

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Počítačová podpora výučby astronomie



2021

Vedoucí práce: Mgr. Tomáš Kühn,
Ph.D.

Lucia Mahútová

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Lucia Mahútová
Název práce: Počítačová podpora výučby astronómie
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2021
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Tomáš Kühn, Ph.D.
Počet stran: 48
Přílohy: 1 CD/DVD
Jazyk práce: slovenský

Bibliographic info

Author: Lucia Mahútová
Title: Software for Teaching of Astronomy
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2021
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Tomáš Kühn, Ph.D.
Page count: 48
Supplements: 1 CD/DVD
Thesis language: Slovak

Anotácia

Webová výučbová aplikácia zobrazuje pohyby planét a mesiacov v Slnечnej sústave. Užívatelia môžu voliteľne zobrazit obežné dráhy telies a pridať umelé kozmické objekty na orbitu vybranej planéty či mesiaca. Aplikácia ďalej zobrazuje zmenu trajektórie umelého objektu pomocou zmeny rýchlosti alebo pôsobenia sily.

Synopsis

A web learning application is showing the movements of planets and moons in the Solar system. Users can optionally view the orbits of bodies and add artificial space objects to the orbit of the selected planet or moon. The application also shows the change in the trajectory of an artificial object by changing the speed or the force.

Kľúčové slová: Slnечná sústava; vesmír; astronómia; simulácia; výučbová aplikácia

Keywords: Solar system; universe; astronomy; simulation; software for teaching

Rada by som poďakovala vedúcemu bakalárskej práce pánovi Tomášovi Kührovi za odborné vedenie, konzultácie a užitočné návrhy k práci. Ďalej by som chcela poďakovať rodičom a priateľovi, ktorí ma podporovali počas štúdia aj tvorby tejto bakalárskej práce.

Čestne vyhlasujem, že som celú prácu vrátane príloh vypracoval/a samostatne a za použitia iba zdrojov spomínaných v texte práce a uvedených v zozname literatúry.

dátum odovzdania práce

podpis autora

Obsah

1	Úvod	8
2	Existujúce aplikácie	9
2.1	The sky live	9
2.2	Solar system scope	9
2.3	Solar walks lite – Planetarium 3D: Planets system	9
3	Vývoj aplikácie a prostredie behu	11
3.1	Aplikačný software	11
3.1.1	Typy aplikačného softwaru	11
3.2	Webové aplikácie a ich vývoj	11
3.2.1	Webová aplikácia vs Webová stránka	12
3.2.2	Progresívne webové aplikácie	12
3.2.3	Typy webových aplikácií	12
4	Programovacie jazyky pre vedecké a astronomické aplikácie	14
4.1	Príklady programovacích jazykov	14
4.1.1	Jazyk Python	14
4.1.2	Jazyk C++	15
4.1.3	Jazyk IDL	15
4.1.4	Jazyk Fortran	15
4.2	Použité technológie	16
4.2.1	Jazyk HTML a CSS	16
4.2.2	Jazyk JavaScript	16
4.2.3	Knižnica jQuery	18
4.2.4	Knižnica Three.js	18
5	Fyzika vo vesmíre	19
5.1	Retrográdny a prirodzený pohyb	19
5.2	Retrográdna a prirodzená rotácia	19
5.3	Gravitačný ťah	19
5.4	Hlavná a vedľajšia poloos	20
5.5	Excentricita	21
5.6	Rýchlosť a akcelerácia	21
5.7	Umelý kozmický objekt	21
5.7.1	Typy orbít	21
5.7.2	Rýchlosť pohybu a úniková rýchlosť	22
5.7.3	Zmena naklonenia dráhy	22
5.7.4	Zmena obežnej dráhy	23
6	Programátorská dokumentácia	24
6.1	Triedy	24
6.1.1	Triedy InitScene, PlanetScene, ModelScene	24

6.1.2	Trieda JSONManager	26
6.1.3	Trieda SidebarManager	27
6.1.4	Trieda RayCaster	28
6.1.5	Trieda InitPlanets	28
6.1.6	Triedy Sun, Moon, Planet	29
6.1.7	Trieda Orbits	30
6.1.8	Trieda CosmicObject	31
6.2	Ovládanie prekreslenia obrazovky	33
6.3	Nasadenie aplikácie na server	33
6.4	Použité matematické vzorce	34
6.4.1	Výpočet pozície telesa na orbite	34
6.4.2	Výpočet vzdialenosti medzi dvoma telesami	35
7	Užívateľská dokumentácia	36
7.1	Funkcionalita menu	36
7.1.1	Model Slnecnej sústavy	36
7.1.2	Planéty	36
7.1.3	O aplikácii	36
7.1.4	Jazyky aplikácie	36
7.2	Pravý bočný panel	38
7.2.1	Otvorenie a zatvorenie bočného panelu	38
7.2.2	Mená vesmírnych objektov	38
7.2.3	Mesiace	38
7.2.4	Orbity vesmírnych objektov	38
7.2.5	Tlačidlo pre zastavenie/spustenie aplikácie	38
7.2.6	Tlačidlo "Potvrdiť"	38
7.3	Tlačidlá	38
7.3.1	Tlačidlo pre umelý kozmický objekt	38
7.3.2	Tlačidlo pre kameru	39
7.3.3	Tlačidlo pre vycentrovanie modelu	40
7.4	Funkcie pre posúvanie modelu	40
7.5	Klikanie na vesmírne telesá	40
	Záver	41
	Conclusions	42
	A Knížnice, moduly a licencie	43
	B Obsah priloženého CD	44
	Literatúra	45

Zoznam obrázkov

1	Webová aplikácia: The sky live	9
2	Webová aplikácia: Solar system scope	10
3	Mobilná aplikácia: Solar walks lite	10
4	Semi-major a semi-minor os	20
5	Perihélium a afélium	20
6	Pohyb vesmírneho telesa a akcelerácia	21
7	Zmena naklonenia dráhy pre vesmírne teleso	23
8	Zmena obežnej dráhy pomocou prenosovej dráhy	24
9	Menu – časť Planéty	37
10	Pravý bočný panel na ovládanie modelu	37
11	Tlačidlo pre umelý kozmický objekt	39
12	Kliknutie na vesmírne teleso	40

Zoznam viet

1	Definícia (Výpočet prenosovej dráhy)	23
2	Definícia (Výpočet bodov elipsy pre pohyb telesa)	35
3	Definícia (Vzdialenosť dvoch bodov v 3D priestore)	35

Zoznam zdrojových kódov

1	Výpočet pozície planét pomocou JSON súboru	30
2	Nastavenie pozície pre kozmický objekt	32
3	Ovládanie prekreslenia obrazovky – výpočet času	33

1 Úvod

Táto bakalárska práca sa zaoberá zobrazením planét a vybraných mesiacov ako modelu v Slnecnej sústave. Aplikácia je vyvinutá ako nástroj pre vyučovanie fyziky na základných a stredných školách, preto bol uprednostnený 2D model, ktorý pôsobí prehľadnejšie. Model je doplnený o interakciu s vesmírnymi objektmi a rozšírený o pôsobenie fyzikálnych javov na teleso obiehajúce okolo planéty, model teda poukazuje na využitie základných javov vo vesmíre. Študenti si môžu vytvoriť lepšiu predstavu o tom, čo sa deje s vesmírnymi loďami a raketami vyslanými do vesmíru.

Vyučovanie fyziky na základných či stredných školách pozostáva z rôznych odvetví ako sú optika, elektrina, termodynamika, kvantová fyzika, kmitanie, vlnenie, elektromagnetické a gravitačné pole, teória relativity a ďalšie. Osnovy jednotlivých škôl sa líšia a v mnohých prípadoch chýba jednoduchý nástroj pre praktickejšie vyučovanie a zobrazenie fyzikálnych javov. Na prebudenie záujmu u študentov a lepšie pochopenie fyziky v praxi môžu dobre poslúžiť rôzne pokusy, animácie či simulácie s možnosťou zasahovania študentov do ich priebehu. Praktické ukážky poukazujú na to, že fyzikálne vzorce a veličiny majú aj reálne využitie, netýkajú sa teda len fyzikálnych pokusov či olympiád na školách.

Téma vesmíru je dnes často spájaná s osídľovaním planéty Mars či ďalších planét alebo mesiacov v našej Slnecnej sústave. Takýto pokrok ľudstva však vyžaduje prebudenie záujmu o vesmír a astronómiu u budúcich generácií, preto by mohlo byť zaradenie astronómie do vyučovania na školách prínosné pre budúcnosť.

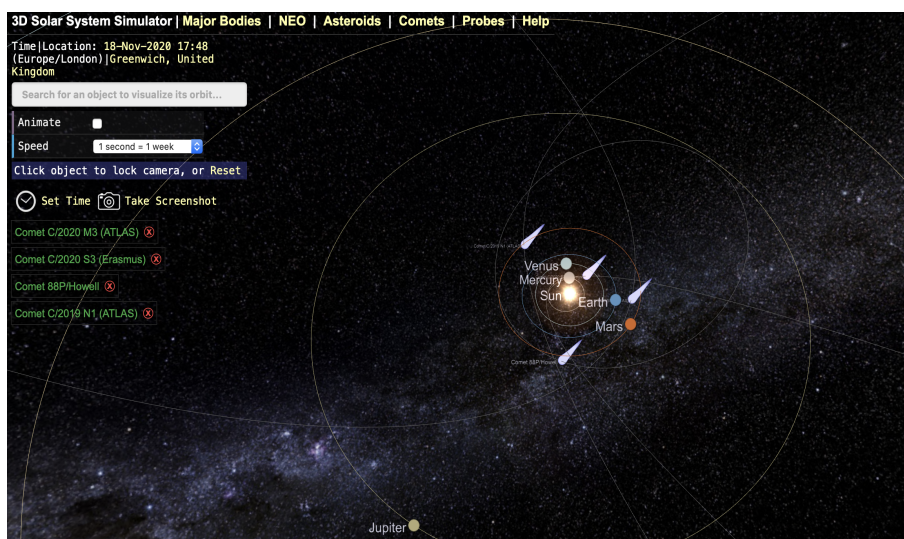
Simulácia pohybu vesmírnych telies je navrhnutá a spracovaná ako webová aplikácia, pre spustenie postačí internetové pripojenie a moderný webový prehliadač. Aplikácia nevyžaduje prihlasovanie užívateľov ani neobsahuje rušivé reklamy, ktoré sú vo veľkom zastúpené na webových stránkach a aplikáciách. Aplikácia je vyvinutá pomocou jazyka JavaScript a základné grafické užívateľské rozhranie je vytvorené pomocou jazykov HTML a CSS. Hlavnou výhodou zvolenej technológie je priama podpora vo webových prehliadačoch, ktoré nie sú zatažené kompiláciou kódu do iného jazyka. Výhodou aplikácie bežiacej vo webovom prehliadači je, že pre svoje spustenie nevyžaduje špecifikáciu platformy ako sú veľmi známe MacOS a Microsoft Windows pre počítače či iOS a Android pre mobilné telefóny.

Pre zobrazenie 3D objektov je využívaná knižnica Three.js, doplnená o textúry planét pre reálnejšie zobrazenie modelu. Mierka jednotlivých planét a orbít vesmírnych telies je upravená vzhľadom na zobrazený model tak, aby boli jednotlivé telesá dobre rozpoznateľné. Pohyb telies okolo stredu Slnecnej sústavy či iných planét je upravený tak, aby bol viditeľný. Vytvorený model nezobrazuje rotácie v reálnom čase.

2 Existujúce aplikácie

2.1 The sky live

Webová aplikácia The sky live [1] obsahuje 3D simulátor nočnej oblohy, kde je možné zobrazit dráhy vesmírnych telies. Užívateľia môžu nastaviť čas, ktorý ovplyvní rýchlosť pohybu planét v modeli. Nastavenie času je však trochu mäťúce a obmedzené na niekoľko predvolených možností. Aplikácia ďalej obsahuje planetárium, ktoré presne popisuje polohy súhvezdí. Medzi negatíva aplikácie patrí zobrazenie planét, aplikácia neposkytuje možnosť ich bližšieho zobrazenia a taktiež lepšie zobrazenie ich povrchu.



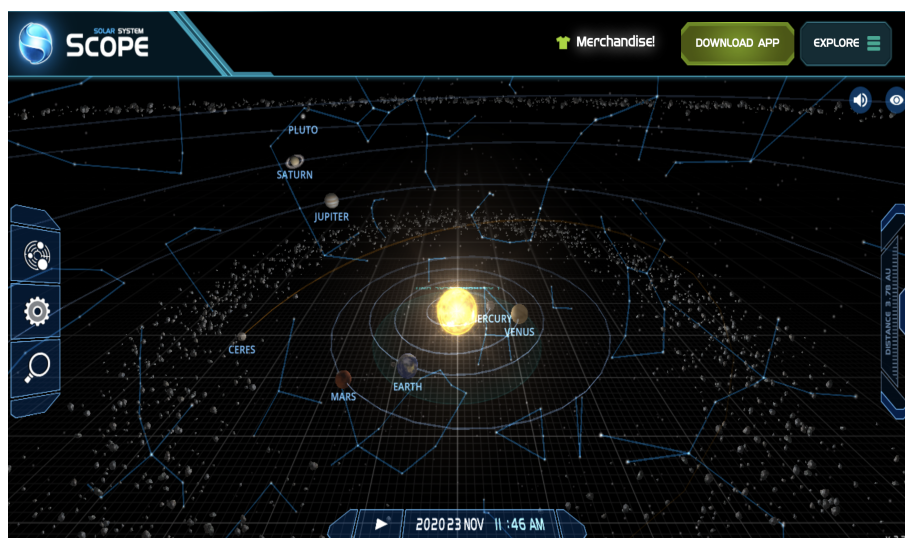
Obr. 1: Webová aplikácia: The sky live

2.2 Solar system scope

Ďalšou webovou aplikáciou je Solar system scope [2]. Výhodou je možnosť zobrazenia vo webovom prehliadači aj stiahnutie na desktop zariadenia. Prvý model predstavuje zobrazenie planét spolu so súhvezdiami. Tento model má veľmi príjemnú grafickú stránku, planéty používajú textúry z vesmírnych pozorovaní. Druhý model zobrazuje pozorovanie súhvezdí nočnej oblohy z povrchu Zeme, grafická stránka je menej prepracovaná. Výhodou je ľahké ovládanie. Nevýhodou aplikácie je základný model, ktorý zobrazuje súčasne polohy planét aj súhvezdí a pôsobí neprehľadne.

2.3 Solar walks lite – Planetarium 3D: Planets system

Mobilná aplikácia Solar walks lite – Planetarium 3D [3] je aplikácia s veľmi pekne prepracovaným 3D modelom Slnecnej sústavy. Aplikácia zobrazuje presné



Obr. 2: Webová aplikácia: Solar system scope

polohy planét a poskytuje užívateľom bližšie informácie o jednotlivých objektoch. Užívateľia uvítajú intuitívne a interaktívne prostredie. Výhodou aplikácie je jej ľahká dostupnosť na viacerých platformách. Nevýhodou je zobrazenie na malej ploche a množstvo reklám, ktoré prerušujú skúmanie modelu.



Obr. 3: Mobilná aplikácia: Solar walks lite

3 Vývoj aplikácie a prostredie behu

3.1 Aplikačný software

Aplikačný software predstavuje program alebo skupinu programov, ktoré sú používané koncovými užívateľmi na špecifické účely ako zapisovanie poznámok, nastavenie alarmu, online vyhľadávanie či hranie hier a ďalšie účely. Účelom softwaru môže byť taktiež pomoc užívateľom s konkrétnym procesom ako zlepšenie produktivity, koncentrácie, komunikácie a zaznamenávanie aktivity.

Každý software okrem systémového softwaru považujeme za aplikačný software, skrátene aplikáciu. Rozdielom je, že systémový software interaguje s iným softwarom alebo hardwarom namiesto interakcie s užívateľmi aplikácií. Medzi známe aplikácie patrí program MS PowerPoint určený na vytváranie prezentácií alebo webový prehliadač, ktorý slúži na prehľadávanie Internetu.

3.1.1 Typy aplikačného softwaru

Pre vývoj aplikácie je dôležité vedieť prostredie behu, pre aký typ koncových užívateľov je aplikácia určená a určiť vhodné nástroje pre vývoj. Príklady aplikačného softwaru:

- Word procesory (Google Docs, MS Word, Pages),
- Databázový software (MySQL, PostgreSQL, Oracle, MongoDB),
- Multimediálny software (Adobe Photoshop, VLC Media Player, Windows Media Player),
- Vzdelávací software (Stellarium, Moodle, MindPlay, KidPix),
- Grafický software (Adobe Photoshop, Maya, Blender),
- Aplikácie prehľadávajúce webové prehliadače (Google Chrome, Safari, Opera, Mozilla Firefox, Microsoft Edge, Internet Explorer),
- Simulačný software (napr. pre inžinierstvo, vzdelávanie, testovanie),
- Software pre vytváranie prezentácií (MS PowerPoint, Adobe Persuasion),
- a iné. [4]

3.2 Webové aplikácie a ich vývoj

Narozdiel od desktopových aplikácií, ktoré sú lokálne spustené na operačnom systéme zariadenia, webové aplikácie bežia na webovom serveri alebo na klientskej časti, v závislosti od použitej technológie. Serverová časť aplikácií zabezpečuje funkcionality pre správnu interakciu s užívateľom. Pre svoj beh vyžadujú internetové pripojenie a sú sprístupnené cez webové prehliadače. Na vybudovanie

webových aplikácií sa používajú základné webové technológie ako HTML, CSS a JavaScript. Taktiež môžu vyžadovať prihlásenie užívateľov.

Webové aplikácie môžu byť doplnené o databázy. Webový prehliadač nemá priamy prístup do databázy, získavanie dát alebo ich ukladanie do databázy sprostredkuje serverová časť.

3.2.1 Webová aplikácia vs Webová stránka

Webová aplikácia je určená pomocou vstupu. Pri vývoji sa používa metóda CRUD čo znamená, že v rámci webovej aplikácie vytvárame, čítame, aktualizujeme a odstraňujeme dáta. Webová stránka je určená pomocou výstupu, kde užívatelia čítajú texty a nemenia ich. Stránky teda slúžia ako zdroj informácií. [5]

3.2.2 Progresívne webové aplikácie

Aplikácie, ktoré sa chovajú ako natívne aplikácie. To znamená, že môžu byť stiahnuté a uložené na domovskú stránku zariadenia. Na rozdiel od webových aplikácií môžu byť používané aj v offline režime, tj. bez využitia internetového pripojenia.

Veľkou výhodou progresívnych aplikácií je, že sú automaticky aktualizované pre užívateľov. Odlišný prístup predstavujú natívne aplikácie, ktoré musia byť aktualizované užívateľom. [6]

Nevýhodou je slabšia podpora webových prehliadačov, v súčasnosti sú progresívne aplikácie podporované prehliadačmi Chrome, Firefox či Opera. Spotrebujú viac batérie ako v prípade natívnych aplikácií, ich beh je pomalší a známe funkcie ako Bluetooth¹ či NFC² nie sú zatiaľ podporované. Taktiež sú webové aplikácie považované za menej bezpečné ako desktopové aplikácie. [7]

3.2.3 Typy webových aplikácií

Vlastnosti aplikácie na strane klienta:

- Kód aplikácie je vyhodnotený webovým prehliadačom (tj. klientom),
- Kód je písaný pomocou skriptovacieho jazyka ako JavaScript, ktorý priamo interaguje s HTML elementmi (napr. tlačítka, zoznamy, textové polia),
- Rýchla odozva na interakciu s užívateľom, pretože jazyky používané na strane klienta sú priamo podporované webovým prehliadačom,
- Výhodné použiť ak nie je potrebné čerpať dáta z databázy pre zmenu informácií na webovej stránke či aplikácii,

¹Bluetooth – bezdrôtová technológia, ktorá slúži na prenos a výmenu dát medzi pripojenými zariadeniami v krátkej vzdialenosti (zvyčajne do vzdialenosti zariadení 10m).

²NFC – bezdrôtová technológia na komunikáciu medzi kompatibilnými zariadeniami, funkčná v krátkej vzdialenosti. Signál je vysielaný pomocou rádiových vln. V súčasnosti je NFC rozšírenou technológiou bezkontaktného platby pre mobilné telefóny či Smartwatch.

- Príklady technológií používaných pri vývoji: HTML, CSS, JavaScript, Ajax, jQuery, [8]
- Uloženie dát na strane klienta pomocou súborov Cookies, úložiska prehliadača alebo tzv. Cache API, ktorá predstavuje systém na ukladanie či získavanie sieťových požiadaviek a ich odpovedí,
- Klient nedokáže čítať dáta zo serveru priamo, na komunikáciu so serverom musí využívať HTTP požiadavky.

Vlastnosti aplikácie na strane servera:

- Kód aplikácie je vyhodnotený webovým serverom a odoslaný klientovi na zobrazenie vo webovom prehliadači,
- Aplikácia prijaté HTTP požiadavky (zo strany klienta) spracuje a odošle späť, napr. aj požiadavky týkajúce sa dát z databázy alebo súborov,
- Vykonáva väčšinu spracovania (namiesto klienta),
- Server renderuje stránky a spracováva vstupy od užívateľa ako napr. overenie užívateľa pri prihlasovaní, uloženie a vrátenie dát, nasmerovanie na iné stránky,
- Nevýhodou je, že užívateľ musí čakať na spracovanie zo strany servera, až potom budú dáta zobrazené, [9]
- Serverová časť by nemala byť prístupná užívateľom z dôvodu jej zabezpečenia,
- Príklady technológií používaných pri vývoji: PHP, ASP, ASP.NET, C#, Ruby on Rails, Python.

4 Programovacie jazyky pre vedecké a astronomické aplikácie

4.1 Príklady programovacích jazykov

Medzi často používané jazyky na vývoj vedeckých a astronomických aplikácií patria jazyky Python, C/C++, IDL, Fortran či JavaScript (knížnica Three.js).

4.1.1 Jazyk Python

Python je skriptovací jazyk s veľmi dobrým využitím pre vedecké aj astronomické aplikácie a dátovú analýzu.

Python predstavuje pre vývojárov v oblasti vedy niekoľko výhod. Syntax jazyka je ľahká pre naučenie aj používanie. Pozitívum predstavujú vedecké knižnice a balíky, viaceré z nich môžu vývojári použiť pre rovnaký účel, preto je vhodné riadiť sa dokumentáciami jednotlivých knižníc a balíkov. Nevýhodou môže byť pomalší čas vyhodnotenia kódu ako u jazykov C, C++ či Fortran.

Vedecké a matematické knižnice, ktoré sú užitočné pre tvorbu astronomických aplikácií:

- SciPy – hlavná vedecká knižnica,
- NumPy – knižnica pre numerické výpočty, poskytuje napr. rýchle operácie na poliach, manipuláciu s tvarom, Fourierové transformácie, lineárnu algebru a základné štatistické operácie, [10]
- SunPy – voľne dostupná knižnica pre solárne dáta,
- Astropy – snaha komunity vytvoriť jednotný balík určený pre astronómiu,
- Matplotlib – 2D a 3D vykresľovacia knižnica. Možné vytvoriť animované aj interaktívne vizualizácie dát,
- IPython – pokročilá interaktívna konzola,
- CosmoPy – kozmologická knižnica,
- a ďalšie. [11]

Na vytvorenie webového užívateľského prostredia už v súčasnosti nemusí byť využívaný len JavaScript. Pre zobrazenie Python aplikácie vo webovom prehliadači existuje niekoľko spôsobov, proces zobrazenia môže však prinášať určité problémy. Pri využití kompilácie do jazyka JavaScript nemusí dôjsť k prekladu všetkých špecifických vedeckých knižníc jazyka Python, preto je v súčasnosti Python vhodnejší pre vývoj desktopového softwaru.

Medzi známe kompilátory do JavaScriptu patrí: Transcrypt, Brython a Skulpt. Opačným prístupom je beh programu priamo vo webovom prehliadači, ktorý

môže byť o niečo pomalší ako prvý prístup. Medzi webové kompilátory patrí: PyPy.js, Batavia a Pyodide. [12]

4.1.2 Jazyk C++

Jazyk C++ sa často používa na vývoj celých softwarových balíkov, operačných systémov, hier, webových prehliadačov, fyzikálnych simulácií založených na skutočnom čase, cloudových systémov a mnoho ďalších.

Taktiež môže byť použitý na implementáciu dôležitých výpočtových algoritmov pre zlepšenie ich výkonnosti. Používa sa aj v kombinácii s jazykom Python, ktorý poskytuje skriptovacie rozhranie a jazyk C++ zabezpečuje časti dôležité pre rýchlu výkonnosť. Nie je však veľmi vhodný pre interaktívne čítanie, vyhodnocovanie či tlač. [13]

4.1.3 Jazyk IDL

IDL predstavuje Interaktívny dátový jazyk, vhodný pre dátovú analýzu či tvorbu interaktívnych aplikácií pre rôzne platformy. Pomocou širokej podpory matematickej analýzy a grafických zobrazovacích techník patrí medzi vhodné nástroje pre astronómov aj vedcov. [14]

Použitie jazyka IDL:

- Vytváranie dát a aplikácií,
- Spracovanie signálu,
- Spracovanie obrázkov – filtrácia šumu, odhalenie anomálií,
- Povrchové a obrysové grafy – zobrazenie ľubovoľného 2D dátového súboru ako povrch,
- Zobrazenie modelov v 3D,
- Animácie – dynamická vizualizácia údajov, vytvorenie sekvencie obrázkov a ich spätné prehranie ako animácia,
- a ďalšie. [14]

4.1.4 Jazyk Fortran

Fortran je aj v dnešnej dobe používaným jazykom v oblasti simulácie fyzikálnych javov. Posledná aktualizácia a stabilná verzia jazyka vyšla v roku 2018. Príkladom použitia je astro-fyzikálne modelovanie hviezd a galaxií, rozsiahla molekulárna dynamika alebo rozsiahle klimatické modely. [15]

4.2 Použité technológie

4.2.1 Jazyk HTML a CSS

HTML (ang. HyperText Markup Language) predstavuje značkovací jazyk pre dokumenty zobrazované vo webovom prehliadači. Základnými pojmami pri tvorbe sú elementy, tagy a atribúty. Samotný dokument predstavuje sémantické rozloženie prvkov webovej stránky alebo aplikácie, ktoré môžu byť pozmenené či doplnené ďalšími technológiami. Pre vytvorenie vizuálnej stránky dokumentu býva jazyk často doplnený o technológiu CSS a pre vytvorenie funkcionality využíva skriptovací jazyk JavaScript.

Spracovanie HTML dokumentu prebieha priamo vo webovom prehliadači pomocou renderovania, ktorý získa HTML dokument z webového serveru alebo lokálneho úložiska. Poslednou aktuálnou verziou je HTML5.

CSS (ang. Cascading Style Sheets) predstavuje Jazyk kaskádových štýlov, používaný na rozvrhnutie jednotlivých prvkov webovej stránky, štýl písma, nastavenie farieb, vyhľadávacích polí a ďalšie úpravy. Jednotlivé štýly je možné písať priamo do HTML dokumentu. V prípade častého používania môžu viesť k redundancii kódu a spôsobiť pomalé spracovanie a načítanie webovej stránky. Odporúčaným prístupom je vytvorenie samostatného .css súboru, ktorý sa následne vloží do hlavičky HTML súboru a môže mať opakované využitie pre viaceré prvky alebo dokumenty, v závislosti od vytvorených štýlov.

Pri použití kaskádových štýlov môže dôjsť ku kolízii pravidiel. Platí, že neskôr umiestnené pravidlo v .css súbore má prednosť. Poradie pravidiel môže byť potlačené špecifickosťou, kedy viac špecifický selektor³ dostane prednosť. Posledným riešením je určenie dôležitosti príkazom *!important*. Podľa zanorenia jednotlivých HTML elementov je možné využiť na jeho tvorbu dedičnosť, ktorá súvisí so spôsobom a hĺbkou zanorenia týchto elementov.

4.2.2 Jazyk JavaScript

Zvoleným jazykom pre tvorbu webovej výučbovej aplikácie je JavaScript, ktorý predstavuje skriptovací jazyk vhodný na tvorbu webových stránok či webových aplikácií. Podpora jazyka zo strany webových prehliadačov uľahčuje prácu vývojárom, jazyk nie je potrebné prekladať pre zobrazenie vo webovom prehliadači.

Skriptovanie prebieha 2 spôsobmi:

1. Skriptovanie na strane klienta – nie je potrebná interakcia ani spracovanie zo strany servera. Skripty sú uložené v HTML dokumente a sú vyhodnotené podľa umiestnenia v dokumente. Rýchlosť a poradie vyhodnotenia skriptov závisí od ich umiestnenia v HTML dokumente. Externé skripty je vhodné pridať na koniec dokumentu, vyhodnotenie prebehne až po zobrazení obsahu dokumentu a v prípade ich robustnosti nebude užívateľom

³Selektor – CSS pravidlo pozostávajúce z HTML elementov, ich tried alebo identifikátorov.

zobrazená ako prvá prázdna obrazovka. Druhou možnosťou je použitie interných skriptov, ktoré sú vyhodnocované priebežne s HTML kódom, v prípade častého používania môžu spôsobiť pomalé načítanie webovej stránky či aplikácie. Príkladom skriptovacích jazykov na strane klienta sú HTML, CSS a JavaScript.

2. Skriptovanie na strane servera – vyžaduje sa činnosť servera na spracovanie skriptov (inštrukcií či programov), ktoré nie sú viditeľné koncovým užívateľom. Výhodou skriptovania na strane servera je napr. tvorba dynamických webových stránok. Príkladom jazykov sú PHP, RubyOnRails, Python, Java, ASP.Net a ďalšie.

JavaScript spĺňa určité pravidlá špecifikácie ECMAScript, ktorá predstavuje štandard pre vytváranie skriptovacích jazykov. Špecifikácia obsahuje základné vlastnosti a spôsob ich implementácie, ktoré by mal skriptovací jazyk poskytovať. Okrem predpísaných vlastností poskytuje JavaScript aj ďalšiu funkcionálnosť, ktorá nie je uvedená v špecifikácii (napr. DOM). [16]

JavaScript má široké využitie. Príklad tvorby:

1. Webové stránky – s interaktívnymi prvkami (napr. tlačítka, carousel, animácie a ďalšie),
2. Webové aplikácie, webový server,
3. Serverové aplikácie – pomocou Node.js sa rozšírilo použitie JavaScriptu aj na vývoj desktopových aplikácií (backend oproti pôvodnému využitiu len pre frontend),
4. Vývoj hier – hry bežiace vo webovom prehliadači,
5. Aplikácie pre Smartwatch⁴ – napr. hodinky Pebble pomocou frameworku Pebble.js alebo aplikácia Alpha WatchBench pre Apple Watch a vývoj iOS aplikácií,
6. Mobilné aplikácie – napr. Airbnb, Facebook, Walmart, Instagram,
7. Lietajúce roboty a drony,
8. Prezentácie ako webové stránky – využívajú sa technológie určené na vývoj webových stránok ako náhrada za PowerPoint alebo Keynote, [17]
9. Čítanie a zápis Cookies⁵,
10. Generovanie HTML kódu.

⁴Smartwatch – prenosné zariadenie v podobe hodiničiek na zápästie, podporujú rôzne aplikácie, meranie zdravotných funkcií človeka alebo NFC platby.

⁵Cookies – predstavujú dáta uložené webovým prehliadačom na počítači užívateľa. Najčastejším použitím je ukladanie osobných údajov o užívateľovi, neskôr použité napr. na ciele návrhy produktov a reklám. Ukladanie údajov môže byť zakázané alebo neskôr odstránené.

4.2.3 Knižnica jQuery

jQuery je pomerne malá multiplatformová knižnica pre JavaScript, podporovaná vo viacerých webových prehliadačoch. Slúži na zjednodušenie kódu písaného pomocou JavaScriptu, teda na jednoduchšie skriptovanie HTML dokumentov zo strany klienta.

Výhodou knižnice je jej široké využitie pre animácie, interaktivitu na webovej stránke, spravovanie udalostí a pridanie efektov ako reakciu na pohyb kurzorom. Knižnica je rozšírená o technológiu Ajax, ktorá môže byť použitá pre získanie dát z databázy či komunikáciu so serverom bez opätovného načítania webovej stránky či aplikácie.

4.2.4 Knižnica Three.js

Three.js je knižnica JavaScriptu určená na tvorbu animovanej a realistickej 3D grafiky. Táto knižnica sa označuje ako API (ang. Application Programming Interface) doplnené o WebGL renderer.

Základné vlastnosti:

- Scéna – obsahuje všetky vytvorené objekty,
- Kamera – sníma objekty na scéne. 2 základné typy kamier sú Perspective a Orthographics, odlišujú sa v zobrazení scény a v možnosti ako s týmito objektmi narábať,
- Renderer – predvolený WebGL renderer, uzatvára trojicu povinných prvkov, bez ktorých by neboli objekty na scéne zobrazené,
- Svetlá – rôzne typy, napr. AmbientLight, DirectionalLight, PointLight, SpotLight. K zvoleným svetlám je možné nastaviť aj vrhanie tieňa,
- Materiály – napr. Lambert, Phong, vloženie rôznych textúr pre realistickejší vzhľad,
- Objekty – napr. Mesh, ktorý sa používa na zobrazenie povrchu objektu alebo Sprite, ktorý predstavuje písmo vložené na scénu a je možné ho nasmerovať tak, aby bolo vždy otočené na kameru,
- Geometria – napr. kocka, guľa, 3D text. Rôzne typy ako Geometry, BoxGeometry či BufferGeometry, ktorá je odporúčaná kvôli jej rýchlejšiemu načítaniu,
- Efekty – napr. animácia, pohyb objektov či scény pomocou kurzora, simulácia fyzikálnych javov,
- Export a import – možnosť vytvoriť modely externe, napr. pomocou softwaru Blender⁶,

⁶Blender – voľne dostupný software, určený na tvorbu 3D počítačovej grafiky a animácií.

- Transparentnosť – možné nastaviť pre rôzne objekty. Nevýhodou je spomalenie transparentných objektov, ich pohybu aj rotácie.

Knižnica Three.js je doplnená o WebGL renderer, ktorý je použitý na renderovanie a zobrazenie 3D scény. Okrem WebGL môže knižnica využiť ďalšie renderery ako Canvas 2D, SVG alebo CSS3D.

WebGL je označovaný ako JavaScript API, ktorý dokáže renderovať 2D a 3D interaktívnu grafiku a je vyhodnocovaný na grafickej jednotke počítača (GPU). Väčšina webových prehliadačov ho podporuje, preto je vhodným nástrojom pre tvorbu či zobrazenie 3D grafiky v prehliadači, bez potreby dodatočných programov (plug-in) ako sú Flash alebo Java. WebGL ako doplnok Three.js knižnice nie je prioritne určený na tvorbu 3D grafiky, preto stačí ovládať túto knižnicu spolu s jazykom JavaScript.

5 Fyzika vo vesmíre

Pomocou fyzikálnych javov a veličín je možné popísať základný pohyb telies vo vesmíre. Na ich pohyb majú vplyv rôzne faktory ako sú gravitačná sila a ťah, hmotnosť telesa, akcelerácia, rýchlosť pohybu a ďalšie.

5.1 Retrográdny a prirodzený pohyb

Vesmírne teleso sa pohybuje retrográdne po orbite vtedy, keď je smer jeho pohybu opačný ako pohyb primárneho telesa, ktoré obieha. Vo vesmíre sa vyskytuje niekoľko mesiacov, ktoré majú retrográdny pohyb voči planéte, ktorú obiehajú. Pre planéty Slnecnej sústavy platí, že sa všetky pohybujú v rovnakom smere.

Pri pozorovaní pohybu planét Slnecnej sústavy sa môže zdať, že niektorá planéta sa pohybuje retrográdne voči inej planéte. Tento jav je spôsobený tým, že planéta s menšou obežnou dráhou sa pohybuje rýchlejšie ako planéta s väčšou obežnou dráhou a v momente, kedy sa jedna planéta dostane pred druhú, ich pohyb po orbite sa zdá byť v opačnom smere. [18]

5.2 Retrográdna a prirodzená rotácia

Retrográdna rotácia predstavuje rotáciu vesmírneho telesa v opačnom smere, ktorým sa pohybuje po obežnej dráhe. Príkladom sú planéty Venuša, Urán alebo Pluto, ktoré bolo oficiálne vyradené spomedzi planét Slnecnej sústavy už v roku 2006. O dva roky neskôr bolo Pluto zaradené medzi trpasličie planéty. [18]

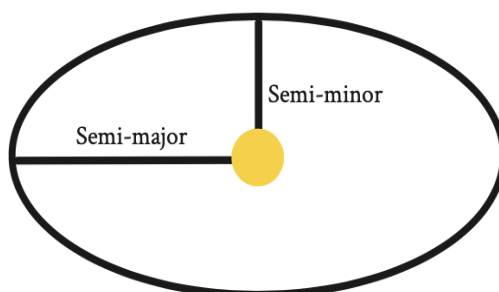
5.3 Gravitačný ťah

Gravitačný ťah alebo gravitácia je neviditeľná sila, ktorou vesmírne teleso priťahuje iné vesmírne telesá do svojho stredu a udržuje planéty Slnecnej sústavy na

obežných dráhach okolo Slnka alebo iného vesmírneho telesa. Sila gravitačného ťahu závisí aj od vzdialenosti, s narastajúcou diaľkou medzi dvomi telesami sa sila ťahu znižuje.

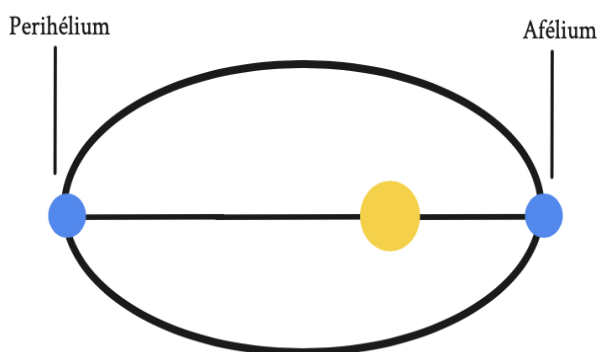
5.4 Hlavná a vedľajšia poloos

Vesmírne telesá rotujú po obežných dráhach tvaru elipsy, rotácia okolo vesmírneho telesa teda neprebíha v dokonalom kruhu. Hlavná poloos orbity telesa sa označuje *semi-major* a je najdlhšou vzdialenosťou medzi stredom obiehajúceho telesa po elipse a stredom elipsy. Opakom je vedľajšia poloos nazývaná *semi-minor*, ktorá predstavuje najkratšiu vzdialenosť medzi stredom obiehajúceho telesa po elipse a stredom elipsy.



Obr. 4: Semi-major a semi-minor os

Podľa pozície vesmírneho telesa je možné určiť perihélium a afélium. Perihélium, tiež nazývané ako *perihelion*, je najbližší bod na obežnej dráhe vesmírneho telesa, v ktorom sa teleso nachádza pri obehú okolo iného telesa. Afélium, tiež nazývané *aphelion*, je naopak najvzdialenejší bod. Keď nastane afélium, pohyb Zeme po obežnej dráhe okolo Slnka sa spomalí. [19]



Obr. 5: Perihélium a afélium

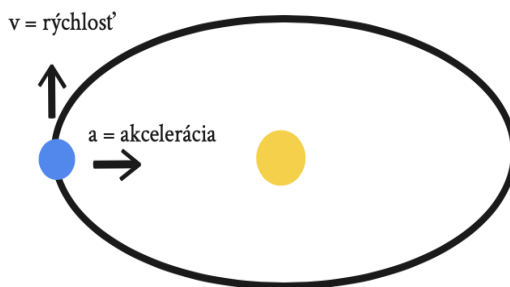
5.5 Excentricita

Excentricita je bezrozmerný parameter, ktorý určuje tvar orbity vesmírneho telesa. Hodnota 0 znamená, že tvar orbity je kruhový. Hodnota medzi 0-1 určuje eliptickú dráhu, vyššia hodnota znamená dlhšiu *semi-major* os a kratšiu *semi-minor* os. Od tvaru orbity závisí napr. rýchlosť pohybu vesmírneho telesa po orbite. [20]

5.6 Rýchlosť a akcelerácia

Rýchlosť pohybu vesmírneho telesa môže byť konštantná, teleso sa vtedy pohybuje rovnakou rýchlosťou po celej dĺžke obežnej dráhy, ktorá má tvar kruhu. Keďže dráhy vesmírnych telies majú eliptický tvar, mení sa aj rýchlosť pohybu telies. V momente, kedy dôjde k zmene rýchlosti, nastáva akcelerácia.

Akcelerácia je vektorová veličina, ktorá predstavuje zmenu rýchlosti pohybu telesa v čase. Termín sa zvyčajne používa pre zrýchlenie pohybu a pre spomalenie pohybu je zavedený termín *decelerácia*. Spomalenie pohybu vesmírneho telesa nastáva v momente, keď je teleso na vzdialenejšej časti obežnej dráhy. [21]



Obr. 6: Pohyb vesmírneho telesa a akcelerácia

5.7 Umelý kozmický objekt

5.7.1 Typy orbít

Existuje niekoľko typov orbít, na ktorých môže obiehať kozmický objekt okolo planéty. Výber orbity závisí napr. od toho, na čo je kozmický objekt alebo satelit určený:

- Geostacionárna orbita – satelit sa pohybuje okolo planéty v smere jej rotácie a rovnakou rýchlosťou. Satelit obiehajúci planétu Zem sa preto javí byť v rovnakej pozícii. V praxi sa takto umiestňujú telekomunikačné satelity, [22]
- Nízka obežná dráha – dráha je vo vzdialenosti menej ako 2000km a jej rovina sa môže meniť. Rýchlosť pohybu satelitu je vyššia ako pohyb planéty po orbite, preto dokáže satelit obehnúť planétu niekoľkokrát za deň. Pri

porovnaní s planétou Zem ju dokáže obehnúť až 16-krát za deň. V praxi sa nízka obežná dráha používa pre fotografie vysokého rozlíšenia alebo pre medzinárodnú vesmírnu stanicu ISS, [23]

- Stredná obežná dráha – v praxi sa využíva pre navigačné satelity,
- Polárna obežná dráha – dráha prechádza ponad Severný a Južný pól, niekedy aj s miernym odchýlením, [24]
- Synchronná dráha so Slnkom – špeciálny prípad polárnej obežnej dráhy. Satelit má na dráhe fixnú pozíciu vzhľadom k Slnku, preto je umiestnený na rovnakom mieste nad planétou každý deň. V praxi sa využíva na fotografie oblastí planéty, ktoré môžu neskôr slúžiť na porovnanie vývoja oblasti, zmenu počasia, zachytenie lesných požiarov, odlesňovanie, záplavy alebo výbuchy sopiek [24]
- Geostacionárna prenosová dráha – dráha satelitu sa môže meniť a s využitím fyzikálnych zákonov je možné zmeniť napr. obežnú dráhu okolo Zeme na dráhu obiehajúcu okolo planéty Mars. Pre zmenu dráhy sa môže využiť známy Hohmann prenosová dráha.

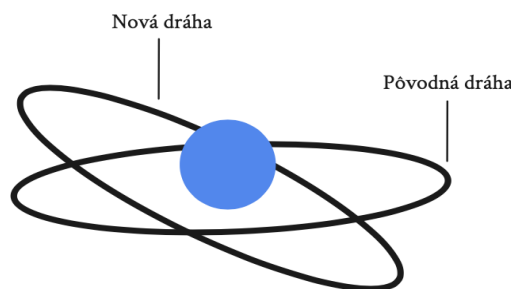
5.7.2 Rýchlosť pohybu a úniková rýchlosť

Rýchlosť pohybu predstavuje rotáciu vesmírneho telesa alebo umelého objektu po obežnej dráhe okolo Slnka alebo inej planéty. Rozdielom je úniková rýchlosť, ktorá predstavuje minimálnu rýchlosť na únik z gravitačného vplyvu vesmírneho telesa, okolo ktorého sa teleso pohybuje. Úniková rýchlosť je preto dôležitá pri prechode z obežnej dráhy telesa na obežnú dráhu iného telesa.

5.7.3 Zmena naklonenia dráhy

Zmena naklonenia obežnej dráhy vyžaduje zmenu vektora pre rýchlosť. Ak je cieľom zachovať veľkosť dráhy a zmeniť len naklonenie, tento manéver sa považuje za jednoduchý. Problémom je vysoká spotreba paliva pre zmenu rýchlosti pohybu. Na minimalizovanie spotreby paliva je vhodné meniť naklonenie dráhy až v bode, keď je pohyb telesa po dráhe najpomalší. Tento jav nastane, keď sa teleso nachádza najďalej od planéty, teda nastane afélium. [25]

V niektorých prípadoch môže byť výhodnejším riešením zmena dráhy na vyššiu dráhu, kde sa upraví jej naklonenie a následne sa teleso vráti na pôvodnú orbitu. Ďalším riešením je využitie až 3 zapálení, ktoré spôsobia potrebnú zmenu v rýchlosti pohybu a ušetria palivo. Nevýhodou je dlhší čas potrebný na vykonanie manévru. [25]



Obr. 7: Zmena naklonenia dráhy pre vesmírne teleso

5.7.4 Zmena obežnej dráhy

Zmena obežnej dráhy na inú dráhu vyžaduje zmenu vektora rýchlosti v jeho veľkosti alebo smere. Manéver sa vykonáva v bode, kde sa pretína pôvodná dráha kozmického objektu s novou dráhou. V prípade, že sa dráhy v žiadnom mieste nepretínajú, je nutné využiť prenosovú dráhu, ktorá pretína počiatočnú aj konečnú dráhu. Príkladom je využitie geostacionárnej prenosovej dráhy s využitím Hohmannovho prenosu. [26]

Prenosová dráha má tvar elipsy a predstavuje dotyčnicu pôvodnej a konečnej dráhy. Hohmannov prenos patrí medzi najúspornejšie prenosy medzi dvoma kruhovými dráhami. Dôležitá je zmena rýchlosti vo vybranom smere pohybu. Pri prechode z menšej obežnej dráhy na väčšiu dráhu sa použije zmena rýchlosti v smere pohybu. Pre opačný prechod je potrebná zmena rýchlosti proti smeru pohybu telesa. [26]

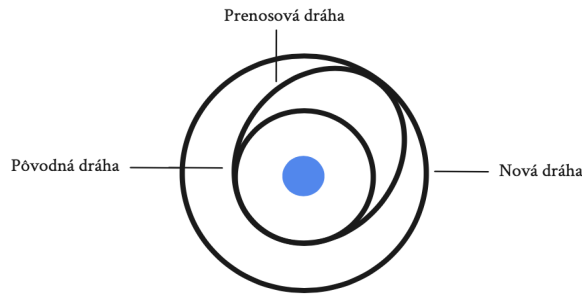
V momente, keď sa kozmické teleso dostane z prenosovej dráhy na cieľovú dráhu, je potrebná opätovná zmena rýchlosti pohybu tak, aby sa tvar dráhy kozmického telesa zmenil z eliptickej dráhy na tvar cieľovej dráhy.

Vysvetlenie premenných:

- Premenná v – rýchlosť pohybu obežného telesa po orbite,
- Premenná G – gravitačná konštanta, ktorej hodnota je $6.67 * \frac{1}{10^{11}} \frac{m^2}{kg^2}$,
- Premenná M – hmotnosť centrálného (obiehaného) telesa,
- Premenná r – vzdialenosť obežného telesa od centrálného telesa, ktoré obieha,
- Premenná a – veľkosť hlavnej poloosi (semi-major) prenosovej dráhy.

Definícia 1 (Výpočet prenosovej dráhy)

$$v^2 = G * M * \left(\frac{2}{r} - \frac{1}{a} \right)$$



Obr. 8: Zmena obežnej dráhy pomocou prenosovej dráhy

6 Programátorská dokumentácia

Programátorská dokumentácia obsahuje a stručne popisuje základné triedy aplikácie a vybrané funkcie z každej triedy. Aplikácia bola prioritne testovaná vo webovom prehliadači Google Chrome.

6.1 Triedy

6.1.1 Triedy `InitScene`, `PlanetScene`, `ModelScene`

Trieda `InitScene`

Trieda obsahuje funkcie na inicializáciu potrebných prvkov, ktoré slúžia na zobrazenie scény a ďalších objektov. Scéna `THREE.Scene` slúži na umiestnenie všetkých vytvorených objektov, renderer `THREE.WebGLRenderer` slúži na prekresľovanie grafických zmien a 3D kamera `THREE.PerspectiveCamera` slúži na snímanie scény a zobrazenie jej obsahu. Ďalšou možnosťou je použitie kamery `THREE.OrthographicCamera`, ktorá je vhodná na snímanie 2D scény. Funkcie:

- `initRenderer` – inicializácia renderera `THREE.WebGLRenderer`, pomocou ktorého sa prekresľuje obrazovka (približne 60x za sekundu),
- `initCamera` – inicializácia kamery, ktorá sníma celú scénu aj s vytvorenými objektami,
- `resizeBackground` – funkcia vytvorí udalosť, ktorá prispôsobí veľkosť scény aktuálnej veľkosti okna webového prehliadača,
- `setPointLightOnScene` – inicializácia svetla `THREE.PointLight`, ktoré dokáže vrhať tieň a napodobniť slnečné žiarenie,
- `setPointLightOnSun` – inicializácia svetla `THREE.PointLight`, ktoré presvetľuje objekt Slnka,
- `traverseSceneToCastShadows` – funkcia prejde všetky objekty (potomkov) na scéne a priradí im vlastnosť prijímania a vrhania tieňov.

Trieda PlanetScene

Trieda obsahuje samostatnú scénu pre zobrazenie jedného interaktívneho modelu planéty, ktorý sa dá ovládať pomocou kurzora myši. Na rovnomerné osvetlenie celého povrchu planéty je použité špeciálne svetlo *THREE.AmbientLight*, ktoré rovnomerne presvetlí všetky objekty na scéne, bez možnosti vrhať tieň. Trieda ďalej využíva knižnicu *OrbitControls.js*, pomocou ktorej je možné ovládanie planéty. Funkcie:

- `initInteractivePlanetScene` – nastavenie kamery a WebGL renderera, ktorý má na starosti prerisovanie obrazovky,
- `createAllPlanets` – vytvorené samostatné objekty planét,
- `createPlanetMesh` – vytvorený povrch planét,
- `setOrbitConstrols` – použitá knižnica *OrbitsControls.js* z hlavnej knižnice *three.js*, pomocou ktorej je možné ovládať modely planét kurzorom myši,
- `showInteractivePlanet` – zobrazenie konkrétneho modelu planéty pre vybranú stránku z rozcestníka planét,
- `animate` – prerisovanie grafických zmien, ktoré sú ovládané pomocou *OrbitControls* a WebGL renderera.

Trieda ModelScene

Trieda obsahuje samostatnú (hlavnú) scénu pre zobrazenie celého modelu Slnecnej sústavy. Je doplnená o druhú scénu s vlastnou kamerou, ktorá sníma statické pozadie v podobe JPG obrázku. Trieda je doplnená o udalosti prepojené s kurzorom myši, ovládanie pohybu a priblíženia objektov. Funkcie:

- `setStaticBackground` – vytvorenie statického pozadia (JPG obrázok), ktoré nebude ovplyvnené pohybom scény, jej priblížením a ďalšími akciami,
- `createBgScene` – pre statické pozadie je potrebné vytvoriť samostatnú scénu, ktorá nebude reagovať na akcie prevedené na scéne s modelom Slnecnej sústavy,
- `createBgCamera` – pre statické pozadie je potrebné vytvoriť samostatnú kameru, ktorá bude snímať pozadie z nezmenenej pozície,
- `moveSceneOnPressedArrow` – pohyb hlavnej scény s modelom pomocou šípok na klávesnici a vrátenie modelu do stredu obrazovky pomocou klávesy *Escape*,
- `moveSceneToOriginalPosition` – vrátenie scény s modelom do stredu obrazovky pomocou klávesy *Escape*, funkcia je aktivovaná tlačidlom na vycentrovanie modelu,

- `mousemoveEvent` – funkcia pre zachytenie stlačenia kurzora až po jeho pustenie. Slúži na ťahanie a pohyb celého modelu,
- `addEventListenerFunctions` – hromadné pridanie udalostí pre kurzor. Typy udalostí: *keydown*, *click*, *mousedown*, *mousemove* a *mouseup*,
- `rotateSceneObjects` – dôležitá funkcia, ktorá má na starosti pohyb objektov na scéne. Hodnota z posúvača, ktorá slúži na zmenu dráhy umelého objektu, je ďalej predaná potrebným funkciám,
- `activateCameraToObjectButton` – aktívne tlačidlo pre kameru dočasne zruší pridávanie umelých objektov a pohyb scény pomocou udalosti *mousemove*,
- `findClickedPlanetForCamera` – nastavenie parametrov kamery tak, aby pohybom nasledovala planétu, ktorá je vybraná kliknutím na ňu alebo jej orbitu,
- `animate` – prekresľovanie grafických zmien, ktoré sú ovládané pomocou WebGL renderera. Je volaná aj animácia pre triedu `RayCaster`, ktorá slúži na rozpoznanie kliknutia na objekt,
- `startRendererOnMouseEvents` – pridanie WebGL renderera k udalostiam: *keydown*, *click*, *mousedown*, *mousemove* a *mouseup*. Funkcia je využitá počas zastavenia pohybu planét a mesiacov, kedy bude prekresľovanie volané len pri pohybe a iných akciách prevedených pomocou kurzora. Funkcia je vytvorená a volaná z dôvodu šetrenia záťaže webového prehliadača.

6.1.2 Trieda `JSONManager`

Trieda slúži na čítanie dát z JSON súboru a vytvorenie dvoch vlákien: vlákno pre čítanie dát o planétach a vlákno pre čítanie dát o mesiacoch. Súbor obsahuje číselné údaje, ktoré sú použité pri vytváraní orbít objektov a pre nastavenie pozície planét, mesiacov a umelého kozmického objektu.

Ďalšie využitie súboru je pre zobrazenie číselných údajov v tabuľke s fyzikálnymi vlastnosťami vybraného objektu, ktorá sa zobrazí po kliknutí na objekt scény alebo jeho orbitu. Súbor obsahuje upravenú mierku pre veľkosť jednotlivých objektov a orbít. Funkcie:

- `readDataJSON` – asynchrónna funkcia, ktorá prečíta potrebné dáta z JSON súboru a vráti ich v podobe príslubu,
- `addAllPlanetDataJSON` – vytvorenie vlákna do JSON súboru, ktoré prečíta všetky dáta o planétach. Príslub je vytvorený len jedenkrát a zdedený v ďalších triedach,
- `addAllMoonDataJSON` – vytvorenie vlákna do JSON súboru, ktoré prečíta všetky dáta o uložených mesiacoch planét. Príslub je vytvorený len jedenkrát a zdedený v ďalších triedach.

6.1.3 Trieda SidebarManager

Trieda obsahuje funkcionalitu na ovládanie pravej bočnej tabuľky, ktorá slúži na ovládanie viditeľnosti názvov a orbít objektov, zobrazenie všetkých mesiacov či zobrazenie mesiacov pre konkrétnu planétu. Ďalej je možné ovládať zastavenie či spustenie pohybu objektov. Počas zastavenia pohybu nie je obmedzená ďalšia funkcionalita aplikácie. Funkcie:

- `showHideSidebarToRight` – zobrazenie alebo skrytie pravej bočnej tabuľky,
- `hideSidebarToLeft` – skrytie ľavej tabuľky s fyzikálnymi údajmi o vybranom objekte, prebieha pomocou kliknutia na ľubovoľné miesto v tabuľke. Jej skrytie zároveň zruší farebné označenie vybraného objektu a jeho orbity,
- `startNewSimulation` – funkcia obsahuje správu na potvrdenie užívateľom. Správa je zobrazená v prípade, že aktuálne okno webového prehliadača obsahuje spustenú simuláciu modelu Slnecnej sústavy a užívateľ vyberie položku z Menu aplikácie na jej opätovné otvorenie. Správa obsahuje upozornenie, že po potvrdení budú nastavené vlastnosti modelu vymazané,
- `openInNewTab` – otvorenie položiek Menu *Planéty* a *O aplikácii* v novej záložke webového prehliadača. Táto možnosť je využitá len v okne, v ktorom je spustená simulácia modelu Slnecnej sústavy. Funkcia je využitá za účelom zachovania nastavených vlastností simulácie od užívateľa,
- `pauseApplication` – zastavenie pohybu vesmírnych objektov a funkcionality tlačidiel pre kozmický objekt a priblíženie kamery na vybranú planétu. Ďalej je uložený čas zastavenia pomocou `Date.now()`, ktorý je neskôr použitý na výpočet správneho pokračovania pohybu vesmírnych objektov,
- `pauseApplicationOnInactiveWindow` – zastavenie pohybu vesmírnych objektov v prípade, že simulácia je otvorená v niektorej záložke webového prehliadača ale nie je aktuálne prezeranou záložkou,
- `playApplication` – spustenie pohybu vesmírnych objektov a funkcionality pre tlačidlá, ktoré boli zastavené pomocou funkcie `pauseApplication`. Ďalej je uložený čas spustenia pomocou `Date.now()`, ktorý je neskôr použitý na výpočet správneho pokračovania pohybu vesmírnych objektov,
- `playApplicationOnActiveWindow` – spustenie pohybu vesmírnych objektov v prípade, že je záložka so simuláciou aktuálne prezeranou záložkou,
- `confirmButtonBehavior` – volanie funkcií pre ovládanie viditeľnosti názvov a orbít objektov, zobrazenie všetkých mesiacov a zobrazenie mesiacov pre konkrétnu planétu,
- `disableConfirmButtonForMoment` – tlačidlo *Potvrdiť* je po stlačení pozastavené na pár sekúnd, počas ktorých sú prevedené všetky voľby z pravej tabuľky,

- `showHideMoonsOfSinglePlanet` – zobrazenie všetkých mesiacov pre konkrétnu planétu. Názvy a orbity mesiacov sú zobrazené podľa voľby ostatných možností z tabuľky. V zoznamoch pre výber názvov a orbít mesiacov sú sprístupnené len tie možnosti, ktoré patria k vybranej planéte,
- `disableDropDownOptionsJupiter` – pomocná funkcia pre `showHideMoonsOfSinglePlanet`, ktorá v zozname pre výber názvov a orbít mesiacov sprístupní len tie možnosti, ktoré patria k vybranej planéte,
- `addEnabledObjectsForSaturn` – funkcia skontroluje výber možností mien mesiacov a ich orbít pre vybranú planétu a podľa toho ich následne zobrazí.

6.1.4 Trieda `RayCaster`

Trieda používa `THREE.Raycaster` na detekciu 3D objektu, ktorý bol vybraný pomocou kurzora myši. Celková funkcionálnosť je popísaná vo funkciách triedy:

- `onMouseMove` – dôležitá funkcia, ktorá rozpozna kliknutie na objekt scény. Funkcia má na starosti rozpoznanie objektu, farebné odlíšenie objektu aj jeho dráhy a skrytie či zobrazenie tabuľky s informáciami o vybranej planéte alebo mesiaci,
- `disableRaycasterThroughOverlayObjects` – dočasné vypnutie funkčnosti `Raycastera`, ktorý dokáže rozpoznať objekty scény aj v prípade ich prekrytia tabuľkou, tlačidlom či posúvačom,
- `getPhysicalValuesOfClickedObjectFromJSON` – funkcia pracuje s príslubom z JSON súboru a doplní potrebné dáta o vybranom objekte do tabuľky s fyzikálnymi vlastnosťami. Podľa označeného objektu (alebo jeho orbity) pracuje s príslubom, ktorý obsahuje dáta o planéte alebo mesiaci,
- `animate` – zmeny vo vyššie popísaných funkciách je potrebné kontrolovať pri každom prekreslení obrazovky a v prípade potreby previesť. Funkcia je volaná spolu s WebGL rendererom, ktorý riadi prekresľovanie pre model scény.

6.1.5 Trieda `InitPlanets`

Trieda obsahuje všeobecné funkcie na vytvorenie samostatných objektov, povrchov, názvov a ich pridanie na scénu. Funkcie sú využité v potomkoch triedy. Funkcie:

- `createPlanetObject` – vytvorenie objektov (planét aj mesiacov) pomocou `THREE.SphereBufferGeometry`,
- `setNewMesh` – vytvorenie povrchu objektov (planét a Mesiaca Zeme) pomocou `THREE.MeshPhongMaterial`,

- `createMesh` – vytvorenie *THREE.Mesh* s využitím nového objektu a povrchu z predošlých funkcií. *Mesh* slúži na ďalšie ovládanie 3D objektu na scéne,
- `createTextGeometry` – vytvorenie názvov pre planéty a mesiace na scéne pomocou *THREE.TextGeometry*. Názvy sú vytvorené v 3 jazykoch: český, slovenský a anglický. Pre mesiace je preložené len meno mesiaca pre planétu Zem,
- `traverseSceneToFindPlanetNames` – funkcia slúži na prehľadanie objektov (potomkov) scény a nastavenie ich viditeľnosti. Vo funkcii *scene.traverse(...)* nie je možné pridávať a odoberať objekty scény, preto je využitá pre nastavenie viditeľnosti názvov na scéne.

6.1.6 Triedy Sun, Moon, Planet

Trieda Sun

Trieda obsahuje vytvorenie objektu Slnka a nastavenie svetiel *THREE.PointLight* na jeho pozíciu. Svetlo osvetľuje všetky objekty scény s možnosťou vrhania tieňov a napodobňuje tak prirodzené slnečné žiarenie. Funkcie:

- `setLightsToSunPosition` – nastavenie svetiel scény na pozíciu Slnka. Svetlá kopírujú jeho pozíciu aj pri pohybe scény.

Trieda Moon

Trieda obsahuje vytvorenie objektov mesiacov a nastavenie ich pozícií, ktoré závisia od číselných údajov z JSON súboru a od pozície planéty, ktorú obiehajú.

Objekty mesiacov majú malú veľkosť, preto je ostatným mesiacom okrem Mesiaca pre planétu Zem priradený rovnaký sivý povrch. Mesiac Zeme má štruktúrovaný povrch ako ostatné planéty. Funkcie:

- `createMoons` – vytvorenie objektov mesiacov,
- `positionMoonToOrbit` – dôležitá funkcia, ktorá má na starosti nastavenie pozície pre mesiace a ich názvy na scéne. Pozícia je nastavená na základe číselných údajov z JSON súboru a pozície orbity pre mesiac,
- `traverseSceneToFindMoons` – prehľadávanie scény pomocou funkcie scény *scene.traverse(...)* na skrytie alebo zobrazenie mien mesiacov. Táto funkcia môže byť užitočnejšia ako použitie funkcie *for(...)*, ktorej vykonanie zaberie dlhší čas a prejaví sa na grafickej stránke aplikácie.

Trieda Planet

Trieda obsahuje vytvorenie objektov planét so štruktúrovaným povrchom a nastavenie ich pozícií, ktoré sú neustále prekresľované pomocou WebGL renderera. Ďalej obsahuje nastavenia uhla rotácie, vytvorenie a pridanie mien planét na scénu v troch jazykoch: český, slovenský a anglický. Funkcie:

- `addNamesToPlanetObject` – vytvorenie názvov planét v troch jazykoch. Názvy sú pridané na scénu spolu s objektami a kopírujú ich meniace sa pozície,
- `setRotationAngleForSinglePlanet` – nastavenie uhla naklonenia pomocou vektora *THREE.Vector3*. Nastavenie prebehne čítaním dát z JSON súboru,
- `rotateAllPlanets` – funkcia zobrazí mená planét na scéne podľa aktuálneho jazyka aplikácie a nastaví pozíciu objektov planét a ich mien pomocou volanej funkcie *positionPlanetOnOrbit*,
- `positionPlanetOnOrbit` – dôležitá funkcia, ktorá má na starosti nastavenie správnych pozícií pre planéty a ich mená. Funkcia ďalej určuje rýchlosť pohybu objektov. Hodnoty sú nastavené čítaním dát z príslušného z JSON súboru.

```

1 positionPlanetOnOrbit(planetMesh, planetName, planetNameOnScene,
   time) {
2   var dataOfCurrentPlanetJSON = (this.getPlanetDataJSON())[0];
3   var orbitalSpeed = 0;
4
5   // Promise from JSON
6   dataOfCurrentPlanetJSON.then(function(result) {
7     orbitalSpeed = result[planetName]["orbitalSpeed"] / 3;
8     planetMesh.position.x = result[planetName]["c"] +
9       (result[planetName]["a"] *
10        result[planetName]["scaleFactor"] *
11        Math.cos(orbitalSpeed * 0.0001 * time));
12    planetMesh.position.z = -1 * (result[planetName]["b"] *
13      result[planetName]["scaleFactor"] *
14      Math.sin(orbitalSpeed * 0.0001 * time));
15
16    if (planetNameOnScene !== undefined) {
17      planetNameOnScene.position.x = planetMesh.position.x +
18        result[planetName]["planetSize"] + 1;
19      planetNameOnScene.position.z = planetMesh.position.z;
20    }
21  });
22 }

```

Zdrojový kód 1: Výpočet pozície planét pomocou JSON súboru

6.1.7 Trieda Orbits

Trieda obsahuje vytvorenie orbít pomocou *THREE.EllipseCurve*. Jednotlivé parametre orbity sú nastavené pomocou číselných údajov z JSON súboru. Orbity planét sú umiestnené podľa stredu Slnecnej sústavy a orbity mesiacov sú umiestnené podľa planéty, ktorú ich mesiace obiehajú. Funkcie:

- `createCurveForOrbit` – funkcia pracuje s príslubom z JSON súboru a číselné údaje používa pre vytvorenie orbít planét a mesiacov. Funkcia pracuje asynchrónne, aby po vyhodnotení dát z príslubu mohla k týmto dátam pristupovať,
- `setParamsForOrbitCurve` – pomocná funkcia pre funkciu `createCurveForOrbit`, ktorá nastavuje parametre pre orbitu,
- `positionSingleMoonOrbit` – nastavenie pozície orbity podľa planéty, ktorú mesiac obieha.

6.1.8 Trieda `CosmicObject`

Trieda obsahuje vytvorenie umelého kozmického objektu a jeho orbity, po ktorej sa pohybuje čítaním jej bodov. Medzi základné funkcie patrí aktivácia tlačidla pre zobrazenie kozmického objektu, detekcia vybranej planéty, dočasné skrytie jej mesiacov a nastavenie pozície kozmického objektu.

Pri aktivácii tlačidla pre umelý kozmický objekt sa zobrazí posúvač, ktorý slúži na zmenu rýchlosti pohybu objektu a pôsobenia sily na objekt. Zmeny sa prejavia v tvare orbity a pohybe objektu, ktorý sa pohybuje pomalšie s väčšou vzdialenosťou od planéty. Funkcie:

- `createCosmicObject` – vytvorenie objektu pomocou `THREE.ConeGeometry` a `THREE.Mesh`,
- `activateCosmicObjectButton` – aktivácia tlačidla pre pridanie kozmického objektu k vybranej planéte. Funkcia pozastaví činnosť tlačidla pre zaostrenie kamery na planétu a farebne zvýrazní aktívne tlačidlo. V prípade odobratia kozmického objektu budú zobrazené mesiace planéty, ktoré boli skryté z dôvodu prehľadnosti modelu,
- `findClickedPlanet` – funkcia rozpozna označenú planétu, pri ktorej bude zobrazený umelý kozmický objekt. Pre planétu budú skryté mesiace z dôvodu prehľadnosti modelu,
- `moonsVisibilityOfSelectedPlanet` – nastavenie viditeľnosti mesiacov konkrétnej planéty,
- `addCosmicObjectToOrbit` – funkcia, ktorá pridá kozmický objekt na orbitu v prípade, že je tlačidlo pre kozmický objekt aktívne a užívateľ vybral kurzorom myši niektorú z planét (nie mesiace),
- `positionCosmicObject` – dôležitá funkcia, ktorá nastaví pozíciu kozmického objektu na jeho orbite okolo vybranej planéty. Objekt sa pohybuje po orbite čítaním jej bodov. Argumentmi ďalších volaných funkcií sú hodnoty z príslubu, ktoré je možné použiť až po jeho vyhodnotení. Pre získanie hodnôt je potrebná asynchrónna funkcia `await`,

- `changeRotationSpeedOfObject` – dôležitá funkcia, ktorá zistí pozíciu kozmického objektu a podľa jeho vzdialenosti od planéty určí rýchlosť jeho pohybu. Rýchlosť pohybu závisí od toho či sa objekt nachádza v najviac vzdialenom bode od planéty (perihélium), alebo v najbližšom bode (afélium). Na zistenie vzdialenosti sú použité dva trojrozmerné vektory, ktoré sú vhodné na zistenie vzdialenosti medzi ľubovoľnými dvoma bodmi,
- `changeShapeOfObjectOrbit` – dôležitá funkcia, ktorá mení tvar orbity pôsobením zmien v rýchlosti a sile.

```

1 positionCosmicObject(selectedPlanet, cosmicObject, orbit, force) {
2   var dataOfCurrentPlanetJSON = (this.getPlanetData())[0];
3   var changeX, changeY, orbitPoint = 0;
4
5   (async() => {
6     var valueX = this.getCurrentForceForAxisX();
7     var valueZ = this.getCurrentForceForAxisZ();
8     var time = this.getCosmicObjectMovingTime();
9     var isSpeedChanged = this.getIsSpeedChanged();
10
11    var promiseValue = dataOfCurrentPlanetJSON.then(function(
12    result) {
13      changeX = 0.5 * valueX *
14        result[selectedPlanet]["cosmicObjectDistanceX"];
15      changeY = 0.5 * valueZ *
16        result[selectedPlanet]["cosmicObjectDistanceZ"];
17
18      orbit.scale.set(changeX, changeY, 1);
19      orbitPoint = new THREE.Vector3(
20        Math.cos(changeX * Math.PI * (-0.00001) * time *
21          result[selectedPlanet]["cosmicObjectSpeed"]), 0,
22        Math.sin(changeY * Math.PI * (-0.00001) * time *
23          result[selectedPlanet]["cosmicObjectSpeed"]));
24      cosmicObject.position.set(orbit.position.x + orbitPoint.x,
25        orbit.position.y, orbit.position.z + orbitPoint.z);
26    });
27    // await - to read data from Promise.then()
28    await promiseValue;
29    this.changeShapeOfObjectOrbit(isSpeedChanged, selectedPlanet,
30      orbitPoint.x, force, orbit);
31    this.changeRotationSpeedOfObject(orbit, cosmicObject, force);
32  }) ();
33 }

```

Zdrojový kód 2: Nastavenie pozície pre kozmický objekt

6.2 Ovládanie prekreslenia obrazovky

Hlavná funkcia aplikácie, ktorá má na starosti prekresľovanie obrazovky pomocou *THREE.WebGLRenderer* a ovláda správne nastavenie času pri zastavení a spustení aplikácie. Čas je po spustení aplikácie vypočítaný tak, aby všetky objekty pokračovali z rovnakého miesta ako pred zastavením ich pohybu. Pre určenie času je použitá hodnota *Date.now()*. Pre aktualizáciu animácie pred jej prekreslením je potrebné zavolať funkciu *requestAnimationFrame()*.

Počas zastavenia pohybu vesmírnych objektov je obmedzený preklad stránky, kedy mená objektov budú preložené až po spustení aplikácie.

```
1 var i = 0;
2 function animate() {
3     modelScene.animate();
4     if (document.getElementById("pauseButton").disabled == false) {
5         modelScene.rotateSceneObjects(
6             i + modelScene.sidebarManager.getPlayTime() -
7             modelScene.sidebarManager.getPlayPauseTimeDifference());
8         i += 50;
9         modelScene.startRenderer(renderer, scene, camera);
10    } else {
11        modelScene.startRendererOnMouseEvents(renderer, scene,
12            camera);
13    }
14    requestAnimationFrame(animate);
15 }
16 animate();
```

Zdrojový kód 3: Ovládanie prekreslenia obrazovky – výpočet času

6.3 Nasadenie aplikácie na server

Pre webovú aplikáciu bol vytvorený virtuálny server s operačným systémom Linux Ubuntu a vo verzii 20.04 LTS, tento server bol poskytnutý školou.

Pre nasadenie NodeJS aplikácie je najskôr potrebné naprogramovať serverové prostredie, to je definované v súbore *serverhttps.js*. Sú v ňom vytvorené 2 servery: HTTP a HTTPS. HTTP server pracuje na porte č.80 a obsahuje len presmerovanie 301 na HTTPS server s portom č.443, ktorý poskytuje užívateľovi zabezpečenú komunikáciu, preto je preferovaný. Aby mohol HTTPS server fungovať korektne, vyžaduje certifikáty, ktoré sú overené dôveryhodnou autoritou, v tomto prípade bola využitá služba *Let's Encrypt* [27]. S pomocou nástroja *certbot* v Ubuntu boli vygenerované certifikáty, overené službou *Let's Encrypt* a následne využité pre HTTPS server. Platnosť certifikátov je 90 dní, voliteľne môže byť nastavená automatická obnova týchto certifikátov a to na serverovom prostredí.

Aby bolo možné takto naprogramovaný server spustiť, je nutné, aby server obsahoval NodeJS runtime. V práci je využitá NodeJS verzia 14.x (x ako voliteľné číslo jednotlivých verzií). Príkazom `node serverhttps.js` by bol server spustený. Tento prístup sa dá použiť len v testovacom prostredí, pretože beh serveru je aktívny len v dobe, keď je k serveru pripojený užívateľ, ktorý daný príkaz spustil. Pretože je nutné, aby tento server bežal na pozadí nezávislo na pripojenom užívateľovi, bol využitý nástroj *forever* [28], ktorý sa stará o beh skriptu na pozadí. Server je potom spustený príkazom `forever start serverhttps.js`.

Webový server má priradenú verejnú IPv4 adresu 158.194.92.109. Pretože pre užívateľov nie je potrebné pamätať si konkrétnu IP adresu servera, bol pre server vytvorený doménový záznam typu A. Na server sa dá pripojiť pomocou adresy `planets.luciamahutova.eu`, ktorá odkazuje na IPv4 adresu webového serveru. Ak by bola skúmaná dátová komunikácia, tak by bolo vidieť, že užívateľ najskôr pristúpi na webovú aplikáciu s využitím protokolu HTTP na porte č.80, potom je aplikované presmerovanie 301 na HTTPS a potom užívateľ komunikuje s webovým serverom len pomocou protokolu HTTPS na porte č.443.

Dôvodom pre využitie komunikácie pomocou HTTPS namiesto HTTP je zabezpečenie komunikácie medzi serverom a klientom, pretože s využitím HTTPS je komunikácia zašifrovaná. Moderné webové prehliadače často označujú webové aplikácie, ktoré používajú HTTP namiesto HTTPS, za nebezpečné a potenciálne nebezpečné pre užívateľov. Protokoly HTTP a HTTPS využívajú transportný protokol TCP, ktorý je spojový, preto poskytuje spoľahlivú komunikáciu medzi klientom a serverom.

6.4 Použité matematické vzorce

6.4.1 Výpočet pozície telesa na orbite

Najčastejšie používaný je vzorec pre výpočet pohybu telesa na jeho orbite. Pohyb musí kopírovať orbitu v tvare elipsy a pohybovať sa po scéne okolo stredu Slnčnej sústavy, teda okolo Slnka. Uvedený vzorec je v základnom tvare, pri použití je doplnený o ďalšie potrebné údaje (s ohľadom na vytvorený model). Matematický vzorec je použitý na správne umiestnenie planét, mesiacov a kozmického objektu na ich orbitu.

Vysvetlenie premenných:

- Premenné x a y – x a y súradnice stredu orbity (elipsy),
- Premenná a – veľkosť hlavnej osi orbity (semi-major os),
- Premenná b – veľkosť vedľajšej osi orbity (semi-minor os),
- Premenná *speed* – rýchlosť pohybu telesa po orbite. Pre zobrazenie pohybu telies je potrebná zmena tejto hodnoty, ktorá je zabezpečená pomocou meniacej sa hodnoty `Date.now()`. Uvedené hodnoty sú navzájom vynásobené a použité vo funkciách *cos* a *sin*, [29]

- Premenné X a Y – vypočítané X a Y súradnice telesa, ktoré určujú jeho aktuálnu pozíciu na orbite.

Definícia 2 (Výpočet bodov elipsy pre pohyb telesa)

$$X = x + (a * \cos speed);$$

$$Y = y + (b * \sin speed);$$

6.4.2 Výpočet vzdialenosti medzi dvoma telesami

Ďalším dôležitým výpočtom je zistenie vzdialenosti umelého kozmického objektu od planéty, ktorú obieha. Výsledná vzdialenosť je použitá na výpočet rýchlosti pohybu a zmeny rýchlosti pohybu podľa toho, či sa objekt vzdaluje alebo približuje k planéte. Na výpočet vzdialenosti dvoch telies sú použité 3D vektory *THREE.Vector3*, ktoré sú nastavené na pozíciu dvoch telies – pozíciu umelého kozmického objektu a pozíciu planéty, ktorú objekt obieha. Výpočet sa označuje aj ako *Euklidovská vzdialenosť*.

Vysvetlenie premenných:

- Premenná d – vypočítaná vzdialenosť dvoch bodov,
- Premenné p a q – jednotlivé body, pre ktoré je počítaná ich vzdialenosť,
- Premenné $p1$, $p2$, $p3$ a $q1$, $q2$, $q3$ – pri výpočte vzdialenosti dvoch bodov p a q v 3D priestore sú použité 3 karteziánske súradnice týchto bodov. Pre výpočet vzdialenosti dvoch bodov v n -rozmernom priestore bude použitých n -súradníc týchto bodov.

Definícia 3 (Vzdialenosť dvoch bodov v 3D priestore)

$$d(p, q) = \sqrt{(p1 - q1)^2 + (p2 - q2)^2 + (p3 - q3)^2}$$

7 Uživatelská dokumentácia

Dokumentácia slúži na vysvetlenie základnej funkcionality aplikácie a je doplnená o ukážky. Aktuálne vysvetľovaná časť aplikácie je zvýraznená pomocou červených rámečkov na obrázkoch.

7.1 Funkcionalita menu

7.1.1 Model Slnecnej sústavy

Predstavuje možnosť zobrazenia interaktívneho modelu. V prípade, že je model zobrazený v aktuálnom okne a užívateľ vyberie v menu túto možnosť, zobrazí sa mu upozornenie o strate všetkých nastavených vlastností pre model. Po akceptovaní zmien bude model znovu načítaný do pôvodného stavu. Po zamietnutí zmien ostane otvorený aktuálny model.

V prípade, že je vo webovom prehliadači otvorených viac záložiek pre aplikáciu (napr. záložka pre model a záložka pre planéty), v jednej záložke je zobrazený model Slnecnej sústavy a v inej záložke webového prehliadača užívateľ zvolí možnosť v Menu pre ďalšie zobrazenie modelu, zobrazí sa mu upozornenie o už spustenom modeli a aktuálna záložka bude zatvorená. Spustenie druhého modelu v novej záložke je zamietnuté z dôvodu menšej záťaže webového prehliadača.

7.1.2 Planéty

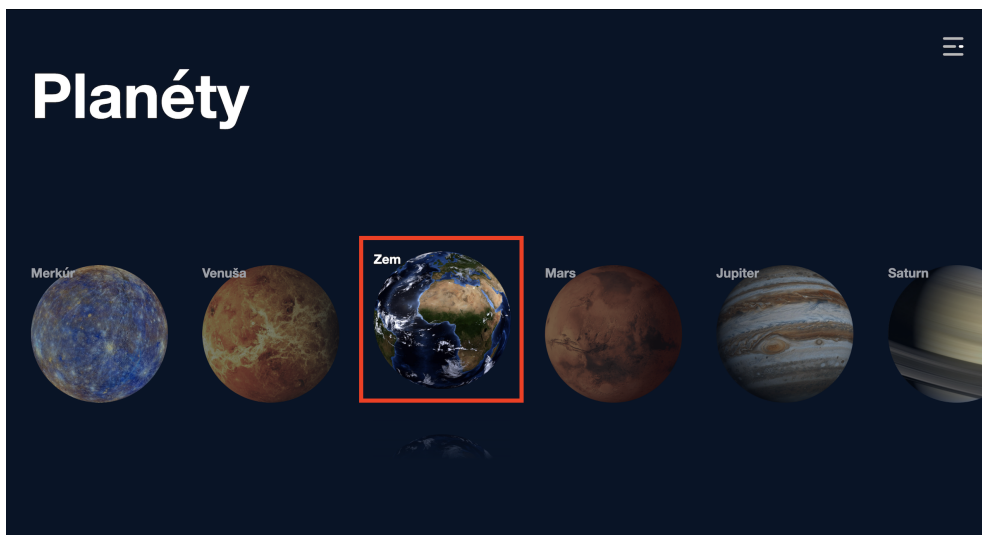
Ak bude položka vybraná zo záložky vo webovom prehliadači, v ktorej je spustená simulácia modelu Slnecnej sústavy, otvorí sa nová záložka s ponukou planét. Po výbere a kliknutí na jednu planétu sa zobrazí stránka so stručným popisom a interaktívnym modelom planéty, ktorý je možné ovládať kurzorom. Model Slnecnej sústavy bude dočasne pozastavený a znovu spustený pri otvorení záložky s modelom.

7.1.3 O aplikácii

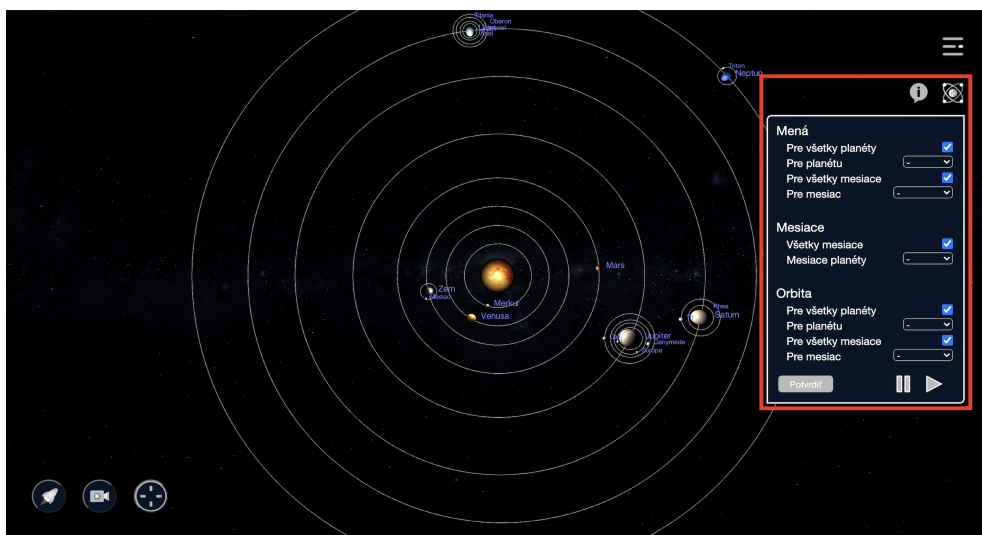
Ak bude položka vybraná zo záložky vo webovom prehliadači, v ktorej je spustená simulácia modelu Slnecnej sústavy, otvorí sa nová záložka s krátkymi vysvetlivkami o základnej funkcionalite aplikácie. Model bude dočasne pozastavený a znovu spustený pri výbere záložky s modelom.

7.1.4 Jazyky aplikácie

Aplikácia ponúka 3 jazyky: CZ ako český jazyk, EN ako anglický jazyk a SK ako slovenský jazyk. Výberom jazyka bude preložený text v aplikácii aj so zobrazenými menami planét v modeli Slnecnej sústavy.



Obr. 9: Menu – časť Planéty



Obr. 10: Pravý bočný panel na ovládanie modelu

7.2 Pravý bočný panel

7.2.1 Otvorenie a zatvorenie bočného panelu

Ovládanie funguje pomocou ikony umiestnenej nad panelom. Kliknutím na ikonu sa panel zatvorí/otvorí.

7.2.2 Mená vesmírnych objektov

Pre zobrazenie len jedného mena pre planétu alebo mesiac je potrebné zrušiť možnosť zobrazenia všetkých mien, vybrať z ponuky konkrétne meno a potvrdiť voľbu tlačidlom "Potvrdiť".

7.2.3 Mesiace

V ponuke sú možnosti zobrazenia všetkých mesiacov, skrytie všetkých mesiacov alebo zobrazenie mesiacov pre zvolenú planétu. Zobrazenie mien a orbít mesiacov závisí od ostatných volieb v bočnom paneli. Voľbu je potrebné potvrdiť tlačidlom "Potvrdiť".

7.2.4 Orbity vesmírnych objektov

Pre zobrazenie len jednej orbity pre planétu alebo mesiac je potrebné zrušiť možnosť zobrazenia všetkých orbít, vybrať z ponuky konkrétnu planétu alebo mesiac, pre ktorú bude zobrazená orbita a potvrdiť voľbu tlačidlom "Potvrdiť".

7.2.5 Tlačidlo pre zastavenie/spustenie aplikácie

Tlačidlo zastaví pohyb vesmírnych telies na orbitách a dočasne deaktivuje tlačidlo pre umelý kozmický objekt a tlačidlo pre kameru. Ostatné funkcie sú stále prístupné aj počas pozastavenia pohybu vesmírnych objektov.

7.2.6 Tlačidlo "Potvrdiť"

Tlačidlo slúži na potvrdenie a prevedenie všetkých zvolených možností v bočnom paneli. Po jeho stlačení bude tlačidlo dočasne nefunkčné, kým sa prevedú všetky zmeny.

7.3 Tlačidlá

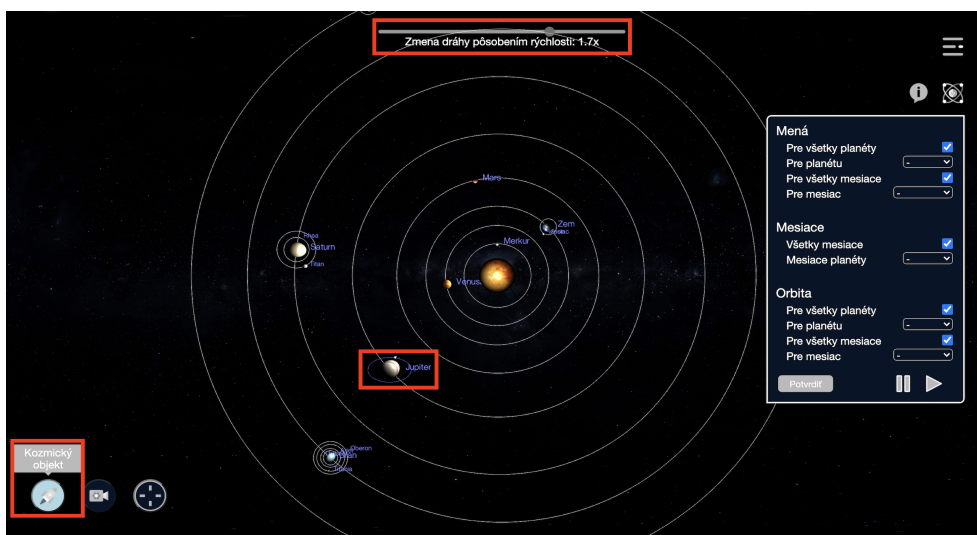
7.3.1 Tlačidlo pre umelý kozmický objekt

Kliknutím na tlačidlo je možné pridať 1 umelý kozmický objekt na orbitu poslednej vybranej planéty. Ak užívateľ zatiaľ neklikol na žiadnu planétu, po aktivácii tlačidla je potrebné vybranú planétu označiť kliknutím na ňu alebo jej orbitu. Počas aktívneho stavu tlačidla sú dočasne nefunkčné funkcie klikania na objekty

so zobrazením ich fyzikálnych vlastností a tlačidlo pre priblíženie kamery na vybranú planétu.

V prípade aktívneho tlačidla pre kozmický objekt sa zobrazí posúvač na ovládanie dráhy objektu, ktorý je umiestnený v hornej časti obrazovky. Číselné hodnoty z posúvača predstavujú zmenu v rýchlosti pohybu objektu. Príkladom je pohyb kozmickej lodi, rakety alebo satelitu, ktoré využívajú zmenu rýchlosti na zmenu tvaru či naklonenia obežnej dráhy. Zmena môže byť použitá aj na opustenie obežnej dráhy planéty a prechod na obežnú dráhu iného vesmírneho telesa. Pokiaľ nedôjde k tejto zmene, na základnú rýchlosť pohybu závisí viacero faktorov ako gravitačný ťah telesa, vzdialenosť objektu od telesa alebo poloha objektu na obežnej dráhe (perihélium – najmenšia vzdialenosť medzi dvoma objektami alebo afélium – najväčšia vzdialenosť).

Na obrázku je zobrazený umelý kozmický objekt pri planéte Jupiter. Pridanie objektu dočasne skryje zobrazené mesiace zvolenej planéty kvôli prehľadnejšiemu zobrazeniu. Deaktivovať tlačidlo je možné pomocou jeho opätovného stlačenia, prejaví sa zmenou farby tlačidla.



Obr. 11: Tlačidlo pre umelý kozmický objekt

7.3.2 Tlačidlo pre kameru

Pomocou tlačidla je možné priblížiť kameru na poslednú vybranú planétu. Ak užívateľ zatiaľ neklikol na žiadnu planétu, po aktivácii tlačidla je potrebné vybranú planétu označiť kliknutím na ňu alebo jej orbitu. Počas aktívneho stavu tlačidla ostávajú prístupné funkcie klikania na objekty a zobrazenia ich fyzikálnych vlastností.

Priblíženie kamery ponúka bližší pohľad na vesmírne teleso a jeho mesiace. Narozdiel od priblíženia celého modelu kamera nasleduje zvolený objekt pri rotácii po orbite. Deaktivovať tlačidlo je možné pomocou jeho opätovného stlačenia, prejaví sa zmenou farby.

7.3.3 Tlačidlo pre vycentrovanie modelu

Tlačidlo posunie model do stredu obrazovky. V prípade ak modelom nebolo pohnuté pomocou kurzora alebo šípok na klávesnici, model ostane na rovnakom mieste.

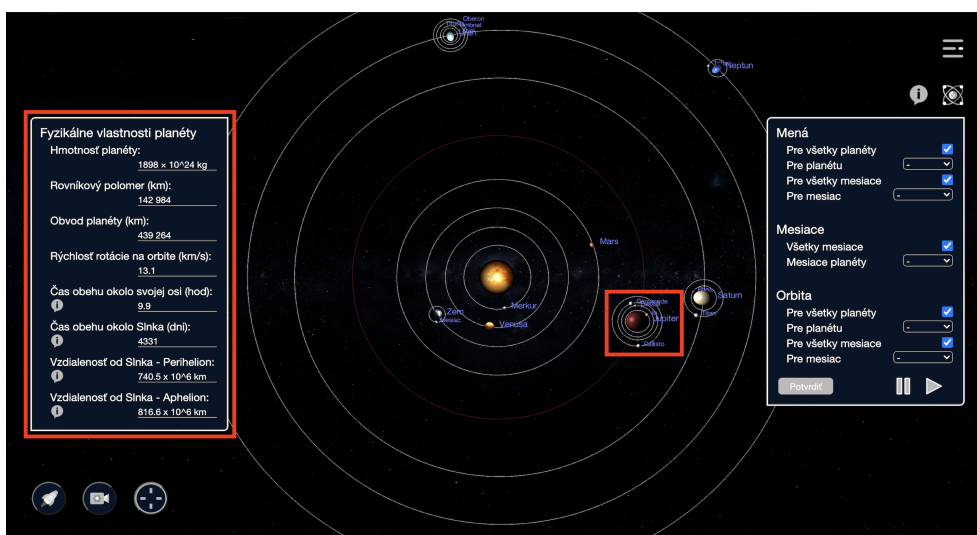
7.4 Funkcie pre posúvanie modelu

Pohybovanie modelu je možné spraviť pomocou kliknutia na ľubovoľné miesto obrazovky (okrem objektov modelu) a potiahnutím pomocou kurzora. Ďalšou možnosťou je využitie šípok na klávesnici. Vrátenie modelu na pôvodnú pozíciu je možné pomocou tlačidla na vycentrovanie modelu alebo klávesou *Escape*.

7.5 Klikanie na vesmírne telesá

Model je rozšírený o možnosť klikania na jednotlivé planéty, mesiace alebo ich orbity. Kliknutím na vesmírny objekt alebo jeho orbitu sa v ľavej časti obrazovky zobrazí tabuľka s vybranými fyzikálnymi vlastnosťami a údajmi o zvolenom telese.

Pre odlišenie je vybraný objekt aj jeho orbita farebne odlišený. Údaje v tabuľke sú dostupné len na čítanie, tabuľka je doplnená o informačné bubliny pre vysvetlenie vybraných fyzikálnych vlastností, ktoré reagujú na umiestnenie kurzora nad nimi. Zatvorenie tabuľky je možné pomocou kliknutia na ľubovoľné miesto v tabuľke. Zatvorením tabuľky bude zrušené farebné odlišenie vesmírnych telies.



Obr. 12: Kliknutie na vesmírne teleso

Záver

Cielom práce bolo vytvoriť webovú výučbovú aplikáciu, ktorá zobrazuje pohyb planét a mesiacov v Slnčnej sústave, vrátane zobrazenia ich obežných dráh aj vybraných fyzikálnych vlastností. Aplikácia mala ďalej umožniť prídanie umelého kozmického objektu na dráhu niektorého telesa a zmenou rýchlosti zobrazí pôsobenie rýchlosti a sily na zmenu dráhy telesa.

Výsledkom je interaktívna aplikácia s modelom Slnčnej sústavy, ktorá je určená prioritne pre žiakov základných a stredných škôl. Aplikácia je vyvinutá pomocou jazykov HTML, CSS a JavaScript, s využitím 3D knižnice *three.js*. Mierka pre veľkosť obežných dráh a veľkosť telies je upravená tak, aby bol zobrazený model prehľadný a náučný pre žiakov. Väčšina číselných hodnôt aplikácie je nastavená pomocou čítania dát z JSON súboru.

Webová aplikácia využíva NodeJS a je nasadená na virtuálny server s operačným systémom Linux Ubuntu. Webový server má pridelenú verejnú IPv4 adresu, ktorá bola poskytnutá školou. Pre zabezpečenie aplikácie je použitá služba HTTPS, ktorá je doplnená o vygenerované certifikáty na zabezpečenie komunikácie medzi serverom a klientom. Užívatelia sa môžu pripojiť na server pomocou adresy *planet.luciamahutova.eu*.

Aplikácia je doplnená o bočný panel, ktorým je možné ovládať zobrazenie či skrytie vesmírnych telies, ich názov a orbít. Model neobsahuje všetky mesiace planét. Zobrazené mesiace boli vybrané podľa ich veľkosti a rozlíšenia v modeli.

Ďalším rozšírením je tlačidlo pre kameru, ktorá sníma celú scénu aj s objektami. Tlačidlo priblíži kameru na vybranú planétu a nasleduje ju počas rotácie. Takto môžu žiaci skúmať konkrétnu planétu aj s jej zobrazenými mesiacmi. Pre podrobnejšie prezeranie jednotlivých planét sú vytvorené samostatné stránky spolu s modelom každej planéty. Model má priradený štruktúrovaný povrch a dá sa otáčať pomocou kurzoru myši.

Časť pre umelý kozmický objekt obsahuje ovládanie pre zmenu tvaru obežnej dráhy objektu. Túto zmenu dráhy je možné rozšíriť o známy *Hohmannov prechod* z dráhy jedného telesa na dráhu iného telesa, napr. prechod z dráhy okolo planéty Zem na dráhu okolo planéty Mars.

Aplikácia môže byť v budúcnosti rozšírená o 3D model Slnčnej sústavy, pohybovanie sa v 3D priestore a zmenu rýchlosti pohybu vesmírnych telies. Jednotlivé interaktívne modely planét je možné doplniť o vyznačenie známych a pomenovaných kráterov. Rozšírením môže byť tiež upravenie modelu pre zobrazenie vo webovom prehliadači, ktorý bude spustený na mobilnom telefóne.

Interaktívny model môže byť ďalej doplnený o meteority a zobrazenie ich zrážky s iným vesmírnym telesom, prípadne s vesmírnou loďou alebo satelitom.

Conclusions

The aim of the thesis was to create a web-based educational application that displays the motion of the planet and moons in the Solar system, including the display of their orbits and selected physical properties. The application was further intended to allow the addition of an artificial cosmic object to the trajectory of a body and by changing the velocity display the effect of velocity and force on changing the trajectory of the object.

The result is an interactive application with a model of the Solar system, which is mainly developed for students of primary and secondary school. The application is developed using HTML, CSS and JavaScript, using 3D library *three.js*. The scale for the size of the orbits and the size of the objects is adjusted so that the shown model is clear and educational for students. Most of the application's numeric values are set by reading data from a JSON file.

The web application uses NodeJS and is deployed on a virtual server with the Linux Ubuntu operating system. The web server is assigned a public IPv4 address, which was provided by the school. To secure the application, the HTTPS service is used, which is supplemented by generated certificates to ensure communication between the server and the client. Users can connect to the server using the address *planet.luciamahutova.eu*.

The application is complemented by a sidebar, which can control the visibility space objects, their name and orbit. The model does not contain all moons of planets. Several moons were selected according to their size and resolution in the model. The model can be resized by using the slider located at the bottom of the screen.

Another complement is the button for the camera, which captures the entire scene with objects. The button zooms the camera to the selected planet and follows it during rotation. In this way, students can explore a particular planet with its displayed moons. For a more detailed view of the individual planets, separate pages with the model of each planet are created. The model has structured surface and can be rotated with the mouse cursor.

The section for an artificial space object contains controls for changing the shape of the object's orbit. This change of path can be extended by known *Hohmann transition* from the path of one body to the path of another body, for example transition from orbit around planet Earth to orbit around planet Mars.

In the future, the application can be extended with a 3D model of the Solar system, moving in 3D space and changing the speed of motion of space objects. Individual interactive models of planets can be supplemented by known and named craters. An extension can also be to modify the model to be displayed in a web browser that will run on a mobile phone.

The interactive model can be further supplemented with meteorites and display of their collision with another space object, a spaceship or satellite.

A Knižnice, moduly a licencie

V aplikácii boli využité nasledovné knižnice a moduly:

- Node.js – použitá verzia 14.15.0, [30]
- Knižnica three.js – vytváranie 3D objektov a ich povrchov, WebGL rendereru, kamery, scény a názvov objektov pomocou *THREE.TextGeometry* vychádzalo z dokumentácie knižnice, [31]
- knižnica jQuery – použitá verzia 3.6.0, [32]
- knižnica OrbitControls.js. [33]

Autorské práva pre kódy a obrázky:

- Súbor *duplicate.js*: MIT licencia. Modul je použitý na zabránenie otvorenia simulácií vo viacerých oknách webového prehliadača. Bude otvorená len jedna simulácia z dôvodu zníženia záťaže webového prehliadača, [34]
- Trieda ModelScene – funkcia *mouseMoveEvent* na pohyb scény, [35]
- Trieda JSONManager – funkcia *readDataJSON* na čítanie dát z JSON súboru, [36]
- Upravenie veľkosti scény podľa veľkosti okna webového prehliadača, [37]
- Textúry pre planéty v HD rozlíšení. [38]

Číselné údaje pre JSON súbor a text:

- Rýchlosť pohybu planét po orbite, v JSON súbore označená ako *orbitalSpeed*, [39]
- Hodnoty pre nastavenie tvaru orbít pre planéty, v JSON súbore označené ako: *a*, *b*, *c*, *e*. Hodnoty sú z tabuľky č.1, [40]
- Hodnoty pre planéty, v JSON súbore označené ako *mass*, *diameter*, *rotationPeriod*, *rotationSpeed*, *orbitalPeriod*, *perihelion*, *aphelion*. Hodnoty sú ďalej použité na stránkach s interaktívnym modelom planéty, [41]
- Číselné údaje pre mesiace, uložené v JSON súbore. Hodnoty sú ďalej použité na stránkach s interaktívnym modelom planéty, [42]
- Úvodný text pre samostatné stránky s interaktívnym modelom planéty. [43]

B Obsah priloženého CD

Súčasťou práce je CD so štruktúrou:

doc/

Text práce vo formáte PDF, vytvorený s použitím záväzného štýlu KI PrF UP v Olomouci pre záverečné práce, vrátane všetkých príloh, a všetky súbory potrebné pre bezproblémové vygenerovanie PDF dokumentu textu (v ZIP archíve), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletné zdrojové texty webovej aplikácie so všetkými potrebnými textami, knižnicami a ďalšími súbormi potrebnými pre bezproblémové vytvorenie adresárovej štruktúry a skopírovanie na webový server.

readme.txt

Inštrukcie pre nasadenie webovej aplikácie na webový server, vrátane všetkých jeho požiadavkov pre jej bezproblémový beh, a webová adresa, na ktorej je aplikácia nasadená pre účel testovania pri tvorbe posudkov práce a pre účel obhajoby práce.

Naviac CD obsahuje:

install/

Inštalátor aplikácií, runtime knižníc a iných súborov potrebných na beh webovej aplikácie, ktoré nie sú štandardnou súčasťou operačného systému na beh webovej aplikácie.

literature/

UML diagram tried, zobrazuje dedičnosť a funkcionálnosť vytvorených tried aplikácie.

Literatúra

- [1] SKYLIVE. *The Sky Live* [online]. [cit. 2020-11-23]. Dostupné z URL: <<https://theskylive.com/3dsolarsystem>>.
- [2] TEAM, Solar system scope. *Planet textures* [online]. [cit. 2020-11-23]. Dostupné z URL: <<https://www.solarsystemscope.com>>.
- [3] SOLAR WALKS. *Solar Walks Lite - Planetarium 3D* [online]. [cit. 2020-11-23]. Dostupné z URL: <<https://apps.apple.com/us/app/solar-walk-lite-planetarium-3d/id1176819919>>.
- [4] FINGENT. *A Detailed Guide to Types of Software Applications: Application Software and Types of Application Software* [online]. [cit. 2021-01-30]. Dostupné z URL: <<https://www.fingent.com/blog/a-detailed-guide-to-types-of-software-applications/#Application>>.
- [5] JOHNSTON, Joe (ed.). *A beginners guide to web application development (2021): Web applications vs website* [online]. [cit. 2021-01-30]. Dostupné z URL: <<https://www.budibase.com/blog/web-application-development/>>.
- [6] RAJPUT, Mehul (ed.). *Which One to Choose: Native Apps or Progressive Web Apps?: The Benefits of Progressive Web Apps* [online]. [cit. 2021-01-30]. Dostupné z URL: <<https://www.mindinventory.com/blog/native-apps-vs-progressive-web-apps/>>.
- [7] RAJPUT, Mehul (ed.). *Which One to Choose: Native Apps or Progressive Web Apps?: The Drawbacks of Progressive Web Apps* [online]. [cit. 2021-01-30]. Dostupné z URL: <<https://www.mindinventory.com/blog/native-apps-vs-progressive-web-apps/>>.
- [8] SEGUE TECHNOLOGIES. *Client-Side vs. Server-Side Code: What's the Difference?: Client-Side Code* [online]. [cit. 2021-01-30]. Dostupné z URL: <<https://www.seguetech.com/client-server-side-code/>>.
- [9] SEGUE TECHNOLOGIES. *Client-Side vs. Server-Side Code: What's the Difference?: Server-Side Code* [online]. [cit. 2021-01-30]. Dostupné z URL: <<https://www.seguetech.com/client-server-side-code/>>.
- [10] THE SCIPY COMMUNITY. *What is NumPy?* [online]. [cit. 2021-02-15]. Dostupné z URL: <<https://numpy.org/doc/stable/user/whatismumpy.html>>.
- [11] ASTROPYTHON. *Packages* [online]. [cit. 2021-02-14]. Dostupné z URL: <<http://www.astropython.org/packages/>>.
- [12] KHALID, Yasoob (ed.). *Running Python in the Browser: SIX SYSTEMS* [online]. [cit. 2021-02-14]. Dostupné z URL: <<https://yasoob.me/2019/05/22/running-python-in-the-browser/#six-systems>>.

- [13] RASSOKHIN, Dmitrii (ed.). *The C++ programming language in cheminformatics and computational chemistry*: Conclusion: C++ as a language for scientific software development [online]. [cit. 2021-02-14]. Dostupné z URL: <<https://jcheminf.biomedcentral.com/articles/10.1186/s13321-020-0415-y#Sec19>>.
- [14] WOOD, Adam Michael (ed.). *Learn IDL Programming to Visualize Data with: About the Language* [online]. [cit. 2021-02-14]. Dostupné z URL: <<https://www.whoishostingthis.com/resources/idl/>>.
- [15] ELTON, Daniel C. (ed.). *Why physicists still use Fortran* [online]. [cit. 2021-02-14]. Dostupné z URL: <<http://www.moreisdifferent.com/2015/07/16/why-physicsts-still-use-fortran>>.
- [16] SINGH, Prateek (ed.). *The relationship between JavaScript and ECMAScript* [online]. [cit. 2021-02-21]. Dostupné z URL: <<https://js.plainenglish.io/the-relationship-between-javascript-ecmascript-6d17706a576>>.
- [17] INTERSOG. *Cool Stuff You Can Build With JavaScript* [online]. [cit. 2021-02-21]. Dostupné z URL: <<https://intersog.com/blog/code/cool-stuff-you-can-build-with-javascript/>>.
- [18] *Retrograde Motion* [online]. [cit. 2021-03-22]. Dostupné z URL: <<https://cseligman.com/text/sky/retrograde.htm>>.
- [19] BOECKMANN, Catherine (ed.). *WHAT ARE PERIHELION AND APHELION?* [online]. [cit. 2021-03-22]. Dostupné z URL: <<https://www.almanac.com/content/what-aphelion-and-perihelion>>.
- [20] BRITANNICA. *Solar system: Orbits* [online]. [cit. 2021-03-22]. Dostupné z URL: <<https://www.britannica.com/science/solar-system#ref515540>>.
- [21] JOHNSON, Lee (ed.). *What is the Difference Between Velocity And Acceleration?* [online]. [cit. 2021-03-22]. Dostupné z URL: <<https://sciencing.com/difference-between-force-velocity-8620764.html>>.
- [22] BRITANNICA. *Geostationary orbit* [online]. [cit. 2021-03-31]. Dostupné z URL: <<https://www.britannica.com/science/geostationary-orbit>>.
- [23] EUROPEAN SPACE AGENCY. *Low Earth orbit* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://www.esa.int/ESA_Multimedia/Images/2020/03/Low_Earth_orbit>.
- [24] EUROPEAN SPACE AGENCY. *Polar and Sun-synchronous orbit* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://www.esa.int/ESA_Multimedia/Images/2020/03/Polar_and_Sun-synchronous_orbit>.

- [25] BRAEUNIG, Robert A. (ed.). *ORBITAL MECHANICS: Orbit Maneuvers: Orbit Plane Changes* [online]. [cit. 2021-03-23]. Dostupné z URL: <http://www.braeunig.us/space/orbmech.htm#maneuver>.
- [26] BRAEUNIG, Robert A. (ed.). *ORBITAL MECHANICS: Orbit Maneuvers: Orbit Altitude Changes* [online]. [cit. 2021-03-23]. Dostupné z URL: <http://www.braeunig.us/space/orbmech.htm#maneuver>.
- [27] LET'S ENCRYPT. [online]. [cit. 2021-03-30]. Dostupné z URL: <https://letsencrypt.org>.
- [28] NPM. *forever: Usage* [online]. [cit. 2021-03-30]. Dostupné z URL: <https://www.npmjs.com/package/forever>.
- [29] UNITY. *Move Object allong an ellipsoid path* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://answers.unity.com/questions/27620/move-object-allong-an-ellipsoid-path.html>.
- [30] NODE.JS. *Download for macOS (x64)* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://nodejs.org/en/>.
- [31] THREE.JS. *Creating a scene* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://threejs.org/docs/index.html#manual/en/introduction/Creating-a-scene>.
- [32] *Downloading jQuery* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://jquery.com/download/>.
- [33] THREE.JS. *OrbitControls* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://threejs.org/docs/#examples/en/controls/OrbitControls>.
- [34] *jQuery plugins: Check Duplicate Browser Tabs/Windows With jQuery - duplicate.js* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://www.jqueryscript.net/other/check-duplicate-browser-tab-window.html>.
- [35] ELLIS, Daniel (ed.). *Implementing a custom drag event function in JavaScript and THREE.js* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://uxdesign.cc/implementing-a-custom-drag-event-function-in-javascript-and-three-js-dc79ee545d85>.
- [36] SINGHAL, Gaurav (ed.). *Get JavaScript Objects from a JSON File: Rendering JSON Objects from an API Call* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://www.pluralsight.com/guides/get-javascript-objects-from-a-json-file>.
- [37] *Making Our Scenes Responsive (and also Dealing with Jaggies): Extend the Resizer Class* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://discoverthreejs.com/book/first-steps/responsive-design/>.

- [38] SOLAR SYSTEM SCOPE. *Solar Textures: Planets* [online]. [cit. 2021-03-31]. Dostupné z URL: <<https://www.solarsystemscoope.com/textures/>>.
- [39] WATKINS, Thayer (ed.). *The Orbital Velocities of the Planets: Orbital Velocity Relative to That of Earth's* [online]. [cit. 2021-03-31]. Dostupné z URL: <<https://www.sjsu.edu/faculty/watkins/orbital.htm>>.
- [40] RAMANAKUMAR, V. (ed.). *Mathematically Describing Planetary Orbits in Two Dimensions: Table 1: Astronomical units* [online]. [cit. 2021-03-31]. Dostupné z URL: <<https://www.hilarispublisher.com/open-access/mathematically-describing-planetary-orbits-in-two-dimensions-2168-9679-1000414.pdf>>.
- [41] WILLIAMS, Dr. David R. (ed.). *Planetary Fact Sheet - Metric* [online]. [cit. 2021-03-31]. Dostupné z URL: <<https://nssdc.gsfc.nasa.gov/planetary/factsheet/>>.
- [42] WIKIPEDIA. *List of Solar System objects by size: List of objects by mean radius* [online]. [cit. 2021-03-31]. Dostupné z URL: <https://en.wikipedia.org/wiki/List_of_Solar_System_objects_by_size>.
- [43] SPACE FACTS. *Planet Facts: The Planets* [online]. [cit. 2021-03-31]. Dostupné z URL: <<https://space-facts.com/planets/>>.