

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

3D Simulácia Ruského Kola

Počítačová Grafika

Systémová Príručka

Šk. rok:
2024/2025

Róbert Smrek
Dušan Mikloš

Obsah

1. Funkcia programu	3
2. Popis programu.....	4
2.1. Popis riešenia	4
2.2. Use-case diagram (zjednodušený):	4
3. Popis algoritmov a údajových štruktúr.....	5
4. Popis modulov, tried a podprogramov.....	6
5. Preklad programu.....	7
5.1. Zoznam zdrojových textov.....	7
5.2. Požiadavky na technické prostriedky pri preklade.....	7
5.3. Požiadavky na programové prostriedky pri preklade.....	7
5.4. Vlastný preklad.....	7
6. Nadväznosť na iné programové produkty	8
7. Zhodnotenie riešenia.....	9
8. Zoznam použitej literatúry.....	10

1. Funkcia programu

V našom projekte sa zaoberáme simuláciou ruského kola v 3D prostredí pomocou knižnice Three.js. Naším cieľom je vytvoriť interaktívny model, ktorý umožňuje používateľovi meniť rôzne nastavenia, ako napríklad osvetlenie, farbu kola, farby kabínok či pozadie scény. Chceme, aby bola simulácia čo najviac flexibilná, preto sme zahrnuli aj možnosť upraviť fyzikálne vlastnosti, čo zaručuje realistickejší pohyb a dynamiku. Tento projekt ma za úlohu odprezentovať naše technické zručnosti z oblasti predmetu Počítačovej grafiky. Model zároveň poskytuje používateľovi nástroje na prispôsobenie prostredia podľa jeho predstáv. Keďže projekt je vedený ako simulácia, zahrnuli sme taktiež možnosť importovať a exportovať ruské kolo v aktuálnom stave, so všetkými svojimi vlastnosťami

2. Popis programu

2.1. Popis riešenia

- Scéna pracuje s 3D objektmi a textúrami v reálnom čase.
- Používa WebGL renderer s tieňovaním a postprocessingom.
- Základom je stromová štruktúra komponentov (kód v priečinku `src`).
- Na vykreslenie scény slúžia triedy a funkcie z knižnice Three.js (pozri Three.js docs).
- Pri algoritmizácii kola vychádza aplikácia z využitia trigonometrie (pohyb kabínok po kružnici) a z parametrického riadenia (amplitúda, rýchlosť).

2.2. Use-case diagram (zjednodušený):

Používateľ	Akcia
Viewer	Prezerá scénu
Viewer	Kliká na kabínu
Viewer	Exportuje a importuje nastavenia
Viewer	Upráva fyzikálne parametre

3. Popis algoritmov a údajových štruktúr

- **Animácia kolesa**
 - Otočka kolesa sa aktualizuje v hlavnej funkcii ``animate()``.
 - Výpočet kabínkovej pozície: $(x, y) = (\cos(uhol) * polomer, \sin(uhol) * polomer)$.
 - Parameter ``uhol`` je časovo menený o rýchlosť otáčania.
- **Swing efekt kabínok**
 - Amplitúda kývania: ``swingAmplitude``.
 - Rýchlosť kývania: ``swingSpeed``.
 - V každom snímku sa kabínka pootočí o ``Math.sin(čas)``.
- **Štruktúra**
 - Objekty kabín sa ukladajú do poľa ``cabins``.
 - Každá kabína má ``userData`` s atribútmi: ``angle``, ``swing``.

4. Popis modulov, tried a podprogramov

- ``main.js``
 - Vytvára scénu a renderer.
 - Inicializuje kameru, svetlá a ovládacie prvky.
 - Definuje slučku ``animate()`` na vykreslenie.
- ``/src/scene/wheel.js``
 - Funkcia ``createWheel(scene)`` vracia objekt kola s kabínkami.
 - Obsahuje exportované funkcie ``updateWheelColor`` a ``updateCabinColor``.
- ``/src/scene/island.js``
 - Vytvára základ ostrova.
 - Obsahuje modely (tráva, oplotenie, stánky).
- ``/src/components/aboutButton.js``
 - Vytvára informačné tlačidlo v rohu obrazovky.
 - Po prejdení kurzorom zobrazí text v modálnom okne.
- ``/src/gui/debugGUI.js``
 - Používa knižnicu ``lil-gui`` na nastavenie parametrov scény.
 - Umožňuje export a import nastavení do JSON.
- **Global Variables**
 - ``physics`` s atribútmi ``swingAmplitude``, ``swingSpeed``, ``rotationSpeed``.
 - ``raycaster``, ``mouse`` pre interakcie.

5. Preklad programu

5.1. Zoznam zdrojových textov

- ``index.html``
- ``main.js``
- ``style.css``
- ``vite.config.js``
- ``src/components/``
- ``src/scene/``
- ``src/gui/``
- ``src/utils/``

5.2. Požiadavky na technické prostriedky pri preklade

- Počítač s aspoň 8 GB RAM.
- Priestor na disku aspoň 500 MB.
- Podpora hardvérovej akcelerácie (bežné GPU).

5.3. Požiadavky na programové prostriedky pri preklade

- Operačný systém (napr. Windows, Linux).
- Node.js (odporúčaný 16.x a vyšší).
- Balíčkovací manažér (npm alebo yarn).
- Vite dev server (pozri ``vite.config.js``).
- Knižnice ``three``, ``lil-gui``, ``three/examples/jsm/postprocessing/*``.

5.4. Vlastný preklad

1. Nainštalujte Node.js a vytvorte projekt príkazom ``npm init vite@latest``.
2. Skopírujte zdrojové súbory do vytvoreného projektu (podľa štruktúry).
3. V koreňovom priečinku spustíte ``npm install`` na inštaláciu závislostí.
4. Spustíte ``npm run dev``.
5. V prehliadači otvorte adresu ``http://localhost:5173`` (port podľa ``vite.config.js``).

6. Nadväznosť na iné programové produkty

- Podpora importu a exportu JSON konfigurácií (možné prepojenie s inými webovými systémami).
- Využitie WebGL a textúr so súbormi `.png`, `.jpg`.
- Prípadné databázové napojenie (momentálne neimplementované).

7. Zhodnotenie riešenia

Kód je funkčný, vizuálne zrozumiteľný a je možné ho ďalej upraviť. Animácia je priamočiara, ale ponúka parametrizáciu kolesa. Niektoré grafické prvky sa dajú doplniť (napríklad pokročilejšie textúry). Rozhranie v `debugGUI` je jednoduché, dá sa rozšíriť o nové nastavenia.

8. Zoznam použitej literatúry

- Three.js dokumentácia. <https://threejs.org/docs/>
- MDN Web Docs. <https://developer.mozilla.org/>
- Lil-GUI GitHub. <https://github.com/georgealways/lil-gui>