

Секция 3 «Автоматизация исследований»

Председатели:

Воруев Андрей Валерьевич, канд. техн. наук, доцент.

Бычков Павел Валерьевич, канд. физ.-мат. наук, доцент.

М. О. Адинцов, Е. В. Рафалова
(ГГУ имени Ф. Скорины, Гомель)

ОДНОПОТОЧНАЯ МОДЕЛЬ OpenGL КАК ОГРАНИЧИВАЮЩИЙ ФАКТОР ПРОИЗВОДИТЕЛЬНОСТИ

OpenGL (Open Graphics Library) является устаревшим графическим API, разработанным в 1990-х годах. Несмотря на его важную роль в развитии компьютерной графики, архитектура OpenGL не соответствует современным требованиям к производительности и функциональности.

Основной недостаток OpenGL заключается в том, что она ориентирована на однопоточное выполнение операций [1], что ограничивает возможности использования многопоточной обработки, которое становится стандартом для современных графических библиотек, таких как Vulkan или DirectX 12.

В качестве примера указанного недостатка можно рассмотреть игру Minecraft версии 1.12.2 с установленной модификацией HBM's NTM Extended Edition. На скриншоте, представленном на рисунке 1, отображён F3-экран отладки, который демонстрирует производительность игры и её взаимодействие с OpenGL.



Рисунок 1 – Отладочная информация Minecraft 1.12.2 с модификацией
HBM's Nuclear Tech Mod Extended Edition

На изображении заметно, что игра работает с низкой частотой кадров (24/13 FPS). Это связано с однопоточной архитектурой OpenGL, где все графические операции выполняются в основном потоке приложения. При высокой загрузке, вызванной сложными модификациями, такими как HBM's NTM, однопоточная обработка становится узким местом в производительности.

Кроме того, отладочная информация показывает отсутствие обновления чанков (0 chunk updates), что свидетельствует о значительных ограничениях при обработке данных. Система, использующая OpenGL, не может эффективно распределять задачи между потоками, что приводит к снижению общей производительности игры в данной конфигурации.

Этот пример хорошо иллюстрирует недостатки OpenGL, связанные с невозможностью эффективного использования многопоточности в современных графических

приложениях, где требуется обработка больших объёмов данных. Переход к более современным API, таким как Vulkan, может решить эту проблему, предоставив возможность распределения графических задач между несколькими потоками.

Литература

1. **Microsoft Learn.** Multithread OpenGL Drawing Strategies [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/en-us/windows/win32/opengl/multithread-opengl-drawing-strategies>. – Дата доступа: 22.03.2025.