

Dokumentacja Projektu

Wykonali:

- Piotr Leymańczyk
- Kamil Sojka

Link do GitHuba: <https://github.com/GrimDent/WMZF>

1. Wstęp

• Cel Projektu

Celem projektu jest symulowanie zjawiska zderzenia dwóch ciał niebieskich.

• Opis modelowanego zjawiska fizycznego

Zderzenie ze sobą dwóch odizolowanych ciał niebieskich, wskutek przyciągania grawitacyjnego. Opory i czynniki zewnętrzne nie istnieją. Ciała są jedynymi obiektami w symulowanej przestrzeni. Program będzie wykorzystywał prawo powszechnego ciążenia

$$F = G \frac{m_1 * m_2}{r^2}$$

• Opis wykorzystywanych narzędzi

- Język programowania Python w wersji 3.8
- IDE PyCharm Community Edition 2020.2.3
- moduły NumPy oraz Matplotlib
- GitHub

2. Ogólny opis projektu i możliwe alternatywy

Projekt polega na stworzeniu programu, który wykona obliczenia na podstawie zmiennych podanych przez użytkownika. Jako zmienne będą wprowadzane masy dwóch ciał oraz ich promienie, jak i odległość między ciałami. Następnie program na podstawie danych obliczy siłę zderzenia ciał oraz ich prędkość w chwili zderzenia. Alternatywy: Dodanie możliwości obliczenia czasu po jakim 2 obiekty się zderzą.

3. Specyficzne wymagania

• Wymagania funkcjonalne

Stworzenie programu obliczającego, siłę zderzenia dwóch ciał niebieskich oraz ich

prędkość w chwili zderzenia, dla podanych przez użytkownika odległości ciał od siebie, promieni i mas obu obiektów.

-zapisanie wyników do pliku .txt

-wyświetlanie wykresów zależności prędkości obiektów od odległości

● Wymagania niefunkcjonalne

Program poprosi użytkownika o wpisanie wybranych przez niego: masy i promienia obu ciał oraz odległości pomiędzy powierzchniami obu obiektów za pomocą polecenia input. Następnie funkcja test sprawdza czy podane wartości są prawidłowe (czy są liczbami i czy są dodatnie). Po wykonaniu obliczeń program otwiera plik w trybie nadpisywania i dopisuje wyniki do pliku. Na koniec wykonują się dwie funkcję które korzystając z NumPy i Matplotlib tworzą odpowiedni wykres.

4. Harmonogram prac z zadaniami do wykonania

Pierwszy commit (do 26.11)

-opracowanie działającego prototypu, z ograniczoną funkcjonalnością.

Drugi commit (do 3.12)

-wprowadzenie możliwości podawania danych wejściowych przez użytkownika oraz zapisywanie wyników działania programu do pliku tekstowego.

Trzeci commit (do 10.12)

-wprowadzenie funkcjonalnych wykresów wyświetlających zależność pomiędzy prędkością a czasem dla obydwu obiektów.

Czwarty commit (do 17.12)

-dodanie zabezpieczeń przed wprowadzeniem błędnych danych przez użytkownika.

Piąty commit do (7.01)

-pozbycie się ewentualnych błędów i optymalizacja kodu.