

Вопросы для самоконтроля по теме 5

- | | |
|---|---|
| 1) Назовите наиболее распространенные способы представления пространственных форм . | 1 |
| 2) Что собой представляют параметрические бикубические куски? | 2 |
| 3) Что собой представляет полигональная сетка ? | 2 |
| 4) Какие типы элементов должна хранить полигональная сетка? | 2 |
| 5) Назовите способы представления полигональной сетки их особенности, достоинства и недостатки. | 2 |
| 6) Назовите свойства полигональных сеток . | 3 |
| 7) Что такое триангуляция ? | 4 |
| 8) Назовите теорему о существовании триангуляции . | 4 |
| 9) Какой многоугольник называется простым? | 4 |
| 10) Назовите алгоритмы триангуляции . | 4 |
| 11) Опишите шаги алгоритма триангуляции «разделяй и властвуй» | 4 |
| 12) Назовите алгоритмы удаления невидимых линий , сущности этих алгоритмов, их достоинства и недостатки. | 5 |
| 13) Назовите правила задания граней . | 7 |
| 14) Какие алгоритмы удаления невидимых линий реализованы в OpenGL? | 8 |
| 15) Что произойдет, если грани задавать с использованием разных обходов вершин (и против часовой стрелки и по часовой стрелки)? | 8 |

- 1) Назовите наиболее распространенные способы представления пространственных форм.

Полигональная сетка и параметрические бикубические куски.

- 2) Что собой представляют параметрические бикубические куски?

Параметрические бикубические куски - это совокупность связанных между собой частей криволинейных поверхностей. Аналогично В-сплайну ПКБ позволяет осуществлять интерполяцию (сглаживание) некоторой совокупности точек, только в данном случае не с помощью кривых линий, а с помощью кривых поверхностей. Границами поверхностей являются кубические кривые.

- 3) Что собой представляет полигональная сетка?

Полигональная сетка - это совокупность вершин, ребер и граней, которые определяют форму многогранного объекта в трехмерной компьютерной графике и объемном моделировании. Гранями обычно являются треугольники, четырехугольники или другие простые выпуклые многоугольники (полигоны), так как это упрощает рендеринг. Полигональная сетка представляет собой совокупность связанных между собой плоских многоугольников и наиболее хорошо описывает объекты с плоскими поверхностями.

- 4) Какие типы элементов должна хранить полигональная сетка?

Объекты, созданные с помощью полигональных сеток должны хранить разные типы элементов, такие как **вершины, ребра, грани, полигоны и поверхности**. Во многих случаях хранятся лишь вершины, ребра и либо грани, либо полигоны.

5) Назовите способы представления полигональной сетки их особенности, достоинства и недостатки.

Способы представления полигональной сетки:

- явное задание граней;
- задание граней с помощью указателей в списке вершин;
- явное задание ребер;
- другие.

Они отличаются:

- объемом требуемой памяти;
- определением ребер, принадлежащих вершине;
- определением многоугольников, общих для ребра;
- поиском вершин, образующих ребро;
- определением всех ребер, образующих грань;
- изображением полигональной сетки;
- обнаружением ошибок в представлении.

Недостатками **явного задания граней** являются избыточность (требует больше памяти), неудобство изменений, отсутствие явного описания общих ребер и вершин для соседних многоугольников.

Достоинствами **задания граней с помощью указателей в списке вершин** являются необходимость в меньшем количестве памяти, так как вершины задаются единожды, а также способность легко изменять координаты вершин. Недостатком является сложность в нахождении многоугольников с общими ребрами.

Достоинствами **явного задания ребер** является необходимость в меньшем количестве памяти, так как вершины задаются единожды, способность легко изменять координаты вершин, а также явное задание общих ребер.

6) Назовите свойства полигональных сеток.

Свойствами полигональных сеток являются:

- **Монолитность.** Сетка представляет собой монолитный объект, если совокупность его граней заключает в себе некоторое конечное пространство;
- **Связность.** Сетка называется связной, если между любыми двумя вершинами существует непрерывный путь вдоль ребер полигона;
- **Простота.** Сетка называется простой, если отображаемый ею объект является монолитным и не содержит отверстий. Это означает, что объект может быть деформирован в сферу, не подвергаясь разрезанию;
- **Плоскостность.** Сетка называется плоской, если каждая грань представляемого ею объекта является плоским полигоном, то есть вершины каждой грани лежат в одной плоскости;
- **Выпуклость.** Сетка представляет выпуклый объект, если прямая, соединяющая любые две ее точки внутри этого объекта, целиком лежит внутри него.

7) Что такое триангуляция?

Триангуляция - это процесс разбиения полигональной области со сложной конфигурацией в набор треугольников.

8) Назовите теорему о существовании триангуляции.

У любого простого n -вершинного многоугольника P всегда существует триангуляция, причем количество треугольников в ней $n-2$ независимо от самой триангуляции.

9) Какой многоугольник называется простым?

Простым многоугольником является фигура, ограниченная одной замкнутой ломаной, стороны которой не пересекаются.

10) Назовите алгоритмы триангуляции.

Алгоритмы триангуляции:

- Жадный алгоритм;
- Оптимальная триангуляция;
- Алгоритм с индексированием треугольников;
- Алгоритм Зейделя;
- Инкрементальный алгоритм построения триангуляции Делоне;
- Алгоритм “Разделяй и властвуй”;
- Другие.
-

11) Опишите шаги алгоритма триангуляции «разделяй и властвуй»

Самым простым алгоритмом триангуляции является метод “Разделяй и властвуй”.

Многоугольник рекурсивно делится на части путем проведения хорды вплоть до треугольников. Общая стратегия триангуляции состоит из двух этапов:

1. Декомпозиция полигона на монотонные части;
2. Триангуляция монотонных частей.

Шаг 1. Разбиение исходного множества на более мелкие множества. Для этого производятся вертикальные или горизонтальные прямые в середине множества и уже относительно этих прямых разделяются точки на две части, примерно по $N/2$. После, для каждой группы точек рекурсивно запускается процесс деления в зависимости от их количества.

- Если число точек $N > 12$, то делим множество с помощью прямых;
- Если число точек $N \leq 12$, то делим множество на 3 и $N-3$ точек;
- Если число точек $N = 8$, то делим множество на 2 группы по 4 точки. Деление продолжается до тех пор, пока не останется 3 или 4 точки.

Шаг 2. Построение триангуляции для множества из 3 или 4 точек. Для трех точек триангуляция очевидна - просто соединяем попарно точки отрезками. Для четырех точек возможны два варианта:

- Если точки образуют невыпуклый четырехугольник, то просто соединяем все 4 точки отрезками;
- Если точки образуют выпуклый четырехугольник, то берем любые 3 точки и проверяем положение четвертой точки относительно окружности, описанной вокруг первых трех точек.

Шаг 3. Объединение оптимальных триангуляций. Сначала находятся две пары точек, отрезки которых образуют в совокупности с построенными триангуляциями выпуклую фигуру. Они соединяются отрезками, и один из полученных отрезков выбирается как начало для последующего обхода.

12) Назовите алгоритмы удаления невидимых линий, сущности этих алгоритмов, их достоинства и недостатки.

Алгоритм художника

Подразумевает последовательное отображение граней в порядке уменьшения удаленности от ТН (*точки наблюдения ?*) и их заливкой

Недостатки: Не универсальность

Алгоритм z-буфера

В нем используется буфер как часть памяти, в которой для каждого пикселя на экране хранится некоторое число z , обозначающее удаленность от точки наблюдения (видовое z) последней точки объекта, спроецированной в данный пиксель

Алгоритм:

1. Для всех пикселей установить z
2. Перебор граней (произвольно)
 - а) Проецирование (определение совокупности пикселей, принадлежащих грани)
 - б) Перебор пикселей
 - Вычисление z_e (преобразование из пиксельной в видовую СК)
 - Если $z_e < z_b$ подсветка пикселя и $z_b = z_e$ (обновление z_b для данного пикселя)

Достоинство - универсальность

Недостатки: Большой объем памяти под буфер и
Низкое быстродействие (избыточность пикселей, ПСК ВСК)

Алгоритм, основанный на ориентации вершин

Объект состоит из граней, грани - из ребер, соединяющих вершины

(Упрощение - треугольные грани)

Структура ФД (?) :

- 1) Объектная точка
- 2) Блок описания вершин
- 3) Блок описания граней

Вершины обходятся против часовой стрелки при наблюдении грани с внешней стороны (не через тело объекта)

При проецировании (задана ТН (точка наблюдения)):

грани, для которых ТН расположена с той же стороны (внешней), что и при задании объекта имеют тот же порядок обхода (против часовой)

грани, для которых ТН расположена с противоположной, чем при задании объекта, стороны имеют противоположный порядок обхода (по часовой)

Для треугольника на плоскости ориентация вершин определяется знаком детерминанта D

$D < 0$ - обход вершин по часовой стрелке

$D > 0$ - обход вершин против часовой стрелки

$D = 0$ - вершины лежат на прямой

Строки в матрицу заносятся в порядке следования вершин из файла данных

Общая схема алгоритма

Разбиение многоугольников на треугольники (самостоятельный алгоритм)

Отсев самозакрывающихся граней (см.ранее)

Обработка ребер (общая идея)

13) Назовите правила задания граней.

Вершины граней стоит задавать против часовой, чтобы OpenGL подсвечивал эту грань цветом, иначе грань будет считаться не видимой и не будет подсвечиваться.

1. Грани должны быть плоскими многоугольниками;
2. При описании грани необходимо смотреть на нее с внешней стороны, а не через тело объекта;
3. Обход вершин, образующих внешние ребра, осуществляется против часовой стрелки;
4. Последовательность вершин при описании грани является циклической, т.е. считается, что за последней в списке вершиной идет первая, встречающаяся в нем;
5. Обход вершин, принадлежащих отверстию, осуществляется по часовой стрелке;
6. При переходе от внешних ребер к ребрам, образующим отверстие и обратно, используются отрицательные номера вершин;
7. При описании грани первые три вершины в последовательности не должны образовывать внутренний угол более 180 градусов;
8. Свободные отрезки в блоке описания граней задаются лишь двумя номерами вершин;
9. Бесконечно тонкие грани описываются с двух сторон;
10. Не плоские грани разбираются на треугольники.

14) Какие алгоритмы удаления невидимых линий реализованы в OpenGL?

Буфер глубины (z-буфер) и удаление нелицевых граней.

15) Что произойдет, если грани задавать с использованием разных обходов вершин (и против часовой стрелки и по часовой стрелки)?

