

ПРОЦЕДУРНАЯ СИМУЛЯЦИЯ ДВИЖЕНИЯ ТРАНСПОРТА В ИГРАХ С ПОМОЩЬЮ ВЫЧИСЛИТЕЛЬНОГО ШЕЙДЕРА

М.Р. Юрьев (Академические классы Гимназии № 642)

Научный руководитель – Р.Н. Юрьев (аспирант Университета ИТМО)

Аннотация. В работе представлен метод использования ресурсов GPU для симуляции движения транспорта в игровых движках. Разработан пример системы трафика на примере реализации в движке Unity. Проведена оценка полученного решения в сравнении с многопоточным и однопоточным решениями для CPU.

Введение. Симуляция движения транспорта является одной из наиболее сложных задач при разработке игр. Для некоторых жанров, таких как градостроительные симуляторы, очень важно, чтобы при большом количестве транспорта симуляция оставалась стабильной и не тормозила игру. В последнее время вычислительные мощности графических процессоров растут. В больших симуляциях транспорт может управляться при помощи одинаковых инструкций (по принципам SIMD), поэтому стоит задуматься об использовании вычислительных возможностей GPU.

Основная часть. Чтобы производить вычисления на видеокарте вне графического пайплайна можно использовать вычислительный шейдер. В этот шейдер нужно передать необходимые данные в формате больших буферов. Нужно разделять неизменные и изменяемые буферы во избежание копирования излишних данных. Для оптимизации следует распределить потоки для различных вычислений в зависимости от их индекса, что сократит время выполнения полноценной инструкции для всего трафика, но может увеличить объем требуемой видеопамяти (в зависимости от роста количества буферов, используемых в шейдере). Разработанная реализация направляет часть потоков на выполнение инструкций для машин, а часть для обработки состояний светофоров.

Так как информация о положении машин необходима для получения на стороне CPU, после выполнения шейдера нужно вернуть значения различных элементов буфера в компоненты движка. Если движок изначально спроектирован в SIMD и ECS парадигмах, то копирование и применение данных будет проще и эффективнее (в нашем случае ECS и SIMD на стороне CPU не используются и это является одним из наиболее неэффективных компонентов системы).

Выводы. Проведены тесты на устройстве с параметрами: Laptop, процессор AMD Ryzen 55600H, RAM 32 Гб, Nvidia RTX 3050Ti. Результаты показывают, что метод действительно позволяет добиться лучшей производительности при симуляции движения транспорта (при количестве в 8000 машин разница между симуляцией на CPU и GPU составляет до 1,7 мс). При небольшом количестве машин, менее приблизительно 2048 (1 стандартный буфер для вычислительного шейдера DirectX) разница в производительности между решениями будет незначительна, при этом решение для CPU будет оптимальнее, так как не занимает дополнительной видеопамяти.

Список использованных источников:

- [1] Jin, Z. 3D Traffic Simulation for Autonomous Vehicles in Unity and Python / Z. Jin, T. Swedish, R. Raskar // ArXiv. – 2018. – arXiv:1810.12552. – URL: <https://arxiv.org/abs/1810.12552> (дата обращения: 16.02.2025).
- [2] Cuyos, K. Computer simulation model for traffic enforcement using unity engine / K. Cuyos, D. Ubanan, A. Ceniza // IOP Conference Series: Materials Science and Engineering. – 2019. – Vol. 482. – P. 012018. – DOI: 10.1088/1757-899X/482/1/012018.
- [3] Unity [Электронный ресурс]. – URL: <https://unity.com> (дата обращения: 16.02.2025).
- [4] Writing Portable Rendering Code with NVRHI [Электронный ресурс]. – URL: <https://developer.nvidia.com/blog/writing-portable-rendering-code-with-nvrhi/> (дата обращения: 16.02.2025)