

Phaser a jeho praktické využití

Petr Fordey

Bakalářská práce
2018



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Fordey**
Osobní číslo: **A15256**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační technologie v administrativě**
Forma studia: **prezenční**

Téma práce: **Phaser a jeho praktické využití**
Téma anglicky: **A Phaser Framework and its Use in Practice**

Zásady pro vypracování:

1. Seznamte se s webovou technologií HTML5 a CSS3. V práci popište nové funkce, které tyto technologie přinášejí.
2. Prostudujte programovací jazyk JavaScript a framework Phaser, který se používá především pro vývoj her ve webových prohlížečích.
3. Vytvořte výukové tutoriály pro framework Phaser, které usnadní jeho studium začínajícím programátorům.
4. Vytvořte podrobnou elektronickou příručku k frameworku Phaser.
5. Pomocí frameworku Phaser navrhnete a realizujete komplexnější počítačovou hru, která bude demonstrovat jeho možnosti.

Rozsah bakalářské práce: -

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **SUEHRING, Steve a Jakub ZEMÁNEK. JavaScript: krok za krokem. Brno: Computer Press, 2008. ISBN 978-80-251-2241-9.**
2. **BROWN, Tiffany B., Kerry BUTTERS, Sandeep PANDA a Ondřej BAŠE. HTML5 okamžitě: [ovládněte HTML5 za víkend]. Brno: Computer Press, 2014. ISBN 978-80-251-4296-7.**
3. **MEYER, Eric A. a Jan POKORNÝ. Eric Meyer o CSS: kompletní průvodce. Brno: Zoner Press, 2007. Encyklopedie webdesignera. ISBN 978-80-86815-64-0.**
4. **Phaser [online]. [cit. 2017-01-18]. Dostupné z: <https://phaser.io/>**
5. **GameDev Academy [online]. [cit. 2017-01-18]. Dostupné z: <https://gamedevacademy.org/>**

Vedoucí bakalářské práce:

Ing. Pavel Pokorný, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

1. prosince 2017

Termín odevzdání bakalářské práce:

25. května 2018

Ve Zlíně dne 14. prosince 2017

doc. Mgr. Milan Adámek, Ph.D.
děkan



doc. Ing. Martin Sysel, Ph.D.
garant oboru


Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 14.5.2018


.....
podpis diplomanta

ABSTRAKT

Hlavním úkolem této bakalářské práce bylo seznámit čtenáře s novinkami, které přinesla nová verze značkovacího jazyka HTML5 a kaskádové styly CSS3. Dalším úkolem bylo popsat, k čemu slouží JavaScript a co nabízí použití frameworku. Následně šlo o rozebrání jednotlivých technologií, pomocí kterých lze tvořit dokumenty ve formátu pro web. Důležitým úkolem této práce bylo také detailně popsat možnosti frameworku Phaser pro tvorbu her ve webovém formátu a vytvořit pomocí něj komplexní hru, která předvede jeho možnosti. Dalším úkolem bylo připravit učební pomůcku ve formě tutoriálů, které usnadní jeho použití začínajícím programátorům a zjednoduší jeho studium.

Klíčová slova: Phaser, framework, HTML, CSS, Web, JavaScript

ABSTRACT

Main purpose of this bachelor thesis was to show features of new version of markup language HTML5 and Cascade Style Sheets 3. Another important goal was to describe use of JavaScript and frameworks. Next important thing was explaining web technologies that are used for creating web documents. Another goal was to describe use of framework Phaser, which is framework used to games creation for HTML5. In the next part it was to demonstrate framework features and creating tutorials for it, that will serve as guide for beginner programmers that will simplify his use.

Keywords: Phaser, Framework, HTML, CSS, Web, JavaScript

Poděkování, motto a čestné prohlášení, že odevzdaná verze bakalářské práce a verze elektronická, nahraná do IS/STAG jsou totožné ve znění:

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Chtěl bych poděkovat panu Ing. Pavlovi Pokornému, Ph.D. za vedení práce, pomoc a poskytnutí rad během jejího zpracování.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 WEBOVÉ TECHNOLOGIE	10
1.1 HTML5.....	10
1.1.1 Historie.....	11
1.1.1.1 HTML 1.0.....	11
1.1.1.2 HTML 2.0.....	11
1.1.1.3 HTML 3.0.....	11
1.1.1.4 HTML 3.2.....	11
1.1.1.5 HTML 4.0 a 4.01.....	12
1.1.1.6 XHTML 1.0.....	12
1.1.2 Novinky v HTML5.....	12
1.1.3 Možnosti HTML5.....	13
1.2 CSS3.....	14
1.2.1 Novinky ve třetí verzi CSS.....	15
1.3 JAVASCRIPT.....	16
1.4 FRAMEWORKY.....	17
2 FRAMEWORK PHASER	18
2.1 HISTORIE VERZÍ.....	19
2.2 STATISTIKY.....	20
2.3 LICENCE.....	21
2.4 PŘEHLED MOŽNOSTÍ.....	21
2.4.1 WebGL a Canvas.....	21
2.4.2 Načítací jádro.....	21
2.4.3 Sprites a Spritesheets.....	21
2.4.4 Fyzikální výpočty.....	22
2.4.5 Částicové systémy.....	23
2.4.6 Kamera.....	23
2.4.7 Responzivnost.....	23
2.4.8 Vstup.....	23
2.4.9 Zvuky.....	24
2.4.10 Dlaždicové mapy.....	24
2.4.11 Rozšíření.....	24
2.4.12 Komunita a podpora.....	25
2.5 ALTERNATIVY K FRAMEWORKU PHASER.....	25
2.5.1 PixiJS.....	25
2.5.2 MelonJS.....	25
2.5.3 Engine Cocos Creator a Cocos frameworky.....	26
2.5.4 Construct 2.....	27
2.5.5 GameMaker.....	27
2.5.6 ImpactJS.....	28
2.5.7 CreateJS.....	28
II PRAKTICKÁ ČÁST	29
3 PŘÍRUČKA	30

4	TUTORIÁLY	32
5	HRA VYTVOŘENÁ POMOCÍ FRAMEWORKU PHASER	33
5.1	PŘÍPRAVA.....	33
5.2	POUŽITÝ SOFTWARE.....	33
5.3	TVORBA ZDROJOVÉ GRAFIKY.....	35
5.3.1	Sada dlaždic.....	35
5.3.2	Hráčův model.....	37
5.3.3	Nepřátelé.....	37
5.3.4	Sběratelské předměty.....	39
5.3.5	Ostatní.....	39
5.4	TVORBA ZVUKŮ.....	40
6	TVORBA HRY	41
6.1	HIERARCHIE HRY.....	41
6.2	PROPOJENÍ SOUBORŮ INDEXEM.....	41
6.3	NASTAVENÍ HRY.....	42
6.4	STATICKÁ GRAFIKA.....	42
6.4.1	Mapy.....	42
6.4.2	Ostatní statické objekty.....	44
6.5	POZADÍ A EFEKTY POČASÍ.....	45
6.6	HRÁČ.....	45
6.7	AUTO.....	47
6.8	KAMERA.....	47
6.9	SBĚRATELSKÉ PŘEDMĚTY.....	47
6.10	TELEPORTÉRY A PŘENAŠEČE.....	48
6.11	NEPŘÁTELE.....	49
6.12	ZVUKY.....	50
6.13	POČÍTADLO SKÓRE.....	51
	ZÁVĚR	53
	SEZNAM POUŽITÉ LITERATURY	54
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	57
	SEZNAM OBRÁZKŮ	58
	SEZNAM PŘÍLOH	60

ÚVOD

Počátky webu sahají do osmdesátých let 21. století. Má za sebou tedy 30 let vývoje. Ačkoliv původním účelem bylo pouze propojení počítačů kvůli pracovním důvodům, během let vývoje se jeho využití rozšířilo i k ostatním účelům a k webu se dostala i veřejnost. Přes počáteční omezené možnosti se prostřednictvím webu začaly postupně šířit i programové části kódu. Protože jazyk pro programové části nebyl pro zábavní účely dostatečný, začal se používat Flash a doplněk do prohlížeče pro jeho podporu. S příchodem nové verze značkovacího jazyka HTML5 byla většina funkcí, pro kterou byl dříve potřeba Flash nahrazena. V aktuálních verzích již není nutné používat žádné doplňky pro novou funkčnost a jsou zde obsaženy nové funkce pro vykreslení složitější grafiky, což zábavnímu průmyslu dosti pomáhá. Dnes se doplňky ve formě frameworků používají spíše pro zjednodušení syntaxe pro programátory.

Cílem práce je seznámit čtenáře s aktuálními verzemi jazyků pro tvorbu webových dokumentů, multimediálních aplikací a her a zejména se samotným frameworkem Phaser. Součástí práce bude také uživatelská příručka k Phaseru a soubor tutoriálů, který čtenářům osvětlí, jak ve frameworku pracovat a na co si dát při tvorbě her pozor. Další důležitou částí je otestování možností frameworku Phaser tím, že v něm bude vytvořena hra, kterou bude práce v praktické části popisovat.

I. TEORETICKÁ ČÁST

1 WEBOVÉ TECHNOLOGIE

Webové technologie jsou skupinou prvků, díky které mohou uživatelé zobrazovat dokumenty napsané webovým jazykem. Zjednodušeně se dá říct, že díky nim lze prohlížet webové stránky. Mezi tyto technologie se řadí internetové prohlížeče, značkovací a programovací jazyky, kaskádové styly a podpůrné knihovny či frameworky.

Většina webových stránek je napsána za použití značkovacího jazyka HTML. Formátování, které jednotlivé značky přinášejí, je definováno v kaskádových stylech CSS. Při čtení webové stránky pak internetový prohlížeč překládá tuto stránku do požadované podoby.

1.1 HTML5

Značkovací jazyk je souborem menších prvků, které společně tvoří celek. Aktuálně používaná verze značkovacího jazyka pro definování struktury hypertextových dokumentů je HTML5. Tato verze byla zveřejněna v říjnu v roce 2014 konsorciem W3C, které má na starost vývoj webových protokolů za účelem rozvoje webu. Primárním účelem nové verze je přinést podporu nových multimediálních funkcí při zachování jednoduché čitelnosti.

Jednou ze základních myšlenek jazyka HTML5 bylo odstranění nutnosti využívat zásuvné moduly a přejít na variantu, kdy bude možno využívat požadované funkce přímo v základu.

Celkový rozsah technologií, které HTML5 přináší je obrovský. HTML5 překračuje pouhý značkovací jazyk, ale nově definuje například i komunikaci prohlížeče se serverem, nebo časování a ukládání dat. Dá se také chápat jako samotná platforma, na kterou je aplikace cílena. Tvorba pak může probíhat pomocí různých nástrojů, aniž by autor osobně definoval syntaxi. [1]



Obrázek 1: logo HTML5 [2]

1.1.1 Historie

Jazyk HTML má za sebou okolo dvaceti let vývoje a možnosti, které dnes programátorům nabízí, nebyly vždy samozřejmostí. Od první verze do současnosti se samozřejmě razantně zvyšuje i počet uživatelů, kteří jej mají zájem aktivně využívat.

1.1.1.1 HTML 1.0

Jednalo se o první verzi HTML, zveřejněnou v roce 1991. Tehdy o vývoj webových stránek nebyl zájem a jazyk navíc nenabízel příliš mnoho možností. V podstatě šlo o zobrazení prostého textu, propojeného pomocí odkazů.

1.1.1.2 HTML 2.0

V roce 1994 byla uvolněna druhá verze HTML, která přidala pouze pár novinek. Mezi ně patřilo přidání grafiky a formulářů. Tento jazyk byl používán v době začátku rozmachu internetu až do roku 1997 z důvodu neúspěšnosti verze 3.0.

1.1.1.3 HTML 3.0

Návrh třetí verze jazyka HTML vznikl v době, kdy měla společnost Netscape na trhu internetových prohlížečů monopol. Prohlížeč Netscape Navigator byl tehdy nejpoužívanější a není tedy divu, že se HTML 3.0 zaměřila na něj. Možnosti, které třetí verze jazyka přinesla se nazývaly Netscape extension tags (přídavné značky pro Netscape). Ostatní prohlížeče se snažily zobrazovat stránky stejným stylem, avšak bez těchto přídavných značek. Toto ovšem nebylo dost dobře možné, a verze HTML 3.0 se brzy přestala používat. Vývojáři, kteří v té době měli HTML na starost, vydali aktualizovanou verzi 3.2, která pracovala s prvky podporovanými všemi prohlížeči.

1.1.1.4 HTML 3.2

Z důvodu normalizace zobrazení jazyka ve všech prohlížečích byla v roce 1994 založena společnost W3C, jejíž první prací byla verze HTML 3.2. Tato verze byla zaměřena spíše na optimalizaci, než vývoj. Verze byla poprvé uvedena do provozu v roce 1997 a některé prohlížeče ji podporují dodnes. V této verzi stále neexistovaly kaskádové styly CSS.

1.1.1.5 HTML 4.0 a 4.01

HTML 4.0 byla první verze, která oddělovala značky a styly. Verze 4.01 vznikla v roce 1999. Tyto verze již používaly CSS, což zjednodušilo formátování elementů jako jsou barvy, písma, nebo pozadí. Po této verzi se prohlížeč od společnosti Netscape přestal používat.

1.1.1.6 XHTML 1.0

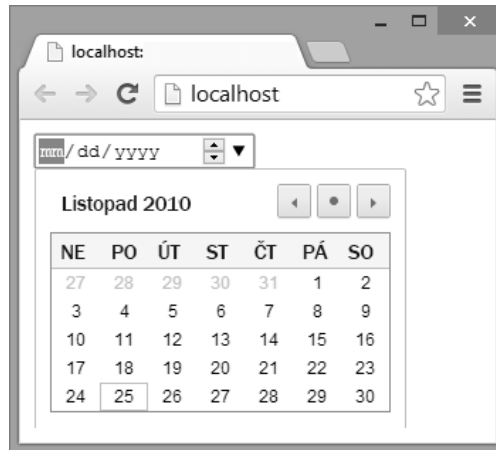
Kolem roku 2000 vydalo konsorcium W3C specifikaci XHTML a označilo jej jako doporučený značkovací jazyk. Šlo o novou odnož HTML, která používala spojení HTML a XML prvků, ale vyžadovala transformaci kódu kvůli zpětné nekompatibilitě. V této době byl nejpoužívanějším prohlížečem Internet Explorer od společnosti Microsoft. V roce 2005 byl vydán Firefox od společnosti Mozilla a v roce 2008 prohlížeč Google Chrome. Od roku 2009 se začalo přecházet na HTML5 a v roce 2010 Steve Jobs oznámil, že společnost Apple přestane podporovat Flash z důvodu nahrazením HTML5. [3]

1.1.2 Novinky v HTML5

S verzí HTML5 byly představeny nové značky, jako například značky `<video>`, či `<audio>`, které slouží pro snadné přidání multimediálních prvků, a to bez použití dalších pluginů či add-onů. Další možností použití této značky je například vytvoření zvukové odezvy pro tlačítka.

Díky elementu `<canvas>` lze vykreslovat prvky vektorové grafiky přímo pomocí jazykových syntaxí ve skriptovacím jazyce JavaScript. Není tak již nutno používat soubory obsahující grafiku, což vede ke značné optimalizaci webových dokumentů. Tohoto elementu je také možno využít aplikačním rozhraním WebGL pro vykreslování 3D grafiky, používané mimo jiné u her, které jsou v tomto jazyce napsány.

Mezi nové zjednodušující prvky patří také nové typy vstupních hodnot ze strany uživatele, jako je datum, telefonní číslo, barva, odkaz, či e-mail. Pole pro vstupní data mohou být také nově pro uživatele označena jako povinná, nebo mohou být zvýrazněna.



Obrázek 2: Vstupní typ datum

Mezi zjednodušené zápisy se řadí také DOCTYPE, u kterého už není nutné zapisovat verzi a specifikaci dokumentu. Dále byly přidány značky, které zvyšují přehlednost ve struktuře dokumentu, jako například <header>, <main>, <footer>. Nová značka pro navigaci v dokumentu <nav> umožňuje odkazovat na jinou část webu. Element <figure> usnadňuje vložení obrázku zároveň s popisem. Nové značky <article> a <section> umožňují lepší integraci do vyhledávače, která vyústí ve větší návštěvnost webu.

1.1.3 Možnosti HTML5

Díky některým novým značkám bylo možné zdokonalit multimediální prvky. Jako typický příklad takového vylepšení lze uvést animované náhledy videí na webu Youtube. Podpora některých prvků se ovšem liší podle použitého prohlížeče. Jedním z důvodů je použití různých vykreslovacích jader. Díky novým multimediálním možnostem také spousta webů přestala využívat **Flash**, který pro ně již není nutný. Pro funkčnost těchto technologií je ovšem nutno implementovat je do kaskádových stylů.

Díky novinkám v podobě aplikačních rozhraní jsou zde další možnosti. Aplikační rozhraní **Geolocation** umožňuje lokalizaci počítače, z kterého je dokument prohlížen. Rozhraní **Drag and Drop** umožňuje přímé přetažení objektu na zvolenou pozici v dokumentu. Díky rozhraní **Web Storage** mohou webové aplikace uchovávat data přímo v prohlížeči uživatele a ta nemusí již být přenášena na server. Data již nemusí být uchovávána v souborech cookies, což vede ke zrychlení načítání webu. Rozhraní **Web Workers** umožňuje spuštění skriptu na pozadí nezávisle na akcích uživatele. Díky rozhraní **Media Capture** může uživatel povolit přístup k mikrofonu, webkameře, souborovému systému a následně těchto výhod využívat. Mezi další novinky patří možnost používat aplikaci bez internetového připojení. [4]

1.2 CSS3

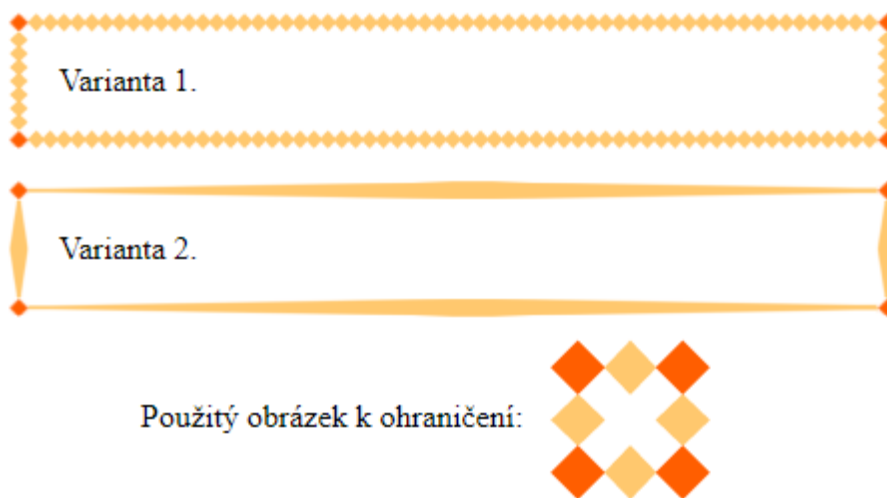
Kaskádové styly jsou standardem pro doplňkové formátování webových dokumentů. Definiují formátování jednotlivých značek pro dokumenty ve formátu HTML, XHTML a XML. Specifikace těchto standardů vydává stejně jako specifikace HTML konsorcium W3C. Jednotlivé verze se postupně updatují a přinášejí nové možnosti. Díky oddělení zápisu stylu od značek stačí definovat styl značky pro všechny následující použití jednou. To vede k následné optimalizaci. Mezi elementy formátování, které lze díky oddělení stylu a značky definovat pro celý dokument patří například styl písma, jeho barva, odsazení, či různé efekty při interakci s jednotlivými elementy. Další možností je vytvoření jednotlivých verzí stránky pro různé typy zařízení. Při návštěvě stránky nebo otevření dokumentu je pak detekován typ zařízení a podle toho se zobrazí požadovaná verze. Výhody se ale neomezují pouze na usnadnění zápisu, ale přinášejí také možnost rozsáhlejšího formátování nebo možnost uložení stylu do paměti cache, což vede k rychlejšímu načtení dokumentu. V dnešní době již kaskádové styly udávají kompletní vzhled stránky. Nevýhodou je, že některé prohlížeče volí jiný přístup ke čtení CSS a následkem toho vznikají různé interpretace stránky. Aktuální verzí jazyka pro zápis kaskádových stylů je CSS3, kde byly pro větší přehlednost rozděleny jednotlivé typy specifikací na několik modulů. Aktuální počet modulů je okolo 50. Z důvodu různého stavu dokončení jednotlivých modulů je podpora pro všechny moduly nasazována postupně. Výhodou je zpětná kompatibilita s CSS2. [5]



Obrázek 3: Logo CSS3 [6]

1.2.1 Novinky ve třetí verzi CSS

Z pohledu grafického designu mezi nejzajímavější novinky v aktuální verzi patří možnost přidání zaoblených rohů pro požadované elementy stránky pomocí definování poloměru zaoblení. Staré ohraničující prvky lze nyní nahradit ohraničujícím obrázkem. Velikost ohraničení již není vázána na velikost prvku uvnitř, ale lze ji definovat pomocí nového atributu. Pro vybrané elementy či text je také nově možno přidat stín. Další grafickou novinkou je možnost přidání animace, nebo přechodu. Pro každou animaci lze vytvořit smyčku pro opakování, pro přechod poté definovat styl vektoru přechodové animace (škálování, posun, rotace, překlápění atd...). Vylepšená je také škála barev, kterou je možno použít. Zatímco dříve se používaly pouze základní typy barev, dnes lze využít vektorových přechodů jako je kruhový, nebo gradientní přechod. Je také možno vybírat z různých barevných modelů. Z hlediska pozadí přibyla nově možnost skládání více pozadí na sebe, což má za následek podobné chování jako v programech pro editaci grafiky. Zlepšila se také podpora průhlednosti prvků. [7]



Obrázek 4: Princip ohraničení obrázkem [8]

S novou verzí přišla také možnost výběru z více variant písma. O to se mimo jiné zasloužila služba Google Fonts od společnosti Google a standart Typekit od společnosti Adobe. Velikost postavení těchto firem na trhu pomohlo k adaptaci jejich standardů.

Mezi nové prvky pro návrh stránky patří Selectory, které určují na které značky se má definovaný styl uplatnit. Novinkou usnadňující rozvržení stránky je také prvek calc(), který umožňuje stanovení měřítka jednotlivých elementů, na které se poté aplikují stanovené hodnoty pro umístění. To návrháři poskytne zpětnou vazbu, díky čemuž může upravit hodnoty

na optimální. Díky prvku `columns` lze text na stránce rozdělit do oddílů jednodušeji, než kdy dříve.

1.3 JavaScript

JavaScript je skriptovací jazyk, který se používá převážně pro manipulaci s webovými dokumenty. Je možno jej vložit přímo do HTML kódu, nebo využít externího souboru. Na jeho vývoji se podílely významné společnosti Netscape Communications Corporation a Mozilla Foundations, které stojí také za vývojem známých webových prohlížečů. Využití skriptovacího jazyka je záměrně omezené. Název JavaScript nemá nic společného s programovacím jazykem Java. Byl vybrán pouze z důvodu popularity jazyka Java v době jeho vzniku. Původním názvem byl ECMAScript. Spolu s HTML a CSS patří JavaScript do trojice základních jazyků pro webové dokumenty. Jeho podpora je stejně jako u ostatních jazyků závislá na použitém webovém prohlížeči. JavaScript původně vznikl pro tvorbu počítačového programu, spustitelného přímo ve webovém prohlížeči. Díky němu se při přenosu webového dokumentu může provést programová část kódu. JavaScript také umožňuje provést různé programové operace při interakci s jednotlivými webovými elementy. Je ho navíc možno využít pro přístup k databázím a ke tvorbě her. Provádění JavaScriptových operací probíhá v internetovém prohlížeči na straně klienta. Oproti například PHP se jedná o klientsky orientovaný jazyk. [9]

Jednou z částí programového kódu, kterou JavaScript nabízí, je definování podmínek. Podmínky fungují stejně jako v programovacích jazycích C. Pro každou podmínku je možno definovat akci, která bude provedena, pokud je podmínka splněna, či nesplněna. Podmínky je možno skládat na sebe, což vede k více možnostem výstupu. Pro jejich definování se používají operátory. Ty lze rozdělit na přiřazovací, početní a logické. Přiřazovací operátory slouží k nastavení hodnot proměnných. Početní operátory slouží k matematickým operacím. Logické operátory se používají při větvení.

Další možností je tvorba funkcí. Na začátku se definuje funkce, která je následně volána. Volání může být spuštěno libovolnou operací, není ho tedy nutné spouštět přímo. U funkcí je možné definovat parametry, které se napíší do závorky. Tyto parametry poté mohou měnit výstup funkce. Je také možno vytvořit událost. Událost definuje požadovanou akci, při které se daná funkce provede.

JavaScript také umožňuje deklaraci proměnných. Do proměnných se ukládají hodnoty. Proměnná poté představuje uloženou hodnotu.

Stejně jako u ostatních programovacích jazyků je i u JavaScriptu možno vkládat komentáře. Komentář způsobí, že část, která je označena jako komentář, bude programem ignorována a nebude tudíž provedena. V JavaScriptu se komentáře značí různými způsoby, podle typu komentáře. Pro jednořádkový komentář se používá značka `//`. Pro víceřádkový komentář se používá značka `/*` na začátku komentáře a konec komentáře se značí `*/`. U komentářů podmínek se navíc přidává znak `@`.

1.4 Frameworky

I přes širokou škálu možností, které jednotlivé nástroje poskytují, se občas vyskytnou případy, kdy nabízené možnosti nestačí. K tomuto účelu vznikly aplikační a webové rámce (frameworky), které nabízejí rozšíření základních funkcí a zjednodušují vrstvené příkazy do jednodušších. Protože se jednotlivé frameworky mohou soustředit na konkrétní odvětví, nebo na konkrétní cíl, mohou nabídnout specifické funkce. Na druhou stranu se frameworky zaměřují hlavně na zjednodušení běžných problémů, aby tím uživateli ulehčily práci. Tvůrce kódu si pak může vybrat, zda mu použití daného rozšiřujícího rámce přinese výhody, či nikoliv. Většina frameworků pro vývoj webových aplikací se zaměřuje na formulaci rozsáhlých syntaxí do jednoduchých zápisů. Aplikační rámce mohou fungovat na bázi poskytnutí knihoven s předpřipraveným obsahem, nebo překladačů, či aplikačních rozhraní.

Většina frameworků je charakteristická tím, že funguje na principu obrácené kontroly. V běžných případech má provádění kódu a posílání dotazů na jednotlivé knihovny na starost samotný program. U frameworku to funguje opačně - framework v případě potřeby volá programátorův kód a má nad ním plnou kontrolu. [10]

Framework můžeme rozdělit do dvou částí. První částí je kód, který je určen konkrétním projektem. Tuto část musí programátor vymyslet osobně. Druhou část tvoří kód, který je plně definován tvůrcem frameworku. Sem patří definice jednotlivých funkcí a základních vztahů mezi nimi. [11]

Mezi nejznámější frameworky pro webové aplikace patří ASP.NET, Ruby on Rails, Django, nebo AngularJS.

2 FRAMEWORK PHASER

Phaser je jedním z nejrozšířenějších, zdarma dostupných frameworků, zaměřený na vývoj her v jazyce HTML5, vyvinutý firmou Photon Storm. Používá upravenou verzi vývojářského jádra PixiJS pro vykreslování obsahu pomocí rozhraní WebGL či Canvas. Pokud vývojář explicitně nedefinuje specifické vykreslovací rozhraní, proběhne výběr rozhraní automaticky. Pokud je zařízení schopno vykreslovat grafiku přes WebGL, je použito WebGL. Pokud ne, použije se klasické vykreslování přes Canvas. Použití WebGL ovšem nabídne hráči plynulejší zážitek. [12]

Co se zvuku týká, Phaser podporuje nejmodernější rozhraní Web Audio, poskytující vícekanálový zvuk s různými zabudovanými efekty, které ale nemusí být dostupné vždy. Z tohoto důvodu je implementovaná i podpora pro starší zvukové rozhraní. To funguje tak, že spojí více zvuků do balíčku a přehrání jednotlivých částí využívá oddělovacích záložek, podobných jako při střihu videa či zvuku. Výběr zvukového rozhraní je opět plně automatizován podle podpory v daném zařízení. [13]



Obrázek 5: Logo frameworku Phaser [13]

Díky použití webového programovacího jazyka umožňuje podporu napříč více platformami, ať už pro osobní počítače, mobilní platformy iOS a Android. Pro vývoj využívá skriptovací jazyky JavaScript nebo TypeScript. Hlavním cílem je poskytnout vývojářům předpřipravené funkce, které jim usnadní práci. Při použití wrapperu je pak možno zabalit hru a vytvořit nativní aplikaci, bez nutnosti přístupu přes webový prohlížeč.

Kvůli používání pro tvorbu počítačových i mobilních her současně jsou poskytnuty podpory pro různé druhy vstupů. Zatímco u hraní na počítači se používá klávesnice a myš, u moderních mobilních zařízení se počítá spíše se vstupními impulzy pro ovládání přes dotykovou plochu. V dnešní době má většina mobilních zařízení podporu vylepšeného dotykového vstupu multitouch a proto je tato podpora ve frameworku zabudována také.

Důvodem, proč by měli vývojáři dát přednost frameworku Phaser před jeho alternativami, jsou dlouhá léta zkušeností tvůrců frameworku Phaser s vývojem webových her a dalších aplikací. Další výhodou je poskytnutí zabudovaného nástroje pro načtení zdrojových herních souborů.

Kdyby autorům her nějaké funkce chyběly, je možno použít další pluginy, které požadované funkce doplní. Tyto doplňky mohou být vyvinuty i externími vývojáři a to díky zveřejnění kompletního zdrojového kódu frameworku. Ačkoliv je framework dostupný zdarma, tvůrci frameworku neomezili vývojářům pluginů možnost nabízet tyto doplňky za úplatu a proto je většina pluginů zpoplatněná. [13]

2.1 Historie verzí

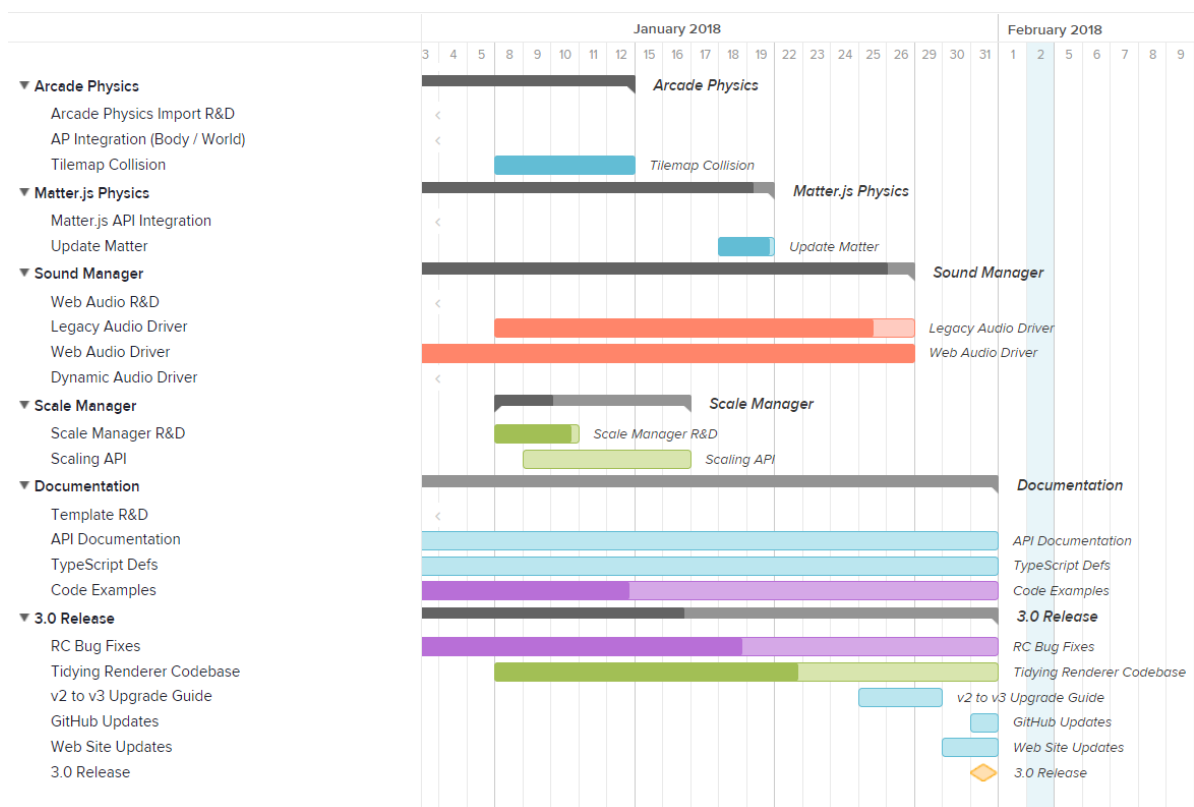
Před prací na tvorbě Phaseru se autor podílel na vývoji herního engine pro jazyk HTML5 Kiwi.js. Verze Phaseru 0.5, která byla první uvolněnou verzí, postrádala některé základní funkce. První významná verze frameworku Phaser (1.0) byla zveřejněna v dubnu roku 2013. Tato verze již obsahovala podporu vstupu pro dotykové zařízení a byla zde také mimo jiné doplněna podpora pro fyzikální kolize. Autor také zveřejnil kompletní dokumentaci k jednotlivým funkcím.

Postupně byla přidána podpora pro WebGL. Následně bylo upuštěno od jednoduchého JavaScriptu a framework přešel na Pixi.js. Další důležitou verzí byl tak až Phaser 2.0, vydaný v březnu 2014, ve kterém byly opraveny některé původní chyby a zjednodušena struktura programu. [14]

V době psaní této práce jsou doporučenými verzemi verze CE (Community edition – aktuální 2.10). Tyto verze nahradily verzi 2.6.2 a liší se od ní hlavně v přístupu. Zatímco obsah starších verzí byl kompletně v rukou společnosti Photon Storm, v edici CE si komunita sama určuje, co do ní přidá. Tvůrci zveřejnili kód na komunitní vývojářské stránce GitHub a předali několika komunitním vývojářům práva na zápis změn. Ostatní mohou nadále posílat požadavky na nové možnosti, které budou zváženy a v případě využitelnosti přidány. Tento krok byl ze strany tvůrců učiněn z důvodu, aby se mohli věnovat tvorbě kompletně přepracované nové verze.

Tvůrci frameworku aktuálně připravují verzi 3, která je ve vývoji od října roku 2016. Většina kódu je ve verzi 3 kompletně přepracována, avšak bez vzrůstu náročnosti na tvorbu pro autory her, a to díky podobnému aplikačnímu rozhraní mezi verzí 2.x a 3. Nově je rozšířena

vzdálenost, pro kterou je možno definovat parametry jednotlivých objektů. Bohužel verze 3 s sebou přináší jednu obrovskou nevýhodu, kterou je nekompatibilita s kódem, který fungoval ve verzích 2.x. S přechodem na verzi 3 také Phaser přestává využívat PixiJS a namísto něj používá vlastní vykreslovací jádra pro Canvas a WebGL. V případě zájmu je možno si ji vyzkoušet v beta verzi, ačkoliv je nutno počítat s tím, že nemusí být tak stabilní jako aktuálně doporučená verze CE 2.1 a nemusí spolehlivě fungovat všechny novinky. O změnách tvůrci informují každý týden a nabízí ukazatel stavu vývoje jednotlivých částí pomocí Ganttova diagramu.



Obrázek 6: Postup vývoje Phaseru [15]

2.2 Statistika

Týdenní novinky si přečetlo již přes 8800 vývojářů. Na serveru GitHub si ho přidalo do seznamu oblíbených položek přes 17 tisíc lidí. Phaser nabízí více než 700 výukových příkladů. Je ve vývoji už 5 let. Bylo vydáno přes 80 verzí a k jeho vývoji přispělo již více než 100 lidí. [12]

2.3 Licence

Majitelem Phaseru je společnost Photon Storm. Na framework se vztahují autorská práva. Phaser patří mezi frameworky se svobodnou licenci. Je dovoleno ho zdarma využívat a modifikovat bez jakýchkoliv omezení a prodávat projekty vytvořené na jeho bázi za podmínky, že bude obsažena původní licence s oznámením o autorských právech. Modifikace mohou být prodávány bez poskytnutí zdrojového kódu. [16]

2.4 Přehled možností

Phaser nabízí širokou škálu možností pro tvorbu her a multimediálních aplikací. Tyto možnosti lze rozdělit do skupin podle podobností a souvislostí. [13]

2.4.1 WebGL a Canvas

Phaser obsahuje zabudovanou podporu pro vykreslovací jádra WebGL a Canvas. Novější a kvalitnější metodou je WebGL, ale stále nemá tak širokou podporu jako Canvas. V aktuální verzi je framework schopen automaticky používat lepší metodu podle možností zařízení. Definovat velikost hry a použití automatického vykreslovacího rozhraní lze následovně:

```
var game = new Phaser.Game(1280, 720, Phaser.AUTO, 'gameDiv',);
```

Obrázek 7: definice hry

2.4.2 Načítací jádro

Vývojáři se zaměřili na jednoduchost přidání zdrojových souborů do hry. K přidání stačí velice jednoduchý zápis, ať už jde o načítání zvuků, obrázků, skupin, nebo XML dat. Všechny načtené soubory framework uchovává v mezipaměti cache. Příklad načtení zdrojového obrázku či animace může vypadat následovně: [17]

```
preload: function() {  
    this.load.image('pozadi', 'soubory/pozadi.png');  
    this.load.spritesheet('hrac', 'soubory/animace.png', 1725, 529, 5);  
}
```

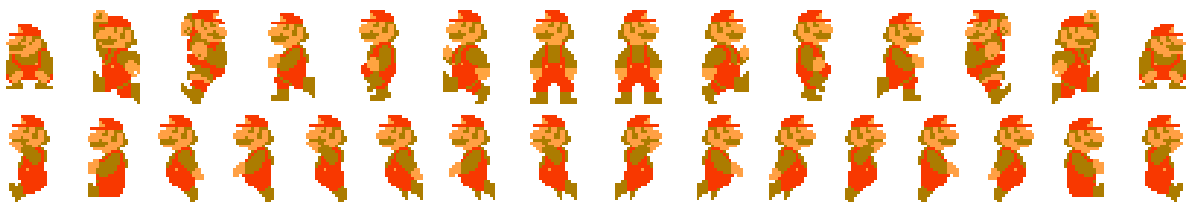
Obrázek 8: Načtení zdroje

2.4.3 Sprites a Spritesheets

Jako Sprite se označuje objekt importovaný do hry. Na těchto grafických spritech je možno provádět různé geometrické operace, jako je škálování, rotace, nebo rozmístění. Tyto objekty je možno také umísťovat ve skupinách, nebo importovaný objekt kreslit vzorkovacím

štetčem. Pro tyto objekty je možno nastavit různé efekty při interakci s nimi, nebo s nimi v případě potřeby pohybovat.

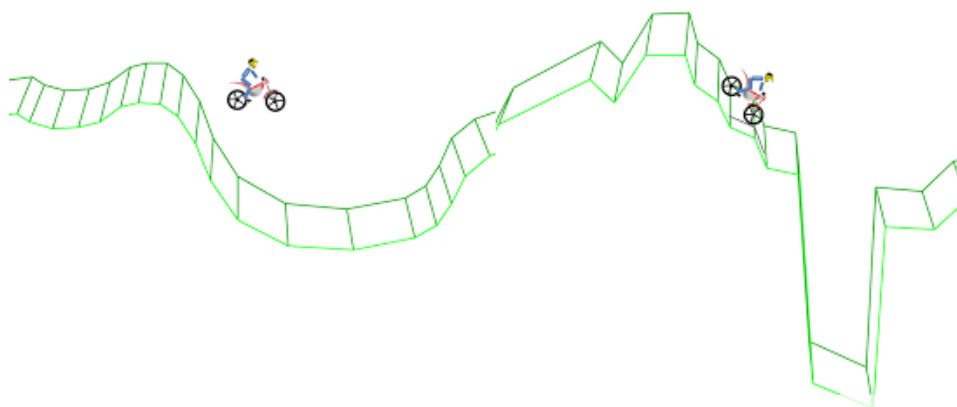
Spritesheet funguje na stejném principu jako Sprite s tím, že se jedná o skupinu, která se používá jako animační balíček. To představuje obrovskou výhodu v hierarchii souborů a také z hlediska zjednodušení zápisu. Není tak nutno definovat jednotlivé animační snímky a jejich geometrické operace, nýbrž stačí provést pouze jeden zápis, ve kterém se specifikuje kolik animačních snímků tvůrce importuje, a jakou velikost mají jednotlivé snímky. Jako spouštěč impulsu pro změnu snímku lze nastavit libovolný vstup. Například při stisknutí klávesy šipky vpravo se může snímek hráče přepnout na postavu otočenou doprava. Tím lze nasimulovat pohyb.



Obrázek 9: Příklad Spritesheetu [18]

2.4.4 Fyzikální výpočty

Phaser nabízí možnost fyzikálních výpočtů a kolizí. Nabízí výběr ze tří přesností. Použití jednotlivých přesností s sebou nese daň v podobě náročnosti výpočtů. Díky možnosti použít nenáročnou arkádovou fyziku je možno vytvořit hru s podporou fyzikálních výpočtů i pro starší zařízení. Simulace fyziky s sebou logicky přináší i simulaci gravitačního pole.



Obrázek 10: Ukázka účinku fyzikálních výpočtů na zobrazení objektů [19]

2.4.5 Částicové systémy

Částicové systémy jsou pro hrátelnost velmi důležité. Jejich prostřednictvím lze do hry přidat například střelivo, exploze, nebo i efekty počasí, jako je sníh či déšť. Díky těmto systémům je také možno tvořit vodu, lávu, nebo další kapaliny. Dalším použitím mohou být animace nepohyblivých prvků při interakci.

Jako příklad lze uvést následující funkci pro emitaci načteného zdroje. Druhý řádek zde určuje místo pro emitaci. Třetí řádek definuje částici, která má být emitována. Čtvrtý řádek limituje počet současných částic maximálně na 10 a určuje jejich životnost. [20]

```
function create() {  
    emitter = game.add.emitter(game.world.centerX, 500, 500);  
    emitter.makeParticles('emitovanyzdroj');  
    emitter.start(false, 60, 10);  
}
```

Obrázek 11: Přidání částicového systému

2.4.6 Kamera

Podpora kamery jako samostatného prvku přináší základní možnosti pro změnu úhlu pohledu, či perspektivy. Navíc se díky efektům kamery dá vytvářet zpětná vazba pro hráče. Je možno simulovat například otřes, či roztočení. Efektem pro přiblížení kamery se dá navodit důležitost aktuální situace ve hře. Autor také může definovat, zda má kamera následovat hráčovu pozici, nebo ne.

2.4.7 Responzivnost

Phaser podporuje vývoj her pro různé druhy platform. Vzhledem k tomu, že jednotlivá zařízení mají různé velikosti obrazovky, poměry stran a rozlišení, měla by hra na všech zařízeních vypadat správně. Díky responzivnímu designu je tento požadavek do značné míry řešen automaticky.

2.4.8 Vstup

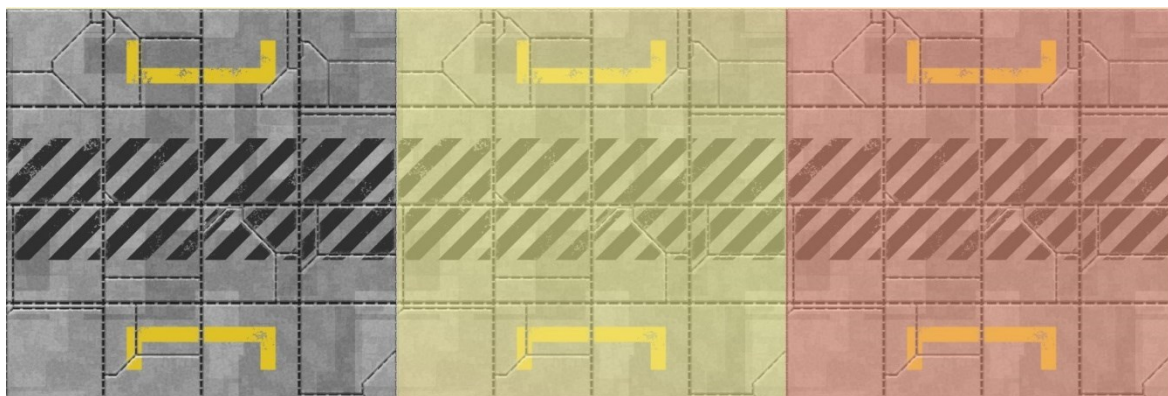
Kvůli použití různých druhů zařízení je také nutno podporovat různé druhy vstupu. Phaser podporuje jak standartní vstup ve formě myši a klávesnice, tak vstup prostřednictvím dotykové obrazovky mobilních zařízení. Lze využívat kombinaci více vstupů najednou, nebo definovat gesta. [12]

2.4.9 Zvuky

Ať jde o efekty při interakci nebo o hudbu, zvuk značně vylepšuje dojem ze hry. Phaser nově nabízí podporu Web Audia, ale v případě, že dané zařízení nemá plnou podporu, může automaticky přepínat na starší zvukové rozhraní HTML. Díky přidávání zvukových zdrojů je možno vytvářet efekty, a pomocí zabudovaných nástrojů je přímo modifikovat podle potřeby. Lze tak například vytvořit zvukovou smyčku pro lepší optimalizaci.

2.4.10 Dlaždicové mapy

Dlaždicové mapy nabízejí možnost využít opakovaně importovanou texturu a následně ji roztáhnout tak, aniž by to porušilo její kvalitu. Princip je založen na tom, že tam, kde jedna dlaždicová mapa končí, hned další začíná. Tyto mapy je také možno zrcadlit, popřípadě roztahovat v různých poměrech. Kromě standartních typů opakování dlaždicových map je možno importovat CSV soubory, nebo textová data map, která v číselné soustavě definují pozici rozmístění jednotlivých prvků. To eliminuje například nutnost kreslit celé pozadí, namísto toho stačí definovat opakovanou texturu.



Obrázek 12: Roztažitelnost dlaždicové mapy [21]

2.4.11 Rozšíření

Pokud by vývojářům her nestačily možnosti, které framework Phaser nabízí v základu, je zde možnost stažení jednotlivých doplňků, které přidávají nové funkce. Některé jsou zdarma, některé zpoplatněné. Některá rozšíření se namísto nových funkcí zaměřují hlavně na zjednodušení stávajících zápisů funkcí. Mezi rozšíření doporučená přímo tvůrci Phaseru patří: [22]

- Box2D – Plugin zaměřený na fyziku, s optimalizací výkonu. Autoři slibují, že zařízení nepocítí takovou zátěž ani při použití náročnějších fyzikálních výpočtů.

- Particle Storm – Plugin pro rychlé vytvoření složitých částicových efektů. Plugin obsahuje přes 200 předpřipravených scén, které slouží jako příklad využití. Nabízí předpřipravené ohnivé efekty a podobně.
- Virtual Joystick – Plugin, který přidá do hry virtuální joystick pro ovládání her na mobilních zařízeních pro ty hráče, kteří preferují tento typ ovládání před používáním gyroskopu. Kromě joysticku nabízí také různé typy obvykle používaných tlačítek.

Ostatní internetové stránky nabízejí ke stažení i jiná rozšíření.

2.4.12 Komunita a podpora

Vzhledem k rozsáhlosti využití Phaseru je velikost komunity obrovská. Díky zpětné vazbě od jednotlivých uživatelů a dlouhodobému vývoji bylo opraveno nesčetné množství chyb, na které uživatelé narazili. Vzhledem k počtu vyvinutých her mohla být doplněna řada významných funkcí, které dřív mohly při vývoji her chybět. Díky vysoké oblíbenosti je také k dispozici velice podrobná dokumentace, která usnadňuje první kroky.

2.5 Alternativy k frameworku Phaser

Phaser samozřejmě není jediným frameworkem pro tvorbu her využívajícím platformu HTML5. Na trhu je k dispozici výběr z dalších nástrojů a každý z nich má své výhody a nevýhody. Některé nástroje nejsou přímo frameworky, ale kladou si za cíl stejný výstup. [23]

2.5.1 PixiJS

Za jednu z alternativ lze považovat zdarma dostupný engine PixiJS. Toto řešení taktéž využívá jazyka HTML5. Tento engine využíval dříve samotný Phaser jako renderer. Byl navržen především pro tvorbu 2D aplikací. Mezi výhody se řadí jeho schopnost automaticky přepínat mezi vykreslovacími metodami WebGL a Canvas a podobnost se skriptovacím jazykem ActionScript. Bohužel se nejedná o kompletní řešení, jelikož PixiJS poskytuje pouze renderer. [24]

2.5.2 MelonJS

Podobným řešením jako Phaser je framework MelonJS, který je rovněž zdarma. Již v základní verzi obsahuje možnost přidat detekci kolizí, fyzikální výpočty, podporu dlaždicových map, skládání několika vrstev grafiky na sebe, škálování na různě velké obrazovky a

mnoho dalšího. Hlavním pozitivem je ovšem editor map, který umožňuje tvořit mapy bez použití textových příkazů. Zaměřuje se zejména na hry běžící přímo v prohlížečích, ale díky dnešní architektuře mobilních zařízení je zde jako u všech ostatních řešení možnost využití wrapperu a vytvoření mobilní aplikace. Z hlediska kompatibility podporuje všechny webové prohlížeče s majoritním zastoupením na trhu. Tvůrci frameworku také umožňují vyzkoušet si přímo na webu hru, kterou vytvořili ve svém frameworku, což poskytuje případným zájemcům obrázek využitelnosti tohoto frameworku pro jejich účely. [25]

2.5.3 Engine Cocos Creator a Cocos frameworky

Společnost Cocos nabízí alternativ hned několik. Všechny její nástroje spadají do svobodné licence. Jednou možností je použití engine Cocos Creator. Engine funguje na bázi aplikace, vytvořené v jazyce C++, která je zaměřena na vývojáře, kteří se nechtějí příliš zdržovat psaním kódu. Aplikace je dostupná jak pro Windows, tak pro MacOS. Díky podpoře většiny skriptů se není nutno učit nic dalšího. Na stránkách výrobce nabízí rozsáhlou dokumentaci a komunita je doplňuje nespočtem videí obsahujících tutoriály. Nabízí také možnost využití 3D modelů, importování vazeb a zabudovaný nástroj pro debugging. Stejně jako jiné alternativy umožňuje použití různých doplňků. Oproti ostatním řešením nabízí podporu vykreslování přes standardní OpenGL.

Kromě engine Cocos Creator je od stejného studia k dispozici i samotný framework Cocos2d-x, psaný taktéž v C++. Nejčastěji se používá pro tvorbu her pro mobilní zařízení, ale zvládá i tvorbu zaměřenou pro počítač. U her pro mobilní platformy má podíl okolo 20 % a jde pravděpodobně o nejpoužívanější řešení vůbec. Může se pyšnit tím, že díky němu byly vytvořeny již tisíce her, včetně těch nejstahovanějších, jako je Badland, Clash of Kings, Castle Clash a dalších. Stejně jako engine Cocos Creator podporuje vykreslování přes upravenou verzi OpenGL. Tvůrci navíc nabízejí nástroj SDKBOX, který by měl zjednodušit integraci dalších SDK (sad nástrojů pro vývojáře).

Jako odnož projektu engine Cocos2d-X vznikla i alternativa Cocos2d-HTML5 pro tvorbu 2D HTML5 her. Oproti verzi X vývojář narazí na mnohá omezení. HTML verze také podporuje pouze 2D grafiku. Jediná použitelná dokumentace je dostupná přímo na GitHubu a odkazuje na domovskou stránku, která je dostupná pouze v čínštině. Pokud vývojář hry potřebuje dokumentaci, musí si vystačit s tou, která byla vytvořena pro Cocos2d-X.

Společnost Cocos je opravdu gigantem a kromě vysokého zastoupení na trhu také spolupracuje s korporacemi jako je Samsung, Google, Facebook, nebo s výrobcí čipů jako jsou Qualcomm, Intel, či ARM. [26]

2.5.4 Construct 2

Dalším z řad řešení pro tvorbu HTML her, které nevyžadují znalosti kódování je aplikace Construct 2 od studia Scirra. Studio Scirra nabízí placenou verzi i verzi zdarma. Tato varianta je nabízena pro ty, kteří chtějí svoji hru vytvořit co nejrychleji. Tvůrcům hry pak stačí pouze sehnat si zdrojové soubory, které budou ve hře použity. Hra je vytvořena pomocí jednoduchého přetažení připravených herních souborů do scény. Poté tvůrci hry stačí jen vybrat vložený objekt a označit ho jako jeden z předdefinovaných typů herního objektu, které jsou poskytnuty od autorů aplikace. Pokud by to nestačilo, je zde také možnost editovat základní atributy objektu. Pro interakci objektů mezi sebou jsou opět předpřipraveny různé druhy akcí. Aplikaci Construct 2 je možno získat ze stránek výrobce, nebo z distribuční platformy Steam. Výhodami této aplikace jsou jednoduchost celého řešení, okamžitá odezva náhledu hry a rychlost tvorby. Aplikace je dostupná pouze pro Windows a výstupem je hrubý kód v jazyce HTML5, který na slabších mobilních zařízeních není příliš použitelný. Nevýhodou pak může být také to, že plná verze není zdarma a možnosti se omezují na předdefinované typy od tvůrců. [27]

2.5.5 GameMaker

Podobným nástrojem je i GameMaker. Vývoj probíhá pomocí aplikace, kterou je možné provozovat na Windows, nebo na MacOS. Aplikace je na trhu již téměř 20 let a má za sebou dlouhý vývoj. GameMaker nabízí zabudovaný wrapper pro tvorbu EXE souborů pro tvorbu her na platformě Windows. Samotná tvorba probíhá v jednoduchém prostředí, hra vzniká přetahováním jednotlivých souborů a objektů. Autoři opět nabízejí předdefinované události, či typy akcí. Je zde ale také implementován doplňující programovací jazyk, bohužel se jedná o jazyk, který tvůrci vytvořili sami, proto je nezbytné naučit se příkazy. Aplikace automaticky podporuje spuštění hry na většině platform, aniž by bylo nutno jakkoliv upravovat kód. Kromě možnosti tvořit nativní aplikace na jednotlivé platformy je zde také možnost pro tvorbu her v HTML5. Oproti ostatním řešením podporuje i herní konzole PS4 a Xbox One. Autoři umožňují vyzkoušení zdarma, ale plná verze je k dispozici za 200 dolarů. [28]

2.5.6 ImpactJS

ImpactJS je dalším z řady JavaScriptových frameworků pro vývoj her v jazyce HTML5. Výhodou je podpora mobilních zařízení a poskytnutí editoru map. Na stránkách výrobců je také k dispozici několik desítek vzorových scén, u kterých je možno stáhnout si zdrojový kód. Velkou nevýhodou je, že se nejedná o bezplatné řešení. Tvůrci si za použití tohoto nástroje účtují 99 dolarů. Oproti Phaseru tento framework také není pravidelně aktualizován. Dostupná verze je již několik let stará. Jeden z komunitních tvůrců tento framework rozšířil a vytvořil verzi Impact++. [29]

2.5.7 CreateJS

CreateJS je spojením několika JavaScriptových knihoven pro usnadnění vývoje. Všechny tyto knihovny pocházejí od stejného autora. EaselJS poskytuje integraci s vykreslovacím modulem Canvas pro snadné vykreslování. Knihovna TweenJS slouží k animování a nastavení JavaScriptových parametrů. Knihovna Sound JS zpracovává zvuk a knihovna PreloadJS zajišťuje načítání jednotlivých zdrojů. Soubor těchto knihoven tvoří celek, který pracuje společně. Jedná se o nástroj, který se zaměřuje na vývoj obecného obsahu, není specificky cílen na hry. Proto může být tvorba pro začínající programátory složitější. Na druhou stranu má širší využití a lze díky němu tvořit například i webové reklamy. Výhodou je také otevřená licence a poskytnutí zdarma. [30]

Název	Cena	Popularita	Hodnocení	Počet ★ (github)
Construct 2	Zdarma		★★★★★	—
ImpactJS	99 dolarů		★★★★★	—
EaselJS	zdarma		★★★★★	6625
Phaser	zdarma		★★★★★	17355
pixi.js	zdarma		★★★★★	17518
GameMaker	200 dolarů		★★★★★	—
melonJS	Zdarma		★★★★★	2118
Cocos2d-X	Zdarma		★★★★★	11508
Cocos Creator	Zdarma	Není k dispozici	Není k dispozici	386
Cocos2d-HTML5	Zdarma	Není k dispozici	Není k dispozici	2340

Obrázek 13: Porovnání alternativ

II. PRAKTICKÁ ČÁST

3 PŘÍRUČKA

Příručka k frameworku slouží jako dokumentace, ukazuje jeho specifika a poskytuje celkový přehled o jeho možnostech. Příručka je dostupná na přiloženém DVD. Nachází se v dokumentu příručka.pdf. Příručka obsahuje celkem 20 stran.

Seznam kapitol příručky:

1. O FRAMEWORKU

- 1.1. Licence
- 1.2. Možnosti frameworku
 - 1.2.1. WebGL a Canvas
 - 1.2.2. Načítací jádro
 - 1.2.3. Sprites a Spritesheets
 - 1.2.4. Fyzikální výpočty
 - 1.2.5. Částicové systémy
 - 1.2.6. Kamera
 - 1.2.7. Responzivnost
 - 1.2.8. Vstup
 - 1.2.9. Zvuky
 - 1.2.10. Dlaždicové mapy
 - 1.2.11. Rozšíření
 - 1.2.12. Komunita a podpora

2. NÁSTROJE PRO TVORBU

- 2.1. Textový editor
- 2.2. Grafický editor
 - 2.2.1. Placené řešení
 - 2.2.2. Alternativní řešení zdarma
- 2.3. Server
- 2.4. Webový prohlížeč
- 2.5. Internet
- 2.6. Dodatečný software k tvorbě map

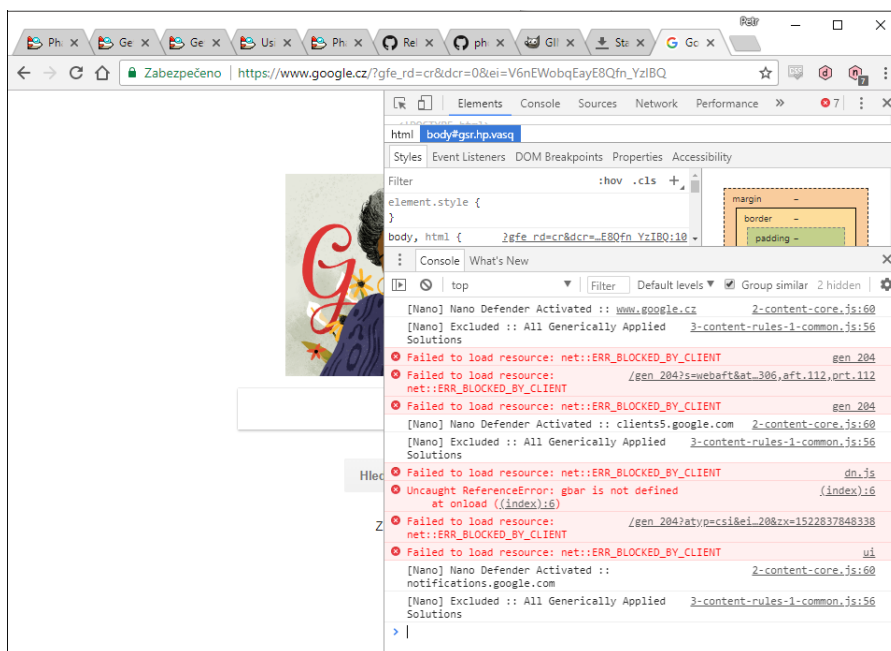
3. STAŽENÍ FRAMEWORKU A NASTAVENÍ

- 3.1. Stažení frameworku
 - 3.1.1. Použití v cloudu

- 3.2. Propojení frameworku a úrovní
4. HERNÍ ZDROJE
 - 4.1. Pozicování zdrojů
5. STRUKTURA PROGRAMU
 - 5.1. Preload
 - 5.2. Create
 - 5.3. Update
 - 5.4. Definice herní úrovně
6. SKUPINY
 - 6.1. Postup načtení skupiny
7. PŘEHLED STATES
8. ZÁVĚR

Ukázka z příručky:

Prohlížeč poskytuje jednak náhled spuštěné hry, ale také ladící konzoli. Tato konzole zobrazí v případě problémů varování, či chyby, které brání spuštění hry, nebo ty, které nastanou při provedení specifické akce ve hře. Zároveň poskytuje informaci o tom, na kterém řádku se chyba nachází. Konzoli je možno otevřít tlačítkem pro vývojářské nástroje F12. Řádek s informací o chybě slouží jako odkaz a po kliknutí otevře kód na pozici s chybou.



Obrázek 14: Ukázka z příručky - Vývojářská konzole zobrazující chyby

4 TUTORIÁLY

Tutoriály jsou specifické návody k tomu, jak pomocí frameworku dosáhnout určitých cílů. Navazují na uživatelskou příručku. Jsou dostupné k nahlédnutí na přiloženém DVD. Samotný dokument je pojmenován Tutoriály.pdf, ve složce tutoriály lze poté dohledat zdrojové soubory použité v tutoriálech. U zdrojových souborů jsou také kompletní kódy pro případ, že by se čtenář v tutoriálu ztratil. Tutoriály obsahují celkem 5 velkých kapitol na 19 stranách.

Seznam tutoriálů

1. NAČTENÍ A UMÍSTĚNÍ ZDROJŮ A NÁSLEDNÁ ANIMACE

- 1.1. Načtení běžného obrázku
- 1.2. Načtení roztažitelného obrázku
- 1.3. Přidání animovaného obrázku a nastavení jeho animace
- 1.4. Škálování objektu
- 1.5. Změna kotevního bodu

2. DLAŽDICOVÁ Mapa

- 2.1. Nastavení počátku mapy
- 2.2. Nastavení smrti

3. OVLÁDÁNÍ

- 3.1. Ovládání pomocí vlastních kláves
 - 3.1.1. Pohyb do stran
 - 3.1.2. Skok
- 3.2. Ovládání pomocí šipek

4. POHYB NEPŘÁTEL A OBJEKTŮ

- 4.1. Import do hry
- 4.2. Nastavení pohybu
- 4.3. Kontakt s nepřítelem
- 4.4. Nepřítel bez otočení

5. ZVUK

- 5.1. Přidání hudby
- 5.2. Přidání zvuků

5 HRA VYTVOŘENÁ POMOCÍ FRAMEWORKU PHASER

Hra Robo je hrou, jejíž hlavní postavou je robot, který spadl z vesmírné lodi na nejbližší planetu. Jeho úkolem je dostat se pryč ze země, v kterých mu hrozí nebezpečí. Robot by měl pokud možno sbírat hvězdičky, které se z lodi ztratily spolu s ním.

5.1 Příprava

Před tvorbou samotného kódu je potřeba vytvořit objekty, které do hry budou vkládány. Mezi ně primárně patří grafika (ať už vektorová, nebo rastrová) a soubory pro herní ozvučení. Tyto objekty budou následně vloženy do hry pomocí různých syntaxí na zvolené pozice. Pro tvorbu grafiky a zvuků je doporučeno použít speciální software, jelikož pro tyto účely jsou nástroje, které jsou předinstalovány v operačním systému nedostatečné. Spuštění hry vyžaduje lokální server, a to z důvodu bezpečnosti. Samotný herní kód je možno psát v libovolném editoru, ale i přesto se doporučuje nepoužívat editor obsažený v systému. Poskytuje to výhodu ve formě zvýrazňování syntaxí, což vede ke zlepšení přehlednosti kódu. Použitou verzí frameworku byla verze 2.9.2 (CE 2.10).

5.2 Použitý software

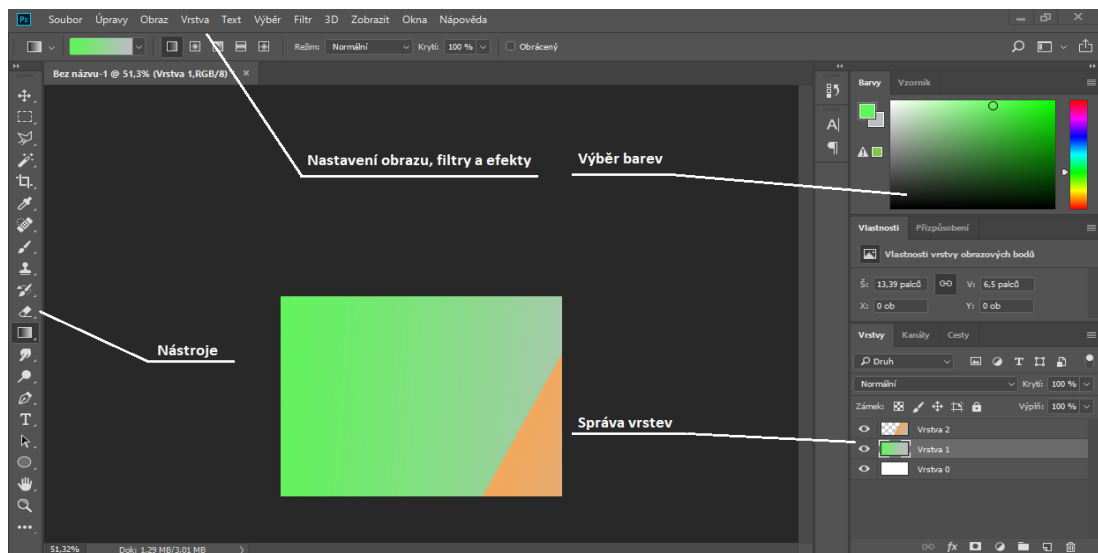
Pro programátorskou část byl vybrán pokročilý editor Brackets od společnosti Adobe. Kromě zvýrazňování kódu nabízí také živý náhled přes spuštění zabudovaného serveru. Není tedy nutno instalovat další software navíc. Program je dostupný ke stažení zdarma.



Obrázek 15: Prostředí programu Brackets

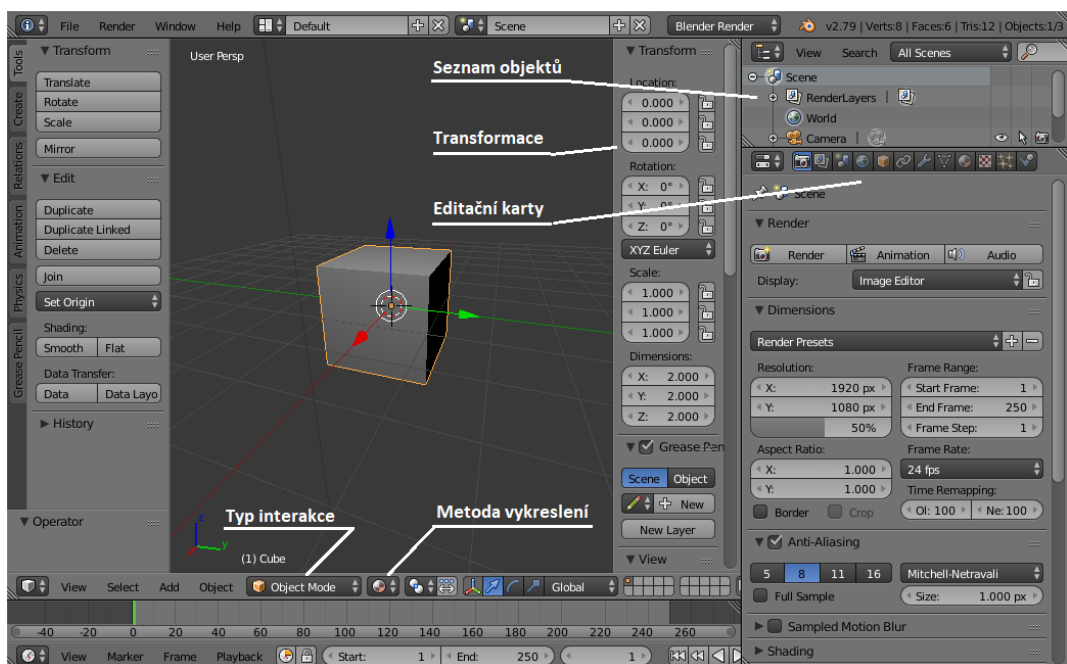
Pro tvorbu grafických zdrojů byl použit program Photoshop, který se používá pro editaci rastrové grafiky. Jedná se o nejrozšířenější editor vůbec, který není zdarma, ale nabízí velmi

komplexní sadu nástrojů a možností. Mezi ně patří například vykreslení různých křivek přes vektorově definovanou cestu, editace vrstev, různé filtry, makování, výběr podle barev a mnoho dalšího. Nadstandardní výhodou jsou také nástroje pro 3D, které nabízí například vygenerování normálových a dalších map pro 3D modely přímo z difúzních textur, nebo otáčení obrázků v 3D prostoru.



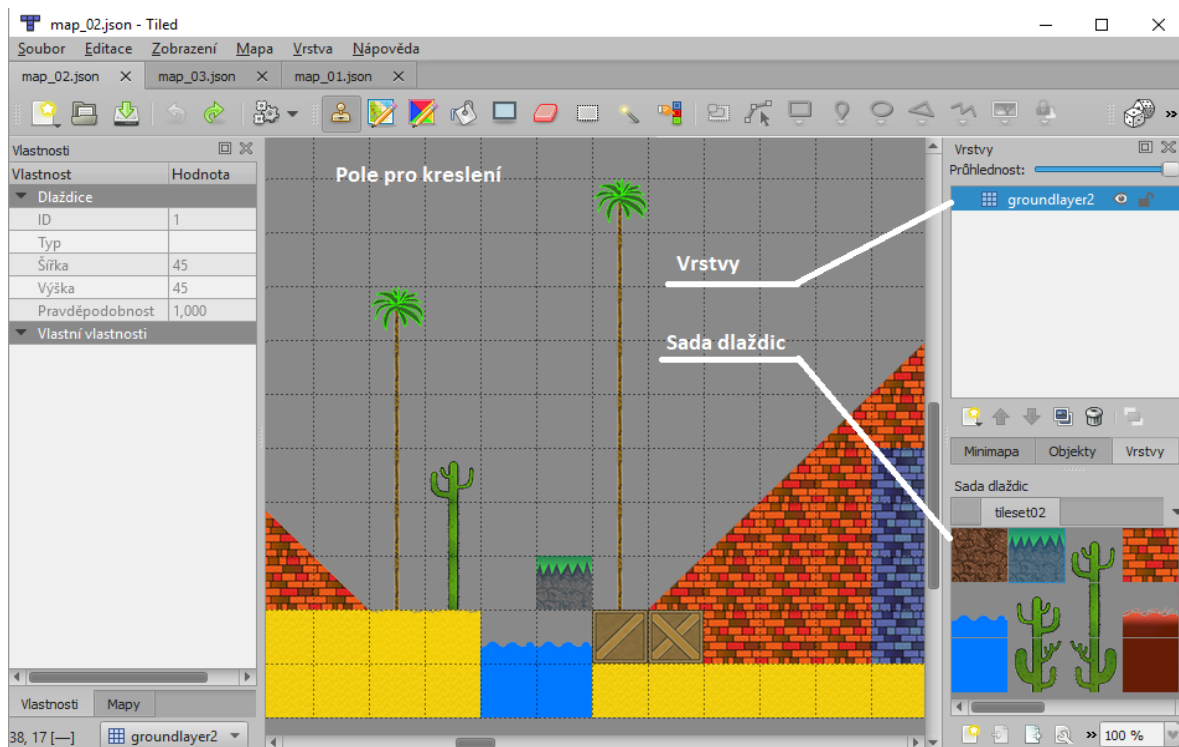
Obrázek 16: Prostředí programu Photoshop

V situacích, kde bylo vyžadováno nakreslení z různých pohledů byl použit program Blender, který se používá pro tvorbu 3D grafiky. Díky němu nebyla při kreslení nutná rozsáhlejší orientace v 3D prostoru.



Obrázek 17: Prostředí programu Blender

Posledním použitým programem je Tiled, kterým je možno vytvořit mapu levelu z vytvořeného setu dlaždic. Mapa je poté exportována do textového formátu json a interpretována Phaserem. Použití tohoto programu není nutné, ale dokáže výrazně usnadnit vkládání opakovaných vzorů.



Obrázek 18: Prostředí programu Tiled

5.3 Tvorba zdrojové grafiky

Jelikož Phaser nepodporuje vektorovou grafiku, byla použita grafika rastrová. Byl použit formát png, který podporuje alfa kanál pro průhlednost. Tento formát doporučují samotní tvůrci frameworku, což je zřejmé z jeho použití v tutoriálech.

5.3.1 Sada dlaždic

Sada dlaždic byla vytvořena jako mapový podklad. Pro roztažitelnost dlaždic v obou osách byl zvolen poměr stran 1:1.

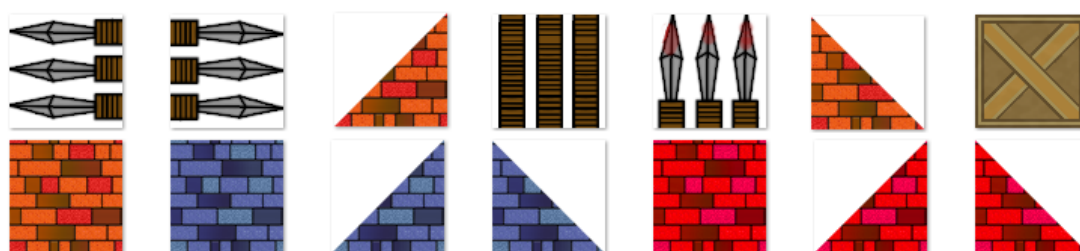
Bodce, bedna a cihly byly nakresleny pomocí nástroje pero. Byla nakreslena cesta, která byla následně rastrována. Výsledek sloužil jako ohraničení. Poté byla pod aktuální vrstvou vytvořena vrstva, která slouží k vyplnění barvou. U složitějších kreseb bylo použito více vrstev pro jednotlivé barevné výplně. Do horních vrstev byly následně dokresleny detaily.

U cihel musela být navíc brána v potaz roztažitelnost při použití, začátek tudíž musel navazovat na konec v obou osách.



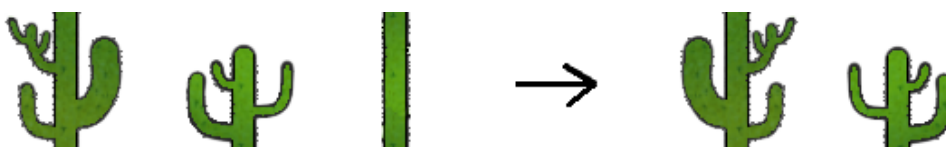
Obrázek 19: Komplexnost dlaždic

Z bodců a bedny vznikly pomocí jednoduchých transformačních operací další odvozené obrazy, jako druhá bedna, či otočené bodce. Další jednoduchá derivace bodců vznikla přikreslením červené barvy do horní vrstvy, která simulovala efekt krve.



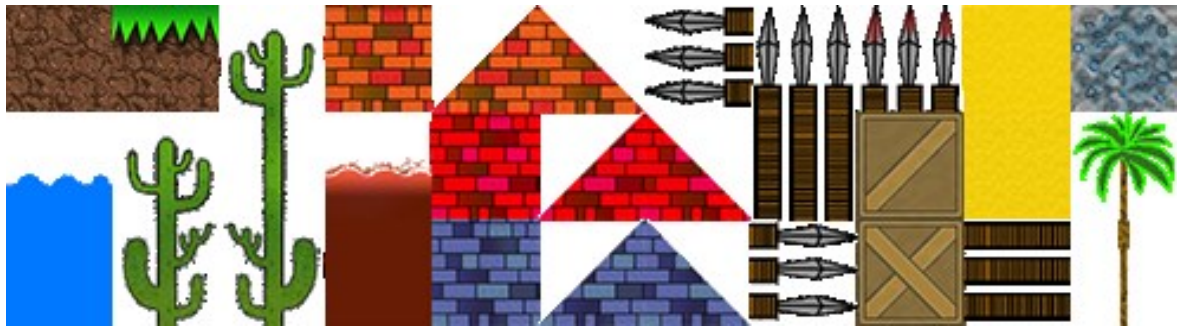
Obrázek 20: Odvozené dlaždice

Dalším zdrojem pro sadu dlaždic byly variace kaktusu. Nakresleny byly základní tři. Ty byly vytvořeny podobným postupem jako předchozí objekty, rozdíl byl jen v barvě výplně, kde byl místo jednoduché barvy použit vzorek, který evokoval ostny. Dvě ze tří základních částí kaktusu byly vytvořeny tak, aby se mohly opakovat v ose y. Po tvorbě základních tří obrázků vznikly další dva pomocí překlopení v ose x.



Obrázek 21: Roztažitelnost kaktusu v ose y a jeho variace

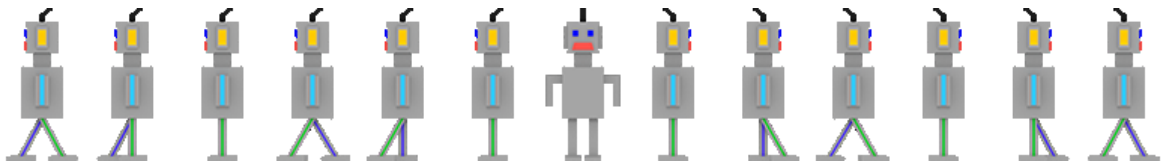
Stejným stylem jako předchozí grafické zdroje byly nakresleny i ostatní grafické zdroje pro hru. Na některé bylo navíc nutno použít rozmazávací efekt, či nějaký filtr. Následně bylo nutné jednotlivé zdroje spojit do stejného souboru s nulovými mezerami mezi sebou.



Obrázek 22: Kompletní sada dlaždic

5.3.2 Hráčův model

Z důvodu nutnosti pohledu z více směrů pro různé animace hráčovy postavy byl vytvořen 3D model. Jelikož měl model znázorňovat robota, stačilo pro jeho vymodelování spojit dohromady několik kusů primitivní geometrie. Následně se stanovila jedna skupina pro každou nohu pro animaci chůze. Poté stačilo vyrenderovat několik různých postojů. Aby model nebyl jednobarevný, několik polygonů bylo před renderováním změněno na jinou barvu.



Obrázek 23: Spritesheet hráče pro animaci pohybu

5.3.3 Nepřátelé

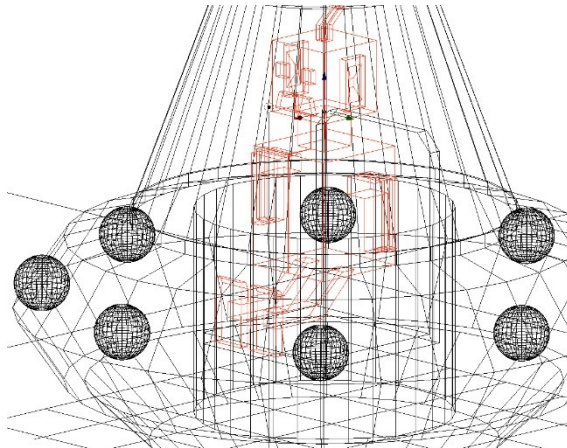
Aby hra nebyla příliš jednotvárná, bylo vytvořeno několik druhů nepřátel. Někteří nepřátelé byli vytvořeni jako spritekit, ostatní jako obyčejný sprite.

Prvním nepřítelem, u kterého stačilo nakreslit jeden snímek byl měsíc, stačilo tedy nakreslit kruh a poté z něj vymazat vyosenou výseč. Následně se do horních vrstev dokreslily detaily. Dalším byl žralok, u kterého stačilo nakreslit část, která bude vidět ve vlnách nad hladinou. Posledním nepřítelem, u kterého nebylo potřeba kreslit animaci byl fiktivní mimozemšťan. Z něj se poté vytvořila variace pomocí přebarvení.



Obrázek 24: Nepřátelé bez animačních snímků.

Tvorba animovaných nepřátel byla složitější. Kvůli jednotné tématice hry vycházel design některých nepřátel z hráčova modelu. Model nepřátelského robota letícího v UFO byl vytvořen ve 3D z důvodu animace objektu ve stylu rotace ve třetí ose. Tvorba se skládala ze dvou částí. První částí bylo nakreslit samotný létající talíř. Ve druhé části byl poté použit model hráče se změněnou barvou očí a jeho tvar byl změněn na sedící postavu. Ta byla nakonec přemístěna dovnitř talíře.



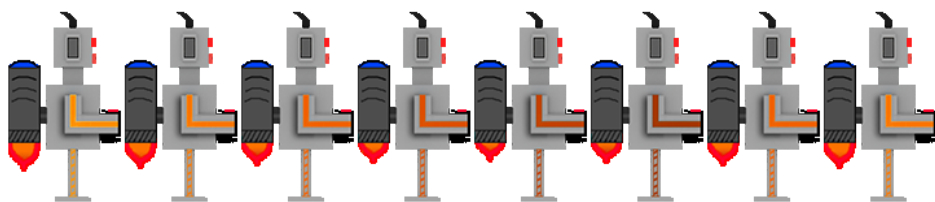
Obrázek 25: Tvorba létajícího talíře a zasazení nepřítele dovnitř.

Po správném umístění postavy byl použit nástroj rotace na létající talíř, postava se nepohybovala. V animaci lze vidět rozdílná pozice koulí na létajícím talíři.



Obrázek 26: Výsledný Spritesheet nepřítele v létajícím talíři.

Dalším nepřítelem vytvořeným s ohledem na hráčův model byl létající robot s tryskovým batohem. K vytvoření došlo jednoduchou editací hráčova modelu ve Photoshopu. Byla vytvořena spodní vrstva, ve které byl tryskový batoh a další vrstva pod ní, ve které byl samotný plamen. Pro jednotlivé snímky animace se pohybovalo s pozicí nejnižší vrstvy, která znázorňovala plamen a zároveň byla měněna barva jiné vrstvy, což evokovalo efekt záření.



Obrázek 27: Spritesheet nepřítele s tryskovým batohem

Posledním animovaným nepřítelem byl šedý červ, ze kterého následně vznikly další dvě barevné variace. Kresba snímku červa začínala tvorbou navazujících kruhů v horních vrstvách, pod které byla následně přidána výplň. Do posledního kruhu byl dokreslen obličej. Pro tvorbu animace se použilo posunu prostředního kruhu v ose y. Navazující kruhy byly také posunuty, ale o nižší hodnoty. Barevné variace červů byly vytvořeny pomocí nástroje, který selektivně vybral šedou barvu, která byla následně změněna.



Obrázek 28: Spritesheet červů a odvození jiných variant přes změnu barvy

5.3.4 Sběratelské předměty

Vybraný sběratelský předmět pro hru byla hvězda. V první části byly vytvořeny dvě vrstvy, jedna pro ohraničení a jedna pro výplň. Ve druhé části bylo použito překrytí vrstvy barvou. Tato barva byla pro každý snímek animace odlišná, díky čemuž animace vytvoří záři.



Obrázek 29: Animační snímky pro záři hvězdy

5.3.5 Ostatní

Dalšími grafickými objekty jsou obrázky, u kterých hráč začíná, nebo končí úroveň. Patří sem začáteční dveře a černá díra. Dveře byly vytvořeny standartním způsobem, jen u nich byl použit místo barvy pozadí šum, který ve hře znázorňuje hvězdy. Kvůli barvě hvězd bylo nutné zvolit monochromatický šum. Pro větší zářivost hvězd byl upraven kontrast. Stejným způsobem byla pak vytvořena černá díra, která hráče přenese do další úrovně. Na jejím okraji byl dokreslen vír. Z těchto objektů vznikly variace, které byly použity v dalších úrovních. Pro finální variaci černé díry byly použity hvězdy nakreslené pro sběratelský předmět.



Obrázek 30: Grafika pro přechod mezi úrovněmi a její variace

Jedním z animovaných objektů je také raketa. Jedná se o jednoduchou kresbu, u které se následně dokreslil postupný oheň z motorů pro jednotlivé snímky.



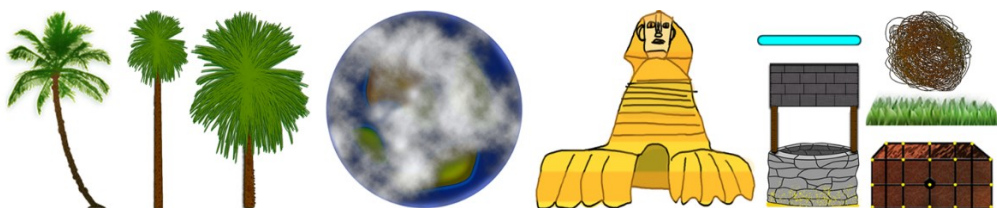
Obrázek 31: Animační sada pro let rakety

Zajímavým grafickým objektem, který je ve hře použit k interakci, je auto. Auto bylo nakresleno podle podkladu. Podkladem byl reálný automobil značky Mercedes, model G 6x6 [31]. Pro hru byl vytvořen ve kreslenějším stylu, kvůli jednoduchosti a jednoduchosti škály barev hry. Auto bylo pro hru vytvořeno i s animací. Animace obsahuje točení kol, které bude použito při jízdě. Pro animační snímky bylo označeno kolo, které bylo pomocí nástroje transformace postupně otáčeno o inkrementální hodnoty.



Obrázek 32: Animační sada pro auto

Posledním typem grafických objektů jsou objekty, použité pro level design, jako jsou palmy, zeměkoule, sfinga, studna, pokladnice, stepní běžec, tráva, či jednoduchá plošina. Tyto objekty byly nakresleny stejným způsobem jako objekty předchozí. Horní vrstva sloužila jako okraj, a spodní vrstva jako výplň. U objektu zeměkoule byla do horní vrstvy navíc dokreslena oblaka, která lze vytvořit přímo zabudovaným efektem ve Photoshopu, následně na ně byl použit deformační filtr, který je stočil a dodal jim větší realističnost. U okrajové vrstvy zeměkoule bylo navíc potřeba nastavit vnitřní záři.



Obrázek 33: Zdroje detailů pro level design

5.4 Tvorba zvuků










Pro každý level byla připravena samostatná hudba. Dále byly připraveny zvuky pro pohyb hráče a jeho interakci s některými objekty. Všechny herní zvuky byly staženy z internetu. Zvuky hry lze rozdělit na hudbu [32] [33] [34] a na zvukové efekty [35] [36] [37] [38] [39].

6 TVORBA HRY

Hra byla rozdělena do tří samostatných herních úrovní, které jsou propojeny pomocí cílové interakce v předchozí úrovni.

6.1 Hierarchie hry

Hra je rozložena do JavaScriptových úrovní, které Phaser označuje jako States. O indexování jednotlivých úrovní se stará HTML soubor. Vzhledem k tomu, že má hra 3 levely, je rozdělena do sedmi JavaScriptových souborů. Pro každý level je vyhrazený samostatný soubor, poté jsou zde soubory navíc pro herní nastavení, menu hry a informaci o dokončení hry. Posledním JavaScriptovým souborem je samotný framework Phaser, o který se opírá herní kód. Předvytvořená grafika je v samostatné složce assets.

 assets	Zdrojová grafika
 index.html	Indexový soubor
 endlevel.js	Stav při výhře hry
 game.js	Herní nastavení
 level1.js	První kolo
 level2.js	Druhé kolo
 level3.js	Třetí kolo
 menu.js	Menu hry
 phaser.js	Framework Phaser

Obrázek 34: Hierarchie herních souborů

6.2 Propojení souborů indexem

Propojení je zajištěno standartním způsobem v HTML souboru. Nejprve bylo třeba definovat, že se jedná o HTML5, poté název, následně seznam skriptů a na závěr herní Div.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Robo</title>
    <script type = "text/javascript" src = "phaser.js"></script>
    <script type = "text/javascript" src = "game.js"></script>
    <script type = "text/javascript" src = "level1.js"></script>
    <script type = "text/javascript" src = "level2.js"></script>
    <script type = "text/javascript" src = "level3.js"></script>
    <script type = "text/javascript" src = "endlevel.js"></script>
    <script type = "text/javascript" src = "menu.js"></script>
  </head>
  <body>
    <div id="gameDiv"></div>
  </body>
</html>
```

Obrázek 35: Obsah indexového souboru

6.3 Nastavení hry

Nastavení hry probíhá pomocí vložení nastavovací syntaxe do horní části kódu. Nastavení stačí provést pouze v jednom souboru a je tedy vloženo do souboru, který bude načítán jako první, čímž je game.js. Syntaxe vypadá následovně:

```
var game = new Phaser.Game(1280, 720, Phaser.AUTO, 'GameDiv');
```

Výše uvedeným kódem v podstatě definujeme proměnnou game, pro kterou nastavujeme rozlišení 1280x720. Značka AUTO označuje, že framework má volit automaticky mezi vykreslovací metodou WebGL a Canvas.

Po provedení tohoto kódu pokračuje program do souboru menu.js, ten obsahuje pouze nápis, že hru lze spustit stisknutím klávesy S a definuje akci, která přepne State do souboru level1.

```
var menuState = {
  create: function() {
    var startLabel = game.add.text(80, 80, 'press S to start',
    { font: '40px Arial', fill:
    '#ddfcff' });
    var skey = game.input.keyboard.addKey(Phaser.Keyboard.S);
    skey.onDown.addOnce(this.start, this);
  },

  start: function() {
    game.state.start('level1');
  },
};
game.state.add('menu', menuState);
game.state.start('menu');
```

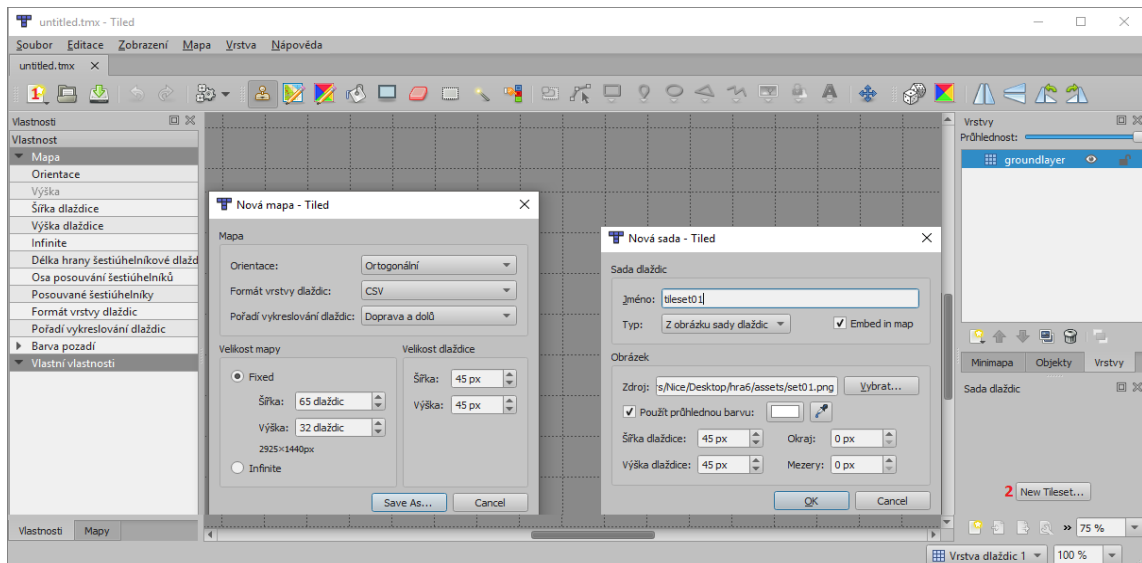
Obrázek 36: Předpis nastavovací úrovně

6.4 Statická grafika

Rozmístění statické grafiky proběhlo ve dvou fázích. V první fázi byla vytvořena dlaždicová mapa v programu Tiled s následným vložení do hry a v druhé fázi byla vložena ostatní statická grafika.

6.4.1 Mapy

Pro vytvoření mapy pro první kolo hry byl vybrán počet dlaždic 65x32. Při rozlišení 45x45 pixelů na jednu dlaždici to vyústilo ve velikost prvního levelu 2925x1440 pixelů. Při importování sady bylo potřeba obsáhnout sadu přímo do mapy, protože tento způsob Phaser doporučuje.



Obrázek 37: Tvorba mapy

Po vybrání grafického zdroje pro sadu dlaždic se zpřístupnily jednotlivé dlaždice. Nakreslením do mřížky vznikla mapa.



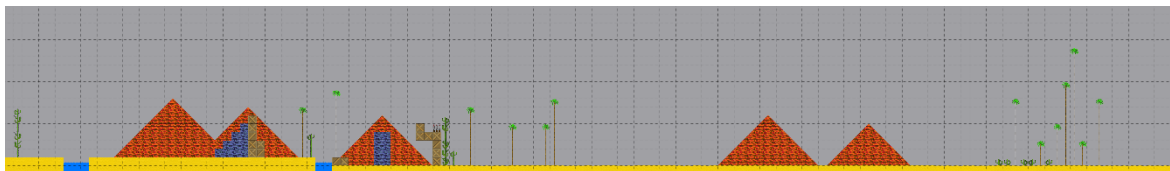
Obrázek 38: Hotová mapa pro první level bez detailů

Pro vložení mapy do hry bylo potřeba definovat proměnné map a layer; následně ve funkci preload načíst json soubor exportovaný z programu Tiled a vložit jej v části create.

```
var map;
var layer;
var levelState = {
  preload: function() {
    game.load.tilemap('map_01', 'assets/map_01.json', null, Phaser.Tilemap.TILED_JSON);
    game.load.image('tileset01', 'assets/set01.png');
  },
  create: function() {
    game.world.setBounds(0, 0, 2925, 1440);
    map = game.add.tilemap('map_01');
    map.addTilesetImage('tileset01');
    groundLayer = map.createLayer('groundlayer');
```

Obrázek 39: Načtení mapy ze souboru ve formátu json

Mapy pro další kola byly vytvořeny totožným způsobem, lišily se pouze rozměry mapy a jiný výběr dlaždic ze sady. Soubor se sadou byl ale stejný. Druhá mapa byla vytvořena v širokoúhlejší poměru stran o velikosti 6300x900 pixelů.



Obrázek 40: Hotová mapa druhého kola bez detailů

Třetí mapa byla vytvořena o velikosti 3150x2025 pixelů.



Obrázek 41: Hotová mapa třetího kola bez detailů

6.4.2 Ostatní statické objekty

Většina statických objektů sloužila jako doplnění detailů do level designu. Všechny objekty byly importovány a načteny stejným stylem. U většiny objektů bylo také definováno měřítko a různý kotevní bod, který zajišťoval odsazení objektu od souřadnic. Různé měřítko umožnilo vkládat variace objektu a použitím záporného měřítka bylo možno objekt převracet.

```
preload: function() {  
    game.load.image('treasure', 'assets/treasure.png');  
},  
create: function() {  
    treasure = this.game.add.sprite(375, 839, 'treasure');  
    treasure.scale.setTo(0.7);  
},
```

Obrázek 42: Způsob načtení dodatečného zdroje

6.5 Pozadí a efekty počasí

Pozadí lze v Phaseru definovat stejně, jako pozadí stránky v HTML a to v části funkce create. Lze použít barvy z nabízených vzorků, nebo definovat přímo kód barvy. Pro první kolo byl použit odstín tyrkysové barvy s kódem. "#ddfcff". Pro druhé kolo byl použit odstín oranžové s kódem "#e28e5a". U druhého kola byl navíc přidán efekt částicového systému, který měl nabudit dojem padajícího písku. U třetího kola nebyla použita barva pozadí, ale roztažitelná textura s hvězdami o velikosti 500x500 pixelů.

```
//level1
game.stage.backgroundColor = "#ddfcff";
//level2
game.stage.backgroundColor = "#e28e5a";
var emitter = game.add.emitter(game.world.centerX, 0, 300);
emitter.width = game.world.width;
emitter.angle = 3;
emitter.makeParticles('sand');
emitter.minParticleScale = 2;
emitter.maxParticleScale = 5;
emitter.setYSpeed(300, 500);
emitter.setXSpeed(-500, 500);
emitter.minRotation = 0;
emitter.maxRotation = 0;
emitter.start(false, 1600, 5, 0);
emitter.gravity = 10;
emitter.forEach(function(particle) {
particle.body.allowGravity = false;
}, this);
//level3
preload: function() {
    game.load.image('background', 'assets/bg3.png');},
background = game.add.tileSprite(0, 0, 3150, 2025, 'background');
```

Obrázek 43: Nastavení pozadí pro jednotlivé úrovně

6.6 Hráč

Hráč byl vložen jako sada obrázků neboli spritekit. Prostřední snímek určoval, jak má hráčova postava vypadat v klidovém stavu. Snímky okolo poté sloužily jako jednotlivé animční snímky pro pohyb. Ke zprovoznění hráče bylo potřeba definovat hráčovu proměnnou a proměnnou pro ovládací klávesy controls. Ovládání hráče bylo nastaveno na klávesy W, A, S, D z důvodu, aby zůstaly šipky neobsazené. V aktualizující části funkce byla vytvořena podfunkce, která zajišťovala, že při stisknutí požadované klávesy bude mít hráč určitou rychlost, při pohybu v ose x, při skoku v ose y. Dále musela být nastavena funkce, která zajišťovala, že pokud nebude stisknuta žádná klávesa pro pohyb, hráčova rychlost se okamžitě vrátí na nulu. Pro správné fungování animací bylo nutno definovat, jak má postava hráče vypadat v klidovém stavu, jak za pohybu směrem vpravo a jak směrem vlevo. Tyto animace byly

poté spouštěny podle stavu stisknutí kláves pro pohyb. U hráče bylo nutno také povolit fyziku a kolize s řadou objektů, mezi něž patří i sada dlaždic. Jinak by postava hráče propadla mapou dolů. Smrt hráče při kolizi s určitými dlaždicemi ze sady byla zajištěna funkcí `death`, která je volána, pokud se hráč dotkne definovaných dlaždic ze sady.

```

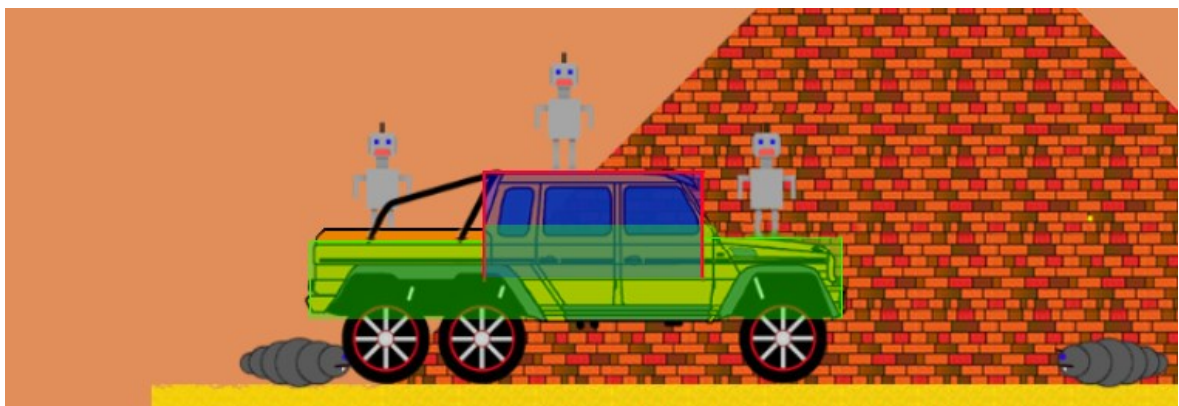
var player;
var controls = {};
var map;
preload: function() {
  game.load.spritesheet('player', 'assets/player.png', 60, 100, 13);
  var levelState = {
  create: function() {
    game.physics.arcade.gravity.y = 1600;
    map.setCollision([1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17,
    18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34]);
    groundlayer = map.createLayer('groundlayer');
    game.physics.arcade.enable('groundlayer');
    map.setTileIndexCallback([7,8,9,12], this.death, groundlayer);
    player = this.game.add.sprite(230, 1200, 'player');
    player.anchor.setTo(0, 0);
    player.scale.setTo(0.7);
    game.physics.arcade.enable(player);
    player.body.collideWorldBounds = true;
    player.animations.add('idle', [6], 0, false);
    player.animations.add('motion', [7,8,9,10,11,12], 10, true);
    player.animations.add('motionback', [5,4,3,2,1,0], 10, true);
    game.physics.arcade.enable(player);
    game.camera.follow(player);
    player.animations.play('idle');
    this.game.world.scale.setTo(1, 1);
  //ovládání
  controls = {
    right: this.input.keyboard.addKey(Phaser.Keyboard.D),
    left: this.input.keyboard.addKey(Phaser.Keyboard.A),
    up: this.input.keyboard.addKey(Phaser.Keyboard.W),
  };
  //smrt
  death: function(player, groundlayer) {
    game.state.restart();
  };
  update: function() {
    game.physics.arcade.collide(player, groundlayer);
    if(controls.right.isDown){
      player.animations.play('motion');
      player.scale.setTo(0.7);
      player.body.velocity.x = +250;
    } else if(controls.left.isDown) {
      player.animations.play('motionback');
      player.scale.setTo(0.7, 0.7);
      player.body.velocity.x = -250;
    } else {
      player.animations.play('idle')
    }
  }
  //skok
  if (controls.up.isDown){
    if (player.body.onFloor()){
      jumpsound.play('', 0, 0.05, false);
      player.body.velocity.y = -800;
      player.body.velocity.x = 0;}}}

```

Obrázek 44: Nastavení animací hráče a jeho kolizi

6.7 Auto

Auto bylo použito ve druhém kole hry. Idea byla využít ho k rychlejšímu posunu hráče. Protože model auta nepředstavuje typický obdélník, bylo potřeba upravit kolize. Původní kolize auta byla v ose y obsazena o několik bodů. Následně byl přidán neviditelný pomocný objekt, který přidal kolizi pro střechu auta, aby skrze ni hráč nemohl chodit. Jde pouze o kosmetické vylepšení, hráč bude při hraní stát hlavně na korbě auta. Z hlediska herní fyziky bylo potřeba udělat, aby auto mohlo vézt hráče, ale hráč nemohl pohybovat autem. Auto používá tři animační snímky, které se přehrávají dokola. V animačním snímku je rozdílná pouze rotace kol. Protože je mapa hry v ose y rovná, nebylo nutno využít fyzikální výpočet pro gravitaci.



Obrázek 45: Pokročilá kolize auta

6.8 Kamera

Aby bylo možno měnit pozici obrazovky, bylo potřeba přidat do hry kameru. Kamera byla nastavena tak, aby následovala pozici hráče. Ve výsledku kamera vždy zobrazovala obraz tak, aby byl hráč uprostřed, pokud zrovna není blízko hranic herní zóny.

```
create: function() {  
  game.camera.follow(player);  
}
```

Obrázek 46: Nastavení kamery

6.9 Sběratelské předměty

Jako sběratelský předmět ve hře slouží hvězda. V každé herní úrovni jsou rozmístěny tři. Hráč není povinen hvězdy sbírat, do dalších úrovní se dostane i bez nich. Jde pouze o bonus, který hráč získá za cenu větší obtížnosti. Pro hvězdu je použita animace z vytvořených pěti

grafických snímků. V jednom animačním cyklu animace přehraje postupně snímky vzestupně a poté sestupně. Při doteku hráče a hvězdy je hvězda považována za posbíranou, je tedy definována funkce, která zajišťuje, že při oné situaci hvězda zmizí.

```

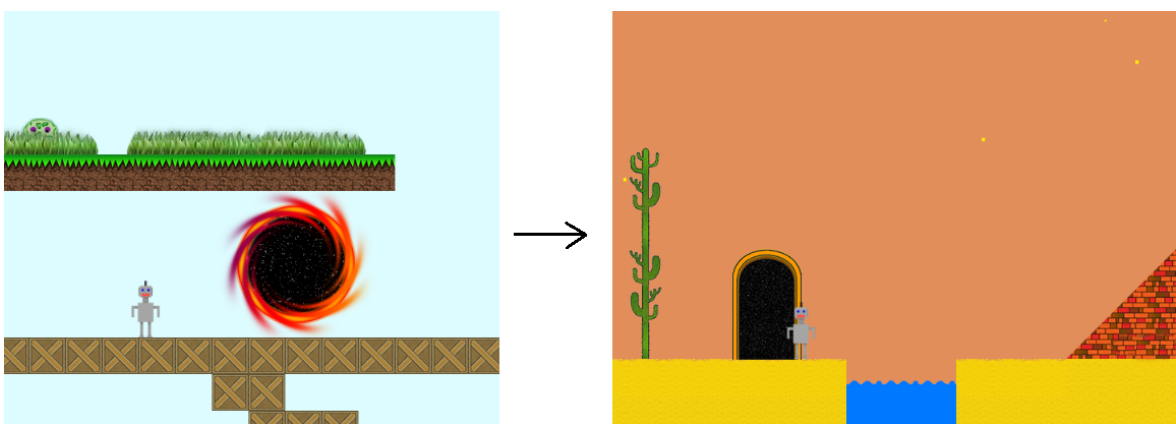
preload: function() {
  game.load.spritesheet('star', 'assets/star.png', 35, 35, 5);
  game.load.spritesheet('player', 'assets/player.png', 60, 100, 13);
},
create: function() {
  star = this.game.add.sprite(140, 710, 'star');
  star.animations.add('starshine', [0, 1, 2, 3, 4, 5, 4, 3, 2, 1, 0], 35, true);
  star.animations.play('starshine');
  star.anchor.setTo(0, 0);
  game.physics.arcade.enable(star);
  star.body.allowGravity = false;},
update: function() {
  game.physics.arcade.overlap(player, star, collisionHandler, null, this);
function collisionHandler(player, star) {
  star.destroy();
}
}

```

Obrázek 47: Funkce pro sbírání hvězd

6.10 Teleportéry a přenašeče

Z důvodu více herních úrovní bylo potřeba definovat nějaký způsob přepínání mezi nimi. Standartním postupem ve hrách bývá, že se hráč do druhého levelu dostane poté, co dokončí level první. I přes to ale musí být definována akce, která ho do druhého levelu přepne. Z toho důvodu byly vytvořeny víry s černou dírou, které se nacházejí na konci levelu, a pokud se k nim hráč přiblíží, vtáhnou ho do další úrovně. To je zajištěno pomocí funkce, která je spuštěna při doteku modelu hráče s vírem. Po přenesení do další úrovně se hráč objeví u obrázku dveří.



Obrázek 48: Přejít z jedné herní úrovně do druhé

Kromě teleportů jsou ve hře také použity objekty, které přenáší hráče uvnitř levelu. Tyto objekty mění pozici bez možnosti ovlivnit ji hráčem. Mezi tyto objekty patří například raketa, nebo jezdící plošiny.

6.11 Nepřátelé

U nepřátel nebyl vzhledem k jejich vlastnostem použit výpočet pro gravitaci. Prvního nepřítele představuje měsíc, jehož pohybová animace je podobná jako u ostatních nepřátel, ale je u něj zároveň současně prováděna rotace.

Druhým nepřítelem byl robot v létajícím talíři, jehož pohybovou animací je obyčejný pohyb tam a zpět. Tento jednoduchý pohyb byl poté použit i u ostatních nepřátel. U některých nepřátel byl změněn pohyb v osy x na pohyb v ose y, nebo byla použita kombinace obou.

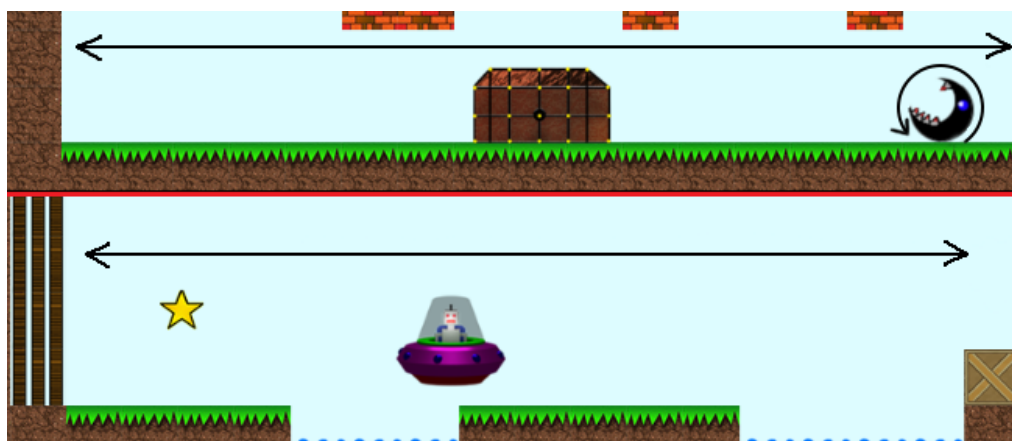
```
preload: function() {
    game.load.image('killingmoon', 'assets/killingmoon.png');
    game.load.spritesheet('enemy2', 'assets/enemy2.png',105,105,12);},
create: function() {
    killingmoon = this.game.add.sprite(100, 874, 'killingmoon');
    killingmoon.anchor.setTo(0.5, 0.5);
    game.physics.arcade.enable(killingmoon);
    killingmoon.body.allowGravity = false;
    killingmoonTween = game.add.tween(killingmoon).to({
        x: killingmoon.x + 650
    },3500,'Linear',true,0,100,true);

    enemy2 = this.game.add.sprite(1465, 1345, 'enemy2');
    enemy2.animations.add('enemy2rotation',[0,1,2,3,4,5,6,7,8,9,10,11],30,true);
    enemy2.anchor.setTo(0.5, 0.5);
    game.physics.arcade.enable(enemy2);
    enemy2.body.allowGravity = false;
    enemy2.scale.setTo(0.85);
    enemy2.animations.play('enemy2rotation');
    enemy2Tween = game.add.tween(enemy2).to({
        x: enemy2.x + 500,
    },5000,'Linear',true,0,100,true);
    enemy2.body.setSize(105,105,60,70);
update: function() {
    killingmoon.angle -= 10;
    if(checkOverlap(player,killingmoon)) {
        game.state.restart();
    }
    if(checkOverlap(player,enemy2)) {
        game.state.restart();
    }
    //funkce vracející hodnotu podle toho, zda se dva zdroje dotýkají
    function checkOverlap(sprite1,sprite2) {
        var bounds1 = sprite1.getBounds();
        var bounds2 = sprite2.getBounds();
        return Phaser.Rectangle.intersects(bounds1,bounds2);
    };
};
},
```

Obrázek 49: Nastavení pohybu nepřátel

Pro smrt hráče při střetnutí s nepřítelem bylo nutno definovat funkci, která restartuje level, pokud se některý z nepřátel dotkne hráče. Tato funkce závisela na další funkci, kterou bylo potřeba definovat. Ta vracela hodnotu podle toho, zda se objekty dotýkaly.

Pohyb byl ve frameworku vytvořen pomocí funkce Tween, což je zkratka z anglického between. Tato funkce zajišťuje interpolaci mezi počáteční a finálním bodem. K dispozici jsou různé parametry, které umožňují mimo jiné nastavit, zda je vyžadován i pohyb zpět na počáteční bod, či opakování cyklu. V této hře se nepřátelé opakovaně pohybují tam a zpět.



Obrázek 50: Typ pohybu nepřátel

6.12 Zvuky

Ve hře bylo použito celkem 8 zvukových souborů, z toho tři byly použity jako hudba pro jednotlivé herní úrovně. Dva zvuky byly použity pro pohyb hráče a jeden pro jeho smrt. Další zvuk byl použit při sebrání hvězdičky. Poslední zvuk byl použit pro pohyb auta.

Hudba pro jednotlivé úrovně byla nastavena pomocí načtení zdroje a následného přehrání. Při každém restartu úrovně, nebo přechodu do jiné, bylo důležité hudbu zastavit. Hudba byla zastavena pokaždé, když hráč umřel. Při té příležitosti byl zároveň přehrán zvuk smrti.

```
preload: function() {
    game.load.audio('sound', 'assets/sound.mp3');
    game.load.audio('deathsound', 'assets/deathsound.wav', 105, 105, 12);
},
create: function() {
    sound1 = game.add.audio('sound1');
    sound1.play('', 0, 1, true);
},
death: function(player, groundlayer) {
    walksound.stop();
    sound1.stop();
    deathsound.play('', 0, 1, false);
    game.state.restart();
},
```

Obrázek 51: Způsob načtení a zastavení hudby

Nastavení zvuků hráče bylo složitější, protože bylo potřeba přehrát zvuk při pohybu, či při skoku. Zároveň nesměl být zvuk pohybu přehráván ve chvílích, kdy se hráč nepohybuje po zemi. Jelikož se pohyb hráče realizuje přes funkci update, bylo také potřeba vytvořit novou funkci, která zamezila přehrávání zvuku pohybu od začátku pokaždé, když se funkce obnovila. Výsledkem byl opakující se zvuk při pohybu, ovšem v plné délce.

```
preload: function() {
    game.load.audio('jumpsound', 'assets/jumpsound.mp3');
    game.load.audio('walksound', 'assets/walksound.mp3');
    game.load.audio('deathsound', 'assets/deathsound.wav', 105, 105, 12);},
create: function() {
    walksound = game.add.audio('walksound');
    walksound.allowmultiple = false;
    jumpsound = game.add.audio('jumpsound');
},
update: function() {
    if(controls.right.isDown && player.body.onFloor()) {
        if(!walksound.isPlaying) {
            walksound.play();
        }
    } else if(controls.left.isDown && player.body.onFloor()) {
        if(!walksound.isPlaying) {
            walksound.play();
        }
    } else {
        walksound.stop();
    }
    //zastavit hudbu při restartu
    if(checkOverlap(player, killingmoon)) {
        game.state.restart();
        sound1.stop();
        deathsound.play('', 0, 1, false);
        walksound.stop();
    }
}
```

Obrázek 52: Nastavení zvuků hráče a střet s nepřítelem

6.13 Počítadlo skóre

Počítadlo skóre bylo vytvořeno pomocí proměnné, ve které byl neustále aktualizován momentální počet sebraných hvězd. Na začátku hry měla proměnná přidělenou hodnotu 0. Při střetu hráče s hvězdou došlo k navýšení skóre o 1 bod. Vypsání skóre pro hráče hry bylo zajištěno pomocí textu, který vracel aktuální hodnotu proměnné pro skóre. Tento text nebyl umístěn na specifickou souřadnici ve hře, namísto toho byl ukotven pomocí pozice kamery do spodního levého rohu.

```
var score = 0;
var ScoreText
create:function() {
  scoreText = game.add.text(15, 15, 'Score = ' + score, { fontSize: '15px', fill: 'black' });
  scoreText.fixedToCamera = true; //ukotvení textu ke kameře
  scoreText.cameraOffset.setTo(50,660); //pozice textu na obrazovce
}
update:function() {
  function collisionHandler(player, star) { //při střetu hráče s hvězdou:
    star.destroy(); //hvězda zmizí
    score += 1; //zvýší skóre o bod
    scoreText.text = 'Score = ' + score; //vypsaný text zvýší hodnotu o bod
  }
  if(checkOverlap(player,enemy)) { //při úmrtí
    score = score1end; //načte hodnotu z dokončení předchozího levelu
    game.state.restart(); //restartuje level
  }
}
```

Obrázek 53: Výpočet skóre v druhé úrovni

Při dokončení úrovně byla hodnota skóre uložena do další proměnné, aby se v případě smrti hráče v dalším kole bylo možno vrátit k hodnotě, s kterou se hráč objevil na začátku dané úrovně.

Při úmrtí byla hodnota skóre resetována na hodnotu, která byla vypočítána při začátku kola. K tomu napomohly pomocné proměnné, které před teleportováním hráče do dalšího kola uložily finální hodnotu skóre z aktuálního kola. Tato hodnota byla následně volána, pokud hráč v dalším kole umřel.

```
var score2end;
function collisionHandler4(player, stars) { //při střetu s cílem
  this.state.start('level3'); //změnit level
  score2end = score; //zkopírovat hodnotu skóre do pomocné proměnné
}
```

Obrázek 54: Uložení hodnoty skóre na konci levelu

ZÁVĚR

V teoretické části tato bakalářská práce seznámila čtenáře se značkovacím jazykem HTML, který se používá pro tvorbu webových dokumentů. Objasnila, jaké novinky přinesla aktuální verze 5 a zároveň osvětlila i další jazyky, které se v souvislosti s HTML používají, jako je JavaScript, či CSS. Následně vysvětlila, co je to framework a k čemu se používá. V další kapitole rozebrala samotný framework Phaser, což bylo hlavním účelem této práce. Bylo ukázáno, jaké možnosti a výhody použití framework Phaser nabízí.

V praktické části bylo vytvořeno několik doprovodných dokumentů, které mají různé účely. První dokument, který se k této práci váže, je uživatelská příručka, která seznamuje čtenáře, kteří mají o framework Phaser zájem, se základními kroky. Příručka byla seřazena do kapitol tak, aby se v ní vyznal i méně zkušený čtenář.

Druhým příloženým dokumentem je série výukových tutoriálů, která poskytuje krok za krokem návod k tomu, jak pomocí frameworku dosáhnout specifických cílů. Tutoriály se nezačínají tvorbou konkrétní hry, pouze řeší typové problémy, po jejichž zvládnutí bude čtenář schopen vytvořit vlastní hru.

Poslední přílohou bakalářské práce je hra zvaná Robo, která demonstruje možnosti frameworku Phaser díky tomu, že byla vytvořena přímo v něm. Hra se zaměřuje na příběh robota, který v jednotlivých úrovních postupně cestuje po různých planetách. O vytvoření této hry je v praktické části obsáhlá kapitola, která popisuje samotnou hru a jednotlivé body, které při vytváření hry probíhaly. Kvůli náročnosti tvorby zdrojových souborů pro hru bylo také použito několik programů, které usnadnily jejich tvorbu.

Framework je již v aktuální verzi pro tvorbu her velice nápomocný. I přes to se ale vývojáři rozhodli vyvinout kompletně přepracovanou novou verzi, díky které bude odstraněna většina nedostatků. Verze 3 bude také efektivněji využívat výkon počítače, což povede k nižší náročnosti.

SEZNAM POUŽITÉ LITERATURY

- [1] About HTML5 [online]. [cit. 2018-03-28]. Dostupné z: <https://www.techradar.com/news/internet/web/html5-what-is-it-1047393>
- [2] HTML5 logo [online]. [cit. 2018-03-28]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/6/61/HTML5_logo_and_wordmark.svg
- [3] History of HTML [online]. [cit. 2018-03-28]. Dostupné z: <http://www.yourhtmlsource.com/starthere/historyofhtml.html>
- [4] HTML5 features [online]. [cit. 2018-03-28]. Dostupné z: <https://code.tutsplus.com/tutorials/28-html5-features-tips-and-techniques-you-must-know--net-13520>
- [5] About CSS [online]. [cit. 2018-03-28]. Dostupné z: <http://www.css3-tutorial.net/introduction/what-is-css/>
- [6] CSS3 logo [online]. [cit. 2018-03-28]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/d/d5/CSS3_logo_and_wordmark.svg
- [7] New features in CSS3 [online]. [cit. 2018-03-28]. Dostupné z: <https://courses.telegraph.co.uk/article-details/181/what-is-css3/>
- [8] Border by image [online]. [cit. 2018-03-28]. Dostupné z: https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_border-image
- [9] About JavaScript [online]. [cit. 2018-03-28]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- [10] Frameworks [online]. [cit. 2018-03-28]. Dostupné z: <https://django-stars.com/blog/what-is-a-web-framework/>
- [11] Frameworky [online]. [cit. 2018-03-28]. Dostupné z: <https://cs.wikipedia.org/wiki/Framework>
- [12] Phaser Github page [online]. [cit. 2018-03-28]. Dostupné z: <https://github.com/photonstorm/phaser>
- [13] Phaser homepage [online]. [cit. 2018-03-28]. Dostupné z: <https://phaser.io/>
- [14] Phaser changelog [online]. [cit. 2018-03-28]. Dostupné z: <https://github.com/photonstorm/phaser-ce/blob/master/CHANGELOG.md>
- [15] Phaser version 3 developement [online]. [cit. 2018-03-28]. Dostupné z: <https://phaser.io/phaser3>

- [16] Licence of Phaser [online]. [cit. 2018-03-28]. Dostupné z: <https://phaser.io/download/license>
- [17] Loading spritesheets [online]. [cit. 2018-03-28]. Dostupné z: <https://phaser.io/examples/v2/loader/load-spritesheet>
- [18] Sprajty [online]. [cit. 2018-03-28]. Dostupné z: <http://sketchapp.me/rabota-s-graficheskim-redaktorom-sketch-chast-1-sprajty/>
- [19] Gravity defied [online]. [cit. 2018-03-28]. Dostupné z: https://appda.ru/uploads/posts/2015-10/1444756965_skachat-gravity-defied-classic-dlya-android.png
- [20] Phaser emitter example [online]. [cit. 2018-03-28]. Dostupné z: <https://phaser.io/examples/v2/particles/diamond-burst>
- [21] Tileable texture [online]. [cit. 2018-03-28]. Dostupné z: <https://www.filterforge.com/filters/11733.jpg>
- [22] Plugins for Phaser [online]. [cit. 2018-03-28]. Dostupné z: <https://phaser.io/shop/plugins>
- [23] Alternatives [online]. [cit. 2018-03-28]. Dostupné z: <https://github.com/collections/javascript-game-engines>
- [24] PixiJS [online]. [cit. 2018-03-28]. Dostupné z: <http://www.pixijs.com/>
- [25] Melon [online]. [cit. 2018-03-28]. Dostupné z: <http://melonjs.org/>
- [26] Cocos products [online]. [cit. 2018-03-28]. Dostupné z: <http://www.cocos2d-x.org/products>
- [27] Construct [online]. [cit. 2018-03-28]. Dostupné z: <https://www.scirra.com/construct2>
- [28] Gamemaker [online]. [cit. 2018-03-28]. Dostupné z: <https://www.yoyogames.com/gamemaker>
- [29] ImpactJS [online]. [cit. 2018-03-28]. Dostupné z: <http://impactjs.com/>
- [30] CreateJS [online]. [cit. 2018-03-28]. Dostupné z: <https://createjs.com/>
- [31] G class 6x6 [online]. [cit. 2018-03-28]. Dostupné z: <https://www.mercedes-benz.com/en/mercedes-benz/innovation/the-automotive-declaration-of-independence/>
- [32] Tentacle Wedding [online]. [cit. 2018-03-28]. Dostupné z: <https://free-sound.org/people/Romariogrande/sounds/396231/>

- [33] Tristan Lohengrin [online]. [cit. 2018-03-28]. Dostupné z: https://freesound.org/people/Tristan_Lohengrin/sounds/343835/
- [34] Rivalry – 8 bit music loop [online]. [cit. 2018-03-28]. Dostupné z: <https://freesound.org/people/michorvath/sounds/412344/>
- [35] Cartoon jump [online]. [cit. 2018-03-28]. Dostupné z: <https://freesound.org/people/Robinhood76/sounds/95557/>
- [36] Casual Death Loose [online]. [cit. 2018-03-28]. Dostupné z: <https://freesound.org/people/GameAudio/sounds/220203/>
- [37] Car engine [online]. [cit. 2018-03-28]. Dostupné z: <https://freesound.org/people/InspectorJ/sounds/345557/>
- [38] Sfx tech effect [online]. [cit. 2018-03-28]. Dostupné z: <https://freesound.org/people/Oddworld/sounds/125106/>
- [39] Walking Sound [online]. [cit. 2018-03-28]. Dostupné z: <https://freesound.org/people/Mativve/sounds/414331/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML HyperText Markup Language.

CSS Cascading Style Sheets.

JS JavaScript.

SEZNAM OBRÁZKŮ

<i>Obrázek 1: logo HTML5 [2]</i>	10
<i>Obrázek 2: Vstupní typ datum</i>	13
<i>Obrázek 3: Logo CSS3 [6]</i>	14
<i>Obrázek 4: Princip ohraničení obrázkem [8]</i>	15
<i>Obrázek 5: Logo frameworku Phaser [13]</i>	18
<i>Obrázek 6: Postup vývoje Phaseru [15]</i>	20
<i>Obrázek 7: definice hry</i>	21
<i>Obrázek 8: Načtení zdroje</i>	21
<i>Obrázek 9: Příklad Spritesheetu [18]</i>	22
<i>Obrázek 10: Ukázka účinku fyzikálních výpočtů na zobrazení objektů [19]</i>	22
<i>Obrázek 11: Přidání částicového systému</i>	23
<i>Obrázek 12: Roztažitelnost dlaždicové mapy [21]</i>	24
<i>Obrázek 13: Porovnání alternativ</i>	28
<i>Obrázek 14: Ukázka z příručky - Vývojářská konzole zobrazující chyby</i>	31
<i>Obrázek 15: Prostředí programu Brackets</i>	33
<i>Obrázek 16: Prostředí programu Photoshop</i>	34
<i>Obrázek 17: Prostředí programu Blender</i>	34
<i>Obrázek 18: Prostředí programu Tiled</i>	35
<i>Obrázek 19: Komplexnost dlaždic</i>	36
<i>Obrázek 20: Odvozené dlaždice</i>	36
<i>Obrázek 21: Roztažitelnost kaktusu v ose y a jeho variace</i>	36
<i>Obrázek 22: Kompletní sada dlaždic</i>	37
<i>Obrázek 23: Spritesheet hráče pro animaci pohybu</i>	37
<i>Obrázek 24: Nepřítelé bez animačních snímků.</i>	37
<i>Obrázek 25: Tvorba létajícího talíře a zasazení nepřítele dovnitř.</i>	38
<i>Obrázek 26: Výsledný Spritesheet nepřítele v létajícím talíři.</i>	38
<i>Obrázek 27: Spritesheet nepřítele s tryskovým batohem</i>	38
<i>Obrázek 28: Spritesheet červů a odvození jiných variant přes změnu barvy</i>	39
<i>Obrázek 29: Animační snímky pro záři hvězdy</i>	39
<i>Obrázek 30: Grafika pro přechod mezi úrovněmi a její variace</i>	39
<i>Obrázek 31: Animační sada pro let rakety</i>	40
<i>Obrázek 32: Animační sada pro auto</i>	40

<i>Obrázek 33: Zdroje detailů pro level design</i>	40
<i>Obrázek 34: Hierarchie herních souborů</i>	41
<i>Obrázek 35: Obsah indexového souboru</i>	41
<i>Obrázek 36: Předpis nastavovací úrovně</i>	42
<i>Obrázek 37: Tvorba mapy</i>	43
<i>Obrázek 38: Hotová mapa pro první level bez detailů</i>	43
<i>Obrázek 39: Načtení mapy ze souboru ve formátu json</i>	43
<i>Obrázek 40: Hotová mapa druhého kola bez detailů</i>	44
<i>Obrázek 41: Hotová mapa třetího kola bez detailů</i>	44
<i>Obrázek 42: Způsob načtení dodatečného zdroje</i>	44
<i>Obrázek 43: Nastavení pozadí pro jednotlivé úrovně</i>	45
<i>Obrázek 44: Nastavení animací hráče a jeho kolizí</i>	46
<i>Obrázek 45: Pokročilá kolize auta</i>	47
<i>Obrázek 46: Nastavení kamery</i>	47
<i>Obrázek 47: Funkce pro sbírání hvězd</i>	48
<i>Obrázek 48: Přejechod z jedné herní úrovně do druhé</i>	48
<i>Obrázek 49: Nastavení pohybu nepřátel</i>	49
<i>Obrázek 50: Typ pohybu nepřátel</i>	50
<i>Obrázek 51: Způsob načtení a zastavení hudby</i>	50
<i>Obrázek 52: Nastavení zvuků hráče a střet s nepřítelem</i>	51
<i>Obrázek 53: Výpočet skóre v druhé úrovni</i>	52
<i>Obrázek 54: Uložení hodnoty skóre na konci levelu</i>	52

SEZNAM PŘÍLOH

Příloha P I: Úvodní úroveň hry

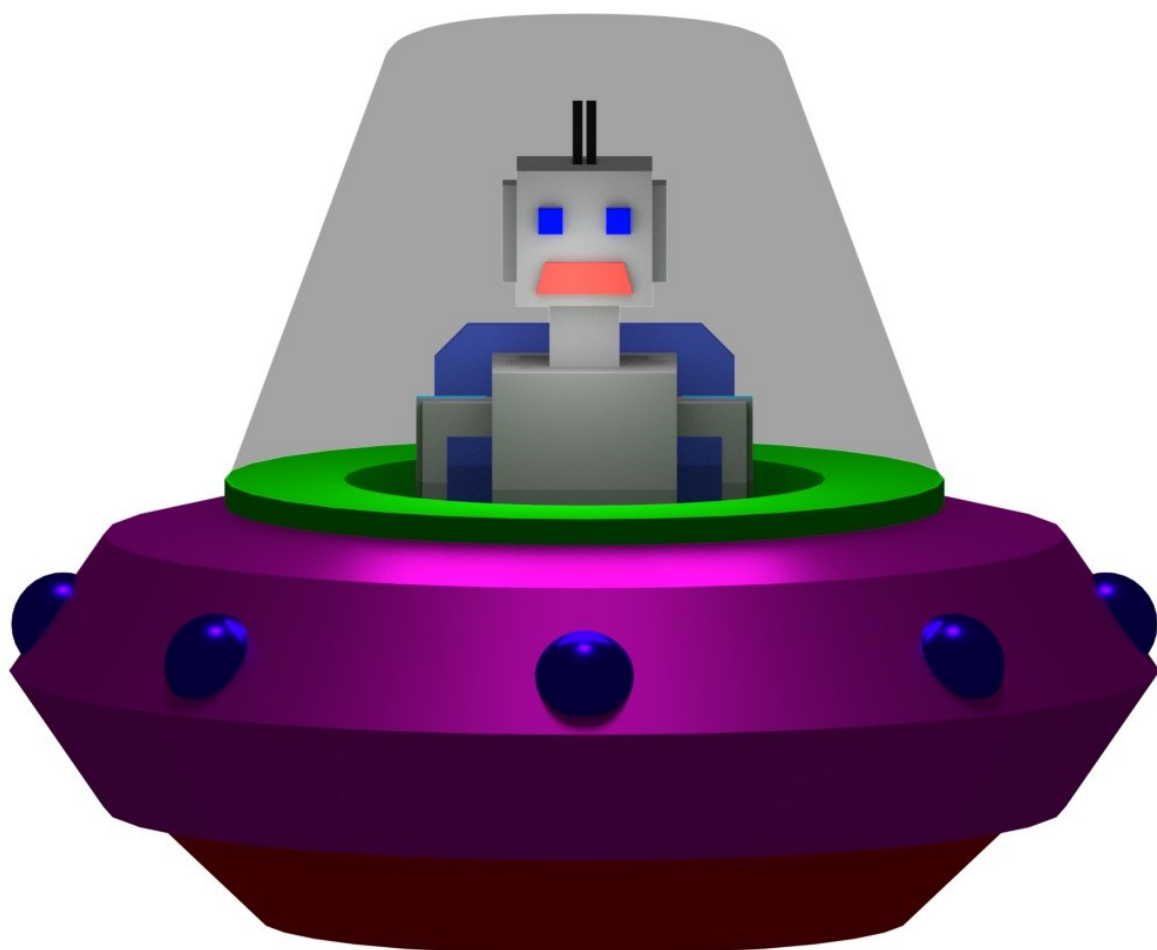
Příloha P II: Detail jednoho z nepřátel

Příloha P III: Obsah přiloženého CD

PŘÍLOHA P I: ÚVODNÍ ÚROVEŇ HRY



PŘÍLOHA P II: DETAIL JEDNOHO Z NEPŘÁTEL



PŘÍLOHA P III: OBSAH PŘILOŽENÉHO CD

