

**Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach**

**Brankár
Počítačová grafika**

Systémová príručka

Šk. rok: 2021/2022

Meno učiteľa: Ing. Miriama Mattová

Autor: Samuel Kucko

Obsah

Zoznam obrázkov	2
1. Funkcia programu	3
2. Popis programu	3
2.1. Popis riešenia	3
2.2. Popis algoritmov a údajových štruktúr, globálnych premenných	4
2.2.1. Globálne premenné modulu <i>scene.js</i>	4
2.2.2. Funkcie v module <i>scene.js</i>	4
2.3. Popis modulov, tried a podprogramov	5
3. Preklad programu	6
3.1. Zoznam zdrojových textov	6
3.2. Požiadavky na technické prostriedky pri preklade	6
3.3. Požiadavky na programové prostriedky	6
3.4. Vlastný preklad	6
4. Nadväznosť na iné programové produkty	7
5. Zhodnotenie riešenia	8

Zoznam obrázkov

Obrázok 1 - Use case diagram	3
------------------------------------	---

1. Funkcia programu

Témou môjho semestrálneho projektu bola jednoduchá športová 3D hra - brankár. Dôležitou súčasťou riešenia bolo zamerať sa na použitie zákonov fyziky v hre a správne animovanie pohybov brankára.

Webová aplikácia jednoduchej športovej 3D hry – brankára 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, podstránku kde hrá prebieha (Play), informačnú podstránku (How to play). Druhá súčasť projektu pozostáva zo samotnej grafickej scény, ktorá vykresľuje aktuálny hru v 3D priestore.

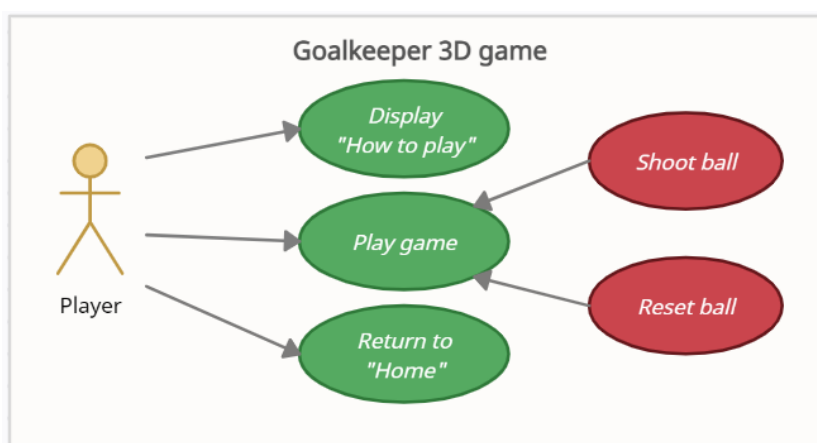
Na podstránke „How to play“ sa používateľ dozvie všeobecné informácie o hre, a aj ako funguje a ako ju ovládať. V hre je cieľom streliť gól a snažiť sa ich steliť čo najviac po sebe bez chytenia brankára.

2. Popis programu

Webová aplikácia hry Brankár je tvorená komplexnou funkcionalitou vybraných webových technológií, ktoré sú detailnejšie charakterizované v kap. 5. Nasledujúce časti systémovej príručky sú zamerané na opis ťažiskových algoritmov, modulov a funkcií, ktoré súvisia s generovaním herného poľa, hlavným modulom, ktorý zabezpečuje inicializáciu a vykresľovanie 3D scény.

2.1. Popis riešenia

Na obrázku nižšie je uvedený diagram prípadov použitia z pohľadu možností hráča v rámci hry ale aj z pohľadu výstupu webovej aplikácie. Ovládanie hry, čiže jej funkčné prvky sú zobrazené červenou farbou.



Obrázok 1 - Use case diagram

2.2. Popis algoritmov a údajových štruktúr, globálnych premenných

Nasledujúce podkapitoly rozoberajú opis vytvorenie herného poľa a zoznam globálnych premenných v module *scene.js*.

2.2.1. Globálne premenné modulu *scene.js*

`camera` – premenná, ktorá drží objekt kamery Three.js.

`scene` – premenná, ktorá drží objekt scény Three.js, resp. Ammo.js.

`renderer` – premenná držiaca objekt rendereru Three.js.

`clock` – premenná, ktorá drží objekt času Three.js..

`glTFLoader` – premenná, ktorá drží objekt načítavania objektov formátu glTF Three.js.

`ball` – premenná, ktorá drží objekt lopty.

`shootBallObj` – premenná, ktorá drží objekt vystrelenej lopty.

`goalkeeper` – premenná, ktorá drží objekt brankára.

`shootsound` – premenná, ktorá drží objekt zvuku kopu do lopty.

`curve` – premenná, ktorá drží objekt krivky pohybu brankára.

`posIndex` – premenná, ktorá drží číslo pozície.

`moveDirection` – premenná, ktorá drží smery pohybu lopty.

`FLAGS` – konštanta, ktorá slúži na definovanie kinematického objektu.

`physicWorld` – premenná, ktorá drží objekt sveta, kde platia fyzikálne zákony - Ammo.js.

`tmpTrans` – premenná, ktorá drží objekt transformácie Ammo.js.

`rigidBodies` – premenná (pole), ktorá drží objekty, na ktoré je aplikovaná fyzika.

2.2.2. Funkcie v module *scene.js*

Zoznam funkcií (*scene.js*):

- **`function init()`** – funkcia inicializuje elementárne objekty Three.js a Ammo.js a postupne volá funkcie na nastavenie grafických objektov, fyziky a vytvorenie objektov.
- **`function render()`** – sa vykonáva v slučke. Volá funkciu *moveBall()* ak by chcel používateľ hýbať s loptou. Zabezpečuje update fyzikálnej stránky sveta v reálnom čase, a zisťuje, či užívateľ vystrelil loptu.

- **function setupGraphics()** – funkcia, ktorá inicializuje objekty ako *clock*, *renderer*, *scene*, *cameral*, pridáva svetlo do scény, načítava zvuk kopu do lopty.
- **function setupPhysicalWorld()** – funkcia, ktorá nastavuje fyzikálne zákony na vytvorenú scénu.
- **function updatePhysics(deltaTime)** – funkcia, ktorá kontroluje objekty, na ktorých sú aplikované fyzikálne zákony v reálnom čase.
- **function addLight()** – funkcia, ktorá pridáva svetlo do scény.
- **function addPlane()** – funkcia, ktorá vytvára plochu pre scénu.
- **function addSphere()** – funkcia, ktorá vytvára sféru pre scénu.
- **function addObject()** – funkcia, ktorá vyvoláva jednotlivé funkcie na vytvorenie objektov.
- **funkcia addGoal()** – funkcia, ktorá pridá do scény bránku a nastaví pre ňu fyzikálne zákony.
- **funkcia addBall()** – funkcia, ktorá pridá do scény loptu a nastaví pre ňu fyzikálne zákony.
- **function shootBall(shootNum)** – funkcia, ktorá na základe parametra *shootNum* zistí, kam ma strela smerovať a zavolá funkciu pre vykonanie strely.
- **function createShootBall(shootVector)** – funkcia, ktorá na základe parametra *shootVector* zistí vektor pre smer lopty, vytvorí objekt lopty a aplikuje na ňu daný smerový vektor.
- **function addGoalkeeper()** – funkcia, ktorá pridá do scény brankára a nastaví na neho fyzikálne zákony.
- **function setupCurveForMoveGoalkeeper()** – funkcia, ktorá vytvorí krivku pre pohyb brankára.
- **function moveGoalkeeper(newPos)** – funkcia, ktorá pohne s brankárom podľa novej pozície v parametre *newPos*.
- **function loadShootSound()** – funkcia, ktorá načíta zvuk kopu do lopty.
- **function moveBall()** – funkcia, ktorá riadi voľný pohyb lopty po ploche.
- **function setupEventHandlers()** – funkcia, ktorá riadi zisťovanie stlačenia tlačidla na klávesnici a podľa toho vyvoláva nejaký úkon.

2.3. Popis modulov, tried a podprogramov

Riešenie tohto semestrálneho zadania obsahuje iba jeden modul (*scene.js*).

3. Preklad programu

3.1. Zoznam zdrojových textov

Všetky opísané algoritmy, moduly a funkcie sú k dispozícii v projekte „pg1-zadanie“ na platforme GitHub, konkrétne na URL adrese <https://github.com/kucko99/pg1-zadanie>.

3.2. Požiadavky na technické prostriedky pri preklade

Nižšie uvedené požiadavky na programové prostriedky a aplikácia opísaných webových technológií popísaná v kap. 4, bola overená na nasledujúcom hardvérovom technickom vybavení, ktoré poskytovalo výborný výkon.

- **Operačný systém** – Windows 11 Home
- **Procesor** – Intel Core i5-8265U 1.60 GHz
- **Grafická karta** – NVIDIA GeForce MX230/2GB
- **Operačná pamäť** – 20.00 GB

3.3. Požiadavky na programové prostriedky

Webová aplikácia nebeží priamo vo webovom prehliadači a je potrebné si stiahnuť .zip súbor celého projektu z platformy GitHub. Následne po stiahnutí treba extrahovať súbor a spustiť /html/index.html súbor. Programové prostriedky pre správny chod aplikácie preto spočívajú vo výbere vhodného webového prehliadača, ktorý má plnú podporu pre vykresľovanie 3D grafiky pomocou knižnice WebGL, resp. jej nadstavby Three.js.

Aplikácia bola otestovaná na týchto webových prehliadačoch – Mozilla Firefox (96.0.3) a Microsoft Edge (97.0.1072.76) v desktopovej verzii. Odporúča sa využívanie oboch prehliadačov. Najlepší výkon v rámci 3D scény sa podarilo dosiahnuť použitím prehliadača Microsoft Edge.

3.4. Vlastný preklad

Webová aplikácia po stiahnutí .zip súboru projektu, jeho extrahovaní a spustení z platformy GitHub nepotrebuje pre nasledujúci vývoj aplikácie vlastný preklad či inštaláciu doplnkových knižníc alebo modulov. Stačí projekt otvoriť v ľubovoľnom IDE, ktoré má podporu písania zdrojových kódov

v jazyku JavaScript. Projekt bol vyvíjaný použitím IDE JetBrains WebStorm vo verzii 2019.3.4. Pre budúci vývoj ale použitie tohto vývojového prostredia nie je podmienkou.

4. Nadväznosť na iné programové produkty

Na vývoj webovej aplikácie boli využité nasledujúce webové technológie. Pred vykonávaním dodatočných úprav, či pokračovaní v ďalšom vývoji sa preto odporúča ich stredná až pokročilá znalosť.

- **Three.js** – Javascript knižnica používaná na tvorbu a zobrazovanie 3D počítačovej grafiky vo webovom prehliadači.
- **Ammo.js** – Plugin pre Three.js. Je to fyzikálny engine „Náboja“ do JavaScriptu.

Okrem uvedených doplnkových programových prostriedkov sa odporúča minimálne základná až pokročilá znalosť elementárnych webových technológií ako HTML, CSS a JavaScript.

5. Zhodnotenie riešenia

Implementácia webovej aplikácie jednoduchej športovej 3D hry – brankára splnilo podľa môjho názoru dostatočnú časť z vopred stanovených požiadaviek. Na druhej strane, kvôli určitým obmedzeniam využitých technológií a obmedzeniam časových kapacít boli niektoré návrhové postupy vynechané a predstavujú možnosti pre budúce doplnenie funkcionality.

- **Nekontrolovanie, či padol gól alebo nie – zobrazenie počtu gólov na obrazovke.**
- **Počítanie počtu gólov daných po sebe, následne ukladanie dát do databázy.**
- **Vytvorenie rebríčku najlepších strelcov.**
- **Vytvorenie animácii zámkrov brankára.**

V tomto zozname sa budúci vývojári môžu inšpirovať a doimplementovať nedopracované časti aplikácie. Pri nasledujúcom vývoji je možné zamerať sa na skvalitnenie herného zážitku alebo vylepšenie vizuálnej stránky.