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:

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. 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, где площадь шара это число Пи \* радиус шара в квадрате. После появления шаров на экране при помощи мышки их можно схватить и бросить в любом направлении. При столкновении со стеной шарик отразится от неё.