

Specyfikacje dotyczące wymagań oprogramowania

dla

<Projektu z WDMZF >

Wersja 1.0

Przygotowane przez <Szymon Niewdana , Maciej Szröder >

<19.11.2020-14.01.2021>

Spis treści

Spis treści

1. Wprowadzenie

1.1 Nazwa projektu

1.2 Źródło inspiracji projektu

1.3 Cel projektu

2.Ogólny opis modelowanego zjawiska fizycznego

2.1 Kinematyka

2.2 Dynamika

3.Narzędzia pracy

3.1 Wykorzystywane narzędzia

4.Opis ogólny

4.1 Opis ogólny programu i możliwe alternatywy

4.2 Wymagania funkcjonalne

4.3Wymagania нефункционалне

5.Ogólny harmonogram prac

1.Wprowadzenie

1.1 Projekt - Symulacja ruchu jednostajnie zmiennego

Wstęp do modelowania zjawisk fizycznych

Wydział Fizyki Technicznej i Matematyki Stosowanej

1.2 Źródło inspiracji projektu

Życie to wielka podróż, zazwyczaj pokonujemy je jednostajnie, ale czasami wymaga od nas przyspieszania, zwolnienia lub całkowitego zatrzymania się. Ta prosta metafora ludzkiego życia stała się dla nas inspiracją do stworzenia programu zajmującego się badaniem i klasyfikacją ruchu jednostajnie zmiennego.

1.3 Cel projektu

Stworzenie programu analizującego ruch jednostajnie zmienny zakładający możliwość edycji wprowadzanych danych, na podstawie których powstaje symulacja.

2.Opis modelowanego zjawiska fizycznego

2.1 Kinematyka

Ruch jednostajnie zmienny to ruch prostoliniowy, w którym wartość przyspieszenia jest stała ($a = \text{const.}$). Jeżeli przyspieszenie powoduje wzrost prędkości ciała ($a > 0$) to ruch nazywamy jednostajnie przyspieszonym, a jeżeli prędkość maleje ($a < 0$) to ruchem jednostajnie opóźnionym. W ruchu jednostajnie przyspieszonym/opóźnionym szybkość wzrasta/maleje w każdej jednostce czasu o taką samą wartość, czyli liniowo.

wzór na prędkość końcową V_k ciała po czasie t

$$V_k = V_p \pm a \cdot t$$

gdzie:

V_k - prędkość końcowa

V_p - prędkość początkowa

a - przyspieszenie

t - czas

wzór na drogę s ciała po czasie t

$$s = V_p \cdot t \pm \frac{1}{2} \cdot a \cdot t^2$$

gdzie:

s - droga

V_p - prędkość początkowa

a - przyspieszenie

t - czas

2.2 Dynamika

II zasada dynamiki Newtona brzmi „Jeżeli na ciało działające siły nie równoważą się, to ciało to porusza się ruchem zmiennym z przyspieszeniem wprost proporcjonalnym do działającej siły wypadkowej i odwrotnie proporcjonalnym do masy ciała.”

$$a = \frac{F_w}{m}$$

gdzie:

a - przyspieszenia

F_w - siła wypadkowa działająca na ciało

m - masa ciała

Siła tarcia/ tarcie

Tarcie jest siłą, która przeciwdziała względnemu ruchowi między ciałami będącymi w kontakcie.

Jeśli dwa spoczywające ciała pozostają w kontakcie, to działa między nimi siła nazywana tarcie statycznym (ang. static friction). Jeśli dwa ciała pozostające w kontakcie przesuwają się względem siebie, to siła występująca między nimi nazywana jest tarcie kinetycznym (ang. kinetic friction).

Wartość siły tarcia statycznego i kinetycznego obliczamy na podstawie wzoru:

$$T = \mu_{s/k} \cdot F_n$$

gdzie:

T - siła tarcia

$\mu_{s/k}$ - współczynnik tarcia statycznego lub kinetycznego

F_n - siła nacisku

3.Narzędzia pracy

3.1Wykorzystywane narzędzia

1. Zintegrowane środowisko programistyczne wraz z niezbędnymi bibliotekami dla języka programowania Python (PyCharm)
2. System kontroli wersji - GitHub - (link)

4.Ogólny opis

4.1Ogólny opis projektu i możliwe alternatywy

Zamysłem całego programu jest obliczanie i wyświetlanie parametrów związanych z ruchem jednostajnie zmiennym. Użytkownik programu po jego uruchomieniu w menu konsoli będzie miał możliwość wyboru uwzględnienia w symulacji siły tarcia, oraz podanie podstawowych danych potrzebnych do przeanalizowania przez program poszczególnego ruchu - prostoliniowego,

przyspieszonego lub opóźnionego w zależności od wartości przyspieszenia. Dzięki temu programowi użytkownik będzie mógł w łatwy i przede wszystkim w szybki sposób otrzymać potrzebne mu wartości i dane związane z owym zagadnieniem.

4.2 Wymagania funkcjonalne

Program będzie wyświetlał dane, które można wyliczyć z podanych wartości przez użytkownika. Przedstawi osobie użytkującej również trzy wykresy- oczywiście jeśli będzie to możliwe: $V(t)$, $S(t)$, $a(t)$. Podstawowymi pytaniami programu będą pytania potrzebne do wykonania symulacji np. "Czy w całym ruchu występuje tarcie?", jeśli tak to program automatycznie przeniesie nas odpowiednio do dalszych części. Użytkownik będzie mógł podać wartości liczbowe, jeśli poda inny rodzaj danych program automatycznie zasugeruje błąd i zapyta ponownie o nie. Na samym końcu wykonywania programu osoba korzystająca jeśli będzie chciała będzie mogła zapisać wszystkie dane w osobnym pliku.

4.3 Wymagania нефunkcjonalne

Program ma działać optymalnie szybko oraz być łatwy w obsłudze. Nie powinien zajmować dużo przestrzeni dyskowej oraz reagować sprawnie i szybko na polecenia użytkownika. Także jako program modelujący pewne zjawisko fizyczne musi być niezawodny w swoim działaniu i bezbłędnie obliczać żądane wartości. Program będzie przeznaczony dla pojedynczego użytkownika. Jego interfejsem będzie konsola. Będzie działał na podstawowych systemach operacyjnych, na których obsługiwany jest język Python.

5. Ogólny harmonogram prac

26.11.2020 - stworzenie zarysu funkcji programu
3.12.2020 - implementacja funkcji umożliwiających pobieranie danych z klawiatury i z pliku
10.12.2020 - implementacja praw fizycznych modelu
17.12.2020 - implementacja wizualizacji danych wyjściowych oraz zapisu do pliku
7.01.2021 - wprowadzanie poprawek do programu
14.01.2021 - prezentacja projektu