Муниципальное бюджетное общеобразовательное учреждение средняя общеобразовательная

школа с. Красное

Кафедра

Создание программы для помощи изучения закона сохранения импульса.

Проект

Выполнил

А. Мальцев, ученик 10 А класса

Научный руководитель —

А.Г. Щедрин, учитель информатики

Красное — 2021

Цель проекта: Создание программы для помощи изучения закона сохранения импульса.

Задачи:

1. написать программу для отображения главного меню
2. написать программу для симуляции столкновений
3. создать описание кода главного меню и симуляции

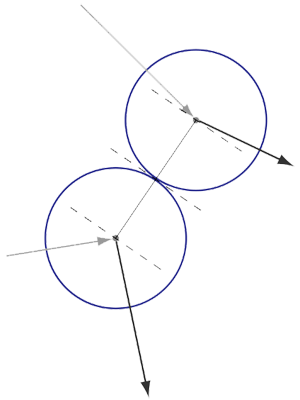
Введение:

Закон сохранения импульса - Векторная сумма импульсов тел, составляющих замкнутую систему, не меняется с течением времени при любых движениях и взаимодействиях этих тел. При повторении закона сохранения импульсов в 10 классе, я заметил, как сложно учителю физики разобрать множество вариантов столкновения тел и их дальнейшего движения. Тогда я задумался о важности этой темы. Полное понимание и осознание этого физического явления является очень важным как для учеников, которые продолжат углублённое изучение физики и будут применять этот закон при решении задач на ЕГЭ, так и для остальных учеников, ведь именно закона сохранения импульсов лежит в основе любых движений после столкновения тел. Сейчас я вспомнил, как мне было сложно изучить эту тему не имея никаких интерактивных пособий. Изучив язык программирования Python я решил помочь ученикам и учителям с данной проблемой, написав интерактивное приложение, которое наглядно продемонстрирует закон в действии. Оно поможет учителям быстрее и проще объяснить эту сложную тему ученикам, а школьники смогут самостоятельно рассмотреть все интересующие их варианты столкновений и уже после этого задать вопрос преподавателю.

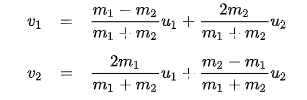
Описание работы главного меню приложения:

Главное меню приложения написано на языке программирования Python с использованием библиотек PyQt5, Qt Designer и random. PyQt5 позволяет использовать фреймворк Qt GUI (GUI — это графический интерфейс пользователя) в Python. Модуль random предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности. Qt Designer — дизайнер графического интерфейса пользователя. При помощи Qt Designer было создано окно 600 \* 600 пикселей. В этом окне уже находятся: 2 горизонтальных слайдера отвечающих за сопротивление воздуха и эластичность шаров, 2 кнопки, и подвижная область, где находятся настройки каждого шара. При нажатии кнопки “Сгенерировать” программа в подвижной области создаёт для каждого шара по 2 горизонтальных слайдера отвечающих за размер и плотность шара.

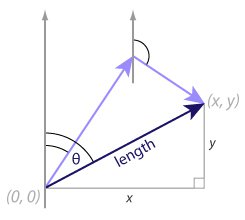
Описание работы симуляции столкновений:

Симуляция столкновений написано на языке программирования Python с использованием библиотек Pygame, random и math. Pygame — набор [модулей](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) (библиотек) языка программирования [Python](https://ru.wikipedia.org/wiki/Python" \o "Python), предназначенный для написания [компьютерных игр](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) и [мультимедиа](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0) - приложений. Pygame базируется на мультимедийной библиотеке [SDL](https://ru.wikipedia.org/wiki/Simple_DirectMedia_Layer). math - математические функции. Модуль math предоставляет доступ к математическим функциям и константам. Модуль math понадобится нам, так как мы будим использовать тригонометрию. Из этого модуля были использованы функции: math.sin()[3], math.cos()[4], math.hypot()[1], math.atan2()[2], math.pi[5]. При запуске приложения создаётся окно 900 \* 900 пикселей с названием “Simulation” и белым фоном. Запускается цикл отрисовки приложения до тех пор, пока приложение не будет закрыто. С учётом полученных данных из главного меню в случайных местах создаются и отрисовываются шары. Шары отличаются насыщенностью синего цвета в зависимости от их плотности (чем более насыщенный цвет, тем плотнее шар). Масса шариков рассчитывается по формуле: масса = плотность \* площадь шара. После появления шаров на экране при помощи мышки их можно схватить и бросить в любом направлении. При столкновении со стеной шарик отразится от неё. Когда две частицы сталкиваются, они отскакивали друг от друга. Теоретически, когда две круглые частицы сталкиваются, они соприкасаются в бесконечно малой точке. Угол этой точки является касательной частицы в этой точке. Как видно из приведенной ниже диаграммы, этот угол перпендикулярен линии, соединяющей центр двух частиц. Мы можем рассматривать столкновение так, как если бы частицы отскакивали от плоской поверхности.

Я беру вектор частицы и добавляю второй, угол которого перпендикулярен углу столкновения, и величина которого основана на импульсе (массе умноженной на скорость) второй частицы. А новые скорости данных объектов я рассчитываю по данной формуле[6].



Функция addVectors() принимает два вектора (каждый-угол и длину) и возвращает одиночный комбинированный вектор. Сначала мы движемся вдоль одного вектора, затем вдоль другого,чтобы найти координаты x, y для того места, где окажется частица (обозначенная (x, y) на рисунке ниже).Затем мы вычисляем вектор, который попадает туда непосредственно. Для этого мы построим прямоугольный треугольник, как показано на рисунке ниже.



Новая длина вектора (скорость частицы) равна гипотенузе треугольника, которую можно вычислить с помощью math.hypot()[1]. Он берет координату x,y и вычисляет ее расстояние от начала координат (0,0). Заметим, что хотя положение нашей частицы на экране не равно (0,0), все векторы находятся относительно положения частицы, поэтому можно считать, что они начинаются с 0,0.

Угол нового вектора немного сложнее вычислить. Сначала мы находим угол в треугольнике, вычисляя арктангенс y/x. Мы могли бы сделать это с помощью функции math.atan(), но тогда нам нужно было бы иметь дело со случаем x=0 и вычислить знак угла. Однако Python предоставляет нам удобную функцию math.atan2()[2], которая принимает координаты x , y, вычисляет знак угла для нас и ведет себя правильно, когда x=0. Как только у нас есть угол треугольника, мы вычитаем его из pi/2, чтобы вычислить угол вектора.

Заключение:

В заключении хотелось бы сказать, что я надеюсь, моё приложение поможет ученикам в изучении закона сохранения импульса. Ведь понять это физического явления достаточно сложно без наглядного и интерактивного пособия.

Литература:

1. Python 3 - Числовая функция hypot() <https://andreyex.ru/yazyk-programmirovaniya-python/uchebnik-po-python-3/python-3-chislovaya-funkciya-hypot/> Дата обращения: 19.05.2021.
2. [Функция](https://en.wikipedia.org/wiki/Function_(mathematics)) atan2 <https://en.wikipedia.org/wiki/Atan2> Дата обращения: 19.05.2021.
3. Python | функция math.sin () <http://espressocode.top/python-math-sin-function/> Дата обращения: 19.05.2021.
4. Python | функция math.cos () <http://espressocode.top/python-math-cos-function/> Дата обращения: 19.05.2021.
5. Python math.pi Константа <https://www.w3schools.com/python/ref_math_pi.asp> Дата обращения: 19.05.2021.
6. Упругое столкновение – Википедия <https://en.wikipedia.org/wiki/Elastic_collision#One-dimensional_Newtonian> Дата обращения: 19.05.2021.