SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-16605-115069

interaktívna aplikácia simulujúca slnečnú sústavu od jej vzniku po zánik

Bakalárska práca

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-16605-115069

interaktívna aplikácia simulujúca slnečnú sústavu od jej vzniku po zánik

Bakalárska práca

|  |  |
| --- | --- |
| Študijný program : | Aplikovaná informatika |
| Názov študijného odboru: | Informatika |
| Školiace pracovisko: | Ústav informatiky a matematiky |
| Vedúci záverečnej práce: | Ing. Dominik Janecký |

Obrázok, na ktorom je text, písmo, snímka obrazovky, dokument

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, písmo, biely

Automaticky generovaný popis

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE   
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

|  |  |
| --- | --- |
| Študijný program : | Aplikovaná informatika |
| Bakalárska práca: | Interaktívna aplikácia simulujúca Slnečnú sústavu od jej vzniku po zánik |
| Autor: | Daniel Janík |
| Vedúci záverečnej práce: | Ing. Dominik Janecký |
| Miesto a rok predloženia práce: | Bratislava 2024 |

Témou tejto bakalárskej práce je návrh a implementácia interaktívnej aplikácie v hernom engine Unity simulujúcej históriu a budúcnosť našej slnečnej sústavy. Cieľom tejto práce je vytvoriť aplikáciu, ktorá umožní prístup bežným ľudom k veľkému množstvu zaujímavých informácií o celej slnečnej sústave v jednom ucelenom celku. V prvej kapitole sa práca zaoberá analýzou dostupných technológii a ich voľbou. Druhá kapitola predstavuje porovnanie existujúcich riešení a vyvodenie ich nedostatkov. Nasledujúca kapitola sa venuje návrhu a implementácií interaktívnej aplikácie. Finálne riešenie poskytuje používateľovi možnosť výberu počiatočného roku spustenia simulácie, voľný pohyb po celej slnečnej sústave a množstvo informácií o každom vesmírnom telese nachádzajúcom sa v nej. V prílohe sa nachádza návod na inštaláciu a samotná aplikácia s príručkou obsluhy.

Kľúčové slová: Vzdelávacia aplikácia, Slnečná sústava, Unity, Vznik, Zánik, C#, Vesmír

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA   
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

|  |  |
| --- | --- |
| Study Programme: | Applied Informatics |
| Bachelor Thesis: | An Interactive application simulating the Solar System from its formation to its demise |
| Autor: | Daniel Janík |
| Supervisor: | Ing. Dominik Janecký |
| Place and year of submission: | Bratislava 2024 |

The topic of this bachelor’s thesis is the design and implementation of an interactive application made in the game engine Unity that simulates the history and future of our solar system. The goal of this thesis is to create an application that will give access to a huge amount of interesting information about the entire solar system in one comprehensive package for ordinary people. The first chapter deals with the analysis and selection of available technologies. The second chapter presents a comparison of existing solutions and identifies their shortcomings. The following chapter deals with the design and implementation of the interactive application. The final solution provides the user with the ability to select the start year of the simulation, free movement trough the solar system and a wealth of information about each celestial body. The appendix contains an installation guide and the application with a user manual.

Key words: Education application, Solar System, Unity, Formation, Demise, C#, Space

Vyhlásenie autora

Podpísaný Daniel Janík čestne vyhlasujem, že som Bakalársku prácu Interaktívna aplikácia simulujúca Slnečnú sústavu od jej vzniku po zánik vypracoval na základe poznatkov získaných počas štúdia a informácií z dostupnej literatúry uvedenej v práci. Uvedenú prácu som vypracoval pod vedením Ing. Dominika Janeckého.

V Bratislave dňa 17.05.2024

..................................................

podpis autora

Poďakovanie

Ďakujem môjmu vedúcemu práce Ing. Dominikovi Janeckému za poskytnutú podporu a odbornú pomoc. Taktiež by som sa chcel poďakovať svojej rodine a kamarátom za podporu počas štúdia.

Obsah

[Úvod 1](#_Toc166842460)

[1 Výber technológií 2](#_Toc166842461)

[1.1 Herný Engine 2](#_Toc166842462)

[1.1.1 Unreal 2](#_Toc166842463)

[1.1.2 Godot 3](#_Toc166842464)

[1.1.3 Hazel 3](#_Toc166842465)

[1.1.4 Unity 4](#_Toc166842466)

[1.2 Vývojové prostredie 5](#_Toc166842467)

[1.2.1 Visual Studio 5](#_Toc166842468)

[1.2.2 Jetbrains Rider 5](#_Toc166842469)

[1.2.3 Visual Studio Code 6](#_Toc166842470)

[1.3 3D Modelovacie softvéry 6](#_Toc166842471)

[1.3.1 Blender 6](#_Toc166842472)

[1.3.2 Cinema 4D 7](#_Toc166842473)

[1.3.3 Autodesk 3ds Max 7](#_Toc166842474)

[1.4 Grafický softvér 8](#_Toc166842475)

[1.4.1 GIMP 8](#_Toc166842476)

[1.4.2 Adobe Photoshop 9](#_Toc166842477)

[1.5 Generatívna umelá inteligencia 9](#_Toc166842478)

[1.5.1 NVIDIA Canvas 10](#_Toc166842479)

[1.5.2 Midjourney 10](#_Toc166842480)

[1.5.3 Fooocus 11](#_Toc166842481)

[2 Konkurenčné aplikácie 12](#_Toc166842482)

[2.1 Universe Sandbox 12](#_Toc166842483)

[2.2 Space Engine 13](#_Toc166842484)

[2.3 SpaceSim 13](#_Toc166842485)

[2.4 NASA's Eyes 14](#_Toc166842486)

[2.5 Zhrnutie aplikácií 14](#_Toc166842487)

[3 Návrh a Implementácia 15](#_Toc166842488)

[3.6 Návrh 15](#_Toc166842489)

[3.6.1 Špecifikácia požiadaviek 15](#_Toc166842490)

[3.6.2 Konceptuálny návrh 16](#_Toc166842491)

[3.6.3 Očakávania od aplikácie 17](#_Toc166842492)

[3.6.4 Rozkúskovanie vývoju 18](#_Toc166842493)

[3.6.5 Nastavenie projektu 18](#_Toc166842494)

[3.6.6 Rozdelenie časovej osi 18](#_Toc166842495)

[3.6.7 Rozdelenie funkcionality 19](#_Toc166842496)

[3.7 Návrh GUI 20](#_Toc166842497)

[3.7.1 Záznam udalostí 21](#_Toc166842498)

[3.7.2 Informácie o objekte 21](#_Toc166842499)

[3.7.3 Chemické prvky 21](#_Toc166842500)

[3.7.4 Debug časť 21](#_Toc166842501)

[3.8 Implementácia 21](#_Toc166842502)

[4 Testovanie a ladenie 22](#_Toc166842503)

[Záver 23](#_Toc166842504)

[Zoznam použitej literatúry 24](#_Toc166842505)

[Prílohy I](#_Toc166842506)

[Príloha A: Štruktúra elektronického nosiča II](#_Toc166842507)

Zoznam obrázkov a tabuliek

[Obr. 1 Stavový diagram návrhu aplikácie 16](#_Toc166867733)

[Obr. 2: diagram použitia simulácie 17](#_Toc166867734)

[Obr. 3: Časová os simulácia (nie v mierke) 19](#_Toc166867735)

[Obr. 4: Blokové rozdelenie v scéne 19](#_Toc166867736)

[Obr. 5: Počiatočný GUI dizajn 20](#_Toc166867737)

[Obr. 6: periodická tabuľka 21](#_Toc166867738)

Tabuľka 1 Popis tabuľky.............................................................................strana

Zoznam skratiek a značiek

UE5 – Unreal Engine 5

IDE – integrated development environment

VS – Visual Studio

VS Code – Visual Studio Code

MS – Microsoft

CAD – Computer Aided Design

GUI – Graphical User Interface

GIMP – GNU Image Manipulation Program

PS – Photoshop

AI – Artificial intelligence

NASA – Národný úrad pre letectvo a vesmír

UX – User Experience (po slovensky užívateľský zážitok)

UML – Unified Modeling Language

**Úvod**

V posledných rokoch záujem o prieskum vesmíru dosahuje svoje historické maximum, čo sa prejavuje v čoraz častejšími štartmi rakiet smerujúcich k obežnej dráhe mesiacu a aj jeho povrchu v rámci rôznych národných vesmírnych programov. Tento fenomén je často označovaný ako druhé vesmírne preteky, tentoraz medzi USA a Čínou, ktorých cieľom je opätovné pristátie človeka na mesiac [1].

Tento zvýšený záujem o vesmír je poháňaný technologickým pokrokom v oblasti výpočtovej techniky a internetu, ktorý umožňuje bežným používateľom jednoduchý prístup k širokému spektru informácií [2]. Súčasne sa zlepšuje grafické rozhranie, čo umožňuje realistickejšie vizualizácie vesmíru. Dnes sa najčastejšie stretávame s vizualizáciou vesmíru prostredníctvom kníh, dokumentárnych filmov a vedeckých článkov. Avšak obsah týchto médií je často obmedzený a poskytuje iba čiastočný pohľad na problematiku. Okrem toho tieto zdroje neposkytujú možnosť interakcie. V dôsledku toho sa môže stať, že tieto tradičné formy prezentácie vesmíru neuspokoja narastajúci dopyt po interaktívnych a prehľadných zdrojoch informácií.

Ako efektívnu alternatívu, ktorá by prekonala tieto nedostatky, možno považovať interaktívne aplikácie [3]. Tieto umožňujú používateľom aktívne skúmať vesmírne koncepty, poskytujú širšie množstvo informácií a umožňujú interakciu. Navrhovanie a implementácia interaktívnych aplikácií je kľúčové pre vytváranie pútavých vzdelávacích nástrojov, ktoré nielen dopĺňajú tradičné médiá, ale aj otvárajú cestu k lepšiemu pochopeniu a interaktívnemu prieskumu vesmíru [4].

Cieľom tejto bakalárskej práce je navrhnúť a implementovať interaktívnu aplikáciu, ktorá zobrazuje vznik a zánik našej slnečnej sústavy, pričom zjednocuje širokú škálu dostupných informácií a sprostredkúva komplexné javy bežným používateľom. Aplikácia bude následne sprístupnená verejnosti s cieľom zvýšiť záujem o vesmír a podporiť širšie pochopenie jeho fungovania. Zdrojový kód aplikácie je navrhnutý tak, aby bol jednoducho aktualizovateľný, čo umožní rýchlu implementáciu nových poznatkov o vesmírnych telesách alebo na najnovšie objavy v menej preskúmaných častiach slnečnej sústavy. Týmto zabezpečíme, že aplikácia zostane relevantná aj s väčším odstupom času.

1. Výber technológií

Návrh a vývoj počítačovej hry sa od vývoju bežnej aplikácie nelíši vo veľkom. V oboch prípadoch si vieme implementáciu zefektívniť a uľahčiť použitím dostupných knižníc a frameworkov pre daný programovací jazyk. Herný framework obsahuje kolekciu knižníc a nástrojov, ktoré vedia zjednodušiť vývoj. Herný engine naopak je softvér, ktorý ponúka vývojárovi grafické rozhranie spolu so sadou nástrojov pre grafiku, vstup, zvuk a iné. V prípade vývoju komplexných hier preferujeme herný engine, nakoľko ponúka voči frameworku oveľa viac funkcionality [5].

* 1. Herný Engine

Herný Engine je softvérové vývojové prostredie, s mnohými nástrojmi, pre optimalizáciu a zjednodušenie vývoju hier v rôznych programovacích jazykoch. Herný engine zahŕňa 2D alebo 3D grafické rozhranie, kompatibilné s rôznymi formátmi ako sú fyzikálne rozhranie, umelú inteligenciu, ktorá automaticky reaguje na vstup hráča, zvukové rozhranie a animácie a nespočetne veľa ďalších nástrojov. V súčasnej dobe je nepredstaviteľné vyvíjať hry bez použitia herného engine a ďalších vhodných nástrojov. Tieto nástroje sa z odstupom času stali široko dostupné verejnosti [6].

* + 1. Unreal

Je voľne dostupný herný engine vyvíjaný spoločnosťou Epic Games. Najnovšia dostupná verzia je UE5, ktorá dovoľuje vývojárom vytvárať tie najrealistickejšie objekty a herné scény spomedzi všetkých dostupných engine na dnešnom trhu. Vďaka tomu je UE5 je ideálnou voľbou pre graficky a výpočtovo náročné projekty, vrátane filmov a animácií [7]. Výhody UE5:

* Rozsiahla sada nástrojov pre tvorbu 3D objektov, prostredia a herných mechaník
* Vhodné obchodné podmienky, účtovanie začína od dosiahnutí 1 milióna USD s tým, že vývojár má poplatok 5% z hrubého príjmu.
* Výkonný engine napísaný v C++, zvláda výpočtovo náročne úlohy voči konkurenčným engine.
* Svetelné efekty, voči Unity má efektívnejší a optimalizovanejšie efekty.

Nevýhody UE5:

* Programovací jazyk C++, narozdiel od konkurencií, ktoré využívajú vysokoúrovňové jazyky ako napr. C#, UE5 vyžaduje znalosti komplexnejšieho jazyka C++.
* Menšia komunita, v porovnaní s konkurenčnými má Unreal menšiu komunitu vývojárov vytvárajúcich návody a nástroje na zlepšenie workflowu [8].
  + 1. Godot

Godot engine je multiplatformový 2D a 3D open-source herný engine. Vhodný pre začiatočníkov na tvorbu jednoduchých hier [9]. Voči ostatným herným engine sa líši v programovacom jazyku GDScript. Tento jazyk je dynamický so syntaxou podobnou k Pythonu. Avšak Godot podporuje aj C# a technológiu na konverziu na iné jazyky ako C a C++ [10].

Výhody Godotu:

* Open-source poháňaný a aktualizovaný hlavne jeho komunitou s vysokou možnou úpravou aj samotného jadra kódu.
* Nenákladný a efektívny na rôznych platformách vhodný pre hry aj simulácie a multimédiové aplikácie.
* Jednoduché používateľské rozhranie s vizuálnym skriptovaním, bez väčšieho programovania vhodný pre programovacích začiatočníkov.

Nevýhody Godotu:

* Menšia knižnica assetov, voči konkurencií, vývojár si musí mnohokrát vytvoriť vlastné assety od základov.
* Nevhodný pre výpočtovo náročné úlohy a komplexnejšie projekty veľkého rozmedzia.
* Nedostatočná dokumentácia pre vývojárov [11].
  + 1. Hazel

Je interaktívna 3D aplikačná vývojová platforma, inak nazývané ako herný engine. Vyvíjaná malým štúdiom Cherno od roku 2018. Engine začínal ako návod na vývoj herných engine na platforme youtube, neskôr vďaka komunite sa z neho stal plne funkčný engine, v ktorom sa už dnes dajú vytvárať hry. Štúdio Cherno má v budúcnosti v pláne vydať bezplatnú verziu pre všetky platformy [12].

Výhody Hazel:

* Naprogramovaný v C++, čím sa zaručuje podpora pre Windows a Linux.
* C# a .NET ako skriptovací jazyk (podobne ako pri Unity).
* Použitie technológie Vulkan pre vykreslovanie [13].

Hot reloading pre assety.

Nevýhody Hazel:

* Pomalý vývoj, štúdio Cherno sa skladá iba z piatich ľudí.
* Pomerne nový a neuhladený engine, môže prísť k veľkým chybám samostatnej aplikácie počas vývoju.
* Dostupný iba pre predplatiteľov služby patreon [12].
  + 1. Unity

Unity Engine je real-time 2D a 3D multiplatformový vývojársky engine pre umelcov, dizajnérov a vývojárov, ktorý umožňuje vytvárať interaktívne aplikácie. Unity sa využíva taktiež v filmovom, hernom priemysle a je vhodný pre tvorbu mobilných hier, ale taktiež aj pre robustné a komplexné projekty [14].

Výhody Unity:

* Intuitívne a priateľské užívateľské rozhranie Unity zjednodušuje prácu vývojárov. Bez ohľadu na ich znalosti.
* Disponuje vlastnou knižnicou s veľkým repozitárom predpripravených assetov, pluginov a nástrojov. Toto urýchľuje vývoj ponukou hotových assetov.
* Má veľkú a aktívnu komunitu vývojárov, umelcov a nadšencov v ekosystéme. Komunita poskytuje cenné zdroje, návody a podporu, čím vytvára spolupracujúce ekosystému.
* jadro engine je naprogramované v C++ zabezpečuje tým efektivitu pre robustnejšie projekty, avšak samotné hry sú programované v C# pre lepší workflow

Nevýhody Unity:

* Pre používanie pokročilejších funkcií a služieb si zaplatiť cenovo náročné predplatné ak je vývojár samostatná osoba.
* Nedostatočná podpora vizuálneho skriptovania voči Unreal Engine
* Pochopenie pokročilejších funkcií je dosť náročné
* Komplexná optimalizácia pre robustné a vysoko nákladové hry [11].
  1. Vývojové prostredie

IDE je softvér pre tvorbu aplikácií, ktorý sa skladá z bežných vývojárskych nástrojov spojených do jedného grafického rozhrania. V IDE bežne nájdeme editoru kódu, kompilátor a debugger. IDE umožňuje vývojárom zefektívniť programovanie kódu, ďalšie funkcie IDE pomáhajú organizovať pracovný postup vývojárov a riešiť problémy, pričom väčšina moderných IDE obsahuje nástroje na identifikáciu chýb v reálnom čase a zvýrazňovanie syntaxe, meniť kód počas runtime a ďalšie. V dnešnej dobe je programovanie bez IDE priam nepredstaviteľné a časovo náročné, ale stále možné používaním textových editorov [15].

Unity podporuje hŕstku textových editorov a všetky tri najpopulárnejšie IDE na programovanie v C# čo sú voľne dostupné bezplatné Visual Studio a Visual Studio Code od Microsoft a proprietárny editor Rider od spoločnosti JetBrains, pričom Visual Studio je predvolené a najpodporovanejšie IDE na tvorbu C# skriptov v prostredí Unity, rozbor týchto programov si rozoberieme v nasledujúcich podkapitolách [16].

* + 1. Visual Studio

Je integrované vývojové prostredie (IDE) pre vývojárov v .NET a C++ na platformu Windows. Obsahuje kompletnú sadu nástrojov a funkcií na zrýchlenie a vylepšenie všetkých fáz vývoja softvéru [17]. MS poskytuje 3 edície Visual Studio :

* Community – voľne dostupná, ponúka všetky podstatné nástroje na kompletný vývoj.
* Profesional – Platená edícia, potrebná pre ziskové spoločnosti.
* Enterprise – Platená edícia, ponúka navyše voči ostatným edíciám testovacie nástroje, expresnú diagnostiku a dodatočné nástroje pre prácu s Xamarin frameworkom [18].
  + 1. Jetbrains Rider

Je cross-platformové platené IDE pre vývoj .NET a hier vyvíjané spoločnosťou Jetbrains. Ponúka pokročilé nástroje na vývoj webových aplikácií pomocou ASP.NET. Podporu pre herný vývoj integráciou s Unity, cloudový vývoj [19].

* + 1. Visual Studio Code

Je samotný multiplatformový editor zdrojového kód, primárne stavaný na vývoj webu s intergrovanou podporou pre Javascript, Nodejs a TypeScript [17]. VS Code narozdiel od svojej konkurencií obsahuje bohatú sadu rozšírení udržiavaných komunitou. Pomocou týchto rozšírení je možné VS Code premeniť na takmer plne integrované vývojové prostredie pre hocijaký jazyk [20].

* 1. 3D Modelovacie softvéry

3D Modely, ktoré reprezentujú fyzický objekt alebo teleso pomocou skupiny bodov v priestore, spájané rôznymi geometrickými entitami, ako sú čiary, trojuholníky, zakrivené povrchy a pod.. 3D modely, ktoré sú skupinou dátových bodov, môžu byť vytvorené ručne, algoritmicky alebo pomocou skenovacieho softvéru, ich povrchy môžu byť ďalej zadefinované mapovaním textúr.

3D Modelovanie popisuje použitie softvérových nástrojov, ako sú CAD (Computer-Aided Design), k vytvoreniu 3D digitálnej reprezentácie objektu. 3D modelovanie sa využíva v automobilovom dizajne, výrobe priemyselných zariadení, architektúre, strojárstve a vývoje hier [21].

* + 1. Blender

Je bezplatný multiplatformový open-source 3D modelovací nástroj, vhodný pre  jednotlivcov. Podporuje celú 3D pipeline, čo zahŕňa modelovanie, manipuláciu, animácie, simulovanie, vykresľovanie, kompozíciu a sledovanie pohybu, vhodné pre herný dizajn a do videí. Pre pokročilých používateľov ponúka Blender API pre Python skriptovanie na prispôsobenie aplikácie a písanie špecializovaných nástrojov, ktoré sa neskôr zahŕňajú do budúcich verzií Blenderu [22].

Výhody Blender:

* Voľne dostupný open-source pre záujemcov 3D modelovania.
* Možnosť tvorby 2D animácií pomocou Grease Pencil.
* Obsahuje video editačné funkcie, čo konkurencia neponúka.
* Poskytuje Eevee, čo je vykresľovanie v reálnom čase pre svetelné efekty a scény.

Nevýhody Blender:

* Neintuitívne používateľské rozhranie pre začiatočníkov.
* Slabšie vykresľovacie schopnosti ako má konkurencia.
* Chýba mu špecializácia, exceluje v modelovaní a textúrovaní, ale zaostáva v požiadavkách 3D umelcov na pohybové grafiky [23].
  + 1. Cinema 4D

Cinema 4D je profesionálny softvér na modelovanie, animáciu, simuláciu a vykresľovanie 3D. Je to rýchly, výkonný, stabilný a flexibilný balík nástrojov pre efektívny a prístupný dizajn, VFX, AR/MR/VR, vývoj hier a všetkých typov vizualizácie [24]. Cinema 4D je hlavne používaný v kreatívnom priemysle pre 3D vizualizáciu a produktovú vizualizáciu, taktiež je často využívaných vo filmoch a televíznych seriáloch [25].

Výhody Cinema 4D:

* Bezpochyby najľahší na naučenie a ovládnutie pre začiatočníkov.
* Poskytuje vynikajúce možnosti simulácie častíc, ako sú mraky, oheň, voda, dym a prášok.
* Ideálny pre kombináciu pohybových grafík a vizuálnych efektov

Nevýhody Cinema 4D:

* Neexistuje bezplatná verzia, aj študentská licencia stojí 80€ ročne.
* Vyžaduje vysoko výkonný počítač, ak chce používateľ vykresliť scénu za jednu hodinu a menej.
* Pre náročnejšie projekty, je potrebné zakúpiť aj rozšírenia tretích strán [23].
  + 1. Autodesk 3ds Max

Je platený počítačový program na tvorbu 3D modelov a animovania digitálnych obrázkov. Patrí medzi najpopulárnejšie programy v priemysle počítačovej grafiky, obľúbený medzi vývojármi hier, TV štúdiami a architektmi. Stojí za ním spoločnosť Autodesk známa svojím softvérom ako je AutoCad a Maya. 3ds Max je používaný hlavne na modelovanie postáv a na vykresľovanie fotorealistických obrázkov budov a iných objektov [26].

Výhody 3ds max:

* Ponúka širokú škálu funkcií a možností, zahŕňajúc programovanie a prispôsobovanie.
* Skriptovací jazyk maxscript je relatívne ľahký na naučenie.
* Má vynikajúci editačný nástroj edit poly modifier a systém spline na trhu.

Nevýhody 3ds max:

* Cenovo náročný pre zakúpenie, existuje však študentská licencia
* Naučiť sa tento výkonný 3D softvér si vyžaduje čas.
* Môžu nastať kompatibilné problémy s iným softvérom Autodesk, ktoré si vyžadujú veľa doplnkov na funkčnosť.
* 3ds Max nie je multiplatformový, funguje iba na systéme Windows [27].
  1. Grafický softvér

Je typ počítačového programu pre tvorbu a úpravu obrázkov. Na dnešnom trhu je veľký rozsah dostupných programov pre tento účel. Jedná sa o jednoduché programy až po komplexné programy, ktoré umožňujú tvoriť dokonca aj 3D modely a animácie. Grafický softvér vieme rozdeliť do štyroch základných kategórií [28]:

* Grafické programy pre vektorovú grafiku – obrázky sa tvoria z čiar a tvarov, nestrácajú na kvalite pri zväčšení a zmenšení, používajú sa hlavne pri tvorbe GUI.
* Grafické programy pre rastrovú grafiku – obrázok je zložený z pixelov, strata kvality pri zmenšení a zväčšení, používa sa pre tvorbu textúr.
* Grafické programy pre 3D grafiku – tvorba 3D modelov, používa sa prevažne v hernom dizajne.
* Grafické programy pre animácie – tvorba pohyblivých obrázkov, používa sa prevažne pre filmy, reklamy a hry [28].
  + 1. GIMP

Celým názvom GNU Image Manipulation Program je open-source cross-platformový editor obrázkov. Ktorý je ponúka veľkú sadu nástrojov pre grafických dizajnérov, fotografov, vedcov. K nim je ešte možné vďaka open-source distribúcií doinštalovať dodatočné balíčky tretích strán na zvýšenie produktivity [29].

Výhody GIMP:

* Open-source bezplatná alternatíva pre Adobe Photoshop.
* Jednoduchý UI, vhodné pre začiatočníkov a pre malé projekty.
* Možnosť vylepšenia základnej verzie pomocou balíčkou.

Nevýhody GIMP:

* Menší počet nástrojov a funkcionalít ako Photoshop
* Nepodporuje prácu s RAW formátom obrázkov
* Malá integrácia s iným softvérom [32].
  + 1. Adobe Photoshop

Je platená [30] softvérová aplikácia na úpravu obrázkov, retušovanie fotografií, digitálny dizajn pre Windows a MacOS. Ponúka možnosť používateľom upravovať a vylepšovať obrázky a ilustrácie vďaka jeho bohatou sadou nástrojov. Je to najpoužívanejší softvér na úpravu obrázkov dokonca aj videí cez vrstvy. Adobe ponúka niekoľko verzií PS, ktoré sa líšia ich cenou a funkcionalitou [31].

Najnovšie verzie PS obsahujú nástroje s umelou inteligenciou pre generovanie obrázkov, úpravu nedostatkových detailov generatívne rozšírenie obrázku a ďalšie. Týmto kombinuje Photoshop silu umelej inteligencie a presnosť jeho nástrojov a funkcionalít pre väčšiu kontrolu nad tvorbou grafického obsahu [30].

Výhody PS:

* Integrácia generatívnej umelej inteligencie do novších verzií.
* Veľká komunita tvoriaca návody pre nové verzie PS.
* Nespočetne veľký počet funkcionalít a nástroj na editáciu fotiek po grafický dizajn.

Nevýhody PS:

* Pre plnú verziu neexistuje doživotná licencia, iba možnosť kúpy cez mesačné predplatné.
* Študentské licencie sú iba zľavnené, neexistuje možnosť používania zadarmo .
* Voči konkurencií dosť cenovo náročný grafický nástroj [32].
  1. Generatívna umelá inteligencia

Generatívna AI ponúka používateľom rýchlo generovať obsah na základne rôznych vstupov. Typy vstupov môžu byť rôzne napríklad text, obrázky, zvuky, animácie a iné.

Tieto modely umelej inteligencie využívajú neurónové siete na identifikovanie vzorov a štruktúr existujúcich údajov z databázy s cieľom vygenerovať nový originálny obsah. Najväčší prínos pre generatívne modely priniesla schopnosť využitia rôznych prístupov učenia vrátane kontrolovaného a nekontrolovaného učenia na trénovanie. Týmto sa zefektívnilo trénovanie pre veľké množstvá dát a vytvorenie základných modelov pre ďalšie špecializovanie na dané témy. Medzi základné modely patria napríklad GPT-3 a Stable Diffusion, ktoré sa následne použijú ako základ pre tvorbu špecializovaných aplikácií ako je ChatGPT, ktorý generuje na základe vstupu, textový výstup. Stable Diffusion narodiel od toho používa textový vstup pre generáciu obrázkov [33].

* + 1. NVIDIA Canvas

Je grafický nástroj, ktorý pomocou umelej inteligencie mení základné ťahy štetcom na realistické textúry krajín. Intuitívny dizajn aplikácie umožňuje rýchlo nakresliť realistické koncepty a následne ich doladiť v Photoshope vďaka možnosti exportovania obrázkov do formátu podporovaných PS. Základná verzia podporuje niekoľko textúr ako piesok, zem, rôzne typy vôd, skalnaté a kamenisté povrchy, oblohu a trávnaté porasty, k týmto textúram je ešte možné zvoliť niekoľko variácií každej textúry. Umožňuje aj import vlastných textúr. Nevýhodou je že aplikáciu je možné nainštalovať iba na platformu Windows a grafickou kartou série RTX [34].

* + 1. Midjourney

Predstavuje typ pokročilej generatívnej AI, ktorá dokáže na základe textového vstupu vytvoriť takmer dokonale realistické obrázky. Narozdiel od konkurencie funguje Midjourney cez bota na platforme Discord, pričom kvalita generovaných obrázkov sa líši na základe textového vstupu, ktorý treba detailne špecifikovať [35].

Výhody Midjouney AI:

* Ak si používateľ zakúpi predplatné má k dispozícií takmer nekonečnú knižnicu vygenerovaných obrázkov spolu s ich textovým vstupom.
* V minulosti jediná dobrá alternatíva pre rýchle a lacné generovanie kvalitných obrázkov a textúr

Nevýhody Midjourney AI:

* Nie je open-source.
* Na využitie treba zakúpiť predplatné.
* Používateľ potrebuje mať Discord účet a musí byť pripojený aspoň na hlavný Midjourney server aby mal prístup k botovi [35].
  + 1. Fooocus

Je voľne dostupný AI generátor obrázkov, ktorý prijíma textový vstup a následne ho mení na obrázkový výstup. Určený pre marketing, tvorcov pre rýchlu tvorbu. Vychádza zo základného modelu Stable Diffusion XL najnovším modelom od StabilityAI. Popri textovom vstupe si používateľ vie aplikovať rôzne umelecké filtre [36].

Výhody Foocus AI:

* Je voľne dostupný a open-source.
* Možnosť nainštalovať si ho na vlastný počítač a používať bez internetu.
* Možnosť doinštalovania rôznych balíčkov tém, pre rôzne generátory.

Nevýhody Foocus AI:

* Vyžaduje veľa pamäte.
* Generácia obrázku závisí od výkonu grafickej karty [36].

Po analýze dostupných technológií a softvérov sme si pre vývoj aplikácie zvolili, ktorý softvér bude pre nás ponúkať najvhodnejšie funkcionality a nástroje. Zvolili sme si herný engine Unity, pre jeho jednoduchosť a programovací jazyk C#. VS Code na programovanie C# skriptov do Unity, vďaka jeho ľahkosti ako IDE. Jednoduché sférické objekty budú vytvorené pomocou Blender a pre editáciu textúr sme si zvolili Photoshop, ktorý poskytuje veľkú knižnicu štetcov na texturovanie.

1. Konkurenčné aplikácie

Existuje viacero simulácií vesmíru, zameraných na rôzne aspekty jeho funkcionality a zložitosti. V tejto kapitole si preto podrobne rozoberieme štyri najpopulárnejšie simulácie, analyzujeme ich výhody, nevýhody a nedostatky. Cieľom tohto rozboru je poskytnúť podklady pre návrh a koncept našej vlastnej simulácie vesmíru. Keďže téma vesmíru a jeho simulácií je komplexná a náročná, je dôležité mať prehľad o existujúcich riešeniach a ich obmedzeniach pred vlastným návrhom. Tento kritický pohľad na súčasné simulácie nám umožní identifikovať nedostatky a možnosti zlepšenia.

* 1. Universe Sandbox

Je multiplatformová komplexná simulácia vesmíru, dostupná dokonca aj na AR/VR Primárnym cieľom aplikácie je simulácia gravitácie, teploty a klímy. Hoci sa v súčasnosti obmedzuje na newtonovskú fyziku, prebiehajú snahy o integráciu efektov relativity, tak aby neboli obmedzené výpočtovými nárokmi. Hlavným cieľom aplikácie je vytvoriť časovo stabilnú simuláciu n-body problému bez použitia kvantových počítačov [37]. Pre tých, ktorí chcú vytvárať simulácie slnečnej sústavy, predstavuje Universe Sandbox sofistikované riešenie, ktoré umožňuje interaktívne skúmanie a experimentovanie s vesmírnymi javmi [38].

Výhody:

* Rôzne scenária, najvyšší počet zo všetkých dostupných simulácií.
* Databáza vesmírnych telies obsahuje viac ako päťdesiat tisíc objektov.
* Možnosť doinštalovania rôznych balíčkov cez steam workshop vytvorených používateľmi a komunitou.
* Expertná a detailná simulácia atmosféry Zeme a Marsu.
* Možnosť tvorby detailnej vlastnej slnečnej sústavy.
* Náhodne generované vesmírne objekty.

Nevýhody:

* Platený softvér, neexistuje demo ani voľne dostupná verzia.
* Náročný a robustný softvér na výpočtovú silu.
* Pohyb kamery je dosť vyhľadený a je možné, že používateľ aj miernym pohybom pôjde až príliš ďaleko od svojho cieľu.
* Simulácia času neovplyvňuje razantne vlastnosti objektov.
  1. Space Engine

Patrí medzi najpopulárnejšiu simuláciu vesmíru. Primárne využitie Space Engine je vo dokumentárnych filmoch a videí. Jedná sa takisto o typ simulácie sandbox. Pričom rozdiel medzi konkurenciou je možnosť v prvej osobe preskúmať celý viditeľný vesmír, iné galaxie, hviezdne sústavy a podobné. Planéty a telesá sú procedurálne generované. Space Engine sa primárne sústreďuje na aspekt skúmania okolitého vesmíru ako jeho fyzikálnu simuláciu. Preto jediná simulácia gravitácie je možná iba medzi planétami a ich hviezdou [39].

Výhody:

* Veľká databáza vesmírnych telies.
* Vysoko kvalitné textúry a procedurálne generované telesá.
* Mierka vesmíru je 1:1.
* Možnosť simulácie pristátia vesmírnou loďou na planétu.
* Kamera z prvej osoby.

Nevýhody:

* Platený softvér, neexistuje voľne dostupná verzia ani demo.
* Pomerne náročný na grafiku.
* Neexistujúca simulácia času s gravitáciou.
  1. SpaceSim

Je pomerne nová simulácia vesmíru, jedná sa tiež o typ simulácie sandbox. Tento typ simulácie umožňuje používateľovi vkladať dostupné vesmírne telesá a upravovať ich fyzikálne vlastnosti. Taktiež ponúka rôzne preddefinované scenária. Softvér vznikol v roku 2018, čo ho robí najmladším spomedzi porovnávaných softvérov od Českého vývojára Pavla Ševečka. Jeho primárnou úlohou je simulovať n-body problém [37, 40].

Výhody:

* Open-source a voľne dostupný na stiahnutie.
* Rôzne scenária, ktoré konkurenčné programy zatiaľ neponúkajú.
* Možnosť prispôsobenia textúr priamo v aplikácií.
* Aplikácia zaberá minimálny priestor na disku
* Možnosť robenia videa z simulácie

Nevýhody:

* Počtom objektov a častíc klesá rýchlosť simulovania času.
* Nedoladený, môžu nastať chyby, pri ktorých treba aplikáciu reštartovať.
* Aj keď autor tvrdí, že je UI dizajn intuitívny a ľahký na použitie [39], je pre začiatočníka komplexnejší ako UI konkurencie.
  1. NASA's Eyes

Jedná sa o jednoduchú simuláciu našej slnečnej sústavy, poskytnutej na hlavnej stránke Národného úradu pre letectvo a vesmír (NASA). Simulácia zahŕňa všetky planéty našej slnečnej sústavy a po priblížení na ľubovoľnú planétu aj jej mesiace. Pričom je možné simulovať aj čas. Dodatočne je možné zapnúť aj vybrané asteroidy, kométy a súhvezdia. V skratke sa dá povedať, že simulácia obsahuje všetky vesmírne telesá, ktoré NASA v minulosti a súčasnosti preskúmava [41].

Výhody:

* Dostupné na webe.
* Kvalitné textúry vesmírnych telies.
* Zobrazenie historických udalostí ako Voyager 1.

Nevýhody:

* Neintuitívny dizajn UI.
* Simulácia času je možná iba v malom rozmedzí od 1949 po 2049.
* Limitovaný pohyb kamery a polohy hráča.
  1. Zhrnutie aplikácií

Z dôkladnej analýzy sme zistili, že aplikácie sa sústreďujú prevažne na simuláciu gravitácie pomocou N-body problému, pričom obetujú možnosť simulácie v čase, nakoľko ide o komplexný problém, ktorý je veľmi ťažko možné predpovedať pri veľkých časový rozdieloch [37]. Taktiež počas ich analýzy a testovania sme zistili, že pohyb kamery a samotného hráča nemá dobre vyriešená ani jedna aplikácia. Z týchto poznatkov vieme navrhnúť aplikáciu, ktorá by vyplnila tieto nedostatky a zaplnila dieru na trhu.

1. Návrh a Implementácia

Táto kapitola obsahuje návrh a implementáciu aplikácie, ktorá simuluje životný cyklus slnečnej sústavy, od jej formovania až po zánik. Pred každou implementáciou predchádza návrh. Kvalitný návrh je kľúčový pre hladký a bezproblémový vývoj softvéru. Pričom zlý návrh môže viesť k komplikovanému a neprehľadnému stavu samotného vývoju.

* 1. Návrh

Prvé kroky návrhu pozostávajú s vhodne určenými požiadavkami samotného používateľa aplikácie. Tento krok je nesmierne potrebný aby sme zaručili kvalitný UX. Tento návrh následne odkonzultujeme a predstavíme vzorke používateľov a prípadne nedostatkov ich upravíme.

* + 1. Špecifikácia požiadaviek

Používateľské požiadavky sa dajú rozdeliť do 3 skupín a to: funkcionálne, nefunkcionálne, doménové.

Funkcionálne požiadavky

* Používateľ musí byť schopný pozastaviť čas simulácie.
* Používateľ má možnosť voľby perspektívy kamery.
* Aplikácia musí ponúkať informácie používateľovi v adekvátnom časovom intervale.
* Aplikácia musí byť spustiteľná aj pre výkonovo slabšie počítače.
* Používateľ má možnosť pozorovať vlastnosti telies.
* Oneskorenie medzi vstupom používateľa a reakciou aplikácie by malo byť minimálne.

Nefunkcionálne požiadavky

* Aplikácia má moderné a prehľadné používateľské rozhranie.
* Aplikácia funguje správne v časoch veľkej záťaže.
* UI by malo byť intuitívne a ľahko použiteľné pre nových používateľov.

Doménové požiadavky

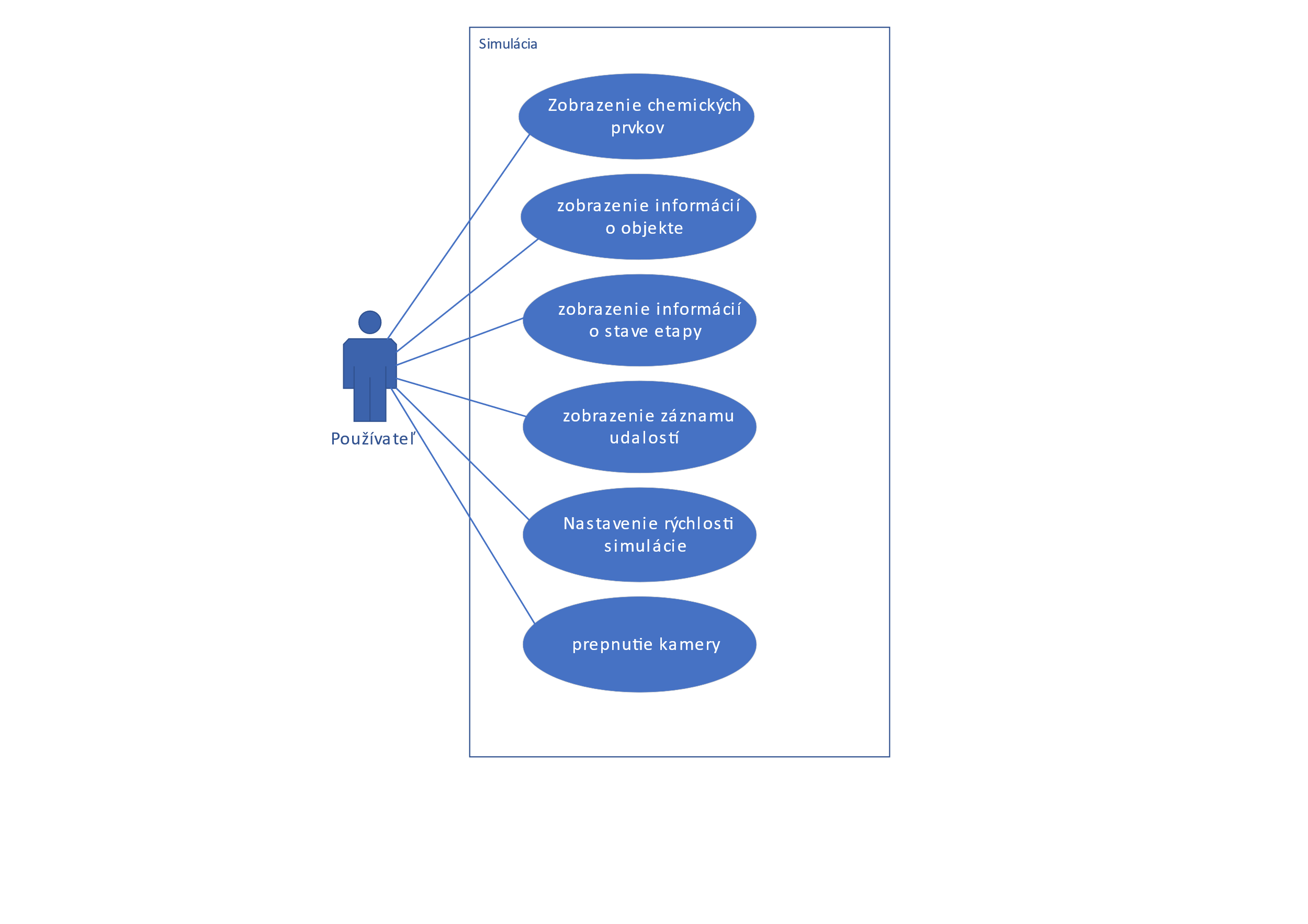
* Aplikácia musí byť prispôsobená na rôzne veľkosti obrazovky.
  + 1. Konceptuálny návrh

Po stanovení správnych požiadaviek, sme spravili koncept aplikácie. Tento sme si navrhli pomocou rôznych UML diagramov [42]. Pomocou stavového diagramu sme si navrhli celkový chod aplikácie od jej spustenia až po vypnutie. Stanovili sme si rôzne stavy, v ktorých sa používateľ môže nachádzať.



Obr. 1 Stavový diagram návrhu aplikácie

Prvý stav je hlavné menu. Používateľ by mal vedieť sa dostať do stavu nastavení, spustiť simuláciu alebo sa dostať výberu spustenia. V nastaveniach má používateľ možnosť úpravu rozličných preferencií, ako napríklad, nastavenie zvuku, hudby, rozlíšenia, levelu detailu. Z hlavného menu sa používateľ vie dostať do výberu etáp. V tomto stave má na výber 4 etapy, začiatok, skorá slnečná sústava, súčasnosť, budúcnosť. Zároveň z každého stavu, ktorý je dostupný z hlavného menu sa dá dostať späť na stav hlavné menu. Najdôležitejší stav je stav simulácie. Tento stav sa delí na 4 podstavy. Každý podstav je etapa vo vývoji slnečnej sústavy. Každý stav etapy nadväzuje pritom na jeho ďalší, pričom sa v čase nedá vrátiť do predošlého stavu. Zo simulácie sa taktiež dá dostať do hlavného menu.

Ako ďalšie sme si spravili diagram prípadov použitia pre používateľa. S týmto návrhom sme si určili čo by mal používateľ v simulácií byť schopný spraviť. Pri tomto sme opäť komunikovali so vzorkov používateľov a upravovali sme diagram podľa ich nápadov a požiadaviek.

Obr. 2: diagram použitia simulácie

Výsledný návrh je vidieť na obrázku. Používateľ má viacero možností počas simulácie zobraziť rôzne informácie, medzi dôležité patrí zobrazenie záznamu udalostí a zobrazenie informácií o objekte a možnosť prepnutie kamery.

Konceptuálny návrh sme s týmto krokom dokončili. Z tohto konceptu sme si vydedukovali čo používateľ očakáva od aplikácií simulujúcej vznik a zánik.

* + 1. Očakávania od aplikácie

Od výsledného stavu aplikácie očakávame, že bude spustiteľná a stabilná. Dôraz sa má hlavne klásť na UX a informovanosť používateľa. Aplikácia by mala disponovať moderným grafickým rozhraním, pričom by nemalo byť nadmerne komplikované a neprehľadné, nado všetkým musí byť responzívne. Vstupy by mali byť pre používateľa prirodzené a nesmie byť veľmi oneskorené tzv. špongiové.

* + 1. Rozkúskovanie vývoju

Vývoj sme si rozdelili na niekoľko kúskov/etáp. Toto rozdelenie je veľmi podstatné nakoľko nám to vie výrazne sprehľadniť vývoj a vyvarovať sa zauzlenia pri neskoršom vývoji.

Etapy :

1. Iniciálne nastavenie projektu
2. Rozdelenie časovej osi
3. Rozdelenie funkcionality na časti
4. Návrh GUI
5. Funkcionalita Simulácie
6. Doladenie výslednej simulácie

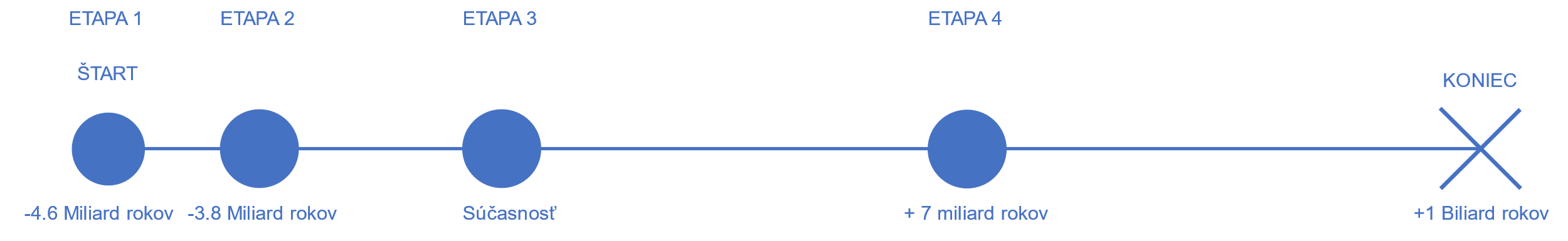
V etapách 2 až 6 sme neustále míľnikové kroky vo vývoju aplikácií testovali a konzultovali s používateľmi zaručili sme s tým, že výsledná aplikácia je presne podľa predstáv používateľa.

* + 1. Nastavenie projektu

V prvej etape sme správne nastavili prostredie Unity, integráciu VS Code. V Unity sme si pomocou Shell skriptu vytvorili základné priečinky na rozdelenie. Kód (ďalej len skripty) sme si nastavili aby sa ukladali do priečinku \_Scripts, podtržník pred S sme doplnili čisto z pohodlia aby sa nám tento priečinok v prehľadávači ukazoval na vrchu. Animation slúži na ukladanie animácií, Prefabs na uloženie prefabrikátov, Resources obsahuje skriptovacie objekty, tieto slúžia ako dátové kontajnery, Sprites slúži na ukladanie textúr a súbor Sound na zvuk. Tieto priečinky sú základné pre každý projekt, neskôr vo vývoji sme si vytvorili ďalšie súbory, ktoré sú špecifické pre našu aplikáciu.

* + 1. Rozdelenie časovej osi

Po nastavení projektu sme si nahliadli do hlavného cieľa aplikácie zobraziť celú časovú líniu od začiatku po koniec našej slnečnej sústavy. Z dostupnej literatúry sme zistili, že slnečná sústava je stará cca. 4,6 miliárd rokov [43], a taktiež že definitívny koniec nášho slnka nastane za približne 1 biliardu rokov (1015 pre lepšiu predstavu) [44]. Z týchto dostupných informácií sme sa rozhodli, že prvý možný štart bude začiatok hneď pri tvorbe a posledný počas transformácie červeného obra na bieleho trpaslíka, keďže v tomto štádiu sa už moc zaujímavého nedialo urýchlili sme simuláciu času tak, aby schladenie bieleho trpaslíka na čierneho [44] netrvalo pridlho. Ako tretí možný štart sme zvolili súčasnosť. Posledný štvrtý štart sme vybrali opäť podľa odbornej literatúry, toto obdobie je primárne zaujímavé zmenami obežných dráh našich planét a migráciou na ich dnešné polohy [43]. Výsledná dejová línia vyzerá s našimi rozhodnutiami nasledovne:



Obr. 3: Časová os simulácia (nie v mierke)

* + 1. Rozdelenie funkcionality

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popisRozdelením funkcionality na menšie bloky sme zaručili nezávislosť skriptov medzi sebou. Takto sme zabezpečili, že napr. debugovacie skripty sú nezávislé od ostatných skriptov, nemajú žiadne priame referencie na iné skripty. Spravili sme si niekoľko blokov, napríklad skripty zodpovedné za GUI majú priame referencie iba medzi inými skriptami na GUI, to platí aj pre kód ovplyvňujúci čas, fyziku, pohyb a pod. V kapitoly „Implementácia“ a jej podkapitol sa budeme viacej zaoberať týmto rozdelením a ako sme využívali jeho benefity v náš prospech. Podobné rozdelenie sme spravili aj v hierarchickom prieskumníku v Unity. To sme dosiahli vytvorením prázdnym objektov v scéne a ich vhodným pomenovaním ako je vidno na obrázku.

Obr. 4: Blokové rozdelenie v scéne

Časom ako sa aplikácia vyvíjala pribúdali objekty do scény, ktoré rýchlo napĺňali priestor v prieskumníku. Spravením tzv. blokových objektov sme si nastavili ostatné objekty relevantné k názvu blokového objektu ako „deti“ tohto objektu, tým sa nám uvoľnil priestor a prehľad objektov v scéne sa zlepšil.

* 1. Návrh GUI

Pri dizajnovaní GUI sme si zase zavolali hŕstku nami zvolených používateľov, predstavili sme im počiatočný dizajn a rozloženie elementov, tlačidiel a textov. Následne sme skúšali rôzne pozície a kombinácie rozloženia spolu s ich návrhmi. Takto sme sa vyvarovali, neskorému redizajnu v neskorších fázach vývoju. Tento krok boľ kľúčový aj z hľadiska našej predošlej logiky pri rozdeľovaní funkcionality. Spravením správneho a pre používateľa akceptovateľného GUI dizajnu sme sa tak vyvarovali neskoršiemu návratu k tomuto vývojárskemu bloku. Dôležité bolo dbať na správne umiestnenie prvkov podľa ich úlohy a dôležitosti, aby sa nestalo, že prvok, ktorý má podstatnú úlohu v našej aplikácií skončí na mieste kde ho používateľský zrak nezachytí. Jeden z takýchto prvkov bola práve časomiera, tento prvok sa skladal z dvoch častí, ikonky a textu, kde text sa aktualizoval podľa aktuálneho urýchlenia času napr. min / sek, kde časť pred lomkou znamená rýchlosť simulácie a časť za lomkou čas v realite. Po komunikácií s používateľom sme sa rozhodli tento grafický prvkov umiestniť do ľavého horného rohu obrazovky, túto praktiku sme tak aplikovali na všetky prvky, ktoré sme sa rozhodli pridať do našej aplikácie. Obrázok, na ktorom je text, snímka obrazovky, diagram, dizajn

Automaticky generovaný popis

Obr. 5: Počiatočný GUI dizajn

Z obrázku je vidieť počiatočný schválené rozloženie GUI podľa používateľa. Ďalšie kroky sa zaoberali návrhom prvkov po ich zakliknutí.

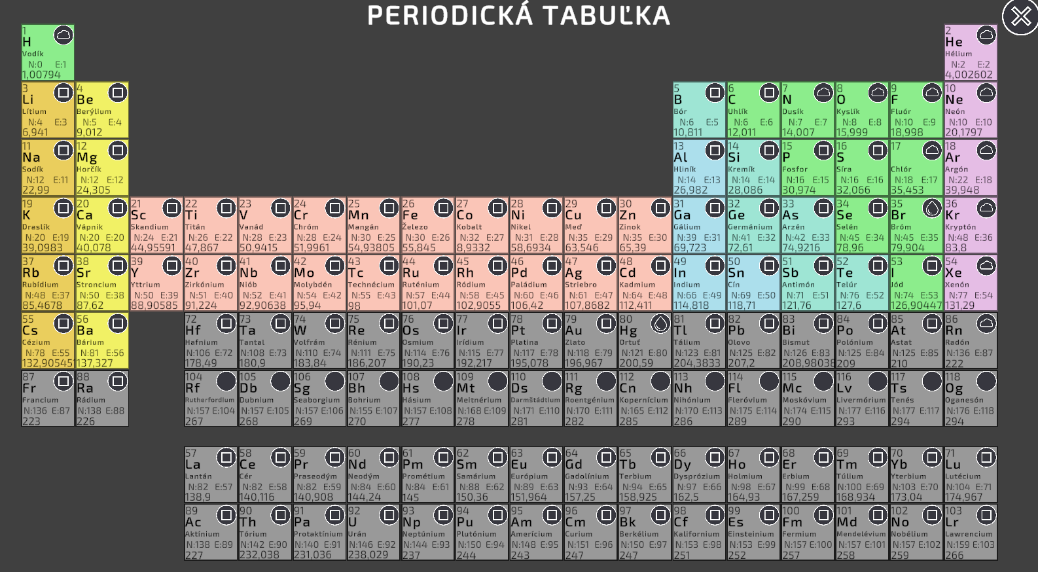
* + 1. Záznam udalostí

Záznam sme navrhli s úmyslom uchovávania všetkých udalostí, ktoré od začiatku simulácie nastali. Tieto udalosti sú rôzneho typu, bližšie sa k nim budeme venovať v nasledujúcich kapitolách. Udalosť sa najprv používateľovi zobrazí v hlavnom GUI hneď pod aktuálnym rokom (viď obrázok), postupom času ako nastanú ďalšie udalosti bude táto udalosť irelevantná a uloží sa do záznamu udalostí, kde si používateľ vie pomocou kurzoru prejsť všetky udalosti v chronologickom poradí.

* + 1. Informácie o objekte

Patrí medzi najpodstatnejšie GUI, používateľovi zobrazí informácie o príslušnom objekte. Toto rozhranie sme navrhli tak aby, používateľ dostával iba relevantné dáta k prislúchajúcemu objektu a aby nenastalo, že sa zobrazia informácie, ktoré s objektom absolútne nesúvisia. Rozhranie sa skladá z viacerých textových prvkov, ktoré bolo treba zase správne umiestniť podľa relevantnosti.

* + 1. Chemické prvky

Pri prvej konzultácií s používateľom, bolo navrhnuté implementovať aj interaktívnu periodickú tabuľku prvkov. Prvotný návrh tabuľky sme spravili s myšlienkou zvýrazňovať chemické prvky podľa ich vytvorenia v slnečnej sústave.

Obr. 6: periodická tabuľka

* + 1. Debug časť

Pri testovaní nám prišla vhod aj spätná väzba od programu, koľko snímok za sekundu dosahuje aplikácia pri záťaži, spotreba pamäte, počet vykresľovaných objektov, vypisovanie chýb atď.. Jednalo sa pritom o jednoduchý návrh, pričom sme zobrali najmenej používanú časť plochy a dosadili sme tam zopár textových elementov, ktoré sme následne napojili na skript.

V tomto okamihu sme úspešne navrhli základný koncept aplikácie. Stanovili sme si základné ciele a požiadavky, ktoré má aplikácia spĺňať. Efektívne sme si rozdelili a rozkúskovali vývoj na viacero prepojených blokov, ktoré nám časovo uľahčia implementačnú časť projektu, ktorej sa budeme podrobnejšie venovať v nasledujúcej kapitole.

* 1. Implementácia

V tejto kapitole sa venujeme vývoju simulačnej aplikácie, opíšeme si praktiky a vymoženosti herného engine Unity, ktoré sme použili, na vývoj a priblížime si k akými komplikáciami sme si museli dať rady.

* + 1. Hlavné menu

Hlavné menu tvorí podstatnú časť každej aplikácie, preto nesmie chýbať ani tu. Z návrhu vieme, že bude pozostávať z viacerých sekcií. Implementácia hlavného menu je celkom priamočiara. Vytvorili sme si novú prázdnu scénu v Unity. Do nej sme ako prvé pridali Canvas. Canvas v Unity je kontajner pre GUI elementy, ktorý slúži na ich vykreslenie [45]. V ňom sme pridali šesť textových elementov, meno aplikácii, štart, etapy, nastavenia, vypnúť a meno autora. Pre texty štart, etapy, nastavenia, vypnúť sme pridali Button komponent, ktorý pridá textu funkcionalitu tlačidla. Posledný krok spočíval v naprogramovaní funkcionality hlavného mena, ktoré je pomerne jednoduché vďaka funkciám, ktoré nám poskytuje Unity. Pre vytvorenie sekcie etapy a nastavenia sme zobrali Canvas hlavného menu a duplikovali sme ho, premenovali sme text, prehodili logiku tlačidiel aby sme sa mohli prepínať medzi všetkými sekciami. V sekcií nastavenia sme pridali dva posuvníky na zvuk a hudbu, zaškrtávacie políčko pre zobrazenie na celú obrazovku a rozbaľovací zoznam pre rozlíšenie. Tlačidlo vypnúť sme premenovali na späť a prehodili funkcionalitu, aby po kliknutí sme sa vrátili na začiatočnú ponuku.

*public* *void* Exit()

{

    StartCoroutine(WaitForAnimationDisable(*null*));

    Application*.*Quit();

Debug*.*Log("Quit");

}

Výpis 1: Ukážka funkcie na vypnutie aplikácie

Tvorenie kódu pre nastavenie aplikácie je taktiež priamočiare a jednoduché vďaka poskytnutým funkciám od Unity. Jedinú časť, pri ktorej sme si museli pomôcť je získanie všetkých dostupných rozlíšení, ktoré máme na výber.

*private* *void* Start()

{

    resolutions *=* Screen*.*resolutions;

    resolutionDropdown*.*ClearOptions();

    List<*string*> options *=* *new*();

*int* resIndex *=* 0;

*for* (*int* i *=* 0; i *<* resolutions*.*Length; i*++*)

    {

*string* option *=* resolutions[i]*.*width *+* " \* " *+* resolutions[i]*.*height   
 *+* " " *+* resolutions[i]*.*refreshRate *+* "hz";

        options*.*Add(option);

*if* (resolutions[i]*.*width *==* Screen*.*currentResolution*.*width *&&*   
 resolutions[i]*.*height *==* Screen*.*currentResolution*.*height)

            resIndex *=* i;

    }

    resolutionDropdown*.*AddOptions(options);

    resolutionDropdown*.*value *=* resIndex;

    resolutionDropdown*.*RefreshShownValue();

}

Výpis 2: Získanie a pridanie rozlíšení do rozbaľovacieho zoznamu [46]

Pre sekciu voľba etapy sme taktiež pomenili iba texty a pridali kratučký skript na načítanie príslušnej scény, ktorú sme stanovili cez Unity editor.

* + 1. Kamerový systém
    2. Pohyb používateľa
    3. Scriptable objekty
    4. Grafické používateľské rozhranie

1. Testovanie a ladenie

**Záver**

V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom.

**Zoznam použitej literatúry**

1. **Petrova, M.** *Why there is a new global race to the moon*, [online]. 20.01.2024, [cit. 09.05.2024]. Dostupné z: <https://www.cnbc.com/2024/01/20/why-there-is-a-new-global-race-to-the-moon-.html>
2. **SAYKILI, A.** *Higher education in the digital age: The impact of digital connective technologies*. Journal of Educational Technology and Online Learning, 2019, 2.1: 1-15. [cit. 10.05.2024]. Dostupné z: <https://eric.ed.gov/?id=ED591364>
3. **Huntington B. Goulding J. Pitchford N. J.** *Pedagogical features of interactive apps for effective learning of foundational skills*, British Journal of Educational Technology, 2023. [cit. 10.05.2024] ISSN 14678535. Dostupné z: <https://bera-journals.onlinelibrary.wiley.com/doi/full/10.1111/bjet.13317>
4. **ABRAMOV, D.** *Interactive Analysis Tools for Visualizing the Universe*. University of California, Santa Cruz, 2023. [cit. 10.05.2024]. Dostupné z: <https://www.proquest.com/openview/68f7fe43529067a68fe04903f9714252/1?pq-origsite=gscholar&cbl=18750&diss=y>
5. **Bhatti K. Abela P.** **Hoppe J.** *How**do you learn and master a new game* *engine or* *game framework quickly and efficiently?,* [online]. 2023, [cit 12.05.2024]. Dostupné z: <https://www.linkedin.com/advice/0/how-do-you-learn-master-new-game-engine-framework-quickly>
6. **arm,** *Game Engines,* [online]. 2021, [cit. 10.05.2024]. Dostupné z: <https://www.arm.com/glossary/gaming-engines>
7. **Unreal Engine**, *Unreal engine features,* [online]. 2018, [cit. 10.05.2024]. Dostupné z: <https://www.unrealengine.com/en-US/features>
8. **AL L., Hussain A. J**. *The Path of unity or the Path of unreal? A Comparative Study on Suitability for Game Development*. Journal of Student Research, 2019, [cit. 11.05.2024]. Dostupné z: <https://www.jsr.org/index.php/path/article/view/976/823>
9. **LINIETSKY, J.** *Godot 2.0: Talking with the Creator,* [online]. 2016, [cit. 11.05.2024]. Dostupné z: <https://imetatech.io/blog/unity-unreal-godot-comparison>
10. **HOLFELD, J.** *On the relevance of the Godot Engine in the indie game development industry*, [online]. arXiv preprint arXiv:2401.01909, 2023, [cit. 11.05.2024]. Dostupné z: <https://arxiv.org/abs/2401.01909>
11. **Sabiq**, *Comparison of Unity vs Unreal Engine Vs Godot,* [online]. 2023, [cit. 11.05.2024]. Dostupné z: <https://imetatech.io/blog/unity-unreal-godot-comparison>
12. **Studio Cherno,**  *Hazel Engine,* [online]. 2022, [cit. 13.05.2024]. Dostupné z: <https://hazelengine.com/>
13. **Chernikov Y.** *I ACCIDENTALLY Created Hazel's Greatest Feature,* [online]. 2024, [cit. 13.05.2024]. Dostupné z: <https://www.youtube.com/watch?v=yMRp9DVZYnI>
14. **Unity**, *Unity Engine* [online]. 2024, [cit. 10.05.2024]. Dostupné z: <https://unity.com/>
15. **Red Hat**,*What is an IDE?,* [online]. 2019, [cit. 12.05.2024]. Dostupné z: <https://www.redhat.com/en/topics/middleware/what-is-ide>
16. **Unity Documentation,** *Integrated development environment (IDE) support,* [online]. 2022, [cit. 12.05.2024]. Dostupné z: <https://docs.unity3d.com/Manual/ScriptingToolsIDEs.html>
17. **Microsoft**, *Visual Studio,* [online]. 2022, [cit. 12.05.2024]. Dostupné z: <https://visualstudio.microsoft.com/cs/>
18. **Microsoft,** *Compare Visual Studio Editions 2022,* [online]. 2022, [cit. 13.05.2024]. Dostupné z: <https://visualstudio.microsoft.com/cs/vs/compare/>
19. **JetBrains**, *Rider,* [online]. 2017, [cit. 12.05.2024]. Dostupné z: <https://www.jetbrains.com/rider/>
20. **Palatkar S.** *VS Code Extensions For Near IDE Experience,* [online]. 2018, [cit. 13.05.2024]. Dostupné z: <https://faun.pub/vs-code-extensions-for-complete-ide-experience-bca5bb2f0f90>
21. **Siemens,** *3D Modeling,* [online]. 2021, [cit. 13.05.2024]. Dostupné z: <https://www.sw.siemens.com/en-US/technology/3d-modeling/>
22. **Blender,** *About,* [online]. 2019, [cit. 13.05.2024]. Dostupné z: <https://www.blender.org/about/>
23. **RebusFarm,** *Cinema 4D vs. Blender: Choosing the Right 3D Software,* [online]. 2024, [cit. 13.05.2024]. Dostupné z: <https://rebusfarm.net/blog/blender-vs-cinema-4d-choosing-the-right-3d-software>
24. **Maxon,** *Cinema 4D,* [online]. 2024, [cit. 13.05.2024]. Dostupné z: <https://www.maxon.net/en/cinema-4d>
25. **Render Farm,** *What is Cinema 4D used for?,* [online]. 2022, [cit. 13.05.2024]. Dostupné z: <https://garagefarm.net/blog/what-is-cinema-4d-used-for>
26. **Petty, J.** *What is 3ds Max & What is it Used For?,* [online]. 2016, [cit. 13.05.2024]. Dostupné z: <https://conceptartempire.com/what-is-3ds-max/>
27. **iMashup,** *3ds Max: Pros, Cons, Quirks, and Links,* [online]. 2018, [cit. 13.05.2024]. Dostupné z: <https://medium.com/imeshup/3ds-max-pros-cons-quirks-and-links-a2a48832dbbe>
28. **GeeksForGeeks,** *What is Graphics Software?,* [online]. 2022, [cit. 13.05.2024]. Dostupné z: <https://www.geeksforgeeks.org/what-is-graphics-software/>
29. **GIMP,** *GIMP - GNU Image Manipulation Program,* [online]. 1998, [cit. 13.05.2024]. Dostupné z: <https://www.gimp.org/>
30. **Adobe,** *Photoshop,* [online]. 2024, [cit. 14.05.2024]. Dostupné z: <https://www.adobe.com/sk/products/photoshop.html>
31. **American Graphics Institute,** *What is Photoshop,* [online]. 2022, [cit. 14.05.2024]. Dostupné z: <https://www.agitraining.com/adobe/photoshop/classes/what-is-photoshop>
32. **Shtanakova A.** *GIMP vs Photoshop: Which Photo Editor is Better?,* [online]. 2023, [cit. 14.05.2024]. Dostupné z: <https://skylum.com/cs/blog/gimp-vs-photoshop>
33. **Nvidia,** *What is Generative AI?,* [online]. 2024, [cit. 14.05.2024]. Dostupné z: <https://www.nvidia.com/en-us/glossary/generative-ai/>
34. **Nvidia,** *Nvidia Canvas,* [online]. 2023, [cit. 14.05.2024]. Dostupné z: <https://www.nvidia.com/en-eu/studio/canvas/>
35. **Wankhede C.** *What is Midjourney AI and how does it work?,* [online]. 2024, [cit. 14.05.2024]. Dostupné z: <https://www.androidauthority.com/what-is-midjourney-3324590/>
36. **Fooocus,** *Foocus: The Most Advanced AI Image Generator 2024,* [online]. 2024, [cit. 14.05.2024]. Dostupné z: <https://fooocusai.com/>
37. **Jar,** *Working Through the N-Body Problem in Universe Sandbox²,* [online]. 2016, [cit. 14.05.2024]. Dostupné z: <https://universesandbox.com/blog/2016/02/n-body-problem/>
38. **Giant Army,** *Universe Sandbox,* [online]. 2018, [cit. 14.05.2024]. Dostupné z: <https://universesandbox.com/>
39. **Cosmographic Software LLC,** *Space Engine,* [online]. 2019, [cit. 14.05.2024] Dostupné z: <https://spaceengine.org/>
40. **Ševeček P.** *SpaceSim,* [online]. 2018, [cit. 15.05.2024]. Dostupné z: <https://pavelsevecek.github.io/>
41. **NASA,** *Solar System Exploration,* [online]. 2024, [cit. 15.05.2024]. Dostupné z: <https://science.nasa.gov/solar-system/>
42. **UÍM**, *Softvérové inžinierstvo*, [online]. 2023, [cit. 15.05.2024]. Dostupné z: <https://uim.fei.stuba.sk/predmet/b-swi/>
43. **SEA,** *The Birth of the Solar System,* [online]. 2021. [cit. 16.05.2024]. Dostupné z: <https://www.youtube.com/watch?v=d-z0eQOEzkE&list=PLCn-jE0y6UU7bCcMjaoy-ir6-brWHXJEY>
44. **SEA,** *The**Final Age of the Solar System,* [online]. 2021. [cit. 16.05.2024]. Dostupné z: <https://www.youtube.com/watch?v=oWaDUiBHDY8&list=PLCn-jE0y6UU7bCcMjaoy-ir6-brWHXJEY>
45. **Unity Documentation,** *Canvas*, [online]. 2021, [cit. 17.05.2001]. Dostupné z: <https://docs.unity3d.com/2021.3/Documentation/Manual/class-Canvas.html>
46. **Thirslund, A.** *SETTINGS MENU in Unity,* [online]. 2017, [cit. 17.05.2024]. Dostupné z: <https://www.youtube.com/watch?v=YOaYQrN1oYQ>
47. **Brush, T.** *Make a A Gorgeous Start Menu (Unity UI Tutorial)!,* [online]. 2019, [cit. 17.05.2024]. Dostupné z: <https://www.youtube.com/watch?v=vqZjZ6yv1lA>
48. **Tesseract.** *20 Advanced Coding Tips For Big Unity Projects*, [online]. 2022, [cit. 17.05.2024]. Dostupné z: <https://www.youtube.com/watch?v=dLCLqEkbGEQ>

**Prílohy**

Príloha A: Štruktúra elektronického nosiča . . . . . . . . . . . . . . . . . . . . . . . . . II

Prílohy sú „číslované“ písmenami A, B, C...

Príloha A: Štruktúra elektronického nosiča

Štruktúra elektronického nosiča (CD, DVD, atď.) s kompletnou digitálnou verziou tlačenej formy práce, vrátane príloh, funkčných zdrojových kódov, programov (aplikácií) pripravených na inštalovanie a iných, vo všeobecnosti ťažko opísateľných ale potrebných častí. Elektronický nosič musí mať obal, pomocou ktorého sa pevne pripevní do práce. Nosič musí mať popis obsahu a meno autora.