

БРС по предмету виртуальная реальность

Общие положения

- $БРС = (ТЗ + баллы) * задания$.
- max баллов за предмет – 100.
- За одно задание max 40 баллов.
- Задание сдано, если набраны 40% баллов за него.
- На доклады етц – не более 30% баллов (т. н. необъективируемые контрольные мероприятия).
- Нет хоть какого-то задания – сразу двойка (не принял участие в контрольном мероприятии).
- Не менее трёх контрольных мероприятий за семестр (это могут быть сдачи лаб).
- Нужен их график (номера пар, когда будет проходить сдача).
- Сдавать задания требуется В СРОК!!! Иначе – 0 баллов. Если не сдали, есть время досдать на консультации при условии наличия уважительной причины.
- Шкала перевода в оценки на стр. 8 в положении.

Задания

Требуется выполнить задания из приведённого ниже списка. Задания выполняются с использованием WebGL.

Проверку каждого задания проводит преподаватель, тестируя наличие и стабильность работы (отсутствие падений и зависаний) заявленной в требовании функциональности. Кроме того, преподаватель смотрит исходный код. За преподавателем остаётся право задавать дополнительные вопросы касательно использованных технологий и приёмов.

Кроме того, в конце семестра каждый студент должен устно ответить на один теоретический вопрос.

Задания

1. Процедурное текстурирование (пара №5, max = 20, min = 8).

Написать программу с использованием WebGL, выводящую на экран квадрат, закрашенный волнами, аналогично первому продемонстрированному на занятии изображению. За основу взять предоставленный пример кода. Затем модифицировать код фрагментного шейдера так, чтобы вместо волн выводились кольца, аналогично второму предоставленному изображению. Продемонстрировать следует оба варианта. Подсказка: Если необходимо, следует соответствующим образом изменить содержимое вершинного буфера. Основные изменения, однако, должны затронуть фрагментный шейдер. В принципе, используя коэффициенты, можно обойтись передаваемыми в шейдерную программу координатами вершин.

2. Альфа-смешивание (пара №8, max = 20, min = 8).

Написать программу с использованием WebGL, выводящую на экран систему из 27 кубов, выстроенных в виде большого куба. Система должна вращаться вокруг своего центра. Каждый куб должен иметь свой цвет и быть полупрозрачным (цвета могут быть заданы случайными, прозрачность лучше сделать у всех одинаковой, например, равной 0.5). Должен быть включен буфер глубины, включено отсечение задних граней, и при этом на изображении должны отсутствовать артефакты альфа-

смешивания.

3. Индивидуальное задание (пара №11, max = 40, min = 16).

Написать программу с использованием WebGL, выводящую на экран освещённую трёхмерную модель с наложенной на неё текстурой. Модель и текстура могут быть загружены из произвольного источника (файл или массив данных). Для данного задания предусмотрено несколько вариантов сложности с разным максимальным баллом. По каждому пункту баллы суммируются.

1. Модель освещения (max = 10).

1. Использование модели освещения на основе закона Ламберта (только диффузный свет) – 5 баллов.
2. Использование модели освещения Фонга (диффузный свет и зеркальный блик) – 6 баллов.
3. Использование более сложной модели освещения – 10 баллов.

2. Текстура (max = 10).

1. Наложение одной текстуры – 5 баллов.
2. Мультитекстурирование – 10 баллов.

3. Интерактивность (max = 5).

4. Стереоскопическая визуализация (max = 15).

1. Анаглиф с демонстрацией – 15 баллов.
2. Side-by-side без демонстрации – 10 баллов.
3. Side-by-side с демонстрацией (использование шлема виртуальной реальности) – 15 баллов.

Доклады

По желанию студенты могут взять доклады по темам, перечисленным ниже. Доклад оценивается в 20 баллов и заменяет собой сдачу теории. Во время докладов необходимо продемонстрировать примеры кода (не обязательно написанные докладчиком самостоятельно, но обязательно им осознанные).

1. Моделирование и текстурирование сфер.
2. Модели освещения.
3. Бамп-мэппинг.

Задача о биллборде

По желанию студенты могут решить теоретико-практическую задачу о биллборде. Первый решивший (либо группа максимум из трёх человек) получает 20 баллов, что заменяет собой сдачу теории. Решение должно состоять из теоретического обоснования и практического подтверждения (должна быть написана демонстрационная программа, использующая полученное решение и доказывающая его правильность).

Формулировка задачи: имеется объект $Obj1$, для которого известна матрица вида-модели $modelView_object$, и только она. Эта матрица является произвольной комбинацией аффинных преобразований. Необходимо найти матрицу вида-модели $modelView_billboard$ для объекта $Obj2$, который вывелся бы как биллборд, «приклеенный» к объекту $Obj1$. То есть нужно найти такое преобразование F , что $F(modelView_object) = modelView_billboard$.

Демонстрационная программа должна использовать все аффинные преобразования, в частности повороты вокруг *всех* осей, одновременно. Например, можно создать сцену, в которой масштабированный куб вращается вокруг некоторого центра относительно вектора

(1;1;1) (как планета вокруг звезды), при этом также вращается вокруг своего геометрического центра относительно вектора (1;1;1), а билборд всё это время «описан» около него.

Теоретический опрос

Каждый студент должен ответить на один вопрос из списка (**пара №11; max = 20; min = 8**).

1. Мультимедиа. Её составляющие. Классификация по степени интерактивности.
2. OpenGL. Его характеристика и особенности.
3. Графический конвейер OpenGL.
4. Машина состояний OpenGL.
5. Использование матриц в OpenGL. Почему все матрицы имеют размерность 4*4?
6. Анимация. Подходы к управлению скоростью анимации.
7. Механизм двойной буферизации.
8. Роль проекции в построении изображений на основе трёхмерной сцены.
9. Буфер глубины.
10. Буфер цвета.
11. Буфер трафарета.
12. Камера на трёхмерной сцене.
13. Трёхмерные объекты. Семантический разрыв между представлением объектов и визуализацией.
14. Освещение. Модель освещения. Виды моделей освещения.
15. Способы закраски полигонов.
16. Нормаль к вершине. Назначение и способы вычисления.
17. Текстура. Текстурные координаты. Карта текстуры.
18. Мипмэппинг.
19. Трёхмерные модели. Способы их хранения.
20. Анимация трёхмерных моделей. Скелетная анимация. Прямая и инверсная кинематика.
21. Кубические сплайны. Построение поверхностей и тел вращения на основе кривых.
22. Кривые Безье. Построение поверхностей и тел вращения на основе кривых.
23. Что представляет собой двумерная графика в OpenGL?
24. Спрайты.
25. Фракталы. Фрактальная размерность.
26. Система итерирующих функций для построения фрактала.
27. Система Линдермайера для построения фрактала.
28. Построение множества Мандельброта.
29. Стереοизображение. Технологии построения и демонстрации.
30. Насколько правильно говорить «фильм в 3D» или «3D-кинотеатр»?
31. Шейдеры. Назначение и виды шейдеров.
32. Фотонная карта.

33. Алгоритм трассировки луча.
34. В чём различие между «затенением» и «тенью»?
35. Основные методы моделирования теней (на уровне идей).
36. Отражение. Основные методы моделирования отражений (на уровне идей).
37. Что такое проблема двух зеркал? Как её решить?
38. Прозрачность и преломление. Основные методы моделирования прозрачности (на уровне идей)
39. Прозрачность и преломление. Основные методы моделирования преломления (на уровне идей).
40. Каустика. Основные методы моделирования каустики (на уровне идей).
41. Оптимизация процесса рендеринга. Алгоритм Z-буфера. Алгоритм художника.
42. Почему алгоритм художника не применим, если используется альфа-смешивание?
43. Оптимизация процесса рендеринга. Отсечение «задних» граней.
44. Оптимизация процесса рендеринга. Отсечение по видимости.
45. Оптимизация процесса рендеринга. Отсечение заслонённых объектов. Отличие данного подхода от алгоритма Z-буфера.
46. Оптимизация процесса рендеринга. Портальный рендеринг.
47. Оптимизация процесса рендеринга. Уровни детализации.
48. Фильтрация изображений.
49. Матрица конволюции.
50. OpenAL. Его характеристика и особенности.
51. Основные понятия OpenAL.
52. Звук и музыка в мультимедийных системах.