1) Особливості зафарбування трикутників. Який алгоритм краще застосувати? Чому?

Краще використовувати алгоритм заповнення лініями. Він більш оптимальний ніж заповнення точками, адже не потрібно перевіряти, чи міститься кожна точка всередині зображення. Достатньо посортувати вершини по у-координаті, обчислити початкову і кінцеву точки ліній та намалювати відрізок.

2) Коли необхідно знаходити нормаль? Як її обрахувати ?

Нормаль – вектор, перпендикулярний до площини. У комп’ютерній графіці нормалі використовуються при обчисленні розсіяної(дифузної) та відбиваючої(спекулярної) складових освітлення, зокрема використовується кут між нормаллю поверхні і вектором від джерела світла.

Для того, щоб обчислити нормаль, наприклад до площини трикутника, потрібно виконати деякі перетворення з координатами його вершин.

Нехай маємо вектор нормалі n(xn,yn,zn)

xn=(y2 - y1) \* (z3 - z1) - (z2 - z1) \* (y3 - y1)  
yn=(z2 - z1) \* (x3 - x1) - (x2 - x1) \* (z3 - z1)  
zn=(x2 - x1) \* (y3 - y1) - (y2 - y1) \* (x3 - x1)

3) Системи координат - світові, екранні. Як і де використовують ?

Система координат — спосіб задання точок простору за допомогою чисел.

Світові системи координат описують стан об’єктів у просторі(тривимірні).

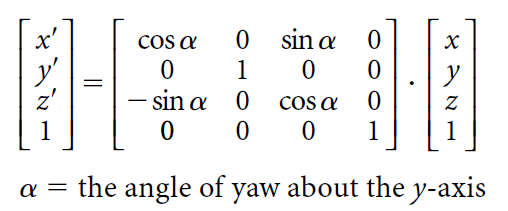
Екранні координати використовуються для відображення на екрані, площині(в заданій проекції), тому вони зазвичай двовимірні.

Щоб перейти від світових координат до екранних:

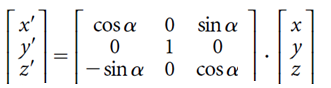
* Перетворюємо світові координати у видові (враховуючи поворот камери)
* Спроектувати їх на площину та отримати координати проекції
* Застосувати масштабування, зсув та інші необхідні перетворення для отримання реальних екранних координат.

4) Які необхідно виконати перетворення при оберті об'єкта навколо осі Y ?

При обертанні об’єкта навколо осі у змінюються його у-координати. Зазвичай для обертання навколо прямої потрібно побудувати матрицю обертання. Так як вісь ординат – не просто пряма, матриця обертання навколо осі ординат відома:

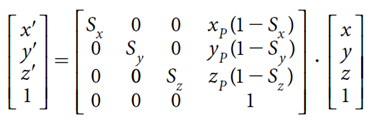


Тобто, щоб обернути об’єкт навколо осі У, необхідно вектор його координат помножити на цю матрицю.



5) Які перетворення потрібно виконати при масштабуванні об'єкта?

Масштабування об’єкта – це його проекція на паралельну площину, збільшена в кілька разів. Тобто, необхідно всі вектори координат об’єкта помножити матрицю

де Sx Sy Sz – числа масштабування

6) Особливості перспективної та аксонометричної проекції. Як і коли застосовують?

Аксонометрична проекція - наочне зображення просторових форм на площині методом паралельного проектування, коли всі проектуючі промені розміщуються під прямим кутом до площини проектування.

Особливості аксонометричної проекції:

* проекції відрізків паралельних прямих завжди паралельні;
* довжини проекцій однакових відрізків паралельних прямих завжди однакові, незалежно від відстані до площини проектування.

Перспективна проекція – нелінійна проекція. Щоб побудувати проекцію потрібно задати точку, яка називається центром проекції. Проекції будуються за допомогою проектуючих променів, які виходять з центру проекції(вони не паралельні).

Особливості перспективної проекції:

* Співвідношення довжин і площ не зберігається
* Прямі лінії зображуються прямими лініями;
* Прямі, паралельні площині екрану, зображуються паралельними прямими;
* Паралельні прямі, що не паралельні площині екрану, зображуються прямими, що сходяться в одній точці;
* Довжини проекцій однакових відрізків прямих будуть різними в залежності від їх віддаленості від площини проектування.

7) Проектування 3D об'єктів на 2D площину. Який алгоритм найпоширеніший ? Який краще?

Проектування 3D об'єктів на 2D площину - це будь-який спосіб відображення тривимірних точок на двовимірній площині. Основна задача проектування – визначити, які частини об'єктів видно з обраного положення перегляду.

Найпоширенішим на сьогодні залишається алгоритм Художника, адже навіть на його базі грунтуються інші алгоритми (Depth Buffer - це попіксельне узагальнення алгоритму художника)

В основі алгоритму художника – відсортовуємо об’єкти за Z-координатою і малюємо їх від найдальшого до найближчого. Хоч цей алгоритм є найпоширенішим, він не є найкращим, адже часто при сортуванні по Z-координаті виникають колізії, зокрема при циклічному перекритті об’єктів.

Найкращим вважається алгоритм Scan-Line. В основі – дві таблиці — ребер та багатокутників в поверхні. Перевага: немає необхідності копіювати координати всіх вершин — лише вершини, що потрапляють в зону видимості. Кожна з вершин зчитується лише раз, що значно підвищує швидкість виконання.

8) Опишіть основні крокі виконання завдання побудови 3D моделі з обертанням та освітленням.

1. Зчитати з файла координати 3Д-моделі
2. відсортувати по З-координаті
3. Обчислити значення трьох складових освітлення
4. вивести на екран початкове зображення з освітленням
5. обчислити координати нового зображення при обертанні (наприклад, з використанням матриці)
6. перейти на крок 3

9) Основні кроки та функції виконання застосування в OpenGL.

* Спочатку, ініціалізуємо вікно. Для цього налаштовуємо відповідні буфери кадру, початкового положення і розмірів вікна, а також заголовка вікна.
* Ініціалізуємо бібліотеку GLUT glutInit()
* Задаємо початкові розміри вікна glutInitWindowSize (Width, Height)
* Задаємо заголовок та створюємо саме вікно glutCreateWindow ( “Glut Shapes”)
* функції Display (), Reshape () і Keyboard () - функції, що викликаються при перемальовуванні вікна, зміні розмірів вікна, та натисканні клавіш на клавіатурі.
* Усі події та виклики функцій здійснюємо у glutMainLoop ().

10) Застосування проекції в OpenGL.

Зазвичай застосовується паралельна ортогональна проекція  
Дл цього викликаємо функцію:

glOrtho (left: GLdouble, right: GLdouble, bottom: GLdouble, top: GLdouble, near: GLdouble, far: GLdouble)

gluOrtho2D (left: GLdouble, right: GLdouble, bottom: GLdouble, top: GLdouble)

Параметри near і far задають відстань до ближньої і дальньої площин і можуть бути від'ємними.

В gluOrtho2D near та far за замовчуванням −1 та 1

Перспективна проекція:

gluPerspective (angley: GLdouble, aspect: GLdouble, znear: GLdouble, zfar: GLdouble)

angley визначає кут видимості в градусах по осі У і мусить бути від 0 до 180 градусів. Параметри zfar і znear задають відстань від спостерігача до площин відсікання по глибині і повинні бути позитивними(на відміну від ортогональної, де можуть бути негативні)

11) Навіщо використовувати буфери в OpenGL?

В OpenGL є кілька буферів кольору, буфер глибини(Z-буфер), буфер-накопичувач (accumulation buffer), буфер маски.

На екрані може з'явитися інформація тільки з буферів кольору.

Буфер глибини використовується для видалення невидимих з поточної перспективи об’єктів.

Буфер - накопичувач можна застосовувати для різних операцій.

Буфер маски використовується для формування піксельних масок (шаблонів), для вирізання із загального масиву тих пікселів, які слід вивести на екран.

При використанні подвійної буферизації говорять про робочий (front) і фоновий (back) буфер. В фоновому буфері створюється зображення, яке потім копіюється в робочий буфер. Тому, получається, що не виникає затримки на створення зображення при анімації. Але при цьому буфери займають значний об’єм пам’яті(наприклад, комп’ютерні ігри іноді «підвисають» через це).