

Uniwersytet Mikołaja Kopernika
Wydział Fizyki, Astronomii i Informatyki Stosowanej
Instytut Fizyki

Rafał Zieliński
nr albumu: 236440

Praca inżynierska
na kierunku Informatyka Stosowana

Ziemia w kosmosie

Opiekun pracy dyplomowej
dr Krzysztof Rochowicz
Instytut Fizyki

Toruń 2013

Pracę przyjmuję i akceptuję

Potwierdzam złożenie pracy dyplomowej

.....

.....

data i podpis opiekuna pracy

data i podpis pracownika dziekanatu

Dziękuję za wsparcie
i kierownictwo w napisaniu
niniejszej pracy promotorowi
doktorowi Krzysztofowi
Rochowiczowi.

UMK zastrzega sobie prawo własności niniejszej pracy inżynierskiej w celu udostępniania dla potrzeb działalności naukowo-badawczej lub dydaktycznej

Spis treści

| | Strona |
|--|---------------|
| 1. Wstęp | 5 |
| 2. Modele Układu Słonecznego | 7 |
| 2.1. Układ geocentryczny (Model Ptolemejski) | 7 |
| 2.2. Układ heliocentryczny | 10 |
| 3. Wykorzystane oprogramowanie | 15 |
| 3.1. Adobe After Effects | 15 |
| 3.2. Adobe Flash | 17 |
| 3.3. Blender | 22 |
| 3.4. FlashDevelop | 25 |
| 3.5. GIMP | 27 |
| 3.6. Inno Setup | 29 |
| 4. Program Ziemia w kosmosie | 32 |
| 4.1. Cele i założenia programu | 32 |
| 4.2. Układ heliocentryczny | 32 |
| 4.3. Porównanie układów | 33 |
| 4.4. Parametry Układu Słonecznego | 34 |
| 4.5. Układ Słoneczny na mapie Polski | 37 |
| 4.6. Skala w kosmosie | 39 |
| 5. Dane techniczne programu Ziemia w kosmosie | 41 |
| 5.1. Struktura programu | 41 |
| 5.2. Wykonanie programu | 41 |
| 5.3. Wymagania sprzętowe | 42 |
| 6. Podsumowanie | 43 |
| 7. Literatura | 45 |

1. Wstęp

Okres edukacji w życiu wielu ludzi jest jednym z najważniejszych, a dla niektórych nawet najważniejszym. Dla dynamicznego rozwoju technologii niezwykle istotne jest, aby w społeczeństwie znajdowało się jak najwięcej ludzi wyedukowanych, a żeby tak było należy pobudzać ciekawość i chęć do samodoskonalenia człowieka w najróżniejszy sposób. Bynajmniej nie należy to do prostych zadań, zwłaszcza przy stosowaniu jedynie tradycyjnych metod nauczania bazujących na przekazaniu wiedzy książkowej ze zdawkowym komentarzem prowadzącego zajęcia. W obecnych czasach, gdy coraz większy nacisk kładziony jest na poznawanie i badanie kosmosu, niezwykle istotne jest, aby kształcić nowych astronomów i fizyków. W pewnej mierze dąży to do przyszłej ekspansji i potencjalnej kolonizacji innych planet, jak i do obserwacji kosmosu w celu wczesnego wykrywania potencjalnych zagrożeń dla Ziemi do czego potrzeba znacznej liczby specjalistów. Z dużą dozą prawdopodobieństwa można stwierdzić, że w gronie astronomów powszechnie znanych i poważanych na świecie wielu po raz pierwszy zainteresowało się kosmosem po spotkaniu się z materiałami dotyczącymi Układu Słonecznego. Dlatego tak ważne jest, aby wzbudzić jak największe zainteresowanie osób dopiero kształcących się tym, co w kosmosie jest najbliższe Ziemi, czyli właśnie Układem Słonecznym. Materiały multimedialne zazwyczaj znacząco zwiększają stopień przyswajania wiedzy i zainteresowania odbiorcy. Dlatego z ich wykorzystaniem spotkać się można coraz częściej.

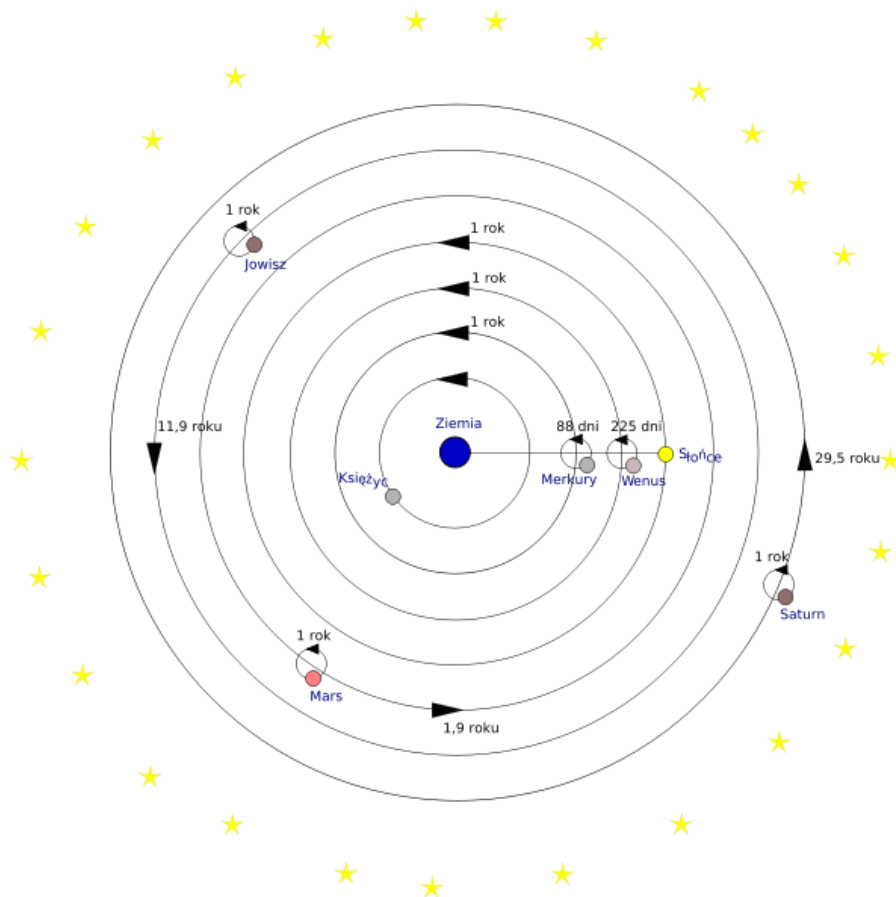
Celem niniejszej pracy było stworzenie programu multimedialnego, który wykorzystany w celach dydaktycznych zainteresuje odbiorców kwestią kosmosu. W szkołach ponadgimnazjalnych, wielu uczniów nie umie zdecydować do ostatniej klasy na czym w życiu dalej chcą się skupić, gdyż uczelnie wyższe oferują dziesiątki różnego rodzaju kierunków, natomiast w szkole uczą się wielu przedmiotów z różnych dziedzin, co z jednej strony oferuje wiele możliwości i rozwija względem wielu aspektów życia, lecz z drugiej utrudnia wybór jednej konkretnej drogi. Nierzadko nauczyciel prowadząc zajęcia w sposób interesujący, czyli na przykład przy wykorzystaniu doświadczeń lub programów multimedialnych, potrafi zainteresować uczniów prowadzonym przez siebie przedmiotem na tyle, aby w przyszłości wybrali kierunki związane z tą dziedziną. Program *Ziemia w kosmosie* pełnić ma rolę właśnie takiego elementu, który pomaga w nakierowaniu odbiorcy na zagadnienia związane z astronomią. Naturalnie spełnia swoją rolę również

w przypadku studentów, czy też osób młodszych, jak gimnazjaliści. Względem tej drugiej grupy możliwe jest, że nie wszystkie osoby będą w stanie w pełni wykorzystać ze zrozumieniem każdy aspekt aplikacji, na przykład niektóre z atrybutów dostępnych w podprogramie Parametry Układu Słonecznego, lecz nawet w takim przypadku ze względu na inne cechy program spełni swoją rolę.

Aby program *Ziemia w kosmosie* mógł w pełni spełniać swoją rolę, powinien wyróżniać się pewnymi cechami. Jedną z nich bez wątpienia jest interaktywność, aby użytkownik musiał sam coś kliknąć czy przełączyć, zamiast dostać jedynie długi tekst do przeczytania, jak to ma niekiedy miejsce. Inną jest wieloplatformowość. Aby aplikacja mogła zostać wykorzystana przez jak największą liczbę odbiorców, powinna zostać wykonana tak, aby jej użytkowanie było możliwe na urządzeniach wyposażonych w różnego rodzaju systemy operacyjne. Ponadto ze względu na fakt pojawiających się co jakiś czas nowych informacji związanych z kosmosem, budowa programu pod względem programistycznym powinna umożliwiać łatwą aktualizację zawartych w nim danych.

2. Modele Układu Słonecznego

2.1. Układ geocentryczny (Model Ptolemejski)



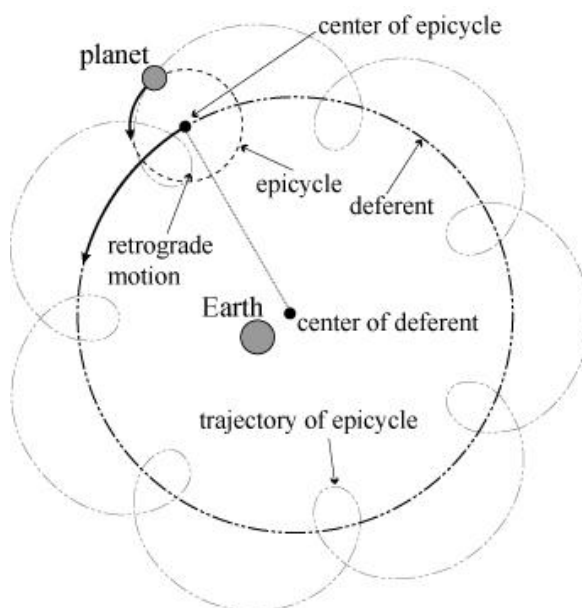
Rys. 1. Ptolemejska wizja modelu Układu Słonecznego

Źródło: http://upload.wikimedia.org/wikipedia/commons/d/d6/Geocentric_system.svg

Od czasów starożytności ludzi intrygowało to, co znajduje się poza Ziemią. W ogólnym przekonaniu społeczeństwa powszechna była dewiza, że ludzie są istotami doskonałymi, znacznie ważniejszymi od wszelkich innych. Toteż nie powinien zaskakiwać fakt, że opracowany w około IV wieku p.n.e. pierwszy model Układu Słonecznego zakładał, że w centrum znajduje się Ziemia, była to teoria sfer homocentrycznych. Jej twórcą był grecki uczoney Eudoksos z Knidos, czyli miasta leżącego w dawnej Karii (czyli krainy leżącej w obecnej południowo zachodniej Turcji). Zajmował się matematyką, astronomią, a nawet geografią czy ogólnie pojętą filozofią. Pobierał nauki między innymi u Archytasa z Tarentu czy Platona. Według jego teorii planety, do których zaliczane również były Księżyc i Słońce, wraz z gwiazdami poruszały się po sferach, których środki

znajdowały się w tym samym miejscu, będącym środkiem Ziemi. Każda ze sfer posiadała nachyloną pod pewnym kątem oś względem której obracała się ruchem jednostajnym [1]. Na wykorzystanie przez Eudoksosa sfer w swoim modelu mogły mieć wpływ poglądy filozofii pitagorejskiej zakładającej, że najdoskonalszą istniejącą figurą jest sfera, której to był zwolennikiem.

Inni uczeni szybko dopatrzili się w teorii sfer homocentrycznych błędów. Najistotniejszym z nich było założenie, że planety i gwiazdy poruszają się po sferach, których środki pokrywają się ze środkiem Ziemi, natomiast ciała niebieskie na nieboskłonie okresowo zmieniały swoją jasność, co sugerowało zmianę ich odległości. Żyjący w III wieku p.n.e. grecki astronom i matematyk Apoloniusz z Pergii (obecne terytorium Turcji) próbował dopracować koncepcje Eudoksosa przy pomocy teorii epicyklów. W II wieku p.n.e. grecki astronom, matematyk i geograf Hipparchos z Nikei (obecne terytorium Turcji), uznawszy teorię sfer homocentrycznych za nieprawidłową opracował teorię geocentryczną bazującą na epicyklach i deferentach. Zakłada, że Ziemia jest w centrum Układu Słonecznego, natomiast poszczególne planety poruszają się po kołowych orbitach, czyli epicyklach, ruchem jednostajnym, których to środki poruszają się po kołowych orbitach, czyli deferentach, których punktem centralnym jest środek Ziemi (Rys. 2.). Stwierdził również, że Słońce wraz z Księżycem poruszają się po orbitach ekscentrycznych, czyli ich środki nie pokrywały się z punktem centralnym całego Układu Słonecznego [2].



Rys. 2. Przedstawienie idei deferentów i epicykli.

Źródło: <http://www.mi.sanu.ac.rs/vismath/liao/images/retrograde%20motion.jpg>

W II wieku n.e. urodzony w Egipcie grecki astronom, matematyk i geograf Klaudiusz Ptolemeusz rozwinął teorię geocentryczną. Swoje koncepcje wraz z dotychczasowymi osiągnięciami w dziedzinie astronomii zapisał w dziele znanym obecnie pod nazwą *Almagest* (oryginalna nazwa to *Mathematike Syntaxis* czyli *struktury matematyczne*). Zwierało ono między innymi katalog 1022 gwiazd wraz z ich położeniem i jasnością [3]. Jako, że dla Greków okrąg był czymś doskonałym, a ciała niebieskie musiały się poruszać w sposób idealny, to w celu jeszcze lepszego dopasowania układu geocentrycznego do obserwacji astronomicznych Ptolemeusz postanowił wprowadzić ekwant. Była to główna zmiana dokonana przez niego. Ekwant jest punktem leżącym na linii łączącej środek Ziemi tak, aby w połowie tego odcinka wypadał punkt centralny okręgu deferentu, po którym poruszała się dana planeta [4]. Ponadto poruszający się względem niego obiekt, czy to środek epicyklu, czy samo ciało niebieskie, musiało mieć stałą prędkość kątową, czyli przemieszczać się zawsze o równy kąt w danym kwancie czasu. Dzięki epicyklom, deferentom i ekwantom, możliwe było wytłumaczenie ruchu prostego oraz wstecznego i przewidywanie położenia ciał niebieskich z dużą dokładnością [5]. Poza tym Ptolemeusz podzielił planety na dwie grupy względem Słońca:

- Dolne:
 - Księżyc
 - Merkury
 - Wenus
- Górne
 - Mars
 - Jowisz
 - Saturn

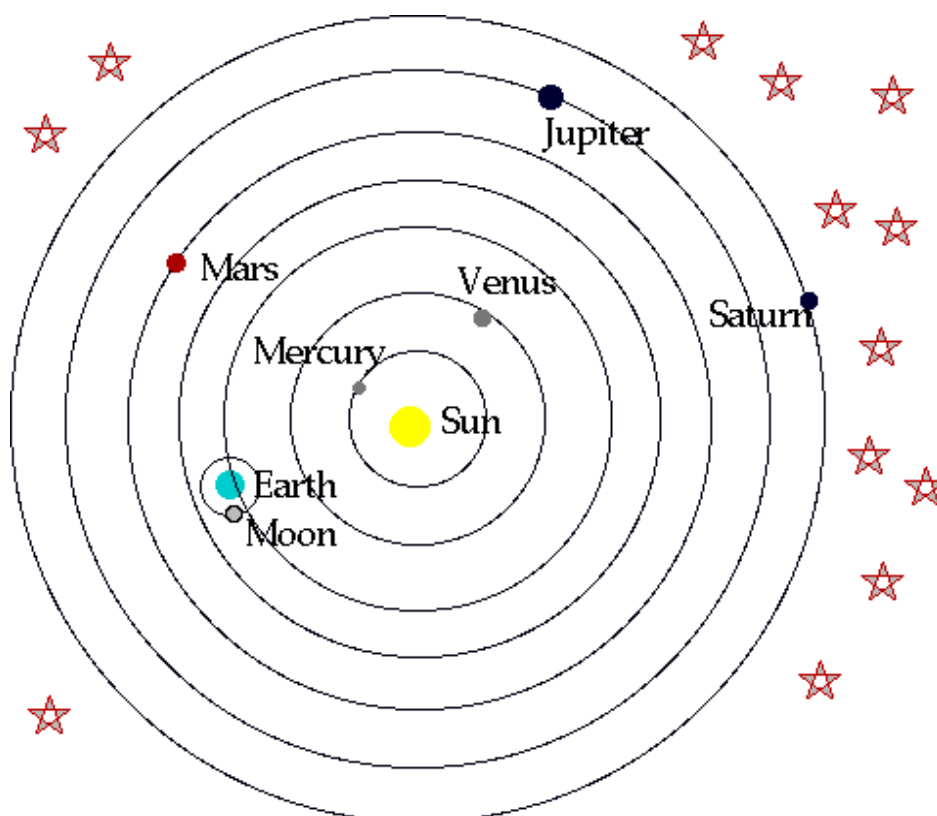
Ponadto planety górne i dolne według Ptolemeusza charakteryzowały się różnymi własnościami. Według jego teorii środki epicyklów po których poruszały się planety dolne leżały na linii łączącej Ziemię i Słońce. Natomiast promienie planet górnych do środków epicyklów są zawsze równoległe w stosunku do promienia między Słońcem za Ziemią. Ptolemeusz oszacował czasy w których poszczególne obiekty obiegały swoje epicykle i deferenty (Tab.1.). Według jego teorii gwiazdy znajdują się w sferze gwiazd stałych umiejscowionej za najdalszą odkrytą do tej pory planetą, czyli za Saturnem. Pełen obrót zajmuje im jedną dobę, czyli 24 godziny i poruszają się wokół Ziemi ze wschodu na zachód [6].

Teoria ta do czasów Mikołaja Kopernika (a nawet XVII wieku) uznawana była powszechnie jako słuszna, natomiast Almagest był źródłem podstawowej wiedzy w dziedzinie astronomii.

Tab.1. Zestawienie wartości obiegów poszczególnych obiektów.

| Obiekt | Okres obiegu po deferencie [dni] | Okres obiegu po epicyklu [dni] |
|---------|----------------------------------|--------------------------------|
| Księżyc | 1 | - |
| Merkury | 365 | 88 |
| Wenus | 365 | 225 |
| Słońce | 365 | - |
| Mars | 693,5 | 365 |
| Jowisz | 4343,5 | 365 |
| Saturn | 10767,5 | 365 |

2.2. Układ heliocentryczny



Rys. 3. Kopernikańska wizja modelu Układu Słonecznego

Źródło: http://weiblephysicsprojectsem1.wikispaces.com/file/view/Heliocentric_Model.gif/102045263/Heliocentric_Model.gif

Przekonanie o centralnym położeniu Ziemi w Układzie Słonecznym i o jej statyczności panowało niezmiennie przez ponad osiemnaście wieków, odkąd Eudoksos z Knidos przedstawił swoją teorię sfer homocentrycznych, aż do czasu ogłoszenia rewolucyjnej koncepcji Mikołaja Kopernika. Jednak nie była to pierwsza w historii próba wypchnięcia Ziemi ze środka Wszechświata. W III wieku p.n.e. grecki matematyk i astronom Arystarch z Samos opracowuje model Układu Słonecznego, w którego centrum znajduje się Słońce, a planety włącznie z Ziemią krążą wokół niego. Jednak odkrycie to nie przyniosło mu zbyt wielkiej chwały. Z powodu błędnych założeń kołowego kształtu orbit wyniki otrzymane przy wykorzystaniu jego modelu były znacznie rozbieżne od pochodzących z obserwacji. Spowodowało to ogólne odrzucenie teorii Arystarcha przez środowiska naukowe, szczególnie przez Arystotelesa. Natomiast środowiska stoickie obwołały jego koncepcję heretycką [7]. Około wiek później astronom i filozof Seleukos pochodzący prawdopodobnie z Seleucji (obecne terytorium Iraku), będący pod silnym wpływem teorii Arystarcha oficjalnie ją poparł, uznając za zgodną z jego osobistymi obserwacjami i założeniami [8]. Około czterystu lat później, po opracowaniu przez Ptolemeusza jego modelu geocentrycznego koncepcje Arystarcha i Seleukosa zostały całkowicie zepchnięte na bok, gdyż obserwacje dawały znacznie bardziej zbliżone rezultaty z założeniami Ptolemeusza. Ponadto dla środowisk naukowych, lecz także społeczeństwa, wygodniejsze było przekonanie, że planeta na której żyją znajduje się w centrum całego Układu Słonecznego.

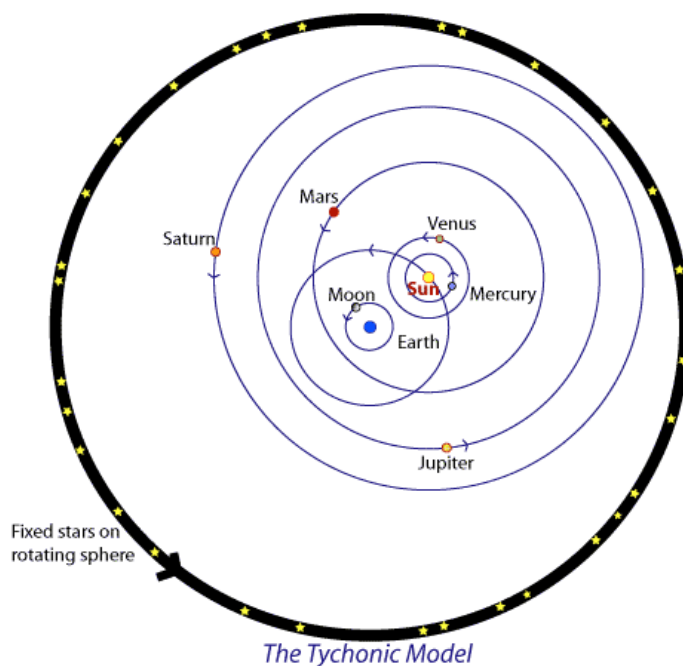
Było tak aż do XVI wieku, kiedy to Mikołaj Kopernik ogłasza swoją rewolucyjną teorię. Kopernik był uczonym pochodzącym z Torunia. Przyszedł na świat 19 lutego 1473 roku w zamożnej rodzinie. Specjalizował się w wielu naukach, lecz kojarzony najbardziej jest z astronomią. Mając podstawy w dziedzinie astronomii, matematyki, jak i znajomości języka łacińskiego rozpoczął naukę wyższą w wieku mniej więcej dziewiętnastu lat w Akademii Krakowskiej. Cztery lata później rozpoczął studia prawnicze w Bolonii, wtedy też, przy współpracy z Dominikiem Marią de Navara, rozpoczął prace prowadzące do podważenia teorii Ptolemeusza o słuszności układu geocentrycznego. W 1501 roku, pomimo ciągłej edukacji w dziedzinie prawa, rozpoczął studia medyczne, również we Włoskim mieście, tym razem w Padwie, które z powodzeniem ukończył. Natomiast studia prawnicze zakończone w 1503 roku, czynią go doktorem prawa kanonicznego, czyli takiego, którym rządzi się kościół katolicki. Mimo skupienia się na wielu innych dziedzinach nauki nie zapomina o astronomii i ok. 1509 roku tworzy pracę: *Nicolai Copernici de hypothestibus motuum coelestium a se constitutis commentariolus*, mającą na

celu przedstawienie własnych hipotez Kopernika na temat ruchu ciał niebieskich, stanowiąc zarazem pewnego rodzaju wstęp do teorii układu heliocentrycznego. Bez wątpienia do rozwoju prac Kopernika przyczyniło się też przekazanie mu w 1513 roku domu we Fromborku przez kapitułę warmińską. Dogodne warunki obserwacyjne pozwoliły mu na przeprowadzenie licznych obserwacji ciał niebieskich, co zaowocowało wykazaniem zmiennego charakteru mimośrodu Ziemi. Około 1515 roku Mikołaj Kopernik rozpoczął pisanie swojego najważniejszego dzieła: *O obrotach sfer niebieskich (De revolutionibus orbium coelestium)*, które w przyszłości zmieni oblicze całej obecnie znanej astronomii, jak i światopogląd wielu ludzi, wywołując ogromną burzę w środowiskach naukowych. To właśnie w nim opisał szczegółowo swoje teorie na temat budowy Wszechświata i o słuszności układu heliocentrycznego nad geocentrycznym. Powstało w znacznej mierze ze względu na wątpliwości Kopernika co do zgodności z rzeczywistością założeń, jakoby Ziemia miała znajdować się nieruchomo w centrum Układu Słonecznego. Wątpliwości te wynikały z paru czynników. Nie zgadzał się między innymi z istnieniem i założeniami opisującymi wprowadzony przez Ptolemeusza ekwant. Ponadto zauważył, że w przypadku teorii sfer homocentrycznych, jak i ptolemejskich, występujące zjawiska astronomiczne nie są ze sobą zgodne. Jednak dysponując tak ograniczonym sprzętem astronomicznym, jaki w tamtych czas był znany, napotkał na swej drodze pewne trudności. Główny problem jaki stał na drodze Kopernika w rozwoju swej teorii był fakt, że jego obliczenia teoretyczne, bliższe rzeczywistości, dawały wyniki zgodne w mniejszym stopniu z obserwacjami niż te pochodzące z błędnych założeń Ptolemeusza. Kopernik założył, że planety poruszają się wokół Słońca po okręgach, chcąc tym samym wykluczyć epicykle. Niestety było to niezgodne z obserwacjami, co zmusiło go do uwzględnienia w swoich koncepcjach epicykli i to w większej ilości niż Ptolemeusz, mimo to otrzymywał wciąż duże nieścisłości. Problem stanowił też brak ruchu gwiazd. Jako że Ziemia obraca się wokół własnej osi, sklepienie niebieskie mogłoby rotować w cyklu dwudziesto cztero godzinnym, lecz jako że porusza się po orbicie powinna być też zauważalna roczna paralaksa.

Mimo swoich licznych kontaktów z duchowieństwem i wysokiej pozycji społecznej obawiał się wydać swoje dzieło zakładając, że będzie obiektem ataków, zwłaszcza ze strony teologów. Jednak po namowach matematyka Jerzego Joachima Retryka w końcu zdecydował się na wydanie i tak w 1543 roku w Norymberdze ukazał się pierwszy egzemplarz dzieła astronoma. Początkowo kościół katolicki nie potępiał teorii Kopernika, jednak protestanci bardzo szybko zakazali jej szerzenia. W miarę wzrostu zainteresowania

władze katolickie odczuwając coraz większe obawy zaczęły potępiać uczonych propagujących idee Kopernika, natomiast w 1616 roku samo *De revolutionibus orbium coelestium* zostało wpisane do Indeksu ksiąg zakazanych [9], z którego zniknęło dopiero w 1757 roku.

Nie wszyscy byli od początku przychylni wobec rewolucyjnych poglądów Kopernika. Żyjący w XVI i na początku XVII wieku duński astronom Tycho Brahe mimo wielkiego szacunku jakim darzył toruńskiego astronoma, nie zgadzał się z jego teoriami. Twierdził, że brak paralaksy gwiazd jak i jego błędne wyniki obserwacji Marsa są wystarczające, by podważyć wiarygodność wszelkich znanych do tej pory modeli. Ponadto Brahe zauważył, że koniunkcja Jowisza z Saturnem, do której doszło w 1563 roku, rozbieżna jest o miesiąc w stosunku do wyników uzyskanych na podstawie modelu ptolemejskiego, natomiast względem kopernikańskiego o 2 dni. Był tak pewien swoich racji, gdyż na duńskiej wyspie Hven (obecnie terytorium Szwecji) posiadał obserwatorium wyposażone w narzędzia astronomiczne tak dokładne, jakich do tej pory nie znano. Postanowił opracować własny model Układu Słonecznego, który opublikował w 1588 roku. Zakładał on, że Ziemia znajduje się w centrum pozostając nieruchoma, natomiast Słońce wraz z Księżycem krążą wokół niej, z kolei reszta planet obraca się wokół Słońca po swoich kołowych orbitach, a sfera gwiazd znajduje się za Saturnem (Rys. 4.) [10].



Rys. 4. Model Układu Słonecznego zaproponowany przez Tycho Brahego.

Źródło: <http://outreach.atnf.csiro.au/education/senior/cosmicengine/images/cosmoimg/tychomodel.gif>

Żyjący na przełomie XVI i XVII wieku niemiecki astronom, astrolog, matematyk i filozof Johannes Kepler był zagorzałym zwolennikiem teorii Mikołaja Kopernika, którą to poznał na studiach. W jednym z pierwszych swoich dzieł, bazując na modelu heliocentrycznym, zakłada istnienie siły pochodzącej ze Słońca, która by wprawiała planety w ruch. Było to pierwsze tego typu rozważanie, gdyż do tej pory zakładano, że planety poruszają się po orbitach, których ruch nie jest niczym uwarunkowany. Niespełna cztery lata później poznaje w okolicach Pragi Tychona Brahego, niedługo potem zostaje jego uczniem. Jako, że Brahe obserwował ruch Marsa na nieboskłonie, Kepler postanowił wykorzystać ten fakt zaczynając badania mające na celu odkrycie cech charakteryzujących orbity Ziemi i Marsa. Po śmierci nauczyciela w 1605 roku po długich badaniach spostrzegł, że prawdziwy kształt orbity Marsa to elipsa, definiując tym samym zależność nazwaną później pierwszym prawem Keplera. Następnie formuje swoje drugie prawo głoszące, że w danym kwancie czasu promień łączący planetę ze Słońcem zakreśla równe pola, czyli prawo to traktuje o niejednostajnym ruchu planet po orbitach. Ostatecznie definiuje ostatnie, trzecie prawo, według którego okres obiegu planety po orbicie jest tym większy, im większa orbita. Są to fundamentalne prawa opisujące ruch planet po orbitach względem Słońca [11].

Żyjący w podobnym czasie co Kepler włoski astronom, filozof, fizyk i astrolog Galileusz był kolejnym wielkim uczonym popierającym teorię Kopernika mimo gróźb ze strony kościoła katolickiego. W 1610 roku przy pomocy teleskopu odkrywa cztery księżycy Jowisza myśląc, że to gwiazdy stałe. Lecz po około tygodniu zorientował się, że krążą po orbitach wokół Jowisza. Było to przełomowe odkrycie zgodne z prawami Keplera, zarazem potwierdzające teorię Kopernika [12].

Mimo praw Keplera i odkryć Galileusza wciąż nie potrafiono określić, czemu nie da się zaobserwować paralaksy gwiazd mimo, że w tym modelu powinna być zauważalna. Brytyjski filozof, fizyk, chemik i architekt Robert Hooke żyjący w XVII i na początku XVIII wieku po raz pierwszy wyznaczył paralaksę gwiazdy Eltanin, co było dowodem na słuszność teorii Kopernika. Jak się później okazało, jego szacowanie było błędne. Między 1832 a 1833 szkocki astronom Thomas Henderson wyznaczył paralaksę dla Alfa Centauri, następnie w 1838 roku niemiecki astronom Friedrich Bessel wyznaczył ją dla 61 Cygni, a Rosjanin Friedrich Wilhelm von Struve dla Wega [13].

W obliczu tak znacznych dowodów dla większości uczonych jak i przeciętnych ludzi stało się oczywiste, że Ziemia nie znajduje się w centrum Układu Słonecznego, a jest jedynie jedną z planet poruszającą się po eliptycznej orbicie wokół Słońca.

3. Wykorzystane oprogramowanie

3.1. Adobe After Effects

Jest to oprogramowanie przeznaczone głównie do tworzenia zaawansowanych animacji 2D. Pierwotnie zostało napisane i wydane przez programistów firmy CoSA (*Company of Science and Art*) na początku 1994 roku. Twórcy do napisania kodu źródłowego swojej aplikacji posłużyli się językiem programowania C++. Pół roku później CoSA została przejęta wraz z wszystkimi projektami przez amerykański Aldus Corporation, która to zdążyła wydać jedynie jedną nową wersję programu After Effects, gdyż rok później została wykupiona przez kolejne amerykańskie przedsiębiorstwo komputerowe Adobe, pod którego to kierownictwem rozwijane jest do dzisiaj [14]. Przejęcie przez dużą firmę niosło ze sobą pewne korzyści, otóż większość wydanych wersji zawierała znaczną liczbę usprawnień i unowocześnień. Od 2003 roku został włączony to powszechnie używanego przez artystów i nie tylko pakietu CS (*Creative Suite*). Rozpowszechniany jest na licencji komercyjnej. Może być zainstalowany i użytkowany na systemach operacyjnych z rodziny Microsoft Windows jak i Mac OS.

Od wersji 10.0, wydanej w 2010 roku wchodzącej w skład pakietu CS5 program oferuje wsparcie dla systemów operacyjnych opartych o 64 bitową architekturę. Pozwala to znacznie lepiej wykorzystać zasoby komputera, co skutkuje znacznym wzrostem wydajności widocznym nie tylko przy renderowaniu, lecz także przy zwykłej pracy z programem.

Za pomocą After Effects nie tylko można stworzyć animację, program ten posiada wiele narzędzi za pomocą których użytkownik ma możliwość przerabiania załadowanych innych animacji, ale również modyfikowania rzeczywistych filmów, a nawet korekty ich jakości czy występujących artefaktów. Jednym z przykładów może być narzędzie usuwające efekt rolling shutter, występujący często przy robieniu zdjęć metodą o tej samej nazwie, objawiający się pochyleniem, a nawet wygięciem krawędzi obiektów znajdujących się w obrębie obiektywu przy dynamicznej scenie, co spowodowane jest sposobem utrwalania obrazu w tej metodzie, polegającym na rejestracji obrazu z jednej strony kadru na drugą, czy to w orientacji pionowej czy w poziomej, czego efektem jest uchwycenie różnych fragmentów kadru w różnym czasie. Innym efektem wykorzystywanym w przemyśle filmowym niemalże przy wszystkich obecnie produkowanych filmach

posiadających efekty specjalne, ale i nie tylko jest Keylight. Polega on na umieszczeniu koloru tła, które musi być w znacznym stopniu jednolite (nazywanym powszechnie Greenscreen lub Bluescreen, gdyż zazwyczaj umieszczone tło jest koloru zielonego lub niebieskiego), w kanale alfa, czyli w kanale przezroczystości. Efektem takiego działania jest uczynienie tła niewidocznym, na którego to miejsce umieścić można cokolwiek innego, jak obraz czy też animacje. Kolejnym istotnym i popularnym narzędziem jest Puppet Tool. Służy do poruszania elementów grafiki wektorowej, ale również rastrowej w taki sposób, aby oddać jak najbardziej rzeczywisty ruch. Poprzez rozmieszczenie na odpowiednim elemencie pinezek, a następnie przemieszczaniem wybranych steruje się ruchem obiektu.

Mimo, że program ten z założenia dedykowany jest pod grafikę 2D, to ze względu na dynamiczny rozwój 3D i coraz to większe zainteresowanie tą dziedziną, postanowiono podjąć pierwsze próby implementacji trzech wymiarów do programu. Miały one miejsce w końcu lat 90tych. Wiele z dodawanych efektów miało na celu jedynie imitowanie efektu trzech wymiarów, czyli 2.5D, które polega na odpowiednim wykorzystaniu obiektu 2D przy zastosowaniu zasad rzutu izometrycznego. W wersji programu dostępnej w pakiecie CS6 została dodana rzeczywista obsługa trzeciego wymiaru, lecz wykorzystanie tej możliwości nie jest zalecane do modelowania obiektów czy też do tworzenia rzeczywistych animacji 3D. Natomiast pomaga to niezmiernie w stworzeniu jak najbardziej rzeczywistych efektów, gdyż dodając i przemieszczając obiekt posiadający widoczną głębie użytkownik nie musi symulować efektu trzech wymiarów poprzez tworzenie odpowiedniego rzutu pod danym kątem i ustawieniem kamery jak i samego obiektu, a następnie tak właśnie go przedstawiać.

Ogromnym atutem programu After Effects jest modułowa struktura. Aby jeszcze bardziej zwiększyć możliwości aplikacji, jak i potencjalne grono odbiorców twórcy pisząc kod aplikacji stworzyli możliwość dodawania zewnętrznych modułów nazywanych *plug-ins*. Wiele z dodatków wydanych przez Adobe i inne jednostki ze względu na komercyjny charakter samego programu również jest płatnych. Atutem jest fakt, że prawo do ich tworzenia ma każdy, również sami użytkownicy. Dlatego też znaleźć można dużą ilość stworzonych w warunkach domowych przez osoby pasjonujące się animacją. Jednak samodzielne stworzenie wtyczki nie jest tak prostym zadaniem dlatego, że trzeba ją napisać w którymś z języków programowania i skompilować. Wykorzystać w tym celu można różne języki, lecz najczęściej stosowany jest C i C++. Bez wątpienia jednym z bardziej złożonych elementów w procesie twórczym wtyczki jest zaprojektowanie jej

struktury i logiki, gdyż za większość efektów graficznych odpowiada złożona matematyka. Ponadto możliwe jest tworzenie pędzli w środowisku graficznym Adobe Photoshop, a następnie importowania ich do After Effects. Wbrew pozorom pędzle są istotnym elementem pracy dla grafika i animatora, zwłaszcza dla takich, którzy zajmują się tworzeniem materiałów promocyjnych, jak zaproszenia, ulotki czy reklamy.

Posiada wiele systemów kodowania obrazu jak i dźwięku. Jako, że możliwe jest otwieranie w nim plików wideo w różnych formatach, a następnie zapisanie ich w innym formacie wraz z odpowiednim kodowaniem, narzędzie to może być traktowane jak transkoder.

3.2. Adobe Flash

Jest to program głównie przeznaczony do tworzenia animacji 2D w oparciu o grafikę wektorową. Początkowo nosił nazwę FutureSplash Animator i został stworzony przez programistę Jonathana Gaya. Po mniej niecałym roku pracy w 1996 roku program został wydany przez firmę FutureWave Software, której to założycielem był właśnie Jonathan Gay jak i Charlie Jackson. Była to odpowiedź na coraz szybciej rosnące zainteresowanie tematem WWW (*World Wide Web*). Program miał służyć w dużej mierze do tworzenia animacji umieszczanych na stronach internetowych. Jako, że wtedy jedynym sposobem na odtwarzanie animacji w Internecie było wykorzystanie Javy, napisali własny odtwarzacz właśnie w tym języku programowania. Okazało się jednak, że działał niezwykle wolno. Niedługo później pojawił się Netscape z ich systemem wtyczek. Wykorzystanie tego faktu było sposobem na implementację animacji w Internecie przy zachowaniu rozsądnej wydajności przeglądarki internetowej. Program FutureSplash Animator szybko zyskał na popularności. Został wykorzystany między innymi przez Microsoft podczas tworzenia portalu MSN, czyli zbioru różnego rodzaju serwisów jak na przykład informacyjnego czy pogodowego. Również przez Disney do budowy Disney Baily Blast, czyli witryny internetowej o płatnym dostępie przeznaczonej dla dzieci. Współpraca z tak znaczącymi organizacjami zwróciła uwagę na małą firmę FutureWave Software, efektem tego było wykupienie jej pod koniec 1996 roku przez amerykańskie przedsiębiorstwo, zajmujące się tworzeniem oprogramowania do budowy stron internetowych i grafiki komputerowej. Nazwa programu FutureSplash Animator została zmieniona na Macromedia Flash. Pod nowym kierownictwem regularnie wydawane były

nowe wersje, których to w sumie było 10. Pod koniec 2005 roku korporacja Adobe System przejęła Macromedia wraz z wydanym przez nich oprogramowaniem. Od tego momentu program Macromedia Flash nosił nazwę Adobe Flash [15], a każde kolejne wersje wydawane były w obrębie nowych pakietów CS (*Creative Suite*), począwszy od CS3. Może być instalowany i użytkowany na systemach operacyjnych z rodziny Microsoft Windows, jak i Mac OS.

Program Flash można było wykorzystać do tworzenia animacji, lecz nie były one interaktywne z użytkownikiem. Jednak zmieniło się to w 2000 roku, gdy został wydany Flash 5 wraz z wbudowanym prostym językiem skryptowym ActionScript. Dawał on możliwość stworzenia logiki do animacji za pomocą typowych elementów jak pętle, warunki czy tablice. Ponadto można było tworzyć funkcje i co niezwykle ważne, obsługiwać zdarzenia. Ostatni z tych elementów pozwalał na tworzenie chociażby przycisków wykonujących określoną akcję. Szybko zauważono, że w tym momencie Flash stał się narzędziem, w którym można zbudować praktycznie całą stronę internetową, lub małe gry obsługiwane z poziomu przeglądarki. Zwłaszcza gry zyskały na popularności w krótkim czasie i cieszą się nią do dnia dzisiejszego. Ich atutem jest to, że są zazwyczaj niewielkich rozmiarów i można uruchomić je z poziomu przeglądarki. Istnieją portale, które gromadzą tego typu gry, które posegregowane są według kategorii, a dostęp do nich jest darmowy. Takie gry zazwyczaj posiadają bardzo prostą fabułę, lub wręcz żadnej, co przyciąga wielu ludzi, którzy szukają krótkiej i prostej rozrywki podczas przerwy w pracy czy między zajęciami. Wiele z nich można ukończyć w paręnaście minut.

Mimo, że idea tworzenia stron internetowych od początku do końca w środowisku Flash wiele osób przyciągała, to niosła za sobą pewne kłopoty. Pierwszym najważniejszym z nich był fakt, że aby w ogóle było możliwe otworenie takiej strony, potrzebna była specjalna wtyczka w przeglądarce internetowej, w której była otwierana. Przeciwnikami takiego rozwiązania byli głównie zwolennicy wolnego i darmowego oprogramowania dlatego, że mimo iż wtyczki niezbędne do obsługi animacji były darmowe, to niektórzy ich wydawcy tworzyli również oprogramowanie komercyjne. Były też osoby, które wykazywały obiekcje w stosunku do programów, które wymagają do poprawnego działania zainstalowania dodatkowych aplikacji, jak w tym przypadku wtyczki. Drugim, również niezwykle istotnym, szczególnie do mniej więcej 2005 roku, był sposób otwierania się wszelkich elementów stworzonych w technologii Flash umieszczanych na stronach. Dlatego, że wszystkie czy to animacje bez zaimplementowanej funkcjonalności, czy gry, czy też całe multimedialne strony wykonane

w tej technice przed wyświetleniem muszą się w pełni załadować, co jest równoznaczne z pobraniem ich do katalogów z zawartością tymczasową przeglądarki. O ile w przypadku gier zazwyczaj użytkownik widząc pasek postępu ładowania jest w stanie poczekać, to w przypadku oczekiwania aż załaduje się strona internetowa może po prostu zrezygnować nie mając pewności, że znajdzie tam poszukiwaną zawartość. W przypadku typowych stron wykonanych przy użyciu HTML (*HyperText Markup Language*), wraz z kaskadowymi arkuszami stylów CSS (*Cascading Style Sheets*) problem ten nie występuje, gdyż poszczególne elementy ładowane są sukcesywnie, tak że przeglądający zazwyczaj widzi najpierw ogólną kanwę na której zbudowana jest witryna wraz z tekstem, a dopiero później grafikę i inne elementy posiadające większy rozmiar.

Problem powolnego ładowania elementów opierających się na technologii Flash może występować do dziś, lecz latach około 2005 roku zaczęto wprowadzać coraz to większe prędkości Internetu. Pozwala to na sukcesywne zmniejszanie się czasu potrzebnego na pobranie aplikacji. Jednak niektóre osoby wciąż posiadają łącza o niższej przepustowości, dlatego też kwestia całkowitego tworzenia stron internetowych w tej technologii jest wciąż dyskusyjna. Czasami stosuje się metodę będącą kompromisem między interaktywną i multimedialną witryną, a prostszą wykonaną w tradycyjny sposób. Polega na tym, że tworzy się obie wersje i po wejściu na stronę należy wybrać, w której chce się ją oglądać.

Adobe Systems kupując firmę Macromedia wraz z ich oprogramowaniem, stało się posiadaczem Macromedia Flash Player, przemianowanego na Adobe Flash Player, czyli wtyczki instalowanej w przeglądarce internetowej, umożliwiającej odtwarzanie animacji stworzonych w formacie Flash, umieszczonych w stronach, jednak pełniła również rolę odtwarzacza tego formatu z poziomu systemu operacyjnego. W 2011 roku wraz z pojawieniem się wersji numer 11 Flash Player oferuje wsparcie dla systemów operacyjnych o architekturze 64 bitowej. Pozwala to na lepsze wykorzystanie zasobów komputera w celu przyspieszenia działania wymagających animacji. Przy wykorzystaniu AIR (*Adobe Integrated Runtime*) wydanego w 2007 roku, będącego środowiskiem wspierającym odtwarzanie aplikacji stworzonych na przykład w środowisku Flash, możliwe jest projektowanie programów, z których następnie będzie można korzystać na urządzeniach mobilnych wyposażonych w system operacyjny Android, iOS czy BlackBerry Tablet OS.

Program Flash posiada wiele elementów ułatwiających i zwiększających komfort pracy. Jednym z nich jest dokowanie, element ten występuje w wielu aplikacjach

wydawanych przez Adobe Systems, jak na przykład w After Effects, Dreamweaver, Illustrator czy Photoshop. Polega na przemieszczaniu za pomocą przeciągnięcia myszką odpowiednich okien funkcyjnych i pasków narzędzi w dowolne miejsce na ekranie, a nie tak jak w przypadku większości aplikacji tylko w ich obrębie. Ponadto możliwe jest łączenie ich ze sobą w samemu przyjętej konfiguracji, tak aby tworzyły jedno zbiorcze okno. Możliwość ta jest niezwykle przydatna i ceniona przez większość grafików, animatorów i osób posługujących się tymi narzędziami, szczególnie przez te posiadające więcej niż jeden ekran, ponieważ daje to możliwość umiejscowienia na jednym monitorze głównej części programu, natomiast na kolejnych znajdując się mogą inne narzędzia. Kolejnym istotnym elementem w przypadku osób tworzących strony internetowe w środowisku Flash i aplikacje umieszczane w Internecie jest symulator pobierania, do którego to użytkownik ma dostęp z poziomu okna testowania animacji pojawiającego się po kompilacji. Użytkownik ma możliwość wybrania jednej z predefiniowanych wartości charakteryzującej przepustowość łącza, natomiast jeśli nie znajduje się na tej liście pożądana opcja, istnieje możliwość samodzielnego dostosowania parametrów z dokładnością do pojedynczego bajta na sekundę. Następnie używając opcji symulacji pobierania, animacja zamknie się i automatycznie włączy ponownie, lecz będzie widoczny jedynie biały ekran w okresie równym czasowi pobierania jej przy założeniu, że przepustowość łącza jest taka jak uprzednio ustalona. Funkcja ta ma duże znaczenie, gdyż pomaga twórcom na oszacowanie czasu po jakim jego aplikacja zacznie się wyświetlać na ekranie przy różnych parametrach Internetu użytkownika, tak więc pomaga niwelować jedną z największych wad plików w formacie wynikowym programu Flash. Kolejnym istotnym elementem jest proste narzędzie do profilowania przepustowości dla wynikowej animacji. Za jego pomocą możliwa jest analiza, jak bardzo poszczególna klatka projektu obciąża łącze o zadanej przepustowości i ile dokładnie potrzebuje czasu na załadowanie. Do narzędzia ma się dostęp, podobnie jak w przypadku symulatora pobierania, z poziomu okna testowego dostępnego po kompilacji. Istotnym usprawnieniem pracy jest panel składników i ustawień predefiniowanych ruchu. Znajdują się tam przygotowane przez twórców programu elementy gotowe go wykorzystania w tworzonej animacji, a każdy z nich posiada estetyczną oprawę graficzną. Przykładowe elementy w obrębie składników to:

- lista
- narzędzie wyboru koloru
- pasek postępu ładowania

- pole tekstowe
- przycisk
- lista
- suwak

Natomiast w ustawienia predefiniowanych ruchu znajdują się efekty, które nakłada się na istniejące obiekty, będące uprzednio przekonwertowane na symbol. Posiadają one z góry ustalone zasady ruchu wybranego obiektu wraz z wpływem na ich kształt, lecz możliwe jest dodawanie własnych zasad ruchu. Przykładowymi możliwymi do wyboru efektami są:

- dym
- fala
- spirala-3D
- wirowanie

Ponadto możliwe jest nakładanie filtrów na poszczególne obiekty, za pomocą których możliwe jest dodawanie przykładowo cieni, rozmycia czy blasku. Każdy z filtrów posiada szereg, ustawień dzięki którym możliwa jest precyzyjna manipulacja poszczególnymi efektami. Używanie ich jest niezwykle proste i intuicyjne, daje przy tym znaczne możliwości.

Program Flash jest kompatybilny z formatami, w jakich zapisywane są projekty w programach Adobe Photoshop i Illustrator, daje to możliwość importowania do biblioteki, czy też bezpośrednio na stół montażowy plików graficznych stworzonych w tych aplikacjach wraz z podziałem na warstwy, konkretne obiekty, pola tekstowe z zapisanym formatowaniem, efekty czy nawet przezroczystości. Jest to o tyle istotne, że daje możliwość zrobienia czy to pojedynczych obiektów, czy całych szablonów stron internetowych w jednym z tych dwóch środowisk graficznych, a następnie zaimportowanie do projektu.

Wynikowy projekt można publikować do standardowego formatu odtwarzanego w Flash Player. Ponadto użytkownik ma możliwość uzyskania pliku o rozszerzeniu .swf, czy też spłaszczenia obrazu do jednego obrazka, jak również publikacji do pliku HTML gotowego do umieszczenia na stronie internetowej, pliku wykonywalnego w systemach operacyjnych z rodziny Microsoft Windows (.exe) i Mac OS (.app). W przypadku plików .swf, HTML i graficznych użytkownik ma możliwość dokonywania szeregu ustawień.

3.3. Blender

Jest to wolne oprogramowanie służące do renderowania obrazów statycznych, jak i animacji. Zostało stworzone przez programistów holenderskiej firmy Neo Geo. Kierownikiem całego przedsięwzięcia jak i głównym programistą był i jest do dziś Ton Roosendaal, założyciel Neo Geo. Program został napisany w językach C, C++ jak i Python. Po raz pierwszy został opublikowany w 1995 roku, będąc w dużej mierze połączeniem gotowych już narzędzi przeznaczonych do robienia grafiki 3D. Mimo, że od początku użytkowanie samego programu było nieodpłatne, to do 2002 roku, aby posiadać program wraz z pełną dokumentacją trzeba było zapłacić, jak również za uzyskanie wsparcia technicznego. W 2002 roku po ogłoszeniu upadłości przez właściciela Blendera, został on odkupiony przez Blender Foundation, będącą organizacją typu non-profit, przewodniczącym której jest Ton Roosendaal. Po przejściu program stał się całkowicie darmowy i jest publikowany na licencji GPL (*General Public License*) [16].

Może być używany na wielu systemach operacyjnych, jak na przykład Microsoft Windows, Linux, Mac OS X. Aby lepiej wykorzystać zasoby komputera i tym samym zmniejszyć czas renderowania poszczególnych klatek, Blender jest wydawany w dwóch wersjach, jedna przeznaczona dla systemów 32 bitowych, natomiast druga dla 64 bitowych. Fakt ten może mieć duże znaczenie już w przypadku kilkuminutowych animacji, których to czas renderowania nierzadko przekracza kilka dni, a nawet tygodni na komputerach wysokiej klasy i w zależności liczby zastosowanych efektów, jak ich jakości i szczegółowości obiektów.

Pomimo znacznych wad dotyczących przejrzystości interfejsu użytkownika, który to został gruntownie zmodernizowany i poprawiony dopiero od wersji 2.5 włącznie (jej wersja stabilna została wydana w 2010 roku), Blender od samego początku ciągnie za sobą wielu zwolenników, a pracują w nim nie tylko amatorzy animacji i grafiki 3D, ale również profesjonaliści, chociażby tacy, którzy zajmują się tworzeniem komercyjnych materiałów promocyjnych. Wielu użytkowników wybiera właśnie ten program ze względu na wspieranie i propagowanie idei wolnego oprogramowania, lecz bezwątpienia silnym bodźcem jest sam fakt możliwości posiadania tak rozbudowanej aplikacji bez konieczności płacenia za nią. Wszystkie inne powszechnie znane programy przeznaczone do tworzenia grafiki 3D są płatne, a koszt jaki trzeba ponieść w związku z zakupieniem licencji jest na tyle duży, że przekracza możliwości finansowe większości osób fizycznych jak i małych firm. Istnienie takiego programu jak Blender, będącego całkowicie wolnym spowodował

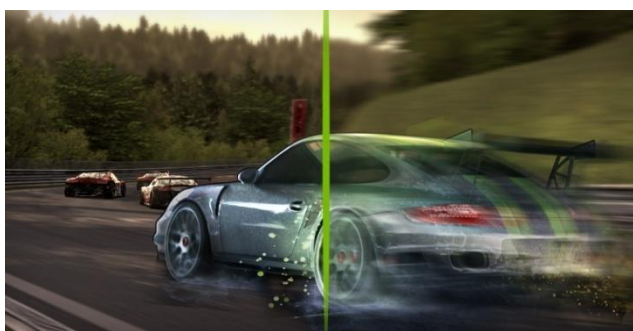
znaczny wzrost zainteresowania tą dziedziną, czego owocem jest wielu utalentowanych artystów. Ponadto dzięki temu nieustannie rośnie liczbą osób będących specjalistami w obsłudze tego programu, dzięki czemu stanowią znaczące wsparcie dla osób rozpoczynających naukę tej dziedziny.

Blender posiada wszystkie funkcje, które są potrzebne do stworzenia profesjonalnej animacji, a nawet małe gry 3D. Ponadto wraz z wydawaniem nowych wersji twórcy sukcesywnie zwiększają możliwości programu. Wśród tych bardziej popularnych wyróżnić można: Anti-Aliasing (czyli wygładzanie krawędzi nienaturalnie ostrych, dając efekt jak gdyby były rzeczywiste (Rys. 5.)), Motion Blur (renderowany obraz, czy też pojedyncza klatka animacji są zawsze statyczne, nawet jeśli przedstawiają ruch, funkcja ta tworzy smugi za sobą tak jak gdyby obiekt szybko się poruszał, znacznie zwiększając stopień realności (Rys. 6.)).



Rys. 5. Idea funkcji Anti-Aliasing

Źródło: <http://i.stack.imgur.com/pA7uy.png>



Rys. 6. Idea funkcji Motion Blur

Źródło: http://www.club-3d.com/tl_files/club3d/uploads/en/content/Graphics%20cards/Nvidia%20600%20Series/GTX

Ray Tracing (czyli sposób renderowania obrazu bazujący jedynie na promieniach generowanych przez źródło światła, które trafiają do kamery, jest to metoda bardzo złożona obliczeniowo, zużywająca znaczne zasoby komputera, lecz może dawać niezwykle rzeczywisty efekt, poprawiając przy tym w znacznym stopniu aspekty wizualne (Rys. 7.)). Wsparcie dla Motion Capture (czyli odzwierciedlanie naturalnych ruchów osoby, która ma założony specjalny kombinezon, na podstawie przyczepionych do niego markerów, przekazujących informacje poprzez obraz z kamer do komputera o dokładnym położeniu każdego z markerów (Rys. 8.)).



Rys. 7. Wynik zastosowania funkcji Ray Tracing

Źródło: <http://www.cs.utah.edu/~wmartin/pics/rtrt/teapot3.large.gif>



Rys. 8. Zastosowanie i rezultat Motion Capture

Źródło: <http://jordandiamond5.files.wordpress.com/2010/09/polar-mocap.jpg>

Blender posiada zaimplementowane funkcję obsługujące fizykę obiektów znajdujących się na scenie. Każdemu z nich można nadać odpowiednie właściwości, tak aby zachowywał się jak dym, tkanina, ładunek wybuchowy, ciecz czy ciało elastyczne wraz ze szczegółowym ustawieniem parametrów odpowiadającym za różne cechy. Ponadto możliwe jest definiowanie kolizji w stosunku do poszczególnych obiektów, tak aby się nie przenikały, odbijały lub deformowały. Za pomocą różnego rodzaju parametrów możliwe jest ustalenie takich elementów jak na przykład sprężystość, gęstość, ciężar, przyspieszenie grawitacyjne z jakim będzie spadał obiekt. Dane obiekty mogą być również emiterami, czyli z nich wydobywać się będzie woda, dym, powietrze, czy też różnego rodzaju cząsteczki. Innymi istotnymi elementami jest możliwość nadawania właściwości takich jak transparentność, odbicia jak lustro, chropowatość powierzchni, przenikalność promieni jak i wiele innych. Każdemu z elementów na scenie można nadać tekstury o różnorodnej charakterystyce.

Jednym z bardziej istotnych elementów, lecz przeznaczonych z założenia dla zaawansowanych użytkowników, znających podstawy programowania jest konsola skryptowa. W Blenderze istnieje możliwość tworzenia własnych skryptów pisanych w języku Python. Można w ten sposób stworzyć logikę zaimplementowaną w animacji. Autorzy wyposażyli swoją aplikację w silnik graficzny przeznaczony do tworzenia gier z pełną obsługą OpenGL, posiadający dwa silniki obsługujące fizykę, czy też obsługę

dźwięku. W połączeniu z logiką zaprogramowaną w skrypcie i z wykorzystaniem urządzeń peryferyjnych, jak na przykład myszki i klawiatury, możliwe jest stworzenie prawdziwej gry o ograniczonych możliwościach, lecz z zaimplementowaną fizyką.

Użytkownik ma możliwość wyrenderowania pojedynczej klatki do różnych formatów obrazu, tak samo jak i animacje mogą być zapisane z zastosowaniem różnego rodzaju kodowania wraz z dostępnością do jego wielu parametrów. Rozmiar maksymalnego wyjściowego obrazu to 10000x10000 pikseli, natomiast rozdzielczość w standardzie HDTV 1080p to 1920x1080, tak więc ta oferowana przez program Blender jest znacząca. Maksymalna długość pojedynczej animacji to 300000 klatek, przy założeniu typowego standardu, czyli 25 klatek na sekundę, daje to 200 minut. W stosunku do jednego pliku jest to bardzo duża wartość. Typowym formatem zapisu całej sceny jest .blend, lecz program ma możliwość importowania jak i eksportowania samych obiektów w formie pliku .obj, jest to niezwykle istotne, gdyż format ten jest szeroko wykorzystywany i kompatybilny z wieloma innymi aplikacjami.

3.4. FlashDevelop

Jest to wolne oprogramowanie pełniące rolę zintegrowanego środowiska programistycznego (*IDE*). Zostało stworzone przez dwoje programistów, Philippe Elsass i Mika Palmu w języku C#. Pierwsza wersja została upubliczniona w 2005 roku. Założeniem twórców było wydanie darmowego środowiska za pomocą którego stworzyć można aplikacje analogiczne do tych jak przy pomocy komercyjnego programu Adobe Flash. Z powodu wykorzystania przez twórców języka C# w połączeniu z platformą programistyczną .NET Framework 2 oficjalnie środowisko to można wykorzystywać jedynie w systemach operacyjnych Microsoft Windows, począwszy od Windows XP

FlashDevelop udostępniony jest na licencji MIT [17], nazwa pochodzi od Instytutu Technologicznego w Massachusetts (*Massachusetts Institute of Technology*), będącego autorem tej licencji. Została opublikowana w 1988 roku, mając na celu rozprzestrzenianie i promowanie idei otwartego oprogramowania (*open source*). Przejawia się to w postanowieniach licencji MIT, które stanowią o całkowitym zrzeczeniu się do praw majątkowych z oprogramowania nią objętego, natomiast każdy użytkownik ma prawo do modyfikowania samego programu oraz kodu według własnego uznania, jak i kopiowania

czy rozpowszechniania. Jedynym warunkiem jest, aby zawsze we wszystkich egzemplarzach aplikacji zachowane były postanowienia licencji [18].

Środowisko FlashDevelop wraz z frameworkiem Adobe Flex SDK stanowi narzędzie gotowe do tworzenia aplikacji pisanych w ActionScript 3, jak i w MXML. Zastosowanie wieloplatformowego języka programowania Haxe daje możliwość kompilacji projektu do kodu JavaScript, ActionScript 3, Neko, PHP, C++, C# i Javy [19]. Natomiast w przypadku wykorzystania kompilatora MTASC (*Motion-Twin ActionScript 2 Compiler*) użytkownik ma możliwość tworzenia aplikacji opartych o ActionScript 2. Każda z trzech powyższych metod umożliwia budowanie aplikacji jak i stron internetowych analogicznych do powstałych z wykorzystaniem narzędzia Adobe Flash. Zarazem Framework Adobe Flex SDK, Haxe jak i MTASC dostępne są na zasadzie wolnego oprogramowania. Natomiast w przypadku posiadania zainstalowanego programu Adobe Flash podczas budowania projektu FlashDevelop automatycznie włączy Flash wykonując kod użytkownika na wybranej scenie.

FlashDevelop posiada wiele cech wpływających na wysoki komfort tworzenia kodu aplikacji. Jednym z najważniejszych z nich jest rozbudowany system podpowiedzi i autouzupełniania kodu. Na ekranie wyświetlane są informacje na temat dostępnych metod dla danego typu obiektu znajdujących się w zaimportowanych pakietach wraz z ich opisem. Między innymi użytkownik jest informowany o liczbie i typie parametrów przyjmowanych jak i zwracanych przez daną metodę. Innym istotnym elementem jest rozwinięte kolorowanie składni kodu. Aby zadbać o optymalizację i dobrą wydajność tworzonego projektu zostało dodane narzędzie do profilowania FlashDevelop Profiler, dzięki któremu możliwe jest znalezienie miejsc w kodzie, które zabierają duże ilości zasobów komputera jak i czasu. Istotnym elementem jest modułowość, która pozwala na implementację zewnętrznych pakietów, wydanych przez autorów aplikacji, czy też samych użytkowników.

Twórcy mając na celu oddanie w ręce użytkownika aplikacji, za pomocą, której stworzyć można nie tylko aplikacje desktopowe czy webowe, lecz także strony internetowe. Dowodem czego jest możliwość tworzenia plików z gotowymi szablonami wraz z pełnym wsparciem autouzupełniania kodu wraz z kolorowaniem składni dla następujących języków [17]:

- HTML
- CSS
- PHP

- XML

Ponadto istnieje możliwość utworzenia nowego pustego pliku tekstowego, a następnie ustawienia dla niego jednej ze składni, które wspiera FlashDevelop wraz z typowym dla wybranego języka kolorowaniem i autouzupełnianiem zawierającym podstawowy zasób informacji. Prócz uprzednio wspomnianych możliwe jest wybranie jednej ze składni:

- Batch
- C++
- C#
- Properties
- Python

3.5. GIMP

Jest to wolne oprogramowanie służące do edycji i tworzenia obrazów rastrowych wszelkiego rodzaju. Nazwa GIMP jest akronimem od słów *GNU Image Manipulation Program*, choć początkowo oznaczała *General Image Manipulation Program*. Pierwszą wersję napisał dwóch programistów: Spencer Kimball i Peter Mattis wydając ją na początku 1996 roku, dopiero później został powołany cały zespół osób pracujących nad rozwijaniem tego projektu. Kod źródłowy programu został napisany w języku C, natomiast interfejs graficzny w GTK+ (*GIMP Toolkit*), który został specjalnie stworzony przez tych samych twórców również w języku C. GIMP jak i GTK+ są wciąż rozwijane, publikowane na zasadzie wolnego oprogramowania i co istotne wieloplatformowe. Mogą być używane zarówno na systemach Microsoft Windows, Linux, Mac OS i innych.

GIMP od samego początku był wydawany na licencji GPL (*General Public License*) [20], która to po raz pierwszy została wydana w roku 1989. Wiele z aplikacji, których twórcy wydają je na zasadzie wolnego oprogramowania decyduje się właśnie na ten typ licencji.

Aby być zaklasyfikowana jako wolne oprogramowanie konieczne było spełnienie czterech warunków. Otóż:

- Programy nią objęte można uruchamiać w dowolnym celu.
- Programy nią objęte można rozpowszechniać.

- Programy nią objęte można analizować w jaki sposób działa i modyfikować w celu dostosowania do własnych potrzeb.
- Programy nią objęte można samemu udoskonalać, jak i rozpowszechniać własne udoskonalenia.

Autorzy podczas tworzenia GIMPa mając na uwadze różnorodność artystów i ich zapotrzebowań stworzyli aplikacje modułową, czyli taką, że każdy użytkownik może samodzielnie dodawać wtyczki wprowadzając tym samym nowe funkcje lub modyfikować te istniejące bez konieczności ingerencji w kod źródłowy całej aplikacji, co łatwo może spowodować destabilizację działania programu, a nawet doprowadzić do tego, że całkowicie przestanie działać. Liczba stworzonych wtyczek jest ogromna, a wiele z nich daje możliwość tworzenia bardzo zaawansowanych efektów, lub pomaga w organizacji projektu. Wśród nich znaleźć można niezwykle ważny między innymi w przypadku usuwania rastrów filtr FFT (*Fast Fourier Transform*), czy zmieniający interfejs graficzny na podobny do tego jaki jest w programie Adobe Photoshop, co jest istotne dla osób pracujących z Photoshopem i znających jego strukturę i funkcjonalność. Innym przykładem modułów zewnętrznych może być taki, który umożliwia operacje na formacie RAW, będącego obrazem nieskompresowanym o najwyższej możliwej jakości jaki tworzy aparat cyfrowy po zrobieniu zdjęcia. Użytkownik może pisać własne wtyczki w języku C, skrypty w Scheme, Perl i Tcl, natomiast rozszerzenia w języku Python.

Z wykorzystaniem narzędzi dostępnych w programie GIMP użytkownik ma możliwość poprawiania jakości zdjęć jak i ich przerabiania, zmiany proporcji kolorów podstawowych w celu równoważenia zachwianej proporcji, lub świadomej destabilizacji proporcji; natomiast w przypadku tworzenia własnych dzieł możliwa jest kreacja nawet niezwykle zaawansowanych grafik czego przykładem jest wielu znanych artystów.

Doskonałym wzbogaceniem funkcjonalności programu, które znacząco poszerzają spektrum możliwości są filtry, których to obecnie dostępnych jest około stu. Wykonują wiele operacji wykorzystujących złożone obliczenia matematyczne takie jak całkowanie czy różniczkowanie. Za ich pomocą możliwe jest wpływanie na kolorystykę obrazu lub jego geometrię. Przykładami mogą być:

- Erozja
- Farba olejna
- Film rysunkowy
- Lekki żar

- Nałożenie płótna
- Rozmycie Gaussa
- Stara fotografia
- Usuwanie efektu czerwonych oczu
- Wyostrenie

Ponadto program GIMP cechuje się dużą różnorodnością w kwestii formatu obsługiwanych obrazów. Standardowym formatem w jakim zapisywane są projekty jest .xcf, lecz możliwe jest operowanie na większości znanych i ogólnie wykorzystywanych rozszerzeniach, włącznie z .psd, będącym formatem Adobe Photoshop.

3.6. Inno Setup

Jest to wolne oprogramowanie służące do tworzenia instalatorów. Zostało stworzone przez programistę Jordana Russella, który to pod wpływem niezadowolenia z programu InstallShield Express dołączonego przy zakupie środowiska Borland Delphi postanawia stworzyć własny program pozwalający na stworzenie bardziej zaawansowanego instalatora. Pierwsza wersja została wydana w 1997 roku, a do napisania jej wykorzystał język Delphi przy zastosowaniu środowiska programistycznego Borland Delphi [21]. Inno Setup do dziś jest rozwijany cały czas przez tego samego programistę, lecz ze względu na otwarty dostęp do kodu źródłowego powstawało wiele usprawnień. Martijn Laan był twórcą jednego z nich, które okazało się wносить tak znaczące możliwości, że Jordan Russella w 2003 roku postanowił włączyć efekt pracy Laana do całości projektu i nadać mu częściowe prawa autorskie.

Inno Setup dostępny jest jedynie na systemy operacyjne Microsoft Windows, począwszy od Windows 2000 na systemach 32 bitowych jak i 64 bitowych. Program daje użytkownikowi dwie metody na tworzenie instalatorów.

Pierwsza przeznaczona jest głównie dla osób nie obeznanych z programowaniem, polega na przejściu przez kolejne kroki kreatora i stworzenie odpowiedniego dla siebie instalatora. Jednakże zakres dostępnych możliwości w tym przypadku jest znacząco ograniczony.

Druga metoda jest kierowana przede wszystkim do osób mających przynajmniej podstawowy zakres wiedzy na temat programowania. W tym przypadku należy napisać odpowiedni skrypt, który po kompilacji utworzy instalator o właściwościach zgodnych

z zawartymi dyrektywami w treści skryptu. Niesie to za sobą znacznie większe możliwości aniżeli te, które dostępne są w poprzednim sposobie. Ponadto tę metodę podzielić można na kolejne dwie części.

Pierwsza z nich odnosi się do tworzenia sekcji, które to wypełnia się dostępnymi dyrektywami wraz z podaniem odpowiednich parametrów (Rys. 9.).

```
[Setup]
; Sekcja zawierająca ogólne ustawienia dotyczące instalatora i deinstalatora
AppName=Ziemia w kosmosie
; Nazwa aplikacji widoczna podczas instalacji
AppVerName=Ziemia w kosmosie 1.0.3
; Pełna nazwa wersji programu, zazwyczaj składająca się z nazwy aplikacji + numeru wersji
AppVersion=1.0.3
; Wersja aplikacji
DefaultDirName={pf}\Ziemia w kosmosie
; Domyślny katalog instalacji
DefaultGroupName=Ziemia w kosmosie
; Nazwa grupy w Menu Start
UninstallDisplayIcon={app}\Data\Lib\ico.dll
; Ikona deinstalatora w Dodaj/Usuń Programy
Compression=lzma2
; Standard kompresji
SolidCompression=yes
; Kompresowanie całego projektu na raz, a nie osobno wszystkich plików
OutputBaseFilename=Ziemia_setup
; Wynikowa nazwa instalatora
OutputDir=E:\MATERIAŁY NAUKOWE\PRACA INŻYNIERSKA\Instalator\instalator docelowy
; Lokalizacja instalatora otrzymanego po komplikacji
AppPublisher=Rafał Zieliński
; Wyświetla podaną nazwę wydawcy w sekcji pomocy w oknie Dodaj/Usuń Programy
UserInfoPage=yes
; Dodaje w instalatorze kartę polami do wprowadzenia nazwy użytkownika i organizacji

[Languages]
; Sekcja odpowiadająca za definiowanie języków instalatora
Name: en; MessagesFile: "compiler:Default.isl"
Name: pl; MessagesFile: "compiler:Languages\Polish.isl"
Name: ru; MessagesFile: "compiler:Languages\Russian.isl"
Name: cns; MessagesFile: "compiler:Languages\ChineseSimplified.isl"
```

Rys. 9. Fragment kodu programu Inno Setup zawierający sekcje i dyrektywy.

Druga polega na pisaniu w obrębie specjalnej sekcji "[Code]" skryptowego kodu o składni ściśle bazującej na języku Pascal. Daje to ogromne możliwości, gdyż użytkownik jest w stanie stworzyć własne szablony instalatora dodając nowe karty czy modyfikując istniejące, ale co istotniejsze, wykorzystując funkcjonalność jaką daje język programowania, możliwe jest wykonywanie zdarzeń w zależności od innych czynników (czyli stosowanie wyrażeń warunkowych), używanie pętli, zdarzeń, funkcji czy też zmiennych (Rys. 10.).

```

[Code]
// Sekcja wprowadzania kodu
var
    currentVersion, currentPluginVersion: string;
    // Tworzenie zmiennych globalnych

function IsDotNetDetected(version, versionPlugin: string): boolean;
// Nagłówek funkcji

var
    keyPath, currentStableVersion, currentPluginStableVersion: string;
    // Tworzenie zmiennych lokalnych

begin
    keyPath := 'SOFTWARE\Macromedia\FlashPlayer';
    RegQueryStringValue(HKEY_LOCAL_MACHINE, keyPath, 'CurrentVersion', currentVersion);
    RegQueryStringValue(HKEY_LOCAL_MACHINE, keyPath + 'Plugin', 'Version', currentPluginVersion);

    currentStableVersion := Copy(currentVersion, 1, Pos(',',currentVersion) + 1);
    currentPluginStableVersion := Copy(currentPluginVersion, 1, Pos('.',currentPluginVersion) + 1);

    result := (currentStableVersion >= version) or (currentPluginStableVersion >= versionPlugin);
    // Zwrócenie wartości funkcji typu Boolean na podstawie porównania leksykograficznego
end;

```

Rys. 10. Fragment kodu programu Inno Setup zawierający fragment sekcji [Code].

Program Inno Setup ma możliwość odnoszenia się bezpośrednio do rejestrów systemu operacyjnego dzięki czemu możliwe jest chociażby sprawdzanie obecności zainstalowanych komponentów i ich wersji, będących niezbędne do poprawnego działania aplikacji do której tworzony jest instalator, również istnieje możliwość modyfikowania istniejących rejestrów czy też dodawania nowych. Poza podstawowymi informacjami jak nazwa i wersja aplikacji będącej w instalatorze, lokalizacji piku wynikowego, czy ścieżki instalacji, użytkownik ma możliwość definiowania własnych ikon pliku instalacyjnego i deinstalacyjnego, tworzenie skrótów w niemalże dowolnym miejscu, uruchomienie programu wraz po zalogowaniu się do systemu, wybór algorytmu kompresji źródeł i jakości, szyfrowania źródeł czy też architektury systemu operacyjnego na którym możliwe będzie włączanie instalatora, nałożenie zadanego hasła na instalator jak i wiele innych możliwości.

4. Program Ziemia w kosmosie

4.1. Cele i założenia programu

Program *Ziemia w kosmosie* jest aplikacją pełniącą rolę dydaktyczną. Ma za zadanie pomóc użytkownikowi w zrozumieniu różnic między modelem geocentrycznym, a heliocentrycznym, wraz z przyswojeniem podstawowych faktów historycznych związanych z ich opracowaniem i dalszym rozwojem. Ponadto na jego podstawie dowiedzieć się można podstawowych informacji na temat poszczególnych planet wchodzących w skład Układu Słonecznego i kilku innych ciał niebieskich. Program zbudowany jest w postaci interaktywnej, aby osoba z niego korzystając wykazała większe zainteresowanie. Przejawia się to tym, że na niektóre elementy można nacisnąć aby wyświetliły się informacje, a inne elementy są ruchome. Aby aplikacja była bardziej atrakcyjna i pobudzała więcej zmysłów odbiorcy zostało dodane tło muzyczne, które użytkownik ma możliwość wstrzymać. Jest to utwór "Billions And Billions" wykonany i stworzony przez amatorskiego kompozytora pochodzącego z Litwy tworzącego muzykę określaną gatunkiem space ambient o pseudonimie Stellardron.

4.2. Układ heliocentryczny

Jest to pierwszy z podprogramów wchodzących w skład całej aplikacji *Ziemia w kosmosie*. Przedstawione zostały w rzucie boczny grafiki reprezentujące kolejno:

- Słońce
- Merkury
- Wenus
- Ziemia
- Mars
- Jowisz
- Saturn
- Uran
- Neptun
- Pluton

Po przyciśnięciu lewym przyciskiem myszy na każdym z obiektów wyświetlają się podstawowe informacje o nim. W początkowym widoku znajdują się informacje dla Słońca. Użytkownik dowiadyuje się na podstawie zawartych danych następujące informacje (wraz z jednostkami w jakich są wyrażone):

- Rok gwiazdowy (dni)
- Odległość od Słońca (km)
- Średnica (km)
- Masa (kg)
- Powierzchnia (km²)
- Przyspieszenie grawitacyjne ($\frac{m}{s^2}$)
- Gęstość ($\frac{g}{cm^3}$)
- Naturalne satelity
- Data odkrycia

4.3. Porównanie układów

Jest to drugi z podprogramów wchodzący w skład projektu. Po uruchomieniu przedstawia wizualne porównanie układu heliocentrycznego z układem geocentrycznym. Model geocentryczny oparty jest na teorii Ptolemeusza pochodzącej z II wieku n.e. wykorzystującym idee epicyklów i deferentów, która to była traktowana niezmiennie do XVI wieku jako jedyna słuszna. Natomiast model heliocentryczny oparty jest na teorii Mikołaja Kopernika. Po kliknięciu na jeden z układów wyświetla się nowe okno zawierające jego podstawowy rys historyczny. Wielkości poszczególnych obiektów, ich czas gwiazdowy jak i czas potrzebny do obejścia całego epicyklu są wzorowane na rzeczywistych danych, lecz z powodów praktycznych jest niemożliwe odwzorowanie danych w sposób bezpośredni, dlatego też wartości te w podprogramie są zmodyfikowane. Występuje on jako drugi w kolejności po modelu układu heliocentrycznego specjalnie, aby użytkownik zaznajomił się wpierw z tym obecnie obowiązującym, a dopiero następnie mógł oba ze sobą porównać.

4.4. Parametry Układu Słonecznego

Kolejny podprogram służy do obliczania różnego rodzaju parametrów poszczególnych obiektów. Użytkownik podaje nową średnicę Słońca wyrażoną w metrach, po naciśnięciu przycisku "Oblicz" zostają wykonane obliczenia i umieszczone w polach tekstowych od razu na wszystkich zakładkach. Wszystkie parametry podzielone są na pięć zakładek (Tab.2.).

Tab.2. Zestawienie parametrów wraz z ich jednostkami w poszczególnych zakładkach.

| Zakładka | Parametr | Jednostka |
|-----------------------|-----------------------------|---------------------|
| Układ Słoneczny | Średnica | <i>metr</i> |
| | Masa | <i>kilogram</i> |
| Układ Słoneczny (2) | Średnia odległość od Słońca | <i>metr</i> |
| | Przyspieszenie grawitacyjne | $\frac{m}{s^2}$ |
| Światło | Prędkość światła | $\frac{m}{s}$ |
| | Rok świetlny | <i>kilometr</i> |
| Odległości w kosmosie | Odległość od Słońca | <i>rok świetlny</i> |
| Rozmiary gwiazd | Średnica | <i>metr</i> |

Po lewej stronie każdego z pól tekstowych, w których pojawiają się nowe wartości znajduje się liczba reprezentująca naturalną wartość danego parametru (Tab.3., Tab.4.). Jako, że różne źródła nie podają zgodnych ze sobą prawdziwych wartości charakteryzujących poszczególne wartości, dlatego też należy traktować je jako przybliżone. Nie jest możliwe jednoznaczne stwierdzenie, które z wartości proponowanych przez różne osoby są zgodne z rzeczywistymi, gdyż do określania poszczególnych parametrów obiektów znajdujących się w kosmosie używa się różnego typu urządzeń, jak i wyników wielu obserwacji, na które wpływ ma znaczna liczba czynników jak na przykład warunki atmosferyczne, ponadto każde urządzenie dysponuje skończoną dokładnością, co skutkuje pojawianiem się błędów pomiarowych, kolejnym elementem jest tak zwany czynnik ludzki, czyli błąd powstały z powodu niedokładności ludzkiej. Jednak różnice są na tyle małe, że dla ogólnego zrozumienia skali możliwe jest przyjęcie większości z nich, aby osoba poznająca te zagadnienia mogła na ich podstawie zbudować poprawne wyobrażenia.

Tab.3. Zestawienie wartości naturalnych poszczególnych parametrów wykorzystanych w podprogramie.
[22][23][24][25][26]

| Obiekt | Średnica | Masa | Średnie odległości od Słońca | Przyspieszenie grawitacyjne |
|---------------|-----------------|-------------------|-------------------------------------|------------------------------------|
| Słońce | 1391000 | $1,989 * 10^{30}$ | - | 274,4345 |
| Merkury | 4878 | $3,302 * 10^{23}$ | 57910006,85 | 3,7047 |
| Wenus | 12104 | $4,869 * 10^{24}$ | 108199995,2 | 8,8710 |
| Ziemia | 12756 | $5,977 * 10^{24}$ | 149599951,1 | 9,8056 |
| Mars | 6787 | $6,419 * 10^{23}$ | 227939920,2 | 3,7198 |
| Jowisz | 142800 | $1,899 * 10^{27}$ | 778330257,4 | 24,8550 |
| Saturn | 120660 | $5,685 * 10^{26}$ | 1429400029 | 10,4234 |
| Uran | 51118 | $8,681 * 10^{25}$ | 2870989229 | 8,8687 |
| Neptun | 49528 | $1,024 * 10^{26}$ | 4504299580 | 11,1471 |
| Pluton | 2324 | $1,25 * 10^{22}$ | 5913000000 | 0,6178 |

Tab.4. Zestawienie wartości naturalnych poszczególnych parametrów wykorzystanych w podprogramie.
[27][28]

| Parametr | Obiekt | Wartość naturalna |
|--|---------------------|------------------------------|
| Światło | Prędkość światła | 299792458 |
| | Rok świetlny | $9,4607 * 10^{12}$ |
| Odległość od Słońca | Alfa Centauri | 4,39 |
| | Syriusz | 8,61 |
| | Procyon | 11,46 |
| | Ypsilon Andromedae | 43,9 |
| | Almak | 355 |
| | Galaktyka Andromedy | $2,52 * 10^6$ |
| | Galaktyka Trójkątna | $2,92 * 10^6$ |
| | Średnica | Gwiazda neutronowa (Geminga) |
| Biały karzeł (Syriusz B) | | 11800 |
| Błękitny nadolbrzym (Rigel) | | 104607360 |
| Czerwony nadolbrzym (Antares) | | 1182867840 |
| Czerwony hiperolbrzym (VY Canis Majoris) | | 2712450000 |

Większość z parametrów obliczana jest na zasadzie przemnożenia odpowiedniej reguły (1) przez poszczególne wartości naturalne. W skład nich wchodzi:

- Średnica (zakładka Układ Słoneczny)
- Średnia odległość od Słońca
- Prędkość światła
- Rok świetlny
- Odległość od Słońca
- Średnica (zakładka Rozmiary gwiazd)

$$p = \frac{nr}{r} \quad (1)$$

gdzie:

p - proporcja

nr - wprowadzona przez użytkownika nowa średnica Słońca

r - naturalna średnica Słońca

Natomiast masa poszczególnych obiektów liczona jest wzorem (2) uwzględniającym średnią gęstość.

$$m = \rho * \frac{4}{3} * \pi * r^3 \quad (2)$$

gdzie:

ρ - gęstość

r - nowa średnica

Z kolei poszczególne wartości przyspieszenia grawitacyjnego liczone są na podstawie wzoru (3) uwzględniającego masę i promień obiektu.

$$a = G \frac{m}{r^2} \quad (3)$$

gdzie:

G - stała grawitacyjna

m - nowa masa

r - nowa średnica

W podprogramie tym do wykonania niektórych obliczeń należało wykorzystać pewne wartości liczbowe, które zostały zaimplementowane w kodzie programu jako stałe (Tab.5.) posiadające przypisane odpowiednie wartości. Jedną z nich jest G, czyli stała grawitacji, która pełni rolę współczynnika proporcjonalności. Kolejną jest mila, będąca jednostką długości. Ze względu na znaczną liczbę odmian, została przyjęta w podprogramie powszechnie stosowana mila międzynarodowa. Wykorzystywana w obliczeniach jest również stała π , będąca w ogólności stosunkiem obwodu koła do jego

średnicy. Ostatnią wykorzystaną wartością jako stałą jest gęstość, dla poszczególnych planet Układu Słonecznego, wraz z Słońcem i Plutonem.

Tab.5. Użyte stałe wraz z ich wartościami i jednostkami. [29]

| Stała | | Wartość | Jednostka |
|--------|---------|--------------------------------|------------------------|
| G | | $6,673848 * 10^{-11}$ | $\frac{m^3}{kg * s^2}$ |
| mila | | 1,609344 | km |
| π | | 3,141592653589793 ¹ | - |
| ρ | Słońce | 1,409 | $\frac{g}{cm^3}$ |
| | Merkury | 5,427 | |
| | Wenus | 5,204 | |
| | Ziemia | 5,515 | |
| | Mars | 3,934 | |
| | Jowisz | 1,326 | |
| | Saturn | 0,687 | |
| | Uran | 1,29 | |
| | Neptun | 1,638 | |
| | Pluton | 2,03 | |

Podprogram ten pomaga w zrozumieniu różnic między parametrami charakteryzującymi poszczególne obiekty w kosmosie. Użytkownik poprzez przeskalowanie całego Układu Słonecznego może stworzyć pełniejsze wyobrażenie na jego temat dzięki, między innymi dzięki wprowadzaniu nowej średnicy Słońca w metrach, co daje możliwość skalowania z jednostkami, z którymi człowiek spotyka się w życiu codziennym.

4.5. Układ Słoneczny na mapie Polski

Jest to podprogram, który ukazuje rozmieszczenie planet Układu Słonecznego na mapie Polski. Jako centralny punkt Układu będący środkiem Słońca, przyjęty został budynek Wydziału Fizyki, Astronomii i Informatyki Stosowanej Uniwersytetu Mikołaja Kopernika, jednocześnie pełniąc rolę średnicy Słońca. Ze względu na niejednorodny kształt budynku, wykorzystana do tego celu została szerokość jednostki od strony ulicy Grudziądzkiej. Użyte mapy, na które zostały naniesione poszczególne ciała niebieskie są zdjęciami satelitarnymi pochodzącymi z serwisu Google Maps.

¹ Jest to wartość liczby π zaimplementowana w bibliotece Math języka Action Script.

Po naciśnięciu lewym przyciskiem myszy na każdy z obiektów, po prawej stronie podprogramu pojawiają się podstawowe informacje (Tab.6.) na jego temat. Ze względu na fakt, że planety skaliste poruszają się ze znaczną prędkością po swoich orbitach to obszar, w którym należy kliknąć myszką, aby daną wybrać został powiększony.

Tab.6. Zestawienie informacji o ciałach niebieskich dostępnych w tym podprogramie.

| Obiekt | Średnica [m] | Masa [kg] | Przyspieszenie grawitacyjne [$\frac{m}{s^2}$] | Odległość od Słońca [m] |
|---------------|---------------------|------------------|---|--------------------------------|
| Słońce | 88 | $5,0311 * 10^9$ | $1,7343 * 10^{-5}$ | - |
| Merkury | 0,3086 | 84,6356 | $2,3724 * 10^{-7}$ | 3663,6093 |
| Wenus | 0,7657 | 1201,3592 | $5,4694 * 10^{-7}$ | 6845,1471 |
| Ziemia | 0,807 | 1518,9653 | $6,2265 * 10^{-7}$ | 9464,2672 |
| Mars | 0,4294 | 163,3033 | $2,3646 * 10^{-7}$ | 14420,354 |
| Jowisz | 9,0341 | 513453,97 | $1,6795 * 10^{-6}$ | 49240,16 |
| Saturn | 7,6334 | 159997,04 | $7,3301 * 10^{-7}$ | 90429,333 |
| Uran | 3,2339 | 20542,098 | $5,2435 * 10^{-7}$ | 181629,8 |
| Neptun | 3,1333 | 28509,585 | $7,752 * 10^{-7}$ | 284959,28 |

Aby uzyskać lepszą przejrzystość obiekty podzielone zostały na dwie zakładki:

- Słońce i Planety skaliste:
 - Słońce
 - Merkury
 - Wenus
 - Ziemia
 - Mars
- Gazowe olbrzymy
 - Jowisz
 - Saturn
 - Uran
 - Neptun

Środki wszystkich planet, zostały naniesione na mapę z dużą dokładnością, tak aby odległości ich do środka Słońca były analogiczne z rzeczywistym dystansem odpowiednio przeskalowanym. Natomiast ze względu na zachowanie czytelności podprogramu wielkości poszczególnych obiektów zostały zmienione. Również rok gwiazdowy został uwzględniony na podstawie rzeczywistych danych przy jednakowym założeniu początkowym dla obu zakładek. W przypadku Merkurego i Jowisza,

czyli pierwszych planet skrajnie wewnętrznych dla obu zakładki, czas obiegu po orbicie został przyjęty jako 4,2 sekundy i do niego zostały dostosowane wartości dla Wenus, Ziemi, Marsa, jak i dla Saturna, Uranu, Neptuna (Tab.7.).

Tab.7. Zestawienie czasów obiegu po orbicie poszczególnych planet Układu Słonecznego.

| Planeta | Rok gwiazdowy [s] | Rzeczywisty rok gwiazdowy [dni] |
|----------------|--------------------------|--|
| Merkury | 4,2 | 87,97 |
| Wenus | 10,6 | 224,7 |
| Ziemia | 17,3 | 365,26 |
| Mars | 32,5 | 686,96 |
| Jowisz | 4,2 | 4333,29 |
| Saturn | 10,3 | 10759,5 |
| Uran | 29,5 | 30708,16 |
| Neptun | 57,9 | 60223,35 |

Podprogram ten w zamiarze pomóc ma zrozumieć rozmiar Układu Słonecznego i dystans między poszczególnymi planetami na przykładzie odległości z jakimi ludzie mają do czynienia na co dzień, co powinno czyni go znacznie bardziej przystępnym dla wielu użytkowników. Ponadto fakt wykorzystania mapy Polski powinien dodatkowo ułatwić przyswojenie informacji, gdyż wszystkie planety, przy zastosowaniu takiej skali, przemieszczają się przez jakieś charakterystyczne miejsce w Polsce czy też dla okolic Torunia, lub w bardzo niewielkiej odległości, jak na przykład orbita Uranu wypadająca przy samej Warszawie, czy też Jowisza przy Bydgoszczy.

4.6. Skala w kosmosie

Jest to animacja ukazująca różnicę w wielkości różnych obiektów we Wszechświecie na zasadzie wizualnego porównania obecnie widocznego z kolejnym. W skład ciał niebieskich pojawiających się w animacji wchodzi planety Układu Słonecznego wraz z Plutonem jak i szereg gwiazd wraz ze Słońcem. Wielkości wszystkich zostały odwzorowane z dużą dokładnością na podstawie ogólnie dostępnych informacji o ich średnicach, natomiast kolory jak i struktura gwiazd została dopasowana na podstawie obecnego stanu wiedzy na ich temat (Tab.8.).

Tab.8. Średnice poszczególnych ciał niebieskich. [30]

| Ciało niebieskie | Średnica (km) |
|-------------------------|----------------------|
| Gemina | 38 |
| Pluton | 2324 |
| ZiemińskiKsiężyc | 3476,2 |
| Merkury | 4878 |
| Mars | 6787 |
| Syriusz B | 11684,4 |
| Wenus | 12104 |
| Ziemia | 12756 |
| Neptun | 49528 |
| Uran | 51118 |
| Saturn | 120660 |
| Jowisz | 142800 |
| Proxima Centauri | 196131 |
| Słońce | 1391000 |
| Procyon | 2848768 |
| Polluks | 12240800 |
| Arktur | 36166000 |
| Gwiazda Polarna | 63986000 |
| Kanopus | 90415000 |
| Rigel | 108498000 |
| La Superba | 299065000 |
| Mira Ceti | 542490000 |
| Antares | 1228253000 |
| Betelgeza | 1321450000 |
| VY Canis Majoris | 2712450000 |

Animacja ta pozwala przyswoić i zrozumieć różnice w wielkości poszczególnych obiektów astronomicznych. Wszystkie dwadzieścia pięć ciał niebieskich, które zostały uwzględnione posegregowane są według promienia od najmniejszego do największego. Elementem mającym na celu ułatwienie zrozumienia skali jest gromadzenie się poszczególnych obiektów po lewej stronie póki nie zostaną przykryte przez kolejne, większe. Dzięki temu użytkownik może porównać kilka jednocześnie, a nie jedynie obecny z następującym po nim.

5. Dane techniczne programu *Ziemia w kosmosie*

5.1. Struktura programu

Aplikacja *Ziemia w Kosmosie*, jako całość, posiada strukturę bazującą na wykorzystaniu mniejszych podprogramów. Bezpośrednio po włączeniu programu *Ziemia w kosmosie* użytkownikowi ukazuje się menu, z którego to ma się dostęp do wszystkich podprogramów i animacji. Każdy z nich jak i animacja oraz ścieżka dźwiękowa są osobnymi plikami, które rozmieszczone są w odpowiednich katalogach w całym projekcie. Konstrukcja taka jest z wielu względów wygodna, chociażby z tego faktu, że odciąża główny plik programu. Kolejnym istotnym elementem jest wygoda przy wprowadzaniu poprawek w poszczególnych podprogramach. Natomiast elementem, który nie jest widoczny dla końcowego użytkownika, lecz odgrywa ogromną rolę dla twórcy jest optymalna praca z projektem, gdyż w przypadku zbyt dużego pojedynczego programu środowisko Flash zaczyna spowalniać pracę i całe się przycina. Konstrukcja tego typu niesie za sobą jednak pewien minus, otóż w momencie otwierania danego podprogramu z poziomu menu następuje bardzo krótka przerwa wynikająca z konieczności jego załadowania.

5.2. Wykonanie programu

Większość elementów programu czyli menu, Układ heliocentryczny, Porównanie układów, Parametry Układu Słonecznego i Układ Słoneczny na mapie Polski wykonane przy pomocy narzędzia Adobe Flash, co skutkuje koniecznością posiadania zainstalowanej wtyczki Adobe Flash Player. Wszystkie elementy Flash zostały wykonane przy zastosowaniu częstotliwości równej dwudziestu czterem klatkom na sekundę, co stanowi standard filmów wyświetlanych między innymi w kinie. Ponadto do ich stworzenia został wykorzystany język ActionScript 3.

Modele planet Układu Słonecznego zostały stworzone na potrzeby tego programu w środowisku Blender. Wszystkie w rzeczywistości stanowią przestrzenne modele z naniesionymi teksturami, jednak ze względu na dwuwymiarowy charakter aplikacji poszczególne obiekty zostały wyrenderowane do obrazu obsługującego przezroczystość,

w tym wypadku PNG (*Portable Network Graphics*). Wykorzystane tekstury zostały pobrane z ogólnie dostępnych źródeł i w dużej mierze bazują na zdjęciach wykonanych przez NASA. Dla uzyskania niezwykle realistycznych efektów w przypadku tworzenia Ziemi zostały zastosowane dodatkowe tekstury:

- Kolorowa mapa świata
- Mapa cieni
- Maski odbić
- Mapa chmur
- Nocna mapa światła

Ponadto mapy cieni zostały zastosowane również dla Merkurego, Wenus, Marsa i Plutona.

Animacja Skali w kosmosie została wykonana przy zastosowaniu narzędzia Adobe After Effects. Jest to animacja dwuwymiarowa stworzona w rozdzielczości Full High Definition, czyli 1920 x 1080 pikseli przy zastosowaniu częstotliwości równej dwudziestu pięciu klatkom na sekundę. Podczas tworzenia jej zostało zastosowanych wiele efektów. Jednym z nich jest niezwykle istotny przy tworzeniu animacji Motion Blur, który to został zastosowany podczas przesuwania się kolejnych obiektów. Innymi efektami jakie zostały wykorzystane w stosunku do pojawiających się i zanikających napisów są dekodery i enkodery. Do stworzenia plam słonecznych posłużył efekt Fractal Noise.

5.3. Wymagania sprzętowe

Do poprawnego działania programu komputer powinien spełniać przynajmniej następujące wymagania:

- System operacyjny: z rodziny Microsoft Windows przynajmniej XP, Linux, Mac OS
- Procesor: 2,33GHz zgodny z architekturą 32 bitową (Intel Atom 1,6 GHz w przypadku netbooków)
- Pamięć RAM: 256 MB (1 GB w przypadku netbooków wyposażonych w systemem operacyjny Windows)
- Pamięć GPU: 128 MB
- Wolne miejsce na dysku: 100 MB
- Adobe Flash Player 11.2

6. Podsumowanie

Wielu uczonych od najstarszych dziejów zastanawiało się nad miejscem Ziemi i ludzi we Wszechświecie oraz jaką rolę odgrywa ich byt. Pierwszy, który to miejsce wskazał wraz z naukową argumentacją był Eudoksos z Knidos formułujący w IV wieku p.n.e. teorię sfer homocentrycznych, umieszczając nieruchomo Ziemię w centrum Układu Słonecznego. Zmieniana i doskonalona przez lata doczekała się w II wieku n.e. ostatecznej postaci, wzbogaconej o system epicyklów i deferentów, jako model geocentryczny opracowany przez Ptolemeusza. Idea ta towarzyszyła ludziom przez następnych czternaście wieków do czasów, gdy Mikołaj Kopernik publikuje swoje rewolucyjne dzieło *De revolutionibus orbium coelestium*, w którym to spycha rolę Ziemi do zwykłej planety, krążącej po kołowej orbicie wokół Słońca. Choć początkowo idee te zostały przyjęte przez kościół pozytywnie to z czasem odrzucono je jako sprzeczne z filozofią i wszelkimi prawdami moralnymi, natomiast samo dzieło astronoma zostało wpisane na indeks ksiąg zakazanych. Dopiero na początku XVIII wieku, po udowodnieniu istnienia paralaksy gwiazd, teoria Kopernika zaczęła stawać się coraz bardziej powszechna i coraz większa liczna ludzi zaczęła akceptować fakt, że nie żyją w centrum Wszechświata.

W ramach niniejszej pracy został zrealizowany program edukacyjny przy pomocy którego użytkownik zapozna się z oboma modelami jak i rysem historycznym związanym z nimi. Posiada on cechy oprogramowania multimedialnego, które mają za zadanie wywołać uczucie zaciekawienia u odbiorcy i zarazem poprawić przyswajalność wiedzy zawartej w nim. Zrealizowane jest to przy pomocy paru elementów. Jednym z nich jest interaktywność, użytkownik ma możliwość samemu coś kliknąć, a menu główne całego programu zrealizowane jest na podobnej zasadzie jak w grach komputerowych, czyli składa się z tła i różnych przycisków odsyłających do poszczególnych elementów aplikacji. Kolejnym elementem jest modułowość, polega na zastosowaniu podprogramów, które podpięte do menu głównego stanowią całość projektu. Pozwala to na wygodną aktualizację w przypadku błędu, lub pojawienia się nowych danych. Kolejnym elementem jest wieloplatformowość. Dzięki zastosowaniu przy tworzeniu projektu technologii Flash, program ten może być użytkowany na komputerach wyposażonych w systemy operacyjne z rodziny Microsoft Windows, Mac OS, Linux, a nawet na urządzeniach mobilnych. Pozwala to znacząco poszerzyć grono odbiorców.

Powyższe aspekty zostały zastosowane celem uatrakcyjnienia aplikacji, tak aby spełniała rolę pomocy naukowej podczas nauczania. Jednak nie stanowi pełnego kompendium wiedzy z tego zakresu, a ma raczej za zadanie zaciekawić i skłonić do pogłębiania wiedzy z dziedziny astronomii.

7. Literatura

- [1] *Popularna Encyklopedia Powszechna*, t. 4, pod red. Jana Pieszcachowicza, Kraków, Oficyna Wydawnicza FOGRA, 2001
- [2] <http://www.britannica.com/EBchecked/topic/266559/Hipparchus> - Strona zawierająca informacje o Hipparchu.
- [3] *Popularna Encyklopedia Powszechna*, t. 1, pod red. Jana Pieszcachowicza, Kraków, Oficyna Wydawnicza FOGRA, 2001
- [4] <http://www.astronomynotes.com/history/s3.htm#A2.4> - Opis idei ekwantu.
- [5] <http://www.britannica.com/EBchecked/topic/482098/Ptolemy> - Strona poświęcona Klaudiuszowi Ptolemeuszowi.
- [6] <http://farside.ph.utexas.edu/syntaxis/Almagest.pdf> - Współczesna wersja Almagestu Ptolemeusza.
- [7] <http://www.britannica.com/EBchecked/topic/34377/Aristarchus-of-Samos> - Strona poświęcona Arystarchowi z Samos.
- [8] <http://grzegorz.paczek.dlf.interia.pl/Tatarkiewicz1.pdf> - Internetowe wydanie pierwszego tomu Historii filozofii autorstwa Władysława Tatarkiewicza.
- [9] <http://copernicus.torun.pl/biografia/1473-1543/2/> - Strona zawierająca biografię Mikołaja Kopernika.
- [10] <http://copernicus.torun.pl/rewolucja/recepcja/3/1/> - Rys historyczny modelu Tycho Brahego.
- [11] <http://copernicus.torun.pl/rewolucja/recepcja/3/2/> - Rys historyczny dokonań Johannes Keplera.
- [12] <http://www.britannica.com/EBchecked/topic/224058/Galileo> - Strona zawierająca biografię Galileusza.
- [13] http://www.bgfax.com/school/distance_history.pdf - Historia odkryć paralaksy.
- [14] http://www.cosa.com/after_fx/ - Historia rozwoju programu After Effects.
- [15] http://www.adobe.com/macromedia/events/john_gay/ - Historia rozwoju programu Flash.
- [16] <http://www.blender.org/blenderorg/blender-foundation/history/> - Strona zawierająca historię powstawiania i rozwoju programu Blender.
- [17] <http://www.flashdevelop.org/> - Strona domowa aplikacji FlashDevelop.
- [18] <http://opensource.org/licenses/MIT> - Licencja MIT.

- [19] <http://haxe.org/doc/intro> - Obsługiwane języki przez wieloplatformowy język HaXe.
- [20] <http://docs.gimp.org/2.8/en/> - Dokumentacja programu GIMP.
- [21] http://www.paranoia.clara.net/articles/inno_setup_review.html - Strona zawierająca opis programu Inno Setup.
- [22] <http://www.aerospaceguide.net/spaceexploration/index.html> - Strona zawierająca średnice poszczególnych planet.
- [23] http://www.smartconversion.com/otherInfo/Mass_of_planets_and_the_Sun.aspx - Strona zawierająca masy poszczególnych planet.
- [24] http://cse.ssl.berkeley.edu/AtHomeAstronomy/act10_datasheet.html - Strona zawierająca dystans poszczególnych planet od Słońca.
- [25] <http://www.asterism.org/ops/iqc/IQC-%20Planets%20table.pdf> - Strona zawierająca parametry poszczególnych planet.
- [26] <http://www.universetoday.com/94252/characteristics-of-the-sun/> - Strona zawierająca charakterystykę Słońca.
- [27] http://math.ucr.edu/home/baez/physics/Relativity/SpeedOfLight/measure_c.html - Strona zawierająca rys historyczny poszczególnych szacowań prędkości światła wraz z wartościami.
- [28] <http://www.astrostudio.org/500nearest.html> - Odległości poszczególnych gwiazd od Ziemi.
- [29] <http://www.cds.hawaii.edu/kahana/downloads/curriculum/SectionII/Unit7/7.C.C.comparingPlanets/7.C.1.TeachersNotes.pdf> - Strona zawierająca średnie gęstości poszczególnych planet.
- [30] <http://listphobia.com/2011/01/18/10-largest-known-stars-of-the-universe/> - Rozmiary dziesięciu największych odkrytych gwiazd.
- [31] Angela Wilkes, *Budowa Wszechświata*, przeł. Grzegorz Szymanowski, Warszawa, Polska Oficyna Wydawnicza BGW, 1991
- [32] <http://copernicus.torun.pl/> - Centrum wiedzy o Mikołaju Koperniku.