Комп'ютерна графіка (КГ) - це спеціальна галузь комп’ютерних дисциплін, що вивчає методи і засоби створення та обробки зображень за допомогою програмно-апаратних обчислювальних комплексів.

**Сфера застосування** КГ включає чотири основні області:

*1. Відображення інформації*

*2. Проектування*

*3. Моделювання (імітація)*

*4. Графічний інтерфейс користувача.*

**Основні напрями КГ**:

·        образотворча комп'ютерна графіка,

·        обробка та аналіз зображень,

·        аналіз сцен (перцептивна комп'ютерна графіка),

·        комп'ютерна графіка для наукових абстракцій (когнітивна комп'ютерна графіка, тобто графіка, сприяюча пізнанню).

**Історія**

Перша офіційно визнана спроба використання дисплея для виводу зображення з ЕОМ - Массачусетський технологічний університет (МТІ), машина Whirlwind-I в 1950 р. А сам термін "комп'ютерна графіка" вперше використав в 1960 р. співробітник компанії Boeing У. Феттер. Перше реальне застосування КГ пов'язують з ім'ям Дж. Уїтні. Він займався кіновиробництвом в 50-60-х роках і вперше використовував комп'ютер для створення титрів до кінофільму.

Наступний крок у  розвитку КГ- робота Айвена Сазерленда, який в 1961р створив програму малювання, названу ним Sketchpad (альбом для малювання). Програма використовувала світлове перо для малювання найпростіших фігур на екрані. Отримані картинки можна було зберігати і відновлювати.

*Перший етап* -  1960-1970-і роки, коли КГ формувалася як наукова дисципліна. У цей час розроблялися основні методи і алгоритми: відсікання, растрова розгортка графічних примітивів, зафарбування візерунками, реалістичне зображення просторових сцен (видалення невидимих ліній і граней, трасування променів, випромінюють поверхні), моделювання освітленості.

*Другий етап* -  1980-ті, коли КГ розвивається більш як прикладна дисципліна. Розробляються методи її застосування в самих різних областях людської діяльності.

*Третій етап* – з  1990-тих років, коли методи комп'ютерної графіки стають основним засобом організації діалогу "людина-комп'ютер" і залишаються такими по теперішній час.

**Основними параметрами монітора є:**

розмір екрану, розмір зерна, швидкість оновлення зображень (частота кадрової розгортки), ступінь плоскості екрану (вища реалістичність зображення, менше відблисків), роздільна здатність.

У залежності від способу **формування зображень КГ** поділяють на:

* растрову;
* векторну;
* фрактальну;
* тривимірну.

**Основні складові векторної графіки:**

* Шлях – це маршрут, що з'єднує початкову та кінцеву точку.
* Сегмент - окрема частина шляху, може бути як прямою, так і кривою лінією.
* Вузол - початкова або кінцева точка сегмента.

Кожен елемент векторної графіки містить ці три основні елементи і дозволяє їх редагування.

**Базові елементи**

Лінія є елементарним об'єктом, якому притаманні певні особливі властивості: форма, товщина, колір, стиль тощо.

Будь-який об'єкт (прямокутник, еліпс, текст і навіть пряма лінія) сприймається як криві лінії.

Зображення описується з допомогою базових примітивів.

Базовий примітив – команда пристрою побудувати певну лінію (вектор).

Вектор – звичайна лінія, коло/еліпс, прямокутник, багатокутник

**Переваги**:

* невеликі за розміром файли, оскільки зберігається не зображення, а лише його основні дані, використовуючи які, програма відновлює зображення;
* об'єкти легко трансформуються, ними легко маніпулювати. Редагуючи векторний об'єкт, можна змінити властивості ліній, з яких складається зображення. Можна пересувати об'єкт, змінювати його розміри, форму та колір, не впливаючи на якість зображення;
* векторна графіка не залежить від роздільності, тобто векторні об'єкти відтворюють на пристроях з різною роздільністю без втрати якості зображення.

**Недоліки**:

* Низька фотореалістичність
* Обмежена область застосування
* Низька швидкість промальовування на растрових пристроях

**Прикладна векторна графіка**

* повноколірні ілюстрації;
* складні креслення;
* логотипи та емблеми;
* графічні зображення для Web;
* мультиплікація;
* рисунки на основі оригіналів

**Найпопулярнішими прикладними програмами** є продукти фірм

* Corel - CorelDraw,
* Adobe - Illustrator,
* Macromedia - FreeHand,
* стандартний додаток у MS Office - Word Editor.
* Vector magic
* Affinity Designer

**Формати файлів векторної графіки**

* **AI** – файли програми Adobe Illustrator. Розширення – **ai**.
* **CDR** - файли програми CorelDraw. Розширення – **cdr**.
* **SVG** – зберігає також і анімацію. Використовується в інженерній графіці та під час розробки веб-сайтів. Розширення – **svg**.
* **WMF** - Універсальний формат для програм, що працюють в ОС Windows. Розширення – **wmf**, **emf**.

**Типи заповнення:**

* *однорідне заповнення* одним кольором або штрихуванням;
* *градієнтне*, при якому кольори або тіні поступово змінюються (лінійна, радіальна, конічна, прямокутна закономірність тощо);
* *візерункове*, при якому об'єкт заповнюється повторювальними зображеннями (двоколірними або повноколірними);
* *текстурне* заповнення (художні зображення)

**Математичні основи векторної графіки:**

Точка. Об'єкт на площині представляється двома числами (х, у) відносно початку координат.

Пряма лінія. Їй відповідає рівняння у=kx+b. Вказавши параметри k та b можна створити пряму лінію у відомій системі координат.

Сегмент прямої. Для опису потрібно додатково вказати параметри х1 та х2, відповідно початок та кінець відрізку.

Крива лінія ІІ порядку. До них належать еліпси, круги, параболи, гіперболи тощо. Пряма лінія є також випадком кривої ІІ порядку. Крива ІІ порядку описується рівнянням а0х2+а1у2+а2ху+а3х+а4у+а5=0. Для побудови відрізка кривої додатково потрібні ще два параметри початку та кінця відрізку.

Крива лінія ІІІ порядку. Важлива наявність точки перегину, що дозволяє відобразити різноманітні об'єкти. Рівняння кривої ІІІ порядку а0х3+а1у3+а2х2у+а3ху2+а4х2+а5у2+а6ху+а7х+а8у+а9=0. Для опису відрізка потрібні ще два параметри початку та кінця відрізку. Зауважимо, що пряма та криві ІІ порядку є частковим випадком кривих ІІІ порядку.

Криві Без'є. Спрощений вид кривих ІІІ порядку. Метод побудови кривих Без'є заснований на використанні пари дотичних, що проведені до лінії в крайніх точках. На форму кривої лінії впливає кут нахилу дотичних та довжина її відрізка. Таким чином, дотичні відіграють роль віртуальних важелів, за допомогою яких керують формою кривої.

**Крива Без’є**

**Крива Безьє** — [параметрично задана крива](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B5_%D1%80%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F), яка використовується в [комп'ютерній графіці](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%27%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0) та суміжних областях. У [векторній графіці](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0), криві Безьє використовуються для моделювання гладких [кривих](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%B2%D0%B0).

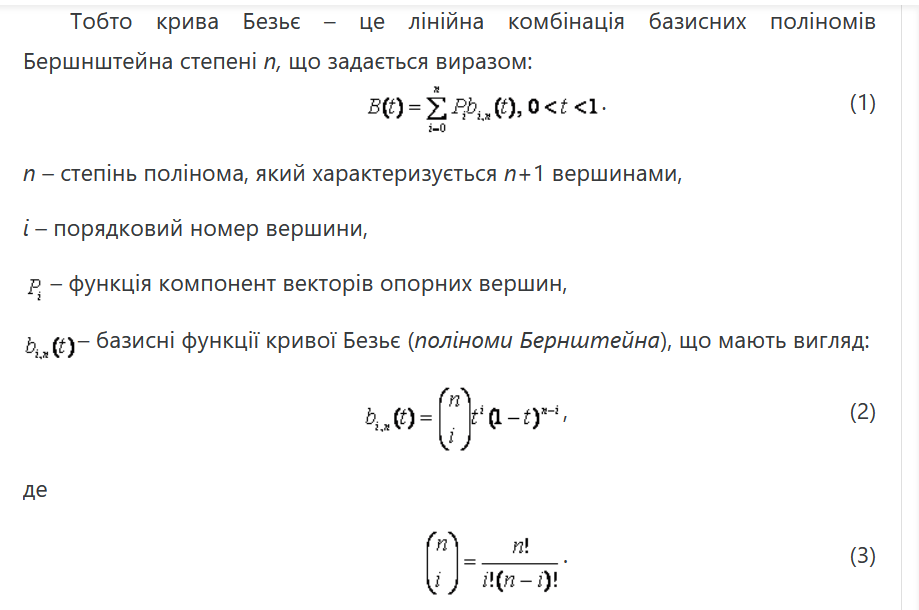
* Гнучкість, наглядність з точки зору користувача-дизайнера
* Простота з точки зору математичного апарату
* Універсальність з точки програмування

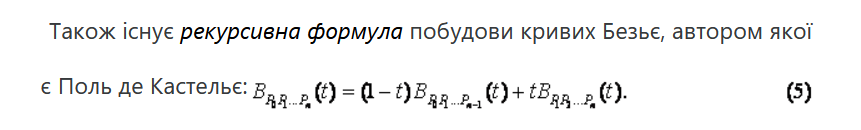
**Елементи керування**:

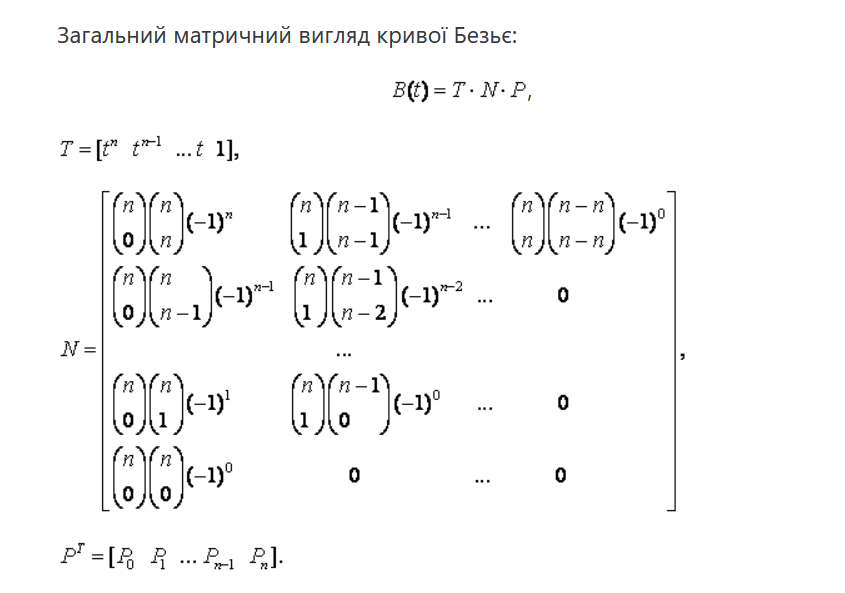
* Контрольні точки є опорні і керуючі
* Дотичні

У процесі інтерактивного конструювання характеристична ламана є засобом управління кривої Без’є.

Для побудови кривої Безьє в комп’ютерній графіці використовують три основні способи: параметричний, рекурсивний, матричний.

**Параметричний:**

**Рекурсивний:**

**Матричний:**

**Властивості многочленів:**

1. Многочлени Бернштейна набувають невід’ємних значень.

2. У сумі вони дають одиницю.

3. Многочлени не залежать від вершин масиву *P*, а залежать лише від кількості точок у масиві.

**Геометричний алгоритм** для кривої Безьє дозволяє обчислити координати (*x, y*) точки кривої Безьє за значенням параметра *t*.

1. Кожна сторона опорної ламаної ділиться у відношенні *t:*(1- *t*), де *t –* аргумент точки поділу.

2. Точки поділу з'єднуються відрізками прямих і утворюють нову ламану (багатокутник). Кількість вузлів нового контуру на одиницю менша, ніж кількість вузлів попереднього контуру.

3. Сторони нового контуру знову діляться пропорційно значенню *t* і т.д. Це продовжується до тих пір, поки не буде отримана єдина точка поділу – точка кривої Безьє.

**Складена кубічна крива Безьє** – це неперервна крива γ, що є об’єднанням елементарних кубічних кривих γ0, γ1, …, γl , тобто

Для побудови складеної кривої Безьє розбиваємо масив точок *P*0, *P*1,…, *Pn* (*n* має бути кратним 3) на *l* підмножин, кожна з яких містить чотири точки так, що остання точка попередньої підмножини – це перша точка наступної підмножини, тобто

{*P*0, *P*1, *P*2, *P*3}, {*P*3, *P*4, *P*5, *P*6}, …, {*P*3*l*, *P*3*l* + 1, *P*3*l* + 2, *P*3*l* + 3}.

На кожній такій підмножині можна побудувати елементарну криву Безьє, тобто окремий фрагмент складеної кривої Безьє. Об’єднання окремих фрагментів дає складену кубічну криву Безьє, яка належить до класу неперервних функцій. Якщо тепер змінити якусь точку масиву, то складена крива Безьє зміниться тільки локально.

**Растрова графіка. Базові елементи та характеристики**

Моделлю називають опис за певними правилами, враховуючи деякі припущення.

## Растрова модель представляє зображення у вигляді комбінації точок (пікселів), яким притаманні свій колір та яскравість і які певним чином розташовані у координатній сітці. Піксель (Pixel) – це скорочення, утворене від терміну Picture Element (англ.). Растром називають матрицю пікселів. Поняття матриці вказує на розміщення пікселів за координатами по горизонталі та вертикалі. Растр має кількісну та якісну характеристики. Кількісна характеристика растру – це розмір растра, тобто кількість пікселів по вертикалі та горизонталі. Записують цю характеристику як NxM, де N, M – відповідні кількості пікселів. Якісна характеристика растру – це роздільна здатність, яка вказує кількість точок на одиницю довжини, вимірюється у точках на дюйм (dpi - dots per inch).

## Існує декілька типів зображень із різною глибиною кольору:

## n    чорно-білі;

## n    у відтінках сірого;

## n    з індексованими кольорами;

## n    повноколірні;

## Прикладні програми:

## Adobe - програма PhotoShop,

## n     Corel – програма PhotoPaint,

## n     Macromedia – програма FireWorks,

## n     Fractal Design – програма Painter,

## n     стандартний додаток у Windows - Paint.

**Формати файлів растрової графіки**

## Найпростіший растровий формат BMP є рідним форматом системи Windows, він підтримується всіма графічними редакторами, які працюють під її управлінням. Цей формат застосовується для збереження растрових зображень, призначених для використання в Windows (і, по суті, більше ні на що не придатний).

## Найпопулярнішим форматом на інтернетівських просторах є достатньо вже давній формат GIF, запропонований CompuServe в 1987 році. Його особливістю є використання режиму індексованих кольорів (не більш 256), що обмежує область застосування формату зображень, які мають різкі кольорові переходи. Цей формат не може застосовуватися для поліграфії

## Найпопулярнішим форматом для збереження фотографічних зображень є JPEG (или JPG) JPEG може зберігати тільки 24-бітові повнокольорові зображення. Хоча JPEG добре стискає фотографії, але цей стиск відбувається з втратами і псує якість, тим не менше, він може бути легко налаштований на мінімальні, практично непомітні для ока, втрати. Проте не варто використовувати JPEG для збереження зображень, які будуть опрацьовуватися, так як при кожному збереженні в цьому форматі процес погіршення якості зображення носить лавиноподібний характер. Таким чином, можна зберегти якість зображення при мінімальному розмірі  файлу.

## Як універсальний формат для зберігання растрових зображень пропонується формат TIFF, який досить широко використовується, в першу чергу, у видавничих системах, що вимагають зображення найкращої якості. До речі, можливість запису зображень у форматі TIFF є однією з ознак високого класу сучасних цифрових фотокамер. Завдяки своїй сумісності з більшістю професійного ПЗ для обробки зображень, формат TIFF дуже зручний при перенесенні зображень між комп'ютерами різних типів (наприклад, з PC на Маc і назад).

## PSD. Є внутрішнім для Photoshop, і дозволяє зберігати інформацію про контури, канали, шари, векторні надписи. Підтримуються всі кольорові моделі і будь-яка глибина кольору. Застосовується стиснення без втрат.

## Формат RAW порівняно молодий формат. Для відкриття такого файлу в графічному редакторі необхідно проінсталювати спеціальне доповнення. У цілому формат RAW має більше переваг, ніж TIFF, по-перше, тому, що дозволяє зберігати початкову кольорову інформацію знімка без застосування спеціальних алгоритмів, по-друге, тому що файли формату RAW займають приблизно в два рази менший об’єм пам’яті, ніж TIFF

# **Переваги растрової графіки**

## n    простота автоматизованого вводу (оцифрування) зображень, фотографій, слайдів, рисунків за допомогою сканерів, відеокамер, цифрових фотоапаратів;

## n     фотореалістичність. Можна отримувати різні ефекти, такі як туман, розмитість, тонко регулювати кольори, створювати глибину предметів тощо.

# **Недоліки растрової графіки**

## n         складність управління окремими фрагментами зображення. Потрібно самостійно виділяти ділянку, що є складним процесом;

## n         ефекти спотворення при масштабуванні (“розмитість” при збільшенні, “ступінчастість” при зменшенні)

## n         розмір файлу є пропорційним до площі зображення, роздільності і типу зображення, і, переважно, при хорошій якості є великим.

## n         растрове зображення має певну роздільність і глибину представлення кольорів. Ці параметри можна змінювати лише у визначених межах і, як правило, із втратою якості;

**Фракталом** називається структура, що складається з частин, які в якомусь сенсі подібні до цілого.

## Базовим елементом є математична формула, ніяких об'єктів у пам'яті комп'ютера не зберігається і зображення будується виключно за формулами, рівняннями.

## Однією з основних властивостей фракталів є самоподібність.  У найпростішому випадку невелика частина фрактала містить інформацію про весь фрактал. Серед важливих властивостей фракталів з точки зору програмної реалізації ітераційність та рекурсивність фрактальних структур.

## Фактично знайдений спосіб легкого представлення складних  об'єктів,  які  схожі на природні. Тобто за допомогою декількох коефіцієнтів можна задати лінії і поверхні дуже складної форми.

# **Класифікація фракталів за способом побудови**

## n   Геометричні (конструктивні)

## n   Алгебраїчні (рекурентні)

## n   Стохастичні (випадкові)

## Геометричні та [алгебраїчні фрактали](http://vns.lpnu.ua/mod/resource/view.php?id=264196) називають детермінованими. Окремо розглядають підвид фракталів, що утворюється через систему ітеруючих функцій – IFS-фрактали.

**Геометричні фрактали:**

## Фрактали цього типу будуються поетапно. Спочатку зображується основа . Потім деякі частини основи замінюються на фрагмент . На кожному наступному етапі частини вже побудованої фігури , аналогічні заміненим частинам основи , знову замінюються на фрагмент , взятий у відповідному масштабі. Кожного разу масштаб зменшується. Коли зміни стають візуально непомітними , вважають , що побудована фігура добре наближає фрактал і дає уявлення про його форму . Однак насправді для отримання фрактала потрібно нескінченне число етапів. Міняючи основу і фрагмент , можна отримати багато різних геометричних фракталів. .У двомірному випадку їх отримують за допомогою деякої ламаної (або поверхні в тривимірному випадку), яку наз. *генератором*.

## За один крок алгоритму кожен з відрізків, складових ламаної, замінюється на ламану-генератор, у відповідному масштабі.

## Найвідоміші фрак тали такого типу - Сніжинка Коха, крива Коха, крива Гільберта-Пеано, множина Кантора, килим Серпінського, трикутник Серпінського, крива дракона, Т-Квадрат, губка Менгера…

# **Алгебраїчні фрактали**

## Це найбільша група фракталів. Отримують їх за допомогою нелінійних процесів в n-мірних просторах.

## Для побудови алгебраїчних фракталів використовуються ітерації нелінійних відображень, що задаються простими алгебраїчними формулами.

## Найбільш вивчені двомірні процеси.

## Прикладами фракталів цього типу є множина Мандельброта, палаючий корабель, фрактал Ляпунова, басейни Ньютона, біоморфи...

## Методів отримання алгебраїчних фракталів декілька. Один із методів являє собою багатократний (ітераційний) розрахунок функції

де Z – комплексне число, а f - деяка функція. Розрахунок даної функції продовжується до виконання певної умови. І коли ця умова виконається – на екран виводиться точка. При цьому значення функції для різних точок комплексної площини може мати різну поведінку:

1 з плином часу прагне до безкінечності,

2. прагне до 0,

3. приймає декілька фіксованих значень і не виходить за їх межі,

4. поведінка хаотична, без будь-яких тенденцій.

## Для фракталів Мандельброта правило f задано формулою : f(z ) = z2 + c , де z і c - комплексні числа. Кожній точці на прямій відповідає те чи інше дійсне число , а кожній точці на площині - комплексне . Для комплексних чисел, як і для дійсних , визначені операції додавання і множення.

# **Стохастичні фрактали**

## Такі фрактали утворюються в тому випадку, якщо  в ітераційному процесі випадковим чином міняти які-небудь його параметри.  При цьому виходять об'єкти дуже схожі на природні – несиметричні дерева, порізані берегові лінії, рельєф місцевості і поверхні моря і т.д

## Плазма. Для його побудови використовують прямокутник і для кожного його кута визначається колір. Далі знаходиться центральна точка прямокутника і розфарбовується  у колір рівний середньому арифметичному кольорів по кутах прямокутника плюс деяке випадкове число. Чим більше випадкове число - тим більш «рваним» буде малюнок. Якщо, наприклад, вважати, що колір точки - це висота над рівнем моря, то отримається замість плазми - гірський масив. Саме на цьому принципі моделюються гори в більшості програм.

## IFS-метод з'явився в середині 80-х років як простий засіб отримання фрактальних структур.

## IFS є системою функцій  деякого фіксованого класу функцій, що відображають одну багатовимірну множину на іншу.

## Найпростіша IFS складається з афінних перетворень :

## A, B, C, D, E, F – деякі коефіцієнти.

## Фрактальний метод стиску використовується для економного збереження зображень без втрати якості. FIF (Fractal Image Format) – спеціальний формат файлів для фрактальних зображень.

## Для синтезу зображень фрактальної графіки існує не дуже багато програмних продуктів. Серед них немає визнаного лідера, як в програмах растрової і векторної графіки. Також особливістю цих програм є те, що вони використовують фрактальний підхід до синтезу зображень разом із звичним набором інструментів растрової і векторної графіки. Відомі фрактальні редактори – Fractal Design Painter, Bruce, Fractint.

## Фрактальна графіка у пакетах наукової візуалізації використовується для побудови ілюстрацій, що імітують природні процеси.

## Серед таких програмних засобів продукти фірми Golden SoftWare:

## *Surfer* - створення дво-, тривимірних поверхонь;

## *Map Viewer* - побудова кольорових карт.

## Surfer дозволяє опрацьовувати та візуалізувати двовимірні набори даних, що описані функцією z=f (x,y). Можна побудувати цифрову модель поверхні, застосувати допоміжні операції і візуалізувати результат.

## Map Viewer дозволяє вводити та корегувати карти - змінювати масштаб, перетворювати координати, обробляти й виводити у графічному вигляді числову інформацію, пов'язану з картами.

## Пакет *Iris Explorer* (фірма Graphics) призначено для створення моделей погодних умов та поверхонь океану.

## Пакет *Earth Watch* (фірма Earth Watch) призначений для моделювання та демонстрації тривимірного зображення метеоумов над Землею, дозволяє будувати топологічні поверхні і прогнозувати погоду.

## 3D  графіка (3 Dimensions) — один з розділів комп’ютерної графіки, який призначений для зображення об’ємних об’єктів (на площині).

## У тривимірній комп'ютерній графіці всі об'єкти зазвичай представляються як набір поверхонь або частинок.

## Мінімальну поверхню називають полігоном. В якості полігону зазвичай вибирають трикутники.

## Тривимірна графіка використовує віртуальний простір, який називають сценою. У цьому просторі проектувальник розміщує необхідні об'єкти, призначає для них певний матеріал (дерево, залізо, скло тощо), розміщує джерела світла, а також віртуальні камери, що визначають точки перегляду сцени.

## *Попередня підготовка* передбачає продумування складу сцени, розміщення об’єктів і їх деталей, які будуть видимими з передбачуваних напрямів спостереження.

## На етапі *створення геометричної моделі сцени* за допомогою різноманітних інструментальних засобів будуються тривимірні геометричні моделі об’єктів сцени, після чого сцену можна розглядати і “фотографувати” з будь-якого потрібного ракурсу.

## *Налаштовування освітлення і знімальних камер* передбачає правильний вибір джерел світла, що дозволяє виконувати імітацію фотографування сцени в будь-яких умовах освітленості. Освітленість всіх об’єктів, тіні від них і бліки світла розраховуються автоматично.

## Моделі знімальних камер дають можливість розглядати тривимірну сцену і виконувати її знімання під будь-яким вибраним кутом зору.

## *Підготовка і призначення матеріалів* надає сцені візуальної правдоподібності. Працюючи з матеріалами, можна налаштовувати такі їх якості, як сила блиску, прозорість, дзеркальність, рельєфність та інші. Реальні фотографії можна включати в склад матеріалів або використовувати для імітації фону.

## *Візуалізація сцени* полягає в проведенні програмою розрахунків і нанесення на зображення всіх тіней, бліків, взаємних відблисків об’єктів і т. п.

## Тобто для отримання тривимірного зображення потрібно виконати *моделювання* - створення математичної моделі сцени і об’єктів в ній та *рендерінг* (rendering) - побудова проекції відповідно до вибраної фізичної моделі .

**СЦЕНА** (віртуальний простір моделювання) включає в себе такі категорії :•Геометрія(побудована за допомогою різних технік модель, наприклад будівля);

•Матеріали(інформація про візуальні властивості моделі, наприклад колір стін і відбиваюча / заломлююча здатність вікон)

•Джерела світла (налаштування напрямків, потужності, спектра освітлення);

•Віртуальні камери (вибір точки та кута побудови проекції);

•Сили та дії (налаштування динамічних спотворень об'єктів, застосовується в основному в анімації);

•Додаткові ефекти (об'єкти, що імітують атмосферні явища: світло у тумані, хмари, полум'я і пр.)

**Існує декілька технологій візуалізації**, часто комбінованих разом.

•Z-буфер –алгоритм, який відповідає за створення зображень 3D-об'єктів, спираючись на глибину елементів зображення. Використовується в «OpenGL» і «DirectX».При створенні (візуалізації)3d-об'єкту його глибина генерується на осі Z-координат і зберігається у Z-буфері. Z-буфер є двомірним масивом (X,Y-координати) із глибиною для кожного екранного пікселя. Коли інший об'єкт сцени повинен бути відображений у цьому пікселі зараз, тоді порівнюється дві глибини та перекривається поточний піксель, якщо об'єкт знаходиться ближче до спостерігача. Обрана глибина зберігається в Z-буфері і замінює попередню. Z-буфер дозволяє правильно відтворювати звичне для нас сприйняття об'єктів, розміщених на різних відстанях.

•Сканлайн(«кидання променю», спрощений алгоритм зворотного трасування променів)-розрахунок кольору кожної точки картинки на основі побудови променю з точки зору спостерігача через уявний отвір в екрані на місці цього пікселя «в сцену» до перетину з першою поверхнею. Колір пікселя буде таким же, як колір цієї поверхні;

•Трасування променів (Ray tracing, рейтрейсинг)-те ж, що і сканлайн, але колір пікселя уточнюється за рахунок побудови додаткових променів від точки перетину променю погляду; Ідея алгоритму полягає в тому, щоб простежити хід променя від уявного ока глядача крізь кожен піксел на уявному екрані і обчислити колір об'єкта, видимого оком крізь нього.

* Глобальне освітлення—розрахунок взаємодії поверхонь і середовищ у видимому спектрі випромінювання за допомогою інтегральних рівнянь. Такі алгоритми враховують не лише пряме світло, що надходить безпосередньо від джерела світла, але також і відбиті промені світла від інших поверхонь об'єктів сцени.

**Програмні пакети:**

Autodesk 3ds Max

•Autodesk Maya

•Autodesk Softimage

•Blender

•Cinema 4D

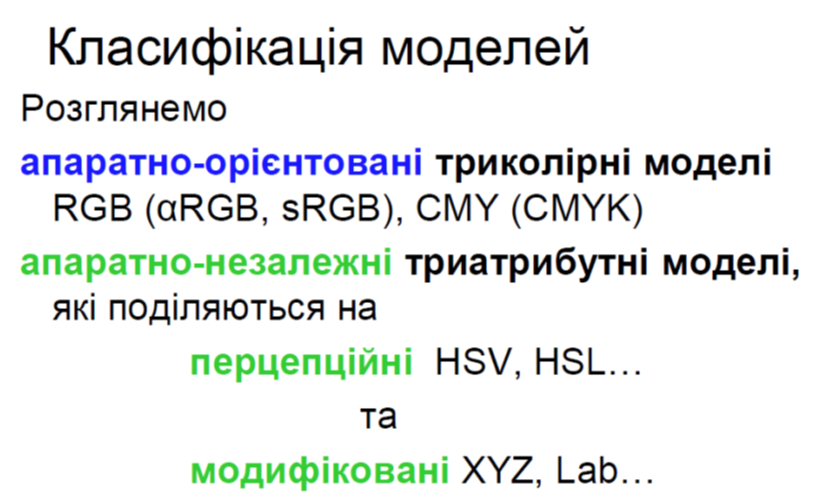
•Zbrush

**Формати файлів:**

* Мова VRML (Virtual Realty Modelling Languagе) призначена для опису тривимірних зображень і оперує об'єктами, що описують геометричні фігури та їх розташування в просторі.
* WRL-файл –звичайний текстовий файл зі списком об'єктів, які названі вузлами.
* 3DS–файл пакету 3D Studio MAX, зберігає 3D-моделі у формі каркасних сіток, текстуру, ефекти тощо.
* МАХ –ще один формат 3D Studio MAX.
* X–“рідний” формат DirectX, структурно-та контекстнонезалежний формат.

**Колір** - це  властивість матеріальних об'єктів, яка сприймається як усвідомлене зорове відчуття та виникає в результаті дії на око потоків видимого електронно-магнітного випромінювання (з довжинами хвиль від 380 до 760 нм).

апаратно-орієнтовані триколірні моделі RGB, CMY та триатрибутні HSV, HSL.

****

**Адитивна модель RGB** - це апаратно-орієнтована модель, в якій кольори описуються за допомогою змішування  трьох базових кольорів – червоного (Red), зеленого (Green), синього (Blue) – в різних пропорціях. Тому модель RGB наз. Адитивною.

Простір RGB геометрично представляють кубом в координатному просторі з початком відліку (0,0,0), вісь Х – компонента червоного кольору, вісь У – зелена компонента, вісь Z – синя компонента.

Будь-який колір у такому випадку представляється точкою з трьома координатами (r,g,b), де r,g,b є [0;1]. Можливе також трактування, що де r,g,b є [0;255].

Недолік RGB - обмеженість у застосуванні, лише на пристроях, які працюють за принципом випромінювання, неможливість відображення деяких кольорів, а саме насичених зелено-синіх.

**Субтрактивна модель CMY** - апаратно-орієнтована модель, яка використовується для формування кольорів на основі принципу віднімання від падаючого світлового потоку частини, яка формується шаром фарби з трьох компонент (блакитний/Cyan, Magenta/пурпурний, Yellow/жовтий).

Тому модель CMY називають субтрактивною (від англ. «sub» віднімати).

Колір в CMY утворюється при відніманні інших кольорів  від загального променя світла (r+g+b).

У цій схемі білий колір утворюється в результаті відсутності всіх кольорів, тоді як їх присутність дає чорний колір.

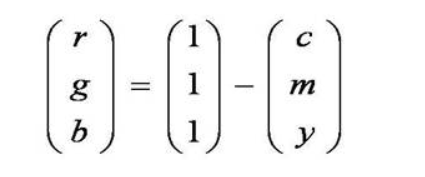
Схема субтрактивних кольорів працює на основі принципу відбиття та поглинання світла.

Недоліки:

* Діапазон кольорів менший
* Інтенсивність кольору не настільки гнучка

У просторі RGB навпроти базових кольорів розміщено базові кольори CMY. Кажуть, що вони доповнюють один одного, поглинають. Кольори моделі CMY є додатковими до кольорів моделі RGB, тобто доповнюючими їх до білого.

Таким чином система координат CMY - той же куб, що і для RGB, але з початком відліку в точці з RGB- координатами (1,1,1), відповідною білому кольору.

Зв’язок адитивної та субтрактивної моделей можна представити математичним виразом.

**Моделі Кольорів HS.** використовуються, щоб позбутися обмежень, що накладаються апаратним забезпеченням. Ці моделі використовуються, коли коректують яскравість, насиченість, перетворюють зображення в сіре та ін.

HSV задається трьома атрибутами:

Hue–колірний тон, визначається довжиною хвилі домінуючої компоненти в спектрі, (“чистий пігмент”). Колірний тон вимірюється в градусах.

Saturation-насиченість, визначається часткою білого кольору. S вказує наскільки колір близький до “чистого” пігменту і визначається відстанню до осі піраміди.

Value–інтенсивність, вказує на загальну кількість світлового потоку, що потрапляє в око, (яскравість). V=0 відповідає вершині і задає чорний колір.

Компоненти S i V змінюються від 0 до 1.

HSL – модель аналогічна до попередньої. Відмінність в компоненті L – освітленість, вказує величину чорного відтінку, доданого до кольору.

Простір HSL – це подвійна піраміда, яку отримують витягуванням  у HSV-піраміді точки S=0, V=1 (білий колір) вгору .

Зміна V дає зміну H. При зменшенні V зелені кольори синіють, сині стають фіолетовими, жовті –оранжевими, оранжеві –червоними.

**Переваги** моделі HSB:

* непогано узгоджується зі сприйняттям людини: колірний тон є еквівалентом довжини хвилі світла, насиченість - інтенсивності хвилі, а яскравість - кількості світла
* дана модель є зручною та зрозумілою і має великий колірний обхват

**Недоліки** моделі HSB:

* необхідність перетворення в модель RGB для відображення на екрані монітора або в модель CMYK для отримання поліграфічного відбитка, а будь-яке перетворення з моделі в модель не обходиться без втрат відображення кольору.

**Модель Lab** базується на людському сприйнятті кольору. При однаковій інтенсивності око людини сприймає промені зеленого кольору найбільш яскравими, дещо менш яскравими –червоного кольору, і ще менш яскравими –синього. Яскравість при цьому є характеристикою сприйняття, а не характеристикою самого кольору. При розробці Lab переслідувалася мета математичного коректування нелінійності сприйняття кольору людиною.

L\* –Ligtness (світлосила, яскравість), в межах [0-100],

a\* –канал кольорів від зеленого до червоного,

b\* –канал кольорів від синього до жовтого, в межах [-200;200]

Оскільки, яскравість у моделі CIELab цілком відділена від кольору, то це робить модель зручною для регулювання тонових характеристик(підвищення контрасту, виправлення похибки тонових діапазонів), видалення кольорового шуму. Враховуючи величезний колірний обсяг, модель знайшла широке застосування в програмному забезпеченні для обробки зображень, через неї відбувається конвертація даних між іншими кольоровими просторами (наприклад, з RGB сканера в CMYK друкованого пристрою).

Перевагою колірної моделі L\*a\*b\* МКО (CIE Lab) є те, що вона не тільки ефективно вирішила проблему розробки рівноконтрастного колірного простору, але й фактично моделює процес представлення кольору апаратом людського зору.

**XYZ–еталонна модель**

Причини створення

Усі кольори, які були отримані Максвеллом шляхом змішування трьох основних кольорів із додатними коефіцієнтами, розташовуються усередині трикутника. Колірний трикутник не показує нам всіх кольорів, які є видимі людському оку. Це саме і є ті кольори, які в дослідах зі зрівнювання кольорів привели до появи від’ємних коефіцієнтів у кривих складання.

Для того, щоб уникнути від’ємних значень кривих додавання, до них були застосовані лінійні перетворення, в результаті яких було отримано нові криві додавання, які позначаються як x(λ) , y(λ) , z(λ) . Ці криві відомі, як криві додавання кольорів для стандартного колориметричного спостерігача МКО 1931 р. Ці кольори є тільки фізичною абстракцією і виконують допоміжну математичну роль.

**Повноколірне кодування** передбачає, що компоненти в моделях задаються числами побайтно. Наприклад, для Windows API таке компонентне кодування кольору в адитивній схемі (4 байти)

00000000 bbbbbbbb gggggggg rrrrrrrr. Перший байт використовується для задання прозорості і тоді використовується система RGBα.

Моделі дозволяють працювати з усіма можливими кольорами. Проте це не завжди потрібно. Можна зекономити пам’ять, якщо працювати з певним набором кольорів, які є актуальними для зображень.

**Палітра кольорів** – це набір актуальних кольорів для певних зображень, представлений у формі таблиці. Палітра застосовується, якщо потреба в економії ресурсів.

* 1. Найпростіший підхід полягає у  переборі всіх пікселів  картинки, і обрахунку, скільки разів зустрічається кожен колір. Палітра складається з тих кольорів, які зустрічаються частіше за інших.

Якщо деякий відтінок синього кольору зустрічається 100 разів, а відтінок червоного тільки 20, то перевага віддається синьому кольору. Основний недолік - деякі кольори будуть зовсім виключені з палітри.

2) Рівномірна розподіленість кольорів - це вибір комплекту кольорів для палітри з рівномірно розподіленими червоною, зеленою і синьою компонентами. Такий підхід забезпечує широкий вибір кольорів.

 Але при цьому не враховується, що в більшості картинок немає рівномірного колірного розподілу.

3) **Метод квантування кольорів медіанним перетином**

Колірний простір розглядається як тривимірний куб. Кожна вісь куба відповідає одному з трьох основних кольорів: червоному, зеленому, синьому. Кожна з трьох сторін розбивається на 255 рівних частин, поділки на осях нумеруються від 0 до 255.

Перший крок полягає у відображенні усіх колірних точок.

Другий крок полягає у відсіканні "країв" куба, які не містять пікселів.

Третій крок - розрізання отриманого паралелепіпеда на два в серединній точці (медіані) найдовшої сторони.

Тепер паралелепіпед розділений на два паралелепіпеди меншого розміру.

Наступні кроки такі ж. Далі попередній процес - відсікання порожніх "країв" і розрізання найдовшої сторони в серединній  - повторюється для двох менших паралелепіпедів.  Тепер початковий куб роздільний на чотири паралелепіпеди.

Медіанний перетин повторно застосовується для того, щоб розділити куб на 8, 16, 32, 64, 128 і 256 паралелепіпедів.

Формування палітри відбувається за центрами паралелепіпедів. Кожен з 256 паралелепіпедів містить піксели приблизно однакового кольору і центр кожного паралелепіпеда представляє оптимальне значення кольору для палітри.

**Wireframe** (каркас) –це простий наочний посібник, який представляє скелетну структуру веб-сайту, системи чи іншого цифрового продукту. Wireframes в деталях показують, яка інформація, контент і елементи управління повинні виводитися на кожній сторінці інтерфейсу будь-якої системи, створюють правильні взаємодії з інтерфейсом та сприяють покращенню користувацького досвіду.

**Типи Wireframes**

Існує три рівні каркасного зв’язку: низький (low fidelity), середній (medium fidelity), та високий (high fidelity).

**ЕТАПИ СТВОРЕННЯ MEDIUM FIDELITY WIREFRAMES**

Етап 1. Дослідження. Зробіть ретельне дослідження. Не починайте процес каркасного проектування до того, як не окреслили цілі проекту, визначили цільову аудиторію та зрозуміли проблеми, які вирішуються продуктом.

Етап 2. Скетч. Мета ескізу - створити структуру дизайну сторінки. Прибережіть тонкі деталі на потім, на даному етапі обдумайте -який функціонал і тип контенту ви будете розміщувати на сторінці.

Етап 3.Сітка. Перед налаштуванням сітки, задайте розміри сторінки (артборду/фрейму). Відштовхуючись від того чи ви працюєте з десктопом чи мобільним інтерфейсом. Далі перейдіть до налаштування сітки та контейнерів. Сітка -це структуроване і просте розташування елементів.

Етап 4. Планування контенту. На даному етапі -проходимо ітерацію низькорівневих Wireframes та створюємо розмітку компоновки (layout) прямокутниками.

Етап 5. Побудова інформаційної ієрархії. Коли вас влаштовує розташування блоків, приступайте до поступового наповнення їх текстом, щоб отримати уявлення, чи добре структурована інформація.. Використовуйте різний розмір шрифту, щоб почати виділяти різні види інформації.. На даному етапі починаємо працювати з функціоналом, текстом, кнопками, інпутами та ін.

Етап 6. Wireflow\*. Wireflow-це поєднання user flow та wireframes. Він потрібен для того щоб розуміти загальну архітектуру проекту, послідовність та розгалуження сторінок.

Точки доступу, де користувачі взаємодіють з дизайном інтерфейсу бувають **трьох форматів**:

* Графічні інтерфейси користувача (GUI)-користувачі взаємодіють з візуальними зображеннями на цифрових панелях управління. Робочий стіл комп’ютера -це графічний інтерфейс.
* Голосові інтерфейси (VUI)-користувачі взаємодіють з ними за допомогою голосу. Більшість розумних помічників -наприклад, Siri на iPhone та Alexa на пристроях Amazon -це VUI.
* Інтерфейси на основі жестів-користувачі взаємодіють із просторами 3D-дизайну за допомогою фізичних рухів: наприклад, у іграх з віртуальною реальністю (VR).

**UI-kit** -це частина дизайн-системи, бібліотека компонентів укоді. Ці елементи малює дизайнер, а фронтенд-розробник регулює їх стан устатиці, при наведенні, в русі. Бібліотека може містити елементи фірмового стилю: типографіку, кольори, інтервали, анімацію, іконки йілюстрації, а може тільки елементи інтерфейсу: кнопки, підказки, меню, чекбокси.

**Композиція**(складання та зв’язування)

Це побудова цілісного витвору, всі складові якого гармонійно узгоджуються. Вона допомагає:

* -Контролювати увагу користувачів. У цьому випадку сприйняття користувача є свого роду шлях, по якому він проходить через інтерфейс і вивчає його візуальні компоненти. Користувач буде слідувати за маршрутом, який йому заздалегідь зададуть.
* -Концентрувати увагу користувача. Люди легше сприймають структуровану інформацію, тому що докладають менше зусиль на її засвоєння. У цифровому світі розглядають різні шаблони сприйняття інформації, найбільш поширені -це F і Z шаблони сприйняття контенту користувачами. Вони пояснюють як наші очі натреновані на пошук інформації.

**Евристики UI**

1) Зробіть кнопки та інші поширені елементи інтерфейсу передбачуваними (включаючи адаптивність та масштабування), щоб користувачі могли їх несвідомо використовувати скрізь. Форма повинна відповідати функції.

2) Робіть правильні акценти та ієрархію. Слід чітко розуміти на чому повинен зосередитись користувач, який основний фокус його уваги і куди зміститься даний фокус при певній дії.

3) Всі елементи повинні бути вирівняні, пам’ятайте про сітку.

4) Зверніть увагу на ключові функції, використовуючи колір, яскравість і контраст. Уникайте використання широкого спектру кольорів, пропрацюйте свою колірну гаму, адже колір тільки посилює дизайн а не формує його.

5) Створіть ієрархію шрифтів, які ви будете використовувати. Текст подається через розмір шрифту, його вагу, динаміку, розмір літер та відстань між ними. Користувачі повинні знаходити потрібну інформацію просто скануючи, без зайвих зусиль.

6) Мінімізуйте кількість дій для виконання завдань, зосередьтеся головній функції на сторінці. Направляйте користувачів, вказуючи бажані дії. Полегшуйте складні завдання, використовуючи прогресивне розкриття інформації.

7) Розташуйте елементи керування поблизу об’єктів, якими користувачі хочуть керувати. Наприклад, кнопка для подання форми повинна знаходитись поруч із формою.

8) Тримайте діалог між користувачем та системою. Інформуйте користувачів про відповіді чи дії системи, підтримуйте зворотній зв'язок.

9) Допомагайте користувачу, якщо є така можливість. Наприклад, користувачу потрібно заповнити форму, ви можете додати автозаповнення вже наявною інформацією.

10) Завжди надавайте наступні кроки, які користувачі можуть зробити природним чином, незалежно від їхнього контексту.

**10 Евристик Нільсена**

* 1. ***Видимість стану системи***

Користувач завжди повинен орієнтуватися і добре розуміти, що відбувається в системі. Взаємодія між користувачем і системою повинна бути якомога більш логічною і швидкою.

* 1. ***Поєднання системи та реального світу***

Система повинна спілкуватися з користувачем на зрозумілій йому мові. Використання слів, фраз і понять, знайомих користувачеві в реальному світі, набагато краще, ніж використання спеціалізованих термінів.

* 1. ***Контроль та свобода користувачів***

Дайте користувачам можливість скасування дій, а також повернення до раніше скасованих дій. Користувачі часто вибирають системні функції помилково, і їм потрібен чітко позначений “аварійний вихід”, щоб залишити небажаний стан без необхідності проходити через розширений діалог з системою.

* 1. ***Послідовність та стандарти***

Користувачі не повинні задаватися питанням, чи означають різні слова, ситуації чи дії одне й те саме. Не плутайте користувача, описуючи одні й ті ж речі різними словами і термінами. Дотримуйтеся однаковості і використовуйте стандартні комунікаційні елементи.

* 1. ***Попередження помилок***

Ретельна конструкція, яка запобігає виникненню проблем – це набагато краще, ніж хороші повідомлення про помилки. Зведіть до мінімуму кількість умов, в яких можуть бути допущені помилки або усуньте умови, що можуть призвести до помилок в роботі з системою, або надайте користувачам можливість підтвердження, перш ніж вони зроблять дію.

* 1. ***Впізнавання замість згадування***

Не змушуйте користувача запам’ятовувати велику кількість інформації, зробивши очевидними предмети, дії та параметри. Відвідувач не повинен тримати в голові інформацію, переходячи з однієї частини системи в іншу. Інструкції щодо використання системи повинні бути видимими або легко доступними, наскільки це можливо.

* 1. ***Гнучкість та ефективність використання***

Прискорювачі, котрі невидимі для користувача початківця – часто можуть пришвидшити взаємодію для досвідченого користувача. Система повинна ефективно обслуговувати як недосвідчених, так і досвідчених користувачів. Спростіть функції, якими відвідувачі вашого сайту користуються найчастіше.  
Не перевантажуйте досвідчених користувачів зайвою інформацією, надайте їм можливість здійснювати часто повторювані дії якомога швидше і простіше.

* 1. ***Естетичний та мінімалістичний дизайн***

Комунікація з користувачем не повинна містити інформації, яка не має значення або рідко потрібна. Кожна додаткова одиниця інформації в діалозі конкурує з корисною інформацією та зменшує її відносну видимість.  
Тексти не повинні містити непотрібної або застарілої інформації. Кожне зайве слово погіршує сприйняття і зменшує ймовірність знайти те, за чим користувач прийшов на сайт.

* 1. ***Допоможіть користувачам розпізнати, діагностувати та відновитись після помилок***

Повідомлення про помилки повинні бути виражені зрозумілою користувачеві мовою, як максимально точно описувати проблему і пропонувати конструктивні варіанти її вирішення.

* 1. ***Довідка та документація***

Дуже добре, якщо систему зручно використовувати без документації, але може виникнути потреба надати допомогу користувачам. Будь-яку таку інформацію слід розмістити у швидкому і зручному доступі, вона має бути орієнтована на завдання користувача. Якщо потрібно, то варто описати конкретні кроки, які слід виконати, вони не повинні бути трудоємкими. Подібні документи повинні бути складені таким чином, щоб в них легко було знайти необхідну інформацію.