

Идеи для итоговых проектов

Сложность ★★★★★

1. Procedural City with LOD System for Mobile GPUs

Цель: Генерация бесконечного процедурного города с системой уровней детализации (LOD), оптимизированной для памяти и пропускной способности мобильного устройства.

Технологии: OpenGL ES 3.2 или WebGL, Geometry Shaders (или инстансинг + генерация на CPU в потоке).

Ключевая фишка: Автоматическое объединение статических мешей (Mesh batching) и переключение LOD без заметных артефактов.

2. Система динамической смены времени суток с атмосферным рассеянием (Precomputed Atmospheric Scattering)

Цель: Реализация реалистичного неба, объемных лучей (god rays) и освещения ландшафта в зависимости от положения солнца.

Технологии: WebGL (для демо в портфолио) или OpenGL.

Фишка: Использование LUT таблиц для предрассчитанных параметров рассеяния Рэля и Ми, чтобы работать в реальном времени на мобильном GPU без пересчета на каждый кадр.

3. Инструмент для анимированного Material Editor на базе графов шейдеров

Цель: Визуальное приложение, где пользователь соединяет ноды (текстуры, маск, noise, blend) и получает на выходе код шейдера.

Технологии: OpenGL + Node Editor.

Поддержка мобильности: Экспорт готового материала под OpenGL ES.

Результат: Показать, как можно автоматизировать рутинную работу художников по созданию материалов.

4. Портал-рендеринг (для неевклидовой геометрии) для игр жанра Antichamber / Portal

Цель: Реализация корректного рендеринга бесконечных пространств и порталов (вид из портала в портал).

Технологии: OpenGL (Stencil Buffer, рекурсивный рендеринг в текстуру).

Сложность: Управление глубиной и отсечением (clipping) в неевклидовых сценах.

Применение к мобильным играм — создание иллюзии большого открытого мира на ограниченном железе.

Дополнительные требования к любому проекту:

1. **Сравнительный бенчмарк:** Замеры FPS, потребления памяти батареи на реальном мобильном устройстве (или эмуляторе).
2. **Кроссплатформенная архитектура:**
3. **Документация:** Объяснение алгоритмов. Защита проекта
4. **Видео-демонстрация** работы.

Проекты уровня: ★★ или ★★★ или ★★★★

1. Интерактивная «Книга теней» (Shadow Mapping с 3 типами фильтров)

Что делает: Сцена с несколькими движущимися источниками света.

Техника: Рендеринг в текстуру глубины + сравнение с PCF (фильтрация краев теней).

Нужно реализовать каскадные тени для directional light.

Оценка: За 1 тип теней — 4, за 3 типа (PCF, Variance, Exponential) — 5.

Сложность: ★★★

2. Атмосферная обсерватория (Fog + Skybox + Day/Night Cycle)

Что делает: Вид на горы с небом, туманом и циклической сменой дня и ночи.

Техника: Процедурное небо (градиент + солнце как спрайт), экспоненциальный туман по высоте, смешивание текстур земли.

Для мобилок: Показать энергоэффективность — no real-time lights, только шейдеры.

Сложность: ★★★

4. Игра-платформер с пером Pixar (Contour Rendering)

Что делает: 3D платформер, где персонажи обведены черным контуром.

Техника: Рендеринг объекта в 2 прохода: 1) раздутый силуэт (back-face culling inverted), 2) нормальная модель.

Платформа: WebGL (чтобы поиграть в браузере).

Продвинуто: Добавить точечные блики (cel shading) + контур.

(классическая техника из 90х, но выглядит как современная).

Сложность: ★★★

5. Мини-карта в реальном времени (RTT для интерфейса)

Что делает: Гоночный трек или стратегия. В углу экрана — вид сверху на машину/юнита через камеру, которая рендерит сцену в текстуру.

Техника: Создать вторую камеру, её FBO прицепить к текстуре UI.

Фишка: Реализовать **Picture-in-Picture** — портал в другую комнату (задняя текстура рендерит вид из другой точки сцены).

Сложность: ★★★

6. Система декалей (Crime scene: пулевые отверстия, пятна крови)

Что делает: При клике мышкой на любую поверхность (стена, пол) наклеивается текстура (дырка, грязь).

Техника: Алгоритм поиска точки столкновения луча с треугольником (ray-AABB), затем рендеринг квадрата в текстуру (или использование воксельной маски).

Для WebGL: Демо в браузере + запись позиций декалей в массив.

Сложность: ★★★★ (из-за необходимости пересчитывать батчи или использовать шейдер с массивом позиций).

7. Танцующие частицы с эффектом Trail (без Compute Shaders!)

Что делает: Рой частиц (2000 шт), которые летают по синусоидам, оставляя за собой шлейф.

Техника: Трансформация вершин на CPU (массив позиций обновляется каждый кадр) — да, не оптимально, но просто. Для шлейфа — инстансинг или геометрический шейдер.

Платформа: Обязательно с замером FPS (показать, что CPU версия жрет батарею).

Сравнение: Реализовать ту же сцену на GPU через transform feedback (это уже продвинуто).

Сложность: ★★.

9. Инструмент для создания «Зелья варенья» (Billboarding + Alpha blending)

Что делает: 3D сцена, где вместо деревьев — спрайты, всегда повернутые к камере.

Добавить траву (cross planes) и прозрачные листья.

Техника: Billboarding в шейдере (модификация моделирующей матрицы).

Продвинутость: Реализовать **цилиндрический билборд** (поворот только по Y) для имитации деревьев в RPG. Добавить альфа-тест (отсечение прозрачных пикселей).

Сложность: ★★ (классика, которую любят на собеседованиях).

10. Web-портфолио «Шейдерный калейдоскоп» (5 интерактивных шейдеров)

Что делает: Одностраничный сайт (Three.js), где на 3D объектах можно переключать шейдеры:

- Волны (vertex shader).
- Проверка расстояния (glow при приближении мыши).
- Мозаика из фрагментов шейдера.
- Дисторсия (эффект жара над асфальтом).
- Сдвиг цветов RGB.

Защита: Студент объясняет, как оптимизировать uniform-переменные под мобильные GPU (минимум дискретизаций).

Сложность: ★★ .

Усложняющие опции к каждому проекту (НА ОТЛИЧНО):

- №1 + Реализовать каскадные тени для большого открытого мира.
- №2 + Добавить объемный свет (god rays) через пост-эффект.
- №8 + Сделать морфинг между двумя эффектами в реальном времени.

1. Система теней с каскадными картами (CSM) для открытого мира

Что делает: Большая сцена (поле, лес, холмы) с динамическим солнцем. Тени не сыпятся на дальних дистанциях.

Техника: 3–4 каскада теней (split by distance), визуализация каскадов разными цветами для отладки.

Почему ок: Расширение классического Shadow Mapping. Добавляется только цикл рендера в несколько FBO.

Сложность: ★★★

2. Эффект «глубины резкости» (DOF) как в кинематографе

Что делает: Камера фокусируется на объекте (пикап по клику), остальное размыто (боке).

Техника: Два прохода рендера в текстуру → размытие по Гауссу с маской глубины →

смешивание.

Платформа: Чистый OpenGL (можно показать на любой десктопной сцене).

Сложность: ★★

4. Динамическое разрушение простых объектов (поломанный забор/ящик)

Что делает: При ударе или выстреле объект разваливается на 5–10 кусков, которые падают с простой физикой (гравитация + вращение).

Техника: Предварительно разбить модель на куски (в Blender или руками). При столкновении — удалить исходный объект, добавить куски с присвоенными скоростями.

Физика: Простейшая (Euler integration без коллизий между кусками).

Сложность: ★★ (нет сложной графики, но результат эффектный)

6. Волны на воде (геометрический шейдер + текстура нормалей)

Что делает: Озеро или океан с анимированными волнами, отражением неба и Френелем.

Техника: Сетка вершин → геометрический шейдер сдвигает Y по синусу (или текстура смещения). Отражение: кубическая карта или рендер в текстуру зеркала.

Фишка: Добавить эффект «пенка» на гребнях волн по высоте.

Сложность: ★★★

7. Система лодов (LOD) для леса из 1000+ деревьев

Что делает: Густой лес. Близкие деревья — высокополигональные, дальние — билборды (2D кресты), очень дальние — исчезают.

Техника: Инстансинг (glDrawElementsInstanced). Расчет расстояния от камеры в вершинном шейдере или на CPU (батчинг по дистанции).

Для десктопа: Можно не париться с memory budget — просто показать, как работает LOD.

Сложность: ★★

10. Десктопная версия игры Pong с продвинутой графикой

Что делает: Классический Pong, но:

- Мяч — сфера с окружением (cubemap reflections).
- Ракетки — 3D модели с эффектом motion blur при движении.
- Пол, стены, фон — текстуры с нормальями и динамическим освещением.
- Счет — рендер 3D текста (символы как меши).

Технологии: OpenGL + GLFW.

Сложность: ★★

Как усилить проекты для сильных студентов (НА ОТЛИЧНО):

- №1 → Добавить soft shadows (PCF фильтрацию с 16 сэмплами).
- №4 → Написать разбиение меша на куски прямо в рантайме (алгоритм Voronoi).

- **№6** → Реализовать физику лодки, плавающей по этим волнам (высота из текстуры нормалей).
- **№10** → Сделать сетевую игру через сокеты (2 игрока на разных компах) — по желанию.