

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації
(повна назва)

Кафедра Радіотехнологій інформаційно-комунікаційних систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ПРОГРАМИ ДЛЯ ТРИВИМІРНИХ ОБЧИСЛЕНЬ ТА ЇХ
ВИКОРИСТАННЯ У ФІЗИЦІ

(тема)

Виконав:
студент 4 курсу, групи ІТІР-20-1

Гаронін В.В.

(прізвище, ініціали)

Спеціальність 126 Інформаційні системи
та технології

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформаційні
технології інтернету речей

(повна назва освітньої програми)

Керівник ст. викл. Штих І.А.

(посада, прізвище, ініціали)

Допускається до захисту

В.о.зав. кафедри

(підпис)

Зарудний.О.А

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти перший (бакалаврський)

Спеціальність 126 Інформаційні системи та технології
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма інформаційні системи інтернету речей
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові ГАРОНІН ВЛАДИСЛАВ ВЯЧЕСЛАВОВИЧ
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програми для тривимірних обчислень та їх використання у фізиці

затверджена наказом університету від 27 травня 2024 р. № 500 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 червня 2024 р.

3. Вихідні дані до роботи _____
Провести аналіз існуючих програм для роботи з фізичними характеристиками тривимірних об'єктів.

Розробити основні види та основні сфери застосування, а також типи програм залежно від них.

Розробити програми для обчислення основних характеристик тривимірних об'єктів та їх фізичних даних.

4. Перелік питань, що потрібно опрацювати в роботі _____
Вступ.

1. Области застосування обраного виду програм
2. Історія розвитку галузі технологій та основні споживачі
3. Складнощі при розробці та застосуванні

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
Слайди у форматі Power Point (назва та мета роботи, вступ, основні функції розробленого коду , області застосування цього типу програм та їх особливості значні імена історія області тривимірних технологій
ВИСНОВКИ

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	ст. викл. Штих І.А.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Ознайомлення із завданням. Уточнення ТЗ.	06.05.2024	виконано
2	Підбір літератури за темою роботи	26.05.–28.05.2024	виконано
3	Пошук подібних програм	28.01.–29.05.2024	виконано
4	Опис основних функцій	29.05.2024	виконано
5	Перевірка та доопрацювання внутрішніх циклів	29.05.–30.05.2024	виконано
6	Додавання фізичних характеристик	30.05.2024	виконано
7	Реалізація закінченого коду	01.06.–02.06.2024	виконано
8	Оформлення презентаційного матеріалу	08.06.–10.06.2024	виконано

Дата видачі завдання 6 травня 2024 р. С

Студент _____
(підпис)

Керівник роботи _____
(підпис)

ст. викл. Штих І.А.
(посада, прізвище, ініціали)

РЕФЕРАТ

Кваліфікаційна робота бакалавра складається з пояснювальної записки, що містить 104 сторінок тексту, 50 рисунків, 19 джерел посилання і 4 додатки.

3D, ФІГУРА, ОБЧИСЛЕННЯ, ФІЗИКА, КООРДИНАТИ

Об'єкт розробки – програми для тривимірних обчислень та їх використання у фізиці

Предмет розробки – код для обчислення математичних та фізичних характеристик тривимірних об'єктів у координатній сітці.

Метод дослідження – збір інформації про подібні програми, написання та тестування коду

Програма це код, який допомагає людині в аналізі та трансформації даних в інші форми. Як окремий випадок можна навести різні форми обчислень.

У ході роботи були розглянуті різні види програм, пов'язані з обчисленням характеристик тримірних об'єктів та їх фізичними характеристиками. На їх основі було розроблено власну програму з подібними функціями.

За рахунок виконаної роботи була створена програма здатна допомогти у швидкому обчисленні декількох типів даних, пов'язаних з тривимірними об'єктами, розташованими в одній координатній сітці.

THE ABSTRACT

The qualification work of the bachelor consists of an explanatory note containing 104 pages of text, 50 figures, 19 literary sources and 4 appendices.

3D, FIGURES, CALCULATIONS, PHYSICS, COORDINATES

The object of development is a program for calculation of characteristics of most common three-dimensional figure types.

The subject of development is a scheme (name) that interacts with the relay and automatically switches the power supply to the router.

The purpose of the work is to develop a scheme for powering the router.

A program is a piece of software that aids person in specific task utilizing input of information, its analysis and transformation into desired output. Calculations are one of the most common forms of task that programs complete for users.

In process of work many programs for calculation of both mathematical and physical characteristics of objects were analyzed as well as other similar pieces of software currently in use.

Due to the procedures described above, the result of which is the creation of a program for calculation of limited number of objects was created.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	8
ВСТУП	9
1 РОЗВИТОК ТЕХНОЛОГІЙ	10
1.1 Основні причини розвитку	10
1.2 Основні фактори стримування розвитку	11
1.3 Важливі імена історія розвитку 3D.....	12
1.4 Важливі компанії історія розвитку 3D	15
2 ОСНОВНІ СПОЖУВАЧІ ТА ВИРОБНИКИ	18
2.1 Галузі науки та виробництва	18
2.2 Компанії виробники	26
2.3 Розповсюджені програми для роботи з 3D	28
3 ПРОГРАМА ДЛЯ ОБЧИСЛЕННЯ ХАРАКТЕРИСТИК ОБ'ЄМНИХ ФІГУР ..	36
3.1 Фігури, внутрішні та зовнішні функції	36
3.2 Основні цикли	43
3.3 Перевірка даних	47
3.4 Проблеми, що виникли під час розробки.....	49
ВИСНОВКИ	57
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТКИ	59
ДОДАТОК А ПУБЛІКАЦІЯ ЗА ТЕМОЮ РОБОТИ	61
ДОДАТОК Б ПОВНИЙ КОД ПРОГРАМИ	66
ДОДАТОК В СЛАЙДИ ПРЕЗЕНТАЦІЇ.....	98
ДОДАТОК Г ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ	103

ПЕРЕЛІК СКОРОЧЕНЬ

3D – тривимірні, у трьох вимірах;

CAD (computer aided design) – дизайн за допомогою комп'ютера;

FDM (Fused Deposition Modelling) – моделювання шляхом декомпозиції
плаваючого матеріала ;

SLS – селективного лазерного спікання ;

SGC (Solid Ground Curing) – Отверждение твердого грунта;

SLA (Stereolithography) – стереолитографії;

UV (ultraviolet) – ультрафіолетовий;

АЕС – атомна електростанція;

ПЗ – програмне забезпечення;

ФПК – фотополімеризуючий композитний матеріал.

ВСТУП

Технології пов'язані з зображенням тривимірних об'єктів, а також маніпуляціями з тримірним простором є в наші дні однією з основ наукового прогресу і займають важливу роль у культурі через область розваг чи це фільми чи ігри.

Можна виділити кілька основних розділів 3D технологій, які виділяються в сучасному світі. Моделювання в тривимірному просторі та симуляція ефектів у ньому ж є на даний момент найбільш поширеними та розвиненими в наукових колах хоч і знайшли своє застосування не тільки там.

Історично тривимірні технології можна вважати дуже молодою галуззю, хоча двовимірні зображення досить швидко навчилися трансформувати в псевдо тривимірні шляхом зміни кольорів і розмірів об'єктів у двовимірному просторі створюючи відчуття глибини або поділу на шари.

Але з розвитком технологій людям були потрібні точніші методи роботи з моделями у трьох вимірах, проведення обчислень щодо тривимірних об'єктів та створення моделей їх поведінки у різних типах середовища. Це було одним з основних факторів того, чому 3D розвивалося настільки стрімко.

Но в своей работе я бы хотел рассмотреть не только то какие отрасли которые развились благодаря появлению этих технологий и более конкретный пример.

Програму для обчислення характеристик різних об'єктів у тривимірному просторі, як математичних, так і фізичних на кшталт об'єму та маси відповідно.

Цей тип програм не настільки далеко від простого калькулятора з складнішими формулами я також додатково ускладнив тим що розмістити створені об'єкти буде можливо в сітці координат, а розташування їх відносно один одного дозволить виділити деякі сумарні значення створених структур.

1 РОЗВИТОК ТЕХНОЛОГІЙ

1.1 Опис плати LM2596

Безліч причин призвело до появи попиту на технології 3D з яких можна виділити наукові, виробничі та розважальні.

Наукова причина була обґрунтована бажанням вчених скласти та простежити процеси всередині складних моделей простору, взаємодій між тривимірними об'єктами різних властивостей.

Виробнича виходить частково з наукової адже конструкторам було набагато дешевше використовувати симулятори для тестування розрахунків майбутніх елементів механізмів. Крім того 3D дозволяло їм реалізовувати куди складніші та багатовимірні креслення простіше.

Розважальна є останньою серед них після того, як вона почала свій розвиток, але мало хто може переоцінити її важливість у збільшенні попиту та швидкості розвитку цих технологій як наслідок.

Якщо трохи заглибитися в історичний аспект теми можна відзначити кілька важливих точок через які пройшли 3D технології (рисунок 1.1).

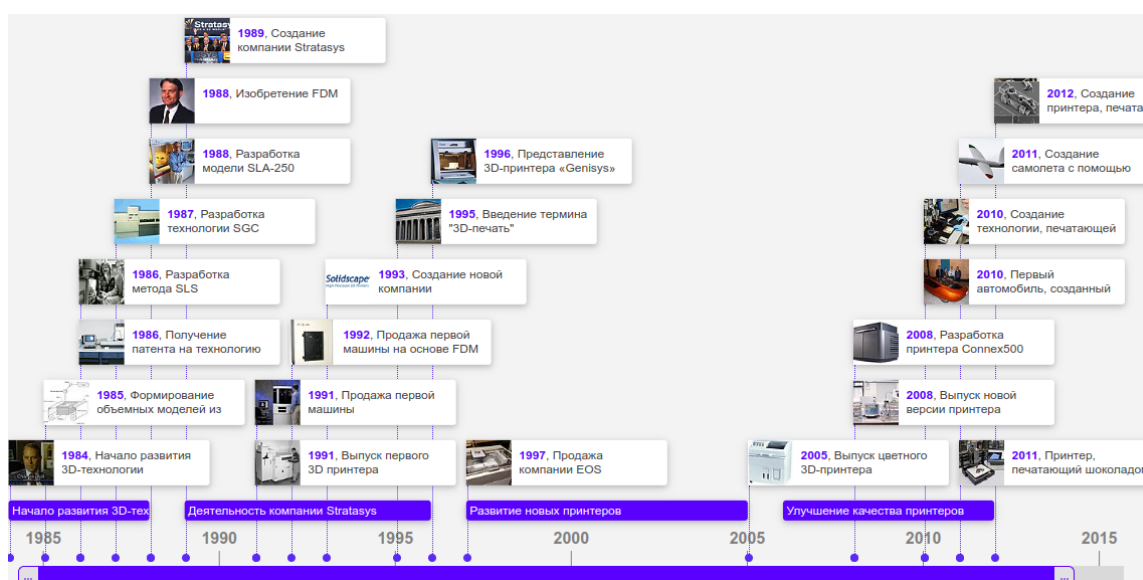


Рисунок 1.1 — Історичний шлях 3D технологій

Першою можна назвати початок з розвитку в 1984 році. Деякі відкриття роками пізніше спростили завдання по роботі з тривимірними моделями і також наблизили створення перших 3D принтерів, що відбулося в 1991 році.

Після цього розвиток технологій пов'язаних з використанням 3D прискорився, адже багато хто побачив у перших паростках майбутньої відросли великий потенціал для покращення наявних процесів.

Таким чином до початку другого десятиліття двадцять першого століття люди змогли побачити роздруковані 3D принтерами машини та літаки, а багато тривимірних симуляторів фізичних законів дозволили вченим проводити тести не витрачаючи реальних ресурсів, крім часу та електрики.

1.2 Основні фактори стримування розвитку

Крім причин розвитку в 3D були проблеми в основному пов'язані з різними технічними обмеженнями комп'ютерів. Одним із основних факторів зі списку була банальна якість зображення, яка могла дозволити собі екрани.

Відображення складних і глибоких зображень вимагало високого розв'язання, що зробило ці технології спочатку далеко не настільки доступними для широкої публіки.

Іншим фактором були потужність і обсяги пам'яті, які були потрібні для обробки і зберігання даних, що багаторазово збільшувалися при створенні тривимірних моделей, особливо якщо вони були динамічними. Хоча ця проблема є і загальною для всіх машин, що працюють з аналізом даних, але ті комп'ютери, які виконували саме подібні завдання, показували ці проблеми більш яскраво.

1.3 Важливі імена історія розвитку 3D

Як додаток до історичних довідок і факторів, що впливали на розвиток цієї галузі, хотілося б згадати і людей, чий відкриття та життя вплинули на те, що ми називаємо 3D технологіями сучасності.

Піонером 3D-моделювання можна назвати американського вченого Айвена Сазерленда (рисунок 1.2), який працював у галузі інформатики. Ще будучи аспірантом, він створив програму SketchPad, яка стала прообразом сучасних програм 3D-моделювання.

Цей учений народився 16 травня 1938 року в Хейстінгсі, Небраска. Отримав премію Тьюринга від ACM в 1988 році за створення «Sketchpad» — прообразу майбутніх САПР, що має ранній прототип графічного інтерфейсу. Одночасно вперше застосував об'єктно-орієнтований підхід до програмування. У 2012 році був удостоєний премії Кіото за ту ж розробку.

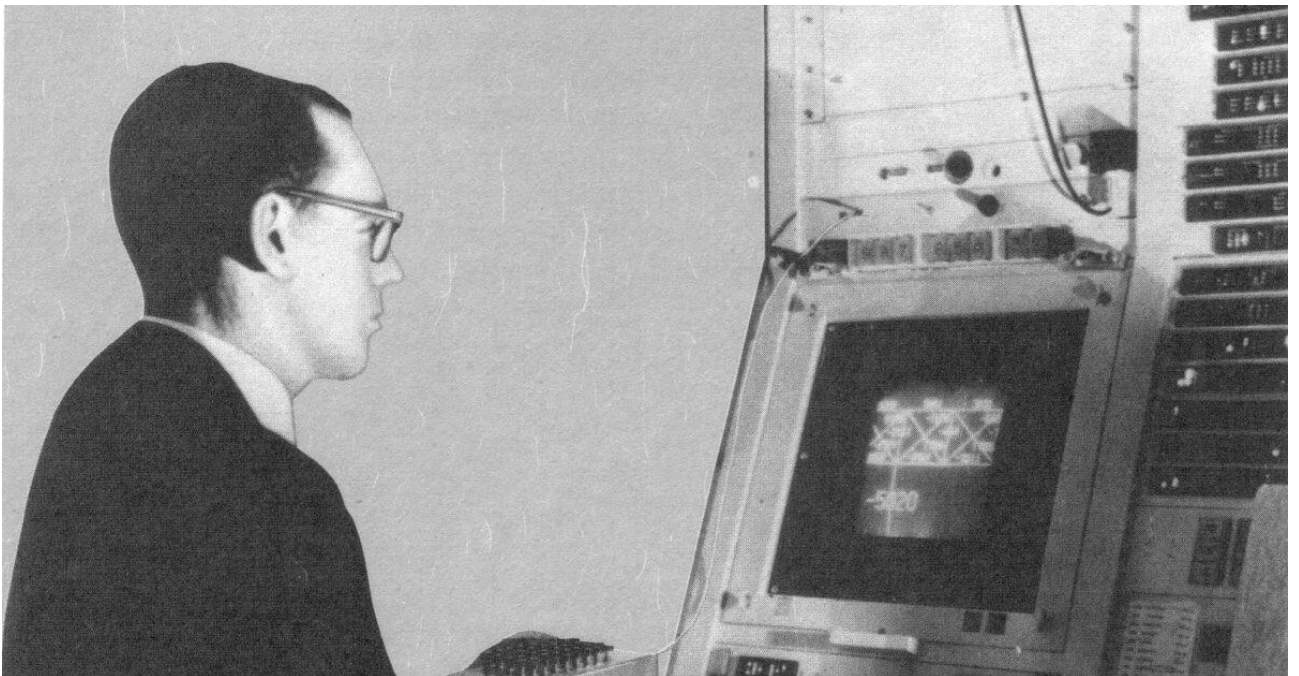


Рисунок 1.2 — Айвен Сазерленд

Сазерленд мав безліч учнів, які зробили величезний внесок у розвиток комп'ютерної графіки. Наприклад, Буй Тонг Фонг створив емпіричну модель розрахунку освітлення тривимірних об'єктів, яка так і називається Phong reflection model. Згодом її удосконалив ще один студент Сазерленда – Джим Блін. Зараз налаштування параметрів матеріалів «Блінн» можна побачити в 3Ds Max та інших програмах моделювання.

Інші студенти Сазерленда також вписали свої імена в історію 3D. Наприклад, Анрі Гуро винайшов метод затінення полігональних моделей, який дозволив імітувати гладкі поверхні, а Ед Катмулл став одним із засновників легендарної студії Pixar, яка випустила перший повнометражний комп'ютерний мультфільм «Історія іграшок».

С. Скотт Крамп (рисунок 1.3) є винахідником моделювання методом наплавлення (FDM). Він розробив технологію швидкого прототипування, яка дозволяє створювати тривимірні об'єкти за допомогою 3D-принтерів. Метод моделювання методом наплавлення полягає у використанні еластичної, воскоподібної або пластмасової нитки, яка твердне, створюючи тривимірні об'єкти із заданим дизайном. Ця технологія дозволяє створювати різнокольорові предмети за допомогою 3D-принтерів.

Чарльз Халл (рисунок 1.4) - співзасновник, виконавчий віце-президент та головний директор з технологій компанії 3D Systems. 1984 року запатентував апарат для стереолітографії. По праву вважається одним із стовпів цієї галузі технологій і той, що винайшов SLA, а також пристосував для застосування в 3D друку ФПК.

Він народився 12 травня 1939 року в Кліфтоні, штат Колорадо. Премію Ранку було отримано ним у 1995 році. У 2017 році ця людина була удостоєна Вашингтонської премії Також у 2023 році вона отримала Національну медаль США в галузі технологій та інновацій.

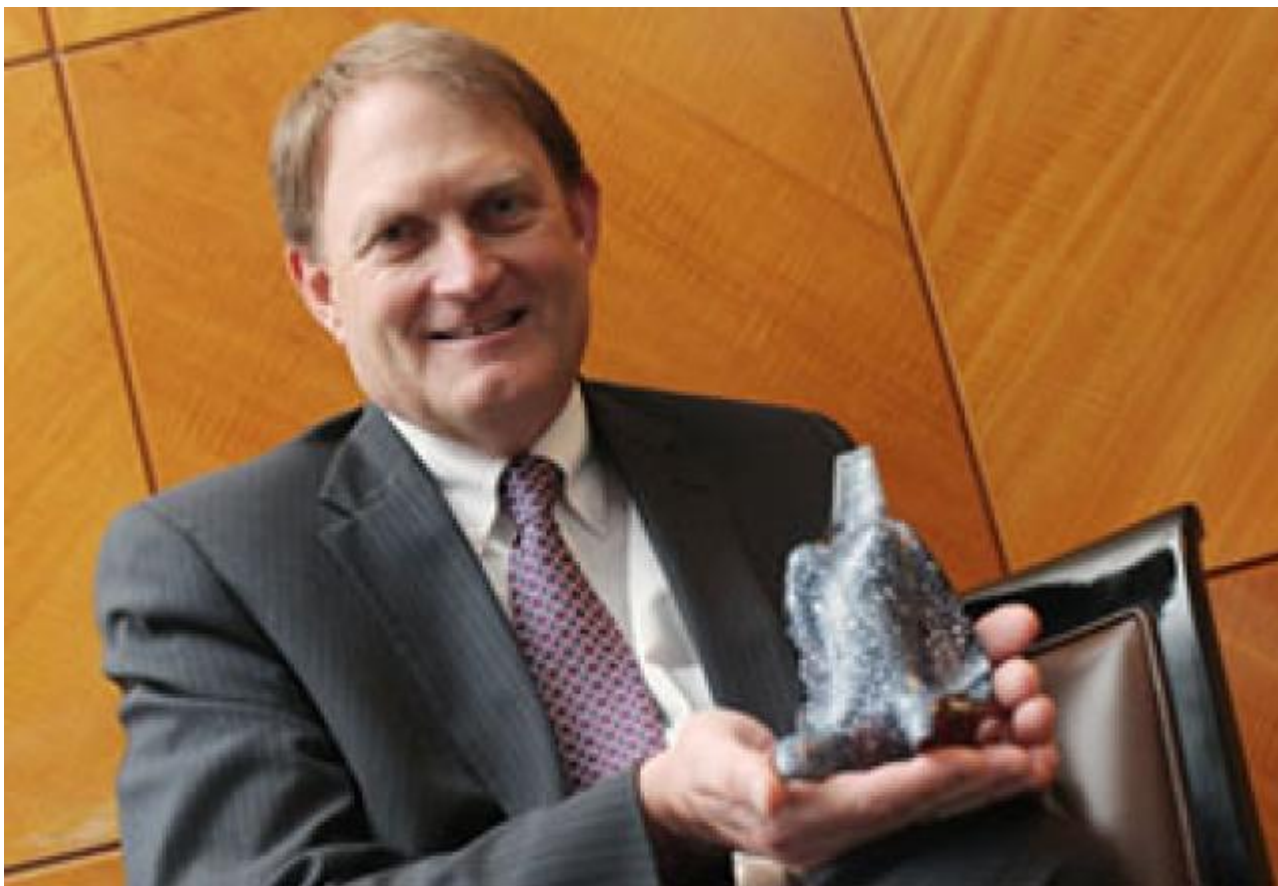


Рисунок 1.3 — С. Скотт Крамп

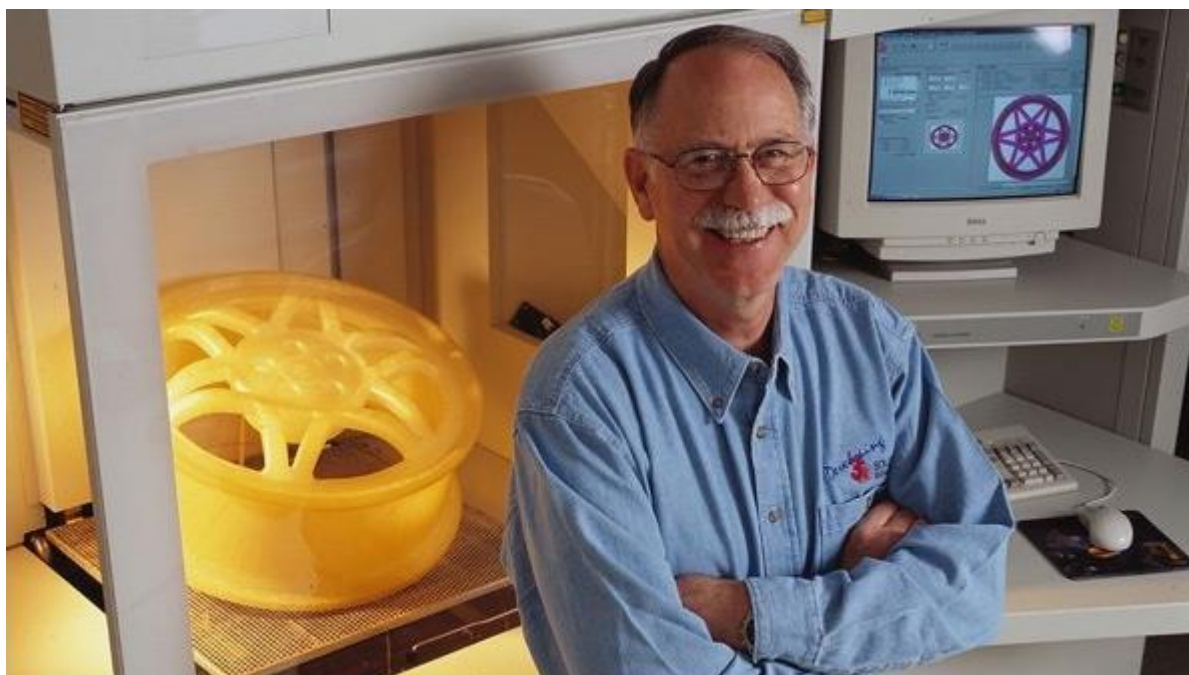


Рисунок 1.4 — Чарльз Халл

1.4 Важливі компанії історія розвитку3D

Є безліч компаній чия робота стала основою для того, як у сучасний час виглядає галузь 3D технологій. Серед них виділяються кілька що привернули більшу увагу людей до цих технологій, показали тим, хто міг би вкласти гроші в розробки, то чого можна домогтися за допомогою цих програм і пристроїв.

У цьому розділі я хотів би розглянути ті з компаній, що залишили найбільш значний слід в історії 3D технологій, а також те, як вони це зробили.

Створена в 1994 році Z Corporation (рисунок 1.5) була першою компанією, що випустила кольоровий 3D-принтер з високою якістю кольорового друку. Spectrum Z510 (рисунок 1.6) став одним із перших кроків до більш масового застосування принтерів для створення тривимірних об'єктів.



Рисунок 1.5 — Логотип Z Corporation

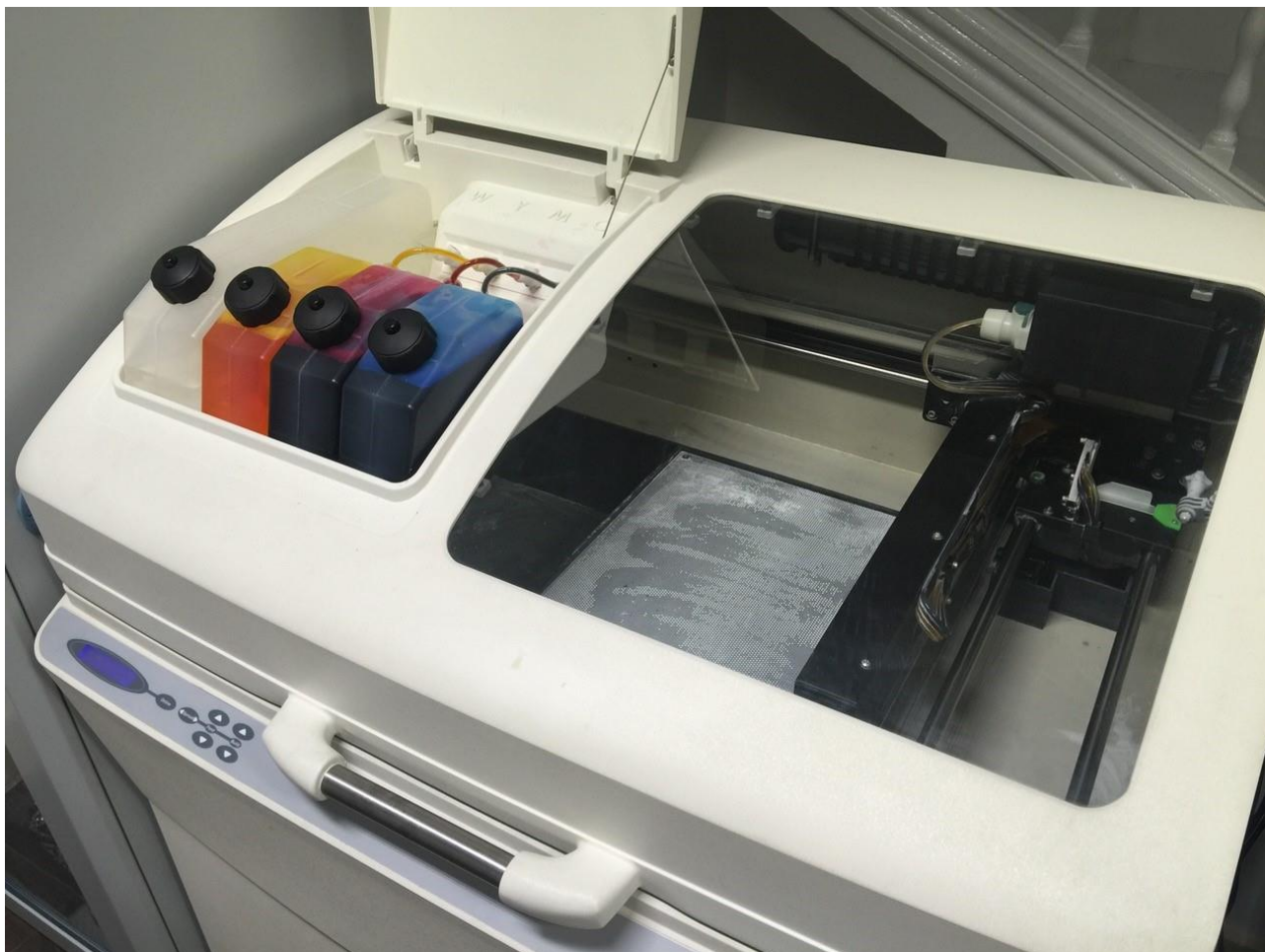


Рисунок 1.6 — Принтер Spectrum Z510

Компанія, що вписала свою назву в історію 3D випуском першого 3D-принтера з можливістю використання відразу декількох різних матеріалів для друку об'єктів (рисунок 1.7).



Рисунок 1.7 — Логотип Objet Geometries

Можна сказати що це відкрило нові горизонти для області, адже подібні принтери дозволяли робити не просто міцні та якісні об'єкти певних форм, а й набагато складніші структури, що працюють шляхом взаємодії матеріалів, що мають різні властивості.

Компанія з медичної галузі що змогла створити принтер здатний до друку кровоносних судин, чим зробив значний крок у медицині, а також зокрема показала частину потенціалу галузі у майбутньому залучаючи інвесторів (рисунок 1.8).



Рисунок 1.8 — Логотип Organovo

Підсумовуючи певний підсумок цього розділу можна сказати що ця область, як і багато хто, навряд би існувала без спільних зусиль багатьох людей і груп зацікавлених у її розвитку. У наступній частині я хотів би детальніше розглянути саме те, яке застосування знайшли створені технології в різних галузях промисловості і то ким, а також які інструменти були створені для роботи з 3D.

Дані для цього розділу були взяті із джерел [3], [2], [15], [16], [17].

2 ОСНОВНІ СПОЖУВАЧІ ТА ВИРОБНИКИ

2.1 Галузі науки та виробництва

Є безліч галузей науки і виробництва, де 3D технології займають важливу роль (рисунок 2.1). У цьому розділі я хотів би розглянути кілька з них, доповнивши інформацію про компанії, що застосовують ці технології.

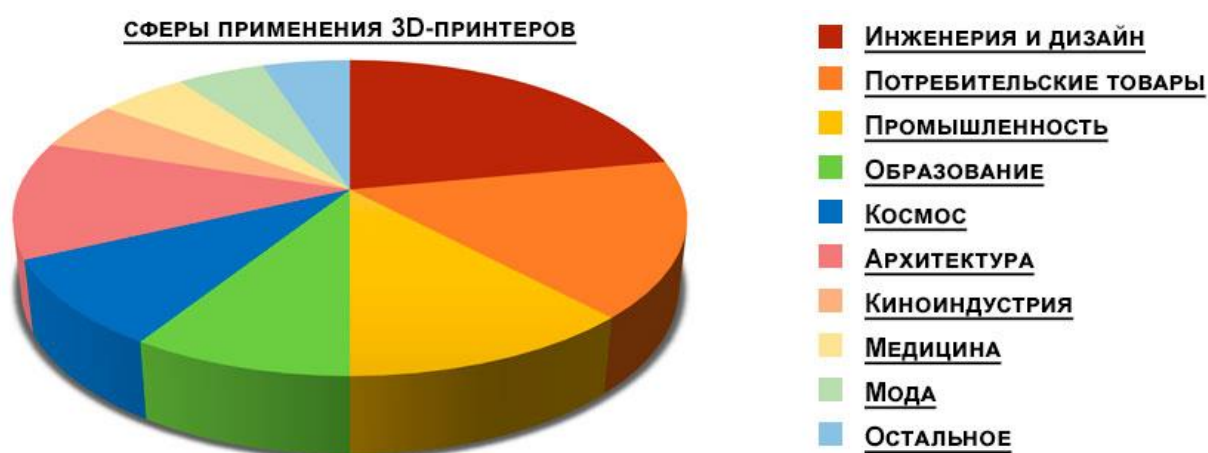


Рисунок 2.1 — Области застосування 3D технологій

У космічній промисловості та авіації використання 3D моделювання для створення деталей та компонентів космічних та авіаційних систем дозволяє інженерам швидко розробляти та тестувати нові конструкції. За допомогою адитивних технологій можна створювати складні та легкі компоненти, які важко чи неможливо виготовити традиційними методами.

3D-технології дозволяють машинобудівним підприємствам скоротити та спростити технологічний процес (рисунок 2.2) та оптимізувати витрати на виробництво.

Завдяки застосуванню 3D-сканерів та ПЗ для реверс-інжинірингу та контролю геометрії виробів у машинобудуванні витрати часу та засобів скорочуються в середньому в 1,5 рази. Серед завдань, які вирішуються за допомогою 3D-друку в машинобудуванні: прототипування створення майстер-

моделей для лиття виготовлення оснастки виготовлення функціональних деталей для агрегатів та вузлів створення складних конструкцій, зокрема цілісних, які раніше збиралися з багатьох елементів проведення технологічних експериментів.



Рисунок 2.2 — Застосування 3D технологій у промисловості

Адитивні технології в енергетичній промисловості ефективно застосовуються для створення як високоточних прототипів, так і складних кінцевих деталей та конструктивних елементів енергетичного обладнання та систем енергопостачання.

Головні переваги 3D-технологій для енергетики: можливість проектувати та виготовляти вироби найскладнішої геометрії, покращувати характеристики та знижувати вагу деталей використовувати інноваційні, екологічно безпечні матеріали Контроль якості та зворотне проектування продукції енергетичного сектора за допомогою 3D-сканера та спеціального

програмного забезпечення дозволяє прискорити та оптимізувати виробничий процес.

У медицині 3D технології дозволяють створювати на замовлення імплантати, такі як кісткові гвинти, зубні коронки та протези, які точно відповідають анатомії пацієнта. 3D принтер застосовується у сфері моделювання органів та тканин (рисунок 2.3). Можна створювати тривимірні моделі органів та тканин для навчання хірургів, планування складних операцій та освітніх цілей.

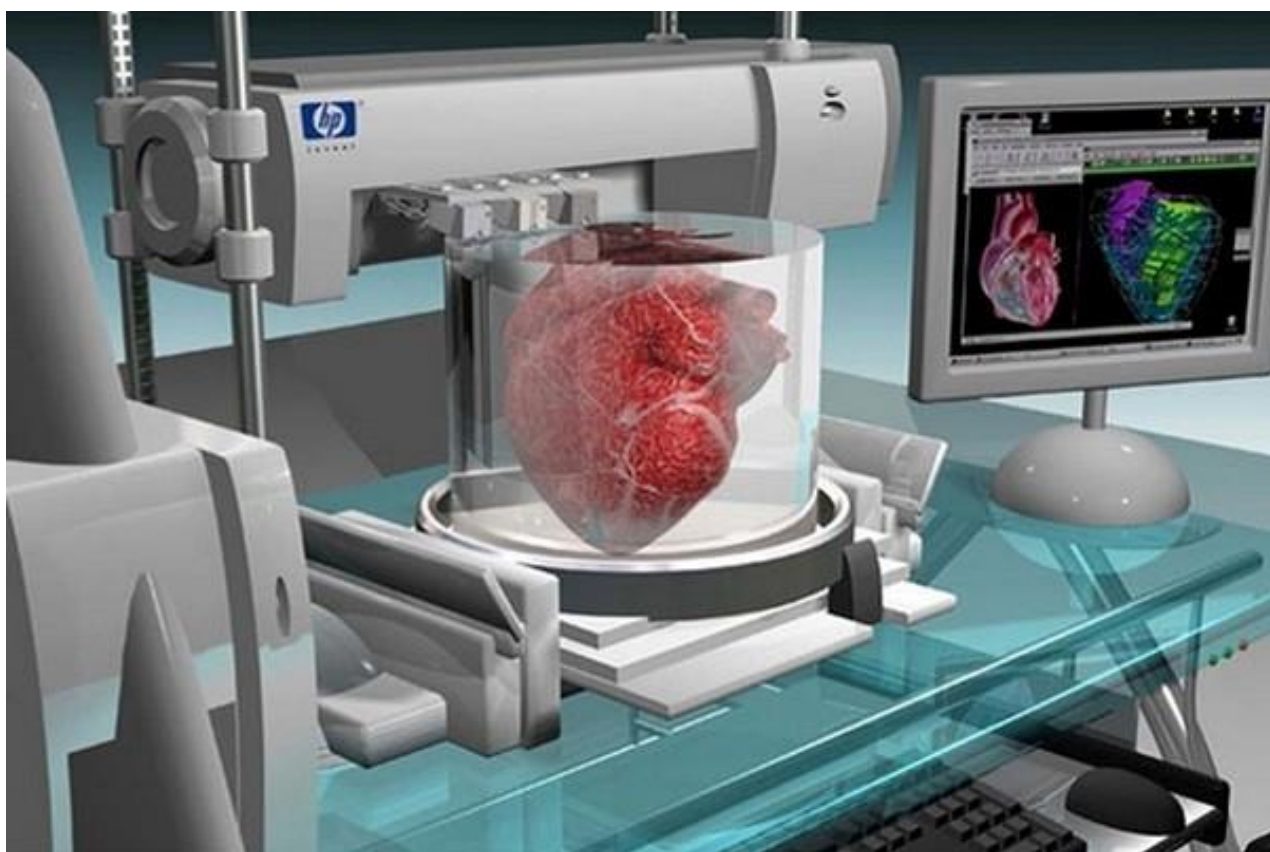


Рисунок 2.3 — Застосування 3D технологій у медицині



Рисунок 2.4 — Застосування 3D технологій у ювелірній промисловості

Ювеліри використовують 3D моделювання ювелірних виробів (рисунок 2.4). Це дозволяє їм легко експериментувати з дизайном, вносити зміни та бачити, як виглядатиме виріб у реальності, перш ніж вони почнуть його створювати. Ювеліри можуть легко створювати персоналізовані вироби для своїх клієнтів. Замовники можуть вибирати дизайн, матеріали та каміння, щоб отримати унікальний виріб.

У високотехнологічних галузях, до яких належить атомна промисловість, 3D-технології незамінні для створення унікальних продуктів та підвищення конкурентоспроможності. 3D-принтер стане таким, що не має аналогів, рішенням для виготовлення виробів складної геометрії – таких, як елементи обладнання та запасні частини для АЕС, деталі тепловіділяючих збірок, елементи автоматики системи аварійного захисту реактора та обладнання перевантаження ядерного палива. 3D-сканування та 3D-моделювання вигідно застосовувати для зворотного проектування та контролю геометрії виробів, а також для вимірювання великих об'єктів, наприклад, контуру реактора АЕС.

Моделювання дозволяє студентам досліджувати складні об'єкти у тривимірному просторі, покращуючи їх розуміння та засвоєння знань. Навчальні заклади можуть створювати віртуальні тури музеями (рисунок 2.5), історичними місцями та іншими локаціями, надаючи студентам можливість занурення в унікальні місця без фізичної присутності. У медичному навчанні 3D моделювання дозволяє студентам вивчати людське тіло в деталях, допомагаючи розумінню анатомії та процесів усередині організму.



Рисунок 2.5 — Застосування 3D технологій в освіті

3D-технології відіграють важливу роль у збереженні культурної спадщини, оскільки допомагають швидко, якісно та з високою точністю відреставрувати чи відтворити історичні цінності та витвори мистецтва. За допомогою 3D-сканера можна отримати тривимірну модель (рисунок 2.6), а потім у програмному забезпеченні виконати зворотне проектування для відновлення пошкоджених або відтворення втрачених об'єктів або їх частин. Тривимірні математичні моделі вносяться до цифрового архіву історичної спадщини та доступні для подальшого редагування з будь-якої точки Земної кулі.

На основі створених 3D-моделей можна виконати 3D-друк копій творів мистецтва, музейних експонатів, майстер-моделей для лиття (наприклад, ювелірних виробів).



Рисунок 2.6

— Застосування 3D технологій в збереженні культури

В області створення креативного одягу та взуття активно використовується 3D-друк гнучкими полімерами. Технологія дозволяє реалізувати проект будь-якого ступеня складності, і вже високо оцінена модниками, стилістами та просто креативними людьми. Створення дизайнерських об'єктів за допомогою 3D друку – нове слово в інтер'єрному та модному дизайні. Найоригінальніша і найнезвичайніша ідея може бути реалізована в найкоротші терміни (застосування). При цьому друкується не лише одяг, а й статуетки, оригінальні світильники та навіть меблі. Особливо виділяється можливість створення оригінальної біжутерії, прикрас та аксесуарів.

Тривимірна графіка (3D, 3 Dimensions, рус. 3 виміри) – це розділ комп'ютерної графіки, що складається з прийомів та інструментів, призначених для зображення об'ємних об'єктів. Зараз тривимірна графіка набула широкого поширення візуалізації інтер'єрів та екстер'єрів, кіноіндустрії, і, напевно, найважливіше в ігровій індустрії. На зорі персональних комп'ютерів графічні ігри були двовимірними у всіх змістах: персонажі були мальованими в проєкції та рухатися могли лише по осях X та Y.

Для імітації ходьби малювалися положення ніг у різний період часу і ці кадри пускалися по колу. Однак із приходом 3D графіки світ комп'ютерних ігор зазнав змін і вийшов новий рівень.

Тривимірна графіка може бути будь-якого рівня складності: від простої тривимірної моделі, з низькою деталізацією та спрощеною формою, до більш складної моделі з опрацюванням дрібні деталі, фактури.

Процес створення повноцінної тривимірної моделі включає кілька етапів: Текстурування (рисунок 2.7) – передбачає проєктування растрових або процедурних текстур на поверхні тривимірного об'єкта у відповідно до картою UV-координат. Висвітлення (рисунок 2.7) – полягає у створенні, напрямі та

налаштуванні віртуальних джерел світла. Анімація (рисунок 2.8) – надання руху тривимірної моделі, або імітація руху серед тривимірних об'єктів. Моделювання (рисунок 2.8) - створення тривимірної математичної моделі сцени та об'єктів у ній.

Крім перерахованих вище стадій роботи з тривимірним зображенням або моделлю є ще кілька .Динамічна симуляція (у деяких випадках) – автоматичний розрахунок взаємодії частинок, твердих/м'яких тіл; Рендеринг (візуалізація) - побудована проекції відповідно до обраної фізичної моделі. Математична просторова модель перетворюється на растрову картинку. Композитинг – доопрацювання зображення; композер поєднує всі частини 3D сцени у єдине ціле.

У цьому етапі легко прослідковуються загальні риси програм до роботи з будь-якими складними об'єктами що мають кілька взаємозалежних характеристик, на відміну від багатьох є поділ середовища проживання і об'єктів хоча рідко але це трапляється й інших програмах що з 3D.

Введення таких, на стадії моделювання та налаштування освітлення або текстурування, а потім робота зі створеним середовищем і тим, що знаходиться в ній, тобто анімація, динамічна симуляція та інші.



Рисунок 2.7 — Застосування 3D в ігровій індустрії, висвітлення та текстурування

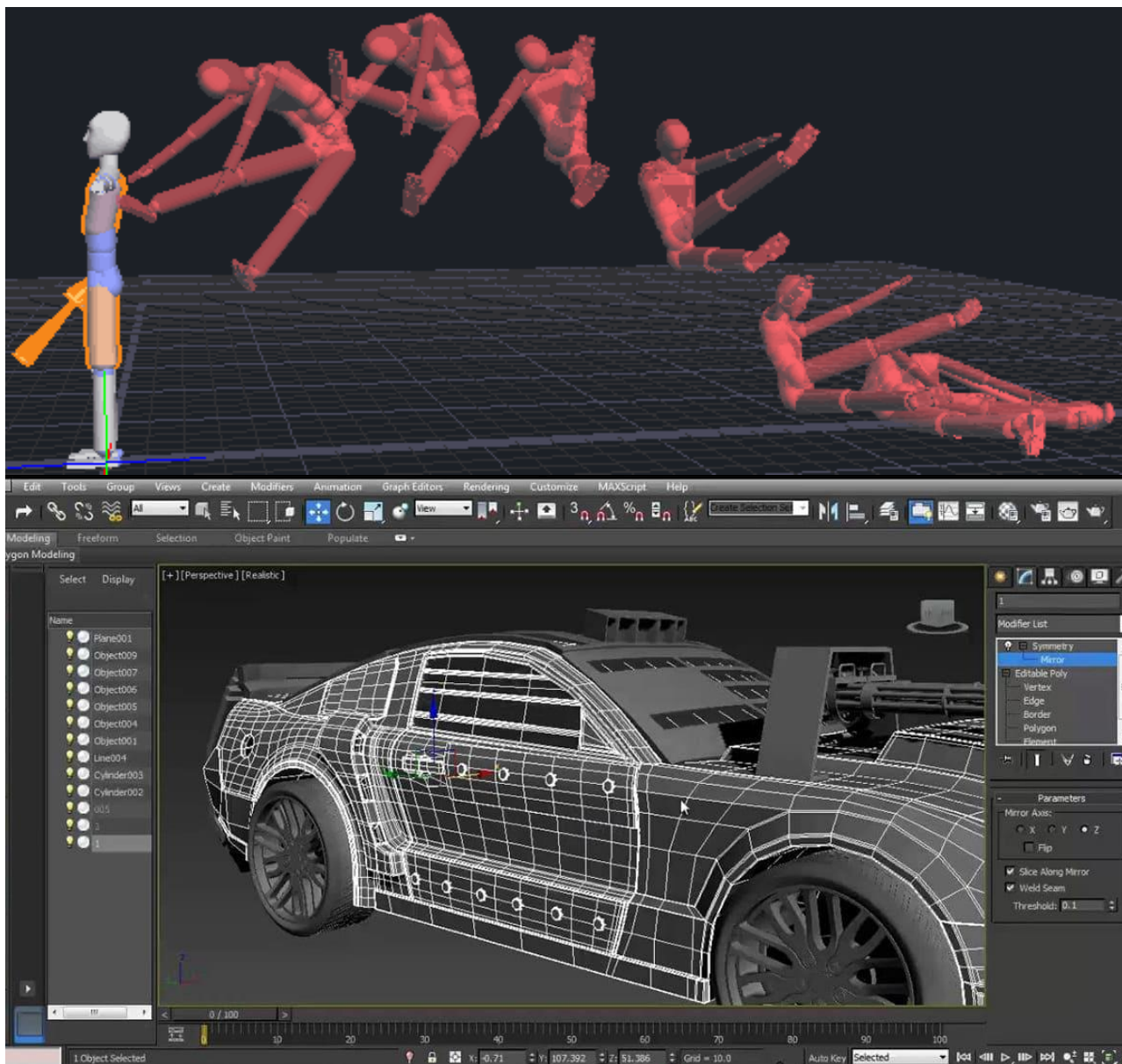


Рисунок 2.8 — Застосування 3D в ігровій індустрії, моделювання та анімація

2.2 Компанії виробники

У цьому розділі я б хотів розглянути компанії, що виробляють технічні пристрої з області 3D або займаються розробкою програмного забезпечення тієї ж спрямованості.

Stratasys (рисунок 2.9) одна з головних компаній, що займається створенням 3D принтерів як для промислових так і особистих цілей з великим вибором матеріалів, що використовуються в процесі друку. Крім цього компанія підтримує програмне забезпечення для створеної ними техніки.



Рисунок 2.9 — Логотипи Stratasys

Materialise (рисунок 2.10) є компанією чий ухил крім виробничого використання 3D також йде у бік застосування цих технологій у галузі медицини будь то використання симуляцій або друк спеціально підігнаних під особисті дані пацієнта речей що спростять процес одужання.

3D Bioprinting Solutions(рисунок 2.11) – лабораторія біотехнологічних досліджень, заснована найбільшою приватною медичною компанією у Росії ІНВІТРО. Діяльність лабораторії – розробка та виробництво біопринтерів та матеріалів у галузі тривимірного біодруку, розробка нових технологій у біофабрикації.



Рисунок 2.10 — Логотипи Materialise

Офіційно відкриття лабораторії 3D Bioprinting Solutions відбулося 6 вересня 2013 року. Співзасновниками лабораторії є один із засновників групи компаній ІНВІПРО, голова наглядової ради лабораторії біотехнологічних досліджень 3D Bioprinting Solutions, голова ради директорів групи компаній ІНВІПРО Олександр Островський та керуючий партнер 3D Bioprinting Solutions- Юсеф Хесуані.



Рисунок 2.11 — Логотипи 3D Bioprinting Solutions

Це лише кілька прикладів з безлічі компаній, що займаються розробками або виробництвом у цій галузі. Зараз майже в кожній розвиненій країні можна знайти місцеві підприємства, що займаються створенням

програмного забезпечення для тривимірного моделювання або обслуговування 3D принтерів.

2.3 Розповсюджені програми для роботи з 3D

Багато видів програм зараз працюють з 3D технологіями, від обробки об'ємних графічних моделей до зчитування та обробки поведінки складних об'єктів або створення симуляцій їх поведінки в різних середовищах, в цьому розділі я б хотів розглянути деякі з програм, що розповсюджені зараз, розібрати причини популярності тих або інших варіантом товару.

Першою я хотів би розглянути таку програму як 3D Slash. Програма для тих, хто почав освоюватись у світі тривимірного моделювання. Є дві версії – безкоштовна та преміальна, щомісячна плата за яку складає \$2. Творцям вдалося розробити інтуїтивний інтерфейс користувача. Функціонал зрозумілий та зручний, відрізняється простим освоєнням. ПЗ полегшує перехід від 3D-моделювання до друку. Для тих, хто не має 3D-принтера, передбачена можливість прямого відправлення створеної моделі в онлайн-сервіс для друку.

Як можна помітити за (рисунок 2.12) програма відмінно показує обсяг, здатна працювати з освітленням і прозорістю об'єктів, що створюються в ній, які в свою чергу дозволяють складати досить складні форми.

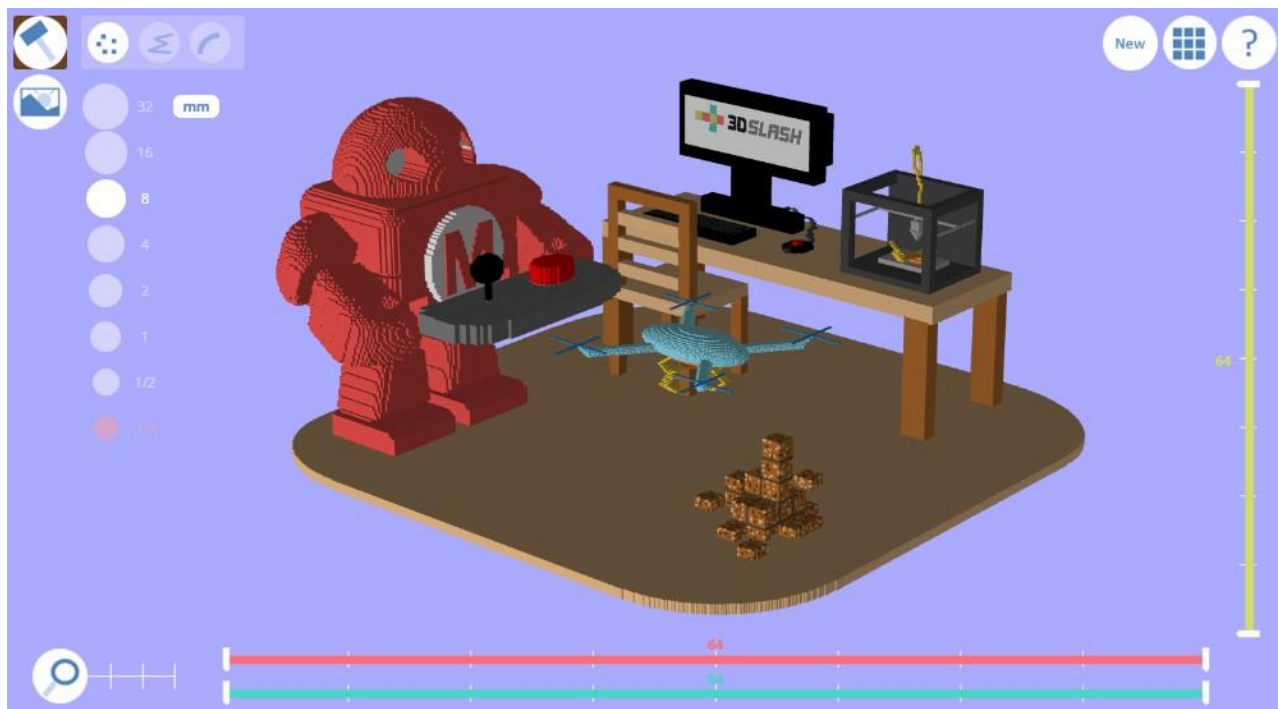


Рисунок 2.12 — Приклад роботи 3D Slash

Другим прикладом подібних програм є Tinkercad (рис. 2.13). Ця браузерна програма для моделювання служить чудовою відправною точкою для початківців та є швидким рішенням для простого моделювання. Ви можете використовувати це програмне забезпечення для створення 3D-моделей з базових форм, налаштувавши параметри для досягнення точності. Tinkercad також дозволяє створювати геометричні або векторні фігури в 2D і перетворювати їх на 3D-моделі.

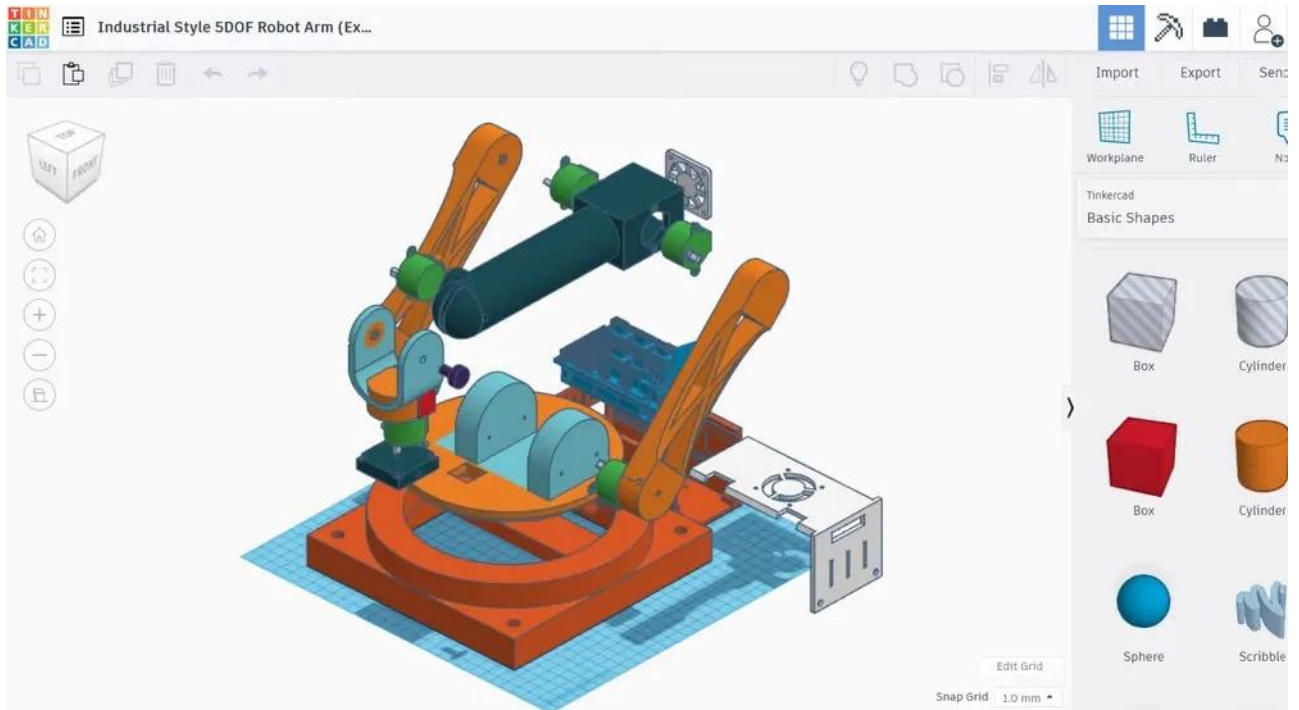


Рисунок 2.13 — Приклад роботи Tinkercad

Ще однією програмою що хотілося б розглянути ZbrushCoreMini (рисунок 2.14). Ця програма - інструмент для художнього 3D-скульптингу з функціоналом, який можна порівняти з ліпленням з глини. Він є фантастичним програмним забезпеченням для 3D-друку, якщо вашою основною метою є створення статуй або фігурок - наприклад, створення бюста вашого улюбленого персонажа з відеоігор або коміксів. Хоча це програмне забезпечення для 3D-моделювання призначене в основному для початківців та середніх користувачів, воно має широкий функціонал, який спрощує процес 3D-скульптингу.

У той час як попередні дві програми мали більш розмиту спеціальність з лише уклоном, що частково простежується, в конструкторському напрямку у випадку то ZbrushCoreMini це представник більш вузько спеціалізованих програм для людей з області мистецтва, хоча її і можна спробувати застосувати для інших цілей.



Рисунок 2.14 — Приклад роботи ZbrushCoreMini

Іншим прикладом програм цієї ж відросли, але цього разу з великим ухилом в анімацію та збирання інформації є Blender (рисунок 2.15).

Професійне безкоштовне програмне забезпечення для тривимірної анімації з відкритим вихідним кодом, Blender включає цілий ряд додатків, таких як візуальні ефекти, анімаційні фільми, відеоігри та багато інших. Blender підтримує все тривимірне дерево, яке включає такі функції, як такелаж, моделювання, анімація, симуляція, композитинг, рендеринг і відстеження руху.

Після я хотів би звернути вашу увагу на трохи інший приклад програми для взаємодії з 3D.

3DCRAFTER (рисунок 2.16) має кілька помітних відмінностей від інших раніше згаданих мною програм для роботи зі створенням моделей та стеження за їх взаємодіями.

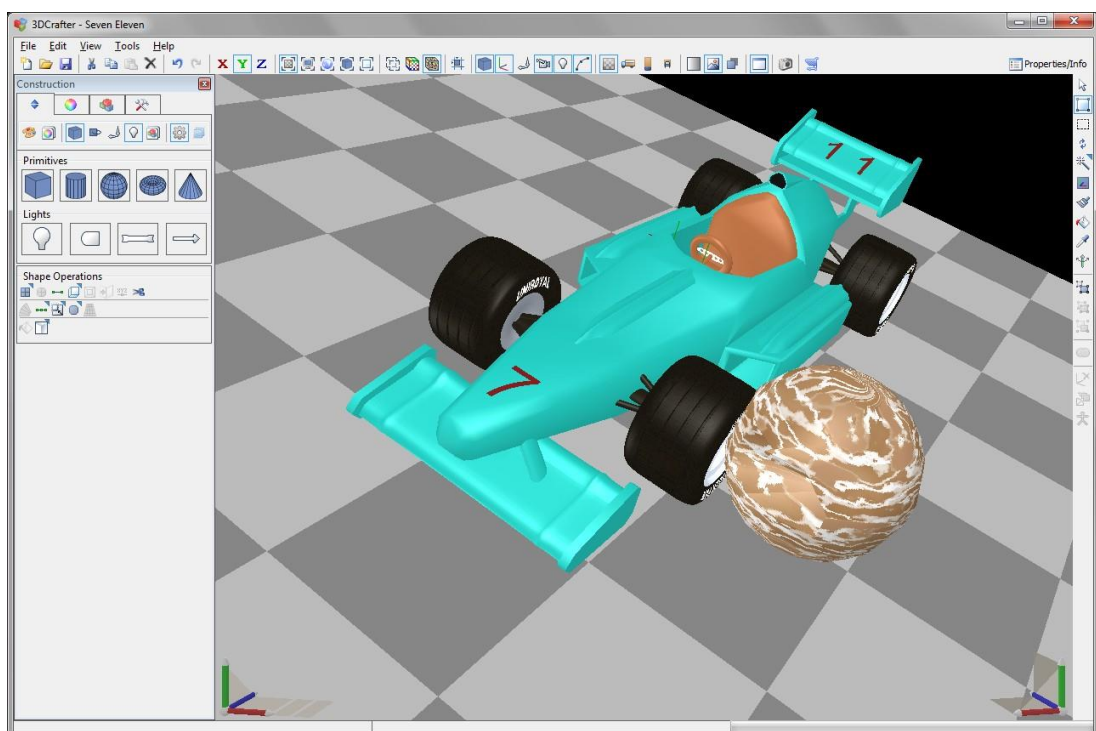
Програм дозволяє більш прямо, за допомогою миші, викривляти та тягнути об'єкти, створені в ній, покадрово відстежувати процес анімації. Крім того, у програмі можна знайти дуже багатий набір готових текстур об'єктів.

Це чудовий безкоштовний редактор для новачків, який дозволяє створювати 3D моделі в режимі реального часу та анімувати їх для проектування та мультиплікації.



Рисун
ок
2.15
—
Прикл
ад
робот
и
Blend
er

Подібні програми більш прямолінійні і менш самостійні в порівнянні з вищенаведеною, адже частіше вимагають прямої уваги для виконання завдання. Це як плюс, так і мінус подібних продуктів.



Рисун
к 2.16
—
Прикла
д
роботи
3DCRA
FTER
Щ
є одним

цікавим прикладом програм для роботи з 3D є Autodesk AutoCAD (рисунок 2.17).

Являє собою програмний комплекс для розробки креслень та проектування в MacOS та Windows. Програма призначена для професійного використання.

Ціна стартує від 210 доларів на місяць, є безкоштовна версія на 30 днів. ПЗ зручно для складних проектів, додавання позначень, зв'язування їх з координатами на місцевості.

Його застосовують у різних галузях промисловості, включаючи будівництво та машинобудування. Підходить для створення моделей для друку та різання лазером.

Цей тип програм є чудовим прикладом більш промислового та утилітарного погляду на 3D технології створення креслень, моделей та симуляторів.

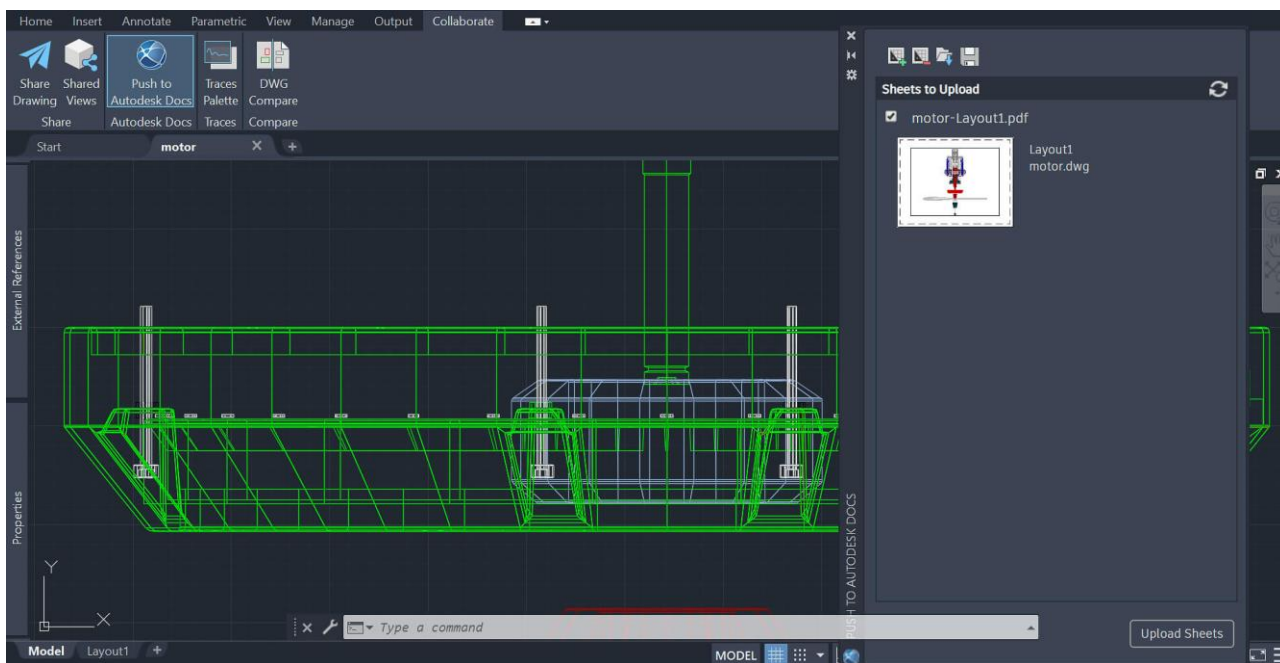


Рисунок 2.17 — Приклад роботи Autodesk AutoCAD

Ще один приклад програмного забезпечення для 3D-моделювання, яке справді виділяється, – це Vectary (рисунок 2.18).

Це дивне поєднання CAD та програмного забезпечення для 3D-моделювання, доступного у браузері. Воно більш доступне для тих, хто прийшов у 3D-дизайн, маючи досвід роботи з програмним забезпеченням для 2D-проектування, зважаючи на його унікальний інтерфейс користувача.

Vectary пропонує рівні, що масштабуються, і майже всі функції безплатний - типи файлів експорту виділяються як платна функція. На відміну від багатьох програм навіть безплатний рівень є безплатний і для комерційного використання, що особливо унікально серед розробників моделей на основі браузера.

Ця програма схоже частково на згаданий раніше ZbrushCoreMini що застосовується в області мистецтва, хоча саме Vectary краще підійде для створення композицій з безлічі об'єктів складових цілісну картинку.

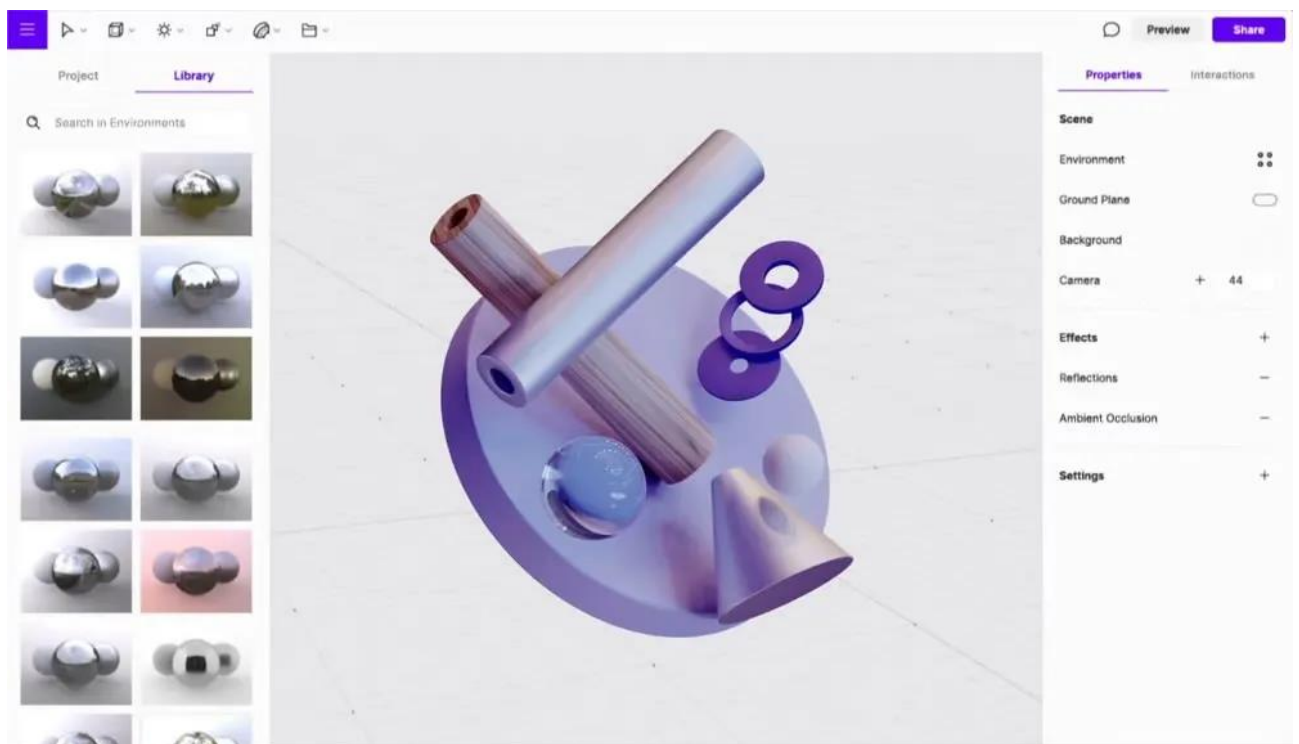


Рисунок 2.18 — Приклад роботи Vectary

Підводячи деякі підсумки цього розділу хотів би помітити що технології 3D чи то промислові принтери, чи програми для роботи з моделями,

середовищем та об'єктами проникли в більшість промислових областей, а також зайняли своє міцне місце в сучасній культурі.

Мистецтво теж змінилося принесеними новою галуззю технологіями та можливостями роботи із зображеннями та об'єктами без обробки реального матеріалу.

Як результат 3D дозволила людям втілювати та перевіряти життєздатність багатьох раніше надто затратних ідей.

Дані для цього розділу були взяті із джерел [11], [12], [13], [14].

3 ПРОГРАМА ДЛЯ ОБЧИСЛЕННЯ ХАРАКТЕРИСТИК ОБ'ЄМНИХ ФІГУР

3.1 Фігури, внутрішні та зовнішні функції

У моїй програмі як і у багатьох все починається із заготівлі фундаменту, у конкретному випадку функцій та класів із їхніми внутрішніми функціями відповідно.

Почнемо з тих типів фігур які були обрані як основні, а потім перейдемо до більш загальних функцій, які будуть потрібні куди пізніше в процесі роботи коду з введеними даними.

Однією з найпоширеніших фігур можна сміливо назвати сферу, від футбольного м'яча до моделі зірки або того, у що збирається рідина в невагомості під силою поверхневого натягу.

До коду створення класу сфери (рисунок 3.1) входить кілька загальних і кілька індивідуальних характеристик. Радіус наприклад можна побачити лише у класі сфери, півсфери, конуса та циліндра відповідно.

Найбільш загальними можна назвати масу, об'єм, плозadzь та поверхні об'єкта.

```

9  class Sphere {
10
11  public:
12      double MpV; // Mass per Volume
13      double R;
14      double Cv;
15      double Cm;
16      double Csa;
17
18      double Csa (double R){ // Calculate Surface area
19          Csa=(4.0*3.0)*3.14*pow(R,2);
20          cout<<"Surface area of the Sphere="<<Csa<<endl;
21          return Csa;
22      }
23
24      double CV (double R){ // Calculate Volume
25          Cv=3.14*pow(R,3);
26          cout<<"Volume of the Sphere="<<Cv<<endl;
27          return Cv;
28      }
29
30      double CM (double R, double MpV){ // Calculate Mass
31          Cm=3.14*pow(R,3)*MpV;
32          cout<<"Mass of the Sphere="<<Cm<<endl;
33          return Cm;
34      }

```

Рисунок 3.1 — Код програми класу сфери

Менш поширеним і складнішим у формулах обчислення є конус (рисунок 3.2) з прикладу якого видно необхідність введення додаткових вступних характеристик, наприклад висоти й бічної лінії що виражає відстань від краю основи до вершини конуса, хоча останнє можна отримати з формули радіуса і висоти.

```

39
40     double R;
41     double H;
42     double L; /// Side line
43     double MpV;
44     double Cv;
45     double Cm;
46     double Csa;
47
48     double CSa(double L, double R){
49         Csa=3.14*(L*R)+3.14*(pow(R,2));
50         cout<<"Surface area of the Cone="<<Csa<<endl;
51         return Csa;
52     }
53
54     double CV (double R, double H){
55         Cv=0.33*3.14*(pow(R,2))*H;
56         cout<<"Volume of the Cone="<<Cv<<endl;
57         return Cv;
58     }
59
60     double CM (double R, double H, double MpV){
61         Cm=0.33*3.14*(pow(R,2))*H*MpV;
62         cout<<"Mass of the Cone="<<endl;
63         return Cm;
64     }

```

Рисунок 3.2 — Код програми класу конусу

Далі за списком йде трохи простіший циліндр(рисунок 3.3). Ця фігура також використовується в багатьох технічних пристроях від двигунів і трубопроводів до різних поршнів або імплантатів, в тій чи іншій формі та масштабі природно.

```

double R;
double H;
double MpV;
double Cv;
double Cm;
double Csa;

double CSa(double R,double H){
    Csa=2*((3.14*R*R)+3.14*R*H);
    cout<<"Surface area of the Cylinder="<<Csa<<endl;
    return Csa;
}

double CV (double R,double H){
    Cv=3.14*R*R*H;
    cout<<"Volume of the Cylinder="<<Cv<<endl;
    return Cv;
}

double CM (double R,double H,double MpV){
    Cm=3.14*R*R*H*MpV;
    cout<<"Mass of the Cylinder="<<Cm<<endl;
    return Cm;
}

```

Рисунок 3.3 — Код програми класу циліндр

Куби (рисунок 3.4), як і сфери є базовим типом фігур з якого багато хто ще в школі починає знайомство з тривимірною геометрією. Прості але тому важливі вони часто є складовими частинами більш складних, сегментами більшого.

```

125 class Cube {
126
127     public:
128
129     double S1;
130     double MpV;
131     double Cv;
132     double Cm;
133     double Csa;
134
135     double CSa(double S1){
136         Csa=6*S1*S1;
137         cout<<"Surface are of the Cube="<<Csa<<endl;
138         return Csa;
139     }
140
141     double CV (double S1){
142         Cv=S1*S1*S1;
143         cout<<"Volume of the Cube="<<Cv<<endl;
144         return Cv;
145     }
146
147     double CM (double S1,double MpV){
148         Cm=S1*S1*S1*MpV;
149         cout<<"Mass of the Cube="<<Cm<<endl;
150         return Cm;
151     }

```

Рисунок 3.4 — Код програми класу куб

Кубоїд (рисунок 3.5), або паралелепіпед, складніший брат куба. Цю форму можна назвати найпоширенішою у нашому сучасному оточенні. Будівлі та машини, блоки комп'ютери та телефони, у кожному прикладі легко вгадується саме цей тип фігур. Його основною складністю є кількість вступних показників.

Рис

98	
уно	99
	100
к	101
	102
3.5	103
	104
—	105
	106
Код	107
	108
про	109
	110
гра	111
	112
ми	113
	114
кла	115
	116
су	117
	118
куб	119
	120
оїд	121
	122
	123
	124

```

double S1;
double S2;
double S3;
double MpV;
double Cv;
double Cm;
double Csa;

double Csa(double S1,double S2,double S3){
    Csa=2*((S1*S2)+(S2*S3)+(S1*S3));
    cout<<"Surface area of the a Cuboid="<<Csa<<endl;
    return Csa;
}

double CV (double S1,double S2,double S3){
    Cv=S1*S2*S3;
    cout<<"Volume of the Cuboid="<<Cv<<endl;
    return Cv;
}

double CM (double S1,double S2,double S3,double MpV){
    Cm=S1*S2*S3*MpV;
    cout<<"Mass of the Cuboid="<<Cm<<endl;
    return Cm;
}

```

о суті з цієї фігури частково виходить і лінза, що має свою окрему функцію, що буде розглянута нижче.

З кубоїду і лінзи, при деякому старанні можна скласти величезну кількість інших фігур через те, як вони мінливі порівняно з тим самим кубом, сферою або півсферою.

214	<code>class Hemisphere {</code>	Рис
215		
216	<code>public:</code>	уно
217		
218	<code>double R;</code>	к
219	<code>double MpV;</code>	
220	<code>double Cv;</code>	3.6
221	<code>double Cm;</code>	—
222	<code>double Csa;</code>	
223		
224	<code>double Csa(double R){</code>	Ко
225	<code> Csa=3*(3.14)*R*R;</code>	д
226	<code> cout<<"Surface area of the Hemisphere="<<Csa<<endl;</code>	
227	<code> return Csa;</code>	про
228	<code> }</code>	
229		гра
230	<code>double CV (double R){</code>	ми
231	<code> Cv=0.66*(3.14)*R*R*R;</code>	кла
232	<code> cout<<"Volume of the Hemisphere="<<Cv<<endl;</code>	
233	<code> return Cv;</code>	су
234	<code> }</code>	
235		
236	<code>double CM (double R, double MpV){</code>	нап
237	<code> Cm=0.66*3.14*R*R*R*MpV;</code>	івс
238	<code> cout<<"Mass of the Hemisphere="<<Cm<<endl;</code>	
239	<code> return Cm;</code>	
240	<code> }</code>	фе

ра

Одна з найбагатших на необхідні вступні характеристики є лінза (рисунок 3.7). У конкретному випадку була обрана лінза покладена на бік, через що у неї лише одна горизонтально лежача основа і потрібно введення висоти окремо від характеристик складових лінзу простих фігур граней.

Є безліч типів цієї фігури в залежності від кількості бічних граней і фігури що використовується як основа призми, через що крім даної формули можна скласти ще безліч і вони теж будуть називатися формулами об'єму або площі поверхні призми.

```

186
187     double S1;
188     double S2;
189     double S3;
190     double H;
191     double MpV;
192     double Cv;
193     double Cm;
194     double Csa;
195
196     double Csa(double S1,double S2,double S3,double H){
197         Csa=(S1*H)+(S2*S3*2)+(S1*S2);
198         cout<<"Surface area of the Prism ="<<Csa<<endl;
199         return Csa;
200     }
201
202     double CV (double S1,double S2,double H){
203         Cv=S1*H*S2*0.5;
204         cout<<"Volume of the Prism ="<<Cv<<endl;
205         return Cv;
206     }
207
208     double CM (double S1,double S2,double H,double MpV){
209         Cm=S1*H*S2*0.5*MpV;
210         cout<<"Mass of the Prism ="<<Cm<<endl;
211         return Cm;
212     }

```

Рисунок 3.7 — Код програми класу призма

Останнім класом фігури є піраміда (рисунок 3.8), яку певною мірою можна назвати похідною або окремим випадком призми. Ми можемо помітити цю фігуру в багатьох стародавніх і сучасних спорудах.

Далі за списком йдуть приклади загальних функцій (рисунок 3.9) (рисунок 3.10) всередині програми, але вже поза основними класами фігур.

Перший тип функцій це порівняння для двох, трьох або чотирьох елементів відповідно, після йде функція для обчислення потенційної енергії об'єкта за допомогою його маси та функція перезавантаження програми.

Обидві функції використовуються в програмі на стадії після створення та позначення початкових характеристик елементів, адже без них не мають особливого сенсу, як і даних, які могли б бути оброблені всередині них.

Рисун
ок 3.8
—
Код
прогр
ами
класу
пірам
іда

```

153 class Pyramid {
154
155     public:
156
157     double S1;
158     double H;
159     double S2;
160     double MpV;
161     double Cv;
162     double Cm;
163     double Csa;
164
165     double Csa(double S1,double S2,double H){
166         Csa=2*S1*S2+S1*S1;
167         cout<<"Surface area of the Pyramid="<<Csa<<endl;
168         return Csa;
169     }
170
171     double CV (double S1,double H){
172         Cv=0.33*S1*S1*H;
173
174         double comp2 (double n, double n1){
175             if ( n>n1 ){
176                 return n;
177             }
178             if ( n1>n ){
179                 return n1;
180             }
181         };
182
183         double comp3 (double n, double n1,double n2){
184             if ( n>n1 && n>n2 ){
185                 return n;
186             }
187             if ( n1>n && n1>n2 ){
188                 return n1;
189             }
190             if ( n2>n && n2>n1 ){
191                 return n2;
192             }
193         };
194
195         double comp4 (double n, double n1,double n2,double n3){
196             if ( n>n1 && n>n2 && n>n3){
197                 return n;
198             }
199             if ( n1>n && n1>n2 && n1>n3){
200                 return n1;
201             }
202             if ( n2>n && n2>n1 && n2>n3){
203                 return n2;
204             }
205             if ( n3>n && n3>n1 && n3>n2){
206                 return n3;
207             }
208         };
209
210         double En (double n){
211             double m=n;
212             double c=(3*pow(10,8));
213             double E=m*pow(c,2);
214             cout<<"Mass of the object ="<<m<<endl;
215             cout<<"Energy of the object ="<<E<<endl;
216             return E;
217         }
218
219         void restart (){
220             cout<<"Restarting the program"<<endl;
221             std::exit(42);
222         }
223     };
224 };

```

Рисунок
3.9 —
Функції
порівня
ння

Рисунок 3.10 — Функції порівняння та інші

3.2 Основні цикли

Закінчивши з тим, що ми будемо користуватися всередині коду, можна перейти і до тієї його частини, яка відповідає за отримання, обробку та видачі результатів користувачеві.

Першим йде вибір числа фігур, після ж вибір типу фігур що ми хотіли б створити разом із введенням їх характеристик після якого нам відразу видається певна інформація за введеними даними, тобто плозась поверхні, об'єм і маса фігури (рисунок 3.11).

```

Choice of the number of figures created -
0 to 3, to create 1-4 figures
1

Figure type options:
0->Sphere
1->Cube
2->Cuboid
3->Prism
4->Hemisphere
5->Pyramid
6->Cone
7->Cylinder

Choice of the figure type created -
0
Radius of the Sphere=2

Mass per volume of the Sphere=3

Charactersitics of the Sphere
Surface area of the Sphere=150.72
Volume of the Sphere=25.12
Mass of the Sphere=75.36

Choice of the figure type created -
1
Side of the Cube=2

Mass per volume of the Cube=1

Charactersitics of the Cube
Surface are of the Cube=24
Volume of the Cube=8
Mass of the Cube=8

```

Рисунок 3.11 — Введення числа фігур та вибір типу фігур

Після того, як ми створили необхідне нам число фігур, йде стадія введення їх вторинних характеристик, а конкретніше їх положення відносно один одного в системі координат (рисунок 3.12).

На цій стадії, як і на стадії вибору типу фігури є перевірка яка не дозволяє помилково введеним даним продовжити роботу всередині програми, у конкретному випадку мається на увазі, що дві фігури не можуть займати одні й ті ж координати в сітці.

```

Input of coordinates for first object
X=1

Y=2

Input of coordinates for second object
X=1

Y=3

Input of coordinates for third object
X=1

Y=4

Input of coordinates for fourth object
X=1

Y=1

```

Рисунок 3.12 — Введення координат об'єктів

Далі є можливість порівняти одну з основних характеристик об'єктів, що ми отримали після введення, точніше порівняти між собою цю характеристику всіх створених об'єктів і отримати дані про найбільшу з них .

Далі є можливість ще одного вибору для отримання даних про сумарну ширину і висоту або потенційну енергію об'єктів. Перший варіант також дозволяє отримати дані про те, як розташовані об'єкти один до одного.

Саме для двох об'єктів програма скаже якщо вони стоять діагонально (рисунок 3.13). Перевірка цього для трьох чи чотирьох сильно ускладнена тому була прописана.

Для трьох програма вкаже якщо об'єкти своїми координатами формують неповний квадрат, або трикутник, як кому зручніше, а також якщо вони вибудовані в горизонтальній або вертикальній лінії.

Для чотирьох об'єктів програма садить якщо вони розташовані по вертикальній або горизонтальній лінії, або якщо їх координати складають з себе кути куба. В інших випадках програма напише, що об'єкти не формують своїми координатами жодної з вище представлених форм.

Рисунок 3.13 — Порівняння створених фігур за:

```

1)Surface area
2)Volume
3)Mass
4)Restart the program
Choose - 3

Mass of the first figure is the largest and =75.36
Further information of created figures or structures formed
1)Energy of the objects created
2)Summary od width and length of the structure created by figures
3)Restart the program
Choose - 2

Width summ=6
Height summ=6
Objects form a diagonal line

```

к, сума ширини та висоти об'єктів

Як видно (рисунок 3.14) програма також видає і характеристику на основі маси об'єкта за формулою, що була зазначена в попередній частині цього розділу.

```

Side of the Cube=3
Mass per volume of the Cube=2
Charactersitics of the Cube
Surface are of the Cube=54
Volume of the Cube=27
Mass of the Cube=54

Input of coordinates for first object
X=2
Y=3

Comparison of created figures by:
1)Surface area
2)Volume
3)Mass
4)Restart the program
Choose - 1

Surface area of the first figure is the largest and =54
Further information of created figures or structures formed
1)Energy of the objects created
2)Summary od width and length of the structure created by figures
3)Restart the program
Choose - 1

Mass of the object =54
Energy of the object =4.86e+18
Energy of the only object created =4.86e+18

```

Рисунок 3.14 — Обчислення потенційної енергії об'єктів

3.3 Перевірка даних

У цій частині я хотів би докладніше розібрати рішення застосовані в перевірках даних з якими працює програма, чи то спочатку надійшли від користувача або результати роботи самого коду.

Крім простіших перевірок згаданих при виборі кількості та типу фігур, а також їх розташування в сітці щодо один одного є ще ряд функцій перевірки, що не перебувають на увазі при роботі коду.

Першим із таких варто назвати вторинну перевірку вибору числа створених об'єктів. Ця перевірка допомогла розділити функції другого та третього ступеня роботи програми, порівняння характеристик та подальша робота зібраними з даними, на кілька гілок використовуючи ранні дану інформацію, тобто вибір числа об'єктів.

Звичайно кожна гілка має подібності, в основному у вигляді формул обчислення прив'язаним до об'єктів або обробці характеристик отриманих з цих формул.

У цьому випадку (рисунок 3.15) цій характеристиці було присвоєно назву C2 від англійського слова choice що перекладатися як вибір. C1 і C3 відповідно належати до виборів типу створених фігур та вибору порівнюваних характеристик відповідно.

```

1048     if (C2==1){
1049         if (C3==1){
1050             double CsaMax= comp2(Csa0,Csa1);
1051             if (Csa0==CsaMax){
1052                 cout<<"Surface area of the first figure is the largest and ="<<Csa0<<endl;
1053             }
1054             if (Csa1==CsaMax){
1055                 cout<<"Surface area of the second figure is the largest and ="<<Csa1<<endl;
1056             }
1057         }
1058         if (C3==2){
1059             double CvMax= comp2(Cv0,Cv1);
1060             if (Cv0==CvMax){
1061                 cout<<"Volume of the first figure is the largest and ="<<Cv0<<endl;
1062             }
1063             if (Cv1==CvMax){
1064                 cout<<"Volume of the second figure is the largest and ="<<Cv1<<endl;
1065             }
1066         }
1067         if (C3==3){
1068             double CmMax= comp2(Cm0,Cm1);
1069             if (Cm0==CmMax){
1070                 cout<<"Mass of the first figure is the largest and ="<<Cm0<<endl;
1071             }
1072             if (Cm1==CmMax){
1073                 cout<<"Mass of the second figure is the largest and ="<<Cm1<<endl;

```

Рисунок 3.15 — Вторинна перевірка кількості створених об'єктів

Іншими типами перевірки є складні “if” що використовують логічні “і”, а також “або” для порівняння координат щодо один одного. Складнішим прикладами цих перевірок є діагональна для двох об'єктів (рисунок 3.16) і не лінійна і не квадратна для чотирьох (рисунок 3.17).

У першому випадку перевірка включає можливість двох напрямів і двох положень об'єктів щодо один одного, вище або нижче, праворуч або ліворуч від іншого, через що створюється додаткова складність.

У другому випадку проблеми створює те, що необхідно перевірити не тільки положення одного об'єкта, що стоїть поза лінією щодо інших по вертикалі або горизонталі, залежно від того, як збудовані інші три, але й те, що він зберігає при цьому з'єднання з хоча б одним з них.

```

1235     }
1236     if (((x1==(x-1) && y1==(y-1)) || (x1==(x+1) && y1==(y-1)) || (x1==(x-1) && y1==(y+1)) || (x1==(x+1) && y1==(y-1)))){
1237         hsm=Ch0+Ch1;
1238         wsm=Cw0+Cw1;
1239         cout<<"Width summ="<<wsm<<endl;
1240         cout<<"Height summ="<<hsm<<endl;
1241         cout<<"Objects form a diagonal line"<<endl;
1242     }
1243     else {
1244         cout<<"Objects are too far away to form united structure"<<endl;
1245     }
1246 }

```

Рисунок 3.16 — Перевірка розташування об'єктів , діагональна для двох об'єктів

```

1327     if (C2==3 && C4==2){
1328
1329         if (x==x1 && x1==x2 && ( x3>x1 || x3<x1 ) && (y3==y1 || y3==y2 || y3==y)){
1330
1331             double CCw;
1332
1333             if (y3==y){                                     ///      X
1334                 CCw=comp2(Cw0,Cw3);                       ///      X1X3
1335                 wsm=CCw+Cw1+Cw2;                           ///      X2
1336                 cout<<"Width summ="<<wsm<<endl;
1337             }
1338             if (y3==y1){
1339                 CCw=comp2(Cw1,Cw3);
1340                 wsm=Cw0+CCw+Cw2;
1341                 cout<<"Width summ="<<wsm<<endl;
1342             }
1343             if (y3==y2){
1344                 CCw=comp2(Cw2,Cw3);
1345                 wsm=Cw0+Cw1+CCw;
1346                 cout<<"Width summ="<<wsm<<endl;
1347             }
1348             hsm=(comp3(Ch0,Ch1,Ch2))+Ch3;
1349             cout<<"Height summ="<<hsm<<endl;
1350
1351         }

```

Рисунок 3.17 — Перевірка розташування об'єктів , не лінійна і не квадратна для чотирьох

3.4 Проблеми, що виникли під час розробки

У цій частині роботи я хотів би показати і розповісти про більш проблемні частини коду програми і про те, як складнощі долалися в міру його створення.

Першим таким пунктом я вважаю створення кількох об'єктів. Тут довелося створювати до 4 копій одного коду з перевіркою "if" залежно від того, яким по порядку був створений об'єкт цього конкретного типу (рисунок 3.18).

Рис	321	if (C1==0 && i==0){
	322	Sphere T;
уно	323	cout<<"Radius of the Sphere=";
	324	cin>>T.R;
к	325	cout<<endl;
	326	cout<<"Mass per volume of the Sphere=";
3.1	327	cin>>T.MpV;
	328	cout<<endl;
8	329	cout<<"Characterisitcs of the Sphere"<<endl;
	330	Csa0=T.CSa(T.R);
—	331	Cv0=T.CV(T.R);
Пер	332	Cm0=T.CM(T.R,T.MpV);
	333	Ch0=T.R*2;
ша	334	Cw0=T.R*2;
	335	}
про	336	if (C1==0 && i==1){ /// sphere 2
	337	Sphere T1;
бле	338	cout<<"Radius of the Sphere=";
	339	cin>>T1.R;
мна	340	cout<<endl;
	341	cout<<"Mass per volume of the Sphere=";
час	342	cin>>T1.MpV;
	343	cout<<endl;
тин	344	cout<<"Characterisitcs of the Sphere"<<endl;
	345	Csa1=T1.CSa(T1.R);
а	346	Cv1=T1.CV(T1.R);

коду

Більш нудним, ніж проблемним, можна назвати всі три варіанти порівняння характеристик (рисунок 3.19), кожен з яких довелося розбивати в залежності від чисельності створених фігур, а відповідно подовжувати код.

Подібна ж проблема виникла при перевірці координат, а точніше того, що жоден з об'єктів не має однакових координат з іншими створеними раніше до нього.

```

1077     if (C2==2){
1078         if (C3==1){
1079             double CsaMax= comp3(Csa0,Csa1,Csa2);
1080             if (Csa0==CsaMax){
1081                 cout<<"Surface area of the first figure is the largest and ="<<Csa0<<endl;
1082             }
1083             if (Csa1==CsaMax){
1084                 cout<<"Surface area of the second figure is the largest and ="<<Csa1<<endl;
1085             }
1086             if (Csa2==CsaMax){
1087                 cout<<"Surface area of the third figure is the largest and ="<<Csa2<<endl;
1088             }
1089         }
1090         if (C3==2){
1091             double CvMax= comp3(Cv0,Cv1,Cv2);
1092             if (Cv0==CvMax){
1093                 cout<<"Volume of the first figure is the largest and ="<<Cv0<<endl;
1094             }
1095             if (Cv1==CvMax){
1096                 cout<<"Volume of the second figure is the largest and ="<<Cv1<<endl;
1097             }
1098             if (Cv2==CvMax){
1099                 cout<<"Volume of the third figure is the largest and ="<<Cv2<<endl;
1100             }
1101         }

```

Рисунок 3.19 — Друга проблемна частина коду

Останнім проблемним моментів я назвав би перевірку положення об'єктів(рисунок 3.20), конкретно для трьох і чотирьох з них відповідно. Це виникло в основному через дуже велику кількість потенційних положень об'єктів щодо один одного.

У разі побудови по квадрату безліч варіантів з незначною різницею виникає через те, яка пара знаходиться вище або нижче за іншу.

Загальна кількість варіантів квадратів дорівнює шести, плюс чотири варіанти повороту кожного окремого варіанту, що благо не потребують окремих перевірок через однакове відносне розташування об'єктів.

```

/// 24 | 13 | 31 | 42
/// 13 | 24 | 42 | 31
if ( (x==x2 && x1==x3) && (x1==x+1 || x1==x-1) && (y3==y2 && y1==y)){
    wsm=comp2(Cw0,Cw1)+comp2(Cw2,Cw3);
    hsm=comp2(Ch0,Ch2)+comp2(Ch1,Ch3);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}
/// 24 | 31 | 13 | 42
/// 31 | 24 | 42 | 13
if ( (x==x2 && x1==x3) && (x1==x+1 || x1==x-1) && (y3==y && y1==y2)){
    wsm=comp2(Cw0,Cw3)+comp2(Cw2,Cw1);
    hsm=comp2(Ch0,Ch2)+comp2(Ch1,Ch3);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}
/// 23 | 14 | 41 | 32
/// 14 | 23 | 32 | 41
if ( (x==x1 && x2==x3) && (x2==x+1 || x2==x-1) && (y1==y2 && y==y3)){
    wsm=comp2(Cw0,Cw1)+comp2(Cw2,Cw3);
    hsm=comp2(Ch0,Ch3)+comp2(Ch1,Ch2);
    cout<<"Height summ="<<hsm<<endl;
    cout<<"Width summ="<<wsm<<endl;
    cout<<"Object's coordinates create a square line"<<endl;
}

```

Рисунок 3.20 — Третя проблемна частина коду

Ще однією помітною проблемою для структур із більш ніж двох фігур виявилось обчислення сумарної висоти та ширини.

Розглядаючи конкретні випадки можна побачити рішення до якого привели ці складності, а саме прив'язка до координат характеристик фігури та відсутність обмежувачів на те ширину та висоту фігури. Таким чином фігура на конкретній координаті може бути в кілька разів більше сусідньої.

Для обчислення ж загальних характеристик використовувалося порівняння стоять один над одним або з однієї зі сторін від іншої фігури. Таким чином за наявності лінії вертикальної її ширина дорівнює найбільшій із ширин фігур з яких лінія складена.

При утворенні складніших структур на зразок кута (рисунок 3.21) або Т образної форми (рисунок 3.22) для трьох і чотирьох об'єктів відповідно

проводитися порівнянням між двома елементами після чого третім додається у

```

1265      // two lined one not variant
1266      if ((y==y1) && (x2==x1)){
1267          hsm=comp2(Ch0,Ch1)+Ch2;
1268          wsm=comp2(Cw2,Cw1)+Cw0;
1269          cout<<"Width summ="<<wsm<<endl;
1270          cout<<"Height summ="<<hsm<<endl;
1271          cout<<"Objects form an L shape"<<endl;
1272          // 01| 2| 2 | 10
1273          // 2| 01| 10| 2
1274      }
1275      if ((y==y1) && (x2==x)){
1276          hsm=comp2(Ch0,Ch1)+Ch2;
1277          wsm=comp2(Cw2,Cw0)+Cw1;
1278          cout<<"Width summ="<<wsm<<endl;
1279          cout<<"Height summ="<<hsm<<endl;
1280          cout<<"Objects form an L shape"<<endl;
1281          // 10| 2| 01| 2
1282          // 2| 10| 2 | 10
1283      }
1284
1285      if ((y2==y1) && (x==x1)){
1286          hsm=comp2(Ch2,Ch1)+Ch0;
1287          wsm=comp2(Cw0,Cw1)+Cw2;
1288          cout<<"Width summ="<<wsm<<endl;
1289          cout<<"Height summ="<<hsm<<endl;
1290          cout<<"Objects form an L shape"<<endl;
1291          // 01| 01| 10| 0

```

виділене значення.

Рисунок 3.21 — Приклад внутрішніх формул для кутової побудови

```

1352
1353   if (y==y1 && y1==y2 &&( y3>y1 || y3<y1 ) && (x3==x1 || x3==x2 || x3==x)){
1354
1355       double CCh;
1356
1357       if (x3==x){
1358           CCh=comp2(Ch0,Ch3);           /// Y Y1 Y2
1359           wsm=CCh+Ch1+Ch2;           /// Y3
1360           cout<<"Height summ="<<hsm<<endl;           ///
1361       }
1362       if (x3==x1){
1363           CCh=comp2(Ch1,Ch3);
1364           wsm=Ch0+CCh+Ch2;
1365           cout<<"Height summ="<<hsm<<endl;
1366       }
1367       if (x3==x2){
1368           CCh=comp2(Ch2,Ch3);
1369           wsm=Ch0+Ch1+CCh;
1370           cout<<"Height summ="<<hsm<<endl;
1371       }
1372       wsm=(comp3(Cw0,Cw1,Cw2))+Cw3;
1373       cout<<"Width summ="<<wsm<<endl;
1374
1375   }
1376   // Sources from blocks 13 / 13 / 14 / 21 / 22 / 24 / 21 / 22 / 24 / 41 /

```

Рисунок 3.22 — Приклад внутрішніх формул для Т образної побудови

Як приклад візьмемо і створимо по одній структурі з кожного варіанта, що були описані раніше.

Для кутової побудови (рисунок 3.23) ми створимо три постаті. Сфера з радіусом рівним трьом, куб зі стороною рівною двом і циліндр чий радіус дорівнює трьом, а висота чотирьом. Вони розставлені за координатами 1:1, 1:2 та 2:1 відповідно.

Для Т образного (рисунок 3.24) будуть створені дві сфери та два куби зі сторонами два і три, по одному в кожній парі фігур. Перші три були розташовані по вертикалі в той час, як остання фігура зайняла місце праворуч від найнижчої.

У першому випадку результатом буде те, що загальна ширина структури зі створених фігур дорівнюватиме восьми, в той час як загальна висота дорівнюватиме десяти.

Другий випадок створить результат у вигляді загальної ширини рівної тринадцяти і висоти рівної восьми.

```

Choice of the figure type created -
0
Radius of the Sphere=3

Mass per volume of the Sphere=1

Charactersitics of the Sphere
Surface area of the Sphere=339.12
Volume of the Sphere=84.78
Mass of the Sphere=84.78

Choice of the figure type created -
1
Side of the Cube=2

Mass per volume of the Cube=1

Charactersitics of the Cube
Surface are of the Cube=24
Volume of the Cube=8
Mass of the Cube=8

Choice of the figure type created -
7
Radius of the Cylinder=3

Height of the Cylinder=4

Mass per volume of the Cylinder=1

Charactersitics of the Cylinder
Surface area of the Cylinder=131.88
Volume of the Cylinder=113.04
Mass of the Cylinder=113.04

Width summ=8
Height summ=10
Objects form an L shape

```

Рисунок 3.23 — Результати для першого випадку

Підсумовуючи своєрідний підсумок, я можу сказати що код вийшов функціональним але не надто красивим і частково громіздким через велику кількість перевірок створеними розшируванням варіантів чисельності створюваних фігур. Число чотири було обрано не просто так, а через те, що

перевірки для п'яти і більше об'єктів зайняли б не менше місця, ніж весь код для чотирьох.

```

Choice of the figure type created - 0
Radius of the Sphere=2
Mass per volume of the Sphere=1
Characteristics of the Sphere
Surface area of the Sphere=150.72
Volume of the Sphere=25.12
Mass of the Sphere=25.12
Choice of the figure type created - 0
Radius of the Sphere=3
Mass per volume of the Sphere=1
Characteristics of the Sphere
Surface area of the Sphere=339.12
Volume of the Sphere=84.78
Mass of the Sphere=84.78
Choice of the figure type created - 1
Side of the Cube=3
Mass per volume of the Cube=1
Characteristics of the Cube
Surface area of the Cube=54
Volume of the Cube=27
Mass of the Cube=27
Choice of the figure type created - 1
Side of the Cube=2
Mass per volume of the Cube=1
1)Energy of the objects created
2)Summary of width and length of the structure created by figures
3)Restart the program
Choose - 2
Width summ=13
Height summ=8

```

Рисунок 3.24 — Результати для другого випадку

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи була створена програма, яка може:

- введення та обчислювати характеристики тривимірних об'єктів;
- розташування створених об'єктів у координатній сітці;
- порівнювати характеристики створених об'єктів;
- проведення подальших операцій для отримання даних про сумарні характеристики створених структур та розташування об'єктів.

Теоретична база та практична розробка програми була описана в трьох розділах, а саме:

- у першому розділі описано історію та причини розвитку тривимірних технологій як відросли;
- у другому розділі описано список основних споживачів технологій цієї галузі у формі програм або пристроїв друку;
- у третьому розділі описано програмний код, його функції як основні та вторинні, основні проблеми з якими зіткнулися під час розробки.

Таким чином вході роботи було створена програма для створення кількох типів тривимірних об'єктів, обчислення їх математичних та фізичних характеристик за введеними параметрами, розташування їх у сітці координат та порівняння їх характеристик отриманих у ході роботи програми.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Network layer. [Електронний ресурс] – URL: <https://en.wikipedia.org/wiki/Organovo> (Дата звернення 23.05.2024).
2. Network layer. [Електронний ресурс] – URL: <https://www.timetoast.com/timelines/3d-7a35b5c1-161f-4397-a789-5d00616f2c42> (Дата звернення 22.05.2024).
3. Data link layer. [Електронний ресурс] – URL: <https://ucvt.org/blog/oblasti-primeneniya-3d-tehnologij-v-sovremennom-mire> (Дата звернення 22.05.2024).
4. Data link layer. [Електронний ресурс] – URL: <https://iqb.ru/industries/power-generation/> (Дата звернення 22.05.2024).
5. Data link layer. [Електронний ресурс] – URL: https://en.wikipedia.org/wiki/Objet_Geometries (Дата звернення 31.05.2024).
6. Data link layer. [Електронний ресурс] – URL: https://en.wikipedia.org/wiki/Z_Corporation (Дата звернення 31.05.2024).
7. Data link layer. [Електронний ресурс] – URL: <https://scientificservices.eu/item/spectrum-z510-3d-printer/819> (Дата звернення 31.05.2024).