

Анализ изменений в проекте Task 8 (DirectX11App_task_8.cpp) по сравнению с Task 7

✦ 1. Обновлённая таблица функций и методов (по лекциям 2–11)

№	Функция/Метод	Назначение и принцип работы	Ссылка на лекцию
1	WinMain	Точка входа в приложение. Инициализирует окно, DirectX, ресурсы и запускает главный цикл.	Лекция 2, стр. 2–4
2	InitWindow	Создаёт и регистрирует окно с заданными размерами и стилем.	Лекция 2, стр. 5–6
3	InitDirectX	Инициализирует DirectX: устройство, контекст, своп-цепь, растеризатор, буфер глубины D32_FLOAT, состояния глубины и blend states.	Лекция 2, стр. 7–15; Лекция 8, стр. 5–20
4	SetupBackBuffer	Создаёт render target view для back buffer и буфер глубины. Теперь также создаёт текстуру для постпроцессинга и соответствующие view.	Лекция 2, стр. 28–29; Лекция 8, стр. 5–6; Лекция 11, стр. 3–4
5	InitCube	Создаёт геометрию куба, шейдеры, input layout. Вершины с нормальными и касательными.	Лекция 3, стр. 5–12; Лекция 6, стр. 11–13; Лекция 9, стр. 13, 25
6	InitBuffers	Создаёт константные буферы для инстансинга и сцены. Добавлен буфер для постпроцессинга.	Лекция 5, стр. 6–8; Лекция 9, стр. 18–20; Лекция 10, стр. 4, 31–33
7	LoadTextureArray	Загружает массив текстур (Brick.dds, Kitty.dds) для инстансинга.	Лекция 10, стр. 12–15
8	LoadNormalMap	Загружает карту нормалей (BrickNM.dds).	Лекция 9, стр. 21–27
9	InitSkybox	Создаёт сферу для skybox, загружает cubemap, компилирует шейдеры с reversed depth.	Лекция 6, стр. 34–42; Лекция 8, стр. 31–32
10	InitTransparentObjects	Инициализирует прозрачные прямоугольники.	Лекция 8, стр. 15–30
11	InitSmallSpheres	Инициализирует маленькие сферы для визуализации источников света.	Лекция 9, стр. 11
12	InitPostProcess	Инициализирует систему постпроцессинга: создаёт текстуру для рендера, RTV, SRV,	Лекция 11, стр. 3–10

№	Функция/Метод	Назначение и принцип работы	Ссылка на лекцию
		шейдеры и константный буфер для эффектов.	
13	WndProc	Обрабатывает сообщения окна: изменение размера, вращение камеры, колесо мыши.	Лекция 2, стр. 2–4
14	ResizeSwapChain	Изменяет размер back buffer, буфера глубины и текстуры постпроцессинга при изменении окна.	Лекция 2, стр. 28
15	UpdateCamera	Обновляет матрицу вида и проекции (reversed depth), записывает в буфер сцены.	Лекция 5, стр. 21–24; Лекция 8, стр. 12–14; Лекция 9, стр. 12
16	UpdatePostProcessBuffer	Обновляет константный буфер постпроцессинга с выбранным типом эффекта.	Лекция 11, стр. 9–10
17	Render	Основной цикл рендеринга. Теперь включает рендеринг сцены в текстуру и применение постпроцессинга.	Лекция 3, стр. 52–54; Лекция 6, стр. 28; Лекция 8, стр. 33; Лекция 9, стр. 11–12; Лекция 10, стр. 3, 34
18	RenderTransparentObjects	Рендерит прозрачные объекты с сортировкой от дальнего к ближнему.	Лекция 8, стр. 25–30
19	RenderSmallSpheres	Рендерит маленькие сферы, представляющие источники света.	Лекция 9, стр. 11
20	RenderPostProcess	Применяет постпроцессинг: рендерит полноэкранный треугольник с шейдером, который обрабатывает текстуру сцены.	Лекция 11, стр. 5–10
21	CullBoxes	Выполняет frustum culling для инстансов.	Лекция 10, стр. 23–30
22	Cleanup	Освобождает все созданные DirectX-ресурсы, включая ресурсы постпроцессинга.	—

2. Сравнение кодов: что нового добавилось в task_8 (по сравнению с task_7)

Компонент	task_7 (старый)	task_8 (новый)	Что изменилось
Постпроцессинг	Не было	Реализована система постпроцессинга с тремя эффектами: сепия, холодный тон, ночное видение	Добавлена текстура для рендера (<code>m_pColorBuffer</code>), RTV, SRV, шейдеры, константный буфер
Рендеринг в текстуру	Сцена рендерилась сразу в back buffer	Сцена рендерится в промежуточную текстуру, затем применяется постпроцессинг и выводится в back buffer	Изменён пайплайн рендеринга: сначала рендер в <code>m_pColorBufferRTV</code> , затем постобработка
Шейдеры постпроцессинга	Не было	Добавлены вершинный и пиксельный шейдеры для постпроцессинга. Вершинный шейдер использует один треугольник вместо квадрата.	Оптимизация: уменьшен <code>overdraw</code> , упрощена геометрия
Управление эффектами	Не было	Добавлено окно ImGui для выбора эффекта (сепия, холодный тон, ночное видение)	Интерактивное переключение эффектов в реальном времени
Текстура для рендера	Не было	Создаётся текстура с флагами <code>D3D11_BIND_RENDER_TARGET</code> и <code>D3D11_BIND_SHADER_RESOURCE</code>	Текстура пересоздаётся при изменении размера окна
Константный буфер эффектов	Не было	Добавлен <code>m_pPostProcessBuffer</code> для передачи типа эффекта в шейдер	Позволяет динамически менять эффект без перекомпиляции шейдеров
Рендеринг полноэкранного треугольника	Не было	Используется треугольник вместо квадрата для покрытия всего экрана	Уменьшение количества вершин и устранение <code>overdraw</code> на границах

3. Детальный конспект (task_8)

Концепция	Реализация в коде	Как работает	Теория
Render Target для постпроцессинга	<code>m_pColorBuffer,</code> <code>m_pColorBufferRTV,</code> <code>m_pColorBufferSRV</code>	Создаётся текстура с двумя флагами: как render target и как шейдерный ресурс. В неё рендерится вся сцена, затем она используется как входная текстура для постпроцессинга.	Лекция 11, стр. 3–4
Вершинный шейдер постпроцессинга	<code>m_pPostProcessVertexShader</code>	Использует один треугольник вместо квадрата. Координаты вершин заданы в clip space. UV вычисляются из позиции.	Лекция 11, стр. 6–8
Пиксельный шейдер постпроцессинга	<code>m_pPostProcessPixelShader</code>	Содержит три функции эффектов: <code>ApplySepia</code> , <code>ApplyColdTint</code> , <code>ApplyNightVision</code> . Выбор эффекта через <code>effectType</code> из константного буфера.	Лекция 11, стр. 9–10
Константный буфер эффектов	<code>m_pPostProcessBuffer</code>	Структура <code>PostProcessBuffer</code> содержит <code>effectType</code> . Обновляется через <code>UpdatePostProcessBuffer()</code> .	Лекция 11, стр. 9–10
Рендеринг в текстуру	В <code>Render()</code> сначала <code>OMSetRenderTargets</code> на <code>m_pColorBufferRTV</code>	Вся сцена рендерится в текстуру, а не сразу в back buffer. Затем вызывается <code>RenderPostProcess()</code> .	Лекция 11, стр. 3–5
Применение постпроцессинга	<code>RenderPostProcess()</code>	Устанавливает back buffer как render target, накладывает текстуру сцены через полноэкранный треугольник с выбранным эффектом.	Лекция 11, стр. 5–10
Оптимизация: треугольник вместо квадрата	В вершинном шейдере постпроцессинга	Три вершины вместо шести, что уменьшает overdraw и упрощает геометрию.	Лекция 11, стр. 12
Интерактивное управление эффектами	Окно ImGui "Post Processing"	Позволяет выбирать эффект из списка: None, Sepia, Cold Tint, Night Vision.	Лекция 11, стр. 14 (задание)

☐ **Итог: Task 8 добавляет систему постпроцессинга, которая включает:**

1. **Рендеринг в текстуру** – создание промежуточного render target для сцены.
2. **Мульти-эффектный шейдер** – поддержка сепии, холодного тона и ночного видения.

3. **Оптимизированный полноэкранный рендеринг** – использование треугольника вместо квадрата.
4. **Интерактивное управление** – выбор эффекта через ImGui в реальном времени.
5. **Динамическую настройку** – возможность легко добавлять новые эффекты в шейдер.

Основное достижение: Реализован профессиональный pipeline постпроцессинга, позволяющий применять различные фильтры к итоговому изображению с минимальными затратами производительности.