



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ FIZYKI TECHNICZNEJ
I MATEMATYKI STOSOWANEJ

Imię i nazwisko studenta: Adam Kaczmarkiewicz

Nr albumu: 180911

Poziom kształcenia: Studia pierwszego stopnia

Forma studiów: stacjonarne

Kierunek studiów: Fizyka Techniczna

Specjalność: Informatyka stosowana

DOKUMENTACJA PROJEKTU

Symulacja zderzeń sprężystych i niesprężystych ciał kulistych.

Spis treści

1	Wstęp	3
1.1	Opis projektu	3
1.2	Opis wykorzystywanych narzędzi	4
2	Ogólny opis projektu i możliwe alternatywy	4
3	Specyficzne wymagania	4
3.1	Wymagania funkcjonalne	4
3.2	Wymagania нефункционалне	5
4	Harmonogram prac z zadaniami do wykonania	5

1 Wstęp

1.1 Opis projektu

Projekt ma na celu stworzenie programu, który będzie opisywał i wyświetlał zderzenia ciał kulistych.

Zderzenia to zjawiska powstające przy zetknięciu poruszających się względem siebie ciał. Podczas zderzeń zachowane są zasady zachowania energii oraz zachowania pędu. Zderzenia są sprężyste i niesprężyste. W fizyce zderzenia analizuje się opisując stan ciał przed i po zderzeniu nie wnikając w szczegóły oddziaływania w trakcie zderzenia. Zderzenia niesprężyste to takie gdzie energia kinetyczna nie została zachowana. Zderzenia sprężyste to takie gdzie obiekty w stanie początkowym i końcowym są takie same i została zachowana energia kinetyczna.

ZDERZENIA SPRĘŻYSTE

Zderzające się dwa ciała oznaczono indeksami 1 i 2, ich prędkości przed zderzeniem oznaczono przez u , a po zderzeniu przez v , a masy przez m . Całkowita energia kinetyczna po zderzeniu jest równa energii kinetycznej ciał przed zderzeniem:

$$\frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2} = \frac{m_1 u_1^2}{2} + \frac{m_2 u_2^2}{2}$$

Całkowity pęd po zderzeniu jest równy pędowi przed zderzeniem:

$$m_1 v_1 + m_2 v_2 = m_1 u_1 + m_2 u_2$$

Z powyższych równań wynikają prędkości ciał po zderzeniu:

$$v_1 = \frac{u_1(m_1 - m_2) + 2m_2 u_2}{m_1 + m_2}$$
$$v_2 = \frac{u_2(m_2 - m_1) + 2m_1 u_1}{m_1 + m_2}$$

ZDERZENIA NIESPRĘŻYSTE

W czasie tego zderzenia nie działają w układzie odosobnionym siły zachowawcze, a zatem nie jest zachowana energia mechaniczna. Natomiast pęd zostaje zachowany, z czego wynika

$$m_1 v_1 + m_2 v_2 = (m_1 + m_2)u$$

stąd prędkość wspólna po zderzeniu wynosi:

$$u = \frac{m_1 v_1 + m_2 v_2}{m_1 + m_2}$$

Znając energię kinetyczną obu ciał przed zderzeniem, jak również energię kinetyczną bryły utworzonej w wyniku zderzenia, można obliczyć stratę energii kinetycznej przekształconą na inne postacie energii:

$$E = \frac{1}{2} \frac{m_1 m_2}{m_1 + m_2} (v_1 - v_2)^2$$

1.2 Opis wykorzystywanych narzędzi

Program zostanie napisany w języku Python 3.7. Zostaną użyte biblioteki numPy do obliczeń numerycznych i pracy z danymi oraz matplotlib do prezentacji wyników. Całość zostanie stworzona w zintegrowanym środowisku programistycznym dla języka Python, czyli PyCharm. Wykorzystany zostanie też system kontroli wersji i commity będą umieszczane na GitHubie.

Link do projektu na GitHubie:

<https://github.com/Adam-Kacz/PROJEKT.git>

2 Ogólny opis projektu i możliwe alternatywy

Zamodelowane zostanie zderzenie dwóch ciał kulistych. Użytkownik będzie miał wybór typu zderzenia, promieni, masy i prędkości ciał. Modelowe zderzenie będzie wyświetlane wraz z danymi. Alternatywą może być dodanie większej ilości ciał, czy zamiast podawania wartości ciał, podać warunki końcowe.

3 Specyficzne wymagania

3.1 Wymagania funkcjonalne

Program powinien:

- Pytać o warunki początkowe (promień, masa, prędkość itd.)
- Sprawdzać czy dane podane przez użytkownika są poprawne
- Obliczać i wyświetlać warunki końcowe
- Możliwość zapisywania prezentacji i wyników do pliku

3.2 Wymagania niefunkcjonalne

Program powinien:

- Działać
- Być niezawodny
- Łatwy w użyciu
- Przejrzysty
- Szybko interpretować
- Nie obciążać urządzenia

4 Harmonogram prac z zadaniami do wykonania

I commit (27.11.2020r.)

Część kodu odpowiedzialna za zderzenie sprężyste

II commit (04.12.2020r.)

Część kodu odpowiedzialna za interfejs użytkownika (interakcję z użytkownikiem)

Wyłapywanie błędów złych danych wprowadzonych przez użytkownika

III commit (11.12.2020r.)

Część kodu odpowiedzialna za wizualizację, interfejs graficzny

IV commit (18.12.2020r.)

Część kodu odpowiedzialna za zderzenie niesprężyste

V commit (08.01.2021r.)

Ostatnie szlify, testowanie, pozbycie się błędów, poprawa estetyczna, przygotowania do prezentacji.