ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd

Předmět UUR

Semestrální práce na téma:

Project: Nebularis (Asteroids-like hra)

Autor: Jan Unger

Datum: 28.05.2025

Obsah

1.1 Přehled projektu

Tento dokument poskytuje komplexní dokumentaci k webové hře inspirované klasikou Asteroids, vytvořené pomocí Reactu a HTML Canvasu. Projekt si klade za cíl poskytnout plynulý a zábavný herní zážitek s moderními prvky, jako je vícejazyčná podpora a dynamické nastavení.

1.2 Cíle a zadání

Hlavním cílem bylo vytvořit robustní a hratelnou webovou aplikaci s následujícími klíčovými vlastnostmi:

- **Plynulé a realistické ovládání lodi:** Implementace setrvačnosti, akcelerace a rotace pro autentický pocit z pilotování vesmírné lodi.
- **Rozmanitá herní mechanika:** Střelba, naváděné rakety, drony a různé typy nepřátel (asteroidy s rozdělením).
- **Progresivní obtížnost:** Zvyšující se výzva s postupem hry a zvyšujícím se skóre.
- **Uživatelsky přívětivé rozhraní:** Intuitivní menu, nastavení a informační prvky během hry.
- **Vícejazyčná podpora:** Možnost přepínání jazyků (CZ, EN, PL).
- **Personalizace:** Možnost změny barev herních prvků a aktivace "Kachničkového módu".

1.3 Popis hry a základní mechaniky

Hráč ovládá vesmírnou loď v prostředí plném asteroidů (nebo kachniček v Kachničkovém módu). Cílem je ničit nepřátele a sbírat skóre.

- **Ovládání:** Loď se ovládá pomocí kláves WASD (W - tah vpřed, A/D - rotace, S - brzda). Střelba se provádí mezerníkem (Spacebar). Naváděné rakety se odpalují klávesou R. Hru lze pozastavit stiskem ESC.
- **Životy:** Hráč začíná se 4 životy. Za každých získaných 8 000 bodů se přičte jeden život (maximálně 4). Kolize s asteroidy nebo zničení dronu sníží životy.
- **Štít:** Hráčova loď je vybavena štítem, který absorbuje jeden zásah. Po zničení štítu následuje cooldown, během kterého se štít regeneruje.
- **Vylepšení střelby:** S rostoucím skóre se hráčova střelba vylepšuje (více projektilů v různých úhlech).
- **Drony:** Po dosažení určitých skóre hráč získává doprovodné drony, které automaticky střílí na nejbližší asteroidy.
- **Rakety:** Hráč může odpalovat naváděné rakety, které sledují nejbližší asteroid.
- **Asteroidy:** Asteroidy se generují z okrajů obrazovky a po zničení se rozdělí na menší kusy.

- **"Kachničkový mód":** Speciální herní režim, který změní vizuál asteroidů na kachničky.

1.4 Technologie

- **Frontend Framework:** React
- **Jazyk:** JavaScript
- **Vykreslování grafiky:** HTML Canvas API (vlastní implementace herního enginu)
- **Správa stavu:** React Context API
- **Build nástroj:** Vite (s pluginem @vitejs/plugin-react)
- **Styling:** Čisté CSS
- **Perzistentní ukládání:** localStorage (pro nastavení jazyka a barev)

2. Architektura a Struktura projektu

Projekt je organizován modulárně s jasně definovanými rolemi pro jednotlivé soubory a adresáře, což usnadňuje čitelnost, údržbu a škálovatelnost.

2.1 Kořenová struktura

Poznámka k assetům: Obrázky asteroidu a kachničky se nachází jak v public/, tak v src/assets/. Pro optimální správu Vite projektu by bylo lepší mít všechny assety, které jsou importovány přímo do JavaScriptu (jako asteroidImage v GameCanvas.jsx), v src/assets/. Soubory v public/ jsou přístupné přímo přes root URL.

2.2 Přehled hlavních modulů a komponent

- **main.jsx**: Vstupní bod aplikace, který renderuje App.jsx do HTML elementu s ID "root".
- **App.jsx**: Hlavní komponenta, která řídí zobrazení jednotlivých "obrazovek" hry (menu, hra, nastavení, about me) na základě interního stavu. Také obaluje celou aplikaci LanguageProviderem pro přístup ke globálnímu kontextu.
- **GameCanvas.jsx**: Jádro hry. Komponenta, která obsahuje veškerou herní logiku, fyzikální simulace, detekci kolizí a vykreslování herního světa na HTML Canvas. Spravuje stav hráče, asteroidů, střel a dronů.
- **LanguageContext.jsx**: Definuje React Context, který poskytuje vícejazyčné texty, herní barvy a stav "Ducky módu" všem komponentám, které ho odebírají. Také spravuje uložení jazyka a barev do localStorage.
- **UI Komponenty (Menu.jsx, Settings.jsx, PauseOverlay.jsx, GameOverScreen.jsx, AboutMe.jsx, Upgrades.jsx)**: Tyto komponenty tvoří uživatelské rozhraní, které se zobrazuje nad herním plátnem nebo na samostatných obrazovkách. Jsou zodpovědné za interakci s uživatelem a zobrazení relevantních informací.

2.3 Správa stavu (Context API)

Aplikace využívá **React Context API** pro efektivní správu globálního stavu, zejména pro:

- **Jazykové preference:** LanguageContext ukládá aktuálně zvolený jazyk a poskytuje funkci pro přepínání jazyků. Všechny komponenty, které potřebují lokalizované texty, mohou k tomuto kontextu přistupovat.
- **Herní barvy:** Konfigurovatelné barvy pro hráče, projektily, asteroidy atd. jsou uloženy v kontextu, což umožňuje snadnou personalizaci vzhledu hry.
- **Ducky mód:** Stav, zda je aktivován "Kačenkový mód", je také sdílen přes kontext, což umožňuje GameCanvas.jsx dynamicky měnit grafiku nepřátel.
- **Perzistence:** Kontext automaticky ukládá zvolený jazyk a barvy do localStorage, aby se tato nastavení zachovala i po zavření prohlížeče.

2.4 Styling

Styling je řešen pomocí čistého CSS, kde každý modul nebo komponenta má svůj vlastní CSS soubor (např. GameCanvas.css, Menu.css). To zajišťuje modularitu stylů a snižuje riziko kolizí jmen. Styly jsou importovány přímo do příslušných komponent.

3. Implementace klíčových částí

3.1 App.jsx: Hlavní aplikační komponenta a správa obrazovek

App.jsx slouží jako hlavní kontejner aplikace. Spravuje, která "obrazovka" je aktuálně zobrazena uživateli (např. hlavní menu, hra, nastavení, informace o autorovi, obrazovka Game Over).

- **Stav currentScreen**: Určuje, co se má vykreslit. Možné hodnoty jsou 'mainMenu', 'game', 'settings', 'aboutMe', 'gameOver'.
- **LanguageProvider**: Obaluje celý obsah, čímž zajišťuje, že všechny podřízené komponenty mají přístup k jazykovému kontextu a dalším globálním nastavením (barvy, Ducky mód).
- **Podmíněné vykreslování**: Využívá if/else if strukturu pro podmíněné vykreslování komponent na základě stavu currentScreen.
- **Ref pro GameCanvas**: Používá useRef a useImperativeHandle (v GameCanvas.jsx) k přímému volání metod restart a pauseGame na instanci GameCanvas z App.jsx, což umožňuje resetování hry nebo její pauzu z hlavního aplikačního stavu.

3.2 main.jsx: Vstupní bod aplikace

Tento soubor je standardním vstupním bodem pro React aplikace vytvořené pomocí Vite.

- Importuje React a ReactDOM pro vytvoření React aplikace.
- Importuje globální styly (i když zde není žádný globální CSS soubor, obvykle by zde byl).

- Vytváří React root a renderuje komponentu <App /> do HTML elementu v index.html.

3.3 GameCanvas.jsx: Jádro herního enginu

GameCanvas.jsx je nejkomplexnější komponenta, která implementuje veškerou herní logiku a vykreslování. Používá useRef pro přístup k HTML <canvas> elementu a udržování mutovatelných herních stavů (např. pole asteroidů, stav hráče). forwardRef a useImperativeHandle jsou použity pro zpřístupnění metod restart a pauseGame z rodičovské komponenty App.jsx.

3.3.1 Základní herní smyčka

- Používá requestAnimationFrame pro plynulé animace a setInterval pro generování asteroidů (když hra není pozastavena).
- update(): Funkce, která aktualizuje stav všech herních objektů (pozice, rychlost, kolize) v každém snímku.
- draw(): Funkce, která vykresluje všechny herní objekty na plátno v každém snímku.

3.3.2 Pohyb hráče a fyzika

- **playerRef**: Ref pro stav hráče (pozice, rychlost, úhel, životy, štít, drony, atd.).
- **Ovládání:** Klávesy WASD mění rotation a přidávají velocityX/velocityY (akcelerace).
- **Setrvačnost a tření:** velocityX a velocityY se postupně snižují friction faktorem, simulujícím tření ve vesmíru (nebo brzdění).
- **Obalování obrazovky:** Hráčova loď se objeví na protější straně obrazovky, pokud opustí jeden z okrajů.

3.3.3 Generování a chování asteroidů (včetně Kačenkového módu)

- **ASTEROID_SIZES**: Konstanta definující různé velikosti asteroidů (large, medium, small) s jejich poloměrem, rychlostí, skóre a typem rozpadu.
- **spawnAsteroid()**: Generuje nové asteroidy na náhodných okrajích obrazovky, s náhodným typem a směrem. Limituje maximální počet asteroidů na obrazovce (MAX_ASTEROIDS).
- **Pohyb a rotace:** Asteroidy se pohybují v určeném úhlu a rotují kolem své osy. Obalují se přes okraje obrazovky.
- **Vykreslování:** Vykresluje buď obrázek asteroidu (asteroidImageElementRef), nebo obrázek kachničky (duckyImageElementRef), v závislosti na isDuckyMode. Pokud obrázky nejsou načteny, vykreslí se jednoduché kruhy.

3.3.4 Střelba (lasery, rakety, drony)

- **Projektily hráče:** Vystřelovány mezerníkem. Vylepšují se na základě skóre, zvyšuje se počet a rozptyl projektilů.

- **Rakety:** Odpalovány klávesou R. Jsou to naváděné střely, které sledují nejbližší asteroid.
- **Drony:** Získávají se za skóre. Drony obíhají kolem hráče a automaticky střílí na nejbližší asteroidy s vlastním cooldownem. Jejich rychlost střelby se zvyšuje s počtem dronů.

3.3.5 Kolize

- **Projektil-Asteroid:** Projektily (hráče, dronů, rakety) kolidují s asteroidy. Zničený asteroid přidává skóre (s multiplikátorem podle typu střely) a případně se rozpadne na menší kusy.
- **Hráč-Asteroid:** Kolize s asteroidem poškozuje hráče (buď zničí štít, nebo sníží životy).
- **Dron-Asteroid:** Drony mohou kolidovat s asteroidy a jsou zničeny, což také ničí asteroid. Ztráta dronu ovlivňuje prahovou hodnotu pro získání dalšího dronu.
- **Cooldown po poškození**: Po utrpění poškození je hráč krátce imunní.

3.3.6 Štít

- **hasShield**: Boolean stav indikující aktivitu štítu.
- **SHIELD_COOLDOWN_TIME**: Doba, po kterou je štít po zničení neaktivní.
- **Vizualizace štítu**: Štít je vykreslován jako pulzující kruh kolem hráče, se speciálním blikajícím efektem během regenerace.

3.3.7 Hvězdné pozadí

- **starsRef**: Pole objektů reprezentujících hvězdy.
- **generateStars()**: Generuje počáteční pozice hvězd.
- Hvězdy se pohybují vertikálně a warpují se na vrchol obrazovky, když ji opustí dole, čímž vytváří iluzi neustálého pohybu.

3.4 Uživatelské rozhraní (UI Komponenty)

Všechny UI komponenty jsou čisté React komponenty, které přijímají props pro řízení jejich chování a obsahu. Přistupují k LanguageContext pro vícejazyčnost.

3.4.1 Menu.jsx

Vykresluje hlavní nabídku hry s možnostmi spuštění hry, přechodu do nastavení, zobrazení informací o autorovi a návratu do hlavního menu. Obsahuje tlačítka pro volbu jazyka.

3.4.2 Settings.jsx

Umožňuje uživateli konfigurovat herní nastavení:

- Přepínání jazyka.
- Přepínání "Kačenkového módu".

- Výběr barev pro hráče, projektily, drony, rakety a asteroidy. Změny se okamžitě projevují díky Context API.

3.4.3 PauseOverlay.jsx

Zobrazuje se, když je hra pozastavena. Nabízí možnosti pro pokračování ve hře nebo návrat do hlavního menu.

3.4.4 GameOverScreen.jsx

Zobrazí se po skončení hry. Zobrazuje konečné skóre hráče a nabízí možnost restartovat hru nebo se vrátit do hlavního menu.

3.4.5 AboutMe.jsx

Komponenta pro zobrazení informací o autorovi nebo projektu.

3.4.6 Upgrades.jsx

(Pokud existuje a nebyla zmíněna dříve, je to komponenta, která by pravděpodobně zobrazovala informace o dostupných vylepšeních nebo průběhu vylepšení během hry.) Z dodaného kódu stylů je patrné, že se vztahuje k nějaké vizuální reprezentaci upgradů.

3.5 Kontext (LanguageContext.jsx)

LanguageContext.jsx je srdcem pro správu globálního, sdíleného stavu:

- **LanguageContext a LanguageProvider**: Vytváří kontext a poskytovatele, který udržuje stav jazyka, barev a Ducky módu.
- **Stav language**: Udržuje aktuální jazyk ('en', 'cs', 'pl').
- **Stav gameColors**: Objekt s aktuálními barevnými schématy pro různé herní prvky.
- **Stav isDuckyMode**: Boolean pro aktivaci/deaktivaci Ducky módu.
- **useEffect pro localStorage**: Automaticky ukládá a načítá nastavení jazyka a barev z localStorage, což zajišťuje perzistenci uživatelských preferencí.
- **T (Translation) funkce**: Poskytuje jednoduchou funkci T(key) pro získání lokalizovaného textu.
- **handleColorChange a toggleDuckyMode**: Funkce pro změnu nastavení, které aktualizují stav kontextu a tím i celou aplikaci.

3.6 Styly (CSS)

Styling je modulární a jednoduchý, s jedním CSS souborem pro každou komponentu.

- **GameCanvas.css**: Styly specifické pro herní plátno, zajišťující správné rozměry a umístění.

- **Menu.css, GameOverScreen.css, PauseOverlay.css, AboutMe.css, Upgrades.css**: Styly pro UI komponenty, zahrnující rozložení, typografii, barvy a interaktivní prvky (např. tlačítka).
- **Overlay.css**: Pravděpodobně obsahuje obecné styly pro překryvné vrstvy (overlays), které se zobrazují nad hrou (např. pro pauzu, menu, game over).
- **Flexbox/Grid**: Předpokládá se použití moderních CSS vlastností pro responzivní a adaptivní rozložení.

4. Nastavení a Spuštění

4.1 Závislosti (package.json, package-lock.json)

I když package.json je prázdný, package-lock.json jasně ukazuje, že projekt využívá:

- **React 19.0.0-rc**: Nejnovější release candidate verze Reactu.
- **React DOM**: Knihovna pro práci s DOM v Reactu.
- **Vite**: Rychlý build nástroj pro moderní webové projekty.
- **@vitejs/plugin-react**: Oficiální plugin pro React ve Vite.
- **ESLint**: Nástroj pro linting kódu a zajištění konzistence.

4.2 Build nástroje (vite.config.js)

Soubor vite.config.js je prázdný, což znamená, že projekt používá výchozí konfiguraci Vite. To je pro menší projekty často dostatečné. Standardně Vite automaticky rozpozná React a aplikuje příslušné optimalizace.

4.3 Lokální spuštění (.bat skript)

Projekt se spouští pomocí .bat souboru (pro Windows prostředí). Typický obsah takového skriptu by byl:

- @echo off npm install npm run dev
  npm install: Nainstaluje všechny závislosti projektu definované v package.json (a upřesněné v package-lock.json).
- npm run dev: Spustí vývojový server Vite, který kompiluje a servíruje aplikaci s hot-module reloading.

Pro spuštění by tedy uživatel potřeboval:

1. Nainstalovaný Node.js a npm.
2. Spustit .bat soubor.

5. Zpětná vazba testerů a Hodnocení

5.1 Zpětná vazba testerů

Zpětná vazba od testerů byla klíčová pro iterativní zlepšování hry a její uživatelské přívětivosti. Projekt se aktivně řídil těmito doporučeními a reflektoval je v následných úpravách:

- **Tester 1 (Vizibilita projektilů):** Tester navrhl zvýšit viditelnost projektilů, což vedlo k úpravě jejich barvy na výraznější, lépe viditelnou růžovou. Tato úprava zlepšila vizuální zpětnou vazbu pro hráče během střelby.
    - **Implementace:** Změna barvy projektilů je řízena stavem gameColors.playerBulletColor v LanguageContext.jsx a aplikována ve vykreslovací logice v GameCanvas.jsx v rámci funkce drawProjectile.
- **Tester 2 (Vykreslování na menším rozlišení):** Nahlášeny problémy s vykreslováním na menších rozlišeních, což bylo opraveno optimalizací přizpůsobení plátna na různé velikosti obrazovky. Tím je zajištěna správná funkčnost a vzhled hry na široké škále zařízení.
    - **Implementace:** Responzivní chování plátna je zajištěno CSS pravidly v GameCanvas.css a logikou v GameCanvas.jsx pro dynamickou změnu rozměrů plátna .
- **Tester 3 (Rychlost otáčení lodi):** Doporučil zvýšit rychlost otáčení lodi, což bylo upraveno pro rychlejší a dynamičtější reakce ovládání. Tato změna přispěla k lepšímu pocitu z kontroly lodi a celkově svižnějšímu hernímu zážitku.
    - **Implementace:** Hodnota rotace hráče je upravena v GameCanvas.jsx v rámci funkce pro zpracování vstupů (handleKeyDown/handleKeyUp) a v rámci herní smyčky (updatePlayer).

5.2 Hodnocení

- Aplikace je technicky náročná díky integraci reaktivního UI s herními mechanismy, správou stavu a perzistencí. Významné je využití vlastních i cizích knihoven a real-time logiky.
- Design je čistý, modulární a dobře organizovaný. Kontext a props jsou vhodně použity pro předávání stavů a funkcí.
- Aplikace je funkčně kompletní, s promyšlenou perzistencí a herní logikou.
- GUI je víceobrazovkové, přehledné a navržené s ohledem na oddělení logických celků hry.
- UX je propracované – kombinace vizuální estetiky, použitelnosti, interakčních detailů a jazykové přizpůsobitelnosti zvyšují kvalitu celkového dojmu.
- Vstupy jsou ošetřeny adekvátně k potřebám aplikace. Předejde se tím neplatným konfiguracím.
- U tabulek v Menu byla zakomponována responzivita pro menší rozlišení obrazovek.
- Většina designu byla dělána pomocí CSS, které jsou ve složce styles.