|  |  |  |
| --- | --- | --- |
| **Univerzita Hradec Králové**  **Fakulta informatiky a managementu**  **Katedra informatiky a kvantitativních metod** | | |
| **Vývoj počítačových her v enginu Unity**  Bakalářská práce | | |
| Autor: Dominik Kolouch  Studijní obor: Aplikovaná informatika | | |
| Vedoucí práce: Ing. Jakub Beneš | | |
| Hradec Králové | Duben 2023 | |
| Prohlášení:  Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury. | | | |
| V Hradci Králové dne 17.4.2023 | | *Dominik Kolouch* | |
| Poděkování:  Děkuji vedoucímu bakalářské práce Ing. Jakubovi Benešovi za metodické vedení práce, odborné rady a vstřícný přístup. Dále děkuji Michaelovi Kraveckému za poskytnuté konzultace a umělecké materiály. | | | |

Anotace

Bakalářská práce se zabývá vývojem počítačových her. Cílem bakalářské práce je prozkoumat a přiblížit čtenáři proces vývoje počítačových her a jeho historii. Od návrhu designu a levelů hry, přes samotný vývoj až po následné vydání hry na existujících platformu pro digitální publikaci her a následné testování hry. Dále porovnat vybrané dostupné herní enginy podle vybraných kritérií. V praktické části práce pak popsat vybrané funkce Unity užité pro vývoj 2D roguelike hry, včetně procedurálního generování světa v Unity enginu za pomoci programu Visual Studio včetně ukázek kódu jednotlivých scriptů a objektů. Zároveň vytvořit dokument herního designu za použití poznatků z teoretické části práce.

Annotation

Title: Development of computer games in the Unity engine

The bachelor thesis deals with the development of computer games. The aim of the bachelor thesis is to explore and introduce the reader to the process of computer game development and its history. From game design and level design, through development, to the subsequent release of the game on existing digital publishing platforms and subsequent playtesting. Furthermore, to compare selected available game engines according to selected criteria. Then, in the practical part of the thesis, describe selected Unity features used for the development of a 2D roguelike game, including procedural world generation in the Unity engine using Visual Studio, including code samples of individual scripts and objects. At the same time create a game design document using the knowledge from the theoretical part of the thesis.

Obsah

[1 Úvod 1](#_Toc132309400)

[2 Cíl práce a metodika 2](#_Toc132309401)

[2.1 Historie 3](#_Toc132309402)

[3 Herní design 4](#_Toc132309403)

[3.1 Dokument herního designu (GDD) 4](#_Toc132309404)

[3.2 Hratelnost 4](#_Toc132309405)

[3.2.1 Principy směru 4](#_Toc132309406)

[3.2.2 Principy chování 5](#_Toc132309407)

[3.2.3 Princip prostředí 6](#_Toc132309408)

[3.3 Herní mechanismy 6](#_Toc132309409)

[3.3.1 Prostor 6](#_Toc132309410)

[3.3.2 Objekty, atributy a stav 8](#_Toc132309411)

[3.3.3 Akce 9](#_Toc132309412)

[3.3.4 Pravidla 10](#_Toc132309413)

[3.3.5 Dovednosti 13](#_Toc132309414)

[3.3.6 Šance 14](#_Toc132309415)

[3.4 Level design 15](#_Toc132309416)

[3.4.1 Rozdělení typu level designu 17](#_Toc132309417)

[3.4.2 Level flow 21](#_Toc132309418)

[3.4.3 Design objektů 24](#_Toc132309419)

[3.5 Design nepřátel 25](#_Toc132309420)

[3.5.1 Design bossů 28](#_Toc132309421)

[3.6 Procedurální generování 29](#_Toc132309422)

[3.6.1 Typy procedurálního generování map 30](#_Toc132309423)

[4 Herní enginy 33](#_Toc132309424)

[4.1 Unity Engine 33](#_Toc132309425)

[4.2 Godot Engine 34](#_Toc132309426)

[4.3 Unreal Engine 34](#_Toc132309427)

[4.4 Porovnání 34](#_Toc132309428)

[5 Praktická část – Vývoj hry v Unity enginu 35](#_Toc132309429)

[5.1 Implementace GDD 35](#_Toc132309430)

[5.1.1 Základní informace 35](#_Toc132309431)

[5.1.2 Hratelnost a mechaniky 36](#_Toc132309432)

[5.1.3 Levely 37](#_Toc132309433)

[5.1.4 Interface 37](#_Toc132309434)

[5.1.5 AI 37](#_Toc132309435)

[5.2 Vývoj hry 38](#_Toc132309436)

[5.2.1 GameObject 38](#_Toc132309437)

[5.2.2 Kamera 45](#_Toc132309438)

[5.2.3 Animator 48](#_Toc132309439)

[5.2.4 Vytváření levelu 49](#_Toc132309440)

[5.2.5 Testování 51](#_Toc132309441)

[6 Shrnutí výsledků 52](#_Toc132309442)

[7 Závěry a doporučení 53](#_Toc132309443)

[8 Seznam použité literatury 54](#_Toc132309444)

Seznam obrázků

[Obrázek 1 Diskrétní prostor piškvorek, převzato [3] 7](#_Toc132629551)

[Obrázek 2 Virtuální prostory pro hru 20 otázek, převzato [3], přeloženo 8](#_Toc132629552)

[Obrázek 3 Informace o stavech, převzato [3] přeloženo 9](#_Toc132629553)

[Obrázek 4 Diagram pravidel převzato [3], přeloženo 11](#_Toc132629554)

[Obrázek 5 Plánek podzemních sklepů do hry DND, Michael Kraveckyj 16](#_Toc132629555)

[Obrázek 6 „Mysterious tunnels“, Příklad „Alley“ level designu, Michael Kraveckyj 17](#_Toc132629556)

[Obrázek 7 „Velvevelkyr“, Příklad „Island“ level designu, Michael Kraveckyj 18](#_Toc132629557)

[Obrázek 8 Příklad „Path“ level designu, Michael Kraveckyj 19](#_Toc132629558)

[Obrázek 9 Příklad „Hub“ Level designu, Michael Kraveckyj 20](#_Toc132629559)

[Obrázek 10 porovnání dvou cest [19], Andrew Yoder 22](#_Toc132629560)

[Obrázek 11 Příklad „critical path“, Michael Kraveckyj 23](#_Toc132629561)

[Obrázek 12 Typ nepřítele se střelbou, Vojtěch Košťál 26](#_Toc132629562)

[Obrázek 13 Typ nepřítele Burrower, Vojtěch Košťál 26](#_Toc132629563)

[Obrázek 14 Typ nepřítele s teleportací, Vojtěch Košťál 27](#_Toc132629564)

[Obrázek 15 Typ nepřítele s blokováním, Vojtěch Košťál 27](#_Toc132629565)

[Obrázek 16 Výsledek generování Simple Room-Placement 30](#_Toc132629566)

[Obrázek 17 Výsledek generování místností s binárním rozdělením prostoru 31](#_Toc132629567)

[Obrázek 18 Výsledek generování místností pomocí celulárního automatu, [33] 32](#_Toc132629568)

[Obrázek 19 Ukázka GameObject „LevelExit“ 38](#_Toc132629569)

[Obrázek 20 Ukázka objektu krabice (Breakable) 39](#_Toc132629570)

[Obrázek 21 Ukázka nepřítele Skeleton 41](#_Toc132629571)

[Obrázek 22 Ukázka komponenty script u nepřítele 42](#_Toc132629572)

[Obrázek 23 Ukázka vzhledu a sekvence bosse 43](#_Toc132629573)

[Obrázek 24 Propojení kamer scény a ovládacích scriptů 46](#_Toc132629574)

[Obrázek 25 Kamera minimapy a Hlavní kamera 46](#_Toc132629575)

[Obrázek 26 Použtí BigMap camery 47](#_Toc132629576)

[Obrázek 27 Řídící body pro animaci chůze hráče 48](#_Toc132629577)

[Obrázek 28 Ukázka animace kotoulu 48](#_Toc132629578)

[Obrázek 29 Editor středu místnosti 49](#_Toc132629579)

Seznam tabulek

[Tabulka 1 Subjektivní porovnání herních enginů podle vybraných kritérii 35](#_Toc132309074)

[Tabulka 2 Zpětná vazba ke hře a jejich řešení 51](#_Toc132309075)

Seznam ukázek kódu

[Ukázka kódu 1 Kód pro obsluhu zničení Breakable objektu 40](#_Toc132309121)

[Ukázka kódu 2 Chování nepřítele "Wanderer" 42](#_Toc132309122)

[Ukázka kódu 3 Přepnutí sekvence podle aktuálních životů nepřítele: 44](#_Toc132309123)

[Ukázka kódu 4 Akce v sekvenci, vyvolání dalších nepřátel bossem 44](#_Toc132309124)

[Ukázka kódu 5 Akce v sekvenci, při které nepřítel má střílet. 44](#_Toc132309125)

[Ukázka kódu 6 Kód pro aktivaci BigMap camery 47](#_Toc132309126)

[Ukázka kódu 7 Kód pro umístění bodu speciální místnosti „obchod“. 50](#_Toc132309127)

[Ukázka kódu 8 Kód pro vygenerování speciální místnosti „místnost se zbraní“ 50](#_Toc132309128)

[Ukázka kódu 9 Kód upraveného switch case pro posun generačního bodu 50](#_Toc132309129)

# Úvod

Téměř všichni lidé z vyspělých zemí se již setkali s pojmy jako počítačová hra nebo videohra. [6, 7, 8] Počítače jsou dnes součástí téměř každého domova. Herní průmysl se stále více dostává do povědomí veřejnosti. Tvorba her se také čím dál více objevuje ve výuce na střední a vysokých školách. V současné době se vývoj her vyučuje na několika vysokých školách v České republice, například na ČVUT v Praze a Univerzitě Karlově v Praze, a také v zahraničí, například na IT univerzitě v Kodani v Německu a na Bournemouth University v Bournemouthu. O IT obor se zajímá stále více lidí. V České republice mezi významné herní vývojářské společnosti patří Warhorse, lidé ze společností 2K Czech a Illusion Softworks. Většina společností spolupracuje se školami, kde se snaží najít nové zaměstnance, ale kvůli nedostatkům lidí a zájmu herní odvětví, jsou nuceni zaměstnávat lidi ze zahraničí. Vývoj her není ztrátou času, umožňuje vývojáři zdokonalit své programátorské dovednosti, které jsou na dnešním trhu práce velmi ceněné a do budoucna se očekává, že se informační technologie budou nadále rozšiřovat a pronikat do dalších odvětví.

První část bakalářské práce se zabývá historii vývoje počítačových her. V další části se pojednává o herním designu a jeho nejdůležitějších částech jako je hratelnost, herní mechaniky, designem levelů, nepřátel a generováním levelů. Tématu herního designu se věnuje Jesse Schell v knize The Art of Game Design A Book of Lenses [3], která se skládá ze samotných principů herního designu a level designu.

Třetí část pojednává o vybraných dostupných herních enginech a jejich vlastnostech. Dochází k jejich porovnání a na základě výsledku k výběru enginu pro vývoj 2D hry v praktické části.

Závěrečná část shrnuje praktickou stránku práce. Je zde implementován dokument herního designu, který slouží jako návrh hry. Jsou zde vysvětleny základní pojmy a princip fungování prvků Unity včetně příkladů, jak byli použity ve výsledné hře. Součástí práce je vytvoření 2D hry od samého začátku ve vybraném enginu a následně úprava hry podle obdržené zpětné vazby.

# Cíl práce a metodika

Cílem práce je představit proces vývoje her a napomoci tak lepší orientaci ve vývoji. Úkolem práce je popsat jednotlivé části herního vývoje. Zkoumá problematiku herního návrhu a prostoru, včetně návrhu a implementace dokumentu herního návrhu. Zabývá se také porovnáním populárních herních technologii jako je Unreal Engine, Godot Engine a Unity engine. Součástí práce je praktická část založena na vlastním reálném příkladu v podobě 2D hry, která zároveň tvoří praktický projekt této práce.

## Historie

První videohry byly vytvořeny v 50. letech 20. století. Rané videohry měly velký úspěch i přes to, že měly velmi jednoduchou nebo žádnou grafiku a zobrazovaly se na černobílých obrazovkách osciloskopu. Mezi první hry patří OXO, Spacewar! a Collosal Cave. V roce 1971 vytvořili budoucí zakladatelé Atari, Ted Dabney a Nolan Bushnell, první arkádovou hru Computer Space. Pro první hry se využívala vektorová grafika a později rastrová, což vedlo k ikonickým hrám jako Pac-Man a Donkey Kong. V 80. letech dominovaly herní scéně arkádové hry, avšak v 90. letech začalo odvětví arkádových her upadat, a nakonec koncerny zkrachovaly a mnoho a mnoho tehdejších konzolí skončilo v rukou sběratelů. [12] V té době vyšly The Legend of Zelda (1991), Super Mario World (1990) a Moral Kombat (1992). Ty se staly kultovními hrami a inspirovaly moderní videohry. Zároveň vyšli první kapesní herní konzole od Nintenda, The Game Boy. Další generací konzolí se stali konzole Sony PlayStation, Nintendo GameCube a Microsoft Xbox. V roce 2012 zažil velký rozkvět i průmysl v mobilních hrách díky hře Angry Birds, která vydělala přes 200 miliónů dolarů. [29] V dnešní době se nejprodávanější hrou všech dob stal Minecraft, sandbox (hráč si určuje sám cíl hry) hra vyvinuta společností Mojang, který následně koupila společnost Microsoft. Jeho hranaty 3D svět vedl k uznání kritiků. Minecraft byl vydán v roce 2011 a prodal více než 238 miliónů kopii. [27] V roce 2023 bylo téměř 170 miliónů aktivních hráčů měsíčně. [28] Další z novodobých populárních her je Grand Theft Auto (GTA), vytvořený Rockstar Games. Hra s kriminální tématikou a poprvé byla hra ze série vydána v roce 1997 a její značka je pořád velmi silná. [27]

# Herní design

Herní design je obor, který se zabývá návrhem herních prvků, jako jsou mechaniky, pravidla, prostředí a interakce, které jsou využívány k vytvoření hry pro zábavu nebo pro vzdělávání, cvičení nebo experimentování. Je to umění spojení designu a estetiky, které umožňují vytvořit hru, která bude pro hráče zajímavá a bude je bavit.[1] Za hlavní body herního designu lze určit hratelnost, mechaniky hry, level design a samotné budování hráčského prožitku, který je u her velmi důležitý.

## Dokument herního designu (GDD)

Game design dokument je slovní popis, podle kterého je vytvořena výsledná hra a každý jeden detail by měl být zaznamenán. Pokud vývoj hry probíhá ve větších týmech a je časově náročný, tím důležitějším se GDD stává. Do GDD se zapisují všechny možné informace, i ty které se do výsledného produktu neimplementují.

## Hratelnost

Hratelnost definuje mechaniky, pravidla a interakce, které budují zážitek z hraní počítačových her. Obsahuje elementy jako je ovládání, objekty zájmu, výzvy a zpětnou vazbu k hráči. Hratelnost je v herním designu kriticky důležitou součástí budování zážitku ze hry a tak, je na ní velmi často zaměřený vývoj a testování. Hratelnost můžeme rozdělit do několika principů.

### Principy směru

První tři zásady se zabývají vedením a usměrňováním hráčova zážitku. Ačkoli je toto médium z velké části založena na osobním interaktivním objevování, stále se jedná o uměleckou formu. Stejně jako obraz vede oko, kniha vede představivost a film vede vyprávění, musí i hra vést interaktivitu, jelikož hráčovu pozornost může zaujmout cokoli. [2]

Bod zájmu

Hráč by neměl být nucen hádat na co by se měl zaměřit, zároveň musí být hráči umožněno soustředit se i na jiné oblasti hry. Úkolem designera je stanovit primární zaměření.[2]

Očekávání

Hráč by měl být včas informovaný, že se něco stane. Při návrhu by měl být vždy zohledněné očekávání. Například včasným zvukovým efektem před příjezdem vlaku.[2]

Oznámení změn

Je nutné hráče informovat o změnách, tento krok probíhá mezi očekáváním a samotnou událostí. Krok se by se měl řídit stupněm vzácnosti, pokud se některá změna vyskytne stokrát za hodinu, nemusí být nutné ji oznamovat. Pokud dochází ke změně jednou za celou dobu herního zážitku, měla by být oznámena například řadou vizuálních upozornění.[2]

### Principy chování

Další čtyři principy pojednávají o velmi důležitém aspektu chování hry. Zabývají se očekáváním hráče, jak těmi vědomými, tak těmi nevědomými. Běžně se zde rozebírají aspekty návrhu jako jsou volby hráče, odměny apod. Principy chování lze aplikovat na další typy designu jako je uživatelské rozhraní nebo příběh hry. [2]

Přijatelnost událostí a chování

Každá událost, chování, akce, reakce, emoce a sdělení musí být v souladu s logikou a očekáváním hráče a musí vyhovět podvědomému testu přijatelnosti hráče. [2]

Překrývající se události a chování

Důležitým aspektem je nalézt správné množství událostí, které se mají vyskytnout v daném časovém okamžiku. [2]

Fyzika

Primární logika hráče se soustředí na známe principy fyziky, jako je gravitace, hmotnost, síla, pružnost atd. Principy fyziky by měly být použity jako základ, ale neměly by nutně být limitem. [2]

Zvuky

Zvuky jsou klíčovými prvky v designu her, protože pomáhají vytvořit atmosféru, umožňují hráči se lépe orientovat a dávají jim informace o dění ve hře. Zvuky také poskytují emoční vyjádření a podporují narativní zážitek.[2] Zvuková zpětná vazba má klíčový význam pro vytváření mentálního spojení mezi hráčem a zvuky, které jsou produkovány v reakci na akce hráče. Díky tomu hráč může vnímat zvuky jako své vlastní a vložit se do postavy a do světa hry. Kvalitní zvuková zpětná vazba umožňuje hráči okamžitou reakci na jeho akce a vytváří silnou citovou vazbu mezi hráčem a postavami v hře. Bez dostatečné zvukové zpětné vazby nebo s příliš velkým zpožděním může hráč ztratit zájem a nedokáže se do hry zcela vložit. [10]

### Princip prostředí

Je nutné rozeznat vztahy mezi prvky, uvědomit si kolik je prostoru na obrazovce a zvážit dopady změn. Při velkém počtu objektu na obrazovce může dojít k chybnému pohybu a celkově ke špatnému dojmu ze hry. [2]

## Herní mechanismy

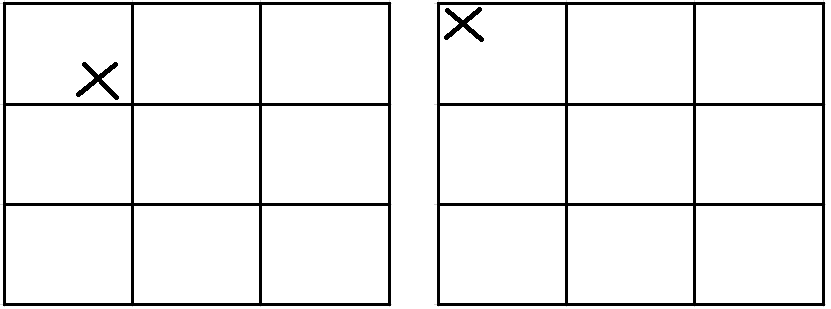
Herní mechanismy jsou základem her, jsou to interakce a vztahy, které zůstávají, když se odstraní všechna estetika, technologie a příběh. I když existuje mnoho jiných prvků v herním designu, není pro mechaniky obecně uznávaná taxonomie. Důvodem je, že herní mechaniky, i u jednoduchých her, jsou poměrně složité a těžko rozebíratelné. Pokusy o zjednodušení těchto složitých mechanik až k dokonalosti jsou velmi náročné a vedou k matematickému pochopení, které je neúplné. [3]

Nejsnadnější způsob, jak pochopit co to jsou herní mechaniky je na příkladu jednoduché hry jako je Tetris. Jednu z nejoblíbenějších a nejznámějších her všech dob. Tetris lze rozdělit na čtyři hlavní herní mechanismy.[4]

1. Systém otáčení: Určuje, v jaké pozici se tetromina (Geometrický útvar složený ze čtyř čtverců) objevují a otáčejí.
2. Náhodný výběr: Pořadí, v jakém se typy tetromin objevují.
3. Systém bodování: Vyčištěné řady získávají hráči body
4. Pohyb: Mechanika umožňující hráči měnit polohu tetromina, včetně rotace.

### Prostor

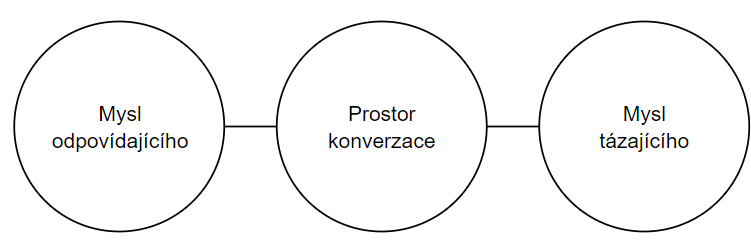
Většina her se odehrává v prostoru definovaném hrou, který lze dále rozdělit na diskrétní a kontinuální. Příkladem diskrétního prostoru může být desková hra jako jsou piškvorky. Každý bod uvnitř čtverce má stejnou hodnotu, zajímá nás pouze hranice.



Obrázek Diskrétní prostor piškvorek, převzato [3]

V FPS hrách se však s velkou pravděpodobností bude použit kontinuální prostor. To, kde postava v souvislém prostoru stojí nebo je v pohybu, bude mít vliv na výstřely, které postava vypálí na protivníka. Ve spojitém prostoru mají malé rozdíly v umístění velký vliv na to, jak se hraje, což platí zejména pro hry v kontinuálním prostoru jako je kulečník. Je důležité také přemýšlet o souvislostech mezi různými prostory ve hře. Příklad šachovnice, kde je určené, jak jsou políčka propojena, ale méně zřejmé je, že některá políčka jsou propojena různými způsoby, pokud se figurky mohou pohybovat různými způsoby. Pěšec se pohybuje dopředu, ale útočí pouze po diagonále. Střelec se pohybuje po diagonálách, věže po řadách a sloupcích. [3] Prostor pro hry, lze rozdělit i podle dimenzí. 2D hry mají jednoduchou grafiku bez třetího rozměru. Na obrazovce budou ploché, bez perspektivy a umožňují pouze pohyb do čtyř stran. U 3D prostoru jde o možnost interaktivity s trojrozměrnou grafikou, lze otáčet o 360 stupňů a prohlížet z libovolného úhlu. Díky 3D lze vymodelovat realistický zážitek ve Virtuálním světě. [37]

Některé prostory ve hrách jsou také vnořené, stejně jako jsou místnosti v budovách. Prostor může být také virtuální ve smyslu, že ve hře vůbec neexistuje. Takové hry jsou triviální a příkladem může být hra „20 otázek“ kde lze mít za to, že využívá tři virtuální prostory. Prostor tazatele, kde promýšlí otázku a odpověď. Prostor odpovídajícího, kde vytváří odpovědi a herní prostor, kde se kladou a odpovídá na otázky.



Obrázek Virtuální prostory pro hru 20 otázek, převzato [3], přeloženo

Při navržení prostoru hry je tedy nutné odpovědět na otázky, jestli se jedná o diskrétní nebo kontinuální prostor, jaké jsou hranice, jestli se zde nachází vnořené prostory a jak jsou prostory propojeny.[3]

### Objekty, atributy a stav

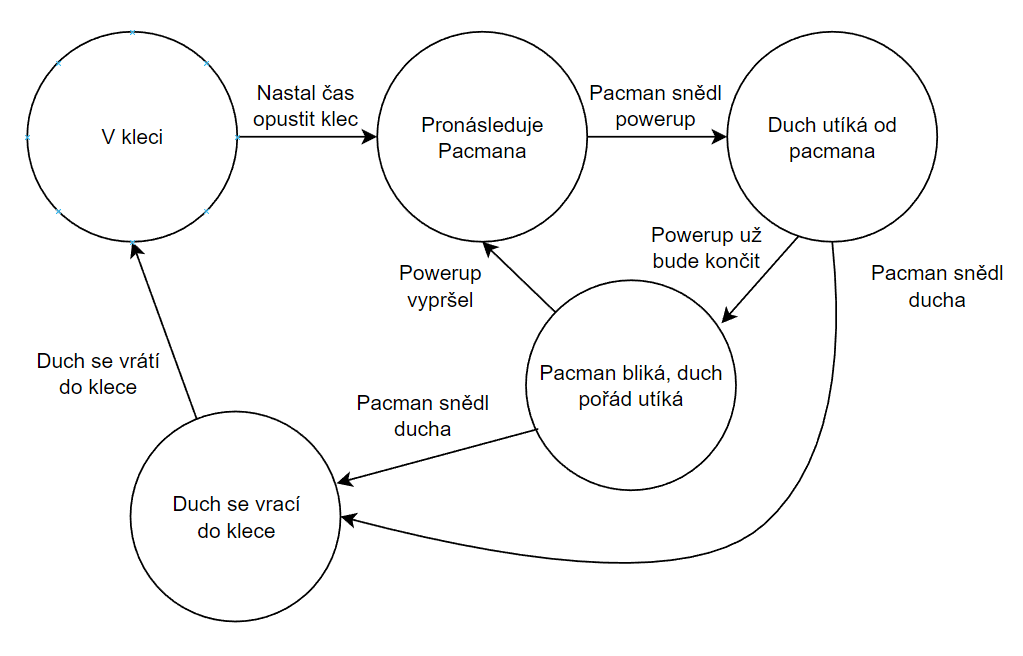
Objekty obsažené ve hře jsou další způsob, jak sledovat informace. V šachu mají figurky různé vlastnosti, které se mohou v průběhu hry měnit. Například šachová figurka může mít vlastnost „pohyb“ a ta může být buď „volná“ nebo „omezená“. Některé figurky mají vlastnosti, které ostatní nemají. Například figurka jako je věž může mít vlastnost „směr“ a „vzdálenost“, jejichž hodnoty jsou spojené a mohou se během hry měnit.

Je důležité však správně rozhodnout, které informace o objektech by měly být přístupné pro všechny, pouze určitým hráčům nebo skupině. Zároveň herní informace mohou být v průběhu hry odhaleny ostatním hráčům jako tomu je při hře „Lodě“ kde dochází k postupnému odhalování herní mapy protihráče.

Informace podle [3] mohou být

* Zcela veřejné – Informace jsou dostupné pro všechny hráče jako v šachu.
* Sdílené mezi více hráči – hráči si mohou sdílet své informace s vybranými hráči – Age of empire uzavírání příměří, obchodní smlouvy
* Soukromé – hráčovy karty v pokeru
* Soukromé pro hru – například hádání hesla, hra zná správnou odpověď, ale hráč jí musí uhádnout, nebo domyslet.
* Náhodná informace – stav známých předmětů není určen, například míchání karet v balíčku. Známe karty, ale nevíme, kde se nacházejí.

Při návrhu objektů tedy musíme uvážit, jaké máme objekty ve hře a jejich vlastnosti, možné stavy, v kterých případech dochází ke změně atributů a kdo bych o těchto informacích měl vědět.[3]



Obrázek Informace o stavech, převzato [3] přeloženo

### Akce

Jako další důležité mechaniky hry jsou akce. Existují dva způsoby, jak lze odpovědět na otázku „Co mohou hráči dělat?“ První z nich jsou operativní akce, což jsou základní akce, které mohou hráči provést v rámci hry. Například v dámě mohou hráči provést pouze tři základní operace jako je posun dopředu, přeskočit soupeřovu dámu a v případě krále přesunout dozadu. Dalším druhem akce jsou výsledné akce, které jsou důležité zejména ve větším kontextu hry a souvisejí s tím, jak hráč využívá operativní akce, aby dosáhl cíle. Seznam výsledných akcí je obvykle delší než seznam operativních akcí a může zahrnovat například ochranu dámy před zajetím, donutit soupeře k nechtěnému skoku nebo obětování dámy k oklamání soupeře. Hrát bez záměru je jako náhodné umisťovat značky v piškvorkách, je to sice možné, ale pravděpodobně to nebude úspěšné ani zábavné. Jde o zábavu, je tedy jasné, že hra by měla stimulovat mysl hráče, aby mohly vznikat výsledné akce. To nás vede k poznání, že by hra měla být přizpůsobena tak, aby vytvářela emergentní akce, tedy akce, které jsou zajímavé díky svým dopadům na hru. Zmíněné akce se nazývají emergentní, protože nejsou vyžadovány pravidly, ale jsou umožněny, jelikož je hráč může shledat zajímavými, zábavnými nebo prospěšnými.

Přidáním více operativních akcí můžeme značně zvýšit možnost výskytu emergentních akcí.

Značnou zábavnost do hry přidávají i cíle, které mohou být dosaženy více různými cestami, to umožní hráči dosáhnout cíle svojí preferovanou cestou a zároveň zvýší šanci nalezení nových cest a zvýší možnost, že si hráč bude chtít hru zahrát znovu jinou cestou.

Další možností je přidání více předmětů/objektů na které se lze zaměřit. V případě her s více charaktery je předmět postava. Více charakterů vede k více druhům interakcí, což vede k více emergentním akcím. Proto je hraní her s jedním charakterem odlišné od hraní se skupinou charakterů. Emergentní hru lze ovlivnit i efekty, které plynou z vedlejších interakcí hráče. Pokud hráč provede některou z vedlejších interakcí, může to mít vliv na omezení hry, herního prostoru nebo možné akce. [3]

U návrhu akcí, aby hra byla co nejvíce zábavná je důležité, kolik má hráč možných akcí, kolik objektu akce ovlivní, počet možným způsobům, jak může hráč dosáhnout cíle, jaké objekty může hráč ovládat a jakým způsobem ovlivňují vedlejší akce svět.

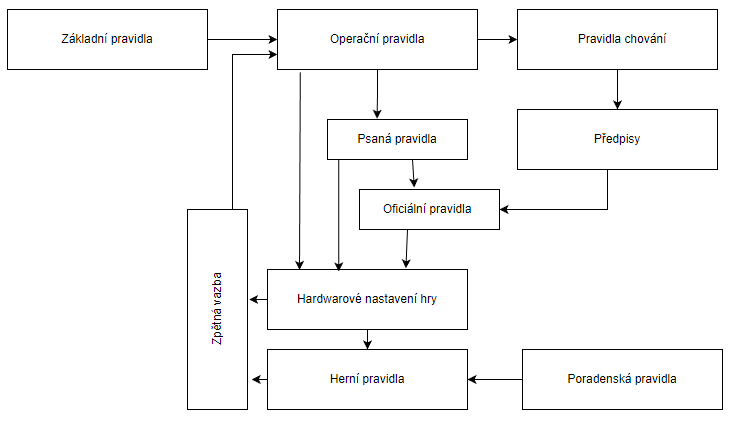
Důvodem proč se hry podobají jedna druhé, je kvůli používání stejného souboru akcí. Hry, které jsou považovány za odvozené, mají stejný soubor akcí jako starší hry. Naopak hry, které se označují jako inovativní, poskytují hráčům nové druhy akcí. Například hra Donkey Kong byla v té době nová kvůli běhání a skákání. Akce, které hráč může provádět, jsou klíčové pro definování herních mechanismů a změnou jedné akce lze získat úplně odlišnou hru. Velké množství designerů touží po hrách, kde by hráči mohli provádět libovolné akce. Některé masivně multiplayerové hry se začínají vydávat tímto směrem a nabízejí širokou škálu akcí pro boj, vytváření předmětů a sociální interakce. V 70-80. letech minulého století byly velmi populární textové adventury, které obvykle obsahovaly desítky nebo stovky možných akcí, ale s nástupem vizuálnějších her se počet možných akcí náhle snížil, protože nebylo možné všechny podporovat vizualizací.

Při návrhu akcí je tedy nutné promyslet jaké akce lze a nelze dělat a proč. Jaké operativní a výsledné akce jsou k dispozici, jaké výsledné akce by designer rád viděl a jak by mohl hru upravit, aby byly možné a které hry by hráči pravděpodobně uvítali, popřípadě jestli je možné tyto akce poskytnout jako operativní nebo výsledné.[3]

### Pravidla

Pravidla jsou nejzákladnějším mechanismem. Definují prostor, objekty, akce, následky akcí, omezení akcí a cíle. Přidávají klíčové věci, která činí hru skutečnou hrou – cíle.[3]

Pravidla digitálních her nejsou stejná jako samotný programový kód, který hru tvoří. Pravidla jsou abstraktními nástroji, které slouží k pochopení formálního uspořádání hry a nemusí být nutně přímo vyjádřena v programovém kódu. [1]



Obrázek Diagram pravidel převzato [3], přeloženo

**Operační pravidla:** Operační pravidla videoher nejsou zaměřena pouze na vnitřní události, ale také na vnější události hry, jako jsou vstupy hráče a výstupy hry, které vyjadřují volby a výsledky pro hráče. Stejně jako u nedigitálních her existuje nejasná hranice mezi operačními a implicitními pravidly digitálních her. [1] Operační pravidla jsou snadno pochopitelná. Jsou to základně "co hráči dělají, aby mohli hrát hru". Když hráči pochopí operační pravidla, mohou hru hrát.[3]

**Základní pravidla:** Základní formální struktura hry je definována základními pravidly. Základní pravidla jsou abstraktnější a popisují matematické vyjádření stavu hry a jak a kdy se mění. Provozní pravidla jsou na druhou stranu přímočařejší a popisují, jaké akce musí hráči udělat, aby hru mohli hrát. Operační pravidlo může například stanovit, že hráč by měl hodit kostkou a nasbírat body, zatímco základní pravidlo může stanovit, že hodnota síly hráče se zvýší o náhodné číslo od 1 do 6. Základní pravidla ovlivňují operační. V současnosti však neexistuje standardní způsob, jak tato pravidla reprezentovat, a není jasné, zda je vůbec možné pro ně vytvořit úplný zápis.[3]

**Pravidla chování:** Pravidla chování jsou silně spjata s hratelností a většina lidí je chápe jako součást „dobrého sportovního chování“. Například při hře šachu by hráč neměl svého soupeře rozptylovat při tahu, nebo svůj tah zbytečně prodlužovat. Pravidla chování se zřídka musí explicitně stanovit, většinou je všichni znají. Fakt, že existují, podtrhuje bod, že hra je druh imaginární dohody mezi hráči. [3] Pravidlům chování se ve své eseji Unwritten rules věnuje Steven Sniderman.

**Psaná pravidla:** jsou pravidla, která přicházejí s hrou, dokument, který hráči musí přečíst, aby získali povědomí o operativních pravidlech. Samozřejmě, ve skutečnosti pouze malé množství lidí tento dokument opravdu čte, většina lidí se seznámí s pravidly hry během samotného hraní. Je velmi těžké zapsat ne-lineární složitosti, jak se hraje, do dokumentu a stejně těžké pochopit takový dokument. Moderní videohry postupně opouštějí písemná pravidla ve prospěch toho, aby samotná hra učila hráče, jak hrát prostřednictvím interaktivních tutoriálů. Praktický přístup je mnohem účinnější, i když může být těžký a časově náročný na navržení a implementaci, protože to zahrnuje mnoho iterací, které nelze dokončit, dokud hra není v jejím konečném stavu. Každý designér hry musí mít připravenou odpověď na otázku: "Jak se budou hráči učit hrát moji hru?" Protože pokud někdo nerozumí vaší hře, tak si ji nezahraje.[3]

**Předpisy**: Vytváří se v případě, že hry jsou hrané v seriózních a kompetitivních prostředích a je tedy nutné vytvořit nebo upřesnit pravidla sportovního chování. Předpisy se také nazývají *Turnajová pravidla*. [3] Příkladem takové hry může být MOBA League of Legends, která se pyšní stále větší a větší oblíbeností. Pravidla mohou pro účastníky upravovat:

* Minimální věk.
* Periferie, které mohou používat.
* Jestli některá z herních postav není zakázána
* Jestli mohou vysílat svůj obraz (streamovat)
* Kolik času mají na jednotlivé úkony (přihlášení, potvrzení účasti, výběr vybavení…)

**Oficiální pravidla**: Jsou to pravidla, která vznikla na potřebě hráčů spojit psaná pravidla s předpisy. Například v šachu, když hráč ohrozí nepřátelského krále, musí svého soupeře upozornit na tuto skutečnost slovem „šach“. Dříve toto pravidlo bylo v předpisech, ale nyní je součástí oficiálních pravidlech. [3]

**Poradenské pravidla:** Poradenská pravidla: často nazývána "pravidla strategie", jedná se pouze o tipy, jak lépe hrát, a ve skutečnosti nejsou z pohledu herního mechanismu opravdu "pravidly". [3]

**House (vlastní) pravidla:** Pravidla, které hráči tvoří, aby byla hra zábavnější (z jejich pohledu). Například ve hře monopoly pokaždé co hráč projde přes políčko parkoviště, vloží na něj peníze. Hráč, který pak následně na toto políčko vstoupí, může peníze vybrat. [3] Upravy oficiálních pravidel hry, a to buď vzájemnou dohodou mezi hráči nebo jedním hráčem nebo skupinou hráčů v zájmu spravedlnosti, zábavy nebo rozmanitosti. [5]

**Nejdůležitější pravidlo:** Základem všech pravidel ve hře je Objekt hry, což je v podstatě účel nebo cíl hry. Je důležité jasně stanovit cíl hry, a pokud je cílů více, jak spolu souvisí. Špatně stanovený cíl může být pro hráče matoucí a bránit jim v radosti ze hry. Například při pokusu o vysvětlení cíle šachu nováčkovi může být vysvětlování pojmu šach a mat komplikované a nesrozumitelné. Skutečným cílem šachů je jednoduše zajmout soupeřova krále, což je jednoduchý a přímočarý koncept. Když hráči jasně chápou cíl hry, je pravděpodobnější, že si jeho dosažení představí a budou motivováni ke hře. [3]

Dobré herní cíle mají tři důležité vlastnosti:

* **Jednoznačnost:** Hráči jasně vědí, čeho mají dosáhnout
* **Dosažitelnost:** Hráči musí vědět, že mají šanci dosáhnout cíle. Pokud se jim to zdá nemožné, rychle to vzdají
* **Odměňování:** Proces dosažení cíle by měl být pro hráče odměňující a samotný cíl by měl být tak odměňující, jako byl hráči prezentován hrou. Čím těžší bylo dosažení cíle, tím větší by měla být odměna.

Při návrhu cílů je žádoucí zaměřit se na aspekty jako jsou hlavní cíl hry, jestli je cíl pochopen hráčem. Pokud existuje sekvence cílů, rozumí jim hráč? Jsou na sebe navázané cíle? Je u cílů jednoznačnost, dosažitelnost a jsou dostatečně odměňovány?

### Dovednosti

Mechanika dovedností přesouvá pozornost ze samotné hry na hráče. Pro jakoukoli hru musí hráči používat specifické schopnosti, a pokud jejich úroveň dovedností odpovídá obtížnosti hry, zažijí pocit výzvy a budou hrát dále. Hry obvykle vyžadují od hráčů kombinaci různých dovedností a při vývoji hry je užitečné vytvořit katalog požadovaných dovedností. Tyto dovednosti mohou být rozděleny podle autorů [3, 11, 12] do několika kategorií:

* Fyzické dovednosti – zahrnují schopnosti jako síla, koordinace, obratnost a výdrž. Tyto dovednosti jsou důležité v akčních hrách, sportovních hrách a v simulátorech.
* Kognitivní dovednosti – zahrnují schopnosti, jako je pozornost, paměť, rozhodování a problémové řešení. Tyto dovednosti jsou důležité v logických hrách, strategiích a simulacích.
* Sociální dovednosti – zahrnují schopnosti jako spolupráce, týmová práce a komunikace s ostatními hráči. Tyto dovednosti jsou důležité v multiplayerových hrách a hrách s kooperativním módem.
* Technické dovednosti – zahrnují schopnosti, jako je ovládání ovladače, znalost uživatelského rozhraní a schopnost používat herní nástroje. Tyto dovednosti jsou důležité v různých typech her a mohou být specifické pro konkrétní platformy.

**Reálné a virtuální dovednosti**

Reálné dovednosti jsou skutečné schopnosti, které musí hráč skutečně ovládat, aby byl úspěšný v hře. Tyto dovednosti jsou nezávislé na herním systému a zahrnují koordinaci pohybů, rychlé reakce na situace na hřišti a přesnou techniku. Na druhé straně jsou virtuální dovednosti definovány samotnou hrou a jsou spojeny s herním systémem. Tyto dovednosti mohou být založeny na reálných dovednostech, jako je koordinace pohybů v rychlé akční hře, ale mohou také být úplně virtuální, jako je schopnost ovládat určitý herní nástroj nebo znát konkrétní herní taktiky. Aby byla hra úspěšná, musí být schopna propojit reálné a virtuální dovednosti takovým způsobem, aby hráči měli pocit, že se dovednosti navzájem doplňují a že jsou v jeho kontrole. Hry by neměly být založeny pouze na virtuálních dovednostech, protože by to mohlo hráče odradit, když by se snažili zlepšit své skutečné schopnosti, které by mohly být potenciálně užitečné i mimo herní kontext. [11]

### Šance

Poslední z mechanik je šance. Prvky náhody do hry přináší nepředvídatelnost, čímž ji tvoří více zábavnou. Využívá se pro určování výsledků určitých událostí, jako například hod kostkou, rozhodnutí souboje, úspěšnost střelby nebo přesnost kouzla. Šance ve hře vytváří napětí a umožňuje hráči zažít různé výsledky v závislosti na jejich schopnostech. [3]

## Level design

Level design je proces vytváření herního světa nebo jednotlivých levelů tak, aby byly plné detailů, ale stále měli jasný cíl hry. [12] Level musí mít přesně určený účel, aby hráči mohli snadno rozpoznat, co se od nich očekává, a aby se cítili odměněni po dosažení cíle. [13] Dále musí být věrohodný, logicky uspořádaný a aby hráči měli pocit, že se mohou svobodně pohybovat. [3]

**Konceptuální návrh**

Návrh základního konceptu a plánu pro level, včetně jeho cílů, vlastností, prvků, vztahů a atmosféry. Návrh slouží jako plán pro tvůrce hry, aby získali představu o tom, jaký bude konečný produkt a usnadnil práci na levelu. [14, 17]

**Tvorba prostoru**

Vytváření herního prostoru musí splňovat požadavky na hratelnost dané hry, a zároveň musí být příjemné pro hráče. Tvorba prostoru přímo vychází z konceptuálního návrhu a přizpůsobuje se technickým omezením herního enginu. Vytvořené prostředí musí být intuitivní a přirozené pro hráče, aby se mohl plně ponořit do hry. [12, 16]

**Design cesty**

Design cesty je tvorba plynulého toku hry prostřednictvím umístění herních objektů a prostorového uspořádání tak, aby postup levelem byl hráči přirozený a snadný. Význam designu cesty je udržet hráče motivovaného a zaměstnaného, aniž by se cítil zmatený nebo ztracen. [12]

**Design herních objektů**

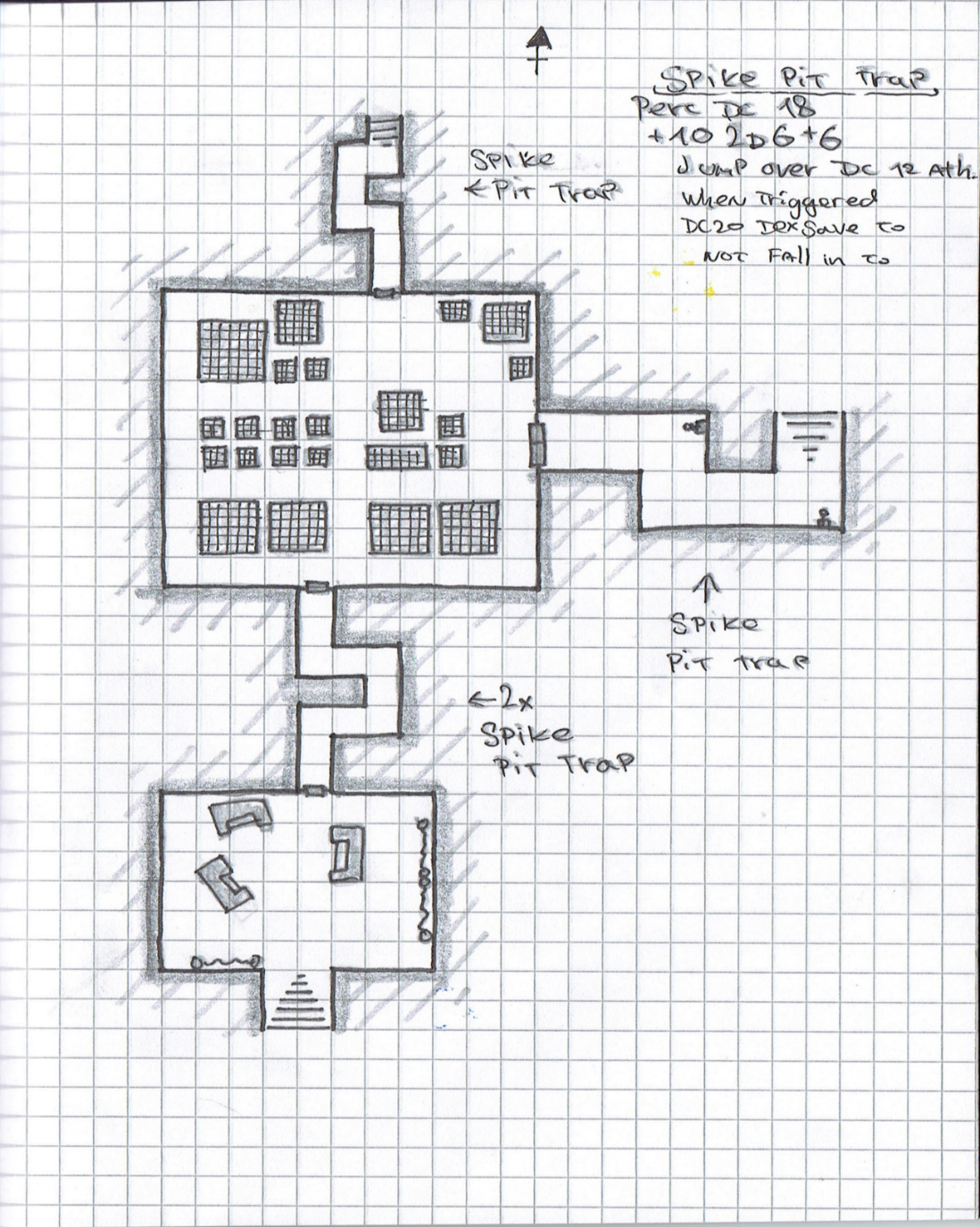
Proces vytváření předmětů, zbraní, překážek a dalších objektů v herním prostředí, který vychází z konceptuálního návrhu. Každý z objektů může být své vlastnosti jako barvu, velikost, interakční možnost, váhu apod. [3, 12]

**Design nepřátel a bossů**

Tvorba nepřátel a bossu je důležitou součástí level designu. Tvorba nepřátel a bossů navazuje na konceptuální návrh a jsou zde stanoveny základní charakteristiky jako je vhodný výběr nepřítele pro level, síla, rychlost, útoky a počet nutných zásahů pro poražení nepřítele. Design bossů je proces vytváření nejsilnějších a nejzajímavějších protivníků, kteří jsou často zařazováni mezi nejzajímavější momenty hry. [12, 15]

Při tvorbě levelu je dobré zaměřit se na všechny užitečné prostředky, které při tvorbě světa mohou pomoci. Postava, akce postavy, děj, témata úrovní a mapa světa. Pro tvorbu celého vesmíru hry jsou tyto prvky nesmírně důležité.

Pro návrh levelu se používá několik různých způsobů, jak daný level zobrazit. Ať už pomocí lega, rychlé prototypování v 3D programech jako je Maya nebo 3D Studio Max a Blender. Nebo pomocí obyčejné tužky a prázdného papíru. (Michael Kraveckyj, 2023)



Obrázek Plánek podzemních sklepů do hry DND, Michael Kraveckyj

### Rozdělení typu level designu

**Alleys** neboli uličky jsou navrženy tak, aby vytvářely soustředěný herní zážitek. Úroveň je vytvořena, aby hráče vedla ke konkrétnímu cílí, kterého má dosáhnout. Alleys nemají určenou velikost. Mohou být úzké, podobně jako ve hře Portál, nebo široké, aby hráč získal pocit svobody, což se uplatňuje například ve hře Call of Duty: Modern Warfate 2. [12]

Designeři používají alleys, jelikož poskytují podle [12] hned několik výhod:

* Jelikož je přesně daný směr, kudy se hráč musí vydat, je mnohem jednodušší umístění spouštěcích zón pro kameru.
* S kamerou lze lépe využít dramatické pohyby, aby byl hráč informován nebo se zdůraznila akce a dramatické momenty.
* V levelu lze odstranit ovládání kamery, takže se hráč bude moci soustředit pouze na ovládání a hratelnost.
* Pomocí znalosti, kde se hráč nachází a kam kouká je možné předem definovaná události.
* Je jednodušší vést boj a další herní události, jako jsou pasti.
* Lze vytvářet úzká místa (tzv. „bottleneck“), které zabrání hráči se vracet do již prozkoumaných míst.
* Návrhář může využít iluzorní narativ (umístění dekorací, zvukových efektů, osvětlení atd.) k vyprávění příběhu levelu.

Obsah obrázku mapa

Popis byl vytvořen automaticky

Obrázek „Mysterious tunnels“, Příklad „Alley“ level designu, Michael Kraveckyj

**Islands** neboli ostrovy jsou z pohledu level designu jedny z náročnějších. [12] Islands se velmi často využívají ve spojení s hrami typu „battle-royal“ nebo hry s otevřeným světem. (Michael Kraveckyj, 2023) Kamera ve hře musí být kvůli výškovým rozdílům dostatečně flexibilní. Události, které jsou skriptování, jsou těžší na provedení, protože hráč do oblastí vůbec nemusí přijít. Situace, kdy má dojít k boji mohou být obcházeny. [12] Některé části mapy mohou obsahovat úseky typu alleys. [18] Díky otevřenosti prostoru poskytuje hráčům možnost zvolit si svojí vlastní cestu. Jedním z prvních příkladů designu typu island je Mario 64, kde hráč může objevovat prostředí v jakémkoli pořadí. Island design dovoluje svobodu v hratelnosti a vyzdvihuje žánr známý jako „sandbox“. [12]

Výhody typu ostrovy podle [12] jsou:

* Levely poskytují pocit prostoru a rozsahu.
* Ostrovy podporují průzkum a podněcují designéry k výplni volných míst tajnými místy, dodatečnými úkoly.
* Herní mechaniky spojené s vozidly (závody, souboje s vozidly) se v širokém otevřeném světě designují mnohem lépe.

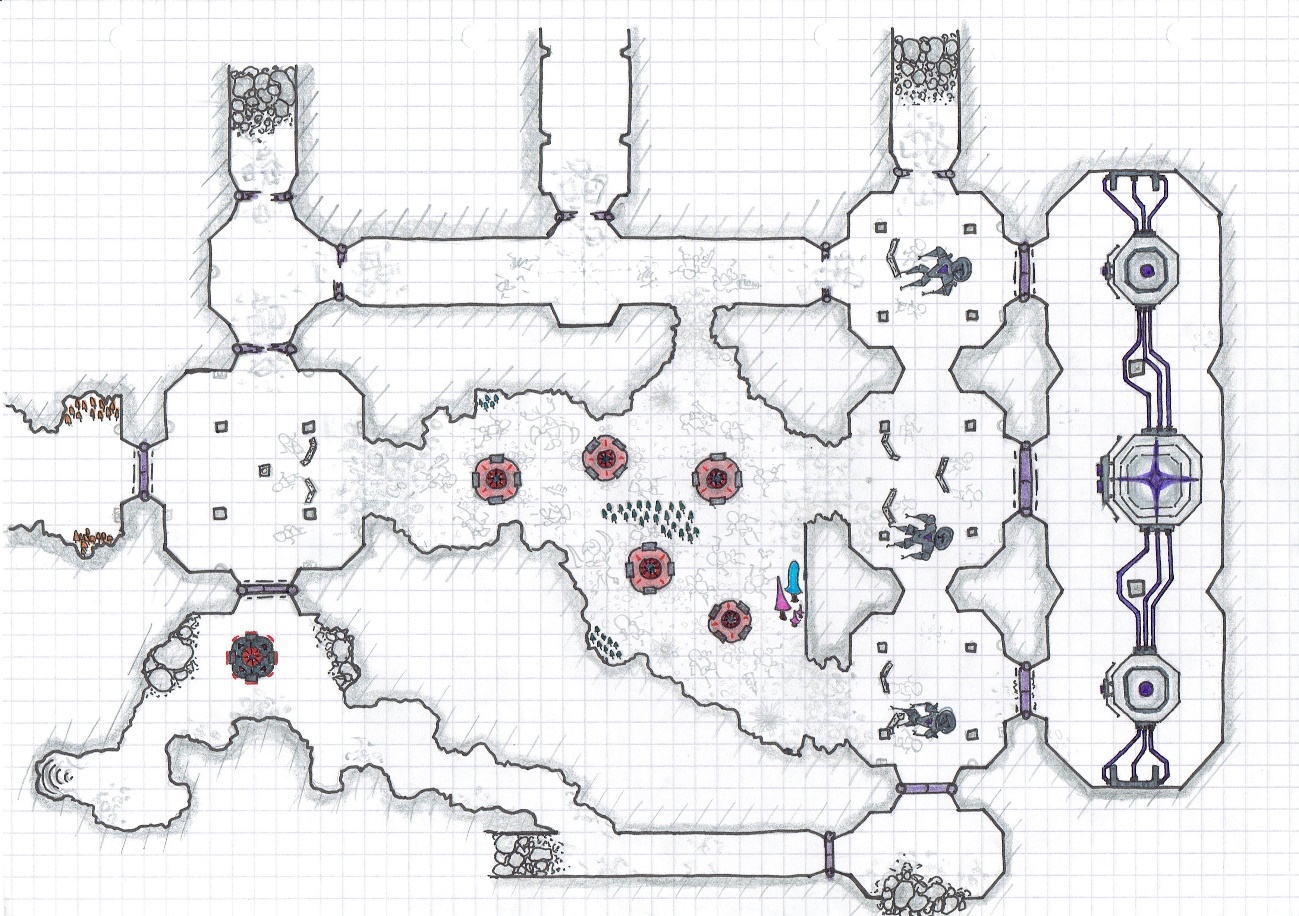
Obsah obrázku mapa, text, perokresba

Popis byl vytvořen automaticky

Obrázek „Velvevelkyr“, Příklad „Island“ level designu, Michael Kraveckyj

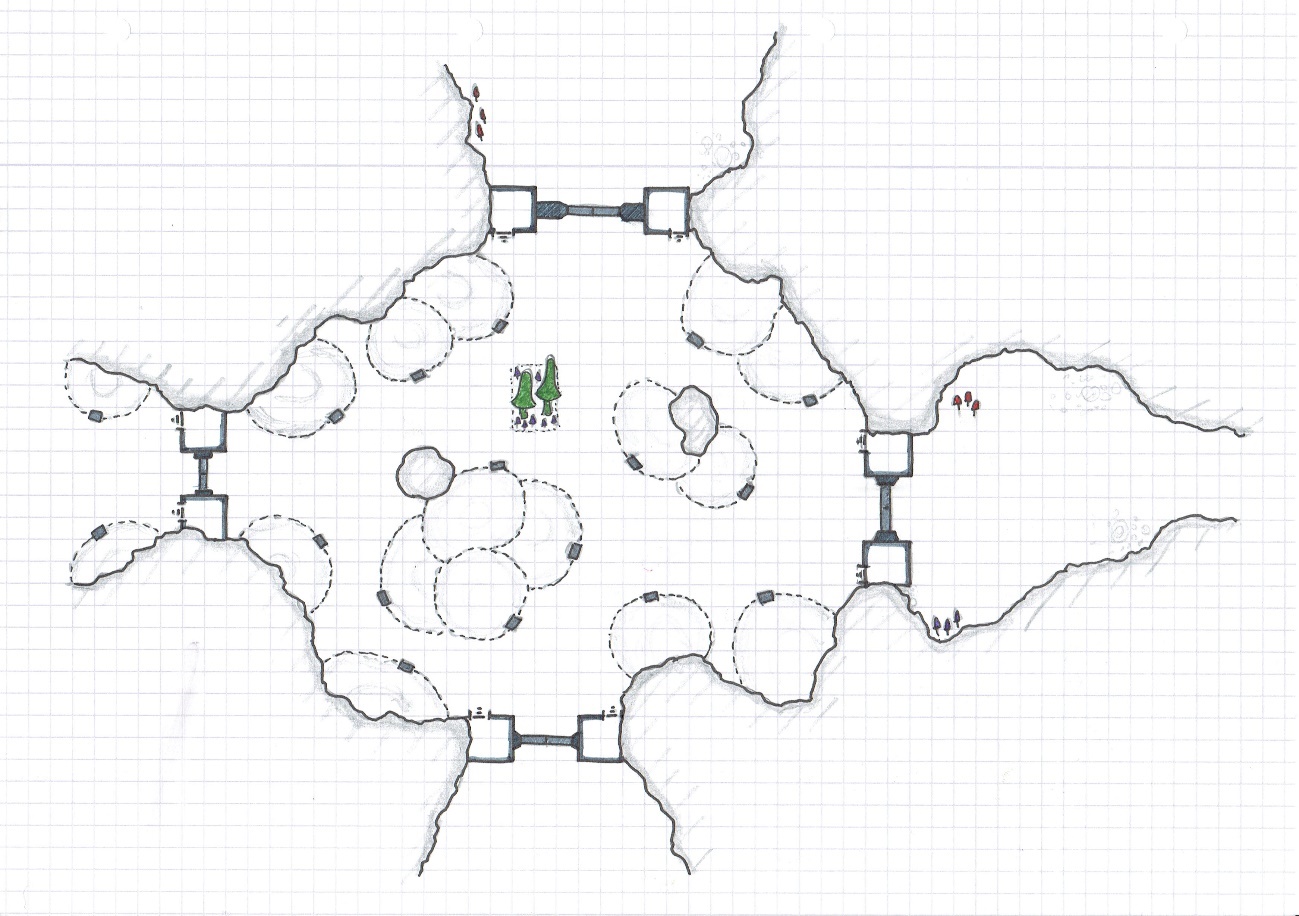
**Maze** (Bludiště) může být zajímavou designovou volbou. Umožňuje umístění skrytých objektů ve hře, může způsobit dezorientaci hráče a tím způsobit, že hráči uniknou některé poklady. Bludiště zároveň lze využít jako prostor, kde hráč musí najít nějaký předmět, či například utéct před nepřítelem, aby se mohl posunout v příběhu. Jednou z výhod může být odměna hráči, za objevování nových, neprozkoumaných oblastí. Na druhou stranu, hráč se může ztratit a postupem času jeho chuť hrát upadne. Příkladem hry, která využívá typ bludiště je Pac-Man. [18]

**Path** (cesta) je styl, kde má hráč možnost volby, kam chce jít, nabízí se mu možnosti, které ale nemusí využít. Příkladem hry, kde máš hráč možnost volby, kam se vydat, je Detroit Become Human. Výhodou stylu Cesta je, že na rozdíl od stylu alley nabízí větší svobodu v tom, kam může hráč jít nebo co může dělat. Nevýhodou stylu je držení hráče v jednom směru, protože existuje omezená oblast, kam se hráč může dostat. [18]



Obrázek Příklad „Path“ level designu, Michael Kraveckyj

**Hub** (rozbočovač) poskytuje hráči možnost se pohybovat v centralizované oblasti, ze které se lze vydat několika směry podle rozložení levelu. Hlavní výhodou je, že když se hráč ztratí, může se vydat zpět na centrální místo, které zároveň slouží jako bezpečné místo, ve kterém mu nic nehrozí a tím mít i představu o tom, kde se zrovna nachází. Nevýhoda hubu je, že hráči jsou omezeni na centrální místo. [18]



Obrázek Příklad „Hub“ Level designu, Michael Kraveckyj

### Level flow

V game designu se pod pojmem "flow" rozumí, jak se hráč cítí při pohybu po různých cestách nebo částech levelu. Zda je cesta jednoduchá nebo složitá, rovná nebo křivolaká, pomalá nebo rychlá – všechny tyto faktory ovlivňují hráčův pohyb prostředím. Navrhování toku hry tedy znamená navrhovat pohyb hráče. S návrhem toku by se mělo začít již během plánování dispozice levelu, ale jak skutečně působí se dá ověřit až po tvorbě hrubého náčrtu a testování. [19] Dobrý tok levelu je pro hráče atraktivní. Snadno se v nich orientují a/nebo nabízejí výhody jako je krytí, zbraně apod. Levely se špatným tokem se vyznačují hlavně špatným pohybem po levelu kvůli špatnému návrhu designu mapy, nenabízejí žádné výhody, v některých případech dokonce způsobují spíše nevýhody a hráč se jim obecně vyhýbá. [20]

**Design toku hry**

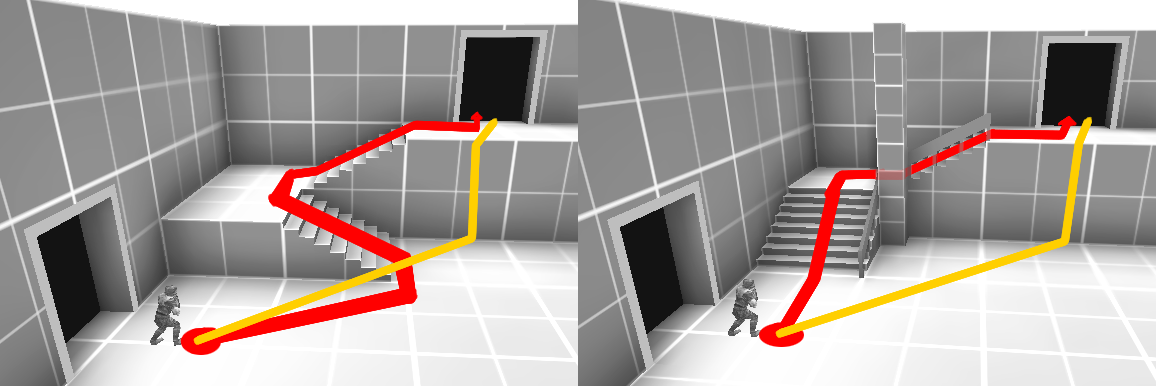
Tok hry ovlivňuje několik faktorů

* Rychlost – Pohyb po cestě nesmí být příliš rychlí, ale ani pomalý.
* Směr – Trasa může být navazující nebo se rozdělovat či „kroutit“.
* Jednoznačnost cesty – Cesty by měly být jednoznačné a hráč by měl dostávat jednoduché náznaky, jak už bylo několikrát naznačeno v práci, kudy se vydat.
* Metrika – Mechaniky pohybu by měly být přizpůsobeny celkové velikosti levelu.

**Desire lines**

„Desire lines“ nebo také „Desire paths“ jsou ideální cesty, které vytvářejí samotní lidé. Jsou vyznačené pomocí pěšího provozu. Příkladem může být vyšlapaná cesta v parku, kde oficiální cesta je vybetonovaný chodník a „desire path“ je cesta, kterou lidé opravdu chtějí a je pro ně nejvíce přirozená a intuitivní.

Desire line jsou vhodné pro design v malém měřítku, například v rámci místnosti. [19]



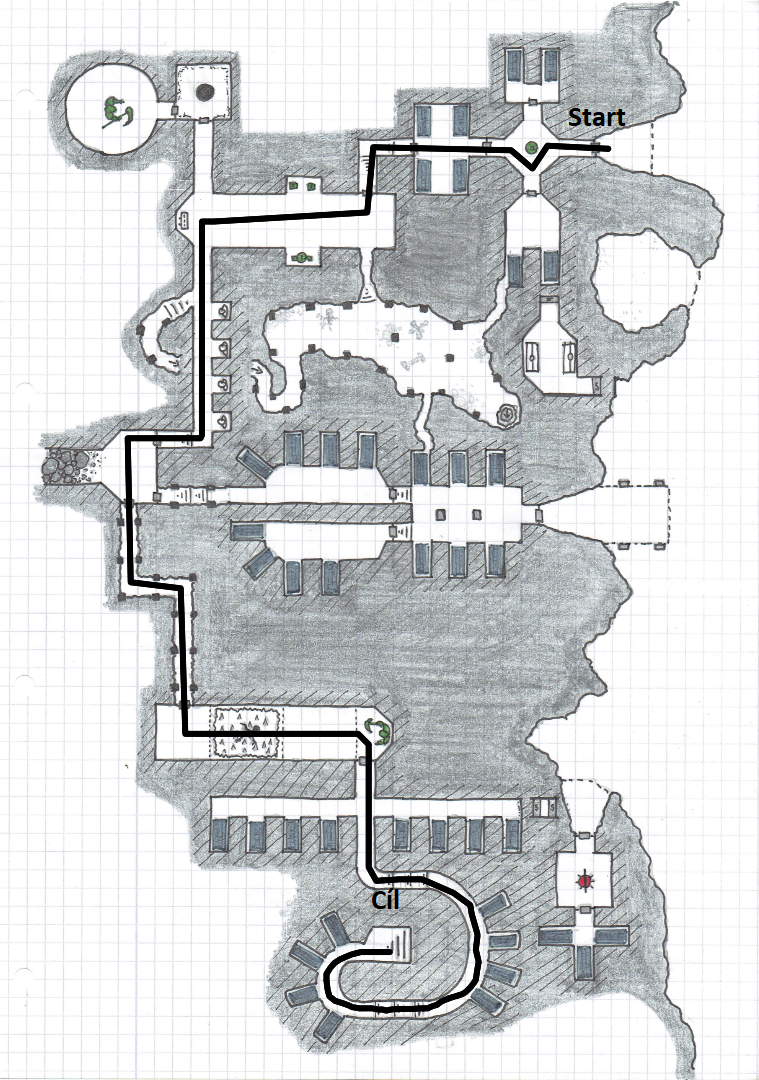
Obrázek porovnání dvou cest [19], Andrew Yoder

Hráčova vytoužená cesta (žlutá) vede přímo na konec levelu v druhém patře, ale oficiální cesta (červená) nutí hráče jít po schodech. Schodiště se zákrutami (vlevo) působí méně přímo, protože při cestě je nutné extra otočení oproti cestě s jedním zalomením (vpravo). I přes to, že se cesta s méně zákrutami může zdát efektivnější, flow nemusí být nutně horší. Schodiště s více zákrutami může být užitečné podle [19] pro:

* Podporuje hráče v prozkoumávání místnosti.
* Může se nabídnout možnost parkour a tím si cestu zkrátit.
* Umístění nepřátel na mezi patro, kde se jim hráč hůře ubrání.

**Critical path**

Critical path, také známa jako „Zlatá cesta“ je nejkratší možná cesta k dokončení levelu. Zjednodušeně se jedná o cestu pro hráče, která poukazuje na základní části úrovně, kudy se každý hráč musí vydat. [19] K cestám, které jsou součástí zlaté cesty, by se vývojáři měli chovat jinak. Po cestě by měli být umístěny indikátory, že hráč postupuje správně, například v podobě mincí, nepřátel a větších odměn. Po cestách mimo zlatou cestu jsou většinou i umístěni nepřátelé větší obtížnosti, aby se vytvořil pocit riziko-odměna. [21]



Obrázek Příklad „critical path“, Michael Kraveckyj

### Design objektů

Při vytváření designu objektů, je nutné zaměřit na několik bodů:

* **Účel a funkce** – Rozhodnout, jaký účel a funkci bude mít objekt v herním světě (Michael Kraveckyj). Než se objekt začne vytvářet, je nutné vědět, k čemu daný objekt je. [3]
* **Forma a tvar** – Fyzický tvar předmětu, včetně velikosti, barvy a tvaru (Michael Kraveckyj). Forma je obecný tvar a vlastnosti fyzického předmětu, včetně velikosti, textury, barvy a hmotnosti. [11]
* **Interaktivita a zpětná vazba** – Je nutné zvážit, jak daný objekt bude interagovat s hráčem a světem okolo, popřípadě zpětnou vazbu, kterou poskytne (Michael Kraveckyj). Interaktivita je jednou z nejvýznamnějších charakteristik videoher. [22]
* **Kontext a prostředí** – Ujištění, zdali umístěný předmět do kontextu a prostředí hry správně zapadá a neruší celkovou estetiku hry (Michael Kraveckyj). Každý předmět v levelu, by měl být umístěný tak, aby to bylo smysluplné a odůvodnitelné, proč tam je. [12]
* **Ikony a symboly** – Podle slov Michaela Kraveckého: „Občas je nutné poukázat na to, že například bodáky hráči ublíží, když na ně vstoupí, takže se na jejich špičku například kreslí krev jako symbol“. Díky používání symbolů a ikon jsme schopni rychle a efektivně informovat hráče o skutečnostech ve hře. [3]
* **Integrita a konzistence** – Zajištění, aby daný design objektu byl správně integrován s herní mechanikou, a aby byl konzistentní s celkovým designem hry (Michael Kraveckyj). Dobrý herní design vyžaduje, aby všechny objekty ve hře byly integrovány a společně vytvořily soudržný celek. [3]

## Design nepřátel

Videohry jsou plné různých bytostí, co se snaží hráče zabít. Je důležité si však uvědomit, že ne všechny videohry se zaměřují na souboj stejným způsobem, mnoho z nich využívají jiné formy konfliktu, jako je například časový limit tahu, konkurence s ostatními hráči nebo samotné hráčovi schopnosti. Ve hrách lze rozlišit tři typy konfliktů. Konflikt člověka s přírodou, kde se hráč potýká s přírodními živly, například hurikány. Konflikt, kdy hráč musí řešit své vlastní vnitřní problémy, jako otázku, kam jít na oběd je konflikt člověka se sebou samým. Konflikt člověka s člověkem, nebo v případě videoher konflikt hráče s bytostmi ve hře. [12] Při návrhu nepřátel je důležité určit vlastnosti nepřátel [23].

* **Životy** – Definují, jak bude nepřítel silný, jak dlouho dokáže většinou přežít.
* **Rychlost** – Definuje rychlost nepřítele, důležitá informace je, jestli se pohybují rychleji nebo pomaleji než hráč.
* **Poškození** – Podle možného poškození nepřítele lze určit nebezpečí, které znázorňuje.
* **Dosah** – Dosah nepřítele popisuje dosah útoku nepřítele.

V ránem stádiu návrhu nejsou konkrétní čísla důležitá, protože se v průběhu návrhu několikrát změní. Stačí odhadem v procentech (10%, 50%, 100% nebo nízká/ střední/ vysoká) a konkrétní čísla doladit později. [23]

**Úvod a umístění nepřátel**

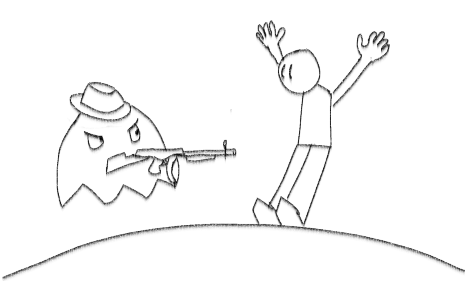
Velmi efektivní způsob, jak hráči sdělit, že se setkává s novým nepřítelem. K tomuto účelu se používá několik triků jako zmrazení kamery nebo její přiblížení, aby si hráč mohl pořádně nepřítele prohlédnout, zobrazení jména nepřítele, vyvoláním napětí pomocí krátkého videa nebo za doprovodu speciálních efektů. [12] Při umisťování nepřátel do levelu, je důležité, aby se nepřátelé v levelu moc neopakovali, byli od sebe odlišitelní, umístěni na smysluplných místech a byli pro hráče výzvou. Nepřátelé by měli být zajímavý a hráč by měl vědět, jak na ně reagovat. [25]

**Chování nepřátel**

Návrh chování nepřítele by mělo řešit odpověď na otázky: Jak se nepřítel pohybuje, co dělá v boji, nebo když je zraněn. Cílem při návrhu chování je neopakovat chování, ale pokusit se, aby se jejich chování vzájemně doplňovalo. [23]

**Základní typy chování**

* **Hlídkování** – Jedno z nejběžněji používaných typů pohybu, při navrhování nepřátel. Je jednoduché na implementaci a nepřátelé díky tomu vypadají „chytře“. [24]
* **Pronásledování** – Pokud se hráč přiblíží k hráči na určitou vzdálenost, nebo nastane splnění jiné podmínky, dojde k jeho pronásledování. V mnoha hrách se z hlídkujících stanou pronásledující poté, co spatří hráče. [12]
* **Střelba** – Hlídkující i pronásledovatelé, pokud k tomu budou mít prostředky a odhalili hráče, se budou pokoušet hráče zasáhnout. [12]



Obrázek Typ nepřítele se střelbou, Vojtěch Košťál

* **Stráž** – Priorita je strážit předmět nebo lokaci. Chování stráží může být jednoduše zkombinováno s pronásledováním a střelbou. [12]
* **Burrower** – Nepřítel může mít mechaniku, která mu umožní dostat se do výhodné pozice pro útok na hráče. Hráč musí poté čekat, než se nepřítel znovu objeví. [12]

Obsah obrázku perokresba

Popis byl vytvořen automaticky

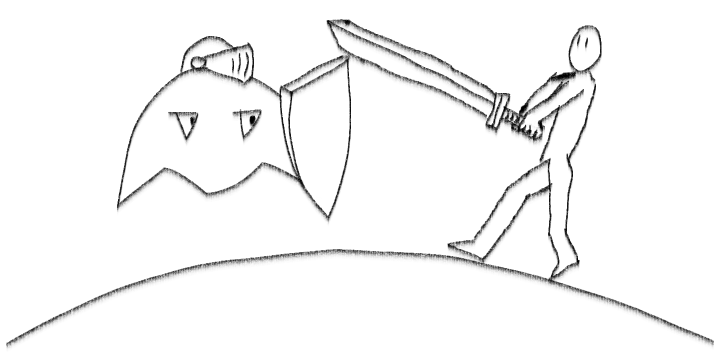
Obrázek Typ nepřítele Burrower, Vojtěch Košťál

* **Teleporter** – Nepřítel, který je schopný rychle změnit pozici, pokud je hráč příliš pomalý, nepřítel může změnit svojí pozici, a tím vykrýt hráčův útok. [12]



Obrázek Typ nepřítele s teleportací, Vojtěch Košťál

* **Blocker** – Nepřátelé mohou mít možnost se bránit proti útokům hráče pomocí štítu nebo jiného obranného zařízení. Štít může zároveň odrážet projektily na různé směry, aby docházelo ještě k samotnému ohrožení hráče. [12]



Obrázek Typ nepřítele s blokováním, Vojtěch Košťál

* Při používání typů nepřátel, kteří mají silně zvýhodňující mechaniku, jako teleport nebo štít, by zároveň měl mít hráč způsob, jak těmto mechanikám zabránit (Michael Kraveckyj).

V hrách se hráč bude pravděpodobně střetávat s mnoha nepřáteli, proto by střed měl být pro hráče zábavný. Výbuchy, vtipné a dramatické animace zásahu nebo odměny, to vše v hráči vyvolá pocit zábavy. Dále je nutné určit, jak bude nepřítel po smrti odstraněn z herního světa, nabízí se hned několik způsobů jako: výbuch, zmizení v oblaku kouře nebo zůstane na obrazovce. Ve většině her není aplikovaná mechanika „stealth“, neboli plížení, proto by měla být většina nepřátel designovaná tak, aby se jim hráč nechtěl vyhnout. Designer by měl hráčům dopomoct k tomu, aby měli pocit, že s nepřáteli bojovat sami chtějí.[12] Hráč musí být informován o tom, že [12]:

* Nepřátelé u sebe mají užitečné věci. Zlato, náboje, lékárničky.
* Blokují cestu, takže hráč přes ně musí projít, pokud chce pokračovat ve hře
* Mají u sebe klíč. Občas je nutné porazit některé nepřátelé, aby se odemkla speciální truhla, nebo dveře.
* Poražení nepřátel pomáhá vylepšovat hráčovi schopnosti.

Je nutné myslet ale i na to, aby samotní nepřátelé měli možnost se bránit a útočit. Pokud mají ruce a nohy, mohou útočit zblízka. Samozřejmě útok se zbraní jako meč, nebo střelnou puškou, které navíc mohou mít výhodu v podobě jedu, který bude následně hráči způsobovat stálé poškození, dokud se neuzdraví. Stejně tak, se budou moct bránit pomocí krytí, odražení, zmražení a paralyzování hráče. [12]

### Design bossů

Velmi podobný proces jako u klasického nepřítele. [23] Vytvoření bosse ve videohře je složitý úkol a vyžaduje mnoho úsilí, pokusů a omylů. Obtížnost nepřítele v podobě bosse by měla odpovídat tomu, v jaké části se hráč zrovna nachází. Není dobré, aby hráč čelil jednomu nepříteli více než šestkrát, protože poté by poražení nepřítele nepůsobilo jak odměna, ale spíše jako úleva. Zároveň nesmí být takovýto nepřítel příliš jednoduchý, měli by být výzvou, pokud lze porazit nepřítele za 30 sekund bez větších problémů, hráči nemusí mít pocit, že něčeho dosáhli. Pokud ale souboj trvá 5 minut a vyžaduje hráčovo soustředění a přesnost, hráči pocítí pocit úspěchu, který lze ještě umocnit tím, že hráčům darujete další život, novou schopnost nebo odemknutí zajímavé zbraně, či oblasti. Při tvorbě je důležitá kreativita, vytvoření zážitku, který si hráči budou pamatovat a hovořit o něm i dlouho poté. Při tvorbě takového bosse pomáhá unikátní design a soubor dovedností. Vzhled a kontext bosse by ve hře měl být jedinečný, měl by správně zapadat do vytvořeného prostoru hry, ale zároveň by měl být dobře odlišitelný od ostatních postav ve hře. Při tvorbě středověké hry, boj s futuristickým robotem by nebyl přirozený ani vhodný pro kontext vyprávění. Mnohem větší smysl by dával speciálně navržený troll s magickými schopnostmi [26]

**Soubor schopností bosse**

Schopnosti jsou stejně důležité jako vzhled. Schopnosti, kterými boss disponuje by měly pro hráče znázorňovat výzvu, se kterou se ještě nepotkali a bude od nich očekáváno použití herních mechanik z jedinečných úhlů. Schopnosti, stejně jako vzhled, musí zapadat do celkového kontextu hry. Zároveň by boss měl mít v zásobě více než jednu schopnost a sekvenci těchto schopností, které se budou řídit například aktuálním počtem životů bosse, protože hráči velmi rychle rozpoznají jejich vzor. [26]

## Procedurální generování

Procedurální generování (PG) je automatizovaný přístup k tvorbě mediálního obsahu. PG lze využít v mnoho oblastech. Např. umění, hudba, poezie, film a hry (deskové i počítačové). V počítačových hrách se PG používá při generování nových levelů, nebo jako reakce na hráčův pohyb, ačkoli lze PG použít na téměř cokoli, používá se hlavně na obsah, který by musel vytvořit herní designer. (textury, postavy, mapy, úkoly…) [30]

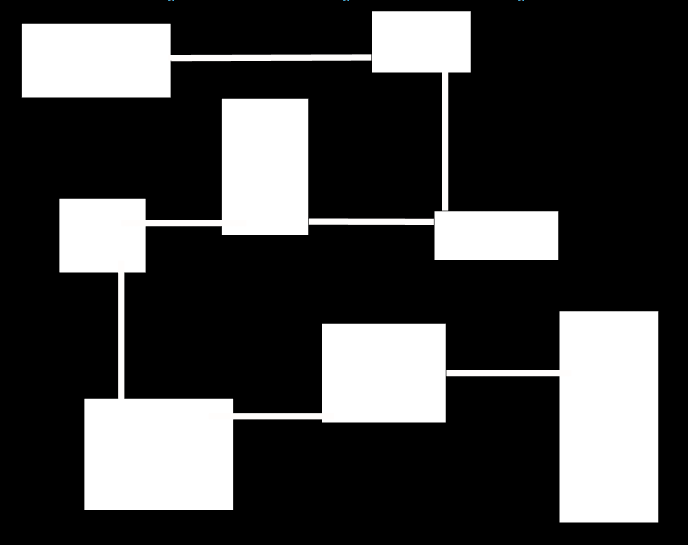
PG se zakládá na schopnosti vytvářet obsah náhodně, což vyžaduje, aby generátor ze stejných vstupních dat vygeneroval různé výsledky. Takový přístup sebou nese několik rizik, protože vývojář částečně ztrácí kontrolu nad výsledným obsahem. Náhodnost je zajištěna pomocí generátoru náhodných čísel (RNG). [31]

### Typy procedurálního generování map

**Simple Room-Placement [33]**

Metoda umísťování náhodných místností.

1. Začne se s obdélníkovou místností náhodné velikosti.
2. Náhodně se zvolí pozice na mapě
   1. Pokud v lokaci není jiná místnost, tak se nová místnost přidá
3. Pokračuje se s dalšími místnostmi, dokud nevznikne dostatečný počet místností
4. Vytvořené místnosti se propojí průchody
   1. Lze využít jednoduchý algoritmus na náhodné přepínání vertikálních a horizontálních průchodů.

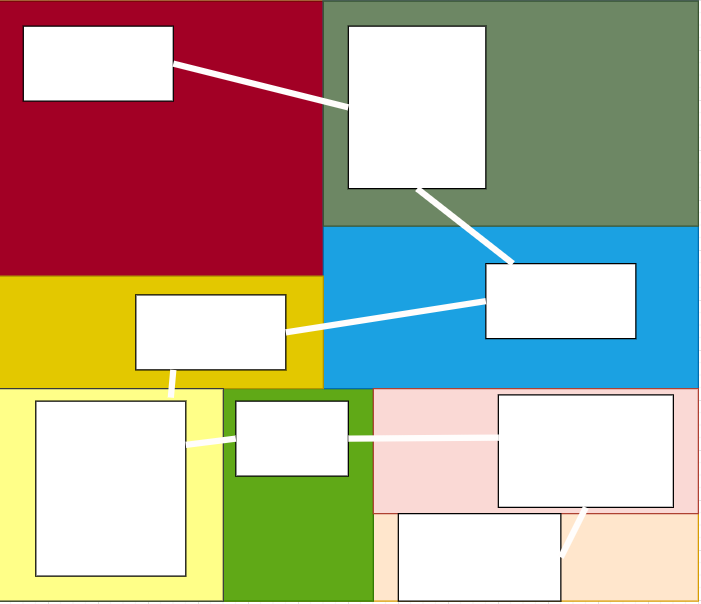


Obrázek Výsledek generování Simple Room-Placement

**Místnosti s binárním rozdělením prostoru [33]**

Výsledek stejný jako u umisťování náhodných místností, ale s lepším rozestupem mezi místnostmi.

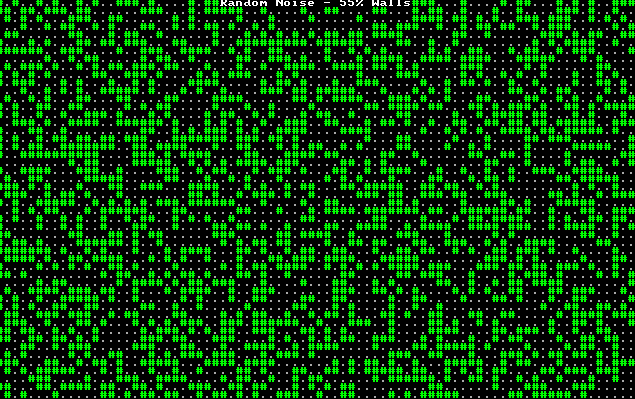
1. Mapa se rozdělí na 2 části, náhodně se rozhodne, jestli vertikálně nebo horizontálně.
   1. S rozdělováním se pokračuje, dokud nedojde k dostatečnému počtu místností
2. Do rozdělených prostorů se umístí náhodně velké místnosti.
3. Algoritmus následně vybere místnosti (v rámci sloupců a řádků), které se propojí.



Obrázek Výsledek generování místností s binárním rozdělením prostoru

**Celulární automat [33]**

1. Vytvoří se náhodná mapa a její kopie.
2. Aplikují se pravidla buněk na každou z dlaždic.
   1. Iteruje se každá z dlaždic, která není na hraně a spočítá se počet sousedů, včetně úhlopříček.
      1. Pokud nemá žádné sousedy, stává se stěnou.
      2. Pokud má jeden až čtyři sousedy, stává se prázdným prostorem.
      3. Pokud je sousedů pět nebo více, stává se stěnou.
      4. Algoritmus by se měl upravit konkrétní hře.
3. Opakování

****

Obrázek Výsledek generování místností pomocí celulárního automatu, [33]

# Herní enginy

Herní enginy jsou nástroje, které jsou určené pro vývoj softwaru, snížení nákladů, složitosti a času. Herní enginy vytvářejí abstraktní vrstvu nad nejčastějšími úkoly při vývoji her. Abstraktní vrstvy jsou baleny do nástrojů tak, aby umožňovaly nahrazení nebo rozšíření o další komponenty od třetích stran. Herní enginy poskytují možnost vývojářům soustředit se pouze na psaní kódu. Dobře navržené herní enginy navíc dobře oddělují interní funkčnosti, kód hratelnosti je oddělen od kódu, který například dekomprimuje mp3 soubory, zároveň kód hratelnosti volá na dobře definované rozhraní api enginu, aby došlo k co nejoptimálnějšímu výkonu. [9]

Nejoblíbenější herní enginy by měly obsahovat podle [9] některé nebo všechny funkce jako:

* Možnost sestavit hru na všechny platformy jako je windows, linux, ios, playstation či xbox
* Graficky renderovací engine, který podporuje 2D a 3D grafiku.
* Fyzikální engine podporující detekce kolizí a gravitace.
* Zvukový engine pro přehrávání hudby a speciálních efektů.
* Podporu skriptování pro implementaci herní logiky.
* Objektový model herního světa.
* Zpracování animací pro načítání a přehrávání snímků animací.
* Možnost více vláknového zpracování
* Správu paměti
* Umělou inteligenci

## Unity Engine

Unity je herní engine pro vývoj 2D a 3D her. První verze Unity byla vytvořena roku 2005 společností Unity Technologies s cílem poskytnout většímu počtu vývojářů přístup k nástrojům pro herní vývoj, což bylo v té době novinkou. Během let se Unity engine dramaticky změnil a rozšířil, a přizpůsoboval nejnovějším technologiím a přístupům. I v dnešní době se pořád zaměřuje na to, aby poskytoval co nejrobustnější soubor nástrojů pro herní průmysl, a aby byl co nejjednodušší pro vývojáře s jakýmkoli stupněm zkušeností, včetně začátečníků. Společnost se zaměřuje i na vývoj real-time 3D, a díky tomu je tento engine jedním z nejvýkonnějších volně dostupných enginů. [34]

## Godot Engine

Godot engine je výkonný herní engine pro tvorbu 2D a 3D her na různých platformách pomocí jednotného rozhraní. Poskytuje soubor nástrojů, aby se vývojáři mohli soustředit na samotný vývoj hry a nemuseli vytvářet již existující prvky. Godot je zcela zdarma a open-source. Uživatelé mají plné vlastnictví svých her, včetně zdrojového kódu enginu bez jakýchkoli podmínek. Vývoj Godot enginu je plně komunitní, takže uživatelé mohou přispívat svými nápady a očekáváními na vývoj enginu. [35]

## Unreal Engine

Unreal engine je software, který umožňuje tvůrcům tvořit 3D obsah nové generace v reálném čase s větší svobodou, věrností a flexibilitou než kdykoli předtím. Unreal engine je vyvíjen společností Epic Games a je dostupný jako bezplatný nástroj pro vývojáře. Podporuje vývoj pro různé platformy, včetně desktopů, mobilních zařízení a herních konzolí. Unreal engine je známý pro své pokrořilé grafické funkce, jako jsou například realistické osvětlení a stíny, dynamické efekty počasí, zvukové efekty, fyziku a umělou inteligenci [36]

## Porovnání

Pro porovnání byla vybrána bodově hodnocená kritéria a stupeň důležitosti, který dané hodnocení násobí. Hodnocení programovacího jazyku, v kterém lze psát kód je jedno z nejdůležitějších částí. V hodnocení je i zahrnuta osobní zkušenost autora s jazyky. Ačkoli Unreal i Godot engine mají možnost programování v C#, jejich podpora, oproti Unity, je velmi malá. Dalším kritériem je aktivní komunita, čím aktivnější komunita daného enginu, tím snazší je hledání řešení konkrétního problému a samotný vývoj hry. V případě uživatelské licence je favoritem Godot engine, jelikož svým uživatelům poskytuje licence zcela zdarma, Unity a Unreal mají zdarma pouze standartní osobní licence. Vyšší licence Unity a Unreal enginu nabízí nadstandartní support, licencované servery apod. Dalším velmi důležitým bodem je podpora vývoje 2D her. Jelikož je Unreal engine vytvořen primárně pro vývoj 3D her a na práci s 2D velmi dlouho nevyšel žádný update, jeho hardwarové nároky jsou oproti Unreal a Godot enginu mnohem vyšší, takže se pro vývoj 2D her nehodí, zároveň Unreal engine nemá vyvinuté funkce pro práci s 2D, ale pouze příkazy, které přepisují 3D funkce. Všechny níže uvedené enginy plně podporují vydání hry na všechny hlavní platformy.

Tabulka Subjektivní porovnání herních enginů podle vybraných kritérii

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kategorie | Unity - Free technology icons  **Unity Engine** | | Unreal Engine icon in Windows 10 Style  **Unreal Engine** | | **Godot Engine** | | Důležitost (násobič) |
| Programovací jazyk | C# | +++ | C++ | + | Gdscript | - | 3 |
| Aktivní komunita |  | +++ |  | ++ |  | ++ | 2 |
| Zdarma | Základní licence | + | Základní licence | + | Zcela zdarma | +++ | 1 |
| 2D |  | +++ | úplný základ | + |  | +++ | 3 |
| Multiplatformní |  | +++ |  | +++ |  | +++ | 2 |
| Výsledek: |  | 31 |  | 17 |  | 22 |  |

# Praktická část – Vývoj hry v Unity enginu

V následující kapitole autor provedl aplikování poznatků z teoretické části k implementaci GDD. V kapitole budou dále popsány důležité a zajímavé části vývoje 2D pc hry s ohledem na zachování kvalitního game designu a na konci kapitoly bude zmíněna zpětná vazba od testerů hry.

## Implementace GDD

### Základní informace

* **Koncept** – Hráč bude procházet jednotlivé levely a na konci třetího levelu bude muset porazit silného nepřítele (bosse).
* **Žánr** – Roguelike.
* **Cílené publikum** – Hra bude cílit na všechny věkové kategorie, zvlášť její obtížnost by mohla zaujmout starší publikum.
* **Vzhled hry** – Hra bude vytvořena ve 2D v pixel art stylu, implementují se animace pohybu i defaultního (idle) stavu, kdy se hráč nehýbe.

### Hratelnost a mechaniky

**Hratelnost**

* **Postup hrou** – Hráč bude procházet jednotlivé místnosti v levelu. Čím vyšší bude level, tím delší bude cesta do dalšího.
* **Výzva** – V každé místnosti se budou moci vyskytovat nepřátelé a na konci třetího levelu na hráče bude čekat silný nepřítel (boss).
* **Cíl hry** – Cílem hry je dojít na konec

**Mechaniky**

* **Fyzika** – Ve hře se implementuje základní fyzika našeho světa.
* **Pohyb** – Hra umožní hráči pohyb pomocí „WSAD“ do čtyř stran a možnost „kotoulu“ pomocí mezerníku. Rozhlížení a určení směru střelby pomocí ukazatele myši.
* **Objekty** – Ve hře se budou nacházet různé objekty rozdělené do kategorii
  + Sběratelské předměty – po jejich sebrání dojde k navýšení hráčových statistik. Pro jejich stačí hráčem přejít přes tento předmět.
    - Léčení – Obnovení chybějících životů hráče.
    - Peníze
    - Zbraně – pokud hráč danou zbraň nemá, po sebrání se přidají hráči do inventáře.
  + Objekty levelu
    - Krabice – lze zničit pomocí střelby nebo „kotoulu“. Uvnitř se mohou nacházet peníze nebo léčení.
    - Bodce – při kolizi s bodci dojde k odečtení životů hráče. Implementuje se mechanika, kdy při použití „kotoulu“ nedojde k odebrání životů hráče.
    - Truhla – Při přiblížení hráče k truhle ze strany, kde se truhla otevírá bude možnost interakce k otevření truhly, ve které se může nacházet nová zbraň.
    - Klec s charakterem – Klec bude sloužit jako možnost odemknutí nového charakteru.
    - Konec levelu – na konci levelů bude umístěn ukazatel konce, při vstoupení k objektu bude hráč přesunut do dalšího levelu.
    - Obchod – Tři různé položky za různé ceny.
* Interakce – Pro interakci s truhlou, klecí či jinými charaktery bude využita klávesa „E“.
* **Souboj** – Pro souboj bude hráč disponovat některou ze zbraní. Nepřátelé budou mít různé způsoby, jak hráče zranit.
* **Ekonomika** – Ve hře z nepřátel či truhel budou padat peníze, které může hráč vždy utratit v obchodu, který se bude nacházet v každém levelu.

### Levely

* Rozložení místností levelu bude generováno procedurálně (náhodně). Obsah dané místnosti se vybere z předem připravených. Každý level bude moci mít místnost s truhlou a obchodem. Ve většině místností se umístí nepřátelé, krabice, bodce.

### Interface

* **Menu**
  + **Hlavní menu** – V hlavním menu bude možnost startu nové hry, vypnutí hry a možnost smazat svůj dosavadní postup ve hře (odemknuté charaktery).
  + **Pause menu** – ve hře bude hráči umožněno v kterýkoli čas hru zastavit (pause), kde bude možnost ukončení hry a pokračování.
* **Audio, hudba, efekty –** Ve hře se nastaví zvukový doprovod v podobě hudby a zvukové efekty při každé činnosti.
* **Systém nápovědy** – na začátku hry bude hráči umožněno zobrazit si nápovědu se základními mechaniky hry.

### AI

* **Nepřátelé** – Simulace umělé inteligence bude dosažena pomocí algoritmů pohybu nepřátel.
  + Reakce na pohyb hráče
    - Sledují ho
    - Utíkají od něj
  + Náhodný pohyb po místnosti
  + Pohyb mezi body místnosti
* **Podpora AI**
  + Kolize – Neimplementuje se samotná detekce (funkce unity), ale pouze události při kolizích.

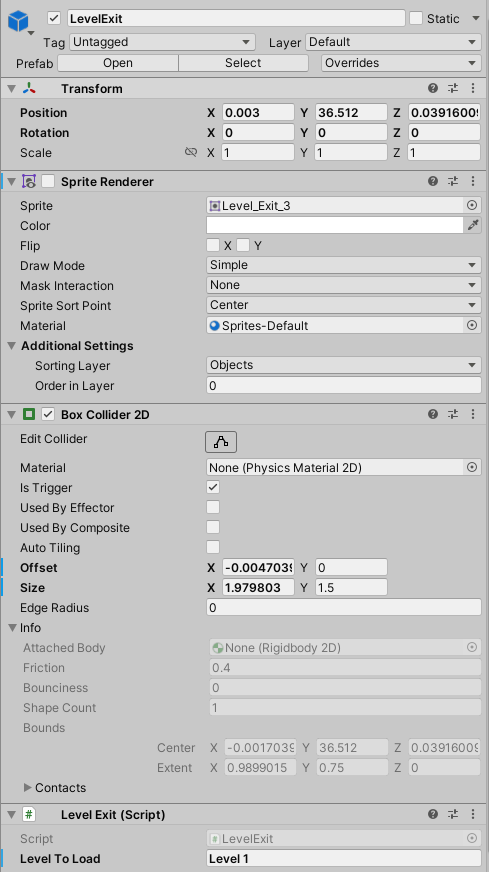
## Vývoj hry

V následující části budou popsány některé části vývoje 2D hry v prostředí Unity enginu, které autor považuje za zajímavé, či důležité.

### GameObject

GameObject je hlavním prvkem v Unity. Reprezentuje herní objekty nebo prvky jako je například hráč, nepřítel nebo statické objekty v herním světě, kterými mohou být stromy, budovy apod. GameObject může mít různé komponenty, které přidávají různé vlastnosti a chování. Tyto prvky lze přidávat a odebírat z GameObjectu dle potřeby, což umožňuje vývojářům vytvářet různorodé hry. Mezi hlavní komponenty patří:

* Vizuální prvky (Sprite renderer)
* Fyzikální modelování (Box Collider 2D)
* Zvukové efekty
* Skripty (Level Exit Script)



Obrázek Ukázka GameObject „LevelExit“

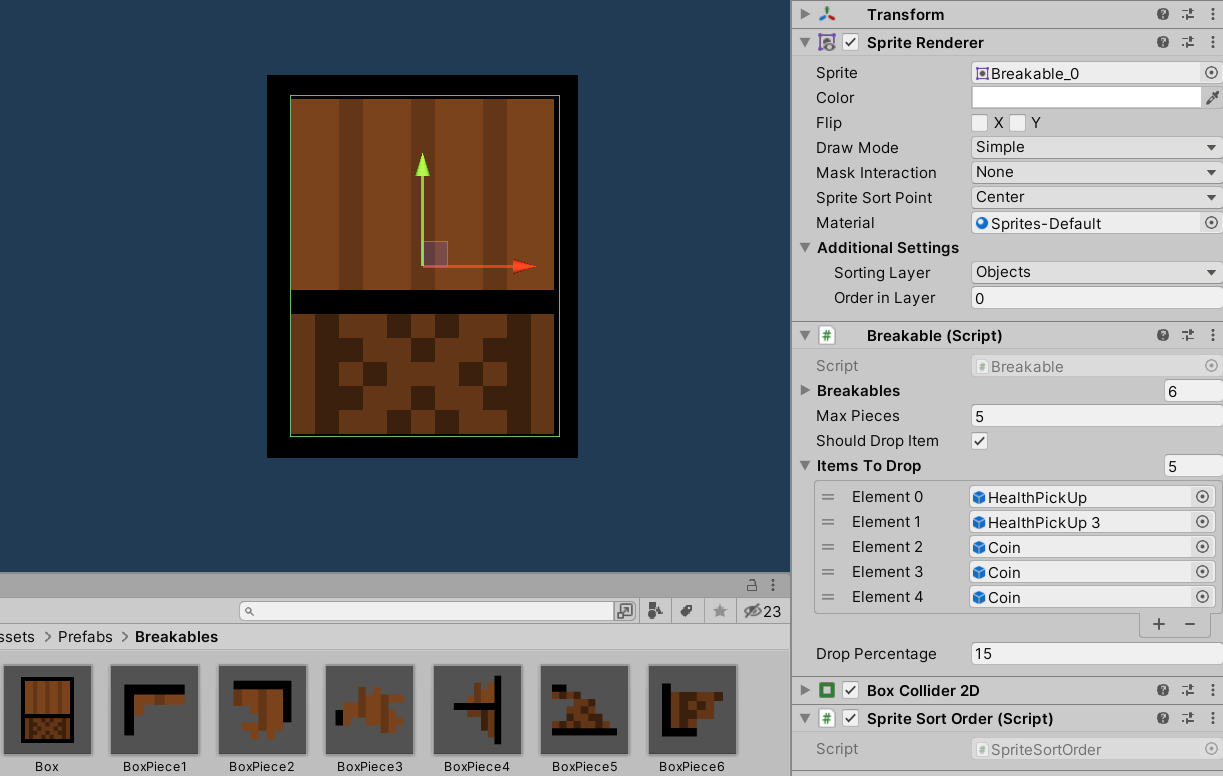
Každý GameObejct má také transformace, které určují jeho polohu, rotaci a měřítko v herním světě. Transformace, stejně jako všechny jiné vlastnosti, mohou být nastaveny přímo v editoru Unity nebo mohou být modifikovány pomocí skriptů. Jako ukázky gameObject budou zmíněny objekty **Krabice, Enemy a Boss.**

**Level objects**

K zachování kvalitního level designu, byly do jednotlivých místnosti umístěny objekty tak, aby měl hráč dojem, že jeho činy ovlivňují okolí. Při střelbě do krabice, se krabice rozpadne na několik kusů, při kolizi hráče s bodci, hráč obdrží poškození. Objekt konce levelu přesune hráče do dalšího levelu nebo na finální obrazovku.

Script pro krabici (breakable) obsahuje 6 dalších objektů (BoxPiece), které reprezentují jednotlivé úlomky krabice, které se při rozbití krabice rozletí do náhodných směrů.

* Max pieces – Označuje kolik se maximálně zobrazí úlomků.
* Should drop item – Označuje, zdali z dané Breakable je možné získat nějaký předmět.
* Items to drop – Seznam předmětů, které mohou být získány z Breakable
* Drop percentage – Šance, že z dané Breakable spadne nějaký předmět.



Obrázek Ukázka objektu krabice (Breakable)

private void DestroyBreakable()

{

Destroy(gameObject);

AudioManager.Instance.PlaySFX(0);

for (var i = 0; i < Random.Range(1, MaxPieces); i++)

{

Instantiate(Breakables[Random.Range(0, Breakables.Count)], transform.position, transform.rotation);

}

if (ShouldDropItem)

{

var dropChance = Random.Range(0f, 101f);

if (dropChance < DropPercentage)

{

Instantiate(ItemsToDrop[Random.Range(0, ItemsToDrop.Count)], transform.position, transform.rotation);

}

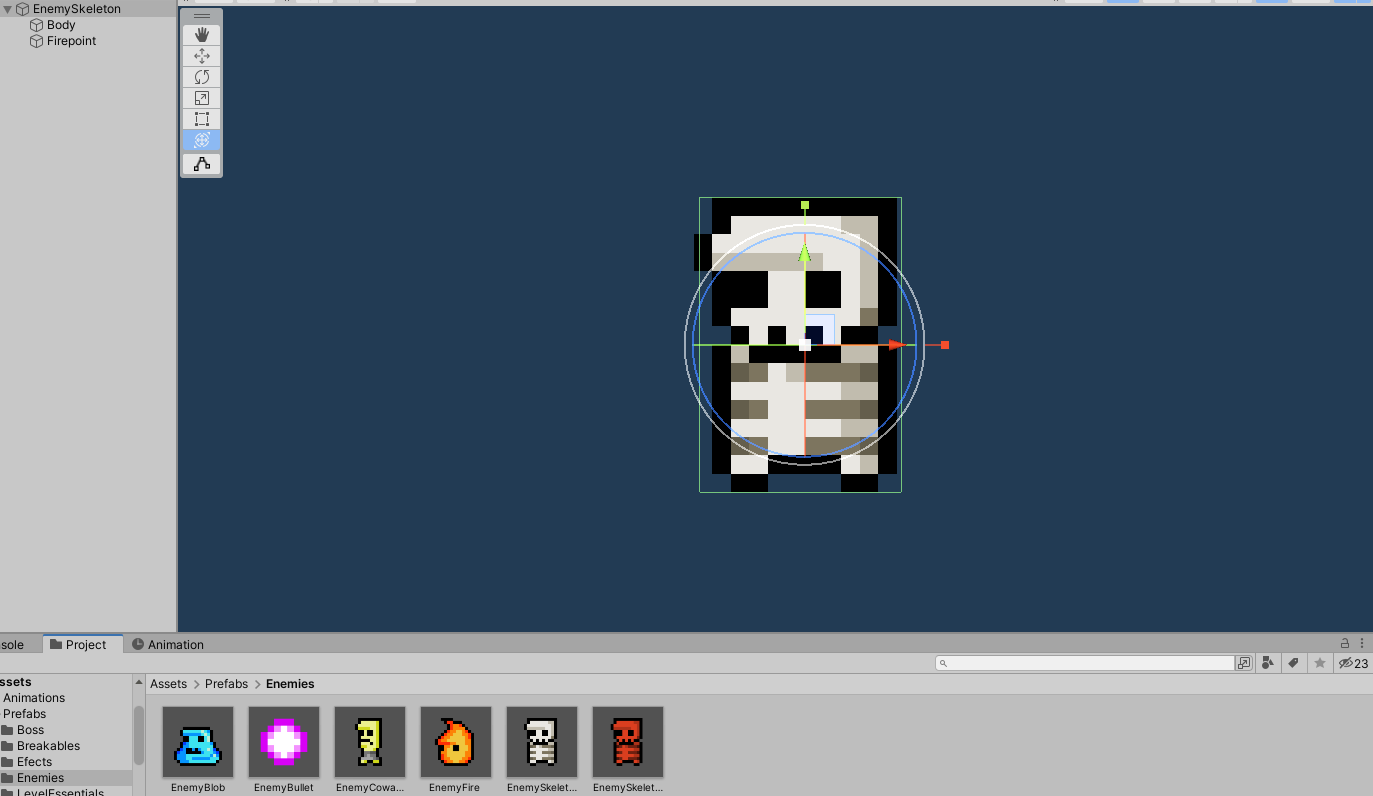
}

}

Ukázka kódu Kód pro obsluhu zničení Breakable objektu

**Nepřátelé (Enemy)**

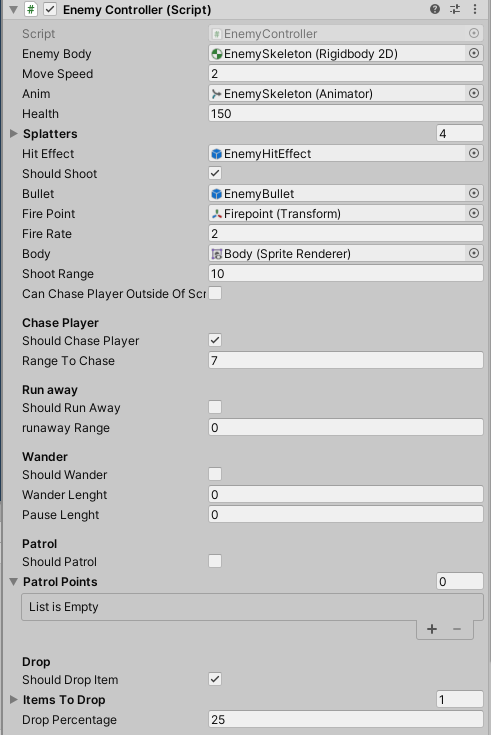
Při tvorbě nepřátel se autor snažil na vytvoření různorodých nepřátel, aby hra byla zábavná a nepřátele vypadali a chovali se „chytře“ a každý z nepřátel má předdefinované chování a vlastnosti, které lze měnit v editoru a tím je zajištěna možnost vytvářet nové originální variace nepřátel.

****

Obrázek Ukázka nepřítele Skeleton

Chování nepřátel lze měnit za běhu v Unity editoru nebo v samotném scriptu „Enemy Controller“. Mezi základní implementované chování ve hře patří:

* Should chase player – rozhodnutí, zdali má nepřítel sledovat hráče.
  + Range to chase – od jaké vzdálenosti má nepřítel sledovat hráče
* Should runaway – při splnění podmínky se nepřítel pokouší dostat od hráče
  + Runaway range – vzdálenost od jaké má nepřítel utíkat od hráče
* Should wander – nepřítel se bude náhodně pohybovat po místnosti
  + Wander lenght – jak dlouho potrvá pohyb
  + Pause lenght – délka pauzy mezi náhodnými pohyby
* Should Patrol – nepřítel se pohybuje mezi definovanými body místnosti
  + Patrol points – body umístěné v mísnosti



Obrázek Ukázka komponenty script u nepřítele

else if (ShouldWander)

{

if (wanderCounter > 0)

{

wanderCounter -= Time.deltaTime;

moveDirection = wanderDirection;

if (wanderCounter <= 0)

{

pauseCounter = Random.Range(PauseLenght \* 0.75f, PauseLenght \* 1.25f);

}

if (pauseCounter > 0)

{

pauseCounter -= Time.deltaTime;

if (pauseCounter <= 0)

{

wanderCounter = Random.Range(WanderLenght \* 0.75f,

WanderLenght \* 1.25f);

wanderDirection = new Vector3(Random.Range(-1f,1f),

Random.Range(-1f, 1f), 0);

}

}

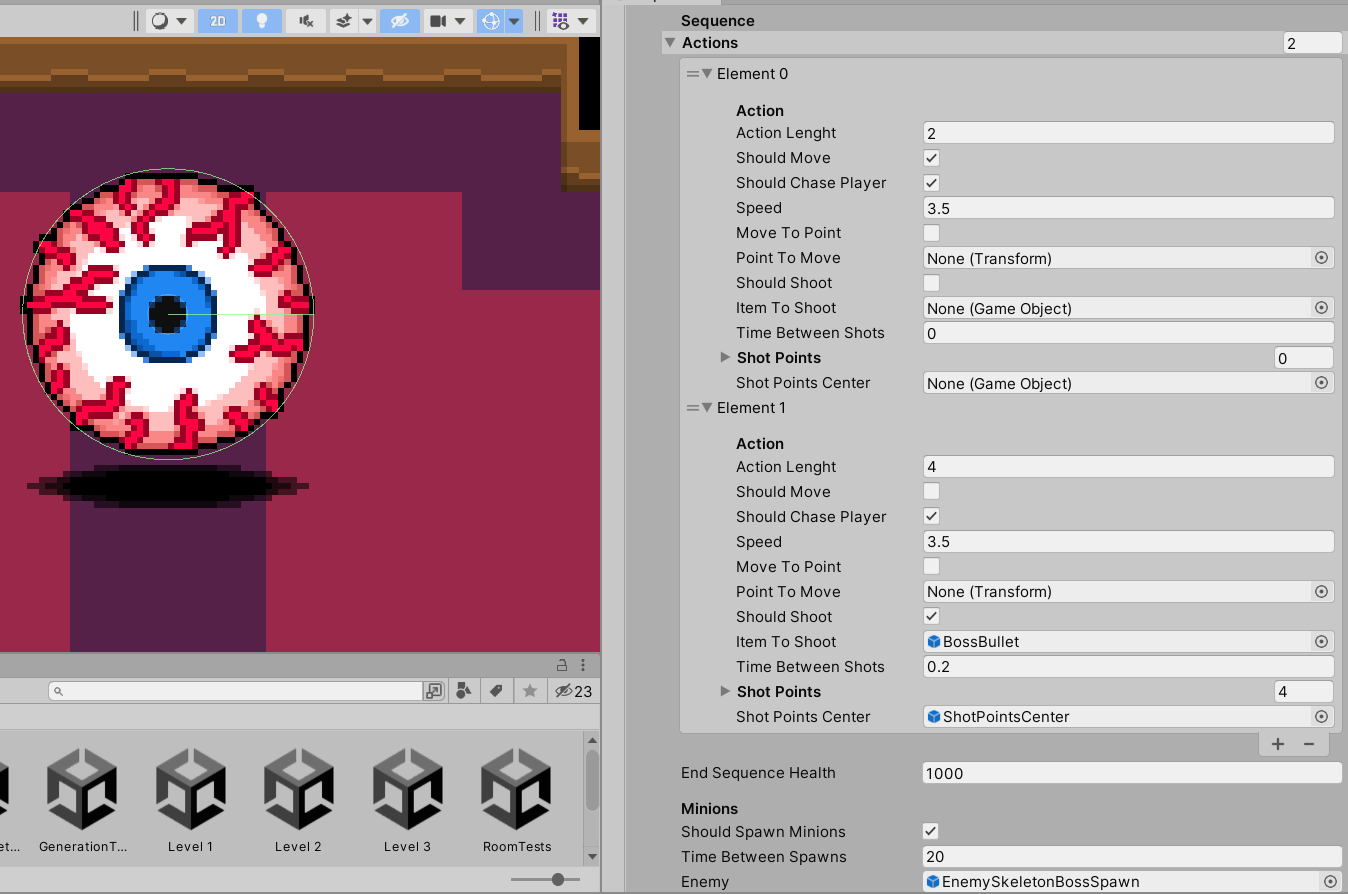
}

}

Ukázka kódu Chování nepřítele "Wanderer"

**Boss**

Podle poznatků z teoretické části byl vytvořen boss, který je velmi silný typ nepřítele a jeho návrh je velmi podobný klasickým nepřátelům. Pro bosse byla vytvořena sekvence schopností, která se odvíjí od počtu aktuálních životů a každá sekvence obsahuje několik po sobě jdoucích akcí bosse.



Obrázek Ukázka vzhledu a sekvence bosse

* Action Lenght – Určuje dobu trvání akce.
* Should move – Aktivní, pokud se nepřítel má pohybovat během akce.
* Should chase player – Aktivní, pokud během akce má následovat hráče.
* Speed – Jakou rychlostí se nepřítel pohybuje.
* Move to point – Akce s pohybem k stanovenému bodu.
  + Point to move – Bod, k jemuž směřuje nepřítel.
* Should shoot – Aktivní, pokud má nepřítel během akce střílet na hráče.
  + Item to shoot – Typ střely.
  + Time between shots – Čas mezi jednotlivými výstřely.
  + Shot points – Identifikace středu nepřítele, kolem kterého následně rotuje bod střelby.
* End sequence health – počet životů, u kterých nastane přepnutí na další sekvenci.
* Should spawn minions – akce mimo sekvenci, dochází k vytváření dalších nepřátel.
  + Time between spawns – čas mezi vytvářením dalších nepřátel.
  + Enemy – určuje, jaký nepřítel se bude bossem vytvářet.

if (Health <= Sequences[currentSequence].EndSequenceHealth

&& currentSequence < Sequences.Count - 1)

{

currentSequence++;

Actions = Sequences[currentSequence].Actions;

currentAction = 0;

actionCounter = Actions[currentAction].ActionLenght;

}

Ukázka kódu Přepnutí sekvence podle aktuálních životů nepřítele:

if (Sequences[currentSequence].ShouldSpawnMinions)

{

spawnCounter -= Time.deltaTime;

if (spawnCounter <= 0)

{

spawnCounter = Sequences[currentSequence].TimeBetweenSpawns;

for(int i = 0; i < Sequences[currentSequence].

StartNumberOfSpawns; i++)

{

var position = gameObject.transform.position;

position.x += Random.Range(-1, 2) \* 2;

position.y += Random.Range(-1, 2) \* 2;

var enemy = Instantiate(Sequences[currentSequence].Enemy, position, transform.rotation);

enemy.transform.parent = EnemySpawns;

}

Sequences[currentSequence].StartNumberOfSpawns++;

}

}

Ukázka kódu Akce v sekvenci, vyvolání dalších nepřátel bossem

if (Actions[currentAction].ShouldShoot)

{

shotCounter -= Time.deltaTime;

if (shotCounter <= 0)

{

shotCounter = Actions[currentAction].TimeBetweenShots;

Actions[currentAction].ShotPoints.ForEach(x =>

{

Instantiate(Actions[currentAction].ItemToShoot, x.position, x.rotation);

});

Actions[currentAction].ShotPointsCenter.gameObject.transform.

rotation \*= Quaternion.Euler(0f, 0f, 5f);

}

}

Ukázka kódu Akce v sekvenci, při které nepřítel má střílet.

### Kamera

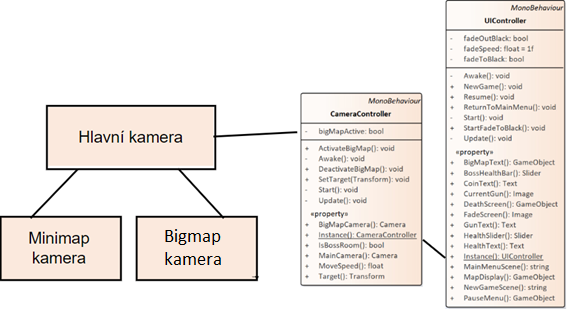
Při vytváření nového projektu v Unity, se vytvoří i hlavní (výchozí) kamera. Hlavní kamera je zodpovědná za zobrazení hry a může být přizpůsobena pomocí různých parametrů. Kamera se v Unity definuje pomocí komponenty „Camera“. Tuto komponentu lze přidat k jakémukoli hernímu objektu, který má transformace, jako je například prázdný herní objekt, nebo hlavní kamera, pokud přidáváme další kameru. Jak bylo zmíněno, komponent kamera má několik nastavení, které ovlivňují její chování. Mezi nejdůležitější patří:

* Near clip plane (z near) – určuje minimální vzdálenost od kamery, při které jsou objekty stále viditelné.
* Far clip plane (z far) – určuje maximální vzdálenost od kamery, při které jsou objekty stále viditelné.
* Background color – určuje barvu pozadí, které je viditelné za objekty ve scéně.
* Target Texture – umožňuje renderování do vybraného souboru v reálném čase, tento soubor lze poté použít například k vytvoření mini mapy.

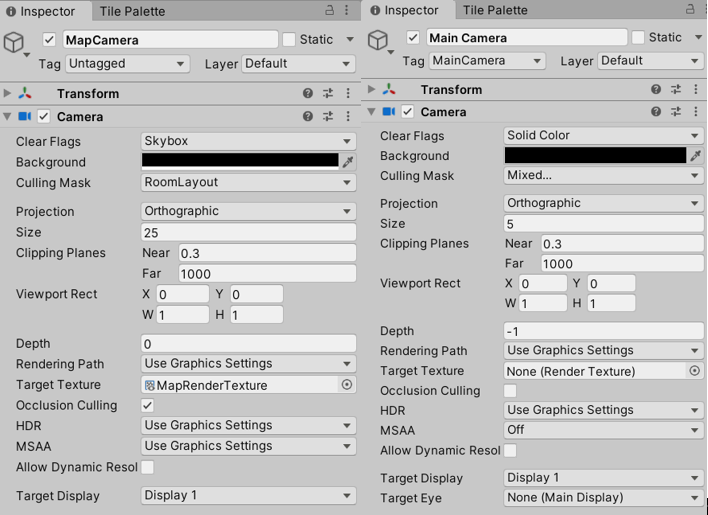
Kamera také umožňuje různé druhy projekcí, jako je perspektivní nebo ortografické projekce. Tyto projekce jsou vhodně pro různé typy her, jelikož mají odlišný vliv na zobrazení scény. Ve hře byly implementovány celkem tři kamery.

* Hlavní (main) kamera, která zobrazuje samotnou hru a UI. Hlavní kamera má 2 potomky
  + Minimap kamera – Určena pro sledování minimapy
  + Map kamera – Kamera je ve výchozím nastavení vypnuta a zapíná se pomocí tlačítka. Při aktivaci překryje celou obrazovku a zobrazí zvětšenou verzi minimapy.

Hlavní kamera sleduje vždy střed místnosti, v které se hráč nachází. Výjimkou jsou úvodní místnost a finální místnost s bossem, kde kamera sleduje hráče.



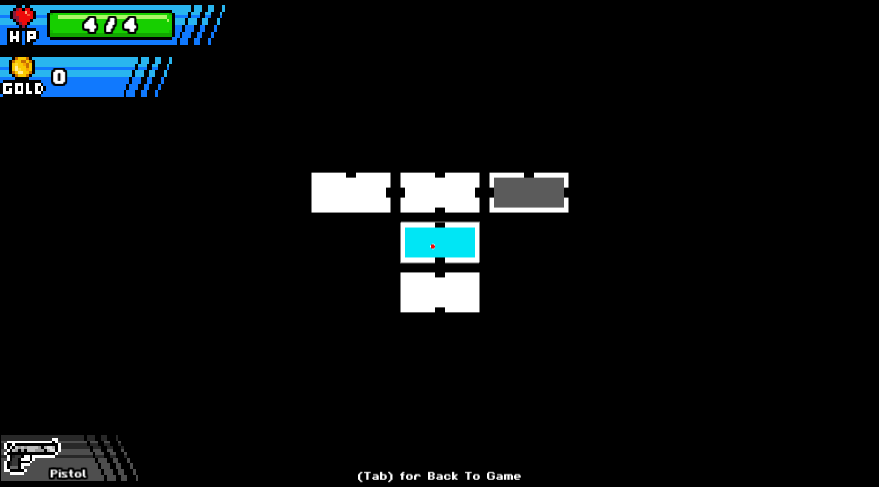
Obrázek Propojení kamer scény a ovládacích scriptů



Obrázek Kamera minimapy a Hlavní kamera

**Target texture**

Target Texture je textura, která slouží jako cílová plocha pro vykreslování grafiky v aplikaci vytvořené v Unity. Tato textura umožňuje ovlivňovat, kam se bude vykreslovat grafika, a umožňuje také získat výstupní data z procesu vykreslování pro další zpracování v rámci aplikace. Autor implementoval Target texture pro vykreslování postupu hrou do textury pomocí minimap camera (viz obr. 25 Map camera, Target texture) a při aktivaci BigMap Camery je použita v UI prvku canvas pro zobrazení v plné velikosti okna.

****

Obrázek Použtí BigMap camery

public void ActivateBigMap()

{

if (LevelManager.Instance.IsPaused) return;

bigMapActive = true;

BigMapCamera.enabled = true;

MainCamera.enabled = false;

PlayerController.Instance.CanMove = false;

Time.timeScale = 0f;

UIController.Instance.MapDisplay.SetActive(false);

UIController.Instance.BigMapText.SetActive(true);

}

Ukázka kódu Kód pro aktivaci BigMap camery

### Animator

Animator v Unity je komponenta, která umožňuje animovat 3D modely a objekty v rámci hry nebo aplikace. Animator umožňuje vytvářet komplexní animace pomocí klíčových snímků, přechodů mezi animacemi a nastavování různých parametrů. Animator pracuje s animačními kontroléry, což jsou soubory, které definují, jaké animace se budou přehrávat v závislosti na různých vstupních podmínkách. Vstupní podmínky mohou být například pohyb postavy, interakce s objekty, stavy zdraví postavy a další. Animator také podporuje vrstvy animací, které umožňují přehrávat více animací najednou, jako například pohyb postavy a animaci zbraně v ruce. Díky tomu lze vytvářet plynulé a realistické animace, které zlepšují výsledný vizuální efekt hry nebo aplikace.

V práci byl využit animátor pro hráče a nepřítele. Idle (postava stojí na místě), chůze a kotoulu u hráče.

Obsah obrázku text, bílé

Popis byl vytvořen automaticky

Obrázek Řídící body pro animaci chůze hráče

Obsah obrázku text, zařízení, stůl, měřidlo

Popis byl vytvořen automaticky

Obrázek Ukázka animace kotoulu

Unity samo dopočítává pozice mezi klíčovými body, aby animace byla plynulá.

### Vytváření levelu

Vytváření levelů bylo v práci realizováno pomocí procedurálního (náhodného) generování místností. Každá místnost se skládá ze dvou částí vytvořených v Tile Palette. První část je střed místnosti, který určuje, co se v dané místnosti nachází (nepřátelé, krabice, konec levelu…). Druhou částí je „obrys“ místnosti, který určuje, jakým směrem jsou umístěny východy (nahoře, dole, vlevo, vpravo).

**Tile Palette**

Tile palette je nástroj v Unity, který slouží pro tvorbu různých dlaždicových map pro hry a umožňuje snadno vytvářet a editovat mapy skládajících se z množství malých dlaždic umístěných vedle sebe. Dlaždice pro tvorbu mapy lze vytvořit pomocí PNG obrázku, který musí být, ale předem připraven. Všechny dlaždice v PNG obrázku musí mít stejnou velikost a umístěny vedle sebe v obrázku. Pokud je obrázek správně naformátován, po přidání obrázku do tile palette, by se dlaždice měly automaticky rozpoznat a objevit v seznamu dlaždic v paletě.

**Obsah obrázku text, snímek obrazovky, výsledková tabule

Popis byl vytvořen automaticky**

Obrázek Editor středu místnosti

**Generování levelů**

Pro náhodné generování autor vytvořil script, který na počátku vytvoří začáteční místnost a následně posune generační bod do náhodného směru. Poté algoritmus začne rozmisťovat body generace (na minimapě jsou tyto body znázorněny barevnými čtverci). Pokud má level obsahovat speciální typy místností jako jsou obchod či místnost s truhlou, označí náhodný bod v určeném intervalu jako speciální místnost. Poté se každému bodu přiřadí obrys (stěny s vchody a východy), který propojí všechny sousedy v okolí místnosti. Jako poslední krok přiřadí speciálním místnostem jejich střed a zbytku střed náhodný.

if (IncludeShop)

{

int shopSelector = Random.Range(MinDistanceToShop, MaxDistanceToShop);

ShopRoom = layoutRoomObjects[shopSelector];

layoutRoomObjects.RemoveAt(shopSelector);

ShopRoom.GetComponent<SpriteRenderer>().color = ShopColor;

}

Ukázka kódu Kód pro umístění bodu speciální místnosti „obchod“.

if (IncludeGunRoom)

{

if (outline.transform.position == GunRoom.transform.position)

{

Instantiate(CenterGunRoom, outline.transform.position, transform.rotation).Room = outline.GetComponent<Room>();

generateCenter = false;

}

}

Ukázka kódu Kód pro vygenerování speciální místnosti „místnost se zbraní“

public void MoveGenerationPoint()

{

GeneratorPoint.position += SelectedDirection switch

{

Direction.Up => new Vector3(0f, yRoomOffset, 0f),

Direction.Down => new Vector3(0f, -yRoomOffset, 0f),

Direction.Left => new Vector3(-xRoomOffset, 0f, 0f),

Direction.Right => new Vector3(xRoomOffset, 0f, 0f),

=> GeneratorPoint.position

};

}

Ukázka kódu Kód upraveného switch case pro posun generačního bodu

### Testování

V rámci testování byla vytvořená hra sestavena v Unity enginu a vydána na platformu k digitální distribuci her Itch.io. Ke hře byl připnut kontakt na autora pro zpětnou vazbu a hra byla v průběhu času podle zpětné vazby upravována. Celkem přišlo 19 zpráv se zpětnou vazbou. Nejčastějšími připomínkami bylo:

Tabulka Zpětná vazba ke hře a jejich řešení

|  |  |  |
| --- | --- | --- |
| **Zpětná vazba ke hře** |  |  |
| Připomínky | Řešení | Poznámka autora práce |
| Bodce poškozují hráče ve velkém okruhu. | Bodcům byla zmenšena kolizní obálka. |  |
| Nepřátelé útočí na hráče hned po vstupu do místnosti, takže hráč nemá šanci se ani zorientovat. | Všechny místnosti s nepřáteli jsou při vstupu zmraženy na 0.5 sekundy. |  |
| Hráč se pohybuje moc pomalu. | Hráčova rychlost byla zvýšena. |  |
| Nepřátelé jsou moc silní. | Neřešeno. | Nepřátelé a hra celkově byla navrhnuta, aby prověřila hráčovi schopnosti. Obtížnost nepřátel v závislosti na rychlosti pohybu, ekonomickému systému hry, zbraním a dalších faktorů tomu byla uzpůsobena. Proto tato připomínka nebyla nijak řešena. |
| Boss působí nezajímavě, je jednoduchý na zdolání. | Bossova sekvence akcí byla upravena, byla přidána další akce, kde boss vytváří další nepřátelé čímž se zvýšila obtížnost. |  |
| Přidání zbraní na blízko, aktivní předměty na jedno použití jako granáty, házecí nože apod. | Neřešeno. | Z časových důvodů nebylo možné implementovat nové funkce do hry. Ale v budoucích verzích hry je v plánu tyto funkce přidat. |

# Shrnutí výsledků

V práci byl shrnut postup vytváření konceptu hry za využití principů herního návrhu. Práce se zabývala správným navržení herního designu a problematikou návrhu a vysvětlila co je to dokument herního designu.

Práce dále pojednávala o tom, jak správně navrhnout hratelnost pomoci několika principů návrhu hratelnosti, tak aby měl hráč co největší přehled o tom, co se ve hře děje, kam má jít a co by měl očekávat. V práci byly dále popsány herní mechanismy. Práce se zabývala popisem některých z důležitých herních mechanismů a jak je správně použít při vývoji her. Práce popisovala herní prostory, v kterých se hry odehrávají, jaké stavy a atributy mohou mít objekty. Práce přibližovala pohled na jednotlivá herní pravidla, dovednostní mechanik a jak prvky náhody zvyšují hráčův požitek ze hry. V práci bylo zmíněno, jak se tvoří levely hry, včetně některých typů levelů a jak vytvořit takový level, který bude mít správný tok a hráč se v něm nebude zasekávat. Práce se zabývala tím, jak levely udělat zajímavé pomocí správného umístění a objektů a nepřátel. Práce dále pojednávala o tom, co by měl obsahovat dobrý design nepřátel a bossů, jejich umístění ve hře a chování, a které typy chování jsou dobré v určitých příležitostech. V práci bylo zmíněno procedurální generování, k čemu lze využít a co lze pomocí procedurálního generování vytvořit. Dále práce popsala funkčnost některých z typů procedurálního generování levelů.

Práce dále zmínila základní informace o populárních herních enginech. Na základě autorových kritérii a zkušeností dané enginy porovnala a kritéria bodově ohodnotila. Engine s nejvíce body byl zvolen pro praktickou část.

Poslední část se zabývala aplikováním poznatků z teoretické části při vývoji 2D hry ve vybraném enginu. Práce popsala jednotlivé části v dokumentu herního designu a dále se zabývala popisem některých důležitých částí využitých při vývoji her a k nim i příklad, jak byly při vývoji použity.

# Závěry a doporučení

V rámci této práce byla vytvořena počítačová 2D hra v Unity Enginu, která byla následně distribuována na platformu Itch.io. Práce bude v budoucnu jistě rozšířena o další funkcionality, a to z důvodu, že hra není v plné verzi. Do hry budou postupně přidávány funkce s ohledem na připomínky testerů a představy autora hry.

Vzhledem k jednočlenné práci na hře nebylo možné implementovat vše, co by správná hra měla mít. Příkladem, není implementován příběh, malá paleta nepřátel a možných místností, animace nejsou dostačující, generování levelů je potřeba nutně vylepšit, AI nepřátel je velmi jednoduché a do budoucna bude nutné implementovat obcházení překážek.

Práce naplnila autorova očekávání a cíle práce byly splněny.

# Seznam použité literatury

1. SALEN, Katie and Eric ZIMMERMAN. Rules of play: Game design fundamentals. B.m.: The MIT Press, 2010.
2. FEBRUARY 27, Matt Allmer Blogger a 2009. The 13 Basic Principles of Gameplay Design. *Game Developer* [online]. 27. únor 2009 [vid. 2023-01-25]. Dostupné z: https://www.gamedeveloper.com/design/the-13-basic-principles-of-gameplay-design
3. SCHELL, Jesse. The Art of Game Design: A book of lenses. B.m.: CRC Press/Balkema, 2008.
4. What are Video Game Mechanics? (Learn for Free) [online]. 11. březen 2017 [vid. 2023-02-06]. Dostupné z: https://www.gamedesigning.org/learn/basic-game-mechanics/
5. PARLETT, David. Oxford History of Board Games. Oxford; New York: Oxford University Press, 1999. ISBN 978-0-19-212998-7.
6. 40 % světové populace hraje hry – INDIAN [online]. [vid. 2023-02-13]. Dostupné z: https://indian-tv.cz/clanek/40-svetove-populace-hraje-hry-8i1cwa
7. Global gaming penetration by country 2022. Statista [online]. [vid. 2023-02-13]. Dostupné z: https://www.statista.com/statistics/195768/global-gaming-reach-by-country/
8. How Many Gamers Are There? (New 2023 Statistics). Exploding Topics [online]. 7. říjen 2022 [vid. 2023-02-13]. Dostupné z: <https://explodingtopics.com/blog/number-of-gamers>
9. HALPERN, Jared. Developing 2D games with Unity: indipendent game programming with C#. [New York, NY]: APress, [2019]. ISBN 9781484237717.
10. BOEN, James. Sound in Video Games: How Sound Is an Important Aspect of the Virtual Experience. 2021
11. FULLERTON, Tracy. Game Design Workshop: A playcentric approach to creating innovative games. B.m.: CRC Press, 2019.
12. ROGERS, Scott. Level up!: The guide to great video game design. Hoboken: John Wiley &amp; Sons, 2014.
13. TOTTEN, Chris. *Game character creation with blender and unity*. Indianopolis, Ind: John Wiley & Sons, Inc, 2012.
14. CO, Phil. *Level design for games: Creating compelling game experiences*. Berkeley: New Riders., 2006.
15. IAN SCHREIBER and BRENDA BRATHWAITE. Challenges for Game Designers. Boston, Mass: Course Technology, 2009.
16. TOTTEN, Christopher W. An architectural approach to level design. Boca Raton: CRC, Taylor &amp; Francis Group, 2014.
17. KOSTER, Raph. A theory of fun for game design. Sebastopol, CA: O'Reilly Media, Inc., 2014.
18. Level design types. Games Design Course - Harrison Preston [online]. 17. září 2018 [vid. 2023-02-25]. Dostupné z: <https://gamesdesign658.wordpress.com/level-design-types/>
19. Flow [online]. [vid. 2023-02-26]. Dostupné z: <https://book.leveldesignbook.com/process/layout/flow>
20. Level Flow - Valve Developer Community [online]. [vid. 2023-02-28]. Dostupné z: <https://developer.valvesoftware.com/wiki/Level_Flow>
21. FEBRUARY 05, Tanya X. Short Blogger a 2014. Level Design in Procedural Generation. Game Developer [online]. 4. únor 2014 [vid. 2023-02-28]. Dostupné z: <https://www.gamedeveloper.com/design/level-design-in-procedural-generation>
22. Dille, F., &amp; Platten, J. Z. (2011). The ultimate guide to video game writing and Design. Lone Eagle.
23. Enemy design [online]. [vid. 2023-03-01]. Dostupné z: https://book.leveldesignbook.com/process/combat/enemy
24. Game AI Basics - Patrolling. Mike Xiao [online]. 14. říjen 2017 [vid. 2023-03-02]. Dostupné z: http://magickaichen.com/game-ai-basics-create-a-wandering-enemy/
25. Characteristics of Good Enemy Design [online]. 2020 [vid. 2023-03-02]. Dostupné z: <https://www.youtube.com/watch?v=01wMwld2_38>
26. GAMEDESIGNLOUNGE. Making a Video Game Boss: Design and Structure. Game Design Lounge | Video Game Level, Character, and Story Design [online]. [vid. 2023-03-02]. Dostupné z: https://gamedesignlounge.com/making-a-video-game-boss/
27. FINALBOSS. The History of Video Games: From Pong to Fortnite. FinalBoss [online]. 21. prosinec 2022 [vid. 2023-03-03]. Dostupné z: <https://finalboss.io/history-video-games/>
28. SHARMA, Aakrit. How many people play Minecraft? 2023 player count. Charlie INTEL [online]. 15. únor 2023 [vid. 2023-03-03]. Dostupné z: <https://www.charlieintel.com/how-many-people-play-minecraft-2022-player-count/195437/>
29. JOHN. The History of Video Games. Culture of Gaming [online]. 22. prosinec 2022 [vid. 2023-03-03]. Dostupné z: <https://cultureofgaming.com/the-history-of-video-games/>
30. Short, T. X., &amp; Adams, T. (2017). Procedural generation in Game Design. Taylor &amp; Francis, CRC Press.
31. TALLE, Job. Random procedural generation [online]. [vid. 2023-03-04]. Dostupné z: <https://jobtalle.com/random_procedural_generation.html>
32. HENDRIKX, Mark, Sebastiaan MEIJER, Joeri VELDEN a Alexandru IOSUP. Procedural Content Generation for Games: A Survey. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) [online]. 2013, 9. Dostupné z: doi:10.1145/2422956.2422957
33. Christian Mills - Notes on Procedural Map Generation Techniques [online]. 9. prosinec 2021 [vid. 2023-03-04]. Dostupné z: <https://christianjmills.com/posts/procedural-map-generation-techniques-notes/index.html>
34. SCHARDON, Lindsay. What is Unity? – A Guide for One of the Top Game Engines. GameDev Academy [online]. 13. leden 2023 [vid. 2023-03-04]. Dostupné z: <https://gamedevacademy.org/what-is-unity/>
35. Godot Engine [online]. C++. B.m.: Godot Engine. 4. březen 2023 [vid. 2023-03-04]. Dostupné z: <https://github.com/godotengine/godot>
36. Unreal Engine 5 - Unreal Engine [online]. [vid. 2023-03-04]. Dostupné z: <https://www.unrealengine.com/en-US/unreal-engine-5>
37. LLP, Vasundhara Infotech. 2D Game VS 3D Game: Which is Better for Game development? Vasundhara Infotech llp [online]. 5. říjen 2022 [vid. 2023-03-06]. Dostupné z: https://vasundhara.io/blogs/2d-game-vs-3d-game

*Oskenované zadání práce*