

## Как будет проходить экзамен

Специальная программа генерирует 5 случайных номеров вопросов; по желанию любой из выпавших вопросов вы можете заменить на другой (тоже случайный). Каждый вопрос оценивается в 10 баллов (частичный ответ может быть оценён в меньшее количество баллов). При подготовке (после получения вопросов) можно использовать любые материалы; время подготовки не ограничено (в разумных пределах).

## Список вопросов

### Графические API

1. **Графические API.** Какие бывают; чем отличаются; GPGPU API.
2. **История OpenGL API.** В чём основные отличия OpenGL 3.0+ от OpenGL 1.0; какие есть API, основанные на OpenGL.
3. **Технические детали OpenGL API.** Контекст OpenGL; что содержит реализация OpenGL; загрузка OpenGL-функций; расширения OpenGL; ошибки OpenGL.

### Математика

4. **Аффинное пространство.** Что такое; зачем нужно.
5. **Аффинные комбинации, выпуклые комбинации, линейная интерполяция.** Что такое; какие свойства.
6. **Аффинные преобразования.** Что такое; что ими можно описать; формула композиции; формула обратного.
7. **Однородные координаты.** Как представляются точки и векторы; как представляются аффинные преобразования; матрицы сдвига, поворота, масштабирования.
8. **Ортографическая проекция.** Когда используется; какие особенности; как описывается; как строится матрица.
9. **Перспективная проекция.** Когда используется; какие особенности; как описывается; как строится матрица.
10. **Перевод из экранных координат в мировые.**
11. **\*Вычисление видимой области.**
12. **Easing functions, сплайны.** Зачем нужны; примеры; виды сплайнов.
13. **Кватернионы.** Что такое; какие свойства; таблица умножения; формула обратного; связь с вращениями.

## Объекты OpenGL

14. **Объекты OpenGL.** Что такое; общие операции работы с объектами; объект с нулевым ID.
15. **Буферы.** Что такое; зачем нужны; какие есть значения target; как загрузить данные; mapped buffer.
16. **Vertex array.** Что такое; зачем нужны; какие данные хранят.
17. **Текстуры.** Что такое; зачем нужны; какие есть значения target; виды текстур.
18. **Текстуры 2.** Как загрузить данные; форматы пикселей; фильтрация; wrapping; mipmaps.
19. **Фреймбуферы.** Что такое; зачем нужны; какие есть значения target; как настраиваются; framebuffer completeness; attachments.
20. **Renderbuffer'ы.** Что такое; зачем нужны; какие есть значения target.
21. **Шейдеры.** Что такое; зачем нужны; как скомпилировать.
22. **Шейдерные программы.** Что такое; зачем нужны; как слинковать.
23. **Texture units.** Что такое; как привязать текстуру к шейдеру.

## Язык шейдеров

24. **Язык GLSL.** Синтаксис (в общих чертах); управляющие конструкции; типы данных (назвать несколько); встроенные функции (назвать несколько); принципиальные отличия от C, C++ и т.п.
25. **Uniform-переменные.** Что такое; зачем нужны; как задавать значение.
26. **Samplers.** Что такое; виды; какие есть функции для чтения текстуры из шейдера; что означают текстурные координаты для разных видов sampler'ов.

## Рендеринг в OpenGL

27. **Графический конвейер.** Все этапы обработки данных в конвейере.
28. **Атрибуты вершин.** Какие есть форматы хранения; как указывается расположение данных атрибута; как описываются в вершинном шейдере.
29. **Индексация вершин.** Зачем нужно; где хранятся индексы; как вызывать индексированный рендеринг; primitive restart.

- 30. **Primitive assembly.** Что такое; какие есть примитивы; как они превращаются в линии/треугольники.
- 31. **Вершинный шейдер.** Зачем нужен; что принимает на вход; что выдаёт на выходе.
- 32. **Геометрический шейдер.** Зачем нужен; что принимает на вход; что выдаёт на выходе.
- 33. **Back-face culling.** Как работает; зачем нужно; как настраивается.
- 34. **Clipping.** Как работает; по каким уравнениям происходит отсечение.
- 35. **Perspective divide.** Зачем нужен; как работает.
- 36. **Растеризация.** Что такое; правила растеризации OpenGL; viewport.
- 37. **Фрагментный шейдер.** Зачем нужен; что принимает на вход; что выдаёт на выходе.
- 38. **Тест глубины.** Зачем нужен; как настраивается; Z-буфер и его формат; Z-fighting; Z-clamping.
- 39. **Stencil-тест.** Зачем нужен; как настраивается; stencil буфер и его формат.
- 40. **Blending.** Зачем нужен; как настраивается.

## Освещение

- 41. **Освещение.** Зачем нужно; что происходит со светом; виды материалов; BRDF, BTDF, BSDF; уравнение рендеринга.
- 42. **Источники света.** Точечные источники; направленные источники; как описывается освещённость в точке.
- 43. **Модель Фонга.** Ambient освещение; диффузное освещение; specular освещение.
- 44. **Ambient occlusion.** Что такое; зачем нужно; baked ambient occlusion.
- 45. **Normal mapping.** Что такое; зачем нужно; как реализуется; система координат нормалей; формат хранения нормалей.
- 46. **Material mapping.** Что такое; зачем нужно; как реализуется.
- 47. **Отражения.** Плоское зеркало; environment mapping; reflection mapping.
- 48. **Тени.** Что такое; мягкие и жёсткие тени; umbra, penumbra, их размер.
- 49. **Shadow volumes.** Принцип работы; плюсы и минусы; детали реализации.

- 50. **Shadow mapping.** Принцип работы; плюсы и минусы; детали реализации; shadow acne, shadow bias; peter panning.
- 51. **Shadow mapping: проекции.** Как строятся; отличия между точечным и направленным источниками.
- 52. **Shadow mapping: вариации.** PCF; ESM; VSM; PSM; CSM.

## Эффекты

- 53. **Размытие.** Что такое; как реализуется; separable kernels; bloom; depth blur.
- 54. **Toon shading.** Что такое; как реализуется; edge detection; color grading.
- 55. **Сглаживание.** Что такое; зачем нужно; supersampling; multisampling; FXAA.
- 56. **Tone mapping.** Что такое; зачем нужен; пример tone mapping оператора; как реализуется.
- 57. **Gamma-correction.** Что такое; зачем нужно; как реализуется; sRGB.
- 58. **Color banding, dithering.** Что такое; когда возникает; как реализуется; где применяется; варианты dither mask.

## Прочее

- 59. **Billboards.** Что такое; зачем нужны; как реализуются.
- 60. **Покадровая анимация (моделей и изображений).** Что такое; как реализуются.
- 61. **Скелетная анимация.** Чем лучше покадровой; как реализуется.
- 62. **Рендеринг текста.** Кодировки; шейпинг; bitmap-шрифты; векторные шрифты; SDF-шрифты.
- 63. **Volume rendering.** Что такое; зачем нужен; основные понятия и уравнения; виды рассеяния.
- 64. **Volume rendering: методы реализации.** Slicing; raymarching.

## Оптимизация

- 65. **\*Оптимизация.** Общие идеи; метод поиска bottleneck'а.
- 66. **\*Instanced rendering.** Что такое; зачем нужно; как работает.
- 67. **\*Frustum culling.** Что такое; зачем нужно; SAT.
- 68. **\*Occlusion culling.** Что такое; зачем нужно; детали реализации.
- 69. **\*LOD.** Что такое; зачем нужно; детали реализации.