

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Plošinová počítačová hra využívající engine Godot



2022

Vedoucí práce:
Mgr. Tomáš Kühn, Ph.D.

Marek Schindler

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Marek Schindler
Název práce: Plošinová počítačová hra využívající engine Godot
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2022
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Tomáš Kühr, Ph.D.
Počet stran: 26
Přílohy: 1 CD/DVD
Jazyk práce: slovenský

Bibliographic info

Author: Marek Schindler
Title: Platformer computer game using the Godot engine
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2022
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Tomáš Kühr, Ph.D.
Page count: 26
Supplements: 1 CD/DVD
Thesis language: Slovak

Anotácia

Bakalárska práca je venovaná implementácii plošinovej hry v hernom engine Godot. Text pojednáva o plošinových hrách, pixelovom umení a využitých technológiách. V programátorskej dokumentácii sú opísané skripty implementovanej 2D plošinovej hry.

Synopsis

Bachelor thesis is dedicated to implementation of platformer game in Godot engine. Text describes platformer games, pixel art and used technologies. Scripts of implemented 2D platformer game are described in program documentation.

Kľúčové slová: plošinová hra; Godot engine; pixelové umenie

Keywords: platformer game; Godot engine; pixel art

Ďakujem vedúcemu práce za vedenie, cenné rady a pripomienky.

Čestne vyhlasujem, že som celú prácu vrátane príloh vypracoval/a samostatne a za použitia iba zdrojov spomínaných v texte práce a uvedených v zozname literatúry.

dátum odovzdania práce

podpis autora

Obsah

1	Úvod	6
2	Plošinová hra	7
2.1	Komponenty plošinových hier	7
2.2	Typy plošín v plošinových hrách	7
2.3	Znaky plošinových hier	8
2.4	Príklady plošinových hier	8
3	Pixelové umenie	10
3.1	Techniky pixelového umenia	10
3.2	Využitia pixelového umenia	11
4	Použité technológie	12
4.1	Godot engine	12
4.2	C#	12
4.3	Visual studio	13
4.4	Figma	13
4.5	NAudio knižnica	13
5	Programátorská a užívateľská dokumentácia	15
5.1	Kľúčové pojmy Godotu	15
5.2	Užívateľská dokumentácia	15
5.3	Pomocné skripty	17
5.4	Menu skripty	17
5.5	Editor skripty	18
5.6	Skripty objektov v leveloch	19
6	Štatistika hry	21
	Záver	22
	Conclusions	23
A	Obsah priloženého CD/DVD	24
	Literatúra	25

1 Úvod

Cielom bakalárskej práce bolo zoznámiť sa s herným enginom Godot a prostredníctvom neho vytvoriť hru zo žánru plošinových hier. Úvodná kapitola popisuje komponenty, typy a znaky plošinových hier. Pri tvorbe hry boli inšpiráciou populárne hry 20. storočia. Retro atmosféru implementovanej hry dotvára jednoduchý pixelový dizajn. Tejto problematike sa bližšie venuje druhá kapitola. Medzi technológie využité pri tvorbe plošinovej hry, ktoré opisuje tretia kapitola, patrí v prvom rade programovací jazyk C#. Avšak skriptovací editor enginu plne nepodporuje niektoré editovacie funkcie tohto jazyka, preto musel byť využitý externý editor Visual studio. Dôležitou súčasťou práce je užívateľská a programátorská dokumentácia. V užívateľskej dokumentácii je, pre lepšiu orientáciu v hre, opísaná funkcionálna hra z pohľadu užívateľa. Programátorská dokumentácia opisuje skripty implementovanej plošinovej hry a ich obsah. V závere práce bola vytvorená štatistika s cieľom zistiť, či bola obtiažnosť levelov implementovanej hry dobre rozložená. Úlohou hráča v hre je dostať sa z počiatočnej pozície k zadanému cieľu, pri čom musí cestou prekonať prekážky. Levely vyžadujú logické myslenie a správne načasovanie pohybov. Hra navyše obsahuje nástroj na vytvorenie vlastného jednoduchého levelu z dostupných komponentov.

2 Plošinová hra

Plošinová hra nazývaná aj plošinovka, je typ video hry, v ktorej hráč ovláda postavu dynamickými pohybmi, a to skákaním, behaním, lezením alebo odrážaním sa. Plošinové hry začali vychádzať na začiatku 80. rokov 20. storočia. Hráč v nich ovláda postavu v single screen leveloch. Single screen plošinovky zobrazujú celý level na obrazovke a ich obtiažnosť sa graduálne zvyšuje s každým levelom. Väčšina starých arkádových hier je založená na tejto metóde z dôvodu technických limitácií. O pár rokov neskôr začali vychádzať hry s posuvnou kamerou, ktorá zobrazuje časť levelu podľa toho, ako sa hráčova postava pohybuje. Plošinová hra odkazuje na sériu plošín, na ktoré hráč môže vybehnúť alebo skočiť. V plošinovej hre sa hráč môže stretnúť aj s ďalšími úlohami, ako je riešenie hádaniek, boj s nepriateľmi, vyhýbanie sa prekážkam či zbieranie predmetov. Plošinové hry sa považujú za podžáner akčných hier. Väčšinu elementov môžeme nájsť v iných žánroch, ako napríklad adventúry alebo stratégie. [1]

2.1 Komponenty plošinových hier

Level plošinovej hry sa skladá z komponentov, ktoré kategorizujeme podľa ich významu v leveli. Hlavné kategórie plošinových hier sú plošiny, prekážky, podporovače pohybu, zberateľské predmety a spúšťacie mechanizmy. Komponent môže patriť do viacerých kategórií.

Platformy sú objekty, cez ktoré hráč behá alebo skáče pri prechode levelu.

Prekážky sú objekty, ktoré môžu postavu zraniť, zabiť alebo spomaliť.

Podporovače pohybu sú objekty, ktoré postave pomáhajú v pohybe pri prechode levelom, napríklad trampolíny, laná alebo rebríky.

Zberateľské predmety sú objekty, ktoré hráča odmeňujú, napríklad mince alebo bonusový život.

Spúšťacie mechanizmy sú objekty meniace stav levelu, napríklad páky alebo tlačidlá, ktoré aktivujú platformy alebo objekty meniace vlastnosti postavy. [2]

2.2 Typy plošín v plošinových hrách

Plošina je hlavným stavebným blokom v plošinových hrách. Existuje niekoľko variácií plošín, ale základným typom je plošina, na ktorej môže charakter stáť a nemôže na ňu skočiť cez jej spodnú stranu. Ďalej rozlišujeme plošinu, na ktorú sa dá skočiť zospodu. Následne sú to šmyklavé plošiny, na ktorých má hráč menšiu kontrolu nad pohybom a skokmi charakteru. Plošiny predstavujú šmyklavé povrchy ako olej alebo ľad. Opačným typom je lepkavá plošina, kde hráč zostane prilepený a môže odolať gravitácii. Ďalším typom sú pásové plošiny, ktoré

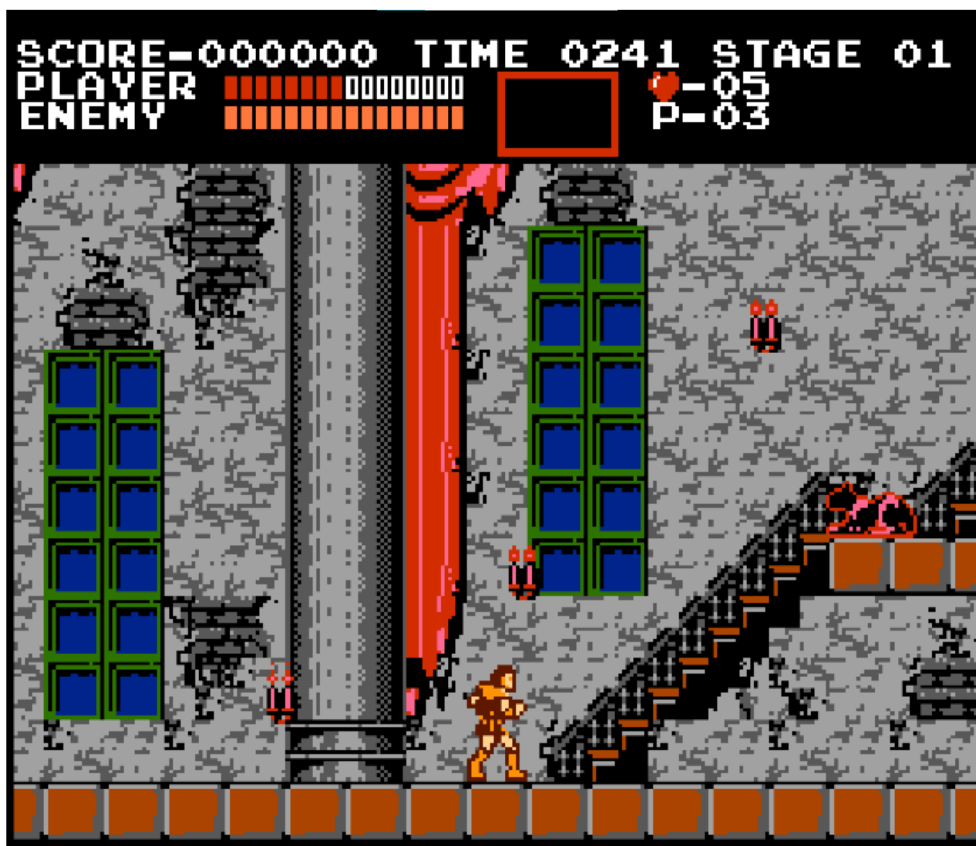
hráča posúvajú daným smerom zvyčajne k pasciam. Taktiež rozlišujeme miznúce, pohyblivé alebo odrážacie plošiny, ktoré charakteru umožňujú skočiť na vyššie miesta. [3]

2.3 Znaký plošinových hier

Prvým znakom plošinových hier je interaktívne prostredie. Je to prostredie, ktorého dizajn vo veľkej miere rozhoduje o činnostiach charakteru. Prostredie plošinových hier je nadizajnované tak, aby vytvorilo pre hráča výzvu. Hry obsahujú rôzne pasce ako bodáky, pohyblivé steny alebo kotúčové píly. Za druhý znak je možné považovať pohľad tretej osoby. Väčšina plošinových hier je navrhnutá práve z pohľadu tretej osoby. Takže hráč môže vidieť ovládaný charakter na obrazovke. Hry navrhnuté z pohľadu prvej osoby hráč ovláda z pohľadu charakteru. Tretím znakom je vertikálny a horizontálny pohyb. Väčšina plošinových hier je dvojdimenzionálna a založená na štýle side-scroller, čo znamená, že hráč vidí charakter z boku. Charakter sa hýbe v jednej rovine vertikálne alebo horizontálne. Novodobé plošinové hry sú rozšírené do trojdimenzionálneho priestoru, ktorý umožňuje dynamickejší pohyb a preskúmavanie. Posledným z uvedených znakov je kontrolovateľné skákanie. Hráč dokáže manévrovať s charakterom počas skákania. [1]

2.4 Príklady plošinových hier

Existujú rôzne štýly plošinových hier vyvinutých pre retro až moderné konzoly. Jednou z prvých plošinových hier pre Nintendo konzoly bola Castlevania (1986). Charakterom v tejto 2D plošinovej hre je lovec upírov. Úlohou hráča je preskúmať a loviť upírov v Drakulovom zámku (pozri obr. 1). Hra je rozdelená na 18 levelov, ktoré musí hráč prejsť, aby sa dostal k finálnemu bosovi skôr, než si vyplýtvá životy. Ďalšou známou 2D hrou bola Donkey Kong (1981), pôvodne navrhnutá ako arkádová hra predtým, než vyšla pre konzoly. V hre sa hráč v úlohe Mária snaží vyliezť po rebríkoch na vrchol konštrukcie a zachrániť princeznú. Počas toho opica Donkey Kong hádže na Mária zvrchu sudy. Donkey Kong bola jedna z prvých hier, ktorá zahŕňala skákanie. Veľmi populárnou bola hra Sonic the Hedgehog (1991). Hráč v tejto hre ovláda rýchleho, modrého ježka nazývaného Sonic. Hlavným cieľom hry je zbierať prstene, porážať nepriateľov a dostať sa čo najrýchlejšie na koniec levelu. Ďalšia už 3D plošinová hra je Crash Bandicoot (1996), v ktorej hráč ovláda postavu nazývanú Crash. Hráč má za úlohu prejsť rôzne krajiny a dostať sa k finálnemu bosovi, šialenému vedcovi, ktorý túži ovládnuť svet. Mirror's Edge (2008) je taktiež 3D plošinová hra. Ovláda sa z perspektívy prvej osoby, čo umožňuje hráčovi z pohľadu charakteru zdolávať výšiny budov a prekážky (pozri obr. 2). [1]



Obr. 1: Ukážka z hry Castlevania, vytvorené z [4]



Obr. 2: Ukážka z hry Mirror's Edge

3 Pixelové umenie

Pixelové umenie je technika tvorenia obrázkov a animácií prostredníctvom pixelov. Pixel je najmenšia jednofarebná jednotka v tvare štvorca, ktorú dokáže počítač nakresliť. V pixelových obrázkoch je rozmiestnenie každého pixelu zámerné. Do pixelového umenia nepatrí 3D grafika, vektorová grafika alebo obrázky, ktoré boli nakreslené a importované do počítača. Tie nepatria do pixelového umenia aj napriek tomu, že sú v nich viditeľné jednotlivé pixely. Kvôli limitáciám DPI obrazoviek a veľkosti pamäte kaziet a kompaktných diskov bolo v minulosti pixelové umenie veľmi dôležité pre počítače a digitálne hry. Pri tvorbe pixelového umenia si grafik musí, podobne ako pri starších hardvéroch, limitovať rozlíšenie a farebnú škálu. [5]

3.1 Techniky pixelového umenia

Z dôvodu pevne stanovenej farebnej škály dostupných farieb vznikli rôzne spôsoby ditheringu [6]. Je to technika simulujúca farby alebo tieňovanie. Kreslenie 2 farieb v šachovnicovom vzore je najbežnejšia technika ditheringu. Zmenou hĺbky jednej z farieb vzniká na danom mieste obrázku odtieň.

Technika nazývaná spriting [5] sa používa najmä vo videohrách. Je to vytváranie čiastočne transparentných obrázkov. Tieto obrázky, nazývané sprites, sa používajú pri animáciách charakteru a objektov, ktoré musia byť v popredí. Sprites sa môžu vyskytovať aj v 3D prostredí, kde sú vždy umiestnené čelne ku kamere. V tomto prostredí ich nazývame billboards [7]. Billboarding sa používa kvôli lepšej výkonnosti a estetickosti.

Ďalšou technikou je anti-aliasing [8], ktorá sa využíva najmä pri objektoch na pozadí. Technika spočíva vo vyhladení ostrých prechodov medzi pixelmi tak, aby výsledný objekt hladko splynul s pozadím, a to prostredníctvom príbuzných farebných pixelov. Obrázok tak vyzerá hladšie a detailnejšie. Technika sa využíva hlavne pri zložitejšom pixelovom umení, pomáha zvýrazniť obrysy spritov v popredí. Spolu s ditheringom by sa jednotlivé pixely mali aplikovať výhradne ručne.

Pixelové obrázky sa ukladajú vo formátoch GIF a PNG kvôli ich bezstratovej kompresii. Vďaka tomu obrázky zostanú v pôvodnej kvalite po ich rekonštrukcii z komprimovanej podoby. Pri použití stratového formátu JPEG je obrázok menší, ale počítač vyhladí jeho textúru. Vo vysokých rozlíšeniach nie je zmenu vôbec vidieť, ale pri pixelovom umení sa obrázok rozmaže, pokazí dithering, farby alebo vzniknú artefakty.

Izometrické pixelové umenie [5] znázorňuje projekciu, pri ktorej sa pixelové objekty podobajú 3D objektom. Jednou z klasických hier založených na tomto štýle bola hra Q*Bert (1982). V hre je charakter stvárnený plocho a bloky, po ktorých sa pohybuje, majú dve bočné a vrchnú stranu. Vďaka takémuto pohľadu sa charakter v hre hýbe diagonálne. Pri tvorbe izometrického pixelového umenia sa využíva mriežka, ktorá pomáha udržať správnu perspektívu a náklon. V

pixelovom umení sa ďalej využíva bočná projekcia alebo projekcia zhora dolu. [5, 6, 7, 8]



Obr. 3: Izometrická projekcia v hre Q*Bert, vytvorené z [4]

3.2 Využitia pixelového umenia

V modernom pixelovom umení je obmedzenie farieb, dĺžky animácií a rozlíšenie textúr estetická voľba. Štýl je populárny kvôli nostalgii ku starším hrám a jeho jednoduchosti. Efekt pixelového umenia vynikne viac pri hrách s menej komplikovanými animáciami a uhlami pohľadu kamery. Pre herné štúdiá je využívanie retro a jednoduchého dizajnu hier výhodné pre jeho nízku finančnú náročnosť. Z hľadiska pamäte je pixelové umenie nenáročné a vyzerá dobre na malých obrazovkách s nízkym rozlíšením. Práve preto je populárne v mobilných hrách a na sociálnych platformách, ako je Facebook. Okrem hier sa pixelové umenie zvyklo využívať aj na webových stránkach, ktoré mali GIFy, avatarov alebo samotnú štruktúru stránky nakreslenú v tomto štýle. Ďalej sa pixelové umenie využívalo pri emotikonoch alebo pri počítačových ikonách. [5]

4 Použité technológie

Táto kapitola sa venuje technológiám, ktoré boli využité pri tvorbe platformovej hry. Ako programovací jazyk bol použitý jazyk C# [10]. Godot má svoj vlastný code editor, ale kvôli nízkej podpore jazyka C#, bolo využité externé vývojové prostredie Visual studio [11] od Microsoftu. Na nakreslenie textúr boli využité editory Photoshop a Piskel. Pre verziovanie, rozvrhnutie prvkov a ďalšie úpravy textúr bol použitý webový editor Figma [12] (pozri obr. 4). NAudio [13] knižnica bola využitá na spracovanie importovanej hudby.

4.1 Godot engine

Godot [9] je open source, multiplatformový herný engine, ktorý slúži na vývoj 2D a 3D hier. Jeho prvá verzia vyšla v roku 2014. Pre developerov poskytuje komplexnú sadu nástrojov, ako napríklad zabudovaný code editor, grafický renderovací engine, nástroj na animácie alebo nástroj na prehrávanie audia.

Godot disponuje množstvom prvkov pre podporu 3D grafiky. Patrí sem svetelný a fyzikálny systém alebo materiálové vzťahy (odraz, lom). Taktiež podporuje zložitejšie grafické prvky ako tieňovanie a častice. Godot má špecializovaný 2D engine. Takže efektívne zvláda nielen 2D backend výpočty, ale aj pixelové jednotky. Pre podporu 2D grafiky Godot disponuje nástrojmi, ako tilemap editor, 2D fyzikálnym a svetelným systémom. Godot je založený na princípoch objektového programovania. Princípy sú postavené na uzlovej a scénovej štruktúre. V Godote je každý objekt definovaný ako scéna (pozri kapitolu 5.1). Každá scéna môže obsahovať niekoľko uzlov. Engine disponuje stovkami uzlov, ktoré poskytujú funkcionálnu dohodu do hry. Teda detektor kolízií, fyzikálne telesá, animácie alebo zvuky. Godot ako multiplatformový herný engine podporuje Windows, macOS, Linux, iOS alebo Android. Vytvorené hry môžu byť vďaka exportu pre HTML5 hrateľné na webe. Herný engine umožňuje vyvíjať hry v niekoľkých programovacích jazykoch. Primárnym jazykom vyvinutým priamo pre prácu s enginom je GDScript. Godot má k dispozícii aj vizuálny skriptovací nástroj (pozri obr. 3), ktorý veľmi dobre pracuje s uzlovou štruktúrou a pre svoju jednoduchosť je vhodný pre začiatočníkov. Godot taktiež podporuje jazyky C++ a C# [10]. [9]

4.2 C#

C# [10] je objektovo orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft. Jazyk C# kombinuje výpočetnú silu jazyka C++ s ľahkosťou programovania jazyka Visual Basics. C# slúži na vývoj zložitých aplikácií, ktoré bežia na frameworku .NET [10]. Medzi prvky C# patrí automatický garbage collector, ktorý z pamäte odstraňuje nedosiahnuteľné a nepoužívané objekty. Ďalej sú to typy s hodnotou null, ktoré chránia pred objektami bez priradenej pamäte. Tiež rozsiahly prístup detekcie výnimiek a ich ošetrenie. V neposlednom rade lambda výrazy, ktoré podporujú funkcionálne programovanie. [10]

4.3 Visual studio

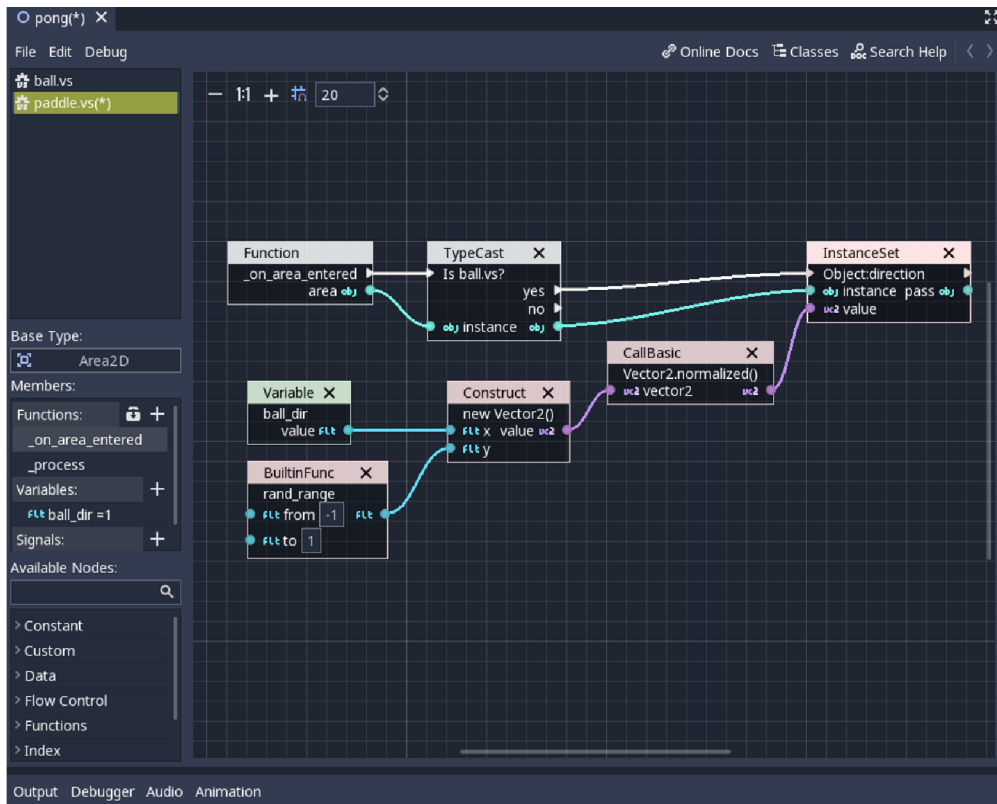
Visual studio [11] je integrované vývojárske prostredie vyvinuté spoločnosťou Microsoft. Je to softwérový program pre developerov, v ktorom môžu písať a editovať svoj kód. Používateľské rozhranie programu Visual studio sa používa na editovanie, debugovanie a zostavenie kódu. Visual studio má code editor, ktorý podporuje IntelliSense a rekonštrukciu kódu. Medzi ďalšie nástroje programu patria nástroje na profilovanie kódu, na tvorbu GUI aplikácií, databázových schém, webový dizajnér a dizajnér tried. [11]

4.4 Figma

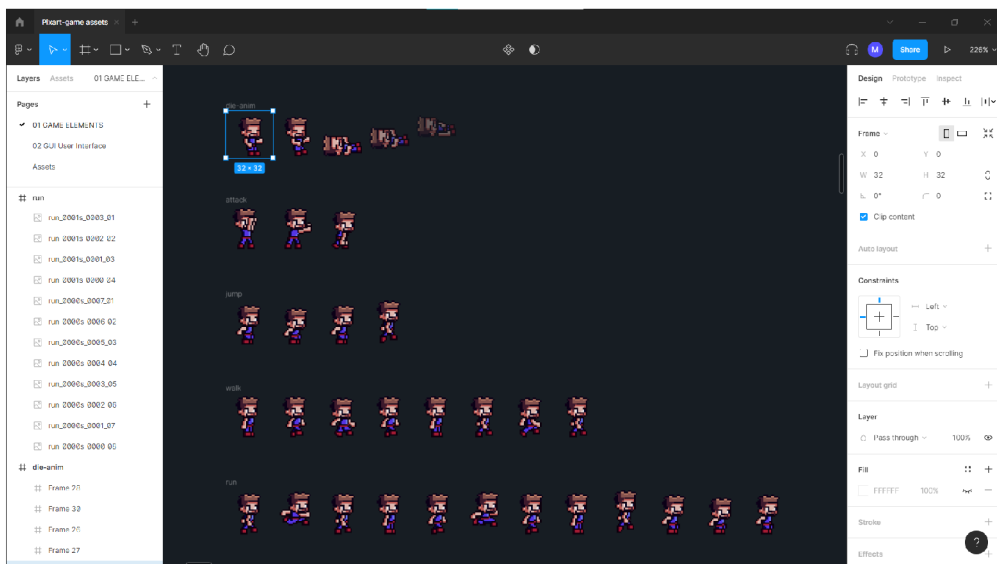
Figma [12] je grafický editovací nástroj. Slúži na tvorbu wireframov, rozhraní mobilných aplikácií alebo prototypov. Od ostatných grafických editorov sa odlišuje tým, že beží priamo na webovom prehliadači. Užívateľ má tak prístup ku svojim projektom z akéhokoľvek počítača. Figma podporuje real-time kolaboráciu, vďaka čomu umožňuje tímu dizajnérov súčasne pracovať a robiť zmeny na projekte. [12]

4.5 NAudio knižnica

NAudio [13] je open source audio knižnica vyvinutá pre platformu .NET [10]. Jej autorom je Mark Heath. Knižnica poskytuje množstvo tried pre prácu s audiom a vývojom audio aplikácií. Dokáže prehrávať audio prostredníctvom rôznych API, čítať audio zo štandardných formátov a miešať alebo manipulovať s audio prúdmi použitím 32-bitového enginu. Ďalej dokáže konvertovať medzi typmi neskomprimovaného audia alebo ho zakódovať. NAudio bolo vytvorené kvôli nízkej podpore audia na platforme .NET 1 a .NET 2. [13]



Obr. 4: Vizuálny skriptovací jazyk Godot engine, prevzaté z [14]



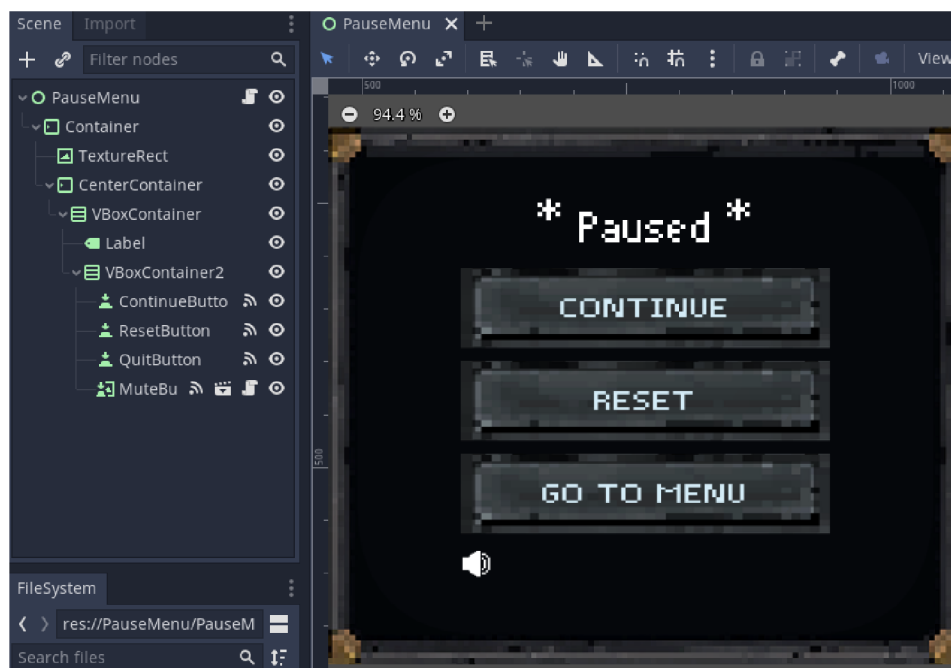
Obr. 5: Editor Figma

5 Programátorská a užívateľská dokumentácia

Táto kapitola sa venuje základným pojmom Godot enginu [9], užívateľskej dokumentácii a skriptom implementovanej plošinovej hry, ktoré sú obsiahnuté v kapitolách menu, editor, pomocné a skripty objektov v leveli.

5.1 Kľúčové pojmy Godotu

Podstatou hier v Godote sú uzly usporiadané v stromovej štruktúre, ktoré spoločne tvoria scény, ktoré sú flexibilné a môžu byť opakovane použité. Scénou môže byť charakter, zbraň, menu v užívateľskom prostredí alebo celý level. Súčasťou jednej scény môžu byť ďalšie scény skladajúce sa z jedného alebo viacerých uzlov. Najmenšie časti hier, uzly, vysielajú počas udalostí signály, umožňujúce komunikáciu medzi jednotlivými uzlami. Singleton je nástroj, ktorý slúži na uchovanie informácií medzi scénami. Pomocou singletonu vznikajú objekty, ktoré sú vždy načítané bez ohľadu na súčasnú scénu. V Godote charakteristiku singletonu simulujú AutoLoad uzly. [15]



Obr. 6: Ukážka scény skladajúcej sa z uzlov a ďalšej scény

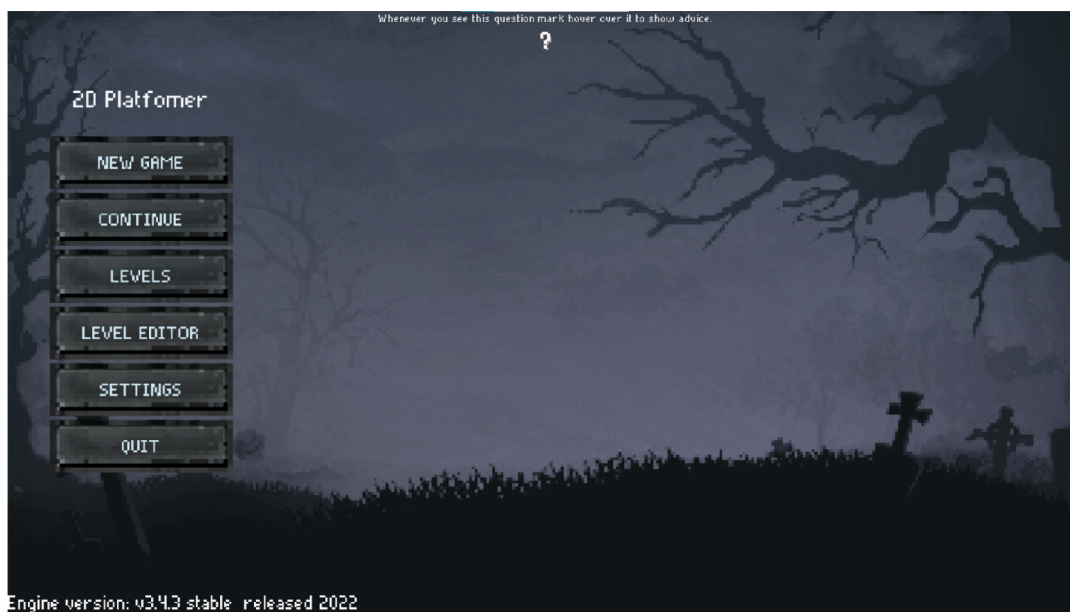
5.2 Užívateľská dokumentácia

Po spustení hry sa zobrazí hlavné menu. Nový hráč má možnosť začať novú hru alebo prejsť do LEVEL EDITORu a vytvoriť si vlastný level z dostupných komponentov. Vytvorený level v editore si môže užívateľ uložiť prostredníctvom

tlačidla PLAY alebo SAVE a pri opätovnom spustení sa v editore automaticky načíta. O uložení levelu v editore informuje ikona, ktorá sa zobrazí v pravom dolnom rohu obrazovky. Pri pohybe kurzora myši nad ikonou otáznika, sa zobrazujú nápovedy. Súčasťou level editora je nápoveda, ktorá informuje o možnostiach ovládania myšou a o polohovaní objektov.

Možnosť LEVELS obsahuje výber levelov, ktoré sú pri novej hre zamknuté. Levely sa postupne odomykajú po prvom úspešnom pokuse. Hráč sa môže kedykoľvek počas hrania vrátiť do hlavného menu a v možnosti SETTINGS nastaviť ovládanie charakteru, hlasitosť hudby a rozlíšenie. Hra ponúka možnosť importovať si vlastnú hudbu formátu .wav, .ogg alebo .mp3. Po úprave nastavení sa hráč môže vrátiť k poslednému hranému levelu cez možnosť CONTINUE.

Hra obsahuje nápovedy, ktoré sprevádzajú hráča v menu a v samotných leveloch. Cieľom každého levelu je dostať sa k vyznačenému koncu. Hráč má na zdolanie levelu neobmedzený čas a množstvo pokusov. Počítadlo pokusov a časomiera sa nachádzajú v ľavom hornom rohu. Hráč sa môže kedykoľvek k prejdеныm levelom vrátiť v možnosti LEVELS, zopakovať ich a zlepšiť si svoj čas prechodu levelom. V leveloch sa ukrýva množstvo nástrah, ktoré hráč musí zdolať použitím logiky a správnym načasovaním pohybov. V hre sa nachádzajú špeciálne objekty, ktoré sú transparentné. Charakter s týmito objektmi interaguje pomocou špeciálnej schopnosti, ktorou ich dokáže podľa potreby zhmotniť. V hre je dostupných sedem levelov, po ich prejdení sa užívateľovi na záver zobrazí obrazovka s poďakovaním a titulkami.



Obr. 7: Hlavné menu implementovanej hry

5.3 Pomocné skripty

Pomocné skripty slúžia na uchovanie parametrov medzi scénami, obsahujú defaultné hodnoty alebo slúžia na prácu so súbormi. Ich úlohou je ukladať, načítavať a vytvárať súbory s dátami hry.

Globals.cs slúži ako singleton (pozri kapitolu 5.1). Uchováva premenné, ktorých hodnota má zostať zachovaná počas prechodu medzi scénami. Je to napríklad premenná `PlayerActionsInitialized`, ktorá pri každom zobrazení hlavného menu udáva, či sa majú z konfiguračného súboru načítať hodnoty, ktorými sa ovláda charakter.

ConfigINI.cs, ak neexistuje konfiguračný súbor s uloženými nastaveniami hry, vytvára nový, do ktorého vloží základné hodnoty nastavení zo skriptu `DefaultSettings.cs`. Taktiež obsahuje funkcie pre načítanie a ukladanie hodnôt do tohto súboru. Hodnoty do súboru ukladá zašifrované. Pri spustení skriptu si súbor načíta a rozšifruje.

DefaultSettings.cs obsahuje základné hodnoty pre nastavenie hudby, grafiky a ovládanie charakteru. Disponuje aj hodnotami, ktoré udávajú, či bola hra spustená po prvýkrát alebo bola prejdená.

LevelSaveSystem.cs, ak neexistuje súbor s uloženými dátami levelov, vytvorí nový so základnými hodnotami. Dáta každého levelu sú v súbore uložené v slovníkovej dátovej štruktúre. Skript obsahuje funkcie pre načítavanie a ukladanie týchto hodnôt. Taktiež disponuje funkciou pre zresetovanie hodnôt, to znamená, že ich v súbore prepíše základnými hodnotami. Ďalej obsahuje funkciu pre nájdenie posledného neprejdeného levelu. Rovnako ako skript `ConfigINI.cs` ukladá hodnoty levelov do súboru zašifrované a pri jeho načítaní súbor rozšifruje.

AudioPlayer.cs spúšťa hudbu a zvuky animácií v hre. Disponuje aj funkciou prehrávajúcu importovanú hudbu, pri ktorej kontroluje, či je súbor formátu `.wav`, `.ogg` alebo `mp3`. Pri skončení prehrávania prehrá nasledujúci súbor príslušného formátu vo zvolenom priečinku.

5.4 Menu skripty

Menu skripty sú všetky skripty, ktoré sa starajú o navigáciu v menu a nastaveniach a dohliadajú na zmeny parametrov hry, ako rozlíšenie, hlasitosť audia a ovládanie.

Menu.cs sa stará o navigáciu v hlavnom menu hry. Pri spustení hry načíta uložené ovládanie charakteru a rozlíšenie okna hry. Ďalej spúšťa audio pri zapnutí hry. V prípade uloženého progresu hry zobrazí okno s upozornením, že sa progres vymaže po stlačení tlačidla `NEW GAME`. Po stlačení tlačidla `CONTINUE` načíta posledný neprejdený level.

Options.cs sa stará o navigáciu v nastaveniach hry. Prepína medzi grafickými, zvukovými a ovladacími nastaveniami.

Gameplay.cs nastavuje a ukladá hodnoty klávesových znakov viazané na ovládanie charakteru. Po kliknutí na tlačidlo, ktoré reprezentuje uložený klávesový znak sa zobrazí okno pre nastavenie nového znaku. Tlačidlá generuje podľa počtu akcií ovládania uložených v `DefaultSettings.cs`. Skript disponuje funkciou na zresetovanie nastavenia ovládania na pôvodné.

KeySelectPopUp.cs predstavuje okno, ktoré sa otvorí po kliknutí na hodnotu klávesového znaku v `GAMEPLAY` nastaveniach. Skript obsahuje funkciu, ktorá kontroluje, či je možné klávesový znak prenastaviť na novú hodnotu zadanú z klávesnicového vstupu. Ďalej hodnotu vstupu preloží na jeho textový názov a vyšle signál skriptu `Gameplay.cs`, aby danú hodnotu nastavil pre akciu, ku ktorej sa viaže.

Sound.cs nastavuje a ukladá hlasitosť hudby a zvuky animácií charakteru. Skript pri stlačení tlačidla `IMPORT` otvorí správcu súborov, kde má užívateľ možnosť vybrať si vlastnú hudbu. Obsahuje funkciu, ktorá umožňuje zresetovanie nastavení hlasitosti hudby a zvukov na jej východiskové parametre. Taktiež zmení importovanú hudbu na pôvodnú. Ďalej obsahuje funkciu na vypnutie hudby (`mute button`).

Video.cs nastavuje a ukladá režim okna hry. Pri vypnutí `fullscreen` režimu umožňuje nastaviť veľkosť okna. Ďalej nastavuje a ukladá vertikálnu synchronizáciu.

Levels.cs zobrazí prehľad levelov hry vo forme tlačidiel. Zo súboru načíta údaje o dokončení levelu. Ak sa do levelu hráč nedostal, dané tlačidlo uzamkne. Po presunutí kurzoru myši nad tlačidlo zobrazí štatistické údaje o leveli. Po kliknutí na tlačidlo spustí level, ktorý reprezentuje.

InfoButton.cs zobrazuje nápovedy, ktoré sa ukážu pri presune kurzoru myši na tento objekt.

EngineVersion.cs zobrazuje v hlavnom menu verziu Godot enginu použitú pri tvorbe hry. Taktiež obsahuje funkciu, ktorá po desiatich kliknutiach ľavým tlačidlom myši na tento objekt odomkne všetky levely.

5.5 Editor skripty

Editor skripty sa starajú o navigáciu v level editore. Tieto skripty disponujú funkciami pre vytvorenie vlastného jednoduchého levelu z dostupných komponentov z výberového okna.

LevelEditor.cs zahŕňa funkciu pre spustenie vytvoreného levelu, ktorý sa spustí pri stlačení tlačidla `PLAY`. Pokiaľ v leveli nie je umiestnený charakter zobrazí pri stlačení tlačidla upozornenie s výzvou o jeho umiestnenie. Skript

taktiež disponuje funkciou pre uloženie levelu, ktorý ukladá, pokiaľ v ňom nastala zmena od jeho posledného uloženia. Level uloží pri stlačení tlačidla PLAY alebo SAVE. Ďalej skript obsahuje funkciu pre návrat do hlavného menu a funkciu pre vynulovanie levelu.

EditorLevelLogic.cs je upravená a zjednodušená verzia LevelLogic.cs (pozri kapitolu 5.6), ktorá sa stará o chod a logiku levelu v editore.

EditorObject.cs obsahuje funkcie pre pokladanie objektov a blokov. Taktiež disponuje funkciami, ktoré zamedzujú prekrývaniu objektov, objektov s blokmi alebo objektov s GUI. Ďalej obsahuje funkcie pre odstránenie objektov a blokov.

Item.cs predstavuje objekt z výberového okna, ktorý sa po kliku nastaví do objektu EditorObject.

SaveIcon.cs je ikona, ktorá informuje o uložení levelu v editore. Ikona sa zobrazuje po dobu troch sekúnd.

5.6 Skripty objektov v leveloch

Skripty predstavujú objekty, ktoré sa nachádzajú v samotných leveloch alebo spravujú jeho funkčnosť. Sú to pasce, platformy, nápovedy, charakter, pozastavovacie okno a z objektov, ktoré spravujú funkčnosť je to štatistika a level logika.

LevelLogic.cs sa stará o prepojenie signálov objektov v leveli. Po prejdení levelu presunie charakter do ďalšieho levelu. Pri neúspešnom pokuse prechodu levelom charakter zabije a level resetuje. Pri neúspešnom aj úspešnom pokuse vysiela signál pre aktualizovanie štatistických dát.

PauseMenu.cs v leveloch zobrazí pause menu a zastaví hru po stlačení klávesového znaku ESCAPE. Má na starosť navigáciu, ktorá umožňuje návrat do hlavného menu, pokračovanie v leveli alebo jeho resetovanie. Po opätovnom stlačení klávesového znaku ESCAPE sa pause menu zavrie a level spustí.

Stats.cs zobrazuje čas, počet neúspešných pokusov a najlepší čas prechodu levelom. Tieto hodnoty načítava a ukladá do súboru. Hodnotu neúspešných pokusov ukladá pri každej zmene. Pri prvom prechode levelu uloží čas, ako najlepší čas prechodu levelom. Pri opätovnom hraní prejdeného levelu nezapisuje ďalšie neúspešné pokusy. Čas prechodu pri opätovnom prejdení levelu porovná s najlepším časom a pokiaľ je lepší, zapíše ho.

Player.cs disponuje funkciou pre horizontálny pohyb charakteru. Funkcia je implementovaná tak, aby pri pohybe charakter nenabral hneď najvyššiu rýchlosť alebo hneď nezastal pri zastavení alebo otočení, čo vytvára efekt šmýkania. Ďalej disponuje funkciou pre skákanie, ktorá okrem klasického

skákania umožňuje charakteru skočiť v časovom limite od opustenia platformy. Taktiež obsahuje funkciu, ktorá sa stará o gravitáciu charakteru alebo funkciu, ktorá kontroluje či sa charakter nachádza medzi objektmi alebo vo vnútri objektu. V prípade takéhoto typu kolízie charakter zabije a vyšle signál o zresetovaní levelu.

Lever.cs predstavuje objekt páky, ktorá kontroluje či je v blízkosti hráč. Páka pri kolízii s hráčom spustí pozastavené platformy, ktoré sú prepojené pomocou atribútu `LeverId`.

AdviceStone.cs pri stretnutí s hráčom zobrazí v leveli nápovedu. Pri nápovedách týkajúcich sa ovládania charakteru využíva regulárny výraz nahradzujúci akciu ohraničenú v znakoch `<`, `>` klávesovým symbolom, ktorý je pre akciu uložený v konfiguračnom súbore.

RollingStone.cs predstavuje objekt kotúlajúceho sa balvanu. Obsahuje funkciu pre horizontálny pohyb a funkciu gravitácie. Taktiež obsahuje funkciu, ktorá kontroluje či nedošlo ku kolízii s hráčom. Ak áno vyšle signál levelu, že pokus bol neúspešný. Ďalej obsahuje funkciu, ktorá kontroluje, či balvan zmizol za hranice obrazu a vráti ho na štartovaciu pozíciu.

MovingPlatform.cs disponuje funkciou pre stály opakovaný pohyb. Plošina sa podľa konečných súradníc polohy môže pohybovať do všetkých smerov. Disponuje aj nastavením, kde sa na určené miesto premiestni raz a už nevráti. Ďalej disponuje nastavením, v ktorom je plošina vypnutá a treba ju v leveli aktivovať. Taktiež obsahuje funkciu, ktorá zmení platformu na transparentnú a pozastaví ju pri použití schopnosti charakteru.

BouncePlatform.cs predstavuje plošinu, ktorá charakter odrazí pri kolízii s týmto objektom. Skript disponuje hodnotou, ktorá nastavuje intenzitu, v akej sa od nej charakter odrazí. Taktiež, ako objekt `MovingPlatform` obsahuje funkciu, ktorá zmení plošinu na transparentnú a zastaví ju pri použití schopnosti charakteru.

Spikes.cs predstavuje statický objekt vo forme bodákov. Bodáky reagujú na kolíziu s hráčom, pri ktorej vyšlú signál levelu, že pokus bol neúspešný. Rovnako bodáky disponujú funkciou, ktorá reaguje na použitie hráčovej schopnosti.

InvinciblePowerUp.cs predstavuje superschopnosť, ktorá reaguje na kolíziu s charakterom a na krátku chvíľu ho spraví nezraniteľným. Pri kolízii hráča a objektu v leveli zobrazí ikonu s odpočtom, ktorá informuje o trvaní efektu.

6 Štatistika hry

Hlavným cieľom kapitoly bolo porovnať obtiažnosť jednotlivých levelov a zhodnotiť výkon vybraných respondentov. Tabuľka porovnáva hodnoty časov a počet neúspešných pokusov naprieč levelmi. Medzi respondentmi boli skúsenejší, ale aj menej skúsení hráči počítačových hier. Dáta v tabuľkách sú od respondentov, ktorí hru hrali prvýkrát. Spätná väzba respondentov poslúžila k vylepšeniam hry.

Tabuľka 1: Priemerný čas a počet pokusov

Level	Priemerný čas (minúty)	Priemerný počet pokusov
1	0:15	1
2	0:48	11
3	0:43	15
4	1:33	31
5	0:46	25
6	0:53	21
7	0:30	9

Prvý level v hre je úvodný, zoznamuje užívateľa s pohybom a prostredím. Level neobsahuje žiadne nástrahy, preto bol priemerný počet pokusov a čas najnižší. Najťažší bol pre hráčov štvrtý level. Na jeho prejdienie potrebovali v priemere 31 pokusov a 1:33 minút.

Tabuľka 2: Výsledky zo štvrtého levelu

Respondent	1	2	3	4	5	6	7	8	9	10
Čas (minúty)	1:13	1:12	1:50	1:11	1:36	3:32	1:42	0:55	1:03	1:21
Pokusy	47	20	47	34	20	14	12	41	35	29

Tabuľka 2 ukazuje čas a počet pokusov, ktorými respondenti dokončili štvrtý level. Tento bol vyhodnotený ako najťažší, vzhľadom na vysoký počet pokusov potrebných na jeho prejdienie.

Rozloženie obtiažnosti je dôležitou súčasťou dizajnu plošinových hier. Obtiažnosť je tvorená špecifickým rozložením komponentov v leveli a schopnosťami hráča. Schopnosti nadobúda postupne pri prechode levelov. Levely sú vyskladané tak, aby každý obsahoval nový komponent. Hráč sa pri každom, aj nevydarenom pokuse zoznamuje so vzorom levelu, a tým sa zdokonaľuje v načasovaní pohybov. Dá sa teda predpokladať, že pri viacerých odohraniach hry by počet pokusov mal klesať, pretože hráči už budú s levelmi oboznámení.

Záver

Cielom bakalárskej práce bolo zoznámiť sa s herným enginom Godot a jeho prostredníctvom vytvoriť plošinovú počítačovú hru, v ktorej hráč musí naprieč levelmi zdolať rôzne prekážky a úlohy pomocou logického myslenia a správneho načasovania pohybov. Levely implementovanej hry sú zložené z komponentov typických pre plošinové hry, ako napríklad platformy, prekážky, podporovače pohybu a spúšťacie mechanizmy. Pri tvorbe hry nebol využitý jazyk GDScript, ktorý je pre Godot engine primárny. Zvolený bol jazyk C#, pre ktorý bolo nutné použiť externý editor prostredia Visual studio. Godot engine je vhodný pre začínajúcich herných developerov hlavne vďaka jednoduchému prostrediu. Výhodou Godotu v porovnaní s ostatnými hernými enginmi je špecializovaný 2D engine, ktorý bol vhodný pre implementáciu zadanej hry. Engine efektívnejšie spracováva 2D grafiku, fyzikálne výpočty alebo 2D osvetlenie. Prácu s enginom uľahčovala aj dobre spracovaná dokumentácia. Štatistika obsiahnutá v závere práce zistila, že obtiažnosť levelov mohla byť vhodnejšie rozložená od najľahšej po najťažšiu.

Conclusions

The goal of bachelor thesis was to get familiar with Godot game engine and use it to create platformer computer game. Player must overcome diverse types of obstacles and tasks with use of logical thinking and good timing of actions. Levels are composed of components that are typical for platformer games, for example, platforms, obstacles, movement aids and triggers. GDScript is primary language of Godot engine. However language C# with external editor Visual studio was used instead of GDScript to create the computer game. Godot engine is suitable for beginner game developers mainly because of simple environment of engine. Advantage of Godot engine in comparison to other engines is specialised 2D engine which was convenient for the creation of the platformer game. Godot engine process 2D graphics, physical calculations and 2D lightning more effectively. Godot also has outstanding documentation which makes work easier. Statistics contained in the end of the thesis discovered that difficulty of levels could be more suitably arranged from easiest to hardest.

A Obsah priloženého CD/DVD

bin/

Hra spustiteľná priamo z CD/DVD a všetky súbory potrebné k jej spusteniu.

doc/

Text práce vo formáte PDF a všetky súbory potrebné pre vygenerovanie PDF dokumentu.

src/

Kompletné zdrojové kódy k hre.

readme.txt

Návod k spusteniu hry.

Literatúra

- [1] MASTERCLASS. Learn About Platform Game: 7 Examples of Platform Games [online]. 2022-03-02. [cit. 2022-03-08]. Dostupné z: <https://www.masterclass.com/articles/platform-game-explained#7-examples-of-platform-games>
- [2] GILLIAN, Smith; MEE, Cha; WHITEHEAD, Jim. A Framework for Analysis of 2D Platformer Levels [online]. 2008-08-09. [cit. 2022-03-08]. Dostupné z: <https://www.semanticscholar.org/paper/A-framework-for-analysis-of-2D-platformer-levels-Smith-Cha/cd4fd502049b3144ba6eb0a58fb02c6aa2ad044d>
- [3] ID TECH. 10 Types of Platforms in Platform Video Games [online]. 2012-10-22. [cit. 2022-03-08]. Dostupné z: <https://www.idtech.com/blog/10-types-of-platforms-in-platform-video-games>
- [4] WWW.RETROGAMES.CZ. ONLINE muzeum starých videoher [online]. [cit. 2022-03-08]. Dostupné z: <https://www.retrogames.cz/>
- [5] WINKLER, Mary. What is Pixel Art? [online]. 2016-01-13. [cit. 2022-03-15]. Dostupné z: <https://design.tutsplus.com/articles/what-is-pixel-art-cms-21759>
- [6] LUX. Dithering for Pixel Artists [online]. 2020-06-9. [cit. 2022-03-15]. Dostupné z: <https://pixelparmesan.com/dithering-for-pixel-artists/>
- [7] PAYNE, Paul. Billboards [online]. 2013-02-18. [cit. 2022-03-15]. Dostupné z: <https://relativity.net.au/gaming/java/Billboard.html>
- [8] LUX. Anti-Aliasing Fundamentals for Pixel Artists [online]. 2020-09-14. [cit. 2022-03-15]. Dostupné z: <https://pixelparmesan.com/anti-aliasing-fundamentals-for-pixel-artists/>
- [9] SCHARDON, Lindsay. What is Godot? The Free Engine for Making 2D & 3D Games [online]. 2022-02-22. [cit. 2022-03-24]. Dostupné z: <https://gamedevacademy.org/what-is-godot/>
- [10] WAGNER, Bill. A tour of the C# language [online]. 2022-03-18. [cit. 2022-03-24]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [11] INCREDIBUILD. Visual Studio [online]. [cit. 2022-03-24]. Dostupné z: <https://www.incredibuild.com/integrations/visual-studio>
- [12] THEME JUNKIE. What is Figma? (And How to Use Figma for Beginners) [online]. 2020-09-17. [cit. 2022-03-24]. Dostupné z: <https://www.theme-junkie.com/what-is-figma/>
- [13] HEATH, Mark. NAudio [online]. 2021-02-6. [cit. 2022-03-24]. Dostupné z: <https://github.com/naudio/NAudio>

- [14] LINIETSKY, Juan; MANZUR, Ariel. Getting started with Visual Scripting [online]. [cit. 2022-03-24]. Dostupné z: <https://docs.godotengine.org/en/stable/tutorials/scripting>
- [15] LINIETSKY, Juan; MANZUR, Ariel. Overview of Godot's key concepts [online]. [cit. 2022-04-02]. Dostupné z: https://docs.godotengine.org/en/stable/getting_started/introduction