

Space Pilots



Gymnázium, Praha 6, Arabská 14

Předmět Programování, vyučující Jan Lána

Marek Daňa, Tomáš Průšek, Jan Mucha, Viktor Vašek 3. E

květen 2019

Prohlášení

Prohlašujeme, že jsme jedinými autory tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené.

Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne ____ května 2019

Podpisy:

Obsah

1) Zadání projektu	1
2) Anotace.....	1
3) Abstract	1
4) Úvod.....	2
5) Návod k projektu.....	3
6) Informace o fungování programu.....	4
6.1) Použité technologie	4
6.2) Postup instalace a systémové požadavky	4
6.3) Struktura programu	4
6.4) Datové struktury.....	4
7) Popis zdrojového kódu.....	5
7.1) Main.js.....	5
7.2) MenuScene.js	5
7.3) LeaderboardScene.js	6
7.4) PlayScene.js	7
7.5) LevelScene.js.....	7
7.6) UpgradeScene.js.....	7
7.7) GameoverScene.js a VictoryScene.js.....	8
7.8) PauseScene.js	8
8) Chování nepřátel	9
8.1) Alien1	9
8.2) Alien2.....	9
8.3) Alien3.....	10
8.4) Alien4.....	10
8.5) Alien5.....	11
8.6) Velitelé v 1. a 2. úrovni.....	11
8.7) Velitelé 3. a 4. úrovně	11
8.8) Uložení a načtení hry.....	13
8.9) Žebříček nejlepších hráčů.....	13
8.10) Chování nepřátel	13
8.10.a) Vyhýbání se střelám (třída Alien3).....	13
8.10.b) Chování pokročilých mimozemských velitelů	14
8.11) Struktura.....	15
8.12) Pozadí.....	15
8.13) Textury	16

9) Závěr.....	17
10) Zdroje	18

1) Zadání projektu

Space Pilots

Zadáním je vesmírná 2D hra až pro 2 hráče – hráči (lidé) budou připojeni na server, připojí se přes web, na kterém také bude ukázka hry, návod a příběh). Každý z hráčů ovládá jednu loď a společně se snaží zničit nepřátelské lodě v aréně. Loď se bude moct pohybovat čtyřmi směry a střílet bude před sebe (více druhů střel). Nepřátelské lodě budou reagovat na chování hráče (uhýbání, míření). Na konci úrovně se objeví boss a po dokončení bude mít hráč možnost vylepšit si svou loď (peníze na vylepšení bude hráč získávat ničením nepřátelských lodí) a poté bude další úroveň. Hra bude obsahovat základní menu s možností výběru úrovně, uložení hry a nastavením.

2) Anotace

Cílem našeho ročníkového projektu je naprogramovat vesmírnou 2D hru Space Pilots v prohlížeči. V této hře hráč ovládá vesmírnou vybranou loď, se kterou se snaží zničit nepřátelské lodě v jednotlivých úrovních. Mezi úrovněmi si může uložit hru a vylepšit svou loď a dozví se informace k další misi. Loď má několik typů útoků a pohybuje se do čtyř směrů. Nepřítel v úrovních je více druhů, každý má jiné chování a střelbu. Na konci každé úrovně hráč musí porazit nepřátelského velitele. Hra obsahuje základní menu s nastavením zvuku a hudby, instrukcemi a žebříčkem nejlepších pilotů na daném počítači v daném prohlížeči.

3) Abstract

The goal of our project is to program a space 2D game Space Pilots played in a browser. In this game, the player controls a selected spaceship which he uses to destroy enemy ships in each level. Between levels, he can save the game and upgrade his ship and learn something about the next mission. The ship has several types of attacks and moves in four directions. There are more types of enemy ships, each has different behavior and attacks. At the end of each level, the player must defeat the enemy commander. The game includes a basic menu with audio and music settings, instructions and a leaderboard of the best pilots in the browser.

4) Úvod

V této práci popíšeme, jak jsme postupovali při programování našeho ročníkového projektu Space Pilots. Tuto hru jsme si vybrali, protože rádi hrajeme vesmírné hry a také jsme se chtěli naučit programovat webové hry. Vzhledem k tomu, že jsme s hrami tohoto typu strávili mnoho času, napadlo nás, že by bylo zajímavé ji naprogramovat, abychom poznali zákonitosti těchto her. Od této práce očekáváme, že se naučíme používat framework Phaser 3, zlepšíme se v programování, týmové spolupráci a vytváření grafiky. V této práci popisujeme, jakou strukturu náš projekt má, jak jsme postupovali při jeho vytváření a jaké jsme řešili problémy. Dále obsahuje návod a základní popis našeho programu. Také vysvětluje námi vytvořené a použité algoritmy a grafickou stránku projektu.

5) Návod k projektu

Prohlížeč se hrou se zapne spuštěním souboru Play.bat. Po spuštění hry se objeví webová stránka s oknem, kde se nachází samotná hra a tlačítkem Menu, pomocí kterého je možné se dostat na stránky s příběhem a ukázkou hry. Pokud se hráč rozhodne hrát, může pokračovat pomocí tlačítka Play nebo tlačítka Load, pokud hru už předtím hrál a uložil. Pokud hru hraje poprvé, může se dozvědět, jak se hra ovládá, pokud klikne na tlačítko Instructions.

Po celou dobu hry hraje hudba, kterou je možno ztlumit pomocí tlačítek ovládajících hlasitost, popř. změnit. Zmáčknutím tlačítka Leaderboard se spustí scéna, která obsahuje žebříček nejlepších hráčů (uloženo v paměti prohlížeče). Po zmáčknutí tlačítka Play se dostane do scény, ve níž si zvolí obtížnost, herní režim a loď. Ovládání je uživatelsky přívětivé a snadno pochopitelné. Po potvrzení nastavení se hráč dostane do hry samotné. Hru hráč ovládá pomocí klávesnice (WSAD – pohyb lodi, B – použití bomby, X – střelba laseru, mezerník – střelba plasmových střel, escape – pozastavení hry).

V pravé horní části obrazovky má hráč ukazatele energie, štítů, peněz, skóre a počtu bomb. Úkolem hráče je vyhýbat se nepřátelským lodím a střela a snažit se je zničit pomocí svých zbraní. Za každé sestřelení nepřátelské lodi získává hráč peníze, které později může použít na vylepšení své lodi. Po několika vlnách přiletí nepřátelský velitel (větší loď, s velkým množstvím životů), po jehož zničení se hráč dostane do scény, kde si může vylepšit svou loď a dozví se základní informace o dalším levelu. V této scéně si hráč může hru uložit, vrátit se zpět do menu, a hlavně zlepšit atributy své lodi (poškození střel, regenerace energie atd.). Pokud hráčovy štíty klesnou na nulu, hra pro něj končí a vrací se do menu. Po zničení finálního bosse ve čtvrtém levelu hráč zvítězí a dostane se do vítězné scény.

6) Informace o fungování programu

6.1) Použité technologie

Jako vývojové prostředí jsme použili Visual Studio Code. Hra byla naprogramována v javascriptovém frameworku Phaser 3, což je framework určený na vytváření 2D her. Jako kompilátor webových aplikací a zároveň server jsme použili Parcel. K tomu, aby Parcel bylo možné nainstalovat a používat bylo nutné nainstalovat Node.js. Pro vytváření grafické části hry byl využit GIMP, jak k vytváření vlastních obrázků, tak k úpravě již existujících.

6.2) Postup instalace a systémové požadavky

Ke spuštění hry je potřeba operační systém Windows pro počítače a nainstalovaný Node.js. Hra se spustí pomocí souboru Play.bat.

6.3) Struktura programu

Program se dělí na dvě hlavní části. První je html část, která zobrazí webové stránky s hrou. Druhou částí je samotný kód hry napsaný ve phaseru (javascript).

Html část se skládá ze tří html dokumentů. První html dokument zajišťuje načtení scriptu hry a její zobrazení. Dále obsahuje menu, pomocí kterého se lze dostat na dvě zbývající stránky. Druhý html dokument zobrazuje video ukázkou naší hry. Třetí html dokument zobrazí stránku s příběhem hry.

Zdrojový kód hry psaný v jazyce Javascript je rozložen do několika souborů s příponou „.js“. V souboru main.js se inicializuje třída Phaser.Game, která se stará o chod aplikace. Nastaví se velikost okna, způsob renderování, fyzika, používané scény. Základním principem frameworku Phaser 3 je přepínání mezi jednotlivými scénami, přičemž každá scéna má vlastní soubor a funkci.

Každá scéna má metody init, preload, create, update a vlastní proměnné. Metoda init slouží k příjmu dat z předchozí scény. Metoda preload načítá vizuální a zvukové soubory (např. obrázky, zvuky) ze složky assets (v našem případě). V metodě create se vytváří třídy, objekty, skupiny, animace atd. Metoda update je nekonečný cyklus, který se stará o aktualizování scény, např. pohyb nepřátel a jejich střelba.

6.4) Datové struktury

Z datových struktur je v našem programu nejvíce zastoupen primitivní var pro uložení jednoduchých informací - například počet životů, energie atd. Dále je využíváno pole na uložení nepřátel, střel a některých grafických prvků. Ze složitějších struktur je využíván objekt pro uložení hry.

7) Popis zdrojového kódu

7.1) Main.js

V tomto souboru se vytváří třída naší hry a nastavují se její vlastnosti. Nastavuje se velikost a styl (ohraničení) okna hry, způsob renderování a fyzika, např. gravitace, a registrují se scény, které má hra využívat, a importují se.

```
let game = new Phaser.Game({
  type: Phaser.AUTO,
  width: 1280, // šířka okna
  height: 896, // výška okna
  canvas: document.getElementById('myCustomCanvas'),
  //seznam scen
  scene: [
    MenuScene, CreditsScene, PlayScene, InstructionsScene,
    Level1Scene, Level2Scene, Level3Scene, Level4Scene, UpgradeScene,
    PauseScene, GameoverScene, VictoryScene, LeaderboardScene,
    InfiniteLevelScene, InfiniteLevelPauseScene
  ],
  render: {
    pixelArt: true
  },
  // nastavení fyziky(gravitace, fps ...)
  physics: {
    default: "arcade",
    arcade: {
      fps: 60,
      gravity: { y: 0 },
    }
  }
});
```

Ukázka kódu - inicializace hry

7.2) MenuScene.js

Tato část kódu vytváří hlavní menu a je to první scéna, která se zobrazí, když si hráč hru otevře ve webovém prohlížeči. Mimo třídu scény se vytváří var proměnné pro celý soubor.

V metodě `init` se z dat odeslaných předchozí scénou získají souřadnice pozadí, aby mělo stejné posunutí jako v předchozí scéně, a hlasitost zvuku.

V metodě `preload` se načtou grafické a zvukové soubory pro celou hru (pro všechny scény, takže se nemusí načítat vícekrát). Dále se každému z nich nastaví jméno, aby se poté na něj dalo odkázat.

V metodě `create` se nastaví background, hudba a zvuky, tlačítka a jejich akce. Nastavují se akce pro tlačítka nastavení zvuku (změní se hlasitost zvuku, nebo se změní hudba na pozadí) a tlačítka pro změnu scény (změní se číslo, podle kterého se poté v metodě `update` spustí odpovídající scéna).

V metodě update je udělané posouvání pozadí a přesun do další scény podle čísla, které se nastavilo, když hráč zmáčknul nějaké tlačítko (např. do scény, kde si vybírá loď, žebříčku nejlepších hráčů nebo instrukcí). Když se mění scéna, tak se posílají také nějaké údaje další scéně, např. souřadnice pozadí (aby se pozadí náhle neposunulo), hlasitost zvuků a hudby atd. Posouvání pozadí u všech scén, kde se pozadí posouvá, funguje tak, že se od y souřadnice odečítá při každém updatu 0.5, čímž je zajištěn

pohyb. Aby se pozadí neposunulo mimo okno hry a hráč neviděl černou plochu, tak se pozadí skládá ze dvou obrázků. Když obrázek vyjede mimo okno hry, tak se přesune pod ten druhý, čímž je zajištěn nekonečný pohyb pozadí.

```
background.y -= 0.5;
if (background.y === -1792) {
    background.y = 5376;
}
background2.y -= 0.5;

if (background2.y === -1792) {
    background2.y = 5376;
}
```

7.3) LeaderboardScene.js

Do této scény se lze dostat z hlavního menu a zobrazí se v ní seznam deseti nejlepších hráčů (uložení skóre nejlepších hráčů a opětovné načtení je podrobněji popsáno v kapitole 9). Funguje to tak, že se ze souboru načte pole s deseti nejlepšími hráči a jejich skóre. Poté se vytvoří 20 textových polí, která jsou ve dvou sloupcích, do prvního sloupce se z pole načtou jména a do druhého sloupce se načte skóre.

7.4) PlayScene.js

Do této scény se lze dostat z hlavního menu a při přechodu z hlavního menu do této scény zadá své jméno. Hráč si zde vybere obtížnost hry, herní režim a loď. Poté svoji volbu potvrdí tlačítkem confirm. V horní části okna hry je zobrazeno přes textová pole, co si hráč zatím zvolil za jméno, loď, obtížnost a herní režim. Ve střední části okna hry je obrázek lodi, již má aktuálně vybranou, na každé straně je jedna šipka na změnu vybrané lodi a pod lodí jsou graficky znázorněny její silné a slabé stránky. To je uděláno tak, že se vytvoří grafické pole a v něm se nakreslí obdélníky ve 4 řadách, každá řada má 5 obdélníků a vedle každé řady je přes textové pole napsáno, co označuje. Obdélníky jsou červené, nebo zelené. Grafické pole se znovu vykresluje, pouze když hráč změní loď.

7.5) LevelScene.js

Všechny Level[X]Scene.js soubory mají stejnou strukturu, liší se pouze počtem vln nepřátel, druhy nepřátelských lodí a velitelskou lodí. Mimo třídu scény se vytváří var proměnné pro celý soubor.

V metodě init se načítají ze vstupních dat hodnoty do proměnných, např. úrovně vylepšení, jméno hráče, typ lodi atd.

V metodě create se vytváří třídy, objekty a animace, např. nepřátelské lodě, střely, animace výbuchu atd. Každá třída má svoji skupinu, která hlídá, že od dané třídy nebude více instancí, než je limit. Dále se zde nastavuje zvuk, pozadí, hodnoty proměnných a vstup z klávesnice. Dále se zde vytváří textová a grafická pole, která se poté využijí v metodě update.

V metodě update jsou definované funkce, např. kolize střely s lodí, kolize hráče s nepřátelskou lodí atd. Zda dva objekty spolu kolidují, hlídá Phaser 3 sám. Nastaví se, které dvě skupiny objektů (nebo dva objekty, nebo skupina objektů a objekt) mají spolu kolidovat a funkce, která se při kolizi spustí. Dále se zde zajišťuje posun pozadí, regenerace energie, vykreslování ukazatelů energie a štítu přes grafické pole, střelbu nepřátel, pohyb a útoky hráče a kontrola, jestli již hráč zvítězil, nebo prohrál. Poté se data odešlou do další scény.

7.6) UpgradeScene.js

Tato scéna se spustí poté, co skončí první, druhý anebo třetí level. Hráč si v ní může uložit hru (uložení a načtení hry je popsáno v kapitole 9), vylepšit svoji loď, přečíst příběh k další misi, odejít do hlavního menu anebo spustit další level. Hráč může vylepšit poškození, redukci poškození od nepřátel, cenu střelby plasmových střel a laseru, regeneraci energie a počet bomb. Na levé straně jsou tlačítka pro vylepšení, na kterých je napsána cena daného vylepšení. Napravo od nich jsou řady obdélníků, které ukazují aktuální a maximální úroveň vylepšení. Zelené obdélníky vyjadřují již zakoupená vylepšení a červené ještě nezakoupená. Obdélníky jsou udělány přes grafické pole, ve kterém se pomocí for-cyklu s podmínkami vykreslí obdélníky. Napravo od nich je ještě přes textové pole ke každému upgradu jednoduše vysvětleno, co ovlivňují. Vpravo nahoře je příběh k další misi, který je opět zobrazen přes textové pole.

7.7) GameoverScene.js a VictoryScene.js

Tyto scény se spustí, když hráčovi vybuchne loď, nebo když vyhraje poslední level. Do těchto scén se z levelu pošle skóre a jméno hráče a uloží se. Z těchto scén se poté hráč může vrátit do hlavního menu.

7.8) PauseScene.js

Tato scéna se spustí v levelu, když hráč zmáčkne klávesu escape a z levelu se pošle, jaká scéna byla pozastavena a poté se případně znovu rozběhne. Hráč se může vrátit z této scény do hry, nebo do hlavního menu, přičemž přijde o veškerý postup v daném levelu.

8) Chování nepřátel

Jak se nepřátelské letouny pohybují po hrací ploše a jak střílí, je popisováno právě v této části. Vysvětleny jsou zde popořadě algoritmy určující pohyb jednotlivých druhů mimozemských stíhačů a systém jejich střílení, a to v pořadí, v jakém se vyskytují v příběhovém režimu.

Ve všech úrovních jsou mimozemské stíhačky generovány ve vlnách. V každé vlně letí vždy 1 stíhač od každého druhu obsaženého v dané úrovni. Jejich souřadnice x je určena náhodně, aby byla každá vlna jedinečná a při každém spuštění hry byla úroveň trochu jiná. Náročnost však zůstává přibližně stejná. To, kdy mimozemšťané přiletí do herního okna, je určeno číslem vlny, ale také drobnou též náhodně určenou výchylkou. Úmysl je zde stejný jako v případě náhodného určení souřadnice x.

8.1) Alien1

Tento se vyskytuje jen a pouze v úrovních 1 a 2. Jde o nejjednoduššího ze všech nepřátel, avšak pokud má být sestřelen, je nutné zaútočit co nejdříve a vyhnout se jeho pravidelným výstřelům.

Jedná se o prostý letoun, jenž se pohybuje stále přímo dolů. Při každém snímku se jeho souřadnice y zvýší o jeho rychlost, což alespoň zaručuje, že z hrací plochy rychle odletí a pro hráče je trochu těžší stihnout ho sestřelit, a získat tak peněžní odměnu.

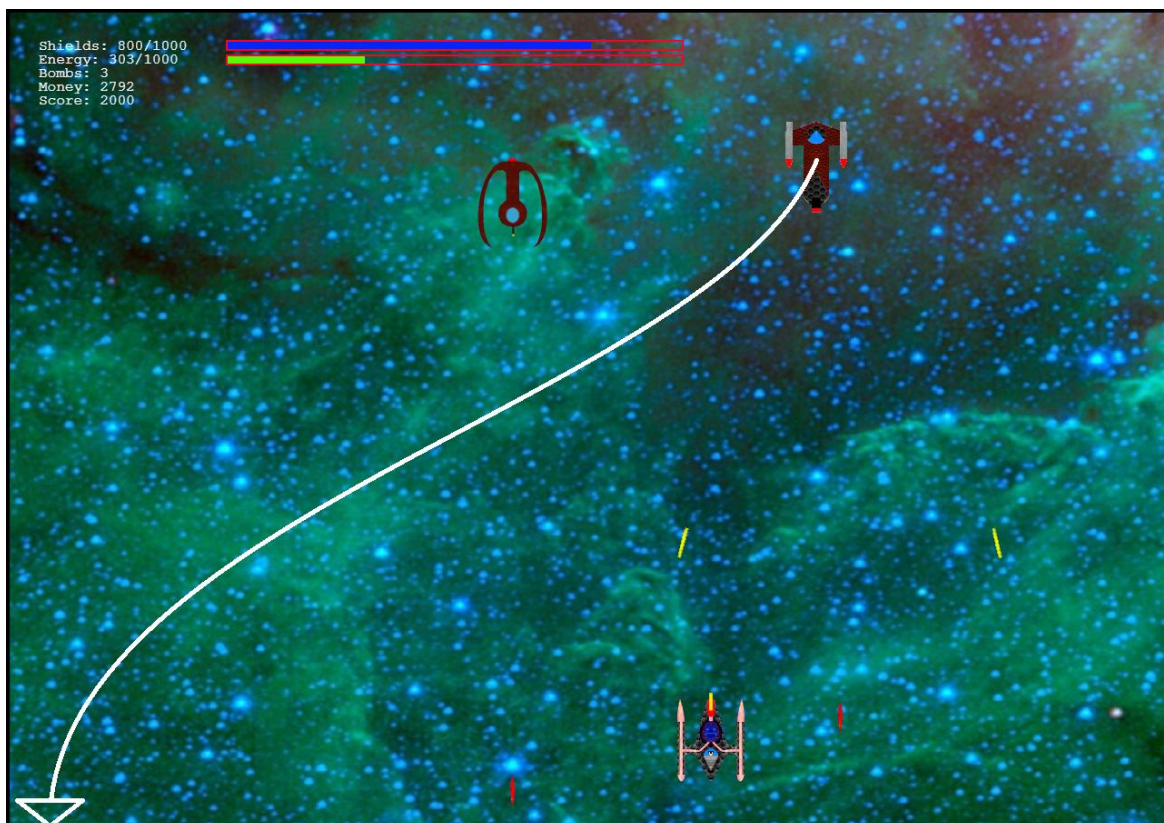
Náboje letí rovně dolů, samozřejmě tak musí činit rychleji, než se pohybuje střelec, aby je nepředletěl. Tím by ztratily svůj význam.

```
var Alien1 = new Phaser.Class({
    Extends: Phaser.GameObjects.Image,
    initialize:
        function Alien1(scene) {
            this.attackSpeed = 325;
            this.lastShot = 0;
            this.health = 500;
            this.speed = 2;
            Phaser.GameObjects.Image.call(this, scene, 0, 0, 'Alien1');
        },
});
```

8.2) Alien2

Druhý typ nepřátel vyskytující se v prvních třech úrovních využívá výpočty poněkud složitější, ačkoli ještě nereaguje na podněty, které by mu dával hráč.

Trajektorie těchto stíhačů se skládá ze dvou částí elipsy, a tak připomíná vlnu. Jejich matematika funguje tak, aby svou cestu, pokud do té doby nejsou sestřeleny, končily v jednom z dolních rohů. Ve kterém z nich je určeno následovně: Pokud se stíhač ve hře objeví v levé polovině okna, cílem jeho cesty je pravý dolní roh, v opačném případě pak levý. Níže se nachází ukázka ze hry s vyznačenou trajektorií tohoto druhu nepřátel.



Ukázka ze hry Space Pilots – pohyb nepřátel Alien2

Podobně jako střely předchozího typu nejsou ani střely těchto nepřátel ničím zvláštním. Nicméně jsou vypuštěny vždy tři střely naráz, přičemž dvě z nich letí šikmo dolů po polopřímce, která s vertikálou svírá úhel necelých 14 stupňů. Tato vícenásobná střelba činí zde popisovaný druh mimozemských stíhačů nejsilnějším po stránce maximálního způsobovaného poškození za jednotku času.

8.3) Alien3

Od úrovně 2 proti hráči nastupují poněkud inteligentnější jednotky. Tento druh zvládá jednoduché uhýbání střelám a těmi svými míří přímo na hráčovu loď. To hráče nutí věnovat pozornost vyhýbání každých několik vteřin.

Tyto nepřátelé doletí na jistou souřadnici svislé osy y a zůstávají na svém místě, dokud se k nim neblíží hráčova střela. V tu chvíli se takové střele vyhnou buď doleva, nebo doprava – záleží, kudy se vyhnou nejrychleji. Výjimka nastane, pokud se tyto lodě nachází na kraji a měly by z herního okna uletět pryč.

Projektily letí od místa výstřelu přímo k pozici hráčovy lodě v čase výstřelu. K tomuto bylo potřeba obrázek náboje správně otáčet pomocí goniometrických funkcí. Také výpočet rychlostí pohybů náboje po obou osách funguje na bázi funkcí sinus a kosinus.

8.4) Alien4

V úrovních 3 a 4 se vyskytují nepřátelé, jejichž předností je rychlost. Na druhou stranu jim byl snížen počet životů. Na své cestě položí pár objektů, min, které zůstávají ve hře, dokud je hráč neseštelí, nebo do nich nenarazí a neutrpí tím poškození.

Letouny schopné klást miny po přiletu do herního okna zamíří do nejvzdálenějšího rohu přímou čarou. V jednoduchém algoritmu probíhají výpočty s použitím goniometrických funkcí, avšak nejsou tak složité jako v případě typu Alien2, neboť tento se pohybuje přímočaře.

Minám, byl stejně tak jako nepřátelským stíhačům přidělen určitý počet životů, a to tak, aby ji bylo možné zneškodnit jedním hráčovým výstřelem, nebo na pár okamžiků zapnutým laserem.

8.5) Alien5

Nepřátelé objevující se pouze v poslední a nejnáročnější úrovni jsou vybaveni algoritmem, jenž sleduje pozici hráčovy stíhačky. Frekvence jejich střelby přitom také závisí na jejich pohybu. Těž disponuje schopností pomalé obnovy životů.

Zrcadlení vertikální souřadnice zvyšuje náročnost sběru bonusových vylepšení, které se někdy objevují po zničených nepřátelských lodích. Pokud jde o pohyb po horizontální ose, tento druh nepřátel se pohybuje tak, aby se jeho první souřadnice přibližovala k té hráčovo, případně zastaví.

Výstřel nastává ve chvíli, kdy je stíhač tohoto typu nucen změnit horizontální směr letu. Může tak nastat situace, kdy bude vystřeleno několik nábojů za sebou, a to právě tehdy, když hráč pod tímto stíhačem létá rychle ze strany na stranu – vždy jen na malý zlomek vteřiny.

V tomto případě nebylo vůbec nutné střely vybavovat složitým algoritmem. Zde popisovaný typ stíhače se pohybuje většinu času přímo nad hráčovou lodí, a tudíž i náboje letí přímo dolů od místa výstřelu.

8.6) Velitelé v 1. a 2. úrovni

Na koncích prvních dvou úrovní jsou velké jednotky s více než desetinásobně větším počtem životů (přesněji 8000). Jejich zničení je následováno spuštěním meziúrovňové scény s vylepšeními. Pro oba velitele je charakteristický podobný pohyb. V horní části herního okna se pohybují ze strany na stranu. Uprostřed pomyslné kolejnice, po které létají, je jejich pohyb pozastavován, aby mohly být použity jejich speciální útoky včetně vystřelování několika nábojů naráz, a to do všech stran. Ve svých útocích se z větší části příliš neliší kromě náhlého vyletění směrem dolů v případě velitele úrovně 2 ve snaze do hráče narazit. Jimi používaný laser a dvojité střelby míří směrem dolů.

Při pozastavení jejich letu je vystřeleno dvakrát po 16 nábojích. Jsou rozptýleny tak, aby pokryly většinu herního okna a bylo nutné najít mezi nimi mezeru pro bezpečný průlet. Pro zvýšení náročnosti a zároveň pro jejich lepší viditelnost jim byl přisouzen kulový tvar.

8.7) Velitelé 3. a 4. úrovně

Poslední 2 úrovně jsou zakončeny souboji s pokročilejšími jednotkami. Veliteli úrovně 3 byl přidán počet životů rovný 10000 a veliteli úrovně 4 byl zvýšen na 16000. Byl pro ně zvolen kyvadlový pohyb v horní části okna hry. Střely těchto majestátních jednotek jsou vytvářeny neměnnou frekvencí nezávisle na pohybu svých střelců.

Kromě střelby se v úrovni 4 při závěrečné přestřelce jednou za čas objevují jednotky třídy Alien3 a Alien4.


```

// po několika vystrelech spawnuje Alien5 do max. počtu 3.
if (BossList[0].movement % (BossList[0].attackSpeed * 16) === 0) {
    var Alien4 = aliens4.get();
    Alien4.create(BossList[0].x, BossList[0].y + 80);
    aliens4List.push(Alien4);
} else if (BossList[0].movement % (BossList[0].attackSpeed * 16) - 8
    * BossList[0].attackSpeed === 0) {
    var Alien3 = aliens3.get();
    Alien3.create(BossList[0].x, BossList[0].y + 80);
    aliens3List.push(Alien3);
}

```

Ukázka kódu – vytváření jednotek při boji s velitelem

Projektily velitele 3. úrovně jsou vytvářeny frekventovaně a směry každých dvou z nich po sobě vystřelených se liší o 15 stupňů. Jejich spojnicí je tak vykreslována spirála až do chvíle, kdy je směr jednoho rovnoběžný s horizontálou. Spojnicí projektilů vytvářených po tomto okamžiku je tvořena druhá spirála stáčející se na opačnou stranu. Pohyb těchto nábojů je přímočarý.

Poslední velitel byl vybaven mířenými střelami. Těm je navíc v průběhu letu směr několikrát obnoven, aby opět zamířili k pozici hráčova stíhače. Po zlomku vteřiny se otáčí směrem k nové pozici hráče. Toto je provedeno několikrát. Poté se schopnost projektilu otáčet se vytratí a následuje vyletění střely z herního okna, nebo případná kolize se stíhačkou hráče.

9) Klíčové problémy

9.1) Uložení a načtení hry

Uložení hry probíhá pomocí vytvoření objektu `saveObject`, který obsahuje jednotlivé proměnné, které je potřeba uložit (počet peněz, úroveň, ve které se hráč nachází atd.). Pomocí metody `JSON.stringify` se objekt konvertuje na JSON string. Toto převedení velmi usnadňuje uložení hry, protože nyní pouze stačí uložit tento JSON string do lokální paměti prohlížeče (`localStorage.setItem("save", saveObject)`).

Načtení uložené hry probíhá obdobně - pouze se celý proces obrátí. Z lokální paměti prohlížeče získáme JSON string "save" obsahující data a pomocí `JSON.parse` získáme opět objekt s uloženými daty.

9.2) Žebříček nejlepších hráčů

Po skončení hry (vítězství/porážka) se vytvoří objekt se jménem hráče a skórem, kterého dosáhl. V lokální paměti prohlížeče je uloženo pole těchto objektů a při skončení hry se do něj tento objekt přidá. Poté se toto pole objektů seřadí podle velikosti skóre (základní Bubble sort, od největšího po nejmenší) a seřazené se opět uloží do paměti prohlížeče.

9.3) Chování nepřátel

V pozdějších fázích vývoje hry se objevovaly mimo jiné i problémy tkvící v nedostatku nápadů. Například jak budou střílet objekty třídy `Alien4` a `Alien5` a jak se budou pohybovat. Níže je popsáno pár významnějších komplikací.

Vyhýbání se střelám (třída `Alien3`)

Ačkoli projektily hráče letí pouze rovně, což činí tento úkol několikanásobně snazším, byl nejobtížnějším problémem týkajícím se chování lodí mimozemšťanů algoritmus, jenž by zvládal vypočítávat nejrychlejší možnou cestu pryč z roje hráčových střel.

Ukázalo se, že bude nutné vytvořit pole, do kterého budou ukládány všechny hráčovy náboje, které se dosud nachází v herním okně. Pokud je nějaký stíhač s umělou inteligencí vyhýbající se střelám v cestě náboji, či nábojům hráče, následuje vyhodnocení dvou možností, jakými se lze (ve většině případů) takovému projektilu vyhnout, a poté výběr té rychlejší. V případě, že se nějaká hráčova střela ocitne v takové blízkosti stíhače, že je pro něj srážka s ním nevyhnutelná, nemá smysl se jí snažit vyhnout, a tak takovou střelu ignoruje.

Pro menší složitost algoritmu byla každému objektu třídy `Alien3` připsány dvě proměnné: První určuje směr letu, druhá jeho délku. Pokud je délka letu 0, znamená to, že má stíhač zůstat na místě. V opačném případě se pohybuje příslušným směrem a pamatuje si, že prozatím ví, co dělat, a tudíž po tu dobu žádné výpočetní algoritmy neprovádí.

Když se poté přišlo na algoritmus schopný vyhýbat se několika nábojům hráče najednou, vyšlo najevo, že pouze uhýbání jednotlivým střelám do stran poskytuje vybaveným nepřátelům dostatečně velkou výhodu. Z toho důvodu se další vývoj algoritmu pro nepřátelské uhýbání střelám musel zastavit.

Zbývalo již pouze zabránit mimozemšťanům v úniku z herního okna. Ti, kteří se nachází příliš blízko okraji, automaticky preferují pohyb směrem dovnitř, a to i v případě kdy je rychlejší vyhnout se střele směrem opačným.

Chování pokročilých mimozemských velitelů

Za problém společný pro velitele třetí a čtvrté úrovně lze označit výpočty pro jejich shodný pohyb. Následně pak střelbu každého z nich.

Byla vyvíjena snaha dosáhnout plynulého pohybu s hladkým zpomalováním a zrychlováním. Řešení prvního zmíněného problému bylo opět zkonstruováno za použití goniometrických funkcí, a tak byl definován kyvadlovitý pohyb do stran.

Prvním navrhovaným řešením střelby velitele 3. úrovně byly provazy kulových střel. Toto řešení se ukázalo jako optimální. Klíčem k jeho ztvárnění byl převod stupňů na radiány a goniometrické funkce. Následovalo ladění frekvence, jakou budou náboje vystřelovány a odchylky směrů 2 po sobě vystřelených nábojů.

Z několika navrhovaných možností střelby, která by mohla být přidělena veliteli mimozemšťanů v poslední úrovni, byla vybrána následující: dvojité mířené střely letící od místa výstřelu směrem k pozici hráči v okamžiku výstřelu. Byly zde použity podobné příkazy jako při programování nábojů nepřátel třídy Alien3. Zapotřebí bylo otáčení obrázků projektilů a vypočítání směru letu.

Nicméně o to náročnější se stal úkol vyvážit nyní již značně vylepšenou loď hráče a efektivitu těchto střel. V průběhu tohoto procesu byly několikrát měněny parametry náboje, jako jsou počet otočení a frekvence otáčení, tj. po jaké době střela obnoví směr svého pohybu. Navíc původně měly být náboje vystřelovány pouze po jednom a až poté padlo rozhodnutí o dvojitých výstřelech pro zvýšení akčního aspektu hry.

Ze stejného důvodu a také pro posílení respektu, který by měl být velitelem poslední úrovně vzbuzován, mu byla přidělena schopnost vyrábět malé jednotky. Nejvhodnějšími typy nepřátel byly pro tento účel shledány třídy Alien3 a Alien4. První zmíněný se po vyrobení pohybuje na úrovni pod velitelem, aby ho mohl případně zaštitit a zároveň bylo pro hráče možné ho sestřelit i klasickými náboji včas, tj. než se zrodí další.

10) Grafická část projektu

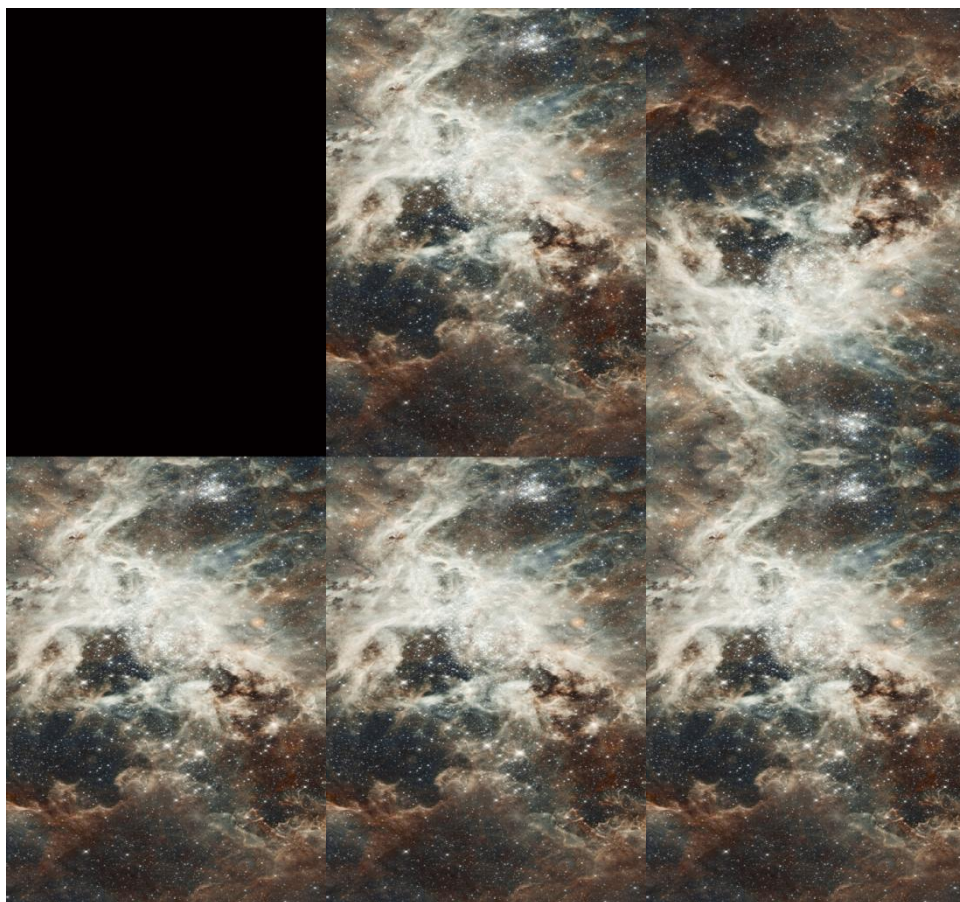
10.1) Struktura

Pro vytvoření grafiky byl použit grafický editor GIMP. Ve hře se nachází dva typy textur. Textury pro pozadí a samotné objekty (vesmírné lodě, střely, laser). Textury jsou uloženy ve formátu JPG (pozadí), nebo PNG (ostatní textury).

10.2) Pozadí

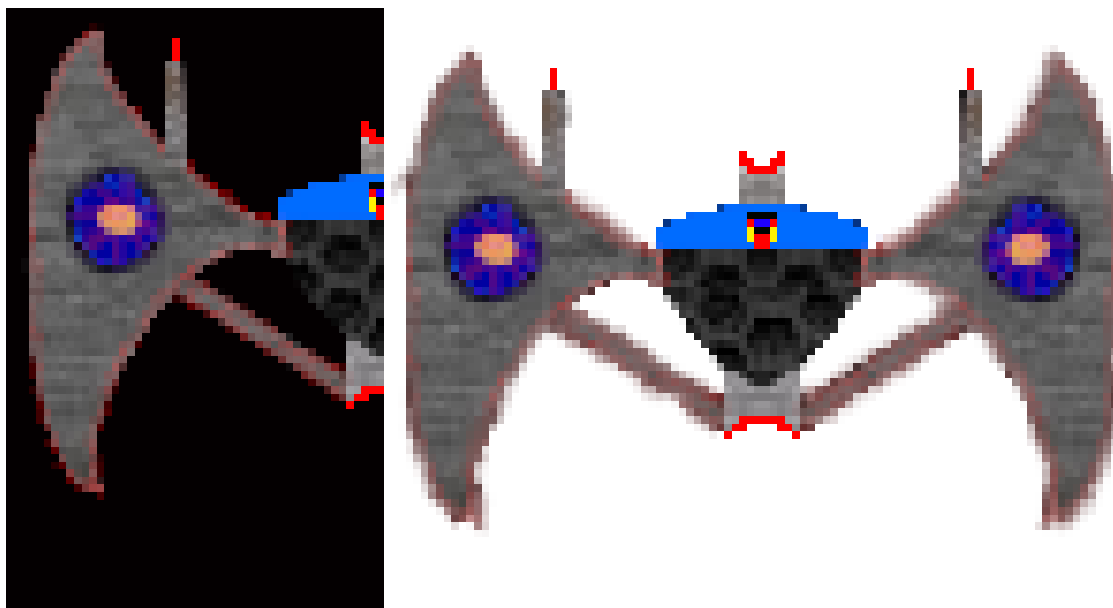
Jak bylo zmíněno výše, tak textura pozadí je uložena ve formátu PNG. Jedná se o upravené obrázky, které jsou zmíněné ve zdrojích. Ve hře jsou dva typy pozadí. Pozadí úrovní a menu, které mají rozměr 1280 x 3584 pixelů a ostatní textury (např. instrukce) s rozměry 1280 x 896 pixelů, což je velikost herního okna.

Pozadí bylo vytvořeno vyříznutím plochy odpovídající velikostí. Například u instrukcí byla u originálního obrázku vybrána požadovaná plocha 1280 x 896 pixelů a poté přidány další objekty (text, střela, vylepšení). Textury pro pozadí úrovní a menu byly vytvořeny podobným způsobem. Opět byla vyříznuta část původního obrázku o velikosti 1280 × 1792 pixelů. Druhá polovina obrázku byla poté vytvořena pomocí osové souměrnosti.



10.3) Textury

Textury jsou uloženy ve formátu PNG. Formát PNG byl nutný, neboť formát JPG nepodporuje soubory bez pozadí. Některé objekty jsou vytvořeny nesymetrickým způsobem (střely, vylepšení atd.). Jiné jsou vytvořené pomocí osové symetrie (lodě). Byly načrtnuty obrysy textur (pomocí několika nástrojů v GIMPu, především pomocí path tool). Poté byl do daných oblastí nanesen vzor, který byl vyroben ze stažených textur zmíněných ve zdrojích. Dále byly textury upravovány do své finální podoby pomocí dalších nástrojů v GIMPu (především burn tool a airbrush). Na závěr bylo odstraněno pozadí textury a u lodí byly složeny obě dvě poloviny k sobě pro vytvoření finální textury.



11) Závěr

Zadání jsme z velké části splnili, a dokonce jsme implementovali některé zajímavé funkce nad rámec zadání. Hráč může pohybovat lodí čtyřmi směry, střílet plasmové střely, laserový paprsek a odpalovat bomby. Ve hře je několik úrovní, na konci každé z nich musí hráč porazit nepřátelského velitele. Mezi jednotlivými úrovněmi si může vylepšit loď za získané peníze. Každý typ nepřátelské lodi má svůj vlastní vzorec chování a způsob útoku. Hráč si může nastavit hlasitost zvuku a vybrat hudbu, uložit hru, vybrat obtížnost a loď a vybrat si mezi příběhovým a nekonečným režimem. Hra dále obsahuje žebříček nejlepších hráčů a instrukce. Hra je na webu, kde je také ukázka hry a příběh. Přestože jsme v zadání hry měli napsáno, že hra bude až pro 2 hráče, po zvážení všech proměnných, jsme dospěli k názoru, že pro tento typ hry se nehodí, aby hráli 2 hráči spolu. Neboť by spolupráce hráčů byla téměř nemožná, překáželi by si a hra by kvůli tomu nebyla zábavná. Při vývoji hry jsme využívali framework Phaser 3, se kterým jsme se museli naučit pracovat, tudíž jsme se museli mnoho nového naučit, než jsme plně pochopili, jak funguje. Spolupráce týmu byla zpočátku náročná, nicméně postupně se zlepšila.

12) Zdroje

Při tvorbě této práce jsme vycházely z dokumentace Phaseru 3 a vzorových kódů na stránce Phaseru 3 a tutoriálu, jak zprovoznit vývojové prostředí. Většina obrázku je vlastních, část z nich byla převzata z internetu a upravena. Zvuky a hudba byly také převzaty z internetu.

Tutoriál:

Phaser 3 Game Tutorial - 01 - Setting Up The Environment, Youtube[online], 5. 10. 2018, [cit. 6. 1. 2019]. Dostupné z: <https://www.youtube.com/watch?v=wDOym-mXxO4>, kanál uživatele: jest array

Phaser 3 dokumentace:

Davey, R., z 8. 5. 2019, Phaser 3 API documentation[online] dostupné z

<<https://photonstorm.github.io/phaser3-docs/index.html>>

Phaser 3 příklady:

1. Goloskokovic, P., z 8. 5. 2019, Phaser 3 example[online] dostupné z <<http://labs.phaser.io/edit.html?src=src/audio/HTML5%20Audio/AudioSprite.js>>
2. Není uveden, z 8. 5. 2019, Phaser 3 examples[online] dostupné z <http://labs.phaser.io/edit.html?src=src/games/topdownShooter/topdown_combatMechanics.js>
3. Není uveden, z 8. 5. 2019, Phaser 3 examples[online] dostupné z <<https://labs.phaser.io/edit.html?src=src/scenes/changing%20scene.js>>
4. Není uveden, z 8. 5. 2019, Phaser 3 examples[online] dostupné z <<https://labs.phaser.io/edit.html?src=src/game%20objects/particle%20emitter/particle%20processor.js>>

Obrázky:

1. Skeeze, www.pixabay.com [online], 6. dubna 2015, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/photos/mlhovina-orel-prostor-kosmos-plyn-708839/>>
2. WikiImages, www.pixabay.com [online], 24. ledna 2013, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/photos/mlhovina-orel-prostor-kosmos-plyn-708839/>>
3. gerald, www.pixabay.com [online], 23. srpna 2014, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/illustrations/ohniv%C3%A1-koule-po%C5%BE%C3%A1r-zna%C4%8Dky-422746/>>
4. Yuri_B, www.pixabay.com [online], 8. srpna 2016, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/illustrations/prostor-galaxie-planeta-vesm%C3%ADr-1569133/>>
5. geralt, www.pixabay.com [online], 23. srpna 2014, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/illustrations/ohniv%C3%A1-koule-v%C3%BDbuch-katastrofa-422750/>>
6. Pexels, www.pixabay.com [online], 29. května 2016, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/photos/astronomie-jasn%C3%BD-souhv%C4%9Bzd%C3%ADtmav%C3%BD-1867616/>>
7. thobar, www.pixabay.com [online], 1. dubna 2018, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/photos/ohe%C5%88-planety-po%C5%BE%C3%A1r-planeta-hork%C3%A9-3281255/>>
8. WikiImages, www.pixabay.com [online], 3. ledna 2013, [cit. 27.4.2019] dostupné z <<https://pixabay.com/cs/photos/ml%C3%A9%C4%8Dn%C3%A1-dr%C3%A1ha-prostor-vesm%C3%ADr-galaxie-67504/>>

Zvuky a hudba:

1. Dexter Britain, [www.freemusicarchive.org](http://freemusicarchive.org/music/Dexter_Britain/2018100594213119/Summers_Coming) [online], [cit. 27.4.2019], Summers Coming, dostupné z http://freemusicarchive.org/music/Dexter_Britain/2018100594213119/Summers_Coming
2. BoxCat Games, [www.freemusicarchive.org](http://freemusicarchive.org/music/BoxCat_Games/Nameless_the_Hackers_RPG_Soundtrack/BoxCat_Games_-_Nameless-the_Hackers_RPG_Soundtrack_-_04_B-3) [online] [cit. 27.4.2019], B-3, dostupné z [http://freemusicarchive.org/music/BoxCat_Games/Nameless_the_Hackers_RPG_Soundtrack/BoxCat_Games - Nameless- the Hackers RPG Soundtrack - 04 B-3](http://freemusicarchive.org/music/BoxCat_Games/Nameless_the_Hackers_RPG_Soundtrack/BoxCat_Games_-_Nameless-the_Hackers_RPG_Soundtrack_-_04_B-3)
3. Dexter Britain, [www.freemusicarchive.org](http://freemusicarchive.org/music/Dexter_Britain/2018100594213119/Waking_Up_Instrumental) [online], [cit. 27.4.2019], Waking Up Instrumental, dostupné z http://freemusicarchive.org/music/Dexter_Britain/2018100594213119/Waking_Up_Instrumental
4. Dexter Britain, [www.freemusicarchive.org](http://freemusicarchive.org/music/Dexter_Britain/2018100594213119/Wonderland_Instrumental) [online], [cit. 27.4.2019], Wonderland Instrumental, dostupné z http://freemusicarchive.org/music/Dexter_Britain/2018100594213119/Wonderland_Instrumental
5. Dexter Britain, [www.freemusicarchive.org](http://freemusicarchive.org/music/Dexter_Britain/2018100594213546/Calling_Instrumental_1420) [online], [cit. 27.4.2019], Calling Instrumental, dostupné z http://freemusicarchive.org/music/Dexter_Britain/2018100594213546/Calling_Instrumental_1420
6. Kei Engel, [www.freemusicarchive.org](http://freemusicarchive.org/music/Kai_Engel/The_Scope/Kai_Engel_-_The_Scope_-_06_Wake_Up) [online], [cit. 27.4.2019], Wake Up, dostupné z [http://freemusicarchive.org/music/Kai_Engel/The_Scope/Kai_Engel - The Scope - 06 Wake Up](http://freemusicarchive.org/music/Kai_Engel/The_Scope/Kai_Engel_-_The_Scope_-_06_Wake_Up)
7. Harvey, G., Martyn, C., 9. 5. 2019 Purple Planet Music [online] dostupné z <https://www.purple-planet.com/inspirational> pod názvem Hard Fought Victory
8. Mrthenoronha, [www.freesound.org](https://freesound.org/people/Mrthenoronha/sounds/397702/) [online], [cit. 30. 3. 2019] dostupné z <https://freesound.org/people/Mrthenoronha/sounds/397702/>
9. Zagi2, [www.freesound.org](https://freesound.org/people/zagi2/sounds/178126/) [online], [cit. 30. 3. 2019] dostupné z <https://freesound.org/people/zagi2/sounds/178126/>
10. nsstudios, [www.freesound.org](https://freesound.org/people/nsstudios/sounds/344276/) [online], [cit. 30. 3. 2019] dostupné z <https://freesound.org/people/nsstudios/sounds/344276/>
11. ZapSplat, [www.zapsplat.com](https://www.zapsplat.com/music/massive-explosion-heavy-distortion-ott-designed-as-an-enhancer-to-other-explosions-1/) [online], [cit. 30. 3. 2019] dostupné z <https://www.zapsplat.com/music/massive-explosion-heavy-distortion-ott-designed-as-an-enhancer-to-other-explosions-1/>
12. ZapSplat, [www.zapsplat.com](https://www.zapsplat.com/music/multimedia-button-click-5/) [online], [cit. 30. 3. 2019] dostupné z <https://www.zapsplat.com/music/multimedia-button-click-5/>
13. Sound Spark LLC, [www.zapsplat.com](https://www.zapsplat.com/music/laser-noise-synth-burst-vintage/) [online], [cit. 30. 3. 2019] dostupné z <https://www.zapsplat.com/music/laser-noise-synth-burst-vintage/>