



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

3D First-person Shooter hra

Bakalářská práce

Studijní program:

B0613A140005 Informační technologie

Studijní obor:

Aplikovaná informatika

Autor práce:

Adrian Bartoň

Vedoucí práce:

Ing. Jan Hybš

Ústav nových technologií a aplikované informatiky





Zadání bakalářské práce

3D First-person Shooter hra

Jméno a příjmení: **Adrian Bartoň**
Osobní číslo: M19000193
Studijní program: B0613A140005 Informační technologie
Specializace: Aplikovaná informatika
Zadávací katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2021/2022**

Zásady pro vypracování:

1. Proveďte rešerši herních engine vhodných pro realizaci 3D FPS her.
2. Navrhněte jednoduchý herní děj a prostředí, ve kterém bude herní postava existovat.
3. Implementujte demonstrační 3D FPS hru, obsahující kolizní systémy hit-scan a projectile.
4. Kriticky zhodnoťte výsledek práce po stránce optimalizace, kompatibility a rozšiřitelnosti.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
30-40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] Hae Kyung Rhee, Doo Heon Song and Kim, J. (2019). *Comparative analysis of first person shooter games on game modes and weapons-military-themed, overwatch,...* [online] ResearchGate. Available at: https://www.researchgate.net/publication/330159552_Comparative_analysis_of_first_person_shooter_games_on_game_modes_and_weapons-military-themed_overwatch_and_player_unknowns_battleground
- [2] Soo-Chang Kang and Chris SeoYun Choi (2015). *A Case Study on PVP Level Design Patterns of FPS games – Focusing on 'Point Blank' -*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/305207858_A_Case_Study_on_PVP_Level_Design_Patterns_of_FPS_games_-_Focusing_on_'Point_Blank'_
- [3] Lafortune, S.R., Blackman, J.C. and Myers, A.E. (2016). *The Effects of Latency and Jitter on a First Person Shooter: Team Fortress 2*. [online] Digital WPI. Available at: https://digital.wpi.edu/concern/student_works/v118rf17s?locale=en

Vedoucí práce:

Ing. Jan Hybš
Ústav nových technologií a aplikované informatiky

Datum zadání práce:

12. října 2021

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2021

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

12. 5. 2022

Adrian Bartoň

Poděkování

Především bych chtěl poděkovat vedoucímu mé bakalářské práce panu Ing. Janu Hybšovi za podporu při tvorbě praktické části videohry a následné dokumentaci. Také děkuji za poskytnutý čas v podobě konzultací a poskytnutí cenných rad. Rád bych také poděkoval své rodině.

Abstrakt

Bakalářská práce je zaměřena na tvorbu 3D FPS hry, s implementací rozdílných metod střel, a to hitscanu a projektilu. Nejprve bylo nutné vybrat herní framework pro tvorbu 3D videohry a modelovací nástroj pro tvorbu herních assetů. Výběr herního frameworku byl realizován nejprve teoretickou rešerší s následným praktickým otestováním v podobě demonstrativní videohry. Koncept herního děje, byl vytvořena na základě filmových postav, podle kterých bylo následně připraveno prostředí s magickou tematikou. Při tvorbě herní mapy bylo využito procedurální generování, s využitím Perlinova šumu. Pro přírodně vypadající rozmístění stromů, kamenů a dalších předmětů na mapě, byla využito funkce rozmísťování objektů, podle vytvořené množstevní mapy. Tvorba herních modelů probíhala v modelovacím nástroji Blender, kde bylo vymodelováno celkem 104 herních modelů. Kompletní implementace celé videohry probíhala ve vybraném herním enginu. Implementovány byly animace, různé herní systémy, nastavení osvětlení, metodiky střely, vizuální a zvukové efekty, umělá inteligence nepřátel a herní úkoly.

Klíčová slova:

Unity, Hitscan, Projektil, 3D FPS, Modelování

Abstract

Bachelors thesis focuses on creation of 3D FPS videogame, with implementation of varied shooting methods, such as hitscan and projectile. First of all, it was necessary to choose the game engine and modelling toolset for creation of game assets. Choosing of game engine was realized by theoretic research and following practical testing with the use of demonstrative videogame. The concept of game plot was based on film characters, according to which environment with magical atmosphere was developed. While creating game map there was used the procedural generation, with the use of Perlin noise. To scatter naturally trees, stones and other objects on the game map, there has been used function of placement of objects according to weight map. Creation of game items was realized in toolset called Blender, where 104 game items were created. Whole implementation of the videogame was developed in the chosen game engine. The implementation consists of animation, varied game systems, light settings, shooting methods, visual and sound effects, artificial intelligence of enemies and game quests.

Keywords:

Unity, Hitscan, Projectile, 3D FPS, Object modeling

Obsah

1	Úvod	12
2	Vývoj videoher	13
2.1	Historie	13
2.2	Rozšířenost herního průmyslu	14
2.2.1	Český herní průmysl	15
2.2.2	Tituly od českých vývojářů	15
2.3	Herní framework	15
2.3.1	Oblíbené herní frameworky	16
2.3.2	Srovnání Unreal a Unity frameworků	16
2.3.3	Demonstrativní hra	16
2.4	Unity	16
2.4.1	Subsystemy	16
2.5	Trojrozměrná grafika	19
2.6	Modelování	19
2.6.1	Vývojové prostředí	19
2.6.2	Objemová reprezentace	19
2.6.3	Hraniční reprezentace	20
2.7	Textura	20
2.8	Animace	20
2.9	Grafické styly	21
2.9.1	Low poly	21
2.9.2	Stylizovaný	22
3	Tvorba hry	23
3.1	Návrh prostředí	23
3.1.1	Inspirace	23
3.1.2	Tvorba 3D assetů	23
3.1.3	Herní assety	24
3.2	Tvorba postav	24
3.2.1	Hráč	25
3.2.2	Postava Baby Jagy	25
3.2.3	Živé stromy	26
3.3	Animace postav	26
3.3.1	Hráč	26

3.3.2	Postava Baby Jagy	26
3.3.3	Živé stromy	26
3.4	Herní mapa	27
3.4.1	Perlinův šum	27
3.5	Návrh herního děje	28
3.5.1	Herní prostředí	28
3.5.2	Herní úkoly	28
4	Implementace	29
4.1	Osvětlení scény	29
4.1.1	Baked lightmaps	30
4.1.2	Mlha	30
4.2	Tvorba grafické pipeline	30
4.3	Kamera	30
4.4	UI Systém	30
4.5	Manažer úkolů	31
4.6	Manažer skóre	31
4.6.1	Systém událostí	31
4.7	Hráč	32
4.7.1	Ovládání	32
4.7.2	Animace	32
4.8	Záporné postavy	32
4.8.1	Umělá inteligence	33
4.8.2	Deformace kostry	33
4.8.3	Rozdělení záporných postav	33
4.9	Zvukové efekty	33
4.9.1	Weberův–Fechnerův zákon	34
4.10	Vizuální efekty	34
4.10.1	Částicový systém	35
4.10.2	Shader	35
4.11	Herní logika	36
4.11.1	Možnosti hráče	36
4.12	Kolizní systém	37
4.12.1	Pokročilá detekce kolizí	37
4.12.2	Interaktivní předměty	37
4.13	HP systém	37
4.13.1	Herní postavy	38
4.14	Hitscan	38
4.14.1	Výhody	38
4.14.2	Nevýhody	39
4.14.3	Raycast	39
4.14.4	Kontrola kolize	39
4.15	Projektíl	39
4.15.1	Výhody	40
4.15.2	Nevýhody	40

4.15.3	Fyzické vlastnosti	40
4.15.4	Kontrola kolize	40
5	Kritické zhodnocení	41
5.0.1	Srovnání	41
5.1	Optimalizace	41
5.1.1	Grafický styl	41
5.1.2	Herní prostředí a děj	42
5.1.3	Herní logika	42
5.1.4	Metoda střelby	43
5.1.5	UI Systém	43
5.2	Kompatibilita	43
5.3	Rozšíření	43
6	Závěr	44
	Použitá literatura	46
7	Seznam příloh	47
7.1	GitHub odkaz	47
7.2	Odkaz na YouTube	47
7.3	Odkaz na YouTube	47
7.4	Příběh	47
7.5	Dokumentace na ovládání videohry	47

Seznam zkratk

- FPS - First-person shooter
- MIT - Massachusettský technologický institut
- AAAA - Videohry vydávané středně velkým nebo velkým vydavatelem
- GPU - Grafický procesor
- GLS - OpenGL Shading Language
- CG - Programovací jazyk pro psaní shaderů
- HLSL - High Level Shader Language
- VR - Virtuální realita
- AR - Rozšířená realita
- AI - Umělá inteligence
- NPC - Nehratelná postava
- WASD - Klávesnice W, A, S, a D
- HP - Hit-Point
- CCD - Continuous collision detection

1 Úvod

Bakalářská práce je zaměřena na tvorbu 3D FPS hry neboli hry, kde hráč hraje z pohledu první osoby. Nejprve bylo zapotřebí vybrat herní framework, ve kterém se vhodně vypracuje požadovaný prototyp videohry. Výběr herního frameworku byl realizován za pomoci krátké demonstrativní hry. Pro srovnání byly vybrány pouze dva frameworky, nejvýše postavené v herním žebříčku podle oblíbenosti samotných indie vývojářů¹ [1]. Po výběru následovalo vytvoření herního děje, který byl inspirován světovými filmy. Po upřesnění děje bylo možné vymodelovat herní předměty, jako je les s magickou tematikou, herní postavy a jejich animace. Při animaci postav bylo zapotřebí nejprve každé postavě vytvořit její kostru, která následně ulehčila tvorbu animace dané postavy. Postavy vznikly spojením několika geometrických objektu, které vizuálně působí jako člověk a následně byla vytvořena jejich kostra. Propojení geometrických objektů s kostrou umožnilo předávat pozici, rotaci a velikost od přiřazené kosti. Byly animovány akce jako nečinný postoj, chůze a útok. Dále také byla vymodelována hráčská postava, jelikož se jedná o FPS videohru, což znamená, že hráč nahlíží na videohru, jako kdyby byl součástí videohry, tedy spodní část těla, trup a hlava není viditelná pro hráče samotného, tudíž nebylo zapotřebí je modelovat. Hráč je reprezentován jako vymodelované pohyblivé ruce, tudíž kostra jsou pouze ruce, které jsou detailněji propracovány v oblasti zápěstí, na rozdíl od ostatních herních postav, jako jsou záporní hrdinové a další. Na závěr bylo zapotřebí provést implementaci hry, spojit herní objekty a herní logiku do finální podoby, ve které je možné si videohru zahrát z uživatelské perspektivy.

¹indie vývojář je nezávislý vývojář, který většinou pracuje samostatně

2 Vývoj videoher

Vývoj videohry je náročný proces z pohledu finančního i pracovního vytížení. Skládá se z několika podprocesů. Nejprve je nutné sestavit tzv. design dokument, který obsahuje detailní popis hry, tedy děj, grafický styl videohry, herní úkoly, vlastnosti předmětů a postav[2]. Tento dokument obvykle obsahuje přes tisíce stran textu a je sestaven herním návrhářem. Po sestavení konceptu následuje představení vydavatelské společnosti, která následně schválí vývoj hry a zafinancuje (v případě že se jedná o tzv. indie herní studia složena z jednoho nebo několika vývojářů, kteří vyvíjejí na vlastní náklady). Po schválení konceptu následuje vývoj, který nejčastěji trvá od jednoho roku až po několik let[2]. Proces vývoje obsahuje tvorbu modelů, textur, animací, audia a skriptů. V takovémto procesu, zejména pro větší herní tituly je zapotřebí si stanovit milníky, neboli dílčí cíle a termíny dokončení. V období kdy se tyto milníky blíží k termínu splnění, řada klíčových členů týmu a samotné týmy přechází do módu pracovních přesčasů, aby efektivně vyřešili všechny nedostatky. Další důležité a pracovně náročné období nastává, když se blíží vydání samotné hry, tehdy pracují všichni vývojáři naplno a skupiny testerů se snaží najít chyby, které by ovlivnily hratelnost videohry. [3]

2.1 Historie

Historie vývoje videoher se píše od roku 1947. Nebyly to videohry, jak je známe dnes, jelikož tehdejší technologie neumožňovaly vykreslit herní prvky, ale odpovídaly definici videohry, splňovaly interakci mezi hráčem a strojem. Hra vyvinutá Thomasem Goldsmithem a Estlem Mannem byla kreslena na průhlednou fólii a obsahovala celkem osm katodových trubice, které se využívaly k simulaci střely odpálené na cíl. Hra disponovala tlačítky umožňujícími měnit dráhu a rychlost střely. V roce 1958 William Higinbotham vytvořil hru, která měla zabavit návštěvníky v Národní laboratoři v New Yorku, videohru nazval „Tennis for Two” (Tenis pro dva). Mnozí autora považují za zakladatele moderního herního průmyslu. Videohra obsahovala zjednodušený tenisový kurt z pohledu z boku. Míček podléhal gravitaci a odehrával se přes síť, hráč ovládal trajektorii a rychlost odpálení míčku pomocí ovladačů s tlačítky.[4]

Další pokrok přišel v 60. letech, kdy byly hry vyvíjeny programátory jednotlivci v jejich volném čase. Tyto hry běžely na univerzitních serverech ve Spojených státech, ale kvůli omezenosti tehdejšího hardwaru bylo takovýchto her velmi málo.

Jedna ze známějších her celého 60. desetiletí minulého století byla hra „SpaceWar”, vytvořena skupinou studentů v roce 1961. Šlo o vesmírnou válku, kde se proti sobě postavili dva hráči. Každý hráč měl možnost ovládat vesmírnou loď s možností střelby. Cílem hry bylo sestřelit nepřátelskou loď. O rok později byla hra uvedena na americkém MIT, stala se první vydanou a rozšířenou hrou. V této dekádě byly také položeny základy pro významnou část herního odvětví a tím jsou herní konzole. Základem byla hra Chase(Honička) od Ralpha Beara, která byla volně dostupná na běžném televizním přijímači. Později když Bear pokračoval ve vývoji, v roce 1968 zkompletoval prototyp herní konzole, na které bylo možné si zahrát několik různých her. Bear v 70. letech prodal svůj prototyp firmě Magnavox, která ho v květnu roku 1972 vydala po jménem Odyssey, první herní konzole toho typu na světě. V 70. letech také vznikla jedna ze známějších her i v současném světě a to byla hra Pong. Byla vydána firmou Atari, která byla tak úspěšná, že v USA rozpoutala automato-vou mánií.[4]

Další desetiletí docházelo k rozmachu herního vývoje. Vyvíjely se nové hry a také herní konzole jako Odyssey 2. Větší pokrok byl dosažen v 90. letech, kdy Nintendo přišlo se svojí herní konzolí zvanou SNES (Super Nintendo Entertainment System) nebo také Sony přišla s dnes tak známou ságou herních konzolí Playstation. Společně s konzolemi začaly vycházet taky známější hry dnešní doby jako Doom, série Grand Theft Auto a Need for Speed. Tento rozvoj herního průmysl pokračoval i na počátku druhého tisíciletí, kdy společnost Sony přišla s další generací Playstation. Na trhu se také objevil její přímý konkurent konzole Xbox od firmy Microsoft. Dále se také objevovaly nové hry jako Call of Duty, Battlefield a další herní tituly, které jsou dodnes známé.[4]

2.2 Rozšířenost herního průmyslu

Videoherní průmysl zahrnuje několik profesí jako jsou grafik, herní producent, vývojář a další. Mezi největší herní giganty, kteří přicházejí na trh s těmi nejhranějšími tituly jako je Mario, Pokémon, Counter Strike, Grand Theft Auto nebo série FIFA, patří Nintendo, Valve, EA Sports nebo Rockstar Games. Počet zaměstnanců u těchto studií je v řádu tisíců. Herní průmysl expanduje a to v jakékoliv zemi, vznikají nová herní studia nebo ta stávající nabírají více vývojářů. Ačkoliv herní průmysl má velký potenciál a každý rok nabírá více a více vývojářů, jeho výuka stagnuje.[3] Předměty na vývoj videoher na střední škole stěží najdeme, pouze pár vysokých škol se tímto odvětvím zabývá. [5]

2.2.1 Český herní průmysl

Česká republika nezaostává a ani tady nebyl pominut trend herního vývoje. Herní průmysl patří k jednomu z nejrychleji rostoucích průmyslových odvětví v České republice. Vznikla zde řada úspěšných studií na vývoj videoher, která vydala tituly známé po celém světě. K neznámějším herním studiím v České republice patří 2K Czech a Bohemia Interactiv. [6]

2.2.2 Tituly od českých vývojářů

Herní studia v České republice produkují ročně několik videoher, v řádu jednotek. Podobně jako ve světě vznikají a vydávají se videohry denně, tak v České republice vznikají videohry, sice v menším měřítku. Většina vydaných videoher jsou obyčejná, rychle upadne zájem podobně jako ve světě. Avšak existují v historii videohry, které dobyly nejen tuzemský trh, ale také ten světový. Mezi úspěšné české herní tituly se řadí herní série Mafie, Arma, Beat Saber nebo Euro Truck Simulátor 2. Tyto videohry se staly nedílnou součástí historie videoher. Díky takovýmto herním titulům, které jsou ve všech směrech odladěné a dokáží přilákat potenciální hráče, čeští herní vývojáři nezaostávají za těmi světovými.

2.3 Herní framework

Herní framework je nástroj určený pro herního vývojáře, který dokáže zlevnit a zrychlit vývoj nových videoher. Obsahuje subsystemy (2.3) usnadňující práci vývojáři, jako jsou vykreslovací, fyzikální, zvukový podsystém a další funkce. Jednotlivci nebo malá herní studia využívají licencování cizích frameworků, naopak pro větší herní studia je výhodnější si vyvinout vlastní herní framework, který si upravují na míru podle vyvíjené hry. [7]

Subsystemy herního frameworku

- vykreslovací podsystém,
- fyzikální podsystém, který simuluje fyzikální zákony,
- zvukový podsystém, dekomprimuje a přehrává zvukové efekty,
- podpora ukládání a nahrávání dat pro videohry,
- podpora různých platforem a jednotlivých ovládacích zařízení,
- podpora síťové komunikace,
- podpora skriptování neboli Visual Scripting,
- podpora umělé inteligence, řešení spíše obecných problémů jako hledání, nejkratší cesty mezi dvěma body.

2.3.1 Oblíbené herní frameworky

Mezi oblíbené herní frameworky využívané jednotlivci nebo malými skupinami herních vývojářů patří Unreal, Unity, GameMaker a další. Jak už bylo zmíněno, velká herní studia si vytváří své vlastní herní frameworky, které nepublikují.

2.3.2 Srovnání Unreal a Unity frameworků

Pro vypracování bakalářské práce bylo důležité provést výběr vhodného herního frameworku. Srovnání bylo provedeno prakticky i teoreticky, z oblíbených herních frameworků pro indie vývojáře. Pro srovnání byly vybrány dva herní frameworky a to přesně Unreal a Unity, ve kterých byl vytvořen jednoduchý herní prototyp a následně porovnán výsledný dojem. V úvahu byly brány i aspekty, jako je vizuální grafické vykreslení, předešlé zkušenosti a časová náročnost tvorby. Následně byl vybrán herní framework Unity na základě malé demonstrativní hry implementované v obou editorech.[8]

2.3.3 Demonstrativní hra

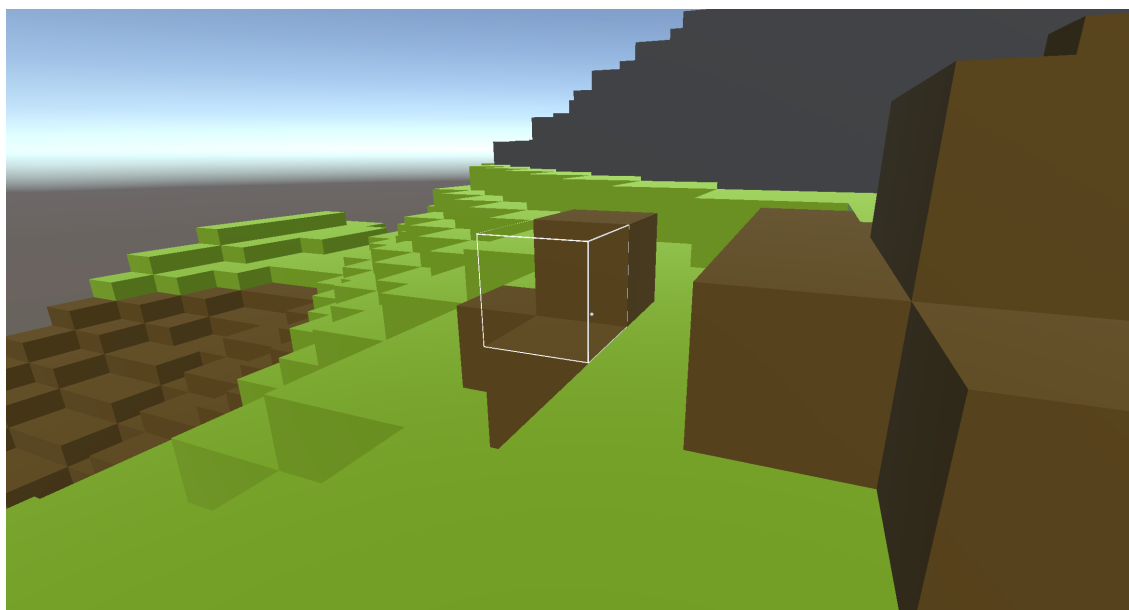
Byla vytvořena krátká demonstrativní hra. Nejprve byla realizována v herním frameworku Unity, především kvůli zkušenosti tvůrce projektu. Demonstrativní hra byla jednodušší forma připomínající již vytvořenou videohru Minecraft umožňující generování herního světa a stavění a demolice čtyř předdefinovaných bloků. V herním frameworku Unity byla videohra realizována za čtyři hodiny, zatímco v Unreal frameworku se za stejný časový úsek, kvůli neznalosti a špatné orientaci v herním frameworku, realizovalo pouze vytvoření čtyř materiálů aplikovaných na bloky a částečné generování světa. Byl vybrán herní framework Unity, především kvůli rychlejší orientaci a lepšímu konečnému výsledku.

2.4 Unity

Herní framework od společnosti Unity Technologies, je jeden z nejpoužívanějších herních frameworků na světě. Historicky byl poprvé představen na celosvětové konferenci Applu v roce 2005. První verze byla podporována pouze na operačním systému MacOS. Od počátku představení v roce 2005 se do současné podoby rozšířil o více než 15 platforem. Unity poskytuje možnosti vývoje pro 2D i 3D hry libovolného zaměření a nebo žánru. Unity výhradně podporuje tvorbu skriptů v jazyce C-sharp.[9]

2.4.1 Subsystémy

Subsystém je framework, který se stará o fyziku, vykreslování, ukládání dat, vizuální a zvukové efekty. Podobně jako ostatní herní frameworky, je tento založen na vývoji v C++, kde je podpora knihoven jako GLFW, GL a podobně. Unity obsahuje následující subsystémy.



Obrázek 2.1: Demonstrativní hra (Minecraft) v Unity

Subsystém vykreslování

Subsystém zajišťuje vykreslování (renderování). Renderování je tvorba reálného obrazu na základě počítačového modelu. Rendering obsahuje v závislosti na softwaru mnoho parametrů a nastavení, kterými lze ovlivnit konečný vzhled scény. Unity poskytuje pokročilejším verzím kromě základní, možnost ovládání renderování pomocí skriptů (funkcionalita Scriptable rendering pipeline), díky které je možné vytvářet realistickou grafiku.

Vykreslovací kanál

Model, který popisuje, jaké kroky musí grafický systém provést k vykreslení 2D a 3D obrazovky. Každý 3D model je zobrazován do 2D pohledu. Unity disponuje několika režimy jako jsou SRP, URP a HDRP.

The Scriptable Render Pipeline (SRP) je pipeline, která umožňuje vývojáři kontrolovat vykreslování za pomoci C-sharp skriptu. Na této bázi stojí herní režimy HDRP a URP.

The Universal Render Pipeline (URP) je odvozena od SRP, navíc umožňuje rychle a jednoduše optimalizovat grafický vjem.

The High Definition Render Pipeline (HDRP) je odvozena od SRP, která je navíc upravena pro tvorbu herních titulů řady AAAA, což podporuje nejnáročnější grafické vykreslování. HDRP využívá fyzikální osvětlování a požaduje výkonný GPU hardware.

Materiály Materiál definuje vykreslování povrchu objektu včetně odkazů na textury. Materiál se odkazuje na shader objektu, také obsahuje informace o barvě a textuře. Dalším důležitou vlastností materiálu je jeho přenositelnost, neboť framework Unity podporuje několik grafický pipelines.

Shadery Shader je program obsluhující jednotlivé části programovatelného grafického řetězce. Využívá specifické programové jazyky, jako jsou GLSL, CG a HLSL. Shader je použitelný na vykreslování reálných objektů, jako je kámen, sklo a další. Unity poskytuje navíc fyzický shading, simulace interakce mezi materiálem a světlem.

Fyzikální subsystém

Fyzikální podsystém zaručuje základní simulaci fyzikálních zákonů jako gravitace, pohyb těles, působení síly a dalších. Aby bylo možné v Unity přidat objekty, na které působí fyzikální zákony, tak se na tento objekt přidá Rigidbody, které přidává gravitaci objektu. Další fyzikální možnosti jsou například kolizní boxy, které umožňují detekovat kolizi. Dále také ragdoll, která je zahrnuta v této práci. Ragdoll funguje jako kostra, vytvoří se instance síly působící na objekty zásahu podle vyhodnocení kolizního systému. Unity disponuje mnoho dalšími fyzikálními funkcemi.

Skriptování

Skriptování je důležitá část každého vývoje, dává smysl herní logice a mechanice. Unity podporuje skriptování v jazyce C-sharp, skript zajišťuje funkcionalitu samotné videohry. Unity také nabízí možnost řízení veškerých vestavěných funkcí pomocí skriptu, například posun objektů, rozměry, animace, zvukové efekty, kolizní systémy a další. V dnešní době je přidána možnost psát skripty tzv. bez skriptování, jednoduše pomocí úkonů. Uživateli je poskytnuto grafické prostředí, kde si sestavuje předepsané skripty.

Zvukový subsystém

Zvukový podsystém je nedílnou součástí tvorby prvotřídní videohry, tento podsystém disponuje funkcemi načtení, dekomprimace a přehrání různých stop. Unity také nabízí možnost audio mixéru, tedy zkombinování více zvukových zdrojů. Podsystém také nabízí další pokročilé funkce jako ozvěna, změna výšek tónů a další.

Další komponenty

Unity framework disponuje multithreadingem, správou paměti a síťováním, které zajišťuje nejen internetovou komunikaci (multiplayer mode), ale i animace, umělou inteligenci a další.

Podpora platformem

Unity podporuje více než 25 platformem, mezi které patří i nejrozšířenější operační systémy jako Microsoft, Linux a MacOS. Unity má také úspěch i na jiných platformách, dominuje na trhu s mobilními videohrami.[10] Více než polovina videoher na trhu je vytvořena přes tento framework, také podobně ovládá VR a AR trh.

2.5 Trojrozměrná grafika

Trojrozměrná grafika je speciální část počítačové grafiky pracující s trojrozměrnými objekty, které jsou převedeny do dvou dimenzionálního zobrazení. Mezi využití počítačové grafiky patří modelování objektů a animace, využitelné ve filmech, počítačových videohrách nebo průmyslových simulacích. [11]

2.6 Modelování

Modelování je způsob tvarování a vytváření 3D modelu, který je možný reprezentovat několika způsoby, nejčastější je hraniční reprezentace a méně častá objemová reprezentace. Modely mohou být vytvořeny na počítači člověkem pomocí modelovacího nástroje, podle dat získaných měřicím přístrojem z reálného světa nebo na základě počítačové simulace.[11]

2.6.1 Vývojové prostředí

Blender je volně dostupný nástroj pro modelování a vykreslování trojrozměrné grafiky a také na tvorbu animací. Blender vykresluje vlastní rozhraní pomocí knihovny OpenGL, která díky své rozšířenosti zaručuje snadnou přenositelnost a také hardwarovou akceleraci (výpočtové části úlohy jsou vykonány hardwarem, což zvyšuje rychlost vykonání operace).

2.6.2 Objemová reprezentace

Objemová reprezentace se využívá pro tvorbu 3D objektů, u kterých jsou důležité jejich fyzikální vlastnosti. Tato reprezentace shledává využití v medicínském a technickém odvětví, jako 3D scanner nebo v lékařství tomograf. Reprezentace se skládá z geometrických primitiv (kvádr, koule, kužel apod.) a jejich kombinacích. Komplexní tvary jsou vyvářeny pomocí množinových operátorů jako jsou průniky, sjednocení, rozdíly či doplňky mezi základními objemovými geometriemi. Systém odvození konkrétního modelu je zapsán pomocí tzv. CSG stromu (Constructive Solid Geometry). [12]

2.6.3 Hraniční reprezentace

Hraniční objekt je mnohostěn s pevnými hranami, vrcholy a stěnami. Využití nachází ve filmech a videohrách. Objemově reprezentované objekty se při vykreslování převádějí na hraniční modely, kvůli menší náročnosti na vykreslování.

2.7 Textura

Textura poskytuje detailnější renderování objektu, jelikož objektová síť podporuje pouze hrubou aproximaci tvaru. Textura je standardní bitmapový obrázek aplikovaný na povrch sítě. V jednodušší formě se textury využívají pro obarvení modelu. Objekty mohou obsahovat i více vrstev textur například lesklost, průhlednost a další. Při mapování textur se ke každému vrcholu definovaného souřadnicemi X, Y, Z přiřadí koordináty U a V určující umístění textury (proces UV mapování).

2.8 Animace

Animace definuje nejen pohyb objektů, ale i osvětlení objektu, úhel pohledu kamery a další aspekty. Jednodušší možnou animací je animace keyframing. Je založená na principu 2D animace, tedy definování klíčových pozic objektu, kde se mezi jednotlivými pozicemi vytváří přechod (2.2)¹. Při animaci postav se nejčastěji využívá kostra postavy, kde jednotlivé části kostry mají připnuté objekty. Animuje se jenom kostra a objekty (noha, ruka, hlava a atd.) se hýbají díky závislosti na kostře. Existuje mnoho dalších technik vývoje animací, mezi které patří také tzv. Motion capture vytvářené zachycením pohybu skutečného objektu. [13]



Obrázek 2.2: Schéma animace

¹proložení přímkou dva sousední body

2.9 Grafické styly

Grafický styl použitý v bakalářské práci není tak jednoduché určit, jelikož většina objektů je vytvořena na pomezí grafického stylu low poly (modely vytvořené s malým počtem polygonů) a stylized styl (typ low poly s větším počtem polygonů). Objekty vymodelované za účelem zakomponování do scény obsahují poměrně vysoký počet polygonů s vizuálním vjemem, ale ne tolik, aby je bylo možné zařadit do stylového grafického stylu. V případě této bakalářské práce se herní scéna skládá ze dvou podobných grafických stylů.

2.9.1 Low poly

Low poly je jednodušší grafický styl, který kvůli svému malému množství polygonů je optimální na vykreslování. 3D objekty ve scéně jsou tvořeny jednoduchými geometrickými tvary. Vizuální vjem low poly stylu je na nízké realistické úrovni, detailně jsou vymodelovány jen ty hlavní prvky, pouze pro rozeznání předmětu. Styl je využíván pro zařízení, která nemají dostatečné zastoupení grafické karty, tj. mobilní zařízení a podobně.[14]



Obrázek 2.3: Low poly model

2.9.2 Stylizovaný

Stylizovaný styl je přechodem mezi Low poly a realistickým stylem. Klade větší důraz na detail, ale přitom si drží relativně malý počet polygonů. Dojem assetů vymodelovaných v tomto stylu je detailně nakreslený model, tedy hlavní prvky modelovaného předmětu jsou zachovány. Vizuální dojem se přiklání k realistickému stylu, více než k Low poly, přitom se jedná o modely spíše kreslené. Pracnost modelování v tomto stylu se podobá spíše realistickému modelování, klade se zde důraz na fantazii tvůrce, která přetváří reálný předmět.



Obrázek 2.4: Stylizovaný Low poly model

3 Tvorba hry

Tvorba hry je časově náročným bodem, nejen v této práci, ale i při vývoji komerční videohry. V tomto bodu se modelují objekty použité ve hře, volí se grafický styl celé videohry a také se určuje, jaký bude herní cíl. Dále zahrnuje návrh zajímavého herního děje, tvorba herních úkolů a vedlejších úkolů.

3.1 Návrh prostředí

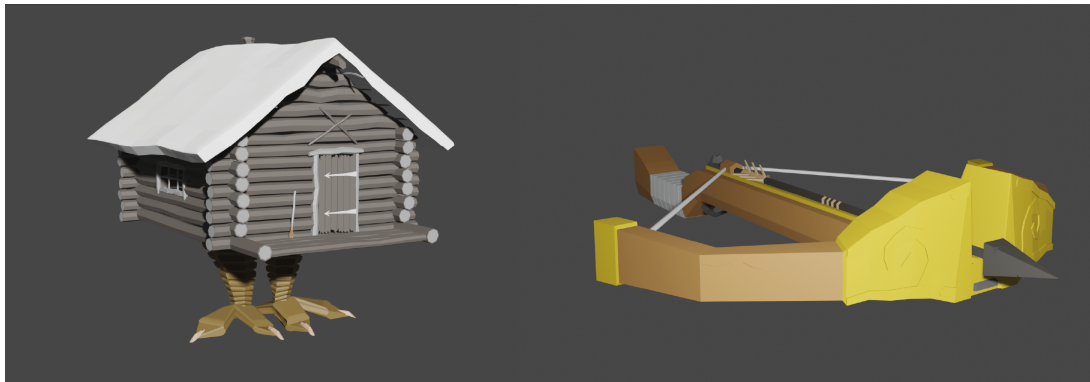
Návrh herního prostředí je jedním z důležitých bodů této bakalářské práce. Vizuální dojem z videohry je důležitý, jelikož první dojem rozhodne, zda se potenciální zájemce rozhodne videohru vyzkoušet. Herní prostředí je také jedním z důležitých aspektů, které rozhodují o přetrvávajícím zájmu u uživatele, který si hru už vyzkoušel.

3.1.1 Inspirace

Herní prostředí hráči navodí atmosféru magického lesa, především zářící předměty, jako houby, řeka a také navigační kameny podél cesty. Magickou atmosféru vytváří nejen zářící měsíc, ale také záporné pohádkové postavy. Herní prostředí je inspirováno pohádkovými postavami. Jelikož se jedná o projekt, na kterém obvykle pracuje v řádu let několik set programátorů, nebylo možné vytvořit videohru, která by se rovnala výsledku tak početného týmu vývojářů. Tento prototyp videohry obsahuje pouze jednu část z několika možných. Část neboli úsek hry zaměřený na film Mrazík, přesněji postavy z něj jako jsou postavy Baby Jagy a živé stromy bránící srub na kuřích nožkách.

3.1.2 Tvorba 3D assetů

Tvorba trojrozměrných objektů, proces modelování jednotlivých herních prvků, je časově náročný a v této práci jeden z hlavních bodů. Tvorba probíhala transformací jednotlivých vrcholů objektu na pozice, nebo přidáváním jednotlivých vrcholů, které se následně spojily stěnami a hranami. Následným přidáním textur na jednotlivé stěny. V tomto projektu se jednalo pouze o jednobarevné textury, tedy každá stěna byla obarvena jen jednou barvou.



Obrázek 3.1: Herní model kuše a chaloupky

3.1.3 Herní assety

Velké množství herních assetů bylo vymodelováno výhradně pro použití v tomto projektu, kompletní seznam vytvořených assetů a jejich tvůrců (3.1). V rámci vypracování práce se vymodelovalo úctyhodných 104 herních assetů a to od těch jednodušších až po ty náročnější, mezi které například patřilo modelování postav.

Tabulka 3.1: Tabulka herní assetů a jejich tvůrců

Herní model	Autor
Polyverse Sky	Boxophobic[15]
Zvukové efekty	Epidemic Sounds[16]
Herní mapa Postavy Stromy Chaloupka Interaktivní předměty Fauna	Adrian Bartoň

3.2 Tvorba postav

Tvorba herních postav je komplexní proces, který byl časově náročnější. Zejména tvorba záporných postav jako byla postava Baby Jagy nebo živého stromu. Nešlo jen o pouhé vymodelování hlavních rysů, ale také o vytvoření kostry, která následně ulehčila tvorbu animace. Na vytvořenou kostru, přesněji na její kosti, se přiřadí jednotlivé části postavy, které následně dědí od kostry její škálovatelnost, pozici a rotaci.

3.2.1 Hráč

First-person shooter (hra z první osoby) - hráč vidí pouze ruce ovládané postavy. Důležité bylo vymodelovat ruce a kostru ovládající pouze horní část končetin. Modelování je nízkopolygonové, tudíž se zaměřením na malý počet polygonů. Horní končetiny jsou modelovány jednoduchými geometrickými tvary (hranoly a kvádry). Náročnost tvorby se zmírnila se zaměřením pouze na horní část končetin, avšak musel se klást větší důraz na detail v oblasti zápěstí až po konečky prstů. Tento segment je hráči zobrazován detailně. Kostra se skládá z 18 kostí na každé paži. Své využití našla i inverzní kinematika, která zajišťuje při pohybu dlaní pohyb lokte. Nebo také předávání rotace, kde rotující kost, v tomto případě článek prstu, předává své nadřazené kosti svojí částečnou rotaci. Kostra byla vytvořena na principu fungování lidských kostí, dalo by se říci že simuluje pohyb kostry člověka. Hráčská kostra také obsahuje vnořenou kostru obsluhující chování zbraně.



Obrázek 3.2: Postava Baby Jagy a živého stromu

3.2.2 Postava Baby Jagy

Komplexním modelovacím procesem prošla postava Baby Jagy, neboli zlé ježibaby. Časově náročnější postava z celé hry se modelovala v čistém čase 57 hodin (tvorba kontur obličeje a oblečení postavy) a její kostra zabrala několik dalších jednotek hodin. Kostra obsahující 25 kostí není nijak detailně vytvořena vzhledem k malému množství požadovaných animací, které postava obsluhuje.

3.2.3 Živé stromy

Záporná postava chráníci chaloupku postavy Baby Jagy. Model stromu je sestaven z pěti šestiúhelníků, které jsou obrostlé větvemi. Kostra má strukturu podobnou té lidské, avšak obsahuje pouze 13 kostí a to pouze ty hlavnější.

3.3 Animace postav

Animace postav spočívá v definování klíčových vlastností kostry, mezi kterými se automaticky vytváří přechod. 3D objekty nabízí tvorbu kostry, která tvorbu animací usnadňuje. U některých postav byl realizován jednoduchý systém založený na rychlosti postavy. Systém který z rychlosti určuje aplikující se animaci.

3.3.1 Hráč

Hráč disponuje animací v klidu a pohybu. Jednoduchý systém, který na základě rychlosti pohybu hráče určuje, která animace se má spustit. Tyto animace běží cyklicky. Postava také obsahuje animace spojené se střelnou zbraní, jako střelba, zamíření a nabíjení zbraně. Animace zamíření a střelba ze zbraně je řízena na bázi skriptu, kde se vyhodnocují stisknutí levého a pravého tlačítka myši. Animace nabití je vykonána řetězově po animaci střelby.

3.3.2 Postava Baby Jagy

Postava má celkem dvě animace. První animace běžící ve smyčce je rozhlížejí animace, kdy postava postoupí o krok dopředu rozhlédne se a vrátí se zpět. Další animace vyžaduje podnět k aktivaci. Pokud se hráč přiblíží na určitou vzdálenost, postava Baby Jagy začne střílet projektily, ohnivé koule ubírající životy. Tato animace má konstantní palebnou dobu, tj. doba, kdy se animace tzv. obnovuje. Postava Baby Jagy se mezi střelami vrací ke své cyklicky se opakující animaci. Postava jako model obsahuje také bod střely, neboli bod instance střely.

3.3.3 Živé stromy

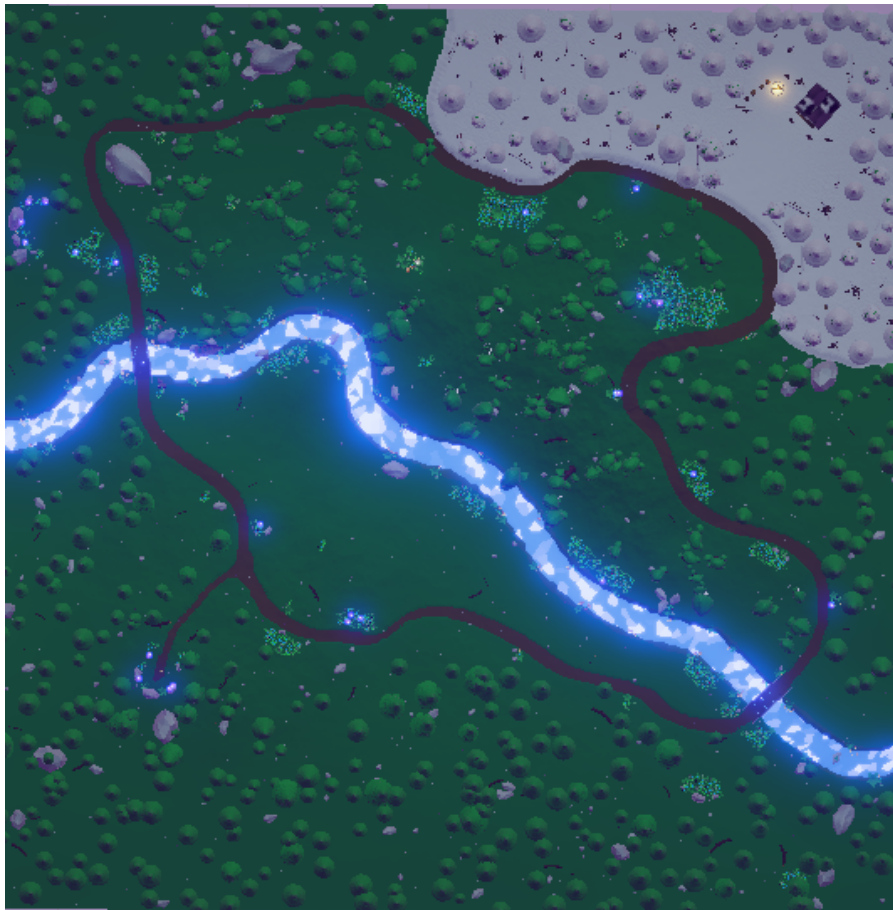
Animace postavy se odvíjí od její rychlosti pohybu. Jelikož se jedná o pohybující se postavu, je sledována její rychlost, podle které se potom určuje, jaká animace bude přehrána (například pokud je rychlost poloviční z maximální stanovené, tak se animace prolínají, obě animace proloženy navzájem). Postava dále obsahuje animaci útoku, která vyžaduje, aby hráč byl v určitém vzdálenostním rozmezí od postavy. Pokud je to splněno, postava maximální rychlostí pronásleduje hráče. Animace obsahuje čas, kdy se útok obnovuje, tj. čas mezi jednotlivými údery.

3.4 Herní mapa

Herní mapa je vytvořena za použití vestavěné funkce použitého vývojového prostředí, funkce generování světa Perlinův šum. Byl vygenerován svět podle příslušných parametrů s následným upravením (lehké zjemnění kontur a také vytvoření koryta řeky a pěší cesty). Modelovací nástroj disponuje také náhodným umístěním objektů, podle vytvořené mapy hustoty. Tento nástroj byl využit pro náhodné umístění stromů a kamenů v daném velikostním rozmezí.

3.4.1 Perlinův šum

Metoda funguje na součtu stejné šumové funkce, avšak o různé intenzitě a v různém měřítku. Pro takovou šumovou funkci se často používá náhodný, deterministický generátor. Výsledkem součtu šumových funkcí je šum, kterým je možné napodobit rozličné přírodní věci, nejčastěji oblaka, vodu, dřevěné materiály a mramor.[17]



Obrázek 3.3: Herní mapa

3.5 Návrh herního děje

Dějová linka obsahuje několik hlavních úkolů, které je možné splnit jednotlivě. Hráč má možnost také splnit vedlejší úkoly, jako je střelba na terče, které jsou umístěny na stromy po celé herní mapě. Splnění hlavních úkolů není nijak hodnoceno, ale pro další hráčův postup je důležité. Na druhou stranu jsou zde vedlejší úkoly, neboli tzv. „Easter eggs“, za které hráč dostává body podle toho, s jakým kruhem střela koliduje (na každém terči se počítá pouze první zásah, jakýkoliv další pokus se už nepočítá). Hlavní úkoly budou hráče posouvat k dalším a náročnějším úkolům a ty vedlejší jsou pouze na získání skóre, které se hráči zhodnotí až na konci hry. Jelikož práce pojednává o prototypu videohry, tak se v tomto projektu neřeší vyhodnocování ani posun k dalším hlavním úkolům, ale pouze dokončení jednoho hlavního úkolu a získání skóre.

3.5.1 Herní prostředí

Herní prostředí vychází z konceptu herního děje. Bylo potřeba vytvořit takové prostředí, aby předměty působily atmosféricky stejně a zároveň byly navigátorem ve videohře, jako například kameny symbolizující sílu provázenou podél cesty, nebo také houby s magickou tematikou vytvořené pro doplnění životů hráče.

3.5.2 Herní úkoly

Úkoly ve hře se rozdělují na hlavní a vedlejší (Easter eggs). Hlavními úkoly, kterými se tento prototyp zabývá je porážení zlé postavy Baby Jagy, která zaklela celý les do temnoty. Úkolem hráče je nejprve nelézt zimní biom, kde se nachází chaloupka na kuřích nožkách. Dále se musí zbavit živých stromů, které chrání chaloupku zlé ježibaby před hráčovým příchodem. Po dokončení předešlého úkolu, hráče čeká poslední úkol k dokončení úseku hry a tím je porazit hlavní zápornou postavu a rozbít kletbu skrytou v zářící hvězdě nad vchodovými dveřmi do chaloupky.



Obrázek 3.4: Dějová linka

4 Implementace

Dalším bodem byla implementace, proces vykonávaný ve vybraném herním enginu. Jednalo se o sestavení a přidání funkčnosti vymodelovaných objektů. Do implementace spadá tvorba skriptů, ve kterých byly vytvořeny herní systémy. Dále také využití vestavěných funkcí pro tvorbu animací, fyziky a dalších. Sekce dále popisuje jednoduchou umělou inteligenci nepřátel, pohyb hráče a další aspekty obsažené ve hře, jako tvorba efektů ať vizuálních nebo zvukových. Druhy osvětlení ve scéně, systémy použité ve videohře a vytvořené herní manažery.

4.1 Osvětlení scény

Herní scéna je osvětlena několika typy světel, které vrhají jemné stíny. Hlavní **směrové světlo**, vhodně využito jako svit měsíce, lze považovat za nekonečně vzdálený zdroj světla. Směrové světlo nemá žádnou identifikovatelnou pozici, tudíž se nedá spočítat vzdálenost k jednotlivým objektům, které jsou všechny osvětleny stejnou intenzitou. Nastavuje se pouze jeho rotace, neboli směr osvětlení.[18]

Dalším typem osvětlení jsou **bodová světla**, které na rozdíl od předešlého typu mají pevně danou pozici a rádius osvětlení. V závislosti na vzdálenosti od zdroje se intenzita osvětlení zmenšuje (intenzita světla je nepřímo úměrná druhé mocnině vzdálenosti od zdroje). Osvětlení je rozmístěno rovnoměrně do všech stran kolem světla. Bodové světlo je ve videohře použito jako simulace svitu plamenů ohně, nebo jako záře hvězdy nad vchodem do chaloupky.[18]

Posledním typem osvětlení jsou **emisní materiály**. Emise je vlastnost shaderu, který umožňuje statickým objektům ve scéně vyzařovat světlo. Pro emisní osvětlení neexistuje žádný rozsah osvětlení, ale intenzita opět klesá kvadraticky (objekty přijímající emisního osvětlení jsou označeny jako statické). Ve videohře je využito osvětlení na zářící kameny podél cesty, zářící řeku a houbu.[19]

4.1.1 Baked lightmaps

Lightmapping je proces předběžného výpočtu osvětlení ve scéně, využito při tvorbě objektů v reálném čase. Jelikož proces osvětlování je náročný, využívá se generování lightmapy, tedy uložení všech možných osvětlení objektů ve scéně do lightmapy, ze které se v průběhu videohry načítá pouze textura objektu. Světla v reálném čase lze překrývat a aditivně používat v kombinaci s lightmapou.

4.1.2 Mlha

Mlha je v herním světě využita za účelem zvýšení efektivity vykreslování. Díky výrazně omezené viditelnosti není potřeba vykreslovat předměty za mlhou a zároveň předměty v mlze mají menší kvalitu vykreslení. Funkcionalita spočívá v překrytí předmětu barvou s intenzitou závislou na vzdálenosti od kamery.

4.2 Tvorba grafické pipeline

Videohra využívá univerzální vykreslovací pipeline (URP). Součástí implementace je také tvorba vlastní pipeline. V rámci vykreslovací pipeline se nastavila kvalita (míra náročnosti GPU), post-processing, stíny a pokročilé nastavení. Kvalita vykreslování je přímo úměrná zvolenému grafickému stylu videohry.

4.3 Kamera

Scéna obsahuje pouze jednu kameru, která v průběhu každého cyklu dostává novou pozici podle pohybu hráče. Kamera tedy není pod stejným objektem jako hráč. Hráč má pouze bod na své postavě, do kterého se přepisuje pozice kamery. Přepis pozice kamery se vykonává jako poslední úkon v cyklu, jelikož kamera sleduje objekty, které by se mohly uprostřed cyklu přesunout. Kamera využívá post-processing, který zvyšuje kvalitu obrazu videohry. Pomocí post-processingu se dají tvořit efekty, jako plavání pod vodou, efekt koukání dalekohledem a další.

4.4 UI Systém

Framework Unity nabízí sadu nástrojů pro správu a vytvoření grafického uživatelského rozhraní. Uživatelské rozhraní je založeno na herním objektu Canvas, do kterého je možné přidávat UI komponenty jako text, obrázky, tlačítka a další. Komunikaci mezi moduly je zprostředkována pomocí systému událostí. Uživatelské rozhraní ve videohře je použito jako nápověda s ukazatelem vyhotovených úkolů (umístěna v levém horním rohu). V systému uživatelského rozhraní se hráči zobrazuje jeho skóre a počet aktuálních životů, případně nápověda pokud koliduje se zónou interaktivního předmětu.

4.5 Manažer úkolů

Manažer úkolů je samostatný herní objekt, který nemá fyzické zastoupení ve videohře. Správce herních úkolů, zajišťuje komunikaci mezi ukazatelem vyhotovených úkolů a kontroluje dokončení úkolů. Vstupním parametrem je UI komponenta ukazatele vyhotovených úkolů a referenci na objekt zapojený do HP systému. V případě dokončení úkolu se odešle požadavek do manažera úkolů a ten jej promítne hráči do ukazatele.

4.6 Manažer skóre

Manažer skóre je samostatný herní objekt, který nemá fyzické zastoupení ve videohře. Správce je vytvořený na bázi skriptu, počítá zasažené terče, které spadají do vedlejších úkolů. Vstupním parametrem je UI komponenta ukazující počet zasažených terčů z celkového množství. Každý terč má implementovaný skript, který v případě kolize se šípem vyhodnotí udělení bodů za zásah, deaktivuje kolizní těleso a pošle požadavek na vyhodnocení a připsání skóre hráči.



Obrázek 4.1: Uživatelské rozhraní

4.6.1 Systém událostí

Systém událostí je založený na odesílání požadavku na základě vstupu (klávesnice, dotyk nebo myš). Samotný systém je navržený jako správce, pro jeho plnou funkcionalitu je zapotřebí implementovat moduly systému, jako jsou tlačítka a podobné vstupy od uživatele. Systém spravuje vstupní moduly, raycasting, výběr herního objektu a aktualizaci vstupních modulů.

4.7 Hráč

Důležitá postava ve videohře, reprezentováno jako model rukou, které hrající uživatel vidí celou dobu hraní. Hráčská postava je ovládána za pomoci klávesnice a myši. Postava hráče je reprezentována v herní schématu jako pohybující se bod. Ten obsahuje vestavěný Character Controller, sloužící pro snadný pohyb hráče mezi kolizními objekty bez využití funkce působení fyzické síly (Rigid Body). Postava má implementovaný bod, na který se v každém cyklu videohry přepisuje pozice hlavní kamery ve scéně. Dále obsahuje skripty pro kontrolu pohybu hráče, kontrolu činností a také kolizní a HP systém.

4.7.1 Ovládání

Ovládání hráče a jeho pohyb vpřed, vzad a do stran je pomocí klávesnice. Přesněji klávesy WASD nebo také možnost využití alternativy v podání šipek. Je zde také možnost zvýšit rychlost pohybu hráče za pomoci levé klávesy „Shift” a nebo skok hráče za pomoci mezerníku. Uživatel může ovládat změnu zorného pole za pomoci posunu myši, nebo střelbu a zamíření z kuše za pomoci levého a pravého tlačítka myši. Interakce s předměty je hráči předem oznamována na obrazovce. Pokud je hráč v dosahu interaktivního předmětu, který může využít, objeví se mu nápověda s instrukcemi pro aplikování předmětu.

4.7.2 Animace

Animace fungují na principu rutiny se spouštěči provedení jiných akcí. Animace hráče má rutinní smyčku pohybu, kde se podle rychlosti hráče rozhoduje, zda se využije animace ve stoje nebo v běhu. Na tuto rutinní část jsou navázány animace, které jsou podmíněny spouštěčem, přechod ke spuštěné animaci je okamžitý bez prodlevy, po vykonání animace se poté už s prodlevou přejde zpět do rutinní smyčky. Animace, které jsou podmíněné spouštěčem je střelba z kuše, zamíření a útok na blízko. Dále také je zde speciální animace újmy na zdraví, do které se může přejít z jakéhokoliv stavu.

4.8 Záporné postavy

Záporní hrdinové příběhu je možné rozdělit do dvou skupin podle počtu životů a také poškození, které způsobují. Každá záporná postava je zapojena do HP systému, který však neumožňuje udělit poškození mezi sebou. Hra je implementována tak, že pouze hráč může poškodit nepřítele a naopak. Postavy obsahují kolizní systém a umělou inteligenci na řízení pohybu.

4.8.1 Umělá inteligence

Umělá inteligence je z části tvořena vestavěnou systémovou komponentou pro pohyb těchto objektů a také rozhodovacím skriptem. Postavy rutinně vykonávají svoji hlídkovou cestu, která je dána body na mapě, pořadí bodů je dáno strukturou a postava je prochází podle nastaveného cíle od skriptu. Nepřítel obsahuje konstantní rádius, který slouží pro detekci hráče. Pokud se hráč objeví v daném poli, tak jej postava začne následovat dokud se nedostane na dostatečnou vzdálenost pro udělení poškození hráči, nebo dokud se hráč neztratí ze zorného pole.

4.8.2 Deformace kostry

Záporné postavy na rozdíl od postavy hráče mají systém na deformaci kostry. Ragdoll vestavěná funkce v herním frameworku umožňuje provést deformaci kostry podle bodu poškození, tedy vytvoření určitého defektu. Po úplném vypršení životů postavy se kostra deformuje podle posledního vytvořeného defektu a postava padá na zem.

4.8.3 Rozdělení záporných postav

Ve videohře jsou vytvořeny dva typy záporných postav, které se navzájem liší v počtu životů, způsobujícím poškození a počtem výskytů na mapě. Hlavní záporná postava je Baba Jaga, která má přidělený vyšší počet životů ze všech postav. Postava Baby Jagy je umístěna ve sněžném biomu, kde brání vlastní území. Postava Baby Jagy střílí projektily, které udávají vysoké poškození ve videohře. Postavy živých stromů brání postavu Baby Jagy, navíc je tento typ nepřátel schopen napadnout hráče, nebo jej pronásledovat, zatímco postava Baby Jagy pouze stojí na prahu svého území a nepohybuje se.

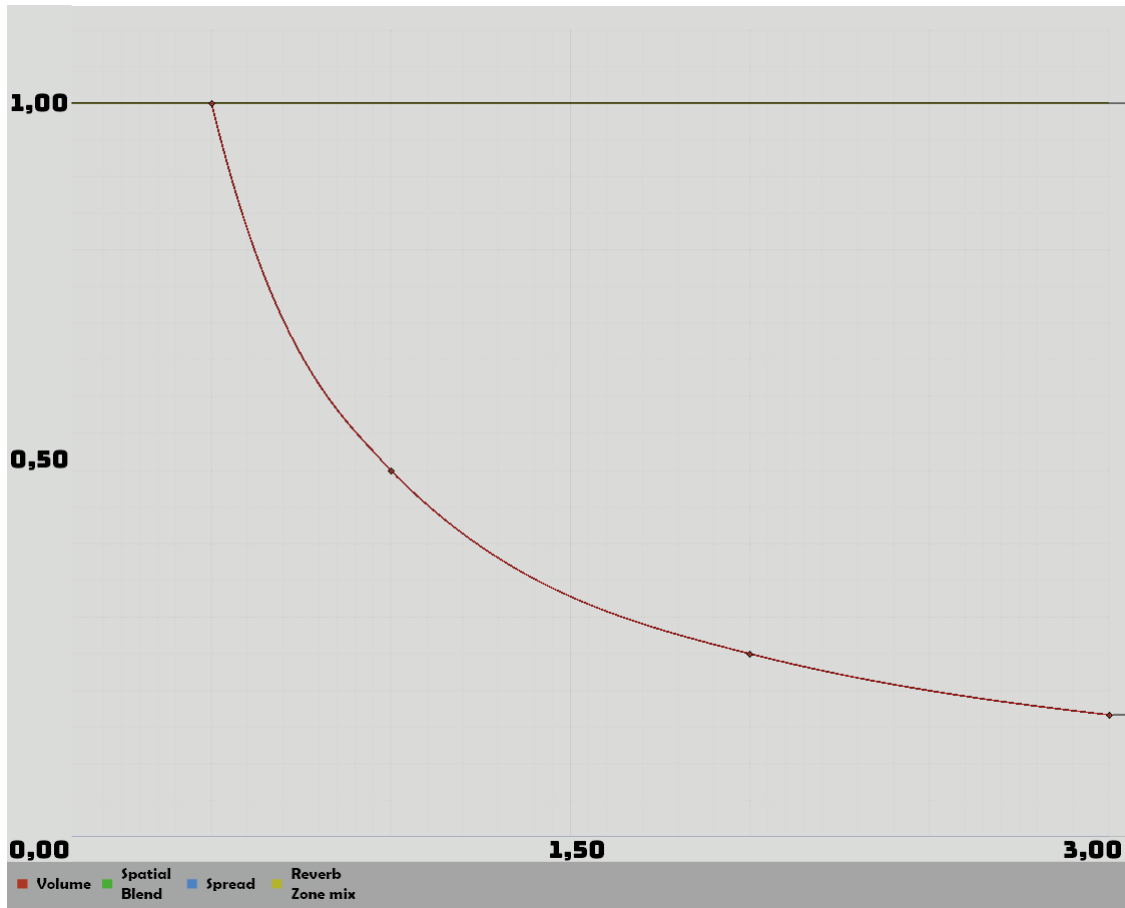
4.9 Zvukové efekty

Zvukové efekty použité ve hře pocházejí od firmy Epidemic Sounds, pro komerční využití. Ve frameworku bylo potřeba určit zvukové zdroje, neboli předměty vydávající zvuk. Zvukový zdroj je přidán na samotný objekt, má pevně daný rádius, který logaritmickou funkcí počítá hlasitost zvuku závislou na vzdálenosti objektu. Objekty ve hře mají většinou pouze jeden vydávaný zvuk, který je pevně přiřazen. Naopak objekty s více efekty jsou řízeny skriptem s přiřazením efektu na základě určité akce. Objekt určený pro příjem zvuku, je implementován pomocí vestavěné funkce audio naslouchač.

Seznam zvukových efektů

- střelba z kuše,
- chůze hráče,
- dopad šípu,

- exploze střely,
- zurčení vody,
- oheň.



Obrázek 4.2: Zvukový zdroj s logaritmickou funkcí

4.9.1 Weberův–Fechnerův zákon

Weberův–Fechnerův zákon popisuje vztah frekvence k lidskému vnímání jako logaritmický jakožto oktáva = $\log_2(f_n/f_{n-1})$, kde n značí celočíselné násobky frekvence. [20]

4.10 Vizuální efekty

Součástí implementace je tvorba vizuálních efektů. Zvukové efekty jsou rozděleny do dvou typů podle jejich funkce a procesu tvorby, kde bylo využito nejen vestavěných funkcí, ale také tvorby vlastního shader grafu. Mezi vizuální efekty ve videohře patří oheň, tok řeky, létání svatojánských mušek a efekt padajícího sněhu chumelenice.

4.10.1 Částicový systém

Pro tvorbu přírodních efektů kouře a ohně, byl využit částicový systém, který je vestavěnou funkcí herního frameworku Unity. Funkce požaduje minimálně tři vstupní parametry pro vygenerování objektů (intervalové rozmezí, velikost vygenerovaného objektu a počet objektů). Částicový systém se spouští v daných intervalech, který omezuje životnost vygenerovaných útvarů. Na začátku intervalu se vygeneruje objekt, který se v průběhu času deformuje až do konce intervalu. Systém také umožňuje omezit počet aktivně vygenerovaných objektů, který má vliv na náročnost vykreslování.

Jedním z příkladů použití částicového systému je roj svatojánských mušek. **Svatojánské mušky** jsou rozmístěny po mapě, kde se na náhodném místě vygenerovaly objekty, které na začátku své instance svítí žlutě a postupně snižují intenzitu světla až objekt zanikne.

Efekt padajícího sněhu je implementovaný jako částicového systému. Objekt vločky je reprezentován dvou-dimenzionálním čtvercem. Kvůli své velikosti, kterou hráč není schopen detailně rozpoznat a kvůli náročnosti vykreslování čtverce a vločky samotné. Vločky se vygenerují nad mapou v pevně dané výšce s postupným pohybem k mapě a postupným zmenšováním.

4.10.2 Shader

Shader je vytvářen v herním frameworku pomocí návrhu grafu, nebo samotným psaním kódu. Shader je aplikovatelný na dvourozměrné i trojrozměrné objekty. V rámci bakalářské práce byly vytvořeny shadery aplikované na 2D objekty, kde byl následným nakombinováním vytvořen 3D vjem.

Jedním z příkladů implementace ve videohře je **řeka**. Pohyb vody v řece je vytvořen pomocí shaderu, vytvářející plošné geometrické útvary, které se náhodně deformují a pohybují, při pohybu náhodně mění barvu a také intenzitu zářivého světla, pro přírodně vypadající odlesk nočního svitu. Veškerá funkcionality vychází ze závislosti na čase, kde se využívá například počítání normály, hluk přechodu, násobení a různé odsazení. Vrcholy objektu, na který je shader aplikovaný postupně mění pozici a vytvářejí vlnovitý pohyb.

Dalším příkladem jsou **plameny ohně**. Plameny stoupající vzhůru jsou vytvořeny pomocí shaderu, který na začátku generuje různé geometrické plošné útvary stoupající vzhůru s postupným vytrácením. Efekt plamene je tvořen v závislosti na čase a rychlosti vykreslování. Stoupající útvary jsou filtrovány pomocí textury ohně. Textura ohně představuje černobílý obrázek s různou intenzitou alfa kanálu (tmavé filtrují kde se oheň nebude promítat, zatímco stupně bílé určují intenzitu vykreslení alfa kanálu).

4.11 Herní logika

Důležitým aspektem hry je herní logika obsažena nejen v hráčských úkolech, ale také v pohybujících se předmětech. Herní logika řeší funkčnost vyhodnocování úkolů, herní systémy, interakci s uživatelem a pohyb předmětů. Tato práce má svojí herní logiku schovanou ve skriptech a různých funkcích. Většina herních objektů ukrývá skript, který obsluhuje jejich pohyb a interakci.

4.11.1 Možnosti hráče

Hráč má možnost interakce s herními objekty, které patří k hlavním a vedlejším úkolům. Hráč má také možnost interakce s předměty, které nejsou důležité pro herní děj, jako objekty sloužící pro doplnění například životů hráče. Interaktivita předmětu je založena na zapojení objektu do kolizního systému.

Hlavní úkoly jsou v herním prototypu celkem tři. Principiálně založené na detekci předmětů a následném vyhodnocení úkolu. Objekt zahrnut v úkolu je řízen skriptem, který detekuje vstup objektu do zóny. Zóna předmětu je určena geometrickým tělesem nejčastěji koulí nebo šestistěnem, který je nastavený jako průchozí kolizní objekt, funkční jako spouštěč akce. Pokud hráč koliduje s hranicí tohoto tělesa, tak se spustí metoda na detekci objektu. Po zavolání se porovná identifikátor herního objektu. Pokud se identifikátor herního objektu rovná požadovanému, je úkol vyhodnocen jako hotový.

Vedlejší úkoly v podobě střelby na terč, je založena na detekci šípu a zasaženého kruhu. Objekt šíp má určený identifikátor jako „Arrow”, terč je rozložen do segmentů, podle bodů udělených za zásah (nejmenší kruh uděluje nejvíce bodů, a naopak největší kruh uděluje nejméně bodů). Objekty disponují dvěma kolizními tělesy. První kolizní těleso kopíruje mesh objektu a druhé je určeno pro detekci kolize s objektem. První kolizní těleso kopírující předmět je zde implementováno z důvodu připevnění šípu na terč, v případě kolize se objekt šípu upne na terč, aby bylo zabráněno svévolnému pádu šípu na zem. V případě úspěšného vyhodnocení bodů a následném připsání se objekt terče stává neaktivním.

Využitelný herní předmět slouží pro doplnění životů hráče. Objekt je koncipován jako houba a na herní mapě je reprezentována jako modrá zářící houba. Objekt má implementovaný kolizní radius pro detekci vstupu hráče. Pokud hráč vstoupí do zóny objektu, zobrazí se mu nápověda s pokyny pro zvednutí (nápověda zmizí v případě, že hráč opustí zónu, nebo využil předmět).

4.12 Kolizní systém

Kolizní systém zajišťuje nejen interakci mezi hráčem a předmětem, ale také zamezuje nepřirozené průchodnosti objektů. Systém využit pro detekci kolize slouží jako spouštěč, který funguje na principu vstupního parametru pro metodu jiného systému. Veškeré herní objekty v mapě jsou zapojeny do kolizního systému. Všechny herní objekty jsou zapojeny do kolizního systému a obsahují základní geometrické těleso kopírující jejich vlastní mesh (základní kolizní těleso podporuje pouze neprůchodnost objektu).

4.12.1 Pokročilá detekce kolizí

Unity využívá algoritmus pokročilé detekce kolizí (CCD), který využívá metody Sweep-based CCD a spekulativní CCD. Spekulativní algoritmus vybírá v každém fyzikálním kroku všechny potencionální objekty na základě algoritmu blízkých bodu. Fyzikální krok střely je vypočítán za pomoci Sweep-based, který využívá výpočet doby dopadu na základě dopředné trajektorie a pohybové rychlosti objektu. Pokud se herní objekt nachází v trajektorii nebo podél směru, je zařazen do množiny kontrolovaných kolizních objektů, která cyklicky provádí kontrolu, zda se objekty překrývají. [21]

4.12.2 Interaktivní předměty

Interaktivní předměty navíc obsahují krom základního kolizního objektu, také průchodné geometrické těleso sloužící jako spouštěč. Předměty mají nastavený speciální identifikátor, který se využívá při vyhodnocování kolize. Vyhodnocení kolize probíhá na bázi skriptu, kde se porovnává identifikátor předmětu, který koliduje.

4.13 HP systém

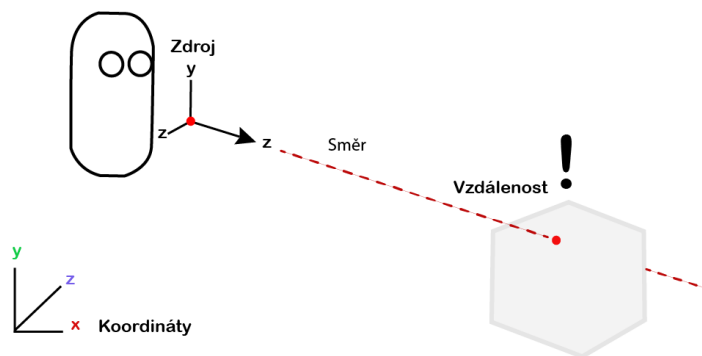
HP systém vyžaduje implementaci kolizního systému, který slouží jako spouštěč požadavku na vyhodnocení. Postava zapojená do HP systému má na začátku hry přidělený počet životů, které se mu během hry mění podle uděleného poškození nebo doplnění životů, zároveň maximum životů je pevně stanoveno. Po dosažení maximálního počtu životů není umožněno postavě doplňovat životy. Součástí kolizního systému je také udělení poškození nepříteli. Poškození je udáváno samotným modelem náboje, který při zjištění kolize udělí dané postavě poškození (velikost poškození je pevně dána a nesouvisí na zasažené části těla postavy).

4.13.1 Herní postavy

Zatímco hráč a postava Baby Jagy mají implementovaný pouze kapsloví tvar kolizního tělesa pro detekci střely, tak živé stromy mají na každé kosti vlastní kolizní objekt, jednak kvůli implementaci ragdoll a také kvůli jejich pohybu. Při pohybu kosti v těle rotují a zasazené střely by nepřírozně levitovaly.

4.14 Hitscan

Hitscan je nejčastější implementací v akčních videohrách z první osoby. Ve videohrách se hitscan nazývá jako „okamžitý zásah“, jelikož při výstřelu se okamžitě aplikuje poškození na cíl. Ve videohře je implementován hitscan pomocí raycastu¹. Při výstřelu je předem známa trajektorie střely, funkčně je vyslán paprsek s lineární trajektorií směrem od zdroje ze kterého vychází. Pokud je v dráze střely detekován objekt, je registrován zásah. Pro zlepšení realistického dojmu střely se používá náhodná porucha cesty k simulaci nepřesnosti.[22] Ve videohře je navíc implementována metoda, která cyklicky kontroluje detekci nového předmětu v trajektorii střely.



Obrázek 4.3: Raycast

4.14.1 Výhody

Hlavní výhodou je jednoduchá implementace oproti jiným typům střel, díky poměrně jednoduchému výpočtu matematickým vzorcem. Metoda na detekci zásahu předmětu je implementována pouze jednou funkcí, což zjednodušuje nejen vyhodnocování, ale i programování zbraně. Hráč nalezne výhodu v lineární trajektorii střely. Hráči stačí namířit na objekt a ten zasáhne přesně, nemusí mířit nad objekt pro kompenzaci doby, kterou střela potřebuje k dosažení cíle.[22]

¹přímka vedoucí od zdroj až po cíl zásahu

4.14.2 Nevýhody

Nevýhoda hitscanu spočívá v okamžitém zásahu. Jelikož střela zasahuje svůj cíl okamžitě, tak bez přídavného vyhodnocování kolize předmětu, je náboj pouhý předmět bez významu. Místo, kam střela dopadne, nemusí nutně odpovídat skutečnému zásahu. Další nevýhoda spočívá v obtížnosti změny trajektorie střely, jelikož při výstřelu je vyslán paprsek, který zasáhne cíl okamžitě, nemůže dojít k žádným změnám dráhy střely. [22]

4.14.3 Raycast

Vestavěná metoda na tvorbu paprsku. Podle vstupních parametrů metoda vygeneruje paprsek a vrátí logickou hodnotu true, pokud paprsek koliduje, v opačném případě hodnotu false. Mezi vstupní parametry patří pozice zdroje (zdroj vysílaného paprsku), směr (dědí od zdroje) a také maximální vzdálenost paprsku. Volitelně je možné implementovat masku, která odfiltruje všechny nechtěné kolize.

4.14.4 Kontrola kolize

Kontrola kolize je prováděna hned po výstřelu, využití raycast funkce nám okamžitě vrátí zasažený cíl. Po obdržení zasaženého předmětu se porovnává identifikátor herního objektu. Pokud se identifikátor rovná „ArrowTarget“, posílá požadavek na vyhodnocení do manažera skóre. V opačném případě se pošle požadavek s uvedeným poškozením na daný herní objekt.

4.15 Projektil

Projektil je střela s obloukovou trajektorií, způsobena působením fyzických sil (gravitace). Při výstřelu se vytvoří fyzikální objekt, která má vlastní hmotnost, rychlost a sledovaný hitbox. Implementace je vytvořena instancí objektu, kde se v každém cyklu hry kontroluje kolize. Na začátku se vytvořenému náboji přiřadí rychlost a směr, kterým se bude pohybovat. Dráha vrženého tělesa je ovlivněna fyzikálními faktory jako je gravitace. Gravitace v reálném světě způsobuje přitahování hmotného tělesa směrem do středu země, ve videohře se jedná pouze o směr k mapě. Střela v závislosti na čase zpomaluje. Náboj se zastaví v případě kolize s herním objektem. Implementace probíhá na bázi skriptu, kde je konstantně určené udělené poškození střely a metoda na detekci kolize (složitost algoritmu pro výpočet potenciálních kolizí je $O(n^2)$, kde n představuje množinu potenciálních kolizních objektů, vypočítaných za pomoci CDD).[22]

4.15.1 Výhody

Výhodou projektilu je reálná simulace trajektorie střely, kterou ocení nejen hráč. Využívá se především v realistický videohrách. Výhodou je také kolize s herními objekty, kde na rozdíl od metody hitscanu, který může projít až několika objekty. Projektil při první kolizi přestane působit a deaktivuje svojí rychlost.[22]

4.15.2 Nevýhody

Nevýhoda spočívá v neúměrné rychlosti střely vzhledem k počtu framů za sekundu. Když je rychlost příliš vysoká, je možné, že objekty nezaznamenají kolizi předmětu, ve videohrách je tento jev nazýván jako „bug“. Jedná se kolizi, která proběhne mezi jednotlivými framy videohry.

4.15.3 Fyzické vlastnosti

Projektil má ovlivněnou trajektorie především gravitací. Působení fyzikálních jevů na objekt je v herním frameworku reprezentováno jako komponenta Rigid Body, která udává použití gravitace. Hlavními vstupními parametry jsou hmotnost, kinematika a podobné vlastnosti.

4.15.4 Kontrola kolize

Kontrola kolize střely probíhá ve skriptu, pokud projektil koliduje s jakýmkoliv objektem. Kontrola probíhá na základě porovnání identifikátoru kolizního objektu. Jelikož ve videohře je implementován projektil na střely hlavní záporné postavy, která střílí pouze po hráčovi, tak vyhledávaný identifikátor kolizního objektu je nastaven na identifikátor „Player“. V případě úspěšné kontroly identifikátoru objektu, je poslán požadavek do HP systému na udělení poškození hráčovi.

5 Kritické zhodnocení

Bakalářská práce byla vytvářena osm měsíců. Videohra má hned několik nedostatků týkající se optimalizace a herní logiky. Vývoj videohry, jak už bylo řečeno, trvá několik let a pracují na ní stovky vývojářů a designerů. Velký nepoměr v čase a v počtu členů týmu staví tuto hru do pozice, kdy se nemůže srovnávat s komerčním vývojem videoher. Naopak srovnání je možné vyhledávat mezi indie videohrami. Indie videohra je hra vytvářena jednotlivcem nebo malým týmem.

5.0.1 Srovnání

Srovnání je možné hledat v komunitě videoher vytvořených indie vývojáři. Mezi známé indie videohry patří Among Us, Overcooked, The First Tree a Valheim. The First Tree videohra¹ má podobně pojatý koncept herního stylu a celkově hry. David Weil videohru vyvíjel dva roky, v porovnání s výsledkem práce, která vznikla za 8 měsíců. Doba je méně než poloviční. 5.1

5.1 Optimalizace

Tvorba herního prototypu byla časově omezenou záležitostí, proto videohra skrývá mnoho nedostatků. Vývoj videoher je projekt, na kterém lze pracovat roky a stále bude co zlepšovat. Například jedna z lépe propracovaných videoher současnosti Grand Theft Auto V se vyvíjela mezi lety 2008-2011 s početným týmem převyšujícím 1000 členů.

5.1.1 Grafický styl

Zvolený grafický styl byl zvolen jako méně náročný a tudíž i jednodušší na tvorbu assetů. Určitě je možné zvolit lepší grafický styl, který bude ovšem časově náročnější na tvorbu assetů, ale bude vypadat lépe. V dnešní době jsou technologie na vzestupu a vyvíjejí se stále novější a výkonnější grafické karty. Optimalizačním krokem by bylo zvolit realistický grafický styl, lepší propracovanost herních postav a také modelů. Takovýto proces vyžaduje několik let tvorby modelů.

¹autor The First Tree byl David Weil, prvně vydaná v roce 2017



Obrázek 5.1: Pohled The First Tree [23] a výsledku práce

5.1.2 Herní prostředí a děj

Herní prostředí je v práci vytvořeno v Low Poly grafickém stylu, který je významný hlavně nízkou náročností na vykreslování, díky malému množství polygonů v modelu. Optimalizačním krokem by mohl být rozvoj stávající herní mapy, rozšíření o další herní biomy, obohacení herní flóry, přidání fauny a nové oblasti. Rozšíření herní mapy by vedlo k většímu počtu herních úkolů. Zlepšení by vyžadovaly také herní postavy, které by mohly být detailněji vymodelovány. Bylo by možné zakomponovat nové záporné hrdiny, nebo nové interaktivní předměty.

5.1.3 Herní logika

Herní logika je aktuálně založena tak, aby herní interaktivní objekty, které jsou součástí hlavních úkolů, byly jednoduše zvládnutelné. Herní logika by mohla být obohacena o úrovně obtížnosti, které by se lišily například v počtu životů nepřátel, bojových dovedností a podobně. Herní logiky je také tvorba animací, kde v aktuální práci bylo využito animace aplikující se na celou postavu. Unity framework nabízí aplikování animací na různé části postavy (například spodní část končetin je ovládána jinou animací než vrchní část těla). Možnost rozšíření o další herní úkoly, které

by mohly vyžadovat pro splnění přemýšlení a určité herní dovednosti. Nové herní systémy, jako obchodní, inventář a character upgrade systém, který by umožnil správu silných vlastností hlavního hrdiny. Zmíněné optimalizace v oblasti herní logiky by mohly zlepšit celkový herní dojem, avšak tyto optimalizace jsou otázkou početnějšího týmu vývojářů (počet členů je přímo úměrný kvalitě hry).

5.1.4 Metoda střelby

Práce využívá dvě metody střelby, metody hitscan a projektil. Krok optimalizace přináší hybridní chování střely podle vzdálenosti cíleného objektu. Pokud zbraň střílí na blízko, je málo faktorů, které by mohly ovlivnit trajektorii, tudíž pokud cíl bude blízko využije hitscan. Když cíl bude dál, bude muset hráč využít metodu projektilu, kde střela bude ovlivněna gravitací a také možností, že do trajektorie vstoupí nový předmět.

5.1.5 UI Systém

Aktuálně videohra disponuje pouze UI komponenty na zobrazování aktuálního počtu životů hráče, ukazatel úkolů a ukazatel využitých Easter eggs. Pro produkční nasazení by videohra potřebovala vytvořit herní menu, kde je možné si zvolit herní styl, popřípadě nastavení videa, zvuku a ovládání.

5.2 Kompatibilita

Aktuálně je možné si videohru vyzkoušet na desktopových zařízeních s nainstalovaným operačním systémem Windows. Možnost stažení instalačního balíčku z GitHubu. Videohru je možné rozšířit na ostatní desktopová zařízení jako MacOS a Linux, nebo také rozšíření na mobilní zařízení. Zvolený grafický styl je přímo určený pro vývoj videoher pro mobilní zařízení, která nemají dostatečné hardwarové zastoupení (po krátkém čase se přehřívají apod.).

5.3 Rozšíření

Rozšířitelnost je zcela vhodná, už jen kvůli zvolenému grafickému stylu, který je vhodně vybrán pro plynulé vykreslování na jakémkoliv podporované platformě. Možnost rozšíření přidáním nových herních objektů, záporných hrdinů a zvětšení herní mapy. Možnost přejít k novějšímu grafickému nastavení ve frameworku Unity, například HDRP, pro který je nutná podpora materiálů a shaderů.

6 Závěr

Cílem bakalářské práce bylo seznámit se s oblíbenými herními frameworky a vybrat vhodný framework pro tvorbu videohry z první osoby (First-person Shooter). Součástí videohry by měla být nejen implementace hit-scanu a projektilu, ale i tvorba herního prostředí a příběhu videohry.

Prvotním bodem bakalářské práce byla volba vhodného herního frameworku pro vytvoření videohry. V rámci práce byla vypracována jednoduchá demonstrativní videohra, která rozhodla, jaký herní framework bude vhodný. Byl vybrán herní framework Unity na základě předchozí zkušenosti autora práce. Dalším důležitým vývojovým prostředím byl Blender, zvolený modelovací nástroj pro tvorbu herního prostředí a postav.

Vytvoření herního prostředí a postav byl komplexní a časově náročný proces. Bylo využito procedurální generování herního světa, za pomoci Perlinova šumu. V rámci bakalářské práce bylo vymodelováno 104 herních assetů, z toho časově náročnější byl model postavy Baby Jagy, trvající 64 hodin. Koncept herního děje je pojatý jako magická fantazijní pohádka, kde hlavní záporné hrdiny hrají filmové postavy. Jelikož tvorba videohry je časově náročný proces i pro několik set herních vývojářů, práce byla pojata jako tvorba prototypu obsahující pouze jednu část z několika možných. Tento herní prototyp se zaměřuje na les navozující magickou atmosféru. Prototyp dále obsahuje postavy nepřátel ve formě živých stromů a zlé ježibaby.

Implementace probíhala ve vybraném herním frameworku, kde bylo vytvořeno několik herních animací a celková herní logika. Implementace obsahovala tvorbu herních systémů, jako HP a kolizní systém, který byl přímo provázaný s metodami střel. První metodou byl Hit-scan, jednodušší metoda instance střely, realizována pomocí raycastu (při výstřelu je znám cíl zásahu). Naopak druhá metoda už není tak přímočará, její implementační složitost je vynahrazena realistickou trajektorií střely. Projektil je střela, na kterou působí fyzické síly (gravitace).

Výsledkem bakalářské práce je videohra se zajímavým herním dějem. Videohra je koncipována jako fantazijní akční videohra. Tomu odpovídají herní předměty, postavy a celkově zvolený grafický styl, který je zvolen z důvodu menší náročnosti na vykreslování. Omezujícím faktorem v celé bakalářské práci byl čas, jelikož autor samotný zastoupil několikačlenný tým složený z vývojářů, grafiků apod.

Použitá literatura

- [1] TYLER, Dustin. *How to Choose the Best Video Game Engine*. 2022-03. Dostupné také z: <https://www.gamedesigning.org/career/video-game-engines/>.
- [2] SAIQA ALEEM, Luiz Fernando Capretz; AHMED, Faheem. *Game development software engineering process life cycle: a systematic review*. 2016. Dostupné také z: <https://jserd.springeropen.com/track/pdf/10.1186/s40411-016-0032-7.pdf>.
- [3] SAIQA ALEEM, Luiz Fernando Capretz; AHMED, Faheem. *Critical success factors to improve the game development process from a developer's perspective*. 2016. Dostupné také z: <https://arxiv.org/ftp/arxiv/papers/1801/1801.04293.pdf>.
- [4] WIKIPEDIA. *Dějiny videoher*. Dostupné také z: https://cs.wikipedia.org/wiki/D%C4%9Bjiny_videoher#:~:text=Ralph%20Baer%20vytvo%C5%99il%20v%20roce,p%C5%99%C3%ADstrojem%20tohoto%20typu%20na%20sv%C4%9Bt%C4%9B..
- [5] BOJNANSKÁ, Katarína. *Kde v Česku studovat počítačové hry?* 2021-03. Dostupné také z: <https://visiongame.cz/kde-v-cesku-studovat-pocitacove-hry/>.
- [6] BARÁK, Pavel. *Monitoring herního průmyslu*. Dostupné také z: <https://gda.cz/cs/monitoring/#:~:text=V%20C4%8Desku%20je%20aktu%C3%A1ln%C4%9B%20110,skl%C3%ADz%C3%AD%20velk%C3%A9%20C3%BAsp%C4%9Bchy%20ve%20sv%C4%9Bt%C4%9B.>
- [7] GAMECAREERGUIDE.COM. *Game Engine*. 2013-11. Dostupné také z: https://www.gamecareerguide.com/features/529/what_is_a_game_.php.
- [8] ARORA, Simran Kaur. *Unity vs Unreal Engine: Which Game Engine Should You Choose?* Dostupné také z: https://hackr.io/blog/unity-vs-unreal-engine#google_vignette.
- [9] DEALESSANDRI, Marie. *What is the best game engine: is Unity right for you?* 2020-01. Dostupné také z: <https://www.gamesindustry.biz/articles/2020-01-16-what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you>.

- [10] MILES KRUPPA, Tim Bradshaw a. *Epic and Unity rev their engines for the next era of entertainment*. 2020-08. Dostupné také z: <https://www.ft.com/content/f77b7979-c943-4b9d-b7b7-7953b63bea7e>.
- [11] WIKIPEDIA. *Počítačová 3D grafika*. Dostupné také z: https://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A1_3D_grafika.
- [12] ING. KATEŘINA RŮŽIČKOVÁ, Ph.D. Geometrické reprezentace 3D modelů. 2021. Dostupné také z: https://homel.vsb.cz/~ruz02/3D_modelovani/pr2_geometricke_reprezentace.pdf.
- [13] UNITY. *Unity User Manuel 2021.3 (LTS)*. Dostupné také z: <https://docs.unity3d.com/Manual/index.html>.
- [14] THOMPSON Jim a Berbank-Green, Barnaby. *Game design course: principles, practice, and techniques*. 2007. Dostupné také z: https://books.google.cz/books?id=Zyh4HXuHTPMC&pg=PA144&redir_esc=y#v=onepage&q&f=false.
- [15] BOXOPHOBIC. *Polyverse Skies — Low Poly Skybox Shaders*. Dostupné také z: <https://assetstore.unity.com/packages/vfx/shaders/polyverse-skies-low-poly-skybox-shaders-104017#publisher>.
- [16] SOUNDS, Epidemic. *Epidemic Sounds*. Dostupné také z: <https://www.epidemicsound.com/>.
- [17] PERLIN, Ken. *Perlin noise*. 1999-12. Dostupné také z: <https://web.archive.org/web/20071011035810/http://noisemachine.com/talk1/>.
- [18] UNITY. *Unity Documentation, Lights*. Dostupné také z: <https://docs.unity3d.com/Manual/Lighting.html>.
- [19] UNITY. *Unity Documentation, Emissive materials*. Dostupné také z: <https://docs.unity3d.com/Manual/lighting-emissive-materials.html>.
- [20] WIKISKRIPTA. *Weberův–Fechnerův zákon*. 2019-05. Dostupné také z: https://www.wikiskripta.eu/w/Weber-Fechner%C5%AFv_z%C3%A1kon.
- [21] UNITY. *Continuous collision detection (CCD)*. Dostupné také z: <https://docs.unity3d.com/Manual/ContinuousCollisionDetection.html#sweep>.
- [22] JUNG, Tristan. *How Do Bullets Work in Video Games?* 2019-12. Dostupné také z: <https://www.gamedeveloper.com/programming/how-do-bullets-work-in-video-games->.
- [23] WEHLE, David. *The First Tree*. 2017. Dostupné také z: <https://gamerinfo.net/game/the-first-tree/>.

a

7 Seznam příloh

7.1 GitHub odkaz

Odkaz na GitHub, kde je možné si stáhnout instalační soubory. [GitHub](#)

7.2 Odkaz na YouTube

Odkaz na video z celé videohry. [Dlouhé video](#)

7.3 Odkaz na YouTube

Odkaz na krátké video ze hry. [Krátké video](#)

7.4 Příběh

Příběh vypráví o zlé čarodějnici, která zaklela celý les do temnoty a zvířátka uvrhla do zimního spánku, kvůli porušení úmluvy o vstupu do zimního teritoria zlé ježibaby. Kletba byla vložena do zářící hvězdy, která je umístěna nad vchodovými dveřmi chaloupky. Jediným způsobem zlomení kletby je rozbít zářící hvězdu. Poté se život v temném lese navrátí opět k normálu a zvířátka se probudí ze zimního spánku. Úkolem hráče je zlomit kletbu uvrženou na temný les.

7.5 Dokumentace na ovládání videohry

Instalace a ovládání 3D FPS hry

Instalace a ovládání 3D FPS hry

Obsah

1. Instalační soubory	1
Google Disk	1
GitHub	1
Kontaktovat Support	1
2. Instalace videohry	2
1. Stáhnout instalační soubor	2
GitHub	2
Google Drive	2
2. Rozbalte ZIP	2
3. Spustit soubor "Setup.exe"	3
4. Volba umístění instalace	3
3. Ovládání hry	4
Supstít hru	4
Ovládání klávesnicí	4
Pohyb vpřed	4
Pohyb vzad	4
Pohyb vlevo	5
Pohyb vpravo	5
Skok	5
Interakce s předměty	6
Sprint	6
Ovládání myši	6
Změna zorného pole	7
Střelba ze zbraně	7
Zamíření zbraní	7
4. Řešení problémů	8
Nemohu najít instalační soubory	8
Nemohu najít instalační soubor Setup.exe	8
Nemohu nalézt uvedené zdroje s instalačními balíčky	8
Kontrola správnosti URL	8
Pokud problémy přetrvávají	8
Ostatní	8
5. Kontakty	9

Kapitola 1. Instalační soubory

Odkazy pro potřeby stáhnutí instalačních souborů. V případě neúspěšného stažení z níže uvedených odkazů, kontaktuje prosím náš support, kde už Váš konkrétní problém je v procesu řešení nebo vyřešen.

Google Disk

Odkaz, kde jsou uchovány instalační soubory pro 3D FPS videohru <https://drive.google.com/drive/folders/1ekV-yhuUsuJrJq1NNotllyYhhXfGICvOc?usp=sharing>

GitHub

Odkaz, kde jsou uchovány instalační soubory pro 3D FPS videohru <https://github.com/BartonAdrian/Bachelor-thesis>

Kontaktovat Support

Možnost vyžádat si externí odkaz na stáhnutí od našeho support týmu. Tato možnost je až v případě, že vaše pokusy o stáhnutí z předešlých dvou zdrojů byly neúspěšné.

Kapitola 2. Instalace videohry

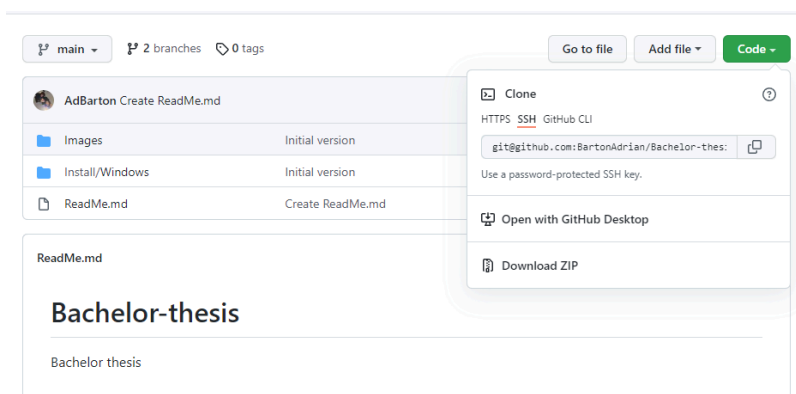
Oblast zabývající se instalací hry. Možnost stažení instalačních balíčků z GitHubu nebo Google Disku. Dále také možnost kontaktovat náš Support, který Vám poskytne podporu při stažení.

1. Stáhnout instalační soubor

Stáhnutí instalačních souborů pro instalaci hry 3D Frist person shooter. Jako náš prvotní atchiv, kde jsou nejaktuálnější instalační soubory je GitHub. Dále také soubory ukládáme na Google Disk, ale ten je použit jako záložní zdroj.

GitHub

Přejděte na Github [<https://github.com/BartonAdrian/Bachelor-thesis>] a stáhněte si repozitář jako ZIP.

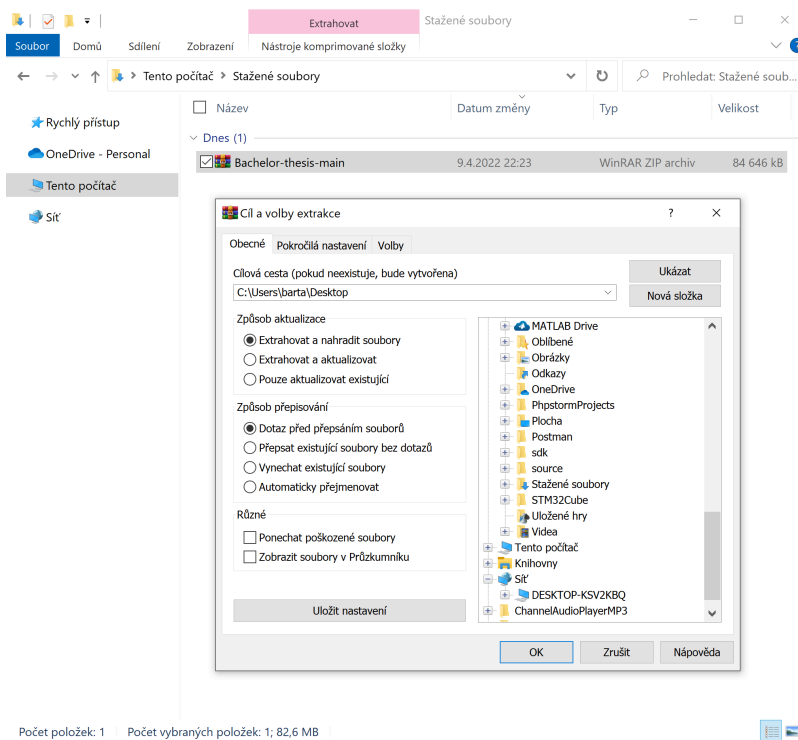


Google Drive

Přejděte na Google Drive [<https://drive.google.com/drive/folders/1ekV-yhuUsuJrJq1NNotlIYhhXfGICvOc?usp=sharing>] a stáhněte soubor "Setup.exe"

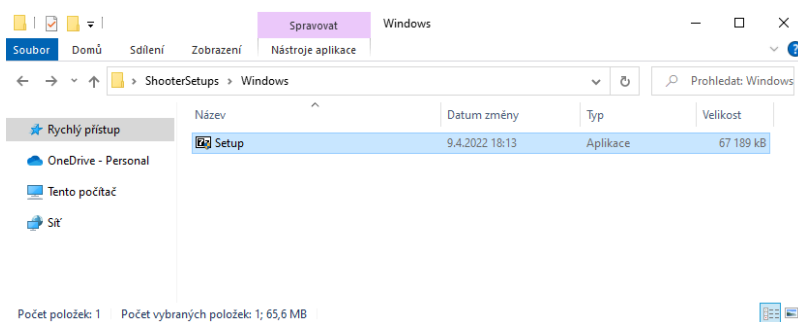
2. Rozbalte ZIP

Klikněte pravým tlačítkem myši na stáhnutý ZIP soubor a zvolte "Extrahovat soubory" a následně potvrďte svojí volbu.



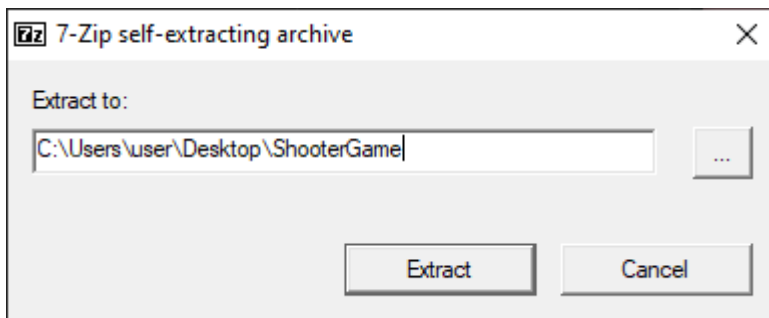
3. Spustit soubor "Setup.exe"

Přejděte do rozbalené složky, v adresáři Install\Windows najdete instalační soubor "Setup.exe".



4. Volba umístění instalace

Vyberte místo na disku, kde bude nainstalována aplikace.

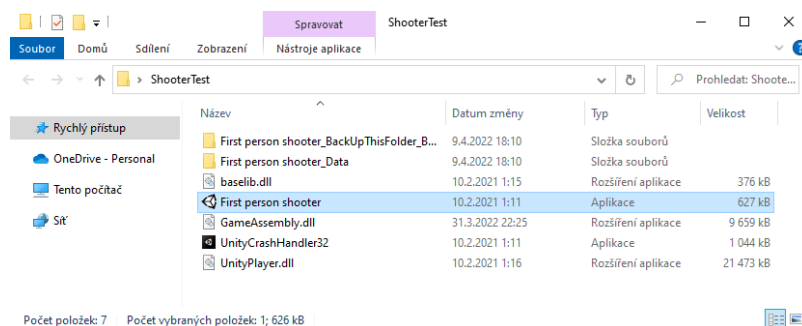


Kapitola 3. Ovládání hry

Stručný popis funkcí hráčských možností. Počínaje zapnutím hry až po střelbu ze zbraně. Pro tento bod je podmíněný splněním bodu 2 – „Instalace videohry“ („2. Rozbalte ZIP“, „3. Spustit soubor "Setup.exe"“ a „4. Volba umístění instalace“)

Spustit hru

Přejděte do složky s nainstalovanou videohrou a spustě soubor "First person shooter".

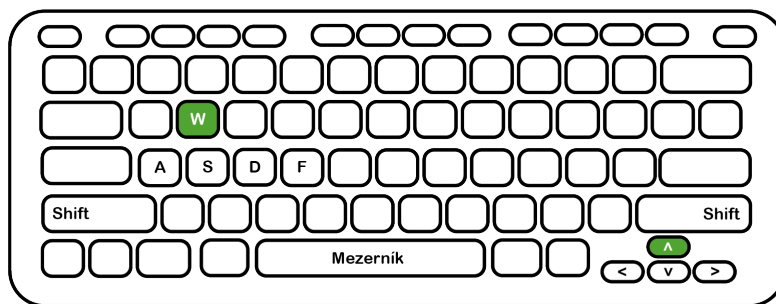


Ovládání klávesnicí

Ovládání hráče pomocí klávesnice a myši. Klávesnice obsluhující pohyb vpřed, vzad a do stran..

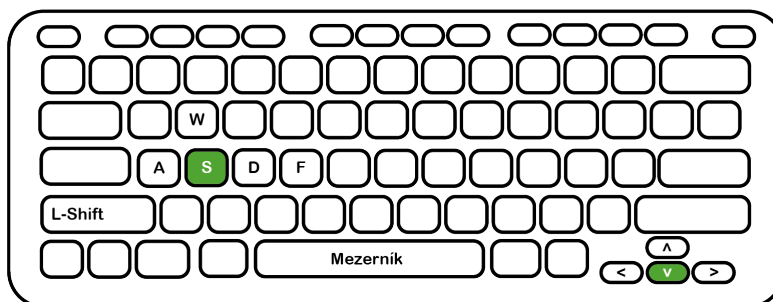
Pohyb vpřed

Pohyb vpřed obsluhuje klávesnice "W" nebo také šipka nahoru v pravé části klávesnice.



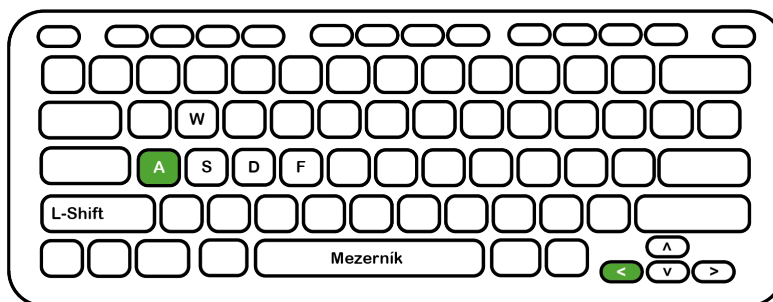
Pohyb vzad

Pohyb vzad obsluhuje klávesnice "S" nebo také šipka dolů v pravé části klávesnice.



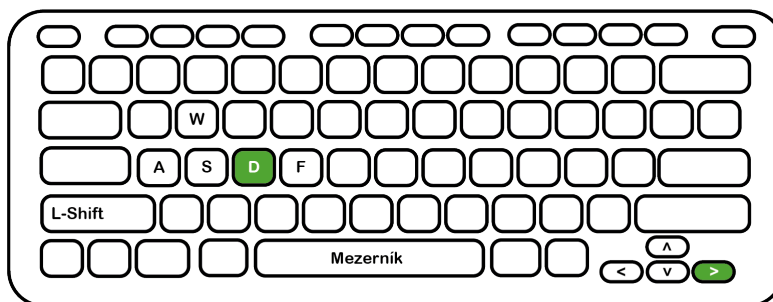
Pohyb vlevo

Pohyb vlevo obsluhuje klávesnice "A" nebo také šipka doleva v pravé části klávesnice.



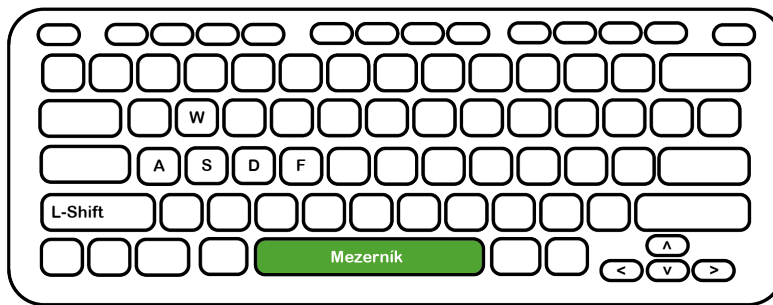
Pohyb vpravo

Pohyb vpravo obsluhuje klávesnice "D" nebo také šipka doprava v pravé části klávesnice.



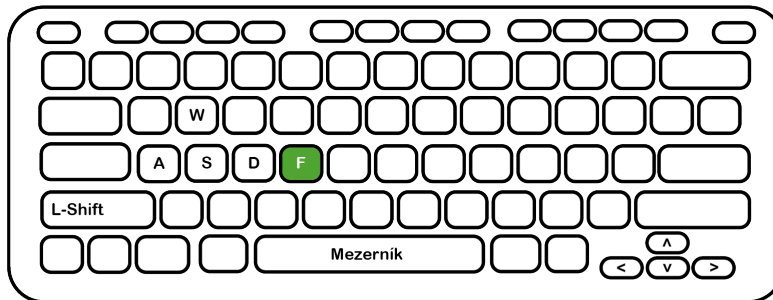
Skok

Skok obsluhuje klávesnice "Mezerník". Funguje pouze jeden skok, nejde zde udělat dvojskok.



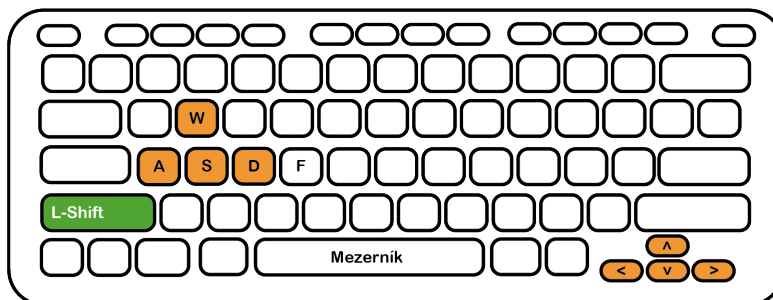
Interakce s předměty

Hra obsahuje také předměty, kde je vyžadována akce uživatele. Například doplnění HP za pomoci sebrání zářivé houby. Takovéto předměty mají u sebe nápovědu a vyžadují stiknutí klávesy "F".



Sprint

Zvýšení rychlost hráče se obsluhuje pomocí libovolné klávesy na pohyb hráče (krom skoku "Mezerník") + klávesy "L-Shift".

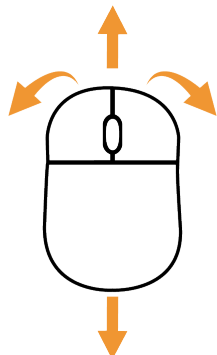


Ovládání myši

Ovládání myši pro změnu zorného pole hráče a nebo také pro zamíření a střelbu ze zbraně.

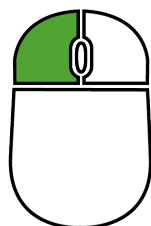
Změna zorného pole

Změnu zorného pole obsluhuje myš, pokud se myš pohybuje směrem nahoru, tak i hráčovo zorné pole se posune nahoru.



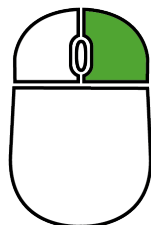
Střelba ze zbraně

Střelba ze zbraně se ovládá stisknutím levého tlačítka na myši.



Zamíření zbraní

Zamíření na cíl se ovládá stisknutím pravého tlačítka na myši.



Kapitola 4. Řešení problémů

Nemohu najít instalační soubory

Vyzkoušet jiné zdroje než jste použil.

Nemohu najít instalační soubor Setup.exe

V tomto případě vyzkoušejte stáhnout instalační soubory z jiného zdroje.

Nemohu nalézt uvedené zdroje s instalačními balíčky

Prosím zkontrolujete si předchozí kroky, než nás budete kontaktovat.

Kontrola správnosti URL

Prosím zkontrolujte si správnost zadané URL

Pokud problémy přetrvávají

V případě stálého problému kontaktujte Support.

Ostatní

V případě jakýkoliv dalších problémů kontaktujte support, za pomoci webových stránek společnosti nebo přes emailovou adresu.

Kapitola 5. Kontakty

www.3Dfps.com/support

support@3dfps.com

adrian.barton@tul.cz