**Katedra počítačov a informatiky**

**Fakulta elektrotechniky a informatiky**

**Technická univerzita v Košiciach**

|  |  |
| --- | --- |
| **Šk. rok: 2024/2025** | **Ladislav Chvastaš, Šimon Pavlišin** |

**Môj Pou**

**Počítačová grafika**

**Systémová príručka**

1.Obsah

[2. Funkcia programu 3](#_Toc186305057)

[3. Popis programu 3](#_Toc186305058)

[3.1. Popis riešenia 3](#_Toc186305059)

[3.2. Popis algoritmov a údajových štruktúr, globálnych premenných 4](#_Toc186305060)

[3.3. Popis modulov, tried a podprogramov 4](#_Toc186305061)

[4. Preklad programu 6](#_Toc186305062)

[4.1. Zoznam zdrojových textov 6](#_Toc186305063)

[4.2. Požiadavky na technické prostriedky pri preklade 6](#_Toc186305064)

[4.3. Požiadavky na programové prostriedky 6](#_Toc186305065)

[4.4. Vlastný preklad 6](#_Toc186305066)

[5. Nadväznosť na iné programové produkty 7](#_Toc186305067)

[6. Zhodnotenie riešenia 7](#_Toc186305068)

[7. Zoznam použitej literatúry 7](#_Toc186305069)

Zoznam obrázkov

Obr. 1 Navrhnutý Use Case Diagram (Diagram prípadov použitia).................................................... 5

# Funkcia programu

Témou tohto semestrálneho projektu bola implementácia programu 3D labyrint pre hľadanie správnej cesty používateľom. Dôležitou súčasťou riešenia bolo zamerať sa na správnosť kolízie používateľa so stenami labyrintu, ako aj na vytvorenie niekoľkých herných kôl (levelov). Výsledné skóre používateľov bolo potrebné ukladať a zobrazovať v tabuľke úspešnosti.

Webová aplikácia hry 3D labyrint pozostáva z dvoch základných súčastí a k nim prislúchajúcich funkcionalít. Prvou z nich je menu, ktoré sa v rámci štruktúry projektu člení na úvodnú alebo štartovaciu stránku, menu pre tvorbu a nastavenie hry (New game), informačnú podstránku (About game) a podstránku zobrazujúcu tabuľku úspešnosti (Hall of fame). Druhá súčasť projektu pozostáva zo samotnej grafickej scény, ktorá vykresľuje aktuálny herný level v 3D priestore.

Vo všeobecnosti programové vybavenie hry umožňuje používateľovi v úvode získať potrebné informácie o samotnej hre a jej ovládaní. V druhom kroku si používateľ môže nastaviť herný cyklus podľa svojho uváženia napr. výberom avatara, pridaním či odstránením levelu alebo nastavením veľkosti herného poľa. Výhodou je, že jednotlivé levely sú vždy generované náhode a nie je možné sa spoliehať na „poznanie daného levelu“. Po úspešnom dokončení všetkých levelov v hre je používateľove skóre (čas) zapísané v tabuľke úspešnosti pri jeho mene. Skóre (čas) je počítané na základe časomiery, ktorá plynie v priebehu celej hry. Získané dáta sú ukladané lokálne a plnia iba informačné účely. Počas priebehu hry je možné hru pozastaviť. V tomto prípade prestane plynúť časomiera a používateľ následne pokračuje od posledného zapamätaného bodu.

# Popis programu

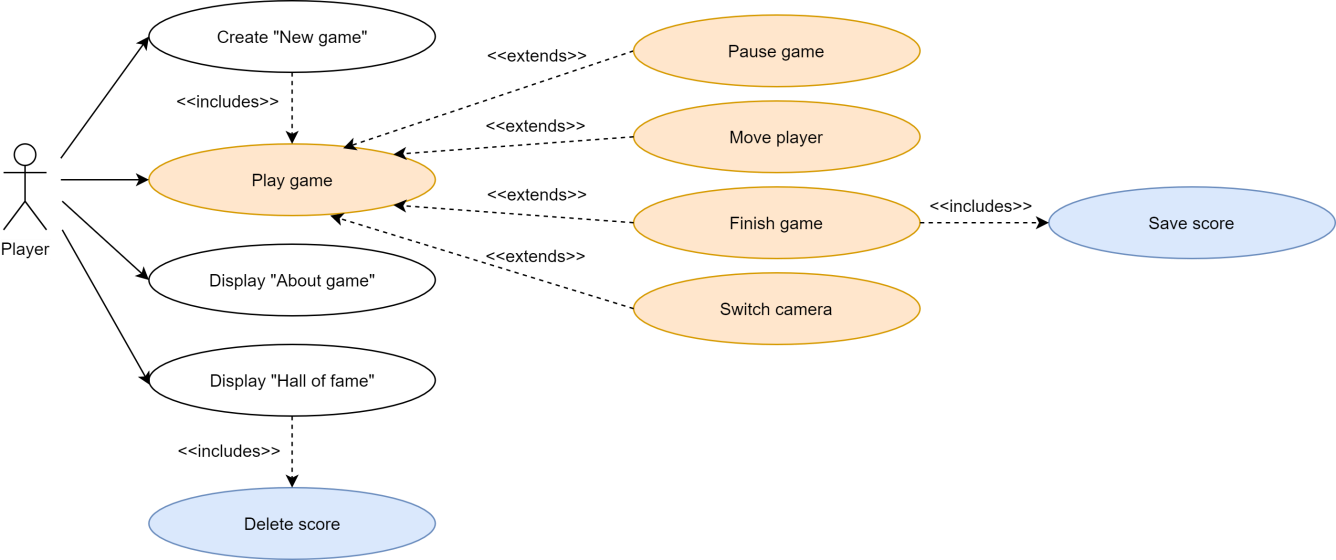
## Popis riešenia

Program je navrhnutý modulárne, čo umožňuje ľahkú rozšíriteľnosť a úravy. Na implementáciu sa využíva:

* Knižnica **Three.js** na tvorbu 3D prostredia.
* Ovládacie prvky s použitím DOM elementov na interakciu.
* Správa stavu prostredníctvom objektu **PouState**.

Hlavné riešenie sa zakladá na zobrazovaní miestností a interakcii hráča s objektmi (napríklad ťahanie jablka v kuchyni). Herná časť minihry obsahuje dynamické prešĺhovanie prekážok s využitím kolíznej detekcie.

Teoretický základ zahŕňa prácu s 3D geometriou, detekciou kolízií cez bounding boxy a vykresľovanie tienľov.



Obr. 1 Navrhnutý Use Case Diagram (Diagram prípadov použitia)

## Popis algoritmov a údajových štruktúr, globálnych premenných

* **Algoritmus kolíznej detekcie**: Využíva triedu THREE.Box3 na vytvorenie ohraničenia pre objekty. Funkcia checkCollision zisťuje prienik dvoch objektov.
* **Správa stavu Poua**: Trieda PouState uchováva hodnoty health, hunger a joy, ktoré sú pravidelne aktualizované a vizualizované pomocou kociek.

Globálne premenné:

* gameState na udržiavanie stavu hry (napr. či je hra ukončená).
* movement na zaznamenanie vstupov klávesnice.
* DIRECTIONS – pole smerov a ich vektorov (*'N', 'S', 'E', 'W'*)

## Popis modulov, tried a podprogramov

* **cubeUtils.js**: Tento modul obsahuje funkcie pre tvorbu stavových kociek, ktoré vizualizujú zdravie, hlad a radosť Poua. Funkcia createStateCube vytvára kocku so sprievodným ikonickým sprite. Funkcia updateCubeScaledynamicky upravuje veľkosť kocky podľa aktuálnej hodnoty stavu.
* **pouState.js**: Obsahuje triedu PouState, ktorá spravuje herný stav Poua. Trieda ponúka metódy na zvyšovanie a znižovanie hodnôt zdravia, hladu a radosti, pričom zaručuje, že hodnoty zostanú v rozsahu 0-100.
* **rollingGameAnimate.js**: Obsahuje hlavnú animačnú slučku hry. Táto slučka zabezpečuje pohyb postavy, posun kamery, dynamické osvetlenie, detekciu kolízií a prechod medzi úrovňami.
* **ui.js**: Tento modul generuje ovládacie prvky používateľského rozhrania, ako sú tlačidlá na prepínanie miestností (obývačka, kuchyňa, ihrisko, minihra). Ovládacie prvky umožňujú jednoduchú interakciu s hrou.
* **rollingGameLevels.js**: Definuje jednotlivé úrovne minihry. Modul poskytuje funkcie setupLevel1 a setupLevel2, ktoré pridávajú prekážky do hernej scény a zabezpečujú zvýšenú obtiažnosť s každou úrovňou.
* **rollingGameCheckCollision.js**: Implementuje funkciu checkCollision, ktorá zisťuje kolízie medzi objektmi pomocou ich ohraničujúcich boxov (bounding boxes).
* **rollingGameEndGame.js**: Poskytuje funkciu na ukončenie hry, nastavuje herný stav na "game over" a voliteľne zobrazí správu "Game Over".
* **rollingGameCreateCamera.js**: Tento modul obsahuje funkciu na vytvorenie perspektívnej kamery s nastavením počiatočnej pozície a uhlu pohľadu.
* **rollingGameCreateRenderer.js**: Poskytuje funkciu na vytvorenie WebGL rendereru s povolením tieňovania a jemných tieňov (soft shadows).
* **rollingGameOnWindowResize.js**: Obsahuje funkciu na dynamické prispôsobenie kamery a renderera veľkosti okna pri zmene rozlíšenia.
* **rollingGameAddBackground.js**: Umožňuje pridanie sférického pozadia pomocou textúry. Táto funkcia vytvára veľkú guľu s obrázkom a zabezpečuje, aby pozadie vyzeralo realisticky.
* **rollingGameAddObstacle.js**: Obsahuje funkcie na pridávanie prekážok do hernej scény. Pridáva rôzne objekty s využitím GLTF modelov, ktoré slúžia ako dynamické prekážky v hre.
* **rollingGameAddSunLight.js**: Poskytuje osvetlenie scény pomocou DirectionalLight, čo simuluje slnečné svetlo. Svetlo sleduje postavu a zabezpečuje, že tiene sú realistické.
* **rollingGameControls.js**: Implementuje ovládanie postavy pomocou klávesnice. Funkcie handleKeyDown a handleKeyUp spracúvajú pohybové vstupy hráča.
* **rollingGameCreateBoundaries.js**: Vytvára neviditeľné hranice na okrajoch hernej platformy, ktoré zabezpečujú, že hráč sa nemôže dostať mimo definovaného priestoru.
* **main.js**: Slúži ako vstupný bod aplikácie. Inicializuje hru a spúšťa hlavný animačný cyklus.
* **init.js**: Obsahuje základnú inicializáciu scény, kamery, rendereru a pridanie ovládacích prvkov. Zabezpečuje tiež načítanie textúr a iných zdrojov.
* **pou.js**: Pridáva postavu Poua do scény pomocou 2D textúr, ktoré sú pripojené na rovinu.
* **textureLoader.js**: Umožňuje načítanie a správu textúr pre hru. Obsahuje funkcie na prednačítanie textúr a zabezpečuje ich dostupnosť pre ostatné moduly.
* **kitchen.js**: Vytvára kuchyňu ako jednu z herných miestností. Obsahuje interakčné prvky ako jablko, ktoré môže hráč premiestňovať.
* **livingRoom.js**: Implementuje obývačku, základnú miestnosť hry, kde sa Pou nachádza na začiatku.
* **playground.js**: Vytvára ihrisko ako zábavnú miestnosť hry.
* **rollingGame.js**: Spája všetky komponenty minihry vrátane prekážok, osvetlenia, a dynamického pohybu postavy.

# Preklad programu

## Zoznam zdrojových textov

* cubeUtils.js
* pouState.js
* rollingGameAnimate.js
* ui.js
* rollingGameLevels.js
* rollingGameCheckCollision.js
* rollingGameEndGame.js
* rollingGameCreateCamera.js
* rollingGameCreateRenderer.js
* rollingGameOnWindowResize.js
* rollingGameAddBackground.js
* rollingGameAddObstacle.js
* rollingGameAddSunLight.js
* rollingGameControls.js
* rollingGameCreateBoundaries.js
* main.js
* init.js
* pou.js
* textureLoader.js
* kitchen.js
* livingRoom.js
* playground.js
* rollingGame.js

## Požiadavky na technické prostriedky pri preklade

* **Počítač**: Moderný PC alebo notebook
* **RAM**: Min. 8 GB
* **Diskový priestor**: 500 MB na inštaláciu knižníc a zdrojových textov

## Požiadavky na programové prostriedky

* **Operačný systém**: Windows, macOS alebo Linux
* **Prehliadač**: Chrome, Firefox (na spustenie hry cez localhost)
* **Node.js**: Na správu knižníc a servera
* **Knižnice**: Three.js, Webpack, Babel

## Vlastný preklad

* Spustite príkaz npm install na inštaláciu všetkých potrebných knižníc.
* Použite príkaz npm run build na zbalenie projektu do spustiteľného formátu.
* Spustite projekt cez npm start a otvorte prehliadač na localhost:8080.

# Nadväznosť na iné programové produkty

Hra využíva knižnicu Three.js na správu 3D grafiky a DOM na ovládací prvky. Projekt je ľahko škálovateľný pre integráciu webových technológií alebo databáz.

# Zhodnotenie riešenia

* **Obmedzenia**: Program zatiaľ neobsahuje ukladaní stavov Poua do databázy ani multiplayer podporu.
* **Možnosti čalšieho vývoja**: Pridať čalšie miestnosti, vylepšiť grafiku a rozšíriť herný obsah (napr. minihry).

# Zoznam použitej literatúry

1. Three.js Documentation: https://threejs.org/docs/
2. MDN Web Docs: https://developer.mozilla.org/