

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«__» _____ 2023 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «Метод афінних перетворень в комп'ютерних системах обробки
графічної інформації»

Захищено на засіданні
Атестаційної комісії № 40
протокол № __ від __.06.2023 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ Скоб Ю. О.

(підпис)

(прізвище та ініціали)

Виконав:
студент 4 курсу, групи КІ– 41
Галузь знань: 12 – Інформаційні
технології
Спеціальність: 123 – «Комп'ютерна
інженерія»
Веремієнко Богдан Олександрович

(підпис)

Керівник:
Чуб Ольга Ігорівна

(підпис)

Рецензент:
Фролов В'ячеслав Вікторович

(підпис)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, чотирьох розділів, висновків, списку використаних джерел. Загальний обсяг роботи складає 64 сторінки, із яких 41 сторінка основної частини містить 16 рисунків, 8 формул та 25 найменувань за списком використаних джерел.

Метою даної кваліфікаційної роботи є дослідження методу афінних перетворень в комп'ютерних системах обробки графічної інформації. Зокрема, розглянуто математичні основи, властивості та застосування методу.

Об'єктом дослідження є дослідження методу афінних перетворень та розуміння його математичних основ, аналізу його використання в різних аспектах обробки графічної інформації та розробки програмної реалізації для практичного використання.

Предметом дослідження є метод афінних перетворень в контексті комп'ютерних систем обробки графічної інформації. Досліджуються принципи, властивості та практичне застосування цього методу для трансформації 2D та 3D об'єктів.

Отримані результати є новизною і важливим внеском у галузь комп'ютерної графіки та обробки графічної інформації. Розроблені рішення можуть бути використані для створення реалістичних графічних зображень та візуалізації даних.

Використання результатів роботи може сприяти покращенню функціональності і візуального вигляду програмних продуктів в комп'ютерних системах обробки графічної інформації.

Ключові слова: афінні перетворення, комп'ютерні системи, графічна інформація, програмне рішення, графічний інтерфейс користувача, 2D та 3D моделі, Java, IntelliJ IDEA.

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, four chapters, conclusions, and a list of references. The total volume of the work is 64 pages, of which 41 pages of the main part contain 16 figures, 8 formulas and 25 references.

The purpose of this qualification work is a detailed study of the method of affine transformations in computer systems for processing graphic information. In particular, the mathematical foundations, properties and applications of affine transformations are considered.

The object of research is to study the method of affine transformations and understand its mathematical foundations, analyze its use in various aspects of graphic information processing and develop a software implementation for practical use.

The subject of the study is the method of affine transformations in the context of computer systems for processing graphic information. The principles, properties, and practical application of this method for transforming 2D and 3D objects are studied.

The results obtained are novel and an important contribution to the field of computer graphics and graphic information processing. The developed solutions can be used to create realistic graphic images and data visualization.

The use of the results of the work can help to improve the functionality and visual appearance of software products in computer systems for processing graphic information.

Keywords: affine transformations, computer systems, graphical information, software solution, graphical user interface, 2D and 3D models, Java, IntelliJ IDEA.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. МАТЕМАТИЧНІ ОСНОВИ АФІННИХ ПЕРЕТВОРЕНЬ.....	9
1.1 Поняття і властивості	9
1.2 Ключові аспекти методу	10
1.3 Особливості методу на площині та у просторі	14
Висновки за розділом 1	16
РОЗДІЛ 2. ВИКОРИСТАННЯ МЕТОДУ У КОМП'ЮТЕРНИХ	18
СИСТЕМАХ	18
2.1 Трансформація 2D об'єктів та 3D моделей	18
2.2 Зображення та текстури	20
2.3 Проекції та перспектива.....	20
2.4 Практичне застосування методу.....	22
Висновки за розділом 2	25
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ	27
3.1. Аналіз існуючих програмних рішень	27
3.2. Вибір інструментів розробки	30
3.3. Реалізація функціоналу	31
3.4. Інтерфейс	33
Висновки за розділом 3	36
РОЗДІЛ 4. РОБОТА КОРИСТУВАЧА З ПРОГРАМОЮ	38
4.1. Системні вимоги.....	38
4.2. Встановлення і запуск додатку.....	39
Висновки за розділом 4.....	45
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	52

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

- АП – афінні перетворення;
- 2D – двовимірний простір;
- 3D – тривимірний простір;
- JDK – комплект розробки Java (Java Development Kit);
- IDE – інтегроване середовище розробки (Integrated Development Environment);
- AWT – Abstract Window Toolkit, бібліотека для створення графічних інтерфейсів у Java;
- SWING – набір компонентів та класів для створення графічних інтерфейсів у Java.

ВСТУП

У сучасному світі комп'ютерна обробка графічної інформації є необхідною складовою багатьох сфер діяльності, включаючи веб-дизайн, комп'ютерну графіку, віртуальну реальність, медичне моделювання та багато інших. Одним із ключових елементів, що використовується в цих системах, є метод афінних перетворень. Він дозволяє змінювати розташування, форму та орієнтацію графічних об'єктів, що відіграє важливу роль у створенні реалістичних та естетично привабливих візуальних ефектів.

Актуальність даної теми обумовлена постійним розвитком комп'ютерних систем обробки графічної інформації та необхідністю ефективного управління та трансформації графічних об'єктів. Використання афінних перетворень дозволяє реалізувати широкий спектр графічних ефектів, від простого масштабування та обертання до складних перспективних деформацій. Дослідження цього методу сприятиме поліпшенню розуміння принципів комп'ютерної графіки та забезпечить основу для подальшого розвитку і оптимізації графічних програм.

Об'єктом нашого дослідження є метод афінних перетворень в комп'ютерних системах обробки графічної інформації. Ми будемо аналізувати та вивчати його математичні основи, властивості, а також способи його практичного застосування у графічних програмах та системах.

РОЗДІЛ 1

МАТЕМАТИЧНІ ОСНОВИ АФІННИХ ПЕРЕТВОРЕНЬ

1.1 Поняття і властивості

Афінні перетворення є важливими математичними концепціями в області комп'ютерних систем обробки графічної інформації. Вони використовуються для зміни кута нахилу, розташування, масштабування та орієнтації об'єктів у двовимірному просторі та тривимірному просторі. Ці перетворення дозволяють створювати різноманітні ефекти та впливати на вигляд та положення об'єктів, що є важливими аспектами роботи з графічною інформацією. У цьому підрозділі ми розглянемо поняття афінних перетворень, а також їх властивості, які роблять їх потужним інструментом для обробки графічної інформації.

Афінне перетворення - це геометрична трансформація, яка зберігає прямі лінії та паралельність. Воно включає зсув, масштабування, обертання та зеркальне відображення [1]. Афінні перетворення можуть бути виконані за допомогою математичних формул або матричних операцій.

Властивості афінних перетворень [2]:

- **Лінійність:** Афінні перетворення є лінійними, що означає, що вони зберігають лінійну комбінацію точок. Нехай A та B - дві точки у просторі, λ та μ - дійсні числа. Тоді афінне перетворення T виконує умову (1.1) - перетворені точки.
- **Збереження паралельності:** Афінне перетворення зберігає паралельність ліній. Якщо дві прямі лінії у вихідній системі координат є паралельними, то їх відображення за допомогою афінного перетворення також буде паралельними.
- **Збереження пропорційності:** Афінні перетворення зберігають пропорціональність відстаней. Якщо відстані між точками у вихідній системі координат пропорційні відстаням між їх

відображеннями після афінного перетворення, то це перетворення зберігає пропорціональність.

- Збереження точок: Афінні перетворення зберігають точки, які лежать на одній прямій. Якщо три точки у вихідній системі координат лежать на одній прямій, то їх відображення після афінного перетворення також будуть лежати на одній прямій.

$$T(\lambda A + \mu B) = \lambda T(A) + \mu T(B). \quad (1.1)$$

1.2 Ключові аспекти методу

Ключовими аспектами методу афінних перетворень вважаються: матричне представлення, координатна система, композиція та інверсія. Для успішного виконання роботи необхідно розуміти ці поняття.

Матричне представлення є одним з ключових аспектів методу афінних перетворень в комп'ютерних системах обробки графічної інформації [3]. Воно надає зручний спосіб опису трансформацій і дозволяє ефективно виконувати їх на комп'ютерах, використовуючи матричні операції.

Матриця зберігає інформацію про афінне перетворення у тривимірному просторі, зокрема про зсув, масштабування, обертання та зеркальне відображення. Матриця афінного перетворення зазвичай є квадратною 4x4 матрицею, де перші три рядки та стовпці відповідають за перетворення координат точок у просторі, а останній ряд та стовпець відповідають за проєктивну координату точок.

Матричне представлення дозволяє зручно комбінувати декілька афінних перетворень. Наприклад, якщо потрібно спочатку здійснити зсув, а потім обертання, можна просто помножити матриці цих перетворень одна на одну. Це дає можливість створювати складніші трансформації шляхом послідовного застосування простіших перетворень.

Матриці афінних перетворень мають певні властивості, які роблять їх ефективними для обробки графічної інформації. Зокрема, їх можна легко комбінувати за допомогою матричних операцій, що дозволяє швидко і точно виконувати послідовні перетворення. Крім того, матриці афінних перетворень можна ефективно обернути, що дозволяє легко відтворити початковий стан об'єкта після здійснення перетворень.

Наприклад, матриця афінного перетворення для зсуву об'єкта на вектор (t_x, t_y, t_z) має наступний вигляд (1.2):

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

Для обертання навколо осі x на кут α матриця має наступний вигляд (1.3):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

Матричне представлення афінних перетворень дозволяє ефективно реалізовувати широкий спектр графічних ефектів. Розуміння матричного представлення афінних перетворень є важливим для розробників графічних програм та систем, що працюють з комп'ютерною графікою.

Координатна система є також важливим інструментом у роботі з афінними перетвореннями в комп'ютерних системах обробки графічної інформації. Вона надає зручний спосіб опису розташування та форми об'єктів у просторі. У цьому підрозділі ми розглянемо вплив афінних перетворень на координатну систему та способи розрахунку перетворених координат.

При застосуванні афінного перетворення до об'єкта змінюється його розташування, форма або орієнтація. Це можна розглядати як перехід від

початкової координатної системи до нової координатної системи, в якій об'єкт знаходиться у зміненому стані.

Існують два основних підходи до роботи з афінними перетвореннями: відносний і абсолютний. У відносній системі координат перетворення здійснюються відносно поточного розташування об'єкта. Наприклад, зсув на певний вектор здійснюється шляхом додавання цього вектора до поточних координат об'єкта. У абсолютній системі координат перетворення зберігаються у вигляді послідовності, що дозволяє накладати їх одне на одне.

У абсолютній системі координат, незалежно від поточного розташування об'єкта, перетворення виконуються у відношенні до глобальної координатної системи. Для цього можна використовувати матрицю перетворення, яка діє на всі точки об'єкта.

Афінні перетворення можуть змінювати розташування об'єкта, його форму та орієнтацію. Наприклад, зсув змінює положення об'єкта вздовж осей координат, масштабування збільшує або зменшує розмір об'єкта, обертання поворотом змінює орієнтацію об'єкта.

Ці зміни можна моделювати за допомогою математичних формул або матричних операцій. Наприклад, для зсуву об'єкта на вектор (t_x, t_y) у відносній системі координат, нові координати точки (x, y) обчислюються за формулою (1.4):

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases} \quad (1.4)$$

У випадку абсолютної системи координат, можна використовувати матрицю перетворення T (1.5):

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

Обернене афінне перетворення також може бути застосоване в контексті координатних систем. Воно дозволяє повернути об'єкт до його початкової позиції відносно глобальної або локальної координатної системи.

Ще одним важливим поняттям в області обробки графічної інформації є композиція. Вона дозволяє поєднувати різні афінні перетворення в одне складне перетворення, що дозволяє здійснювати більш складні трансформації об'єктів [4].

Композиція двох афінних перетворень A та B - це виконання спочатку перетворення B , а потім перетворення A . В результаті отримується нове перетворення, яке відображає об'єкт через два послідовних перетворення.

Матричне представлення афінних перетворень надає зручний спосіб здійснювати композицію перетворень. Якщо матриця A представляє перше перетворення, а матриця B - друге перетворення, то композиція відповідає множенню матриць (1.6):

$$C = A * B, \quad (1.6)$$

де C - матриця, що представляє композицію перетворень.

Порядок застосування афінних перетворень впливає на кінцевий результат. Зазвичай, композиція перетворень відбувається зліва направо. Це означає, що спочатку застосовується друге перетворення, а потім перше.

Також важливою операцією АП є інверсія, яка дозволяє повернути об'єкт до початкового стану після застосування афінного перетворення. Обернене афінне перетворення є таким, що скасовує ефект оригінального перетворення і повертає об'єкт до його початкового розташування, форми і орієнтації.

Обернене афінне перетворення A^{-1} до афінного перетворення A визначається таким чином, що композиція цих двох перетворень дорівнює тотожному перетворенню (1.7):

$$A * A^{-1} = A^{-1} * A = I, \quad (1.7)$$

де A - оригінальне афінне перетворення, A^{-1} - його обернене перетворення, I - тотожне перетворення (яке не змінює об'єкт).

Матричне представлення афінних перетворень надає нам зручний спосіб знаходження оберненого перетворення. Якщо матриця A представляє оригінальне афінне перетворення, то обернена матриця A^{-1} буде матрицею, що задає обернене перетворення.

Для знаходження оберненої матриці A^{-1} до матриці A можна скористатися різними методами, такими як метод Гауса-Жордана, метод оберненої матриці та інші. За допомогою цих методів можна обчислити матрицю A^{-1} , яка буде використовуватися для скасування ефекту оригінального перетворення.

Обернене афінне перетворення має широкий спектр застосувань у комп'ютерній графіці. Воно може бути використане для відновлення початкової форми об'єктів після їх трансформації, для виправлення помилкових перетворень або для редагування вже зроблених перетворень.

Наприклад, якщо ми застосували перетворення зсуву до об'єкта, щоб змістити його на певну відстань, ми можемо застосувати обернене перетворення зі зсувом в протилежному напрямку, щоб повернути об'єкт у його початкове положення [5].

1.3 Особливості методу на площині та у просторі

У цьому підрозділі ми детально розглянемо особливості афінних перетворень на площині та їх математичні властивості.

Афінні перетворення відображають одну площину на іншу, зберігаючи прямі лінії та паралельність. Формально, афінне перетворення задається у вигляді лінійного відображення зі зсувом (1.8):

$$T(v) = Av + b, \quad (1.8)$$

де v - вхідний вектор, A - матриця лінійного відображення, b - вектор зсуву.

Афінні перетворення можна представити як комбінацію лінійних перетворень (матричних множень) та зсувів [6].

Зсув є базовим елементом афінних перетворень. Він зміщує кожну точку на площині на певний вектор b . Зсув не змінює розмірів та форми об'єкта, а лише зсуває його у просторі.

Масштабування дозволяє змінювати розміри об'єкта на площині. Це досягається шляхом множення кожної точки на певний коефіцієнт масштабу. Масштабування може бути однорідним (однакове зміщення в усіх напрямках) або нерівномірним (різне зміщення в різних напрямках).

Поворот змінює орієнтацію об'єкта на площині навколо певної точки або вісі. Він досягається за допомогою матриці повороту, яка виконує лінійне перетворення точок відносно обраної точки або вісі.

Відображення змінює положення об'єкта відносно деякої вісі або площини. Воно може бути симетричним (відображення відбувається відносно вісі або площини) або несиметричним (відображення відбувається відносно довільної прямої або площини).

Афінні перетворення можна складати, застосовуючи послідовність зсувів, масштабувань, поворотів та відображень. Порядок складання перетворень впливає на кінцевий результат, тому важливо враховувати правильну послідовність операцій.

Афінне перетворення у тривимірному просторі задається у вигляді лінійного перетворення точок, яке включає зсув, масштабування, повороти та відображення. Воно відображає один тривимірний простір на інший, зберігаючи прямі лінії та паралельність [7].

Зсув у тривимірному просторі зміщує кожну точку на певний вектор. Він виконується шляхом додавання вектора зсуву до координат кожної точки. Зсув

не змінює форму та розміри об'єкта, а лише зсуває його у тривимірному просторі.

Масштабування у тривимірному просторі дозволяє змінювати розміри об'єкта. Це досягається множенням координат кожної точки на коефіцієнти масштабу по трьом осях. Масштабування може бути однорідним (однаковий масштаб усіх напрямків) або нерівномірним (різний масштаб по різних осях).

Поворот у тривимірному просторі змінює орієнтацію об'єкта навколо певної вісі або точки. Він досягається за допомогою матриці повороту, яка виконує лінійне перетворення точок відносно обраної вісі або точки. Повороти можуть бути виконані по одній або кількох осях.

Відображення у тривимірному просторі змінює положення об'єкта відносно деякої вісі або площини. Воно може бути симетричним (відображення відбувається відносно вісі або площини) або несиметричним (відображення відбувається відносно довільної прямої або площини).

Особливості афінних перетворень у просторі дозволяють ефективно маніпулювати тривимірними об'єктами у комп'ютерних системах обробки графічної інформації. Знання про математичні властивості афінних перетворень та їх застосування допомагають розробникам створювати реалістичні тривимірні сцени.

Висновки за розділом 1

У розділі "Математичні основи афінних перетворень" ми розглянули ключові аспекти методу афінних перетворень в комп'ютерних системах обробки графічної інформації. Під час дослідження ми ознайомилися з поняттям і властивостями афінних перетворень, що дозволило нам розуміти їх сутність та значення у графічних програмах та системах.

Матричне представлення афінних перетворень виявилось надзвичайно корисним і зручним для виконання трансформацій у тривимірному просторі. Ми досліджували матричні операції, які дозволяють комбінувати та послідовно

застосовувати афінні перетворення з метою створення складних графічних ефектів. Також ми вивчили властивості матриць афінних перетворень, зокрема їх оборотність, що дозволяє легко відтворити початковий стан об'єкта після здійснення перетворень.

Розглянули метричну систему координат і її значення для опису розташування та орієнтації об'єктів у просторі. Вона дозволяє точно визначити положення об'єкта за допомогою координат та векторів.

Вивчили поняття композиції афінних перетворень і їх послідовність. Було показано, як застосування декількох перетворень підряд дозволяє створювати складніші трансформації та змінювати форму та розташування графічних об'єктів.

Досліджували поняття інверсії афінних перетворень, що дозволяє повернути об'єкт у початковий стан після виконання перетворень. Розуміння інверсії є важливим для забезпечення правильного виконання трансформацій та збереження інтегральних характеристик об'єкта.

Результати дослідження цього розділу підкреслюють важливість розуміння математичних основ афінних перетворень у графічних програмах та системах. Ці знання є необхідними для розробників та користувачів, оскільки вони надають засади для ефективної роботи з графічною інформацією і дозволяють створювати реалістичні та складні графічні ефекти.

РОЗДІЛ 2

ВИКОРИСТАННЯ МЕТОДУ У КОМП'ЮТЕРНИХ СИСТЕМАХ

2.1 Трансформація 2D об'єктів та 3D моделей

Використання афінних перетворень знаходить широке застосування в комп'ютерних системах обробки графічної інформації, зокрема в рендерингу та трансформації 2D об'єктів. Рендеринг є процесом перетворення векторних або растрових об'єктів у зображення, які можуть бути відображені на екрані або в друкованій формі. Трансформація, у свою чергу, охоплює зміну положення, розміру, орієнтації та інших параметрів об'єктів з метою досягнення бажаного візуального ефекту [8].

У цьому підрозділі ми розглянемо роль афінних перетворень у рендерингу та трансформації 2D об'єктів та роль АП у просторовій трансформації 3D моделей. Афінні перетворення, які включають зсув, масштабування, обертання та відображення, дозволяють змінювати положення, розмір та форму об'єктів у двовимірному просторі. Вони забезпечують потужний інструментарій для створення різноманітних графічних ефектів та реалізації важливих функцій у комп'ютерній графіці.

Застосування афінних перетворень у рендерингу та трансформації 2D об'єктів дозволяє досягнути багатьох важливих результатів. Зокрема, зсув дозволяє змінити положення об'єкту відносно вісей координатної системи. Це може бути використано для переміщення об'єкту на нову позицію або для анімації його руху. Масштабування дозволяє змінювати розмір об'єкту, що є корисним для збільшення або зменшення його відображення. Обертання використовується для повороту об'єкту навколо заданої точки або осі, що дозволяє створювати ефект обертання об'єкту. Відображення використовується для створення дзеркального відображення об'єкту вздовж заданої прямої, що

дозволяє створювати ефект симетрії та реалістично відображати об'єкти, наприклад, у водяних поверхнях або дзеркалах.

Для реалізації цих афінних перетворень можна використовувати матриці трансформацій, які дозволяють представити перетворення у вигляді лінійних операцій над координатами об'єкту. Застосування матриць трансформацій у рендерингу та трансформації 2D об'єктів дозволяє зручно та ефективно виконувати різноманітні перетворення та забезпечує гнучкість та точність у роботі з графічною інформацією.

У сучасних комп'ютерних системах обробки графічної інформації використання афінних перетворень має важливе значення для реалізації просторової трансформації 3D моделей. Просторова трансформація включає зміну положення, розміру, орієнтації та форми 3D об'єктів з метою створення різноманітних візуальних ефектів, реалістичного відображення та анімації.

Такі перетворення як зсув, масштабування, обертання та відображення, дозволяють змінювати положення, розмір, орієнтацію та форму об'єктів у тривимірному просторі. Ці перетворення використовуються для створення різноманітних ефектів, таких як переміщення об'єктів у сцені, зміна їхнього розміру або форми, обертання об'єктів навколо вісей або точок, а також створення дзеркального відображення.

Просторова трансформація 3D моделей зазвичай виконується за допомогою матриць трансформацій. Кожне афінне перетворення може бути представлене матрицею, що відображає відповідні лінійні операції над координатами об'єкту. Застосування матриць трансформацій у просторовій трансформації 3D моделей дозволяє зручно та ефективно виконувати різноманітні перетворення та забезпечує точність і гнучкість у роботі з тривимірною графікою.

Важливим аспектом просторової трансформації 3D моделей є управління поглядом, яке дозволяє змінювати точку спостереження та напрямок перегляду об'єкта в сцені. Це дає змогу створювати різні камерні ракурси, панорамні зображення та інші ефекти, які залежать від точки спостереження. Управління

поглядом є важливою складовою просторової трансформації 3D моделей і активно використовується у віртуальній реальності, комп'ютерних іграх та інших додатках, де взаємодія користувача з тривимірними об'єктами є ключовою.

2.2 Зображення та текстури

У комп'ютерних системах обробки графічної інформації, АП використовуються для зображення та текстуризації об'єктів. Зображення та текстури графічних об'єктів додають реалізм, деталізацію та візуальну привабливість до графічних сцен [9].

Афінні перетворення використовуються для позиціонування та масштабування текстур на поверхні об'єктів. Це дозволяє створювати різноманітні ефекти, такі як повторювані текстури, спотворення текстур залежно від кута нахилу поверхні, розтягування або стискання текстур тощо. Завдяки афінним перетворенням, ми можемо змінювати розміри та пропорції текстур, адаптуючи їх до потреб конкретної графічної сцени.

Крім того, афінні перетворення дозволяють змінювати позицію та орієнтацію зображень на поверхні об'єктів. Це особливо корисно для створення ефектів проекції зображень, тіней, відбиття та інших реалістичних візуальних елементів. Наприклад, застосування афінних перетворень може дозволити нам змінювати положення та орієнтацію текстур, що використовуються для створення тіней або відбиття, залежно від положення та орієнтації об'єктів у сцені.

2.3 Проекції та перспектива

Проекції та перспектива дозволяють створювати вражаючі тривимірні візуальні ефекти та передавати відчуття глибини та просторовості у графічних сценах [10]. У цьому підрозділі ми детально розглянемо принципи та методики

проекцій та перспективи, а також роль афінних перетворень у їхньому використанні.

Проекція є процесом відображення тривимірних об'єктів на площину з метою створення двовимірного зображення. У комп'ютерних системах, для представлення тривимірних сцен, використовуються різні види проекцій, такі як паралельна проекція, перспективна проекція, ортогональна проекція тощо.

Паралельна проекція є одним з основних методів проекції, при якому паралельні промені проходять через об'єкт та візуальну площину, що призводить до збереження пропорцій та розмірів об'єктів. Вона широко використовується у графічних системах для створення двовимірних зображень з тривимірних моделей.

Перспективна проекція дозволяє передати відчуття глибини та просторовості, використовуючи концепцію збільшення об'єктів у міру їх віддаленості від спостерігача. Цей вид проекції використовується у віртуальній реальності, комп'ютерних іграх та архітектурному моделюванні для створення реалістичних зображень та вражаючих візуальних ефектів.

Афінні перетворення відіграють ключову роль у процесі проекцій. Вони використовуються для зміни розмірів, позиції, орієнтації та спотворення об'єктів у тривимірному просторі перед їхнім проекцією на двовимірну площину. Завдяки афінним перетворенням, ми можемо керувати перспективою, масштабуванням та розташуванням об'єктів у проекціях, надаючи їм відповідну просторову глибину та реалістичність.

Для реалізації проекцій у комп'ютерних системах використовуються математичні моделі, які базуються на афінних перетвореннях. Ці моделі описують перетворення координат об'єктів у тривимірному просторі на координати відповідних точок у двовимірному просторі. Для цього використовуються матриці перетворень, які враховують параметри проекцій, такі як точка спостереження, позиція камери, кут огляду тощо.

Проекції та перспектива мають велике значення в сфері віртуальної реальності та комп'ютерних ігор. Вони допомагають створювати ілюзію

просторовості, забезпечуючи глибину та реалістичність графічних сцен. Гравці можуть відчувати присутність у віртуальному світі завдяки правильно налаштованим проекціям та перспективам.

У сфері архітектурного моделювання та візуалізації, проекції та перспектива є важливими інструментами для створення реалістичних зображень будівель та споруд. Вони дозволяють архітекторам та дизайнерам відображати об'єкти з різних ракурсів та показувати їхню просторову структуру, що сприяє кращому розумінню проекту та його візуалізації.

У цьому підрозділі ми розглянули основні принципи та методики проекцій та перспективи у комп'ютерних системах обробки графічної інформації. Зрозумівши роль афінних перетворень у процесі проекцій, ми бачимо, як вони допомагають створювати вражаючі візуальні ефекти та реалістичні графічні сцени. Використання афінних перетворень у цих процесах важливо для досягнення реалістичності, просторовості та візуальної якості графічних образів.

2.4 Практичне застосування методу

Графічні редактори є невід'ємною частиною сучасних комп'ютерних систем обробки графічної інформації. Вони надають користувачам зручні та потужні інструменти для створення, редагування та маніпулювання зображеннями. Одним із ключових аспектів роботи графічних редакторів є здатність змінювати форму, розмір, положення та орієнтацію об'єктів на зображенні. Для досягнення цих ефектів широко використовуються методи афінних перетворень [11].

У графічних редакторах афінні перетворення використовуються для реалізації різноманітних функцій, таких як переміщення, масштабування, обертання, відображення тощо. Ці перетворення дозволяють користувачам змінювати положення, розмір та орієнтацію об'єктів на зображенні з високою точністю та гнучкістю. Застосування афінних перетворень у графічних

редакторах дозволяє досягнути великої кількості ефектів та трансформацій, що значно розширює можливості користувача при створенні та редагуванні зображень.

Один з найпоширеніших сценаріїв використання афінних перетворень в графічних редакторах - це зміна розміру та масштабування об'єктів на зображенні. Використовуючи афінні перетворення, користувач може збільшити або зменшити розмір об'єкта, зберігаючи при цьому його пропорції та якість. Цей процес може бути виконаний шляхом множення координат об'єкта на матрицю масштабування, що дозволяє змінювати розмір об'єкта усіх його точок одночасно. Такий підхід дозволяє зручно та ефективно змінювати розмір об'єктів на зображенні без необхідності перерахунку окремих точок.

Ще одним важливим аспектом застосування афінних перетворень в графічних редакторах є переміщення об'єктів у просторі. Завдяки афінним перетворенням користувач може зручно зміщувати об'єкти вгору, вниз, вліво або вправо, змінюючи їх положення на зображенні. Цей процес може бути реалізований шляхом додавання відповідних значень до координат об'єкта, що забезпечує його переміщення у потрібному напрямку та відстані. Застосування афінних перетворень у переміщенні об'єктів дозволяє зручно та точно розміщувати їх на зображенні з мінімальними зусиллями користувача.

Іншим важливим застосуванням афінних перетворень у графічних редакторах є обертання та зміна орієнтації об'єктів. Цей процес дозволяє змінити кут нахилу об'єкта, що дає можливість створювати цікаві та динамічні ефекти на зображенні. Використання афінних перетворень дозволяє обертати об'єкти навколо певної точки або вздовж певної вісі з заданим кутом повороту. Це забезпечує гнучкість та контроль над орієнтацією об'єктів, що є важливим аспектом в графічних редакторах.

Ще одним застосуванням афінних перетворень в графічних редакторах є зведення тривимірних об'єктів до двовимірного простору. Завдяки афінним перетворенням, тривимірні об'єкти можуть бути проекційно відображені на плоску площину, що дає можливість їх подальшого редагування та

використання в двовимірному контексті. Це особливо корисно для створення реалістичних рисунків та ілюстрацій, де тривимірні об'єкти можуть бути зображені в перспективі або пласко.

Загалом, застосування методу афінних перетворень в графічних редакторах відкриває широкі можливості для творчого вираження та маніпуляції зображеннями. Від простого зміщення та масштабування до складних перетворень об'єктів у тривимірному просторі, афінні перетворення є потужним інструментом для редагування та створення графічного контенту.

Віртуальна реальність є однією з найперспективніших технологій сучасності, яка дозволяє користувачам зануритися у інші світи та взаємодіяти з ними у віртуальному просторі. Для досягнення реалістичного ефекту та плавної інтерактивності в віртуальній реальності використовуються різноманітні графічні техніки, серед яких методи афінних перетворень відіграють важливу роль. У цьому підрозділі ми розглянемо застосування методу афінних перетворень у віртуальній реальності і роль, яку вони відіграють у створенні іммерсивного та реалістичного віртуального досвіду.

Одним з основних використань афінних перетворень у віртуальній реальності є позиціонування об'єктів у тривимірному просторі. Завдяки афінним перетворенням, об'єкти можуть бути зміщені, обернуті та масштабовані відносно віртуальної камери чи користувача. Це дозволяє створювати реалістичні сцени та іммерсивні досвіди, де об'єкти рухаються та взаємодіють з користувачем у віртуальному просторі.

Відображення перспективи є важливим аспектом створення реалістичних сцен у віртуальній реальності. Використання афінних перетворень дозволяє створювати віртуальні камери, які відтворюють ефекти перспективного спотворення, що спостерігаються в реальному світі. Це допомагає створити враження глибини та просторовості віртуальних сцен, що підсилює іммерсивність досвіду користувача.

Афінні перетворення використовуються також для анімації об'єктів у віртуальній реальності [12]. Зміна положення, орієнтації та масштабу об'єктів за

допомогою афінних перетворень може створювати рух, трансформації та зміни форми об'єктів у віртуальному просторі. Це дозволяє створювати живі та динамічні сцени, які взаємодіють з користувачем та реагують на його дії.

Текстурування є важливою графічною технікою, що використовується в віртуальній реальності для створення реалістичних та деталізованих об'єктів. За допомогою афінних перетворень, текстури можуть бути прикладені до поверхні об'єктів, що дозволяє зображати різні матеріали, текстури та декоративні ефекти на їх поверхнях. Це сприяє створенню більш реалістичних та докладних віртуальних об'єктів.

Висновки за розділом 2

У даному розділі було розглянуто використання методу афінних перетворень у комп'ютерних системах. Ми дослідили трансформацію 2D об'єктів та 3D моделей за допомогою цього методу.

Також, ми дослідили використання методу афінних перетворень для зображень та текстур. За допомогою цього методу можна змінювати розмір, обертати та переносити зображення, а також застосовувати текстури до об'єктів для створення реалістичних ефектів.

Далі, ми вивчили проєкції та перспективу, які є важливими аспектами методу афінних перетворень. За допомогою проєкцій можна зображати 3D об'єкти на 2D площині, забезпечуючи відчуття глибини та реалістичності. Перспектива дозволяє створювати ілюзію віддалених об'єктів та змінювати їхні розміри залежно від відстані до спостерігача.

Нарешті, було розглянуто практичне застосування методу афінних перетворень у різних галузях. Його потужність і універсальність дозволяють створювати вражаючі візуальні ефекти та забезпечувати високий рівень графічної якості.

В цьому розділі ми вивчили основні аспекти використання методу афінних перетворень у комп'ютерних системах і побачили його значення для

створення реалістичних та ефектних графічних зображень.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ

3.1. Аналіз існуючих програмних рішень

У даному підрозділі проводиться аналіз існуючих програмних рішень, які використовуються для реалізації методу афінних перетворень. Описуються недоліки цих рішень, що можуть впливати на їхню ефективність, точність та зручність використання. Розглядаються різні аспекти, такі як функціональність, продуктивність та інтерфейс користувача. Аналіз допоможе виявити проблемні моменти та визначити потребу у розробці нової програмної реалізації методу афінних перетворень.

Перелік деяких програм, які використовують метод афінних перетворень у своїй роботі:

- Autodesk AutoCAD: AutoCAD є програмою для проектування та моделювання в області архітектури, інженерії та будівництва. Вона використовує афінні перетворення для зміни масштабу, переміщення, обертання та перетворення об'єктів у тривимірному просторі [13].
- Blender: Blender є вільним програмним забезпеченням для створення тривимірних моделей, анімації та візуалізації. Він використовує афінні перетворення для трансформації об'єктів у тривимірному просторі, включаючи зміну масштабу, переміщення, обертання та зсув [14].
- MATLAB: MATLAB є популярним середовищем для чисельних обчислень та програмування. Він має вбудовані функції для виконання афінних перетворень над графічними об'єктами, такими як зображення та графіки [15].

- GIMP: GIMP є безкоштовним редактором зображень з відкритим вихідним кодом. Він використовує методи афінних перетворень для зміни розміру, масштабування, обертання та зсуву об'єктів у зображеннях [16].
- Inkscape: Inkscape є векторним графічним редактором, який використовує афінні перетворення для зміни розміру, масштабування, обертання та переміщення об'єктів у векторних зображеннях [17].
- CorelDRAW: CorelDRAW є комерційним векторним графічним редактором, який використовує афінні перетворення для зміни розміру, масштабування, обертання та пересування об'єктів у векторних зображеннях [18].

Ці програми є лише деякими прикладами існуючих програм, які використовують метод афінних перетворень. Залежно від конкретного застосування та галузі, можуть існувати й інші програми, які використовують цей метод для обробки графічної інформації.

Blender, AutoCAD в цьому програмному рішенні інтерфейс користувача також має певні недоліки. Він може бути складним у використанні та недостатньо інтуїтивно зрозумілим для користувачів. Відсутність зручних інструментів для вибору об'єктів, задання параметрів перетворень та перегляду результатів може ускладнити процес роботи та знизити ефективність використання програми. Наступним недоліком є низька продуктивність програми. При застосуванні афінних перетворень до великих об'єктів або великої кількості об'єктів швидкість обробки знижується значно. Це може призвести до значних затримок під час роботи з програмою та утруднити продуктивну та швидку роботу з графічними об'єктами.

MATLAB має певні недоліки, які варто відзначити. Одним з недоліків є обмежені можливості функціональності. Програма дозволяє здійснювати лише базові афінні перетворення, такі як зсув, масштабування та поворот. Відсутня підтримка більш складних перетворень, наприклад, деформації об'єктів або

складних просторових трансформацій. Це обмеження знижує потенціал програми для роботи з складними графічними об'єктами.

Такі програми, як GIMP, Inkscape, CorelDRAW також мають свої недоліки, які варто врахувати при аналізі. Одним з найбільших недоліків є складність установки та налаштування програми. Вона вимагає наявності певного набору зовнішніх залежностей, які потрібно встановити окремо. Це може створювати проблеми для користувачів, особливо для початківців або тих, хто не має достатнього досвіду в установці програмного забезпечення. Крім того, ці програми мають недостатню документацію та підтримку. Інформація про функціональність, можливості та правила використання програм не завжди доступна або недостатньо детально описана. Відсутність достатньої документації може ускладнити розуміння можливостей програми та використання її відповідно до вимог користувача. Також, програмне рішення цих програм має обмежену підтримку форматів графічних файлів. Не всі популярні формати, такі як PNG, JPEG або SVG, можуть бути імпортовані або експортовані з програми. Це може створювати обмеження у роботі зі зовнішніми файлами та обміні даними з іншими програмами.

Аналіз існуючих програмних рішень показав наявність певних недоліків, які можуть впливати на ефективність, точність та зручність використання методу афінних перетворень. Обмежені можливості функціональності, низька продуктивність, складний інтерфейс користувача, проблеми установки та налаштування, недостатня документація та підтримка, а також обмежена підтримка форматів графічних файлів є основними недоліками, які вимагають уваги та можуть бути вирішені у новій програмній реалізації методу афінних перетворень. З урахуванням цих недоліків виникає необхідність у розробці нового програмного забезпечення, який вирішує ці проблемні питання та буде максимально ефективним, швидким, точним та зручним у використанні для роботи з афінними перетвореннями графічних об'єктів та моделей у комп'ютерних системах.

3.2. Вибір інструментів розробки

У даному підрозділі розглядається вибір інструментів розробки для програмної реалізації методу афінних перетворень у комп'ютерних системах. Зокрема, розглядається використання мови програмування Java та інтегрованого середовища розробки IntelliJ IDEA.

При виборі мови програмування для розробки програмної реалізації методу афінних перетворень було обрано Java. Java є потужною та популярною об'єктно-орієнтованою мовою програмування, яка забезпечує широкі можливості для розробки графічних програм та обробки графічної інформації. Вона пропонує зручний синтаксис, багатий набір стандартних бібліотек та широку спільноту розробників [19].

Для розробки програми було обрано IntelliJ IDEA як інтегроване середовище розробки (IDE). IntelliJ IDEA є одним з найпопулярніших та потужних IDE для розробки програм на мові Java. Воно надає широкий набір функцій, зручний інтерфейс користувача та підтримку різних інструментів для покращення продуктивності розробника. IntelliJ IDEA також має потужну систему підказок, автодоповнення коду та інструменти для тестування та налагодження програм [20].

Після вибору мови програмування та інструменту розробки, було налаштовано проект у IntelliJ IDEA. Це включало створення нового проекту, налаштування залежностей та імпорту необхідних бібліотек. Крім того, були налаштовані параметри компіляції, забезпечено належну структуру проекту та налаштовано систему збирання проекту.

За допомогою мови програмування Java та інтегрованого середовища розробки IntelliJ IDEA було реалізовано метод афінних перетворень. Це включало написання відповідних класів та функцій для здійснення афінних перетворень, обробки графічних об'єктів та взаємодії з користувачем. Також було використано стандартні бібліотеки Java для роботи з графікою та обробки зображень [21, 22].

Після реалізації методу афінних перетворень було проведено тестування програми для перевірки правильності роботи. Використовуючи різноманітні тестові дані та сценарії, було перевірено коректність виконання афінних перетворень та інші функціональні можливості програми. При виявленні помилок або недоліків, було здійснено налагодження та виправлення програми.

Використання мови програмування Java та інтегрованого середовища розробки IntelliJ IDEA забезпечує зручність, ефективність та потужність в процесі розробки програми. Налаштування проекту, реалізація методу афінних перетворень, тестування, налагодження та оптимізація гарантують стабільну та ефективну роботу програми.

3.3. Реалізація функціоналу

У цьому підрозділі буде розглянуто реалізацію функціоналу програми, яка використовує метод афінних перетворень. Програмна реалізація дозволяє виконувати різноманітні операції з графічною інформацією, використовуючи афінні перетворення для зміни положення, масштабування, обертання та інших трансформацій об'єктів.

У програмі були реалізовані функції для виконання афінних перетворень. Це включає обчислення матриць перетворень, застосування їх до графічних об'єктів та оновлення їх положення та вигляду. Реалізація забезпечує точність та ефективність виконання перетворень. Детальний код реалізації програми наведений у Додатку Г (Лістинг 1).

Для реалізації методу було використано клас пакетування `java.awt.geom.AffineTransform`. `AffineTransform` клас представляє 2D афінне перетворення, яке виконує лінійне відображення від 2D координат до інших 2D координат. Афінні перетворення можуть бути створені, використовуючи послідовності перетворень, масштабів, дзеркальних віддзеркалень і обертань [23].

На рисунку 3.1 можна побачити створення об'єкту класу `AffineTransform`, об'єкт застосовує параметри зміни положення, масштабування та повороту, що потрібно застосувати до зображення.

```
private void applyAffineTransform(int x, int y, double k, int corner) {
    // Застосовуємо афінне перетворення до зображення
    java.awt.geom.AffineTransform at = new java.awt.geom.AffineTransform();
    at.translate(x, y);
    at.rotate(Math.toRadians(corner));
    at.scale(k, k);
    image = applyAffineTransform(image, at);
}
```

Рисунок 3.1 – Створення об'єкту класу `AffineTransform`

Віддзеркалення було вирішено помістити в окремий метод для зручності. На рисунку 3.2 відображено застосування віддзеркалення зображення відносно осей x та y .

```
private void checkMethod() {
    java.awt.geom.AffineTransform at = null;
    if (checkBoxX.isSelected()) {
        at = java.awt.geom.AffineTransform.getScaleInstance( sx: 1, sy: -1);
        at.translate( tx: 0, -image.getHeight());
    }
    else if (checkBoxY.isSelected()) {
        at = java.awt.geom.AffineTransform.getScaleInstance( sx: -1, sy: 1);
        at.translate(-image.getWidth(), ty: 0);
    }
    image = applyAffineTransform(image, at);
}
```

Рисунок 3.2 – Метод віддзеркалення

На рисунку 3.3 можна побачити перемалювання зображення із застосованими змінами.

```
private BufferedImage applyAffineTransform(BufferedImage image, java.awt.geom.AffineTransform at) {
    BufferedImage newImage = new BufferedImage(this.getWidth(), this.getHeight(), image.getType());
    Graphics2D g2d = (Graphics2D) newImage.getGraphics();
    g2d.setRenderingHint(RenderingHints.KEY_INTERPOLATION, RenderingHints.VALUE_INTERPOLATION_BICUBIC);
    g2d.setTransform(at);

    Dimension panelSize = getSize();
    double imageWidth = image.getWidth();
    double imageHeight = image.getHeight();
    double scaleX = panelSize.getWidth() / imageWidth;
    double scaleY = panelSize.getHeight() / imageHeight;
    double scale = Math.min(scaleX, scaleY);
    int scaledWidth = (int) (imageWidth * scale);
    int scaledHeight = (int) (imageHeight * scale);

    // Определение координат отрисовываемого изображения
    int x = (int) ((panelSize.getWidth() - scaledWidth) / 2);
    int y = (int) ((panelSize.getHeight() - scaledHeight) / 2);
    super.paint(g2d);

    // Отрисовка масштабированного изображения
    g2d.drawImage(image, x, y, scaledWidth, scaledHeight, observer: null);
    g2d.dispose();

    return newImage;
}
```

Рисунок 3.3 – Метод зміни зображення

3.4. Інтерфейс

Користувач може виконувати різні дії, такі як вибір зображення, задання параметрів перетворень, відображення результатів, скасування результатів та збереження результату. Інтерфейс забезпечує зручність використання програми та можливість швидко виконувати потрібні дії.

Програма дозволяє завантажувати графічні об'єкти з файлів, відобразити їх на екрані та виконувати різні операції з ними. Вона підтримує різноманітні формати зображень, такі як JPEG, PNG, SVG тощо, і здатна ефективно відобразити їх на екрані.

Програма дозволяє зберігати результати роботи, зокрема зображення після застосування афінних перетворень. Користувач може вказати шлях для збереження. Це дає можливість зберегти результати для подальшого використання або аналізу.

Програма надає зручний та простий інтерфейс для взаємодії з користувачем, це можна побачити на рисунку 3.4.

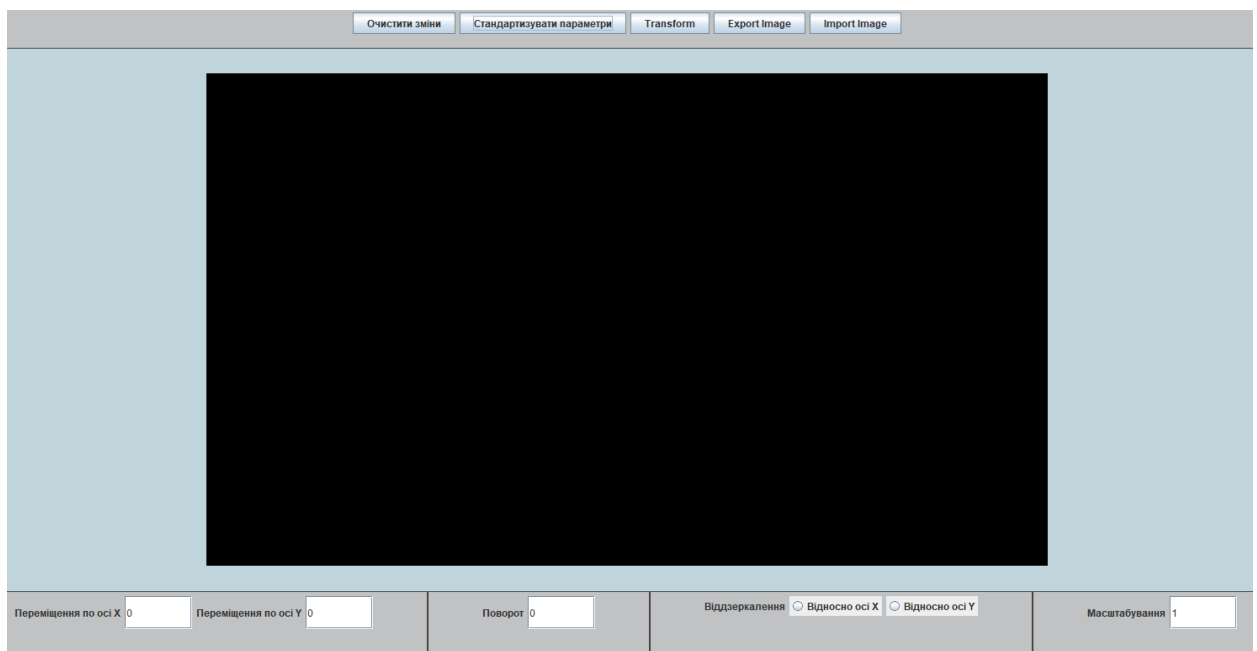


Рисунок 3.4 – Загальний вигляд інтерфейсу

Загалом, інтерфейс користувача можна розділити на 3 частини: верхня панель, центральна панель та нижня.

Верхня панель містить 5 кнопок: «Очистити зміни» (повертає початкове зображення, скасувавши всі застосовані перетворення), «Стандартизувати параметри» (скидає всі значення полів параметрів до стандартних), «Transform» (виконує застосування афінного перетворення до зображення), «Export Image» (зберігає результат на комп'ютері), «Import Image» (обирає зображення для редагування). Верхня панель відображена на рисунку 3.5.

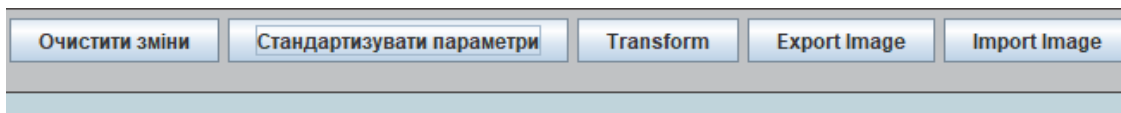


Рисунок 3.5 – Верхня панель

Центральна панель – вікно роботи над зображенням. Тут користувач буде бачити усі застосовані зміни до зображення, аналізувати та змінювати до потрібного результату. Центральна панель відображена на рисунку 3.6.



Рисунок 3.6 – Центральна панель із зображенням

Нижня панель потрібна для взаємодії з користувачем. У кожне поле прописується значення, що потрібно застосувати методом афінного перетворення до зображення. Візуальний вигляд нижньої панелі зображено на рисунку 3.7.

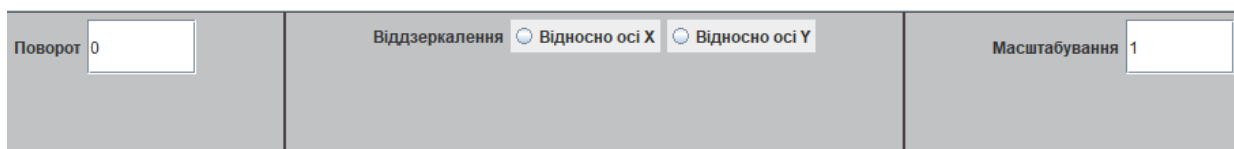


Рисунок 3.7 – Нижня панель

Висновки за розділом 3

У розділі "Програмна реалізація методу" було проведено аналіз існуючих програмних рішень, вибрані необхідні інструменти розробки, реалізовано функціонал та розроблено графічний інтерфейс користувача.

Аналіз існуючих програмних рішень показав, що на ринку існує кілька програм, які використовують метод афінних перетворень. Однак, були виявлені деякі недоліки, зокрема обмеження функціональності, складність у використанні або недостатня підтримка платформ.

При виборі інструментів розробки було приділено увагу таким аспектам, як підтримка мови програмування, наявність необхідних бібліотек та фреймворків, зручність у використанні та розширення, а також спільнота розробників. В результаті була обрана Java як мова програмування, а розробка проводилась з використанням IntelliJ IDEA - потужного та зручного інтегрованого середовища розробки.

Реалізація функціоналу передбачала впровадження методу афінних перетворень для зміни кута нахилу, розташування, масштабування та орієнтації об'єктів у двовимірному просторі.

Графічний інтерфейс користувача був реалізований з урахуванням зручності та інтуїтивності використання. Користувач мав можливість обирати зображення для редагування, застосовувати до них афінні перетворення та спостерігати за змінами в реальному часі. Застосовувалися сучасні технології розробки графічного інтерфейсу, що забезпечували зручність взаємодії та простий вигляд додатку.

У результаті роботи над розділом було успішно розроблено програму, яка використовує метод афінних перетворень для зміни положення об'єктів у двовимірному просторі. Результати аналізу існуючих програмних рішень, вибір правильних інструментів розробки, реалізація функціоналу та графічного інтерфейсу користувача сприяли створенню потужного та зручного інструменту для використання методу афінних перетворень в комп'ютерних системах.

РОЗДІЛ 4

РОБОТА КОРИСТУВАЧА З ПРОГРАМОЮ

4.1. Системні вимоги

Системні вимоги є важливою складовою при розробці програмної реалізації методу афінних перетворень у комп'ютерних системах. Вони визначають необхідні характеристики апаратного та програмного забезпечення, які повинні бути наявними на комп'ютерній системі для ефективного виконання програми.

Для успішної роботи програми, яка використовує метод афінних перетворень, важливо мати наступні системні вимоги:

- Процесор: Мінімумально рекомендується процесор з двома ядрами, що дозволяє виконувати паралельні обчислення.
- Оперативна пам'ять: Мінімумально рекомендується 2 ГБ оперативної пам'яті. Більша кількість оперативної пам'яті дозволить більш ефективно опрацьовувати великі об'єкти та зображення.
- Відеокарта: Рекомендується наявність відеокарти з об'ємом пам'яті від 1 ГБ.
- Операційна система: Програма може працювати на різних операційних системах, таких як Windows, macOS або Linux. Рекомендується використання останньої доступної версії операційної системи для забезпечення сумісності та отримання оновлень безпеки.

Врахування системних вимог дозволить забезпечити належну продуктивність програми, а також уникнути проблем з сумісністю та недостатньою функціональністю. Перевірка відповідності системних вимог

перед встановленням програми допоможе забезпечити ефективну роботу методу афінних перетворень у комп'ютерних системах.

4.2. Встановлення і запуск додатку

Перед детальним описом користування програмою, не зайвим буде ознайомитися з блок-схемою роботи програми. На рисунку 4.1 зображено принцип роботи усієї програми.

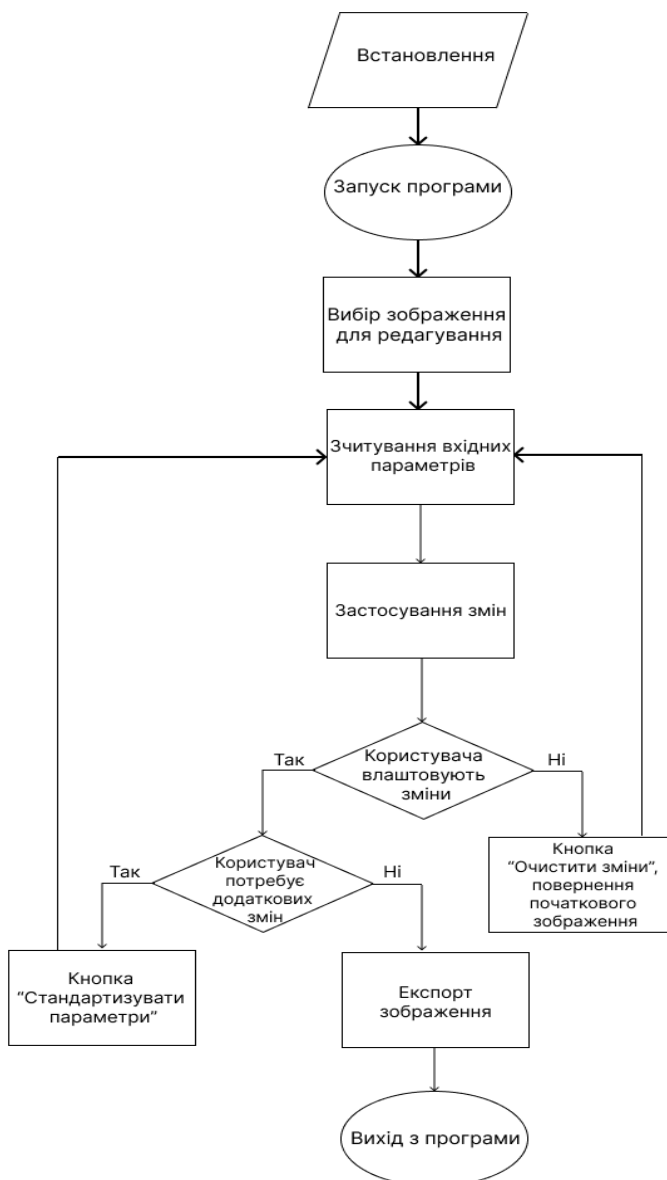


Рисунок 4.1 – Блок-схема роботи програми

Для того, щоб почати використовувати додаток, спершу потрібно встановити на комп'ютер JDK (Java Development Kit). JDK - це пакет розробки Java, який включає в себе необхідні компоненти для компіляції, виконання та розробки програм на мові програмування Java.

JDK надає засоби для компіляції і перетворення вихідного коду додатків на байт-код, який може бути виконаний на віртуальній машині Java (JVM). Це дозволяє забезпечити переносимість програм між різними операційними системами, оскільки вони можуть бути виконані на будь-якій системі, на якій встановлена JVM. Пропонується встановити останню версію JDK 17 з офіційного сайту Oracle [24].

Тепер можна перейти до встановлення додатку. Для зручності користувачів усі файли програми були об'єднані у один архів. Для більш зручного доступу, було вирішено завантажити архів на гугл-диск та надати доступ за посиланням [25], що зображено на рисунку 4.2.

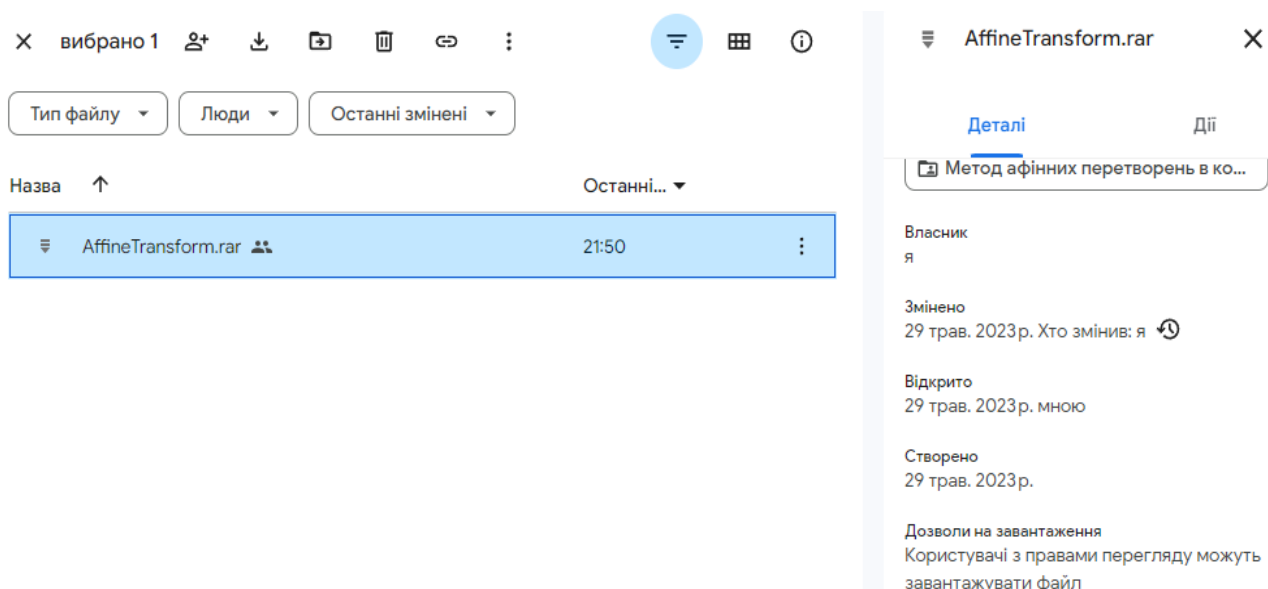


Рисунок 4.2 – Архів на гугл-диску

Після того, як користувач перейшов за посиланням на гугл-диск, необхідно натиснути правою кнопкою миші на файл AffineTransform.rar та у відкритому меню обрати «Завантажити», як це зображено на рисунку 4.3.

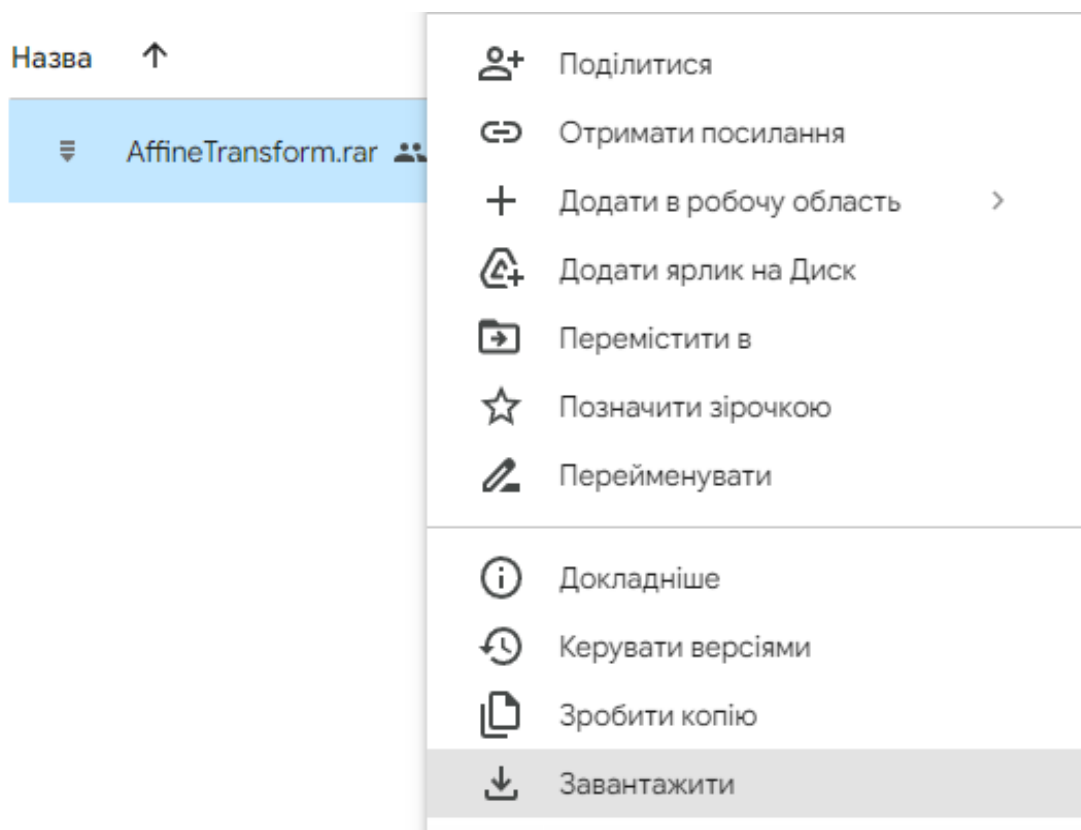


Рисунок 4.3 – Завантаження програми на комп'ютер

Після завантаження архіву, його потрібно розпакувати. Тепер додаток встановлено, можна переходити до використання. Відкриваємо додаток «Метод афінних перетворень у комп'ютерних системах обробки графічної інформації», як це показано на рисунку 4.4.

| dan > AffineTransform

Имя	Дата изменения
.idea	24.05.2023 15:21
out	24.05.2023 15:15
src	24.05.2023 15:17
AffineTransform.iml	24.05.2023 15:12
AffineTransform	24.05.2023 15:18
exported_image	25.05.2023 18:03
Метод афінних перетворень в комп'ют...	24.05.2023 15:20

Рисунок 4.4 – Додаток, завантажений на комп'ютер

Очікуємо, поки програма завантажується, як це показано на рисунку 4.5.

Завантаження програми...



Рисунок 4.5 – Завантаження програми

На початку роботи програми потрібно обрати зображення, яке потрібно відредагувати. Для цього імпортуємо зображення за допомогою кнопки «Import Image», як на рисунку 4.6.

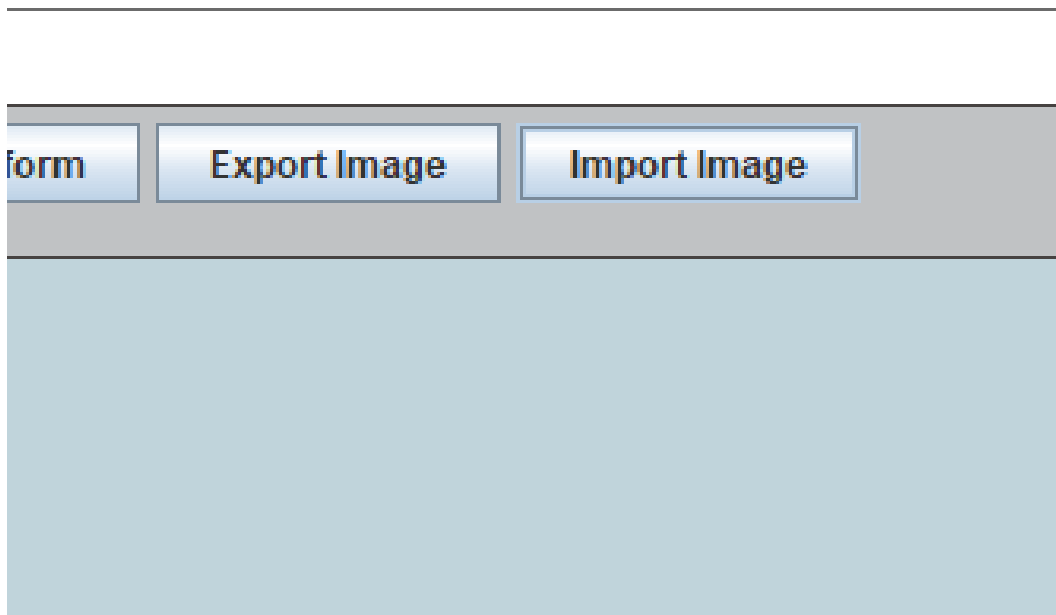


Рисунок 4.6 – Імпортування зображення

Ввести параметри, які необхідні для застосування перетворень у відповідні поля, як на рисунку 4.7.

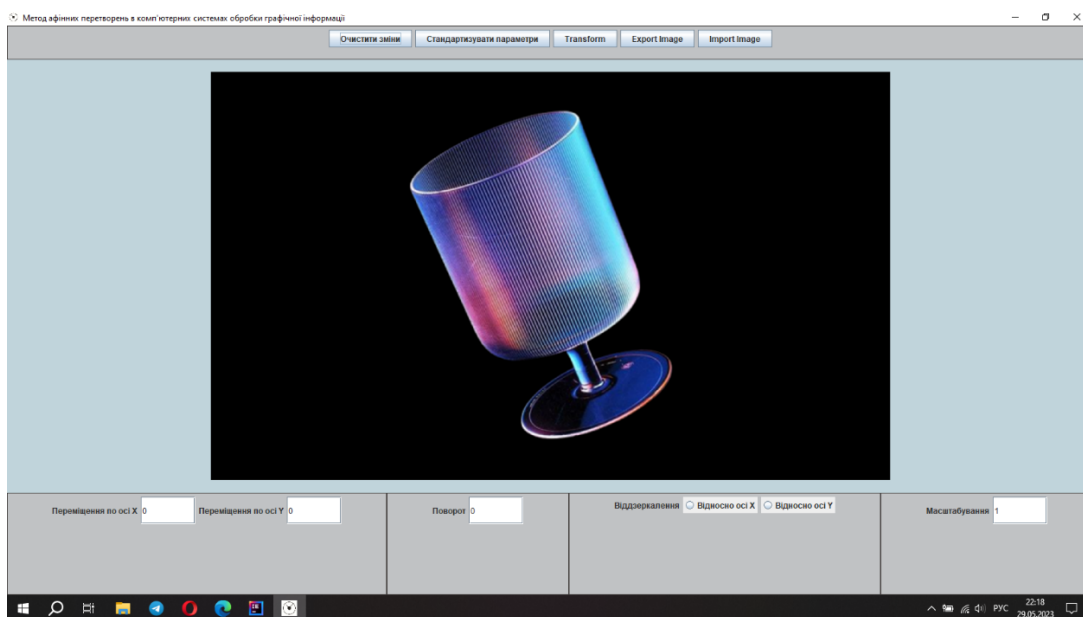


Рисунок 4.7 – Введення параметрів

Натиснути кнопку «Transform», оцінити результат. За необхідності натиснути кнопку «Стандартизувати параметри», для того, щоб скинути значення параметрів до базових. У випадку, якщо було застосовано зайві перетворення можна натиснути кнопку «Очистити зміни», і продовжити роботу з початкового зображення. Повторити перетворення до бажаного результату. Отримати результат зображення, наприклад, як на рисунку 4.8.

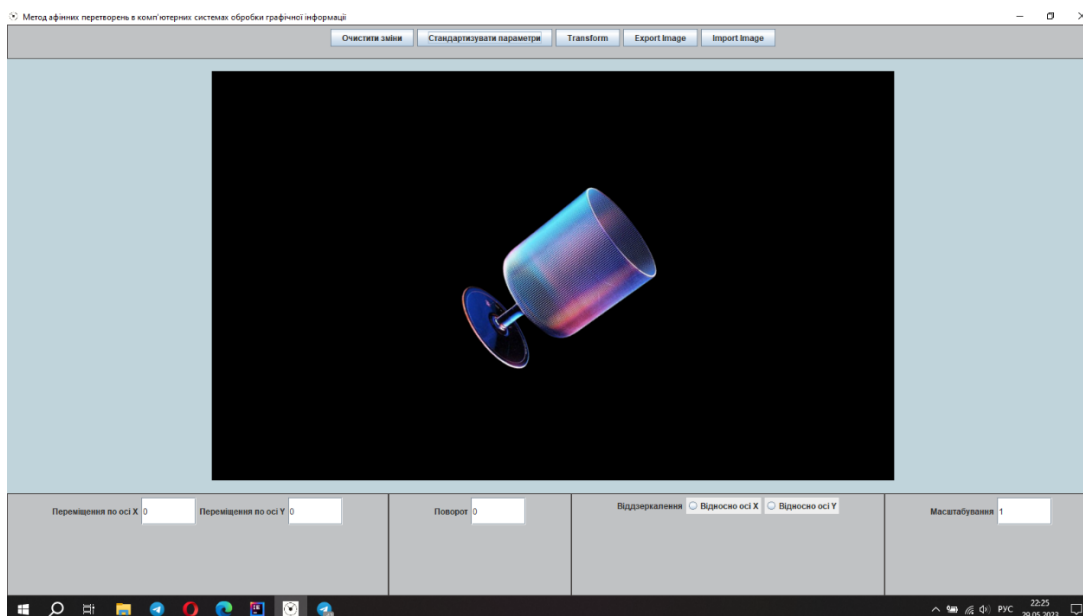
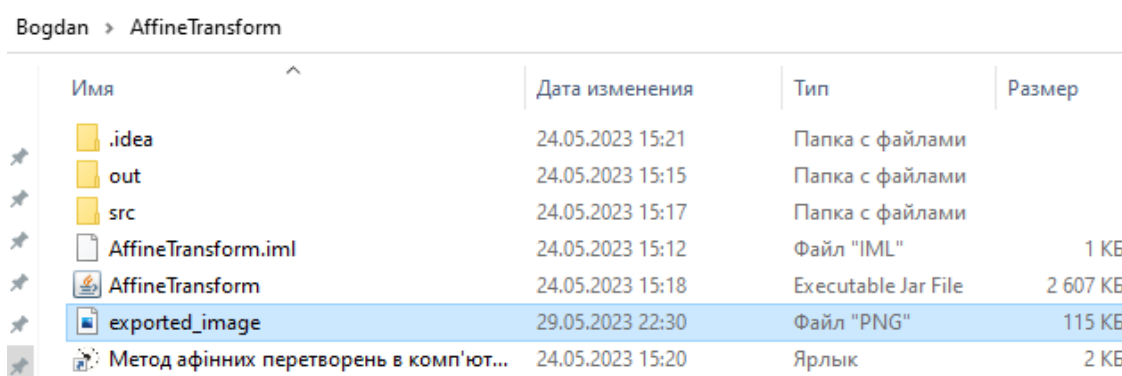


Рисунок 4.8 – Результат зображення

При досягненні бажаного результату, зберігаємо зображення за допомогою кнопки «Export Image». Зображення буде збережено у папці додатку, як на рисунку 4.9.



Имя	Дата изменения	Тип	Размер
.idea	24.05.2023 15:21	Папка с файлами	
out	24.05.2023 15:15	Папка с файлами	
src	24.05.2023 15:17	Папка с файлами	
AffineTransform.iml	24.05.2023 15:12	Файл "IML"	1 КБ
AffineTransform	24.05.2023 15:18	Executable Jar File	2 607 КБ
exported_image	29.05.2023 22:30	Файл "PNG"	115 КБ
Метод афінних перетворень в комп'ют...	24.05.2023 15:20	Ярлык	2 КБ

Рисунок 4.9 – Файл зображення

Висновки за розділом 4

У даному розділі було розглянуто системні вимоги та процес встановлення і запуску додатку. Проаналізувавши ці аспекти, можна зробити наступні висновки.

Системні вимоги є важливою частиною роботи з будь-якою програмою, оскільки вони визначають необхідність певного апаратного та програмного забезпечення для успішного функціонування додатку. У цьому випадку, для користування додатком на комп'ютері необхідно мати встановлену JDK (Java Development Kit), що забезпечить необхідне середовище для виконання програми.

Встановлення і запуск додатку є важливим етапом перед початком роботи з програмою. Користувачам потрібно мати доступ до встановлювального

пакету програми та виконати відповідні кроки для успішного встановлення. Після встановлення, додаток може бути запущений за допомогою відповідного ярлика.

В результаті проведення аналізу системних вимог, опису процесу встановлення і запуску додатку, було визначено, що для користування програмою на комп'ютері необхідно встановити JDK та виконати встановлення архіву. Ці кроки дозволять користувачам успішно запустити додаток і почати його використання.

В цілому, розділ "Робота користувача з програмою" зосереджується на аспектах, пов'язаних з вимогами до системи та процесом встановлення і запуску додатку. Дотримання цих вимог дозволить користувачам безперешкодно почати використання програми і отримати доступ до її функціоналу.

ВИСНОВКИ

В рамках дослідження було проведено аналіз теоретичних основ та практичного застосування методу афінних перетворень. У першому розділі були розглянуті поняття і властивості афінних перетворень, а також висвітлено ключові аспекти цього методу. Особливу увагу було приділено його застосуванню на площині та у просторі.

У другому розділі описано використання методу афінних перетворень у комп'ютерних системах. Було розглянуто трансформацію 2D об'єктів та 3D моделей, їх зображення та текстурування, а також принципи проекції та перспективи. Також були представлені практичні застосування методу в різних областях комп'ютерної графіки.

Третій розділ присвячено програмній реалізації методу афінних перетворень. Здійснений аналіз існуючих програмних рішень, та розроблено нове рішення мовою програмування Java. Крім того, розроблений графічний інтерфейс користувача для зручного використання програми.

Четвертий розділ описує вимоги до системи та процес встановлення і запуску розробленого додатку. Були розглянуті системні вимоги, необхідні для успішного виконання програми, а також кроки, що потрібно виконати для встановлення та запуску додатку.

В результаті дослідження було встановлено, що метод афінних перетворень є потужним інструментом в комп'ютерних системах обробки графічної інформації. Він дозволяє здійснювати різноманітні трансформації об'єктів, покращувати їх візуальну якість та реалістичність. Розроблений додаток на основі методу афінних перетворень відповідає поставленим вимогам і забезпечує зручну роботу з графічними об'єктами. Отже, дипломна робота засвідчує глибоке розуміння методу афінних перетворень та його застосування в комп'ютерних системах обробки графічної інформації. Результати

дослідження можуть бути корисними для розробників програмного забезпечення, які працюють у сфері комп'ютерної графіки та візуалізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Рудавський Ю. К., Костробій П. П., Луник Х. П. та ін. Лінійна алгебра та аналітична геометрія. Львів : Видавництво Державного університету «Львівська політехніка», 1999. 262 с.
2. Яковець В. П., Боровик В. Н., Ваврикович Л. В. Аналітична геометрія. Суми : Університетська книга, 2004. 296 с.
3. Boreskov A., Shikin E. Computer Graphics: from Pixels to Programmable Graphics Hardware. Boca Raton : CRC Press, 2014. 568 p.
4. Kokkos A. Expanding Transformation Theory Affinities between Jack Mezirow and Emancipatory Educationalists. Taylor & Francis Ltd, 2021. 234 p.
5. Геометричне моделювання і комп'ютерна графіка. К.: КНУБА, 2010. 288 с.
6. Шишкін О. В., Боресков А. В. Комп'ютерна графіка. М.: Діалог-МІФІ, 1995. 288 с.
7. Руденко Є.А, Семушева О.Ю. Афінні перетворення об'єктів у комп'ютерній графіці на прикладі 3D-моделі, що обертається, Київ: Наукова думка, 2020. 102 с.
8. Томашевський В.М. Моделювання систем. Київ, Видавнича група ВНУ, 2019. 352 с.
9. Веселовська Г. В. Комп'ютерна графіка: Навчальний посібник для вузів. Херсон: ОЛДІ-плюс, 2004. 582 с.
10. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. Чернівці: Рута, 2009. 343 с.
11. Аналітична геометрія. ч.2. Навчально-методичний посібник / Укл.Ю.В. Яременко, Л.І. Лутченко, Кіровоград: "Антураж А", 2005. 116 с.
12. Комп'ютерна анімація : посібник для студентів "Видавничо-поліграфічна справа" / О. С. Євсєєв., Х.: Вид. ХНЕУ ім. С. Кузнеця, 2014. 152 с.

13. Бойко А. П. Комп'ютерне моделювання в середовищі AUTOCAD. Частина 1. Геометричне та проєкційне креслення : навч. посіб. / А. П. Бойко. Миколаїв : Вид-во ЧНУім. Петра Могили, 2017. 116 с.
14. Allan Brito "Blender 3D: Architecture, Buildings, and Scenery" Packt Publishing, 2016.
15. Гаєв Є.О., Нестеренко Універсальний математичний пакет MATLAB і типові задачі обчислювальної математики. Навчальний посібник. К.:НАУ, 2004. 176 с.
16. Klaus Goelker "GIMP 2.10 for Photographers: Image Editing with Open Source Software" Rocky Nook, 2018.
17. T. Van. Inkscape: Guide to a Vector Drawing Program. Prentice Hall, 2011. 504 p.
18. Gary David Bouton "CorelDRAW: The Official Guide" McGraw-Hill Education, 2019.
19. Paul Deitel, Harvey Deitel "Java: Advanced Features and Programming Techniques", 2017.
20. JetBrains: Встановлення та налаштування середовища розробки IntelliJ IDEA [Електронний ресурс]. – режим доступу: URL: <https://www.jetbrains.com/help/idea/getting-started.html> (дата звернення – 01.06.2023).
21. Oracle: Документація Oracle з бібліотеки Swing для розробки графічного інтерфейсу [Електронний ресурс]. – режим доступу: URL: <https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/javax/swing/package-summary.html> (дата звернення – 01.06.2023).
22. Oracle: Документація Oracle з бібліотеки AWT для розробки графічного інтерфейсу [Електронний ресурс]. – режим доступу: URL: <https://docs.oracle.com/javase/8/docs/technotes/guides/awt/> (дата звернення – 01.06.2023).
23. Oracle: Документація класу AffineTransform, бібліотеки java.awt.geom [Електронний ресурс]. – режим доступу: URL:

<https://docs.oracle.com/javase/8/docs/api/java/awt/geom/AffineTransform.html>

(дата звернення – 01.06.2023).

24. Oracle: Встановлення та налаштування JDK на комп'ютер користувача

[Електронний ресурс]. – режим доступу: URL:

[https://www.oracle.com/java/technologies/javase/jdk17-archive-](https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html)

[downloads.html](https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html) (дата звернення – 01.06.2023).

25. Google Drive: Завантаження додатку на комп'ютер користувача

[Електронний ресурс]. – режим доступу: URL:

[https://drive.google.com/drive/u/1/folders/1a-](https://drive.google.com/drive/u/1/folders/1a-2OgRul7bzqMJK4diaeqb0JUCZrgAAX)

[2OgRul7bzqMJK4diaeqb0JUCZrgAAX](https://drive.google.com/drive/u/1/folders/1a-2OgRul7bzqMJK4diaeqb0JUCZrgAAX). (дата звернення – 01.06.2023).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Факультет комп'ютерних наук

Кафедра теоретичної та прикладної системотехніки

Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр

Галузь знань: 12 – Інформаційні технології.

Спеціальність 123 – Комп'ютерна інженерія.

ЗАТВЕРДЖУЮ

**Завідувач кафедри теоретичної
та прикладної системотехніки**

д.т.н., проф. Шматков С. І.

«17» листопада 2022 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Веремієнка Богдана Олександрівна

1. Тема роботи «Метод афінних перетворень в комп'ютерних системах обробки графічної інформації»

керівник роботи Чуб Ольга Ігорівна, к.е.н, доц., доцент кафедри теоретичної та прикладної системотехніки.

затверджені наказом по університету від «23» травня 2023 року № 4101-5/895

2. Строк подання студентом роботи 26 травня 2023

3. Перелік питань, які потрібно розробити :

1. Дослідження математичних основ методу афінного перетворення.
2. Вивчення особливостей афінних перетворень на площині.
3. Вивчення особливостей афінних перетворень у просторі.
4. Використання методу у комп'ютерних системах.
5. Аналіз існуючих програмних рішень.
6. Вибір технологій для розробки.
7. Розробка програмної моделі.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми роботи	Жовтень 2022
2	Аналіз та пошук методичної літератури	Листопад 2022- Грудень 2022
3	Реалізація теоретичного матеріалу	Січень 2023
4	Розробка структури майбутньої програми	Січень - Лютий 2023
5	Реалізація графічного модулю, який включає в себе інтерфейс для роботи з користувачем	Березень 2023- Квітень 2023
6	Оформлення пояснювальної записки	Квітень-травень 2023
7	Перед захист кваліфікаційної роботи	Травень 2023
8	Представлення кваліфікаційної роботи керівнику та рецензенту	Травень 2023

5. Дата видачі завдання 19 жовтня 2022

Студент


 підпис
Веремієнко Б. О.

ініціали, прізвище

Керівник роботи


 підпис
Чуб О. І.

ініціали, прізвище

Додаток Б

**Технічне завдання
на розробку програмного виробу
«Метод афінних перетворень в комп'ютерних системах обробки графічної
інформації»**

Назва розділу	Назва та зміст підрозділу
1. Вступ	1.1. Назва програмного виробу: Метод афінних перетворень в комп'ютерних системах обробки графічної інформації. 1.2. Галузь застосування: комп'ютерна графіка.
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – „Комп'ютерна інженерія”. 2.2. Завдання на кваліфікаційну роботу студента (бакалавра) із наказом №4101-5/895 23.05.2023.
3. Призначення розробки	3.1. Мета розробки програмного виробу: створення графічних дизайнів, що дозволяють дизайнерам використовувати метод афінних перетворень для масштабування, зсуву та обертання об'єктів на дизайні. 3.2. Призначення програмного виробу: редагування зображень. 3.3. Вихідні дані для розробки: математична теорія, програмування, комп'ютерна графіка, розробка інтерфейсу користувача, тестування.
4. Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: інтерактивний інтерфейс, підтримка різних методів афінних перетворень, можливість редагування зображень, можливість зберігати зображення, простота використання. 4.2. Вимоги до надійності : надійне збереження даних, відповідність стандартам безпеки. 4.3. Вимоги до умов експлуатації : мінімальні вимоги до апаратного забезпечення, доступність. 4.4. Вимоги до складу параметрів технічних засобів: можливість відображення зображень високої якості, підтримка різних форматів зображень, швидкодія. 4.5. Вимоги до інформаційної та програмної сумісності: сумісність з різними операційними системами, сумісність з різними версіями операційних систем, сумісність з різними форматами зображень. 4.6. Вимоги до маркіровки та упаковки не пред'являються. 4.7. Вимоги до транспортування та зберігання не пред'являються. 4.8. Спеціальні вимоги до програмного виробу не пред'являються.
5. Вимоги до програмної документації.	Програмною документацією до виробу «Метод афінних перетворень в комп'ютерних системах обробки графічної інформації» вважати: 1) Справжнє Технічне завдання на розробку програмного

	<p>виробу;</p> <p>2) Програму та методику випробувань розробленого програмного виробу.</p>
6. Техніко-економічні показники	<p>1) Орієнтовна оцінка ефективності: Розробка цього програмного виробу дозволить зменшити витрати на проектування та розробку графічних об'єктів на комп'ютері, а також знизити час на їхнє створення. При цьому, програмний виріб буде надійним та забезпечуватиме високу якість обробки графічних об'єктів. Це дозволить залучити нових клієнтів та підвищити конкурентоспроможність компанії.</p> <p>2) Терміни та витрати коштів: Терміни розробки програмного виробу можуть становити до 6 місяців. Витрати на розробку включатимуть заробітну плату розробників, придбання необхідного обладнання та програмного забезпечення. Вартість розробки може сягати від \$50,000 до \$100,000.</p> <p>3) Порівняння з аналогами: На ринку існує кілька програмних виробів для роботи з афінними перетвореннями в комп'ютерній графіці, такі як "Adobe Photoshop", "CorelDRAW", "GIMP" та інші. Однак, на відміну від інших програм, "Метод афінних перетворень у комп'ютерній графіці" має спеціалізований функціонал, який дозволяє більш точно та ефективно працювати з афінними перетвореннями.</p>
7. Стадії та етапи розробки	<p>1) Аналіз вимог і дослідження: на цій стадії визначаються вимоги до програмного виробу, а також проводиться аналіз ринку та конкурентів.</p> <p>2) Проектування: на цьому етапі розробляється детальний план роботи, складається технічне завдання та визначається архітектура програмного продукту.</p> <p>3) Розробка: це етап, на якому програмісти пишуть код та тестують програмний продукт.</p> <p>4) Тестування: на цій стадії проводяться тестування програмного виробу для виявлення та виправлення помилок та недоліків.</p> <p>5) Випробування та валідація: на цьому етапі перевіряється, чи задовольняє програмний продукт вимоги та специфікації, які були визначені на початкових етапах розробки.</p> <p>6) Впровадження та підтримка: це завершальний етап, на якому вирішуються питання пов'язані з розгортанням програмного виробу в робоче середовище, його підтримкою та забезпеченням безперебійної роботи.</p>
8. Порядок контролю та приймання	<p>1) Перевірку ходу розробки програмного виробу Керівнику робіт виконувати 1 раз на 3 тижні.</p> <p>2) Випробування програмного виробу відповідно до Програми та методики випробувань провести на базі комп'ютерного класу.</p> <p>3) Захист розробленого програмного виробу провести на засіданні ДЕК.</p> <p>4) Пояснювальну записку подати на паперових носіях в одному екземплярі, в електронному вигляді – на CD-диску в одному екземплярі.</p>

Виконавець Веремієнко Веремієнко Б. О.

Замовник _____ Чуб О. І.

Додаток В

Програма та методика випробувань програмного виробу «Метод афінних перетворень в комп'ютерних системах обробки графічної інформації»

1. Об'єкт випробувань

1.1 Найменування програмного виробу: Метод афінних перетворень в комп'ютерних системах обробки графічної інформації.

1.2 Область застосування: створення графічних дизайнів та ілюстрацій.

1.3 Умовне призначення розробки: ця розробка може бути корисною для фахівців у галузі комп'ютерної графіки, а також для студентів, що вивчають цю тему. Вона може бути використана для створення різноманітних графічних ефектів, реалізації анімацій, віртуальної реальності та інших графічних додатків.

2. Мета випробувань

Підтвердження відповідності функціональних та інших характеристик розробленого програмного виробу вимогам, зазначеним у Технічному завданні.

3. Загальні положення

3.1 Підстави щодо випробувань

Підставою для проведення випробувань є: 1) Наказ ХНУ про затвердження тем кваліфікаційних робіт бакалаврів; 2) Навчальний план дисципліни.

3.2 Місце та тривалість випробувань

Приймальні випробування проводяться на базі комп'ютерного класу кафедри.

Підрозділ 3.3. «Обсяг випробувань»

Приймальні випробування програмного виробу проводяться в обсязі відповідно до цієї Програми та методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться за участю Замовника, Виконавця та членів комісії, призначених для прийому курсової роботи.

4. Вимоги до програмного виробу

Вимогами вважається: інтерактивний інтерфейс, підтримка різних методів афінних перетворень, можливість редагування зображень, можливість зберігати зображення, простота використання, відповідність стандартам безпеки, мінімальні вимоги до апаратного забезпечення, доступність, підтримка різних форматів зображень.

5. Вимоги до програмної документації

Програмною документацією до виробу «Метод афінних перетворень в комп'ютерних системах обробки графічної інформації» вважати:

- 1) Справжнє Технічне завдання на розробку програмного виробу;
- 2) Програму та методику випробувань розробленого програмного виробу.

6. Засоби та порядок випробувань

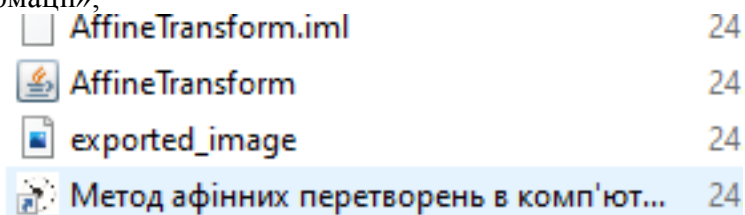
6.1 Засоби випробувань

Вимоги до складу технічних засобів: можливість відображення зображень високої якості, підтримка різних форматів зображень, швидкодія.

Випробування провести на комп'ютері з операційною системою Windows, з мінімальними характеристиками RAM 2 гб, процесор 2 ядра, відеокарта 1 гб.
Надається інсталяційна версія розробленої програми.

6.2 Порядок проведення випробувань

- 1) Якість програмної документації перевіряється на відповідність вимогам ДЕСТ 19.301-79 ЄСПД «Програма та методика випробувань».
- 2) Програма працює відповідно до умов експлуатації ОС Windows 10.
- 3) Для роботи необхідне встановлена JDK 17.
- 4) Порядок проведення випробувань:
 - Запуск програми через ярлик «Метод афінних перетворень в комп'ютерних системах обробки графічної інформації»;



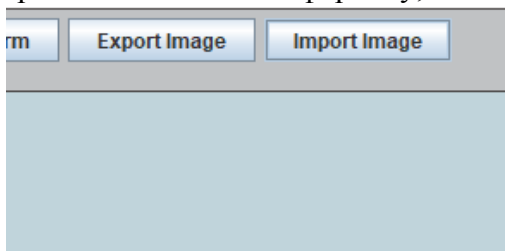
Метод афінних перетворень в комп'ютерних системах обробки графічної інформації

- Зачекати поки завантажується програма;

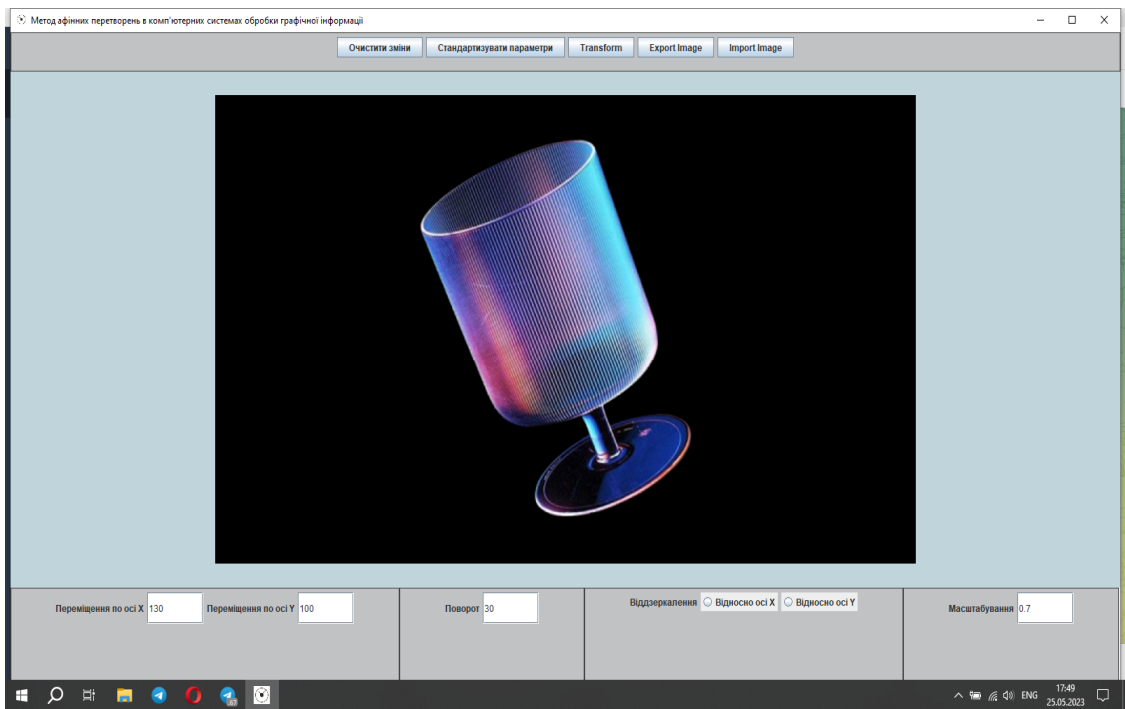
Завантаження програми...



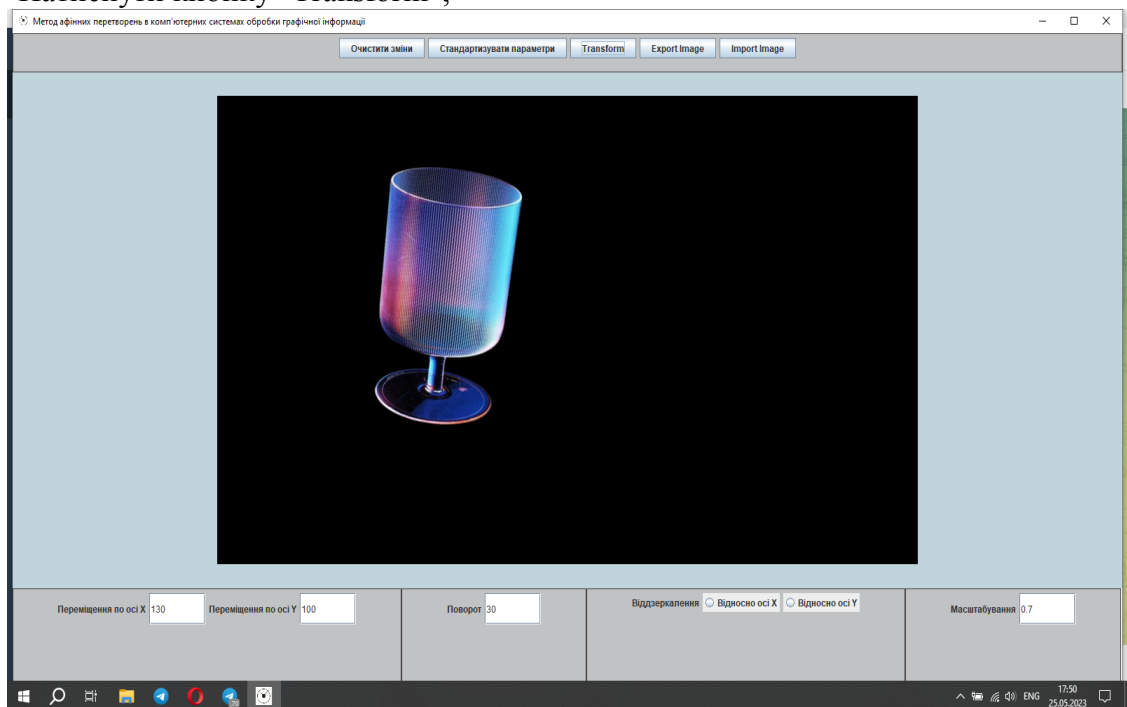
- Імпортуємо довільне зображення довільного формату;



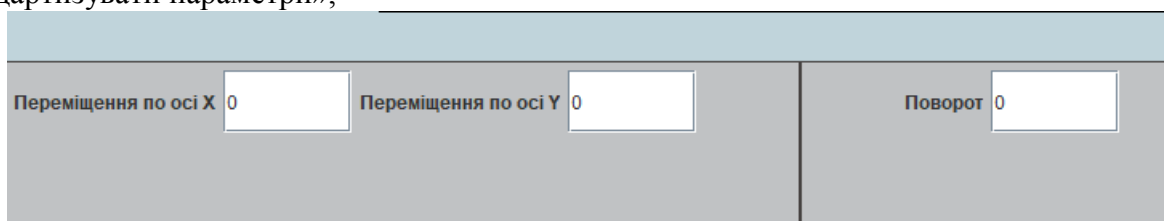
- Налаштовуємо параметри, що будуть застосовані до зображення;



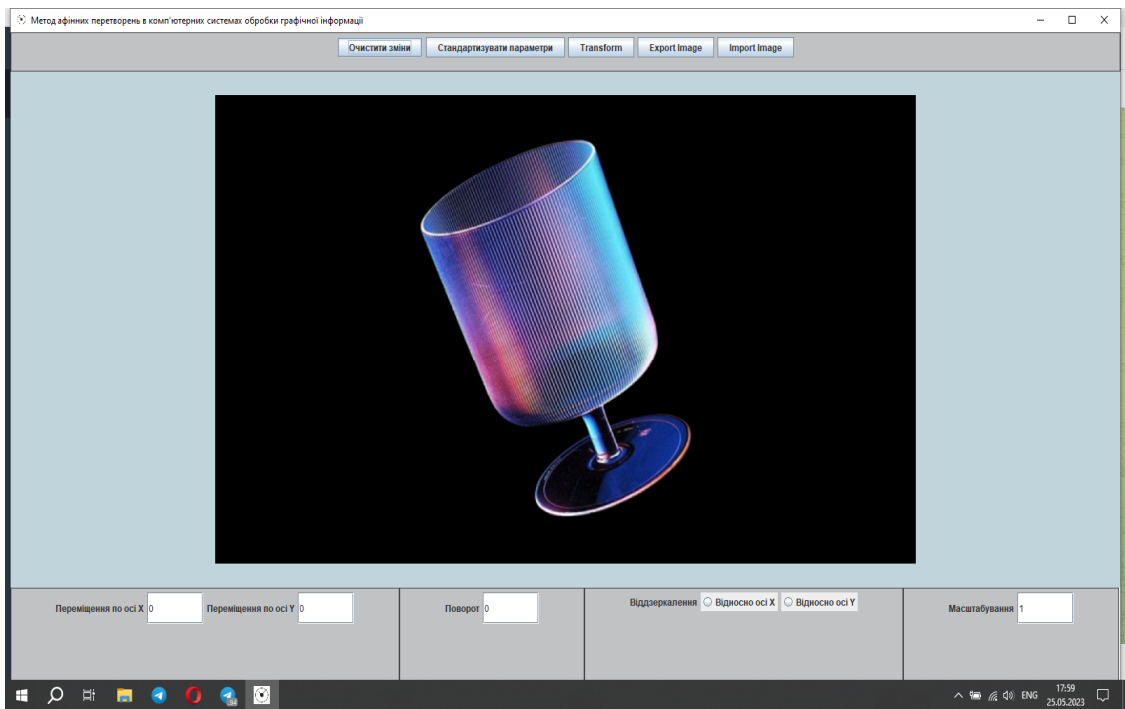
-Натиснути кнопку “Transform”;



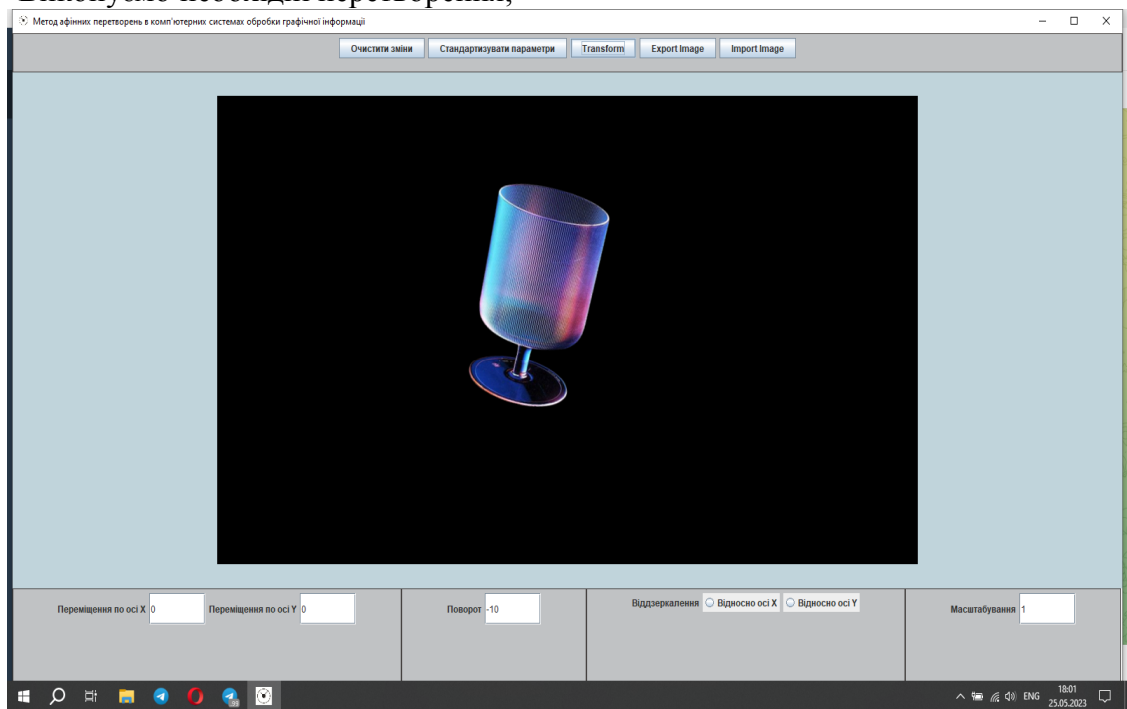
-При необхідності змінити параметри до бажаних і застосувати їх необхідну кількість разів; Для зручності, щоб скинути усі параметри до базових - натиснути кнопку «Стандартизувати параметри»;



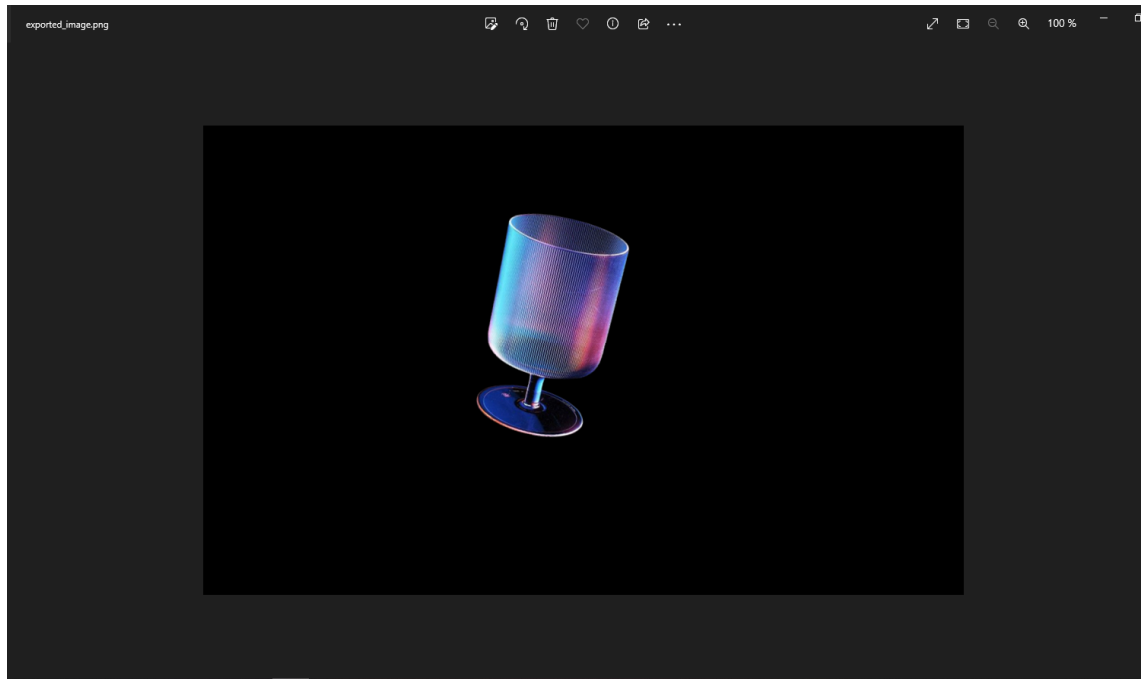
-У випадку застосування небажаних змін до нашого зображення – натиснути кнопку «Очистити зміни» для повернення початкового зображення;



-Виконуємо необхідні перетворення;



-При досягненні потрібного результату натискаємо кнопку "Export Image".
Перевіряємо результат;



5) Для проведення випробувань пропонується тест, опис якого міститься у додатку. Тест вважається пройденим, якщо не виявлено жодних помилок компілятором нашої IDE.

6) Програмний виріб вважається таким, що пройшов випробування в цілому, якщо збережено файл зображення із застосованими змінами.

Виконавець Веремієнко Веремієнко Б. О.

Лістинг 1

```

import javax.imageio.ImageIO;
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

public class AffineTransform extends JPanel {
    private BufferedImage image;
    private BufferedImage chosenImage;
    private static final JButton at2D = new JButton("Почати
роботу");
    private static JButton transformButton;
    private static final JButton cleanButton = new
JButton("Очистити зміни");
    private static final JButton cleanParamButton = new
JButton("Стандартизувати параметри");
    private static final JButton exportButton = new
JButton("Export Image");
    private static JButton importButton;
    private static final JPanel bottomPanel = new JPanel();
    private static final JPanel topPanel = new JPanel();
    private static final JPanel panel1 = new JPanel();
    private static final JPanel panel2 = new JPanel();
    private static final JPanel panel3 = new JPanel();
    private static final JPanel panel4 = new JPanel();
    private static final JLabel label1x = new JLabel("Переміщення
по осі X");
    private static final JLabel label1y = new JLabel("Переміщення
по осі Y");
    private static final JTextField textField1X = new
JTextField("0");
    private static final JTextField textField1Y = new
JTextField("0");
    private static final JLabel label2 = new JLabel("Поворот");
    private static final JLabel label3 = new
JLabel("Віддзеркалення");
    private static final JLabel label4 = new
JLabel("Масштабування");
    private static final JTextField textField2 = new
JTextField("0");
    private static final JTextField textField4 = new

```

```

JTextField("1");
    private static final JRadioButton checkBoxX = new
JRadioButton("Відносно осі X");
    private static final JRadioButton checkBoxY = new
JRadioButton("Відносно осі Y");
    public AffineTransform() {
        this.setPreferredSize(new Dimension(1000, 600));
        this.setBackground(Color.BLACK);

        importButton = new JButton("Import Image");
        transformButton = new JButton("Transform");
        transformButton.addActionListener(e -> {
            if(image != null) {
                try{
                    int x =
Integer.parseInt(textField1X.getText());
                    int y =
Integer.parseInt(textField1Y.getText());
                    double kef =
Double.parseDouble(textField4.getText());
                    int corner =
Integer.parseInt(textField2.getText());
                    applyAffineTransform(x, -y, kef, corner);

                    if(checkBoxX.isSelected() ||
checkBoxY.isSelected()) {
                        checkMethod();
                    }
                    repaint();
                }
                catch (NumberFormatException nfe){
                    nfe.printStackTrace();
                }
            }
        });
        cleanButton.addActionListener((event) -> {
            try {
                image = chosenImage;
                textField1X.setText("0");
                textField1Y.setText("0");
                textField2.setText("0");
                textField4.setText("1");
                checkBoxX.setSelected(false);
                checkBoxY.setSelected(false);
                if(image!=null){
                    repaint();
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        });
        cleanParamButton.addActionListener((event) -> {

```

```

        try {
            textField1X.setText("0");
            textField1Y.setText("0");
            textField2.setText("0");
            textField4.setText("1");
            checkBoxX.setSelected(false);
            checkBoxY.setSelected(false);
        } catch (Exception e) {
            e.printStackTrace();
        }
    });
    importButton.addActionListener((event) -> {
        JFileChooser fileChooser = new JFileChooser();
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            File selectedFile = fileChooser.getSelectedFile();
            try {
                image = ImageIO.read(selectedFile);
                chosenImage=ImageIO.read(selectedFile);
                repaint();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
    exportButton.addActionListener((event) -> {
        BufferedImage image = new
        BufferedImage(this.getWidth(), this.getHeight(),
        BufferedImage.TYPE_INT_RGB);
        Graphics2D g = image.createGraphics();
        this.printAll(g);
        g.dispose();

        try {
            ImageIO.write(image, "png", new
            File("exported_image.png"));
            System.out.println("Image exported
            successfully.");
        } catch (IOException ex) {
            System.err.println("Failed to export image: " +
            ex.getMessage());
        }
    });
}

@Override
public void paint(Graphics g) {
    if(image!=null) {
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_INTERPOLATION,
        RenderingHints.VALUE_INTERPOLATION_BICUBIC);
        super.paint(g);
    }
}

```

```

        Dimension panelSize = getSize();
        double imageWidth = image.getWidth();
        double imageHeight = image.getHeight();
        double scaleX = panelSize.getWidth() / imageWidth;
        double scaleY = panelSize.getHeight() / imageHeight;
        double scale = Math.min(scaleX, scaleY);
        int scaledWidth = (int) (imageWidth * scale);
        int scaledHeight = (int) (imageHeight * scale);

        // Определение координат отрисовываемого изображения
        int x = (int) ((panelSize.getWidth() - scaledWidth) /
2);

        int y = (int) ((panelSize.getHeight() - scaledHeight)
/ 2);

        // Отрисовка масштабированного изображения
        g2d.drawImage(image, x, y, scaledWidth, scaledHeight,
null);

        g2d.dispose();
    }
}

private void applyAffineTransform(int x, int y, double k, int
corner) {
    // Застосовуємо афінне перетворення до зображення
    java.awt.geom.AffineTransform at = new
java.awt.geom.AffineTransform();
    at.translate(x, y);
    at.rotate(Math.toRadians(corner));
    at.scale(k, k);
    image = applyAffineTransform(image, at);
}

private BufferedImage applyAffineTransform(BufferedImage
image, java.awt.geom.AffineTransform at) {
    BufferedImage newImage = new
BufferedImage(this.getWidth(), this.getHeight(), image.getType());
    Graphics2D g2d = (Graphics2D) newImage.getGraphics();
    g2d.setRenderingHint(RenderingHints.KEY_INTERPOLATION,
RenderingHints.VALUE_INTERPOLATION_BICUBIC);
    g2d.setTransform(at);

    Dimension panelSize = getSize();
    double imageWidth = image.getWidth();
    double imageHeight = image.getHeight();
    double scaleX = panelSize.getWidth() / imageWidth;
    double scaleY = panelSize.getHeight() / imageHeight;
    double scale = Math.min(scaleX, scaleY);
    int scaledWidth = (int) (imageWidth * scale);
    int scaledHeight = (int) (imageHeight * scale);

    // Определение координат отрисовываемого изображения

```

```

int x = (int) ((panelSize.getWidth() - scaledWidth) / 2);
int y = (int) ((panelSize.getHeight() - scaledHeight) /
2);
super.paint(g2d);

// Отрисовка масштабированного изображения
g2d.drawImage(image, x, y, scaledWidth, scaledHeight,
null);
g2d.dispose();

return newImage;
}

private void checkMethod() {
java.awt.geom.AffineTransform at = null;
if (checkBoxX.isSelected()) {
at = java.awt.geom.AffineTransform.getScaleInstance(1,
-1);
at.translate(0, -image.getHeight());
}
else if (checkBoxY.isSelected()) {
at = java.awt.geom.AffineTransform.getScaleInstance(-
1, 1);
at.translate(-image.getWidth(), 0);
}
image = applyAffineTransform(image, at);
}

public static void main(String[] args) {
JFrame frame = new JFrame("Метод афінних перетворень в
комп'ютерних системах обробки графічної інформації");
ImageIcon icon = new
ImageIcon("C:\\Users\\Вогдан\\AffineTransform\\src\\images\\icon.j
pg");

// Встановлення зображення іконки для фрейму
frame.setIconImage(icon.getImage());

//Встановлення розміру на весь екран
Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
int screenWidth = screenSize.width;
int screenHeight = screenSize.height;
Dimension sizeScreen= new
Dimension(screenWidth, screenHeight);
frame.setPreferredSize(sizeScreen);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

textField2.setPreferredSize(new Dimension(80,40));
textField4.setPreferredSize(new Dimension(80,40));
textField1X.setPreferredSize(new Dimension(80,40));
textField1Y.setPreferredSize(new Dimension(80,40));

```

```

AffineTransform ate = new AffineTransform();
FlowLayout fl = new FlowLayout();
Color three = Color.decode("#464241");
Border border = BorderFactory.createLineBorder(three);

JPanel panelGif = new JPanel();
panelGif.setLayout(new BorderLayout(panelGif,
BorderLayout.Y_AXIS));

JLabel labelStart = new JLabel("Завантаження
програми...");
ImageIcon gifIcon = new
ImageIcon("C:\\Users\\Богдан\\AffineTransform\\src\\images\\gif.gif");
JLabel gifLabel = new JLabel(gifIcon);

gifLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
labelStart.setAlignmentX(Component.CENTER_ALIGNMENT);
at2D.setAlignmentX(Component.CENTER_ALIGNMENT);
panelGif.add(at2D);
panelGif.add(labelStart);
panelGif.add(gifLabel);

panelGif.setBackground(Color.white);
frame.getContentPane().add(panelGif);
at2D.setVisible(false);
Timer timer = new Timer(4200, e -> {
    at2D.doClick(); // Натискання кнопки
});
timer.setRepeats(false); // Вимкнути повторення таймера

// Почати таймер при запуску програми
timer.start();

at2D.addActionListener((event) -> {
    frame.getContentPane().remove(at2D);
    frame.getContentPane().remove(panelGif);

    frame.setLayout(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.gridx = 0;
    gbc.gridy = 1;
    gbc.weightx = 1.0;
    gbc.weighty = 1.0;
    gbc.fill = GridBagConstraints.CENTER;
    frame.add(ate, gbc);
    topPanel.setLayout(fl);

    topPanel.add(cleanButton);
    topPanel.add(cleanParamButton);
    topPanel.add(transformButton);

```

```

topPanel.add(exportButton);
topPanel.add(importButton);

gbc.gridx = 0;
gbc.gridy = 0;
gbc.weightx = 1;
gbc.weighty = 0;
gbc.fill = GridBagConstraints.HORIZONTAL;
topPanel.setPreferredSize(new Dimension(1000, 50));
bottomPanel.setPreferredSize(new Dimension(1000,
150));

Color two = Color.decode("#C0C2C4");
topPanel.setBackground(two);
topPanel.setBorder(border);
frame.add(topPanel, gbc);
topPanel.getParent().setComponentZOrder(topPanel, 0);

gbc.gridx = 0;
gbc.gridy = 2;
gbc.weightx = 1;
gbc.weighty = 0;
gbc.fill = GridBagConstraints.HORIZONTAL;

bottomPanel.setLayout(new BorderLayout(bottomPanel,
BoxLayout.X_AXIS));
panel1.setBackground(two);
panel2.setBackground(two);
panel3.setBackground(two);
panel4.setBackground(two);
panel1.setBorder(border);
panel2.setBorder(border);
panel3.setBorder(border);
panel4.setBorder(border);
bottomPanel.add(panel1);
bottomPanel.add(panel2);
bottomPanel.add(panel3);
bottomPanel.add(panel4);
panel1.add(label1x);
panel1.add(textField1X);
panel1.add(label1y);
panel1.add(textField1Y);

panel2.add(label2);
panel2.add(textField2);

panel3.add(label3);
panel3.add(checkBoxX);
panel3.add(checkBoxY);

panel4.add(label4);
panel4.add(textField4);

```

```
        frame.add(bottomPanel, gbc);  
bottomPanel.getParent().setComponentZOrder(bottomPanel, 0);  
  
        frame.revalidate();  
        frame.repaint();  
    });  
  
    Color one = Color.decode("#C0D4DB");  
    frame.getContentPane().setBackground(one);  
    frame.pack();  
    frame.setLocationRelativeTo(null);  
    frame.setVisible(true);  
    }  
}
```