

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Наталія АУШЕВА
“ ” _____ 2023р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему: Візуалізація деформації тривимірних тіл методами
полікоординатних перетворень

Виконав: студент 2 курсу, групи ТР-21 мп

ВОДОПШИН Володимир Володимирович

(прізвище, ім’я, по батькові)

_____ (підпис)

Науковий керівник к.т.н., доцент каф. ЦТЕ Юлія СИДОРЕНКО

(посада, вчене звання, науковий ступінь, ім’я ПРИЗВИЩЕ)

_____ (підпис)

Рецензент к.т.н., доцент каф. ТАЕ Артур РАЧИНСЬКИЙ

(посада, вчене звання, науковий ступінь, ім’я ПРИЗВИЩЕ)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ
ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти другий (магістерський)

За освітньою програмою “Цифрові технології в енергетиці”

Спеціальності 122 Комп’ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Наталія АУШЕВА
(підпис)

« _____ » _____ 2023 р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

ВОДОПШИНУ Володимиру Володимировичу

(прізвище, ім’я, по батькові)

1. Тема дисертації «Візуалізація деформації тривимірних тіл методами
полікоординатних перетворень»

Науковий керівник Сидоренко Юлія Всеволодівна, доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “06” листопада 2023 року №5152-с

2. Строк подання студентом дисертації 18 грудня 2023р

3. Вихідні дані до роботи: мова програмування Python, середа розробки PyCharm

4. Перелік питань, які потрібно розробити: аналіз методів візуалізації деформацій
тривимірних об’єктів, проектування алгоритму для полікоординатних перетворень,
створення користувацького інтерфейсу програми, вибір програмних бібліотек для
імплементції алгоритму, реалізація та тестування програмного забезпечення,
оцінка ефективності системи для кінцевого користувача

5. Орієнтований перелік ілюстративного матеріалу: постановка завдання,
математичний апарат програми, функціональна схема реалізації системи,
результати роботи програми

6. Орієнтований перелік публікацій: «Система деформаційного моделювання
тривимірних тіл на основі полікоординатних перетворень» VIII - а Міжнародна
науково-практична конференція «Сталий розвиток – XXI століття (наукові читання
імені Ігоря Недіна)»

7. Консультанти розділів дисертації _____

8. Дата видачі завдання «24» жовтня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Реалізація методу політочкових перетворень	01.09 – 08.09.2023	
2	Проектування архітектури програмних модулів	09.09 – 16.09.2023	
3	Розробка та реалізація розрахункового модуля	17.09 – 30.10.2023	
4	Створення графічного інтерфейсу користувача	01.10 – 08.10.2023	
5	Документація результатів дослідження	09.10 – 16.10.2023	
6	Внутрішнє тестування системи	17.10 – 22.10.2023	
7	Захист програмного продукту	23.10.2023	
8	Оформлення магістерської дисертації	24.10 – 03.12.2023	
9	Предзахист	04.12.2023	
10	Захист		

Студент

(підпис)

Володимир ВОДОПШИН

(ім'я ПРИЗВИЩЕ)

Науковий керівник

(підпис)

Юлія СИДОРЕНКО

(ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Структура і обсяг роботи. Магістерська дисертація складається зі вступу, п'яти розділів, висновку, переліку посилань з 42 найменувань, 1 додатку, і містить 26 рисунків, 15 таблиць. Повний обсяг магістерської дисертації складає 90 сторінок, з яких перелік посилань займає 3 сторінки, додатки – 6 сторінок.

Актуальність теми. В умовах стрімкого розвитку комп'ютеризації та інформаційних технологій тема деформаційного моделювання стає все важливішою в багатьох напрямках - від дизайну до машинобудування та медицини. Моделювання деформації тривимірних тіл методами полікоординатних перетворень є ключовим для візуалізації змін об'єктів у тривимірному просторі і потребує розробки відповідного програмного забезпечення.

Метою дослідження є створення програмної системи, яка дозволить візуалізувати деформацію тривимірних тіл в режимі реального часу з використанням методів полікоординатних перетворень.

Відповідно до поставленої мети були сформовані наступні **завдання дослідження:**

- проаналізувати існуючі програмні рішення;
- обрати методи та засоби програмної реалізації;
- розробити математичний апарат системи;
- створити структуру системи;
- розробити програмний продукт візуалізації деформації тривимірних тіл методами полікоординатних перетворень;

Об'єктом дослідження є комп'ютерна система деформаційного моделювання.

Предметом дослідження є система деформаційного моделювання за допомогою полікоординатних перетворень.

Методи дослідження. Розв'язання поставлених задач виконувались з використанням наступних методів:

- метод політочкових перетворень для побудови перетвореного тривимірного тіла;

— метод модульного програмування для розробки системи, що складається з різних, самостійних модулів;

Практичне значення одержаних результатів роботи полягає в розробці системи візуалізації деформацій, що надалі може бути використана дослідниками та інженерами у сфері деформаційного моделювання, а також у суміжних галузях для освіти, наукових досліджень та розробки нових продуктів.

Апробація результатів роботи. Основні положення роботи доповідались і обговорювались на :

- VIII Міжнародній науково-практичній конференції «Сталий розвиток – XXI століття (наукові читання імені Ігоря Недіна)» (м. Київ, 16-17 листопада 2023 року);

Наукові положення дипломної роботи опубліковані у 1 роботі.

Ключові слова: *деформаційне моделювання, тривимірна візуалізація, полікоординатні перетворення, програмне забезпечення, інтерактивна графіка, реальний час, куля, куб, шестикутна призма.*

ABSTRACT

Structure and extent of the work. The master's thesis consists of an introduction, five chapters, a conclusion, a list of 42 references, 1 appendix, and contains 26 figures, 15 tables. The total volume of the master's thesis is 90 pages, of which the list of references occupies 3 pages, appendices - 6 pages.

Relevance of the topic. In the context of rapid development of computerization and information technologies, the theme of deformation modeling is becoming increasingly important in many areas - from design to mechanical engineering and medicine. Modeling the deformation of three-dimensional bodies using poly-coordinate transformations is key for visualizing changes in objects in three-dimensional space and requires the development of corresponding software.

The goal of the research is to create a software system that will allow for the visualization of the deformation of three-dimensional bodies in real-time using methods of poly-coordinate transformations.

In accordance with the set goal, the following **research tasks** were formulated:

- analyze existing software solutions;
- select methods and tools for software implementation;
- develop the mathematical apparatus of the system;
- create the structure of the system;
- develop a software product for visualizing the deformation of three-dimensional bodies using poly-coordinate transformations;

The object of the study is a computer system for deformation modeling.

The subject of the study is a system for deformation modeling using poly-coordinate transformations.

Research methods. The tasks were solved using the following methods:

- the method of poly-point transformations for constructing a transformed three-dimensional body;
- the modular programming method for developing a system consisting of various modules;

The practical significance of the obtained results lies in the development of a system for visualizing deformations, which can be used by researchers and engineers in the field of deformation modeling, as well as in related fields for education, scientific research, and new product development.

Approval of the research results. The main points of the work were reported and discussed at:

- VIII International Scientific and Practical Conference "Sustainable Development – XXI Century (scientific readings in honor of Igor Nedin)" (Kyiv, November 16-17, 2023);

Scientific positions of the diploma work are published in 1 paper.

Keywords: *deformation modeling, three-dimensional visualization, poly-coordinate transformations, software, interactive graphics, real-time, sphere, cube, hexagonal prism.*

ЗМІСТ

ВСТУП.....	10
1 ЗАДАЧА ВІЗУАЛІЗАЦІЇ ДЕФОРМАЦІЇ ТРИВИМІРНИХ ТІЛ МЕТОДАМИ ПОЛІКООРДИНАТНИХ ПЕРЕТВОРЕНЬ	12
Висновки до розділу 1	14
2 МАТЕМАТИЧНИЙ АППАРАТ СИСТЕМИ.....	15
2.1 Перетворення площин полікоординатним методом	15
2.2 Перетворення гранованих тіл полікоординатним методом	24
Висновки до розділу 2	26
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ	27
3.1 Загальна структура системи.....	27
3.2 Графічний модуль	29
3.3 Модуль розрахунків перетворень	31
3.4 Інтерфейсний модуль	34
3.5 Вибір інструментарію та середовища розробки для програмного продукту	36
3.5.1 Критерії та раціональний вибір мови програмування для проекту	37
3.5.2 NumPy, як інструмент наукових обчислень у Python.....	38
3.5.3 PyGame - бібліотека для розробки графічного інтерфейсу	39
3.5.4 Інтегроване середовище розробки PyCharm	40
Висновки до розділу 3	41
4 РОБОТА КОРИСТУВАЧА З РОЗРОБЛЕННОЮ СИСТЕМОЮ	43
4.1 Системні вимоги до комп'ютера.....	44
4.2 Встановлення програми на комп'ютер.....	45
4.3 Меню вибору початкового тіла для побудови з подальшим налаштуванням	48
4.4 Перетворення та деформації тіла діями користувача	51
4.4.1 Обертання фігури навколо своєї осі.....	53
4.4.2 Деформація фігури методом полікоординатних перетворень.....	55
4.4.3. Зміна параметру деталізації при побудові фігури	58
4.4.4 Масштабування моделі.....	61
4.5 Завершення роботи з програмою	62
Висновки до розділу 4	64
5 РОЗРОБКА СТАРТАП ПРОЕКТУ	65
5.1 Основні положення стартап-проекту	65
5.2 Огляд існуючих програмних рішень	66

5.3 Вибір технічних інструментів для втілення проектної ідеї.....	69
5.4 Прогнозування етапів реалізації стартап-проекту	70
5.5 Аналіз потенційних клієнтів.....	72
5.6 Аналіз факторів загроз та можливостей зі сторони ринку	74
5.7 Розроблення стратегії ринкового впровадження проекту	76
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	82
ДОДАТОК А.....	85

ВСТУП

У світлі стрімкого прогресу технологій та їх широкого впровадження в найрізноманітніші сфери людської діяльності, науковий інтерес до методів комп'ютерного моделювання стає дедалі актуальнішим. Цінність тривимірної графіки для аналітичних, навчальних та інженерних досліджень важко переоцінити, адже вона відчиняє двері до віртуальної реальності, що надає можливість детального розгляду об'єктів з будь-якого боку та у будь-яких умовах. Інтерактивність та висока реалістичність візуальної інформації усувають бар'єри між реальним та уявним світами, водночас надаючи сильний імпульс для розвитку наукових досліджень та індустріальних розробок.

Сфера моделювання деформаційних процесів тривимірних тіл вимагає глибокого та багатогранного підходу з огляду на комплексність задач, що виникає при дослідженні фізичних властивостей та поведінки тіл під час їх трансформації. Візуалізація таких процесів несе в собі велике значення для підвищення точності інженерних розрахунків, кращого розуміння біомеханічних процесів, створення віртуальних прототипів у дизайні та навіть при плануванні хірургічних втручань.

Науковий інтерес до цієї теми пов'язаний із потребою розробки нових, більш ефективних методів візуалізації деформацій, які могли б забезпечити більшу точність та гнучкість у моделюванні. Полікоординатні перетворення як метод моделювання, що дозволяє деталізувати та змінювати форму тіл в тривимірному просторі, має особливу цінність у цьому аспекті.

Незважаючи на значний прогрес у цій області, існуючі методи моделювання часто зазнають впливу обмежень, обумовлених внутрішніми властивостями програмних продуктів, які використовуються для їхнього відтворення. Саме тому дослідження альтернативних підходів, здатних розширити можливості практичного застосування візуалізацій, становить значний науковий інтерес.

Метою даної магістерської роботи визначено розробку та аналіз моделі візуалізації деформації тривимірних тіл з використанням полікоординатних перетворень, яка б дозволила узагальнити існуючі методики та надала б нові можливості для їх практичного застосування.

Для досягнення цієї мети необхідно розв'язати такі завдання:

- провести аналіз сучасного стану методів візуалізації деформацій;
- визначити головні принципи полікоординатних перетворень у контексті візуалізації деформацій;
- розробити алгоритмічну базу для комп'ютеризованих методів полікоординатних перетворень;
- оптимізувати та валідувати створену модель шляхом експериментів та тестувань;
- розробити рекомендації щодо інтеграції та використання моделі в практичній діяльності.

Для подальшого прогресу в області комп'ютерного моделювання важливим є розвиток і удосконалення методів деформаційного моделювання. Поглиблене вивчення полікоординатних перетворень і їх застосування до тривимірних тіл може відкрити нові перспективи для створення більш точних і гнучких моделей. Це стане основою для покращення розрахунків і алгоритмів, які мають застосування у широкій сфері наукових і інженерних досліджень.

Крім розробки нових алгоритмів і підходів, існує потреба у створенні комплексних комп'ютеризованих систем, що дозволяють проводити більш швидкі та точні розрахунки деформацій, забезпечуючи можливість моделювання в режимі реального часу. Такі системи могли б значно підвищити ефективність конструкційних, аеродинамічних, біомеханічних та інших досліджень, що вимагають точної візуалізації і моделювання складних динамічних процесів.

1 ЗАДАЧА ВІЗУАЛІЗАЦІЇ ДЕФОРМАЦІЇ ТРИВИМІРНИХ ТІЛ МЕТОДАМИ ПОЛІКООРДИНАТНИХ ПЕРЕТВОРЕНЬ

У вік швидкоплинних технологічних інновацій надзвичайно зросла потреба забезпечити візуальні технічні рішення, спроможні демонструвати складні фізичні процеси, як-от деформація тривимірних тіл. Актуальність створення досконалих методів візуалізації деформацій із залученням новітніх досягнень у цифровій обробці даних і аналітичному моделюванні визначається їх потенціалом у провідних промислових галузях, від авіабудування до медичної інженерії.

Підготовка надійної та зрозумілої візуалізації деформаційних процесів тривимірних об'єктів викликає посилену увагу до комплексного аналітичного підходу, що забезпечить безперервний процес від абстрактної концепції до достовірного відображення реальних фізичних явищ. В контексті магістерського дослідження центральними є завдання розробки та вдосконалення алгоритмічних підходів, інструментальних засобів та методик, які можуть бути адаптовані до високоспецифічних візуальних завдань.

Процес роботи поділяється на кілька стратегічних напрямків. Насамперед, головним завданням є розробка математичного апарату, що включає формулювання математичних моделей, здатних описувати самі деформації у всіх їхніх аспектах. Ця робота вимагає використання математичного аналізу, диференціальної геометрії та комп'ютерних наук, щоб на складне інтердисциплінарне завдання знайти обґрунтоване і точне рішення.

Також ключовим є завдання вибору і оптимізації алгоритмів візуалізації, що дозволить посилити точність відображення деформацій. Для цього потрібно притримуватися базових принципів графічного рендерингу та програмування шейдерів, а також працювати з сучасними графічними бібліотеками і двигунами.

Наступна задача — розробка і впровадження користувацького інтерфейсу. Важливість цього аспекту важко переоцінити, оскільки зручність та інтуїтивність користувацького досвіду безпосередньо впливають на ефективність роботи з системою. Належне проектування користувацького інтерфейсу вимагає розуміння основ UX (User Experience) та UI (User Interface) дизайну.

Не на останньому місці стоїть задача забезпечення гнучкості системи щодо модифікації параметрів моделювання у реальному часі. Це означає, що користувач повинен мати можливість не тільки запускати прогони сценаріїв деформацій, але і безпосередньо взаємодіяти з візуальною моделлю - змінюючи її параметри, що відіграють роль у деформаційному процесі. Впровадження інтерактивних систем і налаштування, яке змінюється динамічно в залежності від дії користувача, призведе до підвищення інформативності візуалізації та сприятиме глибшому розумінню причинно-наслідкових деформаційних зв'язків.

Важливим є забезпечення можливості реагувати на нестандартні умови чи непередбачувані стани системи. Програмне забезпечення повинно мати вбудовані захисти від некоректного вводу даних користувачем та засоби діагностики для точного виявлення та корекції помилок. Робота з наслідками некоректного запуску сценаріїв або помилок у програмному коді має бути продуманою та безболісною для кінцевого користувача.

Останнє, але не менш важливе завдання — це розробка документації, яка матиме наглядний і зрозумілий вигляд та забезпечить користувачу всю необхідну інформацію для використання візуалізаційної системи. Документація має включати як технічний опис компонентів системи, так і практичні керівництва, для швидкого оволодіння функціоналом.

Комплексний підхід до вирішення зазначених завдань вкаже шлях до розвитку надійного, зручного в користуванні програмного засобу для візуалізації деформацій, який враховує всі аспекти від математичної обробки даних до реального використання в індустрії чи освіті. Ретельна підготовка до кожного з цих завдань та продумане вирішення будуть служити запорукою розв'язання основної мети дослідження — створення і втілення в життя чіткої, наукоємної, але водночас із юзер-френдлі інтерфейсом компільованої системи візуалізації тривимірних деформацій, здійснюваної за допомогою методів полікоординатних перетворень.

В кінцевому вигляді розроблена система повинна стати ефективним інструментом для інженерних та наукових кіл, що займаються моделюванням і аналізом фізичних процесів. Для досягнення цих цілей потрібно провести детальне тестування, яке встановить точність, що може забезпечити реалізована система,

визначити допустимі межі її застосування, а також забезпечити засади для майбутніх технічних та функціональних покращень системи на основі результативності її практичного застосування.

Ідеалом залишається створення такої програмної системи, що не буде мати рамок у своєму використанні: від базових сценаріїв до складних промислових досліджень. Цьому сприятиме зосередження на модульності дизайну, що дасть можливість користувачам персоналізувати і систематично розширювати її функціонал.

Висновки до розділу 1

У цьому розділі було визначено ключові вимоги та задачі для створення системи візуалізації деформацій тривимірних тіл, поєднуючи математичні методи з алгоритмами комп'ютерного моделювання. Основною метою дослідження стало розроблення ефективних інструментів для реалізації такої системи, здатної надавати глибокий аналіз і зрозуміле відображення деформаційних процесів. Було вирішено досягнути високої гнучкості системи, завдяки інтеграції користувацького інтерфейсу, що дозволить користувачу взаємодіяти з тривимірною моделлю в реальному часі. Програма має бути розроблена таким чином, щоб забезпечити наочність, доступність та розуміння процесів деформації.

Підсумки дослідження надають підґрунтя для подальшого розвитку програмної системи, адаптації її до складних інженерних та наукових задач з використанням як систематичних так і динамічних методів моделювання деформацій. Уточнені цілі та підходи дозволяють створити міцну основу для оптимізації та розширення можливостей системи візуалізації в майбутньому.

2 МАТЕМАТИЧНИЙ АППАРАТ СИСТЕМИ

Візуалізація деформації тривимірних тіл з використанням концепції полікоординатних перетворень потребує ретельного розуміння принципів і механізмів цих перетворень у контексті тривимірної геометрії. Полікоординатні перетворення у тривимірному просторі дозволяють нам не лише маніпулювати базовими геометричними формами, але й моделювати більш складні деформації, що відображають реальні фізичні процеси.

Концепція полікоординатних перетворень базується на ідеї взаємодії тривимірних просторових об'єктів з множиною управляючих точок. У двовимірному просторі управління формою об'єктів відбувається через маніпулювання точками на плоскій поверхні. Проте в тривимірному світі контрольні точки розташовані не лише на поверхні, а й заглиблені в просторі навколо об'єкта, формуючи об'єм. Тому коли йде мова про перетворення об'єктів у 3D, мається на увазі вплив на них за допомогою переміщення точок усередині цього об'єму.

Аналізуючи тривимірні деформації, ми виходимо з просторових об'єктів, що можуть бути представлені за допомогою тривимірної сітки координат. Подібно до того як у двовимірному випадку прямі лінії можуть бути згодом деформовані, у тривимірному випадку ми оперуємо з поверхнями та їх зламами.

2.1 Перетворення площин полікоординатним методом

Розглянемо методологію та інструментарій, які дозволяють здійснювати полікоординатні трансформації об'єктів, існуючих на площині, та екстраполюємо їх на багатовимірний, зокрема, тривимірний простір. Коли ми говоримо про полікоординатні трансформації лінії на площинному рівні, то розуміємо, що фундаментом таких трансформацій є сукупність точок самої площини, і, у свою чергу, об'єктами для трансформування служать одна або декілька ліній. Це необхідне для того, щоб виконати процес деформації предмета, який первісно визначається визначеними наборами ліній [15,19,22]. Отже, при переході до

полікоординатних трансформацій у тривимірному вимірі, геометричною основою для трансформацій служить весь набір точок тривимірного простору, в той час як об'єктами трансформації стають фігури площини або навіть повний набір площин, коли мова йде про тривимірні об'єкти, які можуть бути представлені через їхній набір площин [14,18,26].

Розглянемо особливості розрахункових відношень, здатних виконувати полікоординатні трансформування конкретної площини у тривимірному просторі. Залучаючи до уваги принцип дуалізму у формульному представленні, відносно однорідного простору. Однорідний простір визначаємо через введення невизначеного множника, що служить як четверта координата:

$$X_{\text{однорідний}} = WX$$

$$Y_{\text{однорідний}} = WY$$

$$Z_{\text{однорідний}} = WZ$$

$$h_{\text{однорідний}} = W$$

У вигляді однорідного простору, рівняння площини перетворюється як наступне:

$$aX_{\text{однорідний}} + bY_{\text{однорідний}} + cZ_{\text{однорідний}} + dh_{\text{однорідний}} = 0$$

Це рівняння є симетричним щодо координат $X_{\text{однорідний}}$, $Y_{\text{однорідний}}$, $Z_{\text{однорідний}}$ та $h_{\text{однорідний}}$, а також коефіцієнтів a , b , c , d . Відповідно, виходить так, що фігура площини характеризується «координатами» a , b , c , d , тоді як будь-яка точка в просторі отримує значення $X_{\text{однорідний}}$, $Y_{\text{однорідний}}$, $Z_{\text{однорідний}}$, $h_{\text{однорідний}}$.

В області тривимірних перетворень, полікоординатні трансформації будуть визначені як трансформації площин, за умов, що за основу трансформацій береться набір точок, які можуть змінюватись відповідно до вимог, які формуються згідно вирішення завдань деформації [20,21].

Візуально підкреслимо задачу полікоординатних трансформацій у тривимірному просторі.

На рисунку 2.1 ми бачимо каркас початкових точок 1, 2, 3, 4, 5, позначених за допомогою декартових координат x_i , y_i та z_i . Визначаємо початкову площину (тобто площину, яка підлягає трансформації), а також обчислюємо відстань від

точок початкового базису до цієї площини. Далі рівняння отримуємо у нормалізованому вигляді і, після підстановки координат базових точок у рівняння, отримуємо величини γ_i , які є полікоординатними координатами площини.

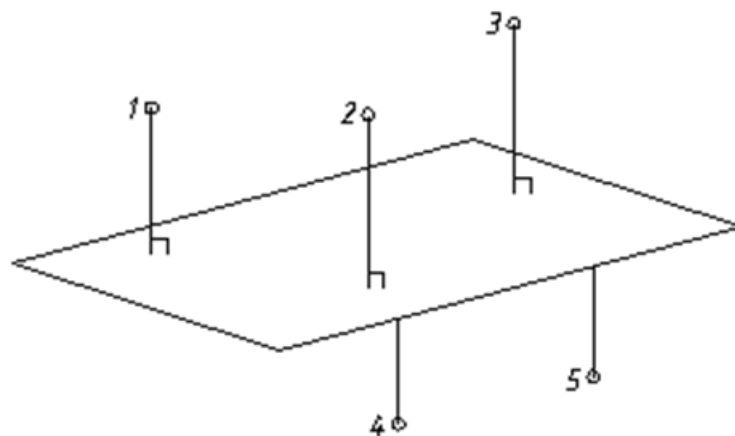


Рисунок 2.1 - Площина у початковому базисі

Модифікуємо каркас: отже, ми змінюємо декартові координати точок первинного базису, що зображено на рисунку 2.2. Нові точки 1', 2', 3', 4', 5' формують новий базис, де нам необхідно визначити розташування нової (трансформованої) площини.

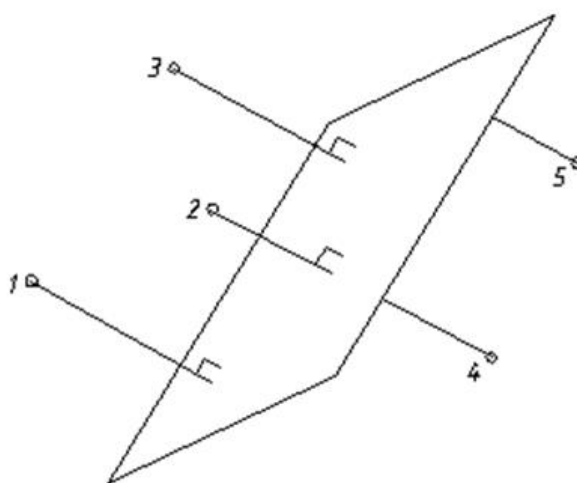


Рисунок 2.2 - Перетворена площина після зміни базису

Більш глибоко аналізуючи полікоординатні трансформації в умовах багатоточкового каркасу, ми визначаємо образ-площість на початковому базисі набором рівнянь:

$$\gamma_i = ax_i + by_i + cz_i + dh_i, \quad i = 1 \dots n, \quad (2.1)$$

де a, b, c, d – коефіцієнти опису площини,

x_i, y_i, z_i, h_i – координати точок початкового базису.

Тепер необхідно визначити нове положення площини у оновленому базисі, Трансформована площина визначена системою рівнянь:

$$\phi_i = AX_i + BY_i + CZ_i + DH_i, \quad i = 1 \dots n,$$

де A, B, C, D – шукані коефіцієнти нової площини,

X_i, Y_i, Z_i, H_i – координати точок у новому базисі,

ϕ_i – відстані від точок початкового базису до образу-площини.

Коли ми крок за кроком аналізуємо конкретну площину у контексті множини точок первісного каркасу, її полікоординати перед та після трансформації, як правило, будуть різнитись, тобто $\gamma_i \neq \phi_i$, де $i = 1, 2, \dots, p$. В тому випадку, коли ми захочемо звести площину у трансформованому базисі, виходячи із умови $\gamma_i = \phi_i$, для $i = 1, 2, \dots, p$, ми виявимо, на превеликий жаль, не площину, а деякий об'ємний простір [1, 4, 13]. Внаслідок цього, стоїть завдання отримати чітку і недвозначну відповідь на запитання полікоординатного трансформування простору R^3 , тобто встановити функціональний зв'язок в базисах між координатами γ_i та ϕ_i , де $i = 1, 2, \dots, p$.

Аналогічно до площинних полікоординатних трансформацій, і з урахуванням того, що координати площини у полікоординатному базисі корелюють із її відстанню від базових точок, полікоординатне перетворення у тривимірному просторі можна представити формулою:

$$\phi_i = \omega_i \gamma_i, \quad i = 1, 2, \dots, n.$$

Підставивши ϕ_i у рівняння (2.1), ми отримаємо:

$$\omega_i \gamma_i = Ax_i + By_i + Cz_i + Dh_i, \quad i = 1, 2, \dots, n.$$

Ця система рівнянь містить n рівнянь з $n+4$ невідомими (ω_i , де $i = 1, 2, \dots, n$, та A, B, C, D). Тому для однозначного рішення цієї системи потрібно встановити чотири додаткові умови.

Щоб визначити ці додаткові умови, розглядаємо знаходження ω_i , для $i = 1, 2, \dots, n$, при мінімальному відхиленні від одиниці, таким чином ϕ_i буде мінімізувати своє відхилення від γ_i [15,16]. Отже, пропонується звести до мінімуму функціонал, який матиме наступний вигляд:

$$\sum (\phi_i - \gamma_i)^2 \rightarrow 0, \quad i = 1 \dots n.$$

Задача зводиться до пошуку цього функціоналу екстремуму, для чого необхідно розрахувати часткові похідні за змінними A, B, C , а четверте рівняння системи буде покликано підтримати нормалізований вигляд трансформованої площини[7]. З цим ми отримуємо нелінійну систему рівнянь, яку належить розв'язати.

$$\begin{cases} Ax_1 + By_1 + Cz_1 + D = \beta_1, \\ Ax_2 + By_2 + Cz_2 + D = \beta_2, \\ Ax_3 + By_3 + Cz_3 + D = \beta_3, \\ A^2 + B^2 + C^2 = 1. \end{cases}$$

$$\begin{cases} A(x_1 - x_2) + B(y_1 - y_2) + C(z_1 - z_2) = \beta_1 - \beta_2, \\ A(x_2 - x_3) + B(y_2 - y_3) + C(z_2 - z_3) = \beta_2 - \beta_3, \\ A^2 + B^2 + C^2 = 1. \end{cases}$$

$$\begin{cases} A \frac{(x_1 - x_2)}{(z_1 - z_2)} + B \frac{(y_1 - y_2)}{(z_1 - z_2)} + C = \frac{\beta_1 - \beta_2}{z_1 - z_2}, \\ A \frac{(x_2 - x_3)}{(z_2 - z_3)} + B \frac{(y_2 - y_3)}{(z_2 - z_3)} + C = \frac{\beta_2 - \beta_3}{z_2 - z_3}, \\ A^2 + B^2 + C^2 = 1. \end{cases}$$

$$\begin{cases} A \left(\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3} \right) + B \left(\frac{y_1 - y_2}{z_1 - z_2} - \frac{y_2 - y_3}{z_2 - z_3} \right) = \frac{\beta_1 - \beta_2}{z_1 - z_2} - \frac{\beta_2 - \beta_3}{z_2 - z_3}, \\ A^2 + B^2 + C^2 = 1. \end{cases}$$

$$\begin{cases} A \left(\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3} \right) + B \left(\frac{y_1 - y_2}{z_1 - z_2} - \frac{y_2 - y_3}{z_2 - z_3} \right) = \frac{\beta_1 - \beta_2}{z_1 - z_2} - \frac{\beta_2 - \beta_3}{z_2 - z_3}, \\ A^2 + B^2 + \left(\frac{\beta_2 - \beta_3}{z_2 - z_3} - A \frac{x_2 - x_3}{z_2 - z_3} - B \frac{y_2 - y_3}{z_2 - z_3} \right)^2 = 1. \end{cases}$$

Введемо позначення:

$$t_1 = \frac{x_1 - x_2}{z_1 - z_2}; \quad t_2 = \frac{x_2 - x_3}{z_2 - z_3}; \quad t_3 = \frac{y_1 - y_2}{z_1 - z_2};$$

$$t_4 = \frac{y_2 - y_3}{z_2 - z_3}; \quad t_5 = \frac{\beta_1 - \beta_2}{z_1 - z_2}; \quad t_6 = \frac{\beta_2 - \beta_3}{z_2 - z_3}.$$

$$\begin{cases} A(t_1 - t_2) + B(t_3 - t_4) = t_5 - t_6, \\ A^2 + B^2 + (t_6 - At_2 - Bt_4)^2 = 1. \end{cases}$$

$$\begin{cases} A(t_1 - t_2) + B(t_3 - t_4) = t_5 - t_6, \\ A^2 + B^2 + t_6^2 - 2t_6(At_2 + Bt_4) + (At_2 + Bt_4)^2 = 1. \end{cases}$$

$$\begin{cases} A(t_1 - t_2) + B(t_3 - t_4) = t_5 - t_6, \\ A^2 + B^2 + t_6^2 - 2At_2t_6 - 2Bt_4t_6 + A^2t_2^2 + 2ABt_2t_4 + B^2t_4^2 = 1. \end{cases}$$

$$\begin{cases} A(t_1 - t_2) + B(t_3 - t_4) = t_5 - t_6, \\ A^2(1 + t_2^2) + 2ABt_2t_4 - 2At_2t_6 - 2Bt_4t_6 + B^2(1 + t_4^2) - 1t_6^2 = 0. \end{cases}$$

Розглянемо перше рівняння:

$$A = \frac{t_5 - t_6 - B(t_3 - t_4)}{t_1 - t_2}.$$

$$A = \frac{t_5 - t_6}{t_1 - t_2} - B \frac{t_3 - t_4}{t_1 - t_2}.$$

$$\left(\frac{t_5 - t_6}{t_1 - t_2} - B \frac{t_3 - t_4}{t_1 - t_2} \right) (1 + t_2^2) + 2 \left(\frac{t_5 - t_6}{t_1 - t_2} - B \frac{t_3 - t_4}{t_1 - t_2} \right) (Bt_2t_4 - t_2t_6) -$$

$$-2Bt_4t_6 + B^2(1 + t_4^2) - 1 + t_2^6 = 0.$$

Введемо позначення:

$$a_{-1} = \frac{t_{-5} - t_{-6}}{t_{-1} - t_{-2}}; \quad a_{-2} = \frac{t_{-3} - t_{-4}}{t_{-1} - t_{-2}}; \quad a_{-3} = 1 + t_{-2}^2; \quad a_4 = t_{-2}t_{-4};$$

$$a_5 = t_2t_6; \quad a_6 = t_4t_6; \quad a_7 = 1 + t_4^2; \quad a_8 = -1 + t_6^2.$$

$$(a_1 - Ba_2)^2a_3 + 2(a_1 - Ba_2)(Ba_4 - a_5) - 2Ba_6 + B^2a_7 + a_8 = 0.$$

$$a_3a_1^2 - 2Ba_1a_2a_3 + B^2a_2^2a_3 + 2(Ba_1a_4 - B^2a_2a_4 - a_1a_5 + Ba_2a_5) - \\ - 2Ba_6 + B^2a_7 + a_8 = 0.$$

$$a_3a_1^2 - 2Ba_1a_2a_3 + B^2a_2^2a_3 + 2Ba_1a_4 - 2B^2a_2a_4 - 2a_1a_5 + 2Ba_2a_5 - \\ - 2Ba_6 + B^2a_7 + a_8 = 0.$$

$$B^2(a_2^2a_3 - 2a_2a_4 + a_7) + B(-2a_1a_2a_3 + 2a_1a_4 + 2a_2a_5 - 2a_6) + \\ + a_3a_1^2 = 2a_1a_5 + a_8 = 0.$$

Введемо позначення.

$$K_1 = a_2^2a_3 - 2a_2a_4 + a_7; \quad K_2 = -2a_1a_2a_3 + 2a_1a_4 + 2a_2a_5 - 2a_6;$$

$$K_3 = a_3a_1^2 - 2a_1a_5 + a_8.$$

Отримаємо квадратне рівняння.

$$B_2 = \frac{-K_2 + \sqrt{K_2^2 - 4K_1K_3}}{2K_1}. \quad B^2K_1 + BK_2 + K_3 = 0.$$

$$D = K_2^2 - 4K_1K_3. \\ B_1 = \frac{-K_2 - \sqrt{K_2^2 - 4K_1K_3}}{2K_1}.$$

$$\text{де } K_1 = a_2^2a_3 - 2a_2a_4 + a_7 =$$

$$\begin{aligned}
&= \left(\frac{t_3 - t_4}{t_1 - t_2} \right)^2 (1 + t_2^2) - 2 \left(\frac{t_3 - t_4}{t_1 - t_2} \right) t_2 t_4 + 1 + t_4^2 = \\
&\quad \left(\frac{t_3 - t_4}{t_1 - t_2} \right) \left[\frac{(t_3 - t_4)(1 + t_2^2)}{(t_1 - t_2)} - 2t_2 t_4 \right] + 1 + t_4^2 = \\
&= \frac{(t_3 - t_4)}{t_1 - t_2} \cdot \frac{((t_3 - t_4)(1 + t_2^2) - 2(t_1 - t_2)t_2 t_4)}{t_1 - t_2} + 1 + t_4^2 = \\
&= \frac{\frac{y_1 - y_2}{z_1 - z_2} - \frac{y_2 - y_3}{z_2 - z_3}}{\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3}} \cdot \left[\frac{\left(\frac{y_1 - y_2}{z_1 - z_2} - \frac{y_2 - y_3}{z_2 - z_3} \right) \left(1 + \left(\frac{x_2 - x_3}{z_2 - z_3} \right)^2 \right) -}{\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3}} \right. \\
&\quad \left. - 2 \left(\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3} \right) \cdot \frac{x_2 - x_3}{z_2 - z_3} \cdot \frac{y_2 - y_3}{z_2 - z_3} \right] + 1 + \left(\frac{y_2 - y_3}{z_2 - z_3} \right)^2.
\end{aligned}$$

$$\begin{aligned}
K_2 &= -2a_1 a_2 a_3 + 2a_1 a_4 + 2a_2 a_5 - 2a_6 = \\
&= -2 \frac{t_5 - t_6}{t_1 - t_2} \cdot \frac{t_3 - t_4}{t_1 - t_2} \cdot (1 + t_2^2) + 2 \frac{t_5 - t_6}{t_1 - t_2} t_2 t_4 + \\
&\quad + 2 \frac{t_3 - t_4}{t_1 - t_2} t_2 t_6 - 2t_4 t_6 = \\
&= 2 \left[\frac{t_5 - t_6}{t_1 - t_2} \left(\frac{(t_3 - t_4)(1 + t_2^2)}{t_1 - t_2} + t_2 t_4 \right) + t_6 \left(\frac{(t_3 - t_4)t_2}{t_1 - t_2} - t_4 \right) \right] = \\
&= 2 \left[\frac{t_5 - t_6}{t_1 - t_2} \left(\frac{(t_3 - t_4)(1 + t_2^2) + (t_1 - t_2)t_2 t_4}{t_1 - t_2} \right) + \right. \\
&\quad \left. + t_6 \left(\frac{(t_3 - t_4)t_2 - t_4(t_1 - t_2)}{t_1 - t_2} \right) \right] = \\
&= \frac{2}{t_1 - t_2} \left[\frac{t_5 - t_6}{t_1 - t_2} \left((t_3 - t_4)(1 + t_2^2)(t_1 - t_2)t_2 t_4 \right) + \right.
\end{aligned}$$

$$\begin{aligned}
& +t_6((t_3 - t_4)t_2 - t_4(t_1 - t_2))\Big] = \\
& = \frac{2}{\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3}} \left[\frac{\frac{\beta_1 - \beta_2}{z_1 - z_2} - \frac{\beta_2 - \beta_3}{z_2 - z_3}}{\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3}} \left(\left(\frac{y_1 - y_2}{z_1 - z_2} - \frac{y_2 - y_3}{z_2 - z_3} \right) \times \right. \right. \\
& \times \left(1 + \left(\frac{x_2 - x_3}{z_2 - z_3} \right)^2 \right) + \left(\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3} \right) \frac{x_2 - x_3}{z_2 - z_3} \cdot \frac{y_2 - y_3}{z_2 - z_3} \Big) \\
& + \frac{\beta_2 - \beta_3}{z_2 - z_3} \left(\left(\frac{y_1 - y_2}{z_1 - z_2} - \frac{y_2 - y_3}{z_2 - z_3} \right) \cdot \frac{x_2 - x_3}{z_2 - z_3} - \right. \\
& \left. \left. - \frac{y_2 - y_3}{z_2 - z_3} \left(\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3} \right) \right) \right].
\end{aligned}$$

$$\begin{aligned}
K_3 &= a_3 a_1^2 - 2a_1 a_5 + a_8 = \\
& (1 + t_2^2) \cdot \left(\frac{t_5 - t_6}{t_1 - t_2} \right)^2 - 2 \frac{t_5 - t_6}{t_1 - t_2} t_2 t_6 + t_6^2 - 1 = \\
& = \left(1 + \left(\frac{x_2 - x_3}{z_2 - z_3} \right)^2 \right) \cdot \left(\frac{\frac{\beta_1 - \beta_2}{z_1 - z_2} - \frac{\beta_2 - \beta_3}{z_2 - z_3}}{\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3}} \right)^2 - \\
& - 2 \left(\frac{\frac{\beta_1 - \beta_2}{z_1 - z_2} - \frac{\beta_2 - \beta_3}{z_2 - z_3}}{\frac{x_1 - x_2}{z_1 - z_2} - \frac{x_2 - x_3}{z_2 - z_3}} \right) \cdot \frac{x_2 - x_3}{z_2 - z_3} \cdot \frac{\beta_2 - \beta_3}{z_2 - z_3} + \left(\frac{\beta_2 - \beta_3}{z_2 - z_3} \right)^2 - 1.
\end{aligned}$$

Коефіцієнт В служить ключем до виявлення інших коефіцієнтів у рівнянні: А, С і D, проте процедура їх обчислення виявляється складною і обтяжливою, тому пропонується взяти до розгляду варіативний метод [8]. Для того, щоб продовжити, розрахуємо часткові диференціали по змінних А, В, С та D.

$$\frac{\partial \sum (\phi_i - \gamma_i)^2}{\partial A} = 2 \sum X_i (A X_i + B Y_i + C Z + D H_i - \gamma_i) = 0.$$

$$\frac{\partial \sum (\phi_i - \gamma_i)^2}{\partial B} = 2 \sum Y_i (AX_i + BY_i + CZ_i + DH_i - \gamma_i) = 0.$$

$$\frac{\partial \sum (\phi_i - \gamma_i)^2}{\partial C} = 2 \sum Z_i (AX_i + BY_i + CZ_i + DH_i - \gamma_i) = 0.$$

$$\frac{\partial \sum (\phi_i - \gamma_i)^2}{\partial D} = 2 \sum H_i (AX_i + BY_i + CZ_i + DH_i - \gamma_i) = 0.$$

Для лінійної системи рівнянь, в матричному вигляді буде представлено:

$$A \cdot V = B,$$

де

$$A = \begin{bmatrix} \sum X_i^2 & \sum X_i Y_i & \sum X_i Z_i & \sum X_i H_i \\ \sum Y_i X_i & \sum Y_i^2 & \sum Y_i Z_i & \sum Y_i H_i \\ \sum Z_i X_i & \sum Z_i Y_i & \sum Z_i^2 & \sum Z_i H_i \\ \sum H_i X_i & \sum H_i Y_i & \sum H_i Z_i & \sum H_i^2 \end{bmatrix}.$$

$$V = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}.$$

$$B = \begin{bmatrix} \sum X_i \gamma_i \\ \sum Y_i \gamma_i \\ \sum Z_i \gamma_i \\ \sum H_i \gamma_i \end{bmatrix}.$$

Отже, вирішенням даної системи лінійних рівнянь стануть значення змінних А, В, С та D, які, в свою чергу, представляють собою шукані параметри трансформованої площини.

Таким чином, полікоординатне перетворення площин відкриває шлях для подальшого моделювання об'єктів в тривимірному просторі. Завдяки цьому, тривимірні об'єкти, котрі можуть бути розділені на сукупності площин, можна трансформувати шляхом застосування полікоординатних перетворень до кожної з цих площин окремо, що дозволить нам отримати кінцевий перетворений тривимірний об'єкт.

2.2 Перетворення гранованих тіл полікоординатним методом

У процесі виконання полікоординатних трансформацій, де об'єкт визначений у тривимірному просторі, він представляється як сукупність граней, які мають форму

трикутників. Одну із можливостей представлення кожної з таких трикутних граней визначається за допомогою чотирьох площин, що перетинаються у вершинах трикутника, так як показано на рисунку 2.3.

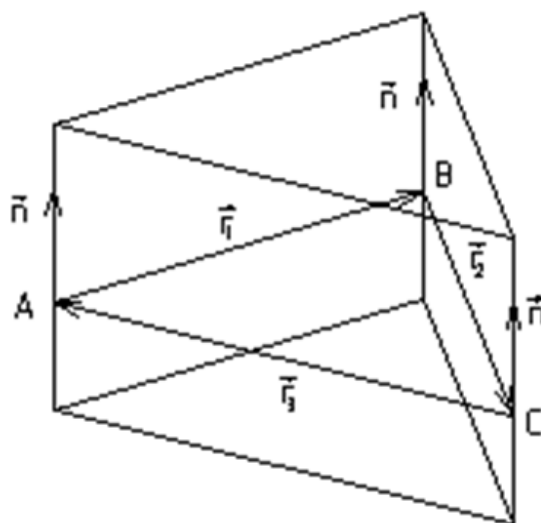


Рисунок 2.3 - Візуалізація трикутника як елемента трансформації

Для перетворення кожної площини, що складають грані об'єкта, застосовується полікоординатний метод. Унаслідок цього, в новій системі координат визначаються нові розташування граней, які ось і визначають положення граней вже трансформованого об'єкта. Поєднуючи трансформовані трикутники, ми формуємо фасетний об'єкт - результат трансформації заданої моделі [9]-[12].

Процедуру трансформації тривимірної моделі можна описати в декілька етапів [26,29]:

1. Початкова підготовка базису, у який "занурюється" тривимірний об'єкт. Відбір таких базисних точок вимагає врахування специфічних вимог до кінцевого аспекту трансформованого об'єкта та здійснюється вручну спеціалістом.
2. Репрезентація об'єкта, який підлягає трансформації, через сукупність трикутних площин, що може бути досягнуто, зокрема, шляхом тріангуляції моделі.
3. Послідовне застосування полікоординатних трансформацій до кожної з трикутних площин. Це дозволяє визначити перетини та отримати репрезентацію граней об'єкта у новому базисі.

4. Звернення до вже відомих методів та технік просторової інтерполяції або можливостей сучасного програмного забезпечення для графічної обробки, аби виконати оптимальне згладжування поверхонь отриманої фасетної моделі.

В результаті виконання такої процедури, ми одержуємо перетворений об'єкт, який демонструє зміни відповідно до заданих раніше умов. Весь алгоритм характеризується високою ефективністю особливо при рішенні задач деформування моделей у трьох вимірах, а також завдяки своїй структурі вимагає мінімально можливого числа операцій для досягнення точних результатів.

Висновки до розділу 2

В результаті аналізу властивостей і механізмів полікоординатних перетворень у тривимірному просторі, було встановлено, що ця концепція є потужним інструментом для візуалізації і моделювання деформацій об'єктів реального світу. Подано докладний опис процедур необхідних для перетворення об'єктів в тривимірному просторі, що включає в себе представлення об'єктів через тривимірні сітки координат і здійснення деформації через маніпуляцію площинами.

Особлива увага була приділена властивостям і алгоритмам розрахунку полікоординатних перетворень площин і гранованих тіл. Описано методологію перетворення площин, яка передбачає використання однорідних координат і мінімізацію функціоналу для визначення кращого рішення в перетвореному базисі.

Дослідження показали, що для успішного застосування полікоординатних трансформацій, важливо детально підготовлювати початкові умови та враховувати аспекти точної геометрії тривимірного об'єкта.

У підсумку, полікоординатні перетворення пропонують строго науковий та практично застосовний підхід до задач деформації та візуалізації тривимірних об'єктів, можуть бути реалізовані з високою ступенем точності і відкривають нові можливості для більш складних симуляцій в сучасній графіці та інженерії.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Загальна структура системи

В рамках даного дослідження було проведено ретельний аналіз специфіки предметної області, з особливим акцентом на дослідження та розуміння складних механізмів полікоординатних перетворень. Результатом цієї роботи стало створення технологічно вдосконаленої програмної системи, яка об'єднує в собі як теоретичні знання, так і практичні інструменти для моделювання деформацій тривимірних тіл. Система дозволяє реалізовувати складні деформації з високою точністю, надаючи користувачам потужні можливості для аналізу та візуалізації фізичних процесів у різних областях, від промислового дизайну до розважальних медіа.

Важливою частиною створення системи була її модульна архітектура, яка забезпечує ефективність та надійність виконання складних обчислювальних завдань. Система складається з трьох основних компонентів: модуля візуалізації, модуля обчислень та користувацького інтерфейсу, кожен з яких був розроблений як окремий виконуваний файл. Цей підхід не тільки спрощує процедуру інтеграції програмного продукту в різні технічні середовища, але й забезпечує високий рівень гнучкості та контролювання робочих процесів з боку користувачів.

З метою детального розуміння внутрішньої логіки роботи системи та взаємодії між її окремими модулями, було створено візуальну схему, яка представлена на рисунку 3.1. Ця схема детально відображає роль кожного модуля, їх поділ за функціями, а також показує, яким чином відбувається передача даних та команд між ними. Такий підхід у візуалізації архітектури системи відіграє ключову роль у забезпеченні її зрозумілості та ефективності на етапі проектування та подальшої експлуатації.

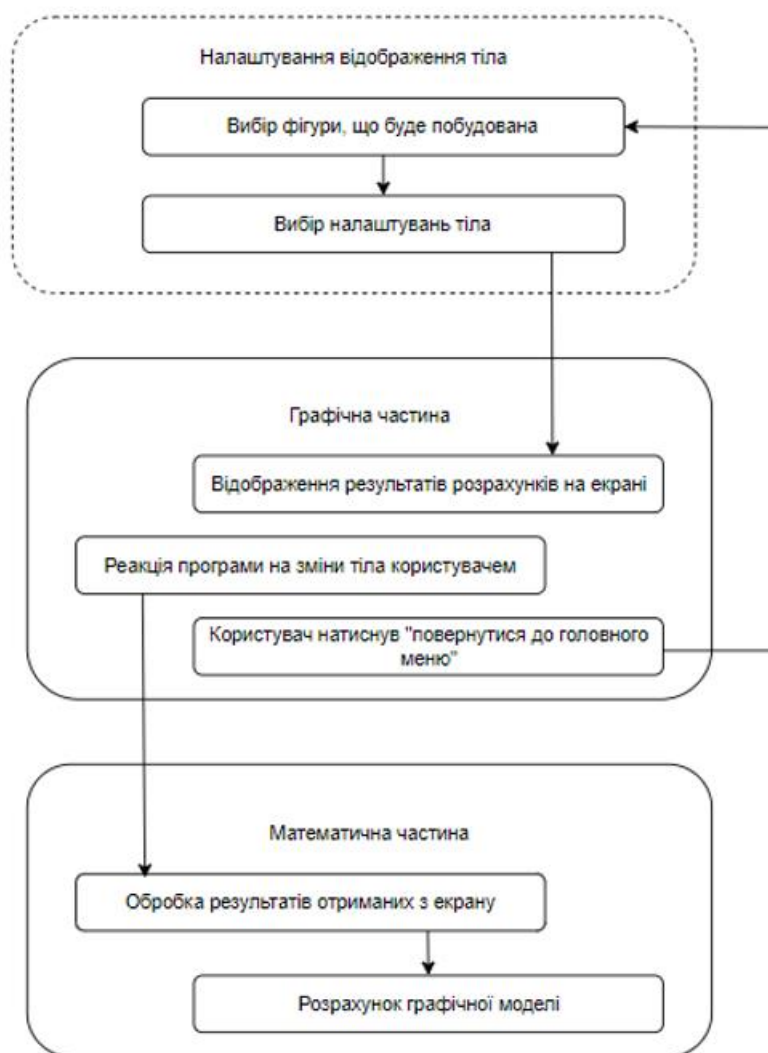


Рисунок 3.1 – Структура розробленої системи

З рисунку 3.1 видно, що робота кожного з модулів налаштована таким чином, що вона розпочинається лише тоді, коли завершується функціонування попереднього модуля. Це забезпечує плавність та послідовність робочого процесу, виключаючи можливість виникнення помилок або конфліктів під час обробки даних. Після завершення процесу побудови і деформації об'єкта, система не переходить у режим очікування, а зберігає активний стан, знову знаходячись в готовності до взаємодії з користувачем. Це означає, що користувач може без затримок подавати нові дані для обробки або, у разі необхідності, може скористатися можливістю повернення до головного меню системи. У головному меню користувач має змогу вибрати інші об'єкти для подальшого моделювання, встановити параметри їх деформації і запустити весь процес заново. Весь цикл

роботи з системою може бути повторений необмежену кількість разів, забезпечуючи таким чином високу продуктивність та ефективність використання програмного забезпечення для досягнення поставлених цілей.

3.2 Графічний модуль

Графічний модуль системи виконує критично важливе завдання, перетворюючи абстрактні математичні концепції та числові розрахунки в конкретні візуальні зображення, які мають велике значення для користувачів. Цей модуль не просто відображає результати роботи розрахункового модуля, але і створює візуальний інтерфейс, що дозволяє користувачам наочно бачити зміни, внесені в параметри тривимірної моделі.

Графічний модуль забезпечує плавну та ефективну обробку даних, отриманих з розрахункового модуля, і відіграє вирішальну роль у забезпеченні точного та чіткого представлення об'єктів. Блок-схема графічного модуля, яка зображена на рисунку 3.2, демонструє складність взаємодії між різними компонентами модуля і як вони працюють разом, щоб перетворити оновлені координати точок у фінальну візуалізацію. Кожен крок обробки відтворюється на схемі, що допомагає розробникам та користувачам зрозуміти, як інформація проходить через систему, і як різні елементи взаємодіють для створення кінцевого зображення.

Детально відпрацьований інтерфейс графічного модуля, розроблений на основі глибокого розуміння процесів візуалізації, дозволяє досягти максимальної точності у відтворенні даних, забезпечуючи, таким чином, високі стандарти якості зображень. Враховуючи складність завдань, які вирішує графічний модуль, використовується низка оптимізацій та алгоритмів для забезпечення швидкої обробки великих масивів даних і створення реалістичного зображення з високим рівнем деталізації.



Рисунок 3.2 - Блок-схема графічного модулю

Процес обробки даних, введених користувачем, є одним з найбільш важливих аспектів роботи графічного модуля. Ця функція вимагає точної координації між різними частинами системи, щоб забезпечити, що кожна дія користувача, така як переміщення курсору або вибір точки на моделі, миттєво реєструється та перетворюється в зміни в тривимірній моделі. Це дозволяє користувачам в реальному часі бачити результати своїх дій на екрані, що підвищує

взаємодію та задоволення від роботи з програмою. Використання фреймворку PyGame значно спрощує цю задачу, оскільки він надає велику кількість вбудованих функцій для обробки графічних даних і взаємодії з користувацькими пристроями вводу.

Побудова тіла на екрані є другою значущою функцією графічного модуля. Використовуючи складні алгоритми та техніки візуального рендерингу, модуль створює деталізовані та об'ємні візуалізації на основі точок тіла, розрахованих математичним модулем. Завдяки PyGame, графічний модуль може використовувати вже реалізовані графічні інструменти, забезпечуючи гладкість анімацій та перехідних ефектів, що робить процес моделювання більш привабливим та інтуїтивно зрозумілим.

Після завершення побудови, створена модель транспортується до інтерфейсу користувача, де вона стає видимою на екрані. Відображення готової фігури на екрані є завершальним етапом взаємодії між користувачем і системою, і важливо, щоб цей етап був виконаний без затримок і помилок. Графічний модуль повинен забезпечувати, що кожна зміна, зроблена користувачем, миттєво і точно відображається на екрані, дозволяючи користувачу бачити результат своєї роботи і робити необхідні корекції в реальному часі. Це вимагає від системи великої обчислювальної потужності та оптимізованого коду, здатного швидко адаптуватися до змін і відповідати на запити користувача.

В результаті, графічний модуль виконує вирішальну роль у створенні інтуїтивно зрозумілого, візуально багатого, і взаємодійного досвіду для користувачів. Він перекладає технічні дані в зрозумілі візуальні образи, що робить процес моделювання доступним і зрозумілим для широкого спектру користувачів.

3.3 Модуль розрахунків перетворень

Розрахунковий модуль займає центральне місце в архітектурі цієї комплексної програмної системи, відіграючи вирішальну роль у обчислювальному процесі перетворення тривимірних об'єктів. Цей модуль, як місток між користувачем та кінцевим візуальним результатом, виконує серію обчислень, які

забезпечують точне та динамічне моделювання форм. Він ретельно аналізує інформацію, введену користувачем, трансформує її за допомогою складних математичних операцій, заснованих на теорії полікоординатних перетворень, і виробляє нові координати точок, які служать основою для побудови деталізованої тривимірної моделі.

У цьому процесі використовується програмний фреймворк NumPy, який є одним з найпотужніших інструментів у сфері наукових обчислень і даних. Він дозволяє оптимізувати роботу з великими об'ємами даних, включаючи масиви та матриці, і забезпечує швидкість обчислень на рівні нативних мов програмування. Завдяки цьому, розрахунковий модуль може швидко виконувати складні математичні перетворення без затримок, забезпечуючи плавну взаємодію з користувачем і негайну відповідь на його дії.

Розрахунковий модуль починає свою роботу з прийняття вхідних даних – координат, які були введені або змінені користувачем за допомогою інтерфейсу додатку. Ці дані проходять через низку алгоритмів і обчислень, які оновлюють стан точок, з яких буде складатися модель. Результатом є новий набір точок, який вже відображає зміни, зроблені користувачем, і готовий до подальшої візуалізації.

Для наочності процесу роботи розрахункового модуля було розроблено графічне зображення у вигляді блок-схеми, яку можна побачити на рисунку 3.3. Ця схема детально відображає кожен крок обробки вхідних даних, показуючи їх потік від моменту отримання від користувача до моменту передачі обчислених результатів до наступного компонента системи – графічного модуля. На схемі відзначені ключові вузли та з'єднання, які ілюструють логіку та послідовність обчислювальних процесів, що гарантують ефективне та правильне виконання завдань.

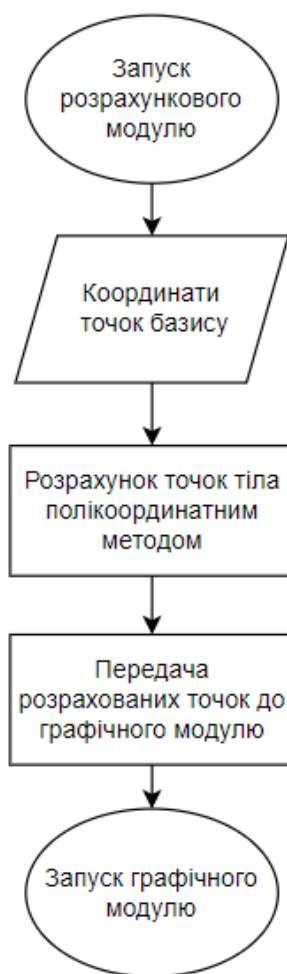


Рисунок 3.3 - Блок-схема розрахункового модулю

Однак, потужність розрахункового модуля не обмежується лише перерахунком координат. Він також обладнаний внутрішніми механізмами для оптимізації обчислень, здатний виконувати паралельні обчислення та використовувати кешування результатів для підвищення швидкодії при повторних обрахунках. Ці техніки дозволяють мінімізувати час, необхідний для обробки інформації, і роблять розрахунковий модуль надзвичайно ефективним в умовах, коли користувач вносить швидкі і часті зміни до тривимірної моделі.

Коли розрахунковий модуль завершує обчислення, він передає оновлені координати точок до графічного модуля, який з цієї інформації створює візуальну репрезентацію тривимірного об'єкта. Це перетворення є останнім кроком у ланцюжку обробки даних, де тривимірна модель набуває форми, видимої на екрані, що дозволяє користувачу бачити результат своїх дій у реальному часі. Таким

чином, розрахунковий модуль не лише виконує функції обрахунків, а й є інструментом, що забезпечує безперебійну інтерактивність між людиною та машиною, пропонуючи досвід, що базується на інтуїтивності, швидкодії та точності.

3.4 Інтерфейсний модуль

Інтерфейсний модуль є важливою частиною програмної системи, яка забезпечує зручний та інтуїтивно зрозумілий спосіб взаємодії користувача з різними можливостями програми. Цей модуль діє як центральний вузол, через який протікає обмін інформацією між користувачем та іншими компонентами системи, такими як розрахунковий та графічний модулі. Він відповідає за прийом інструкцій від користувача, їх аналіз та передачу необхідних даних для виконання запитів, а також за відображення отриманих результатів у зрозумілій формі.

Інтерфейсний модуль дає користувачу змогу вибрати параметри для побудови тривимірних моделей, таких як кулю, куб чи шестикутну призму. Він передбачає введення даних, таких як кількість граней або точок, які будуть використані у конструкції обраної фігури. Ці дані мають прямий вплив на складність та детальність майбутньої моделі, а тому важливо, щоб інтерфейс був організований таким чином, щоб користувач міг легко та точно вказати свої переваги без ризику внесення помилок.

На рисунку 3.4 зображена блок-схема інтерфейсного модуля, яка демонструє процедуру взаємодії користувача з програмою. Вона показує, як користувач може обрати тип тіла, яке він хоче побудувати, і які параметри потрібно зазначити для кожного типу тіла. Ця схема допомагає зрозуміти логіку роботи інтерфейсу та взаємодію користувача з програмою на різних етапах налаштувань.

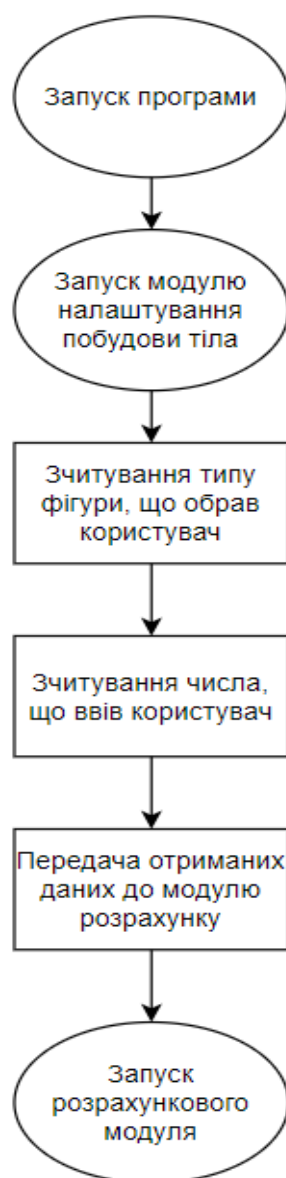


Рисунок 3.4 - Блок-схема інтерфейсного модулю

Після того, як користувач визначився з параметрами моделювання, інтерфейсний модуль перетворює введені дані в команди, які можуть бути інтерпретовані іншими модулями системи. Це дозволяє інтерфейсу служити мостом між людиною та машиною, перекладаючи дії користувача в технічні директиви, що можуть бути виконані програмою. Завдяки цьому, користувач отримує можливість керувати складним процесом створення тривимірних моделей через простий і зрозумілий графічний інтерфейс.

Інтерфейсний модуль також впливає на продуктивність програми, адаптуючи її для роботи на різноманітному обладнанні. Користувачі можуть змінювати рівень

деталізації моделі для оптимізації вимог до обчислювальної потужності. Це дає змогу програмі ефективно функціонувати на пристроях з меншими технічними ресурсами, роблячи її доступною для ширшого кола користувачів із різноманітним обладнанням.

Від інтерфейсного модуля залежить, наскільки зручною та зрозумілою буде програма для користувача. Цей модуль не лише відображає інформацію, але й активно допомагає користувачам у прийнятті рішень, надаючи їм рекомендації та візуальні підказки при виборі параметрів. Саме тому розробка інтерфейсу вимагає уважності до деталей та дослідження користувацького досвіду, щоб забезпечити високу якість візуалізації та плавність взаємодії.

Велика увага приділяється логіці навігації, зрозумілості візуальних елементів, та ергономіці інтерфейсу. Все це має на меті створити безперешкодне та приємне середовище для користувачів, яке сприяє залученості та підвищує ефективність використання програми. Інтерфейсний модуль, як інструмент взаємодії, таким чином, відіграє критичну роль у загальній архітектурі програмних рішень, підсилюючи зв'язок між користувачем і обчислювальними потужностями, що стоять за програмою.

3.5 Вибір інструментарію та середовища розробки для програмного продукту

Розробка програмного продукту завжди починається з ретельного планування та визначення ключових технічних аспектів, які будуть формувати основу проекту. Перед тим як приступити до написання коду, необхідно здійснити глибокий аналіз вимог, визначити цілі та задачі, які програма повинна вирішувати, а також врахувати майбутнє середовище її використання. Це включає розгляд можливих технічних обмежень, потреб користувачів, а також перспектив розвитку та підтримки програмного продукту.

3.5.1 Критерії та раціональний вибір мови програмування для проекту

Одним з найважливіших рішень у процесі розробки є вибір мови програмування, який визначає не тільки технологічний стек, але й спосіб реалізації багатьох функціональних можливостей програми. Слід враховувати, що мова програмування повинна відповідати ряду важливих критеріїв, таких як:

- продуктивність;
- гнучкість;
- доступність ресурсів для навчання та підтримки;
- можливість інтеграції із супутніми технологіями та інструментами;

Виходячи з цих міркувань, було проведено детальне дослідження існуючих мов програмування, їхніх особливостей та екосистем. Важливим фактором стала підтримка різноманітних бібліотек, особливо тих, що дозволяють працювати з графікою, адже програма вимагала реалізації складних графічних функцій. Також було важливо знайти мову з сильною підтримкою і розгалуженою спільнотою, де можна було б швидко отримати допомогу або вирішити проблеми, що виникають під час розробки. Іншим значним аспектом була доступність високоякісного інтегрованого середовища розробки, яке надавало б широкий спектр інструментів та функцій для полегшення всього процесу розробки.

З урахуванням цих параметрів вибір звузився до декількох мов програмування, які займають лідируючі позиції в індустрії і мають велику спільноту користувачів. Мови, такі як C++, Python, Java та C#, кожна зі своїми унікальними перевагами та особливостями, були розглянуті як потенційні кандидати для використання в проекті. Проте, після додаткового аналізу та оцінки особистого досвіду роботи з кожною з цих мов, вибір припав на Python, оскільки він пропонує багату екосистему, високу портативність та чудову легкість читання коду. Більше того, Python має широкий вибір бібліотек та модулів для роботи з графікою, що є вирішальним для реалізації графічного модуля програми, а також велику кількість навчальних матеріалів та прикладів реалізації різноманітних задач.

3.5.2 NumPy, як інструмент наукових обчислень у Python

Створення програмного продукту – це процес, що вимагає не тільки технічних навичок та досвіду, але й глибокого розуміння проблем, які можуть виникнути на шляху розробки. Кожен етап від ідеї до втілення має бути ретельно продуманий, щоб мінімізувати ризики та максимізувати шанси на успіх. Перед самим кодуванням стоїть завдання визначення найбільш оптимальних інструментів, які будуть задіяні у процесі створення програми. Це стосується не тільки вибору мови програмування, але й вибору бібліотек та середовища розробки, яке полегшить та прискорить розробку, дозволяючи зосередитися на реалізації креативної та функціональної частини проекту.

Інструментальна частина розробки насправді є своєрідним каркасом проекту. Від правильного вибору інструментарію залежить не тільки швидкість розробки та якість кінцевого продукту, але й легкість його подальшого супроводу та розширення можливостей. В контексті програми, що займається візуалізацією та обробкою даних, важливим аспектом є робота з масивами даних і обчисленнями, тому вибір пав на бібліотеку NumPy, яка є стандартом у світі наукових обчислень на Python.

NumPy – це бібліотека з відкритим кодом, яка надає дуже велику швидкість та гнучкість при роботі з багатовимірними масивами та матрицями. Її популярність та значення в сучасній науці та інженерії обумовлені широкими можливостями та оптимізацією. Використання NumPy дозволяє здійснювати велику кількість обчислень на високій швидкості, що критично необхідно для програм, що вимагають обробки складних даних, як це є у випадку з моделюванням тривимірних тіл. Бібліотека включає в себе широкий спектр математичних функцій, які можуть вирішити більшість завдань, пов'язаних з лінійною алгеброю, статистикою, та іншими областями, що вимагають інтенсивних обчислень.

Незважаючи на свої численні переваги, NumPy також має певні обмеження, такі як треба використання однотипних даних в масивах, що може бути незручним при деяких сценаріях використання. Це обмеження можна обійти за допомогою різноманітних методів, зокрема, використання "об'єктних" масивів, що дозволяють

зберігати дані різних типів. Також було впроваджено додаткові перевірки та механізми обробки для забезпечення коректності даних. Важливо також врахувати, що робота з NumPy вимагає розуміння основ роботи з масивами та матрицями, але завдяки об'ємному ресурсу навчальних матеріалів та прикладів, можливо ефективно освоїти цю бібліотеку для розробки проекту.

3.5.3 PyGame - бібліотека для розробки графічного інтерфейсу

PyGame було обрано як ключовий інструмент для розробки візуального інтерфейсу та графічних аспектів програми, оскільки ця бібліотека відома своєю універсальністю та ефективністю у створенні двовимірних графічних додатків. Вона надає широкий спектр можливостей для створення інтерактивних елементів, динамічних анімацій, обробки подій від користувача, а також управління зображеннями і спрайтами. Розглянемо більш докладно, чому PyGame є відмінним вибором для таких завдань.

PyGame базується на бібліотеці SDL, що надає низькорівневий доступ до графічного апаратного забезпечення, що дозволяє максимально ефективно використовувати ресурси комп'ютера. Це означає, що PyGame добре підходить не тільки для створення графічних інтерфейсів, але й для розробки ігор та інших програм, де потрібна висока продуктивність та швидке відгук на дії користувача.

Бібліотека пропонує інтуїтивно зрозуміле API для малювання фігур, тексту, завантаження та відображення зображень у різноманітних форматах. Вона також підтримує створення складних сцен із багатьма об'єктами, що мають власні властивості та поведінку. Розробники можуть легко створювати багатопланові сцени, використовуючи прозорість та перекриття, анімувати об'єкти та управляти їх взаємодією.

Ще однією перевагою PyGame є його активна та допоміжна спільнота. Незліченна кількість навчальних посібників, форумів та документації є доступною для розробників будь-якого рівня. Це не тільки сприяє легкому освоєнню бібліотеки, але й забезпечує високий рівень підтримки, коли виникають складніші питання або проблеми.

PyGame також дозволяє розробникам інтегрувати код, написаний на інших мовах програмування, якщо це необхідно для певних аспектів проекту, що робить його гнучким рішенням для комплексних проектів. Разом з тим, PyGame можна поєднувати з іншими бібліотеками Python, такими як NumPy, для підвищення продуктивності та розширення можливостей.

Що стосується обмежень PyGame, зокрема в області 3D-графіки, ці обмеження були успішно подолані шляхом залучення додаткових інструментів та використанням гібридного підходу до розробки. Такий підхід дозволяє використовувати PyGame як надійний інструмент для тих частин проекту, де його можливостей достатньо, а для більш складних завдань залучати інші технології.

У підсумку, PyGame є високоефективним інструментом для створення графічних елементів, який може бути адаптований для широкого спектру задач. Завдяки його гнучкості, простоті використання та можливості розширення, PyGame став ідеальним інструментом для створення системи візуалізації деформацій тривимірних тіл методами полікоординатних перетворень.

3.5.4 Інтегроване середовище розробки PyCharm

PyCharm, обраний як інтегроване середовище розробки (IDE), грає ключову роль у процесі створення програмного продукту. Це потужний інструмент, який надає розробникам великий набір функціональних можливостей для підвищення продуктивності та ефективності розробки. Розглянемо детальніше переваги та функції, які пропонує PyCharm, що стали вагомими аргументами для його вибору.

PyCharm відрізняється своєю комплексністю та вбудованими інструментами, які включають розширену систему підказок коду, глибокий аналіз якості коду, інструменти для рефакторингу, інтегровані засоби для дебагу та тестування коду. Ці засоби забезпечують інтуїтивно зрозуміле та ефективне оточення для розробників, сприяючи зменшенню часу на пошук та виправлення помилок, а також покращенню якості коду.

Однією з ключових особливостей PyCharm є його інтеграція з різноманітними версійними системами контролю, такими як Git, Mercurial,

Subversion та інші, що дозволяє легко управляти змінами в коді, співпрацювати з іншими розробниками та відстежувати прогрес проекту. Інтеграція з системами баз даних та веб-сервісами також надає можливість прямої роботи з цими компонентами зсередини IDE, що значно спрощує процес розробки складних застосунків.

PyCharm також дозволяє налаштовувати середовище під конкретні потреби проекту, забезпечуючи індивідуальний підхід до розробки. Гнучкі налаштування дозволяють настроїти робоче середовище відповідно до особистих вподобань, що підвищує продуктивність та спрощує виконання поставлених задач.

Велика кількість доступних плагінів та доповнень збільшують можливості PyCharm ще більше, дозволяючи інтегрувати додаткові інструменти, фреймворки та бібліотеки. Це підвищує ефективність розробки, оскільки розробник має в розпорядженні все необхідне в одному середовищі.

Втім, PyCharm має деякі недоліки, такі як високі вимоги до ресурсів системи, що може призвести до зниження продуктивності на менш потужних комп'ютерах. Щоб протидіяти цьому, було оптимізовано налаштування PyCharm, обрано лише необхідні плагіни та інструменти, що зменшило навантаження на систему. Крім того, для полегшення роботи з PyCharm було вивчено багато навчальних матеріалів та відеоуроків, що сприяло швидкому освоєнню інструментарію та інтеграції середовища в робочий процес.

Загалом, PyCharm став незамінним помічником у розробці, надаючи можливість ефективної організації робочого процесу, автоматизації багатьох задач та забезпечуючи високу якість коду. Його розширені функціональні можливості, поєднані з можливістю глибокої кастомізації та інтеграції з іншими інструментами розробки, роблять PyCharm ідеальним середовищем для створення складних та високофункціональних програмних продуктів.

Висновки до розділу 3

У даному розділі на основі детального аналізу технічних аспектів і потреб задачі моделювання деформацій тривимірних тіл, було розглянуто комплексний

опис програмної системи з акцентом на інструментарій та середовище розробки. Визначальним є розподіл функціональності по модулях: інтерфейсний модуль служить точкою взаємодії користувача з програмою, розрахунковий модуль відповідає за обчислювальну частину, а графічний модуль, реалізований на основі PyGame, забезпечує візуалізацію результатів.

Окрім того, оглянуто специфікації кожного інструменту та їхню взаємодію в контексті полікоординатних перетворень. Обговорено використання бібліотеки NumPy для модуля розрахунків, що значно посилює здатності системи до ефективного виконання обчислень, пов'язаних із обробкою багатовимірних даних. Також підкреслено, як інтегроване середовище розробки PyCharm сприяє зручності та продуктивності в написанні та тестуванні коду, надаючи розробникам всебічну підтримку і поліпшуючи якість програмного продукту.

Важливо зазначити, що за допомогою модульної структури та ефективного вибору інструментарію, програмна система набуває високого рівня адаптивності, дозволяючи легко вносити зміни або доповнення до проекту. Це створює можливості для масштабування програми відповідно до зростаючих і змінних вимог користувачів та уявлень про деформації тривимірних тіл.

Систематизований підхід до вибору компонентів, починаючи від мови програмування та закінчуючи середовищем розробки, відкриває шлях до ефективної реалізації проекту. Python виявився ідеальним варіантом для реалізації задачі, оскільки забезпечує гнучкість та велику екосистему бібліотек, чого достатньо для розв'язання найбільш складних завдань у контексті візуалізації та обчислень.

Завдяки стабільній та надійній платформі, що спроектована з урахуванням потенційного масштабування, програмна система відкрита для подальшого розвитку, інтеграції нових алгоритмів перетворень або вдосконалення інтерфейсу користувача. Рішення, прийняті на етапі планування та розробки, гарантують, що програма може бути ефективно використана для широкого спектру задач, включаючи наукові, освітні та промислові застосування, де необхідна точна візуалізація та аналіз тривимірних форм.

4 РОБОТА КОРИСТУВАЧА З РОЗРОБЛЕННОЮ СИСТЕМОЮ

Програма для візуалізації тривимірних тіл, розроблена з урахуванням потреб сучасного користувача, володіє унікальною властивістю – підтримкою різних операційних систем, а саме: Windows, Linux та macOS. Це було досягнуто завдяки створенню індивідуальних виконуваних файлів для кожної платформи, що є значним кроком у забезпеченні універсальності та доступності програми.

Для компіляції вихідного коду в автономні виконувани файли були використані спеціалізовані інструменти, а саме PyInstaller. Цей інструмент забезпечує збірку програми з усіма необхідними бібліотеками та залежностями в єдиний файл, що суттєво спрощує процес розповсюдження та встановлення програми на кінцевому обладнанні. Використання таких інструментів дозволяє інкапсулювати всі компоненти програми, включаючи інтерпретатор Python, модулі NumPy та PyGame, а також додаткові ресурси, необхідні для її функціонування.

Переваги цього підходу полягають у високій мобільності програми та зручності для користувача. Немає потреби в додатковому завантаженні та налаштуванні середовища Python чи встановленні бібліотек – усе вже міститься всередині виконуваного файлу. Це також означає, що програма має однакову поведінку та інтерфейс на всіх підтримуваних платформах, забезпечуючи користувачам аналогічний досвід роботи з програмою незалежно від їхньої операційної системи.

Ще однією значною перевагою є те, що програма не потребує традиційного встановлення. Оскільки всі необхідні компоненти вже вбудовані в .exe файл, користувач просто завантажує файл з Google Drive та запускає його локально. Це не тільки спрощує початок роботи з програмою, але й знижує потенційні технічні бар'єри для користувачів без розширених комп'ютерних навичок, так як вони не повинні замислюватися про налаштування та конфігурацію різних програмних складових.

Проте, створення таких самодостатніх виконуваних файлів має і свої недоліки. Наприклад, збільшення розміру файлу, оскільки всі залежності містяться

всередині нього, може стати проблемою для користувачів із обмеженим простором на жорсткому диску або повільним інтернет-з'єднанням. Також відсутність традиційних систем оновлення означає, що для отримання нової версії програми користувачам необхідно заново завантажити повний .exe файл, що може бути незручним і часозатратним.

І хоча ці недоліки є реальністю, вони були враховані під час розробки. Наприклад, для мінімізації розміру виконуваних файлів було виконано оптимізацію ресурсів та коду, вибираючи лише найнеобхідніші бібліотеки для включення до дистрибутива. Щоб знизити навантаження на користувачів при загрузці програми, було введено чіткі інструкції та зручний механізм завантаження виконуваного файлу з Google Drive, де користувачі можуть легко знайти та завантажити відповідні файли для своєї ОС.

Таким чином, програма для візуалізації тривимірних тіл забезпечує користувачам легкий доступ до потужних інструментів моделювання без необхідності обтяжливого процесу встановлення та налаштування, пропонуючи сучасне, зручне і ефективне рішення для вирішення складних завдань в галузі тривимірної графіки та візуалізації.

4.1 Системні вимоги до комп'ютера

Для забезпечення стабільної та комфортної роботи з програмою візуалізації тривимірних тіл, важливо, щоб обчислювальна техніка користувача відповідала певним системним вимогам. Нижня межа вимог слугує гарантією того, що програма зможе ефективно виконувати необхідні обрахунки та забезпечить гладке відображення фігур, без затримок чи інших технічних проблем під час роботи.

Мінімальні системні вимоги для операційної системи Windows включають:

- операційна система: Windows 7 або новіша версія, 64-розрядна;
- відеокарта: мінімум з 1 ГБ відеопам'яті;
- оперативна пам'ять: мінімум 2 ГБ;
- процесор: з базовою тактовою частотою від 2.5 до 2.9 ГГц.

Для користувачів інших операційних систем, таких як macOS та Linux, вимоги можуть легше варіюватися, але загальні рекомендації залишаються схожими:

- операційна система: macOS X Sierra (10.12.6) або новіша, Linux Ubuntu 16.04 або новіша;
- відеокарта: мінімум з 1 ГБ відеопам'яті, підтримка OpenGL;
- оперативна пам'ять: мінімум 4 ГБ для комфортної роботи;
- процесор: з тактовою частотою 2.5 ГГц або вище.

Більш високі конфігурації обчислювальної техніки, очевидно, забезпечать кращу продуктивність та більш високу якість візуалізації, особливо при роботі з високими параметрами деталізації фігур. Якщо система не відповідає мінімальним вимогам, користувач може зіткнутися з різними проблемами, зокрема зі зниженням швидкості реакції програми, затримками в обрахунках або неможливістю коректного відображення складних моделей.

Важливо відзначити, що для забезпечення максимальної зручності та продуктивності роботи з програмою, користувачам рекомендується використовувати обладнання, що перевищує мінімальні вимоги. Це дозволить не тільки з комфортом працювати з базовими функціями програми, але й розширить можливості для більш складних та ресурсоємних операцій, таких як робота з великими даними, складні перетворення та детальна візуалізація тривимірних об'єктів.

4.2 Встановлення програми на комп'ютер

Для початку роботи з програмою з візуалізації тривимірних тіл потрібно виконати кілька простих кроків. З метою спрощення процесу інсталяції та забезпечення максимальної сумісності з різними операційними системами, програма була розподілена на три окремих виконуваних файла, кожен з яких

оптимізований для своєї ОС – Windows, Linux або macOS. Це дозволяє користувачу обрати варіант, який відповідає його операційній системі, і гарантує оптимальну роботу програми.

Всі три версії програми були завантажені на Google Drive, зручний хмарний сервіс для зберігання та дистрибуції файлів. Завантаження файлів з Google Drive гарантує користувачам легкий доступ до них з будь-якого місця та на будь-якому пристрої з підключенням до Інтернету. Кожен файл має пряме посилання, що робить процес завантаження простим та зрозумілим.

Отже, користувач має зайти за посиланням на сторінку Google Drive, як це видно на рисунку 4.1, де він знайде три виконувані файли – кожен під певну операційну систему.

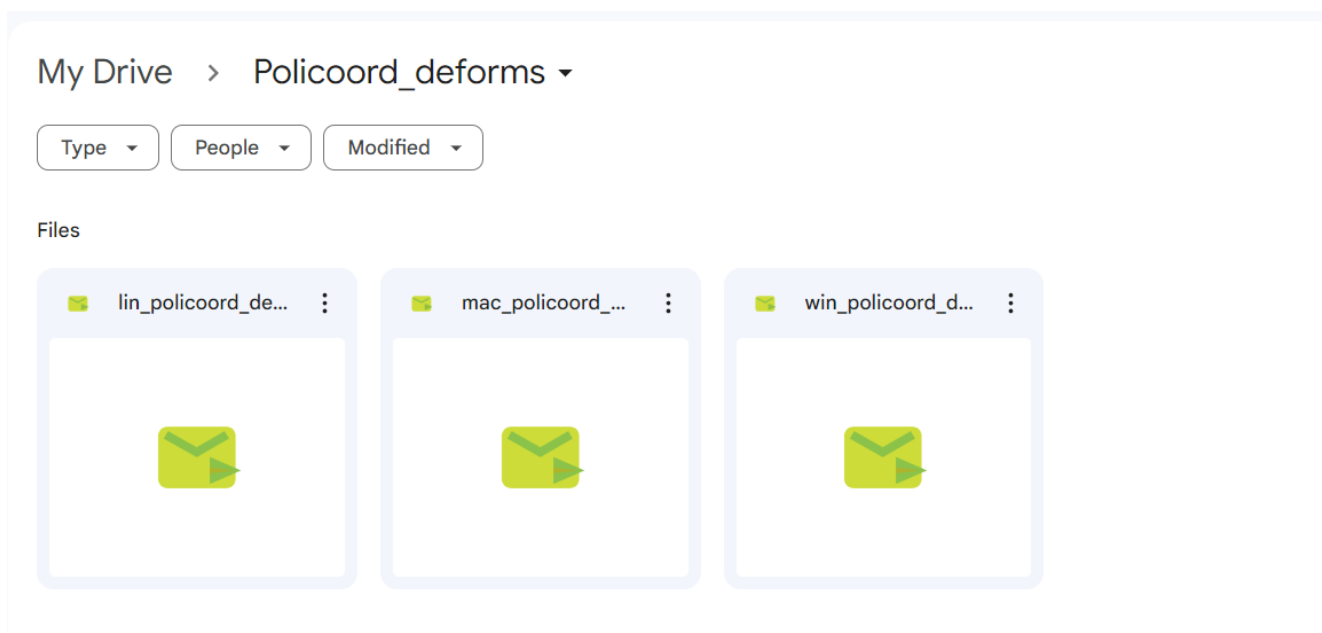


Рисунок 4.1 – Перелік виконуваних файлів на гугл-диску

Натиснувши правою кнопкою миші на потрібному файлі, залежно від операційної системи користувача, з'явиться контекстне меню – рисунок 4.2.

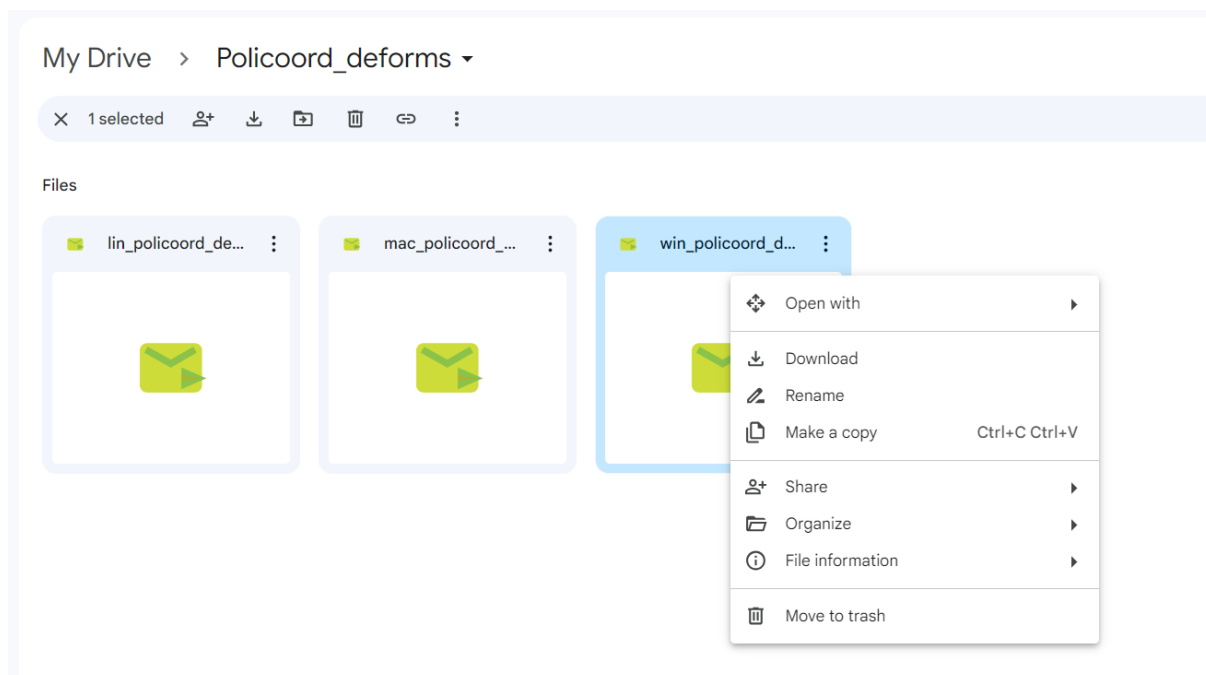


Рисунок 4.2 – Контекстне меню після вибору файлу

У з'явившомуся вікні варто обрати опцію "Download", як це показано на рисунку 4.3. Ця дія ініціює процес збереження виконуваного файлу на локальний диск користувача. Завдяки тому, що всі необхідні модулі вже включені в виконуваний файл, немає різниці в яке місце користувач завантажить .exe файл, адже для коректної роботи програми не потрібні жодні сторонні модулі, або залежності.

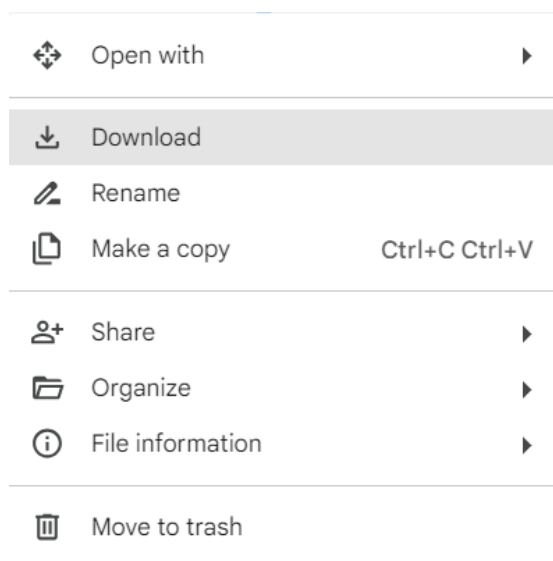


Рисунок 4.3 – Завантаження програми на комп'ютер

Після завершення завантаження, користувачу залишається лише знайти файл у папці до якої його було завантажено та активувати його, зазвичай подвійним кліком миші по іконці файлу, що розпочне виконання програми, як це і відбувається на рисунку 4.4.

Name	Date modified	Type	Size
▼ Today			
 win_policoord_deforms	12/3/2023 4:00 PM	Application	119 KB
 deformations	12/3/2023 4:02 PM	File folder	
 Telegram Desktop	12/3/2023 4:01 PM	File folder	

Рисунок 4.4 – Додаток завантажений на комп'ютер

Відтак на екрані має з'явитися інтерфейс програми, де першим вікном буде вікно вибору тривимірного тіла для побудови та подальшої деформації.

Цілісність такого підходу до дистрибуції програми забезпечує користувачам зручний та надійний спосіб отримання та запуску програми, економлячи час та виключаючи необхідність додаткових кроків по налаштуванню або встановленню зовнішніх залежностей. Різні версії виконуваних файлів для різних ОС гарантують, що користувачі зможуть використовувати програму в різних середовищах без жодних додаткових зусиль, що робить програму максимально доступною та корисною для широкого спектру користувачів.

4.3 Меню вибору початкового тіла для побудови з подальшим налаштуванням

Щойно користувач ініціює запуск програми, перед ним відкривається інтерактивне меню вибору, де він стикається з першим ключовим етапом взаємодії з програмою – вибором фігури для подальшої візуалізації та деформації. Програма демонструє свою багатofункціональність, надаючи користувачеві можливість обрати одну з трьох основних фігур: сферу, куб або шестикутну призму - рисунок

4.5. Цей елемент інтерфейсу виконаний таким чином, щоб користувач зміг легко зрозуміти наявні опції та зробити інформований вибір.

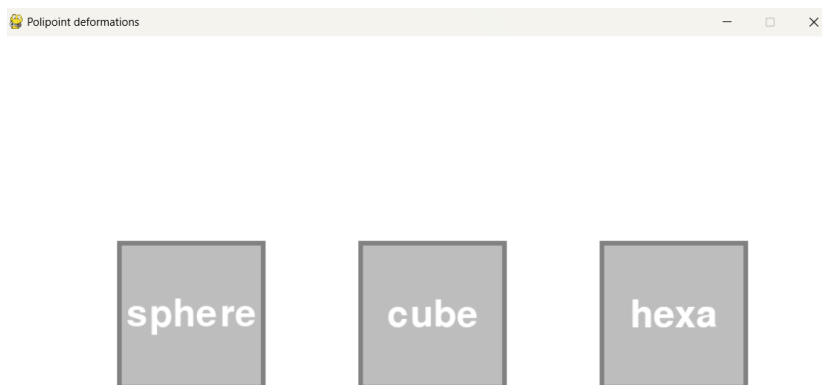


Рисунок 4.5 – Вікно з вибором фігури для побудови

Після того, як користувач робить свій вибір серед трьох доступних фігур для візуалізації – куба, сфери чи шестикутної призми – він переходить до наступного етапу взаємодії з програмою: налаштування параметрів побудови обраної фігури. Цей крок здійснюється через спеціалізоване вікно налаштувань (рис. 4.6), яке має одне основне поле для введення даних, що відповідає за деталізацію візуалізованої фігури.

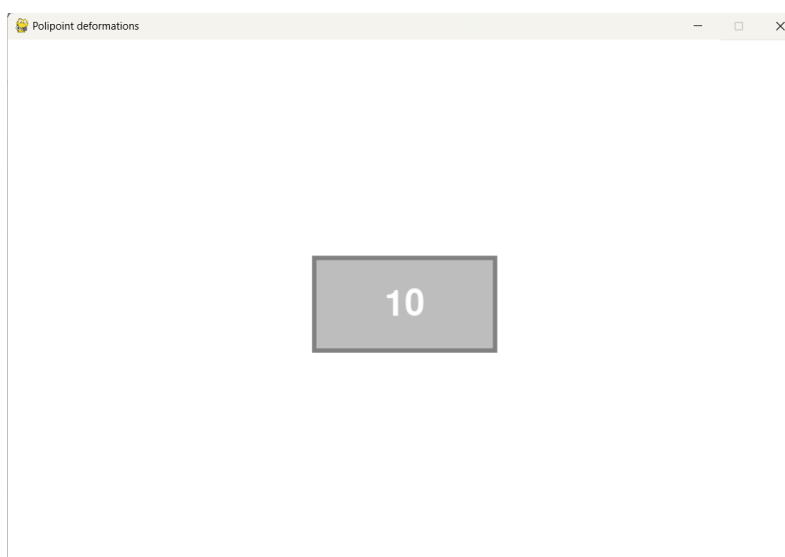


Рисунок 4.6 – Вікно з налаштуванням деталізації побудови фігури

У цьому вікні налаштувань, користувач зустрічається з текстовим полем, де вже задано універсальне значення, яке балансує між якістю візуалізації та оптимальним навантаженням на комп'ютер. Це універсальне значення є рекомендованим стартовим пунктом, яке може бути змінено в залежності від переваг користувача або вимог до якості моделі. Введення більшого числа призведе до створення більш деталізованої фігури з вищою кількістю полігонів, що, в свою чергу, може збільшити навантаження на систему.

Програма включає захист від некоректного введення даних, забезпечуючи, що в текстове поле можуть бути введені тільки числові значення. Якщо користувач спробує ввести літери, символи або спеціальні знаки, ці дані ігноруються, і вони не з'являться в полі. Це запобігає введенню помилкової інформації та забезпечує правильність параметрів для побудови фігури. Якщо користувач спробує ввести нульове значення, то вікно налаштувань підсвітиться червоним кольором, сигналізуючи про недопустимість такого введення і заблокує перехід до наступного кроку, що саме ми і бачимо на рисунку 4.7.



Рисунок 4.7 – Реакція програми на спробу ввести некоректні дані

У процесі налаштування параметрів візуалізації тривимірних фігур, програма забезпечує додаткові візуальні підказки для полегшення введення даних користувачем. Коли текстове поле, призначене для введення числових значень, є активним, користувач може помітити, що навколо нього з'являється зелена рамка (рис.4.8). Це візуальний індикатор, який підкреслює, що поле готове до введення даних, і є частиною вдосконаленого інтерфейсу з метою зробити його максимально зручним та інтуїтивно зрозумілим.



Рисунок 4.8 – Активне поле вводу

У підсумку, процес налаштування побудови фігури у програмі є інтуїтивним і захищеним від помилок, що забезпечує користувачу зручність та впевненість у тому, що він зможе ефективно та безперешкодно керувати процесом візуалізації.

4.4 Перетворення та деформації тіла діями користувача

Після завершення етапу налаштування параметрів тривимірної фігури, користувач стикається з інтерактивною візуалізацією обраного об'єкта, реалізованого в програмі. Розглянемо на прикладі шестикутної призми. На екрані відображається її точний тривимірний аналог (рисунок 4.9), готовий до подальшої деформації та перетворення з використанням методу полікоординатних перетворень. Модель, що відображається, не лише демонструє кінцевий результат налаштувань, але й служить вихідним пунктом для подальших інтеракцій з програмою. Важливо відзначити, що візуалізація забезпечує не просто статичне

зображення, а високоякісне, інтерактивне відображення, яке можна обертати та деформувати відповідно до методу полікоординатних перетворень, на вимогу користувача.

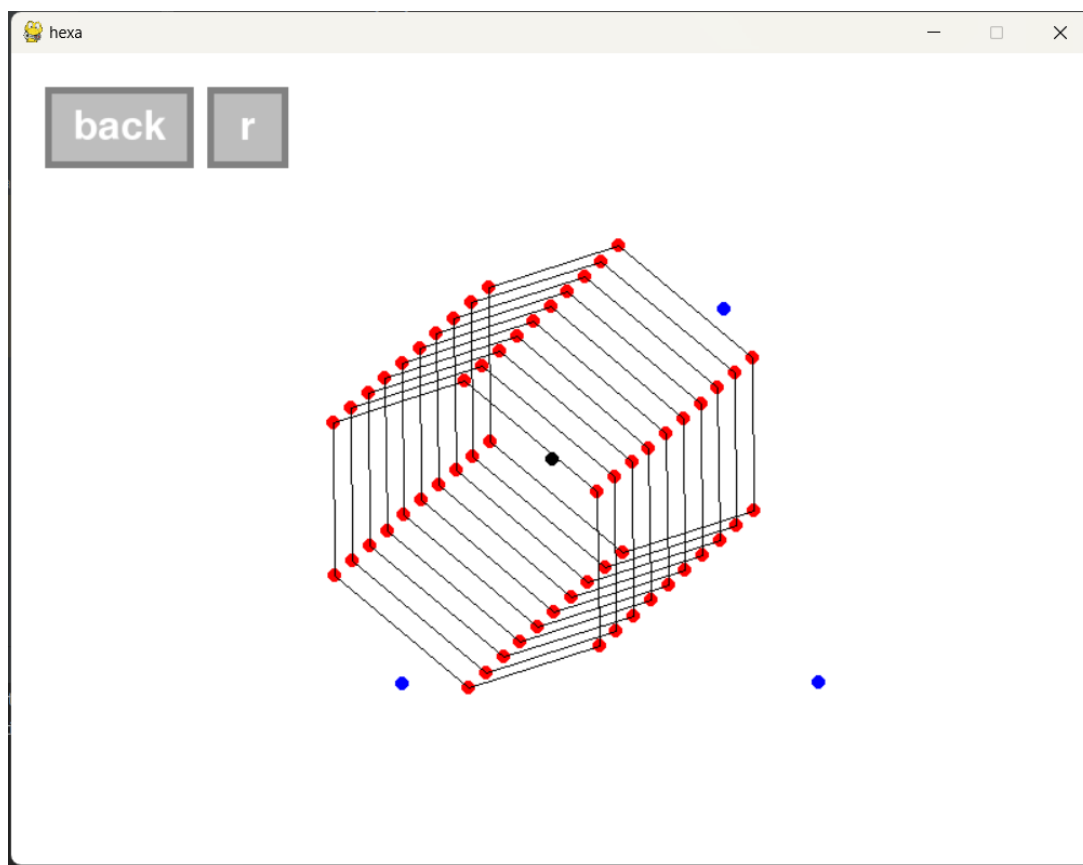


Рисунок 4.9 – Початкова візуалізація шестикутної призми

Коли процес візуалізації фігури завершено, користувач отримує не лише детальне тривимірне зображення обраного об'єкта, але й зіткнеться з додатковими елементами на сцені, які мають вирішальне значення для подальших маніпуляцій з моделлю. Ці елементи – це точки базису, закрашені у яскравий синій колір, які розташовані поза фігурою та відіграють ключову роль у процесі полікоординатних перетворень. Точки базису розглянемо більш детально коли будемо говорити про деформацію фігур.

У центрі кожної фігури відображається точка чорного кольору, яка візуально позначає центр фігури, а також співпадає з центром екрану. Ця точка забезпечує важливий орієнтир для користувача при взаємодії з моделлю. Вона не має властивостей зміщення та залишається нерухомою в процесі редагування, надаючи

стабільну відправну точку для всіх перетворень. Чорна точка в центрі допомагає користувачеві легше візуалізувати розташування фігури в просторі та сприяє більш точному та продуманому підходу до модифікацій.

4.4.1 Обертання фігури навколо своєї осі

Однією з фундаментальних характеристик програми для візуалізації тривимірних тіл є можливість користувача вільно обертати фігуру у тривимірному просторі, що надає детальне уявлення про її форму та об'єм. Ця функціональність є ключовим інструментом для оцінювання та редагування тривимірної моделі, надаючи користувачеві можливість огляду об'єкта з різних перспектив і кутів зору.

Щоб активувати механізм обертання, користувачу достатньо вибрати ділянку на екрані, яка не містить контрольних точок – маленьких синіх маркерів, розташованих поза фігурою. Ці контрольні точки слугують для інших маніпуляцій з моделлю, таких як деформація за методом полікоординатних перетворень. Після вибору вільного простору, користувач затискає ліву кнопку миші та починає вести курсор в бажаному напрямку, що призводить до обертання фігури на екрані, рисунок 4.10.

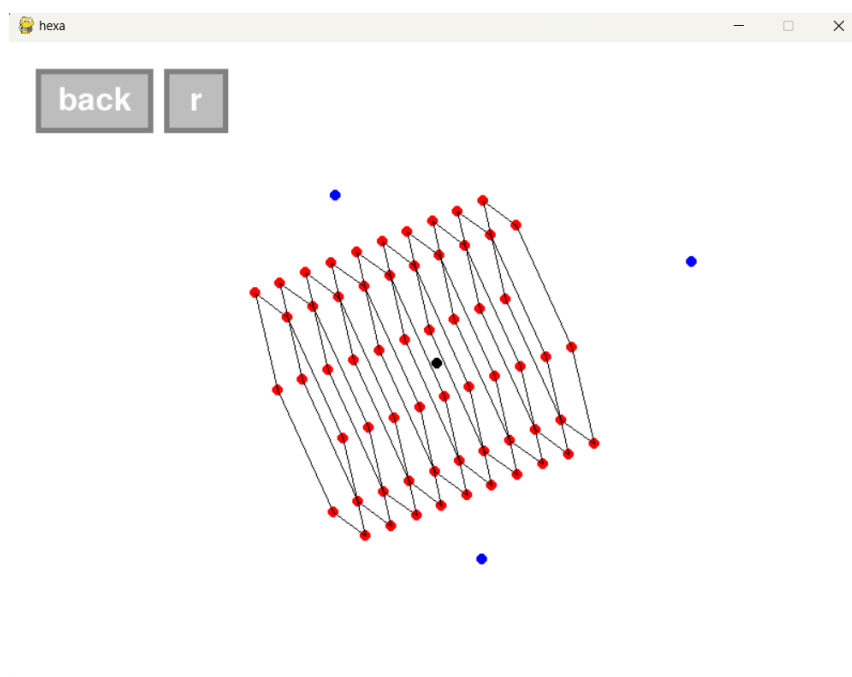


Рисунок 4.10 – Фігура після обертну

Обертання моделі в програмі відбувається плавно і синхронізовано з рухами миші, що дозволяє користувачеві легко контролювати процес і отримувати миттєвий відгук від програми. Це створює ефект безпосереднього взаємодії з моделлю, ніби користувач має можливість фізично торкатися та обертати об'єкт у руках. Візуальне відображення об'єкта оновлюється в реальному часі, що демонструє високу реактивність програми та її здатність до швидкого оброблення даних.

Для більш зручної роботи з моделлю та її огляду з усіх боків, в лівому верхньому куті інтерфейсу реалізовано спеціальну кнопку, яка активує автоматичне обертання фігури навколо всіх її осей одночасно з постійною швидкістю.

Ця кнопка, проста у використанні, забезпечує користувачам зручний механізм для спостереження за фігурою без необхідності вручну змінювати її положення. З її допомогою можна легко оцінити всі аспекти дизайну, а також виявити деталі, які можуть бути непомітні при статичному відображенні. Функція автоматичного обертання є невід'ємною для детального аналізу тривимірних моделей та може бути корисною як для навчальних цілей, так і для професійного застосування, наприклад, у промисловому дизайні чи архітектурі.

Інтерфейсна кнопка для управління автоматичним обертанням має інтуїтивно зрозуміле візуальне кодування стану: коли автоматичне обертання включено, кнопка світиться зеленим кольором, що вказує на активність режиму. Коли функція обертання вимкнена, колір кнопки змінюється на сірий, що універсально сприймається як індикатор відсутності дії або неактивного стану, рисунок 4.11.



Рисунок 4.11 – Порівняння відображення вимкненої та увімкненої кнопки автоматичного обертання

Ця кнопка важлива для тих, хто хоче заощадити час під час роботи з програмою, оскільки дозволяє зафіксувати повний оберт візуалізації фігури без постійного втручання з боку користувача.

Механізм обертання також є важливим інструментом для аналітичної роботи з моделлю. Він дозволяє виявити потенційні недоліки у дизайні, перевірити рівномірність форми та оцінити відповідність до заданих вимог. Це може бути використано як для освітніх цілей, так і в професійних сферах, де точне розуміння геометрії об'єкта є критично важливим.

Програма забезпечує високу точність у відтворенні об'єктів, а розвинуті інструменти обертання та перетворення роблять її могутнім інструментом у руках користувачів, що вивчають, проектують або працюють з тривимірними моделями в різних областях діяльності. Відтак, можливість обертання моделі не обмежується лише однією фігурою, але розповсюджується на усі фігури, доступні в програмі, забезпечуючи універсальність та гнучкість в роботі з тривимірними об'єктами.

4.4.2 Деформація фігури методом полікоординатних перетворень

Програма надає користувачам можливість глибокої взаємодії з тривимірною фігурою через механізм полікоординатних перетворень, а саме шляхом переміщення точок базису, про які ми вже згадували у пункті 4.2.

Користувачу необхідно навести курсор миші на обрану точку базису, після чого зажати ліву кнопку миші, та переміщати курсор у довільному напрямі. Точка базису стає активною та змінює свій колір на зелений, а також переміщується разом з курсором користувача. Це візуальне змінення кольору є сигналом для користувача, що вказує на успішну активацію точки. При переміщенні відбувається візуальна трансформація фігури (рисунок 4.12), що є втіленням полікоординатного перетворення.

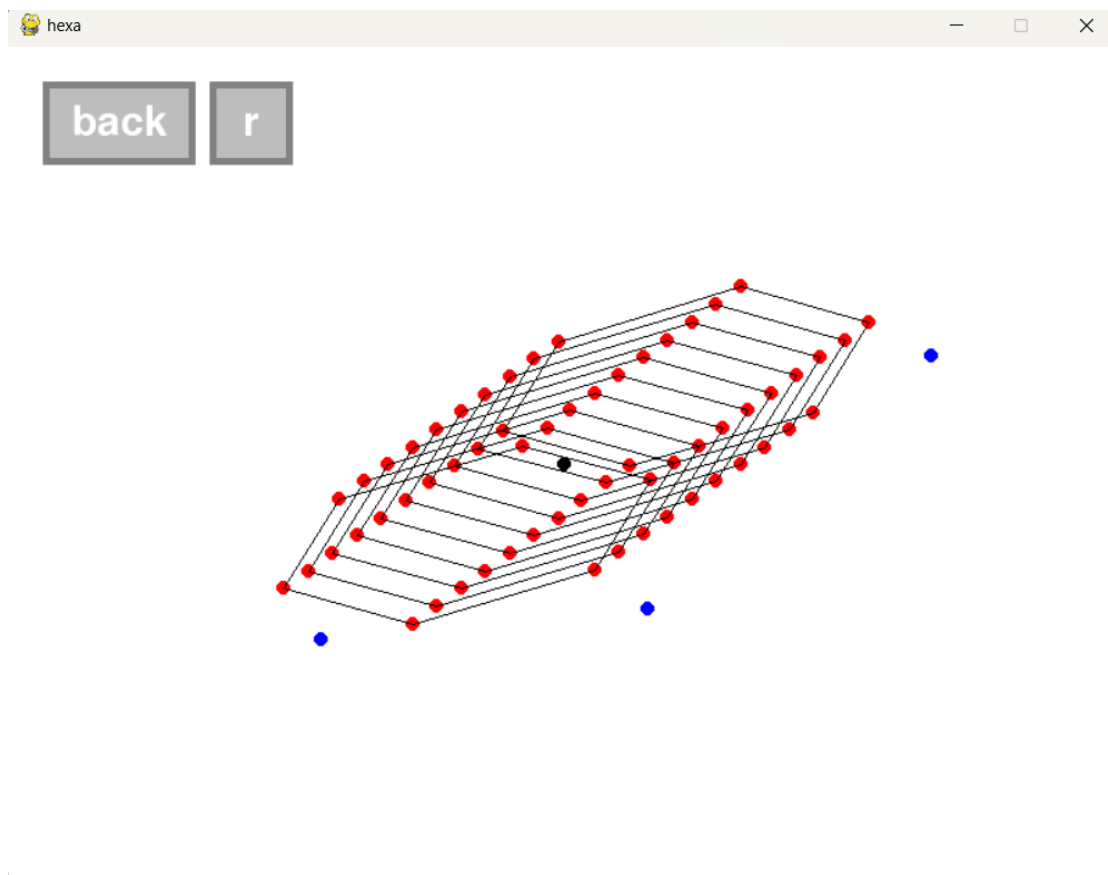


Рисунок 4.12 – Візуалізація призми після полікоординатних перетворень

Цей процес переміщення точок базису дозволяє користувачам застосовувати комплексні математичні алгоритми перетворення до фігури без необхідності глибокого розуміння всіх математичних деталей, що лежать в основі процесу.

Комбінування можливостей обертання та деформації (рисунок 4.13) в програмі є основою функціонування створеної системи. Виконуючи ці дві дії одночасно, можна досягти унікального ефекту, де фігура піддається динамічним змінам у реальному часі, надаючи користувачам змогу спостерігати за цілим спектром перетворень, що відбуваються з моделлю.

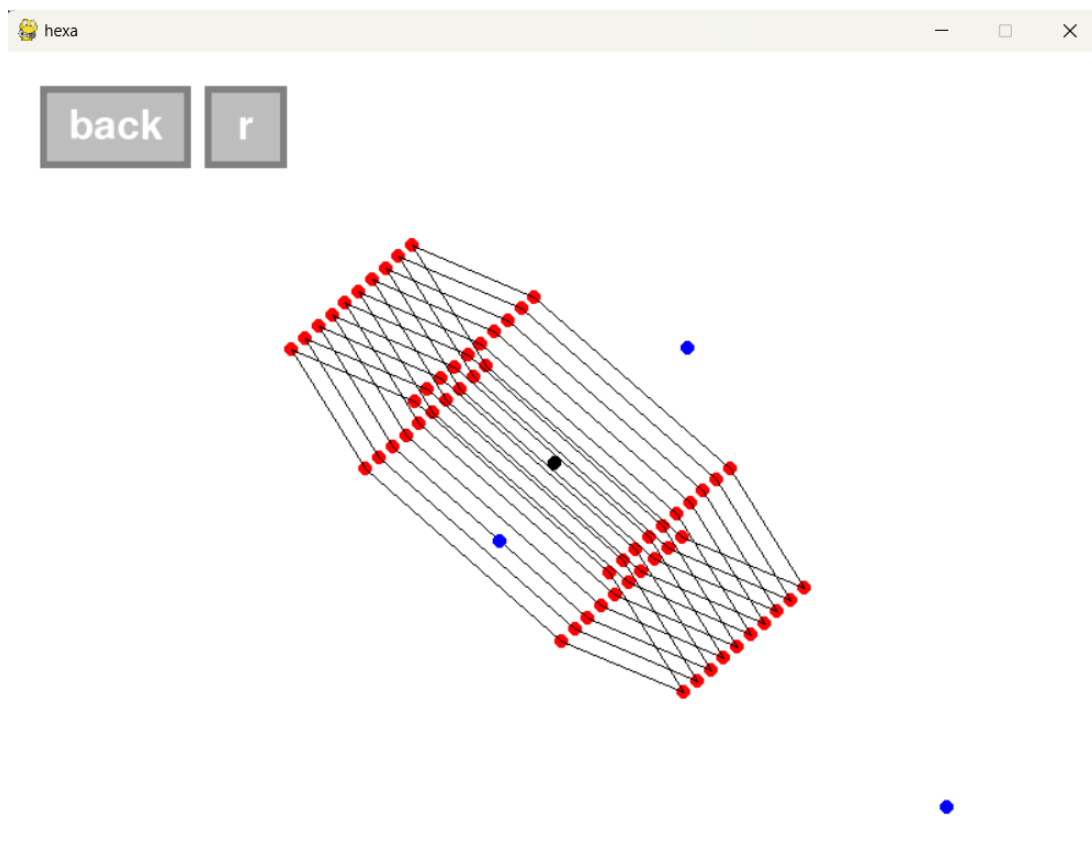


Рисунок 4.13 – Візуалізація призми після деформації та обертання

Цей комплексний підхід до моделювання дозволяє користувачу не просто створювати статичні об'єкти, але й динамічно маніпулювати їх формою, розглядаючи як вони змінюються під час обертання та під дією деформуючих сил. Така функціональність є особливо цінною при створенні наукових візуалізацій, в анімації, у геймдевелопменті, де необхідно демонструвати об'єкти у русі, а також у багатьох інших застосуваннях.

Програма надає високу точність у відтворенні цих перетворень, завдяки чому користувачі можуть ретельно вивчати кожен аспект своїх моделей. Вони можуть відстежувати, як змінюється об'єкт не тільки візуально, але й з точки зору геометрії та фізики моделювання. Завдяки злиттю обертання та деформації виникає більш глибоке розуміння того, як різні параметри впливають один на одного та як діє змінювана модель у тривимірному просторі.

4.4.3. Зміна параметру деталізації при побудові фігури

В процесі розробки програми для візуалізації тривимірних тіл ключовим аспектом є забезпечення балансу між точністю відображення об'єктів та обчислювальним навантаженням на комп'ютер користувача. Однією з основних змінних, що контролює цей баланс, є параметр, який визначає кількість граней, з яких буде побудована фігура, наприклад сфера. Деталізація візуалізації сфери – це важливий момент, який дозволяє не тільки глибше вивчати деформації, але й впливає на загальну продуктивність програми.

Коли користувач збільшує значення деталізації, сфера візуалізується з більшою кількістю граней, що робить її поверхню більш гладкою та більш точно відображає кривизну. Це дозволяє з великою точністю здійснювати полікоординатні перетворення та спостерігати за детальними змінами, які відбуваються з формою об'єкта під час деформації. Приклад сфери з високим ступенем деталізації можна побачити на рисунку 4.14, де сфера містить значну кількість граней і відтворюється з високою точністю.

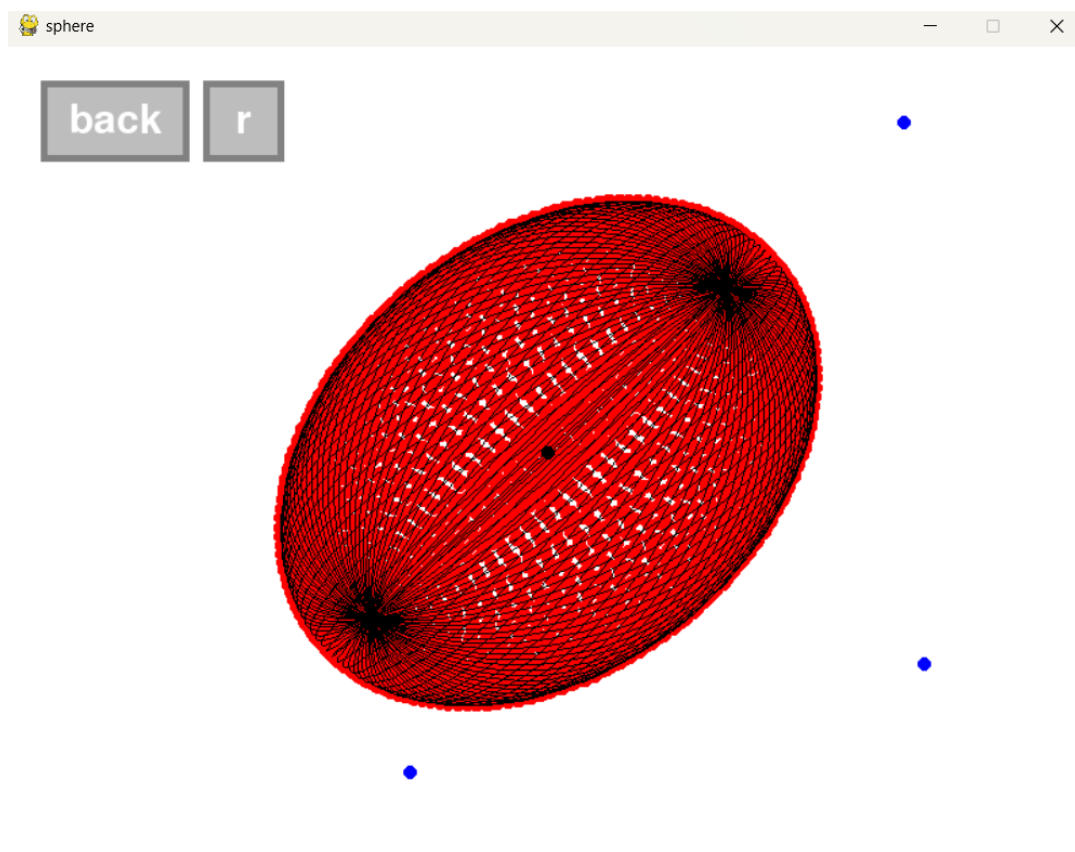


Рисунок 4.14 – Деформована куля з деталізацією в 50 граней

Проте, збільшення деталізації також призводить до вищих вимог до обчислювальної потужності комп'ютера. Якщо параметр деталізації встановлено на дуже високому рівні, можуть виникати затримки у відображенні перетворень, оскільки системі потрібно виконувати значну кількість розрахунків у реальному часі. Це може стати проблемою для комп'ютерів з обмеженими ресурсами або старішими конфігураціями.

З іншого боку, якщо кількість граней зменшити, сфера буде візуалізована з меншою кількістю деталей, що може вплинути на загальну якість зображення. На рисунку 4.15 можна побачити сферу з низькою кількістю граней, де кожна грань виглядає як окрема ламана, і загальна форма має менш гладкий вигляд.

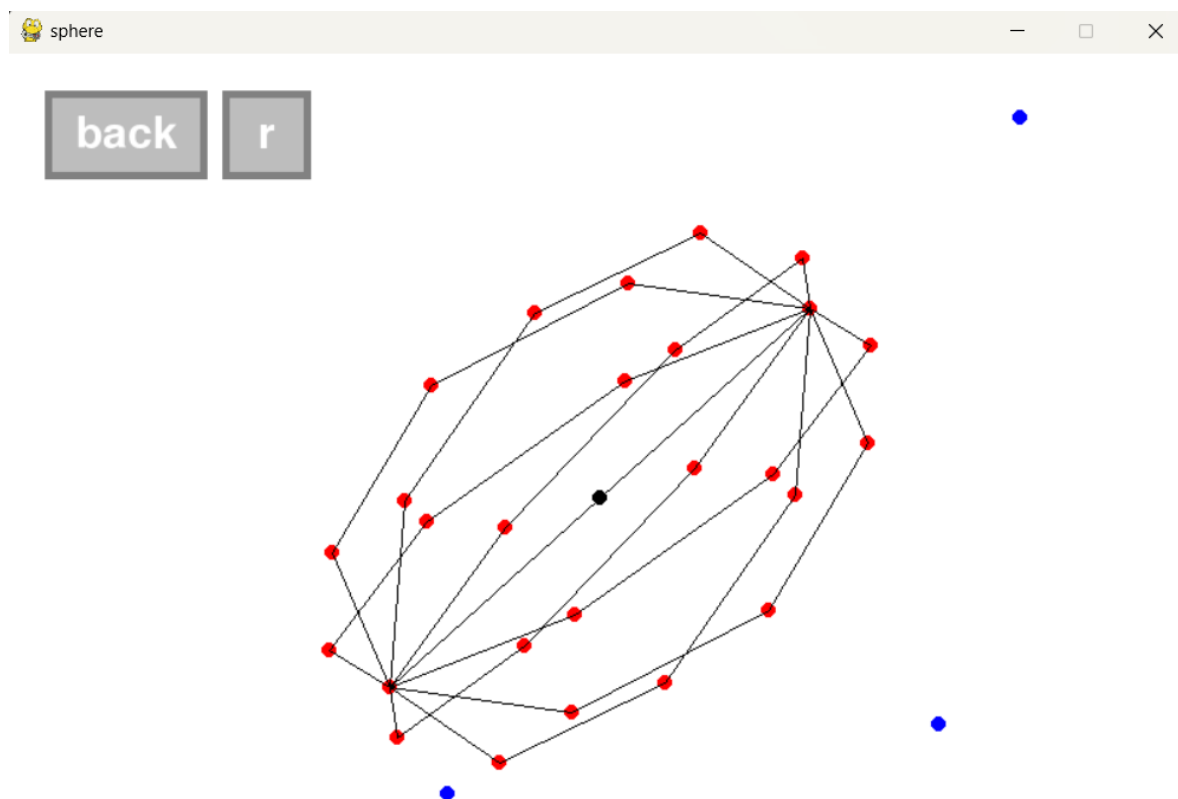


Рисунок 4.15 – Деформована куля з деталізацією в 4 грані

Хоча це знижує точність візуалізації, такий підхід також зменшує обчислювальне навантаження, дозволяючи забезпечити більш швидке відображення змін у програмі.

Коректний вибір параметра деталізації визначає оптимальне співвідношення між якістю моделі та продуктивністю комп'ютера (рисунок 4.16), дозволяючи користувачу регулювати налаштування таким чином, щоб досягти потрібної точності без непотрібних затримок.

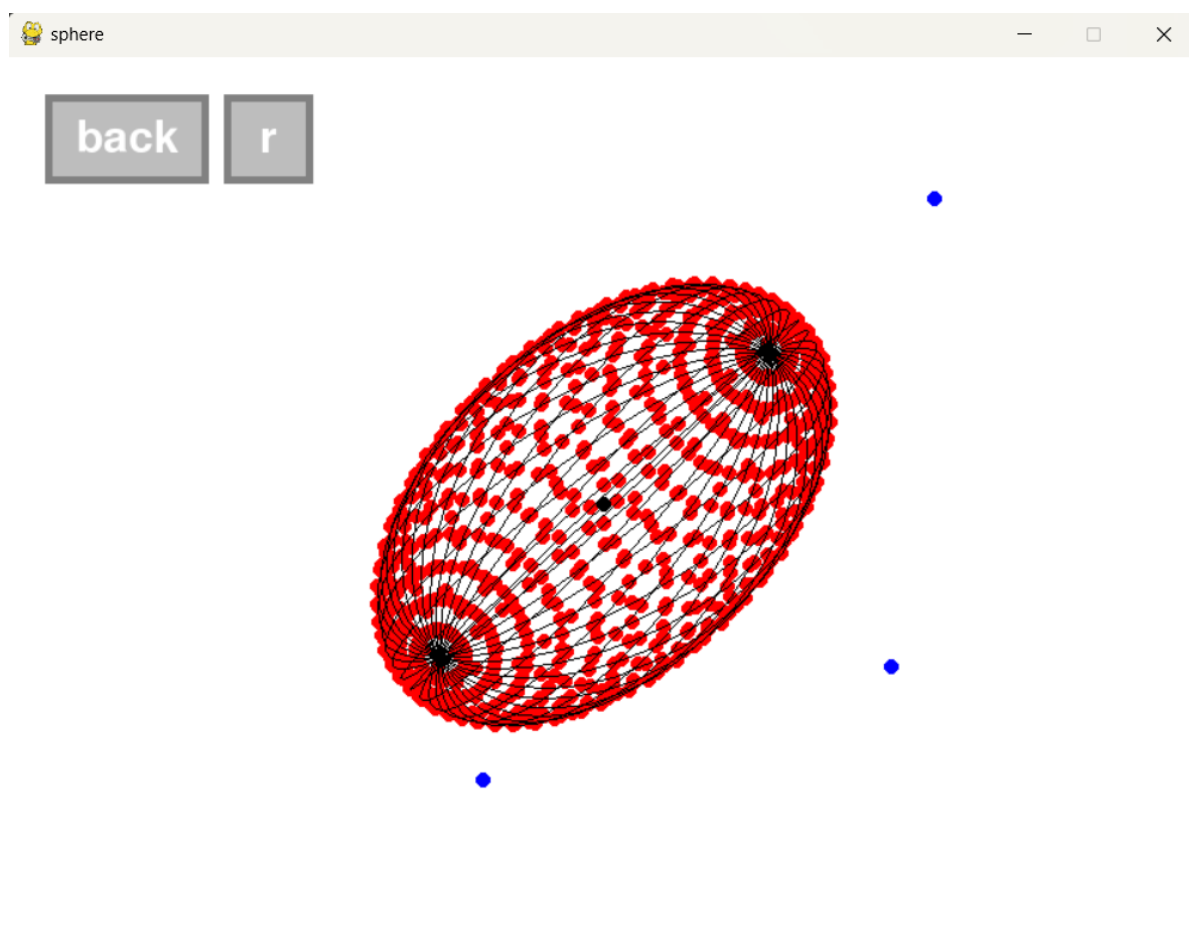


Рисунок 4.16 – Деформована куля з деталізацією в 20 граней

Завдяки цьому користувачі можуть з легкістю експериментувати з різною деталізацією та спостерігати за тим, як змінюються форми та структури об'єктів, навіть якщо їх комп'ютери мають обмежені характеристики.

4.4.4 Масштабування моделі

Однією з важливих функцій програми для візуалізації тривимірних фігур є можливість масштабування зображення, що дозволяє користувачам детально розглядати кожен аспект моделі. Ця функція реалізована через використання коліщатка миші, що є стандартним інструментом на більшості сучасних маніпуляторів. Завдяки цьому користувачі мають можливість легко приближати або віддаляти зображення фігури, просто обертаючи коліщатко миші вперед або назад.

При використанні цієї функції зображення фігури на екрані змінюється – воно стає більшим або меншим залежно від напрямку обертання коліщатка. Це особливо корисно при детальному вивченні геометрії об'єкта, а також при перевірці точності здійснюваних полікоординатних перетворень.

На рисунку 4.17, можна побачити приклад куба, зображеного на екрані з великим масштабуванням. Таке збільшене зображення дозволяє користувачам ретельно розглянути деталі фігури, такі як вершини, ребра та площини, а також забезпечує більш глибоке розуміння просторової структури моделі.

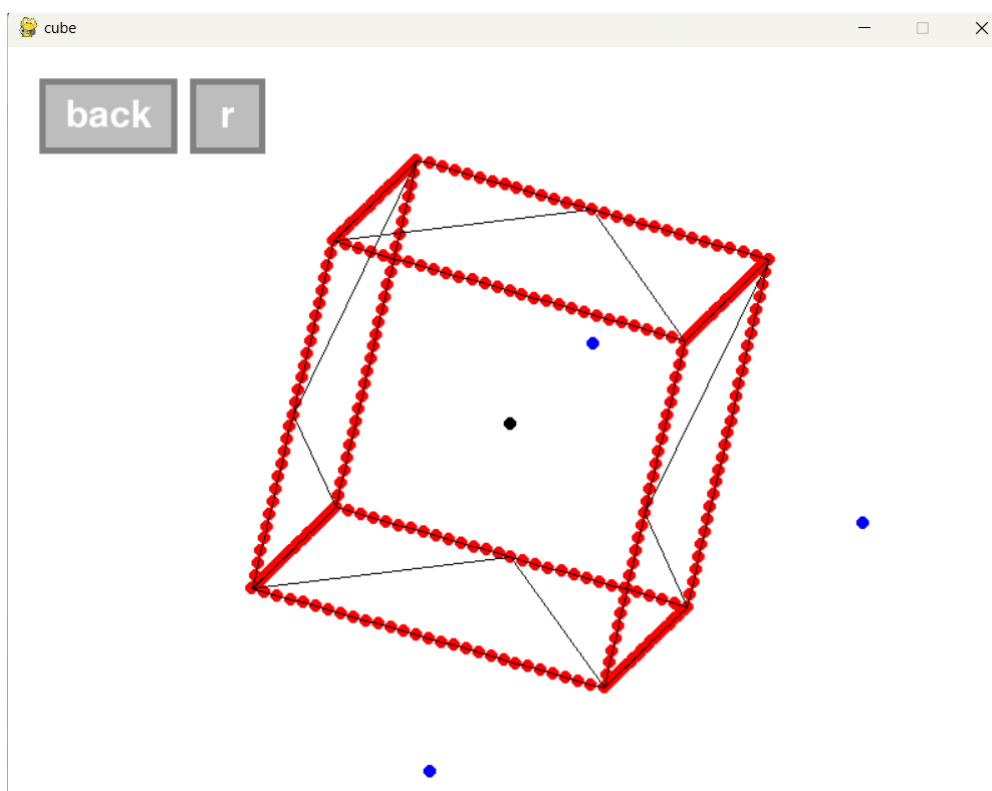


Рисунок 4.17 – Фігура куба після збільшення масштабу

На рисунку 4.18 представлено той же куб, але з значно меншим масштабуванням. Віддалене зображення корисне для отримання загального уявлення про форму та розміри фігури, та дозволяє оцінити її пропорції.

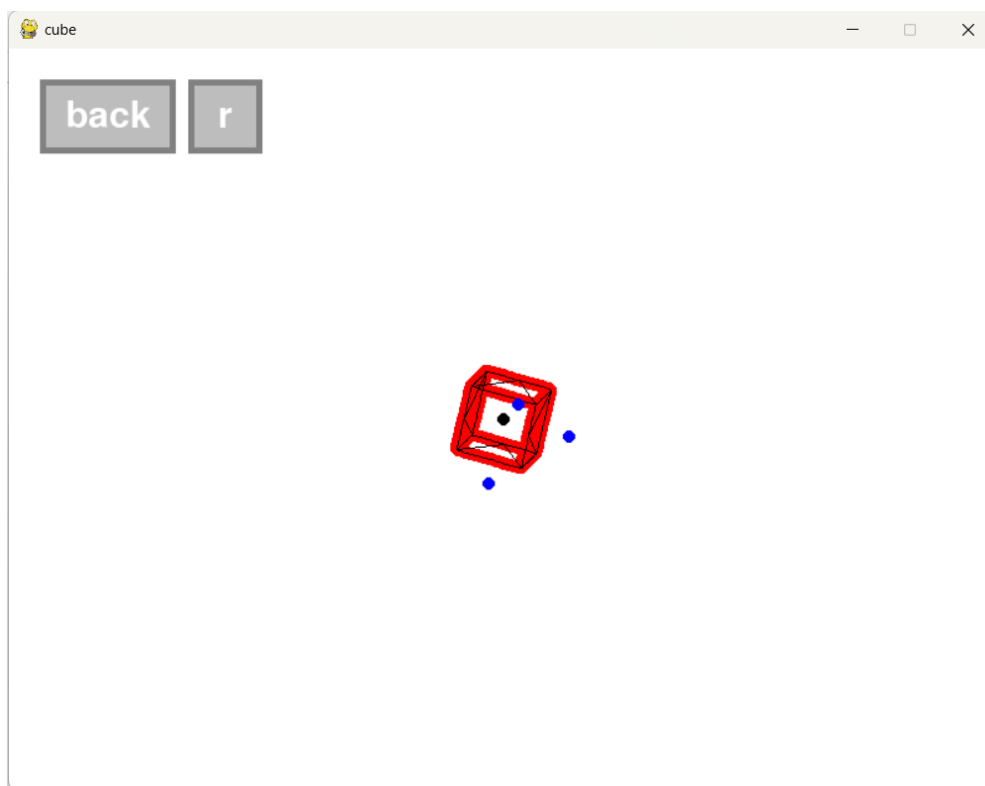


Рисунок 4.18 – Фігура куба після зменшення масштабу

Функція масштабування зображення за допомогою коліщатка миші робить програму більш гнучкою та зручною. Вона дозволяє користувачам вибрати оптимальний рівень збільшення або зменшення для конкретної задачі, забезпечуючи необхідний рівень контролю над візуальним аналізом тривимірних моделей.

4.5 Завершення роботи з програмою

Програма для візуалізації тривимірних тіл надає користувачам повний контроль над процесом роботи, включаючи можливість легкого повернення до попередніх етапів редагування та вибору налаштувань. Якщо користувач бажає деформувати іншу фігуру або змінити параметри деталізації вже обраної моделі,

він може скористатися кнопкою "Back", розташованою в лівому верхньому куті інтерфейсу (рисунок 4.19).



Рисунок 4.19 – Кнопка повернення до головного меню

Кнопка "Back" – це інтуїтивно зрозумілий механізм навігації, який забезпечує швидке повернення до головного меню програми, де користувач може знову вибрати іншу фігуру для деформації або внести зміни до налаштувань вже обраної фігури. Ця функція підвищує ефективність робочого процесу та надає додаткову гнучкість, дозволяючи користувачам експериментувати з різними фігурами без необхідності перезапуску програми чи втрати часу на повторне налаштування параметрів.

Крім того, програма реалізована з думкою про зручність завершення роботи. У будь-який момент користувач може закрити програму, натиснувши на традиційний хрестик у правому верхньому куті вікна або використовуючи клавішу "Esc" на клавіатурі. Ці стандартні методи виходу з програми є загальноприйнятими у більшості операційних систем, що робить процес інтуїтивно зрозумілим для користувачів незалежно від їхнього досвіду в роботі з комп'ютерними програмами.

Така структура управління програмою, з легкодоступними кнопками для навігації та виходу, оптимізує користувацький досвід, роблячи взаємодію з програмою більш зручною та плавною. Це важливо для підтримки робочого потоку користувача та забезпечення його здатності до швидкої реакції на зміни в процесі роботи з тривимірними моделями.

Висновки до розділу 4

У даному розділі було розглянуто процедуру роботи користувача з розробленою системою моделювання деформацій тривимірних тіл. Підкреслено значення універсальності та доступності програми завдяки підтримці різних операційних систем та забезпеченню легкої інтеграції завдяки використанню PyInstaller для створення виконуваних файлів.

Описані кроки, які користувач повинен виконати для початку роботи з програмою: від завантаження виконуваного файлу з Google Drive до його запуску. Висвітлені можливості, які надає програма для керування параметрами тривимірної фігури, зокрема, її деталізацію та деформацію. Обговорено механізми, що забезпечують зручну інтерактивність, включно з можливістю обертання фігури та зміною масштабу зображення, що дають користувачу повне уявлення про властивості та характеристики об'єкта.

Значну увагу приділено діалогу користувача з програмою під час деформації фігури за допомогою полікоординатних перетворень. Наголошено на інтуїтивно зрозумілому графічному інтерфейсі, що дозволяє користувачу легко маніпулювати контрольними точками та спостерігати за динамічними змінами форми об'єкту під впливом його втручань.

Перехід між різними режимами програми, відлагодження деталей до завершення роботи, забезпечується за допомогою лаконічних та легкодоступних інструментів навігації. Описано процес закриття програми, який є стандартним для більшості користувачів і не представляє складності.

Користувачі володіють всіма необхідними засобами для ефективного перегляду та редагування тривимірних об'єктів. Зручність користувацького інтерфейсу та відповідність програми очікуванням користувачів у кожному дії покращують процес створення моделей, роблячи дослідження деформацій полікоординатним методом більш доступними та ефективними.

5 РОЗРОБКА СТАРТАП ПРОЕКТУ

Основною ідеєю стартап-проекту є створення програмного продукту, який здатен візуалізувати та аналізувати деформації тривимірних тіл за допомогою методів полікоординатних перетворень. Ця система спрямована на обслуговування потреб у сферах, де важливим є глибоке розуміння та візуальний аналіз складних просторових структур, зокрема в інженерії, архітектурі, навчальних закладах та багатьох інших галузях.

5.1 Основні положення стартап-проекту

Для представлення основних положень стартапу було створено таблицю 5.1, підкреслюючи основні напрямки застосування розроблюваного продукту та вигоди, які отримують користувачі при використанні цього програмного забезпечення.

Таблиця 5.1 – Основні положення стартап-проекту

Завдання	Сфера застосування	Значення для споживача
Розробка програмного продукту для детальної візуалізації та аналізу деформацій тривимірних тіл на основі полікоординатних перетворень	Інженерне проектування	Інтуїтивне розуміння складності просторових структур
	Дизайн і архітектура	Зручні інструменти для моделювання та зміни форм
	Освітній процес	Наочне подання важливих понять та закономірностей
	Наукові дослідження	Поліпшене відновлення та аналіз деформованих об'єктів

Описаний стартап-проект впливає з актуальності розв'язання сучасних завдань тривимірного моделювання та візуалізації та бажанням внести інновації в цю галузь. Ринковий потенціал проекту високий, оскільки він дозволяє автоматизувати та упростити робочі процеси, які традиційно вимагають значних зусиль та фахової підготовки. Зважаючи на широке поле застосування та орієнтацію на розширення функціональних можливостей користувачів, стартап має великі шанси на успіх та зростання в майбутньому.

5.2 Огляд існуючих програмних рішень

Поточний стан сфери візуалізації тривимірних деформацій представлений широким спектром програмних засобів, кожен з яких має свої особливості, переваги та обмеження. Дослідженням були охоплені найрізноманітніші платформи і технології, від універсальних бібліотек графічного рендерингу до спеціалізованих фізичних двигунів. Розглядалися як комерційні рішення з закритим джерельним кодом, так і проекти з відкритим доступом до коду, що дозволяють користувачам вносити зміни та адаптувати функціонал під конкретні завдання.

Представляючи цей огляд, важливо зазначити, що за сучасного стану програмного забезпечення, існує значна кількість інструментів, що здатні моделювати та візуалізувати деформації в тривимірному просторі. Дослідженням були охоплені такі програмні пакети як ANSYS, ABAQUS та Comsol Multiphysics, котрі часто застосовуються в інженерних сферах для рішення складних завдань.

На основі порівняльного аналізу, основна увага була зосереджена на здатності пакетів відображати зміни геометричних форм в реальному часі, з наданням користувачеві гнучких інструментів для управління процесами деформації. Системи, які були детально проаналізовані протягом дослідження, надають користувачам потужні інтерфейси для контролю над параметрами матеріалів, зовнішнього навантаження та інших факторів, які визначають поведінку моделі під час симуляції.

Для прикладу, програмний комплекс ANSYS дозволяє реалізувати симуляції із використання функцій спектрального аналізу, допускає включення режимів моделювання нелінійних властивостей матеріалів або динамічних відповідей конструкцій. Схожим ресурсом є програма ABAQUS, яка надає більш широкі можливості для чіткого і детального моделювання процесів складного характеру завдяки використанню елементів з розширеними можливостями.

Щоб найбільше об'єктивізувати огляд, була розроблена “Таблиця 5.2” для порівняльного аналізу ключових характеристик розглянутих програмних систем, з подальшим виставленням оцінок від 0 до 5.

Таблиця 5.2 – Порівняння характеристик розглянутих систем

	ANSYS	ABAQUS	Comsol Multiphysics	SolidWorks	Autodesk Inventor
Здатність до візуалізації	4	4	5	4	4
Швидкість обчислень	4	4	3	3	3
Гнучкість налаштувань	4	4	5	4	4
Інтерфейс та зручність користування	3	3	4	4	4
Відкритість вихідного коду	0	0	0	0	0
Можливості інтеграції	4	3	4	4	4
Розширюваність/ Модульність	4	4	4	3	3

Таблиця включає наступні показники:

- здатність до візуалізації: забезпечення реалістичного відображення деформацій;
- швидкість обчислень: ефективність моделювання в тривимірному просторі за часом;
- гнучкість налаштувань: можливість змінювати матеріальні та геометричні властивості об'єктів;
- інтерфейс та зручність користування: наскільки інтуїтивно зрозумілими є налаштування і керування програмою;
- можливості інтеграції: сумісність з іншими програмами та системами;
- розширюваність/Модульність: наскільки програма підтримує створення власних модулів під специфічні завдання;

Для кращого розуміння – нижче наведений приблизний опис граничних оцінок програми.

Оцінка 0.

- не відповідає вимогам: програма зовсім не пропонує дану функціональність або можливість, або вона абсолютно неефективна;
- неможливо використовувати: програма не забезпечує мінімально прийняттого рівня продуктивності, візуалізації, налаштувань, інтеграції чи інтерфейсу користувача і тому повністю непридатна для заданої мети;
- відсутня функціональність: програма взагалі не має запитаної категорії функціональності;

Оцінка 5:

- висока якість/відмінно: програма випереджає інших завдяки своїй функціональності, забезпечуючи користувачам дуже гарну візуалізацію, швидкість обчислень, гнучкість налаштувань, інтерфейс та інтеграцію;
- найкращі в своєму класі: порівняно з конкуруючими програмами, дана програма демонструє найвищі стандарти відповідності та ефективності;
- надзвичайно ефективна: програма забезпечує найкращі можливі результати в своїй категорії і є прикладом для інших продуктів;

Базуючись на проведеному огляді існуючих програмних рішень, було складено список параметрів, які визначатимуть розроблювану програмну систему.

5.3 Вибір технічних інструментів для втілення проектної ідеї

Даний розділ зосереджується на вивченні технічних можливостей, необхідних для реалізації концепції стартапу, зорієнтованого на створення програми для детальної візуалізації деформацій тривимірних тіл за допомогою полікоординатних перетворень. Основна увага приділяється розгляду сучасних програмних рішень та інструментів, які відповідають технологічним вимогам ідеї та доступні для розробників і кінцевого використання.

Інструментальні можливості проекту представлені в таблиці 5.3, в якій описується, які технології передбачається використовувати.

Таблиця 5.3 – Обрані технічні інструменти для реалізації проекту

Технічний інструмент	Використання у проекті
Python	Мова програмування
PyCharm	Інтегроване середовище програмування
Pygame	Бібліотека для візуалізаційних операцій
NumPy	Бібліотека для наукових обчислень

Таблиця 5.3 демонструє, що обрана для проекту технічна платформа - Python - це мова програмування високого рівня, яка забезпечує широке поле для втілення розмаїтих ідей завдяки своїй гнучкості та ефективності. Python є досить популярним у наукових дослідженнях, що пояснюється його спроможністю до швидкої розробки та багатим функціоналом, а також великою кількістю доступних відкритих бібліотек.

PyCharm вже згадувався у контексті основного середовища для програмування, що виділяється потужним набором функцій для відлагодження та тестування, а також інтуїтивною організацією робочого процесу.

Pygame визначається як вдалий вибір для побудови візуальних компонентів проекту, оскільки ця бібліотека є ефективним інструментом для візуалізації і взаємодії з користувачем, надаючи змогу створювати графічний інтерфейс відносно просто та з широким спектром можливостей.

NumPy, бібліотека, чия спеціалізація полягає в реалізації ефективних наукових та математичних обчислень, включаючи роботу з багатовимірними масивами даних, здатна значно прискорити обчислення, пов'язані з деформаційними перетвореннями тривимірних об'єктів.

Враховуючи значення та відкритий доступ до зазначених технологій, стартап має всі шанси на успішну реалізацію програмного продукту, який буде корисним для широкого кола користувачів та вирізнятиметься серед інших платформ та програмних рішень у цій області.

5.4 Прогнозування етапів реалізації стартап-проекту

У вирішенні задач візуалізації деформацій у тривимірному просторі розробка системи передбачає декілька послідовних етапів, кожен з яких включає конкретні задачі:

1. Етап планування:

На цьому етапі визначаються основні цілі проекту, обсяг робіт та часові рамки. Формуються технічне завдання та проектні специфікації, а також розроблюються попередні технічні рішення.

2. Етап проектування:

На даному етапі сформовано архітектуру системи, визначено вибір технологій та рамковий огляд програмних бібліотек та інструментарію, які будуть використовуватись. Здійснено проектування графічного користувацького інтерфейсу та план реалізації алгоритмів обчислення.

3. Етап розробки:

Ця фаза включає в себе безпосередньо кодування та створення програмного продукту згідно з проектною документацією та технічними специфікаціями, розробленими на попередніх етапах.

4. Етап тестування:

Проведення ретельних тестів для перевірки стійкості, надійності та коректності роботи програми. Це включає модульні тести, інтеграційні тести, системні тести та тести користувацького прийняття.

5. Етап впровадження:

Здійснення розгортання системи в середовищі кінцевого користувача. Включає налаштування системи, міграцію даних.

Паралельно з плануванням і реалізацією необхідно здійснити оцінку можливих ризиків та розробку стратегій для їх мінімізації. Для виявлення потенційних ризиків було створено “Таблицю 5.4”, яка структурована за видами ризиків, імовірністю їх появи та потенційним впливом на проект. Поруч з оцінками ризиків були підготовані попередні заходи реагування — від методів запобігання до планів оперативного реагування.

Таблиця 5.4 – Оцінка потенційних ризиків та заходи запобігання ним

Ризик	Імовірність	Вплив	Заходи мінімізації
Технічні проблеми з обладнанням	Низька	Високий	Регулярний технічний огляд, створення резервних копій
Втрата даних через системний збій	Середня	Високий	Використання cloud-сервісів, локальна та віддалена резервна копія
Недооцінка часу на завершення проекту	Висока	Середній	Ретельне планування, чіткий розподіл трудових ресурсів
Брак досвіду в специфічних областях	Середня	Середній	Консультації з керівником, навчання та самоосвіта

Продовження таблиці 5.4.

Зміни в технічних вимогах	Низька	Середній	Регулярне оновлення вимог, адаптивне планування
Проблеми з програмними залежностями	Середня	Середній	Використання надійних і добре підтримуваних бібліотек
Брак знань в області UX/UI	Висока	Низький	Ітеративне тестування
Відсутність формального тестування	Висока	Середній	Розробка і впровадження детального плану тестування

Проаналізувавши викладену вище таблицю ризиків, було визначено, що належне управління проектом та вмiле реагування на потенційні загрози сприятимуть підвищенню стабільності і безперебійності розвитку програмної системи. Заплановані заходи реагування на ризики, їхнє старання виконання та періодичний перегляд будуть слугувати основою для підвищення якісного виконання проекту та гарантуватимуть зниження несподіваних збоїв у процесі розробки і впровадження системи візуалізації деформацій.

5.5 Аналіз потенційних клієнтів

Для ефективного просування проекту на ринку необхідно глибоко зрозуміти потреби потенційних клієнтів у системі візуалізації деформацій тривимірних тіл. Визначення та аналіз цільових аудиторій дозволяє налаштувати проект таким чином, щоб він найточніше відповідав запитам споживачів та задовольняв їх професійні вимоги.

Основні групи клієнтів, що можуть бути зацікавлені у використанні системи включають, але не обмежуються такими, як:

- дослідницькі та освітні інституції, які займаються вивченням механіки тривимірних тіл і їх властивостей;
- інженерні бюро та дизайнерські агенції, для яких важливо візуалізувати результати деформацій та вплив навантажень на різні конструктивні елементи;
- виробничі компанії, що використовують складне обладнання і зацікавлені у прогнозуванні його поведінки під час експлуатації;

Таблиця 5.5 наводиться як орієнтовний перелік потреб різних груп споживачів, що можуть бути зацікавлені у рішенні.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних цільових груп	Вимоги споживачів до товару
Навчальний процес та дослідження у сфері інженерії	Університети, науково-дослідницькі центри	Університети віддають перевагу ліцензійному ПЗ з обширною підтримкою та пакетами оновлень, в той час як науково-дослідницькі центри можуть вибирати більш спеціалізоване ПЗ	Ліцензії для багатьох користувачів, розширена підтримка, наявність навчальних матеріалів
Розробка нових інженерних рішень	Проектувальники, архітектори	Професіонали воліють ПЗ з наявністю додаткових інструментів для розширеного аналізу і продуктивної взаємодії	Висока точність моделювання, можливість інтеграції з іншими ІТ-системами, гнучкість у кастомізації

Згідно з отриманими даними, можна сформулювати ключові вимоги до розробленої системи, які включатимуть наступні технічні характеристики:

- висока швидкість рендерингу для досягнення плавності візуалізації;
- інтуїтивне керування деформаційними параметрами в реальному часі;
- можливості варіативного відображення процесів деформації. адаптивність інтерфейсу до користувацьких потреб;
- оптимізація під наявне апаратне забезпечення;
- уникнення некоректного чи хибного відтворення даних;
- забезпечення точності зображення наукових даних що візуалізуються;

Особлива увага була приділена питанням програмної взаємодії з різними інструментальними засобами та платформами. З урахуванням особливостей методів полікоординатних перетворень була наголошена необхідність створення модульної архітектури програмного забезпечення, що дозволяє вбудовувати та адаптувати специфічні алгоритми, необхідні для розв'язання задач специфічних сфер застосування цих методів.

Нарешті, при визначенні остаточних параметрів реалізації системи, було враховано потребу впровадження тестування продукту з метою виявлення потенційних проблем на ранніх етапах розробки та впевненості у стабільності та надійності роботи при регулярному використанні.

5.6 Аналіз факторів загроз та можливостей зі сторони ринку

Розглядаючи ринкову стратегію для впровадження системи візуалізації деформацій тривимірних тіл, важливо враховувати потенційні ризики, які можуть негативно вплинути на ринкове проникнення проекту, а також розглядати можливості, які сприятимуть успішній реалізації інноваційної програми. На підставі аналізу зовнішнього середовища та поточних ринкових трендів складаються таблиці факторів, що сприяють та перешкоджають успіху проекту, таблиці 5.6 та 5.7 відповідно.

Таблиця 5.6 – Фактори загроз

Фактор	Зміст загрози	Можлива реакція
Конкуренто - спроможність	Поява нових конкурентів з подібним або кращим програмним рішенням	Вдосконалення алгоритмів деформації, впровадження додаткових функцій для покращення результатів візуалізації
Технологічний прогрес	Швидке оновлення програмного забезпечення конкурентами	остійне оновлення технічної бази, автоматизація процесів, інвестування в R&D
Зниження репутації	Негативні відгуки від користувачів або професіоналів	Залучення зовнішніх експертів для оцінки продукції, вдосконалення користувацького досвіду
Економічний спад	Зменшення інвестицій у новітні технології з боку потенційних клієнтів	Розробка стратегій гнучких ціноутворень, розширення географії продажів, пошук альтернативних ринків

Таблиця 5.7 – Фактори можливостей

Фактор	Зміст можливостей	Можлива реакція
Ринковий попит	Зростання ринкової потреби в новітніх рішеннях візуалізації	Оптимізація програми під потреби ринку, розробка більш ефективних перетворень
Технологічні інновації	Поява нових технологій, що відкривають ширші можливості для візуалізації	Інтеграція нових технологій у програмний продукт, надання більш широких сервісів
Розбудова репутації	Визнання на ринку як новатор у сфері 3D візуалізацій	Активний маркетинг, участь у професійних конференціях, публікація результатів у наукових журналах

На підставі отриманих даних буде розроблено стратегію ринкового впровадження продукту, враховуючи при цьому інтереси та поведінкові особливості потенційних клієнтів. Програма візуалізації деформацій тривимірних тіл має великий потенціал для задоволення потреб ключових ринкових ніш, таких як важка промисловість, проектування та навчальні заклади, де високо цінуються точність моделювання і гнучкість візуалізаційних інструментів.

5.7 Розроблення стратегії ринкового впровадження проекту

Для створення ефективної ринкової стратегії проекту візуалізації деформації тривимірних тіл необхідно визначити основні цільові групи потенційних споживачів. Це дозволить зосередитись на вимогах конкретних користувачів та розробити стратегію, яка найточніше відповідатиме їхнім потребам.

Таблиця 5.8 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит	Інтенсивність конкуренції	Простота входу
Університети, науково-дослідні лабораторії	Висока	Високий	Низька	Легко
Інженерні компанії, будівельні фірми	Висока	Середній	Помірна	Помірно
Виробничі підприємства, заводи	Середня	Середній	Середня	Помірно

На основі цього аналізу було зроблено висновок про високі шанси прийняття розробленого продукту обома цільовими групами, при цьому для кожної з них спостерігався високий рівень готовності до впровадження нового продукту та сприйняття його функціональності. Це створює сприятливі умови для активного розгортання діяльності на ринку.

Виходячи з розуміння поведінки цих груп споживачів, була розроблена базова стратегія розвитку, таблиця 5.9.

Таблиця 5.9 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку
Інженерні компанії, будівельні фірми	Стратегія концентрованого маркетингу	Надійність, точність, висока якість деформацій, сумісність з іншими інженерними програмами	Стратегія диференціації
Університети, науково-дослідні лабораторії	Стратегія масового маркетингу	Цінова доступність, наявність навчальних матеріалів, легкість у використанні	Стратегія проникнення на ринок

У таблиці 5.9 було описано базову стратегію розвитку для вибраних сегментів ринку, яка концентрується на стратегії концентрованого маркетингу, розробці надійних та точних рішень для візуалізації деформацій тіл.

Далі була сформована стратегія конкурентної поведінки, що відображена у таблиці 5.10. Вона базується на основі дослідження позиції компанії на ринку, потреби в пошуку нових споживачів, а також оцінки конкурентних переваг.

Таблиця 5.10 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
---	---	--	---

Продовження таблиці 5.10.

Ні	Обидва варіанти	Покращення функціональності, впровадження унікальних властивостей	Стратегія диференціації
----	-----------------	---	----------------------------

Аналізуючи дані таблиці 5.10, можна зробити висновок про те, що стратегія проекту передбачає активну позицію на ринку. Важливо відмітити, що проект не позиціонує себе як «першопрохідець» в галузі, що може свідчити про наявність вже існуючих продуктів із схожим функціоналом. Це ставить питання про необхідність максимально чіткого вироблення унікальної торговельної пропозиції і знаходження незайнятої ринкової ніші.

Із таблиці також видно, що планується робота як із новими споживачами, так і з тими, котрі вже користуються продуктами конкурентів. Це вимагає від компанії гнучкого підходу до формування ринкового меседжу та адаптації пропозицій до потреб обох груп користувачів.

Завершальним етапом є розроблення стратегії позиціонування, яка визначає як потенційні споживачі будуть ідентифікувати продукт та які асоціації повинні сформуватися, що візуалізує собою таблиця 5.11.

Таблиця 5.11 – Визначення стратегії позиціонування

Вимоги до товару	Базова стратегія розвитку	Ключові конкурентоспроможні позиції	Вибір асоціацій із проектом
Легке застосування, інтерактивність, широкі аналітичні можливості	Стратегія спеціалізації	Інтуїтивний інтерфейс, гнучкість налаштувань, точність відображення з деформаціями	Інноваційність, ефективність, надійність

Вивчивши ринкове середовище та розглянувши потенційні можливості та ризики, можна прийти до висновку, що ринкова стратегія для проекту візуалізації деформацій тривимірних тіл має бути зосереджена на диференціації продукту та проникненні в специфічні сегменти ринку.

Основна увага при цьому має бути приділена розвитку унікальних характеристик та переваг, які допоможуть виділити продукт серед конкурентів, а також забезпечать залучення нових споживачів, що вимагають високої точності й якості візуалізації.

Ринкова стратегія, розроблена на основі цих даних, передбачає концентрований підхід до маркетингу з акцентом на спеціалізацію продукту, що спирається на покращену функціональність та якість в порівнянні з іншими інструментами на ринку.

Розробка комплексної позиції та вибір стратегічних напрямків, що охоплюють зручність, швидкість і адаптивність, є базовими для успішного ведення проекту на ринок та його подальшого розвитку.

ВИСНОВКИ

У ході виконання даної магістерської роботи було здійснено розробку та аналіз моделі для візуалізації деформацій тривимірних тіл з використанням полікоординатних перетворень. Дослідження дало змогу узагальнити існуючі методики та відкрити нові шляхи для їх практичного застосування у різноманітних сферах.

1. Проведено аналіз сучасного стану та існуючих методів візуалізації деформацій, що підтвердив актуальність розробки нових підходів та методологій у цій галузі.
2. Досліджено та проаналізовано ключові принципи полікоординатних перетворень, які лягли в основу створеної моделі візуалізації, та продемонстровано їх переваги у гнучкості та точності моделювання.
3. Розроблено алгоритмічну базу для комп'ютеризованого застосування полікоординатних перетворень, що дозволяє з високою точністю відтворювати деформації тривимірних тіл.
4. Виконано оптимізацію та валідацію моделі на основі тестувань, які засвідчили ефективність запропонованого підходу та потенціал його застосування у практичних сценаріях.
5. Надано рекомендації щодо використання розробленої моделі візуалізації в практичній діяльності, що забезпечує основу для її ефективного застосування у різних галузях, де актуальними є точність, динамічність та адаптивність візуального відображення.
6. Розроблено стартап-проект на основі розробленої системи візуалізацій деформації тривимірних тіл полікоординатними методами.

В результаті проведеного дослідження було отримано цінні теоретичні знання та практичні навички, які підкреслюють важливість інтеграції полікоординатних перетворень у системи візуалізації. Розроблена модель є гнучкою та масштабованою, що дозволяє її адаптувати до специфічних потреб різноманітних дослідницьких та комерційних проектів.

Перспективи подальшого розвитку за тематикою дослідження включають розширення функціоналу програми для роботи з більш складними формами, інтеграцію з іншими програмними платформами та інструментами аналізу даних, а також поліпшення користувацького інтерфейсу для забезпечення ще більшої інтуїтивності та зручності використання.

Значний науковий інтерес також представляє подальше дослідження можливостей застосування полікоординатних перетворень у віртуальній та доповненій реальності, робототехніці, а також в галузях, пов'язаних з 3D-друком та прототипуванням.

Таким чином, отримані в результаті дослідження теоретичні і практичні результати відкривають нові можливості для розширення обсягу знання і практичного застосування сучасних методів візуалізації деформацій тривимірних тіл.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сидоренко Ю.В., Водопшин В.В. Система деформаційного моделювання тривимірних тіл на основі полікоординатних перетворень/ Сталій розвиток — XXI століття. Дискусії 2023: колективна монографія / Національний університет “Києво-Могилянська академія” / за ред. проф. Хлобистова Є.В. — Київ, 2023. — Електронне видання.
2. Залевська О.В., Яблонський П.М., Сидоренко Ю.В., Мірошніченко І.В., Ситник А.Ю. Удосконалення методу та алгоритму фрактального стиснення графічного зображення. *Прикладна геометрія та інженерна графіка*. 2021. Вип.100. С.118-125.
3. Залевська О.В., Яблонський П.Н., Фіногенов О.Д., Сидоренко Ю.В., Ситник А.Ю. Реалізація алгоритму фрактального стиснення графічного зображення. *Сучасні проблеми моделювання*. 2021. Вип.22. С.8-11.
4. Сидоренко Ю.В. Підвищення точності алгоритму політочкових перетворень / Ю.В. Сидоренко, О.В. Залевська // *Прикладна геометрія та інженерна графіка* — К.:КДТУБА, 2020, вип.97 — С.129-135.
5. Сидоренко Ю.В. Геометричне моделювання складних об’єктів на основі політочкових відображень відрізків прямих/ Ю.В.Сидоренко, Ю. І. Бадаєв // *Сучасні проблеми моделювання: зб. наук. праць.* - Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2019. -Вип.16.-С.17-24.
6. Сидоренко, Ю., Залевська, О., Городецький, М., & Спірінцев, Д. Аналіз поведінки похибки обчислень при Гаус-інтерполяції функції Рунге. *Сучасні проблеми моделювання*, 2022, 159-167.
7. Andrew Hunt, David Thomas. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley Professional; 1st edition, 1999.
8. Anna Ravenscroft, Alex Martelli. *Python in a Nutshell*. O'Reilly Media; 3rd edition, 2017.
9. Barry Warsaw, Jeremy Hylton, Fred L. Drake Jr. *Python Library Reference Open Source Library*. PythonLabs, 2001.
10. Bradford Cross. *Machine Learning with Python: Deep Insights into Machine*

Learning. Packt; 1st edition, 2018.

11. Bruce Eckel. Thinking in Python. Prentice Hall, 2002.
12. Dan Bader. Python Tricks: A Buffet of Awesome Python Features. Dan Bader; 1st edition, 2017.
13. David Beazley, Guido Van Rossum. Python: Essential Reference. New Riders Publishing, 1999.
14. David Beazley. Python Essential Reference. Addison-Wesley Professional; 4th edition, 2009.
15. Eli Bressert. Scipy And NumPy: An Overview for Developers Paperback. O'Reilly Media; 1st edition, 2012.
16. Eric Matthers. Python Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to Programming. No Starch Press; 2nd edition, 2019.
17. Florian Dedov. The Python Bible 7 in 1: Volumes One To Seven. Independently published, 2020.
18. Francois Chollet. Deep Learning with Python, Second Edition. Manning; 2nd edition, 2021.
19. Ivan Idris. Instant Pygame for Python Game Development How-to. Packt, 2013.
20. Ivan Idris. NumPy CookBook. Packt, 2012.
21. Jake VanderPlas. Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media; 1st edition, 2016.
22. John V. Guttag. Introduction to Computation and Programming Using Python: With Application to Understanding Data. MIT Press; 2nd edition, 2016.
23. Justin Seitz, Tim Arnold. Black Hat Python, 2nd Edition: Python Programming for Hackers and Pentesters. No Starch Press; 2nd edition, 2021.
24. Laura Cassell, Alan Gauld. Python Projects: Practical Programming for Beginners. Wiley; 1st edition, 2014.
25. Luciano Ramalho. Fluent Python: Clear, Concise, and Effective Programming. O'Reilly Media; 1st edition, 2015.
26. Mark Lutz. Learning Python, 5th Edition. O'Reilly Media; Fifth edition, 2013.
27. Mark Pilgrim. Dive Into Python 3. Apress; 1st edition, 2009.
28. Mark Summerfield. Python 3: The Comprehensive Guide to Building Applications.

Apress; 1st edition, 2009.

29. Matt Harrison. *Effective PyCharm: Learn the PyCharm IDE with a Hands-on Approach (Treading on Python)*. Independently published, 2019.
30. Michael Dawson. *Python Programming for the Absolute Beginner*. Course Technology PTR; 3rd edition, 2010.
31. Noah Gift, Kennedy Behrman. *Python for DevOps: Learn Ruthlessly Effective Automation*. O'Reilly Media; 1st edition, 2020.
32. Paul Barry. *Head First Python: A Brain-Friendly Guide*. O'Reilly Media; 2nd edition, 2016.
33. Pedro Kroger. *Modern Python Development With PyCharm*. Independently published, 2015.
34. Roberto Ulloa. *Kivy: Interactive Applications in Python*. Packt Publishing, 2013.
35. Sean McManus. *Mission Python: Code a Space Adventure Game*. No Starch Press, 2018.
36. Sloan Kelly. *Python, PyGame and Raspberry Pi Game Development*. Apress; 1st edition, 2002.
37. Stef Maruch, Aahz Maruch. *Python For Dummies*. For Dummies; 1st edition, 2006.
38. Tim Peters, Antoon Pardon. *Python Programming Language*. Network Theory Ltd; 1st edition, 2001.
39. Tony Gaddis. *Starting Out with Python*. Pearson; 4th edition, 2017.
40. Tony Holdroyd. *Essential Python: Learn to Code with Python*. Independently published, 2019.
41. Will McGugan. *Beginning Game Development with Python and Pygame: From Novice to Professional*. Apress; 1st edition, 2007.
42. Will McGugan, Harrison Kinsley. *Beginning Python Games Development, Second Edition: With PyGame*. Apress; 2nd ed. edition, 2015.
43. Zed Shaw. *Learn Python 3 the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code*. Addison-Wesley Professional; 1st edition, 2017.

ДОДАТОК А. ТЕЗИ НА НАУКОВО-ПРАКТИЧНУ КОНФЕРЕНЦІЮ

Національний університет "Києво-Могилянська академія"



СТАЛИЙ РОЗВИТОК — ХХІ СТОЛІТТЯ

Дискусії 2023

Колективна монографія

Київ, Україна 2023

СИСТЕМА ДЕФОРМАЦІЙНОГО МОДЕЛЮВАННЯ ТРИВИМІРНИХ ТІЛ НА ОСНОВІ ПОЛІКООРДИНАТНИХ ПЕРЕТВОРЕНЬ

Сидоренко Юлія Всеволодівна

Кандидат технічних наук, доцент

Україна, м. Київ, Національний технічний університет України “КПІ ім. Ігоря
Сікорського”, (проспект Перемоги, 37, Київ, 03056)

Доцент

Конт. тел.: 097-954-16-32

Адреса для висилання матеріалів конференції:

Сидоренко Ю.В., вул. Машинобудівна, буд. 27, кв.72, м. Київ, 03067, Україна

E-mail: suliko3@ukr.net

Водопшин Володимир Володимирович

Студент групи ТР-21мп

Україна, м. Київ, Національний технічний університет України “КПІ ім. Ігоря
Сікорського (проспект Перемоги, 37, Київ, 03056)

Студент

Конт. тел.: 097-198-90-73

Адреса для висилання матеріалів конференції:

Водопшин.В.В., вул. Героїв Дніпа, буд. 42, кв.180,

м. Горішні Плавні, 39800, Україна

E-mail: vodopshyn.v@gmail.com

З незворотнім розвитком комп'ютеризації дедалі більше сфер діяльності починають використовувати можливості, які їм готові надати інформаційні технології. Чи не найбільш важливою є тема моделювання деформацій об'єктів, яка є однією з найбільш розповсюджених у використанні. Наразі моделювання об'єктів широко використовується в багатьох сферах, починаючи від дизайну і закінчуючи машинобудуванням та хірургією [1].

Одночасно із широким використанням моделювання, виникає питання про вибір відповідного йому методу. Найбільш підходящими є методи деформаційного моделювання, адже вони застосовують деформаційні зміни до простору, що в свою чергу сприяє адекватному перетворенню об'єкта [2]. Яскравим представником даного класу моделей є метод полікоординатних перетворень. Даний метод дозволяє реалізувати моделювання тривимірних об'єктів. Для цього необхідно представити об'єкт у вигляді множини площин та послідовно провести полікоординатні перетворення для кожної з них [3].

У даній статті приводиться описання розробленої системи моделювання тривимірних тіл та перетворення їх полікоординатним методом, а також надаються результати роботи програмного продукту.

У зв'язку з широкою областю використання методу деформаційного моделювання існує реальна потреба в програмному забезпеченні, що дозволяє візуалізувати перетворення в режимі реального часу. Тому було вирішено провести

дослідження даного методу, результатом якої стала система візуалізації полікоординатних перетворень тривимірних тіл.

Було створено програмне забезпечення, що відображає на екрані процеси перетворення кулі, куба та шестикутної призми методом полікоординатних перетворень, з можливістю проведення деформацій на основі дій користувача, з можливістю обертати тіла в будь-якому напрямку, наближати та віддаляти фігуру та змінювати налаштування побудови тіл.

Основною задачею, яку має виконувати програма, є надання користувачу можливості деформувати тіло шляхом зміни положення точок базису, не впливаючи на сам об'єкт, а тільки на простір, у якому він знаходиться.

Вхідною інформацією системи є: обрана користувачем фігура, дані про налаштування тіла, положення точок базису.

Результатом роботи системи є візуальне зображення об'єкту та оновлене положення точок базису.

На рисунку 1 представлена структурна схема програми.

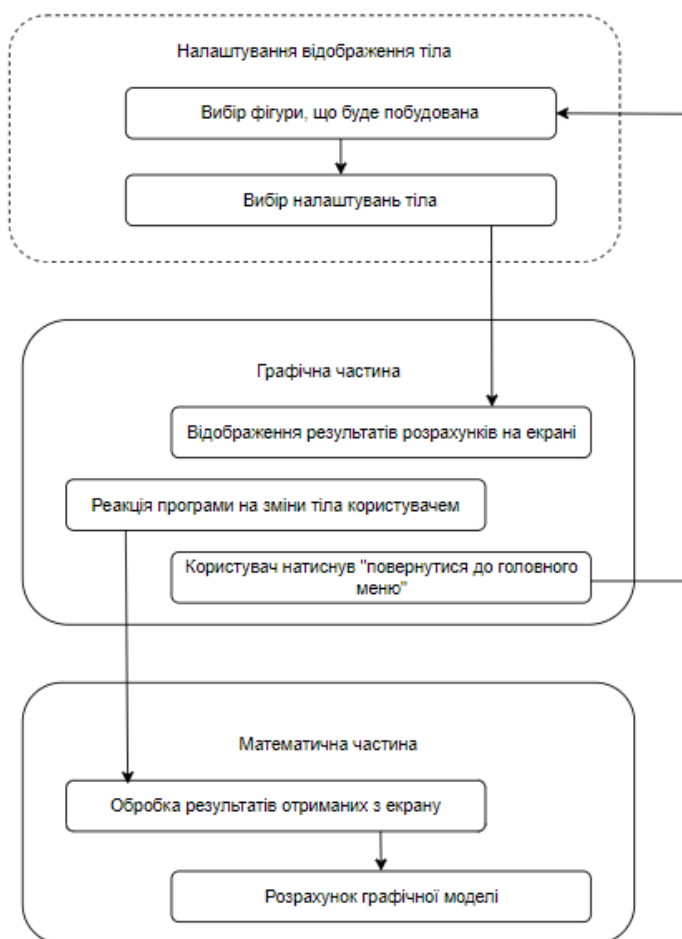


Рис. 1 - Структурна схема програми

Створена система складається з трьох окремих структурних частин, а саме: модулю налаштування відображення тіла, графічного та математичного модулів. Кожна з яких реалізована в вигляді одного виконуваного файлу. Перед початком

роботи користувачу необхідно завантажити програму на свій комп'ютер. Після чого користувачу достатньо запустити завантажений виконуваний файл.

Програма була реалізована на мові програмування Python, та з використанням зовнішніх модулів PyGame та NumPy, після чого була зібрана у виконуваний файл за допомогою утиліти PyInstaller, що дозволило використовувати розроблену систему навіть без встановленого інтерпретатора мови python, та всіх зовнішніх модулів, що необхідні для роботи, адже всі вони були включені до .exe файлу на стадії збірки. Єдиним мінусом використання PyInstaller є неможливість зібрати виконуваний файл, універсальний під всі операційні системи, тому користувач має завантажити саме той .exe файл, що відповідає його операційній системі. За відповідність файлу до системи відповідають перші три літери в назві (рисунок 2). Win для Windows, lin та mac – для Linux та MacOS відповідно.



Рис. 2 – Перелік програм доступних до завантаження залежно від ОС

Після запуску програми користувач має змогу обрати фігуру (рисунок 3) яку він бажає побудувати, а саме – куб, куля або шестикутна призма.



Рис. 3 – Меню з вибором фігури

Після чого йому необхідно ввести значення, що відповідає деталізації побудови фігури (рисунок 4). Чим більше значення, тим більш детальною буде фігура, але це потребує використання більших ресурсів комп'ютера, тому за замовченням користувачу пропонується значення «10», що є достатнім рівнем деталізації, при цьому не будуть спостерігатися затримки навіть на комп'ютерах з помірною потужністю.



Рис. 4 – Поле для значення деталізації фігури

Після усіх цих дій, на екрані з'явиться побудована фігура, що була обрана користувачем. В даному прикладі – шестикутна призма (рисунок 5).

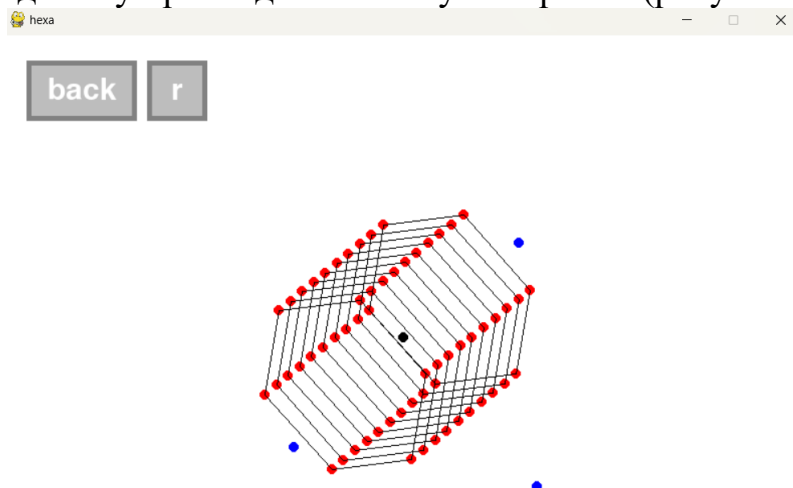


Рис. 5 – Поле для значення деталізації фігури

Окрім фігури на екрані знаходяться точки базису (зображені синім кольором), та кнопки «back» та «r», що дозволяються повернутися до меню вибору фігури (рисунок 3) та увімкнути/вимкнути автоматичне обертання фігури відповідно. Користувач має змогу переміщувати точки базису, навівши курсор на обрану точку та затиснувши ліву кнопку миші. В результаті чого тіло зазнає відповідних деформацій (рисунок 6).

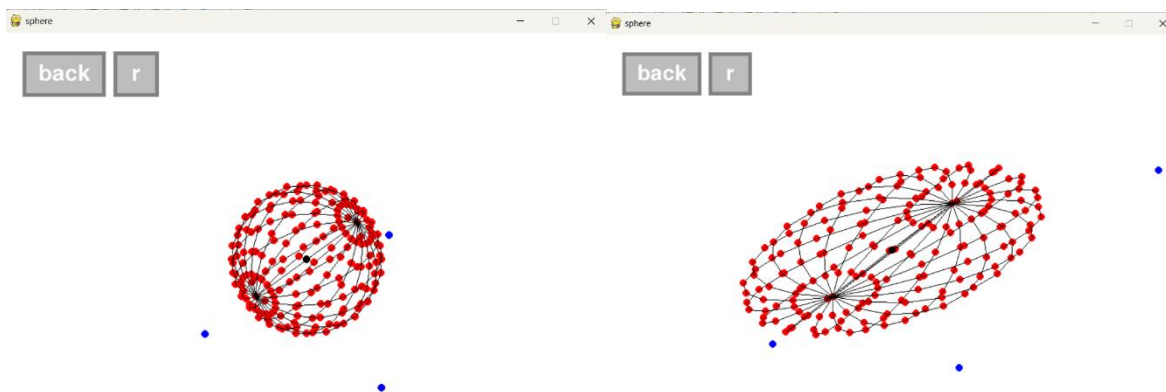


Рис. 6. Куля до та після переміщення точок базису

Окрім цього користувач має змогу обертати тіло навколо будь – якої з осей та збільшувати/зменшувати зображення на екрані (рисунок 7).

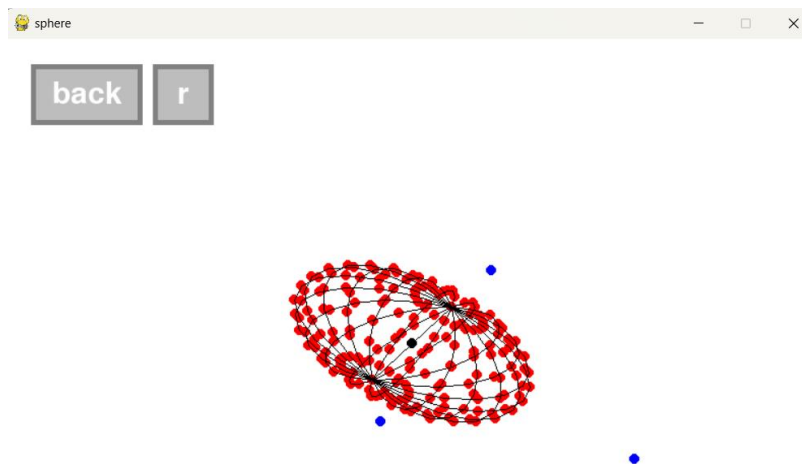


Рис. 7. Зменшенна куля після деформації та обертів.

Висновки. Система деформаційного моделювання тривимірних тіл на основі полікоординатних перетворень надає доступний та простий в використанні спосіб проведення полікоординатних перетворень до реалізованих в системі тіл.

Алгоритм полікоординатних перетворень, який є внутрішньою частиною системи, вважається одним із передових напрямків розвитку практичної геометрії. А сама система має потужний потенціал у модифікуванні, адже для розширення бази тривимірних тіл для побудови та перетворення, достатньо реалізувати побудову необхідної фігури, що в майбутньому дозволить проводити моделювання та деформації більш складних тіл.

Література:

1. Сидоренко Ю.В., Шалденко О.В. Вагові політочкові перетворення при моделюванні можливих результатів пластичної хірургії. Сучасні проблеми моделювання. 2019. – Вип.15. – С. 151-161.
2. Sidorenko Yu, Kryvda O., Leshchynska I. System of modeling of structural elements of ventilation systems by polycoordinate transformations. Strength of Materials and Theory of Structures. 2020. No. 104. P. 221-228
3. Сидоренко Ю.В., Бадаєв Ю.І. Геометричне моделювання складних об'єктів на основі політочкових відображень відрізків прямих. Сучасні проблеми моделювання. 2019. -Вип.16.-С.17-24.