

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA
MLADÁ BOLESLAV

ROČNÍKOVÁ PRÁCE



Tomáš Novák

Mladá Boleslav 2025

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA MLADÁ BOLESLAV

Galaxy Invaders!

Autor: Tomáš Novák

Studijní obor: 18-20-M/01 Informační technologie

Vedoucí práce: Jan Till

Mladá Boleslav 2025

Obsah

Obsah.....	3
1 Úvod	8
2 Herní žánr a inspirace.....	9
2.1 Co je Galaxy Invaders	9
2.2 Herní průběh	10
2.3 Ovládání hráče a střelba.....	10
2.4 Pohyb a chování nepřátel.....	11
2.5 Herní rozhraní a informační prvky	11
2.6 Systém vln a obtížnost	11
2.7 Ukončení hry	12
3 Použité technologie.....	12
3.1 HTML5	13
3.2 CSS3	13
3.3 JavaScript.....	14

4	Vývoj	15
4.1	Main.js.....	15
4.1.1	Definice hráče a jeho vykreslování	15
4.1.2	Ovládání hráče pomocí klávesnice	16
4.1.3	Aktualizace pohybu hráče (update)	16
4.1.4	Správa životů hráče.....	17
4.1.5	Vykreslování skóre, životů a vlny	18
4.1.6	Úvodní obrazovka a start hry.....	19
4.1.7	Obrazovka Game Over a ukládání skóre	20
4.1.8	Herní cyklus (gameLoop)	21
4.2	Sound.js	22
4.3	enemies.js.....	23
4.3.1	Pohyb nepřátel a změna směru.....	24
4.3.2	Detekce kolizí projektilů s nepřáteli	25
4.3.3	Kontrola přistání nepřátel (prohra)	26

4.3.4	Správa přechodu na další vlnu	26
4.3.5	Vykreslování nepřátel	27
4.4	enemyBullets.js	27
4.4.1	Střelba nepřátel	28
4.4.2	Detekce zásahu hráče	29
4.4.3	Vykreslování nepřátelských střel	29
4.5	bullets.js	30
4.5.1	Střelba hráče (střídání kanónů)	30
4.5.2	Aktualizace pohybu projektilů	31
4.5.3	Vykreslování střel	31
4.6	Problematika při vývoji	32
4.6.1	Použití delta time	32
4.6.2	„Start screen“ a „Game over screen“	32
4.7	Testování	33
4.7.1	Drobné úpravy	33

5	Možné budoucí rozšíření.....	33
5.1	Přidání herního příběhu	33
5.2	Různé herní módy	34
5.3	Achievementy.....	34
5.4	Sběr herní měny a obchod	34
6	Závěr.....	35
7	Přílohy.....	36
7.1	Zdroje	36

Prohlášení

Prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti zpřístupňování této ročníkové práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Mladé Boleslavi dne podpis:



1 Úvod

Cílem této práce je vytvoření jednoduché 2D střílečky inspirované klasickou arkádovou hrou zvanou Space Invaders¹. Výsledná webová aplikace je vytvořena pomocí technologií HTML², CSS³ a JavaScript⁴. Projekt je vytvořen bez využití placených komerčních knihoven či jiných nástrojů s omezenou licencí, a respektuje tak zásady svobodného softwaru a autorského práva.

Zvolená hra používá základní principy vývoje herních webových aplikací a současně umožňuje aplikaci dovedností z předmětů, které vyučuje Střední průmyslová škola Mladá Boleslav, jako je práce s funkcemi, správa a aktualizace herního cyklu, logika spojená s kolizemi a vykreslování grafických prvků. Výsledný produkt je rozložen na modulární strukturu kódu a jednoznačné oddělení jednotlivých částí kódu, což přispívá k jeho přehlednosti a budoucí možné rozšiřitelnosti.

Součástí webové aplikace je také základní uživatelské rozhraní zobrazující stav hry, včetně aktuálního skóre, počtu životů a čísla vlny nepřátel. Aplikován je také mechanismus dynamického zvyšování obtížnosti prostřednictvím narůstající agresivity nepřátel. Herní jsou inspirovány principy původní hry, přičemž byly rozšířeny o nově navržené prvky, jako je přidávání životů, dynamicky zvyšující se skóre anebo střídavá střelba z více pozic hráčovy lodi, konkrétně z dvou kanónů, kterou vesmírná loď disponuje.

Tato dokumentace je rozdělena do několika částí, které jsou odděleny nadpisy. První část se věnuje teoretickému základu, prvotnímu nápadu a vysvětlení mechanismů, na kterých je hra postavena. Následuje popis vývojového prostředí a použitých technologií. Další části se

¹ WIKIPEDIA. *Space Invaders!*. Online. 2004, 21.4.2025. Dostupné z: https://en.wikipedia.org/wiki/Space_Invaders. [cit. 2025-04-28].

² WIKIPEDIA. *HTML*. Online. 2006, 19.4.2025. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Markup_Language. [cit. 2025-04-28].

³ WIKIPEDIA. *CSS*. Online. 2006, 11.4.2025. Dostupné z: https://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styl. [cit. 2025-04-28].

⁴ WIKIPEDIA. *JavaScript*. Online. 2006, 20.2.2025. Dostupné z: <https://cs.wikipedia.org/wiki/JavaScript>. [cit. 2025-04-28].



zaměřují na jednotlivé fáze vývoje a popis konkrétního postupu a realizace. Závěrečná část obsahuje zhodnocení práce, zjištěné poznatky a možnosti budoucího rozšíření.

2 Herní žánr a inspirace

2.1 Co je Galaxy Invaders

Galaxy invaders je dvourozměrná střílečí hra inspirovaná legendární arkádovou hrou, která vyšla na přelomu osmdesátých a devadesátých let minulého století v Japonsku zvanou Space Invaders!⁵.

⁵ WIKIPEDIA. *Space Invaders!*. Online. 2004, 21.4.2025. Dostupné z: https://en.wikipedia.org/wiki/Space_Invaders. [cit. 2025-04-28].



2.2 Herní průběh

Základním principem hry Galaxy Invaders je střet mezi hráčem ovládanou vesmírnou lodí, která se snaží ochránit svojí domovskou planetu, a skupinami nepřátelských mimozemských jednotek, které se objevují ve vlnách. Herní pole je omezeno do pevně daného dvourozměrného prostoru, který se ale přizpůsobuje velikosti monitoru, na kterém je hra spuštěna. Hráč má možnost pohybu pouze ve vodorovném směru na spodní hraně obrazovky, zatímco nepřátelé se pohybují synchronizovaně v horní části a postupují směrem dolů, pokaždé když narazí na pravou či levou hranu obrazovky. Cílem hráče je zničit všechny nepřátele v každé vlně a získat co nejvyšší počet bodů, a zároveň se vyhýbat nepřátelské střelbě. Pokud se mimozemský útočník dostane až k dolní hraně obrazovky nebo hráč přijde o všechny životy a hra okamžitě končí. Z herního hlediska se jedná o klasický model postupně zvyšující se obtížnosti v rámci jednoduchého a dynamického cyklu.

2.3 Ovládání hráče a střelba

Hráčova vesmírná loď je ovládána pomocí klávesových šipek pro pohyb doleva a doprava, nebo pomocí kláves písmen A (vlevo) a D (vpravo). Klávesa mezerník slouží pro vystřelení střely směrem vzhůru. Ovládání je plynulé a pohyb je v reálném čase, tedy bez zpoždění, což je zásadní pro správnou hratelnost a hráčský zážitek. Jedním z mnoha prvků, které hru odlišují od originálu, je například začlenění dvou kanónů umístěných po stranách hráčovy lodi. Tyto kanóny nejsou jen estetický detail, naopak, jsou využívány a vystřelují projektily střídavě – nejdříve z levého a poté z pravého. Tento princip vytváří iluzi realističtější střelby, ztěžuje hráčovi se trefit a přidává vizuální pestrost. Každý projektil se pohybuje směrem vzhůru konstantní rychlostí a po střetnutí s nepřítelem je odstraněn z herní plochy. Projektil se také odstraní, pokud dosáhne horní hrany herní obrazovky. Hráč nemůže vystřelit další projektil, dokud se jeho předchozí projektil zobrazuje, tato funkce opět slouží k ztížení hry a k vylepšení hráčova zážitku.



2.4 Pohyb a chování nepřátel

Nepřátelští útočníci se objevují v horní části herního pole a jsou uspořádáni do pravidelných řad. Pohybují se horizontálně v jednom směru až k okraji obrazovky, kde dojde ke změně směru a současnému posunu celé formace o jeden krok směrem dolů. Tento vzorec se neustále opakuje, a tím se zajišťuje plynulý a předvídatelný pohyb. V pokročilejších vlnách se mění rychlost pohybu, i frekvence střelby nepřátel. Ti jsou schopni střílet projektily směrem dolů, a tím ohrožovat hráče. Náhodné prvky ve střelbě projektilů zvyšují obtížnost hry, a nutí hráče k neustálému soustředění a rychlé reakci.

2.5 Herní rozhraní a informační prvky

Herní rozhraní je navrženo jednoduše a přehledně, a zároveň má vzbuzovat dojem arkádové retro hry. V horní části obrazovky se zobrazuje počet zbývajících životů hráče, aktuální skóre a aktuální číslo vlny, ve které se hráč momentálně nachází. Tyto prvky slouží k orientaci během hry a zároveň poskytují hráčovi informace o úspěšnosti. Skóre se zvyšuje za každého zasáhnutého útočníka, a jeho hodnota je přímě úměrná aktuální vlně, to znamená že čím vyšší vlna, tím více bodů je za zásah nepřítele hráčovi přiděleno. To motivuje hráče ke zlepšování se a překonávání předchozích výkonů. Počet životů se zobrazuje v numerické podobě, nikoliv formou ikon, což zajišťuje dostatečnou srozumitelnost.

2.6 Systém vln a obtížnost

Hra je rozdělena na postupně se opakující vlny. Každá vlna má stejný počet mimozemských nepřátel. V počátečních vlnách je hra jednodušší a hráč má čas se seznámit s ovládáním a herním průběhem. S každou další vlnou se obtížnost zvyšuje tím, že se nepřátelé zrychlují, a intenzita jejich střelby se zvyšuje, čímž se zvyšují požadavky na náročnost a na přesnost a rychlost hráčovy reakce.



2.7 Ukončení hry

K ukončení hry dochází pouze ve dvou situacích, a to, pokud hráč přijde o všechny životy, nebo pokud alespoň jeden nepřítel dosáhne dolní hrany herní obrazovky. Po skončení hry se herní smyčka zastaví, a zobrazí se „Game Over“ obrazovka, která informuje o výsledném skóre a umožňuje restart hry. Tento prvek ukončuje herní cyklus, a poskytuje tak hráči možnost analyzovat vlastní výkon, případně motivaci ke zlepšení při dalším pokusu, protože nejvyšších 5 zaznamenaného skóre se ukládá a zobrazuje.

3 Použité technologie

Při vývoji hry Galaxy Invaders byly využity standardní webové technologie, konkrétně HTML5⁶, CSS3⁷ a JavaScript.⁸ Tyto technologie byly zvoleny z důvodu autorovi předchozí zkušenosti, jejich široké dostupnosti, nezávislosti na komerčních licencích a kompatibility s naprostou většinou nepoužívanějších moderních webových prohlížečů. Projekt nevyužívá žádné placené nástroje, knihovny ani externí frameworky, což zajišťuje maximální transparentnost a volnost při případném dalším rozšiřování hry.

⁶ MDN WEB DOCS (MOZILLA DEVELOPER NETWORK). *HTML5*. Online. 2005. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [cit. 2025-04-28].

⁷ WORLD WIDE WEB CONSORTIUM. *CSS*. Online. 2006. Dostupné z: <https://www.w3.org/Style/CSS/Overview.en.html>. [cit. 2025-04-28].

⁸ MDN WEB DOCS (MOZILLA DEVELOPER NETWORK). *JavaScript*. Online. 2005. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [cit. 2025-04-28].



3.1 HTML5

HTML5 slouží jako základní kostra projektu. Struktura této herní webové aplikace je realizována prostřednictvím elementu `<canvas>`, který umožňuje vykreslování 2D hry přímo v prostředí webového prohlížeče. Tento prvek byl vybrán vzhledem k autorovi předchozí zkušenosti, a jeho jednoduchému použití pro potřeby herního vývoje a logiky. Hlavní stránka projektu (`index.html`) obsahuje kromě plátna také základní strukturu rozhraní, například tlačítko pro spuštění hry a prostor pro zobrazování skóre a vlny na „game over“ obrazovce.

3.2 CSS3

Pro vzhled této webové aplikace byla využita technologie CSS3. Hlavním cílem bylo vytvořit jednoduché a funkční uživatelské prostředí. CSS bylo použito k nastavení pozadí herního plátna a úpravě vzhledu tlačítek v celé aplikaci. Důraz byl kladen na arkádový retro efekt, ale i na minimalismus tak, aby nedocházelo k zbytečnému rozptylování hráče během samotné hry.



3.3 JavaScript

JavaScript slouží jako nejdůležitější prostředek pro logiku celé aplikace a herní logiky. Byly využity moderní funkce, jako jsou například arrow funkce, `const` a let proměnné a kód byl rozdělen do jednotlivých oddělených částí. Tyto rozdělené scripty obsahují všechny klíčové herní mechaniky. Veškerý kód byl vyvíjen samostatně bez použití různých online generátorů nebo předpřipravených šablon. Pro řešení jednotlivých problémů byly použity různé volně dostupné zdroje jako je například Stack Overflow⁹, Mozilla Developer Network (MDN)¹⁰, oficiální dokumentace jazyků a technologií. Vývoj hry Galaxy Invaders byl rozdělen do několika částí. Každá část byla zapsána v samostatném scriptu, aby bylo možné efektivněji přidávat různé herní prvky.

⁹ Stack Overflow. Online. 2008. Dostupné z: <https://stackoverflow.com/>. [cit. 2025-04-28].

¹⁰ MDN WEB DOCS (MOZILLA DEVELOPER NETWORK). *MDN Web Docs (Mozilla Developer Network)*. Online. 2005. Dostupné z: <https://developer.mozilla.org/en-US/>. [cit. 2025-04-28].



4 Vývoj

4.1 Main.js

Na začátku souboru jsou provedeny importy všech potřebných funkcí z ostatních souborů (bullets.js, enemies.js, enemyBullets.js, sound.js) a nastavení herního plátna

4.1.1 Definice hráče a jeho vykreslování

Hráč je reprezentován objektem obsahujícím jeho velikost, pozici a rychlost. K jeho vykreslení je využit obrázek rakety.

```
33 const player = {
34   width: 0,
35   height: 0,
36   x: 0,
37   y: 0,
38   speed: 0,
39   dx: 0,
40 };
41 const playerImg = new Image();
42 playerImg.src = "res/img/raketa.png";
43
44 const drawPlayer = () =>
45   ctx.drawImage(playerImg, player.x, player.y, player.width, player.height);
```

Zdrojový kód č. 1 Definice hráče

Objekt player obsahuje vlastnosti potřebné pro vykreslování obrázku¹¹ a pohyb hráče.

¹¹ OPENAI. *Obrázek vygenerovaný pomocí ChatGPT*. Online. 2025. Dostupné z: <https://chatgpt.com/>. [cit. 2025-04-28]



4.1.2 Ovládání hráče pomocí klávesnice

Raketka je ovládána pomocí kláves (šipka vlevo, šipka vpravo, A, D, a, d). Stisknutí kláves nastavuje horizontální rychlost hráče (dx).

```
50 document.addEventListener("keydown", (event) => {  
51   if (event.key === "ArrowLeft" || event.key === "a" || event.key === "A")  
52     player.dx = -player.speed;  
53   else if (event.key === "ArrowRight" || event.key === "d" || event.key === "D")  
54     player.dx = player.speed;  
55 });  
56  
57 document.addEventListener("keyup", (event) => {  
58   if (["ArrowLeft", "ArrowRight", "a", "d", "A", "D"].includes(event.key)) player.dx = 0;  
59 });
```

Zdrojový kód č. 2 Ovládání hráče

Na základě stisku kláves se změní směr a rychlost pohybu hráče.

4.1.3 Aktualizace pohybu hráče (update)

Při každém snímku se aktualizuje pozice hráče podle rychlosti a delta time. Hráč je omezen na hranice plátna.

```
61 const update = (deltaTime) => {  
62   player.x += player.dx * deltaTime;  
63   if (player.x < 0) player.x = 0;  
64   if (player.x + player.width > canvas.width)  
65     player.x = canvas.width - player.width;  
66 };
```

Zdrojový kód č. 3 Aktualizace pohybu hráče

Funkce update posouvá hráče a zabraňuje mu opustit herní pole (canvas).



4.1.4 Správa životů hráče

Při zásahu nepřátelským projektilem se hráči sníží počet životů. Při ztrátě všech životů se hra ukončí.

```
70 export const handlePlayerHit = () => {  
71   gameState.playerLives--;  
72   sfx.damage.play();  
73   console.log(`Zbývající životy: ${gameState.playerLives}`);  
74   if (gameState.playerLives <= 0) showGameOver();  
75 };
```

Zdrojový kód č. 4 Funkce pro odečítání životů

Funkce `handlePlayerHit` se exportuje a když se zavolá tak snižuje počet životů a vyvolává konec hry při jejich vyčerpání.



4.1.5 Vykreslování skóre, životů a vlny

V horní části obrazovky se zobrazuje aktuální skóre hráče, počet jeho zbývajících životů a číslo aktuální vlny¹². Informace jsou vykreslovány na plátno pomocí funkce `drawScore` která používá `fillText`.

```
78 const drawScore = () => {  
79   ctx.fillStyle = "white";  
80   ctx.font = "24px 'Press Start 2P', monospace";  
81   ctx.textAlign = "left";  
82   ctx.fillText(`Score: ${gameState.score}`, 20, 40);  
83   ctx.textAlign = "center";  
84   ctx.fillText(`Lives: ${gameState.playerLives}`, canvas.width / 2, 30);  
85   ctx.textAlign = "right";  
86   ctx.fillText(`Wave: ${waveLevel}`, canvas.width - 20, 30);  
87 };
```

Zdrojový kód č. 5 Funkce pro vykreslování skóre, životů a vlny

Skóre je zarovnáno vlevo nahoře, životy ve středu nahoře a číslo vlny vpravo nahoře.

¹² GOOGLE. *Font Press Start 2P*. Online. Dostupné z: <https://fonts.google.com/specimen/Press+Start+2P>. [cit. 2025-04-28].



4.1.6 Úvodní obrazovka a start hry

Před spuštěním samotné hry se zobrazí úvodní obrazovka s možností spustit hru nebo zobrazit instrukce. Ovládání úvodního menu je uskutečněno pomocí tlačítek.

```
100 const showStartScreen = () => {
101   instructionsScreen.style.display = "none";
102   canvas.style.display = "none";
103   startMenu.style.display = "flex";
104 };
105
106 startGameButton.addEventListener("click", () => {
107   sfx.button.play();
108   startMenu.style.display = "none";
109   canvas.style.display = "block";
110   sfx.gameStart.play();
111   gameStarted = true;
112   gameLoop();
113 });
114
115 instructionsButton.addEventListener("click", () => {
116   sfx.button.play();
117   startMenu.style.display = "none";
118   instructionsScreen.style.display = "flex";
119   showingInstructions = true;
120 });
121
122 const backToStart = () => {
123   sfx.button.play();
124   showingInstructions = false;
125   instructionsScreen.style.display = "none";
126   startMenu.style.display = "flex";
127 };
128 window.backToStart = backToStart;
```

Zdrojový kód č. 6 Zobrazení úvodní obrazovky a proces po stisknutí tlačítek

Funkce showStartScreen přepíná viditelnost HTML prvků tak, aby se zobrazilo hlavní menu.

Funkce addEventListener čeká na zmáčknutí tlačítka a poté provede příslušnou akci.



4.1.7 Obrazovka Game Over a ukládání skóre

Při prohře je hráči zobrazena obrazovka „Game Over“ s jeho výsledným skóre a dosaženou vlnou. Skóre je uloženo do lokálního úložiště prohlížeče (localStorage).

```
147 export const showGameOver = () => {
148   if (gameOverDisplayed) return;
149   frozenFrame = new Image();
150   frozenFrame.src = canvas.toDataURL();
151   sfx.gameOver.play();
152   console.log("Game Over!");
153   gameOver = true;
154   gameStarted = false;
155   gameOverDisplayed = true;
156   finalScore.textContent = `Score: ${gameState.score}`;
157   waveNumber.textContent = `Wave: ${waveLevel}`;
158   saveHighScore(gameState.score);
159   updateHighScoresList();
160   gameOverScreen.style.display = "block";
161 };
162
163 // ukladani score
164
165 const saveHighScore = (score) => {
166   let scores = JSON.parse(localStorage.getItem("highScores")) || [];
167   scores.push(score);
168   scores.sort((a, b) => b - a);
169   scores = scores.slice(0, 5);
170   localStorage.setItem("highScores", JSON.stringify(scores));
171 };
172
173 const updateHighScoresList = () => {
174   const scores = JSON.parse(localStorage.getItem("highScores")) || [];
175   highScoresList.innerHTML = scores.map((s) => `<li>${s}</li>`).join("");
176 };
```

Zdrojový kód č. 7 Zobrazení game over obrazovky a ukládání skóre

Funkce showGameOverScreen zobrazuje „Game Over“ obrazovku a zobrazuje nejvyšší dosažená skóre hráče. Funkce saveHighScore ukládá nejlepší výsledky lokálně do úložiště prohlížeče a funkce updateHighScoreList zajišťuje jejich setřídění od nejvyššího po nejnižší.



4.1.8 Herní cyklus (gameLoop)

Herní cyklus řídí celkový běh hry, aktualizuje stav hry na základě času, vykresluje hráče, projektily i nepřátele i nepřátelské projektily, a řeší přechod mezi různými stavy hry (úvodní obrazovka, hraní, „game over“).

```
184 let gameLoop = (currentTime = performance.now()) => {
185   const deltaTime = (currentTime - lastTime) / 10;
186   lastTime = currentTime;
187
188   ctx.clearRect(0, 0, canvas.width, canvas.height);
189
190   if (gameStarted && !gameOver) {
191     drawPlayer();
192     update(deltaTime);
193     updateBullets(deltaTime);
194     drawBullets();
195     updateEnemies(deltaTime);
196     drawEnemies();
197     updateEnemyBullets(deltaTime);
198     drawEnemyBullets();
199     drawScore();
200   } else if (gameOver && frozenFrame) {
201     ctx.drawImage(frozenFrame, 0, 0, canvas.width, canvas.height);
202     return;
203   } else if (!showingInstructions) {
204     showStartScreen();
205   }
206   requestAnimationFrame(gameLoop);
207 };
208
209 gameLoop();
```

Zdrojový kód č. 8 Herní cyklus gameLoop

Funkce gameLoop spravuje stav hry podle toho, zda se hraje, zda hra skončila, nebo je hráč na úvodní obrazovce. Pomocí requestAnimationFrame a performance.now je zajištěno efektivní vykreslování a správné načasování snímků.



4.2 Sound.js

Script `sound.js` slouží k načítání a správě zvukových efektů, které doprovázejí různé herní akce, jako je start hry¹³, střelba¹⁴, zničení nepřítele¹⁵, postoupení do další vlny¹⁶, konec hry¹⁷, stisknutí tlačítka¹⁸ nebo když hráč přijde o život¹⁹. Zvukové efekty výrazně zvyšují hráčův zážitek ze hry a zároveň poskytují zpětnou vazbu hráčovi.

```
1  export const sfx = {
2      gameStart: new Audio("res/sfx/game-start.wav"),
3      shoot: new Audio("res/sfx/shoot.wav"),
4      gameOver: new Audio("res/sfx/game-over.wav"),
5      pop: new Audio("res/sfx/pop.flac"),
6      damage: new Audio("res/sfx/damage.wav"),
7      levelUp: new Audio("res/sfx/levelup.wav"),
8      button: new Audio("res/sfx/button.wav")
9  };
10
11  sfx.gameStart.volume = 0.6;
12  sfx.shoot.volume = 0.4;
13  sfx.gameOver.volume = 0.6;
14  sfx.pop.volume = 0.6;
15  sfx.damage.volume = 0.6;
16  sfx.levelUp.volume = 0.9;
17  sfx.button.volume = 0.9;
```

Zdrojový kód č. 9

Všechny zvukové efekty jsou načteny jako konkrétní objekty `Audio` s přiřazenou cestou k příslušnému zvukovému souboru. Po načtení je každému zvuku nastavena vhodná hlasitost.

¹³ 8-Bit Arcade Video Game Start Sound Effect, Gun Reload and Jump !!. Online. 2016. Dostupné z: <https://freesound.org/s/368691/>. [cit. 2025-04-28].

¹⁴ Laser2.wav. Online. 2010. Dostupné z: <https://freesound.org/s/89490/>. [cit. 2025-04-28].

¹⁵ Pop 9. Online. 2015. Dostupné z: <https://freesound.org/s/328119/>. [cit. 2025-04-28].

¹⁶ LevelUp.wav. Online. 2021. Dostupné z: <https://freesound.org/s/609335/>. [cit. 2025-04-28].

¹⁷ 8bit_status_4.wav. Online. 2018. Dostupné z: <https://freesound.org/s/425345/>. [cit. 2025-04-28].

¹⁸ Button 7.wav. Online. 2010. Dostupné z: <https://freesound.org/s/107154/>. [cit. 2025-04-28].

¹⁹ 8-bit Deny/Error sound. Online. 2020. Dostupné z: <https://freesound.org/s/551543/>. [cit. 2025-04-28].



4.3 enemies.js

Nepřátelé jsou generováni do formace na základě definovaných parametrů, konkrétně počet řad a sloupců, rozestupy a rychlost pohybu. Pro grafické zobrazení nepřátel jsou náhodně vybírány obrázky.

```
16 export const createEnemies = (  
17   rows,  
18   cols,  
19   spacingX,  
20   spacingY,  
21   enemyWidth,  
22   enemyHeight,  
23   speed  
24 ) => {  
25   enemies.length = 0;  
26  
27   const startX =  
28     (canvas.width - (cols * (enemyWidth + spacingX) - spacingX)) / 2;  
29   const startY = 50;  
30  
31   for (let row = 0; row < rows; row++) {  
32     for (let col = 0; col < cols; col++) {  
33       const img = new Image();  
34       img.src = realEnemies[Math.floor(Math.random() * realEnemies.length)];  
35  
36       enemies.push({  
37         x: startX + col * (enemyWidth + spacingX),  
38         y: startY + row * (enemyHeight + spacingY),  
39         width: enemyWidth,  
40         height: enemyHeight,  
41         speedX: speed,  
42         direction: 1,  
43         image: img  
44       });  
45     }  
46   }  
47 };
```

Zdrojový kód č. 10

Funkce createEnemies generuje nepřátele do pravidelné formace se zadanými rozměry a náhodnými obrázky nepřátel²⁰.

²⁰ OPENAI. *Obrázky vygenerované pomocí ChatGPT*. Online. 2025. Dostupné z: <https://chatgpt.com/>. [cit. 2025-04-28].



4.3.1 Pohyb nepřátel a změna směru

Nepřátelé se pohybují horizontálně v jednom směru. Pokud některý nepřítel dosáhne okraje plátna, změní celá formace směr a posune se o jednu řadu dolů.

```
for (let i = 0; i < enemies.length; i++) {  
  const nextX = enemies[i].x + enemies[i].speedX * enemies[i].direction * deltaTime;  
  if (nextX + enemies[i].width >= canvas.width || nextX <= 0) {  
    changeDirection = true;  
    break;  
  }  
}  
for (let i = enemies.length - 1; i >= 0; i--) {  
  if (changeDirection) {  
    enemies[i].direction *= -1;  
    enemies[i].y += 20;  
  }  
  enemies[i].x += enemies[i].speedX * enemies[i].direction * deltaTime;  
}
```

Zdrojový kód č. 11 Pohyb a změna směru nepřátel

Při dosažení okraje herního pole nepřátelé otočí směr pohybu a seskočí o řádek níže, čímž se postupně čím dál více přibližují hráči.



4.3.2 Detekce kolizí projektilů s nepřáteli

Každý nepřítel kontroluje kolizi se střelami hráče. Při zásahu je nepřítel i střela odstraněn a hráči je připsáno skóre.

```
68     if (  
69         bullets[j].x < enemies[i].x + enemies[i].width &&  
70         bullets[j].x + bullets[j].width > enemies[i].x &&  
71         bullets[j].y < enemies[i].y + enemies[i].height &&  
72         bullets[j].y + bullets[j].height > enemies[i].y  
73     ) {  
74         sfx.pop.play();  
75         enemies.splice(i, 1);  
76         bullets.splice(j, 1);  
77         setCanShoot(true);  
78         increaseScore(waveLevel);  
79         break;  
80     }  
81 }  
82 }
```

Zdrojový kód č. 12 Detekce zásahu nepřítele

Kontrola překrývání hranic zajišťuje správnou detekci zásahů nepřátel hráčovými projektily.



4.3.3 Kontrola přistání nepřátel (prohra)

Pokud některý nepřítel dosáhne dolní hrany obrazovky, hra je okamžitě ukončena a hráči je zobrazeno Game Over.

```
85     if (enemies[j].y + enemies[j].height >= canvas.height - 100) {  
86         console.log("Game Over: enemies landed!");  
87         requestAnimationFrame(() => showGameOver());  
88     }
```

Zdrojový kód č. 13 Ukončení hry při přistání nepřítele

Pokud nepřítel dosáhne spodní hrany plátna, je volána funkce showGameOver, která ukončí hru.

4.3.4 Správa přechodu na další vlnu

Po zničení všech nepřátel aktuální vlny je hráči přidán jeden život a generuje se nová, obtížnější vlna s vyšší rychlostí nepřátel a vyšší intenzitou střelby.

```
91     if (enemies.length === 0) {  
92         gameState.playerLives++;  
93         sfx.levelUp.play();  
94         waveLevel++;  
95         const newSpeed = 1 + waveLevel * 0.5;  
96         createEnemies(3, 6, 20, 20, 60, 60, newSpeed);  
97     }  
98 }
```

Zdrojový kód č. 14 Přechod na novou vlnu

Funkce automaticky zvyšuje obtížnost každé nové vlny a odměňuje hráče přidáním života.



4.3.5 Vykreslování nepřátel

Každý nepřítel je na plátno vykreslen pomocí svého obrázku.

```
100 export const drawEnemies = () => {  
101   enemies.forEach((enemy) => {  
102     if (enemy.image.complete) ctx.drawImage(enemy.image, enemy.x, enemy.y, enemy.width, enemy.height);  
103   });  
104 };
```

Zdrojový kód č. 15 Vykreslení nepřátel

Funkce drawEnemies zajišťuje vizuální zobrazení všech živých nepřátel na herním poli.

4.4 enemyBullets.js

Soubor enemyBullets.js obsahuje logiku náhodné střelby nepřátel a pohybu nepřátelských projektilů, detekci zásahu hráče a vykreslování střel na plátno.



4.4.1 Střelba nepřátel

Každý nepřítel, má při každém snímku malou šanci vystřelit. Pravděpodobnost střelby se zvyšuje s každou další vlnou, čímž roste obtížnost hry.

```
8 export const updateEnemyBullets = (deltaTime) => {
9   const currentShootChance = shootChance + waveLevel * 0.001;
10  enemies.forEach(enemy => {
11    if (Math.random() < currentShootChance) {
12      enemyBullets.push({
13        x: enemy.x + enemy.width / 2 - 5,
14        y: enemy.y + enemy.height,
15        width: 10,
16        height: 20,
17        speed: 5 + waveLevel * 0.5
18      });
19    }
20  });
```

Zdrojový kód č. 16 Náhodná střelba nepřátel

Každý nepřítel má v každém snímku určitou pravděpodobnost vystřelit svůj projektil, který je následně uložen do pole enemyBullets.



4.4.2 Detekce zásahu hráče

Nepřátelské projektily se pohybují směrem dolů. Pokud projektil zasáhne hráče, je střela odstraněna a volána funkce pro odečtení hráčova života.

```
22     for (let i = enemyBullets.length - 1; i >= 0; i--) {
23         enemyBullets[i].y += enemyBullets[i].speed * deltaTime;
24         if (
25             enemyBullets[i].x < player.x + player.width &&
26             enemyBullets[i].x + enemyBullets[i].width > player.x &&
27             enemyBullets[i].y < player.y + player.height &&
28             enemyBullets[i].y + enemyBullets[i].height > player.y
29         ) {
30             enemyBullets.splice(i, 1);
31             handlePlayerHit();
32             continue;
```

Zdrojový kód č. 17 Detekce zásahu hráče

Pokud nepřátelský projektil narazí do hráče, je zavolána funkce handlePlayerHit pro snížení počtu životů.

4.4.3 Vykreslování nepřátelských střel

Nepřátelské střely jsou na plátně zobrazeny jako bílé obdélníky.

```
38     export const drawEnemyBullets = () => {
39         ctx.fillStyle = "white";
40         enemyBullets.forEach(bullet => {
41             ctx.fillRect(bullet.x, bullet.y, bullet.width, bullet.height);
42         });
43     };
```

Zdrojový kód č. 18 Vykreslení nepřátelských střel

Funkce drawEnemyBullets zobrazí všechny aktuálně aktivní nepřátelské střely v herním poli.



4.5 bullets.js

Soubor bullets.js obsahuje logiku střelby projektilů hráče, jejich správu, aktualizaci pohybu projektilů a jejich vykreslení na herním plátně. Odlišnou funkcí od originální hry je střídavá střelba z levého a pravého kanónu.

4.5.1 Střelba hráče (střídání kanónů)

Při stisknutí mezerníku hráč vystřelí projektil. Střelba projektilů se střídá mezi levým a pravým kanónem vesmírné lodi. Po výstřelu je střelba dočasně zablokována, dokud střela neopustí obrazovku nebo nezasáhne nepřítele.

```
13 document.addEventListener("keydown", (event) => {
14   if (event.key === " " && canShoot) {
15     const cannon = cannons[cannonIndex];
16     cannonIndex = (cannonIndex + 1) % cannons.length;
17
18     bullets.push({
19       x: player.x + player.width * cannon.offsetX - 5,
20       y: player.y,
21       width: 10,
22       height: 20,
23       speed: canvas.height * 0.015,
24     });
25     setCanShoot(false);
26
27     sfx.shoot.currentTime = 0;
28     sfx.shoot.play();
29   }
30 });
```

Zdrojový kód č. 19 Střídavá střelba hráče

Funkce přidá nový projektil do pole bullets a přepne kanón, ze kterého bude vystřelen další projektil.



4.5.2 Aktualizace pohybu projektilů

Projektily hráče se pohybují směrem vzhůru konstantní rychlostí. Pokud projektil opustí horní část obrazovky, je odstraněn z pole a hráč může znovu vystřelit.

```
32 export const updateBullets = () => {
33   for (let i = 0; i < bullets.length; i++) {
34     bullets[i].y -= bullets[i].speed;
35     if (bullets[i].y + bullets[i].height < 0) {
36       bullets.splice(i, 1);
37       setCanShoot(true);
38       i--;
39     }
40   }
41 };
```

Zdrojový kód č. 20 Aktualizace pohybu střel

Projektily se pohybují nahoru a po opuštění obrazovky se odstraní, čímž je uvolněna možnost další střelby.

4.5.3 Vykreslování střel

Střely hráče jsou na herním plátně vykreslovány jako červené obdélníky.

```
43 export const drawBullets = () => {
44   ctx.fillStyle = "red";
45   for (let bullet of bullets) {
46     ctx.fillRect(bullet.x, bullet.y, bullet.width, bullet.height);
47   }
48 }
```

Zdrojový kód č. 21 Vykreslení střel

Funkce drawBullets zobrazí všechny aktivní projektily hráče jako malé červené obdélníky.



4.6 Problematika při vývoji

Vývoj hry Galaxy Invaders přinesl několik problémů a výzev, které byl autor nucen v průběhu vývoje práce překonat. Tyto problémy autor řešil hledáním veřejně dostupných informací, testováním a optimalizací kódu.

4.6.1 Použití delta time

Při prvních testech hry se objevilo velmi nežádoucí chování, a to že rychlost hry měnila v závislosti na výkonu konkrétního zařízení a počtu snímků za sekundu. Tento problém byl způsoben tím, že pohyby byly počítány absolutně, na každý snímek, bez ohledu na jeho délku.

4.6.1.1 Řešení

Přidání správného výpočtu „delta time“ umožnila konzistentní rychlost hry napříč různými zařízeními a snímkovacími frekvencemi.

```
185     const deltaTime = (currentTime - lastTime) / 10;  
186     lastTime = currentTime;
```

Zdrojový kód č. 22 Delta time

Přidání „delta time“ však bylo potřeba správně přidat do všech potřebných funkcí, což způsobilo problémy při snaze o vyřešení původního problému.

4.6.2 „Start screen“ a „Game over screen“

Zpočátku bylo obtížné správně přepínat mezi herním plátnem a HTML prvky úvodní obrazovky, instrukcí a „game over“ obrazovky bez chybných přechodů.



4.6.2.1 Řešení

Po mnoha dlouhých pokusech a vyhledávání z veřejně dostupných zdrojů byl kód konečně opraven tak, aby se úvodní obrazovka, instrukce i „Game over“ obrazovka zobrazovali správně ve správný čas.

4.7 Testování

Po dokončení základní hry Galaxy Invaders následovala konečná fáze testování. Cílem bylo ověřit správnou funkčnost všech herních mechanik, stabilitu hry, kompatibilitu v různých prostředích a vyváženost obtížnosti.

4.7.1 Drobné úpravy

Při testování hry autor objevil nevyváženost herní obtížnosti, a poté provedl menší řadu úprav, jako je například přidání života při každé poražené vlně, zmenšení šance na vystřelení nepřátelských projektilů a zpomalení a zvětšení nepřátelských jednotek.

5 Možné budoucí rozšíření

Hra Galaxy Invaders je již ve své základní podobě plně funkční, i přesto však její struktura umožňuje další rozšiřování a vylepšování v budoucnosti.

5.1 Přidání herního příběhu

Přidání krátkého příběhu, který by uváděl hráče do kontextu hry, by zvýšila její atraktivitu. Příběh by mohl být prezentován formou komiksových úvodních scén na začátku nebo mezi jednotlivými vlnami.



5.2 Různé herní módy

Přidání různých módů hry by výrazně zvýšilo její zajímavost, v budoucnu se hra Galaxy Invaders může rozšířit například o: Časový mód – omezený čas na dosažení co nejvyššího skóre, Survival mód – hráč bude mít pouze jeden život, Multiplayer mód – možnost dvou hráčů na jednom zařízení

5.3 Achievements

Rozsáhlý systém odměn za dosažení určitých milníků ve hře (např. „100 zabitých nepřátel“, „5 vln poraženo bez ztráty života“ apod.) by podpořil motivaci hráčů ke zlepšování.

5.4 Sběr herní měny a obchod

Zavedení herní měny během hraní by umožnilo hráčům odemykat různé bonusy anebo vizuální změny charakteru, pozadí, projektilů, štít, zrychlení střelby apod.



6 Závěr

Cílem této práce bylo vytvoření jednoduché, ale plně funkční 2D střílečích hry inspirované klasickým arkádovým titulem Space Invaders. Tento cíl se podařilo úspěšně naplnit a výsledná aplikace Galaxy Invaders obsahuje všechny základní herní mechaniky, včetně pohybu hráče, střelby, pohybu nepřátel, detekce kolizí, správy skóre a životů a dynamického zvyšování obtížnosti.

Vývoj hry probíhal postupně od návrhu herního prostředí, přes implementaci herní smyčky, vykreslování objektů, až po přidání úvodní a koncové obrazovky. V průběhu práce bylo nutné řešit několik problémů, zejména správné fungování delta time, což rozhodně přispělo k rozvoji autorova programátorského myšlení a schopnosti efektivního řešení problémů.

Hra byla testována na různých zařízeních a v různých prohlížečích, přičemž byla ověřena její stabilita, plynulost a správná funkčnost všech klíčových herních mechanik. Díky tomu je výsledný produkt dobře hratelný a technicky stabilní.

Tento projekt zároveň nabízí potenciál k dalšímu rozšiřování v budoucnu, například přidáním příběhu, nových herních módů, achievementů, herní měny a skinů.

Výsledná hra Galaxy Invaders tedy nejen splňuje požadavky zadání, ale zároveň poskytuje pevný základ pro možné další vylepšení a rozšíření, což svědčí o kvalitním návrhu její struktury a kódu.



7 Přílohy

7.1 Zdroje

WIKIPEDIA. *Space Invaders!*. Online. 2004, 21.4.2025. Dostupné z: https://en.wikipedia.org/wiki/Space_Invaders. [cit. 2025-04-28].

WIKIPEDIA. *HTML*. Online. 2006, 19.4.2025. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Markup_Language. [cit. 2025-04-28].

WIKIPEDIA. *CSS*. Online. 2006, 11.4.2025. Dostupné z: https://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styl. [cit. 2025-04-28].

WIKIPEDIA. *JavaScript*. Online. 2006, 20.2.2025. Dostupné z: <https://cs.wikipedia.org/wiki/JavaScript>. [cit. 2025-04-28].

WIKIPEDIA. *Space Invaders!*. Online. 2004, 21.4.2025. Dostupné z: https://en.wikipedia.org/wiki/Space_Invaders. [cit. 2025-04-28].

MDN WEB DOCS (MOZILLA DEVELOPER NETWORK). *HTML5*. Online. 2005. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [cit. 2025-04-28].

WORLD WIDE WEB CONSORTIUM. *CSS*. Online. 2006. Dostupné z: <https://www.w3.org/Style/CSS/Overview.en.html>. [cit. 2025-04-28].

MDN WEB DOCS (MOZILLA DEVELOPER NETWORK). *JavaScript*. Online. 2005. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [cit. 2025-04-28].

Stack Overflow. Online. 2008. Dostupné z: <https://stackoverflow.com/>. [cit. 2025-04-28].

MDN WEB DOCS (MOZILLA DEVELOPER NETWORK). *MDN Web Docs (Mozilla Developer Network)*. Online. 2005. Dostupné z: <https://developer.mozilla.org/en-US/>. [cit. 2025-04-28].

8-Bit Arcade Video Game Start Sound Effect, Gun Reload and Jump !!. Online. 2016. Dostupné z: <https://freesound.org/s/368691/>. [cit. 2025-04-28].

Laser2.wav. Online. 2010. Dostupné z: <https://freesound.org/s/89490/>. [cit. 2025-04-28].

Pop 9. Online. 2015. Dostupné z: <https://freesound.org/s/328119/>. [cit. 2025-04-28].

LevelUp.wav. Online. 2021. Dostupné z: <https://freesound.org/s/609335/>. [cit. 2025-04-28].

8bit_status_4.wav. Online. 2018. Dostupné z: <https://freesound.org/s/425345/>. [cit. 2025-04-28].

Button 7.wav. Online. 2010. Dostupné z: <https://freesound.org/s/107154/>. [cit. 2025-04-28].

8-bit Deny/Error sound. Online. 2020. Dostupné z: <https://freesound.org/s/551543/>. [cit. 2025-04-28].

OPENAI. *Obrázky vygenerované pomocí ChatGPT*. Online. 2025. Dostupné z: <https://chatgpt.com/>. [cit. 2025-04-28].

GOOGLE. *Font Press Start 2P*. Online. Dostupné z: <https://fonts.google.com/specimen/Press+Start+2P>. [cit. 2025-04-28].

- **Mozilla Developer Network (MDN)** – oficiální dokumentace JavaScriptu.
(<https://developer.mozilla.org/>)
- **Stack Overflow** – online programátorské forum
(<https://stackoverflow.com/>)
- **Zvukové efekty** – volně dostupné zvuky z platforem poskytujících royalty free audio
(<https://freesound.org/>).
- **Obrázky nepřátel** – generované pomocí volně dostupného AI nástroje pro generování obrázků



(<https://chatgpt.com/>)

- Font "Press Start 2P" – Google Fonts

(<https://fonts.google.com/specimen/Press+Start+2P/>)