

Коллоквиум по вр:

#### 1. Дать определение компьютерной графике.

Компьютерная графика (также машинная графика) — область деятельности, в которой компьютеры наряду со специальным программным обеспечением используются в качестве инструмента как для создания (синтеза) и редактирования изображений, так и для оцифровки визуальной информации, полученной из реального мира, с целью дальнейшей её обработки и хранения.

#### 2. Типы графики

- Научная графика
- Деловая графика
- Инженерная графика
- Иллюстративная графика
- Художественная и рекламная графика
- Пиксель арт

По способам задания изображений графику можно разделить на категории:

- Двухмерная графика
- Векторная графика
- Растровая графика
- Фрактальная графика
- Трёхмерная графика
- CGI графика

#### 3. Разница между векторной и растровой графикой.

Растровое изображение, как мозаика, складывается из множества маленьких ячеек — пикселей, где каждый пиксель содержит информацию о цвете. Определить растровое изображение можно увеличив его масштаб: на определённом этапе станет заметно множество маленьких квадратов — это и есть пиксели.

Наиболее распространённые растровые форматы: JPEG, PNG.

В отличие от растровых, векторные изображения состоят уже не из пикселей, а из множества опорных точек и соединяющих их кривых. Векторное изображение описывается математическими формулами и, соответственно, не требует наличия информации о каждом пикселе. Сколько ни увеличивай масштаб векторного изображения, вы никогда не увидите пикселей.

Самые популярные векторные форматы: SVG, AI.

#### 4. Что такое фрактальная графика

Это одна или несколько геометрических фигур, каждая из которых подобна другой. То есть, изображение составляется из одинаковых частей.

Само слово "фрактал" может употребляться, если фигура обладает одним или несколькими из этих свойств:

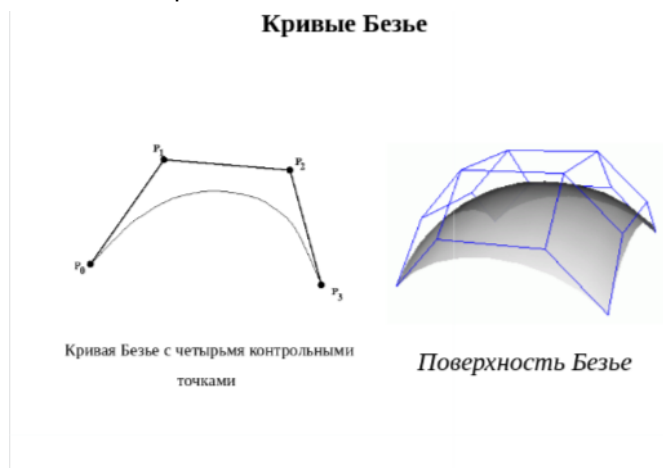
- Нетривиальная структура. Когда рассматривается небольшая деталь всего изображения, то фрагмент схож со всем рисунком. Увеличение масштаба не приводит к ухудшению. Изображение всегда остается одинаково сложным.
- Каждая часть рисунка является самоподобной.
- Имеется математическая размерность.
- Строится при помощи повторения.

## 5. Что такое кривая Безье

Кривая Безье — это математически описанная кривая, используемая в компьютерной графике и анимации. В vector images они используются для создания плавных кривых, которые можно бесконечно сильно масштабировать.

Кривая описывается массивом контрольных точек, которых должно быть, как минимум, две. В веб-графике и анимациях используются кривые Безье с 4 контрольными точками  $P_0$ ,  $P_1$ ,  $P_2$  и  $P_3$ .

Для того чтобы нарисовать кривую, необходимо нарисовать две воображаемые линии, одна из которых будет иметь координаты  $P_0$  и  $P_1$ , а другая —  $P_1$  и  $P_2$ . Затем крайние точки этих линий начинают непрерывно двигаться к следующим точкам. Третья воображаемая линия рисуется с начальной точкой, непрерывнодвигающейся по первой линии, и конечной точкой,двигающейся по второй линии. На этой воображаемой линии рисуется точка, постепенно движущаяся от начала линии до самого конца. Кривая, которую эта точка будет описывать называется кривая Безье



## 6. Отличие NURBS от кривой Безье

"Сплайн" - это математическая конструкция, позволяющая построить по контрольным точкам гладкую (дифференцируемую  $N$ -ое кол-во раз) кривую, впоследствии расширенная до гладких поверхностей. Изначально придумана для аппроксимации графиков функций. Слово "сплайн" ИМНИП означает гибкую инженерную линейку, которую инженер гнул и прикладывал к чертежу для того чтобы начертить плавную кривую, имеющую требуемые координаты и наклон на концах. Математической аппроксимацией такой линейки является кубический сплайн.

Теперь к ответу на вопрос.

В Blender термин "кривая" означает в точности "сплайн". В Blender имеется два вида сплайнов:

- Безье, отличаются тем, что сплайн проходит через контрольные точки.
- NURBS, отличаются тем, что сплайн следует за контрольными точками, но не проходит через них.

## 7. Из чего состоит 3д модель

Любая 3d модель состоит из:

- Точек
- Рёбер (2 точки образуют ребро, это всегда прямая линия, у каждой точки всегда есть своя вертекс нормаль)
- Плоскостей

#### 8. для чего нужна карта нормалей

- 1) Вертекс нормали влияют на видимость полигона. Если нормаль направлены в сторону пользователя, то он видит полигон.
- 2) Вертекс нормали, помимо блика, влияют ещё на видимость объекта. У полигона есть только одна видимая сторона.
- 3) Вертекс нормали влияют на жёсткость граней

Чтобы воссоздать отражающие свойства высокополигональной модели на низкополигональной модели, как раз и применяются карты нормалей — она запекается в виде текстуры и накладываются на нужный объект. Вся информация о том, как модель должна отражать свет, хранится в текстуре. Это и называется картой нормалей.

#### 9. Разница между уникальным и тайловым маппингом

Тайловый маппинг (от англ. tile — плитка) — это раскрашивание модели при помощи повторяющихся фрагментов рисунка небольшого размера. Например, чтобы наложить на всю стену эффект треснувшей штукатурки, будет достаточно иметь небольшой фрагмент текстуры штукатурки, который будет повторяться на стене. Чаще всего тайловый маппинг применяют при моделировании однородных материалов: экстерьеров зданий, настенной плитки, металлической поверхности, ткани, грунтовых или каменных дорожек и т.д.

Уникальный маппинг — это ручное раскрашивание всех элементов развертки. Например, если перед вами стоит задача наложить текстуры на самолет в стиле стимпанк, то, скорее всего, там будет и ржавый металл, и медь, и дерево, и кожа. Но подгружать все эти материалы на модель, особенно если она будет использоваться в игре, компьютеру будет сложно, он будет тратить много ресурсов. Поэтому для развертки автомобиля используется один тайл, раскрашенный вручную. С точки зрения компьютера это будет всего одна текстура, поэтому загружать ее будет в разы легче, чем точно такую же, но сделанную с помощью нескольких тайлов. Однако уникальный маппинг требует от 3D-моделлера художественных навыков.

#### 10. Для чего нужна развертка

3D-модель сделана из объемных форм. На любой объект в редакторе можно положить материал, выбрать его цвет и настроить блик. Но положить текстуры на 3D-объекты нельзя, пока ты не сделаешь развертку. Программа просто не знает как накладывать плоскую текстуру на геометрию.

Отсюда возникает отдельный этап пайплайна — UV-развертка. Развертка переносит объемные формы на плоскость для того, чтобы на них можно было положить текстуры.

Важно: развертку всегда создают из лоуполи модели, которую мы сделали на прошлом этапе — ретопологии.

#### 11. Что такое OpenGL

OpenGL переводится как Открытая Графическая Библиотека (Open Graphics Library), это означает, что OpenGL - это открытый и мобильный стандарт. Программы, написанные с помощью OpenGL можно переносить практически на любые платформы, получая при этом одинаковый результат, будь это графическая станция или суперкомпьютер. OpenGL освобождает программиста от написания программ для конкретного оборудования. Если устройство поддерживает какую-то функцию, то эта функция выполняется аппаратно, если нет, то библиотека выполняет её программно.

OpenGL - это программный интерфейс для графических устройств, таких как графические ускорители. Он включает в себя около 150 различных команд, с помощью которых программист может определять различные объекты и производить рендеринг. Говоря более простым языком, вы определяете объекты, задаёте их местоположение в трёхмерном пространстве, определяете другие параметры (поворот, масштаб, ...), задаёте свойства объектов (цвет, текстура, материал, ...), положение наблюдателя, а библиотека OpenGL позаботится о том чтобы отобразить всё это на экране. Поэтому можно сказать, что библиотека OpenGL является только воспроизводящей (Rendering), и занимается только отображением 3D объектов, она не работает с устройствами ввода (клавиатуры, мыши). Также она не поддерживает менеджер окон.

## 12. Для чего нужен фрагментальный (или фрагментный) шейдер

Фрагментный шейдер (fragment shader) обрабатывает каждую видимую часть конечного изображения.

Подробнее:

Шейдеры - простыми словами это инструкции для наших видео-карт, которые говорят, как правильно отрисовывать и трансформировать объекты в игре.

Вершинные шейдеры, которые принимают на вход атрибуты вершин некоторых примитивов, и вычисляют новые значения тех же самых или других атрибутов вершин. Вершинный шейдер работает над каждой вершиной независимо от других вершин и от примитива, к которому он относится

Фрагментные шейдеры, которые принимают на вход данные, которые когда-то были возвращены вершинным шейдером, а потом прошли дополнительные преобразования (интерполяцию между вершинами примитива по всему примитиву). Задача фрагментного шейдера — вычислить цвет фрагмента примитива (обычно треугольника) на основе ряда величин, характерных именно для этого фрагмента; каждый фрагмент обрабатывается фрагментным шейдером независимо от других фрагментов того же примитива.