Изображение выглядит как Шрифт, логотип, Графика, белый

Автоматически созданное описание**Университет ИТМО**

**Физико-технический мегафакультет Физический факультет**

|  |  |
| --- | --- |
| Группа M3201 | К работе допущен |
| Студенты Ткачук С.A. и Чуб Д.О. | Работа выполнена |
| Преподаватель Шоев В.И. | Отчет принят |

****

**Отчет по моделированию № 1.01**

**Абсолютно упругое взаимодействие**

1. Теоретическая часть

В данной работе изучается абсолютное упругое соударение двух тел. Используются законы сохранения импульсов тел для моделирования движения этих тел. В данной работе используется модель абсолютного упругого столкновения, которая предполагает, что энергия и импульс сохраняются во время столкновения.

1. Используемые формулы:

(1) – закон сохранения импульса

(2) – закон сохранения энергии

Используя эти формулы, можем получить измененные скорости:

1. Численный алгоритм для решения уравнений

Используется метод Эйлера для решения уравнения движения. Мы обновляем значения параметра координаты x на каждом временном шаге и аппроксимируем траекторию движения.

1. Программный код на языке python:

import pygame  
  
pygame.init()  
  
window\_height = 280  
window\_width = 1000  
screen = pygame.display.set\_mode((window\_width, window\_height))  
  
font = pygame.font.Font(None, 28)  
  
  
class Cube:  
 def \_\_init\_\_(self, top\_left\_x, side\_length, color, speed, mass):  
 self.side\_length = side\_length  
 self.top\_left\_x = top\_left\_x  
 self.top\_left\_y = window\_height - self.side\_length  
 self.color = color  
 self.speed = speed  
 self.mass = mass  
  
  
cube1\_speed = 0  
cube2\_speed = 100  
cube1\_mass = 1  
cube2\_mass = 1000  
  
cube1 = Cube(400, 30, (255, 255, 0), cube1\_speed, cube1\_mass)  
cube2 = Cube(600, 60, (255, 0, 0), -cube2\_speed, cube2\_mass)  
  
hits\_counter\_text = font.render('Количество ударов:', True, (0, 0, 0))  
  
running = True  
hits\_count = 0  
  
while running:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 running = False  
  
 screen.fill((255, 255, 255))  
  
 if cube1.top\_left\_x <= 0:  
 hits\_count += 1  
 cube1.speed = -cube1.speed  
  
 if cube1.top\_left\_x + cube1.side\_length >= cube2.top\_left\_x:  
 hits\_count += 1  
 new\_speed0 = (((cube1.mass - cube2.mass) \* cube1.speed) + 2 \* cube2.mass \* cube2.speed) / (cube1.mass + cube2.mass)  
 new\_speed1 = (((cube2.mass - cube1.mass) \* cube2.speed) + 2 \* cube1.mass \* cube1.speed) / (cube1.mass + cube2.mass)  
 cube1.speed = new\_speed0  
 cube2.speed = new\_speed1  
  
 cube1.top\_left\_x = cube1.top\_left\_x + cube1.speed \* 0.016  
 cube2.top\_left\_x = cube2.top\_left\_x + cube2.speed \* 0.016  
  
 if cube2.top\_left\_x <= cube1.side\_length:  
 pygame.draw.rect(screen, cube1.color, (0, cube1.top\_left\_y, cube1.side\_length, cube1.side\_length))  
 pygame.draw.rect(screen, (0, 0, 0), (0, cube1.top\_left\_y, cube1.side\_length, cube1.side\_length), 1)  
 pygame.draw.rect(screen, cube2.color,  
 (cube1.side\_length, cube2.top\_left\_y, cube2.side\_length, cube2.side\_length))  
 pygame.draw.rect(screen, (0, 0, 0), (cube1.side\_length, cube2.top\_left\_y, cube2.side\_length, cube2.side\_length), 1)  
 else:  
 pygame.draw.rect(screen, cube1.color,  
 (cube1.top\_left\_x, cube1.top\_left\_y, cube1.side\_length, cube1.side\_length))  
 pygame.draw.rect(screen, (0, 0, 0), (cube1.top\_left\_x, cube1.top\_left\_y, cube1.side\_length, cube1.side\_length), 1)  
 pygame.draw.rect(screen, cube2.color,  
 (cube2.top\_left\_x, cube2.top\_left\_y, cube2.side\_length, cube2.side\_length))  
 pygame.draw.rect(screen, (0, 0, 0), (cube2.top\_left\_x, cube2.top\_left\_y, cube2.side\_length, cube2.side\_length), 1)  
  
 screen.blit(hits\_counter\_text, (20, 20))  
 hits\_count\_text = font.render(str(hits\_count), True, (0, 0, 0))  
 screen.blit(hits\_count\_text, (250, 20))  
  
 pygame.display.flip()  
 pygame.display.update()  
  
 pygame.time.Clock().tick(120)  
  
pygame.quit()  
  
print("Количество ударов:", hits\_count)

1. Вывод

Представленный код демонстрирует применение численного метода Эйлера для моделирования столкновения двух кубов с использованием базовых принципов физики. Метод обеспечивает аппроксимацию движения объектов, используя законы сохранения энергии и импульса.