

Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт–Петербургский государственный технологический институт

(технический университет)»

КУРСОВОЙ ПРОЕКТ

|  |  |  |  |
| --- | --- | --- | --- |
| **Направление подготовки** | | 09.03.01 Информатика и вычислительная техника | |
| **Направленность** | | Автоматизированные системы обработки информации  и управления | |
| **Факультет** | | Информационных технологий и управления | |
| **Кафедра** | | Систем автоматизированного проектирования и управления | |
| **Учебная дисциплина** | | Информационные технологии и программирование | |
| **Курс** | 1 | **Группа** | 423 |
| **Студент** | | Мухаметов Данил Ильнурович | |

|  |  |
| --- | --- |
| **Тема:** | Разработка программного комплекса для исследования поведения физических тел в гравитационном поле |

Заведующая кафедрой, проф. Т. Б. Чистякова

Лектор, доц. И. Г. Корниенко

Консультант, ст. преп. А. К. Федин

Исполнитель Д. И. Мухаметов

СОДЕРЖАНИЕ

[ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 3](#_Toc138832637)

[ВВЕДЕНИЕ 5](#_Toc138832638)

[1 Аналитический обзор 7](#_Toc138832639)

[1.1 Обзор и анализ программного обеспечения для исследования поведения физических тел в гравитационном поле. Сравнительная характеристика существующих программ. Обоснование актуальности разработки программного комплекса 7](#_Toc138832640)

[1.2 Общая характеристика и особенности программного комплекса для исследования поведения физических тел в гравитационном поле. 13](#_Toc138832641)

[1.3 Обзор и обоснование выбора инструментальных средств разработки программного комплекса для исследования поведения физических тел в гравитационном поле. 14](#_Toc138832642)

[2 Цель и задачи курсового проекта 30](#_Toc138832643)

[3 Технологическая часть 31](#_Toc138832644)

[3.1 Формализованное описание процесса поведения физических тел в гравитационном поле 31](#_Toc138832645)

[3.2 Постановка задачи отрисовки. 32](#_Toc138832646)

[3.3 Разработка функциональной структуры программного комплекса для демонстрации поведения физических тел в поле притяжения. 32](#_Toc138832647)

[3.4 Создание алгоритма взаимодействия объектов 33](#_Toc138832648)

[3.5 Разработка структуры интерфейса пользователя программного комплекса для демонстрации поведения физических тел в поле притяжения 34](#_Toc138832649)

[3.6 Описание структур данных и алгоритмов 36](#_Toc138832650)

[3.7 Описание структур данных и алгоритмов (формат представления данных в памяти и на внешних носителях) 40](#_Toc138832651)

[3.8 Тестирование программного комплекса (на добавление новых объектов, новых источников притяжения). 41](#_Toc138832652)

[ХАРАКТЕРИСТИКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ 44](#_Toc138832653)

[ВЫВОДЫ ПО КУРСОВОМУ ПРОЕКТУ 45](#_Toc138832654)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46](#_Toc138832655)

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В данном отчёте о курсовом проекте применяются следующие сокращения и обозначения:

ОС – операционная система

ЯП – язык программирования

Симуляция – моделирование физических процессов на компьютере.

ООП-язык — язык, построенный на принципах объектно-ориентированного программирования.

Кроссплатформенность - способность программного обеспечения работать с несколькими аппаратными платформами или операционными системами.

Скрипты — это текстовые файлы в языке программирования, которые содержат набор команд. Скрипты используются в программировании для автоматизации выполнения рутинных задач, для выполнения последовательности действий над данными или для автоматической обработки информации.

Фреймворк – структура и набор инструментов, предназначенных для разработки программного обеспечения, упрощающих задачи программиста.

Плагины - независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей.

Аддоны - необязательные данные для компьютерного программного обеспечения, изменяющие функциональность или настройки и могущие распространяться отдельно от самого ПО в виде файлов или на носителях.

Библиотека – сборник программных компонентов, предназначенных для повторного использования в различных проектах.

UI – user interface (пользовательский интерфейс) – визуальное представление программы или приложения, с которым пользователь взаимодействует.

Физический движок - Компьютерная программа, которая производит компьютерное моделирование физических законов реального мира в виртуальном мире, с той или иной степенью аппроксимации. Чаще всего физические движки для физического моделирования используются не как отдельные самостоятельные программные продукты, а как составные компоненты других программ.

Графический движок - Промежуточное программное обеспечение, программный движок, основной задачей которого является визуализация двухмерной или трёхмерной компьютерной графики.

Рендеринг - термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы. Здесь модель - это описание любых объектов или явлений на строго определённом языке или в виде структуры данных.

Паттерн – рефакторинговая и архитектурная конструкция, описывающая типовые проблемы и способы их решения.

Портируемость - В программировании под портированием понимают адаптацию некоторой программы или её части, с тем чтобы она работала в другой среде, отличающейся от той среды, под которую она была изначально написана с максимальным сохранением её пользовательских свойств.

IDE – интегрированная среда разработки – Программное обеспечение, предоставляющее разработчику инструменты для написания, отладки и тестирования кода.

Виджет - элемент интерфейса — примитив графического интерфейса пользователя, имеющий стандартный внешний вид и выполняющий стандартные действия.

API – Application Programming Interface (интерфейс программирования приложений) – Набор методов и функций, предоставляемых программным обеспечением для взаимодействия с другими программами.

ЦА – целевая аудитория.

ЦП – центральный процессор.

ГП – графический процессор.

Отладка - этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки.

Фича — это какая-либо полезная особенность, специфическая черта, дополнение в программном обеспечении

Коллизия – c английского языка столкновение. В разработке игр имеет смысл как моменты пересечения моделей.

ВВЕДЕНИЕ

Физика является одной из наиболее фундаментальных наук, которая занимается изучением природы и ее процессов. Ее результаты имеют широкое применение в различных областях, и поэтому изучение физики является ключевым фактором в познании мира и развитии технологий.

Одним из основных направлений физики является изучение гравитационного поля и его взаимодействия с физическими телами. Эта область науки стала особенно актуальной в связи с развитием космических программ и изучением космических объектов. Физика гравитационных полей позволяет описать такие явления, как движения небесных тел, взаимодействия галактик и звезд.

Один из наиболее распространенных и эффективных методов исследования физики — это симуляции, т.е. моделирование физических процессов на компьютере. Они позволяют более точно описывать состояние системы и ее эволюцию во времени, в то время как аналитические методы, которые рассчитываются вручную, могут быть ограничены в своей применимости и точности.

Для применения симуляций в исследованиях поведения физических тел в гравитационном поле необходимы программные инструменты, которые позволяют легко управлять и настраивать параметры моделирования, а также собирать и анализировать результаты. Разработка программного комплекса, способного реализовывать симуляции на основе современных методов и алгоритмов, является ключевой задачей данного курсового проекта.

При правильном применении симуляций возможно провести детальные исследования различных сценариев поведения физических тел в гравитационном поле. Это позволит углубить наши знания об устройстве и работе Вселенной, о распределении материи в галактиках и других космических объектах. Такие исследования имеют значимость не только в научных целях, но и в области практического применения, например, в аэрокосмической промышленности.

Важность программных средств для моделирования физических процессов в гравитационном поле определяется в их способности достигать большей точности и масштабируемости. Разработанный программный комплекс сможет стать инструментом для исследований на уровне институтов, для наглядной демонстрации законов физики.

Получение результатов исследования и их анализ является одним из ключевых аспектов разработки программного комплекса. Для этого необходимо использовать различные методы визуализации результатов, которые позволяют выявлять и анализировать закономерности и особенности различных сценариев поведения физических тел в гравитационном поле.

В итоге, разработанный программный комплекс будет универсальным инструментом для исследования различных сценариев поведения физических тел в гравитационном поле, что существенно облегчит работу исследователей в этой области науки.

1 Аналитический обзор

1.1 Обзор и анализ программного обеспечения для исследования поведения физических тел в гравитационном поле. Сравнительная характеристика существующих программ. Обоснование актуальности разработки программного комплекса

Разработка программного обеспечения — это процесс создания программного продукта, который будет использоваться для решения определенных задач. В современном мире программное обеспечение является неотъемлемой частью жизни каждого человека, ведь оно используется в различных сферах, в том числе в образовании, медицинском деле, торговле, банковском секторе, технологических компаниях и научных комплексах и многих, многих других сферах жизнедеятельности.

Разработка программного обеспечения — это сложный и многогранный технологический процесс, который должен пройти несколько этапов, начиная от идеи и заканчивая реализацией готового продукта. Разработка программного обеспечения может осуществляться как небольшой командой разработчиков, так и крупными IT-компаниями.

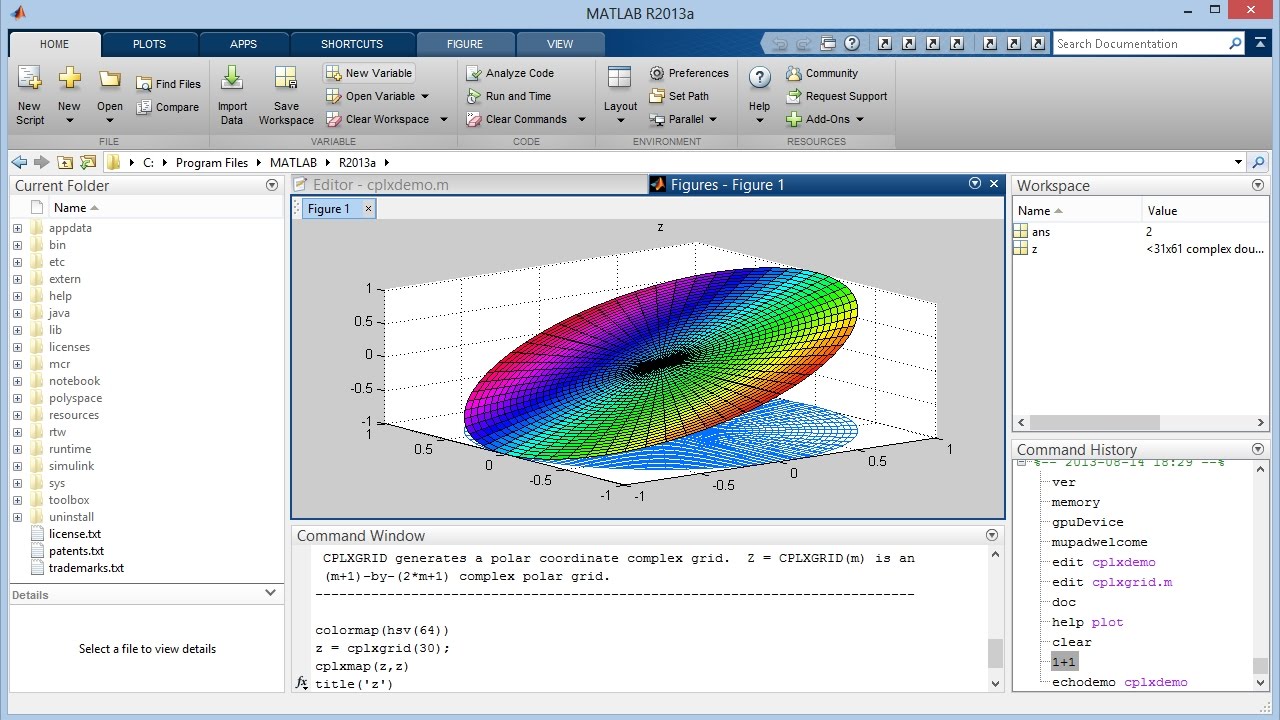
Ключевым этапом разработки программного обеспечения является создание проекта, в рамках которого определяются требования к программному продукту, устанавливаются задачи, расписывается функционал, определяются временные рамки и бюджет проекта. После этого приступают к реализации проекта, где осуществляется разработка архитектуры, дизайн и программирование. Также очень важным этапом является тестирование программного обеспечения, в ходе которого выявляются ошибки и неточности, которые могут возникнуть в процессе использования продукта. И, наконец, после тестирования программное обеспечение выпускается на рынок и начинается обновление и техническая поддержка продукта.

Основной целью разработки программного обеспечения является создание продукта, который будет эффективно решать задачи, наиболее оптимально удовлетворять потребности пользователей, быть удобным и интуитивно понятным в использовании. Как уже было сказано, программное обеспечение имеет широкое применение в сферах знаний и общественной жизни. Оно способно значительно упростить и оптимизировать работу пользователей, повысить их эффективность, а также улучшить качество производимой работы.

В научной сфере, а конкретнее в сфере компьютерных симуляций, существует несколько крупных игроков. MATLAB - программное обеспечение для вычислительной математики, которое может использоваться для моделирования физических процессов. ANSYS – Пользователи ANSYS могут создавать модели и симуляции многих процессов, включая моделирование динамики тел, оптимизацию конструкции, анализ напряженно-деформированного состояния. Autodesk 3ds Max – профессиональная программа для компьютерной графики 3D для создания 3D-анимации, моделей. 3ds Max отличается от своих предшественников тем, что он специализирован на 3D графике, когда другие на симуляции физики.

Таблица 1 – Преимущества и недостатки приложения «MATLAB»

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| MATLAB имеет мощную систему визуализации данных, которая позволяет легко визуализировать и анализировать данные в различных форматах. | Необходимо знать основы программирования для работы с MATLAB, так как для описания работы программ, требуется писать свои скрипты |
| MATLAB также имеет большую поддержку сообщества, которое делает доступным множество бесплатных и платных ресурсов для обучения и развития навыков программирования. | MATLAB распространяется на платной основе. |
| С помощью MATLAB можно моделировать и симулировать широкий спектр физических процессов, включая взаимодействие частиц, электромагнитные поля, тепло- и массообмен, гидродинамику, физику твердого тела и многие другие. | MATLAB разработан не отечественным разработчиком. |

 Рисунок 1 – Демонстрация программы «MATLAB».

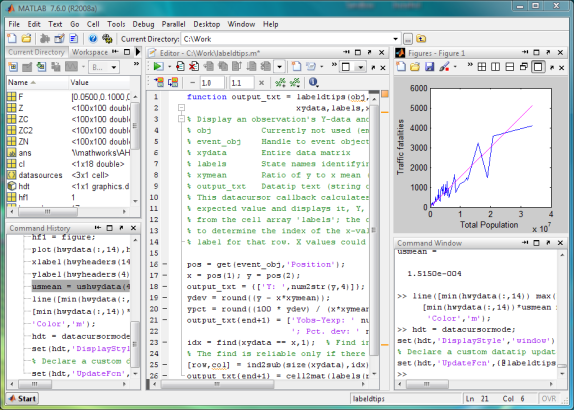


Рисунок 2 – Пример процесса работы с «MATLAB».

Таблица 2 – Преимущества и недостатки приложения «ANSYS»

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| ANSYS является одним из наиболее мощных инструментов для моделирования, симуляций и инженерных расчетов различных физических процессов. | ANSYS может не всегда быть оптимальным выбором для расчетов небольших систем или низкобюджетных проектов. |
| ANSYS имеет широкий спектр инструментов для создания высококачественных моделей, включая системы CAD, импорт/экспорт файлов, автоматизированные процессы построения моделей и многие другие. | Для работы с ANSYS требуются специализированные знания и навыки, что может ограничить доступность для новичков и людей без должной подготовки. |
| ANSYS позволяет получить высокую точность результатов симуляции при работе с большими объемами данных. | ANSYS может быть сложным и многоэтапным, что требует времени на изучение инструментов и процесса симуляции. |
| ANSYS включает в себя модули для симуляции различных физических процессов, включая механику, теплопередачу, акустику и электромагнитную совместимость | ANSYS может быть дорогим в приобретении и использовании. |
| ANSYS может использоваться в различных отраслях промышленности, включая авиацию, автомобильную и кораблестроение, энергетику, машинное производство и другие. | ANSYS разработан не отечественным разработчиком. |

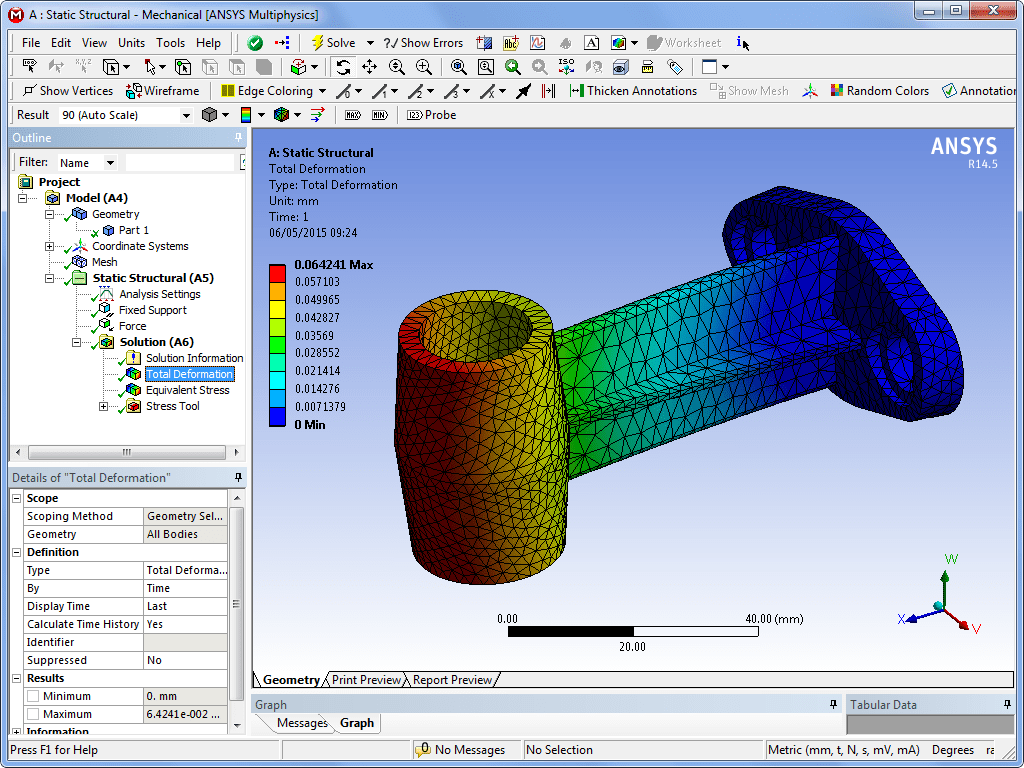


Рисунок 3 – Демонстрация программы «ANSYS».

Таблица 3 – Преимущества и недостатки приложения «Autodesk 3ds Max»

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Autodesk 3ds Max имеет множество инструментов для создания сложных 3D-моделей, анимаций и виртуальных миров. | Autodesk 3ds Max — это сложное программное обеспечение с множеством функций, что может быть непросто для новичков и влечет за собой длительный процесс обучения. |
| Autodesk 3ds Max поддерживает различные форматы файлов, включая 3ds, obj, FBX и другие. | Autodesk 3ds Max может быть дорогим в приобретении и использовании. |
| Autodesk 3ds Max Max имеет большое сообщество пользователей, что делает возможным быстрое решение проблем и заимствование идей. | Autodesk 3ds Max для работы требует множество плагинов и аддонов |
|  |  |
|  |  |
|  |  |
|  | Autodesk 3ds Max разработан не отечественным разработчиком. |

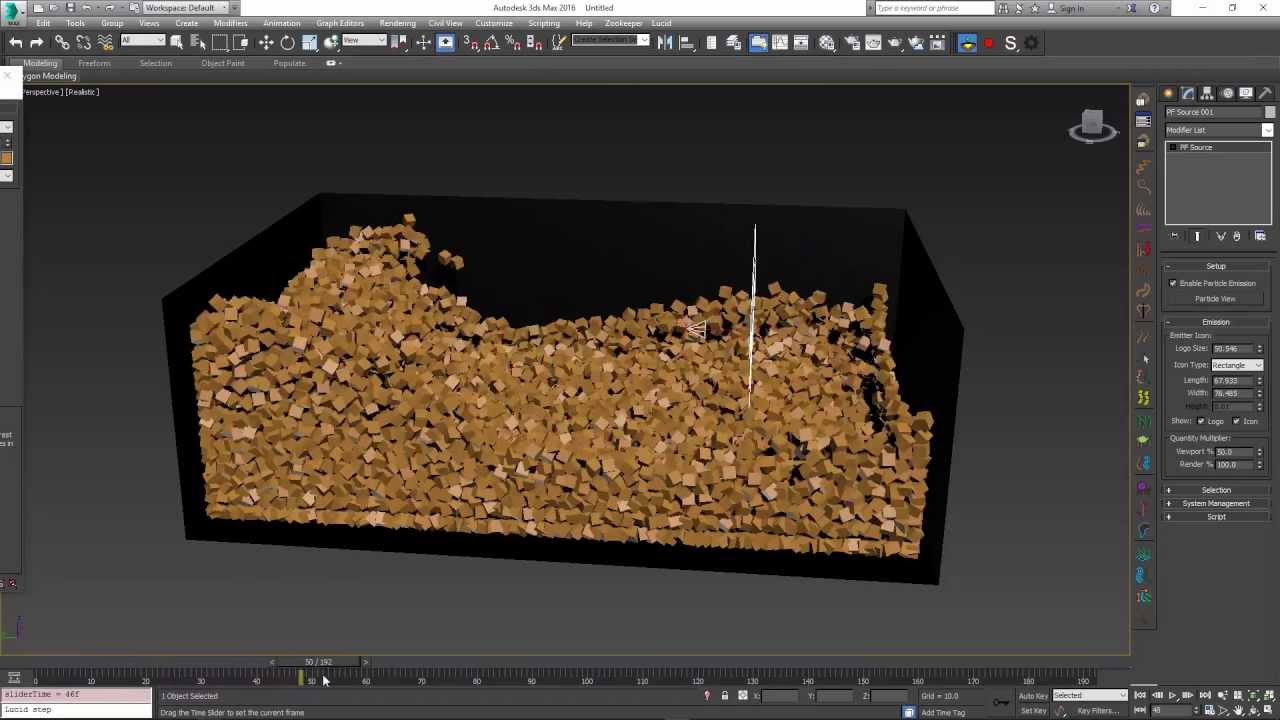


Рисунок 4 –Пример симуляции в программе «Autodesk 3ds Max».

Таким образом, можно сделать вывод, что приложения подобного рода актуальны в разработке, поскольку существующие приложения слишком сложны для в обращении и требуют хороших знаний в предметной области, а также разработаны не отечественным разработчиком и требуют покупки лицензии. Данная разработка актуальна, поскольку разрабатываемый мной проект позволяет без каких-либо материальных и трудозатрат продемонстрировать поведение тел в гравитационном поле.

1.2 Общая характеристика и особенности программного комплекса для исследования поведения физических тел в гравитационном поле.

Приложения и программные комплексы, симулирующие окружающий нас мир, играют важнейшую роль как в научной, так и в образовательной среде. Эти приложения обеспечивают удобную и эффективную систему для проведения опытов, упрощая и автоматизируя множество задач, связанных с восприятием и анализом информации.

Одной из особенностей реализации таких приложений является создание удобного и интуитивно понятного пользовательского интерфейса (UI), который позволяет студентам легко взаимодействовать с системой и задавать свои параметры объектов. Также необходимо разработать оптимизированные алгоритмы обработки физики, которые лягут в основу физического движка, который в свою очередь возьмёт на себя роль модуля, руководствующим всеми взаимодействиями между объектами.

Программный комплекс для симуляций — это важный инструмент, который используется для моделирования реальных процессов и событий, а также для тренировки и обучения людей в различных областях. Для его реализации потребуется множество систем, взаимодействующих заодно. Физический движок будет просчитывать передвижения и столкновения объектов, после чего полученные данные будут отрисованы графическим движок, и между тем UI будет позволять удобно настраивать параметры симуляции.

1.3 Обзор и обоснование выбора инструментальных средств разработки программного комплекса для исследования поведения физических тел в гравитационном поле.

Одно из важнейших мест в разработке приложения занимает выбор платформы. Это рождает вопросы о направленности проекта и о его возможной кроссплатформенности. Данный проект буде направлен на пользователей Windows. Это обусловлено удобством работы с Window обычному пользователю и из-за численного превосходства в пользователях в сравнении с другими ОС.

C++ — это высокоуровневый язык программирования, который широко используется для разработки системного и прикладного программного обеспечения. Он является расширением языка C и предоставляет мощные возможности, такие как объектно-ориентированное программирование (ООП), обработка исключений, шаблоны, многопоточность и многое другое.

DirectX 11, разработанный компанией Microsoft, также является широко распространенным графическим API. DirectX 11 включает в себя функции рендеринга, графический контент, звуковые эффекты и другие периферийные устройства для компьютерных игр и других приложений. Главным преимуществом DirectX11 является его мощная система рендеринга, которая позволяет создавать высококачественные 3D-изображения и передавать их на экран без задержек. DirectX 11 также поддерживает большой объем визуальных эффектов и средства оптимизации, которые позволяют управлять ресурсами, уменьшать нагрузку на ЦП и увеличивать уровень производительности.

ImGui (Immediate Mode Graphical User Interface) — это библиотека для создания пользовательских интерфейсов в режиме немедленного режима. Она разработана специально для работы с OpenGL, DirectX и другими графическими API, и предназначена для помощи в создании инструментария для разработчиков. ImGui отличается от традиционных GUI-библиотек тем, что не использует концепцию "виджета" и не требует от программиста создавать и хранить множество объектов GUI.

Выбор такого стека технологий для реализации программного комплекса для исследования поведения физических тел в гравитационном поле обоснован следующим образом. C++ обеспечивает точность с использования памяти, и быстроту исполнения разработанного программного комплекса, DirectX 11 предоставляет ориентированные на Window инструменты для отрисовки изображения, что упрощает процесс написание графического движка для нашей программы, ImGui идеально подходит для DirectX 11 и С++, так как был специально написан под них, что упрощает работу. Этот стек технологий позволяет создавать высокопроизводительные и надежные приложения, которые эффективно справляются со своими задачами.

Рассмотрим плюсы и минусы ЯП для разработки под ОС Windows: С, C++ и Java. Важными критериями являются: многопоточность, управления памятью, наличие библиотек и фреймворков, расширяющие возможности языка.

**C 17**

С — это компилируемый язык программирования, разработанный в начале 1970-х годов. Он был создан Деннисом Ритчи в лаборатории Bell в рамках разработки операционной системы UNIX. C был разработан с учетом эффективности, портируемости и низкоуровневого программирования, что позволило использовать его для разработки системного программного обеспечения и драйверов устройств.

**Плюсы C 17:**

–Эффективность: C является низкоуровневым языком программирования, позволяющим разработчикам полностью контролировать использование ресурсов компьютера, таких как память и процессорное время. Это делает его особенно подходящим для разработки производительного и оптимизированного по ресурсам программного обеспечения.

– Портируемость: Код, написанный на C, может быть скомпилирован и запущен на различных платформах с небольшими или без изменений. Это обеспечивает высокую портируемость программ, позволяя разработчикам создавать кроссплатформенное программное обеспечение.

–Близость к аппаратуре: C предоставляет возможность непосредственного взаимодействия с аппаратным обеспечением компьютера. Это позволяет оптимизировать код для конкретной архитектуры и использовать специфические возможности и функции аппаратуры.

–Богатый выбор библиотек и фреймворков: существует огромное количество библиотек и фреймворков, разработанных на C, обеспечивающих широкий спектр функциональности, от графики и сетевого взаимодействия до обработки данных и многое другое. Это позволяет разработчикам использовать готовые решения и ускорить разработку программного обеспечения.

-Широкий выбор обучающего материала: существует бесконечно много видеоуроков, лекций, семинаров. Язык входит в учебную программу практически в каждой школе.

**Минусы C 17:**

– Управление памятью: в отличие от более современных языков, таких как Java или Python, в C отсутствует автоматическое управление памятью. Разработчику необходимо самостоятельно выделять и освобождать память для объектов, что может привести к ошибкам и проблемам с утечками памяти.

– Отсутствие безопасности типов: В C отсутствует строгая проверка типов, что может привести к ошибкам во время выполнения программы, особенно если некорректно приводить типы данных или использовать указатели.

– Ограниченная поддержка ООП: В отличие от C++, C не имеет полной поддержки объектно-ориентированного программирования (ООП). Хотя некоторые концепции ООП могут быть реализованы в C, они не являются встроенными и требуют дополнительной работы.

–Ограниченные средства абстракции: C предоставляет базовые средства абстракции, но они не настолько мощны и удобны, как в некоторых более современных языках. Некоторые операции и конструкции, которые могут быть легко выполнены в других языках, могут требовать большего количества кода и усилий в C.

–Отсутствие некоторых современных функций: C является языком программирования, который разработан давно, и в нем отсутствуют некоторые современные функции и конструкции, доступные в более новых языках. Это может затруднить разработку программного обеспечения с использованием последних инноваций и подходов.

**C++ 22**

C++ – это объектно-ориентированный язык программирования, который используется для создания приложений с высокой производительностью. Он является одним из самых популярных языков программирования и используется для различных целей, включая мобильную разработку.

**Плюсы С++:**

–Высокая производительность: С++ обеспечивает низкоуровневый доступ к аппаратным ресурсам, что позволяет создавать мобильные приложения с высокой производительностью.

–Платформенная независимость: С++ позволяет разрабатывать кроссплатформенные приложения, поддерживая различные операционные системы и архитектуры процессоров.

–Большое сообщество разработчиков: С++ имеет обширное и активное сообщество разработчиков, что обеспечивает доступ к обширным ресурсам и поддержке.

–Богатый набор инструментов: С++ предоставляет разнообразные инструменты и библиотеки для разработки мобильных приложений, позволяющие расширить функциональность и повысить эффективность разработки.

–Управление памятью: С++ предоставляет возможность явного управления памятью, что дает разработчикам больше контроля над использованием ресурсов и оптимизацией памяти.

**Минусы С++:**

–Сложность и подверженность ошибкам: С++ требует более аккуратного и тщательного программирования, что может привести к возникновению ошибок и уязвимостей.

–Отсутствие автоматического управления памятью: С++ не предоставляет встроенного механизма автоматического управления памятью, что может привести к утечкам памяти и ошибкам работы с памятью.

–Ограниченная переносимость: С++ является языком, тесно связанным с конкретной платформой, что ограничивает переносимость приложений между различными операционными системами и архитектурами процессоров.

–Большая сложность обучения: С++ является более сложным в изучении и требует от разработчиков более глубокого понимания языка и его особенностей.

**Java 19**

Java – это объектно-ориентированный язык программирования, который был разработан в 1995 году компанией Sun Microsystems (позднее приобретенной компанией Oracle). Java был создан как язык для написания кроссплатформенных приложений, которые могут быть запущены на любой операционной системе, которая поддерживает JVM.

**Плюсы Java:**

– Платформа независимость: Java разработан для работы на любой операционной системе, что позволяет создавать приложения, которые могут быть запущены на различных платформах без необходимости перекомпиляции кода.

– Обширная стандартная библиотека: Java поставляется с обширной стандартной библиотекой, которая предоставляет готовые инструменты и функции для разработки различных типов приложений.

–Богатая экосистема инструментов: Java обладает разнообразными инструментами разработки, IDE и фреймворками, которые упрощают и ускоряют процесс создания клиентского мобильного приложения.

–Объектно-ориентированное программирование (ООП): Java поддерживает принципы ООП, такие как инкапсуляция, наследование и полиморфизм, что способствует созданию модульного и гибкого кода.

–Высокий уровень безопасности: Java имеет встроенные механизмы безопасности, такие как проверка типов во время компиляции и управление памятью, что способствует созданию надежных и защищенных мобильных приложений.

–Многопоточность: Java предоставляет инструменты и синтаксис для создания многопоточных приложений, что позволяет эффективно использовать ресурсы и повышает производительность.

**Минусы Java:**

–Использование памяти: Java требует дополнительного объема памяти для работы виртуальной машины Java (JVM), что может привести к более высокому потреблению ресурсов в сравнении с некоторыми другими языками.

–Отсутствие полной контроля над памятью: В Java отсутствует прямой доступ к памяти, так как управление памятью автоматически осуществляется сборщиком мусора. Это может ограничить возможности оптимизации и управления ресурсами в некоторых случаях.

–Ограниченная поддержка функционального программирования: Java не обладает полноценной поддержкой функционального программирования.

–Производительность: в сравнении с некоторыми низкоуровневыми языками как с++ или с, Java может быть менее производительным в некоторых высоконагруженных и ресурсоемких сценариях.

Таблица 4 – Сравнительная таблица ЯП для реализации клиента (мобильной разработки): С, C++, Java. Часть данных для сравнения скорости выполнения программ на ЯП были взяты с 2 сайтов: “ <https://www.opennet.ru/opennews/art.shtml?num=30784> ”, “ <https://itnan.ru/post.php?c=1&p=347688> ”.

|  |  |  |  |
| --- | --- | --- | --- |
| Особенности | С 17 | C++ 22 | Java 19 |
| Объектно-ориентированный подход | - | + | + |
| Наследование | - | + | + |
| Автоматическое управление памятью | - | - | + |
| Кроссплатформенность | +  (Compilers) | +  (Compilers) | +  (Java Virtual Machine) |
| Высокая скорость выполнения | + | + | - |
| Богатая экосистема | + | + | + |

Таким образом, выбор ЯП С++ оправдан наличием полноценного доступа к памяти, своим быстродействием и наличием полноценного ООП.

Выбор графического API, будет основан на похожих признаках. Важной особенностью должно быть удобство работы OC Windows, и простота работы с С++.

Рассмотрим 3 самых популярных графических API: OpenGL, DirectX11 и Vulkan.

**OpenGL 2.3.1**

OpenGL — это бесплатный, открытый стандарт графического программирования, который поддерживается на всех основных операционных системах. OpenGL используется во многих проектах, не только в играх, но и в научных и инженерных предприятиях, в которых требуется высокоэффективная визуализация данных. Одним из главных преимуществ OpenGL является его кроссплатформенность и универсальность - он может работать на устройствах с различными аппаратными требованиями, такими как мобильные устройства, настольные компьютеры и серверы. Кроме того, OpenGL обеспечивает широкую совместимость с различными устройствами и позволяет программистам работать на языках программирования, таких как C++, Java и Python.

**Плюсы OpenGL:**

–Кроссплатформенность: OpenGL поддерживается на различных операционных системах, включая Windows, macOS, Linux и другие. Это позволяет создавать приложения, которые работают на разных платформах без необходимости переписывать код с нуля.

–Широкая поддержка и активное сообщество: OpenGL является открытым стандартом и имеет широкую поддержку как со стороны производителей видеокарт, так и разработчиков. Существует большое сообщество разработчиков, которые делятся опытом и помогают другим пользователям.

–Гибкость и расширяемость: OpenGL предоставляет гибкий и расширяемый набор функций, что позволяет разработчикам создавать разнообразные графические эффекты и реализовывать свои собственные расширения.

–Удобство работы с С++: это связано с тем, что OpenGL API сам по себе написан на языке C и обладает нативной поддержкой для C++. Множество библиотек и фреймворков, предназначенных для работы с OpenGL, также предоставляют удобные интерфейсы на C++.

**Минусы OpenGL:**

–Уровень абстракции: OpenGL является относительно низкоуровневым API, требующим от разработчика более прямого контроля над графическими ресурсами. Это может потребовать больше усилий и времени для разработки сложных графических приложений.

–Отсутствие встроенной поддержки: OpenGL не предоставляет встроенной поддержки для некоторых распространенных функций, таких как обработка пользовательского интерфейса (UI) и аудио. Разработчику может потребоваться использовать дополнительные библиотеки или API для реализации этих функций.

–Устаревание: В сравнении с Vulkan, более новым и низкоуровневым графическим API, OpenGL может быть считаем устаревшим. Vulkan предлагает более высокую производительность и больший контроль над аппаратными ресурсами, однако он также более сложен в использовании.

**DirectX 11**

**DirectX** — DirectX 11, разработан в Microsoft, пользуется широкой популярностью Основные преимущества DirectX 11 включают мощную систему рендеринга для создания высококачественных 3D-изображений, поддержку визуальных эффектов и оптимизацию для увеличения производительности приложений.

**Плюсы DirectX:**

–Производительность: DirectX 11 обеспечивает высокую производительность графики и рендеринга благодаря использованию современных технологий и оптимизаций. Он позволяет разработчикам использовать мощные возможности современных графических аппаратных устройств для достижения высокой производительности приложений.

–Инструменты разработки: DirectX 11 предлагает разнообразные инструменты разработки, такие как DirectX SDK и Visual Studio, которые облегчают процесс разработки и отладки графических приложений. Эти инструменты позволяют разработчикам эффективно использовать возможности API и оптимизировать свои приложения.

–Широкая поддержка: DirectX 11 является популярным API и имеет широкую поддержку со стороны производителей видеокарт и разработчиков. Это обеспечивает стабильность и совместимость с различным оборудованием и операционными системами.

–Удобство работы с С++: это связано с тем, что DirectX сам по себе является набором API для разработки приложений под операционные системы Windows, что требует низкоуровневого языка на подобии С++.

**Минусы DirectX:**

–Ограничение на платформу: DirectX 11 является проприетарным API, разработанным для операционных систем Windows. Это ограничивает его использование только на платформе Windows, что может быть проблемой, если требуется кроссплатформенная поддержка.

–Сложность и обучение: Использование DirectX 11 может требовать от разработчиков более высокого уровня знаний и навыков в сравнении с другими графическими API. Он предоставляет широкий набор функций и возможностей, что может быть сложно для начинающих разработчиков.

–Ограниченная поддержка более старых версий Windows: Некоторые функции и возможности DirectX 11 могут быть недоступны в более старых версиях операционной системы Windows. Это может ограничить целевую аудиторию приложения и потребовать дополнительного усилия для поддержки старых версий Windows.

**Vulkan 1.3**

Vulkan - открытый стандарт графического API, который разработала группа Khronos. Он предстовляет собой эволюцию концепции OpenGL и обеспечивает полную многопоточность в процессе работы. Vulkan используется в игровых проектах, таких как Dota 2, Doom и Battlefield 1 и предоставляет доступ к широкому спектру инструментов графического программирования, включая управление потоками данных, распределение ресурсов и геометрические преобразования.

**Плюсы Vulkan:**

–Высокая производительность: Vulkan предоставляет более низкоуровневой доступ к аппаратным ресурсам, что позволяет разработчикам более точно контролировать процесс рендеринга и оптимизировать его для достижения максимальной производительности. –Инструменты разработки: DirectX 11 предлагает разнообразные инструменты разработки, такие как DirectX SDK и Visual Studio, которые облегчают процесс разработки и отладки графических приложений. Эти инструменты позволяют разработчикам эффективно использовать возможности API и оптимизировать свои приложения.

–Мультиплатформенность: аналогично OpenGL, Vulkan поддерживается на различных операционных системах, включая Windows, macOS, Linux и Android. Это обеспечивает возможность создания кроссплатформенных приложений с использованием одного и того же кода.

–Масштабируемость: Vulkan разработан с учетом современных многопоточных систем. Он предоставляет возможность распараллеливания команд и задач рендеринга, что позволяет эффективно использовать множество ядер процессора и достичь высокой производительности на современных многопоточных системах.

–Большой контроль над ресурсами: Vulkan предоставляет разработчикам полный контроль над управлением памятью и ресурсами, такими как буферы, текстуры и шейдеры. Это позволяет эффективно управлять памятью, минимизировать задержки и улучшить производительность.

**Минусы Vulkan:**

–Сложность использования: По сравнению с OpenGL, Vulkan имеет более низкоуровневую архитектуру и требует от разработчиков больше усилий для написания кода. Он предоставляет больше возможностей, но также требует более глубокого понимания графического программирования и API.

–Большая кодовая база: поскольку Vulkan предоставляет более низкоуровневой доступ к аппаратным ресурсам, его кодовая база может быть более обширной по сравнению с более абстрактными API, такими как OpenGL. Это может потребовать больше времени и усилий на разработку и отладку приложений.

–Ограниченная поддержка: В отличие от OpenGL, который имеет более длительную и широкую историю, Vulkan является относительно новым API. В связи с этим, поддержка Vulkan может быть ограничена на некоторых платформах или устройствах, особенно в старых операционных системах.

Таблица 5 – Сравнительная таблица графических API: OpenGL, DirectX, Vulkan

|  |  |  |  |
| --- | --- | --- | --- |
| Особенности | OpenGL 3.3 | DirectX 11 | Vulkan 1.3 |
| Простота освоения | + | - | - |
| Кроссплатформенность | + | - | + |
| Поддержка аппаратных устройств | + | + | + |

Таблица 5 – Сравнительная таблица графических API: OpenGL, DirectX, Vulkan

|  |  |  |  |
| --- | --- | --- | --- |
| Особенности | OpenGL 3.3 | DirectX 11 | Vulkan 1.3 |
| Поддержка многопоточности | - | +- | + |
| Высокая скорость выполнения | - | + | + |
| Богатая экосистема | + | + | + |

Таким образом, выбор графического API DirectX 11 обусловлен ориентированностью на ОС Windows, удобством использования С++, а также своей производительностью.

Выбор интерфейса (UI), будет основан на визуальной красоте и на простоте имплементации в кодовую базу.

Рассмотрим 2 популярных графических API: Qt, ImGui.

**ImGui 1.77**

ImGui (Immediate Mode Graphical User Interface) - это легкая и простая в использовании библиотека для создания графического интерфейса пользователя (GUI) в приложениях. Вот некоторые плюсы и минусы использования ImGui.

**Плюсы ImGui:**

–Простота использования: ImGui предоставляет простой и интуитивно понятный API, который позволяет создавать GUI быстро и без особых сложностей. Она хорошо подходит для прототипирования и быстрого создания простых интерфейсов.

–Легковесность: ImGui является легковесной библиотекой, не зависящей от сложных зависимостей или сторонних библиотек. Это означает, что она может быть интегрирована в различные проекты без увеличения размера исполняемого файла или добавления избыточной сложности.

–Платформенная независимость: ImGui доступна на различных платформах и поддерживает множество графических API (например, OpenGL, DirectX). Это позволяет вам создавать кроссплатформенные приложения с GUI.

–Настраиваемость: ImGui предоставляет множество настраиваемых параметров и стилей, которые позволяют адаптировать внешний вид и поведение вашего GUI под ваши потребности. Вы можете настроить цвета, шрифты, расположение элементов и многое другое.

**Минусы ImGui:**

–Ограниченные возможности: ImGui не предоставляет такое множество готовых элементов интерфейса, как некоторые другие GUI-библиотеки. Если вам нужно сложное взаимодействие или специфический элемент интерфейса, вам придется реализовать его самостоятельно или использовать дополнительные библиотеки.

–Возможность возникновения сложностей при интеграции с некоторыми сторонними инструментами и библиотеками.

–Ограниченная кроссплатформенная совместимость: хотя ImGui поставляется с поддержкой для различных платформ и графических API, возможны некоторые различия в поведении и внешнем виде GUI на разных платформах. Это может потребовать некоторой доработки и настройки для достижения желаемого результата на разных платформах.

**Qt 5.2.0**

Qt — это кроссплатформенный фреймворк для разработки приложений, включая графический интерфейс пользователя (GUI). Вот некоторые плюсы и минусы использования

**Плюсы** Qt**:**

–Кроссплатформенность: Qt обеспечивает высокую кроссплатформенную совместимость, что означает, что вы можете разрабатывать приложения один раз и запускать их на различных операционных системах, таких как Windows, macOS, Linux, Android и других. Это сокращает время разработки и упрощает поддержку множества платформ.

–Богатые возможности: Qt предоставляет обширный набор инструментов и функций для создания сложных и мощных GUI-приложений. Он включает в себя готовые элементы интерфейса, такие как кнопки, таблицы, диалоговые окна и другие, а также поддержку графики, анимации, мультимедиа и многопоточности. –Удобство разработки благодаря автоматической настройке и конфигурированию, а также богатому функционалу, включая механизмы безопасности и управления транзакциями.

– Мощные инструменты разработки: Qt поставляется с интегрированной средой разработки (IDE) Qt Creator, которая предлагает удобную среду разработки с отладчиком, автодополнением кода, визуальным редактором интерфейса и другими инструментами. Она упрощает разработку и отладку приложений на основе Qt.

– Широкое сообщество и документация: Qt имеет активное сообщество разработчиков, где вы можете получить поддержку, задать вопросы и найти готовые решения для своих задач. Кроме того, существует обширная документация, учебные материалы и примеры кода, которые помогут вам быстро освоить фреймворк.

**Минусы Qt:**

–Объемный размер и сложность: Использование Qt может увеличить размер исполняемых файлов ваших приложений из-за необходимости включения библиотек Qt. Кроме того, Qt — это мощный фреймворк с обширными возможностями, и его изучение и освоение может потребовать времени и усилий.

–Лицензирование: Qt предоставляется под коммерческой лицензией, которая может требовать определенных затрат для коммерческого использования. Однако существует также версия с открытым исходным кодом (Qt Open Source), которая доступна бесплатно для некоммерческого использования.

– Сложность интеграции с существующим кодом: если у вас уже есть существующий код или проект, интеграция Qt может потребовать некоторой доработки и адаптации, особенно если ваш код был написан на другом фреймворке или с использованием других библиотек.

Таблица 6 – Сравнительная таблица фреймворков для создания UI: ImGui, Qt.

|  |  |  |
| --- | --- | --- |
| Особенности | ImGui **1.77** | Qt 5.2.0 |
| Простота использования и обучения | + | - |
| Кроссплатформенность | + | + |
| Богатая экосистема | + | + |
| Мощные инструмент | - | + |
| Простота встраивания в готовы проект | + | - |

Таким образом, выбор фреймворка ImGui, оправдан его простотой использования и приятным визуальным оформлением.

Рассмотрим несколько популярных сред для разработки под С++ и проведем анализ, чтобы выбрать наиболее подходящую среду для программы.

**Visual Studio 2022**

Microsoft Visual Studio 2022 — это среда разработки интегрированная современная среда, которая предназначена для разработки различных приложений, включая контрактные приложения, приложения Windows для настольных компьютеров, веб-приложения и мобильные приложения. Эта среда разработки, установленная на компьютере, позволяет разработчикам создавать, отлаживать и тестировать программы в единой среде. Ниже приведены плюсы и минусы использования среды разработки Visual Studio 2022.

**Плюсы:**

–Интеграция с технологиями разработки - Visual Studio 2022 интегрируется с различными технологиями разработки, такими как .NET Framework и Visual C. Это позволяет разработчикам создавать приложения на разных языках программирования.

–Простота разработки - Visual Studio 2022 предлагает обширный функционал интеллектуальных средств разработки (Intellisense), которые предсказывают продолжение кода путем подсказок и анализов. Они также проводят анализ кода для поиска потенциальных ошибок, что упрощает процесс разработки.

–Отладка — Это один из наиболее важных плюсов Visual Studio 2022. С этой средой разработки можно отлаживать приложения, находить ошибки, запускать тесты и устранять проблемы без изменения кода приложения.

–Большое количество опций в стандартной функциональности. В Visual Studio 2022 встроены стандартные функциональные возможности, такие как, например, Team Foundation Server и Subversion, что упрощает проектную работу.

–Техническая поддержка - Microsoft обеспечивает техническую поддержку своему продукту, что позволяет разработчикам обращаться за помощью в случае необходимости.

**Минусы:**

–Объемный размер Visual Studio 2022 - очень сложно установить эту IDE на малоемкий компьютер или устройство мобильной связи, так как объем данных этой программы достаточно большой.

–Не соответствующее стандартам использование ресурсов - Visual Studio 2022 потребляет значительное количество ресурсов процессора и оперативной памяти. Этот факт может вызывать проблемы у пользователей с менее мощными компьютерами и устройствами.

– Ограниченные возможности на платформе Mac OS - В мире Mac OS многие функции Visual Studio 2022 не работают, что делает его менее предпочтительным для разработчиков на этой платформе.

**Code::Blocks-20.03**

Code::Blocks — это бесплатная среда разработки, которая предоставляет множество функций и возможностей для создания приложений на разных языках программирования. Ниже перечислены плюсы и минусы Code::Blocks.

**Плюсы:**

–Открытый исходный код - Code::Blocks разрабатывается на основе открытого исходного кода, что позволяет пользователям модифицировать и улучшать его.

–Удобная навигация - Code::Blocks предоставляет удобную и понятную навигацию по проектам, файлам, функциям, переменным и т.д.

–Отладка - среда Code::Blocks предоставляет мощный отладчик, который позволяет искать ошибки и проблемы в коде.

–Поддержка плагинов - Code::Blocks можно дополнительно настроить по своему вкусу, используя большое количество плагинов и расширений, которые могут быть загружены пользователем.

–Интеграция с другими сервисами - Code::Blocks может использоваться вместе с другими инструментами, такими как GitHub, Jupyter Notebook и другие.

**Минусы:**

–Ограниченные возможности - по сравнению с некоторыми другими IDE, Code::Blocks может иметь менее обширный набор функций, чем у Visual Studio.

–Скорость работы - на медленных компьютерах Code::Blocks может работать медленно и иметь задержки.

–Проблемы с совместимостью - Code::Blocks может иметь проблемы с совместимостью с некоторыми операционными системами и требовать дополнительной настройки..

Таблица 7 – Сравнительная таблица сред разработки (IDE) – Visual Studio и Code::Blocks

|  |  |  |
| --- | --- | --- |
| Особенности | Visual Studio 2022 | Code::Blocks-20.03 |
| Поддержка DirectX SDK | + | + |
| Скорость работы | + | - |
| Количество фич | + | - |
| Интеграция с GitHub | + | + |
| Расширенная отладка | + | + |
| Отладка DirectX | + | - |

Исходя из полученных сравнительных данных, выбор Visual Studio 2022 оправдан тем, что IDE от Microsoft более развита, по сравнении с Code::Blocks, так же в Visual Studio имеет отладчик предназначенный для отладки DirectX, всего процесса отрисовки.

2 Цель и задачи курсового проекта

Целью курсового проекта является разработка программного комплекса для исследования поведения физических тел в гравитационном поле.

Программный комплекс будет включать:

–Задание объектов сцены через графический интерфейс (UI).

–Задание параметров сцены (физики), через графический интерфейс (UI).

–Просчёт физики объектов.

–Отрисовка объектов.

В соответствии с поставленной целью были сформулированы следующие задачи:

–Составить формализованное описание процесса тестирования студента как объекта обработки информации.

–Постановить задачу обработки и отрисовки физики объекта.

–Разработать функциональную структуру программного комплекса обработки и отрисовки.

–Разработать компоненты математического обеспечения программного комплекса.

–Разработать структуры интерфейсов для пользователя программного комплекса.

–Описать структур данных и алгоритмов.

–Описать структуры программы.

–Протестировать программный комплекс.

–Оформить документацию по проекту.

3 Технологическая часть

3.1 Формализованное описание процесса поведения физических тел в гравитационном поле

Для разработки данного проекта необходимо было иметь представление о моделируемом объекте, а именно о том, как представить процесс обработки физики в виде алгоритма, представляющего собой последовательность процессов, подпроцессов и условий.



Рисунок 5 – Формализованное описание процесса обработки физики

Пояснения к рисунку 5:

X = { iPoints } – вектор входных данных, где iPoints – объект точки( положение, скорость, ускорение точки).

U = { state } – вектор управляющего воздействия, где state – состояния физического движка.

Y = { oPoints } – вектор выходных данных, где oPoints – объект точки( положение, скорость, ускорение точки).

3.2 Постановка задачи отрисовки.

Заходя в приложение, пользователя встречает удобный и понятный графический интерфейс (UI), через который он может изменять, как и количество объектов, так и их свойства. Также через графический интерфейс (UI) возможно изменять состояния физического движка.

3.3 Разработка функциональной структуры программного комплекса для демонстрации поведения физических тел в поле притяжения.

Программный комплекс состоит из 3 модулей:  
– модуль сцены (через него идёт взаимодействие с объектами),

–модуль физический движок (через него идёт взаимодействие между объектами),

–модуль графического движка (объектов),

–модуль интерфейса (UI).



Рисунок 6 – Функциональная структура приложения автоматизированного тестирования.

3.4 Создание алгоритма взаимодействия объектов

Программный комплекс включает в себя модуль сцены, который хранит в себе информацию о всех точках (точка — это структура, хранящая в себе информацию о положении, скорости, ускорении), и состояния сцены. Модуль физический движок связан со сценой. Он получает информацию о всех точках. Далее он с помощью полного перебора проверяет коллизию между всеми объектами, и если функция коллизии вернула истину, то запускается функция столкновения, с последующим пересчётом скорости. В конце этого цикла идёт увеличение скорости в зависимости от ускорения и перемещение в зависимости от скорости.



Рисунок 7 – Схема вычисления перемещения точек

3.5 Разработка структуры интерфейса пользователя программного комплекса для демонстрации поведения физических тел в поле притяжения

В качестве пользовательского интерфейса будет выступать меню, в котором будет возможно изменять параметры сцены и её объектов.

Меню делится на несколько выпадающих списков:

* Настройки сцены. Данный пункт меню будет содержать поля для задания параметров объектов, а именно: их положения, скорости, ускорения, радиуса и состояний (статический/движимый, создающий притяжение/не создающий притяжение). Так же будут кнопки для быстрого создания точек и для удаления.
* Настройки физики. Данный пункт меню будет позволять изменять параметры физики для сцены, а именно: включение/отключение физики, притяжения, затухания движения и сопротивления воздуха. Так же будет доступ для множителей затухания движения, сопротивления воздуха, перемещения, ускорения и притяжения для возможности более тонкой настройки.

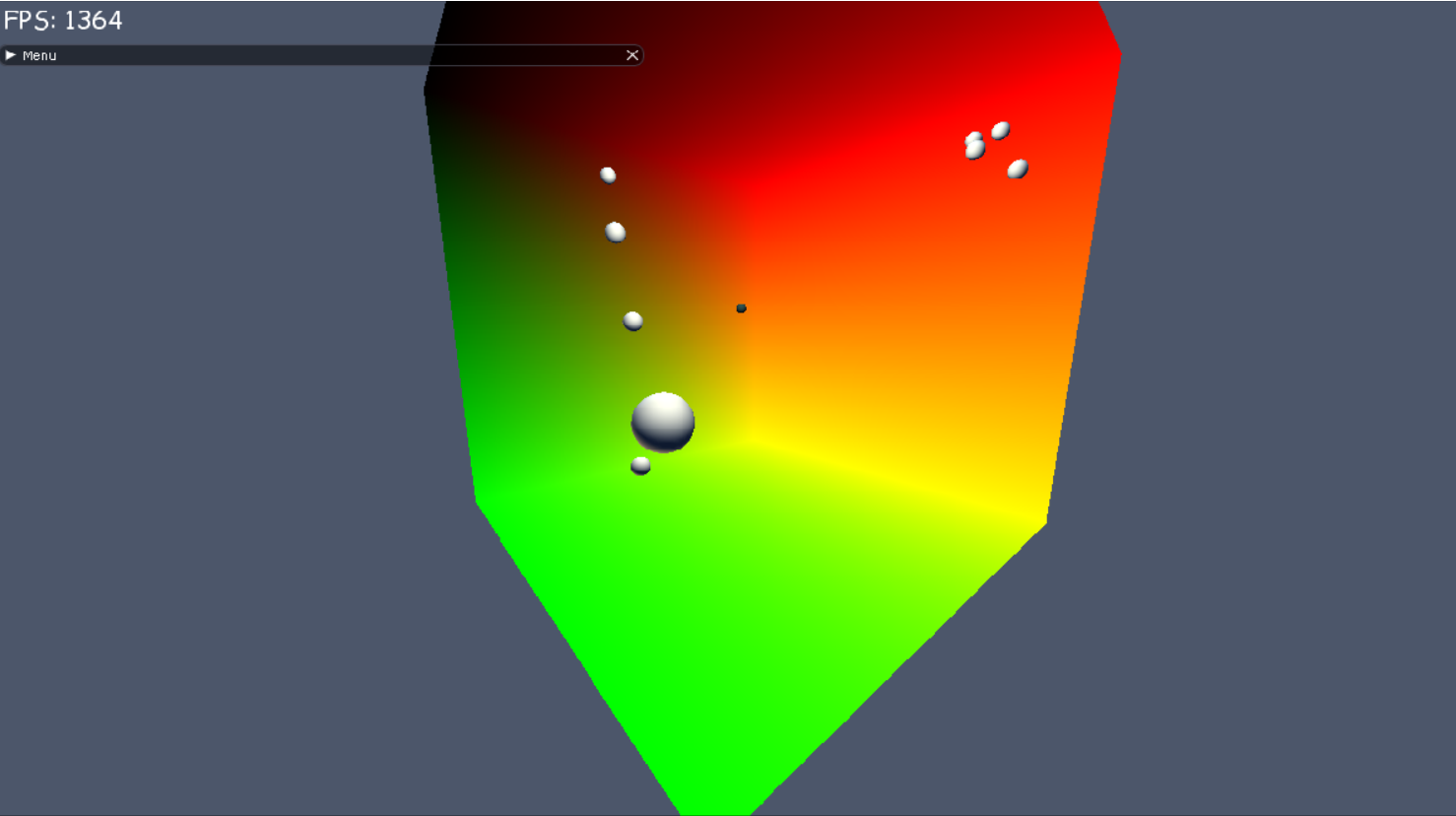


Рисунок 8 – Пользовательский интерфейс программы в закрытом состоянии

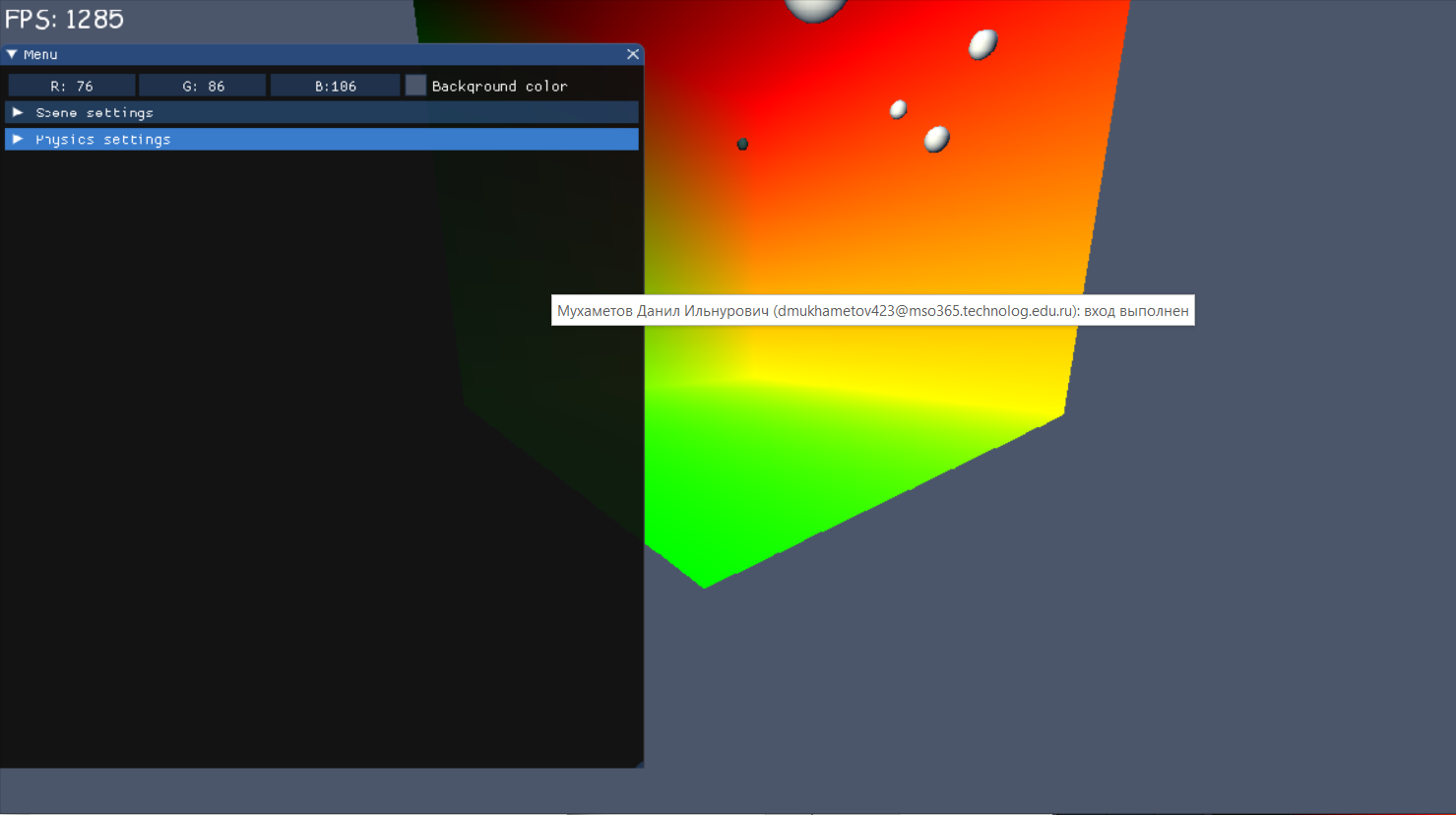


Рисунок 9 – Пользовательский интерфейс программы в полураскрытом состоянии.

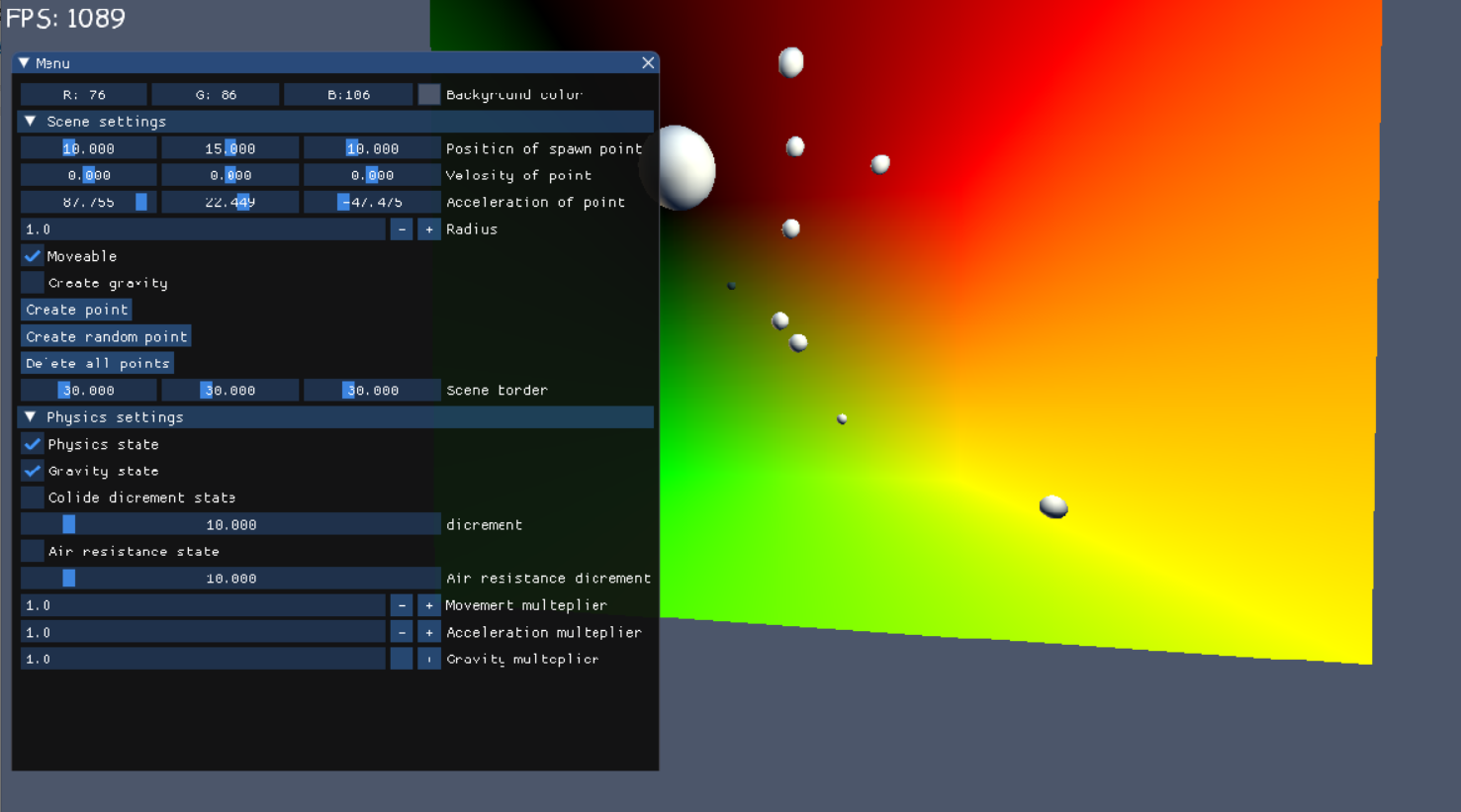


Рисунок 10 – Пользовательский интерфейс программы в раскрытом состоянии.

3.6 Описание структур данных и алгоритмов

Данная программа содержит несколько значимых структур, а именно структура сообщения от интерфейса для передачи всех состояний в модуль сцены, представлена на рисунке 11, и точка (хранит в себе информацию о положении, скорости, ускорения), представлена на рисунке 12.

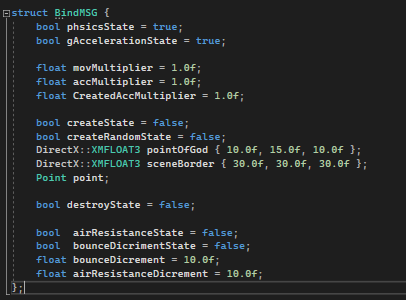


Рисунок 11 – Реализация структуры сообщения от интерфейса (UI).

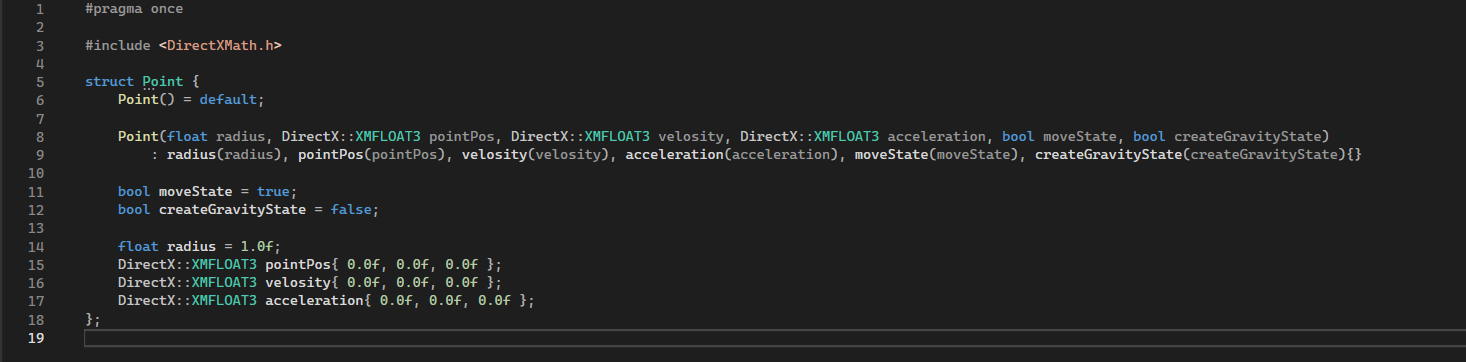


Рисунок 12 – Реализация структуры точки.

Алгоритмы представлены функциями:

* DistanceVector - вычисляет вектор расстояния между точками (вычитает координаты 1 из координат 2, из-за чего мы получаем вектор от 1 точки в направлении 2), представлена на рисунке 13.
* Distance – возвращает значение длину вектора (его модуль), представлена на рисунке 13.
* ObjectCollision – проверяет коллизию между объектами (если длина радиуса больше значения расстояния до другого объекта, то засчитывается столкновение), представлена на рисунке 14.
* BorderCollision - проверяет коллизию с границей сцены и возвращает сторону с которой произошло столкновение (граница сцены задана вектором. Если координаты объекта выходят за значение соответствующей координаты вектора границы, то выводится сторона коллизии также в соответствии), представлена на рисунке 14.
* Move – перемещает точку (к значениям позиции прибавляются значениям скорости по каждой координате), представлена на рисунке 15.
* Accelerate – ускоряет точку в зависимости от состояния сцены (к значениям скорости прибавляются значениям укорения по каждой координате), представлена на рисунке 16.
* PAccelerate – ускоряет точку по значениям ускорения точки, представлена на рисунке 16.
* GAccelerate – ускоряет точку по значению гравитационного ускорения, представлена на рисунке 16.

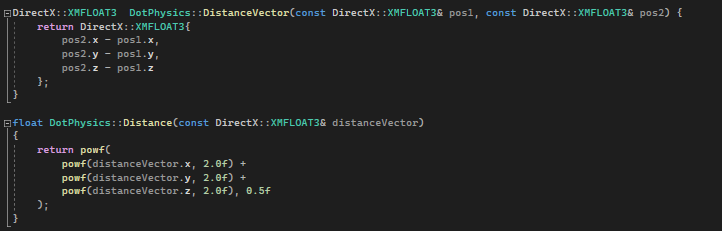


Рисунок 13 – Реализация функций DistanceVector и Distance.

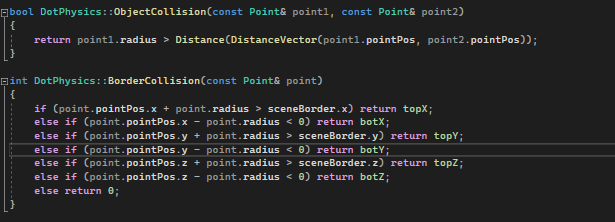


Рисунок 14 – Реализация функций ObjectCollision и BorderCollision.

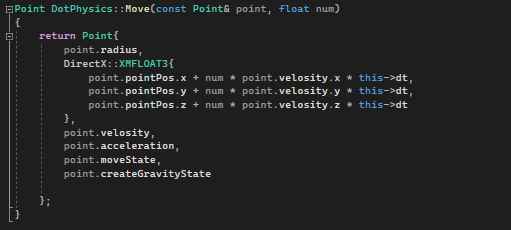


Рисунок 15 – Реализация функций Move.

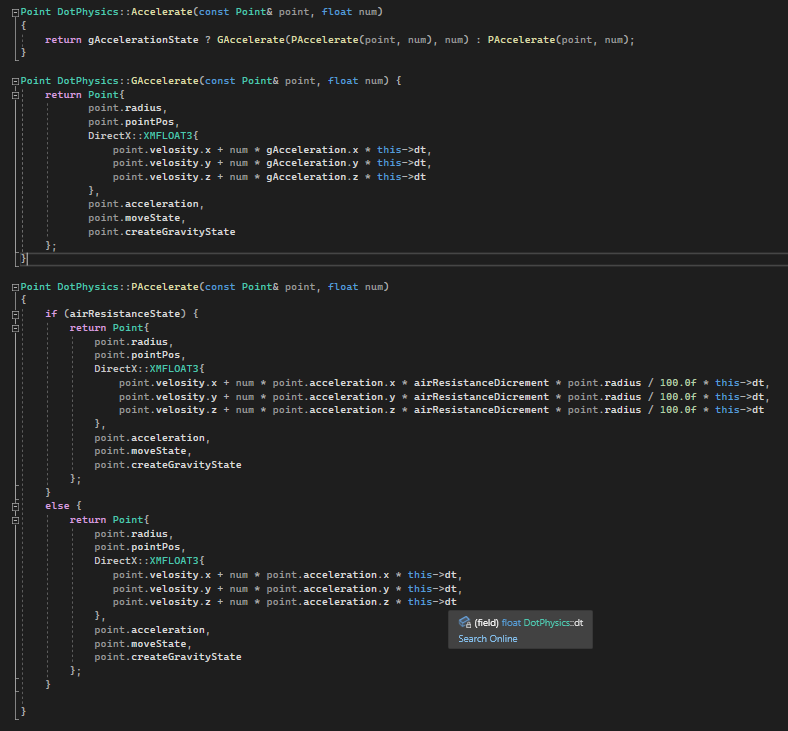


Рисунок 16 – Реализация функций PAccelerate, GAccelerate и Accelerate.

3.7 Описание структур данных и алгоритмов (формат представления данных в памяти и на внешних носителях)

В программе использовались целочисленные переменные, переменные с плавающей точкой, строковые и логические переменные, контейнеры STL, а также множество структур DirectX 11 и Win32.

Основные переменные, используемые в проекте, описаны в таблице 8.

Таблица 8 – Основные переменные программы

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Описание |
| pointOfGod | DirectX::XMFLOAT3 | положение точки создания (на её месте создаются новые объекты) |
| sceneBorder | DirectX::XMFLOAT3 | размер сцены |
| points | std::vector<Point> | Контейнер, хранящий все точки в сцене |
| phsicsState | bool | Флаг для включения/выключения физики |
| gAccelerationState | bool | Флаг для включения/выключения Земной гравитации |
| bounceDicrimentState | bool | Флаг для включения/выключения затухания движения |
| airResistanceState | bool | Флаг для включения/выключения сопротивления воздуха |
| movMultiplier | float | Множитель для перемещения точек |
| accMultiplier | float | Множитель для ускорения точек |
| CreatedAccMultiplier \_ | float | Множитель для создаваемого другими объектами ускорения |
| bounceDicrement | float | Значение, на которое уменьшается скорость затихает |
| airResistanceDicrement | float | Значение зависимости сопротивления воздуха от размера |
| createState | bool | Флаг для создания точки |
| createRandomState | bool | Флаг для создания точки со случайными параметрами |
| destroyState | bool | Флаг для уничтожения всех точек |

3.8 Тестирование программного комплекса (на добавление новых объектов, новых источников притяжения).

Программа была множественно проверена на работоспособность её компонентов (модулей). Все пункты меню выполняют свои функции и корректно модифицируют сцену, а именно создают точки с определёнными параметрами (положение, скорость, ускорение, статическая/движущаяся, создающая притяжение/не создающая притяжение), создаёт случайные точки, также удаление объектов сцены.

На рисунках 17 – 20 продемонстрирована работа программного комплекса для исследования поведения физических тел в гравитационном поле

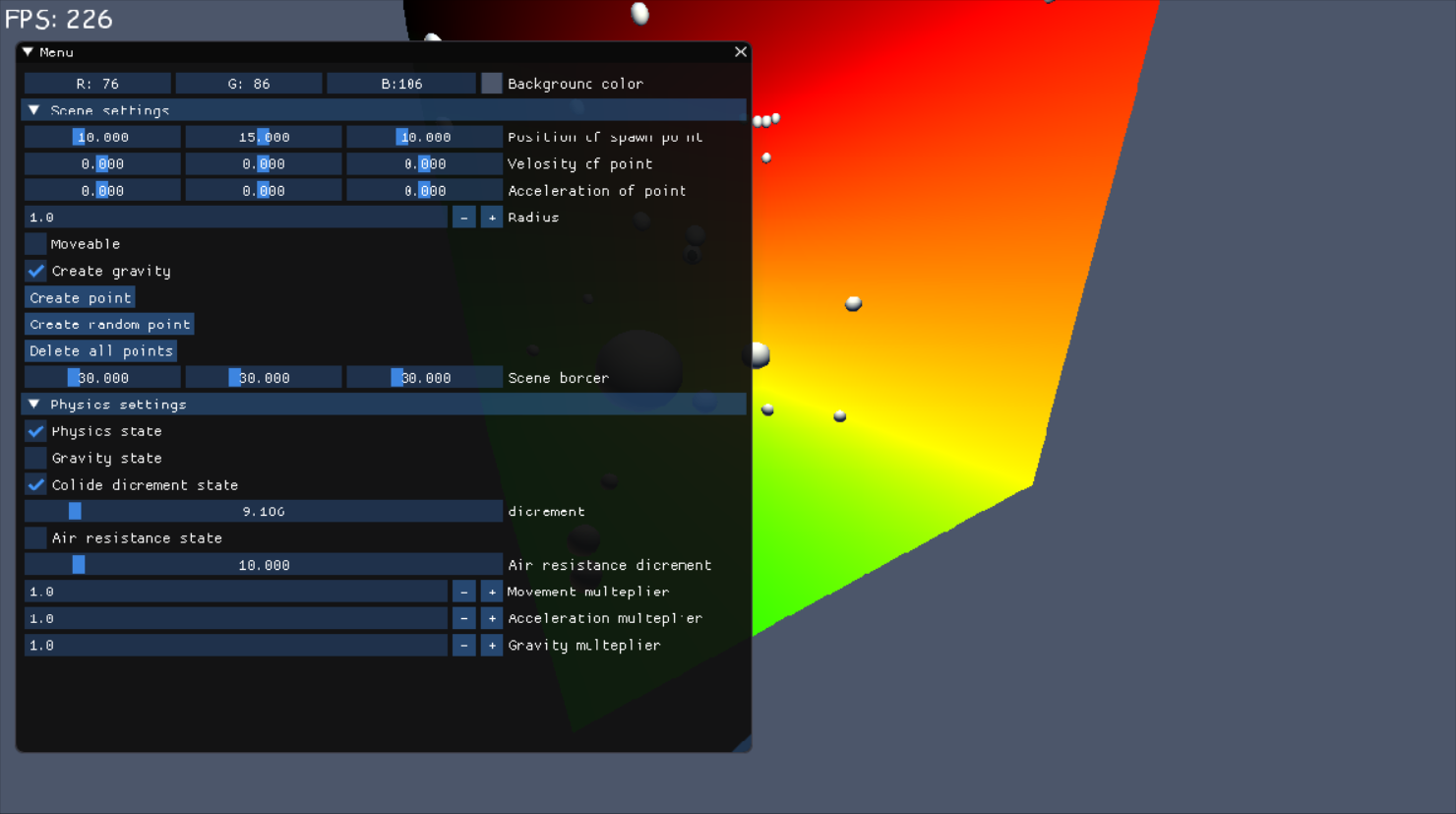


Рисунок 17 – Демонстрация взаимодействия пользователя с графическим интерфейсом.

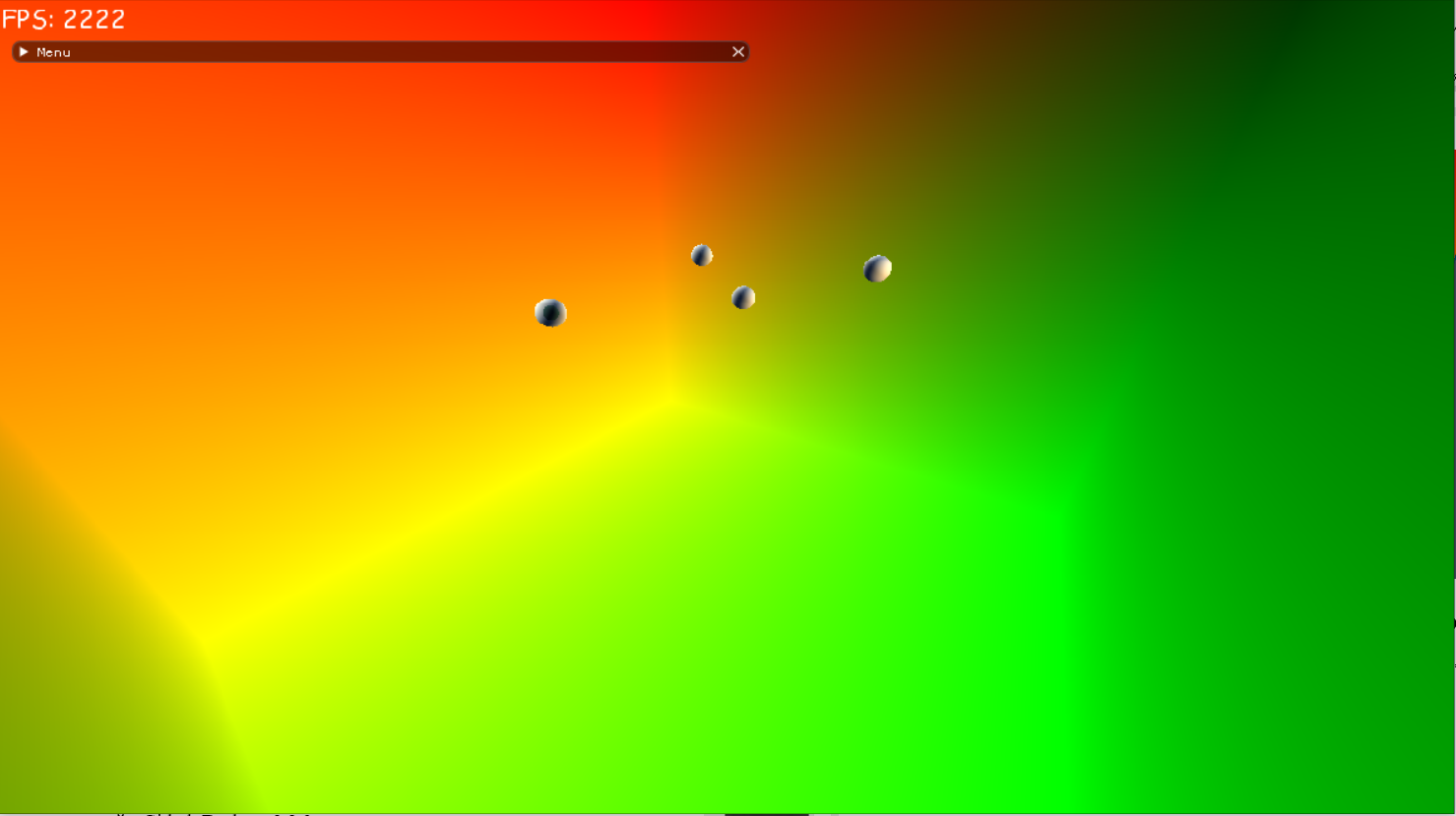


Рисунок 18 – Демонстрация сцены с 3 статическими телами, создающих притяжение для 4 тела в разный момент времени.

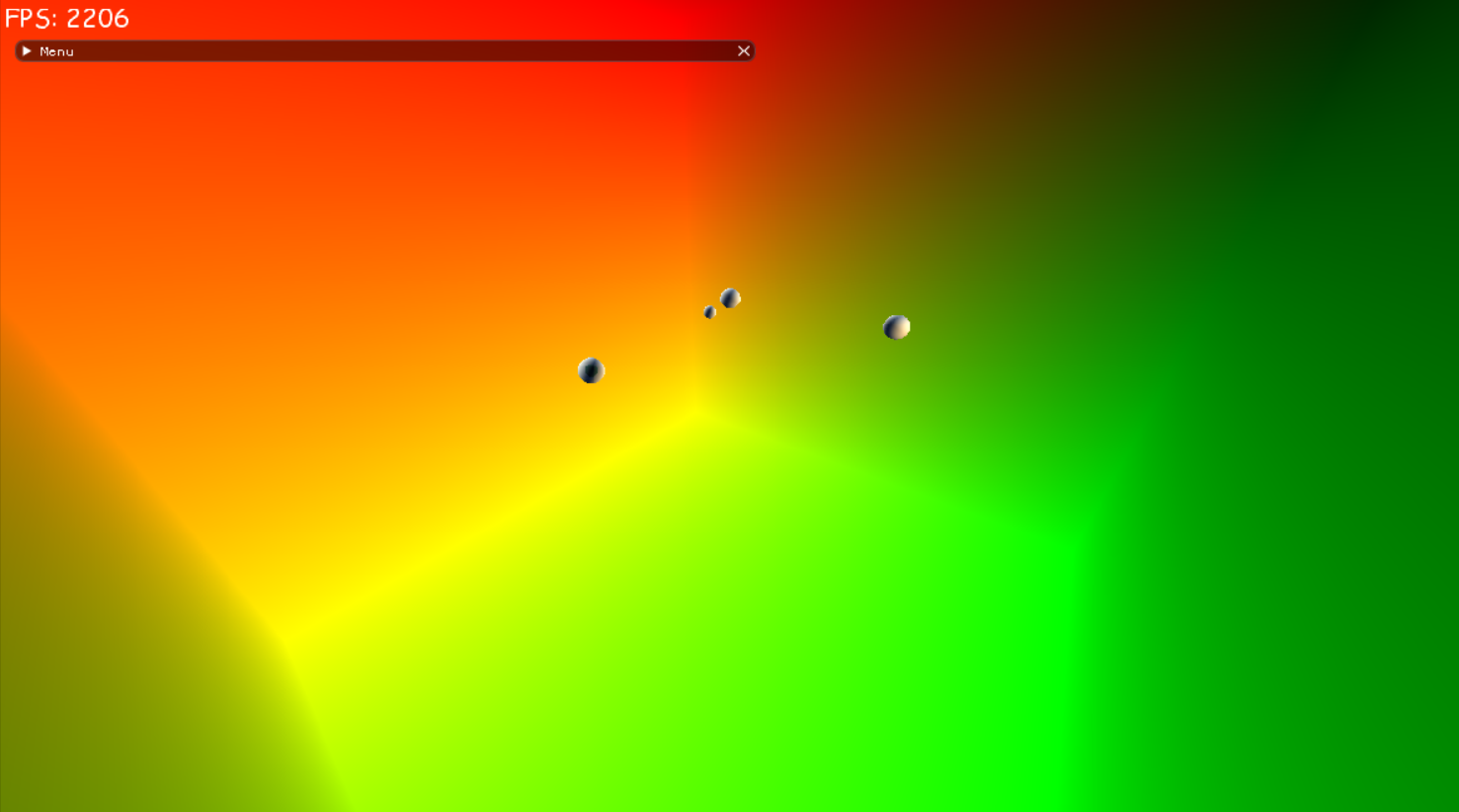


Рисунок 19 – Демонстрация сцены с 3 статическими телами, создающих притяжение для 4 тела в разный момент времени.

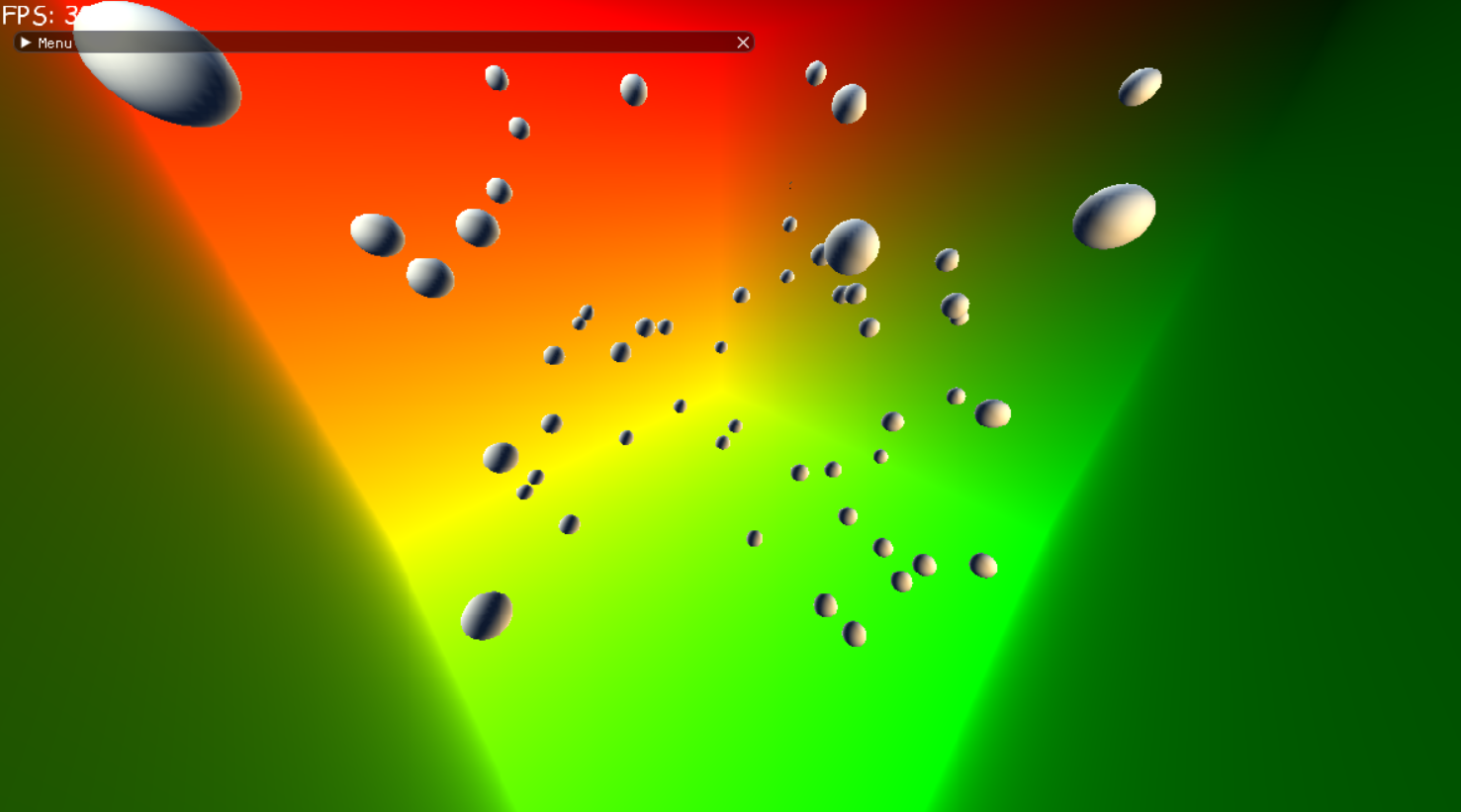


Рисунок 20 – Демонстрация стрессоустойчивости программного комплекса перед большим количеством объектов.

ХАРАКТЕРИСТИКА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ

Таблица 9 – Структура аппаратного обеспечения.

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Компьютер |
| Процессор (CPU) | Ryzen 5 5600H |
| Тактовая частота процессора, ГГц | 3,3 |
| Объем оперативной памяти, ГБ | 16 |
| Операционная система | Windows 10 21H2 |
| Графический ускоритель (GPU) | Radeon Vega 6 (1500 МГц) |

Таблица 10 – Структура программного обеспечения.

|  |  |
| --- | --- |
| Показатель | Значение |
| Тип ЭВМ | Компьютер |
| Среда разработки | Visual Studio 2022 |
| Офисный пакет | Microsoft Office 2021 |
| Система контроля версий | Github Desktop 3.2.0 |
| Графический интерфейс | ImGui 1.77 |
| Графическая API | DirectX 11 |

ВЫВОДЫ ПО КУРСОВОМУ ПРОЕКТУ

В ходе выполнения курсового проекта был разработан программный комплекс под операционную систему (ОС) Windows для исследования поведения физических тел в гравитационном поле, который позволяет наглядно демонстрировать эволюцию динамической системы тел. Данный программный комплекс охватывает такие темы как равноускоренное движение тел, так и общий случай ускоренного движения с возможностью полной настройки сцены. Для использования данным программным комплексом пользователь не обязан иметь какую-либо квалификацию, что позволяет охватить большую часть аудитории. Данный проект использует популярный графический API (DirectX 11), что позволяет наглядно демонстрировать результаты вычислений программный. Общую картину дополняет приятный и понятный графический интерфейс, который позволяет удобно вносить изменения в динамическую сцену: создавать, удалять объекты, а также изменять настройки физического движка программы, что кардинально меняет поведение объектов в сцене. Дальнейшие перспективы данного программного комплекса заключаются в введении системы ввода пользовательских моделей, что позволит симулировать еще больше физических задач, улучшении графической составляющей программы, а также улучшении в области оптимизации программы для дальнейшей возможности симуляции еще большего количества объектов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Frank, D. L. Introduction to 3D Game Programming with DirectX 11 / D. L. Frank. – United State : Mercury Learning and Information; Pap/DVD edition, February 28, 2012. – 600 с. – ISBN 978-1936420223.

2 Direct3D 11 на C++ с нуля : сайт. – URL: http://d3dbegin.narod.ru (дата обращения: 23.05.2023)

3 Jason, Z. Practical Rendering and Computation with Direct3D 11 1st / Z. Jason, P. Matt, H. Jack. – United State : A K Peters/CRC Press; 1st edition, July 27, 2011. – 648 с. – ISBN 978-1568817200.

4 DirectX 11 on Windows 10 Tutorials : сайт. – URL: https://www.rastertek.com/tutdx11win10.html (дата обращения: 23.05.2023)

5 Fletcher, D. 3D Math Primer for Graphics and Game Development 2nd / D. Fletcher, P. Ian. – United State : A K Peters/CRC Press; 2nd edition, November 2, 2011. – 846 с. – ISBN 978-1568817231.