

ОБЗОР И СРАВНЕНИЕ СУЩЕСТВУЮЩИХ МЕТОДОВ ОПТИМИЗАЦИИ ПРОИЗВОДИТЕЛЬНОСТИ ПРИ ОТРИСОВКЕ ОБЪЕКТОВ НА ДАЛЬНИХ РАССТОЯНИЯХ В UNITY 3D

В.В. Фирстов

аспирант, e-mail: ckontro@gmail.com

Омский государственный университет имени Ф.М. Достоевского, Омск, Россия

Аннотация. В статье представлен обзор современных методов оптимизации производительности при отрисовке объектов на дальних расстояниях в Unity 3D. Рассмотрены техники, такие как уровни детализации (LOD), окклюзионное отсекаание, использование билбордов, объединение объектов и GPU Instancing. Проведён анализ их эффективности на основании тестов с использованием сцены с открытым миром. Работа подчёркивает важность комплексного подхода к оптимизации для достижения высоких показателей производительности.

Ключевые слова: оптимизация производительности, разработка игр, Unity 3D.

Введение

Оптимизация производительности является ключевым аспектом разработки игр, особенно при создании больших и динамичных игровых миров. Особенности прорисовки объектов на дальних расстояниях могут значительно влиять на потребление ресурсов игры и конечного пользователя. Настоящая статья представляет обзор и анализ таких методов оптимизации, как LOD, окклюзионное отсекаание (встроенные инструменты Unity3D), билборды, объединение объектов и современные техники GPU Instancing.

1. Цель исследования

Основной целью настоящего исследования является выявление наиболее эффективных подходов для оптимизации сцен с открытым миром. Задача работы – изучить существующие методы оптимизации, произвести их тестирование в одинаковых условиях, сравнить полученные результаты.

1.1. Методы исследования

Анализировались следующие методы оптимизации:

- Уровни детализации (LOD) – автоматическое переключение между высоко- и низкополигональными моделями в зависимости от расстояния до камеры.

- Окклюзионное отсекаание (Occlusion Culling) – исключение из рендеринга объектов, закрытых другими элементами сцены.
- Билборды (Billboards) – замена сложных моделей двухмерными изображениями на дальних расстояниях.
- Объединение объектов (Static Batching) – уменьшение количества вызовов отрисовки путём группировки статичных объектов.
- GPU Instancing – рендеринг множества экземпляров одной модели с минимальными затратами ресурсов.

1.2. Спецификации оборудования, на котором проводилось тестирование

- ОС: Windows 11 Домашняя, 23H2.
- Процессор: AMD Ryzen 5 7500F.
- Видеокарта: NVIDIA GeForce RTX 4070 SUPER.
- Оперативная память: 32,0 ГБ, 6000 МГц.
- Unity 3D 2021.3.45f1.

2. Словарь терминов

- **Draw Calls** – последовательность команд, которые сообщают GPU, как следует обработать рендер.
- **Batches** – количество отрисовок, сделанных видеокартой в текущем цикле.
- **Tris** – количество отрисовываемых треугольников в мешах.
- **Verts** – количество отрисовываемых вершин в мешах.
- **Avg FPS** – среднее количество кадров в секунду.

3. Ход работы

3.1. LOD (Levels of Detail)

LOD (Levels of Detail) – это метод оптимизации, при котором используются разные версии модели с различной степенью детализации в зависимости от расстояния до камеры [1]. Чем дальше объект от игрока, тем более упрощённая модель используется, что снижает количество полигонов, которые необходимо обработать (рис. 1).

Основные преимущества LOD:

- Существенное снижение нагрузки на видеокарту.
- Для каждой модели можно задать свои уникальные настройки.

Однако LOD также обладает и недостатками:

- Для каждой модели необходимо создать несколько уровней детализации.
- Увеличение занимаемой памяти как в игре, так и в билде.

Для тестирования на сцену были помещены 180 объектов (рис. 2) с настроенным LOD. Проведём тест и посмотрим влияние данного метода оптимизации на производительность.

Результаты без LOD:

- Batches: 112074.

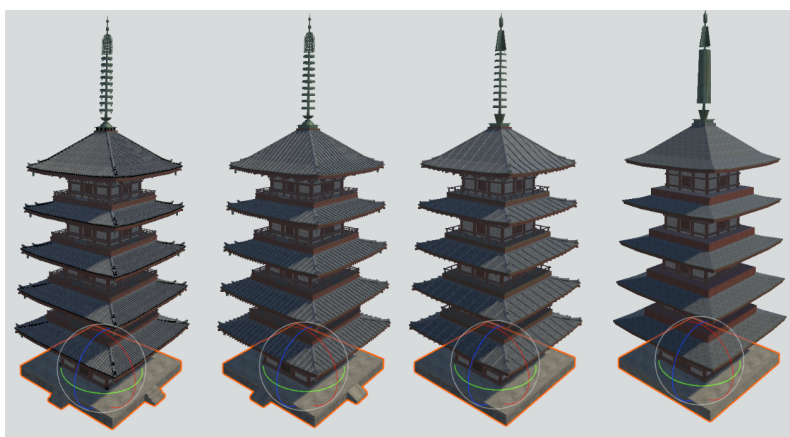


Рис. 1. Пример разных уровней LOD, применённых к одной модели

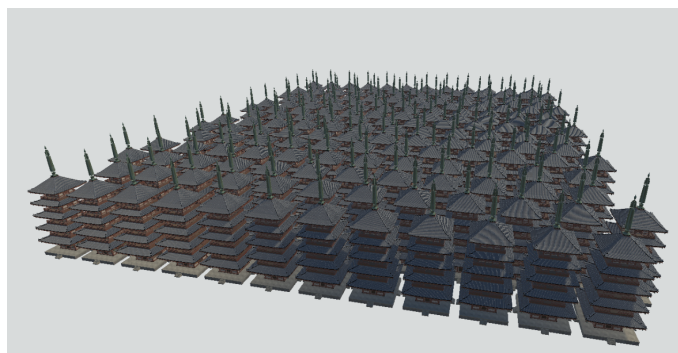


Рис. 2. Сцена, в которой проводятся тесты

- Tris: 178.2M.
- Verts: 492.2M.
- Avg FPS: 15–20.

Результаты с LOD на близком расстоянии:

- Batches: 31158.
- Tris: 31.4M.
- Verts: 84.5M.
- Avg FPS: 130.

Результаты с LOD на дальнем расстоянии:

- Batches: 13681.
- Tris: 2.1M.
- Verts: 5.2M.
- Avg FPS: 260–270.

Вывод: использование LOD позволяет уменьшить нагрузку на видеокарту и сильно увеличить производительность в зависимости от количества объектов на сцене и их расстояния от камеры. Метод отлично подходит для высокополигональных моделей.

3.2. Occlusion Culling

Occlusion Culling — это метод оптимизации, при котором движок не отрисовывает объекты, полностью закрытые другими объектами (рис. 3). В Unity эта функция встроена в инструменты рендеринга и позволяет исключить из обработки объекты, которые невидимы игроку, тем самым снижая нагрузку на графический процессор (GPU) и повышая производительность [2].

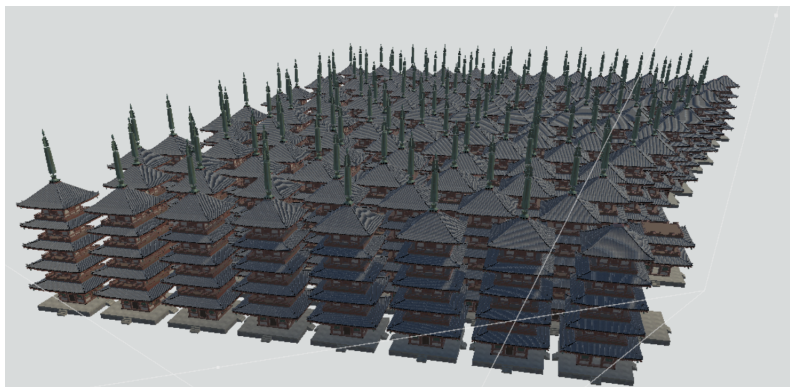


Рис. 3. Визуализация отсекаания объектов вне поля зрения камеры

Этот метод особенно полезен в сценах с большим количеством объектов, где многие из них могут быть скрыты стенами, другими моделями или элементами окружения, например, в городской застройке или в закрытых помещениях.

Основные преимущества Occlusion Culling:

- Снижение количества отрисовываемых объектов.
- Лёгкая настройка.

Недостатки Occlusion Culling:

- Неэффективность на открытых пространствах.
- Запекание может занять длительное время.
- Сложно использовать с динамическими объектами.

Для тестирования объекты были помечены как статические, и включено отсечение.

Результаты без Occlusion Culling:

- Batches: 112074.
- Tris: 178.2M.
- Verts: 492.2M.
- Avg FPS: 15–20.

Результаты с Occlusion Culling:

- Batches: 29411.
- Tris: 107.6M.
- Verts: 292.5M.
- Avg FPS: 35–40.

3.3. Billboards

Billboards – метод оптимизации рендеринга, при котором плоские изображения (спрайты) отображаются вместо сложных трёхмерных объектов (рис. 4). Эти изображения автоматически поворачиваются лицевой стороной к камере, создавая иллюзию объёма [3]. Биллборды используются для растительности, эффектов частиц и декоративных элементов на дальних расстояниях.

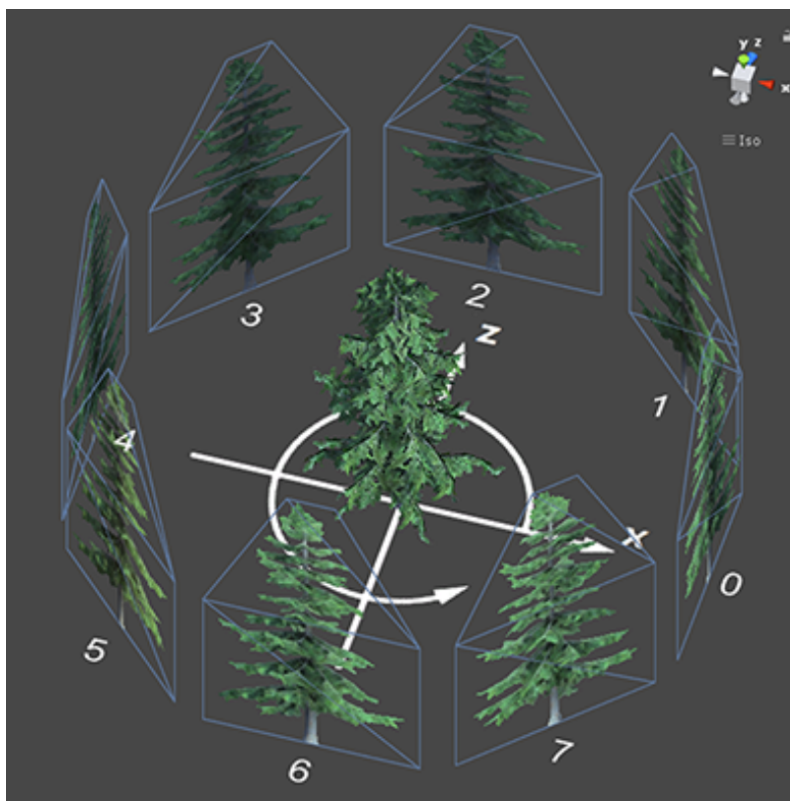


Рис. 4. Визуализация работы Billboards

Основные преимущества Billboards:

- Увеличение производительности: заменяют сложные модели на простые текстуры, уменьшая нагрузку на графический процессор (GPU).
- Экономия ресурсов: позволяют сократить количество полигонов и шейдерных вычислений.
- Универсальность: подходят для объектов, которые не требуют высокой детализации на расстоянии.

Недостатки Billboards:

- Ограниченная реалистичность: при близком приближении выглядит плоско и теряет качество.
- Отсутствие объёмных теней: биллборды не создают корректные тени как полноценные 3D-модели.
- Проблемы с перспективой: могут выглядеть неестественно при изменении углов обзора.

Результаты без Billboards:

- Batches: 112074.
- Tris: 178.2M.
- Verts: 492.2M.
- Avg FPS: 15–20.

Результаты с Billboards:

- Batches: 213.
- Tris: 4.1T.
- Verts: 8.3T.
- Avg FPS: 350–370.

3.4. Прочие методы

Существуют подходы, которые не сравнятся с другими инструментами, но также могут улучшить производительность.

3.4.1. Объединение объектов (Static/Dynamic Batching)

Объединение объектов (Static/Dynamic Batching) – слияние нескольких объектов в один меш для уменьшения количества вызовов рендеринга (Draw Calls).

Преимущества:

- Быстрая и лёгкая настройка, считается базовым методом оптимизации работы видеокарты.

Недостатки:

- Дополнительная нагрузка на CPU.
- Разные подходы для разных типов объектов (статичные/динамичные).

Данный способ на тестируемой сцене сократил количество Batches примерно в 2 раза, облегчив работу видеокарты, но при этом, из-за повышенной нагрузки на CPU, FPS также упал, а не увеличился.

3.4.2. GPU Instancing

GPU Instancing, её техника, позволяющая отрисовывать множество одинаковых объектов за один вызов рендеринга [4].

Преимущества:

- Эффективно для большого количества идентичных объектов.
- Меньшая нагрузка на CPU и RAM по сравнению с Static Batching.

Недостатки:

- Применимо не ко всем материалам.
- Меньшая эффективность при малом количестве объектов.

Данный способ снизил количество Batches более чем на 90 %, тем самым разгружая видеокарту, но из-за того, что нагрузка не перераспределялась на CPU, FPS также незначительно вырос.

3.4.3. Сравнение результатов

Сравнение результатов проводилось путём измерения показателей FPS и Batching на заранее заготовленной сцене, состоящей из 180 высокополигональных объектов. Для обеспечения сопоставимости результатов каждый метод оптимизации тестировался изолированно.

Тестирование началось с базового замера производительности без каких-либо оптимизаций, которое показало средний показатель в 17 FPS и 112,074 Batches. Затем последовательно проверялись различные методы: Level of Detail с тремя уровнями детализации для всех 3D-моделей, Occlusion Culling с автоматической разметкой зон видимости, замена зданий на билборды на дальних расстояниях, статический батчинг и GPU Instancing для материалов. Результаты по каждому из основных способов были получены через стандартные средства диагностики Unity 3D. График результатов тестирования представлен на рис. 5.

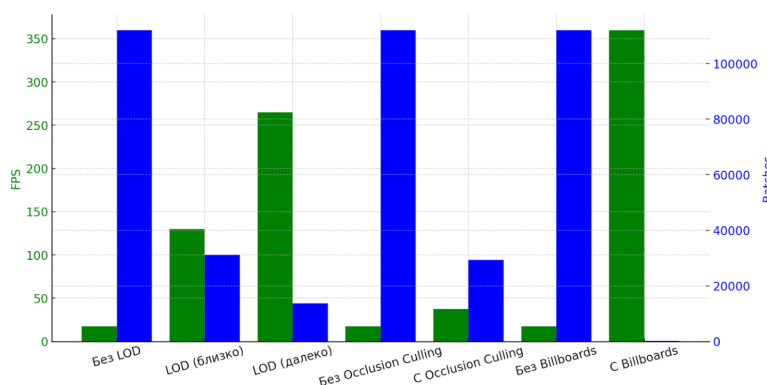


Рис. 5. Результаты работы различных методов оптимизации

Заключение

В данной статье был проведён обзор и сравнение основных методов оптимизации производительности при отрисовке объектов на дальних расстояниях в Unity 3D. Рассмотренные техники, такие как LOD, Occlusion Culling, Billboards, Static/Dynamic Batching и GPU Instancing, продемонстрировали свою эффективность в различных сценариях, особенно в контексте сцен с открытым миром.

LOD показал себя как высокопроизводительный инструмент для снижения нагрузки на GPU за счёт использования упрощённых моделей на дальних расстояниях, что значительно повысило средний FPS. Однако его настройка требует дополнительных временных затрат и увеличивает объём данных проекта.

Occlusion Culling эффективно снижает количество отрисовываемых объектов, особенно в сценах с большим количеством перекрывающихся элементов, но его применение ограничено в открытых пространствах и с динамическими объектами.

Billboards продемонстрировали высокую производительность, заменяя сложные 3D-модели на простые текстуры, что особенно полезно для растительности и деко-

ративных элементов. Однако их использование ограничено из-за потери реалистичности при близком рассмотрении.

Для проектов с открытым миром рекомендуется приоритизировать LOD и Billboards. В сложных сценах с множеством перекрытий (например, здания) Occlusion Culling становится обязательным. Оптимальная стратегия – комбинирование методов в зависимости от доминирующего типа объектов. При этом важно учитывать, что скорость и сложность внедрения различных методов оптимизации существенно различаются. Настройка LOD для ключевых объектов хоть и требует значительных временных затрат (в том числе на создание оптимизированных копий моделей), окупается стабильным приростом производительности. Billboards, напротив, настраиваются относительно быстро и идеально подходят для массовой растительности, но их реализация ограничивается определёнными типами объектов. GPU Instancing требует технической подготовки материалов, однако после правильной настройки практически не нуждается в дополнительном внимании. Наиболее трудоёмким остаётся Occlusion Culling – разметка зон видимости в сложных сценах может отнять много времени, но именно этот метод становится незаменимым для городских локаций с плотной застройкой. Static Batching хоть и прост в реализации, часто оказывается недостаточным для комплексной оптимизации.

Таким образом, при выборе методов следует учитывать не только их эффективность, но и соотношение трудозатрат к потенциальному выигрышу в производительности, отдавая предпочтение тем решениям, которые дают максимальный результат при минимальных усилиях в конкретных условиях проекта.

Литература

1. Дикинсон, К. Оптимизация игр на Unity 5: Практические советы и методы для повышения производительности приложений Unity3D. Бирмингем: Packt Publishing, 2015. 161 с.
2. How to Use Occlusion Culling in Unity - The Sneaky Way // TheGamedev.Guru. URL: <https://thegamedev.guru/unity-performance/occlusion-culling-tutorial/> (дата обращения: 15.01.2025).
3. Billboards in Unity (and How to Make Your Own) // GameDev Beginner. URL: <https://gamedevbeginner.com/billboards-in-unity-and-how-to-make-your-own/> (дата обращения: 16.01.2025).
4. Miroslav Martinovič. Unity3D Adventures in Creating Huge Outdoor Areas & Terrains. Part 4: The Real Adventure of GPU // Medium. URL: <https://midnight-shcode.medium.com/unity3d-adventures-in-creating-huge-outdoor-areas-terrains-part-4-the-real-adventure-of-gpu-cc265a423d53> (дата обращения: 19.01.2025).

REVIEW AND COMPARISON OF EXISTING METHODS FOR OPTIMIZING PERFORMANCE IN RENDERING DISTANT OBJECTS IN UNITY 3D

V.V. Firstov

Ph.D. Student, e-mail: ckontro@gmail.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. The article provides an overview of modern methods for optimizing performance when rendering objects at long distances in Unity 3D. Techniques such as levels of detail (LOD), occlusion culling, billboard usage, object combining, and GPU Instancing are examined. Their effectiveness is analyzed based on tests conducted using an open-world scene. The study highlights the importance of a comprehensive optimization approach to achieve high-performance metrics.

Keywords: performance optimization, game development, Unity 3D.

Дата поступления в редакцию: 14.05.2025