

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Vývoj 2D hry s využitím Unity engine**

Bakalářská práce

Autor: Jan Dušek

Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Michal Macinka

Hradec Králové

Duben 2021

**Prohlášení:**

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 28.4.2021

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Michalu Macinkovi za metodické vedení práce a jeho cenné rady.

## **Anotace**

Bakalářská práce pojednává o tématu vývoje počítačových her. Kromě počítače představuje i ostatní oblíbené platformy a zároveň přibližuje základní pojmy v oblasti videoher. V první části jsou představeny kromě Unity engine i další možné technologie pro tvorbu videoher a následné jejich porovnání. Práce rozebírá různé dimenze prostoru, jejich kombinace a poté konkrétní příklady. Dále je v práci popsán, a současně i vypracován herní návrh, včetně Game Design Dokumentu, jenž má za úkol tvořit hlavní osnovu pro vývoj počítačové hry. Poslední část práce seznamuje s prostředím Unity engine a společně s tím, popisuje implementaci projektu. Bakalářským projektem je funkční počítačová 2D hra vytvořena za pomoci engine Unity.

## **Annotation**

### **Title: Development of 2D game using Unity engine**

The bachelor thesis focuses on the development of computer games. Apart from the computers, it also introduces other popular platforms as well as the basic terms of the area of video games. The first part is not related only to the Unity Engine but also additional possible technologies for the development of the video games together with their comparison. The thesis describes different dimensions and their combinations leveraging particular examples. Furthermore, there is the description and proposal of the PC game, including the Game Design Document, that represents the baseline for PC game development. The last part of the thesis introduces the environment of Unity engine together with the description of the implementation project. The bachelor project is a fully working 2D game developed with the leverage of the Unity engine.

## Obsah

1	Úvod .....	1
2	Cíl práce.....	2
3	Videohry a jejich platformy.....	3
3.1	Videohra.....	3
3.1.1	Dělení her .....	3
3.1.2	Hra více hráčů.....	4
3.2	Platformy .....	5
3.2.1	Počítač .....	5
3.2.2	Konzole.....	6
3.2.3	Smartphone .....	6
3.2.4	Ostatní platformy .....	6
3.2.5	Srovnání platforem .....	7
4	Technologie pro vývoj her.....	9
4.1	Game engine .....	9
4.1.1	Komponenty game engine.....	9
4.2	Unity engine.....	11
4.3	Unreal engine .....	12
4.4	CryEngine .....	13
4.5	GameMaker engine .....	14
5	Dimenze prostoru .....	15
5.1	2D hra .....	15
5.2	3D hra .....	15
5.2.1	Porovnání 2D a 3D her .....	16
5.3	2.5D hra .....	17

5.4	Projekce prostoru .....	18
6	Návrh a herní design.....	20
6.1	Obchodní parametry .....	20
6.1.1	Nízkorozpočtový projekt .....	21
6.1.2	Fixní rozpočet a pevné datum vydání.....	22
6.1.3	Vysokorozpočtové prvotřídní projekty.....	22
6.1.4	Obchodní parametry v praxi .....	23
6.2	Koncept hry.....	23
6.2.1	Unifikovaný Modelovací Jazyk.....	24
6.2.2	Vznik konceptu hry .....	25
6.3	Herní design.....	27
6.3.1	Obsah.....	28
6.3.2	Úvod .....	28
6.3.3	Herní mechaniky.....	29
6.3.4	Umělá inteligence .....	30
6.3.5	Herní prvky.....	30
6.3.6	Příběh.....	31
6.3.7	Herní progres.....	31
6.3.8	Menu.....	31
6.3.9	Tvorba Game Design Dokumentu.....	32
7	Implementace počítačové hry .....	33
7.1	Orientace v prostředí Unity.....	33
7.2	Základní komponenty herních objektů .....	34
7.2.1	Pohyb.....	34
7.2.2	Kolize .....	35
7.2.3	Částicový systém .....	35

7.2.4	Zvuk.....	36
7.3	Skriptování.....	37
7.3.1	Základní funkce.....	37
7.3.2	Ovládání .....	39
7.4	Export hry .....	40
8	Závěr.....	41
9	Seznam použité literatury .....	42
10	Přílohy .....	47
10.1	Příloha 1 – Game Design Dokument.....	47

## Seznam obrázků

Obrázek 1 Ukázka Behavior tree. ....	11
Obrázek 2 Rovnoběžné promítání (vlevo) a středové promítání (vpravo). ....	18
Obrázek 3 Pohledový objem rovnoběžného (vlevo) a středového (vpravo). ....	19
Obrázek 4 Trojúhelník projektu. ....	20
Obrázek 5 Fáze návrhu a vývoje. ....	21
Obrázek 6 Diagram případů užití bakalářského projektu. ....	27
Obrázek 7 Rozložení Unity editoru. ....	34
Obrázek 8 Ukázka komponenty Částicový systém. ....	36
Obrázek 9 Funkce Instantiate a GetComponent. ....	38
Obrázek 10 Lineární interpolace. ....	38
Obrázek 11 Nastavení exportu hry. ....	40



## **Seznam tabulek**

Tabulka 1 Srovnání 2D a 3D her .....	16
Tabulka 2 Případy užití bakalářského projektu .....	26

# 1 Úvod

Vzhledem k dostupnosti a oblíbenosti počítačových technologií, stoupá nabídka souvisejících produktů na poli zábavního průmyslu. Těmito produkty jsou například filmy, seriály i počítačové hry. Právě počítačové hry jsou vyvíjeny velkými i menšími studii všude po světě. Velké množství vývojářských studií na trhu zaručuje hráči široký výběr her z nejrůznějších žánrů. Pro vývoj elementární počítačové hry, není podmínkou absolvovat specializované středoškolské nebo vysokoškolské vzdělání. Díky obrovské komunitě herních vývojářů existuje mnoho návodů, šablon nebo videí, které přispívají ke správnému postupu při vytváření hry.

Firmy, jakou jsou Sony, Nintendo a Ubisoft, mají možnost pověřit vývojem své hry i několik stovek vývojářů, což má dopad na výslednou kvalitu grafického zpracování. Avšak i malá studia, někdy i jednočlenná, převzou prestižní ocenění za jimi vytvořenou hru, a to právě díky originálnímu nápadu. Takové hry nesou označení „Indie hra“ (z anglického Independent Game), neboli nezávislá hra.

Software, který je přizpůsobený pro tvorbu videoher, se nazývá game engine. Každý má svá specifika a je tedy vhodný pro jiné účely. Velké firmy mají finanční a časovou kapacitu na to, aby si mohli příslušný software vyvinout sami. Stejně tak i sami modelují grafické prvky hry. Ostatní vývojářská studia musí vybrat takový engine, jež nejvíce vyhovuje jejich požadavkům. Existuje mnoho firem, které nabízejí svůj engine vývojářům pro tvorbu videoher. Firmy s takovou nabídkou jsou například Epic Games, nebo Unity. Zároveň zprostředkovávají internetový obchod s „assets“, tedy s již vymodelovanými objekty připravenými k použití do vývoje. Všechny firmy, nabízející svůj produkt, se snaží ze svých služeb profitovat, ať už v podobě měsíčních poplatků, nebo provizí z prodaných her.

## **2 Cíl práce**

Cílem práce je seznámit se základními pojmy v oblasti videoher, včetně jejich platforem a pomoci tak k lepší orientaci v oblasti herního vývoje. Práce má za úkol popsat a porovnat populární technologie pro vývoj her, jako jsou Unreal engine, nebo Unity engine. Rozebírá problematiku dimenze počítačové hry, případně kombinaci dimenzí v jedné hře. Dále se zabývá šablonou pro vývoj hry, tzv. Game Design Dokument. Dokument je vyplněn za použití reálného příkladu v podobě 2D hry, která tvoří bakalářský projekt této práce.

## 3 Videohry a jejich platformy

### 3.1 Videohra

Je software, jehož hlavní účel je poskytnout uživateli zábavu, relaxaci, nebo rozvoj osobnosti [10]. Videohra nemusí mít svůj hlavní záměr vždy pobavit. Například didaktické hry, které vznikají s výchovným úmyslem, využívají zábavnost počítačové hry a zároveň učí a rozvíjí jedince.

#### 3.1.1 Dělení her

Existuje mnoho kritérií, podle kterých lze videohry dělit. Časté rozdělení je například podle žánru, platformy, přístupu k internetu a další. V odborném článku [1] jsou rozebírány nejběžnější herní žánry a platební modely her, které jsou rozvinuty v následujících kapitolách. Pět nejvíce populárních žánrů jsou:

1. akční,
2. RPG,
3. strategické,
4. adventura,
5. sportovní. [19]

V Akčních hrách má hráč za úkol pomocí zbraně zneškodnit nepřítele, který je ovládán jiným hráčem, nebo počítačem. Akční hry jsou rozděleny na tzv. „střílečky“ z první osoby (z anglického First Person Shooter, FPS), kdy je postava ovládána z pohledu první osoby nebo hry využívající pohled z třetí osoby (z anglického Third Person Shooter, TPS). V žánru rolové hry (z anglického Role Playing Game, RPG), hráč ovládá svého hrdinu (postavu ve hře) a putuje s ním po herním světě, posouvá se v příběhu hry pomocí plnění úkolů s cílem vylepšení postavy nebo získání bodů. Pro hraní strategických her je předpokladem strategické myšlení. Hráč buduje a rozvíjí svoje město, území, či armádu. Cílem těchto her je zvolit správnou strategii a splnit buď úkol budovatelský, nebo zničit protivníka. Adventura je žánr, ve kterém je zapotřebí rozluštit hádanky, rébusy, záhady, aby se hráč posunul v ději hry. Běžně se používá motiv vyšetřování zločinu, popřípadě honba za pokladem. Sportovní hry jsou založeny na simulaci sportu, kdy hráč ovládá tým, nebo jedince. [19]

Dalším kritériem dělení videoher jsou finanční náklady spojené s pořízením, či hraním hry. Premium hry umožňují neomezené hraní po jednorázovém poplatku. Freemium hry je možné zdarma nainstalovat a hrát, ovšem pro přidání bonusového obsahu, nebo nějaký druh výhody, je už třeba zaplatit poplatek. Freeware hry využívá hráč v plném rozsahu, aniž by musel vložit finanční částku. Zisky z provozování těchto her závisí na reklamách vkládaných do her. Hry s periodickým poplatkem umožňují hráči přístup pouze po dobu, kterou si zaplatil. Hráč si musí koupit herní čas.

### **3.1.2 Hra více hráčů**

Multiplayer umožňuje hrát stejnou hru i s ostatními lidskými hráči. Hra jednoho hráče je singleplayer. Offline multiplayer nevyžaduje připojení k internetu, proto je nutné, aby hráči byli fyzicky u sebe. Takový multiplayer může být zprostředkován na jedné obrazovce pomocí rozdělené obrazovky (z anglického Split Screen), nebo s využitím režimu tzv. „horké křeslo“ (z anglického Hot Seat). Offline multiplayer na více počítačích je možný po vytvoření lokální sítě, na které je připojeno herní zařízení každého hráče. Pomocí rozdělené obrazovky, nejčastěji na dvě poloviny, mají oba hráči možnost ovládat hru na stejném počítači. Hráč může svým ovládním ovlivnit pouze svoji přidělenou část obrazu. V režimu „horké křeslo“ je cílem hráče získat co nejvyšší bodové hodnocení v daném čase, po uplynutí času je na řadě další hráč, který se za stejných podmínek pokouší o to samé. Vítězem se stává ten, kdo si po všech pokusech udržel nejvyšší skóre.

Při hraní online multiplayeru hráč soutěží s ostatními hráči, obvykle v reálném čase. Pokud není nutné, aby hráči hráli hru současně, jedná se o asynchronní multiplayer, příkladem mohou být online tahové hry. Typickým představitelem online her v reálném čase jsou FPS hry, kdy se hráči rozdělí do týmů a hrají proti sobě. Strategie fungují na stejném principu, jenom s menšími týmy, častokrát proti sobě hrají i jedinci.

Specifičtější online žánry jsou MOBA, MMO a Battle Royale hry. Tzv. „online bitevní aréna“ (z anglického Multiplayer Online Battle Arena, MOBA), je hra, kdy v uzavřené aréně soupeří dva týmy. Jeden tým většinou o velikosti 5 hráčů se snaží zničit nepřátelskou základnu. Každý hráč si zvolí svého hrdinu, který je po celou jednu hru neměnný. Průkopníkem tohoto žánru jsou hry League of Legends, nebo Dota 2. Battle royale hry podle [15], zažili největší nárůst hráčů v roce 2017 díky vydání hry PlayerUnknown's Battlegrounds (zkráceně PUBG) společností BlueHole, které se za rok

a půl prodalo přes 50 milionů kopií. V PUBG je na velkou mapu vpuštěno 100 hráčů s prázdným inventářem. Každý hráč se snaží najít zbraň, neprůstřelnou vestu, helmu, náboje, auto a další věci, které mu pomůžou v boji. V několika minutových intervalech je zmenšováno území, v kterém hráč musí být, pokud je mimo povolené území příliš dlouho, tak ve hře končí. Dochází k bojům a ten co zůstane jako poslední naživu vyhrál. Na principu poslední žijící muž (z anglického Last Man Standing) je založen každý Battle Royale, hry se liší hlavně ve vizualizaci a možnostech během hry. Krátce po vydání PUBG, vývojáři z Epic Games vytvořili konkurující hru Fortnite, která je zdarma a svojí popularitou předčila PUBG. Hry s obrovským počtem online hráčů (z anglického Massively Multiplayer Online Game, MMO) je možné rozdělit až do 16 kategorií [23]. Hry jsou založeny na předpokladu velkého až masivního počtu hráčů online v reálném čase sdílející stejný svět. Typickým žánrem pro MMO je RPG, z toho potom zkratka MMORPG.

Další příklady žánrů v MMO:

- strategie v reálném čase (MMORTS),
- MOBA (MMOBA),
- sportovní hry (MMOSG),
- FPS (MMOFPS).

## **3.2 Platformy**

Herní trh převyšuje v roce 2020 hodnotu 150 miliard amerických dolarů. Nárůst oproti předcházejícímu roku 2019 je roven 9,3 %. Přičemž, na celém světě je odhadem přes 2,5 miliardy hráčů [43]

### **3.2.1 Počítač**

Počítač je po smartphonu nejvíce populární herní platforma ze všech. Titul první počítačové hry je připisován hře Spacewar!, vyvinutá v roce 1962 studenty v Massachusettském technologickém institutu. Je to hra pro dva hráče, každý ovládá svoji vesmírnou loď, snaží se zničit protihráče a zároveň se vyhýbá hvězdě uprostřed obrazovky [3]. Ve srovnání s jinými platformami má počítač velký výkon, a tedy i vyšší pořizovací cenu. Platforma PC je i nejoblíbenější pro herní vývojáře, konkrétně 56 % herních vývojářů se v roce 2020 podílelo na vývoji PC hry [43]. Mezi přední prodejce

herních počítačů patří Dell se svojí řadou Alienware, MSI, HP, nebo Asus se svojí řadou Republic of Gamers.

### **3.2.2 Konzole**

Konzole je třetí nejpobulárnější platforma, ve Velké Británii předčila svou popularitou i počítač. Ve spojených státech amerických je zhruba ve stejné oblíbě jako počítač [17]. V roce 1972 spustila prodej první herní konzole Magnavox Odyssey, následována o tři roky později konzolí Atari's PONG od firmy Atari. První hry byly převážně sportovního žánru hlavně ping-pong, tenis, házená a volejbal [20]. Trh s konzolemi vede firma Sony se svým Playstation, pro tuto platformu vyvíjí 25 % herních vývojářů. Největší konkurent firmy Sony je Microsoft a jeho Xbox, kde 23 % vývojářů zabývající se vývojem Xbox her je jen o trošku menší než u Playstationu. Jako další a poslední zmiňovaná konzole je Nintendo Switch od firmy Nintendo, která má pouze 3 % herních vývojářů [43].

### **3.2.3 Smartphone**

V posledních letech je nejpobulárnější platforma smartphone. Ve Spojených státech amerických je ze všech mobilních uživatelů 72.3 % zároveň mobilních hráčů, což znamená, že pouze v USA je více než 210 milionů smartphone hráčů [43]. Historicky první hra na mobil byla představena v roce 1994 s názvem Tetris, běžící na mobilu s označením Hagenuk MT-2000. V roce 1997 Nokia představila hru Snake ve svém mobilu Nokia 6110, která byla vložena do více než 400 milionu zařízení, a tím se stala nejhranější hrou světa. Díky infračerveným portům Nokie 6110 hra Snake byla první multiplayer hra na mobilu [18]. Mobily se systémem Android, nebo iOS tvoří většinu smartphonů a jsou nejčastějším systémem pro hraní mobilních her. Výhodou je jednoduché stažení z Google Play pro systém Android, nebo z App Store pro systém iOS.

### **3.2.4 Ostatní platformy**

Současným, ne ještě úplně rozvinutým odvětvím herního průmyslu je virtuální realita (zkráceně VR) a rozšířená realita (zkráceně AR). AR doplňuje reálný svět nasnímaný kamerou grafickými prvky. Příkladem může být vložení bitevního pole do reálného prostoru, který si hráč zvolí, toto bitevní pole je potom vidět například na stole, který má hráč před sebou. Cílem VR je vytvořit zcela nový virtuální prostor, který je hráči předáván pomocí uzavřené náhlavní soupravy, která vytváří iluzi, že hráč je součástí tohoto

prostoru. Pomocí ovladačů v ruce se ovládá prostředí, v kterém se hráč nachází. Pro minimalizaci počtu kabelů technologické firmy nabízejí herní batohy, které mají hráči na zádech, a tak nejsou limitováni kabeláží. 14 % všech herních vývojářů vyvíjí hry pro platformu VR [43]. Ze světových her jsou tu tituly jako Half-Life: Alyx, The Elder Scrolls V: Skyrim, nebo Beat Saber vyvinuté českým studiem Beat Games.

Méně významnou herní platformou je televize podporující připojení k internetu. Smart TV nabízí ve svých online obchodech hry. Hry je možné ovládat televizním ovladačem s gyroskopem nebo je možné připojit externí herní ovladač.

### **3.2.5 Srovnání platform**

Při výběru vhodné platformy je zásadní místo, kde se hra hraje. Například v dopravních prostředcích je ideální volbou mobil nebo tablet. Příznivci her na mobilu mohou v domácím prostředí využít výhodu velké obrazovky televize. Mobily často slouží jako externí spárovatelné ovladače k televizi. V domácím prostředí je možné volit mezi konzolí a počítačem.

V článku [24] jako hlavní výhoda konzole je uvedena menší pořizovací cena na rozdíl od počítače. Z toho důvodu, že konzole jsou určeny a vyrobeny primárně k hraní her, je možné optimalizovat výkon především na hraní, a tím se sníží pořizovací náklady. Mezi výhody konzole patří také komfort při hraní. Pokud chce hráč hrát online, musí platit poplatek za možnost připojení k oficiálním serverům. U Sony je to Playstation Plus a u Microsoftu Xbox Live Gold. To ovšem neplatí pro všechny konzolové hry. Zpětná kompatibilita je jeden ze záporů konzolí. Novější generace konzolí většinou neumožňují hrát hry starší generace. Je proto zapotřebí opětovné zakoupení hry nebo použití konzole starší generace.

Pro počítač je vyvíjeno více her než pro konzoli, proto je na výběr z více titulů. Univerzálnost neboli všestrannost počítače je dalším kladem. Je mnoho způsobů, jak si sestavit svoje vlastní herní zařízení. Jedním z nejčastějších problémů je vědět jaké správné komponenty použít, nestačí totiž mít pouze výkonnou grafickou kartu, protože by ji ostatní části zpomalovali. Kdežto u konzolí je to jasně dáno a pouze vzácně má uživatel potřebu upgradovat jednotlivé části. Jedna z možností je koupit si levnější počítač, na kterém je možné spustit aktuální hry, ale vzhledem k rychlému vývoji, za poměrně krátký čas výkon na stále aktuálnější hry nebude stačit. Druhou možností je



koupit velice výkonný počítač s mnohdy i několikanásobnou pořizovací cenou, ale ani tak není jisté, jak dlouho bude výkon počítače stačit. [24]

## 4 Technologie pro vývoj her

Zpočátku byl software pohánějící videohry silně závislý na dotyčné hře a na hardwaru, na němž běžela. První změna nastala v polovině 90. let minulého století v souvislosti s populární hrou Doom od studia id Software. Architektura Doom měla chytře oddělené hlavní softwarové komponenty od vizuálních prvků. Velká hodnota architektury hry Doom byla evidentní ihned po zveřejnění. Díky zakoupené licenci vývojáři vytvářeli nové hry s minimálním počtem změn v původních softwarových komponentách. Koncem 90. let byly vydané již takové hry, u kterých se předpokládala pozdější modifikace. Skriptovací jazyky zajišťovaly vysokou přizpůsobivost. Tento typ architektury je nazýván jako game engine. Tedy znovupoužitelné klíčové softwarové komponenty, které díky licenci používají vývojářská studia pro vývoj svých her. [13]

### 4.1 Game engine

Na základě [26], game engine zastává hlavní roli při vytváření hry. Pomáhá vývojářům vytvářet jednotlivé scény, charaktery, zvuky, umělou inteligenci, animace, multiplayer a tak dále. Game engine obsahuje sadu předpřipravených vývojářských nástrojů sloužících k vizualizaci a také pomocné softwarové komponenty. Jedná se o framework upravený speciálně pro konstrukci a vývoj videoher. Používá se pro vytváření konzolových, mobilních i počítačových her.

#### 4.1.1 Komponenty game enginu

Všechny komponenty jsou integrovanou součástí game enginu pro podporu vývoje hry. Hlavní komponenty se starají o vstup, grafiku, zvuk, fyziku a umělou inteligenci [26]. Kromě těchto, většina dnešních game enginů nabízí kompletní podporu pro online hraní a tím výrazně ulehčí implementaci komunikace TCP či UDP protokolem do hry.

Hra by měla zajistit podporu pro velké množství vstupních zařízeních jako je počítačová myš, gamepad, joystick a další. Dva nejvíce používané způsoby uchopení vstupu jsou prostřednictvím dotazování (z anglického Polling), nebo pomocí událostí (z anglického Events). Při používání druhé zmíněné techniky událostí probíhá zachycení vstupu například v podobě kliknutí myši, nebo zmáčknutí specifické klávesy. Konkrétní kód je pak spuštěn na základě vstupu, který počítač přijal. Technika dotazování se používá při zjišťování pozice ukazatele. Typicky se monitoruje pomocí souřadnic x a y. Kromě toho

jsou vypočítávány různé úhly, které svírá herní ovladač, nebo v jaké pozici a natočení se nachází smartphone. [26]

Úroveň grafického zpracování hry, má velký dopad na její komerční úspěch. 3D grafika je složena z 3D assetů, které jsou vytvořeny v externích renderovacích programech. Typický software tohoto rázu je Blender. Vymodelovaný asset je poté importován do game engine, proto je nutné, aby engine podporoval více grafických formátů. Samotný game engine nabízí spoustu možností, jak asset dále upravovat. Pro maximalizování reálného dojmu disponuje vlastnostmi jako jsou světelné efekty, stíny, skládání animací apod. Zvukový engine kontroluje zvukové efekty a společně s renderovacím enginem generují 3D animovanou grafiku ve 2D obraze.

Jednou z komponent je fyzikální engine (z anglického physics engine). Jedná se o software dovolující vykonávat co nejpřesnější simulace reálného světa. Kupříkladu umožňuje provádět pohyb pevného tělesa, posunutí měkké hmoty, upravování rychlostí, simulaci dynamiky tekutin, odražení těles a další běžné jevy. Tyto integrované komplexní fyzikální enginy mají své využití především ve videohrách, kde musí simulace probíhat v reálném čase.

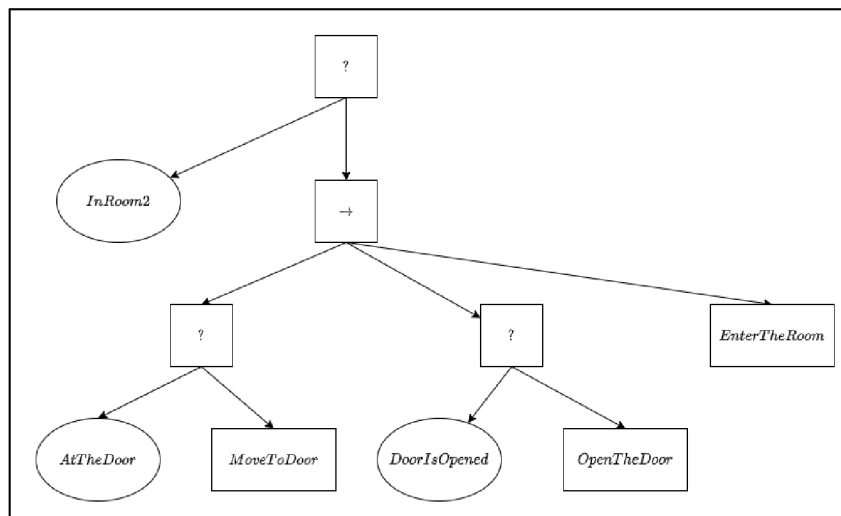
Umělá inteligence hraje velkou roli v rámci herního vývojářství. Předpovídání kroků hráče nebo zaznamenávání jeho chování, na jehož základě jsou uskutečněny příslušné akce, může být docíleno používáním specifického softwaru zabudovaného přímo ve hře. Místo psaní složitěho kódu pro umělou inteligenci, je výhodnější použít strukturu stromu chování (z anglického Behavior tree). Implementace je většinou realizována pomocí hotových scriptů, které byly navrhnuté a vytvořeny softwarovými inženýry v oboru Umělá inteligence. [26]

#### **4.1.1.1 Strom chování**

Podle [22], strom chování je jedna ze struktur pro upřesnění, či přepínání úloh v komponentě umělé inteligence. Úloha může být reprezentována jako strom chování, jež je implementován jako samostatný softwarový modul. Strom je možné složit ze dvou základních typů uzlů, každý uzel může mít status úspěch, chyba nebo běžící. Prvním typem uzlu je control flow, který se dělí na další 4 podtypy:

- Sequence – značí se šipkou „->“. Vráti úspěch svému rodiči, pokud i všichni jeho potomci vrátí úspěch.
- Fallback – značí se otazníkem „?“ . Pokud první potomek vrátí hodnotu úspěch, pak i tento uzel bude mít status úspěch.
- Parallel – značí se dvojitou šipkou. Spustí všechny své potomky, a pokud alespoň jeden vrátí úspěch, vrací úspěch i tento.
- Decorator – značí se delta „δ“. Uzel, který modifikuje status svého potomka.

Druhý typ uzlu je execution nodes, tvoří listy stromu. Může spustit požadovaný příkaz, nebo kontrolovat podmínku. Strom se čte vždy po jednotlivých úrovních, zleva doprava. Obrázek 1 řeší úlohu, při které se robot s umělou inteligencí musí přemístit na jiné místo, konkrétně do místnosti číslo 2. Nejprve se zkontroluje podmínka, zda robot není v místnosti č. 2 (InRoom2), pokud podmínka není splněna, tak pokračuje dále. Následně zajistí, aby se robot dostal ke dveřím (MoveToDoor). Pak pokud je potřeba, otevře dveře (OpenTheDoor). A nakonec vstoupí do místnosti (EnterTheRoom), čímž splní úvodní podmínku (InRoom2). Úloha je tak pomocí stromu chování úspěšně provedena.



**Obrázek 1 Ukázka Behavior tree. Zdroj: [22]**

## 4.2 Unity engine

Unity engine byl v roce 2005 představen společností Unity Technologies pro OS X. Od té doby se rozšířil na 27 platforem. Primárně je používán na vývoj a simulace pro počítače, konzole i mobilní zařízení. [11]

Na základě [8], Unity engine je používán především v tvorbě nezávislých a mobilních her. V září roku 2019 52 % z tisíce předních mobilních her bylo vyvinuto pomocí Unity engine, stejně tak 60 % veškerého AR/VR obsahu. Protože pro studenty a začínající vývojáře je používání engine zadarmo, je pro většinu z nich první volba. Zásobný Asset store nabízí zhruba 56 000 dostupných balíčků, což znamená přes jeden milion jednotlivých assetů, které může vývojář použít. Schopnost vytvářet 2D či 3D hry velmi rychle je další silnou stránkou. Je reálné vytvořit několik prototypů téměř bez programování a následně odstartovat oficiální projekt. Navíc Unity engine poskytuje jednoduché přenesení hry na jinou platformu. Ovšem velká studia si Unity engine jako nástroj pro vytváření svých her nevolí. Tím, že je prostředí Unity přizpůsobeno/vytvořeno pro rychlý a efektivní vývoj spíše menších projektů, nedovolí vývojáři zkoumat či upravovat kód skutečně do hloubky. Proto nástroj není vhodný pro silně specifické, nevšední nebo velmi rozsáhlé videohry.

Z oficiálních zdrojů [28] je možné využívat služeb Unity zadarmo s licenci Personal, pokud zisk za poslední rok nepřekročí 100 000\$. Pokud je pro případného uživatele podmínka nevyhovující, pak může využít vyšší předplatné Plus, které stojí 400\$ ročně, přičemž hranice zisku nesmí přesáhnout 200 000\$. Licence, které nejsou nijak limitovány jsou Pro a Enterprise. Z nichž levnější varianta Pro stojí 1 800\$ ročně.

### **4.3 Unreal engine**

Unreal engine byl uveden na trh v roce 1998 firmou Epic games. V současné době podporuje širokou škálu platform od Windows, Playstation, Xbox, macOS, přes AR, VR, Android, iOS až po SteamOS, či HTML5. Na rozdíl od Unity engine je preferovaný především pro tvorbu prvotřídních a rozsáhlejších titulů. Díky technickým možnostem, propracovanému fyzikálnímu a destruktivnímu systému je v oblibě u velkých studií. Společně s Unity engine dominují vývojářské scéně. [9]

Typickým rysem Unreal engine je fotorealistické grafické zpracování, což zapříčiňuje, že má kromě videoher využití například v architektuře, nebo ve filmu. Flexibilita patří mezi hlavní důvody, proč se tento engine setkává s vysokou popularitou. Přístup ke zdrojovému kódu je zdarma stažitelný ze serveru GitHub. Proto je možné jeho přizpůsobivost maximalizovat na rozdíl od Unity, které zpřístupní zdrojový kód jenom pro vyšší licence. Další klady jsou například vestavěné funkce dovolující rychle vytvářet

prototypy a návrhy bez větší znalosti programování. Renderovací vlastnosti dovolují adaptovat hru na různé platformy. Díky velké oficiální a komunitní podpoře je zajištěn obsáhlý obchod s assety. I když vyniká v tvorbě otevřených světů, paradoxně je možné rychle dosáhnout jeho limitů velikosti, což může znatelně ovlivnit vývoj. Vzhledem k jeho komplexnosti, může být obsluha náročná, rozhodně více než v případě Unity. Nehodí se tedy pro tvorbu malých her, protože příliš mnoho možností zpomaluje vývoj. Unreal engine byl primárně určen pro „střílečky“ a čím specifitější, či odlišnější je hra, tím větší technické znalosti jsou potřeba. [9]

Za používání enginu se neplatí měsíční poplatky, ale provize. Kontrola zisků probíhá každé čtvrtletí vyplněním online formuláře. Používání je plně zadarmo, do chvíle, než zisk přesáhne hodnotu 1 000 000\$. Po překročení této sumy si Epic Games nárokuje 5 % provize. [41]

#### **4.4 CryEngine**

Dle [7], CryEngine od firmy Crytek na sebe upozornil při zveřejnění technologické demo verze, jež ukazovala možnosti tehdejších grafických karet Nvidiá. Následovalo získání partnerství s Ubisoftem, což zaručilo strmý růst popularity CryEnginu. Původní technologické demo dalo za vznik v roce 2004 hře Far Cry. Crytek tehdy zpřístupnil engine pro získání licence a zároveň zahájil spolupráci s Electronic Arts.

Dnes engine podporuje široký výběr žánrů a formátů. Exceluje v tvorbě fotorealistických „stříleček“ a obecně vyniká v hrách z pohledu první osoby. Dále dovoluje vyvíjet pro nejznámější platformy včetně VR. Používání CryEnginu také znamená mít přístup ke zdrojovému kódu, který dovoluje větší přizpůsobivost a flexibilitu. Mnoho vývojářských studií používá silně modifikované verze a to včetně českého Warhorse Studio. Novější hry ze série Far Cry používají také modifikovanou verzi přejmenovanou na Dunia engine. Firma Amazon v roce 2015 podepsala licenční smlouvu, aby směla vytvořit svůj vlastní engine Lumberyard na základě CryEnginu.

CryEngine nabízí spoustu nástrojů, které zjednodušují vývoj. Vyniká v otevřených světech s přírodou. Stejně jako předešlé enginy, i tento dovoluje rychle a efektivně vytvářet kvalitně vypadající prototypy her. CryEngine je dobrou volbou i pro začínající vývojáře, jedná se o silný nástroj a v herní komunitě je velice uznáván. Uživatelská základna je menší než u konkurence, proto není příliš mnoho výukových materiálů. Dále

není doporučen pro „střílečky“ odehrávající se ve velké rychlosti, protože jeho fyzikální systém se na ně nehodí. Stejně tak není příliš vhodný pro komplexní RPG hry. CryEngine podobně jako Unreal engine je zadarmo s provizním systémem. Zisk nesmí překročit během jednoho roku 5 000\$, jinak si Crytek nárokuje 5 % provizi. [7]

#### **4.5 GameMaker engine**

Jak popisuje [42], GameMaker engine byl zveřejněn 15. listopadu 1999 pod názvem Animo. Je navržen pro vytváření her napříč mnoha platformami. Jeho úkolem je vytvářet kvalitní 2D hry. GameMaker engine je zaměřen především na začátečníky s minimální až nulovou znalostí programování. Firma YoYo Games, jež engine vyvinula, zlepšila táhni-a-pust' systém (z anglického Drag-and-Drop System) na dostatečnou a všestrannou úroveň. Je možné si prohlédnout kód, který vznikne po táhni-a-pust' akcích, díky čemuž se programování lze teoreticky naučit. Je možné psát i vlastní scripty. GameMaker používá svůj vlastní programovací jazyk GameMaker Language založený na kombinaci C# a C++.

V porovnání s ostatními enginy je GameMaker jednoduchý a nevyžaduje žádné vzdělání v IT. Velikost komunity je natolik velká, aby vznikl dostatečný počet materiálů, návodů a příruček. Vzhledem k jeho jednoduchosti dává hráčům velkou stabilitu. Hlavní nevýhodou je 3D vývoj, i když GameMaker engine podporuje přístup k 3D assetům, je silně zaměřen jenom na 2D hry, jak je zmíněno výše. Pro začátečníky je dobrou volbou, avšak přechod na jiné technologie je poté obtížný. Stát se profesionálním vývojářem v GameMaker enginu je reálné, ale vzhledem ke specifickým přístupům a vlastnímu programovacímu jazyku je většina těchto znalostí nepoužitelná u jiných enginů.

Oproti konkurenci je jeho používání dražší. Zakoupením nejzákladnější roční licence Creator za 39\$ je možné vyvíjet hry na Windows a Mac. Další kategorií jsou licence typu Developer, díky kterým je vývojáři umožněno za jednorázový poplatek vyvíjet hry pro specifické platformy. Například pro vývoj mobilních her za 199\$. Poslední možností jsou licence typu Console. Fungují obdobně, akorát poplatek je roční a vztahuje se na jednotlivé konzole. Třeba licence pro vývoj hry na Playstation 4, Xbox One nebo Nintendo Switch stojí každá samostatně 799\$. Noví uživatelé mají k dispozici jednoměsíční přístup zdarma, který sice neumožňuje vygenerovat spustitelný soubor, ale pro seznámení se s vývojem je dostačující. [44]

## 5 Dimenze prostoru

Nejjednodušší rozdělení videoher je podle jejich dimenze. Drtivá většina her spadá právě do 2 dimenzionálních her (2D) nebo do 3 dimenzionálních her (3D). Vysvětlením základních vlastností těchto kategorií se zabývá článek [25], který je rozvinut níže.

### 5.1 2D hra

2D hra podporuje pouze dvě osy pohybu. Typicky se jedná o plochý svět, ve kterém je možné se pohybovat doleva, doprava, nahoru a dolů. Protože 2D hry obsahují nižší počet možností pohybu než jejich 3D protějšek, častokrát bývají utvořeny pomocí jednoduššího principu a jsou lineární. Lineární znamená, že primárním cílem je dostat se ze startu úrovně, až na její konec. V mnoha 2D hrách se objekty jmenují „sprites“, což je obrázek se souřadnicemi x a y, které udávají jeho přesnou polohu v herním světě.

Kamera je velice zjednodušená. Většinou je nasměrovaná ze strany nebo shora směrem do hry a tím je způsobená absence perspektivy. Některé hry používají tzv. „parallax scrolling“, který posouvá pozadí jinou rychlostí než popředí, a vytváří tak iluzi hloubky.

#### Příklady

Mezi typické hry pro 2D prostředí podle [6] patří například Terraria, Stardew Valley nebo Don't Starve. V současnosti se nejedná o slepou či zastaralou větev herního vývoje. Ve hře Terraria je možné objevovat svět, vyrábět předměty, těžit suroviny nebo stavět, a to vše ve 2D světě s možností multiplayeru. S pomocí vytěžených materiálů je možné sestavit předměty a ostatní objekty, které je možné použít během dalšího hraní. Ve Stardew Valley je hráč vlastníkem farmy. Plní úlohy, aby jeho farma byla větší, úrodnější a úspěšnější. Kromě toho je možné seznamovat se s ostatními postavami, plnit jejich žádosti či objevovat údolí. V Don't Starve se hlavní hrdina snaží nalézt zdroje za pomoci procházení se po světě, a to vše proto, aby si mohl postavit obydlí, zaopatřit jídlo a přežít noc. Svět je generován náhodně, čímž se vytváří nepředvídatelné prostředí. Původní Don't Starve je hra pro jednoho hráče.

### 5.2 3D hra

3D hry dovolují plný rozsah pohybu ve třech osách. Hráč je tedy schopný chodit volně v herním světě s možností otáčet se o 360 stupňů. Animace charakteru jsou výrazně



komplexnější. 3D model reaguje na ostatní prvky okolního světa. Místo před-renderování objektů jako v mnoha 2D hrách, se renderují textury na konkrétním povrchu, aby měly přirozenější vzhled. Hratelnost 3D her je na vyšší úrovni, navíc nejsou z velké části lineární, ale je možné prozkoumávat jiné úkoly, než o které se stará hlavní dějová linka. [25]

### **Příklady**

Populární hry ve spojitosti s 3D prostředím jsou The Witcher 3: Wild hunt, Rainbow Six Siege, nebo Minecraft [14]. The Witcher 3: Wild hunt byl vydán v květnu roku 2015, ale stále se řadí mezi prvotřídní tituly. Děj je zasazený do fantasy prostředí podle knižní předlohy Zaklánač od polského spisovatele Andrzeje Sapkowského. S možnou hratelností i několik desítek hodin je hráč v tomto RPG slavným lovcem příšer a plní úkoly s tím spojené. V multiplayerové „střílečce“ Rainbow Six Siege je důležitá kooperace jednotlivých hráčů. Každý má svoji specializaci na určitý způsob boje. Cílem je porazit tým protivníků, k čemu může pomoci například zničení bezpečnostních kamer a dodržování správných pozic. Minecraft byl vyvinut v roce 2009, jeho neustálým upravováním a vylepšováním stále baví mnoho milionů hráčů po celém světě. Základní myšlenkou je svět vystavěný pouze z kostek. Ty lze bourat, těžit, stavět nebo z nich vytvořit jiné užitečné předměty. Hře Terraria, je často přezdíváno 2D Minecraft, protože funguje na podobném principu.

#### **5.2.1 Porovnání 2D a 3D her**

Rozdílnosti mezi 2D a 3D hrami jsou také popsány v [27] a jak je vidět v Tabulka 1.

**Tabulka 1 Srovnání 2D a 3D her. Zdroj: [autor]**

	<b>2D</b>	<b>3D</b>
<b>Postavy</b>	Mají tendenci vypadat pohádkově.	I když občas vypadají také pohádkově, díky hloubce působí reálně.
<b>Interakce</b>	Postavu je možné vidět pouze z jedné perspektivy. Nejčastěji ze strany či shora.	Hráč se může vidět z různých úhlů pohledů.

<b>Pozadí</b>	Používá se obrázek.	Místo běžného obrázku se používá skybox. Jedná se o rovinu, na kterou je podle pravidel namapován obraz.
<b>Design úrovně</b>	Záleží na typu hry. Je nutné vědět, kolik hráčů bude ve hře a počet předpokládaných úrovní. Ve 3D hře je třeba klást důraz na textury, světlo a ostatní věci, což ve 2D hře není až tak zásadní.	
<b>Systémové požadavky</b>	Nebývají velké, ani omezující.	Hodně rozdílné, od nenáročných her, na které stačí běžná i postarší grafická karta, až po velice náročné, které jsou bez nových a výkonných karet nehratelné.
<b>Game Engine</b>	Existují enginy určené pro 2D, 3D, či oboje.	
<b>Ovládání</b>	Průměrně se používá 5-7 tlačítek pro ovládání hry. Extrémní případy používají jenom 2, nebo naopak i 10 a více.	Průměrný počet používaných kláves je 15-20, plus ovládání myši.
<b>Uvěřitelnost</b>	Ve věrohodnosti prostředí vzhledem k realitě se 2D hry kvůli absenci hloubky nemohou rovnat s 3D hrami.	
<b>Trh</b>	Nejsou tak populární jako 3D hry, ale občas vývojáři zaujmou s originálním nápadem, či příběhem.	Obrovský trh, kde je o 3D hry veliký zájem.

### 5.3 2.5D hra

Pokud hra není čistě 2D, ani 3D, s největší pravděpodobností se jedná o 2.5D. Tedy hru, která kombinuje obě dimenze, konkrétně moderní 3D grafiku s tradiční 2D hratelností. Například bojová hra, která používá 3D modely postav a vše je sledováno z fixní postranní perspektivy. Pohyb probíhá většinou po 2D rovině. Hráč má málokdy přímou kontrolu nad kamerou. Výraz 2.5D byl používán u prvních „stříleček“ z první osoby jako je Doom nebo Duke Nukem 3D, které používaly 2D sprites. Pravá perspektiva s 3D modely nebyla totiž tehdejšími technologiemi dosažitelná. [12]

#### Příklady

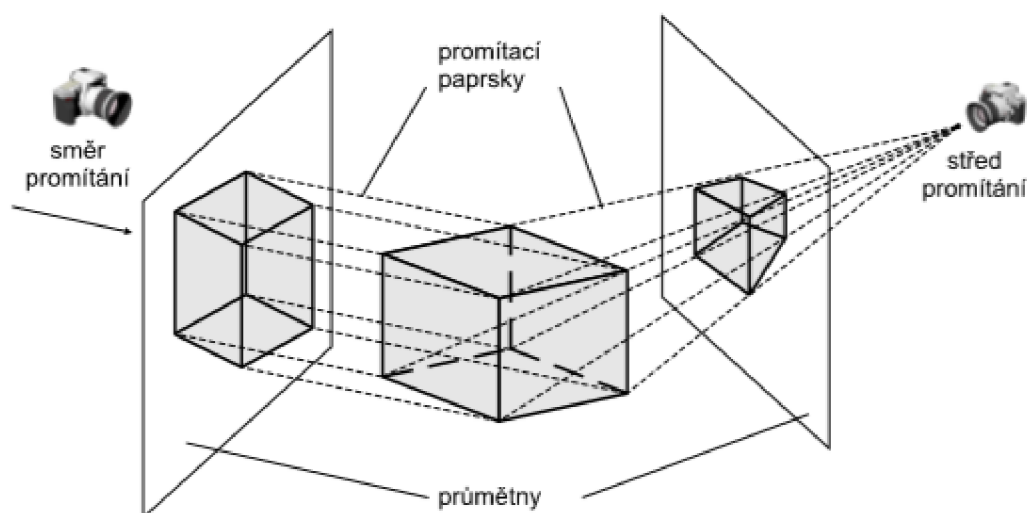
V minulosti sice 2.5D hry vznikaly hlavně kvůli technickým omezením, ale jsou stále vyvíjeny i dnes po překonání technických bariér. Jako příklady z mnoha úspěšných titulů

Lze uvést třeba *Mortal Kombat* či *Trine*. Velmi populární hra *Mortal Kombat* využívá jednoduchý princip. Proti sobě stojí dva hráči s možností pohybu pouze do stran. Navzájem se snaží adekvátními kombinacemi kláves, či tlačítek na ovladači porazit protivníka v boji zblízka. Série *Trine* vypadá jako typická 2D hra s kamerou ze strany, ale díky 3D modelům postav, nepřátel a zbytku světa se řadí mezi 2.5D hry. Hráč může hrát za jednu ze tří postav. Každá postava má specifické vlastnosti a jinou hratelnost. Hru je možné hrát v 1-4 hráčích.

#### 5.4 Projekce prostoru

Protože vytváření hry v game engineu probíhá nejčastěji v 3D prostoru, musí se zvolit způsob transformace 3D bodů do 2D roviny. Tou je nejčastěji monitor, či jiné zobrazovací zařízení. Podle [45] je takový proces, kdy těleso ztrácí prostorovou informaci, nazýván promítání. Rovnoběžné nebo středové promítání tvoří dva základní způsoby, jak transformace dosáhnout.

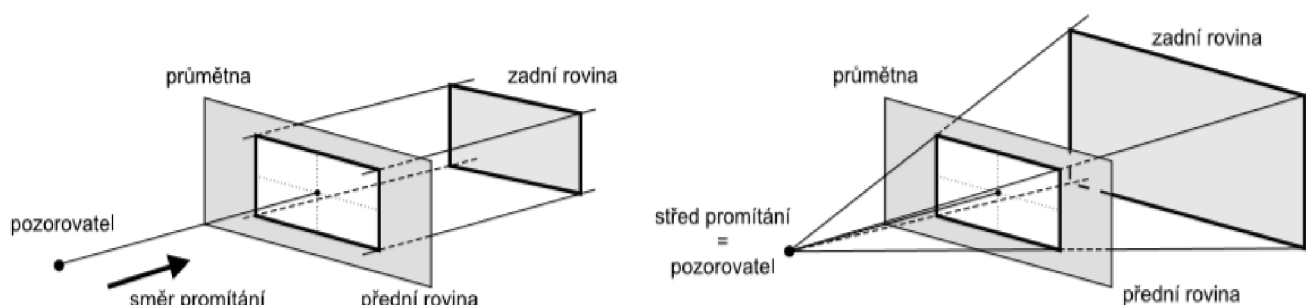
Na Obrázek 2 je možné vidět, jak směr promítacích paprsků závisí na zvolené promítací metodě. Plocha, na kterou dopadají paprsky se jmenuje průmětna (z anglického *viewing plane*). Díky obrazu, který je vytvořen na základě míst dopadů paprsků, vznikne v průmětně 2D reprezentace prostoru nazývaná průmět. Rovnoběžné promítání používá pro všechny paprsky stejný směr na rozdíl od středového, kde paprsky vychází právě z jednoho bodu. Středové promítání vyobrazuje způsob vnímání světa lidským okem, protože se vzrůstající vzdáleností pozorovatele probíhá zmenšování předmětů.



Obrázek 2 Rovnoběžné promítání (vlevo) a středové promítání (vpravo).

Zdroj: [45]

Tělesa spadající do pohledového objemu (z anglického viewing frustum) budou vykreslena, všechna ostatní tělesa musí být ořezána. Není nutné zpracovávat všechna tělesa na scéně, ale pouze ta, která budou na obrazovce vidět. Tím se výrazně urychlí proces promítání. Pohledový objem u středového promítání má tvar komolého jehlanu, kdežto u rovnoběžného je to kvádr, jak je vidět na Obrázek 3. Velmi důležitou roli hraje přední a zadní omezovací rovina. Obě zmíněné roviny zamezují vykreslení příliš blízkých objektů, které by překážely ve výhledu. Stejně tak ořezávají vzdálené, pro pozorovatele nezajímavé objekty. [45]



**Obrázek 3 Pohledový objem rovnoběžného (vlevo) a středového (vpravo).**

**Zdroj: [45]**

Rovnoběžné promítání mohou používat architekti a inženýři pro práci s objekty. Zachovává kompletní nezkrácenou reprezentaci objektů. Žádný objekt se s rostoucí vzdáleností od obrazovky nezmenšuje. Proto i při otáčení tělesa je stále zachována původní velikost i tvar. Tento druh projekce používají 2D hry, kde perspektiva nemá smysl. Naopak vývoj 3D hry se bez středového neboli perspektivního promítání neobejde.

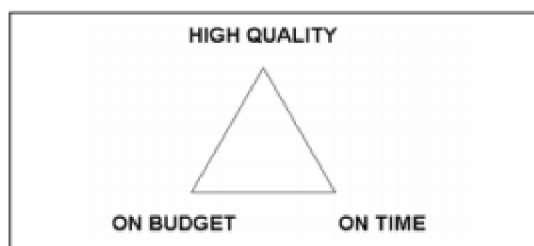
## 6 Návrh a herní design

Na samotné programování přichází řada ve chvíli, kdy je jasně vytyčeno, co a jak by mělo být implementováno. Proto je nutné nejdříve vypracovat podrobný návrh, jak má hra vypadat, hlavní myšlenky, základní herní mechaniky a další nezbytné informace. Soubor, který vytváří základní kostru pro vývojáře, se nazývá Game Design Dokument (z anglického Game Design Document, GDD). Primární cíl je předat programátorům jasné instrukce s veškerými informacemi o připravované hře, a minimalizovat tak nejasnosti v tvorbě projektu.

GDD existuje v mnoha formách. Buď v tištěné podobě nebo jako online brožura. Před začátkem vytváření dokumentu je doporučeno vytyčit si několik stěžejních bodů, jako jsou: za jakým cílem je hra vytvářena, pro jakou skupinu je určena. Podrobnému popisu návrhu se věnuje kniha [2], která je zdrojem následující podkapitoly.

### 6.1 Obchodní parametry

Obrázek 5 v první části (Inception) demonstruje počáteční myšlenku projektu. Avšak reálný start návrhu projektu nastává až v druhé fázi, viz. Obrázek 5 (Business Parameters), což dále vede k zamyšlení se nad obchodními parametry, tedy snaha zaměřit se na upřednostňované hodnoty v průběhu vývoje. Pro ilustraci poslouží Obrázek 4, vrcholy trojúhelníku představují tři základní cíle: zda se překročí rozpočet (ON BUDGET), jestli bude hotovo v čas (ON TIME) a bude-li dosaženo vysoké kvality (HIGH QUALITY). Hrana mezi vrcholy znázorňuje vztah mezi nimi. Je třeba vybrat jednu hranu, díky čemuž se dva cíle stanou prioritní a třetí bude upozaděno. Snaha předejít ztrátě jakéhokoliv vrcholu, a udržet si všechny tři vrcholy, v praxi není realizovatelná. Zvyšuje se šance ztráty ne pouze jednoho vrcholu, ale i dvou, nebo dokonce všech tří. V praxi mohou nastat tři různé scénáře. Prvním z nich je situace, kdy se rozhodne pro



**Obrázek 4 Trojúhelník projektu.  
Zdroj: [2]**

dokončení projektu v čas a zároveň se nepřesáhne stanovený rozpočet. To má za následek nutnost přijetí faktu, že finální produkt nebude požadované kvality. Další možností je držet si stále vysokou kvalitu a nepřesáhnout rozpočet, čímž se opozdí datum vydání. Poslední scénář nastane, pokud je pro studio hlavní dodržet čas a neupustit na kvalitě, tím se zvýší finanční náročnost celého projektu. [2]

Je tedy prioritou důkladně popřemýšlet o ideální volbě strategie. Stanovení „trojúhelníku“ projektu přispívá ke správnému deklarování obchodních parametrů, avšak existují i další metody. Zejména přímočaré zodpovězení následujících otázek:

- 1) Čeho chcete s touto hrou dosáhnout?
- 2) Kdy musíte dokončit projekt?
- 3) Kolik máte peněz na vývoj?
- 4) Koho máte na práci?



**Obrázek 5 Fáze návrhu a vývoje. Zdroj: [2]**

### **6.1.1 Nízkorozpočtový projekt**

Pokud hru vytváří pouze jedna osoba, ve svém volném čase, její rozpočet může být limitující. Omezení se dají částečně eliminovat sdílením projektu mezi další osoby, nebo sdílením na internetu s předpokladem, že se do projektu zapojí další. Koordinace skupiny dobrovolníků zpravidla bývá nelehká a tvůrci takových skupin většinou projekt opouští. Získání zaměstnání v herním průmyslu se stává velice atraktivním právě v kruzích programátorů, kteří se již podílely na takovémto druhu projektu. Obvyklý výstup nízkorozpočtového projektu je modifikace již existující hry, popřípadě vznik úplně nové hry. Vytvoření dema, které nese základní myšlenku, ale nemá kompletní umělou inteligenci, zvuky ani textury. Tento výstup nemá pozitivní dopad na prezentaci projektu herním vydavatelům. Vydavatelé nebudou žádat produkt, který má deset ne zcela kompletních vlastností, ale upřednostní ten, který má tři, čtyři perfektně vyladěné. [2]

### 6.1.2 Fixní rozpočet a pevné datum vydání

Je zapotřebí rozdělit každou požadovanou funkci dané hry právě do jednoho ze tří pilířů, které třídí funkcionality podle priorit – primární, sekundární, terciální. Je to způsob, jakým se identifikují nejzásadnější rysy hry. Nejdříve je vytvořen plán, podle kterého se vývojáři zaměří na vývoj pouze primárních funkcí, tj. primární pilíř. Sekundární, případně terciální pilíř přijde na řadu až po zdárném implementování primárních funkcí. Při nedostatku času může nastat nežádoucí situace a to taková, že některé sekundární položky budou přesunuty do terciálního pilíře a z vybraných primárních vzejdou sekundární.

Aby projekty, s pevně daným rozpočtem a neměnným datem vydání, fungovali podle plánu, je třeba začít od konce a stanovit datum, kdy projekt vstoupí do fáze alfa a beta. Fáze alfa a beta představují časový úsek, během vývoje hry. Pro každou fázi jsou specifické jednotlivé úkony vývojářů. Ve fázi alfa nejsou do projektu přidávány žádné další vlastnosti do zdrojového kódu. Vlastnosti hry jsou například: způsob pohybu, fyzikální zákony v herním světě a další žánrová specifika. To znamená, že všechny základní herní vlastnosti byly již do kódu přidány. V mezičase alfy a bety, je tým zaměřen na opravy známých chyb, zajištění stability hry. Standartně je hra do fáze beta uvedena v okamžiku, kdy jsou odstraněny všechny dosud známé chyby. V samotné fázi beta se provádí testování a hledají se další dříve neviditelné nedokonalosti, poté dochází k finálnímu vybalancování hry. Posléze přichází pouze takzvaná *final gold master* etapa, což je označení pro konečné zpracování hry. Díky stanovení pevných dat fází alfa a beta, je k dispozici hrubý odhad množství zbývajících času pro návrh hry a produkci. Obecně se udává, že 15-35 % celkového času, by měla tvořit pre-produkce. [2]

### 6.1.3 Vysokorozpočtové prvotřídní projekty

Velké firmy jako Blizzard, BioWare, Ubisoft a další, jsou pod velkou tíhou očekávání, protože je předpokládáno, že z každého vydaného titulu předních vývojářů vznikne hit, jehož kvalita stanoví nové, vyšší měřítko pro herní průmysl. Koncentrace, na maximální možnou úroveň interní komunikace ve vývojovém týmu, je nevyhnutelná. Spolupracující týmy programátorů, pracující na stejném projektu, čítají několik desítek, ba i stovek vývojářů. Obtížnost komplexní komunikace vzrůstá společně s počtem pracovníků. [2]

#### **6.1.4 Obchodní parametry v praxi**

Při tvorbě bakalářského projektu je třeba vyřešit problematiku, kterou znázorňuje Obrázek 4. Upřednostnění vrcholu ON TIME plyne ze skutečnosti, že projekt musí být zhotoven na čas. Vzhledem k silně omezenému rozpočtu projektu byl zvolen vrchol ON BUDGET. Účel projektu je především demonstrativní, tedy není třeba dosažení stejné kvality jako u her od profesionálních vývojářů. Jelikož se jedná o nízkorozpočtový projekt, je nutné rozhodnout, zda výsledkem bude nová jednoduchá hra, nebo modifikace existující hry. Odpovědí na tuto otázku je ujasnění cíle, kterým je především předvést vývoj hry od úplného začátku, z toho důvodu bude výstup v podobě nové a fungující hry.

### **6.2 Koncept hry**

Nyní, podle [2] přichází na řadu zamyšlení nad konceptem hry (z anglického Game Concept), jak je výše vyobrazeno i na Obrázek 5. V tomto okamžiku vytyčené obchodní parametry slouží jako důležitý nástroj k tomu, zda prostředky odpovídají předpokládané kvalitě a velikosti hry. Je logické, že vývojářské studio s fixním rozpočtem sto tisíc korun, si nemůže dovolit začít roční vývoj hry, protože náklady pro vývoj takovéto hry by byly mnohonásobně vyšší. V praxi není jednoduché ideálně naplánovat vývoj hry v závislosti na dostupných zdrojích. Úmysl je najít takový koncept, u kterého se po dokončení s vysokou pravděpodobností dostaví příznivé příjmy. Mnoho firem specifikuje obchodní parametry a koncept hry najednou, případně upravují obě fáze do doby, než si navzájem vyhovují. Ať už se jedná o úpravy ve smyslu uvolnění více peněz, pracovníku, či času, nebo naopak snížení požadavků na hru. V případě nenalezení ideální rovnováhy je žádoucí, aby projekt skončil, není totiž nutné najít řešení za každou cenu.

Chyby v rozvržení se projeví na projektu až v pozdější fázi, kdy tým nestíhá implementovat hru podle prvotního plánu. V této situaci vedení projektu bere tuto skutečnost jako nežádoucí, tým je vnímán jako nezralý a následně zruší celý projekt. Poněkud nešťastné je řešení, kdy vedení udržuje projekt za každou cenu, rozšíří zdroje a přidá týmu čas. Přidělení více peněz a odložení data vydání není v herním průmyslu vzácné, z tohoto důvodu si spousta vývojářů mylně myslí, že tím nebudou nepoškozeni, ani že výrazně neoslabí vydavatele. Pokud je herní vydavatel donucen vložit další peníze do projektu a posunout datum vydání, musí tak čelit negativním dopadům. Zprvé, investované finance nebudou k dispozici pro nové, potenciálně úspěšnější tituly. Zadruhé,



vydavatel bude mít zpoždění s kompenzací investic a znalostí zisku. Reklama vyskytující se před vydáním, která bývá zpravidla zásadní, se stane tak nevhodnou. Pokud je hra odložena, ztratí se i veškeré nadšení zákazníků a je potřeba s reklamou začít znovu až to bude relevantní, což stojí vydavatele další nemalou částku. Ostatní vydavatele her může zastrašit právě nastalý konflikt ohledně data vydání a proběhne snížení poptávky po vývojářském studiu. Posledním důvodem, proč se za každou cenu vyhnout špatnému rozvržení herního konceptu, je vůči obchodním parametrům, vznikne tak totiž deziluze, která může nastat u jednotlivých programátorů na jejímž základě se rozhodnou opustit firmu a přestoupit ke konkurenci. [2]

### **6.2.1 Unifikovaný Modelovací Jazyk**

Podle zdroje [5] je při vytváření hry primárním cílem udržovat systematický vývoj, kterého lze dosáhnout používáním notace unifikovaného modelovacího jazyku (z anglického Unified Modeling Language, UML). Jedná se o standardizovaný text a vizuální jazyk pro podporu jednotného návrhu diagramů. V minulosti bylo mnoho pokusů vytvořit jakýsi standard, jak nahlížet na návrh projektu. Až firmě Rational Software se v 90. letech minulého století povedlo vytvořit jednotnou metodiku s názvem UML. Velké firmy začaly projevovat zájem o to, aby tento standard existoval a byl udržován. Ambice udržet standard stále aktuální, vedly ke vzniku konsorcia Object Management Group (zkráceně OMG), kde mají své zástupce například IBM, Rational Software, Microsoft, Oracle a další.

UML slouží také k efektivní komunikaci v týmu. V okamžiku, kdy složitost projektů vzrůstá, je dobré spojení mezi programátory stále více žádoucí. UML diagramy pomáhají lidem, kteří se snaží lépe pochopit stávající požadavky na systém. Důležitou vlastností, kterou jsou diagramy vybaveni, je abstrakce. Ta umožňuje nahlížet na systém z různých úhlů pohledů a zachytit pouze důležité informace pro konkrétní téma. UML plán, vytvářen pomocí softwaru, je detailní zpracování diagramů sloužících programátorům jako předloha při implementaci. Diagramy jsou po dokončení použité jako dokumentace projektu, tedy i nově přichozí programátor porozumí dané problematice. Kromě vytvoření teoretického plánu, má většina softwaru podporujícího UML notaci i funkcionalitu generování kódu. To znamená, že navrhnoutou strukturu je možné exportovat ve formě programátorského kódu, který je použit jako základní šablona pro implementaci.

Aktuálně je UML složeno ze 14 diagramů, které jsou rozděleny do 2 elementárních skupin. Diagramy struktury tvoří první skupinu a popisují co projekt obsahuje. Druhou skupinou jsou diagramy chování, jež mají za úkol vysvětlit, jak systém funguje. Každý diagram z obou skupin má svůj účel, ale není nutné použít všechny. [5]

Nástrojů pro zakreslení diagramů je mnoho. U některých není nutná instalace do počítače, tvorba může probíhat pouze v prohlížeči. Takovým online nástrojem může být například zdarma dostupný diagrams.net. Pokud je upřednostněna desktopová aplikace, pak jsou možnou volbou Modelio, StarUML, nebo poněkud složitější Enterprise Architect.

### 6.2.1.1 Diagram případů užití

Dle zdroje [4], diagram případů užití (z anglického Use Case Diagram) je modelován z pohledu uživatele. Tento diagram často vzniká jako jeden z prvních, protože informuje o tom, co má systém nabídnout uživateli, ale neříká nic o tom, jak toho má docílit. Složen je z případů užití, aktérů a vzájemných vztahů. Diagram případů užití videohry vyjmenovává, co se zobrazí v prostředí hry z pohledu hráče a čeho v ní bude schopen.

Případ užití je soubor akcí, které je třeba podniknout pro dosažení cíle. Typickým příkladem případu užití je sebrat předmět, vidět terén, hýbat s hlavním hrdinou a další. Jeden případ užití má v diagramu nejčastěji tvar elipsy, v které je název funkcionality. Aktér je ten, kdo interaguje s jednotlivými případy užití, což je znázorněno pomocí příslušného vztahu mezi nimi. Nejčastější je vazba <<include>>. Pokud je spuštěn případ užití, jsou vždy spuštěny i všechny případy užití napojené pomocí vazby include. Alternativou se složitějším použitím je <<extend>>. [4]

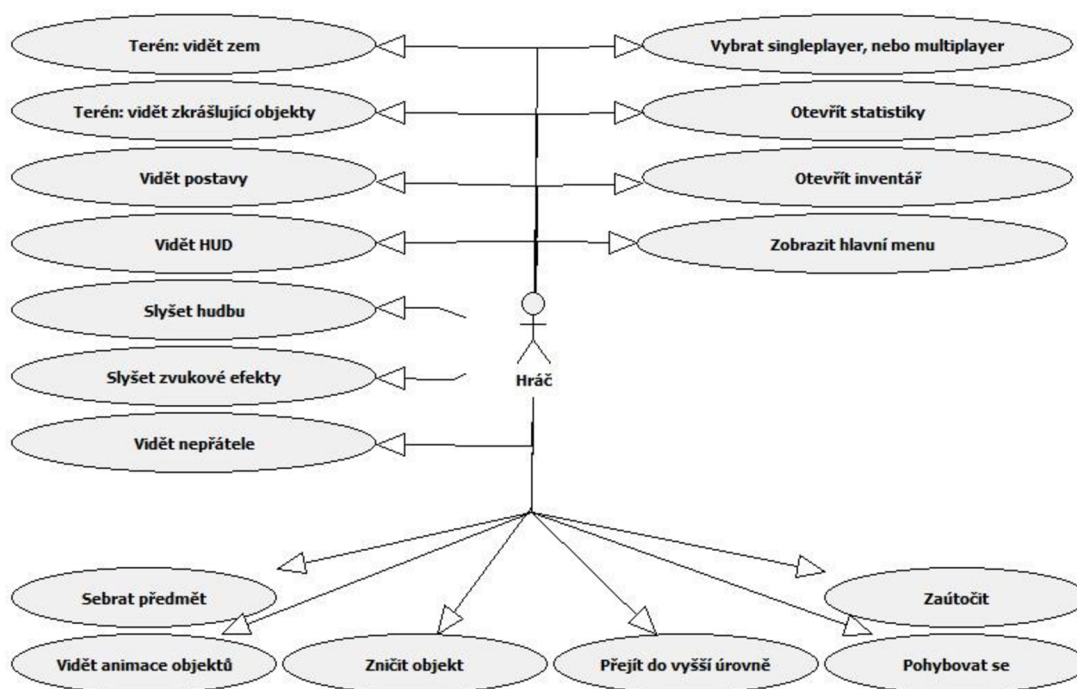
### 6.2.2 Vznik konceptu hry

Pro ideální ztvárnění konceptu hry je vhodné vytvořit use case diagram. Na základě [2] je doporučeno nejdříve si napsat hlavní kategorie funkcionalit a do těch přidávat příslušné případy užití. Tento postup je zvolen i u bakalářského projektu, jak je zobrazeno v Tabulka 2 **Chyba! Chybný odkaz na záložku.** Nejprve je nutné zvolit požadované kategorie, které musí společně pokrýt všechny případy užití, jenž mají být použity. Jediným uživatelem systému, tudíž i aktérem, je hráč. V této hře neexistuje žádný administrátor, nebo jakákoliv jiná role, která má jiná privilegia než samotný hráč. V diagramu jsou modelovány pouze funkcionality, které tvoří dojem ze hry.

**Tabulka 2 Případy užití bakalářského projektu. Zdroj: [autor]**

<b>Prostor a zvuk</b>	<b>Interakce s objekty</b>	<b>Správa</b>
Vidět zem tvořenou z 2D dlaždic (hlína, kameny, tráva)	Sbírat předměty (léky, zbraně atd.)	Zobrazit hlavní menu
Vidět objekty (stromy, cedule)	Pohybovat svým hrdinou (nahoru, dolů, doleva a doprava)	Otevřít statistiky
Vidět postavy	Zaútočit	Otevřít inventář
Zobrazit HUD (objekty, které překrývají herní prostor)	Vidět animace objektů (otevření dveří, zobrazit text, animace hlavního hrdiny a nepřátel, animace neutrálních postav)	
Vysílat hudbu a zvukové efekty	Zničit vybraný objekt	
Vidět nepřátele	Načíst novou úroveň po dosažení správné pozice	

Z vyobrazené Tabulka 2 je zřejmé rozdělení do tří kategorií. První sloupec má na starost pokrýt vše týkající se dojmu z herního prostředí, proto nese název *prostor a zvuk*. Kromě zvuků, jde převážně o vykreslení statického prostoru, jenž bude hlavním faktorem pro navození správné atmosféry během hraní. Střední sloupec je poněkud obsáhlejší než zbylé dva. Je nutné specifikovat veškeré ovládání, spouštění animací, či načítání nové úrovně. S využitím fyzikálního systému poskytovaného přímo enginem, je třeba nastavit pohyby postav a jejich kolize s ostatními objekty. Třetí sloupec zobrazuje další statické záležitosti jako základní informace o stavu hry, pohyb v hlavním menu, či zvolení herního režimu. Vzhledem k rozsahu tabulky je možné vytvořit jednoduchý, ale dostačující diagram případů užití (Obrázek 6), na kterém jsou základní a očekávané funkcionality.



Obrázek 6 Diagram případů užití bakalářského projektu. Zdroj: [autor]

### 6.3 Herní design

Fáze herní design (z anglického Game Design) z Obrázek 5 představuje neoddelitelnou součást celého návrhu, jejíž hlavní částí je Game Design Dokument, zkráceně GDD. Ačkoliv neexistuje žádná standardizovaná šablona, která by určovala, jak by měl vypadat GDD. Avšak mnoho odborníků, díky svým zkušenostem, doporučuje, co by nemělo v GDD chybět. Výsledný dokument může mít podobu fyzickou, ale i čistě elektronickou. Celkovým významem a popisem doporučených částí Game Design Dokumentu se zabývá kniha [21], která je níže parafrázována.

Účel zhotovení GDD je kompletně popsat vyvíjenou hru včetně detailů hratelnosti. Programátoři musí být informováni o tom, jakých akcí bude hráč schopen nebo v jakém prostředí se bude hráč pohybovat. Protože GDD není jediným popisným dokumentem, je vhodné se vyhnout okruhům informací, které by se v něm neměly vyskytovat. Například Technický Design Dokument doplňuje návrh o systémové požadavky hry, strukturu kódu, algoritmy umělé inteligence a další technické informace. Zjednodušeně, Game Design

Dokument určí funkcionality hry, ale nevypovídá nic o jejich technické, či implementační stránce.

Není žádný standard, podle kterého se tvoří dokumenty. Vlastní formát používají hlavně velké firmy pro své interní použití. Ostatní firmy použijí již zhotovený GDD jako předlohu, nebo si vytvoří nový formát. Avšak je určitý druh informací, jež by se měly zaručeně objevit v každém design dokumentu, aby byl užitečný. Zdroj [21], rozděluje popis hry do následujících sekcí:

- obsah,
- úvod,
- herní mechaniky,
- umělá inteligence,
- herní prvky,
- příběh,
- herní progres,
- menu.

V závislosti na typu hry nemusí být v GDD nutně obsaženy všechny sekce.

### **6.3.1 Obsah**

Obsah, tedy seznam všech kapitol i podkapitol, by neměl nikdy chybět v GDD. Když člen vývojářského týmu, který hledá informace, se nejdříve podívá do obsahu. Čím podrobnější a výstižnější bude seznam, tím rychleji vývojář vyhledá informace. Klíčovou vlastností pro obsah je tedy co nejvyšší detailnost, v opačném případě se může jednat o ztrátu času.

### **6.3.2 Úvod**

Je doporučeno zpracovat krátké, maximálně jednostránkové shrnutí o herním designu, na začátku dokumentu. Jeden odstavec by měl být věnován příběhu se zaměřením na hlavní dějovou linii. Ostatní odstavce se zaměřují na různé aspekty hrátelnosti, které se stanou lákadlem pro hráče. Celý souhrn by se měl zaměřit na vlastnosti, které tvoří hru unikátní v porovnání s ostatními. Není nutné uvádět, že je možné ovládat hlavního hrdinu, protože

se jedná o zřejmý fakt. Po dočtení by tak čtenář měl mít představu o tom, čím bude hra na herním trhu unikátní.

### 6.3.3 Herní mechaniky

Herní mechaniky jsou nejdůležitější částí GDD. Právě programátoři většinou první vyhledávají onu mechaniku. Místo „herní mechaniky“ by názvem mohla být rovněž „hratelnost“, protože je popisováno, co hráč může dělat a jak probíhá samotné hraní. Ze všech pasáží, je napsat právě tuto nejsložitější. Mnoho špatných Game Design Dokumentů, vznikne přeskočením této části a zaměřením se více na části, které se píšou o poznání lehčeji, zejména příběh, vzhled, nebo systémová menu. [21]

Až na výjimky, by se mělo vyhnout detailnímu popisu jednotlivých objektů nebo postav, neboť se to zcela jistě neřadí do této části. Záměrem je popsat efekty zbraní, které hrdina může sebrat, a jejich ovládání, ale kompletní podrobný seznam zbraní z herního světa má využití až v pozdější fázi dokumentu. Seřazení jednotlivých herních mechanik je stejné, jako když hráč poprvé zapne hru. Ze všeho nejdřív je vysvětleno základní ovládání, tedy jakým tlačítkem je hrdina uveden v pohyb. Pokud bude stisknuta klávesa pro pohyb doprava, hrdina se vydá tím směrem, bude-li stisknuta a ihned uvolněna, pak se původní pozice změní pouze o malou hodnotu, a tak dále. Je dobré používat výrazy, které jsou na platformě nezávislé, nemusí být totiž známy konkrétní klávesy, ale hlavně přesný počet kláves.

Po představení základních příkazů bude dovysvětlen zbytek pohybového modelu, zdali bude co nejvíce blízky realitě, či naopak. Zda při startu bude postava pomalu zrychlovat do plné rychlosti, nebo s plnou rychlostí již pohyb začne. Nebo zda závisí rychlost chůze na povrchu. Nebo když narazí do jiného objektu, zda se zastaví, nebo odrazí. Přesně tento typ detailů je potřeba do hloubky upřesnit. Na výběr detailů k popsání má samozřejmě největší vliv typ hry. Po základech přicházejí na řadu komplexnější akce, kupříkladu jestli je možné sbírat předměty a jak funguje inventář. Každá část se snaží co nejvíce stavět na té předchozí, čímž čtenář získá povědomí o celkové složitosti hry. Prioritou je vysvětlit, jak fungují všechny interaktivní objekty, jimiž jsou padající zdi, speciální lektvary, otevírání/zavírání bran, magické efekty a jakékoliv další komponenty vyskytující se v herním světě. Stále se však nejedná o seznam všech elementů a jejich podrobné konfigurace, nýbrž o jakousi paletu objektů, díky které se následně skládá herní svět. [21]

Grafické uživatelské rozhraní (z anglického Graphic User Interface, GUI) je taktéž kriticky důležité, proto bude také následovně podrobně popsáno. Jedná se o vyobrazení rozhraní během hraní, které informuje hráče o aktuálním stavu hry. Konkrétně jde o počet životů, mapu světa, různé statistiky, vzhled rozhovorů, vzhled inventáře s dalšími informacemi. Systémová menu mají ovšem v GDD vlastní sekci, proto není na místě je popisovat tady jako součást GUI.

Otázky „Co a jak může hráč dělat?“ slouží jako pomocná vodítka ke správnému popsání herních mechanik, přičemž otázky „kde“, „kdy“ a „proč“ by měli pro tuto sekci zůstat nezodpovězeny. Na herní mechaniky lze nahlížet jako na extrémně detailní manuál k vytváření hry. Není nesnadné zapomenout explicitně definovat funkcionalitu, která se autorovi GDD jeví jako zřejmá. [21]

#### **6.3.4 Umělá inteligence**

Kapitola s názvem Herní mechaniky popisuje, jak hráč interaguje s herním světem. Kapitola jménem Umělá inteligence pak uvádí, jaké jsou reakce herního světa na podniknuté akce hráčem. Také je nutno uvést, jak se herní svět chová, pokud hráč nevykonává žádnou akci. Začlenění této podkapitoly do game design dokumentu je rozhodně ovlivněno typem hry. Jednoduchá hra jako Tetris, nebo i složitější multiplayerové hry, rozhodně nebudou mít samostatnou část pro umělou inteligenci v GDD. Úkolem je plně definovat předpokládané chování hry. Jak objekty reagují, jestli hráče zpočátku ignorují, zda po mapě hlídkují, pokud útočí, tak z jaké iniciativy. Zdali uslyší nebo uvidí hrdinu. Zda budou nepřátelské jednotky bezhlavě útočit, nebo budou využívat výhody terénu ke svému prospěchu. Vystává tu mnoho dalších podobných otázek, které by měly být zodpovězeny, aby programátor umělé inteligence měl co nejjasnější představu chování hry. [21]

#### **6.3.5 Herní prvky**

Nyní, když jsou vysvětleny herní mechaniky a možnosti umělé inteligence, přichází na řadu specifikovat možný obsah hry. Zatím bylo doporučeno vyhnout se vyjmenovávání všech jednotlivých objektů a popisování hlavní myšlenky hry, ovšem teď přichází na řadu pravý opak. Herní prvky tvoří jakýsi zásobník objektů pro designéry úrovní a herního světa. Každý element, který se může objevit ve hře, musí být uveden zde, v této

podkapitole. Prvky je dobré rozdělit do menších podkategorií, například Postavy, Předměty a Objekty/Mechanismy.

Do kategorie Postavy spadá výpis všech postav, včetně nepřátel. Část Předměty obsahuje všechny entity, jimiž hráč může zaplnit inventář. Poslední skupina herních prvků Objekty/Mechanismy zahrnuje elementy vyskytující se ve hře, které nejsou poháněny umělou inteligencí, ale hráč s nimi může operovat, tj. dveře, přepínače a ostatní. Uvádění statistik v tuto chvíli by bylo předčasné. Náročným úkonem se stává vybalancování hry ještě předtím, než je možné ji hrát. Proto veškeré konkrétní statistiky objektů, typicky poškození zbraně, se určují později. Cílem je vyobrazit objekt, jak je vzhledem k ostatním silný a pro jakou situaci se hodí nejvíce. [21]

### **6.3.6 Příběh**

Ačkoliv se nejedná o nezbytně nutnou součást GDD, při vyvíjení je dobré mít alespoň částečný přehled o příběhu, který se ve hře odehrává. Pokud vůbec nějaký příběh hra má, jedná se pouze o souhrn hlavní dějové linie, lze předpokládat, že u „střílečky“ bude shrnutí zhruba jednu stranu dlouhé, kdežto u rolové hry může být i delší.

### **6.3.7 Herní progres**

Část herní progres bude v GDD pravděpodobně ta nejdelší. Účelem je vytyčit si události (úrovně), které mění podobu hry. Nejčastěji je hra rozdělena do úrovní, kdy hráč čelí pokaždé jiné výzvě. Pak je potřeba ke každé úrovni doplnit jaký obsahuje úkol. Zdůraznit předměty a objekty, které nemají v dané úrovni chybět, případně, které nesmějí být použity. Programátoři a designéři by měli být informováni, o náplni jednotlivých úrovní. Zda se jedná spíše o průzkumnickou misi, nebo se zaměřuje spíše na akční boj. Ne všechny hry mají úrovně, proto je třeba najít jiné rozdělení hry samotné. Takové, které se mění po celou dobu hry, například když hráč získává určitý počet bodů. Navíc některé hry vůbec nepotřebují informovat o herním progresu, protože všechny důležité informace jsou obsaženy už v herních mechanikách, zejména hry, kde jsou úrovně generovány náhodně. [21]

### **6.3.8 Menu**

V této kapitole GDD je podrobně popsáno, jak má vypadat hlavní menu po zapnutí hry a jakékoliv další zobrazení možností, na které hráč v průběhu hry může narazit. Důležitou



vlastností je, že tyto menu nijak neovlivňují hratelnost, a proto jsou definovány v samostatné sekci. Je zde zařazen nejen vzhled, ale i ovládání menu, zda klasicky myší, nebo jenom klávesnicí.

### **6.3.9 Tvorba Game Design Dokumentu**

Existuje více způsobů, jak pohodlně vytvořit přehledný Game Design Dokument. Základní možností je začít tvořit strukturu ručně s využitím libovolného textového editoru či procesoru. Variantou může být použití online aplikace, nabízející předpřipravené šablony pro dokument. Typickou aplikací tohoto charakteru je například Nuclino, vybavená knihovnou vzorů s možností vizualizace. Plnohodnotná alternativa k Nuclino je aplikace Dundoc, která byla použita při tvorbě GDD bakalářského projektu, umístěného níže v Příloha 1 – Game Design Dokument.

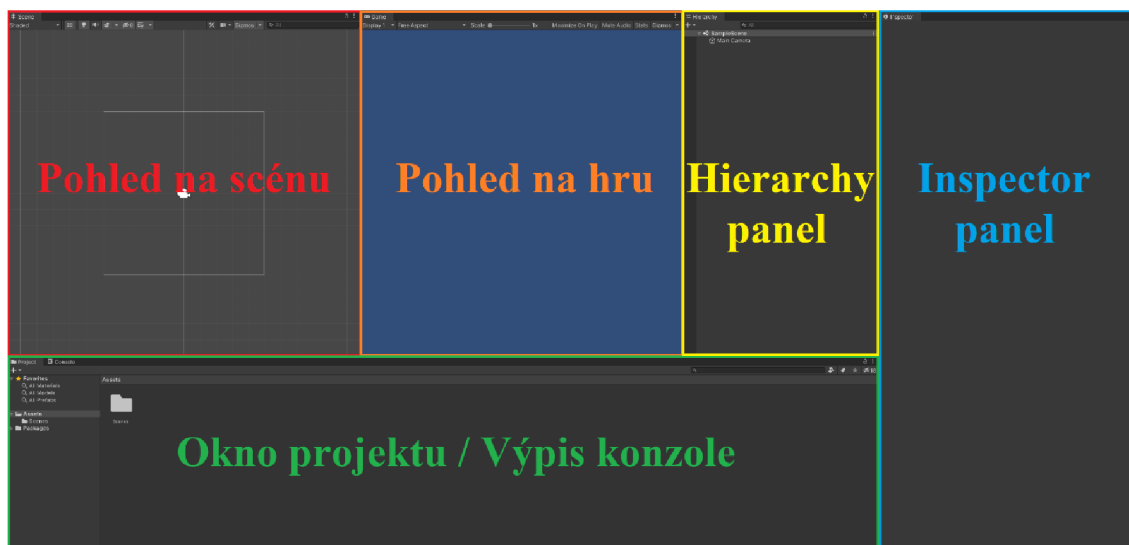
## 7 Implementace počítačové hry

Implementace je vysvětlena s využitím prostředí Unity pro tvorbu 2D her a za pomoci ukázek z bakalářského projektu. Projekt je zaměřen na důležité funkcionality hratelnosti, jimiž jsou zejména pohyb, souboj, správa předmětů, ovládání kamery, zvuky a další. Produktem je počítačová hra složená ze tří úrovní, spadající do žánru RPG. Byla vytvořena jako nízkorozpočtová nezávislá hra.

### 7.1 Orientace v prostředí Unity

Otevření projektu v Unity editoru probíhá nejčastěji přes doprovodnou aplikaci Unity Hub, kde je navíc umožněn rychlý přístup ke komunitním zprávám, výukovým materiálům, rozpracovaným projektům a ke stažitelným podporovaným verzím Unity enginu. Při zakládání nového projektu je na výběr z několika předpřipravených šablon, jejichž počet závisí na právě používané verzi enginu, mezi nejběžnější patří 2D, 3D a URP šablona.

Jak je zobrazeno na Obrázek 7, v Unity editoru figuruje celkem šest základních bloků, napomáhajících k přehlednosti. Okno projektu slouží jako seznam všech souborů a složek uložených v aktuálním projektu. Soubory typu obrázku, zvuku a 3D modelu, jsou obecně označovány jako *assets*. Další část, Výpis konzole, kromě jiného zobrazuje upozornění a varování, včetně užitečných doporučení. Hierarchy panel je seznam všech herních objektů vyskytujících se ve scéně, včetně jejich případných vztahů „rodič a dítě“. Čtvrtým blokem je Pohled na scénu, ve kterém se odehrává hlavní modelování a vytváření hry. Běžný způsob vložení herního objektu do scény probíhá pomocí „táhni-a-pust“ z Okna projektu do Hierarchy panelu, nebo přímo do scény. Zatímco Pohled na scénu umožňuje volný pohyb v celém prostoru. Pohled na hru ukazuje pouze to, co vidí hráč během hraní. Rovněž je možné rozpracovanou hru v editoru spustit a v tomto okně testovat. Při vybrání jakékoliv položky se blok Inspector panel postará o vypsání dostupných informací spojených s vybraným předmětem. [40]



**Obrázek 7 Rozložení Unity editoru. Zdroj: [autor]**

## **7.2 Základní komponenty herních objektů**

Komponenty definují chování objektu vyskytujícího se ve scéně. Výpis v panelu Inspector je rozčleněn do jednotlivých komponent přiřazených k aktuálně vybranému objektu. Manipulace s nimi probíhá přímo v grafickém rozhraní editoru nebo pomocí skriptů. Každý herní objekt obsahuje komponentu Transform uchováající uloženou pozici, rotaci a měřítko, vzhledem k hernímu světu a ke svému „rodiči“. Ve výchozím stavu nově vytvořený projekt začíná s objektem Hlavní Kamera. Ta je nakonfigurována jako primární kamera hry. Vybavení komponentami Transform, Audio listener a Camera zajišťuje kameře snímání scény vizuálně i zvukově. [35]

### **7.2.1 Pohyb**

Rigidbody je název hlavní komponenty umožňující chování objektu podle fyzikálních zákonů hry. Od chvíle, kdy je připojena Rigidbody, není doporučeno měnit polohu objektu komponentou Transform, ale aplikovat sílu ke změně polohy objektu správným směrem a přenechat fyzikálnímu systému výpočet výsledku. Ovšem vyskytují se případy, kdy je žádoucí, aby objekt při pohybu nepodléhal dalším výpočtům fyzikálního systému, zejména když je potřeba nadále detekovat ostatní objekty. Udávat pohyb postavě pouze skriptem je druh pohybu, který se nazývá kinematický [39]. Kromě toho je v Rigidbody možné nastavit další typ pohybu. Tím je pohyb Dynamický, který plně využívá fyzikální systém a je nejnáročnějším na výpočet. Jeho protikladem je statická Rigidbody komponenta, kde pohyb není umožněn vůbec.

Pokud rychlost pohybu klesne pod definovanou hranici minima, fyzikální systém vyhodnotí objekt jako stojící a uvede ho do režimu spánku (z anglického Sleeping mode). Díky této optimalizační vlastnosti procesor nestráví žádný čas aktualizováním Rigidbody „spícího“ objektu, do té doby, než bude uveden zpět do pohybu. Tuto funkci lze v editoru explicitně zakázat. [39]

### **7.2.2 Kolize**

Collider komponenty definují tvar, určený pro výpočet kolizí mezi objekty, přičemž tvar používaný Colliderem nemusí souhlasit s přesným obrysem herního objektu. Z hlediska náročnosti na procesor jsou nejefektivnější takzvané primitivní Collider [32], které mají v 2D prostoru tvar obdélníku či kruhu. Pokud jednoduché tvary nepůsobí dostatečně reálně, je třeba použít mnohoúhelníkový Collider (z anglického Polygon Collider), jímž je možné věrně napodobit tvar objektu. Faktem zůstává, že primitivní Collider nezatěžují výpočet v takové míře, jako mnohoúhelníkové. Z toho důvodu je možné přiřadit jednomu objektu více kolizních komponent kteréhokoliv druhu. Pokud je Collider nastaven jako spoušť (z anglického Trigger), herní objekt se přestane chovat jako pevný předmět, přičemž fyzikální systém dále detekuje ostatní objekty, ale bez vytváření kolizí. Collider společně s Rigidbody vytváří dojem pevného tělesa reagujícího na okolní síly, díky zachycení kolize a možností pohybové reakce. Collider je označován jako statický, pokud herní objekt nemá komponentu Rigidbody. Může sice zaznamenávat kolize, ale není schopen pohybu. Statické Collider jsou ideální pro zdi, střechy a další bezpohybové omezující elementy scény. [32]

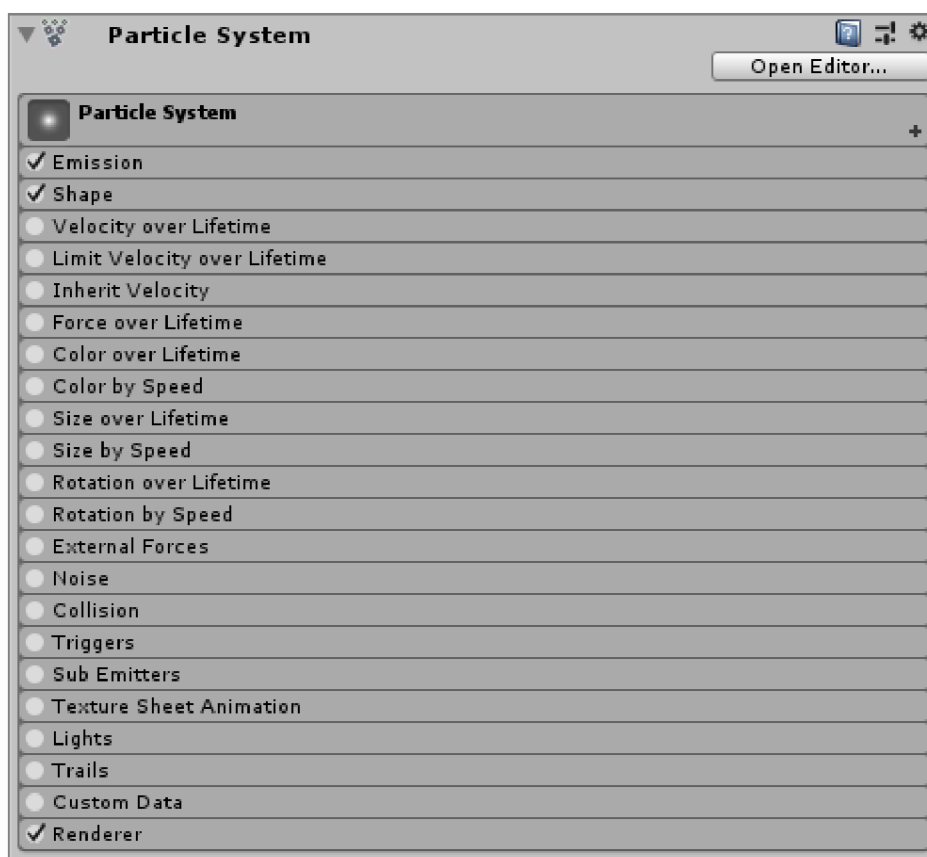
### **7.2.3 Částicový systém**

Užitečná komponenta s názvem Částicový systém, efektivně simuluje a vykresluje velké množství malých obrázků, které jsou pojmenované jako částice, za účelem vytvoření vizuálního efektu. Systém kolektivně simuluje všechny částice k dosažení reálného dojmu, přičemž každý obrázek reprezentuje individuální grafický element. Použití je žádoucí v případě vytváření dynamického objektu, jímž může být oheň, kouř nebo tekutina. [37]

Po zobrazení herního objektu s Částicovým systémem, je v Inspector panelu zobrazeno mnoho vlastností, kterými lze komponentu přizpůsobovat vizuálnímu efektu. Množství

použitelných funkcí lze vidět na Obrázek 8. Mezi nejčastěji používané patří zejména tyto moduly:

- Particle systém: nastavuje základní rysy, například délku trvání efektu, životnost, velikost, rychlost a omezení počtu částic.
- Emission: udává počet generovaných částic za sekundu.
- Shape: omezuje odkud a jakým směrem může být obrázek vyslán.
- Texture Sheet Animation: umožňuje přidat obrázky, které mají být generovány.
- Renderer: vykresluje částice.



**Obrázek 8 Ukázka komponenty Částicový systém. Zdroj: [36]**

## 7.2.4 Zvuk

Flexibilní Unity Audio systém obsahuje sofistikované vlastnosti pro přehrávání zvuků ve 3D světě a umožňuje import souborů do Okna projektu ve standardních zvukových formátech jako AIFF, WAV, MP3 a Ogg. Pro správné fungování zvukového modelu je nutná komponenta Audio Source, jež slouží jako zdroj zvuku ve scéně. Komponenta

Audio Listener, přijímající okolní zvuky, je přiřazena nejčastěji k hlavní kameře. Každý objekt vysílající zvuk musí mít komponentu Audio Source, přizpůsobenou jejímu účelu. Například hudba na pozadí je přehrávána cyklicky a hraje stejnou hlasitostí po celém herním světě, na rozdíl od jednorázových 3D prostorových zvuků, které jsou přehrávány v závislosti na prostředí a vzdálenosti od posluchače. [29]

### **7.3 Skriptování**

Společně s Unity enginem je instalováno Microsoft Visual Studio, sloužící jako integrované vývojářské prostředí, vybavené potřebnými knihovnamí pro vývoj v Unity. Po založení C# skriptu přímo v Unity editoru, vznikne spojení s interním fungováním Unity enginu, díky implementování nové třídy [31]. Do třídy jsou vloženy dvě bezparametrické funkce. První z nich, s názvem Start, je volána před prvním snímkem hry, díky tomu tvoří ideální místo na provedení inicializací. Kód, uvnitř druhé přidané funkce Update, se provádí každý snímek.

#### **7.3.1 Základní funkce**

Pokud se herní objekt vyskytuje ve hře napříč úrovněmi vícekrát, je vhodné uložit jej do Okna projektu jako Prefab, kde bude uchován se všemi komponentami a s jejich nastavením. Systém umožňuje synchronizaci Prefab souborů s objekty ve scéně. Není tedy nutné kopírovat konfiguraci zvlášť u všech jednotlivých objektů. Například často používané typy stromů, nebo běžně se vyskytující nepřátelé, jsou vhodné pro Prefab.

Ne však všechny objekty figurují ve scéně od počátku hraní, kupříkladu projektily musí být tvořeny až při střelbě. K tomu slouží funkce Instantiate, přijímající tři argumenty, které mimo jiné určují, na jaké pozici a s jakou rotací má být objekt vytvořen. Jako příklad je na Obrázek 9 uveden kód, kdy v případě vyhovující podmínky vystřel (shoot), je vytvořen a následně přiřazen Prefab šípu do proměnné arrow (arrowPrefab) s definovanou pozicí (hand.position). Poslední parametr Quaternion.identity je zkratkou pro použití výchozí rotace. Na dalším řádku se s využitím funkce GetComponent, přistupuje k Rigidbody2D komponentě objektu, díky které je šíp vystřelen příslušným směrem (shootingDirection) stanovenou rychlostí (arrowVelocity).

```

if (shoot)
{
    arrow = Instantiate(arrowPrefab, hand.position, Quaternion.identity);
    arrow.GetComponent<Rigidbody2D>().AddForce(shootingDirection * arrowVelocity);
}

```

**Obrázek 9 Funkce Instantiate a GetComponent. Zdroj: [autor]**

Práce s časem a jeho odpočtem je mnohokrát nezbytnou součástí herního vývoje. Třída Time umožňuje vytvořit běžně používaný časovač, pomocí přístupu k parametru deltaTime, jenž určuje časový interval od vykreslení předposledního snímku po současný. Ve spolupráci s další důležitou třídou Mathf, poskytující kolekci běžně používaných matematických operací, je kupříkladu možné dosáhnout plynulého přechodu mezi dvěma hodnotami za definovaný čas. Tento případ je znázorněn na Obrázek 10, kde s využitím funkce Mathf.Lerp, pro výpočet lineární interpolace, je určována nová pozice objektu. Parametr t, může nabývat hodnot od 0 do 1, definuje v jakém místě mezi dolní a horní hranicí má být vypočítána interpolace. Pokud v kódu na Obrázek 10 parametr t přesáhne hodnotu 1, je vynulován a hranice jsou prohozeny. To má za následek, že pozice bude následující sekundu interpolovat v opačném směru. Protože se jedná o vnitřek funkce Update, je tento kód prováděn každý snímek.

```

public class Example : MonoBehaviour
{
    float minimum = -1;
    float maximum = 1;

    float t = 0;

    © Zpráva Unity | Počet odkazů: 0
    void Update()
    {
        transform.position = new Vector3(Mathf.Lerp(minimum, maximum, t), 0, 0);

        t += Time.deltaTime;

        if (t > 1.0f)
        {
            float temp = maximum;
            maximum = minimum;
            minimum = temp;
            t = 0.0f;
        }
    }
}

```

**Obrázek 10 Lineární interpolace. Zdroj: [autor]**

Při vzniku kolize fyzikální systém zavolá specifickou funkci, do které může být vložena odpověď v podobě kódu vzhledem k nastalým situacím. Je-li kolize na začátku, bude volána funkce `OnCollisionStart`, následně bude opakovaně prováděna funkce `OnCollisionStay`. Při opuštění kolize bude pouze jednou zavolána funkce `OnCollisionExit`. To ovšem sebou nese několik podmínek, jednou z nich je, že oba objekty musí mít `Collider` a zároveň nesmí být ani jeden nastaven jako `spoušť`. Jinak je nutné použít obdobné funkce jako `OnTriggerEnter`, `OnTriggerStay` a tak dále.

### 7.3.2 Ovládání

Unity engine podporuje vstupy od mnoha typů zařízení, konkrétně od:

- klávesnice a myši,
- joysticku,
- ovladače,
- dotykové obrazovky,
- akcelerometrů, nebo gyroskopů mobilních zařízení,
- VR a AR ovladačů. [33]

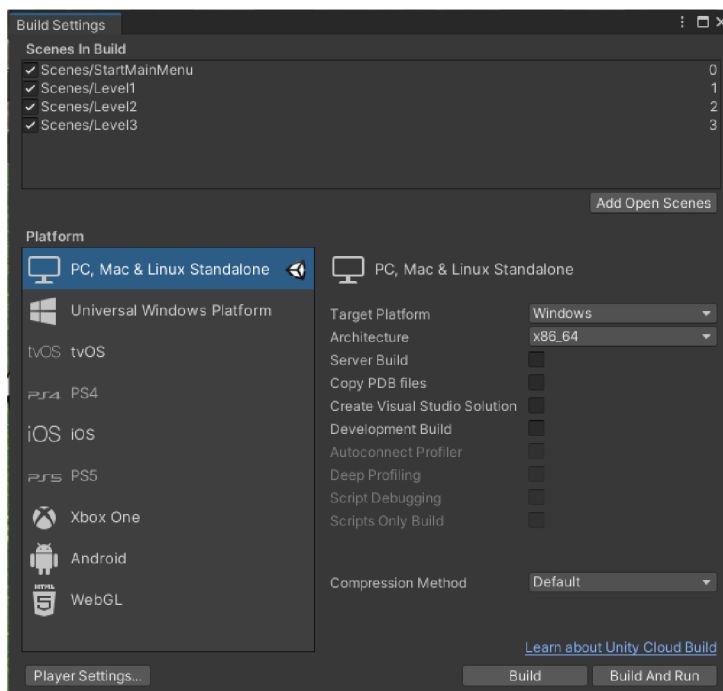
`Input Manager` je základní nástroj pro snímání uživatelského vstupu. Každý vytvořený projekt obsahuje několik vstupních virtuálních os, kterými jsou monitorovány příchozí data. Ze všeho nejdříve je nutné namapovat pozitivní a negativní tlačítka. Například výchozí nastavení horizontální osy pro klávesnici, může nabývat hodnoty  $-1$  až  $1$ , přičemž tlačítka jsou pravá a levá šipka. Pokud není stisknuta žádná klávesa, osa vrací hodnotu  $0$ . Naopak při držení tlačítka se během krátkého časového intervalu hodnota změní na hodnotu hraniční. V praxi to znamená, že ovládaná postava, využívající virtuální osy, při začátku pohybu, pomalu zrychlí. [34]

Při pohybu v herním světě je mnohokrát zapotřebí identifikovat ostatní objekty, aniž by muselo dojít ke kolizi. Řešením je vrhnutí paprsku s použitím funkce `Raycast`. Jedná se o laser paprsek vystřelený z určité pozice daným směrem. Pro úplné využití možností funkce jsou k dispozici volitelné parametry určující délku paprsku nebo filtrování výsledků. První vyhovující zasažený objekt s `Colliderem` je detekován a je možné s ním nadále pracovat. Pro vrácení všech zasažených objektů je nutné použít `RaycastAll` funkci. [38]



## 7.4 Export hry

V libovolné fázi vývoje je možné hru exportovat do samostatně spustitelného souboru. Pokud není definováno jinak, před začátkem exportování Unity automaticky doplní informace týkající se projektu, zejména název hry, číslo verze, jméno studia či ikonu hry. Tyto položky lze změnit a přizpůsobit, aby byly v souladu s projektem. Nastavení exportu probíhá v editoru pomocí okna Build Settings, které je vyobrazeno na Obrázek 11. Do první části (Scenes in Build) musejí být vloženy všechny scény vyskytující se ve hře. Pořadí vkládání je závazné, přinejmenším pro první vloženou scénu s indexem 0, neboť ta bude při zapnutí hry spuštěna jako první. Seznam vyskytující se v levé dolní části okna, slouží k vybrání cílové platformy pro export. Modul pro počítač je automaticky nainstalován společně s Unity engine. Případné další platformy je nutné stáhnout z oficiálních zdrojů, k některým je navíc zapotřebí licence. Zbývající část je určena k upřesnění exportu pro vybranou platformu, kde kromě cílového operačního systému je vhodné vybrat i jeho architekturu. Ostatní volby umožňují například zvolit export pro vývojáře (z anglického Development Build), který je přizpůsoben pro testovací účely. Vše je završeno tlačítky Build a Build And Run, jimiž je po stisknutí odstartován export. Rozdíl je mezi nimi v tom, zda se po dokončení exportu aplikace automaticky spustí či nikoliv. [30]



Obrázek 11 Nastavení exportu hry.

Zdroj: [autor]

## 8 Závěr

Cílem práce je seznámení s prostředím herního vývojářství a oblastí videoher obecně. Hodnota herního průmyslu roste každým rokem o jednotky procent. V současnosti se jedná o trh s několika set miliardovou hodnotou. Firmy neustále pracují na nových technologiích, kterými se snaží docílit co nejuživnějšího a zároveň nejoriginálnějšího zážitku ze hry pro své uživatele. Toto dalo za vznik mnoha kategoriím her a zároveň novým a inovativním platformám.

V první části této práce jsou vysvětleny základní pojmy týkající se herního vývojářství, včetně členění. Následuje výčet technologií používaných v herním průmyslu i v historickém kontextu. Každé technologii je věnována podkapitola, v níž je podrobně charakterizována. Dále se pozornost přesouvá k projekci především 2D a 3D prostoru. V kapitole poté přichází na řadu rozebrání principů návrhu hry, včetně důkladně popsané doporučené struktury Game Design Dokumentu, na jejímž základě byl vytvořen GDD bakalářského projektu, který je k práci připojen. Pro demonstraci funkcionalit vývojového prostředí, je pro tuto práci zvolen Unity engine. Jedná se o komplexní software, který je oblíbený zejména u menších vývojářských studií a zároveň již dal vzniknout spoustě herních titulů. Vydavatel zprostředkovává na svých oficiálních stránkách výukové plány. Firma Unity nabízí svůj engine zdarma, pokud hodnota výtěžku ze hry nepřekročí během jednoho roku 100 000\$. Samotná implementace bakalářského projektu, ve které je využit Unity engine tvoří poslední kapitolu.

## 9 Seznam použité literatury

- [1] BASLER, Jaromír. COMPUTER GAMES, THEIR DIVISION, CONTEMPORARY DEVELOPMENT TENDENCIES AND BASIC INVESTIGATIONS FROM THE FIELD OF COMPUTER GAMES. Trends in Education [online]. 2016, 9(1), 20-27 [cit. 2020-08-05]. ISSN 18058949. Dostupné z: doi:10.5507/tvv.2016.003
- [2] BETHKE, Erik. Game Development and Production. Plano (Texas): Wordware Publishing, 2003. ISBN 1-55622-951-8.
- [3] BURŠOVÁ, Irena, 2015. Na počátku bylo slovo. A to slovo bylo Spacewar!. In: Rádiožurnál [online]. 16.12.2015 [cit. 2020-08-06]. Dostupné z: <https://radiozurnal.rozhlas.cz/na-pocatku-bylo-slovo-a-slovo-bylo-spacewar-6234957>
- [4] ČÁPKA, David. Use Case Diagram. ITnetwork [online]. [2013] [cit. 2021-01-23]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>
- [5] ČÁPKA, David. Úvod do UML. ITnetwork [online]. [2013] [cit. 2021-01-23]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>
- [6] DUBE, Kayla. 8 Best 2D Games for Windows. Online Tech Tips [online]. 24. 10. 2020 [cit. 2021-02-07]. Dostupné z: <https://www.online-tech-tips.com/gaming/8-best-2d-games-for-windows/>
- [7] DEALESSANDRI, Marie. What is the best game engine: is CryEngine right for you? Gamesindustry [online]. 16. 1. 2020 [cit. 2021-02-02]. Dostupné z: <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-cryengine-the-right-game-engine-for-you>
- [8] DEALESSANDRI, Marie. What is the best game engine: is Unity right for you? Gamesindustry [online]. 16. 1. 2020 [cit. 2021-02-02]. Dostupné z: <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you>
- [9] DEALESSANDRI, Marie. What is the best game engine: is Unreal Engine right for you? Gamesindustry [online]. 16. 1. 2020 [cit. 2021-02-02]. Dostupné z: <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unreal-engine-4-the-right-game-engine-for-you>

- [10] DOSTÁL, Jiří. Výukové programy[online]. Olomouc: Univerzita Palackého v Olomouci, 2011. ISBN 978-80-244-2804-8. Dostupné z: <https://books.google.cz/books?hl=cs&id=cgcLE7u7CJ8C&q=Po%C4%8D%C3%ADta%C4%8Dov%C3%A1+hra#v=snippet&q=Po%C4%8D%C3%ADta%C4%8Dov%C3%A1%20hra&f=false> .
- [11] Unity Game Engine Guide: How to Get Started with the Most Popular Game Engine Out There. FreeCodeCamp [online]. 13. 2. 2020 [cit. 2021-02-02]. Dostupné z: <https://www.freecodecamp.org/news/unity-game-engine-guide-how-to-get-started-with-the-most-popular-game-engine-out-there/>
- [12] 2.5D: Concept. Giant Bomb [online]. 3. 1. 2021 [cit. 2021-02-08]. Dostupné z: <https://www.giantbomb.com/25d/3015-660/>
- [13] GREGORY, Jason. Game Engine Architecture. 3rd ed. Boca Raton: Taylor & Francis, CRC Press, 2018. ISBN 9781138035454.
- [14] HETFELD, Malindy, Samuel HORTI a Diego ARGUELLO. 30 best PC games to play right now. GamesRadar [online]. 19. 1. 2021 [cit. 2021-02-07]. Dostupné z: <https://www.gamesradar.com/best-pc-games/4/>
- [15] HORNSHAW, Phil, 2019. The history of Battle Royale: From mod to worldwide phenomenon. In: DigitalTrends [online]. 10.4.2019[cit. 2020-08-05]. Dostupné z: <https://www.digitaltrends.com/gaming/history-of-battle-royale-games/>
- [16] KOLÁČEK, M. Freemium hry. In: JIRKOVSKÝ, J. Game Industry 3. Praha: D.A.M.O., 2013. s. 73–83. ISBN 978-80-904387-4-3.
- [17] Limelight. THE STATE OF ONLINE GAMING – 2019[online]. 25.1.2019-14.2.2019 [cit. 2020-08-06]. Dostupné z: <https://www.limelight.com/resources/white-paper/state-of-online-gaming-2019/>
- [18] 10 things you didn't know about mobile gaming. Microsoft [online]. 16.1.2013, 1.10.2015 [cit. 2020-08-06]. Dostupné z: <https://blogs.windows.com/devices/2013/01/16/10-things-you-didnt-know-about-mobile-gaming-2/>
- [19] PHAN, H. M. Video Gaming Trends: Violent, Action/Adventure Games are Most Popular. Software Usability Research Laboratory [online]. 2011. [cit. 2020-08-05]. Dostupné z: <http://usabilitynews.org/video-gaming-trends-violent-actionadventure-games-are-mostpopular/>

- [20] POH, Michael. Evolution of Home Video Game Consoles: 1967 – 2011. In: Hongkai [online]. 23.6.2020 [cit. 2020-08-06]. Dostupné z: <https://www.hongkiat.com/blog/evolution-of-home-video-game-consoles-1967-2011/>
- [21] ROUSE, Richard a Steve OGDEN. Game Design: Theory and Practise. 2nd ed. Plano (Texas): Wordware Publishing, 2005. ISBN 1-55622-912-7.
- [22] NIRWAN, Debby. Designing AI Agents' Behaviors with Behavior Trees: Designing and Implementing Behaviors for AI Agents. Towards data science [online]. 6. 12. 2020 [cit. 2021-02-01]. Dostupné z: <https://towardsdatascience.com/designing-ai-agents-behaviors-with-behavior-trees-b28aa1c3cf8a>
- [23] SAURAV MAJHI, Samir, 2011. 16 Different Types of Massively Multiplayer (MMO) Online Games. In: DragonBlogger[online]. 23.5.2011 [cit. 2020-08-05]. Dostupné z: <https://www.dragonblogger.com/16-types-massively-multiplayer-mmo-online-games/>
- [24] SMITHEE, Alan, 2019. PC Gaming Vs. Console Gaming – The Pros And Cons[online]. 9.4.2019 [cit. 2020-08-06]. Dostupné z: <https://unwinnable.com/2019/04/09/pc-gaming-vs-console-gaming-the-pros-and-cons/>
- [25] STEGNER, Ben. 2D Games vs. 3D Games: What Are the Differences? Muo [online]. 7. 10. 2020 [cit. 2021-02-06]. Dostupné z: <https://www.muo.com/2d-games-vs-3d-games-differences/>
- [26] What is game engine?: Game Engine and History of Game Development. Studytonight [online]. 17. 7. 2017 [cit. 2021-01-30]. Dostupné z: <https://www.studytonight.com/3d-game-engineering-with-unity/game-engine>
- [27] 2D and 3D Games. UKEssays [online]. říjen 2018 [cit. 2021-02-06]. Dostupné z: <https://www.ukessays.com/essays/video-games/2d-and-3d-games.php>
- [28] Unity [online]. 2021 [cit. 2021-02-02]. Dostupné z: <https://store.unity.com/compare-plans?currency=USD>
- [29] Audio Overview. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-08]. Dostupné z: <https://docs.unity3d.com/Manual/AudioOverview.html>

- [30] Build Settings. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-12].  
Dostupné z: <https://docs.unity3d.com/Manual/BuildSettings.html>
- [31] Creating and Using Scripts. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-09]. Dostupné z: <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>
- [32] Colliders. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-08]. Dostupné z: <https://docs.unity3d.com/Manual/CollidersOverview.html>
- [33] Input. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-10]. Dostupné z: <https://docs.unity3d.com/Manual/Input.html>
- [34] Input Manager. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-11]. Dostupné z: <https://docs.unity3d.com/Manual/class-InputManager.html>
- [35] Introduction to components. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-08]. Dostupné z: <https://docs.unity3d.com/Manual/Components.html>
- [36] Particle system. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-08]. Dostupné z: <https://docs.unity3d.com/Manual/class-ParticleSystem.html>
- [37] Particle systems. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-08]. Dostupné z: <https://docs.unity3d.com/Manual/ParticleSystems.html>
- [38] Physics2D.Raycast. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-11]. Dostupné z: <https://docs.unity3d.com/ScriptReference/Physics2D.Raycast.html>
- [39] Rigidbody overview. Unity: Documentation [online]. 5.4.2021 [cit. 2021-04-08]. Dostupné z: <https://docs.unity3d.com/Manual/RigidbodiesOverview.html>
- [40] Setting Up the Unity Editor. Unity: Learn [online]. 20.1.2021 [cit. 2021-04-08]. Dostupné z: <https://learn.unity.com/tutorial/setting-up-the-unity-editor?uv=2020.2&projectId=5c6166dbedbc2a0021b1bc7c#>
- [41] Unreal engine [online]. 2021 [cit. 2021-02-03]. Dostupné z: <https://www.unrealengine.com/en-US/release>
- [42] W., Bryan. GameMaker Engine Review and Tutorials. GameDesigning [online]. 5. 1. 2019, 20. 12. 2020 [cit. 2021-02-05]. Dostupné z: <https://www.gamedesigning.org/engines/gamemaker/>
- [43] Wepc. Video game industry statistics in 2020 [online]. 14.8.2020 [cit. 2020-08-06]. Dostupné z: <https://www.wepc.com/news/video-game-statistics/>

- [44] YoYo Games [online]. 2021 [cit. 2021-04-04]. Dostupné z:  
<https://www.yoyogames.com/en/get>
- [45] ŽÁRA, Jiří, Bedřich BENEŠ, Jiří SOCHOR a Petr FELKEL. Moderní počítačová grafika [online]. 2. vydání. Brno: Computer Press, 2005 [cit. 2021-04-04]. ISBN 80-251-0454-0. Dostupné z:  
<https://dcgi.fel.cvut.cz/cgg/ModerniPocitacovaGrafika/>

## 10 Přílohy

### 10.1 Příloha 1 – Game Design Dokument

# Bakalářský projekt - Game Design Dokument

Owned by Jan Dušek and Created By Jan Dušek - version: 0.0

Powered by Dundoc

#### 1. Příběh

Příběh hry o honbě za svatým grálem je jednoduchý a přímočarý. Na začátku hry je nastíněna aktuální situace a sice taková, že svatý grál se právě našel, ale má ho v držení nepřátelská frakce. Hlavní hrdina je pověřen, aby se se svým průvodcem Qarellem vydal grál hledat. Jakmile hlavní hrdina doputuje do nepřátelské pevnosti je proti němu poslána armáda rytířů a nemrtvých. Pokud hlavní hrdina uspěje, utká se s hlavním nepřítelem po jehož poražení se objeví svatý grál.

#### 2. Herní mechaniky

##### ◦ Pohyb

Klávesy pro ovládání hlavního hrdiny musí být co nejvíce intuitivní. Při držení tlačítka pro pohyb se hlavní hrdina vydá požadovaným směrem.

Po zmáčknutí klávesy pro pohyb se hlavní hrdina začne pohybovat rychlostí chůze, teprve po malé chvíli se rozeběhne do plné rychlosti.

Hlavní hrdina se zastaví o kolidující předměty. S nepřátelskými postavami se může přetlačovat.

##### ◦ Předměty

Základní výbava hráče se skládá z kompletního brnění, zbraně na blízko, luku, štítu a léčivých lektvarů.

Všechny předměty hráč získá po sebrání ze země. Nový předmět ihned nahradí ten dosavadní. V případě léčivých lektvarů se zvýší počet o jeden.

Při použití léčivého lektvaru bude ihned vyléčena nanejvýš polovina maximálních životů. Lektvar nemůže být použit při plném zdraví.

##### ◦ Inventář/Statistiky

Ve hře je možné zobrazit aktuální statistiky kterými disponuje hrdina.

- Útočná síla
- Maximální zdraví
- Redukce poškození
- Počet léčivých lektvarů

Po sebrání předmětu dojde ihned k aktualizaci dat.

##### ◦ Grafické rozhraní

Nad každou bojeschopnou postavou je zobrazen aktuální stav jejich životů.

Rozhovory a rady se vypisují na tabuli v podobě kamenné, papírové, či dřevěné desky. Věty se po několika vteřinách střídají. Tabule je zobrazena pouze v případě její potřeby, mimo rozhovory není vidět.



#### • Boj

Při stisknutí tlačítka útoku hráč začne provádět útok. Dát poškození může pouze v případě, že zbraň během letu zasáhne nepřítele, který musí být v dostatečné vzdálenosti. To stejné platí i v opačném případě, kdy má poškození utržit hráč.

Hráč může držením příslušného tlačítka využívat štít. Pokud je hráč za štítem, utrží menší poškození. Se štítem se hráč nemůže hýbat.

Pokud postava ve hře zasáhne jinou postavu, přeruší se případný probíhající útok zasažené postavy. Poté je poškození zmenšeno o redukcí poškození pomocí brnění a teprve poté případně i o redukcí štítu.

Například je veden útok proti hrdinovi o síle 100. Hrdina je vybaven brněním ignorujícím 20% poškození a zároveň se chrání svým štítem, který ignoruje 50%. Finální síla útoku je rovna vzorci  $Síla\ útoku * (1 - Redukce\ poškození\ brnění) * (1 - Redukce\ poškození\ štítu) = Reálná\ síla\ útoku$ , tedy  $100 * (1 - 0,2) * (1 - 0,5) = 40$

V případě luku je nutné nejdříve luk natáhnou, zasadit šíp a teprve poté vystřelit.

Šíp je možné vystřelit pod přirozeným úhlem dopředu tj. nesmí střílet šípem za sebe, či přímo nahoru. I když hráč bude chtít vystřelit nad sebe, bude šíp vypuštěn pouze pod jeho maximálním úhlem. Při míření opačným směrem než je postoj hrdiny bude šíp sesazen, nikoliv vystřelen.

#### • Kamera

Ve výchozí situaci je hlavní hrdina usazen doprostřed kamery.

Hráč může držením tlačítka a tažením myši pohybovat kamerou, po uvolnění se kamera vrátí zpět na střed. Hráč má možnost kamerou hýbat pouze do určité vzdálenosti. Hráč se nemůže rozhlížet v případě, že se nachází v jakékoliv budově.

Pokud je hráč uvnitř budovy, venkovní svět je zakryt černou barvou. Z venku je do normálních budov umožněno vidět. Ve speciálních budovách je vnitřek odkryt až při vstoupení.

### 3. Umělá inteligence

Je nutné dohlížet na herní tok. Umělá inteligence spouští monology průvodce pokud hráč dosáhne požadovaného bodu v úrovni. Spouštěčem může být vstoupení na definované území, zničení nepřítele, sebrání předmětu a další. Při nutnosti je možné aktivovat/deaktivovat hráčův pohyb, nebo útok.

Nepřítel se dělí na dva typy: první z nich se při narušení jeho zóny k hlavnímu hrdinovi rozběhne a bezhlavě po něm útočí. Druhý typ má za úkol pouze dojít na nějaké území, nemá potřebu po nikom útočit.

## 4. Herní prvky

---

### ◦ Postavy

Ve hře se vyskytují následující postavy:

- **Hlavní hrdina**
  - ovládán hráčem
  - po celou hru zůstává neměnný (kromě jeho vybavení)
- **Průvodce**
  - pomocník hlavního hráče
  - rozdává úkoly a rady
  - není možné na něj útočit
  - kouzelný neběžný vzhled
  - nenesí zbraň
- **Rytíř vandal**
  - první nepřítel
  - běžný až trochu chudý vzhled
  - agresivní chování
  - neměl by být velkou výzvou
- **Běžný lidský nepřítel**
  - základní nepřítel
  - je možné bojovat i proti více nepřátelům tohoto druhu najednou
  - lehce vznešený vzhled
  - průměrná obtížnost
- **Blázen**
  - nejjednodušší forma nepřítele
  - chudý vzhled
  - špatné statistiky
  - nepředstavuje žádnou výzvu
  - silné agresivní chování
- **Vznešený rytíř**
  - bohatý vzhled lidského rytíře
  - větší obtížnost, pravděpodobně nutné použít léčivé lektvary během boje
  - agresivní chování
  - pomalý pohyb
  - silné útoky

- **Vznešený rytíř**

- bohatý vzhled lidského rytíře
- větší obtížnost, pravděpodobně nutné použít léčivé lektvary během boje
- agresivní chování
- pomalý pohyb
- silné útoky

- **Nebojující nepřátelé**

Tito nepřátelé nebojují, ale mají za úkol dojít do určité destinace. Hlavní hrdina je musí zničit, než se jim to podaří, jinak končí hra.

- Rytíř - běžný lidský vzhled s normálním počtem životů
- Nemrtvý1 - nevznešený vzhled nelidského bojovníka s průměrným počtem životů
- Nemrtvý2 - vzhled mumie, velice rychlý, ale málo životů
- Nemrtvý3 - nebezpečný vzhled, více životů
- Hlavní vůdce tažení - vznešený nelidský vzhled s velkým počtem životů

- **Hlavní nepřátelský vůdce**

- poslední nepřítel, kterého musí hlavní hrdina porazit
- při malém množství životů použije léčivý lektvar (jednou)
- největší výzva z celé hry
- odolný proti šípům
- urostlé postavy
- temný
- ne z našeho světa

- **Předměty a prostředí**

Hra disponuje těmito prvky prostředí:

- stromy - okrasné předměty hry
- zdi, stěny, střechy - neprůchodné části světa
- tráva - základní stavební jednotka pro výplň světa
- cesta - vede celou úrovní
- budovy - ne do všech lze vstoupit
- okna

V hře je možné narazit na tyto předměty:

- běžné vybavení - hrdina může na začátku hry sebrat brnění/štit/meč, které zajistí základní statistiky
- vylepšené vybavení - v průběhu hry hrdina přejde na lepší, vznešenější vybavení, místo meče zaujme sekera
- léčivý lektvar
- luk
- šíp
- cvičný terč - nehybná postava, která po zasažení šípem exploduje

## 5. Herní progres

---

### ◦ Úroveň 1

- především seznamovací mise
- uvedení hráče do příběhu
- představení základního ovládání
- na začátku bez jakéhokoliv vybavení
- vybavení je rozmístěno podél cesty
- na konci první nepřítel
- po poražení nepřítele konec 1. úrovně

### ◦ Úroveň 2

- začátek v budově
- v této úrovni je zaměřeni na boj zblízka
- nejdříve dva běžní nepřátelé, poté Blázen v samostatné budově a na konci Vznešený rytíř
- během úrovně hráč sbírá vzácnější vybavení
- konec nastává pokud hráč dorazí před pevnost a má na sobě požadované vybavení

### ◦ Úroveň 3

- na začátku výcvik s lukem
  - bránění území s pomocí luku před příchozími skupinami
  - v zakryté budově na konci kola se nachází vůdce nepřátel
  - po poražení generála se objeví svatý grál
  - hra končí po jeho sebrání
- 

## 6. Herní menu

Celá hra obsahuje pouze jedno hlavní menu, které se automaticky zobrazuje při zapnutí, nebo během hry, když je zmáčknuta příslušná klávesa. Čas ve hře je pozastaven během pohybování se v hlavním menu. Z menu je možné spustit novou hru odznovu, načíst konkrétní level, otevřít nabídku nastavení, či ukončit hru.

Konkrétní level lze načíst pouze v případě je-li zpřístupněn. Kupříkladu po dohrání úrovně 2 se zpřístupní úroveň 3 atd. Progres se ukládá do lokálního úložiště.

---



## Zadání bakalářské práce

<b>Autor:</b>	<b>Jan Dušek</b>
Studium:	I1800164
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
<b>Název bakalářské práce:</b>	<b>Vývoj 2D hry s využitím Unity engine</b>
Název bakalářské práce AJ:	Development of 2D game using Unity engine

### Cíl, metody, literatura, předpoklady:

Cíl práce: Cílem bakalářské práce je představit technologie pro vývoj počítačových her a následném vytvoření 2D hry s pomocí engine Unity.

#### Osnova:

- Úvod
- Videohry a jejich platformy
- Technologie pro vývoj her
- Dimenze prostoru
- Návrh a herní design
- Implementace počítačové hry
- Závěr

Garantující pracoviště: Katedra informatiky a kvantitativních metod,  
Fakulta informatiky a managementu

Vedoucí práce: Ing. Michal Macinka

Oponent: doc. Ing. Filip Malý, Ph.D.

Datum zadání závěrečné práce: 14.1.2018