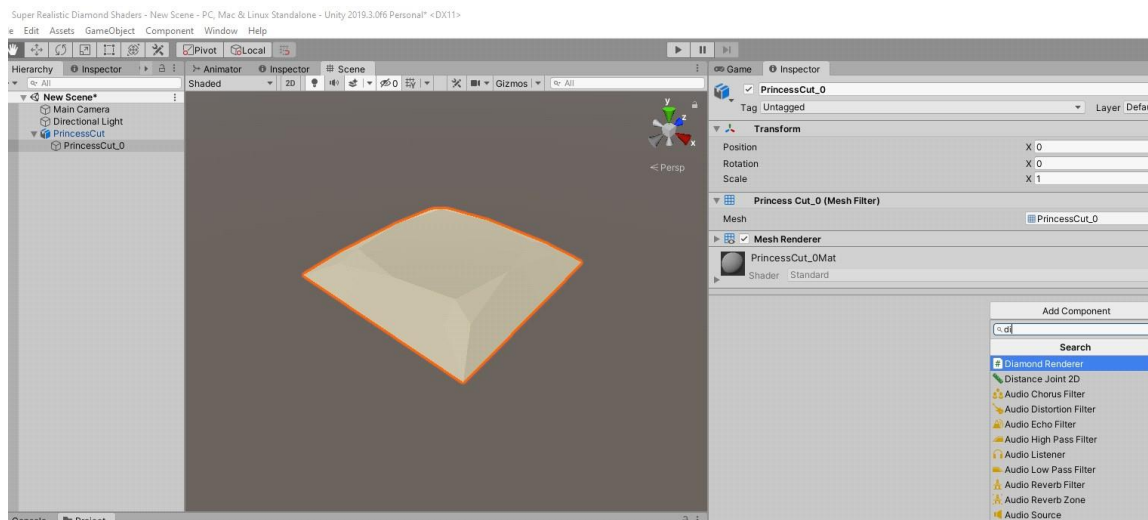
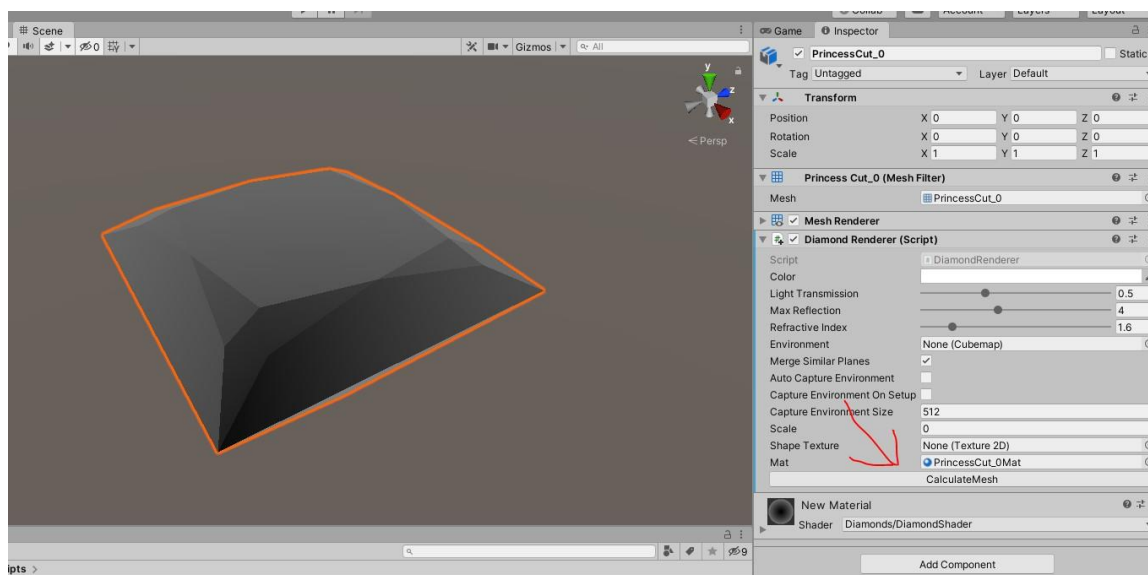


Поместите модель в сцену, добавьте скрипт Diamond Render на модель, под компонентом Mesh Renderer.

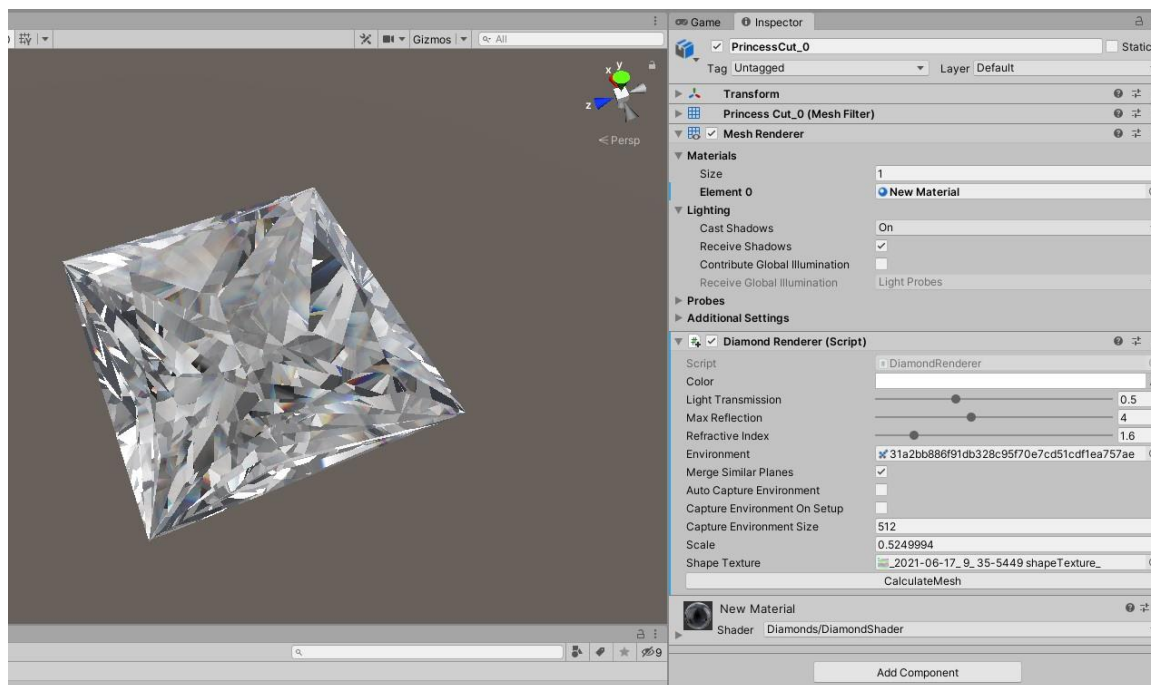
Не забудьте о поставке галочки Read\Write в настройках модели, это обязательно для того чтобы скрипт смог читать модель, так же убедитесь что нормали у модели у вас правильные, не сглаженные. Так же у вашей модели должна быть правильная огранка и правильная топология чтобы рефракция была корректной.



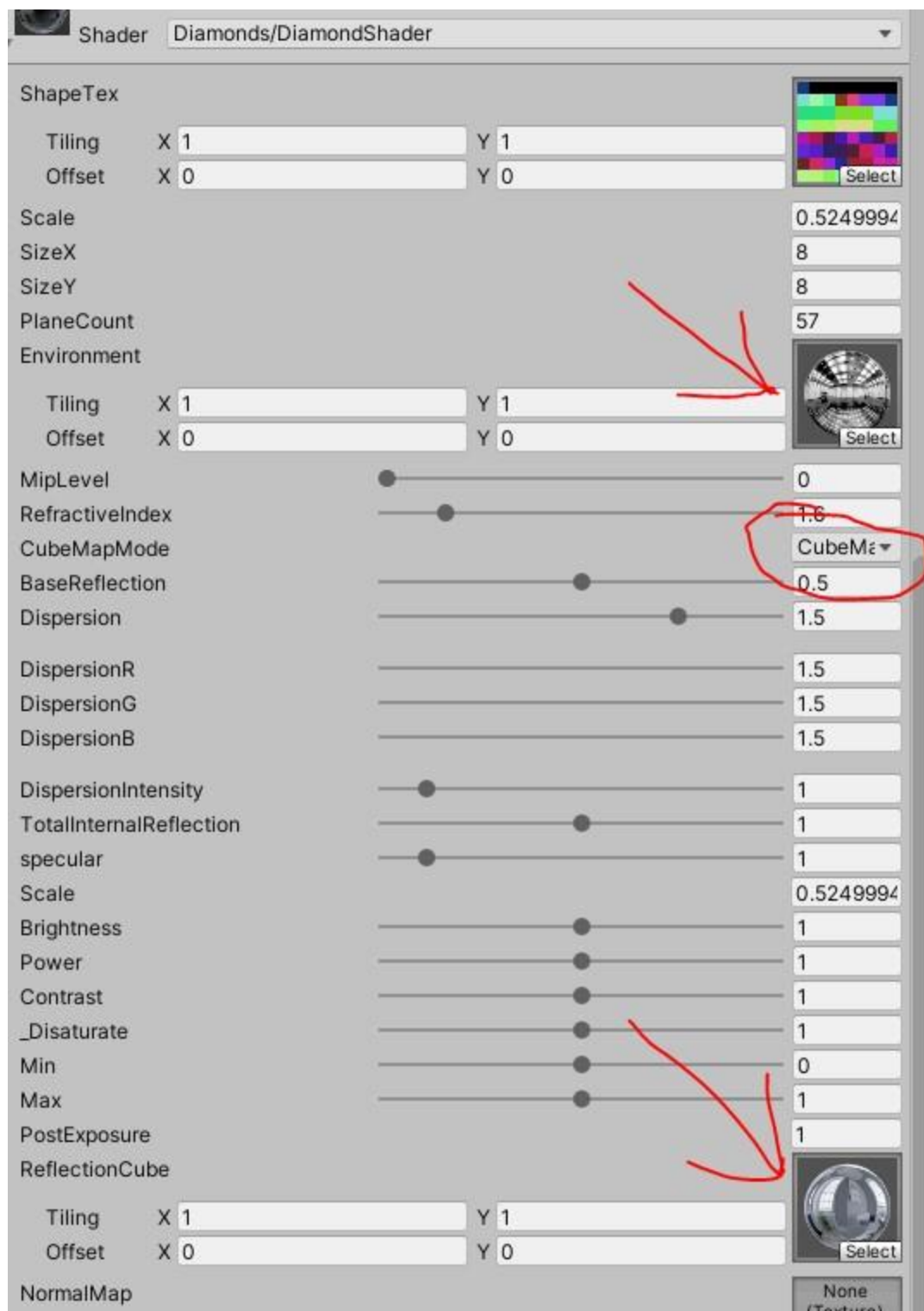
Установите материал с шейдером DiamondShader



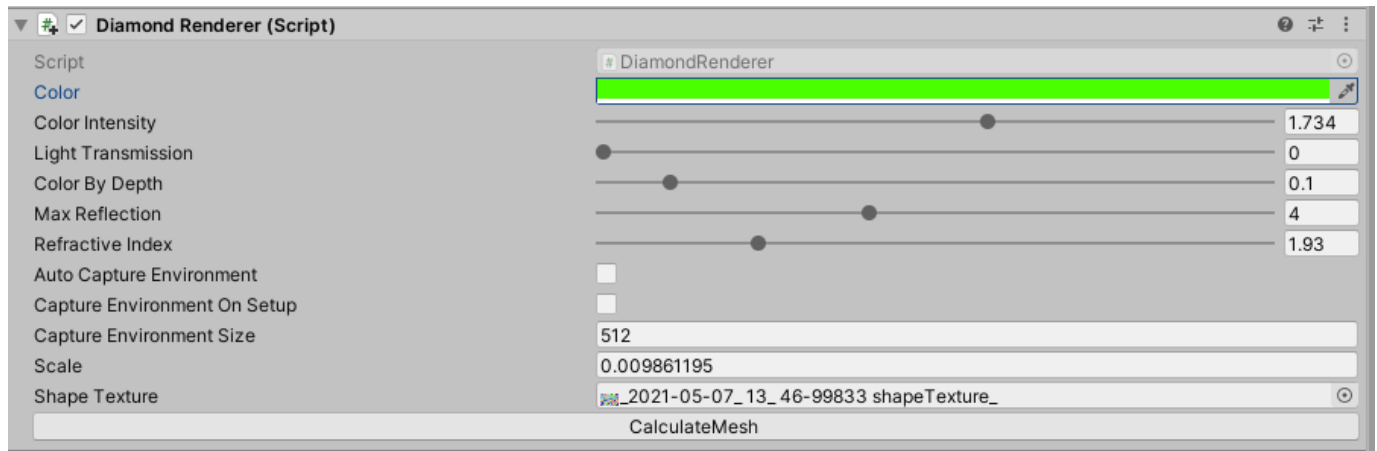
После этого нажмите на кнопку CalculateMesh, скрипт произведет расчеты и создаст на основе модели данные, которые шейдер будет обрабатывать каждый кадр, такой подход даст хорошую производительность. При любом изменении модели или смене геометрии, нажимайте эту кнопку.



Следующий шагом обычно будет выбор нужной кубмапы либо сменить тип на ReflectionProbe. Всего 2 кубмапы, 1 кубмапа нужна для внутренних преломлений и отражений камня а второй для внешнего отражения (регулируется значением Specular)



Разберем параметры скрипта.



Color: устанавливает цвет для выбранной модели, альфа этого цвета отвечает за интенсивность насыщенности.

Color Intensity: Этот параметр усиливает насыщенность цвета

ColorByDepth: этот параметр нужен для того чтобы установить насколько сильно цвета бриллианта должны зависеть от глубины преломления света в бриллианте, если при изменениях этого параметра бриллиант станет черным, уменьшите значение "ColorIntensity"

Light Transmission: этот ползунок отвечает за просвечивание светлых участков кубмапы из цвета, т.е. если у вас например цвет камня задан = зеленый а Light Transmission = больше чем 1, то там где у вас светлые участки кубмапы там будет менее интенсивный зеленый цвет, менее насыщенным, он будет более безцветным, попробуйте с каким нибудь ярким цветом.

Max Reflection: Очень важный параметр, от него сильно зависит производительность и внешний вид бриллианта, это количество проходов которые выполняет шейдер, чем больше проходов тем больше внутренних отражений и преломлений, чем выше параметр тем медленнее работает шейдер, для мобильных телефонов не советую ставить высокие значения, для маленьких камней лучше ставить 3-4.

RefractiveIndex: это коэффициент преломления, обычно в реальном мире у бриллианта он равняется = 2.417, но я часто ставлю ниже это значение так как часто это выглядит лучше, я заметил так делают многие художники, но решать вам, все зависит от вашей сцены и ваших целей.

Auto Capture Environment: Если включить то будет записываться сцена в кубмапу и использоваться в материале (не работает для AR, по поводу AR будет написано в конце документации)

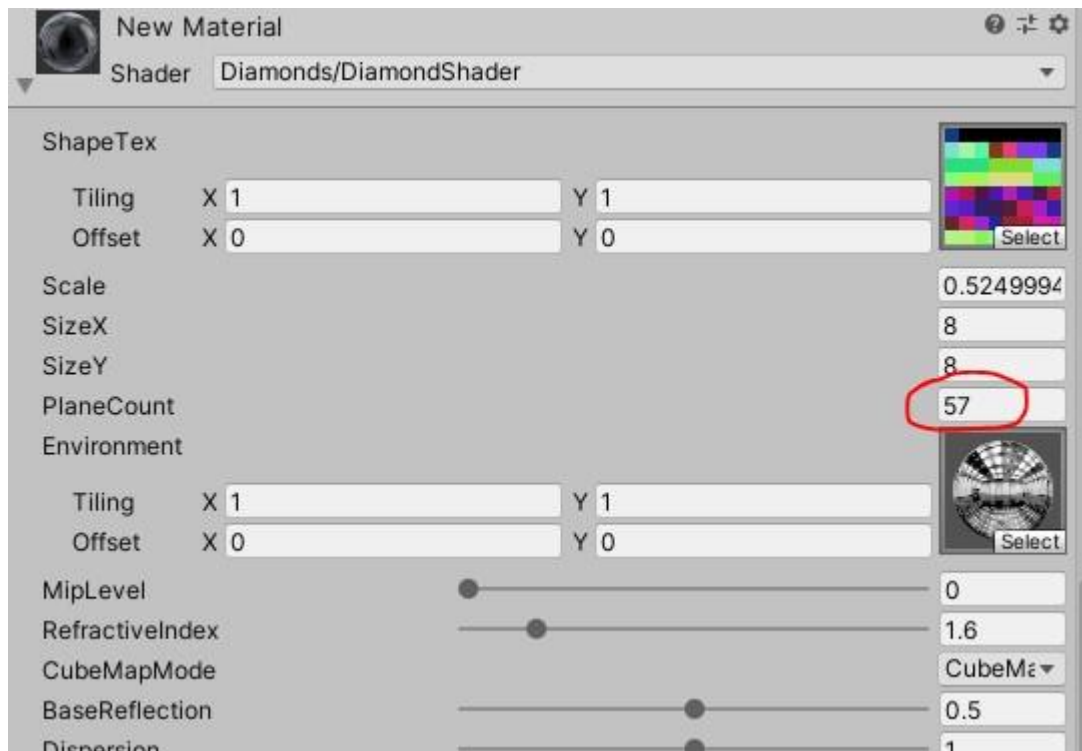
Capture Environment on Setup: кубмапа запишется только при нажатии кнопки CalculateMesh

Capture Environment Size: размер записываемой кубмапы

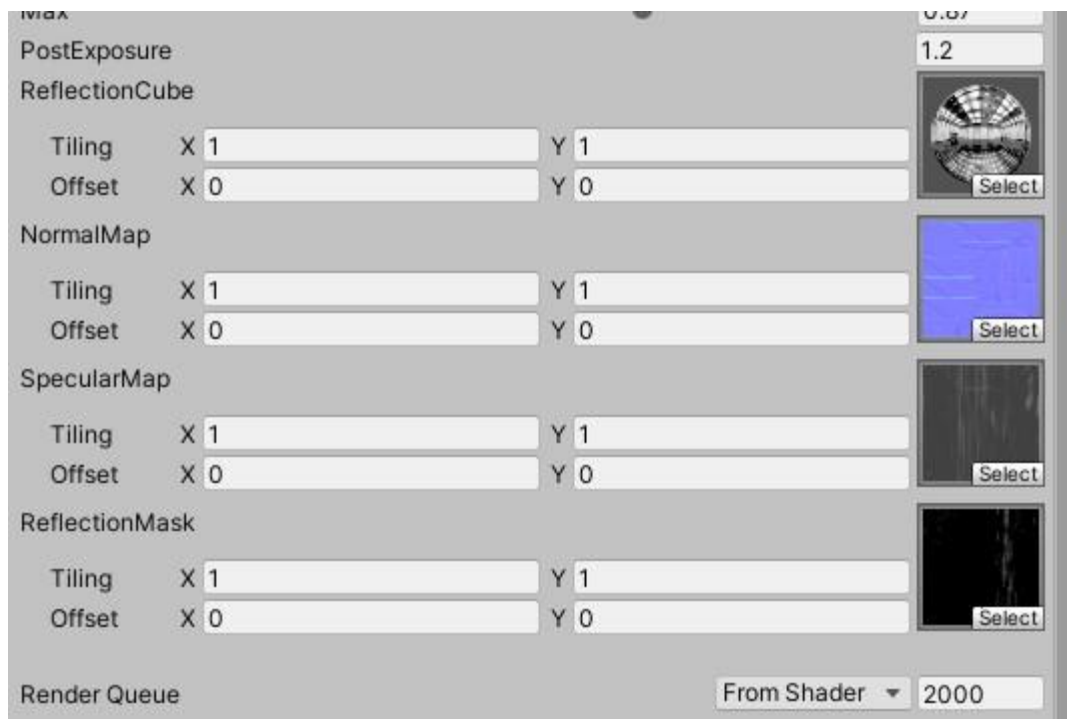
Scale: Значение этого параметра задается автоматически скриптом, он определяет правильный

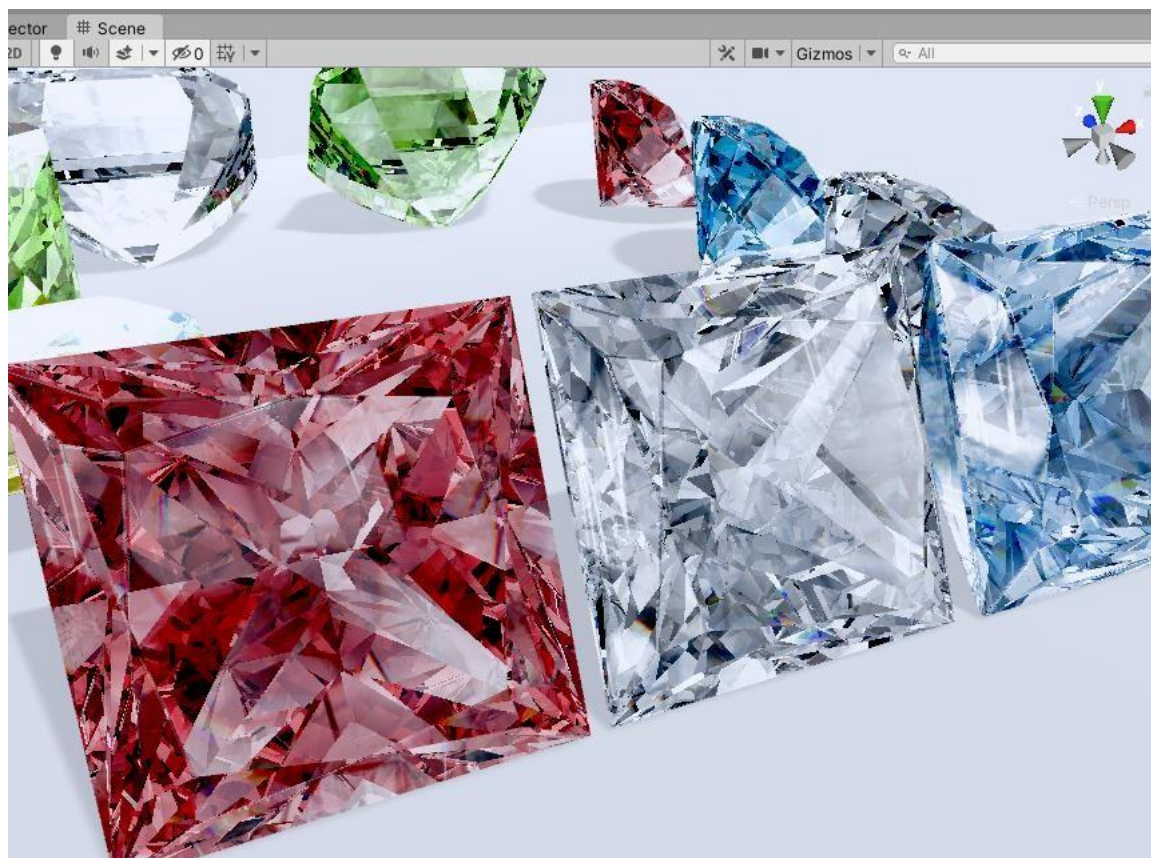
ShapeTexture: Это данные которые скрипт записал в текстуру, если он записал слишком много пикселей то возможно ваша модель имеет слишком много полигонов.

По поводу полигонов, убедитесь что ваша модель имеет достаточно низкое количество полигонов, от этого зависит производительность очень сильно. Ниже пример моей модели, Plane Count в материале означает количество полигонов которые будут просчитываться в шейдере, если у вас слишком большое значение то снизьте количество полигонов в своей программе где вы делаете модели, обычно мне помогает такой инструмент в Cinema4d как polygon reduction, он же есть и в blender и возможно в других программах, если ваша топология изначально правильная то часто этот инструмент создает хорошую огранку при значительном снижении полигонов.



По поводу фасок на ребрах камня, их лучше не делать, вместо этого используйте запеченные нормали и SpecularMap чтобы создать иллюзию фасок, ниже я привел пример с демо сцены, получился неплохой результат при том что я не делал это долго, отдайте эту задачу 3d художнику который умеет работать с запеканием нормалей и текстурированием чтобы сделать отличный результат, эти 3 текстуры позволят так же добавить разводы на камнях и сделать их более реалистичными.





Разберем все параметры материала:

ShapeTex

Tiling X 1 Y 1
Offset X 0 Y 0



SizeX

8

SizeY

8

PlaneCount

57

Environment

Tiling X 1 Y 1
Offset X 0 Y 0



MipLevel

0

RefractiveIndex

1.7

CubeMapMode

CubeMap ▾

BaseReflection

0.4

Dispersion

0.92

DispersionR

-0.326

DispersionG

-0.167

DispersionB

0.068

DispersionIntensity

1

TotalInternalReflection
specular

1

7.05

Brightness

1.075

Power

0.953

Contrast

1

_Disaturate

1

Min

0

Max

0.87

PostExposure

1.2

ReflectionCube

Tiling X 1 Y 1
Offset X 0 Y 0



NormalMap

Tiling X 1 Y 1
Offset X 0 Y 0



SpecularMap

Tiling X 1 Y 1
Offset X 0 Y 0



ReflectionMask

Tiling X 1 Y 1
Offset X 0 Y 0



ShapeTex: Как я уже писал - это просчитанные полигоны из скрипта, они сохраняются в папку ShapeTextures со своим уникальным номером и датой. SizeXY: размеры ShapeTex

PlaneCount: количество полигонов которые будут обрабатываться, убедитесь что он получился достаточно низким

Environment: кубмапа которая используется для преломлений и отражений внутри камня

MipLevel: если включены MipMaps в кубмапе то вы можете выбирать MipLevel который будет использоваться, на производительность почти не влияет.

RefractiveIndex: это коэффициент преломления, обычно в реальном мире у бриллианта он равняется $= 2.417$, но я часто ставлю ниже это значение так как часто это выглядит лучше, я заметил так делают многие художники, но решать вам, все зависит от вашей сцены и ваших целей.

CubMapMode: Выберите что использовать, кубмапу или ReflectionProbe из сцены (ReflectionProbe особенно важен для AR, об этом ниже)

BaseReflection: Это значение определяет насколько сильно будут видны отражение по отношению к преломленному свету, всегда на уровне 0.3 - 0.6

Dispersion: Это настройка дисперсии цветов, поищите в интернете больше информации что это такое, уровень определяет насколько далеко пучки света разделяются друг от друга образуя разный цвет.

DispersionRGB: эти 3 значения нужны для того чтобы разделить каждый цвет друг от друга с разной длиной.

DispersionIntensity: это насыщенность или же интенсивность эффекта дисперсии

TotalInternalReflection: Это значение для того чтобы определить с какого места можно прерывать отражения, желательно оставить его на исходном уровне, но может кому то пригодиться

Specular: Это регулирует интенсивность внешних отражений.

Brightness: Яркость внутренних преломлений\отражений.

Power: Возводит выходный результат в степень, делает более мощный и сочный результат по типу как контрастность но работает немного иначе.

Contrast: Это контрастность выходного результата, светлые участки светлее, темные - темнее

Disaturate: В светлых участках снижает насыщенность

Min: регулирует темноту в камне

Max: Регулирует светлость

PostExposure: Высвечивает итоговый результат, делает светлым либо темным

ReflectionCube: Это кубмапа для внешних отражений

NormalMap: Это карта нормалей, она необходима для предания объему для внешних отражений, без

ReflectionCube нормальная карта бесполезна

SpecularMap: Эта карта для изменения интенсивности внешних отражений

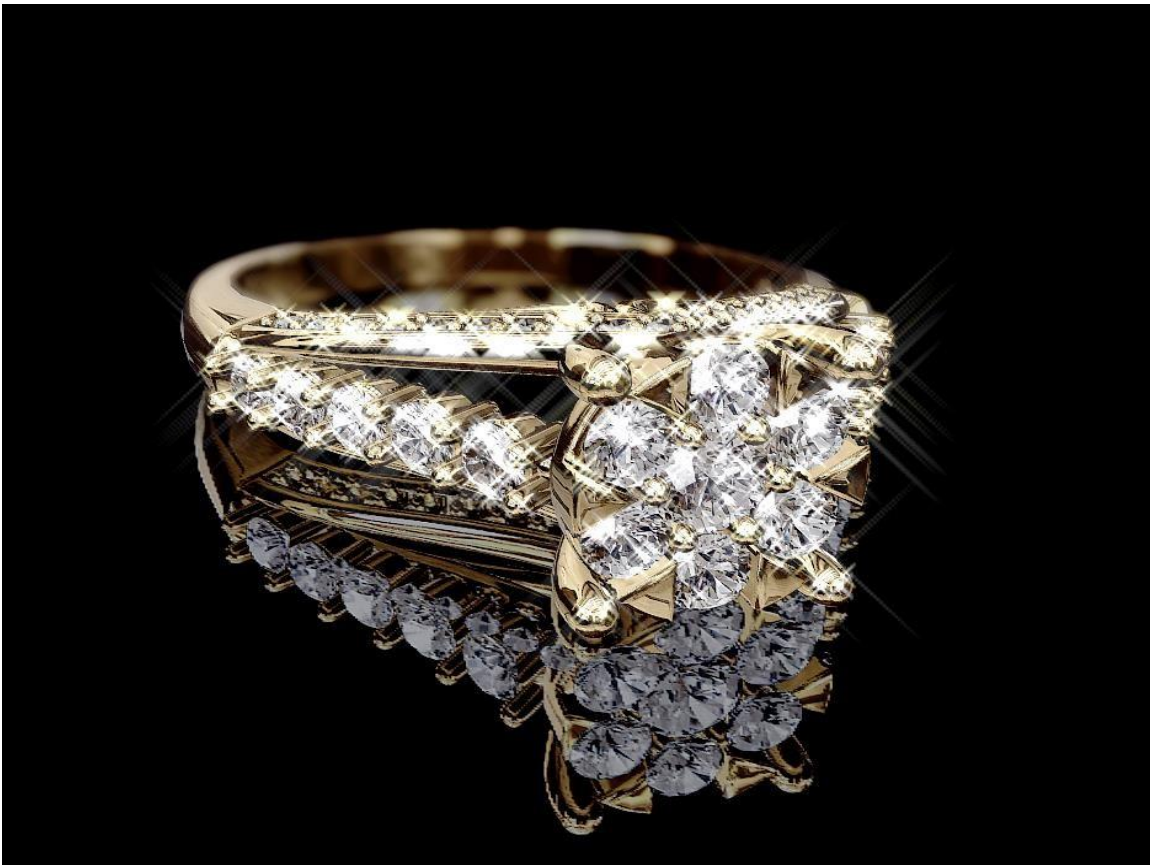
ReflectionMask: добавляет отражения вне зависимости от значения "specular"

Flares

Кроме шейдера для бриллиантов и скрипта, в ассете так же присутствует постэффект с бликами.

URP

Для использования в URP, импортируйте Unitypackage из папки URP. Используйте файл "UniversalRenderPipelineAsset_Renderer Flares" в вашем Universal Render Pipeline Asset.



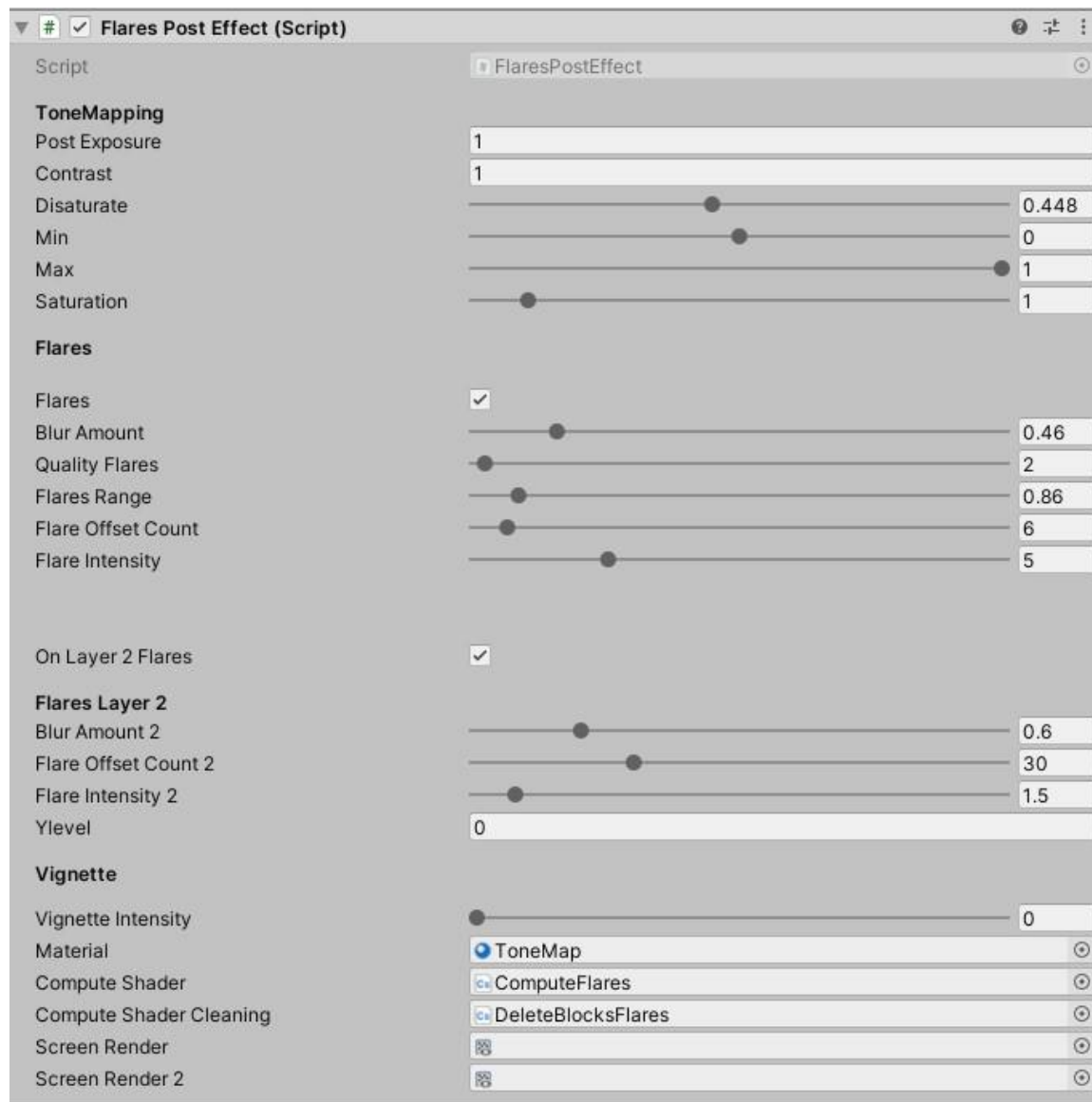
Первый блок это базовые значения для коррекции изображения, чтобы можно было его использовать без стандартного постпроцессинга, так же внизу 2 слоя бликов, для мелких точек и крупных точек на экране, он вычисляет светлые точки и создает в этих местах блики.

Blur Amount: это сдвиг всех пикселей лучей, настройте его так чтобы луч выглядел без шума. Quality Flares: чем ниже значение тем выше качество и больше пикселей используется для создания бликов, это поможет изменить разбег желаемых лучей, на производительность это не влияет.

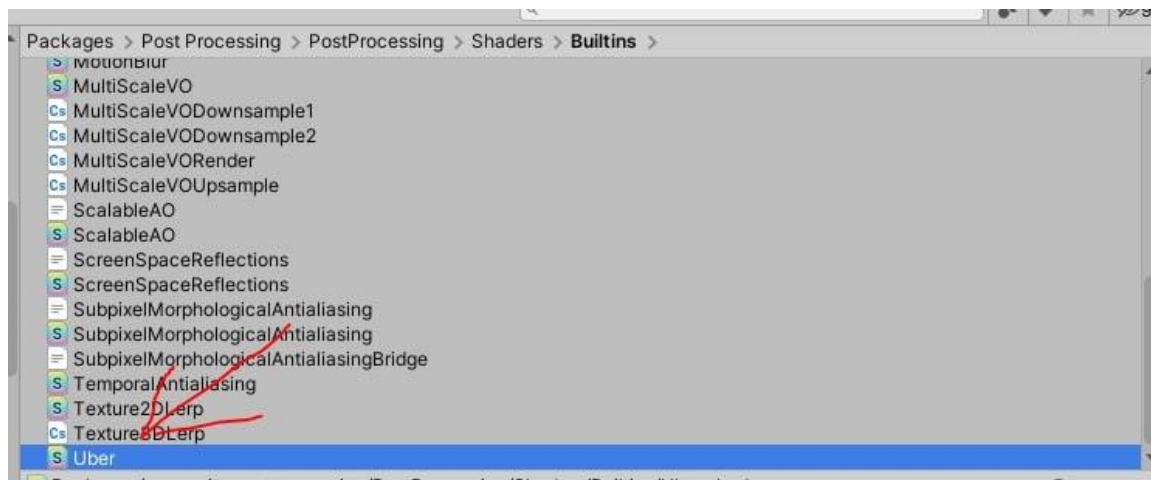
Flares Range: Этим ползунком можно отрегулировать порог для создания бликов

Flares Offset Count: Внимание этот ползунок сильно влияет на производительность, чем выше значение тем больше проходов происходит и тем ровнее получается луч и его можно сделать длинее. Flare Intensity: Это прозрачность лучей

Ylevel: Это позиция в мировом пространстве, за границы которой лучи не появляются, например чтобы лучи не срабатывали от белого пола, стандартное значение = 0, от него и выше и работают лучи



Теперь я опишу что делать если блики слишком сильно мелькают при сочетании с стандартным пост эффектами (без них все нормально), это бывает из за компонента Post-Process Layer и его антиалиасинга, для того чтобы исправить хотя бы часть сильного подергивания - нужно зайти в эту папку ниже и открыть файл Uber.shader



Вам нужно найти строчку которая зеленная, у вас она будет белая и активная, закомментируйте как на скриншоте и добавьте вместо неё запись: `output.rgb = output.rgb;`

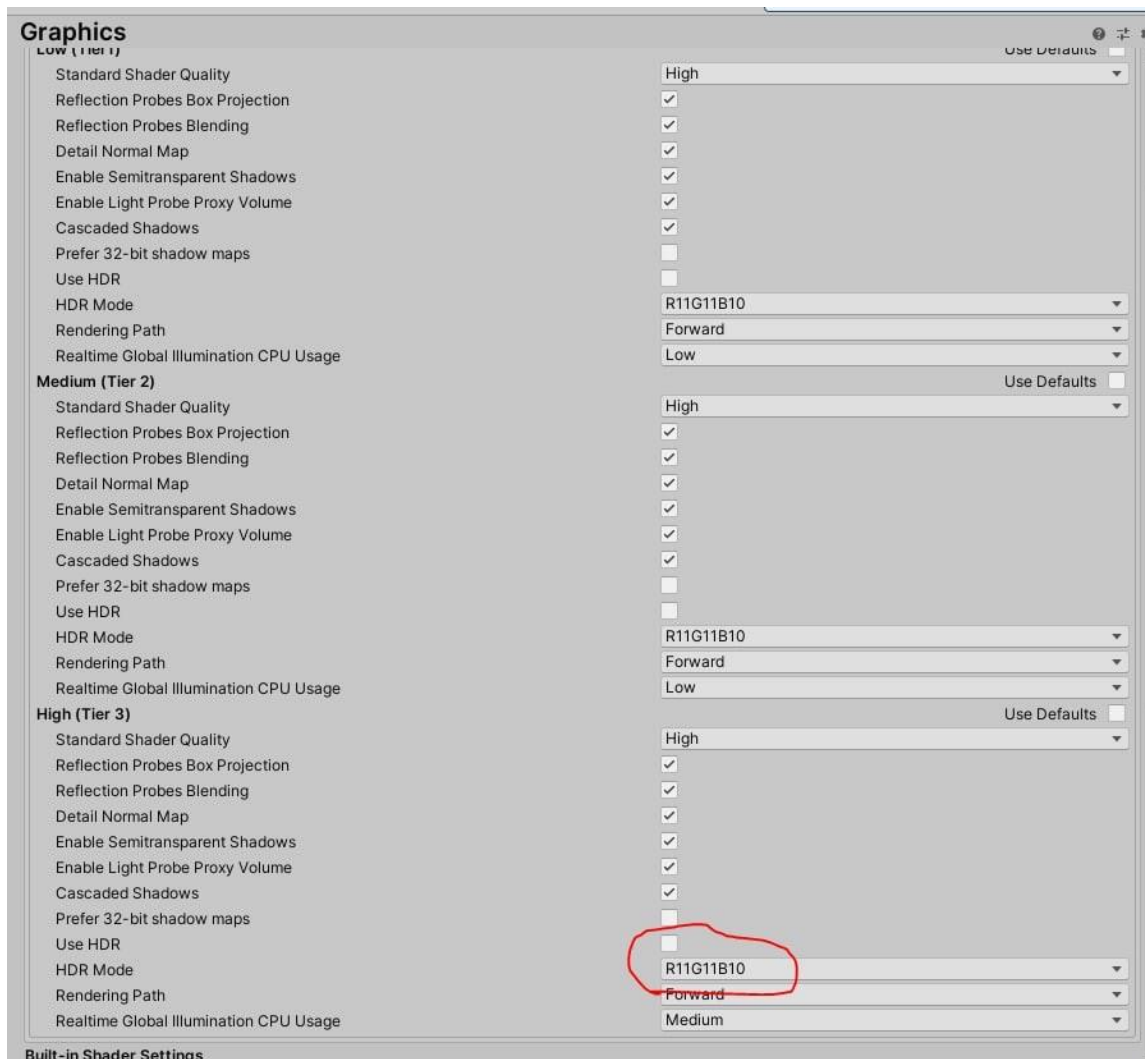
Это отключит встроенный дизеринг и шум который создает Post-Process Layer снизиться

```

240 |         output = LinearToSRGB(output);
241 |     }
242 |     #endif
243 |
244 |     output.rgb = output.rgb;
245 |     //output.rgb = Dither(output.rgb, i.texcoord);
246 | }
247 | #else
248 | {
249 |     UNITY_BRANCH
250 |     if (_LumaInAlpha > 0.5)

```

В моем проекте я использовал такие настройки графики:



Оформленные сцены

Для создания реалистичных камней в сценах, пришлось перебрать 600 HDRI и более подходящие осталось всего меньше десятка, так что напишите в интернете например **"красивый интерьер"** и скачайте штук 300 сразу и проверяйте все их, чтобы подобрать нужное для вашей сцены. Шейдер делает все возможное для создания реалистичных камней, он делает правильный рендер бриллиантов как и жизни, все остальное зависит от оформления сцены.

По поводу антиалиасинга, советую использовать TemporalAntialiasing чтобы грани внутри бриллиантов были более четкие, если подкрутить значение Sharpness.

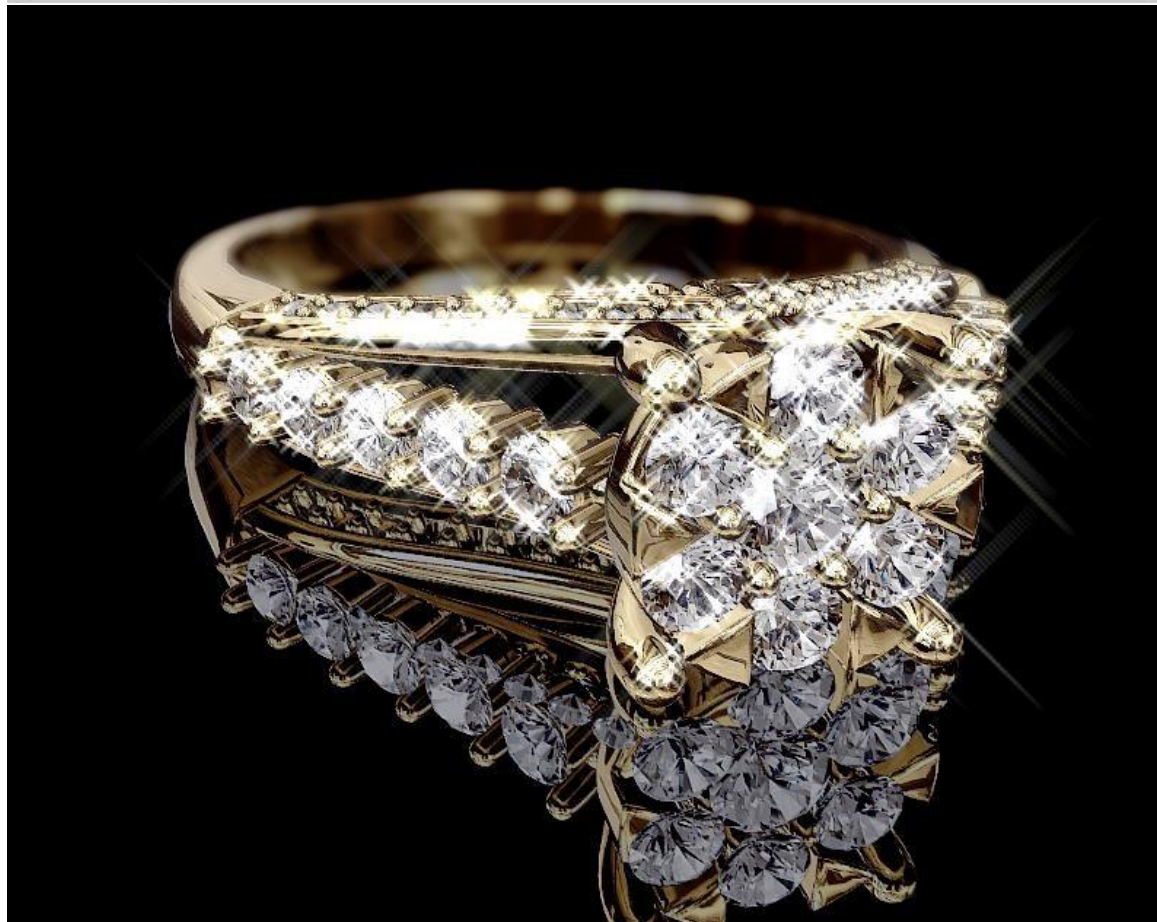
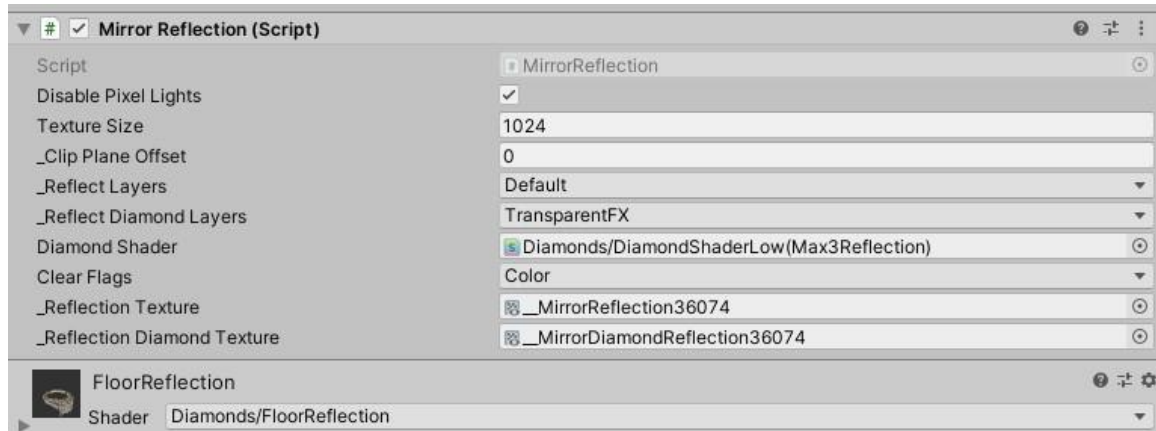
При использовании Color Grading используйте лучше режим custom, чтобы светлые участки не затемнялись.

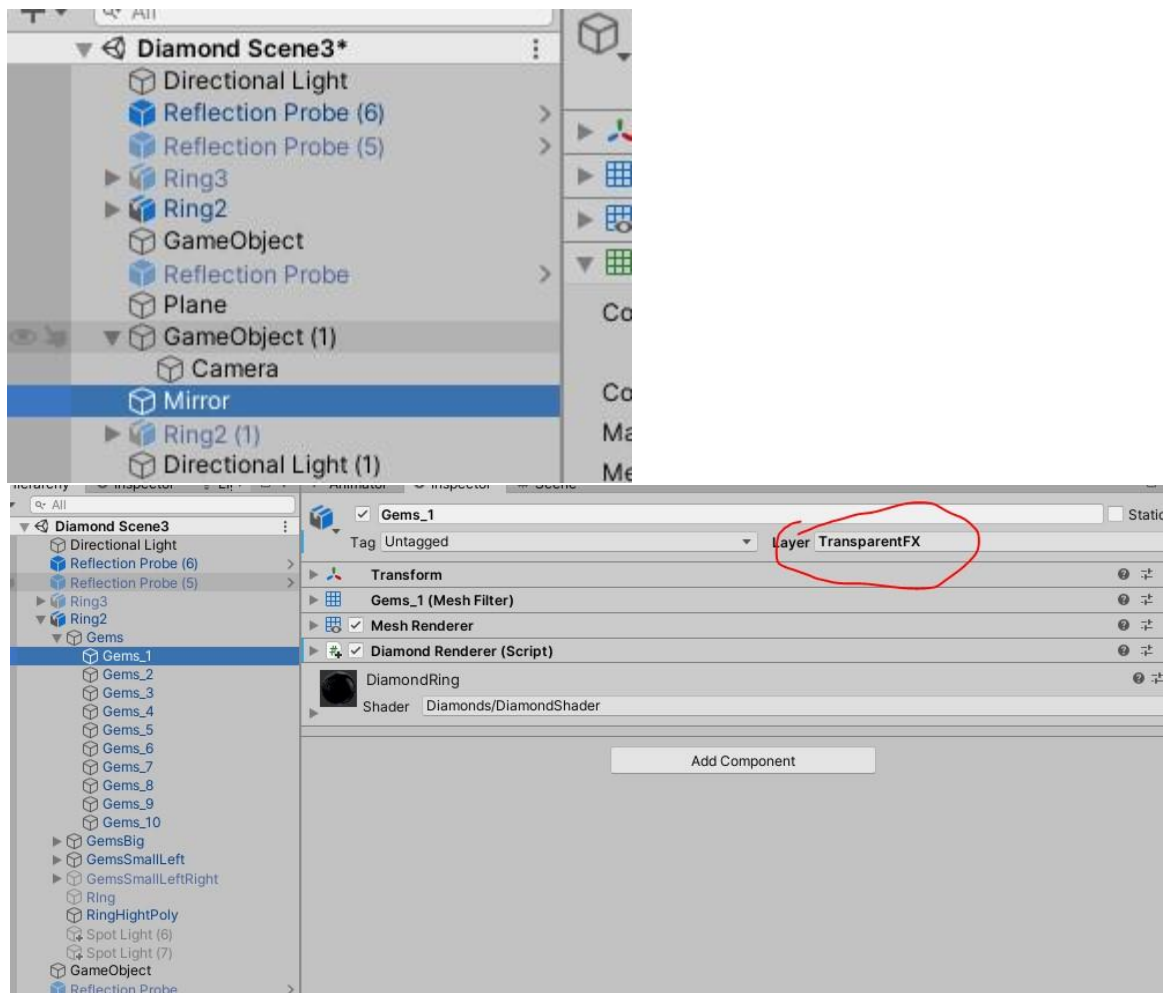
Mirror Reflection

Для некоторых сцен будет полезен скрипт "MirrorReflection"

Он необходим для создания отражений на полу, с помощью шейдера вы сможете изменять прозрачность отражений а в скрипте настраивать размер, размер будет сильно влиять на производительность, я создал так же второй шейдер для бриллиантов у которого есть ограничения по проходам, значение "Diamond Shader".

Он необходим для того чтобы в отражениях рендерились не камни со сцены а камни с замененным шейдером, с более быстрым, для этого в скрипте есть разделение по слоям,
_Reflect Layers: это первый слоя который рендерит камера
_Reflect Diamond Layers: это слой для бриллианта где будет заменен шейдер в материалах





AR

Для того чтобы в камни транслировалось изображение с камеры и делать преломления с учетом окружения, вам нужно использовать "AR Environment Probe Manager"

<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/environment-probe-manager.html>

Вам нужно будет сделать префаб с ReflectionProbe и с этим компонентом и создавать его на сцене, а в материалах камней поставить в режим использования ReflectionProbe.

Мобильные приложения и оптимизация

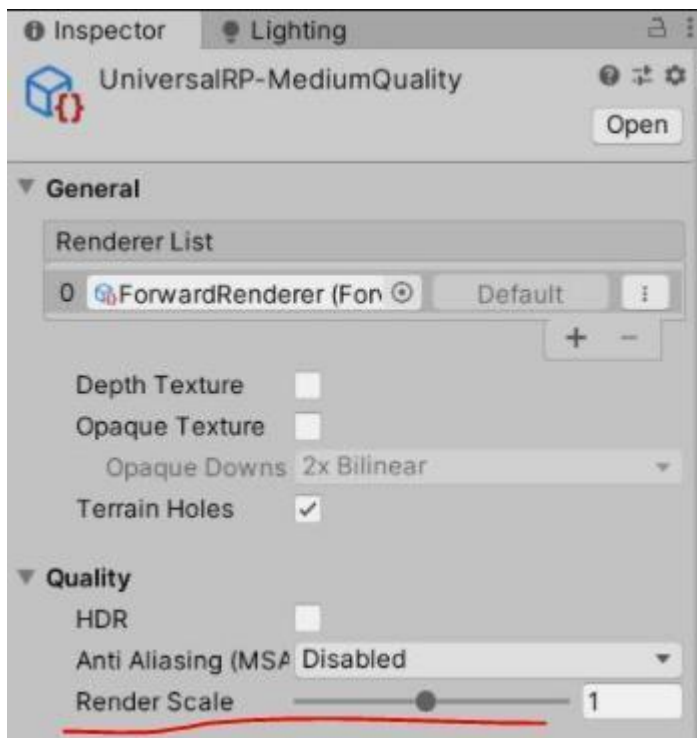
Перед тем как создавать приложения для мобильных устройств, удалите скрипт

"ImportPostProcess.cs" иначе он будет постоянно менять цветовое пространство на Linear, это необходимо было лишь для демонстрации сцен, так как они создавались именно на Linear, используйте все выше описанные параметры которые влияют на производительность чтобы все работало быстро на мобильных устройствах, Самые основные это: Max Reflection в скрипте;

количество полигонов в модели бриллианта;
Flares Offset Count в скрипте для flares;
скрипт MirrorReflection и его размер рендера.

хочу так же отметить что на производительность оказывает большее влияние именно размер бриллиантов на экране а не количество бриллиантов, так как вычисления происходят по пиксельно, чем меньше пикселей занимает бриллиант тем меньше вычислений нужно выполнять, учитывайте это при создании сцен.

Так же для еще большей оптимизации вы можете перейти на URP и использовать такую фичу как размер рендера, чем меньше этот параметр - тем больше будет FPS, но менее качественный будет рендер.



Напишите мне на почту если вам нужна помощь, любая помощь связанная с этим активом, andreygraphics.online@gmail.com

Version: 1.3