

Raport z projektu o Układzie Słonecznym

Maria Budziło, Kateryna Ivasenko

11 Czerwca 2021

Rozdział 1

Generalne informacje

1.1 Wykonawcy

Maria Budziło, Kateryna Ivasenko

1.2 Cele

Przedstawienie koncepcji i realizacji wybranego tematu- Ruch planet w Układzie Słonecznym- w języku Python i C++

Rozdział 2

Opis projektu

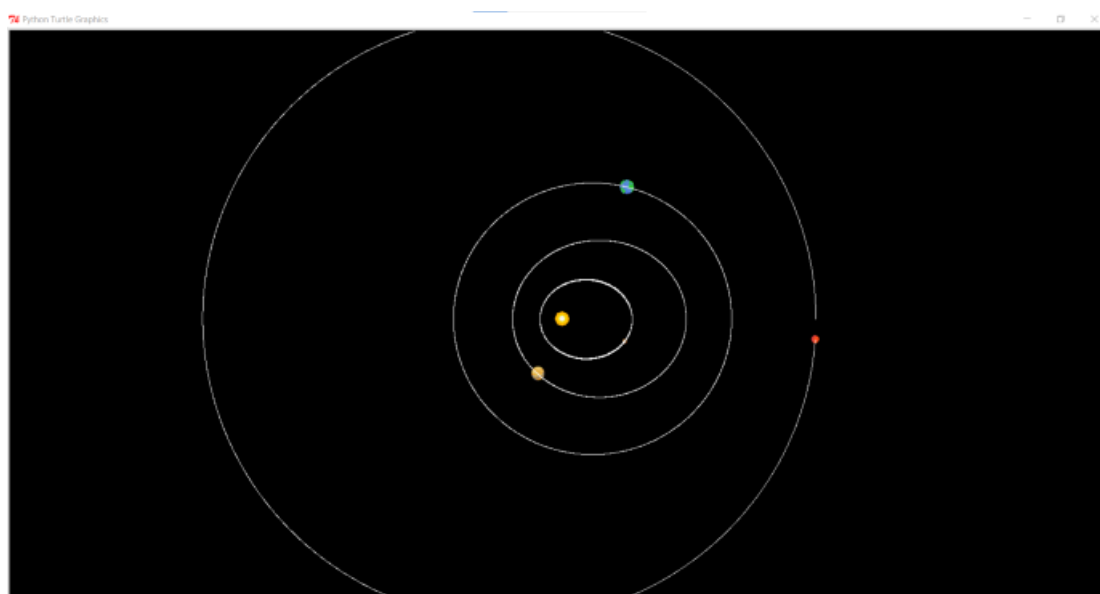
2.1 Kod w języku Python

Projekt, który wybrałyśmy dotyczy Układu Słonecznego. Naszym celem było przedstawienie ruchu planet, który byłby najbliższy ich realnemu ruchowi w przestrzeni. Do realizacji tego kodu w Pythonie wykorzystaliśmy bibliotekę „turtle”. Na początku zostały zebrane informacje dotyczące odległości planet od Słońca oraz wielkości planet, które zostały ujęte w tabeli 2.1. Ze względu na chęć przedstawienia Układu w realistycznych wielkościach, Układ został przedstawiony jedynie do Marsa, gdyż kolejne planety są nieporównywalnie większe co sprawiłoby prawie całkowity brak widoczności tych pierwszych. Również wielkość Słońca nie została przedstawiona realistycznie, aby umożliwić widok całych orbit pozostałych czterech planet.

Odległość planety od Słońca	Średnica planety
Słońce: 0 km	Słońce: 1 392 684 km
Merkury: 58 milionów km	Merkury: 4 879 km
Wenus: 108 milionów km	Wenus: 12 104 km
Ziemia: 150 milionów km	Ziemia: 12 760 km
Mars: 228 milionów km	Mars: 6 787 km

Tabela 2.1: Dane planet [2]

Wielkość planet, poza Słońcem została dopasowana w skali 1.5/1000 pikseli. Odległość od Słońca natomiast w skali 1.5/1000000 + 10 pikseli, ze względu na promień słońca mający 10 pikseli. Zdefiniowaliśmy planety określając ich położenie kolor rysowanych linii/orbit- biały oraz odległość od Słońca. Również kształt znacznika (podstawowej strzałki), który w tym wypadku był plikiem .gif odpowiedniej planety.

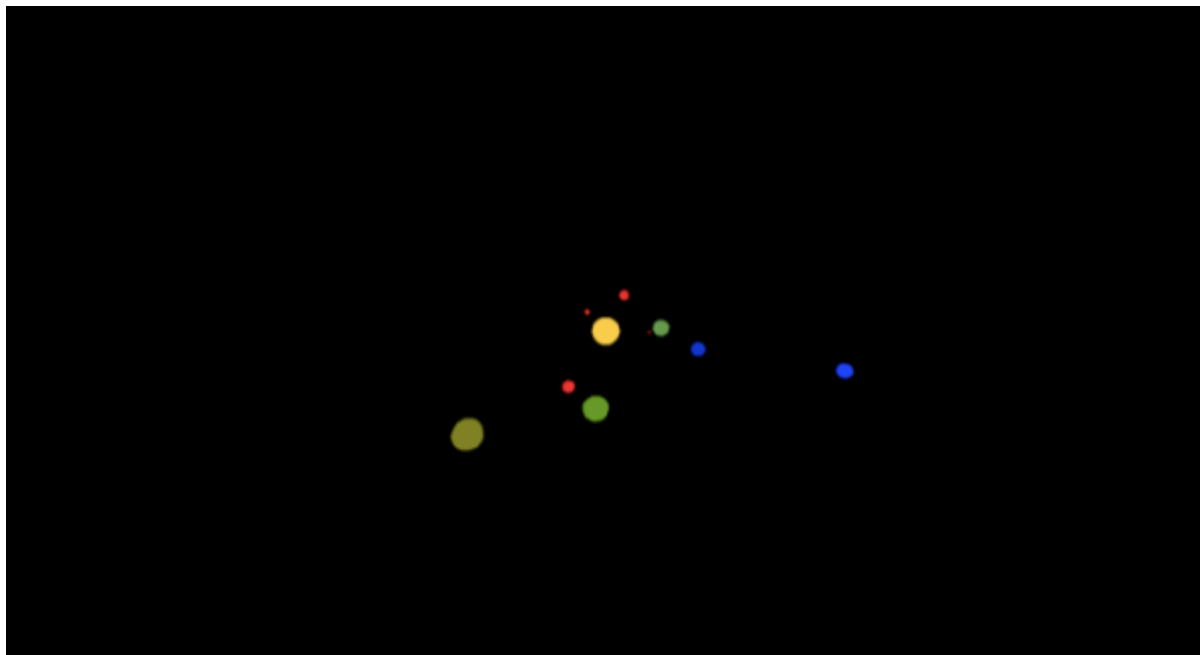


Rysunek 2.1: Efekt końcowy programu w języku Python

Następnie zdefiniowaliśmy ruch planety po orbicie eliptycznej wokół środka masy, którym jest Słońce w tym wypadku. Równanie to niestety nie jest idealne czy bardzo realistyczne, jednak w ciekawy sposób ukazuje przybliżony i uproszczony ruch planet względem siebie, korzystano tutaj z pomocnego kodu na ruch eliptyczny [1]. Na końcu do poszczególnych planet zostały przypisane wartości, kształty oraz ruch. Ostateczny wynik został przedstawiony na rysunku 2.1.

2.2 Kod w języku C++

Kod pisany w języku C++ natomiast używał jednej biblioteki `glut.h`. OpenGL to biblioteka wieloplatformowa, tj. programy napisane w OpenGL można łatwo przenieść do różnych systemów operacyjnych, uzyskując ten sam efekt wizualny.



Rysunek 2.2: Efekt końcowy programu w języku C++

Jedyną złą rzeczą jest to, że dla konkretnego systemu operacyjnego musisz skonfigurować OpenGL na swój własny sposób. Operacje z oknami, interfejs sterowania, operacje z urządzeniami wejścia/wyjścia muszą zostać przepisane w innym systemie operacyjnym. Ostateczny wynik został przedstawiony na rysunku .

Rozdział 3

Wnioski

3.1 Kod w języku Python

Po pierwsze kod działa poprawnie i nie wyświetla żadnych błędów. Założony cel został więc osiągnięty. W przyszłości na pewno można by było ulepszyć ruch planet, by nie tylko sprawiał wrażenie realistycznego, ale takim też był.

3.2 Kod w języku C++

W kodzie C++ udało się osiągnąć, że program wykorzystując wiele funkcji do przesyłania obrazu wolumetrycznego, z powodzeniem uzyskał możliwość, że każda planeta obracała się na swojej orbicie z własną prędkością. I razem wyglądało to jak Układ Słoneczny.

Bibliografia

- [1] kanał Windows 10 Assistant: Simulate Planetary Motion With Python Turtle
<https://www.youtube.com/watch?v=v1GT0jGBR-Iab>*channel = Windows10Assistant*
- [2] Nancy Atkinson: Order Of the Planets From The Sun
<https://www.universetoday.com/72305/order-of-the-planets-from-the-sun/>