

Ни одну из современных "Next-Gen" игр нельзя представить без красивых теней. Существует немало способов создания мягких теней, самым эффективным среди которых является сглаживание границы теней через пост-обработку (Post Effect). Положительные способы этого метода: хорошая производительность и работа с большим числом источников света, все объекты сцены отбрасывают и принимают тени.

Что же такое пост-обработка или пост-эффект (High Dynamic Range – HDR). Изображения в играх хранятся и обрабатываются в представлении, имеющем ограниченную точность, это даёт небольшое количество градаций яркости. HDR позволяет использовать для хранения цвета большее число градаций (96-бит с плавающей точкой).

Для реализации данной методики выбрали движок Ogre3D [www.ogre3d.org].Ogre3D – это объектно-ориентированный графический движок рендеринга. Для работы с ним необходимо хорошее знание C++, так как структура движка построена на объектах. Следует заметить, что Ogre3d -- только графический движок. Это значит, что ничего другого кроме вывода и обработки графики он не делает. Но это значит еще и то, что к нему можно прикрутить сторонние библиотеки для создания полноценной игры. Поддержка Direct3D и OpenGL, поддержка Windows, Linux и Mac OSX, поддержка компиляторов Visual C++ 6 (необходим STLport), Visual C++.NET 2002 (необходим STLport), Visual C++.NET 2003 и Visual C++.NET 2005, Visual C++.NET 2008 в среде Windows. Поддержка компиляторов gcc v.3 и выше в операционных системах Linux / Mac OSX (необходим XCode).

При использовании DSM (Depth Shadow Map) сталкиваемся с проблемой размытия теней. Проблема заключается в том, что мы не можем размыть теневую текстуру до её использования, так как она представляет собой текстуру, в каждом пикселе которой мы храним расстояние, а не цвет как в обычных текстурах. То есть две рядом стоящие точки могут иметь абсолютно разные значения, и их смешивание приведёт только к получению артефактов.

Вывод – теневую карту нельзя размыывать до её использования, а если и размыть, то результат не изменится. Значит, размытие нужно делать после применения. Тут мы и возвращаемся к пост-обработке.

Одним из самых простых примеров является – B&W (Black and White effect). Цветное изображение становится черно-белым.

Как это обычно делается в играх: из положения камеры, согласно ее направлению взгляда, создаётся RTT (Render To Texture), образно говоря, делается снимок того, что мы видим на экране. Затем эта текстура передается в шейдер:

далее мы читаем нашу текстуру float3 image и работаем уже с image.RGB, то есть делается цветокоррекция в [0; 1]. На выходе получаем первоначальное изображение только в черно-белых красках;

создаём quad (квадрат размерами с наш экран) и на него накладываем полученную в пункте 2) текстуру.

Мы же предлагаем использовать следующую методику для создания мягких теней:

рисуются сцена без теней – RTT1 (Render To Texture), то есть на все игровые объекты накладываются текстуры, бамп, параллакс и т.д. Но тени не прорисовываем;

далее рисуем только тени – RTT2. Полная противоположность пункту 1). Делаем все объекты полностью белыми и прорисовываем тени. Получаем черно-белую картинку;

картинку из RTT2 необходимо размыть, так как границы теней зубчатые. Устранить это можно увеличением размера текстуры, что не очень хорошо, или использованием размытия. Размываем RTT2 по вертикали и горизонтали и получаем RTT3;

смешиваем $RTT4 = RTT1 * RTT3$. То есть, получаем float3 image1 и float3 image2. После этого берется текстура без теней и умножается на теневые коэффициенты. Получаем тени с гладкими границами;

выводим RTT4 – получаем мягкие тени.

При реализации данного метода создания мягких теней сталкиваемся с проблемой -- это пункт 3). Так как картинка размывается всегда постоянным коэффициентом (при обычном блюре-размытии), то по мере удаления камеры от тени, тень начинает исчезать (мерцать). Поэтому размывать RTT2 нужно с учетом глубины. Для чего создаётся еще одна дополнительная текстура RTTDepth и смешивание идёт с учетом коэффициентов данной текстуры.

Как результат этого зубчатые границы сглаживаются, картинка получается более реалистичной, тени мягкими.