



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

EDITOR HERNÍHO ENGINE PRO ŽÁNŘ SHMUP

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie
Autor práce: **Bc. Stanislav Franc**
Vedoucí práce: Ing. Martin Rozkovec, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

SHMUP GAME EDITOR

Diploma thesis

Study programme: N2612 – Electrical Engineering and Informatics
Study branch: 1802T007 – Information Technology
Author: **Bc. Stanislav Franc**
Supervisor: Ing. Martin Rozkovec, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Stanislav Franc**
Osobní číslo: **M12000201**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Informační technologie**
Název tématu: **Editor herního engine pro žánr SHMUP**
Zadávající katedra: **Ústav informačních technologií a elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Upravte engine vytvořený v rámci magisterského projektu o časovače a spouštěče.
2. Prozkoumejte možnosti propojení XNA game studio s WPF a skriptovacím jazykem LUA.
3. Vytvořte editor hry umožňující vizuální editaci úrovní (vrstvení pozadí, definice prvků hry, editace skriptů, editace spouštěčů a časovačů).
4. Propojte editor s engine a demonstруйте ho na jednoduché hře.

Rozsah grafických prací: Dle potřeby dokumentace
Rozsah pracovní zprávy: cca 40 - 50 stran
Forma zpracování diplomové práce: tištěná/elektronická
Seznam odborné literatury:

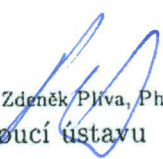
- [1] Reed, A. Learning XNA.. Sebastopol, CA: O'Reilly, c2011, xix, 516p ISBN 14-493-9462-0
- [2] Charles Petzold, Mistrovství ve WPF, Computer Press, EAN: 9788025121412
- [3] Lua: the programming language[online]. 1994-2013 [cit. 9.10.2013], dostupné z <http://www.lua.org>

Vedoucí diplomové práce: Ing. Martin Rozkovec, Ph.D.
Ústav informačních technologií a elektroniky

Datum zadání diplomové práce: 12. září 2013
Termín odevzdání diplomové práce: 16. května 2014


prof. Ing. Václav Kopecký, CSc.
děkan




prof. Ing. Zdeněk Pliva, Ph.D.
vedoucí ústavu

V Liberci dne 12. září 2013

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

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

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce za trpělivost a podporu při tvorbě tohoto díla. Dále bych chtěl poděkovat svému bratrovi za testování editoru, připomínky k jeho funkcím a návrhy na další funkce.

Abstrakt

Cílem práce je vytvořit uživatelsky přátelský editor pro vytváření úrovní v herním enginu Shmup. Práce se zabývá rešerší herního enginu pro hru žánru Shoot 'em up. Tento engine byl vytvořen v rámci magisterského projektu. Jsou zde popsány další úpravy enginu zejména pro podporu spouštěčů a časovačů. V textu se řeší propojení enginu s editorem pomocí jazyka lua. Dále jsou zde popsány jednotlivé části herního editoru. Výsledkem práce je editor rozdělený do několika oken, generující skript lua a připravující obrázky do potřebného formátu. Pomocí tohoto skriptu a obrázkových souborů jsou vytvářeny úrovně v enginu. V tomto editoru jsou schopni pracovat uživatelé bez znalosti programování.

Klíčová Slova

Shmup, Shoot 'em up, střílečka, počítačová hra, herní engine, editor hry

Abstract

The goal of this work is to create a user-friendly editor designed for creating levels for the Shmup game engine. The work refers to the game engine for the Shoot 'em up genre game, that was created earlier within the master project. There are described further additions to the engine, especially support of triggers and timers. The text addresses the connection between the engine and the editor using Lua scripting language. Different parts of the game editor are described as well. The result is a multi-window application, that generates lua scripts and prepares game assets. The editor is targeted on users lacking any programming skills.

Keywords

Shmup, Shoot 'em up, shooter, computer game, game engine, game editor

Obsah

1 Úvod	11
2 Seznámení se s prostředím	12
2.1 Herní platformy.....	12
2.2 Vývojová prostředí.....	12
2.3 Žánr Shoot 'em up.....	12
2.4 Části hry.....	13
2.5 Požadavky na herní engine SHMUP.....	13
3 Principy herního engine SHMUP	15
3.1 Grafika.....	15
3.1.1 Grafické režimy.....	15
3.1.2 Animace.....	16
3.1.3 Pozadí.....	16
3.2 Herní objekty.....	17
3.2.1 Postava hráče.....	17
3.2.2 Nepřátelé.....	18
3.2.3 Kolize.....	19
3.3 Skripty.....	19
3.4 Dodatečné úpravy engine.....	19
3.5 Časový manažer.....	20
3.6 Časovače a spouštěče.....	21
3.6.1 Události.....	21
3.6.2 Podmínky.....	22
3.6.3 Akce.....	22
4 Možnosti propojení enginu s editorem	25
4.1 WPF.....	25
4.2 Lua.....	25
4.3 Vlastní propojení enginu s editorem.....	25
5 Editor herního engine	26
5.1 Požadavky editoru.....	26
5.2 Hlavní okno.....	26

5.2.1 Nabídky.....	26
5.2.2 Zobrazovací část.....	27
5.2.3 Panel nástrojů.....	28
5.3 Editor Animací.....	30
5.4 Editor Pozadí.....	31
5.5 Editor Hráče.....	32
5.6 Editor Trajektorií.....	32
5.7 Editor Objektů.....	34
5.8 Editor Proměnných.....	35
5.9 Editor Zvuků.....	35
5.10 Editor Spínačů.....	36
5.10.1 Hlavní okno.....	36
5.10.2 Vedlejší okna.....	37
5.11 Distribuce.....	39
5.12 Vytvoření hry v editoru.....	39
5.12.1 Příprava herních struktur.....	39
5.12.2 Vytvoření herní scény.....	41
5.12.3 Skriptování.....	41
6 Závěr	43
Seznam použité literatury	44
Přílohy	45
A Obsah Přiloženého CD	45

Seznam Obrázků

Ukázka rozlišení při jiném poměru obrazovek.....	15
Textura animace lodě.....	16
Princip pohybujícího se pozadí.....	17
Catmull-Rom spline.....	18
Kolizní boxy se protínají, ke kolizi nedošlo.....	19
Kolize dvou objektů.....	19
Časový manažer.....	20
Hlavní okno.....	27
Nástroj pozadí.....	28
Nástroj nepřátel.....	29
Editor animací.....	30
Editor pozadí.....	31
Editor hráče.....	32
Editor trajektorií.....	33
Editor nepřátel.....	34
Editor proměnných.....	35
Editor zvuků.....	35
Editor spínačů.....	36
Okno podmínky.....	37
Okno hodnoty.....	38
Okno pozice.....	38
Trajektorie topíra.....	40

Seznam tabulek

Tabulka událostí.....	21
Tabulka podmínek.....	22
Tabulka akcí.....	23

Seznam zkratek

ESA	Entertainment software association
2D	Dvourozměrný prostor
3D	Třírozměrný prostor
NES	Nintendo Entertainment System
Alpha	Průhlednost obrázku
lua	Skriptovací jazyk
Pixel	Obrazový bod
JPG	Obrázek komprimovaný metodou JPEG
PNG	Portable Network Graphics
GIF	Graphics Interchange Format
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language
WAV	Waveform audio file format

1 Úvod

Vývoj počítačových her zabírá značnou část softwarového průmyslu. Podle průzkumu ESA 59% Američanů hraje videohry. V roce 2013 spotřebitelé utratili 21,53 miliard dolarů na videohry, hardware a příslušenství. Průměrný hráč má 31 let a videohry hraje už 14 let. V roce 2013 zisk videoherního průmyslu činil 21 miliard dolarů. [1]

Rozhodl jsem se vytvořit herní engine pro jednoduchou hru typu shoot 'em up. Jedná se o hru, ve které hráč řídí objekt, nejčastěji vojenský dopravní prostředek (např. letadlo, raketu, tank, ale i draka). Kde se sestřelují nepřátelské objekty přilétající nejčastěji ze směru proti pohybu hráče. Existuje spousta lidí co neumí programovat a přesto by si rádi vytvořili nějakou hru. Vzhledem k tomu vytvořím k hernímu enginu editor úrovní navržený tak, aby člověk neznalý programování byl schopen vytvořit vlastní úroveň.

Existují už hotové enginy například Unreal Engine[2] nebo nástroje pro vytváření her například Game Maker Studio[3]. Práce s těmito prostředími však vyžaduje základní znalost programování, nebo jsou příliš komplexní a pro uživatele, nováčka, složité. Vytvořením enginu pro jeden herní žánr lze dosáhnout menší složitosti při vytváření jeho úrovní. Shoot 'em up mají některé věci společné a ty již mohou být připraveny v rámci enginu a editoru.

2 Seznámení se s prostředím

2.1 Herní platformy

Zařízení, na kterých lze hrát videohry nebo počítačové hry, je veliké množství. Některé z nich už patří historii, i když existují vášniví hráči, kteří je stále využívají a k nim příslušné hry stále hrají.

Platformy mají buď zabudovaný displej (např. Gameboy), nebo se připojují k televiznímu přijímači (např. NES, PlayStation, Xbox, Wii). Hry se dají hrát i na přístrojích, které původně nebyly určeny ke hraní her, ale postupně se hry do nich dostaly (osobní počítače, mobilní telefony).

Některé herní platformy a jejich hry byly tak oblíbené, že byly vyvinuty emulátory těchto zařízení pro dnešní počítače. V dnešní době jsou nejrozšířenějšími herními platformami Xbox 360, PlayStation 3, Wii, osobní počítače a mobilní telefony.

2.2 Vývojové prostředí

Chtěl bych se nejvíce soustředit na hry pro osobní počítače s operačním systémem Windows. Pro tuto platformu se hry nejčastěji vyvíjí v programovacím jazyce C++ a to zejména pro jeho rychlost vykonávání kódu. Dále se zde často používá programovací jazyk Java, převážně ve hrách dostupných z webu.

Přímo pro Windows byl vytvořen programovací jazyk C# využívající platformu .NET. Pro vývoj her v tomto jazyce existuje nadstavba s názvem XNA Game Studio, která je určena hlavně pro samostatné vývojáře. Ta má v sobě zabudované řešení některých částí herní problematiky. Například využívá DirectX, grafickou knihovnu, která používá prostředků grafické karty. Má zavedenou základní herní strukturu (načítání obsahu, herní smyčka, smyčka vykreslování) a základní třídy pro řízení a manipulaci hry. Toto vývojové prostředí umožňuje vytvářet hry pro Windows, Xbox 360 a Windows 7 phone.

2.3 Žánr Shoot 'em up

Žánr Shoot 'em up (nebo taky shmup, STG) je podžánr počítačových stříleček. Hráč ovládá jednu postavu, nejčastěji vesmírnou loď nebo letadlo. Bojuje s velkým množstvím nepřátel a snaží se vyhnout jejich útokům. Shoot 'em up obsahuje mnoho poddruhů a kritici se dohadují co přesně definuje tento žánr. Mezi tyto poddruhy patří „fixed shooter“, kde se hráč pohybuje jen po jedné ose, nebo „bullet hell“, ve kterém je téměř celá obrazovka zaplněna střelami.

Hra Space invaders, která vyšla v Japonsku roku 1978, se zasloužila o definování a rozšíření oblíbenosti tohoto žánru. I když byly později vyvíjeny 3D hry, žánr Shoot 'em up zůstal převážně ve 2D prostředí[4].

2.4 Části hry

Každá hra se skládá z několika částí. Hra se neustále musí aktualizovat, tohoto se docílí herní smyčkou. Aby provedla nějakou akci, nečeká na vstup od hráče, ale provádí většinu úkonů nezávisle na vstupu hráče (například pohyb nepřátel nebo střel). Souběžně s aktualizováním pozic a stavu všech objektů herní scény, se tyto objekty musí vykreslovat. Tímto hráč získává informace o dění ve scéně a podle toho může reagovat. Mimo vykreslování hráče a nepřátel se většinou na obrazovku vykreslují zbývající životy hráče, popřípadě dosažené skóre.

Vlastní hra se skládá z herního enginu a herních skriptů. Engine je prostředí, které určuje herní pravidla, způsob vykreslování grafiky a další vlastnosti. Hra bývá členěna do úrovní, jenž využívají pravidel enginu a vytváří z nich pro hráče různě složité úkoly. Úroveň je řízena skriptem.

Většina her je vybavena herním menu, ve kterém se určuje, kdy se zahájí vlastní hra a jaká úroveň hry se spustí. Hra se často dá během hraní pozastavit a později pokračovat. V herním menu se většinou nachází různá nastavení např. ovládání hlasitosti, změna grafických režimů, rozložení kláves a podobně.

2.5 Požadavky na herní engine SHMUP

V rámci magisterského projektu byly stanoveny následující požadavky.

Engine má umožňovat běh v několika grafických režimech, přepínání mezi zobrazením na celou obrazovku nebo v okně. Možnost změnit rozlišení obrazovky. Textury se budou načítat z obrazových souborů, včetně alfa kanálu. Tyto textury se využívají jako objekty nebo se z nich vytvoří animace objektů. Jako pozadí se použije pohyblivý obraz. Pozadí může být tvořeno v několika vrstvách.

Avatar hráče se pohybuje po scéně za pomoci kurzorových kláves. Avatar neopustí herní obrazovku a při změně velikosti herní plochy bude stále omezen na tento prostor. Avatar po stisknutí mezerníku vystřelí. Avatar má určité vlastnosti, počet životů, rychlost, rychlost střel a množství poškození střel. Na obrazovce se budou zobrazovat aktuální životy hráče.

Nepřátelé se pohybují po předem určených trajektoriích. Při kolizi s hráčem se navzájem poškodí. Nepřátelé střílejí a tím se pokoušejí poškodit hráčova avatara. Různé

druhy nepřátel se na určité pozici objevují a to v závislosti na čase. Pozice, čas a vlastnosti nepřítele nejsou pevně určeny kódem enginu.

Průběh hry může být doprovázen zvukovými efekty a hudbou. Textury a vlastnosti hráče, nepřátel a pozadí se načítají podle externího skriptu. Engine bude vybaven základním menu.

3 Principy herního engine SHMUP

V následující části jsou popsány principy herního engine, tak jak byl vytvořen v rámci diplomového projektu.

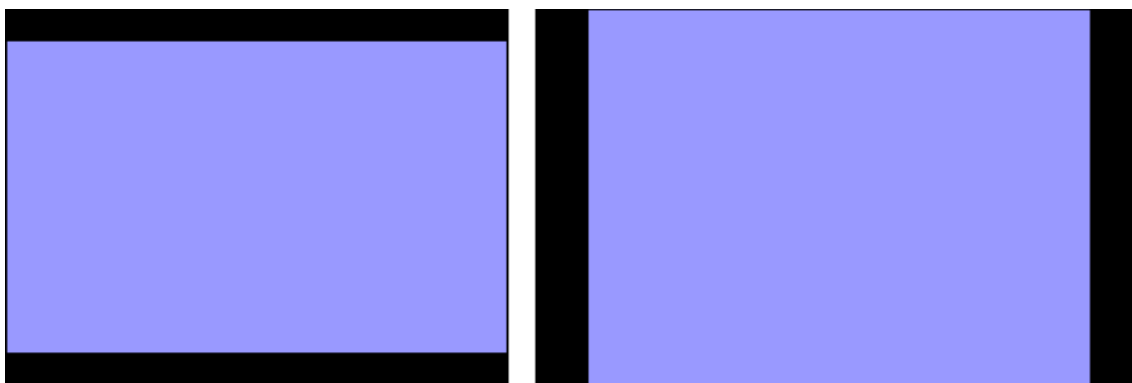
3.1 Grafika

Nedílnou součástí každé hry je grafické prostředí. Soubory s obrazovými daty se musí zpracovávat a odesílat grafické kartě ke zpracování a vykreslení. Ve Windows se grafika ve většině případů zpracovává pomocí rozhraní DirectX. XNA game studio již řeší práci s DirectX a problematiku s ní spojenou.

3.1.1 Grafické režimy

Hra bude mít vždycky nějaké výchozí rozlišení. Do nových rozlišení se vykreslovaný obraz bude muset přepočítávat. XNA už má zabudované metody pro přepínání do režimu celé obrazovky a pro změnu velikosti zobrazované plochy.

Je nutné zajistit, aby se obraz v novém rozlišení roztáhl do plochy a aby se nedeformoval. Nejdříve se zjistí poměr stran výchozího rozlišení. Pokud je poměr stran stejný v novém rozlišení s poměrem ve výchozím rozlišení, zvětšení určíme jako podíl rozměrů původního rozlišení s rozměry aktuálního rozlišení. Podle změny rozlišení se vytvoří matice zvětšení, která se dále použije při vykreslování. Pokud jsou poměry stran různé, obraz se přizpůsobí straně, která má menší podíl původní a aktuální strany. Takto nedojde k deformaci obrazu. Volný prostor se vyplní černou barvou.



Obrázek 1: Ukázka rozlišení při jiném poměru obrazovek

Při změně rozlišení dojde k úpravě rozměrů prostoru, do kterého je možné vykreslovat herní objekty. Aby objekty zaujímaly správnou část herní plochy, musí se před vykreslením změnit jejich rozměry podle předem připravené matice.

3.1.2 Animace

Aby každý objekt více připomínal realitu, měl by se jeho vzhled měnit v čase. Tento jev nazýváme animace. Hráč nabývá dojmu, že objekt opravdu provádí nějakou činnost. Každá animace se skládá z řady po sobě jdoucích snímků, přičemž mezi každým snímkem je prodleva.

Pokud chceme, aby při překrývání dvou obrázků byla některá místa spodního obrázku viditelná skrze horní obrázek, použijeme alfa kanál. Při použití správného formátu obrazu se alfa kanál získá přímo ze souboru s obrazem. XNA podporuje úplnou i částečnou průhlednost[5]. Textury lze získat z obrázků formátu JPG, PNG a GIF.

Pro rychlejší zpracování animace jsou všechny snímky v řadě, případně sloupcích jedné textury. Pro správné vykreslení animace z toho pásu snímků se musí určit rozměry a počet snímků animace. Rychlost přehrávání animace se určuje dobou trvání každého snímku. U animace se určí zda se bude přehrávat neustále dokola nebo se přehraje jen jednou.



Obrázek 2: Textura animace lodě

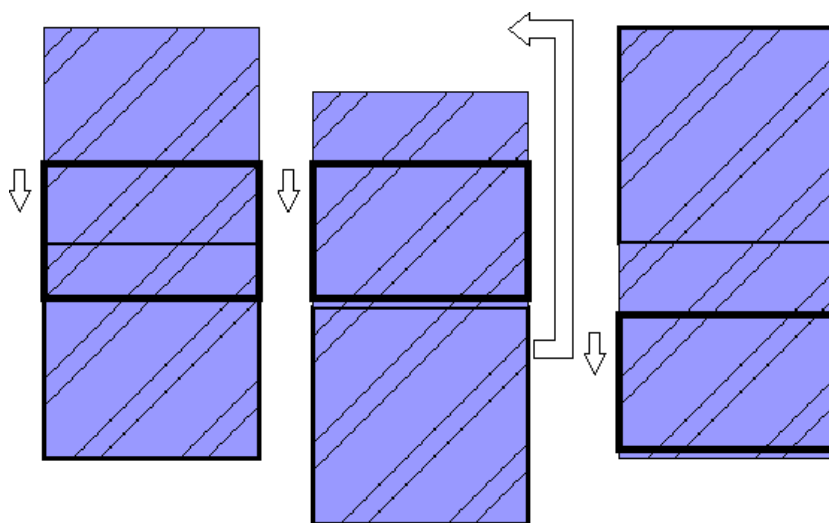
Z výšky, šířky a pořadí snímku se vybere část textury, která se bude vykreslovat na předem určené místo. Po uplynutí času snímku se přejde na další snímek a vykreslí se jiná část textury. Když se přejde k poslednímu snímku, animace se opakuje opět od prvního snímku, nebo se animace označí jako dokončená a přestane se vykreslovat.

3.1.3 Pozadí

Pozadí může být statické, herní objekty se pohybují stále nad jedním obrázkem, který se nehýbe, nebo dynamické, pozadí se pohybuje jedním směrem a vytváří se dojem pohybu hráče nad krajinou. Pozadí se může vytvořit pro celou délku úrovně, nebo je kratší a neustále se opakuje. U opakujícího se pozadí je důležité, aby obrázek po umístění sám na sebe plynule navazoval. Pozadí může být složeno i z několika vrstev, přičemž každá vrstva se může pohybovat jinou rychlostí. Průhlednou částí horní vrstvy je vidět vrstva spodní.

Je-li obrázek s pozadím větší než herní obrazovka, bude stačit vykreslovat pouze dvě instance obrázku. Pokud je pozadí menší, bude třeba tolik obrázků, aby vyplnily

herní plochu, plus jeden navíc. Obrázky pozadí se poskládají k sobě, aby vytvářely celistvou plochu. Všechny obrázky se posouvají určeným směrem a rychlostí. Když se obrázek vysune z obrazovky, posune se na druhou stranu pásu s pozadím.



Obrázek 3: Princip pohybujícího se pozadí

3.2 Herní objekty

Každý herní objekt má nějakou animaci nebo texturu, aby byl ve scéně něčím reprezentován. Musí být zachována informace o jeho pozici, která je s každým kolem herní smyčky předávána jeho příslušné animaci. Objekt může mít více než jednu animaci a tím lze animaci měnit podle stavu objektu. Další z vlastností, které si herní objekt udržuje, je jeho aktuální stav životů, poškození, které udělí při kontaktu s jiným objektem, kolizní obdélník a pixelová mapa kolize.

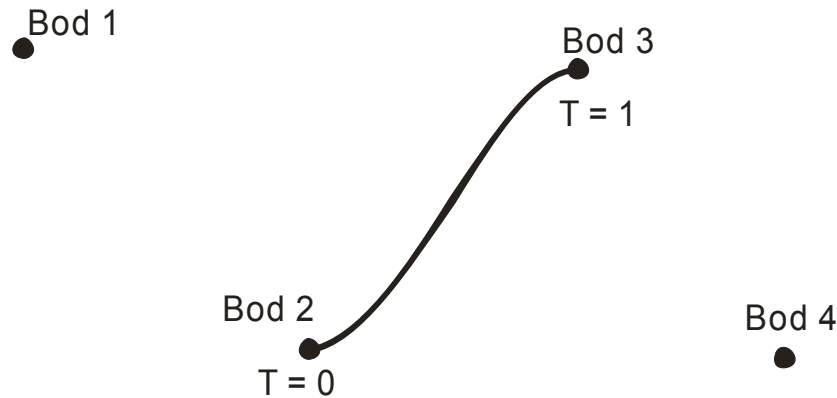
3.2.1 Postava hráče

Postava hráče je odvozena od herního objektu. Oproti hernímu objektu navíc kontroluje stav klávesnice. Pokud je některá ze směrových šipek zmáčknuta, postava hráče se bude pohybovat daným směrem, rychlostí určenou při inicializaci hráče. Když dalším posunem avatar hráče opustí herní obrazovku, jeho pozice se upraví na nejkrajnější pozici v daném směru. Při každém kontaktu s nepřítelem, nebo cizí střelou se hráči odečtou životy.

Objekt hráče si dále uchovává informace o svých střelách. Při stisknutí mezerníku, pokud v předchozím stavu byl mezerník uvolněný, se vytvoří na pozici hráče střela a začne se pohybovat určeným směrem dokud nezasáhne nepřítele, nebo neopustí obrazovku, v takovémto případě střela zaniká.

3.2.2 Nepřátelé

Podobně jako postava hráče jsou nepřátelé odvozeni od herního objektu, ale v případě nepřátel má jiné vlastnosti než objekt hráče. Nepřítel se musí pohybovat a střílet podle předem určených pravidel. Nepřátelé se pohybují po trajektorii vytvořené z Catmull-Rom křivek.



Obrázek 4: Catmull-Rom spline

XNA studio již má hotovou implementaci pro výpočet těchto křivek. Křivka je vytvořena čtyřmi řídicími body. Pro komplexnější trajektorie je potřeba spojit křivky několik. Trajektorie je určena minimálně čtyřmi body. Aktuální pozice bodu na křivce se získá zadáním procenta uražené vzdálenosti od počátku ke konci. Pokud nepřítel opustí herní plochu přepne se do neaktivního stavu.

Je využito tři možností návaznosti trajektorie. V prvním případě se jedná o křivku složenou z několika bodů, křivka vychází z druhého bodu a končí v předposledním. První část křivky je složena z prvních čtyř bodů, další části jsou pak posunuty o jeden bod, tímto se vytvoří jedna celistvá křivka, na konci této křivky se pohyb zastaví.

Druhý případ je podobný prvnímu. Když pohyb dojde na konec křivky, křivka se začne znovu procházet z nového výchozího bodu, který je umístěn na konci předchozí křivky. Tato činnost se neustále opakuje.

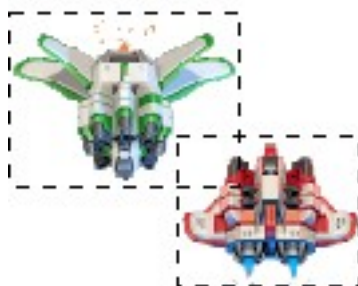
Poslední případ je kružnicová trajektorie. První část křivky se určí jako poslední bod a první tři, poslední část křivky se určuje jako poslední tři body a bod první. Když se dojde nakonec křivky, znovu se pokračuje od začátku.

Nepřítel střílí v určitém intervalu, ten si hlídá podle času posledního výstřelu a aktuálního času hry. Střely nepřítele nevlastní sám nepřítel, ale objekt hry. Tímto se docílí, aby při zničení nepřítele nezanikly i jeho střely.

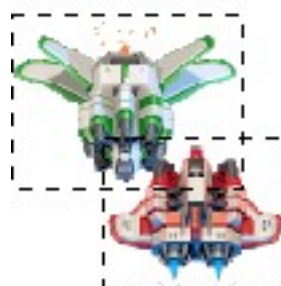
3.2.3 Kolize

Objekty se ohraničí obdélníkem. U dvou obdélníků je jednoduché zjistit jestli se protínají. Je-li vrchní nebo spodní strana prvního obdélníku mezi vrchní a spodní stranou druhého obdélníku a zároveň pravá nebo levá strana prvního obdélníku je mezi pravou a levou stranou druhého obdélníku, tak se tyto obdélníky protínají.

Objekty nemusí mít vždy obdélníkový, nebo přibližně obdélníkový tvar. Může nastat situace, kdy se kolizní boxy střetnou, ale aktuální objekty se ještě nesrazily (viz. Obrázek 5). Pro upřesnění kolize bude třeba provést kontrolu pixel po pixelu. Nastane-li kolize hraničních boxů, můžeme říci, že ke kolizi dvou objektů mohlo dojít v části, kde se obdélníky překrývají. Z obou textur se získá pole pixelů z plochy, ve které se překrývají kolizní boxy. Obě pole mají stejnou délku. Ke kolizi došlo, pokud se při procházení obou polí na stejné pozici ocitnou pixely, které nejsou zcela průhledné[6].



Obrázek 5: Kolizní boxy se protínají, ke kolizi nedošlo



Obrázek 6: Kolize dvou objektů

3.3 Skripty

Úroveň by neměla být natvrdo definována kódem hry, a proto se využívá vnějších skriptů, které ovládají některé prvky hry. Pro tento účel byl zvolen jazyk lua[7]. Pro zprostředkování lua pro C# existuje rozhraní LuaInterface. Podle tutoriálů se propojily vybrané ovládací funkce engineu se lua skriptem v určeném souboru.[8]

Na začátku mise se z lua skriptu načte inicializace hráče, jednotlivé vrstvy pozadí, vytvoří se trajektorie pro nepřátele a inicializují se předlohy nepřátel. Dle hodnoty času se do scény vkládají nepřátelé.

3.4 Dodatečné úpravy engine

Pro zlepšení funkčnosti engineu a rozšíření možností a různorodosti vytváření úrovní bylo nutné provést několik změn v herním engineu. Některé akce, které byly napevno definovány engineem, se předaly pro zpracování spínačům.

Objekty střel hráče byly přesunuty z vlastnictví objektu hráčova avatara do vlastnictví hry. Objekty střel nepřátel byly předdefinovány jako nepřátelé a jejich vytváření bylo předáno spínačům.

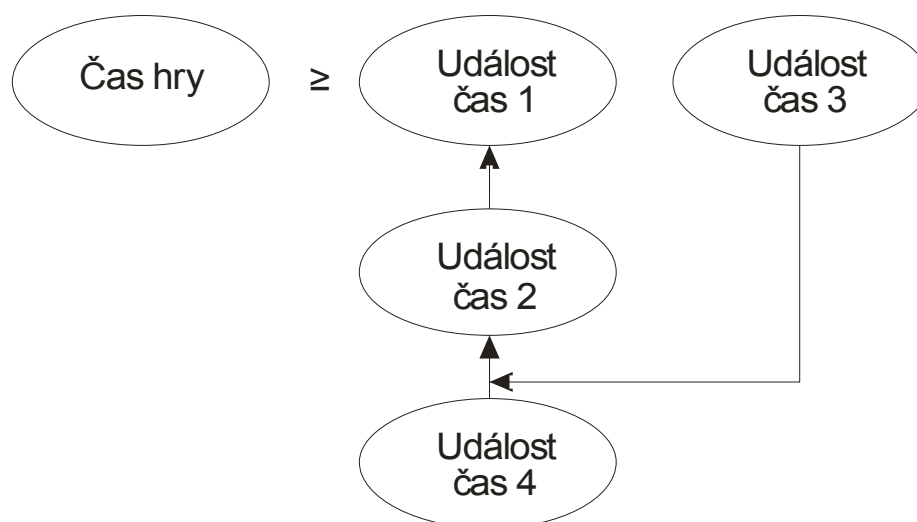
Byly přidány uživatelsky definované celočíselné proměnné a možnost jejich využití. Funkce kolizí mezi hráčem, jeho střelami a nepřáteli zůstala stejná. Ukončení úrovně při smrti hráče bylo předáno spínačům.

3.5 Časový manažer

Časování událostí řídí jeden manažer. Tímto se zajistí rychlejší zpracování oproti možnosti, kde si každý objekt sám hlídal čas. Manažer obsahuje frontu s událostmi, seřazenou vzestupně dle času. Tato fronta se naplní při inicializaci úrovně. Časový manažer řídí několik věcí:

- Čas pro vytváření nepřátel definovaných časem a pozicí.
- Spínače spouštěné časem hry
 - Na začátku úrovně
 - Po uplynutí času
- Spínače spouštěné časem objektů
- Zpoždění akcí spínačů

Při vkládání nových časových událostí se projde fronta a porovnává se čas už vložených událostí. Nová událost se vloží před první událost, která se má vykonat v čase pozdějším než je čas vkládané události. Samotné hlídání času události probíhá při každé aktualizaci hry. Čas první události ve frontě se porovnává s aktuálním herním časem. Pokud je čas události menší nebo roven času hry, provede se vybraná událost.



Obrázek 7: Časový manažer

3.6 Časovače a spouštěče

Časovače a spouštěče se skládají ze tří hlavních částí: událostí, podmínek a akcí. Události spouštějí dané spínače. Podmínky určují, zda se má spínač po příslušné události spustit. Spínač provede nějakou činnost, definovanou soupisem akcí. Spínač může být aktivní od začátku úrovně, nebo jeho spouštění může povolit případně zakázat jiný spínač.

3.6.1 Události

Do implementace enginu jsem zvolil několik spínacích událostí určených pro ovládání úrovně. Jedná se o události závislé na čase, nebo akcí uskutečněných hráčem.

Tabulka 1: Tabulka událostí

Událost	Vstupní parametry	Popis
Začátek úrovně		Spustí spínač na začátku úrovně.
Čas hry	Čas v sekundách	Spustí spínač po určitém čase.
Změna hodnoty proměnné	Proměnná	Spustí spínač při změně proměnné.
Čas objektu	Typ objektu, čas	Spustí spínač po určitém čase od vytvoření objektu, předá referenci na objekt.
Kolize objektu s hráčem	Typ objektu	Spustí spínač po kolizi objektu s hráčem, předá referenci na objekt.
Změna životů objektu	Typ objektu	Spustí spínač při změně životů objektu, předá referenci na objekt.
Změna životů hráče		Spustí spínač při změně životů hráče.
Stisknutí tlačítka střelby		Spustí spínač při zmáčknutí tlačítka střelby.

Časové události, jako je čas hry nebo objektu jsou řízeny manažerem času. Reference na spínače spouštěné objektem se uchovávají u vzorů objektů. Při vytvoření instance daného vzoru se tyto reference předají instanci, při spouštění spínače se předává reference na objekt spínající tuto událost. Pokud se jedná o čas objektu, tak se při vytvoření instance zaregistruje v časovém manažeru událost. Čas této události odpovídá součtu času vytvoření objektu a času provedení události od vytvoření objektu.

U ostatních událostí, jsou ke zdroji této události přidány reference na spínače, které mají být spuštěny. Pokud nastane událost, tak všem referencím na spínače se zašle zpráva, že mají zkontrolovat podmínky a spustit odpovídající akce.

3.6.2 Podmínky

Samotný výskyt události nemusí znamenat, že nastaly všechny uživatelské podmínky pro spuštění požadovaných akcí. Z tohoto důvodu byly přidány struktury na jejich kontrolu. Uživatel může také požadovat logickou operaci mezi několika podmínkami. Z tohoto důvodu jsou podmínky uchovávány ve stromové formě.

Tabulka 2: Tabulka podmínek

Podmínka	Vstupní parametry	Popis
Logická operace	Dvě podmínky, operátor	Provede logickou operaci mezi dvěma podmínkami.
Porovnání herního času	Čas, operátor	Porovná herní čas se zadaným časem, dle vybraného operátoru.
Porovnání dvou hodnot	Dvě hodnoty, operátor	Porovná dvě hodnoty dle zvoleného operátoru.
Porovnání životů objektu	Hodnota, operátor	Porovná životy objektu se zadanou hodnotou, dle vybraného operátoru.
Porovnání času objektu	Čas, operátor	Porovná čas objektu se zadaným časem, dle vybraného operátoru.
Zjištění aktivity objektu		Zkontroluje zda je objekt aktivní, je na obrazovce.
Porovnání životů hráče	Hodnota, operátor	Porovná životy hráče se zadanou hodnotou, dle vybraného operátoru.
Zjištění stavu tlačítka	Logická hodnota	Zjistí zda je nebo není tlačítko střelby stisknuto.

Při porovnávání dvou hodnot, nebo při porovnávání životů se hodnota může získat z několika zdrojů. Hodnota může být získána jako konstanta, proměnná, náhodná hodnota od určeného minima do maxima (včetně), nebo výraz mezi jinými hodnotami. Operace mezi těmito hodnotami mohou být: sčítání, odečítání, násobení nebo celočíselné dělení.

3.6.3 Akce

Uživatel bude chtít vykonávat určitou akci nebo skupinu akcí. U některých akcí bude chtít, aby se vykonaly až po určitém čase po proběhnutí události spouštějící spínač. U těchto akcí se zadává hodnota zpoždění a akce se předá časovému manažeru, který zajistí spuštění akce s daným zpožděním.

Tabulka 3: Tabulka akcí

Akce	Vstupní parametry	Popis
Opožděná akce	Čas, akce	Spustí zadanou akci se zpožděním.
Přehrát zvuk	Zvuk	Jednorázově přehraje zvuk.
Hrát hudbu	Zvuk	Přehraje zvuk ve smyčce, nahradí předchozí zvuk přehrávaný ve smyčce.
Změna životů objektu	Hodnota	Nastaví životy objektu na danou hodnotu.
Změna animace objektu	Animace	Změní animaci objektu, včetně kolizní mapy.
Změna trajektorie objektu	Trajektorie	Změní trajektorii objektu.
Vytvořit objekt	Objekt, pozice	Vytvoří objekt daného typu na určené pozici.
Změna životů hráče	Hodnota	Nastaví životy hráče na zadanou hodnotu.
Změna animace hráče	Animace	Změní animaci hráče.
Vytvoření střely hráče	Objekt, pozice	Na určené pozici vytvoří objekt daného vzoru. Objekt koliduje s nepřáteli.
Nastavení proměnné	Proměnná, hodnota	Nastaví zvolenou proměnnou na danou hodnotu.
Přehrání animace	Animace, pozice	Vytvoří zvolenou animaci na určené pozici.
Ukončení úrovně	Status	Ukončí úroveň vítězstvím nebo prohrou
Spuštění spínače	Spínač	Spustí zadaný spínač s předáním reference na objekt
Povolení spínače	Spínač	Povolí spouštění vybraného spínače
Zakázání spínače	Spínač	Zakáže spouštění vybraného spínače

U přehrání zvuku se ze seznamu zvuků jednorázově přehraje zvukový efekt. Pokud se jedná o hraní hudby tak se vybraný zvuk stále přehrává ve smyčce. Akce změny životů objektu nastaví životy objektu, který vyvolal událost, na určenou hodnotu, u této hodnoty platí stejná pravidla jako u hodnoty popisované v kapitole 3.6.2 Podmínky. Akce změny animace změní animaci objektu, který vyvolal událost, animace se přehrává od začátku. Stejně to tak platí u trajektorie. Podobně to je pak u akcí ovlivňujících hráče.

Při vytváření objektů, nepřátel nebo hráčových střel, lze pozici určit několika způsoby. Zadájí se hodnoty X a Y. Převezme se pozice objektu, nebo hráče. Vytvoří se

nová pozice operací mezi jinými pozicemi. Pozice se mohou sčítat, odčítat, násobit nebo dělit. Tyto operace probíhají mezi hodnotami X nebo Y jednotlivých pozic.

Ze spínače je možné spouštět další spínač, při spuštění se kontrolují podmínky spínače. Dále je možné aktivovat či deaktivovat spouštění spínačů. Pomocí spínačů se i ukončuje úroveň a to s dvěma možnostmi, vítězstvím, nebo prohrou.

4 Možnosti propojení enginu s editorem

4.1 WPF

Windows Presentation Foundation dříve znám jako „Avalon“ je grafický systém od Microsoftu pro vykreslování uživatelského rozhraní aplikací s okny. Je nástupcem Windows Forms. Odděluje grafickou stránku programu od funkční.

Ovládací prvky jsou vytvořeny v XAML, jazyce založeném na XML. WPF využívá pro vykreslování Direct3D a tím pádem je grafika hardwarově akcelerovaná na grafické kartě. WPF je součástí .NET framework 3 a 4. Knihovny pro běh WPF jsou obsaženy od Windows Vista. Pro Windows XP lze potřebné knihovny doinstalovat.[9]

4.2 Lua

Lua je odlehčený procedurální programovací jazyk určený ke skriptování. Jedná se o interpretovaný jazyk. Podporuje boolovské hodnoty, čísla s plovoucí desetinnou čárkou a řetězce. Lze v něm vytvářet pole a asociativní pole. Má automatickou správu paměti a garbage collector. Lua je multiplatformní.

Lua lze snadno vložit do různých programovacích jazyků. Je zdarma. Je používána v mnoha komerčních programech nebo hrách jako je třeba World of Warcraft nebo Angry Birds.[7]

4.3 Vlastní propojení enginu s editorem

V enginu vytvořeném v XNA jsou připravené struktury animací, objektů, trajektorií, spínačů a časovačů. Pomocí knihovny Luainterface pro C# je lua skriptu poskytnuté rozhraní pro ovládání enginu, převážně funkce pro vytváření předloh a instancí herních prvků.

XNA není vhodný pro vytváření editoru, protože má jen omezené možnosti vytváření ovládacích prvků uživatelského prostředí. Oproti tomu WPF je určen pro vytváření aplikací v oknech s podporou vytváření uživatelského prostředí s grafickými ovládacími prvky. Protože WPF využívá Direct3D nebude takový problém s rychlostí vykreslování grafiky jako třeba u WF.

V editoru budou v grafickém prostředí vytvořeny herní objekty a spínače. Tyto data se zapíšou ve formě funkcí do textového souboru s příponou lua. Obrázky se překompilují do formátu PNG. Editor vytvoří složku projektu obsahující obrázky, zvuky a skript. Obsah této složky se nakopíruje do adresáře s enginem, kde k nim bude mít přístup.

5 Editor herního engine

5.1 Požadavky editoru

Editor herního engine by měl být vytvořen v grafickém a uživatelsky přátelském prostředí. Uživateli bude umožněno vytvářet a upravovat vlastnosti animací, hráče, pozadí, objektů, trajektorií, proměnných a podobně.

Práci na rozpracované úrovni by mělo být umožněno přerušit a později na ni opět navázat. Uživatel by měl vidět vzhled připravovaných objektů. V editoru musí být umožněno vygenerovat výstup kompatibilní s herním engine Shmup.

5.2 Hlavní okno

5.2.1 Nabídky

Hlavní okno obsahuje nabídku *Soubor* obsahující další položky. Položka *Nový* vymaže všechna uživatelská data a připraví prázdnou pracovní plochu pro vytváření nového projektu. Položka *Nahrát* otevře dialogové okno, kde je umožněno procházet adresáře a otevírat soubory. Výběr souborů je omezen na soubory s koncovkou *.smp, ve kterých jsou uložena data projektu. Položka *Uložit* uloží rozpracovanou práci otevřeného projektu. Tato položka je aktivní pouze pokud byl projekt nahrán, nebo uložen do určitého souboru za pomoci položky *Uložit jako*. Položka *Uložit jako* otevře dialogové okno pro ukládání souborů. Ukládaný soubor má koncovku *.smp. Poslední položkou je *Generovat skript*, která ve složce projektu vytvoří z dat připravených v editoru skript ve formátu lua.

Při ukládání souborů se vytvoří souborový proud. Aby se omezila šance poškození souboru při ukládání, soubor se nejprve uloží pod dočasným názvem. Pokud při vytváření souboru dojde k chybě, případný původní soubor zůstane zachován. Pokud k chybě nedojde, starý soubor se vymaže a dočasný soubor se přejmenuje. Obrázková data se uloží do příslušného podadresáře adresáře projektu. Zvuková data se zkopírují z pozice odkud byla vybrána. Všechna ostatní textová data se uloží pomocí binárního serializéru, který je součástí knihoven .NET. Nakonec se souborový proud uzavře.

Při nahrávání se z vybraného souboru získají textová data pomocí deserializace. Z tohoto souboru se získají i potřebné cesty k obrazovým a zvukovým datům. Po uložení nebo nahrání souboru se na horní liště hlavního okna objeví cesta k souboru projektu, se kterým se aktuálně pracuje.

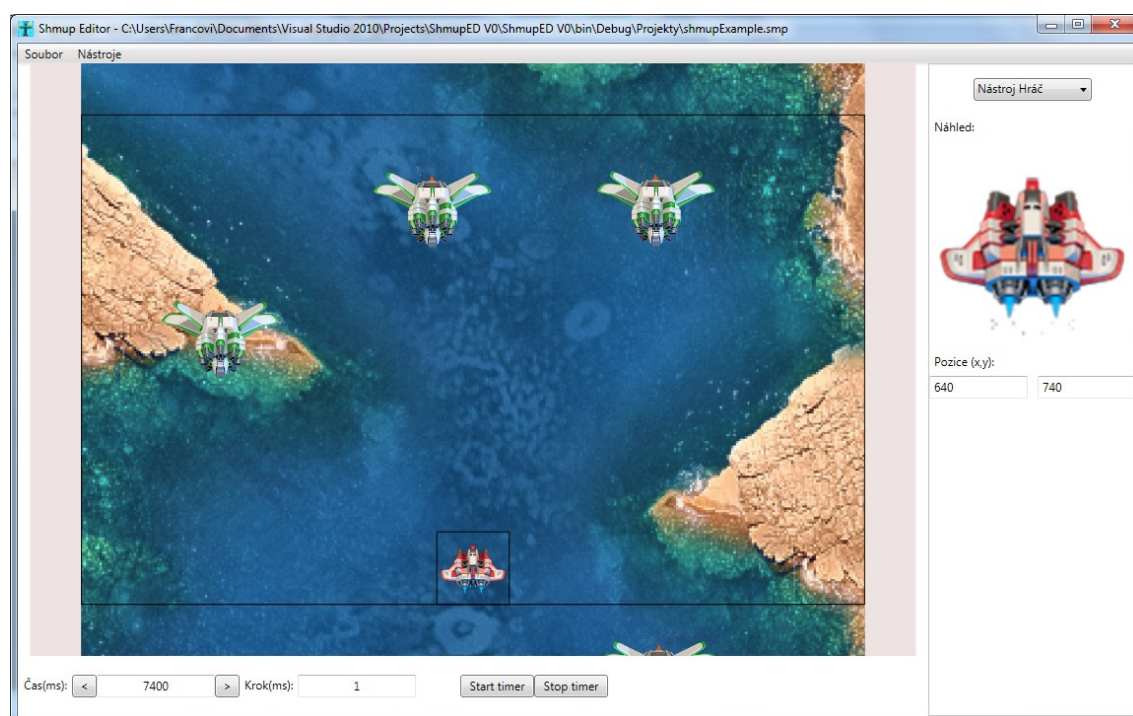
Při generování skriptu se projdou všechny seznamy s připravenými herními strukturami. Zavolá se jejich metoda `toLua`, která z informací v nich uložených vytvoří textový přepis reprezentující kód lua skriptu. Ten se potom řádek po řádku zapisuje do textového souboru s koncovkou `*.lua`.

Další nabídkou jsou *Nástroje*. Zde se nacházejí položky pro otevírání oken na vytváření datových struktur. Jednotlivé funkce těchto oken budou popsány níže. Jedná se o následující položky:

- Editor Animací
- Editor Pozadí
- Editor Hráče
- Editor Trajektorií
- Editor Objektů
- Editor Proměnných
- Editor Zvuků
- Editor Spínačů

5.2.2 Zobrazovací část

Velkou část plochy okna zaplňuje plátno, do kterého se vykresluje vzhled úrovně. Na tomto plátně je vykreslen černý obdélník reprezentující rozměry obrazovky. Celé



Obrázek 8: Hlavní okno

plátno je pak o něco větší. To umožňuje umisťovat objekty, tak aby se objevovaly za okrajem obrazovky. Plátno se přizpůsobuje velikosti okna.

Pod plátnem se nachází ovládání aktuálního času scény. Podle tohoto času se vykresluje stav pozadí a nepřátel. Ovládání času se skládá z textového pole, ve kterém je aktuální čas v milisekundách. Jsou zde dvě tlačítka sloužící k posouvání aktuálního času o krok vpřed nebo vzad. Další textové pole obsahuje informaci o velikosti kroku. Tato informace je opět v milisekundách. Nakonec se zde nacházejí tlačítka pro spuštění a vypnutí časovače. Tento časovač mění hodnotu času při rychlosti padesát snímků za sekundu. Při každé změně hodnoty času se scéna překresluje.

5.2.3 Panel nástrojů

V pravé části okna se nachází přepínací nabídka s panely nástrojů. Vždy je aktivní pouze jeden nástroj. Jedná se o *Nástroj hráče*, *Nástroj pozadí* a *Nástroj nepřátel*.

Nástroj hráče umožňuje určit počáteční pozici hráče. Hráčova pozice se může určit napsáním hodnot X a Y do příslušných textových polí. Další možnost je umístit ukazatel myši na obrázek hráčova avatara na plátně a při stisknutí levém tlačítku myši přemístit ukazatel i s avatarem na zvolenou pozici. V nástroji hráče se zobrazuje náhled vzhledu hráčova avatara.

Další nástroj slouží k umisťování pozadí. Do scény lze umístit libovolné množství pozadí. Zmáčknutím tlačítka *Nové* se vytvoří nová instance pozadí. Tato instance dostane explicitní název, shodující se s vnitřním identifikátorem, který je unikátní v rámci každého projektu. Uživateli je umožněno měnit název viditelný v editoru pro svou vlastní potřebu identifikace. Pod textovým polem pro název se nachází rozbalovací seznam, kde lze vybrat vzor pozadí. Umisťování pozadí se provádí určením referenčního bodu na pozadí a bodu vložení na obrazovce. Referenční bod pozadí je omezen na rohy textury. Podobně jsou omezeny body vložení na rohy obrazovky. Tímto je dosaženo, že lze pozadí umístit



Obrázek 9: Nástroj pozadí

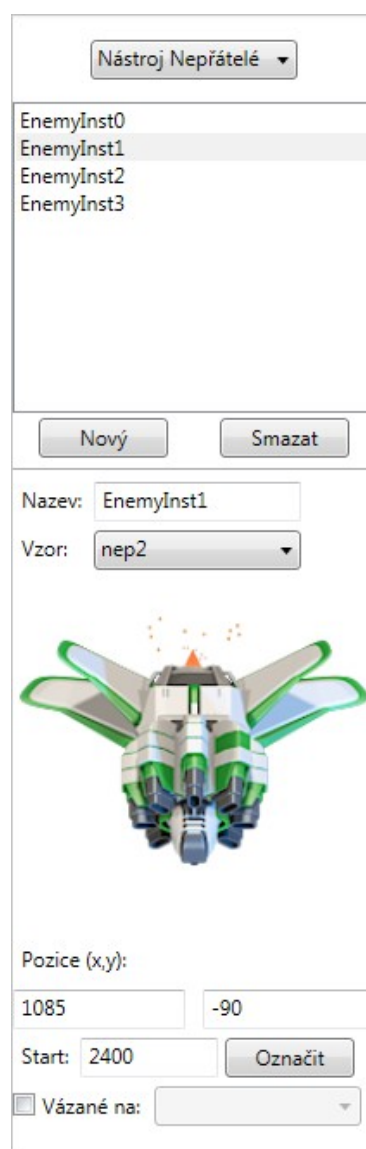
na začátku úrovně zakrýváje obrazovku nebo tak, aby se nové pozadí začalo objevovat od okraje obrazovky.

V jeden okamžik může být ve scéně několik pozadí. Aby uživatel mohl určit, které pozadí bude na vrchu, přiřadí každému pozadí číslo vrstvy. Pozadí s vyšším číslem vrstvy se vykresluje nad pozadím s nižším číslem. Čas objevení pozadí se může napsat do příslušné kolonky, tento údaj je v milisekundách. Pomocí tlačítka lze označit aktuální čas scény jako počáteční čas pozadí. Při vytváření této instance je výchozí hodnota času začátku pozadí určena z aktuálního stavu časovače scény. Poslední určenou hodnotou je koncový čas pozadí. Při překročení tohoto času se pozadí přestane vykreslovat. Opět lze určit tlačítkem, nebo vypsáním hodnoty. Pokud je nastaven na zápornou hodnotu tak se pozadí bude vykreslovat neustále. K vlastnostem umístěných pozadí lze přistupovat vybráním položky v seznamu. Pozadí se vykreslují na plátno reprezentující aktuální stav obrazovky.

Jako poslední je zde nástroj pro umístění nepřátel. Podobně jako u nástroje pro pozadí se tlačítkem *Nový* vytvoří nová instance nepřítele. Uživatel si opět může určit název umístěných objektů. Z rozbalovacího seznamu se vybere vzor nepřítele. Po vybrání vzoru se objeví náhled na vybraný vzor. Uživatel může změnit počáteční pozice změnou textových polí reprezentující souřadnice X a Y, nebo přesunutím objektu na plátně, za pomoci myši.

Startovní čas se určuje stejně jako v případě pozadí, změnou hodnoty, označením, nebo při vytvoření. Instanci nepřítele je možné spojit s určitým pozadím. Toto zajistí správné umístění objektu na pozadí po případných úpravách vlastností příslušného pozadí. Při změně umístění pozadí, nebo rychlosti jeho posunu, se přepočítává startovní čas nepřítele. Pokud změna startovního času nestačí, posune se nepřítel v osách X a Y.

Úpravy nepřítele lze provádět po jeho vybrání. Nepřítele lze vybrat ze seznamu nepřátel, nebo kliknutím myši na obrázek, který jej reprezentuje ve scéně.



Obrázek 10: Nástroj nepřítel

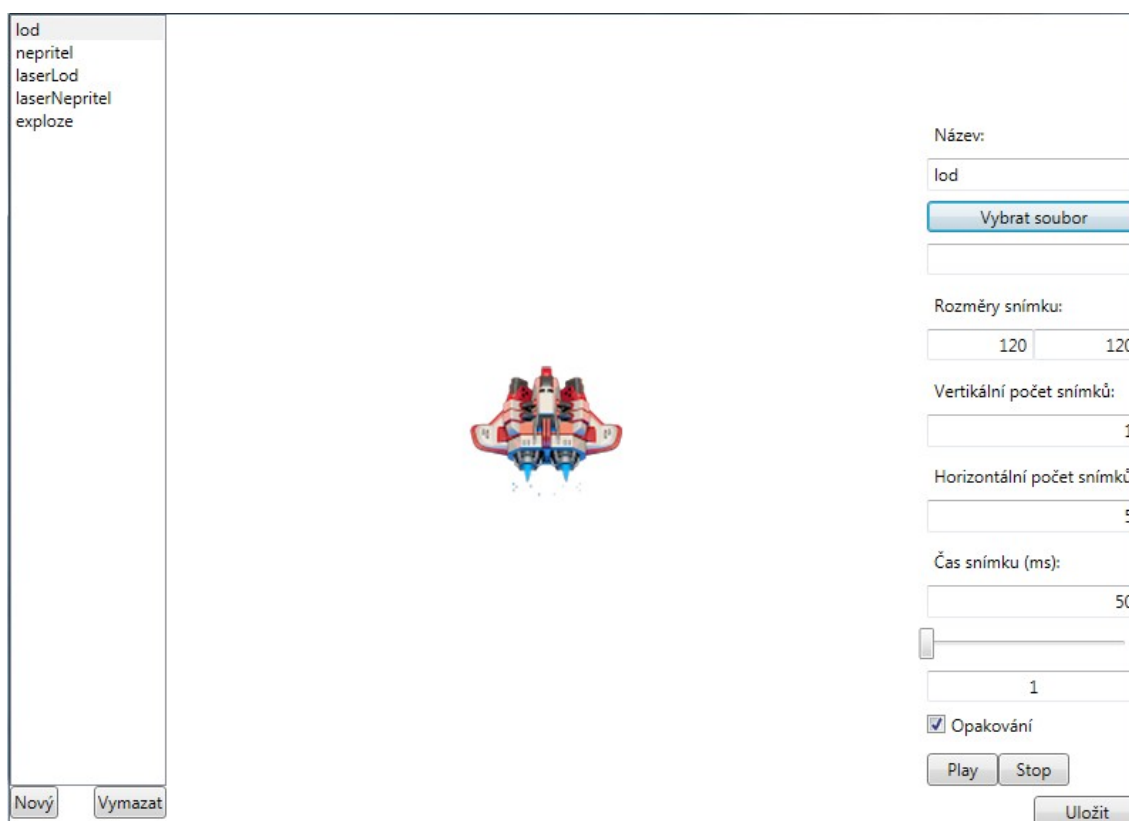
Aktuálně vybraný nepřítel je na plátně ohraničen obdélníkem. Obdélník má černou barvu pokud čas jeho vytvoření odpovídá aktuálnímu času scény, jinak má barvu červenou. U označeného nepřítele se vykresluje i jeho trajektorie, kružnice označuje počáteční pozici.

5.3 Editor Animací

Editor animací má své vlastní okno. V levé části okna se nachází seznam již vytvořených animací. Pod ním se nacházejí tlačítka pro vytváření, nebo případné odstraňování existujících animací. Ve střední části okna se nachází obrázek představující aktuální snímek animace. V pravé části se pak nacházejí prvky určující vlastnosti této animace.

Po vytvoření nové animace se musí vybrat soubor s obrázkem. To se provede tlačítkem *Vybrat Soubor*, otevře se dialogové okno s výběrem souborů. Výběr souborů je omezen na formáty JPG, PNG, GIF. Uživatel může měnit název animace, vnitřní identifikace animace je určena při vytvoření a je unikátní. Zapišou se rozměry jednoho snímku animace a určí se kolik má obrázek s animací sloupců a řádků.

Program kontroluje zda ze zadaných rozměrů a množství snímků lze vytvořit animaci. Pokud animaci lze vytvořit, je možné posouvat s posuvníkem a přepínat zobrazení mezi jednotlivými snímky. Do textového pole *Čas snímku* se zadá prodleva



Obrázek 11: Editor animací

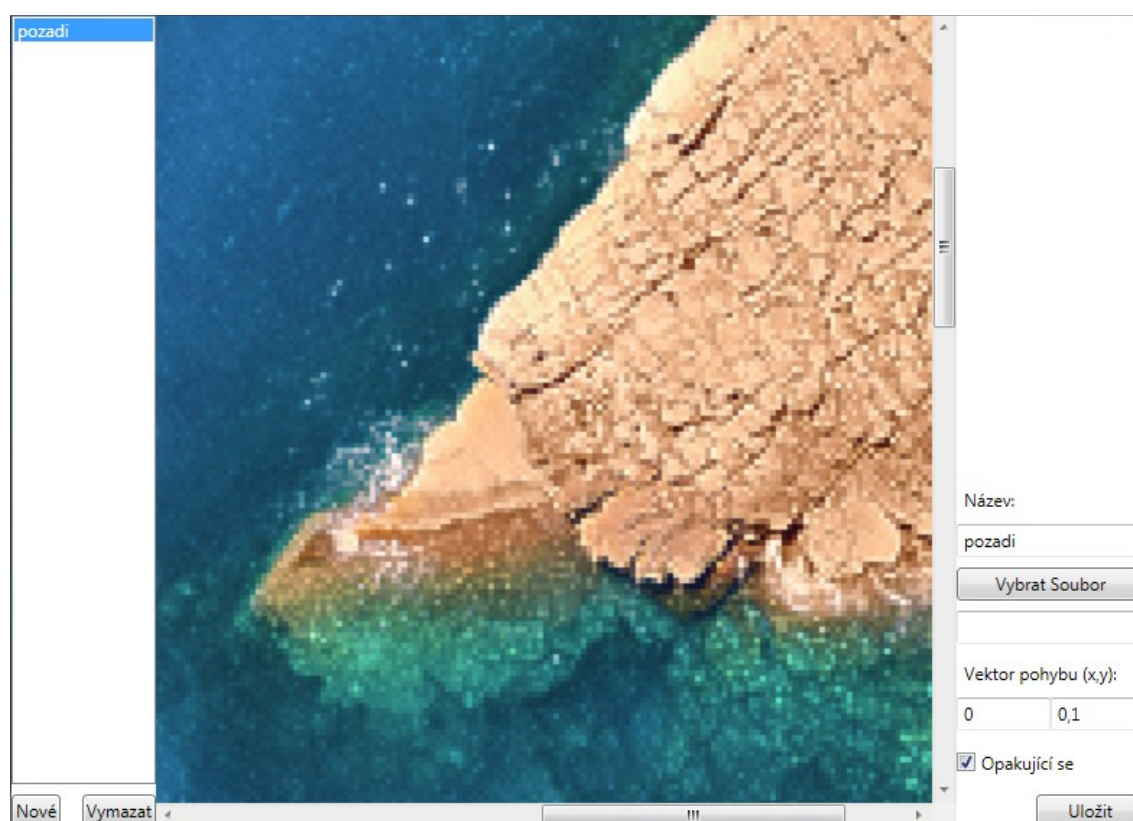
mezi dvěma snímky. Jednorázová animace po přehrání zmizí. Opakovaná se přehrává stále dokola. K určení opakování slouží zaškrtačací políčko. Aby uživatel viděl jak bude výsledná animace vypadat, je zde přidán časovač, který dle zvoleného intervalu mění snímky.

Vlastnosti vytvořené animace se musí tlačítkem *Uložit* zapsat příslušnému objektu. Při vybrání některé animace z levé nabídky se zahodí neuložené změny a zobrazí se údaje z vybrané animace.

5.4 Editor Pozadí

Editor pozadí je vytvořen v podobném stylu jako editor animací. Slouží k přípravě předloh pro pozadí, která budou vykreslována jako prostředí úrovně. V levé části se opět nachází seznam existujících předloh pozadí. Tlačítka *Nové* a *Vymazat* vytváří a mažou předlohy pozadí. Uprostřed se nachází prostor, do kterého se vykresluje náhled vybraného souboru s pozadím. V případě, že se obraz nevejde do okna objeví se posuvníky umožňující prohlédnutí celého obrázku v jeho původní velikosti.

Na pravé straně se nacházejí prvky na určování vlastností pozadí. Jedná se o název, výběr zdrojového obrázku, určení rychlosti posunu a informace zda se pozadí opakuje, nebo se má zobrazit jen jednou. U rychlosti posunu se jedná o počet pixelů za

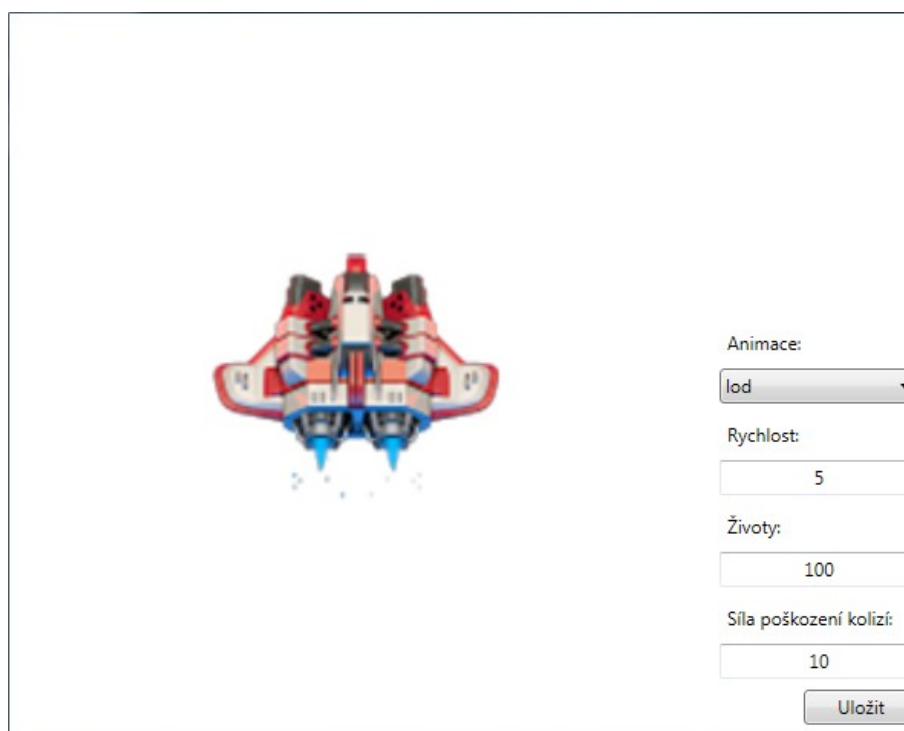


Obrázek 12: Editor pozadí

jednu milisekundu, přičemž toto číslo může být desetinné. Posun se vždy provádí o celé pixely. V případě desetinného čísla se zvětšuje interval mezi posuny.

5.5 Editor Hráče

Editor hráče slouží k určování vlastností avatara hráče. V levé části okna se zobrazuje náhled na zvolenou animaci avatara hráče. V pravé části se nacházejí ovládací prvky. Z vysouvacího seznamu se vybere jedna z animací, která bude sloužit jako reprezentace hráče. Do kolonky *Rychlost* se udává rychlost v počtech pixelů na snímek. Do kolonky *Životy* se vkládá hodnota maximálního počtu životů hráče. Tato hodnota reprezentuje kolik poškození hráč vydrží předtím, než bude jeho postava zničena. Hodnota zadaná do kolonky *Síla poškození kolizí* udává množství životů, kterou nepřátelský objekt ztratí při kolizi s hráčem. Veškeré změny provedené v tomto okně se musí potvrdit tlačítkem *Uložit*.



Obrázek 13: Editor hráče

5.6 Editor Trajektorií

Trajektorie jsou důležitou částí enginu, určují pohyby všech objektů ve scéně, s výjimkou hráčova avatara. V levé části okna se nachází seznam vytvořených trajektorií, pod ním se nacházejí tlačítka na vytváření a mazání trajektorií.

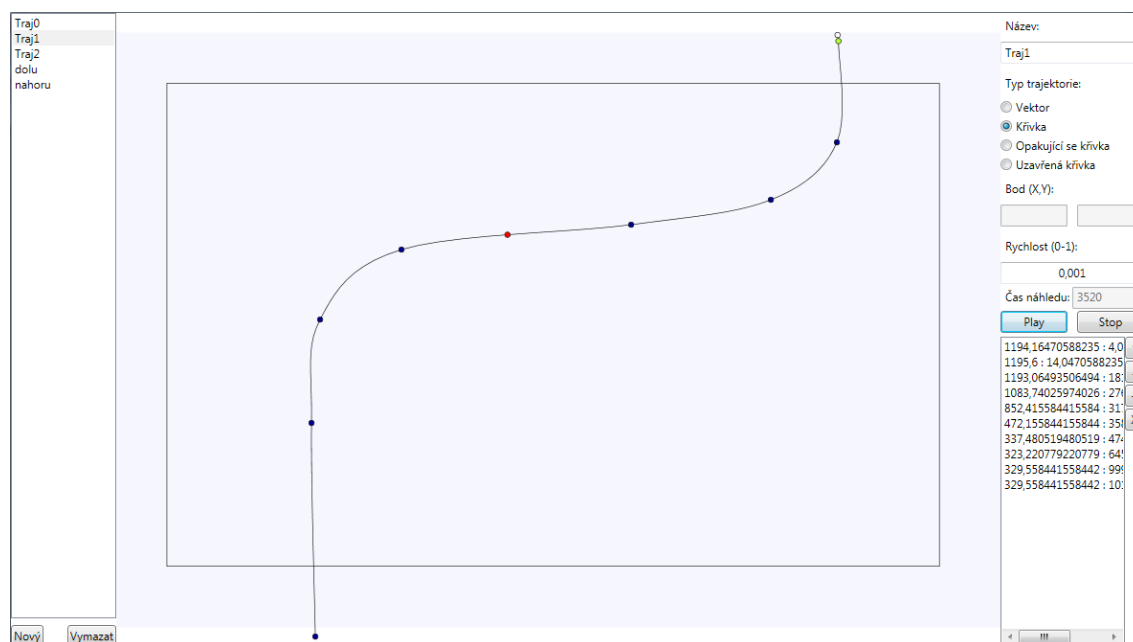
V prostřední části se nachází plátno, do kterého se vykresluje vzhled trajektorie. Je zde nakreslen obdélník reprezentující obrazovku, který pomáhá uživateli určit

velikost trajektorie. V pravé části se nacházejí ovládací prvky pro vytváření trajektorie. V textovém poli lze danou trajektorii pojmenovat, tento název se bude nacházet v místech kde se vybírá trajektorie. Trajektorie může být jedním ze čtyř typů.

- Vektor
- Křivka
- Opakující se křivka
- Uzavřená křivka

Ve spodní části se nachází seznam bodů trajektorie. Vedle tohoto seznamu jsou ovládací prvky umožňující body vytvářet, mazat a měnit jejich pořadí. Pozice vybraného bodu lze měnit v textových polích pro hodnoty X a Y. Hodnota rychlosti určuje jaké procento dráhy mezi dvěma body objekt urazí za dobu jedné milisekundy. Pro sledování této rychlosti je možné zmáčknout tlačítko *Play*. Objeví se červené kolečko, představující aktuální pozici na trajektorii. Pomocí časovače se dle nastavených parametrů terčik pohybuje. V tomto okně se všechny změny ukládají okamžitě.

Mimo použití ovládacího panelu lze křivku vytvářet pomocí kurzoru myši. Na požadované místo na plátně se umístí kurzor myši a dvojitým kliknutím se vytvoří nový bod. Tento bod se vytvoří za bodem označeným v seznamu. Pokud není bod vybrán, tak se vytvoří na konci seznamu. Výběr bodu se provádí kliknutím na vybraný bod. Kliknutím a posunem kurzoru lze body posouvat.



Obrázek 14: Editor trajektorií

Vektor lze určit dvěma body. Výchozí bod je označen zelenožlutou barvou. Koncový bod je označen modrou barvou. Všechny ostatní body jsou označeny průhledným kolečkem a nemají na vektor žádný vliv. V seznamu zůstávají pouze pro případ změny typu trajektorie.

Křivka je určena minimálně čtyřmi body. Přičemž první a poslední bod jsou body řídicí, určují zakřivení. Řídicí body jsou označeny prázdným kolečkem. Počáteční bod křivky je označen žlutozelenou barvou. Body křivky jsou označeny modrou barvou. Opakující se křivka funguje na stejném principu. Jediný rozdíl je, že se tato křivka periodicky opakuje.

Uzavřená křivka vyžaduje minimálně tři body. První bod je zároveň počáteční bod a je označen žlutozelenou barvou. Běžné body křivky jsou označeny modrou barvou. Poslední bod je označen barvou šedivou, aby bylo zřejmé, jak je křivka orientována. Body jsou propojeny do kruhu.

Při vykreslování křivky na plátno se každý úsek křivky rozdělí na deset bodů. Mezi těmito body se nakreslí úsečky. Tak sice nevzniká dokonalá křivka, ale výsledek je pro oko dostačující.

5.7 Editor Objektů

V editoru objektů se vytvářejí objekty, které se budou nacházet na herní scéně. Jedná se převážně o nepřátele, jejich střely a střely hráče. Je velmi podobný editoru



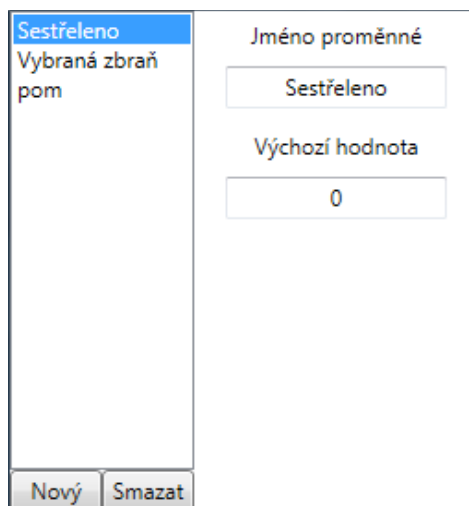
Obrázek 15: Editor nepřátel

hráče. Uprostřed se vykresluje vybraná animace. Na pravé straně jsou ovládací prvky určující název, animaci, životy a sílu kolize objektu.

Na rozdíl od editoru hráče se zde nenastavuje rychlost. Ta je zde nahrazena trajektorií. V levé části se nachází seznam existujících objektů a tlačítka na jejich vytváření a mazání.

5.8 Editor Proměnných

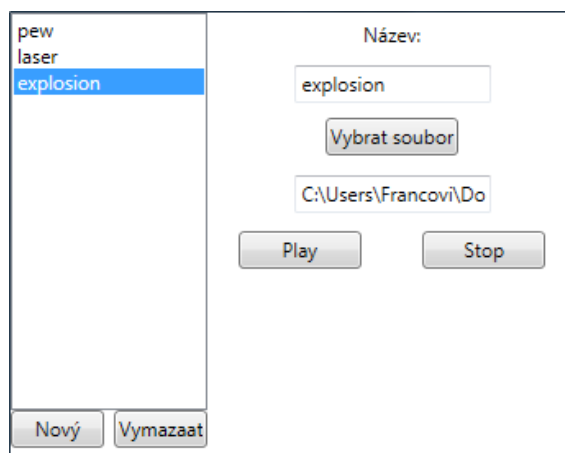
Editor proměnných je velice jednoduchý. Na levé straně se nachází seznam proměnných a tlačítka na jejich vytváření nebo mazání. Na pravé straně se nacházejí jejich vlastnosti. Uživatel zadá jméno proměnné a její výchozí hodnotu. Tato hodnota musí být celé číslo. Změny se zde ukládají okamžitě.



Obrázek 16: Editor proměnných

5.9 Editor Zvuků

Opět se jedná o velmi jednoduchý editor. V levé části je seznam zvuků s tlačítky pro jejich vytváření nebo mazání. V pravé části se pak nachází textové pole na určení



Obrázek 17: Editor zvuků

jména a tlačítko otevírající souborový dialog. Výběr souborů je omezen na formát WAV. Nacházejí se zde tlačítka pro přehrávání vybraného zvuku.

5.10 Editor Spínačů

5.10.1 Hlavní okno

Editor spínačů je jednou z nejrozsáhlejších částí editoru. Zde se vytvářejí události, které ovlivňují tok hry. V levé části se tradičně nachází seznam s vytvořenými spínači a tlačítka pro jejich vytváření a mazání. Pravá část opět slouží k nastavování vlastností spínačů. Spínače se skládají z událostí, podmínky a akcí viz 3.6. Zpožděnou akci nahrazuje akce čekání, ve které se nastavuje zpoždění následujících akcí.

Nachází se zde textové pole pro určení názvu spínače. Zaškrťovací políčko *Zapnuto* určuje zda bude spínač použit. Toto hlavně slouží pro testovací účely. Pro vypnutý spínač se nevygeneruje skript. Zaškrťovací políčko *Na začátku povoleno* určuje zdá má být spínač aktivní od začátku hry, nebo jestli bude čekat až nějaký jiný spínač povolí jeho spouštění. Následuje editovatelný textový blok. Do tohoto bloku může uživatel zapsat své poznámky o účelu konkrétního spínače.

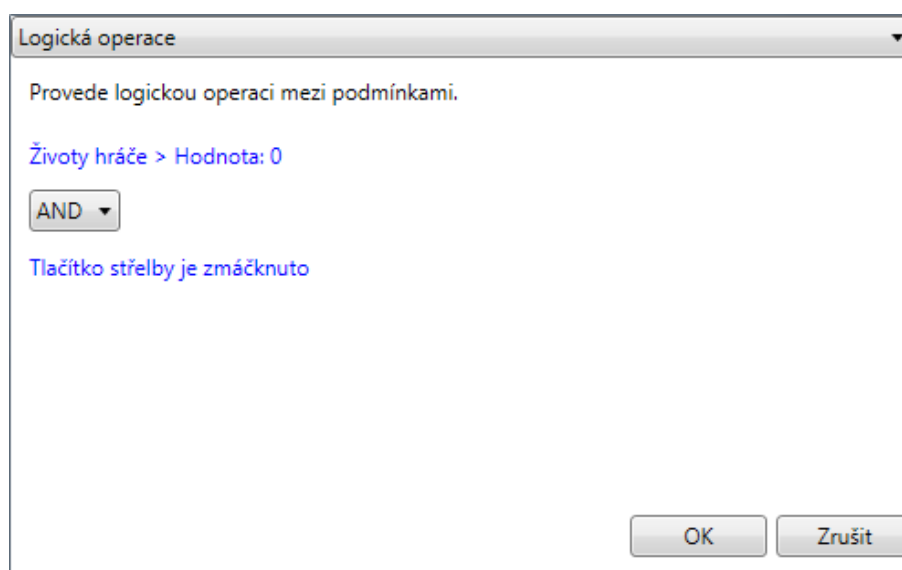
palba pal nep1 nep2 nep3 nepPal bam lost	Název: nepPal ☒ Zapnuto ☒ Na začátku povoleno
	Poznámky k spínači: Všichni nepřátelé typu nep1, nep2 a nep3 budou střílet střelu typu strelaNep, každou sekundu počínaje první sekundou.
	Události: Nová Upravit Vymazat Čas objektu typu nep1 je 0:1(1000ms) Čas objektu typu nep2 je 0:1(1000ms) Čas objektu typu nep3 je 0:1(1000ms)
	Podmínky: Vymazat Objekt spouštějící událost je aktivní
	Akce: Nová Upravit Vymazat Přehraj zvuk pew Vytvoř objekt typu 'strelaNep' na pozici 'Pozice objektu spouštějícího událost' Čekat 0:1(1000ms) Spust' spínač: nepPal
Nový Vymazat	

Obrázek 18: Editor spínačů

Pod poznámkovým blokem se nachází seznam událostí, které spouštějí spínač, tlačítkem Nová se otevře okno pro vytvoření nové události. Tlačítko upravit lze stisknout pouze pokud je vybrána událost. Po jeho stisknutí se otevře okno pro editaci události. Tlačítko smazat vymaže zvolenou událost. Pod listem událostí se nachází textový výraz reprezentující podmínku. Kliknutím na tento text se otevře okno pro úpravu podmínek. Nachází se zde tlačítko nastavující podmínku do takového stavu, ve kterém bude vždy splněna. Dole se nachází seznam s akcemi a tlačítka na jejich vytváření úpravu a mazání.

5.10.2 Vedlejší okna

Události, podmínky a akce se vytvářejí ve vlastních oknech. V horní části okna je rozbalovací seznam, kde se nacházejí dle typu okna všechny události, podmínky nebo akce. Vybráním položky v tomto seznamu se mění spodní část okna, ve které se nacházejí vlastnosti těchto prvků. Pokud se okno otevírá při editaci, v rozbalovacím seznamu je vybrán odpovídající prvek a jsou vyplněny jeho vlastnosti.



Obrázek 19: Okno podmínky

U většiny událostí se nastavuje čas, typ objektu, nebo proměnná. Některé události dokonce nemají žádné parametry. Podmínky se do jisté míry podobají událostem, ale mají přesnější vlastnosti. Při porovnávání hodnot je umožněno vybrat operátor rovnosti nebo nerovnosti. Speciální podmínkou je logická operace mezi dvěma podmínkami. V jejím okně se nachází text reprezentující jednu a druhou podmínku. Kliknutím na tento text se otevře nové okno pro tvorbu podmínek. Tímto lze uspořádat podmínky do stromové struktury. U akcí se většinou přiřazuje hodnota, pozice, nebo některý z objektů hry.

Vyber zdroj hodnoty.

☒ Konstanta: 0

☐ Proměnná: ▼

☐ Náhodná: Min: 0 Max: 0

☐ Výraz: HODNOTA

+ ▼

HODNOTA

OK Zrušit

Obrázek 20: Okno hodnoty

Některé podmínky a akce mají složitější vlastnosti, pro které je vytvořeno vlastní okno. Jedná se o *hodnotu a pozici*. Hodnota lze vzít z konstanty, proměnné, náhodného čísla nebo výrazu mezi dvěma hodnotami. Náhodné číslo je určeno minimem a maximem. Minimum i maximum může být výsledkem hodu. Výraz mezi dvěma hodnotami funguje podobně jako logická operace mezi podmínkami, kliknutím na hodnotu se otevře nové okno pro vytváření hodnoty. Mezi dvěma hodnotami může být operace sčítání, odečítání, násobení nebo dělení. Pozici je možné určit jako hodnoty X a Y. Pozice lze převzít od hráče nebo objektu, který spustil událost. Pozici lze vytvořit i z výrazu mezi dvěma pozicemi. To slouží hlavně k vytváření relativních pozic vzhledem k objektu.

Vyber zdroj pozice.

☒ Nová (x,y): x= Hodnota: 0
y= Hodnota: 0

☐ Pozice hráče

☐ Pozice objektu

☐ Výraz: POZICE

+ ▼

POZICE

(Operátor mezi pozicemi se provádí mezi odpovídajícími hodnotami x a y)

OK Zrušit

Obrázek 21: Okno pozice

5.11 Distribuce

Pro distribuci obou programů, engine a editoru, jsem zvolil Visual Studio Installer Setup project. Výhodou této možnosti je vlastní volba umístění aplikace. Do instalačního balíčku byly vloženy obě aplikace a testovací úroveň. Testovací úroveň je vložena v obou formátech, ve verzi pro editor a pro engine.

Balíček dále obsahuje knihovny pro běh lua skriptu. V balíčku bylo nastaveno, že aplikace vyžaduje *Microsoft .NET Framework 4*, *Microsoft XNA Framework Redistributable 4.0* a *Windows installer 3.1*. Instalátor by měl tyto frameworky doinstalovat, ale ve většině testovacích případech se stalo, že se XNA Framework musel nainstalovat ručně.

5.12 Vytvoření hry v editoru

V následující části bude popsáno vytváření ukázkové úrovně. Tato část textu by také měla sloužit jako pomůcka pro vytváření úrovní v editoru. Jako téma ukázkové úrovně byl zvolen drak bojující s příšerami. Drak bude chrlit oheň. Nepřátel budou dva typy, topír a kobold. Topír bude pouze letět. Kobold bude před sebe střílet ohnivé koule. Drak bude začínat u hnízda a bude letět nad krajinou. Nepřátelé budou mít společnou animaci smrti. Úroveň bude úspěšně dokončena po dvou minutách. Soubory k vytvoření této úrovně jsou přiloženy na CD.

5.12.1 Příprava herních struktur

Před vytvářením samotné úrovně je třeba připravit objekty, které budou umístěny do scény. Začneme s vytvářením animací. Na horní liště v nabídce nástroje vybereme editor animací. Otevře se okno pro vytváření animací. Jako první vytvoříme animaci hráče. Kliknutím na tlačítko nový se vytvoří nosič animace.

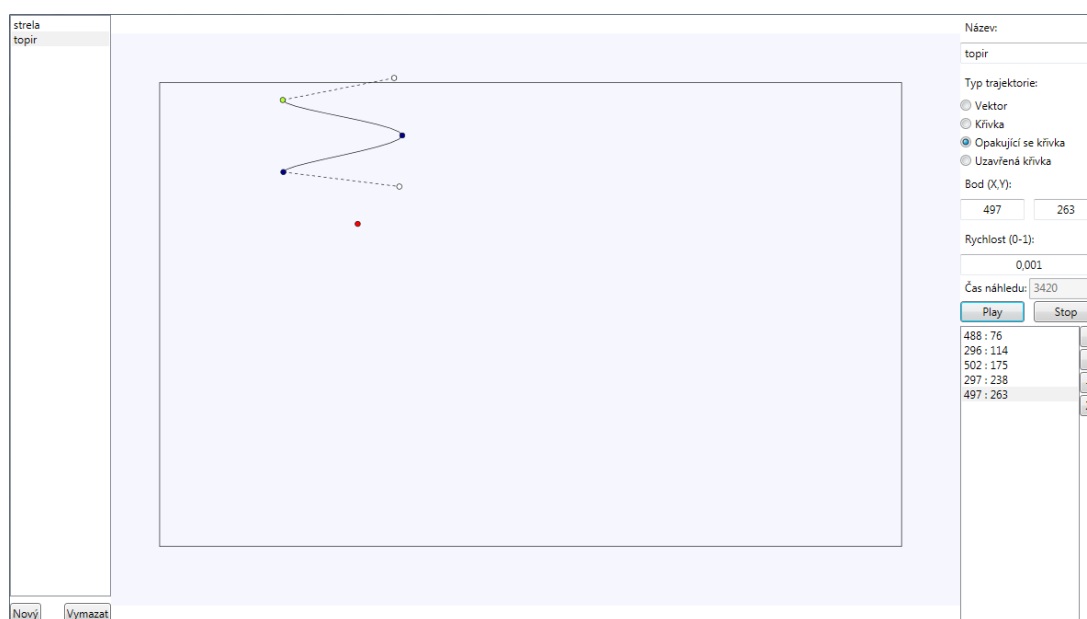
Animaci si pojmenujeme jako *Drak*. Stiskneme tlačítko vybrat soubor. V dialogovém okně nalezneme adresář s obrázky a vybereme soubor *Drak_350x275.png*. Kolonky rozměry snímků vyplníme hodnotami 350 a 275. Vertikální počet snímků nastavíme na 3 a horizontální na 4. Do Času snímku zadáme hodnotu 50 a zaškrtneme políčko opakování. Nyní si tlačítkem play můžeme prohlédnout výslednou animaci. Všechny změny potvrdíme tlačítkem uložit.

Další vytvoříme střelu hráče. Vytvoříme novou animaci, pojmenujeme ji střela. Přidělíme ji obrázek se jménem *Strela_40x45.png*, rozměry jsou obsaženy v názvu souboru, horizontální počet snímků 9 a čas 100 milisekund. Tato animace se bude opakovat. Následují animace nepřátel. Topír má obrázek *Topír_118x79.png*,

7 horizontálních snímků, 75 milisekund mezi snímky a bude se opakovat. Kobold má obrázek *Kobold_87x84.png*, 5 horizontálních snímků, 150 milisekund mezi snímky a bude se opakovat. Koboldova střela má obrázek *koule_30x30.png*, 4 horizontální snímky, 100 milisekund mezi snímky a bude se opakovat. Smrt nepřátel má obrázek *plesk_70x70.png*, 6 horizontálních snímků, 75 milisekund mezi snímky a nebude se opakovat.

Nyní si připravíme vzory pozadí. V nabídce nástroje vybereme položku editor pozadí. Kliknutím na tlačítko nové se vytvoří nosič pozadí. Z každého obrázku ze složky pozadí vytvoříme pozadí. Jedná se o hnízdo, úžinu a krajinu. U všech nastavíme vektor pohybu $X=0$ a $Y=0,2$. Opakování nastavíme jen u krajiny.

Před vytvářením objektů je potřeba připravit trajektorie. Z nástrojů vybereme editor trajektorií. Avatar hráče trajektorii nepotřebuje. Střela hráče se bude pohybovat směrem nahoru. Vytvoříme novou trajektorii, pojmenujeme ji střela a určíme, že se bude jednat o vektor. Tlačítkem plus vytvoříme postupně dva body, první na pozici $X=100$ $Y=100$, druhý s pozicí $X=100$ $Y=0$. Rychlost nastavíme na 0,005. Topír bude kmitat ze strany na stranu a postupovat ke spodní části obrazovky. Trajektorii vytvoříme jako opakující se křivku. Klikáním tlačítka myši na plátno umístíme pět bodů, které budou tvořit tvar periody viz Obrázek 22. Počáteční bod nemusí být umístěn na místě vytvoření objektu. Kobold bude chodit z pravé strany na levou nebo z levé strany na pravou. K tomuto budeme potřebovat dvě trajektorie typu křivka. Rozmístíme dostatek bodů, aby se kobold prošel po scéně a byl schopen odejít za kraj mapy. Koboldova střela bude opět vektor, tentokrát směrem dolů.



Obrázek 22: Trajektorie topíra

Přejdeme k vytváření objektů a začneme avatarem hráče. Z nástrojů vybereme editor hráče. Vybereme animaci drak. Rychlost nastavíme na pět, životy na sto a poškození kolizí na deset. Pro vytváření ostatních objektů musíme otevřít editor objektů. Vytvoříme jeden objekt pro střelu, přiřadíme příslušnou animaci a trajektorii. Počet životů nastavíme na deset a poškození také. To samé provedeme u topíra a střely kobolda. Koboldy vytvoříme dva, každý bude mít jinou trajektorii, jejich životy nastavíme na dvacet a poškození na deset.

5.12.2 Vytvoření herní scény

V hlavním okně na pravé straně vybereme nástroj hráče. Uprostřed plátna by měl být obdélníkem označený avatar hráče. Nastavíme jeho počáteční pozici. Klikneme na hráčova avatara a přesuneme ho do dolní části plátna.

Pravý panel přepneme na nástroj pozadí. Vytvoříme novou instanci pozadí. Jako vzor vybereme hnízdo. Referenční bod a bod vložení nastavíme na pravý dolní roh. Vrstva bude nula, počáteční čas taktéž. Koncový čas necháme na hodnotě mínus jedna. Vytvoříme instanci s pozadím úžiny. Referenční bod nastavíme na levý dolní roh a bod vložení na levý horní roh. Vrstva bude nula, počáteční čas také a koncový čas bude mínus jedna. Nastavíme čas na hodnotu 4650, horní okraj posledního pozadí by měl být na kraji obdélníku reprezentujícího obrazovku. Vytvoříme pozadí krajiny s referenčním bodem v levém dolním rohu a bodem vložení v levém horním rohu. Vrstva bude nula, čas vložení se převezme z nastavení času scény, koncový čas necháme na mínus jedna.

Nyní budeme umisťovat nepřátele. V pravém panelu vybereme nástroj nepřítel. Čas scény nastavíme na vhodnou hodnotu, v tomto čase se objeví vložený nepřítel. Tlačítkem vytvoříme nového nepřítele, zvolíme jeho typ. Zvolený nepřítel bude označen obdélníčkem a ukáže se náhled jeho trajektorie. Pomocí myši ho přesuneme za horní okraj obdélníka. Posuneme čas scény na novou hodnotu, ve kterém budeme chtít vytvořit dalšího nepřítele. Takto budeme pokračovat dokud se nedostaneme k času 12000 milisekund.

5.12.3 Skriptování

Když by se v tuto chvíli vygeneroval skript a byl vložen do enginu, hráč by ovládal avatara a objevovali by se nepřátelé, ale hráč by nemohl střílet a hra by neměla konec. Proto je třeba pro hru vytvořit skript. V nabídce nástroje vybereme editor spínačů.

Začneme s podmínkami pro ukončení hry. Vytvoříme nový spínač, pojmenujeme ho vítězství. Přidáme událost *čas uběhl* a nastavíme hodnotu dvě minuty. Do podmínky přidáme, že životy hráče musí být větší než nula. Vytvoříme akci *ukončit úroveň* a příznak nastavíme na vítězství. Podobně vytvoříme spínač pro prohru. Jako událost určíme změnu životů hráče. Podmínkou bude hodnota životů hráče menší nebo rovna nule. Akce ukončí úroveň jako prohru.

Další vytvoříme spínače pro střelbu hráče a nepřítele typu kobold. Ve spínači střelby hráče vytvoříme událost *zmáčknuto tlačítko střelby*. Akce bude vytvořit střelu hráče, zvolíme objekt střela. Počáteční pozici zadáme jako pozici hráče minus pozice $X=0$, $Y=130$. Střela bude vycházet z hlavy draka a ne z jeho středu. K vytváření střel kobolda použijeme dvě události *čas objektu*, objekt nastavíme na typ kobold1 a u druhé události na kobold2. Čas nastavíme na hodnotu 750 milisekund. Podmínkou bude, že je objekt stále aktivní. Akce vložíme tři. První akce bude vytvoření objektu typu koule na pozici objektu spínajícího událost. Dále vložíme akci čekat s hodnotou 750 milisekund. Poslední akce bude spustit tento spínač, abychom docílili periodického spuštění.

Zbývá vytvořit spínač přehrávající animace smrti nepřátel. Budou zde tři události *změna životů objektu*. Jako typ objektu zvolíme topír, kobold1 a kobold2. Podmínkou budou životy objektu menší nebo rovné nule. Z akcí zvolíme přehrání animace na pozici objektu.

6 Závěr

Během práce byl upraven engine pro hru žánru Shoot 'em up, který byl vytvořen v rámci magisterského projektu. Tento engine byl vytvořen v XNA studiu pro jazyk C#. Největší část úprav tvořily časovače a spínače. Úroveň hry je řízena skriptem lua, který byl vygenerovaný v editoru. Engine je navrhovaný, tak aby bylo možné jej rozšířit například o další podmínky nebo akce. Hra běží s šedesáti snímky za sekundu.

Editor je naprogramovaný v jazyce C# a využívá grafické rozhraní WPF. Editor je členěný do několika oken. V každém okně se definují různé vlastnosti úrovně hry, například animace, objekty, trajektorie, zvuky a spínače. Spínače a časovače jsou vytvářeny ve formě stavebnice. Editor generuje skript lua a ukládá obrázky ve formátu PNG. Soubory jsou ukládány do struktury, aby bylo možné úroveň jednoduše překopírovat do složky s enginem.

V editoru bylo vytvořeno několik testovacích úrovní, z nichž je tvorba jedné popsána. V tomto editoru byl schopen úroveň hry vytvořit i člověk neznalý programování. Editor i engine mám v plánu dále rozšiřovat, například odstranit chyby, které nebyly odhaleny testováním, nebo přidat další možnosti spínačů.

Seznam použité literatury

- [1] *Entertainment Software Association* [online]. Copyright © 2014
[cit. 2014-04-30]. Dostupné z: <http://www.theesa.com/>
- [2] EPIC GAMES, INC. *Unreal Engine* [online]. © 2004-2014 [cit. 2014-04-30].
Dostupné z: <https://www.unrealengine.com/>
- [3] YOYO GAMES LTD. *YOYO GAMES* [online]. © 2013-2014
[cit. 2014-04-30]. Dostupné z: <https://www.yoyogames.com/>
- [4] Games That Defined the Shmups Genre. In: *Racket boy: your guide for a retro gaming lifestyle* [online]. 19. 4. 2012, 12. 5. 2012 [cit. 2014-04-30].
Dostupné z:
<http://www.racketboy.com/retro/shooters/games-that-defined-the-shmups-genre>
- [5] REED, Aaron. *Learning XNA 4.0*. Sebastopol, CA: O'Reilly, c2011, xix, 516 p.
ISBN 14-493-9462-0.
- [6] MICROSOFT. *Xbox LIVE Indie Games* [online]. © 2013 [cit. 2014-04-30].
Dostupné z: <http://xbox.create.msdn.com/en-US/>
- [7] *Lua: the programming language* [online]. © 1994–2013 [cit. 2014-04-30].
Dostupné z: <http://www.lua.org/>
- [8] Scripting with Lua in C#. *Godpatterns* [online]. May 20, 2006
[cit. 2014-05-30].
Dostupné z: <http://www.godpatterns.com/2006/05/scripting-with-lua-in-c.html>
- [9] Windows Presentation Foundation (WPF): Next Generation of User Interfaces.
National Instruments [online]. © 2012 [cit. 2014-04-30].
Dostupné z: <http://www.ni.com/wpf/>

A Obsah Přiloženého CD

- Text diplomové práce
- Instalační balíček obsahující engine a editor
- Soubory potřebné k vytvoření ukázkové úrovně
- Zdrojové soubory