

Контрольна робота з курсу "Комп'ютерна графіка"

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

Scan-Line Algorithm, Boundary Fill Algorithm, Алгоритм Форчуна, Алгоритм Брезенхейма.

Алгоритм Брезенхейма найкращий.

2. Коли необхідно знаходити нормаль? Як її обрахувати ?

В залежності від того, до чого необхідно знайти нормаль. Наприклад, нормаль до вектору знаходиться за допомогою ділення кожної координати вектора на його довжину.

Для обрахування дифузного освітлення необхідно вирахувати нормаль полігону.

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

Світові координати описують стан об'єктів в просторі.

Екранні координати - в системі координат (зазвичай двовимірної) пристрою відображення: зображення об'єктів в заданій проекції зображення тривимірного об'єкту на пристрої відображення необхідно задати перетворення координат з світових в екранні.

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

Необхідно визначити матрицю перетворення:

$$M_y(\alpha) = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

і помножити вектор координат на неї.

Таким чином отримуємо, що нові координати будуть дорівнювати:

$$\begin{aligned} x' &= x \cdot \cos(L) + z \cdot \sin(L) ; \\ y' &= y ; \\ z' &= -x \cdot \sin(L) + z \cdot \cos(L) ; \end{aligned}$$

Це необхідно зробити для кожного полігону об'єкту, і для того, щоб обертання було відносно об'єкту, а не окремо для полігону, необхідно відняти від координат полігону координати об'єкту, помножити на матрицю перетворення, а потім додати координати об'єкту.

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

Для масштабування об'єкту на коефіцієнт S необхідно зробити все так само як і з обертанням, але матриця перетворення буде =

$$\begin{vmatrix} S & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & S & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

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

Аксонометрична проекція — різновид паралельної проекції, коли всі проєктуючі промені розміщуються під прямим кутом до площини проєкування $Z(0) \rightarrow X, Y$. Проекції відрізків паралельних прямих завжди паралельні; довжини проекцій однакових відрізків паралельних прямих завжди однакові, незалежно від їх віддаленості від площини проєкування.

Перспективна проекція — не зберігається співвідношення довжин і площ; паралельні 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 різні, залежно від їх віддаленості від площини проєкування.

Вид перспективної проекції визначається взаємним розташуванням об'єкта, площини проєкування та точки фокуса. Щоб зображення виглядало природним, потрібно, щоб кут L , під яким ми бачимо зображення (на екрані, аркуші та ін.) Відповідав кутку L' , під яким ми бачимо об'єкт. Якщо фокусна відстань занадто мала і кут L' значно більше кута L , отримаємо спотворене зображення з перебільшеною перспективою. Якщо фокусна відстань велика, а кут L' малий, зображення буде близьким до аксонометричної проекції і не буде передавати відчуття глибини простору. У разі рівного розподілу кутів L' і L зображення буде найбільш природним. Проектування на площину зображень з великим кутом огляду призводить до спотворень.

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

Наприклад, для отримання панорамних зображень використовують проектування на циліндричну та сферичну поверхні. В циліндричній проекції точка фокусу розташовується на осі циліндра, а зображення будується по точках перетину променів з поверхнею циліндра.

Циліндрична проекція використовується в панорамних кінотеатрах, де зображення проєкується на вигнутий екран - глядач бачить події, що відбуваються не тільки перед ним, але і зліва, справа і навіть ззаду.

В сферичній проекції поверхнею проєкування служить сфера, а точка фокусу розташовується в її центрі. Сферична проекція використовується в планетаріях для імітації зоряного неба на напівсферичним екрані.

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

Визначити функції для обертання 3D об'єкту через матричні перетворення, для кожного кадру відмальовуємо модель з невеликим обертанням.

Для освітлення спочатку необхідно встановити параметри: ambient coefficient (загальне освітлення всієї сцени або фонове освітлення), diffuse coefficient (розсіяна складова), specular coefficient (дзеркальна складова), напрям світла. Потім при рендері кожного полігону необхідно вирахувати його колір, базуючись на його куті відносно напрямку світла. Для цього вираховуємо вектор нормалі для цього полігону:

```
normal_vector: np.array = np.cross((p.v3 - p.v1).np_array, (p.v2 - p.v1).np_array)
```

```
normal_vector = normal_vector / np.linalg.norm(normal_vector)
```

і множимо його на напрям світла, для отримання значення дифузного освітлення:

```
diffuse_light = max(self._light_direction.dot(normal_vector), 0)
```

Для отримання дзеркального освітлення необхідно вирахувати напрям віддзеркалення і помножити його на напрям світла:

```
reflect_dir = 2 * normal_vector * (normal_vector.dot(self._light_direction) / normal_vector.dot(normal_vector)) - self._light_direction
```

```
specular_light = pow(max(self._view_dir.dot(reflect_dir), 0.0), 32)
```

В кінці множимо отримані значення на коефіцієнти визначені в самому початку і сумуємо:

```
return self._ambient_coef + self._diffuse_coef * diffuse_light + self._specular_coef * specular_light
```

І множимо колір (кожну складову r, g, b) на це число і отримаємо колір полігону.

9. Основні кроки та функції виконання застосування в OpenGL.

Основні кроки — Ініціалізації вікна, налаштування функцій з зворотним викликом і головного циклу обробки подій.

Ініціалізація вікна складається з налаштування відповідних буферів кадру, початкового положення і розмірів вікна, а також заголовка вікна. Функція `glutInit` ініціалізує бібліотеку GLUT.

Команда `glutInitDisplayMode` ініціалізує буфер кадру і налаштовує повноколірний режим RGB. `glutInitWindowSize` використовується для завдання початкових розмірів вікна.

`glutCreateWindow` задає заголовок вікна і візуалізує саме вікно на екрані.

Функції Display, Reshape і Keyboard викликаються відповідно, при перемальовуванні вікна, зміні розмірів вікна, та натисканні клавіші на клавіатурі. Контроль за усіма подіями і виклик потрібних функцій відбувається всередині нескінченного циклу в функції glutMainLoop.

10. Застосування проекції в OpenGL.

Проекції в OpenGL бувають паралельними і перспективними. Паралельні проекції можна створювати за допомогою функції glOrtho та gluOrtho2D. Перспективні — gluPerspective. glViewport — вивід.

11. Навіщо використовувати буфери в OpenGL?

Під час виклику функцій відмальовки, заповнюється декілька буферів — буфер кольору, буфер глибини, буфер трафарету. Також можна прив'язувати додаткові буфери. Головна мета використання — створення різних спецефектів та прив'язування додаткових текстур до зображення.