

Оптимизация построения и хранения lightmap

Гурьев Василий

группа 344

руководитель А. А. Пименов

26 апреля 2019 г.

Введение

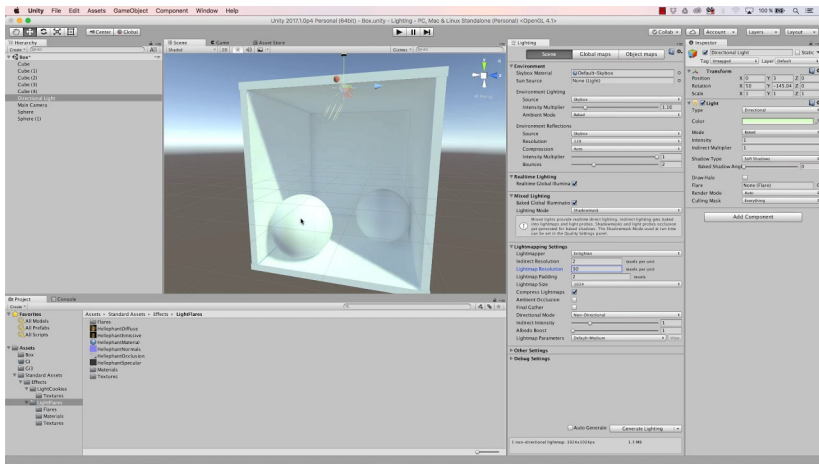


Рис.: Пример создания карты освещенности, Unity

Введение

- ▶ На данный момент применяется во всех крупных 3D проектах и 3D движках
- ▶ По реалистичности превосходит динамическое освещение, при котором освещенность и тень рассчитываются перед каждым кадром
- ▶ Позволяет значительно экономить ресурсы компьютера
- ▶ Работает только со статическими объектами и источниками света

Цель работы

- ▶ Исследовать существующее решение lightmap в игровом проекте World War Z
- ▶ На основе данного исследования предложить несколько оптимизаций по построению и хранению карты освещенности
- ▶ Реализовать данные оптимизации до завершения проекта

Пример работы алгоритма

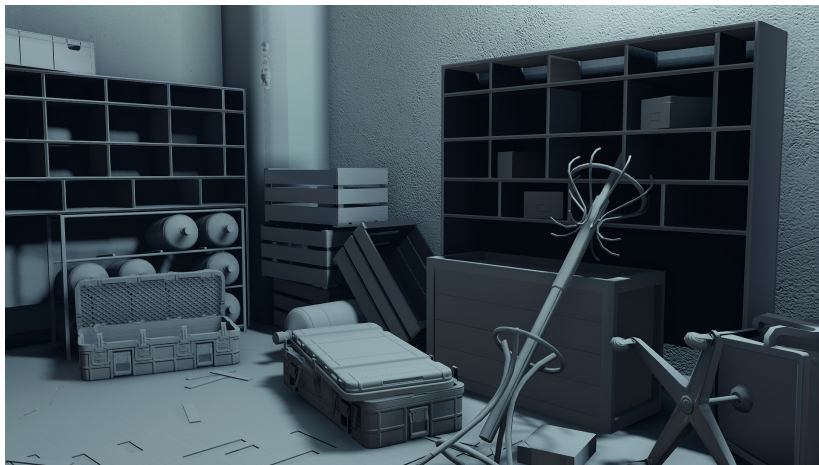


Рис.: Сцена с картой освещенности, World War Z

Алгоритм

- ▶ Изначально все текселы карты освещенности заполняются черным цветом
- ▶ Каждой точке на полигоне 3D объекта сопоставляется тексел карты освещенности
- ▶ Считается освещенность в этой точке от всех видимых источников света, учитываются отражения, преломления и т.д.
- ▶ Результат записывается внутрь карты освещенности
- ▶ При рендере объекта считывается и учитывается освещенность из lightmap

Алгоритм в World War Z

- ▶ Пять текстур вместо одной
- ▶ Для разных настроек качества - разные карты освещенности
- ▶ В вершинах геометрии сохранены одни uv-координаты для всех текстур
- ▶ Формат хранения текстур - BC6/BC7

Алгоритм в World War Z

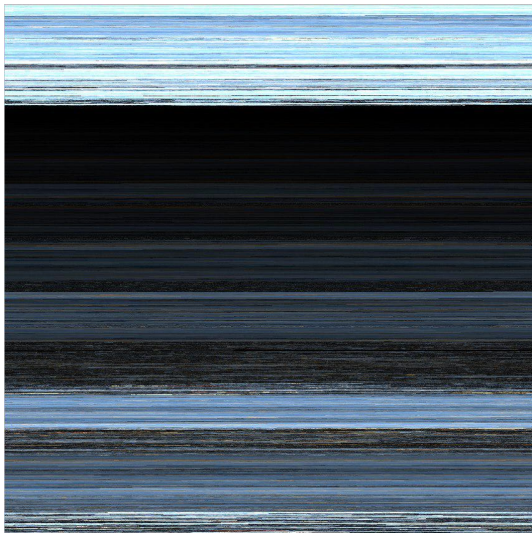


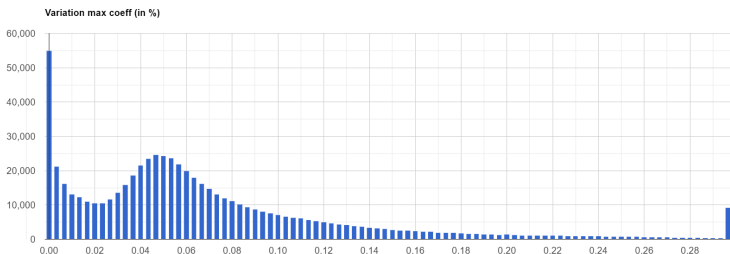
Рис.: lightmap одной из карт, World War Z

Оптимизации

- ▶ Разные uv-координаты для разных настроек качества
- ▶ Найти "похожие" точки в текстурах и объединить их в одну, сделав одни и те же uv координаты на вершинах геометрии

Оптимизации

- ▶ Была реализована программа, показывающая, насколько "похожие" данные хранятся в карте освещенности



▶ **Рис.:** Результат работы программы, максимальное относительное отклонение точек от среднего значения внутри кластера, приведенное к виду от 0 до 1

Оптимизации

Алгоритм нахождения "похожих" точек был реализован следующим образом

- ▶ K-means применяется для начальной кластеризации данных
- ▶ В каждом кластере параллельно ищутся точки из одной ϵ -окрестности, после чего они объединяются в одну точку

Результат оптимизации

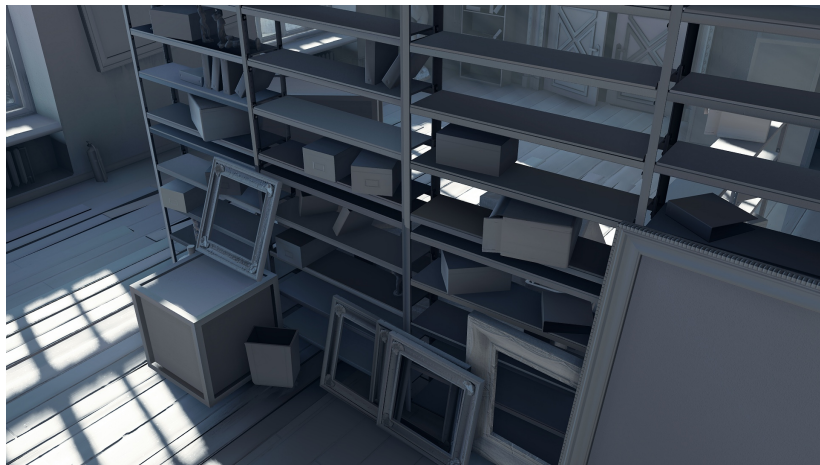


Рис.: lightmap одной из карт, World War Z

Результат оптимизации

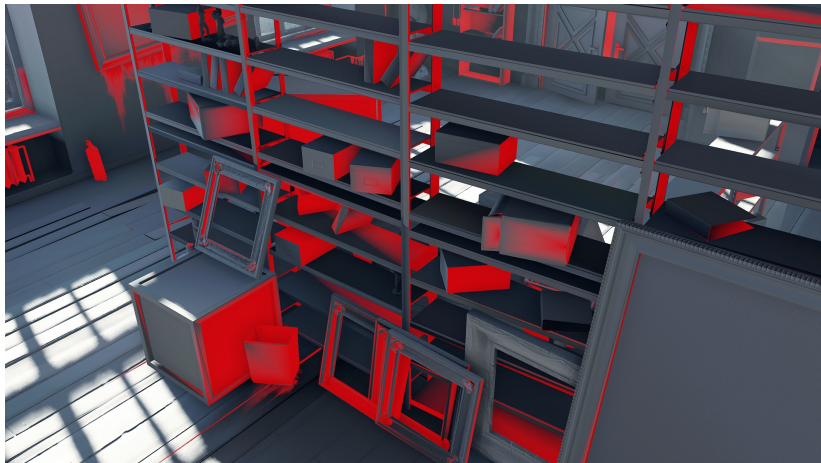


Рис.: lightmap одной из карт с визуализированной оптимизацией, World War Z

Результат оптимизации

- ▶ Удалось сократить количество точек на 38.5%
- ▶ Общее время работы алгоритма для 13 миллионов 30-мерных точек составляет в среднем 8 минут на уровень

Итоги

- ▶ Исследовано существующее решение lightmap в игровом проекте World War Z
- ▶ На основе данного исследования было предложено несколько оптимизаций по построению и хранению карты освещенности
- ▶ Одна из оптимизаций была реализована, существенно сократив размер проекта (порядка 300МБ) без видимой потери качества