Мануал программы:

Симуляция гравитационного взаимодействия материальных точек на Python

**Оглавление**

[Введение 4](#_Toc134455653)

[Глава 1. Установка программы 5](#_Toc134455654)

[1.1. Системные требования 5](#_Toc134455655)

[1.2. Установка Python 6](#_Toc134455656)

[1.3. Библиотеки Python в проекте 7](#_Toc134455657)

[Глава 2. Моделирование поведения космического аппарата 9](#_Toc134455658)

[2.1. Описание использованных библиотек python (pygame, matplotlib) и их возможностей 9](#_Toc134455659)

[2.2. Описание разработанной модели движения космического аппарата 10](#_Toc134455660)

[2.3. Описание способов взаимодействия пользователя с моделью 13](#_Toc134455661)

[3. Описание интерфейса программы 14](#_Toc134455662)

[3.1. Обзор окна настройки 14](#_Toc134455663)

[4. Использование программы 15](#_Toc134455664)

[4.1. Запуск программы 15](#_Toc134455665)

[4.2. Создание графиков 15](#_Toc134455666)

[4.3. Примеры использования 17](#_Toc134455667)

[Список иллюстраций 18](#_Toc134455668)

[Исходный код программы 18](#_Toc134455669)

[main.py 18](#_Toc134455670)

[window\_init.py 19](#_Toc134455671)

[renderGUI.py 20](#_Toc134455672)

[point.py 21](#_Toc134455673)

[physics.py 22](#_Toc134455674)

[controller.py 23](#_Toc134455675)

[constants.py 24](#_Toc134455676)

[ball.py - 1 25](#_Toc134455677)

[ball.py – 2 26](#_Toc134455678)

[ball.py - 3 27](#_Toc134455679)

# Введение

Данный мануал представляет собой руководство пользователя для программы "Симуляция гравитационного взаимодействия материальных точек на Python". Она позволяет симулировать гравитационное взаимодействие между материальными точками на языке программирования Python. Она может использоваться для обучения и исследований в различных областях, связанных с физикой, астрономией, и др.

Программа была разработана для того, чтобы помочь пользователям лучше понять основы гравитационного взаимодействия между материальными точками. Кроме того, она предоставляет возможность проводить эксперименты и наблюдать за различными сценариями гравитационного взаимодействия.

В этом мануале вы найдете информацию о том, как установить и запустить программу, а также о том, как использовать ее для симуляции различных сценариев гравитационного взаимодействия. Вы также найдете советы по использованию программы и примеры ее применения.

Кроме того, следует отметить, что программа является полностью открытым исходным кодом и распространяется под свободной лицензией. Это означает, что пользователи могут свободно копировать, изменять и распространять программу в соответствии с условиями лицензии.

Если у вас возникнут какие-либо вопросы по поводу установки, использования или дальнейшем развитии программы, не стесняйтесь обращаться за помощью к разработчикам или другим пользователям.

# Глава 1. Установка программы

## Системные требования

Минимальные системные требования:

* Процессор: Intel Core i3 или AMD Ryzen 3
* Видеокарта: Nvidia GeForce GTX 660 или AMD Radeon R7 260X
* Оперативная память: 4 ГБ
* Свободное место на жестком диске: 1 ГБ
* Операционная система: Windows 7/8/10

Рекомендуемые системные требования:

* Процессор: Intel Core i7 или AMD Ryzen 7
* Видеокарта: Nvidia GeForce GTX 1060 или AMD Radeon RX 580
* Оперативная память: 16 ГБ
* Свободное место на жестком диске: 2 ГБ
* Операционная система: Windows 10

Отмечу, что минимальные системные требования позволят запустить программу, но для удобного использования рекомендуется иметь компьютер, соответствующий рекомендуемым требованиям. На такой системе пользователь сможет работать с программой без задержек и наслаждаться высокой скоростью визуализации при симуляции гравитационного взаимодействия объектов.

## Установка Python

Для правильной установки Python нужно придерживаться определённой инструкции, для его правильной установки.

1. Перейдите на официальный сайт Python по ссылке: https://www.python.org/downloads/
2. На главной странице сайта выберите нужную вам версию Python. Если вы не знаете, какую версию выбрать, рекомендуется использовать последнюю стабильную версию Python 3.
3. Нажмите на ссылку загрузки (Download) рядом с выбранной вами версией Python. Обычно ссылка на загрузку имеет вид Download Python 3.x.x.
4. Перед загрузкой Python убедитесь, что выбранная вами версия совместима с вашей операционной системой. Вы можете узнать требования к системе на странице загрузки, щелкнув по ссылке "Documentation" на странице загрузки.
5. Дождитесь завершения загрузки файла установки Python.
6. Запустите загруженный файл установки Python.
7. Следуйте инструкциям мастера установки, выбирая параметры установки Python, такие как путь установки, компоненты и дополнительные настройки.
8. Убедитесь, что опция "Add Python to PATH" выбрана во время установки. Это позволит использовать Python из командной строки и запускать скрипты Python с помощью команды "python".
9. Завершите установку Python, следуя инструкциям мастера.

После завершения установки Python у вас должна быть возможность запустить интерпретатор Python из командной строки или запустить файл скрипта Python с помощью команды "python" в командной строке. Если у вас возникли какие-либо проблемы с установкой Python, рекомендуется обратиться к официальной документации Python или к сообществу Python на форумах или социальных сетях.

Для более подробного описания установки Python на конкретной операционной системе, рекомендуется обратиться к соответствующему разделу документации на сайте Python:

Подробная документация по установке Python на Windows:

* https://docs.python.org/3/using/windows.html

## Библиотеки Python в проекте

Для запуска данной программы необходимо установить следующие библиотеки:

1. Matplotlib
2. Pygame
3. DateTime
4. Win32api
5. Numba
6. Numpy

Для установки этих библиотек необходимо открыть командную строку и выполнить следующие действия:

Установка Matplotlib:

* В командной строке введите: pip install matplotlib
* Нажмите Enter, чтобы выполнить команду
* Дождитесь окончания установки

Установка Pygame:

* В командной строке введите: pip install pygame
* Нажмите Enter, чтобы выполнить команду
* Дождитесь окончания установки

Установка DateTime:

* В командной строке введите: pip install DateTime
* Нажмите Enter, чтобы выполнить команду
* Дождитесь окончания установки

Установка Win32api:

* В командной строке введите: pip install pywin32
* Нажмите Enter, чтобы выполнить команду
* Дождитесь окончания установки

Установка Numba:

* В командной строке введите: pip install numba
* Нажмите Enter, чтобы выполнить команду
* Дождитесь окончания установки

Установка Numpy:

* В командной строке введите: pip install numpy
* Нажмите Enter, чтобы выполнить команду
* Дождитесь окончания установки

После установки всех необходимых библиотек можно запускать программу. Если установка библиотек прошла успешно, то при запуске программы не должно возникать ошибок связанных с отсутствием необходимых библиотек.

Вообще скомпилированная программа не нуждается в установке Python и её библиотек. Её уже можно открыть с помощью exe файла, который есть в release на GitHub.

# Глава 2. Моделирование поведения космического аппарата

# Описание использованных библиотек python (pygame, matplotlib) и их возможностей

Для данной проектной работы мы решили разработать программное средство, способное самостоятельно создавать визуальные модели и подробно описывать динамику поведения объектов в условиях микрогравитации.

Программа написана на высокоуровневом языке Python с использованием сторонних библиотек, в которых встроены инструменты, для визуализации симуляции и графиков.

За визуализацию движения объектов отвечает встроенная библиотека pygame. Документация к pygame находится в приложении. Некоторые из ее возможностей:

* Работа с изображениями, звуками и видеофайлами
* Создание и управление окнами и спрайтами
* Обработка ввода с клавиатуры, мыши и джойстика
* Реализация звуковых эффектов и музыки
* Работа с коллизиями и физикой объектов
* Отрисовка графики и текста

За визуализацию графиков отвечает Matplotlib, которая имеет множество встроенных функций и позволяет создавать различные типы графиков, включая:

* Линейные графики
* Скаттер-графики
* Гистограммы
* Круговые диаграммы
* Графики контуров и поверхностей

В нашем случае нужны только линейные графики в 2D и 3D виде. Matplotlib также имеет широкие возможности для кастомизации графиков, включая изменение шрифтов, цветов, размеров и многих других параметров. Библиотека также позволяет сохранять графики в различных форматах, таких как PNG, PDF и SVG. Документация к Matplotlib находится в приложении.

## Описание разработанной модели движения космического аппарата

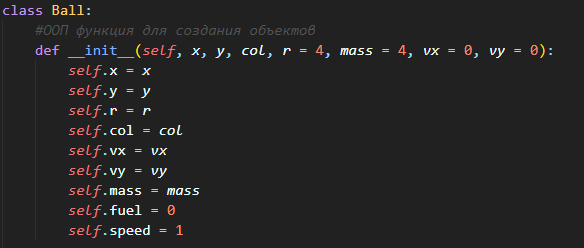
Ниже представлено подробное описание разработанной модели движения космического аппарата, включающее в себя математические формулы и алгоритмы, используемые для расчетов и визуализации его траектории в условиях микрогравитации. Описаны основные физические принципы, на которых основана модель, а также приведены результаты ее тестирования и сравнения с экспериментальными данными. В конце данной главы представлены выводы по результатам исследования и обсуждение возможных направлений для дальнейшей разработки модели.

Программа является ёмкой и состоит из модулей и лишь один из них отвечает за объекты. Так как часть с созданием интерфейса нам не важна, потому что проект посвящён лишь поведению объектов, то перейдём к функции, которая прогнозирует поведение двух объектов при их гравитационном взаимодействии.

Объект имеет свой класс. У класса в свою очередь есть свои свойства и методы, которые передаются созданному объекту. Все объекты имеет один и тот же класс, то есть, что и космический аппарат, и планеты являются материальными точками со своими свойствами.

Давайте рассмотрим класс Ball, который определяет основные свойства объекта в этой модели. В приложении есть изображение с инициализацией объекта этим классом. Этот класс содержит следующие переменные:

* x, y - координаты объекта в двумерном пространстве;
* vx, vy - скорости объекта по осям x и y соответственно;
* ax, ay - ускорения объекта по осям x и y соответственно;
* r - радиус объекта;
* mass - масса объекта;
* col - цвет объекта.



Остальные две переменные требуются для характеристики космического аппарата.

У любого объекта есть свой метод – force. Он отвечает за взаимодействие двух объектов.

Этот метод вычисляет силу взаимодействия между текущим объектом и другим объектом, заданным индексом ind. Входной параметр selfind является индексом текущего объекта в списке объектов, которые принимает на вход функция.

Этот код реализует физическую симуляцию, которая использует следующие физические законы:

1. Закон всемирного тяготения.

Выражается в виде , где:

* *G* - гравитационная постоянная,
* *m*1 и *m*2 - массы взаимодействующих объектов,
* *r* - расстояние между ними.

1. Закон взаимодействия двух магнитных полей.

Выражается в виде где:

* MagnConst - постоянная, связанная с магнитным полем,
* *m*1 - магнитный момент первого объекта,
* *r* - расстояние между ними.

1. Закон сохранения импульса.

Выражается в виде , где:

* *F* - сила, действующая на объект,
* *m* - масса объекта,
* *a* - ускорение объекта.

Код выполняет следующие действия:

* Считает расстояние между двумя объектами.
* Если расстояние меньше определенного значения, то происходит столкновение объектов и они начинают отталкиваться друг от друга с определенной силой.
* Если расстояние больше определенного значения, то между объектами действуют гравитационные и магнитные силы, которые влияют на их движение.
* Рассчитывает силы, действующие на объекты, и вычисляет их ускорение.
* Обновляет скорость и координаты объектов с учетом полученного ускорения.

После завершения сценария, в котором пользователь может принимать прямое участие, то есть управлять космическим аппаратом, предлагается создать графики скорости, истории траектории с течением времени и отношением эффективности потраченного топлива.

Добавлю, что программа не является идеальным алгоритмом расчёта всех составляющих, она лишь демонстрирует поведение космического аппарата в различных условиях гравитации и взаимодействия с другими объектом. Создание всеобъемлющего комплексного программного обеспечения, направленного на расчёт затрат топлива, эффективности работы двигателя и работы с реальными показателями объектов нашей необъятной вселенной, является очень сложным процессом и для её создания требуются высококвалифицированные кадры, то есть, создание в одиночку такой комплексной программы практически невозможна в столь короткий срок длинной в год.

Это отступление послужит ознакомлением, так как без него может быть множество курьезов и вопросов, насчёт точности программы и в целом её работы. Она использует базовые школьные формулы в комплексе и не предоставляет более углубленного алгоритма для расчёта и прогноза движения ракеты.

## Описание способов взаимодействия пользователя с моделью

Данная программа была разработана с учетом возможности управления космическим аппаратом в аркадном режиме. Это означает, что пользователь сможет изменять скорость объекта и управлять им с помощью удобного интерфейса.

Кроме того, в программе предусмотрена возможность создания сценариев, которые можно просимулировать при заданных характеристиках объекта. Это позволяет пользователям создавать и тестировать различные сценарии. Для удобства работы с объектами в программе предусмотрен специальный редактор сценариев, который отображает окно с объектами в виде списка с их характеристиками.

В целом, описание управления есть в файле текстовом файле README, где так же есть и описание программы, её возможности и обновления.

# Описание интерфейса программы

## Обзор окна настройки

Как и в любой другой программе эта имеет окно настроек, перед запуском симуляции. На данный момент она является сырой, поэтому настройки выполняют не все возможные функции, которые я поставил себе в задачи. На данный момент меню позволяет:

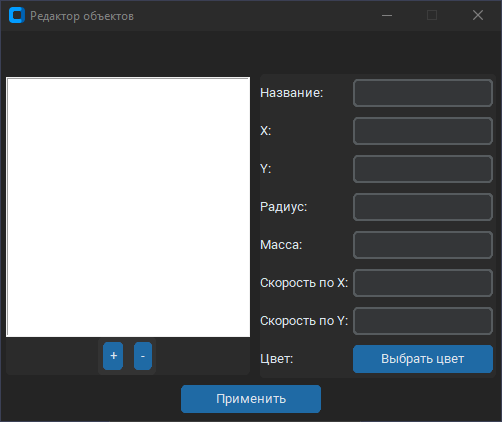
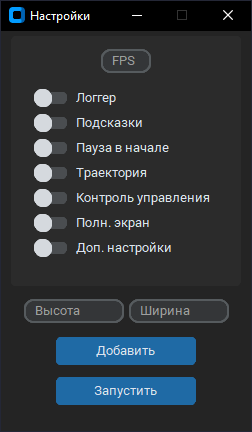
1. Ставить начальное количество кадров в секунду.
2. Включать логгер.
3. Включать историю траекторий тел.
4. Включать в начале паузу.
5. Включать контроль управления за космическим аппаратом.
6. Выбирать экран или выбрать своё разрешение в виде окна.
7. Добавлять объекты со своими характеристиками объектов.

Рисунок 2

Рисунок 1

# Использование программы

## Запуск программы

После завершения настройки программы, для запуска симуляции необходимо нажать на кнопку "Запустить". Программа запросит подтверждение на запуск, после чего симуляция начнется. По умолчанию, система объектов состоит из пародии на систему Земля-Луна со спутником, который можно управлять. В помощнике, представленном в виде виджетов, описано управление, которое поможет разобраться в управлении.

Управление космическим аппаратом осуществляется при помощи стрелок на клавиатуре. Кроме того, можно изменить силу импульса с помощью сочетания клавиш Ctrl + Стрелка вверх или вниз, в диапазоне от 0.01 до 1.01.

После завершения симуляции, можно выйти из программы, нажав кнопку "q". После этого программа предложит создать график скорости, а затем график истории траектории в виде трехмерного графика с третьей осью времени.

## Создание графиков

Графики, созданные с помощью библиотеки Matplotlib, представляют собой важный инструмент для визуализации данных. Эти графики являются удобным способом для исследования данных и анализа результатов.

Более того, с помощью Matplotlib можно создавать красивые и выразительные графики, которые могут быть использованы в научных исследованиях и для создания идей архитектурных моделей.

Для создания 2D-графика в Matplotlib используется функция plot(), которая позволяет строить линейные и нелинейные графики. С помощью этой функции можно настроить ширину линии, цвет и стиль линии, а также добавить название и метки осей.

Для создания 3D-графика в Matplotlib используется модуль mplot3d, который позволяет создавать трехмерные графики и моделировать архитектурные конструкции. С помощью функций plot\_surface() и plot\_wireframe() можно настроить ширину и цвет линий, цвет и прозрачность поверхности, а также добавить подписи к осям.

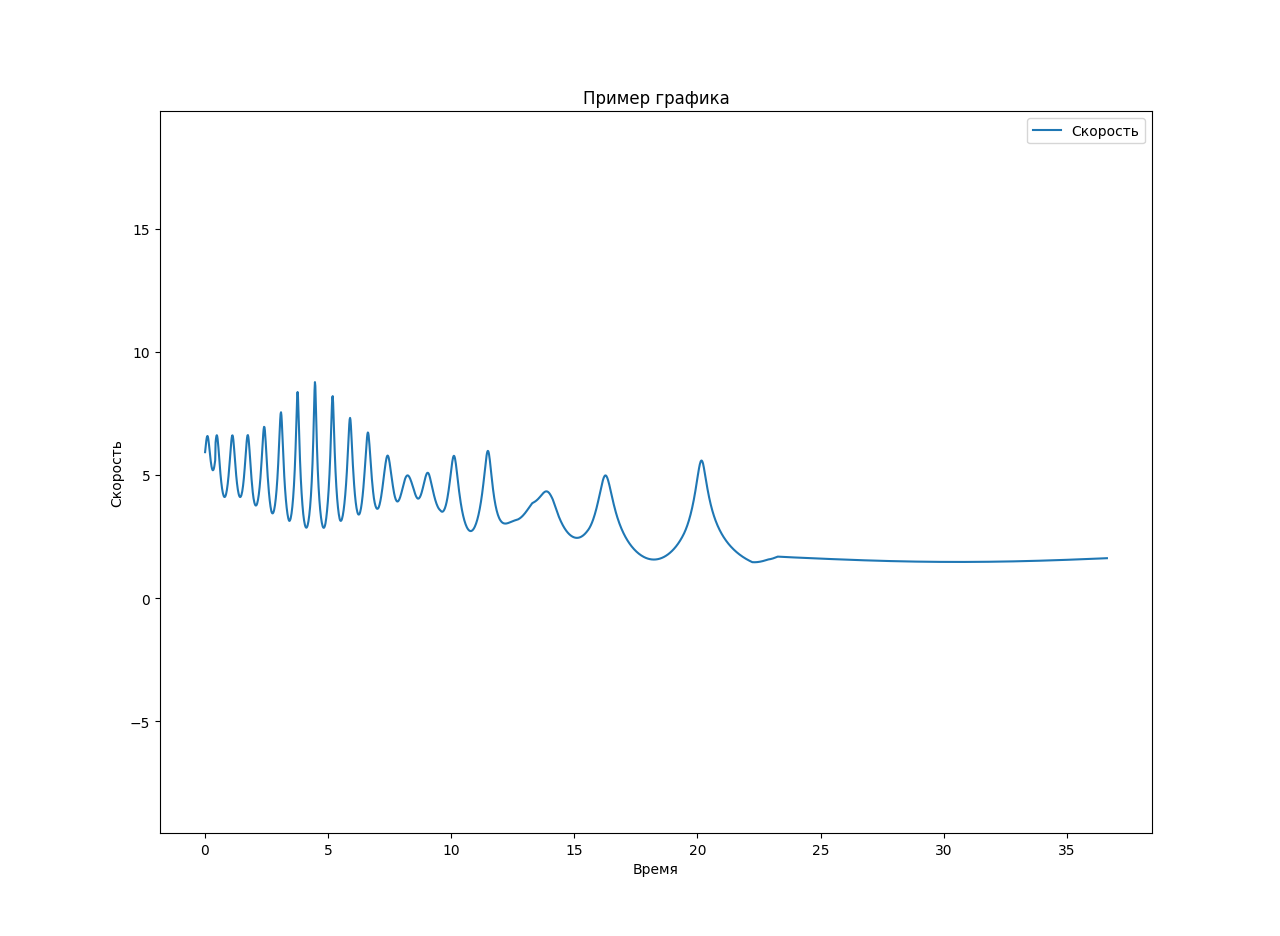
Приведены ниже два графика в качестве примера, один из которых является двухмерным, а другой – трехмерным:

Рисунок 3

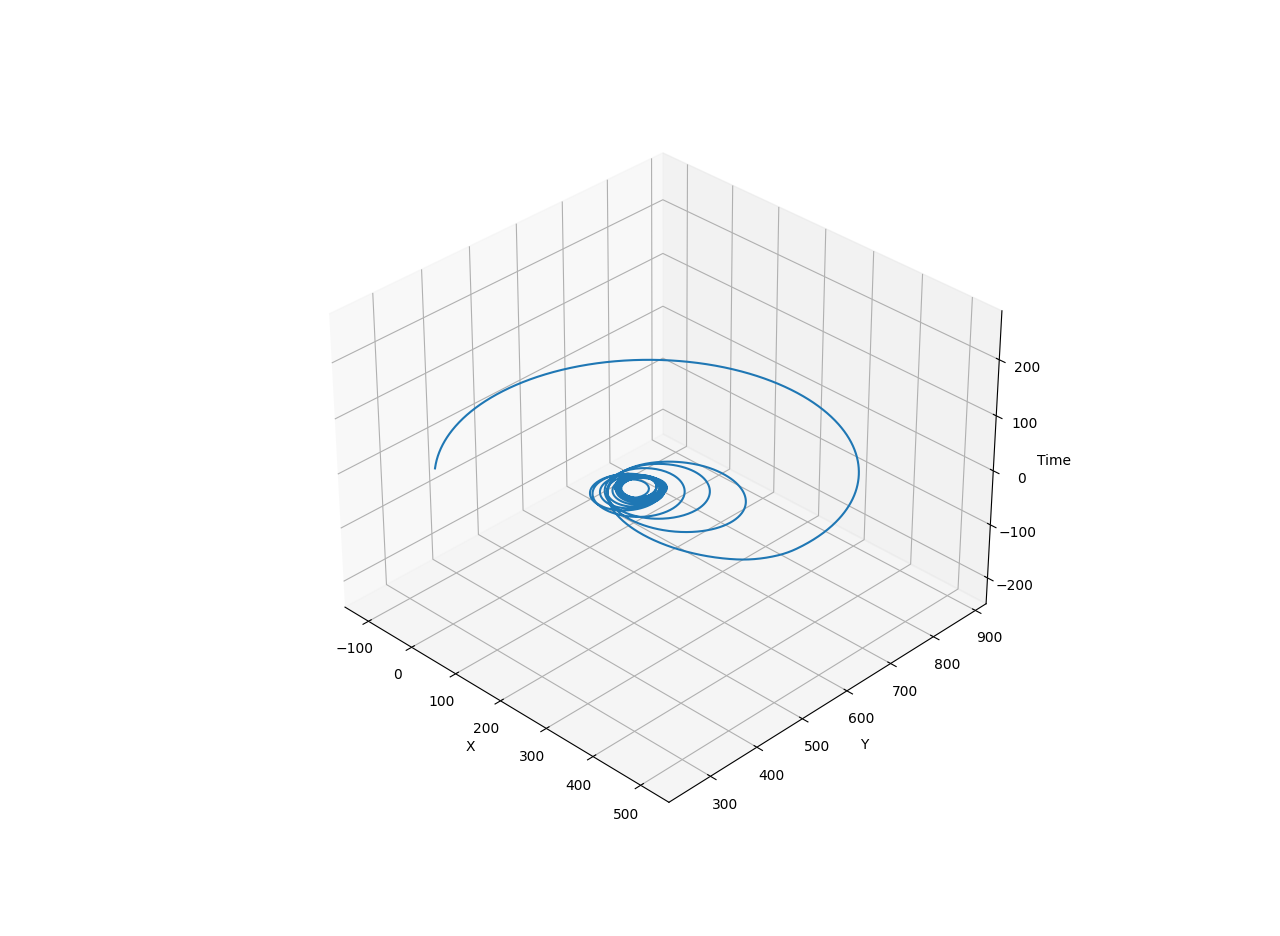


Рисунок 4

## Примеры использования

Программа может быть использована в качестве инструмента для изучения различных сценариев гравитационного взаимодействия. Ниже приведены примеры применения программы для изучения различных сценариев.

1. Движение планет: с помощью программы можно создать сценарий движения планет в солнечной системе. Можно изменять массу и начальные скорости планет, чтобы изучать, как они влияют на их орбиты и движение вокруг солнца.
2. Космические миссии: можно создать сценарий космической миссии на другую планету. Можно изучать, как изменения начальных условий влияют на траекторию космического корабля и время его полета.
3. Изучение законов Ньютона: использовать для изучения законов Ньютона, которые описывают движение материальных точек под воздействием силы гравитации. Можно изменять параметры системы, чтобы изучать, как они влияют на движение искомой точки.
4. Визуализация гравитационных сил: использовать для визуализации гравитационных сил, действующих между материальными точками. Это позволяет лучше понять, как эти силы взаимодействуют в системе.

Программа может быть использована для изучения множества других сценариев гравитационного взаимодействия и поможет лучше понять это явление в физике и космологии.

# Список иллюстраций

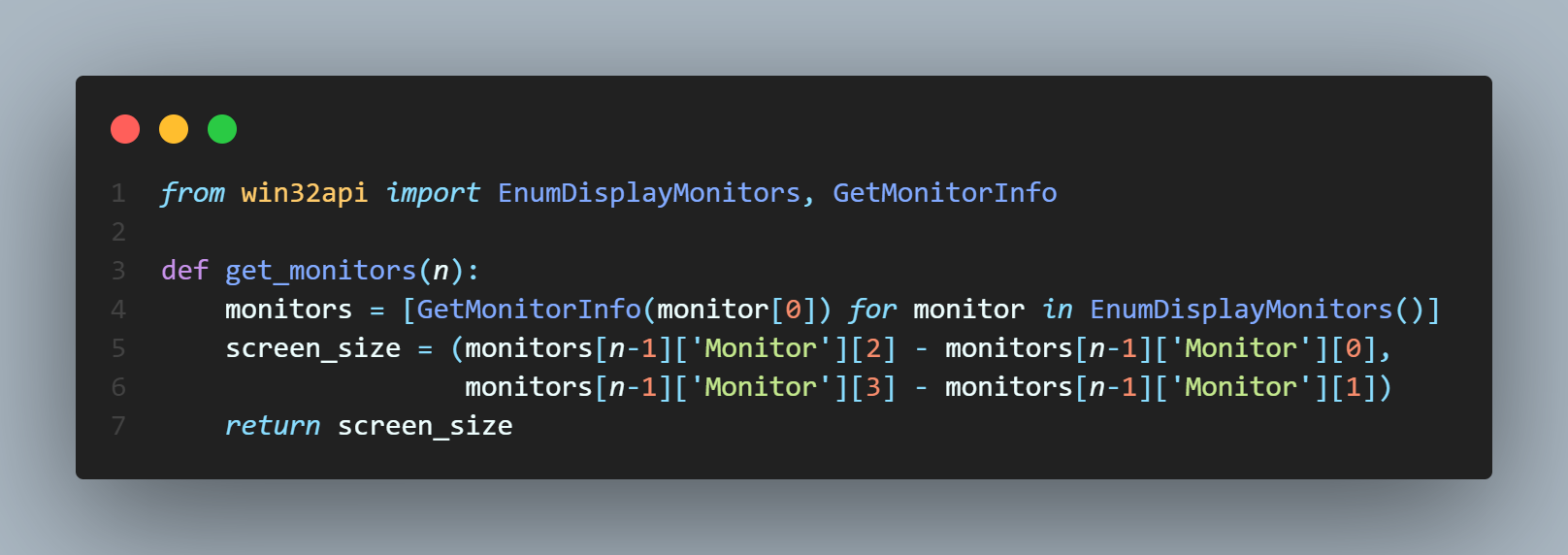
## Исходный код программы

### main.py

### window\_init.py

### renderGUI.py

### point.py

**monitor.py**

### physics.py

### controller.py

### constants.py

### ball.py - 1

### ball.py – 2

### ball.py - 3