Když detekuje hráče v blízkosti vytvoří projektil odpovídající jeho typu, například šíp. Jakmile je projektil vytvořen, letí směrem, kde hráč stál, když byl vystřelen. Jestliže projektil hráče zasáhne, ubere mu na životech, a poté se zničí. Stejně tak se projektil zničí, jestliže narazí do zdi.

4.4.3 Boss

Bossové ve hře mají speciální postavení a jsou navrženi tak, aby poskytovali hráčům vysokou výzvu a napětí. Stejně jako u standardních nepřátel, i bossové využívají koncept dědění, avšak jejich struktura je odlišná.

Na rozdíl od běžných nepřátel, bossové nejsou odvozeni z třídy nepřítel, ale mají vlastní hierarchii dědění. Každý boss je rozdělen do fází, které dědí ze společného skriptu BossPhase. Tento design umožňuje rozdělit bosse do více fází, přičemž každá fáze může mít unikátní chování, schopnosti a strategie boje.

Díky modulární struktuře fází lze bosse designovat tak, aby se měnili během boje a poskytovali hráčům dynamický a vzrušující zážitek. Každá fáze může nabídnout nové výzvy a úkoly pro hráče, což zvyšuje jejich angažovanost a motivaci pro překonání bosse.

4.5 Procedurální generace dungeonu

V následní části budu podrobně vysvětlovat jak se procedurálně vytváří dungeon. Tento proces využívá kombinaci cestové generace a předem definovaných variací místností. Budu detailně popisovat postup, kterým se tato metoda řídí, včetně konkrétních kroků a principů, které jsou při tvorbě dungeonů klíčové.

4.5.1 Příprava místonstí

Než začneme generovat dungeon, musíme se připravit několik prefabů místností. Každá místnost by měla mít svůj unikátní design a atmosféru, která přispívá k celkovému dojmu z hry. Kromě estetického hlediska musíme také zvážit praktické aspekty, jako je velikost místnosti a umístění významných prvků. V přípravné fázi je důležité dbát na to, aby každý prefab obsahoval všechny potřebné prvky. Dále je důležité přidat do každé místnosti vhodné dekorace a objekty, které dodají prostoru autentičnost a zajímavost. Některé místnosti mohou vyžadovat specifické prvky nebo funkce, které musí být implementovány v rámci jejich prefabu. Například místnosti s pastmi nebo úkoly pro hráče mohou vyžadovat speciální skripty nebo interakční prvky. Důkladná příprava prefabů nám umožní snadněji a efektivněji generovat dungeon a zajistit bohatý a zábavný herní zážitek pro hráče. Každá místnost musí mít přiřazené následující prvky:

- Čtyři vchody do všech stran. Čtyři zdi do všech stran.
- Hlavní tělo místnosti.
- Vrstvu pro mini mapu.
- RoomScript který má přidaný:
    - Počet vln nepřátel.
    - Počet nepřátel v každé vlně.
    - Dříve zmíněné vchody a zdi.
    - Obstrukce, nejsou povinné, ale hodně pomohou přidat variaci naší generace.

Po vytvoření různých variant místností s odlišnými rozměry a počty vchodů můžeme tyto místnosti přiřadit do skriptu DungeonGenerator. Tento skript bude sloužit k vytváření dungeonů v naší hře. Vytvoření variant místností nám umožňuje vytvářet různorodé prostory v dungeonu, což přispívá k jeho rozmanitosti a zábavnosti pro hráče. Přiřazení místností do skriptu DungeonGenerator umožňuje dynamicky generovat dungeon na začátku hry na základě určitých pravidel nebo náhodných faktorů, což vytváří plynulý a poutavý herní zážitek.

Máme také předpřipravenou třídu jménem RoomVar která v sobě drží:

- Pozici – Vector2
- Rodiče – Vector2
- Svojí velikost/délku – čislo
- Směry – bool list délky 4
- Typ místnosti – enum

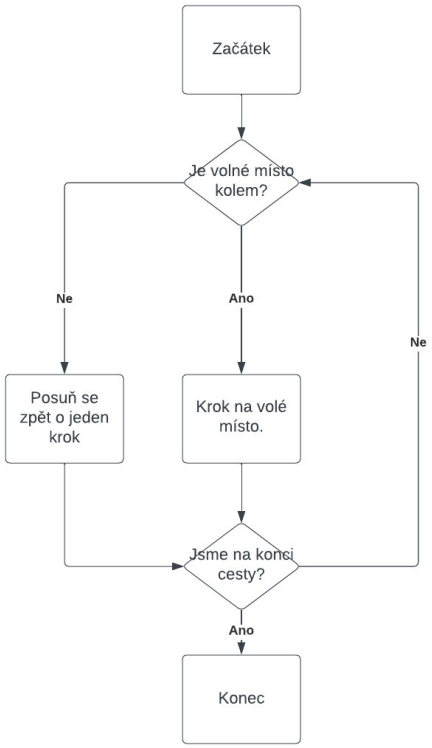
4.5.2 Generace cest

Prvním úkolem je vytvořit hlavní cestu dungeonu. Toho dosáhneme použitím algoritmu, který systematicky prozkoumává okolí dané pozice a hledá volná místa. Pokud žádné volné místo není nalezeno, algoritmus se vrátí na předchozí pozici. Pokud je nalezeno právě jedno volné místo, algoritmus se vydá tímto směrem. Pokud jsou k dispozici více volných míst, algoritmus náhodně vybere jedno z nich a algoritmus se tímto směrem vydá. Tímto způsobem je zajištěno dynamické a náhodné generování hlavní cesty dungeonu. Pro lepší kontrolu procesu generování je možné této funkci předat počáteční pozici, délku cesty a informaci o tom, zda se jedná o hlavní nebo vedlejší cestu. Každý krok, který se provede má šanci uložit svou pozici do listu pozic pro vedlejší větve který se poté projde a pro každou uloženou pozici se vygeneruje další cesta s krátkou délkou která na svém konci má místnost s odměnou.

Každý krok co provedeme tak vytvoříme nový RoomVar s aktuální pozicí jako svou pozici, poslední pozicí jako rodiče, délka místnosti je prozatím u všech 1 ale to že toto číslo můžeme měnit v algorytmu nám v budoucnosti umožní udělat a místnosti větší než 32x32, do směrů prozatím dáváme jen prátdný array bez dat a typ který se vybere záležíc na jestli jsme na konci cesty, jestli jsme na vedlejší cestě a i na náhodné šanci. Tento RoomVar se přidá do listu jménem roomVars. Každý krok co se provede také může náhodně uložit svoji pozici do listu větve. Tento list se generaci hlavní cesty projde a vygenerují se vedlejší cesty s odměnovímy místonstmy.

Po vytvoření hlavní a vedlejších neboli větvových cestách se projde skrz list roomVars a vygenerují se spojení místností pomocí rodiče roomVaru. Tyto spojení se udělají nejen mezi rodičem a dítětem ale i dítětem a rodičem. Tento systém zajisťuje spojení mezi nejen dvoumi nebo třemi místonsti mezi sebou ale i čtyřmi nebo pěti.

Teď jsme připraveni vytvářet naše herní objekty. Začneme procházet list roomVars a pro každý z nich vytoříme náhodnou místnost z některých naších prefabů. Jestli je místnost start, konec nebo odměnová místnost dungonu vytvoří se prefab nejmenší místnosti ale jestli je to normální místnost vytvoří se náhodně jeden z prefabů. Podle směrů naší místnosti se nastavý vstupová cesta nebo zeď.



Obr. 3: Diagram generace cest

4.5.3 Vstup do místnosti

Po vstupu do místnosti se ve všech vchodech zavřou dveře, které hráče zamknou v místnosti do té doby, než ji hráč dokončí. Po zavření dveří se začnou objevovat nepřátelé po vlnách. Velikost a počet vln se dá upravit a záleží na tom, jak je místnost veliká. Poté co hráč zabije všechny nepřátele ve všech vlnách místnost se otevře a hráč může jít do jiné místnosti.

4.5.4 Porovnání různých provedení

Jak bylo dříve řečeno můj systém kombinuje generaci cest a přednastavené variace místností toto není jediný způsob, jak generovat náhodný herní dungeon(žalář). Tento systém je inspirován hrou Enter the Guneon[17] a Soul Knight.[18]

Například video hra Dead Cells[19] využívá generaci pomocí variací místností s přednastavenými body které určují místo napojení další místnosti. Tento systém umožňuje mít místnosti s rozsahem velikostí a můj jen místnosti které mají velikost 32x32.

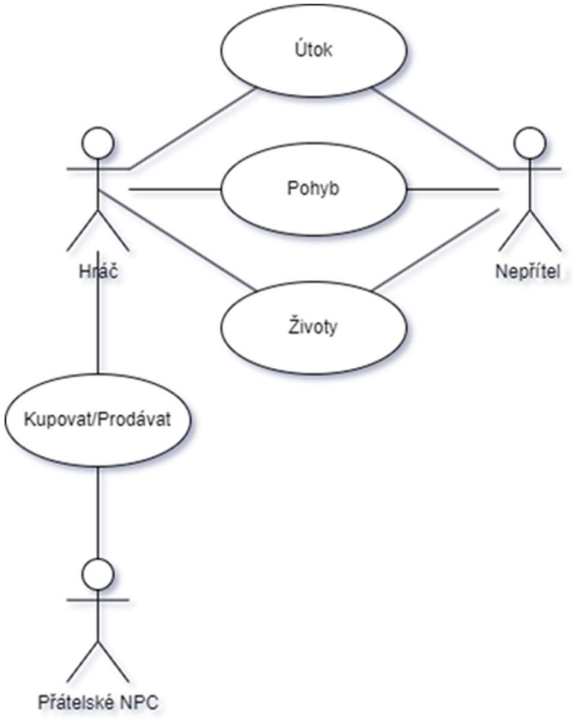
V diplomové práci „Procedural generation and realtime rendering of planetary bodies.“[20]. Autor porovnává několik jiných systémů generace od fraktálu až po šum. Šum je náhodně generovaný vzor, který se dá inicializovat pomocí seedu (semínka). Potom jestli chceme z šumu náhodnou hodnotu předáme mu číslo x a číslo y které můžeme brát jako 2d pozici. Funkce nám vrátí číslo, s kterým poté můžeme pracovat. Na šum se dříve využíval takzvaný Perlinový

šum[21] ale v dnešní době už se využívá modernější, rychlejší a méně nár SimpleX šum[22]. Se šumem jsem také sám už párkrát experimentoval a sice s ním rád pracuji nemyslel jsem si, že se hodí pro tento projekt. Procedurální generace pomocí fraktálu funguje tak, že se složité a detailní struktury generují opakovaným aplikováním jednoduchých matematických pravidel, které vytvářejí samo-podobné vzory na různých měřítcích. S fraktály jsem nikdy nepracoval a také mi nepřišli vhodné pro tento projekt. Autor nakonec vybral kombinaci šumu a fraktálu, která vytváří velice zajímavý výstup. Ale jelikož já používám přednastavené variace místností pro mě tento systém není zas tak užitečný.

V diplomové práci „Procedural Generation of Minecraft Villages utilizing the Wave Function Collapse Algorithm“[23]. Autor vysvětluje využití takzvané WFC (Wave Function Collapse) funkce. WFC je technika pro generování složitých vzorů nebo struktur založených na vstupních datech. Algoritmus pracuje tak, že analyzuje vstupní vzor a snaží se vytvořit nový vzor, který respektuje vzory a vlastnosti vstupního vzoru. Autor využívá WFC na generaci vlastních vesnic ve videohře Minecraft[24]. Minecraft je počítačová hra, která se odehrává v otevřeném světě, kde má hráč neomezenou svobodu pohybu a činnosti. Myslím, že použití WFC na generaci dungeonů by vytvořilo velice zajímavý výsledek. Zajistilo by to nekonečnou generaci dungeonu z malé palety, ale nevybral jsem si ho z důvodu, že je komplikovaný na implementaci a nákladný na výkon počítače.

4.6 Use case diagram hry

V rámci projektu vznikl i jednoduchý usecase diagram, který zobrazuje jednotlivé role v rámci herního světa. Můžeme zde vidět jak hráčská postava interaguje s herním světem a jaké vlastnosti sdílí s ostatními rolemi, které jsou řízené algoritmem.



Obr. 4: Use case diagram hry

5 ZÁVĚR

Cílem práce bylo vytvořit 2D pixel art dungeon-crawler hru Dungeoneer, tak, aby byla opakovatelně hratelná a umožňovala hráčům přizpůsobení své herní postavy.

Ve hře je možné vytvořit nové uložení hráčských dat nebo pokračovat hrát s daty které už jsou uloženy. Interagovat, prodávat a nakupovat výbavu u přátelských nehratelných charakterů. Některá výbava se dá i u přátelských nehratelných charakterů vylepšit. Hráč má možnost spravovat svůj inventář například sundávat nebo nandávat zbroj a kontrolovat svoje hráčské statistiky. Hráč dále může vstoupit do procedurálně generovaného dungeonu pomocí velkých stupních dveří. Poté co do nich vstoupí hráč se objeví v dungeonu, který když začne procházet začne se objevovat variace nepřátel. Potom co hráč dojde na konec dungeonu začne takzvaný boss fight kde bojuje jenom s jedním silným nepřítelem, který má několik fází. Po zabití bosse hráč je odměněn a vrátí se zpátky do města kde může prodat věci co našel cestou v dungeonu a vylepšit si výbavu.

Hru se podařilo doladit do stavu beta verze (dostupná z https://l19uid.itch.io/dungeoneer) která je nyní ve fázi testování menší komunitou hráčů, na základě jejichž zpětné vazby probíhá ladění herních mechanik a opravy chyb. Do budoucna tak hra získává jasnou podobu, ve které by mohla být publikována na platformě Steam.

Projekt má stále ještě drobné nedostatky, například chybí úvodní tréninkový dungeon, ve kterém by se hráč naučil základní ovládání a mechaniky hry. Podle mě není zas tak nutná, jelikož hra není komplikovaná na naučení. Ale více nutnější úkoly neboli mise které by hráč dostával od přátelských nehratelných charakterů, které by přidali ke příběhu hry a hráčovi odměně což by také přidalo znovu hratelnosti.

Celkově projekt hodnotím kladně, hra získala skvělou zpětnou vazbu a dobré ohlasy, a to jak z řad mladších hráčů, převážně spolužáků, tak také od zkušenějšími programátorů. Zdá se tak, že má dobrý potenciál na úspěch i v širší herní komunitě. Budoucí zaměření bude určitě rozšíření komunity pomocí sociálních medií.

6 POUŽITÁ LITERATURA

[1] WIKIPEDIA CONTRIBUTORS, 2019a. Game engine. *Wikipedia* [online]. B.m.: Wikimedia Foundation [vid. 2024-02-26]. Dostupné z: https://en.wikipedia.org/wiki/Game_engine

[2] WIKIPEDIA, 2022. JavaScript Object Notation. *Wikipedia* [online] [vid. 2024-02-26]. Dostupné z: https://cs.wikipedia.org/wiki/JavaScript_Object_Notation

[3] KUDRYASHOV, Serhiy, 2019. Procedurally generated content in computer game. theses.cz [online] [vid. 2024-03-22]. Dostupné z: https://theses.cz/id/kuebew/Serhiy-Kudryashov-Bachelors-Thesis.pdf

[4] WIKIPEDIA CONTRIBUTORS, 2023b. Exkluzivní Disjunkce. Wikipedia [online] [vid. 2024-03-10]. Dostupné z: https://cs.wikipedia.org/wiki/Exkluzivn%C3%AD_disjunkce

[5] TECHNOLOGIES, Unity, 2024a. Unity - Manual: Prefabs. *docs.unity3d.com* [online] [vid. 2024-02-26]. Dostupné z: https://docs.unity3d.com/Manual/Prefabs.html

[6] WIKIPEDIA, 2020. Instance (informatika). *Wikipedia* [online] [vid. 2024-02-26]. Dostupné z: https://cs.wikipedia.org/wiki/Instance_(informatika)

[7] TECHNOLOGIES, Unity, 2024b. Unity - Scripting API: Rigidbody2D. *docs.unity3d.com* [online] [vid. 2024-02-26]. Dostupné z: https://docs.unity3d.com/ScriptReference/Rigidbody2D.html

[8] WIKIPEDIA, 2022. Hladomorna. *Wikipedia* [online] [vid. 2024-02-26]. Dostupné z: https://cs.wikipedia.org/wiki/Hladomorna

[9] CONTRIBUTORS, WIKIPEDIA, 2023a. Výčtový typ. Wikipedia [online] [vid. 2024-02-28]. Dostupné z: https://cs.wikipedia.org/wiki/V

[10] CAPELLO, David, 2019. Aseprite. Aseprite.org [online] [vid. 2024-02-26]. Dostupné z: https://www.aseprite.org

[11] UNITY TECHNOLOGIES, 2019. Unity - Unity. Unity [online] [vid. 2024-02-26]. Dostupné z: https://unity.com

[12] JETBRAINS S.R.O., 2024. Rider: The Cross-Platform .NET IDE from JetBrains. JetBrains [online] [vid. 2024-02-26]. Dostupné z: https://www.jetbrains.com/rider/

[13] MICROSOFT, 2022. A tour of C# - Overview. *learn.microsoft.com* [online] [vid. 2024-02-28]. Dostupné z: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

[14] DŘÍMAL, Bc. Jiří, 2022. Vývoj videoher v herním enginu Unity. *Digitální Knihovna Univerzity Pardubice* [online] [vid. 2024-02-26]. Dostupné z: https://dk.upce.cz/bitstream/handle/10195/79583/DrimalJ_VyvojVideoher_AK_2022.pdf?sequence=1&isAllowed=y

[15] WIKIPEDIA CONTRIBUTORS, 2024. Třída (programování). *Wikipedia* [online] [vid. 2024-02-28]. Dostupné z: https://cs.wikipedia.org/wiki/T%C5%99%C3%ADda_(programov%C3%A1n%C3%AD)

[16] JÖNSSON, Tim a Glasell JOSEFIN , 2023. Active Inventory Systems in Games And What Defines Them. Diva Portal [online] [vid. 2024-03-22]. Dostupné z: https://uu.diva-portal.org/smash/get/diva2:1809544/FULLTEXT01.pdf

[17] DODGE ROLL, 2024. Enter The Gungeon. *enterthegungeon.com* [online] [vid. 2024-03-26]. Dostupné z: https://enterthegungeon.com

[18] CHILLY ROOM, 2024. Soul Knight. *chillyroom.com* [online] [vid. 2024-03-26]. Dostupné z: https://soulknight.chillyroom.com/et
[19] MOTION TWIN, 2024. Dead Cells. *dead-cells.com* [online] [vid. 2024-03-26]. Dostupné z: https://dead-cells.com
[20] KAHOUN, Martin , 2010. *Procedural generation and realtime rendering of planetary bodies* [online] [vid. 2024-03-25]. Dostupné z: https://cgg.mff.cuni.cz/~kahoun/files/kahoun_martin_bachelor_thesis.pdf
[21] WIKIPEDIA CONTRIBUTORS, 2021. Perlin noise. *Wikipedia* [online] [vid. 2024-03-25]. Dostupné z: https://en.wikipedia.org/wiki/Perlin_noise#:~:text=Perlin%20noise%20is%20a%20procedural
[22] WIKIPEDIA CONTRIBUTORS, 2023. Simplex noise. *Wikipedia* [online] [vid. 2024-03-25]. Dostupné z: https://en.wikipedia.org/wiki/Simplex_noise#:~:text=Simplex%20noise%20is%20the%20result
[23] ČERNÝ, Mgr. Vojtěch , 2022. Procedural Generation of Minecraft Villages utilizing the Wave Function Collapse Algorithm. cuni.cz [online] [vid. 2024-03-25]. Dostupné z: https://dspace.cuni.cz/bitstream/handle/20.500.11956/171653/120411674.pdf?sequence=1&isAllowed=y
[24] MOJANG, 2024. Minecraft. *minecraft.net* [online] [vid. 2024-03-26]. Dostupné z: https://www.minecraft.net/en-us

7 ZAKOUPENÉ ASSETY

[1] GOWL, 2024. Nature pixels 2. *itch.io* [online] [vid. 2024-03-25]. Dostupné z: https://gowldev.itch.io/nature-pixels-v2
[2] GRIGOREEN, 2024. Cozy interiors game assets. *itch.io* [online] [vid. 2024-03-25]. Dostupné z: https://grigoreen.itch.io/cozyinteriors
[3] NAMATNIEKS, 2023. Era of Fantasy: Village Props - Pixelart Asset Pack. *itch.io* [online] [vid. 2024-03-25]. Dostupné z: https://aamatniekss.itch.io/era-of-fantasy-village-props
[4] ZEGLEY , Hayden , 2023. 2D Pixel Art Character Template Asset Pack. *itch.io* [online] [vid. 2024-03-25]. Dostupné z: https://zegley.itch.io/2d-platformermetroidvania-asset-pack

8 SEZNAM OBRÁZKŮ A TABULEK

Obr. 1: Class diagram hráčské postavy

Obr. 2: Ukázka nepřítele a vizualizace jeho vize

Obr. 3: Diagram generace cest

Obr. 4: Use case diagram hry

9 ZADÁNÍ MATURITNÍHO PROJEKTU Z INFORMATICKÝCH PŘEDMĚTŮ

Jméno a příjmení: *Tomáš Vik*

Pro školní rok: *2023/2024*

Třída: *3. B*

Obor: *Informační technologie 18-20-M/01*

Téma práce: *Jednoduchá hra – Dungeoneer*

Vedoucí práce: *Jan Mottl*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem projektu bude vytvořit 2D hru, která bude žánrově spadat do typu her rougelike (případně soulslike). Kamera bude hráčskou postavu sledovat ze shora. Hra bude vytvořena v herním enginu Unity a veškeré skripty budou psané v C#. Hra by byla rozdělena na 2 hlavní části.

Pevným bodem ve hře bude město, kde půjde obchodovat s předměty a získávat úkoly v od postav. Z města lze vejít do akční části hry, kterou jsou katakomby (dunegon).

Katakomby (Dungeon) je akční část hry, kde hráčova postava může postupovat skrz jednotlivé náhodně generované místnosti. V těchto místnostech hráč bojuje s nepřáteli a získává předměty se kterými se lze vrátit do města.

V rámci maturitního projekt se vyhotoví mechaniky hráčské postavy tedy pohyb, systém útočení, inventář a hráčova výbava. Taky systém obchodníků, dialogů a přijímáních úkolů. Akční část bude hotova pro základní pár úrovní hry, kdy se představí část palety nepřátel. Nakonec bude navrženo možné rozšíření nejen hratelnosti ale i počet vybavení.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- Září – Studium dokumentace a rozplánování konkrétních mechanik hry.
- Říjen/2023 – Tvorba generátoru katakomb (dungeon).
- Listopad/2023 – Tvorba nepřátel a jejich mechanik.
- Prosinec/2023 – Systém inventáře a uživatelské rozhraní.
- Leden/2024 – Finální úpravy pro beta test.
- Únor/2024 – Betatestování projektu a příprava podkladů pro dokumentaci.
- Březen/2024 – Tvorba dokumentace.