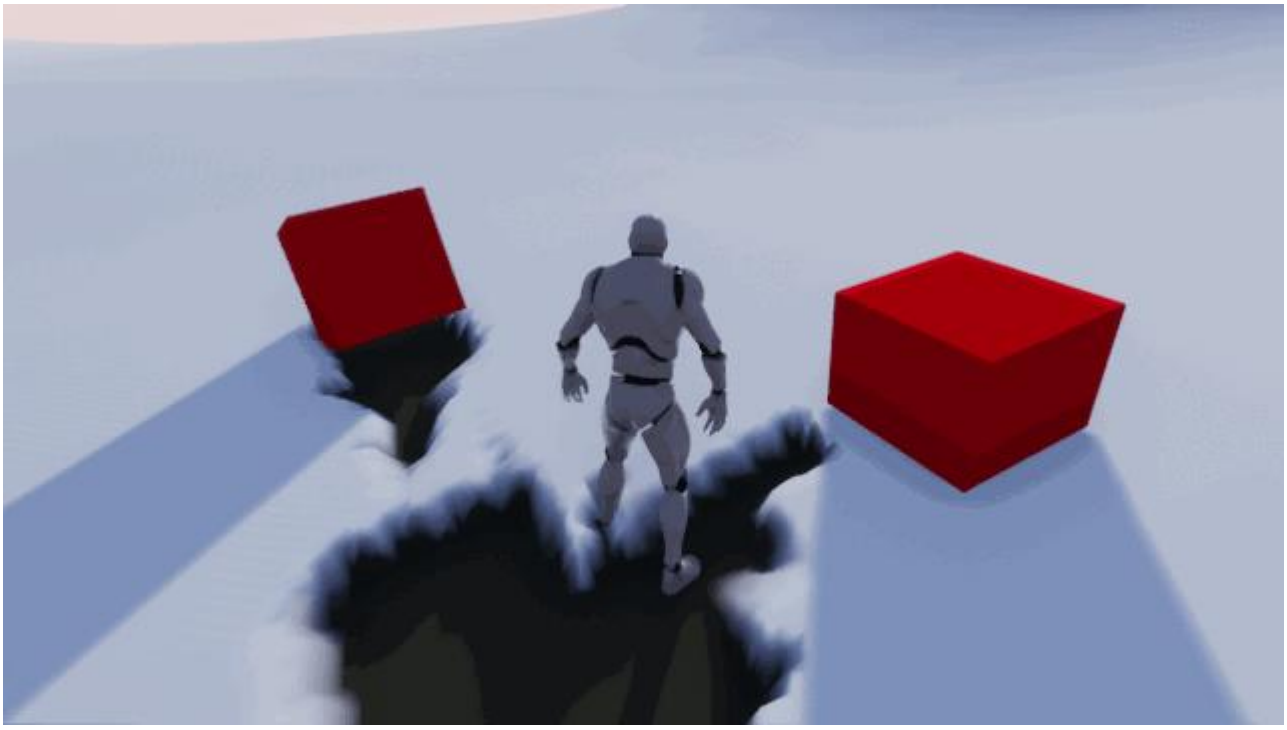
Создание следов на снегу в Unreal Engine 4



Если вы играете в современные AAA-игры, то могли заметить тенденцию использования покрытых снегом ландшафтов. Например, они есть в *Horizon Zero Dawn*, *Rise of the Tomb Raider* и *God of War*. Во всех этих играх у снега есть важная особенность: на нём можно оставлять следы!

Благодаря такому взаимодействию с окружением усиливается погружение игрока в игру. Оно делает окружение более реалистичным, и будем честными — это просто интересно. Зачем тратить долгие часы на создание любопытных механик, если можно просто позволить игроку упасть на землю и делать снежных ангелов?

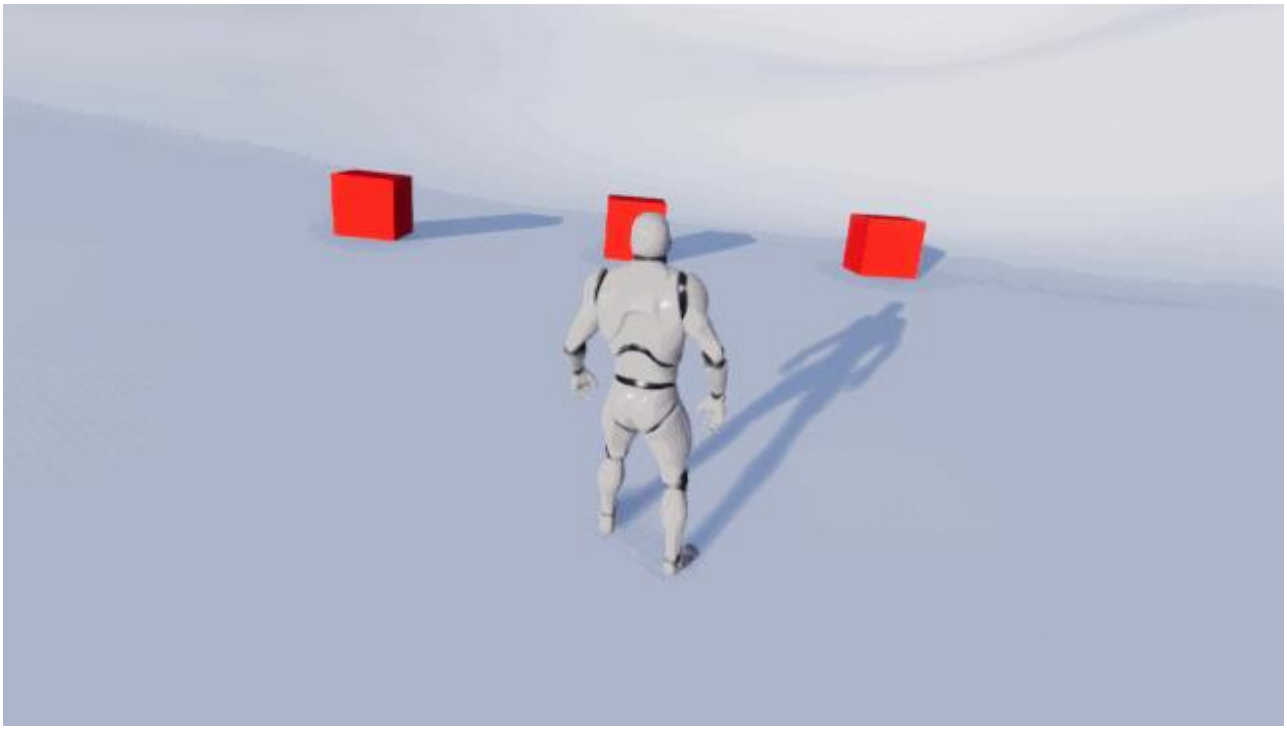
В этом туториале вы научитесь следующему:

- Создавать следы с помощью захвата сцены для маскировки объектов, близких к земле
- Использовать маску с материалом ландшафта, чтобы создавать деформируемый снег
- Для оптимизации отображать следы на снегу только рядом с игроком

Приступаем к работе

Скачайте материалы (https://koenig-media.raywenderlich.com/uploads/2018/06/SnowDeformation.zip) для этого туториала. Распакуйте их, перейдите в *SnowDeformationStarter* и

откройте *SnowDeformation.uproject*. В этом туториале мы будем создавать следы с помощью персонажа и нескольких ящиков.



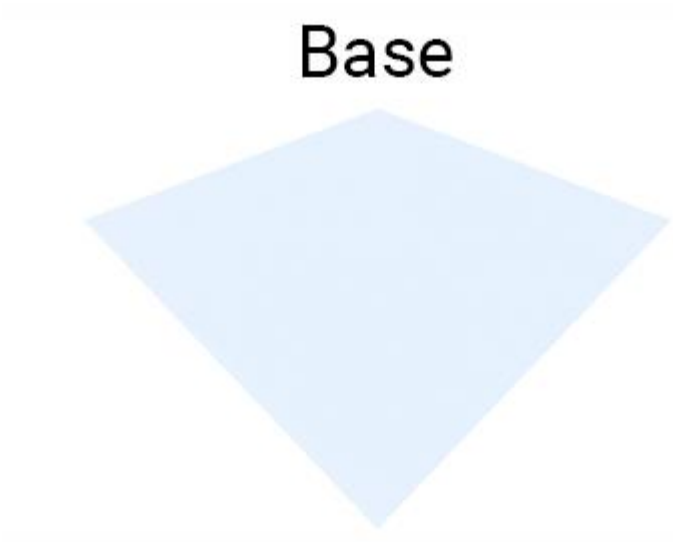
Прежде чем мы начнём, вам нужно знать, что способ из этого туториала будет сохранять следы только в заданной области, а не во всём мире, потому что скорость зависит от разрешения целевого рендера.

Например, если мы хотим хранить следы для большой области, то придётся увеличивать разрешение. Но это также увеличивает влияние захвата сцены на скорость игры и объём памяти под целевой рендер. Для оптимизации необходимо ограничить область действия и разрешение.

Разобравшись с этим, давайте узнаем, что нужно для реализации следов на снегу.

Реализация следов на снегу

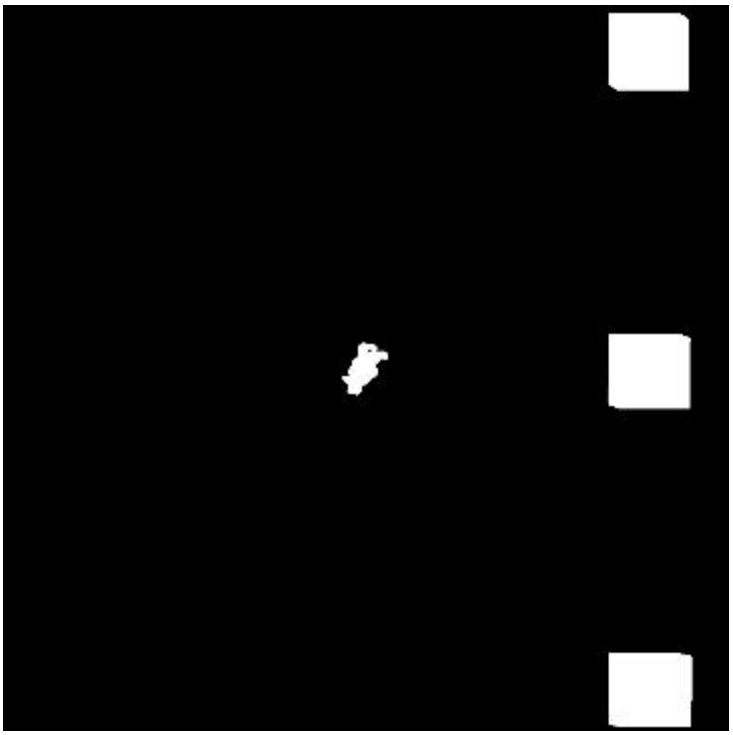
Первое, что нужно для создания следов — это *целевой рендер*. Целевой рендер (render target) будет маской в градациях серого, в которой белый цвет обозначает наличие следа, а чёрный — его отсутствие. Затем мы можем спроецировать целевой рендер на землю и использовать его для смешивания текстур и смещения вершин.



Второе, что нам потребуется — это способ маскирования только влияющих на снег объектов. Это можно реализовать, сначала рендеря объекты в *Custom Depth*. Затем можно использовать *захват сцены (scene capture)* с *материалом постобработки (post process material)* для маскировки всех объектов, отрендеренных в Custom Depth. Потом можно вывести маску в целевой рендер.

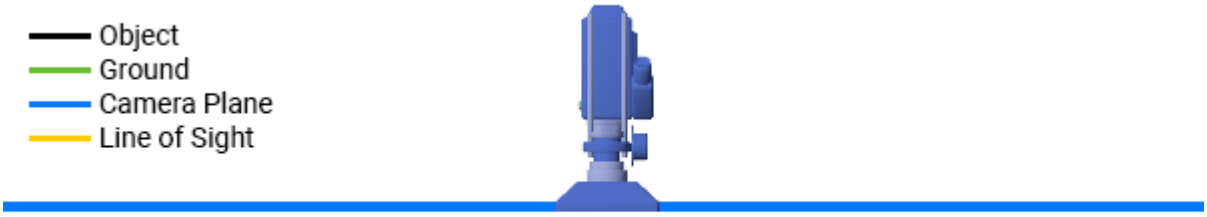
Примечание: захват сцены (scene capture) — это, по сути, камера с возможностью вывода целевого рендера.

Самая важная часть захвата сцены — это *место* его расположения. Ниже показан пример целевого рендера, захваченного из *вида сверху*. Здесь маскируются персонаж от третьего лица и ящики.

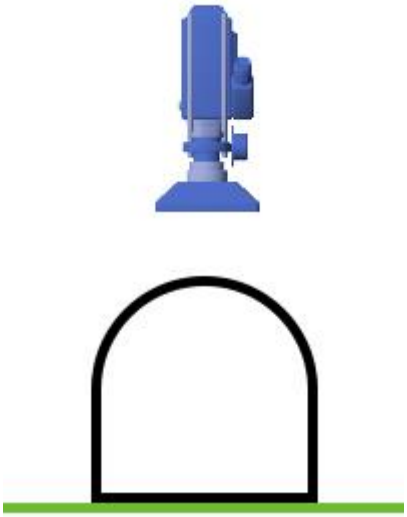


На первый взгляд захват с видом сверху нам подходит. Формы выглядят соответствующим мешам, поэтому проблем быть не должно, правда?

Не совсем. Проблема захвата из вида сверху заключается в том, что он не захватывает ничего под самой широкой точкой. Вот пример:



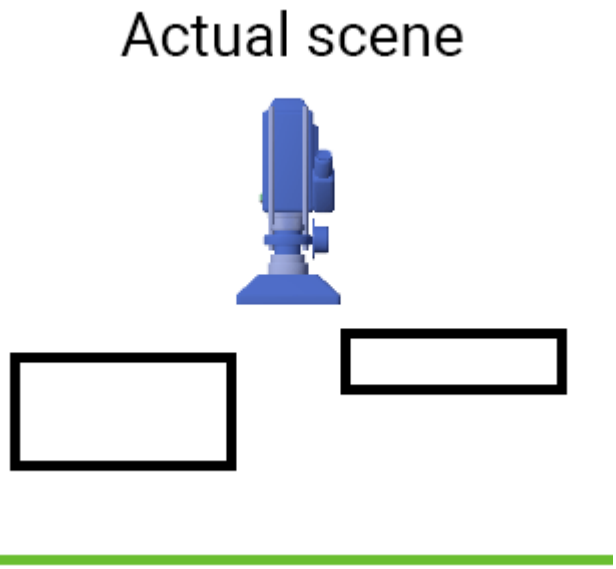
представьте, что жёлтые стрелки проходят весь путь до земли. В случае куба и конуса остриё стрелки всегда будет оставаться внутри объекта. Однако в случае сферы остриё при приближении к земле выходит из неё. Но по мнению камеры остриё всегда находится внутри сферы. Вот как будет выглядеть сфера для камеры:



Поэтому маска сферы будет больше, чем должна, даже если область контакта

с землёй мала.

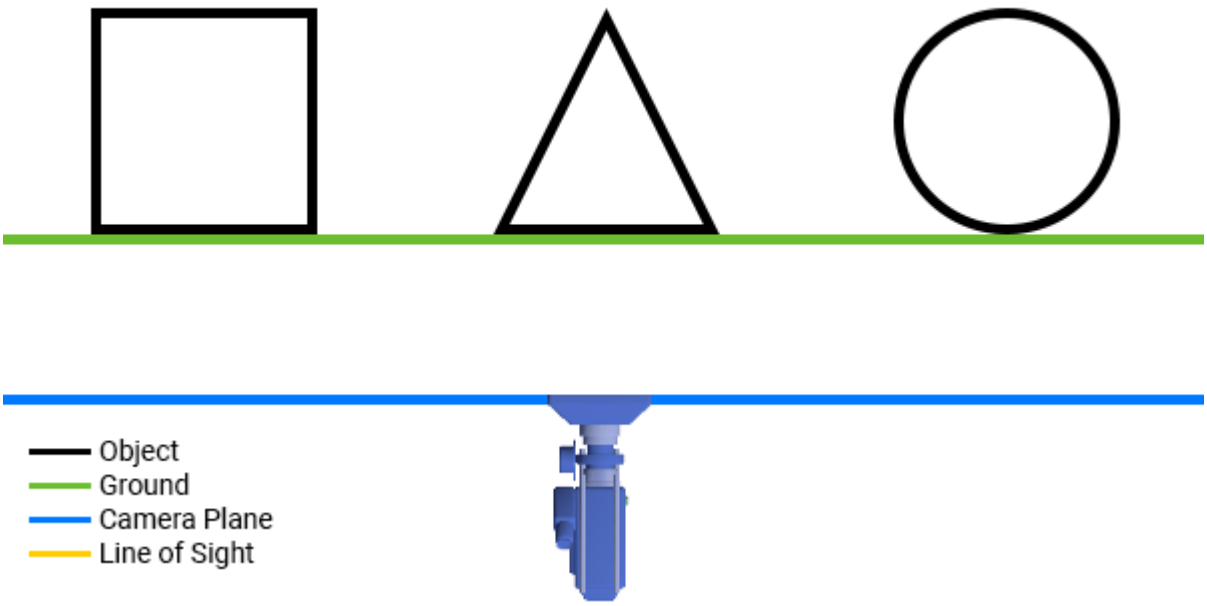
Кроме того, эта проблема дополняется тем, что нам сложно определить, касается ли объект земли.



Справиться с обеими этими проблемами можно с помощью захвата *снизу*.

Захват снизу

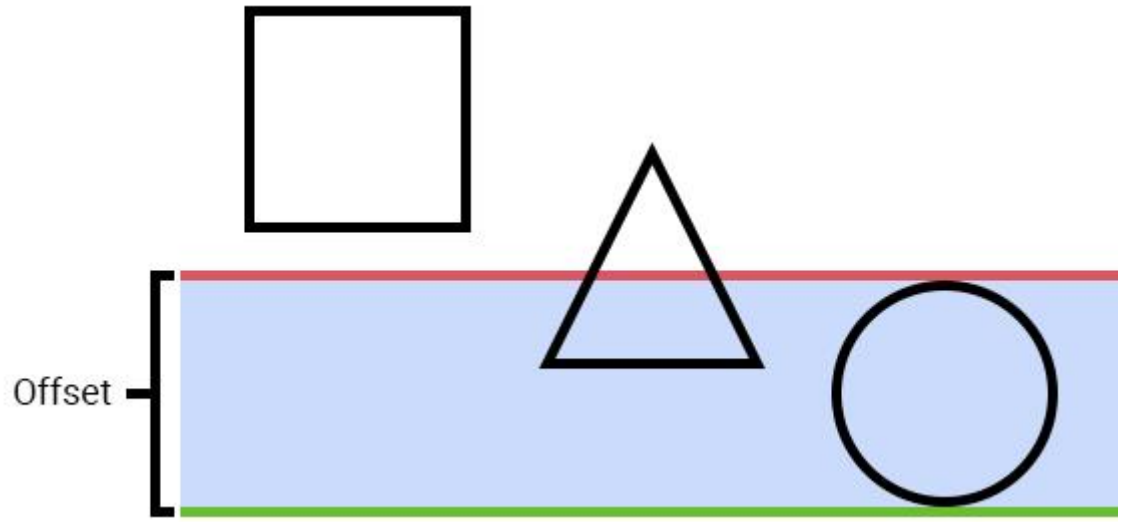
Захват снизу выглядит следующим образом:



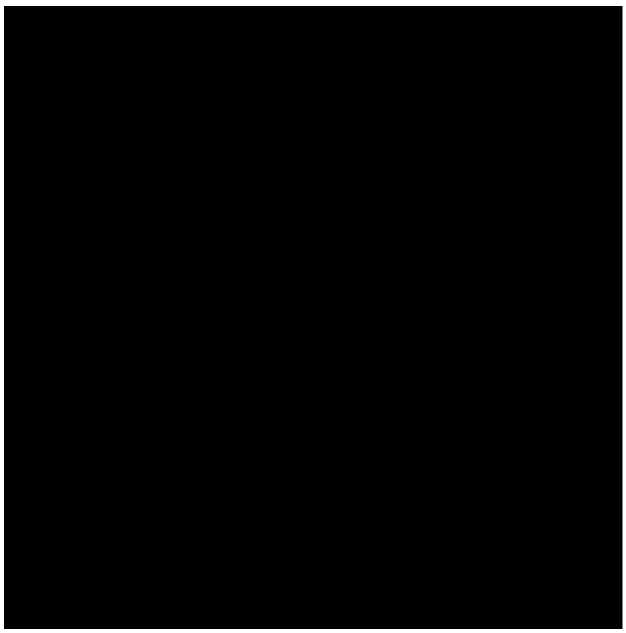
Как видите, камера теперь захватывает нижнюю сторону, то есть ту, которая

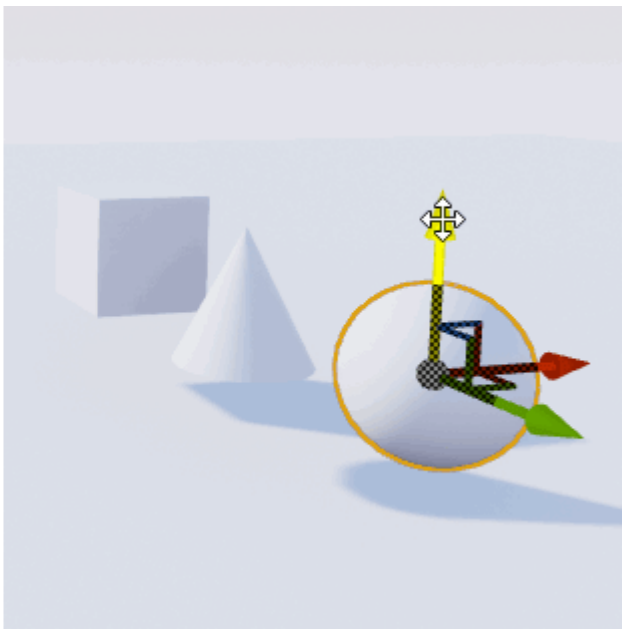
касается земли. Это устраняет проблему «самой широкой области», появляющуюся при захвате сверху.

Чтобы определить, касается ли объект земли, можно для выполнения проверки глубины использовать материал постобработки. Он проверяет, больше ли глубина объекта глубины земли *и* ниже ли она заданного смещения. Если оба условия соблюдаются, то мы можем замаскировать этот пиксель.



Ниже представлен пример внутри движка с зоной захвата в 20 единицах над землёй. Заметьте, что маска появляется только когда объект проходит через определённую точку. Также заметьте, что маска становиться белее при приближении объекта к земле.





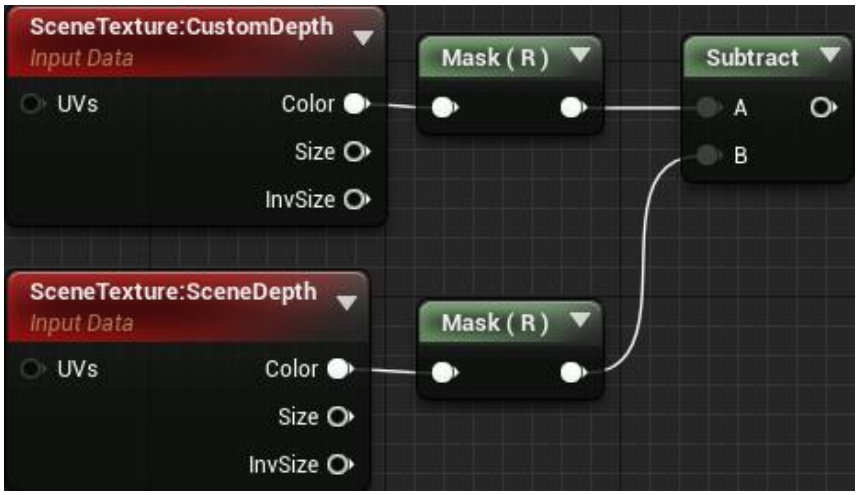
Для начала создадим материал постобработки для выполнения проверки глубины.

Создание материала проверки глубины

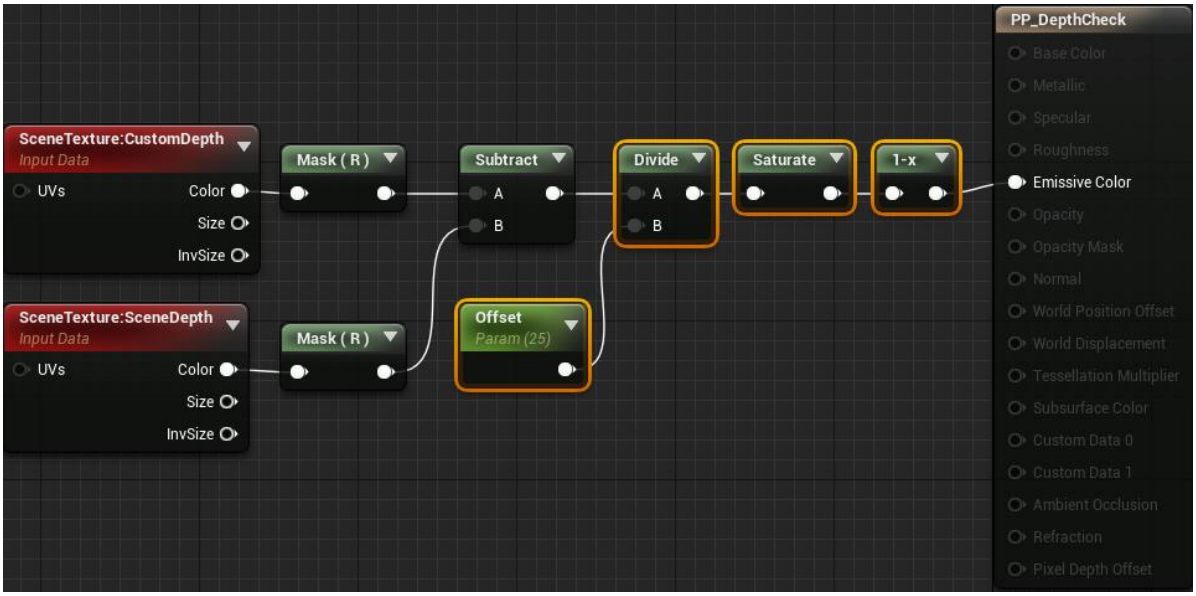
Для выполнения проверки глубины нужно использовать два буфера глубины — один для земли, другой для влияющих на снег объектов. Так как захват сцены видит только землю, *Scene Depth* будет выводить глубину для земли. Чтобы получить глубину для объектов, мы просто будем рендерить их *Custom Depth*.

Примечание: для экономии времени я уже отрендерил персонажа и ящики в Custom Depth. Если вы хотите добавить другие влияющие на снег объекты, то необходимо включить для них *Render CustomDepth Pass*.

Во-первых, нужно вычислить расстояние каждого пикселя до земли. Откройте *Materials\PP_DepthCheck* и создайте следующее:



Далее необходимо создать зону захвата. Для этого добавьте выделенные ноды:



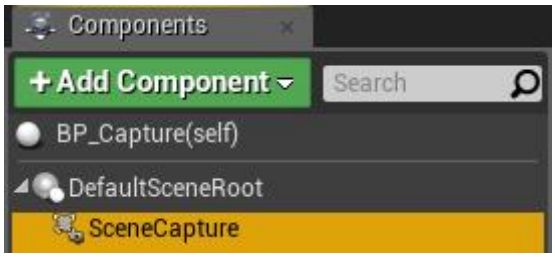
Теперь если пиксель находится в пределах *25* единиц от земли, то он появится в маске. Яркость маскирования зависит от того, насколько пиксель близок к земле. Нажмите на *Apply* и вернитесь в основной редактор.

Далее нужно создать захват сцены.

Создание захвата сцены

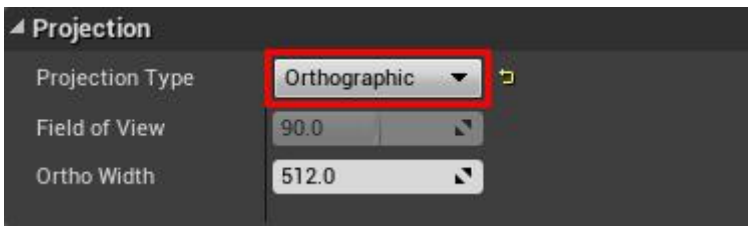
Сначала нам необходим целевой рендер, в который можно записывать захват сцены. Перейдите в папку *RenderTargets* и создайте новый *Render Target* под названием *RT_Capture*.

Теперь давайте создадим захват сцены. В этом туториале мы добавим захват сцены в блюпринт, потому что позже нам понадобится для него скрипт. Откройте *Blueprints\BP_Capture* и добавьте *Scene Capture Component 2D*. Назовите его *SceneCapture*.



Сначала нам нужно задать поворот захвата, чтобы он смотрел на землю. Перейдите в панель Details и задайте *Rotation* значения *(0, 90, 90)*.

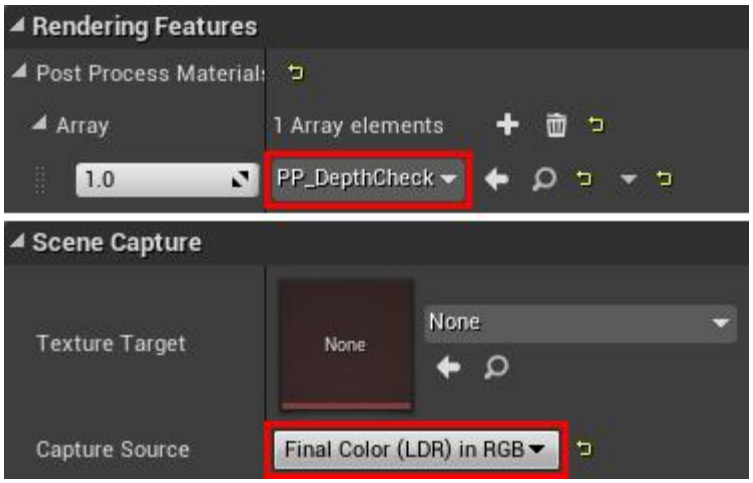
Дальше идёт тип проецирования. Поскольку маска — это 2D-представление сцены, нам нужно избавиться от перспективного искажения. Для этого зададим для *Projection\Projection Type* значение *Orthographic*.



Далее нам нужно сообщить захвату сцены, в какой целевой рендер

выполнять запись. Для этого выберем для *Scene Capture\Texture Target* значение *RT_Capture*.

Наконец, нам нужно использовать материал проверки глубины. Добавим к *Rendering Features\Post Process Materials PP_DepthCheck*. Чтобы постобработка работала, нам нужно также изменить *Scene Capture\Capture Source* на *Final Color (LDR) in RGB*.



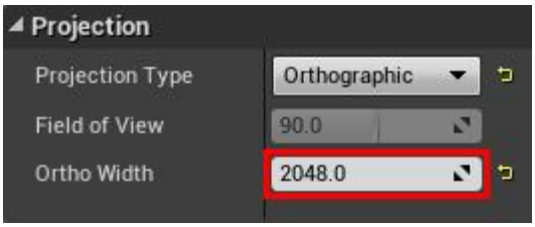
Теперь, когда захват сцены настроен, нам нужно указать размер области захвата.

Задание размера области захвата

Так как для целевого рендера лучше использовать низкие разрешения, нам нужно пользоваться пространством эффективно. То есть мы должны выбрать, какую область будет покрывать один пиксель. Например, если разрешения области захвата и целевого рендера одинаковы, то мы получаем соотношение 1:1. Каждый пиксель будет покрывать область 1×1 (в единицах измерения мира).

Для следов на снегу соотношение 1:1 не требуется, потому что такая детализация нам скорее всего не понадобится. Я рекомендую использовать бОльшие соотношения, потому что это позволит вам увеличить размер области захвата при низком разрешении. Но не делайте соотношение слишком большим, иначе начнут теряться детали. В этом туториале мы будем использовать соотношение 8:1, то есть размер каждого пикселя будет 8×8 единиц измерения мира.

Можно изменить размер области захвата, меняя свойство *Scene Capture\Ortho Width*. Например, если вы хотите выполнять захват области 1024×1024, то задайте значение 1024. Так как мы используем соотношение 8:1, задайте значение *2048* (разрешение целевого рендера по умолчанию равно 256×256).



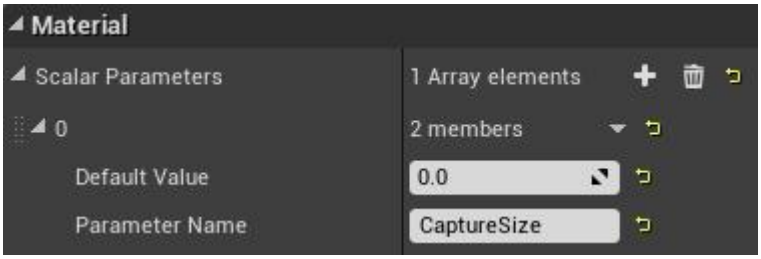
Это значит, что захват сцены будет захватывать область *2048×2048*. Это приблизительно 20×20 метров.

Материалу земли тоже необходим доступ к размеру захвата для правильного проецирования целевого рендера. Проще всего это сделать, сохраняя размер захвата в *Material Parameter Collection*. По сути, это коллекция переменных, к которой может получить доступ *любой* материал.

Сохранение размера захвата

Вернитесь в основной редактор и перейдите в папку *Materials*.
Создайте *Material Parameter Collection*, которая будет находиться в *Materials & Textures*. Переименуйте её в *MPC_Capture* и откройте.

Затем создайте новый *Scalar Parameter* и назовите его *CaptureSize*. Не беспокойтесь о задании его значения — мы займёмся этим в блюпринтах.



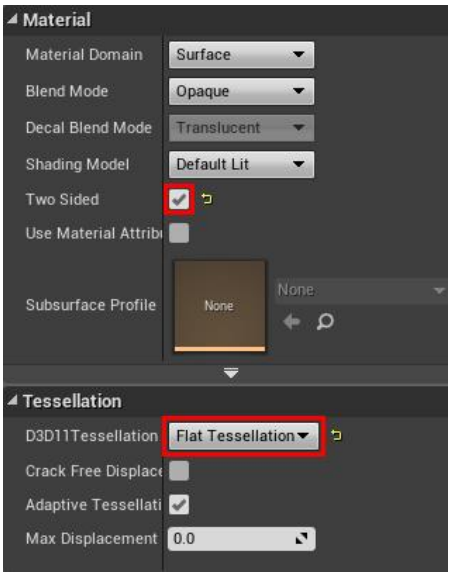
Вернитесь к *BP_Capture* и добавьте к *Event BeginPlay* выделенные ноды.
Выберите для *Collection* значение *MPC_Capture*, а для *Parameter Name* значение *CaptureSize*.

Теперь любой материал может получать значение *Ortho Width*, считывая его из параметра *CaptureSize*. Пока с захватом сцены мы закончили. Нажмите на *Compile* и вернитесь в основной редактор. Следующий шаг — проецирование целевого рендера на землю и использование его для деформации ландшафта.

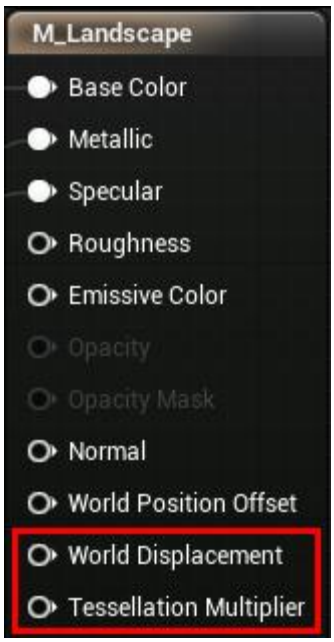
Деформация ландшафта

Откройте *M_Landscape* и перейдите в панель Details. Затем задайте следующие свойства:

- Для *Two Sided* выберите значение *enabled*. Так как захват сцены будет «смотреть» снизу, он будет видеть только обратные грани земли. По умолчанию движок не рендерит обратные грани мешей. Это значит, что он не будет сохранять глубину земли в буфер глубин. Чтобы исправить это, нам нужно сказать движку рендерить обе стороны меша.
- Для *D3D11 Tessellation* выберите значение *Flat Tessellation* (также можно использовать PN Triangles). Тесселляция разобьёт треугольники меша на более мелкие. По сути это увеличивает разрешение меша и позволяет нам получать более тонкие детали при смещении вершин. Без этого плотность вершин будет слишком мала для создания правдоподобных следов.



После включения тесселляции включатся *World Displacement* и *Tessellation Multiplier*.

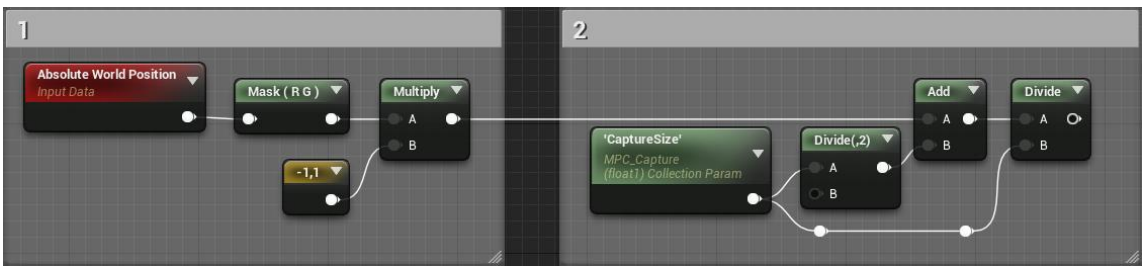


Tessellation Multipler управляет величиной тесселляции. В этом туториале мы не будет подключать этот нод, то есть используем значение по умолчанию (*1*).

World Displacement получает векторное значение, описывающее, в каком направлении и насколько перемещать вершину. Чтобы вычислить значение для этого контакта, нам нужно сначала спроецировать целевой рендер на землю.

Проецирование целевого рендера

Для проецирования целевого рендера необходимо вычислить его UV-координаты. Для этого нужно создать следующую схему:



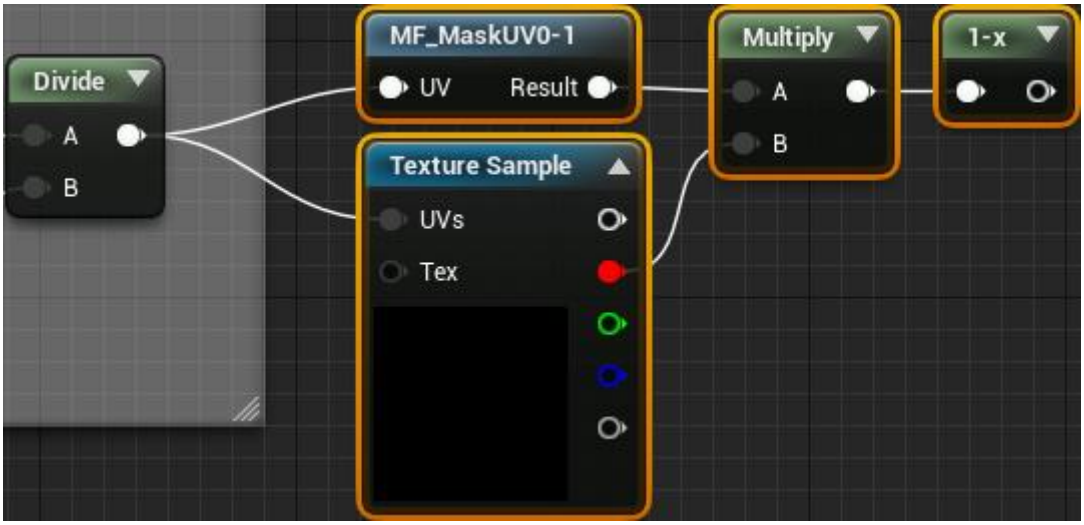
Что здесь происходит:

1. Сначала нам нужно получить позицию по XY текущей вершины. Так как мы выполняем захват снизу, координата X перевёрнута, поэтому

необходимо перевернуть её обратно (если бы мы выполняли захват сверху, нам бы это не понадобилось).

2. В этой части выполняются две задачи. Во-первых, она центрирует целевой рендер таким образом, чтобы его середина находилась в координатах *(0, 0)* мирового пространства. Затем она преобразует координаты из мирового пространства в UV-пространство.

Далее создадим выделенные ноды и соединим предыдущие расчёты так, как показано ниже. Для текстуры *Texture Sample* выберите значение *RT_Capture*.



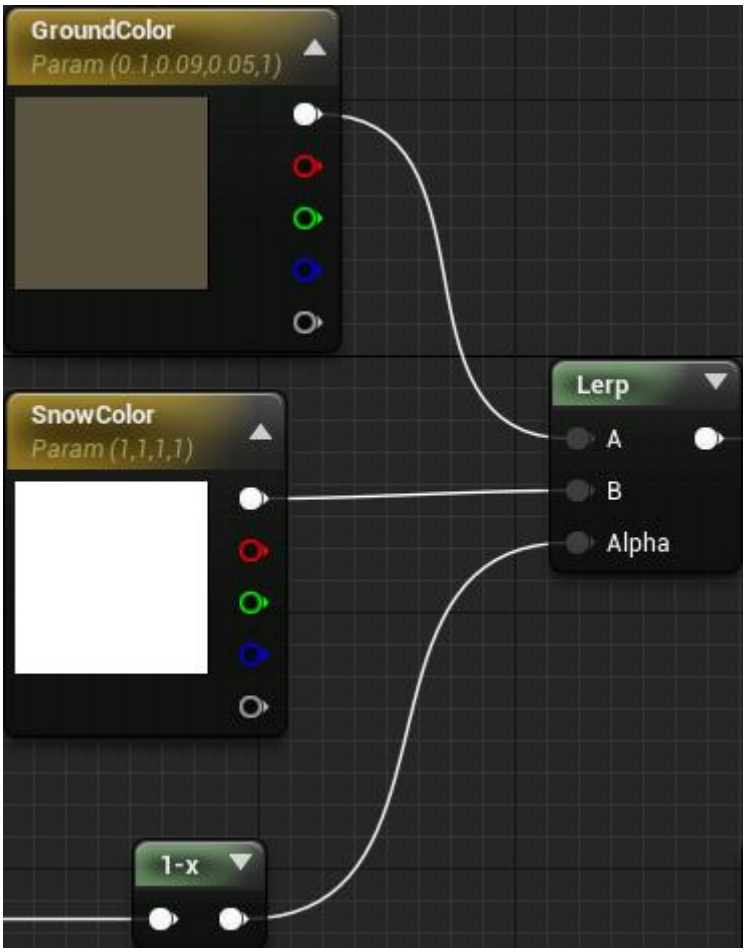
Это спроецирует целевой рендер на землю. Однако все вершины за пределами области захвата будут сэмплировать грани целевого рендера. На самом деле это проблема, потому что целевой рендер должен использоваться только для вершин внутри области захвата. Вот, как это выглядит в игре:

Чтобы исправить это, нам нужно замаскировать все UV, находящиеся за пределами интервала от 0 до 1 (то есть области захвата). Для этого я создал функцию *MF_MaskUV0-1*. Она возвращает *0*, если переданная UV находится за пределами интервала от 0 до 1 и возвращает *1*, если в его пределах. Умножая результат на целевой рендер, мы выполняем маскирование.

Теперь, когда мы спроецировали целевой рендер, можно использовать его для смешивания цветов и смещения вершин.

Использование целевого рендера

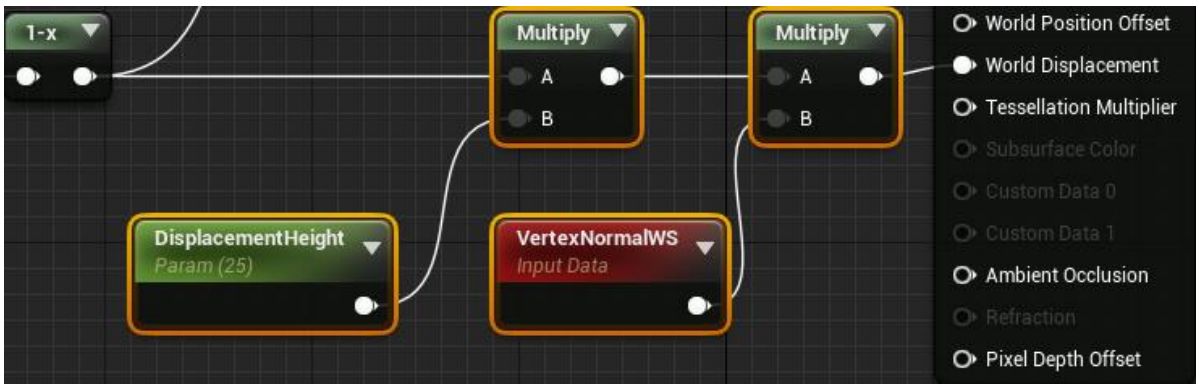
Давайте начнём со смешения цветов. Для этого мы просто соединим *1-x* с *Lerp*:



Примечание: если вы не понимаете, почему я использую *1-x*, объясню — это нужно для инвертирования целевого рендера, чтобы вычисления стали чуть проще.

Теперь, когда у нас есть след, цвет земли становится коричневым. Если цвета нет, он остаётся белым.

Следующий шаг — смещение вершин. Для этого добавим выделенные ноды и соединим всё следующим образом:



Это приведёт к тому, что все снежные области переместятся вверх на *25* единиц. Области без снега имеют нулевое смещение, благодаря чему будет создаваться след.

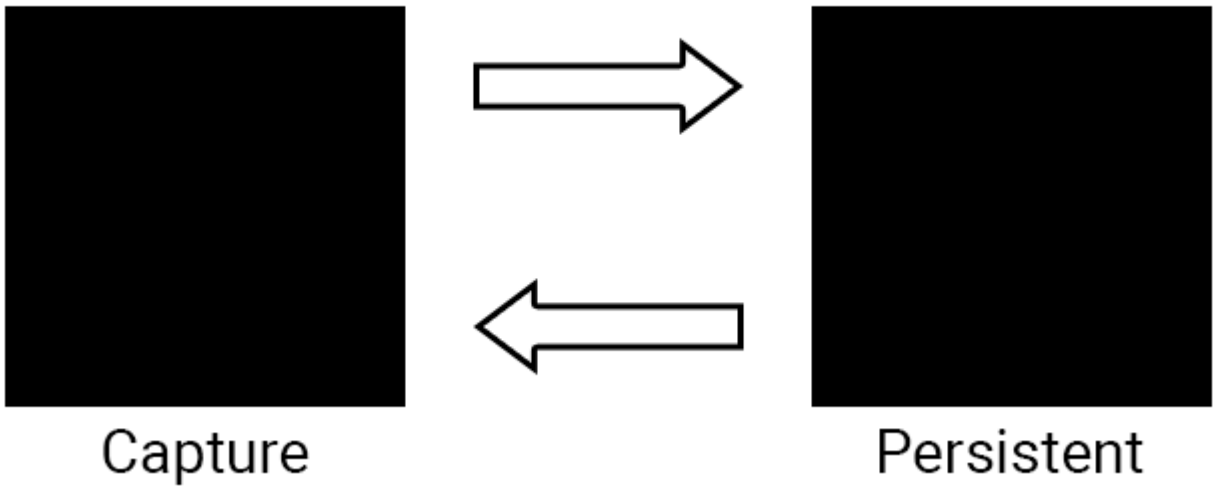
Примечание: можно менять *DisplacementHeight* для повышения или уменьшения уровня снега. Также заметьте, что DisplacementHeight — это то же значение, что и смещение захвата. Когда они имеют одинаковое значение, это даёт нам точную деформацию. Но существуют случаи, когда требуется менять их по отдельности, поэтому я оставил их отдельными параметрами.

Нажмите на *Apply* и вернитесь в основной редактор. Создайте на уровне экземпляр *BP_Capture* и задайте ему координаты *(0, 0, -2000)*, чтобы разместить его под землёй. Нажмите на *Play* и побродите вокруг с помощью клавиш *W*, *A*, *S* и *D*, чтобы деформировать снег.

Деформация работает, но следов не остаётся! Так получилось, потому что захват перезаписывает целевой рендер каждый раз при выполнении захвата. Нам нужен какой-то способ, чтобы сделать следы *постоянными*.

Создание постоянных следов

Для создания постоянства нам необходим ещё один целевой рендер (*постоянный буфер*), в котором будет сохраняться всё содержимое захвата перед перезаписью. Затем мы будем добавлять постоянный буфер к захвату (после его перезаписи). Мы получим цикл, в котором каждый целевой рендер выполняет запись в другой. Вот так мы создадим постоянство следов.

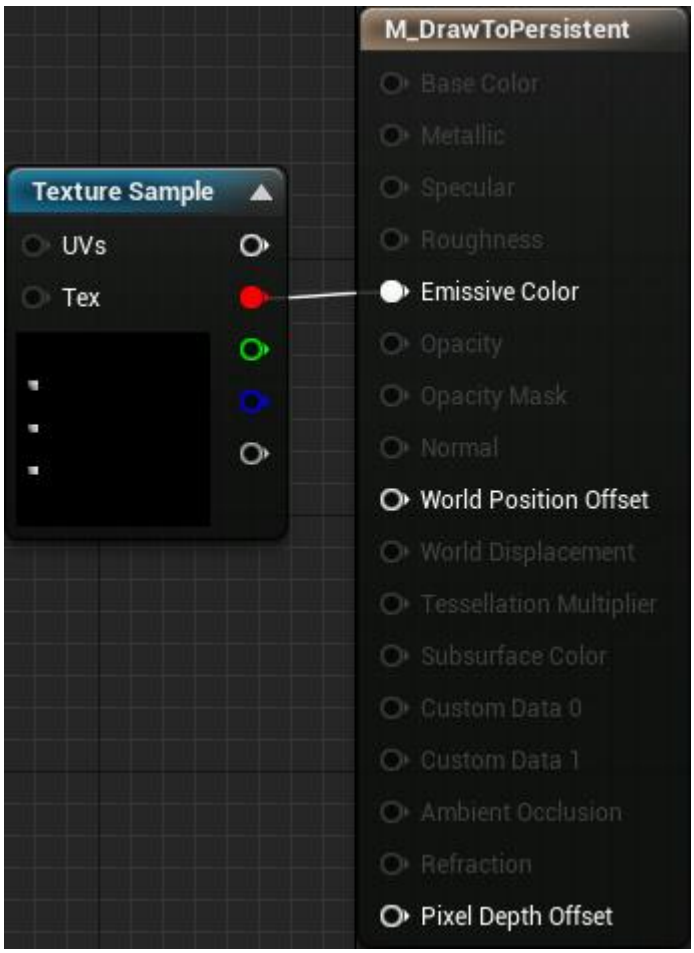


Во-первых, нам нужно создать постоянный буфер.

Создание постоянного буфера

Перейдите в папку *RenderTargets* и создайте новый *Render Target* под названием *RT_Persistent*. В этом туториале нам не придётся менять параметры текстур, но в собственном проекте вам необходимо будет убедиться, что оба целевых рендера используют одинаковое разрешение.

Далее нам необходим материал, который будет копировать захват в постоянный буфер. Откройте *Materials\M_DrawToPersistent* и добавьте нод *Texture Sample*. Выберите ему текстуру *RT_Capture* и соедините его следующим образом:



Теперь нам необходимо использовать материал отрисовки (draw material). Нажмите на *Apply*, а затем откройте *BP_Capture*. Сначала создадим динамический экземпляр материала (позже нам нужно будет передавать в него значения). Добавьте к *Event BeginPlay* выделенные узлы:

Ноды *Clear Render Target 2D* очищают перед использованием каждый целевой рендер.

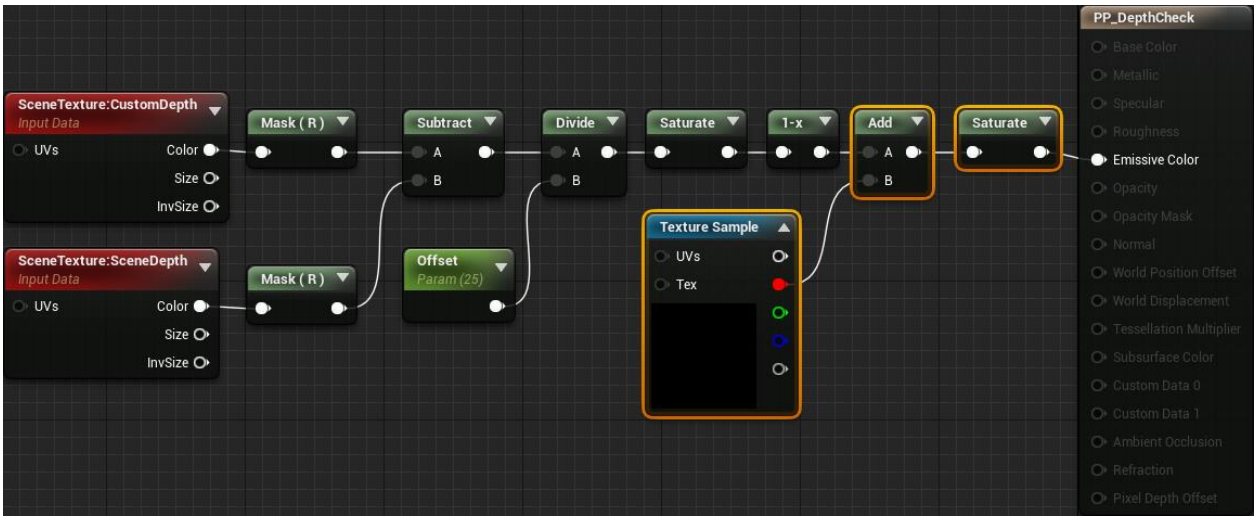
Затем откройте функцию *DrawToPersistent* и добавьте выделенные ноды:

Далее нам нужно сделать так, чтобы отрисовка в постоянный буфер выполнялась в каждом кадре, потому что захват происходит в каждом кадре. Для этого добавим *DrawToPersistent* к *Event Tick*.

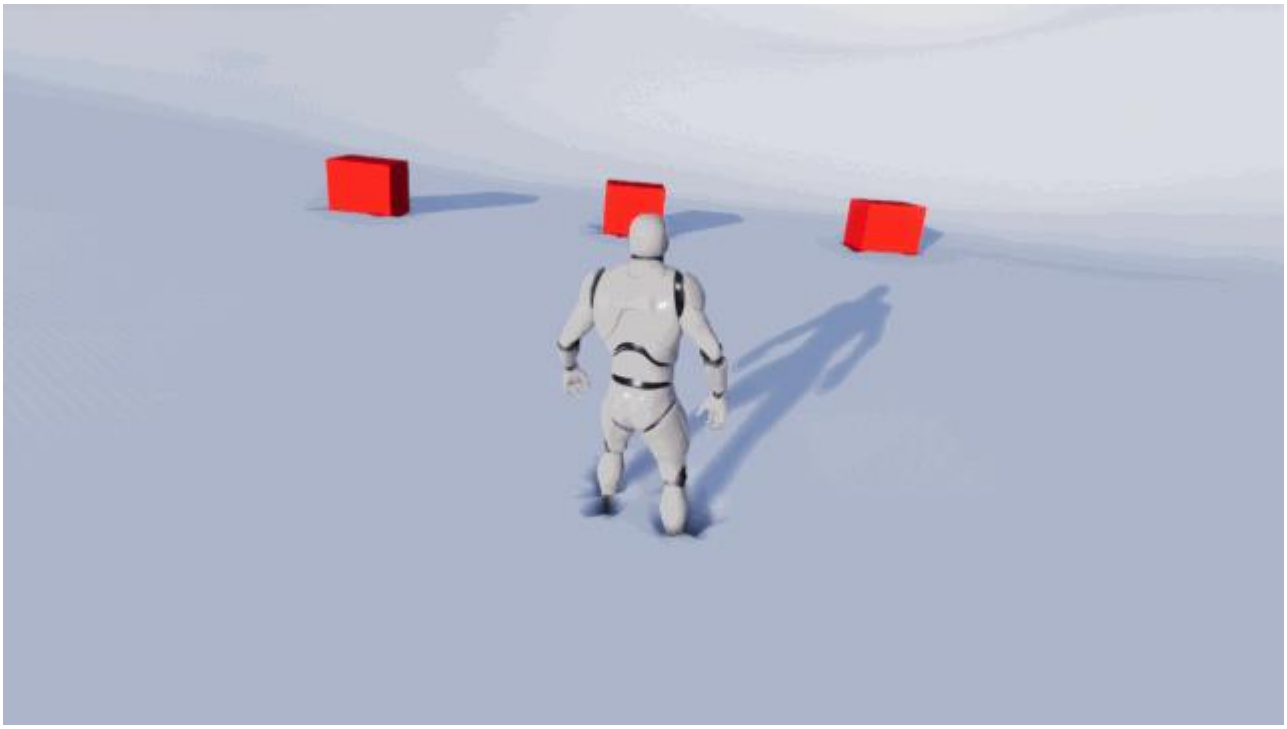
Наконец, нам нужно добавить постоянный буфер обратно в целевой рендер захвата.

Запись обратно в захват

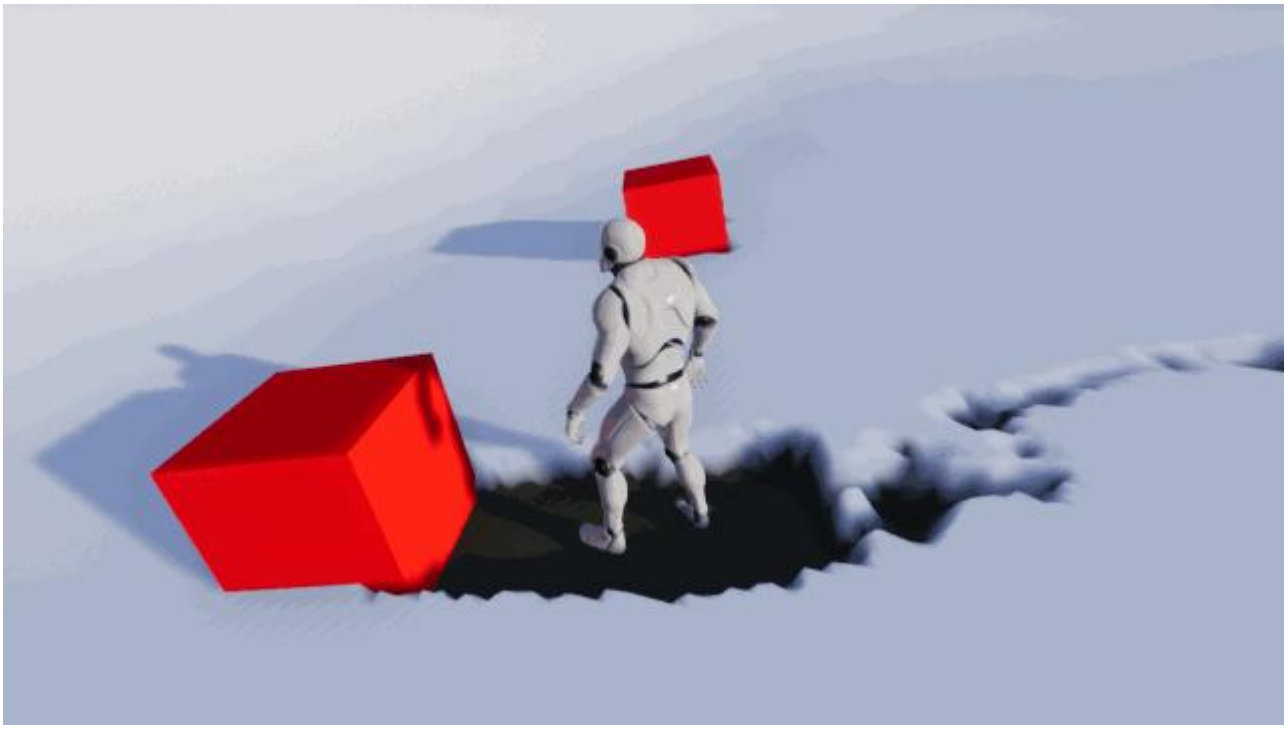
Нажмите на *Compile* и откройте *PP_DepthCheck*. Затем добавьте выделенные ноды. Для *Texture Sample* задайте значение *RT_Persistent*:



Теперь, когда целевые рендеры выполняют запись друг в друга, мы получим сохраняющиеся следы. Нажмите на *Apply*, а затем закройте материал. Нажмите на *Play* и начинайте оставлять следы!



Результат выглядит отлично, но получившаяся схема работает только для одной области карты. Если выйти за пределы области захвата, то следы перестанут появляться.



Можно решить эту проблему, перемещая область захвата *вместе* с игроком. Это означает, что следы всегда будут появляться вокруг области, в которой находится игрок.

Примечание: так как захват перемещается, вся информация за пределами области захвата убирается. Это значит, что если вернуться в область, где уже

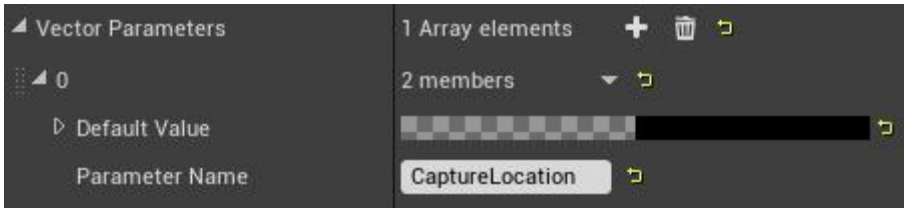
были следы, то они уже пропадут. В следующем туториале я расскажу, как создать частично сохраняющиеся следы.

Перемещение захвата

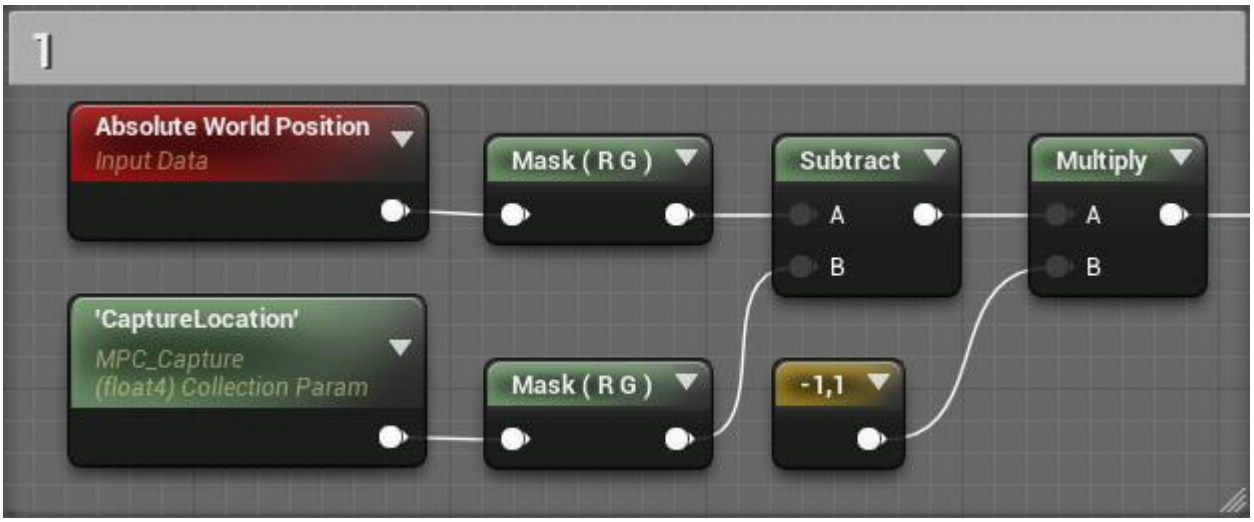
Можно решить, что достаточно просто привязать позицию захвата по XY к позиции игрока по XY. Но если так сделать, то целевой рендер начнёт размываться. Так происходит потому, что мы двигаем целевой рендер с шагом, который меньше пикселя. Когда такое случается, новая позиция пикселя оказывается *между* пикселями. В результате один пиксель интерполируются несколько пикселей. Вот как это выглядит:

Чтобы устранить эту проблему, нам нужно двигать захват дискретными шагами. Мы вычислим *размер пикселя в мире*, а затем переместим захват на шаги, равные этому размеру. Тогда каждый пиксель никогда не окажется между другими, поэтому размытие не будет появляться.

Для начала давайте создадим параметр, в котором будет храниться местоположение захвата. Он понадобится материалу земли для выполнения вычислений проецирования. Откройте *MPC_Capture* и добавьте *Vector Parameter* под названием *CaptureLocation*.



Далее необходимо обновить материал земли, чтобы использовать новый параметр. Закройте *MPC_Capture* и откройте *M_Landscape*. Измените первую часть вычислений проецирования следующим образом:



Теперь целевой рендер всегда будет проецироваться на местоположение захвата. Нажмите на *Apply* и закройте материал.

Далее мы сделаем так, чтобы захват перемещался с дискретным шагом.

Перемещение захвата с дискретным шагом

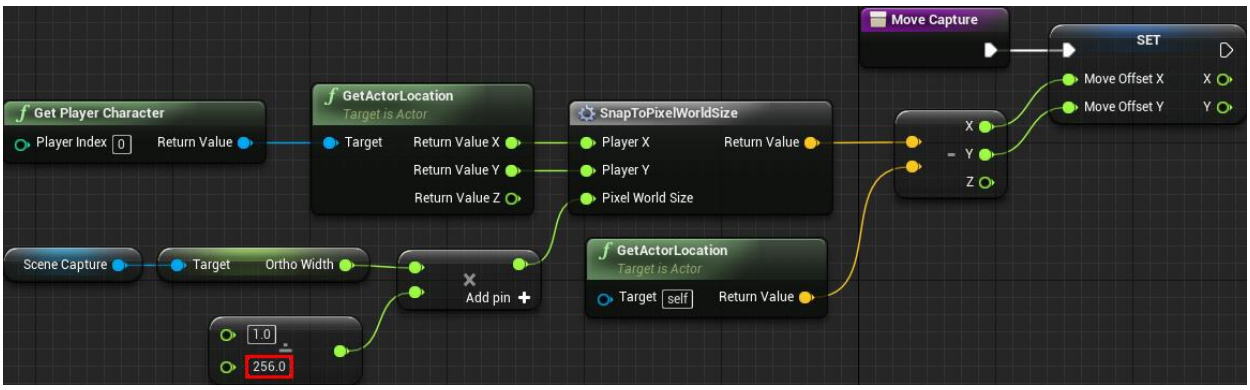
Для вычисления размера пикселя в мире можно использовать следующее уравнение:

```
(1 / RenderTargetResolution) * CaptureSize
```

Для вычисления новой позиции мы используем показанное ниже уравнение для каждого компонента позиции (в нашем случае — для координат X и Y).

```
(floor(Position / PixelWorldSize) + 0.5) * PixelWorldSize
```

Теперь используем их в блюпринте захвата. Чтобы сэкономить время, я создал для второго уравнения макрос *SnapToPixelWorldSize*.
Откройте *BP_Capture*, а затем откройте функцию *MoveCapture*. Далее создайте следующую схему:

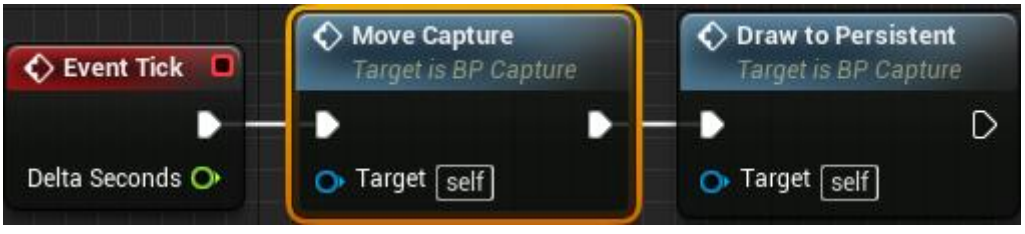


Она будет вычислять новое местоположение, а затем сохранять разницу между новым и текущим местоположением в *MoveOffset*. Если вы используете разрешение, отличающееся от 256×256, то измените выделенное значение.

Далее добавим выделенные ноды:

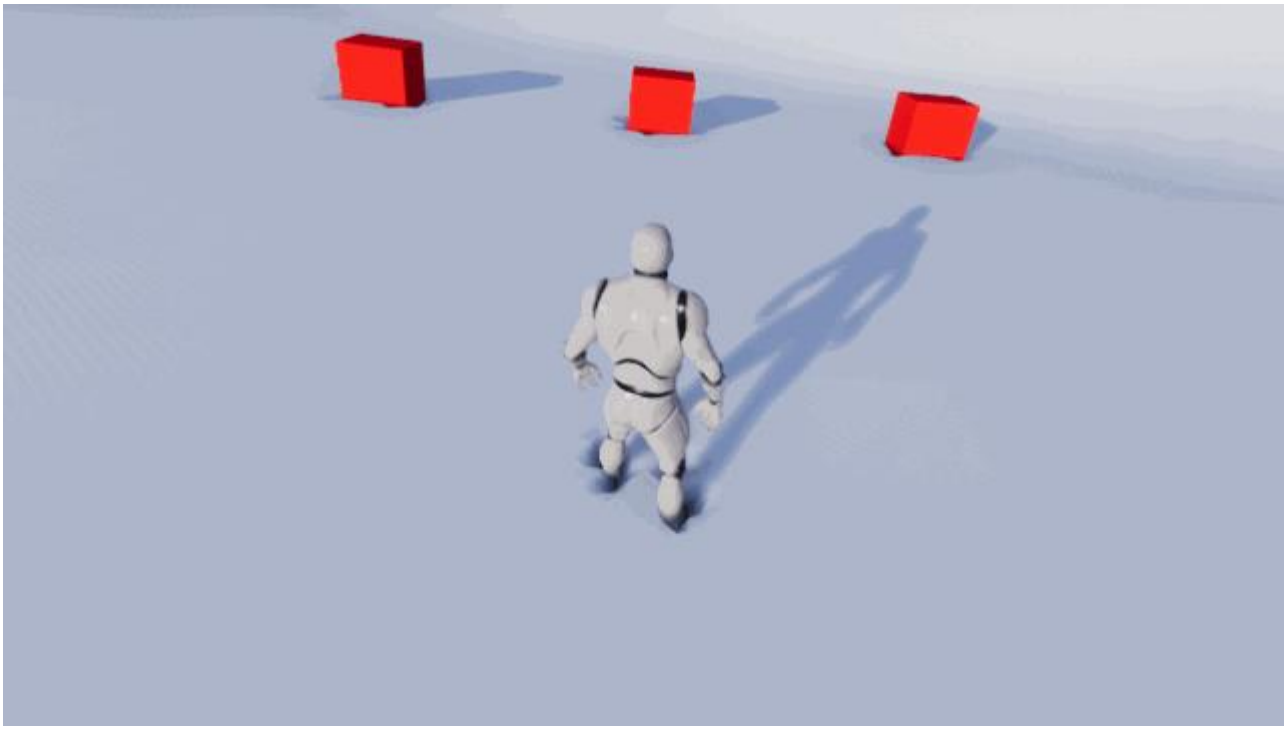
Эта схема будет перемещать захват с вычисленным смещением. Затем она будет сохранять новое местоположение захвата в *MPC_Capture*, чтобы его мог использовать материал земли.

Наконец, нам нужно выполнять в каждом кадре обновление позиции. Закройте функцию и добавьте в *Event Tick* перед *DrawToPersistent MoveCapture*.



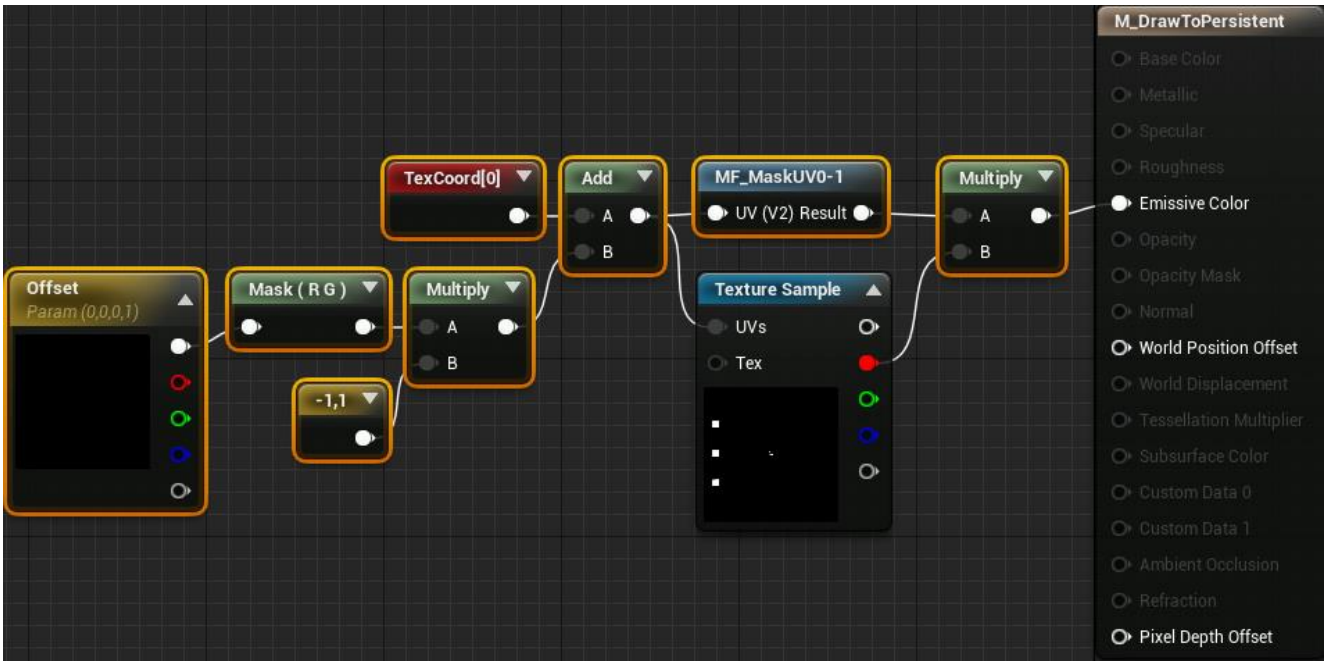
Перемещение захвата — это только половина решения. Также нам нужно

перемещать постоянный буфер. В противном случае захват и постоянный буфер рассинхронизируются и будут создавать странные результаты.



Перемещение постоянного буфера

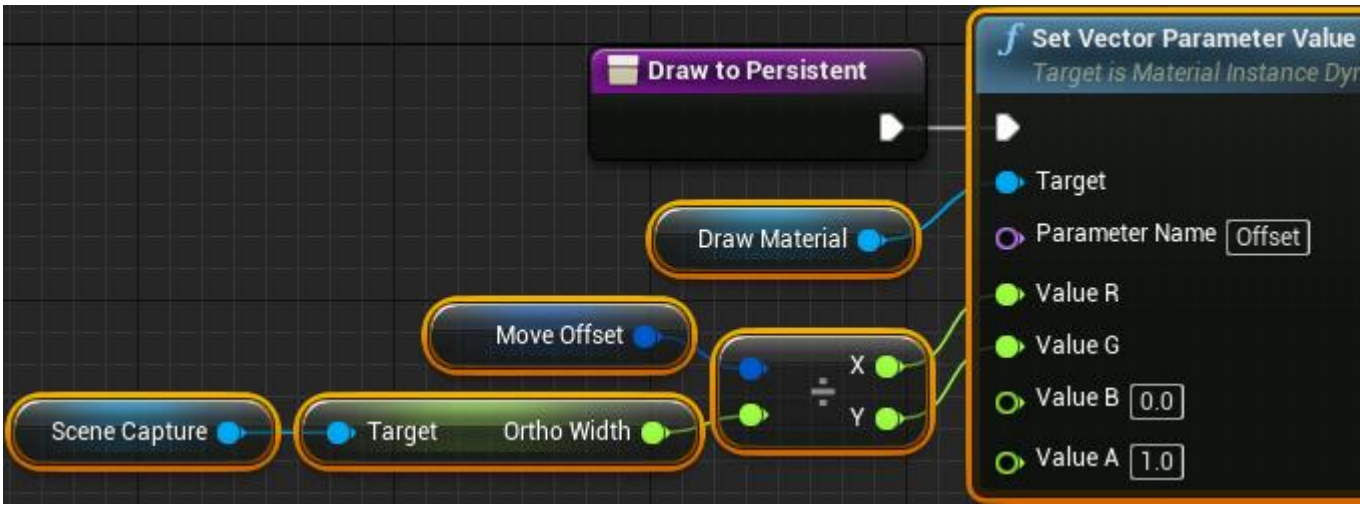
Для сдвига постоянного буфера нам нужно передавать вычисленное смещение перемещения. Откройте *M_DrawToPersistent* и добавьте выделенные ноды:



Благодаря этому постоянный буфер будет сдвигаться на величину переданного смещения. Как и в материале земли, нам необходимо

переворачивать координату X и выполнять маскировку. Нажмите на *Apply* и закройте материал.

Затем необходимо передать смещение. Откройте *BP_Capture*, а затем откройте функцию *DrawToPersistent*. Далее добавьте выделенные ноды:



Так мы преобразуем *MoveOffset* в UV-пространство, а затем передаём его в материал отрисовки.

Нажмите на *Compile*, а затем закройте блюпринт. Нажмите на *Play* и побегайте всласть! Как бы далеко вы ни убежали, вокруг вас всегда будут оставаться следы.