

ЭТОТ СЛИВ – НЕ КОНЕЦ!

Гайд по созданию ассетов для S.T.A.L.K.E.R.2.

Environmental storytelling

Суть:

Задача окружения заключается в том, чтобы рассказать игроку историю.

По сути, отдельный объект в окружении — это лишь маленькая часть большой картины. Человек привык смотреть на все сразу, воспринимая картину в целом, даже не подозревая о том, что все вокруг состоит из маленьких историй незначительных предметов. Но понимание данного вопроса, в конечном итоге, и делает окружение осмысленным и убедительным.

Никакой объект не существует отдельно от мира, он всегда выполняет какую-то роль. Хороший художник по окружению всегда знает? как правильно расставить акценты, чтобы объект выглядел максимально убедительно. Убедительность ассета — это ни в коем случае не *наляпистость*, и не умение как-то мазать кистью под определенным углом, а вполне конкретный навык, который состоит из понимания исходного материала, с последующим анализом и переосмыслением. Эти знания художник не изобретает самостоятельно и не опирается полностью на свою “технику”, он руководствуется несколькими простыми правилами:

1. Назначение объекта;

То, для чего был создан объект, зачастую и определяет его дальнейшую эксплуатацию.

Пример: Бита сделана для игры в бейсбол. Если на ней осколки черепа, это уже см. пункт 3.

2. Путь эксплуатации объекта (что происходило с объектом);

Через что объект прошел, прежде чем попасть к нам в “кадр”? Каков его путь и кем он использовался?

Пример: обычный холодильник “Днепр”, 1957 года выпуска. До трагедии в 1986 году он как-то эксплуатировался купившими его людьми (одним человеком, семьей, государством?). Соответственно, кроме ржавчины, на нем останутся отметины, встречающиеся в обычном быту.

3. Обстоятельства, в которых находится объект;

Где и как объект использовался? Где он находился с момента своего создания? Обстоятельства иногда могут даже перевернуть назначение самого объекта.

Пример: А вы знали что ломик Гордона Фримена был создан для аварийного открывания люков?

Environment artist — это наблюдательный человек, способный анализировать то что он видит и переносить это в виртуальный мир. Бездумное повторение даже исчерпывающего референса никогда не даст удовлетворительного результата без осмысления исходного материала. Все вмятины, любая информация в текстуре, говорящая об эксплуатации или старении объекта, должна учитывать, что с объектом происходило. Насколько интенсивно он использовался, либо же наоборот — сколько времени лежал без дела.

Гайд художнику по созданию убедительного ассета

Итак, что же нам нужно уяснить вне зависимости от конкретных объектов, то есть, принять за правило? Нужно понять пару несложных вещей:

Мир не идеален.

Найти что-либо идеально гладкое, идеально круглое или идеально прямое будет крайне тяжело в реальной жизни.

Нужно принять тот факт, что идеальных прямых линий в реальности не существует. Да, реальность сурова. В ней все несовершенно и особенно это касается предметов и вещей не первой свежести. Когда мы работаем с объектами после эксплуатации, они теряют товарный вид и практически никогда не бывают идеально ровными или идеально чистыми. Это стоит помнить.

Что же для этого нужно делать? Конечно, для этого нужна модель под сабдив. Практически все модели, которые мы хотим визуально повредить, должны иметь топологию под сабдив для дальнейшей ручной доводки их в **ZBrush**. на примере ниже видно как выглядит модель с просто запеченными фасками и как выглядит та же самая модель после **ZBrush**.



Вмятины, проставленные не рандомно, а в местах, где им положено быть, делают ассет реалистичнее.

Например, вот эта вмятина появилась на плите в результате многократного неаккуратного открывания двери:



Вот сюда кто-то, вероятно, ударил ребром гладильной доски:



А это место было повреждено во время переноса плиты:



И так далее.

Также важный момент, что тип вмятины может быть гипертрофирован в ту или иную сторону: быть более явным, или более слабым относительно референса. Также зачастую не имеет смысла полностью повторять то, что есть на картинке, а можно использовать собственное видение объекта для создания смыслов.

Например, у нас есть такой референс:



Как мы можем прочитать данный референс, чтобы сделать убедительный ассет? Допустим, у нас нет других референсов и мы даже не знаем, что это за модель плиты (это и не важно, на самом деле). Чтобы правильно подойти к процессу, достаточно просто разложить то, что мы видим, на этапы.

Итак:

1. Это плита (кэп);
2. Она стоит в комнате уже не первый день (не первый десяток лет, на самом деле);
3. До того как плита стала покинутой, ей кто-то пользовался (скорее всего, обычный быт).
4. Плита вырвана с места, где она стояла. Мы можем придумать, что это сделали мародеры, чтобы временно спрятать в ней свои пожитки.

Все эти этапы мы должны внести в ассет. Для начала нам нужно понять основной конструктив ассета, но это очевидный факт. Далее мы смотрим, как как она старела. Логика здесь следующая: повреждения от бытового использования → запустение → удары от перемещений и каких-то действий мародеров → снова длительное покинутое состояние.

На выходе у нас получается следующий ассет (без декалей паутины, без мусора и т.п.):



На данном примере видно, что некоторые детали не соответствуют на 100% оригиналу, а текстуры и вовсе переосмыслены.



Но идея и ощущения от предмета сохранены, а где-то даже немного усилены за счет более выраженных повреждений.



Все материалы должны читаться хорошо

При создании такого ассета самой важной будет последовательность действий. Например, не стоит стараться сразу сделать побитый и грязный объект, налепить побольше материалов, если какой-либо отдельный материал не понятен до конца (например, как выглядит заводская краска в чистом виде).

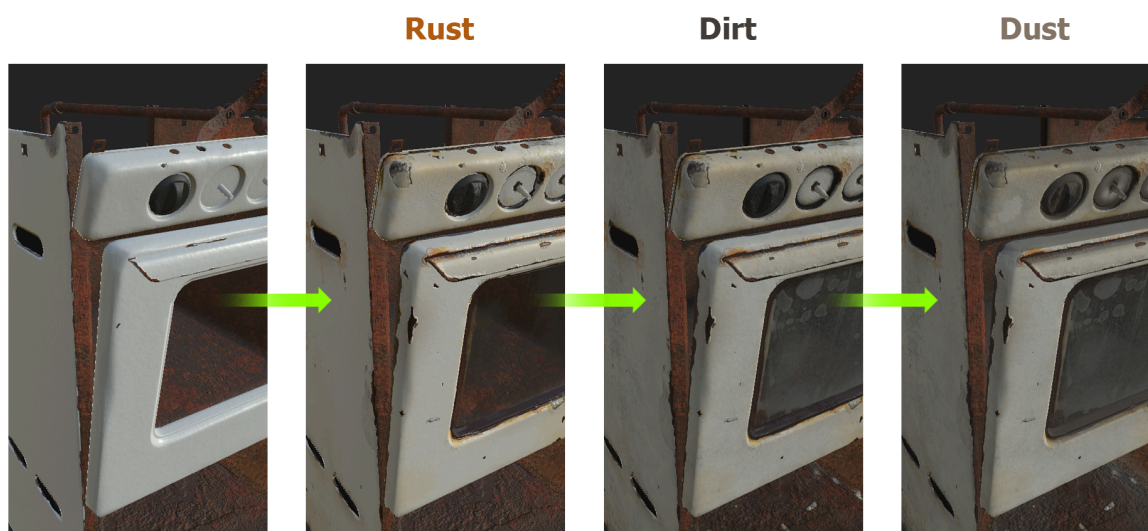
Иногда нужно докрутить gloss



или, например, важным моментом какой-то краски может быть неровный noise.



Подобные вещи никогда не должны быть упущены или заглушены слоями других материалов сверху. Хороший подход — это всегда делать сначала чистый объект, а потом послойно накладывать на него другие материалы: ржавчину, грязь, пыль.



Причем, каждый материал должен читаться и иметь уникальные свойства.

Немного о художественном подходе

Очень важна ручная выделка. Дело в том, что процедурность модели зачастую выдает ее искусственное происхождение из-за излишней равномерности, либо повторяемости. Задача же художника — создать убедительный ассет, который будет работать на реалистичность мира. А не прочитываться игроком, как некая “бутафория”, призванная заполнить пустое пространство на уровне.



Такого происходить не должно. Именно поэтому мы стараемся сделать каждый ассет максимально уникальным и использовать процедурные карты с умом. Если мы делаем ржавчину, то думаем, как она будет располагаться, в каких местах она будет темнее, а в каких — светлее. Если мы кладем грязь, то думаем, как она туда попала, а не просто бездумно пользуемся mask generator.

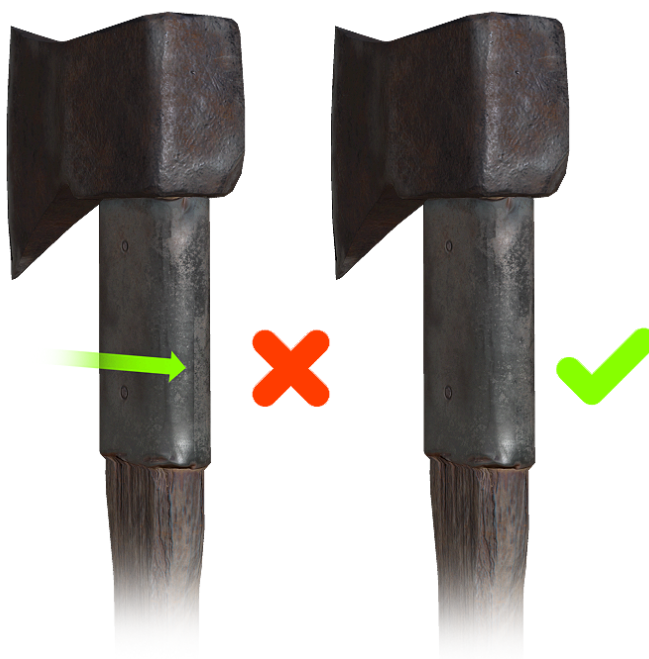
Чтобы лучше видеть косяки в материале, есть смысл смотреть на объект издалека. Это позволяет нам воспринять его более цельно. Например, тут становится видно, что неплохо выглядящая вблизи ржавчина издалека кажется очень равномерной и выглядит процедурно. Это плохо.



Швы на текстурах

Часто бывает так что работая с текстурами, мы различные моменты решаем процедурно.

Подобных вещей следует избегать. Поэтому если вы видите что где-то образовался такой шов после наложения процедурной маски, попробуйте сначала изменить тип наложения с **UV** на **Triplanar**. Если это не решает проблему, создайте **Paint** в слое с проблемой и зарисуйте эти области руками.



Подобные неаккуратности сильно выбивают игрока из общей атмосферы, особенно когда располагаются на видимых сторонах объектов.



Также швы могут сильно портить внешний вид объекта, после чего он теряет целостность.

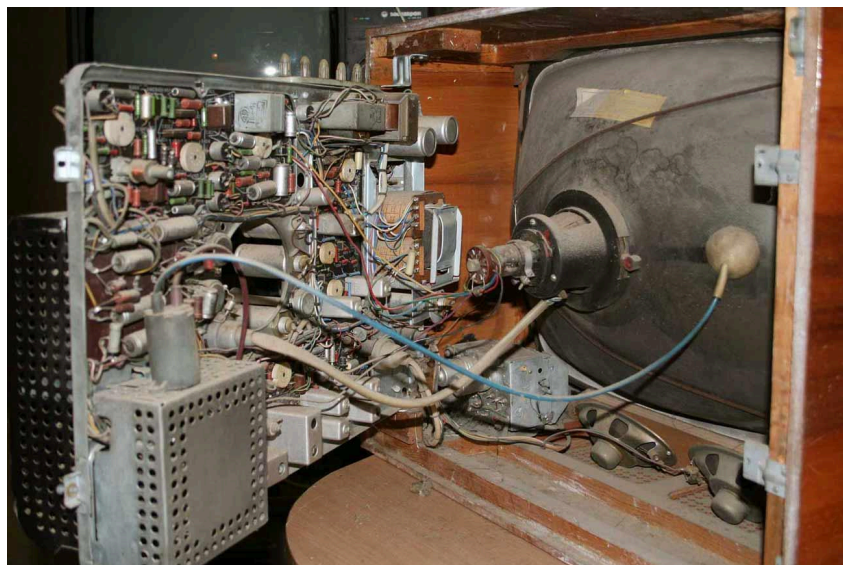
Работа с референсами

Референсы - чуть ли не самое главное в нашей работе. Важно иметь как можно больше информации о том, что мы делаем.



К сожалению, для многих данный момент не понятен. Зачастую нужно самостоятельно искать дополнительную информацию по объекту, который требуется создать.

Конструкция



Важно полностью представлять тот объект, который вы делаете, даже если ваша задача создать его упрощенным. Ищите максимальное количество информации о том, что необходимо сделать. Помните что понимание конструкции всегда позволяет сделать ассет правильным, не стоит пренебрегать дополнительным поиском и всецело полагаться на свое воображение.

Важным моментом также является функционал объекта. Даже если мы упрощаем какие-то элементы, нам значительно легче будет это делать, если мы знаем что именно мы упрощаем.

Пример: для того, чтобы создать упрощенный кинескоп телевизора с обратной стороны, сначала нужно понять как он устроен.



Материалы

Не обязательно у вас должен быть один референс на объект и на его материал, чаще это будет отдельный реф материала на совершенно другом объекте. Имейте это в виду! Большая редкость когда вы располагаете исчерпывающей информацией сразу и на примере одного объекта!

Как работать с материалами

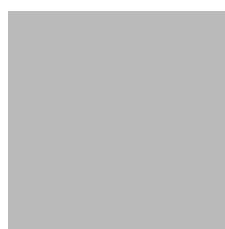
Чисто белые и чисто черные цвета в природе не существуют. В игре чистые цвета также будут интерпретироваться некорректно, белый станет неестественным засветом, а черный превратиться в черную дыру. Об этом следует помнить. Поэтому мы используем фиксированные значения, за рамки которых не выходим:



Об этом следует помнить всегда.

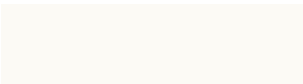
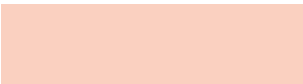
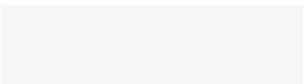
Base color

Нейтральный серый:



186

Цвета для разных материалов:

		
Золото	Железо	Никель
		
Серебро	Медь	Кобальт
		
Алюминий	Титан	Платина

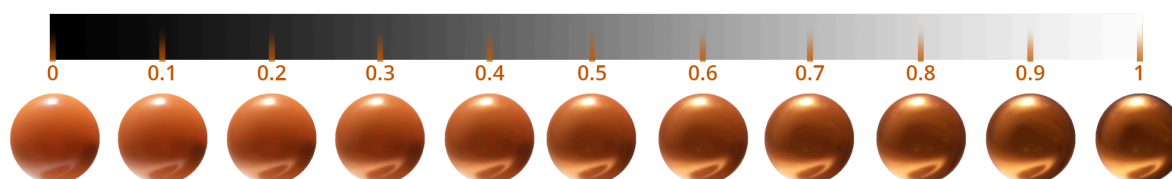
Также может помочь:

	50 уголь
	57 кованое железо
	67 темная земля
	92 изношенный асфальт
	181 чистый цемент
	129 серый пластик
	135 старый бетон

Вышеприведенные цвета — лишь ориентир, но их желательно использовать.

Metallic

Карта **metallic** применяется только на металлические материалы. Белый — металл, черный — не металл.

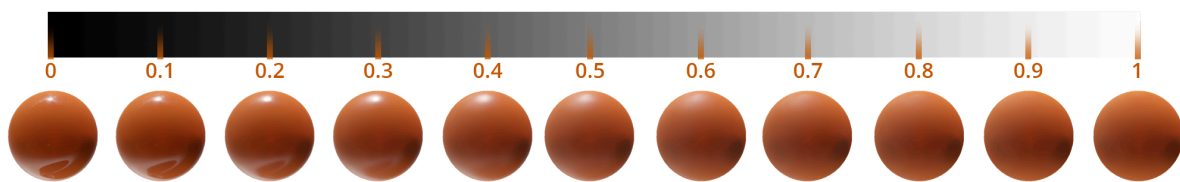


Metalness у материалов, которые не являются металлами, должна быть полностью черной.

Существует распространенная ошибка подкручивать **metalness** для того чтобы добиться красивых бликов, но так делать категорически нельзя. Металлом должен быть только металл.

Roughness

Roughness мы используем для создания глянцевости. Ни в коем случае не компенсируем глянцевость картой **metallic**, это очень грубая ошибка.



Гайд по структуре сцены в Substance Painter:

Для того чтобы правильно собрать материалы, сначала нужно понять как все это будет работать в движке. Для большинства ассетов мы используем систему многослойных шейдеров, что позволяет нам красиво смешивать материалы в движке. Это значит что из сабстенса мы получаем только маски. В основном это касается средних и больших объектов, разрешения текстур которых объективно не хватает для достижения оптимального качества картинки. Но даже если объект относительно маленький и мы делаем его со всеми картами, структура создания материалов остается одной.

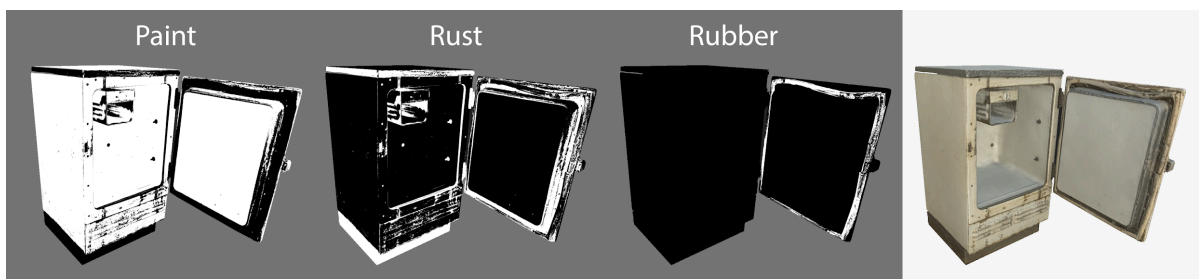
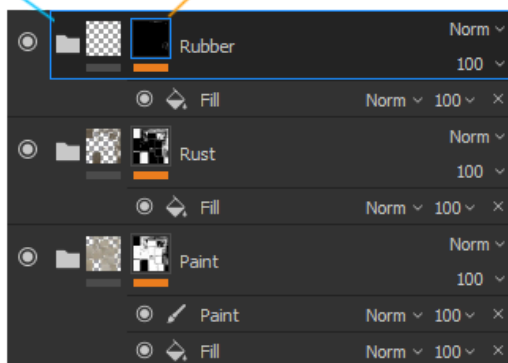
Итак, максимальное количество материалов на одном объекте - 5. Но желательно делать до 4-х. В идеале должно быть 3 материала. Похожие по типу материалы можно просто группировать (например, разные типы грязи можно сделать одной папкой).

У нас есть папка, для которой мы создаем **blackmask**, в которой уже создаем **fill**, **paint** и прочие корректировочные слои.

Выглядят это так:

folder

black mask



Внутри каждой папки может быть группа материалов.

1. Хаос в сцене — это плохо. Все материалы следует структурировать согласно логике их расположения на объекте. Не забываем что папок с масками, то есть, групп материалов, у нас может быть до 5.

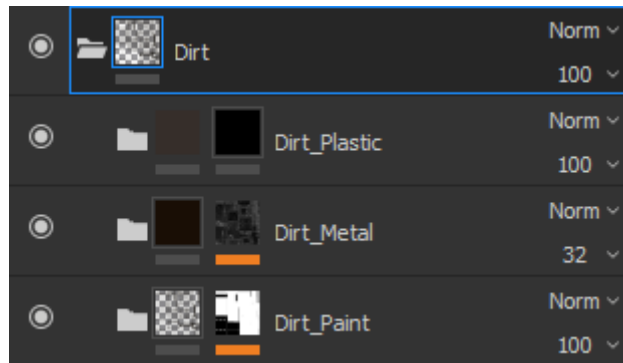
- 5. Dust
- 4. Dirt
- 3. Rust
- 2. Plastic
- 1. Paint

2. Все материалы похожего типа следует группировать по папкам с соответствующим названием. Это значительно упрощает работу.



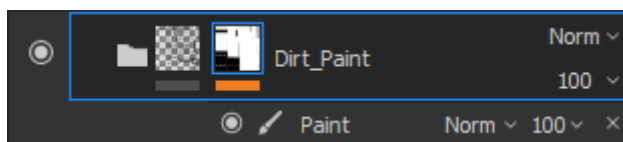
Hint: все базовые материалы разного типа, например: краска, пластик, дерево и т.п., можно группировать в одну папку и делать на них общую маску. Это облегчит структуру сцены.

3. В каждой папке должны находиться материалы для разных групп, либо разные типы материалов этой категории.



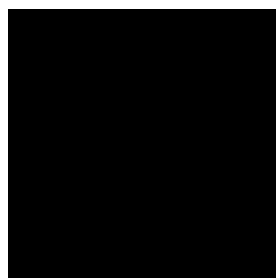
Пример: в папке Dirt может быть много типов грязи, то есть, много папок Dirt с разными материалами грязи, включая разную грязь для отдельных типов материалов со своими масками.

4. Маски мы всегда делаем для папок, а не для отдельных слоев. Информацию заносим в них не прямо в маску, а в **paint**.



Alpha и ее правильное применение

Абсолютное значение цвета мы все-таки используем. Но только для того, чтобы пробивать им дырки. Логика такая: в местах, где должны быть дырки, мы не делаем отдельную альфу, а просто на карте albedo рисуем чисто черным цветом (0,0,0). Поскольку у нас нигде не используется чисто черный, то мы в итоге движком отрезаем эти области через **cutout**.

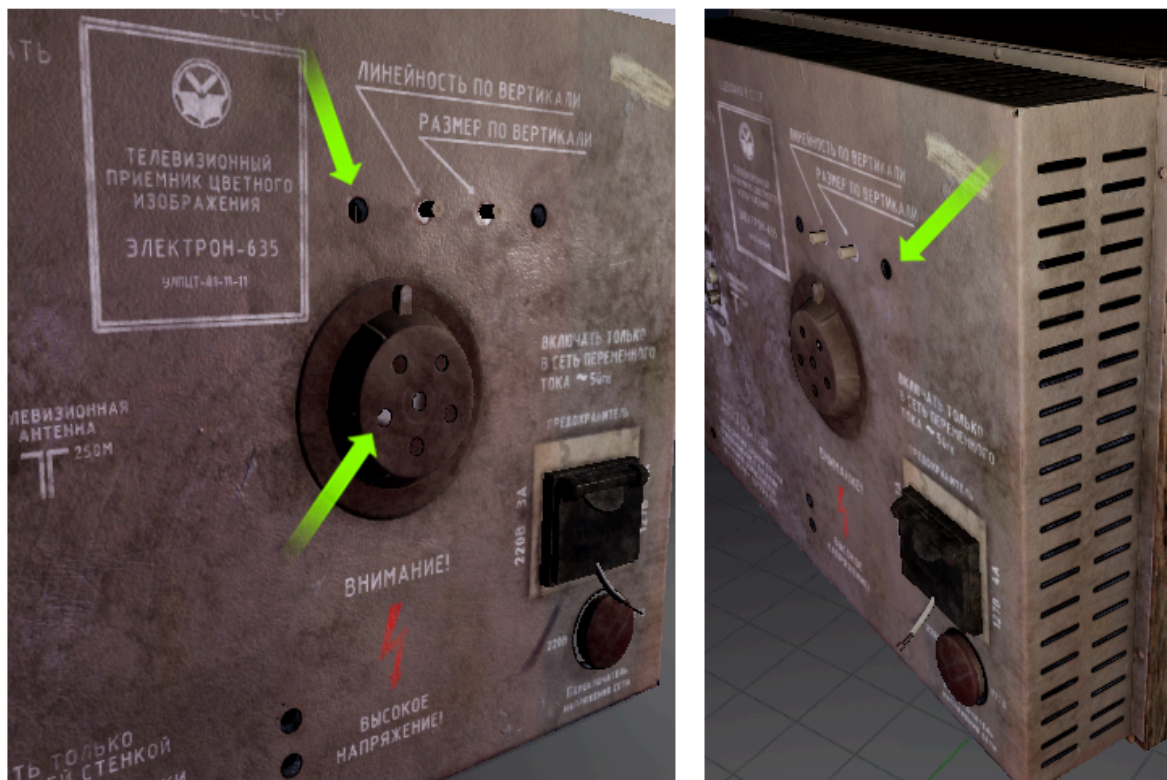


0,0,0

Чтобы альфа была максимально аккуратной, нужно следить за пикселями. Желательно закрашивать только ту область, которая идет с отверстием и больше НИГДЕ(!) на ассете не использовать чисто черный цвет.

При аккуратной работе с альфами, мы можем делать так простые небольшие отверстия, что позволяет увеличивать детализацию объектов за счет карт и не делать отдельных альф для простых ситуаций.

Пример:



А вот в таких ситуациях мы уже используем отдельную альфу:



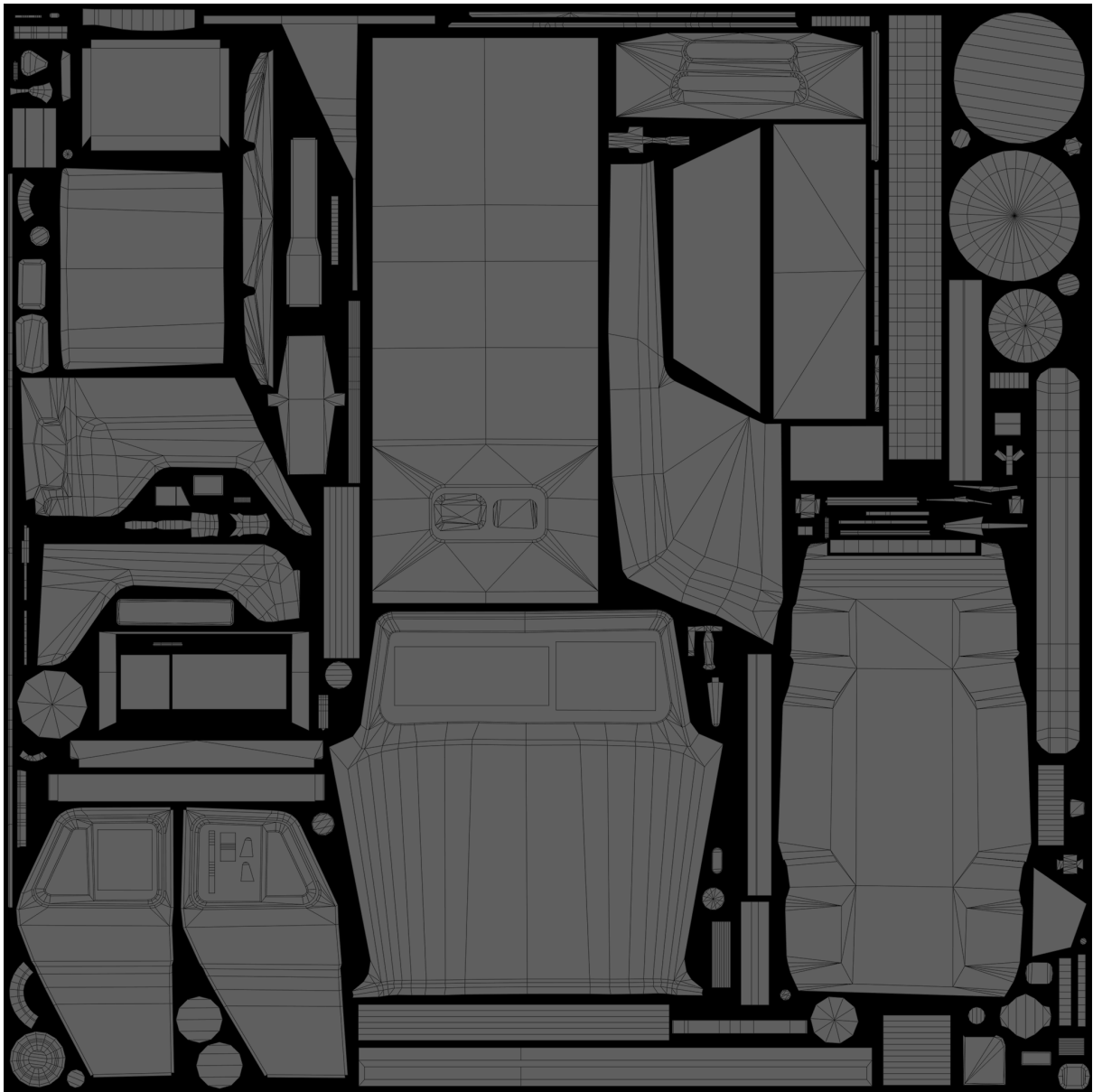
Работа с UV

Аккуратная развертка — залог качественной карты **normal** и хороших текстур. В создании UV есть несколько важных правил.

Когда мы создаем развертку, очень важно не только правильно подготовить геометрию, разрезать и развернуть “ювишки”, но и выполнить пару несложных действий.

1. Layout UV

Раскладывать UV лучше всего руками. Только так мы можем самым лучшим образом организовать пространство. Адекватно разложенная UV, где учтены все пункты данного параграфа, выглядит как-то так:



2. Тексель и правила хорошего тона

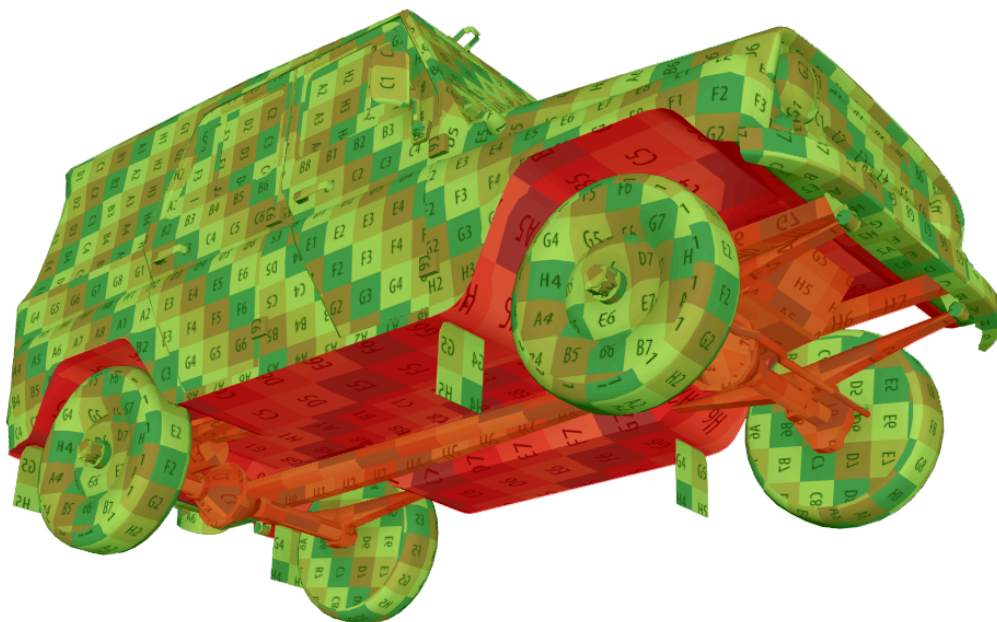
На тексель влияет качественная работа с **UV**. Это значит, что на данном этапе мы сталкиваемся с необходимостью рациональной организации места. Чтобы это сделать, необходимо подойти к процессу развертки с умом и максимально использовать все **UV** пространство.

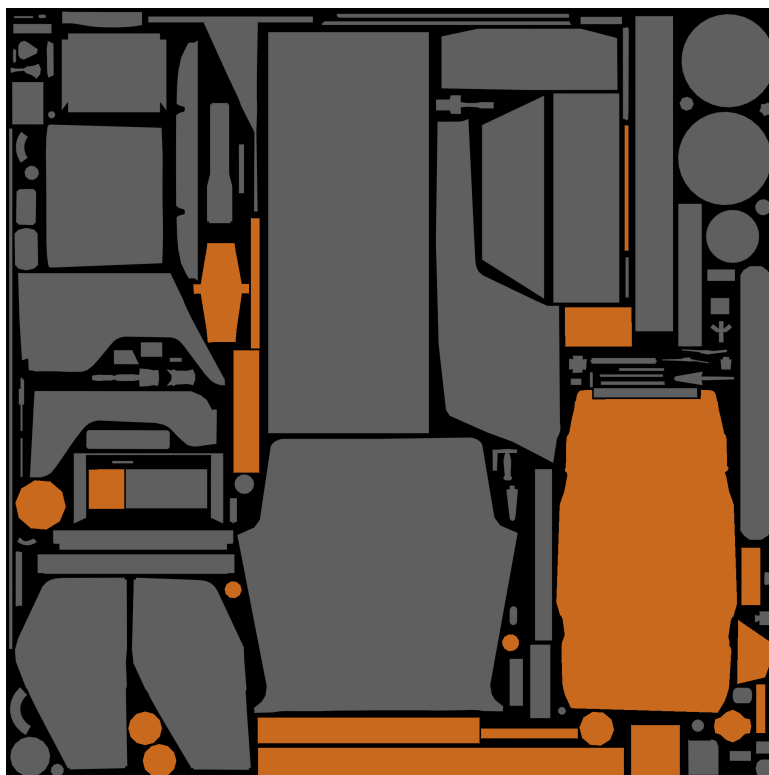
Для начала нам необходимо развернуть всю модель и понять какими UV-шеллами мы располагаем. Потом уже понять какие части модели являются приоритетными для игрока и будут его точкой внимания, а какие — нет (например детали, которые будут видны лишь частично). Таким образом, имея понимание, как модель будет восприниматься игроком, мы можем значительно увеличить общее разрешение текселя, не меняя разрешение самой карты.

Пример: допустим, мы имеем автомобиль, который переворачивать в игре не планируется. Поэтому все элементы, включая днище, которые нам сильно видны не будут, следует отскейлить, чтобы дать больше пространства важным деталям.



На развертке днище будет иметь меньший тексель, чем другие детали:

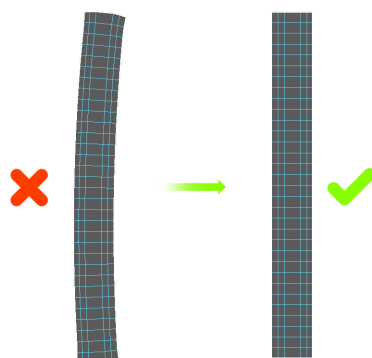




3. Антиалиайзинг, швы и проблемы с местом

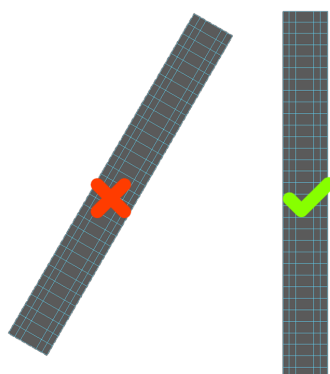
На финальное качество модели в игре влияет несколько факторов, которые необходимо учитывать. Это, в основном, касается генерации мипмап. Многие вещи, которые выглядят хорошо в большом разрешении, могут выглядеть плохо на маленьком, после автоматической генерации мипмап движком. Нам нужно учесть это, чтобы избежать проблем на мипмапах в дальнейшем:

1. Кривые шеллы нужно выравнивать. Держать их в кривом виде плохо как для запекания швов, так и для рационализации пространства. Антиалиайзинг на кривых шеллах в местах швов будет делать нам “лесенки” из пикселей, если мы их не выпрямим. Это особенно ощутимо на картах с маленьким разрешением.



2. Не стоит бросать шеллы на UV-пространстве как попало. Их следует класть так, чтобы прямые линии были прямыми (при скалировании и генерации

мипмап прямая превращается в красивый ряд пикселей, в то время, как кривая становится рубленой и в месте шва начинаются проблемы, как и в пункте 1)



3. Нужно постоянно следить за тем, чтобы UV не были кривыми. Чекер на модели должен выдавать ровные квадраты.

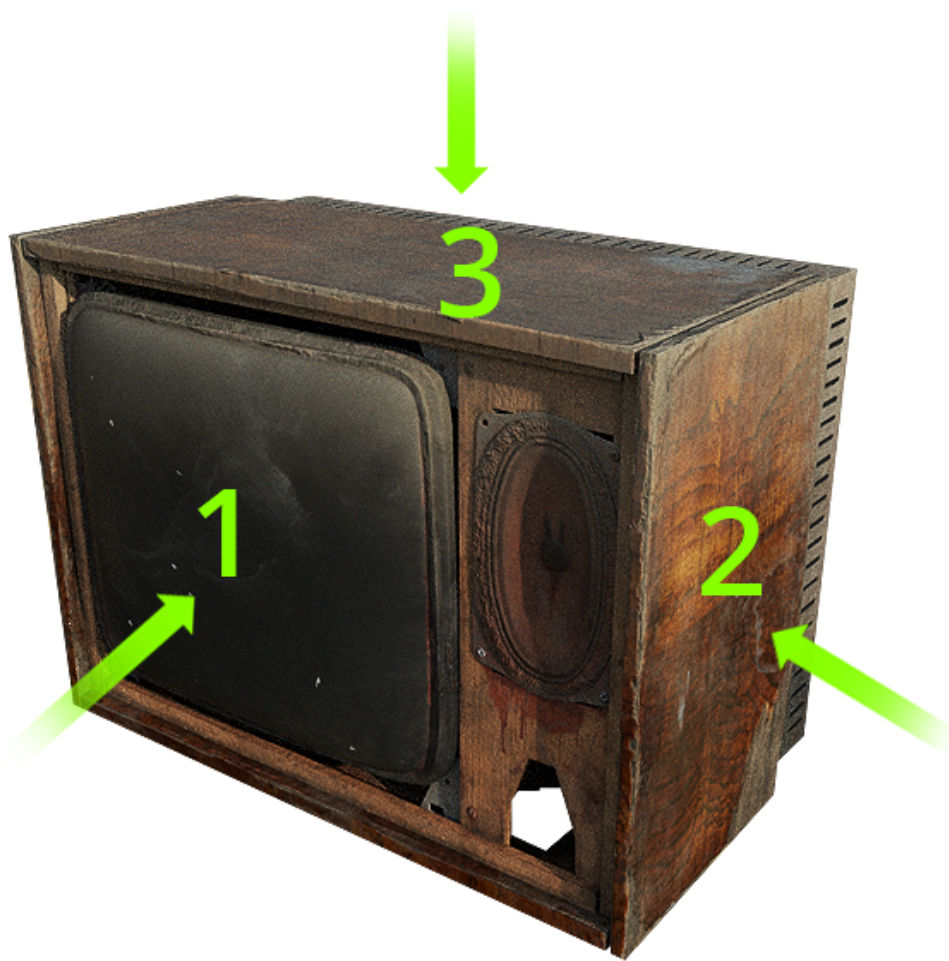


4. Направление UV шеллов

Иногда так случается, когда нам нужно наложить в движке процедурную текстуру чего-нибудь на модель. Чтобы такая текстура легла ровно и без проблем, нужно не только чтобы все части модели, на который планируется назначить конкретный материал, имели одинаковый текстел, но и чтобы направления этих шеллов были одинаковыми. Например, все смотрели вверх и не были флипнутыми.

5. Оверлапинг

Экономить нужно с умом. Например, используя оверлаппинг важно понимать что резать нужно только те части текстуры, которые одновременно не попадают в поле зрения игрока. Когда мы смотрим на объект, то видим максимально 3 его стороны:



В теории все эти 3 стороны можно оверлапнуть. Но на примере выше мы видим что верхняя и нижняя части сильно отличаются, а вот боковые есть прямой смысл зеркалить.

Оверлапинг сильно экономит место на развертке и позволяет нам получить лучший текстур при том же разрешении текстуры.



Особенно много места можно выиграть, если заOVERLAPить крупные куски.



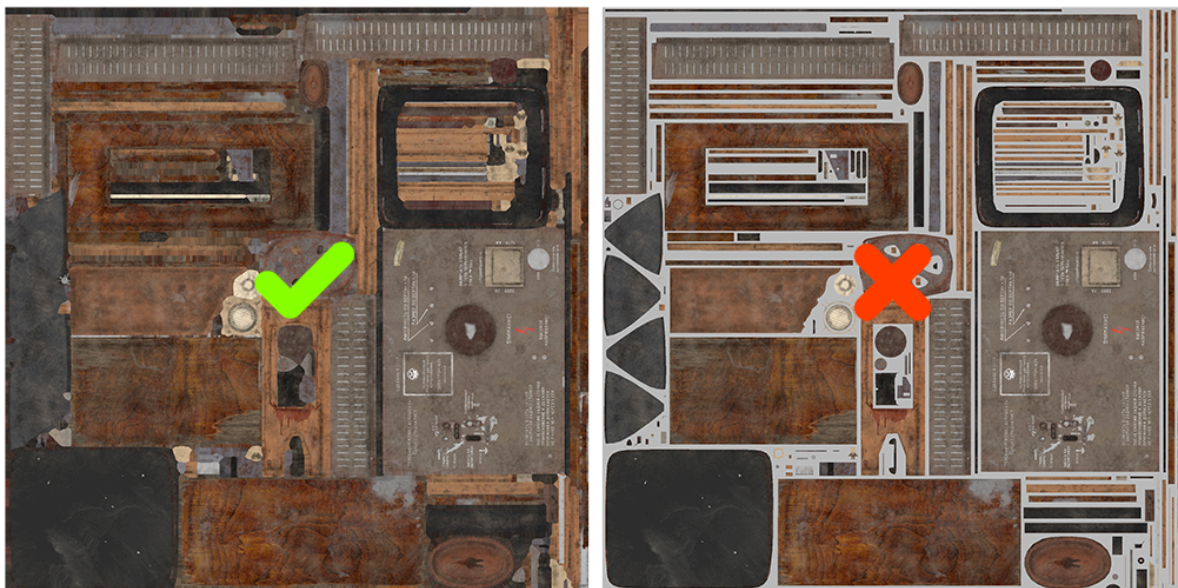
6. Padding

Важно соблюдать паддинг. Нельзя делать шеллы без соблюдения пространства между ними. Паддинг нужно соблюдать.

2048 — 10

4096 — 20

При экспорте текстур также важно чтобы стояло сохранение границ:



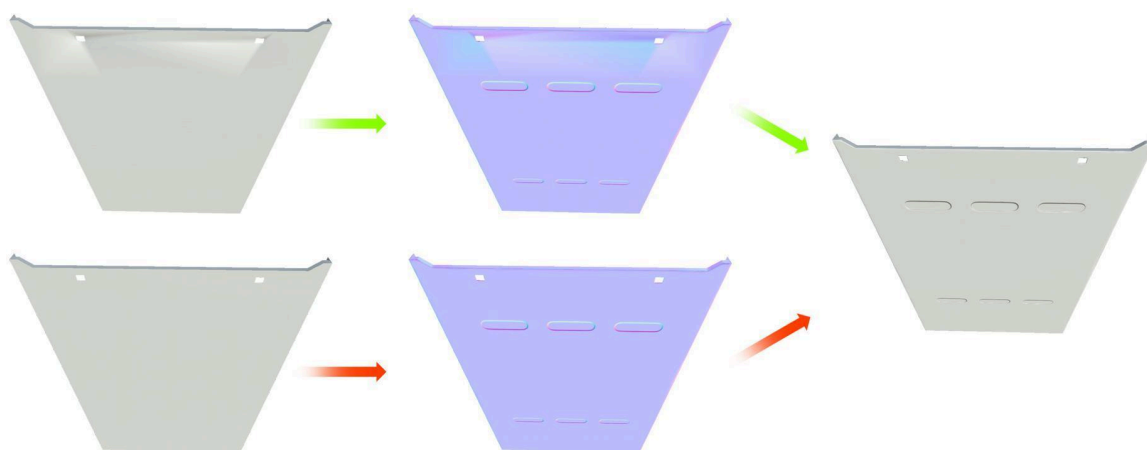
Если этого не сделать, на модели могут возникнуть проблемы в области швов.



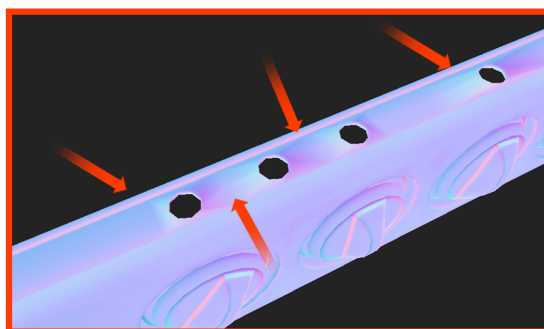
Normal Map

Шейд. Компенсация геометрии картой normal map.

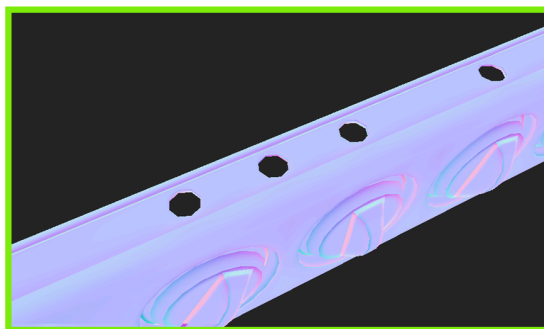
При подготовке lowpoly-модели очень важно, чтобы карта **normal** выполняла задачу добавления геометрии, а не функцию компенсации шейда. Это не только правило хорошего тона, но и необходимое условие для правильной генерации [мипмап](#). Поэтому шейды на лоуполи и на хайполи должны максимально совпадать. Если шейд на лоуполи сильно отличается от того что мы имеем на хайполи, это место компенсируется на карте, как видно на примере ниже.



То есть, на выходе у нас в обоих случаях получается идентичный результат, но карта нормала сильно отличается в области мест, где произошла компенсация разницы геометрии. И эти места в итоге при ресайзе (когда будут генерироваться мипмапы) могут привести к сильной потере в качестве. Для избежания проблем связанных с компенсацией, необходимо добиться максимально чистой карты **normal**, то есть, выровнять геометрию на лоуполи так, чтобы подобные места отсутствовали:



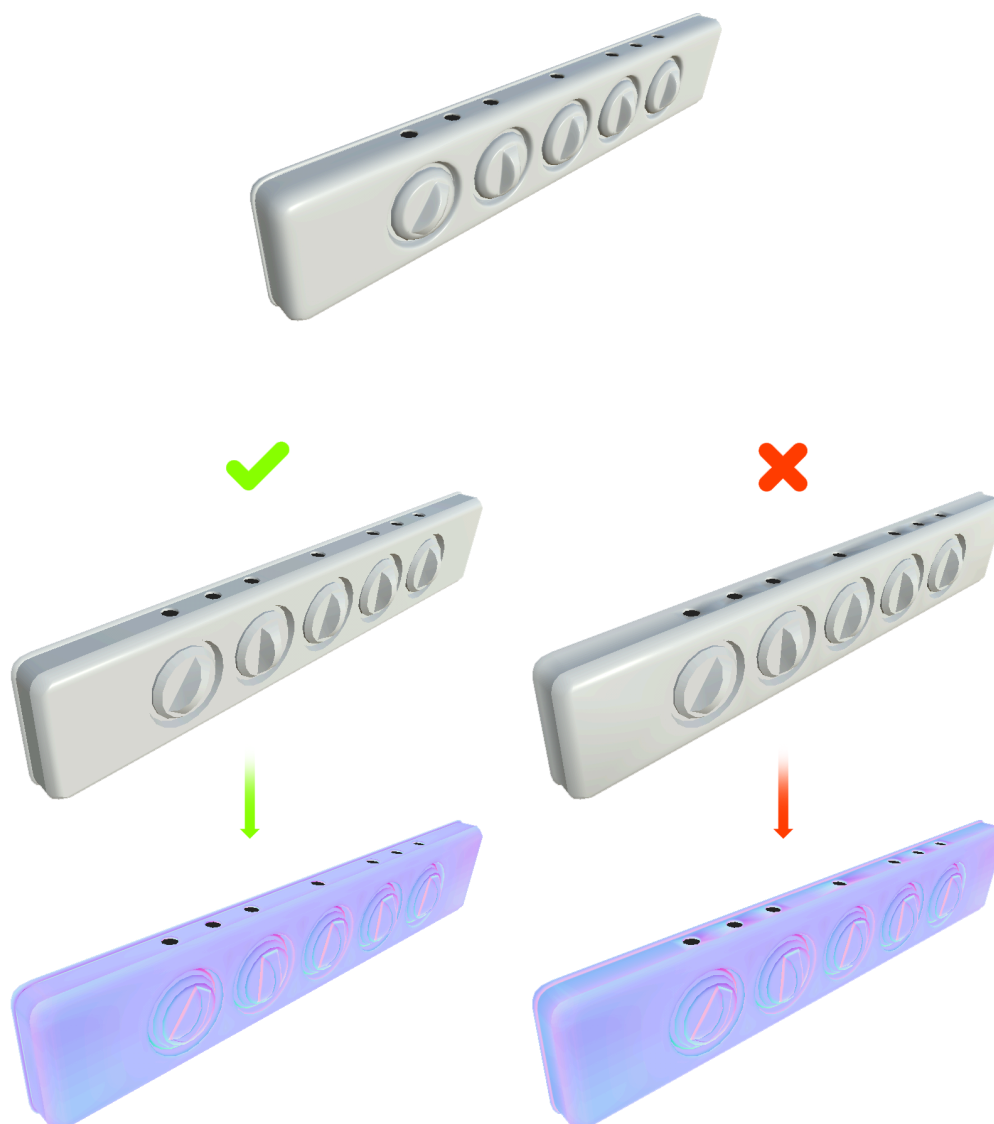
Хороший шейд для данного места выглядит так:



То есть, в карте **normal** нам нужна информация только в тех местах, которые мы реально хотим детализировать за счет хайполи. Если место на хайполи и на лоуполи

ровное, то мы избегаем жестких градиентов в нормале и вообще любой лишней информации, которая не выполняет функцию детализации нашего объекта.

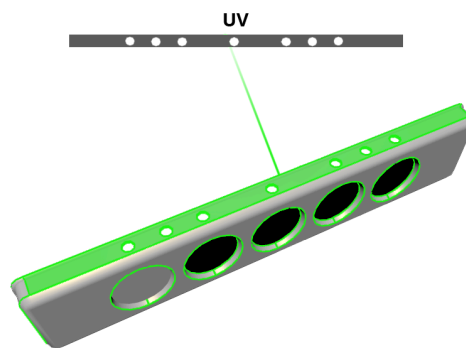
Смотрим еще один пример: сверху у нас лоуполи с запеченным нормалом. Слева ровный шейд в области отверстий, а справа наоборот — кривой шейд, который в итоге компенсирует карта **normal**. Нам нужен такой результат, как слева, где нормал просто добавляет нам фаски на отверстия, а область которая у нас ровная, таковой и остается в карте **normal**.



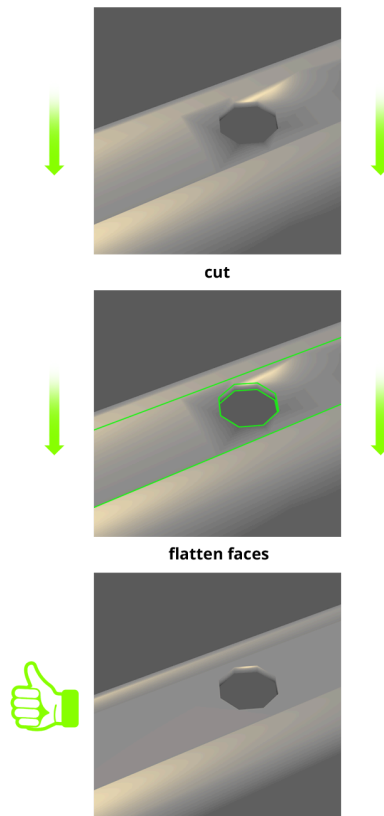
Итак, что же делать, если у нас уже вышел плохой шейд? Для начала убедиться что мы все сделали верно: сделали нашей модели **soften edge**, потом поставили **harden edge** в местах разрывов на uv.

Для выравнивания шейда у нас несколько действий:

1. **Average.** Мы можем выделить проблемную область и применить к ней функцию **average**, если проблема возникла на плоской поверхности. То есть, после того, как мы сделали нашей модели **soft edge** и поставили в местах разрывов uv **hard edge**, мы выделяем нужную нам область и выравниваем на ней нормали так, чтобы они были перпендикулярны к фейсам. Тут стоит учитывать, что кроме стандартных средств также существует и много дополнительных тулов и скриптов под разный софт (как пример, для Maya есть **Flatten Faces** на Python).
2. Если шейд проблемный во многих местах одновременно, либо не выравнивается простыми действиями, то можно просто отрезать проблемную зону на uv, либо даже отделить геометрию (в месте разрыва **uv** получается не **1** вертекс, а **4**, то есть, для видеокарты разрыв на **uv** = разрыв меша. Поэтому мы спокойно можем детачить геометрию для удобства работы с шеллами)



Отрезаем нужный нам кусок и выравниваем его. Если у нас есть примыкающие полигоны, как на примере ниже, мы их тоже отрезаем.



После манипуляций, меш можно склеить, сохранив все смус группы (также в некоторых случаях перед склейкой вертексов можно залочить нормали, чтобы не слетели группы сглаживания).

Bake scene

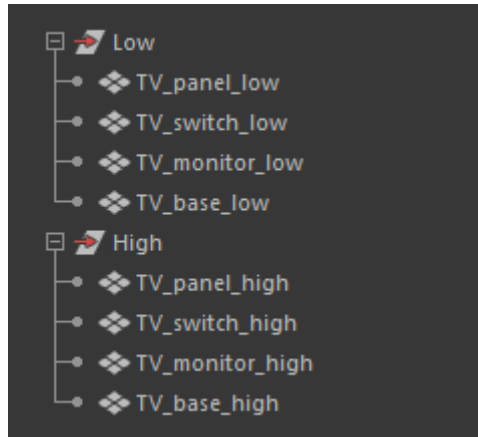
Удобный бейк дает сразу несколько преимуществ:

1. Мы можем сразу запечь все карты на все нужные нам группы элементов и начать в этой же сцене текстурить;
2. Легкость внесения правок: мы всегда можем заменить нужный элемент и перезапечь карты не отходя от кассы.
3. Удобно отдавать, удобно принимать сцену.

Хороший подход заключается в том, что все объекты для запекания у нас имеют уникальные имена. Это позволяет нам не делать отдельную бейк сцену для запекания нормала и отдельную для АО и текстурирования. Чтобы упростить себе жизнь, лучше сразу назвать все меши лоуполи и хайполи одинаковыми именами с разными суффиксами. Вообще в идеале у вас сцена в **Maya** или **3D's max** должна выглядеть так:



2 группы, одна с **lowpoly**, вторая - с **highpoly**.

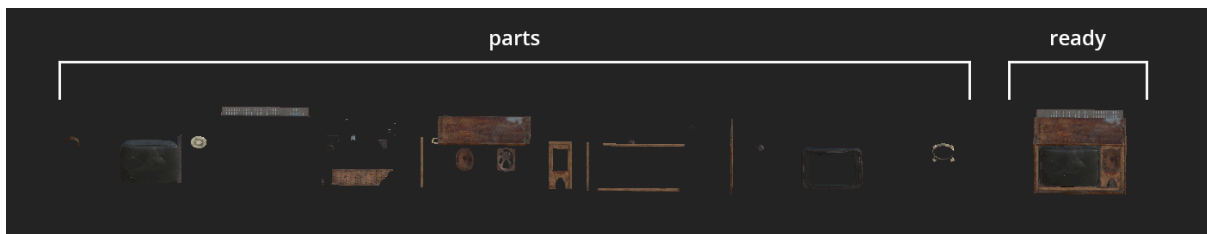


Лоупполи у нас идет с приставкой **_low**, хайполи идет с суффиксом **_high**.

В сабстенсе просто ставим **By Mesh Name** при запекании во вкладке **Match**, и **substance painter** автоматически понимает какой объект на какой печь, если мы не допустили ошибок в именах.

Немного об удобстве работы

Для того чтобы обеспечить себе максимальный комфорт работы, можно сделать сразу 2 варианта объекта: один в разобранном виде, второй - в сборке. Оба в одной сцене. Зачем это нужно? Как минимум, для того чтобы иметь больше контроля над текстурированием и видеть любые изменения как на отдельных деталях, так и на всем объекте в сборе одновременно.



Важно: перед запеканием карт главное не забыть оставить UV только у одной копии объекта, а UV второго перенести на единицу в любой соседний квадрат.

Частые ошибки и методы борьбы с ними

Как гласит старая добрая пословица:

“чтобы сделать все правильно, достаточно просто не допускать ошибок”

Поэтому мы будем стараться свести ошибки к минимуму и работать чисто.

Дырки это плохо.

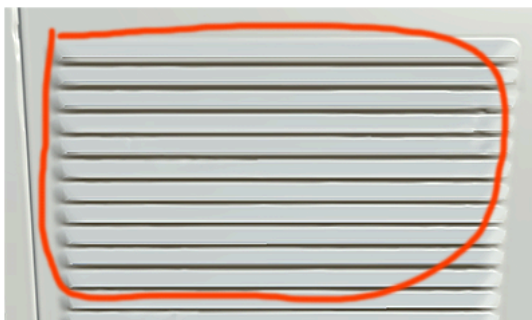
Смотреть сквозь объект это было бы неплохо уметь в реальности, но в игре это выглядит зашкварно. Поэтому в подобных ситуациях всегда следует дублировать геометрию, даже если место очень тонкое в реальности.



То есть, если мы что-то видим насквозь, это плохо. Толщину следует делать даже очень тонким элементам (с оверлапами, конечно).

Харды.

Не забываем ставить харды. Чтобы было проще жить, юзаем скрипты помогающие выбрать бордеры по UV. Иначе после бейка могут быть очень неприятные сюрпризы.



Работа с 3D сканами

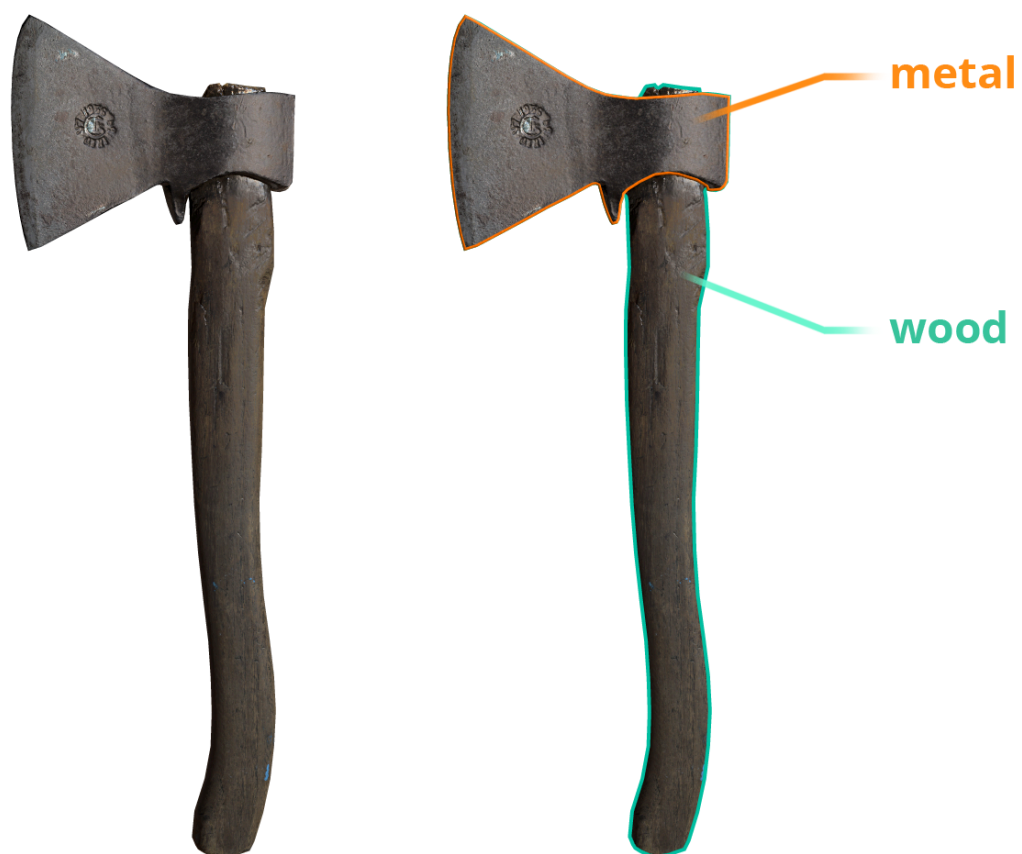
Идея

Создать качественную модель из скана - достаточно ответственная работа. Для того чтобы материалы были максимально близкими к реальным, вместе со сканом также выдаются фотографии объекта в естественной среде.

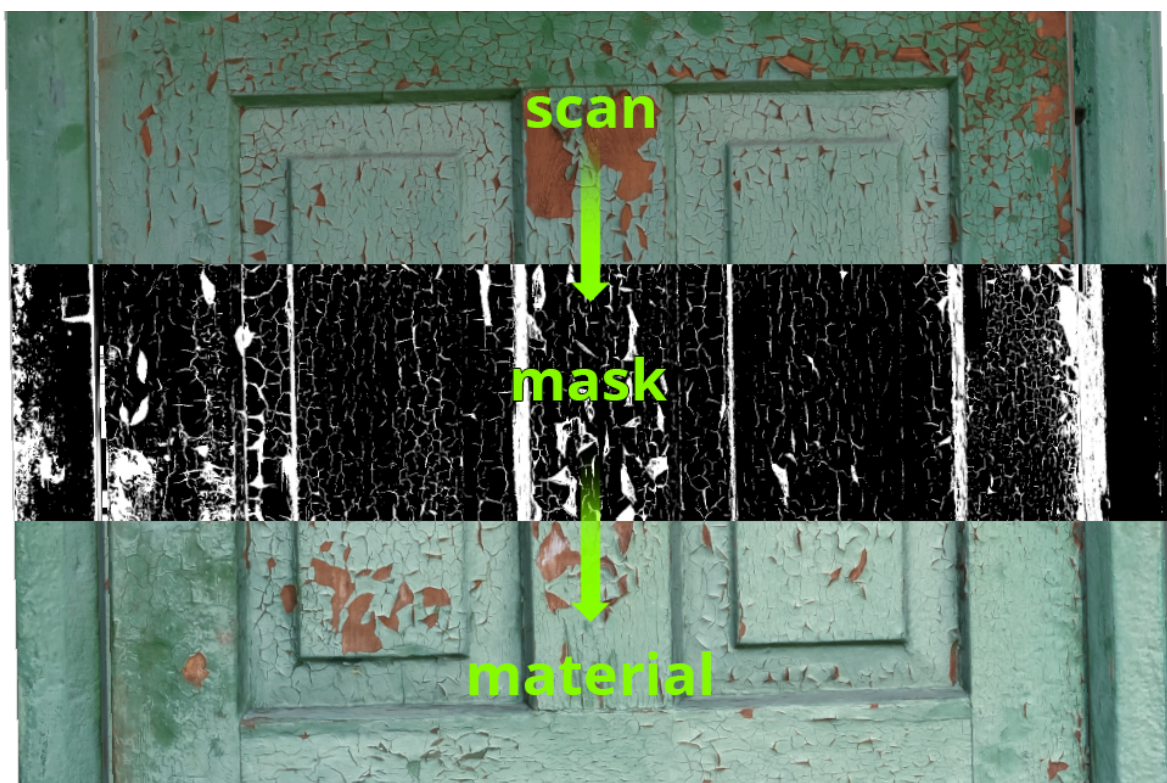
Далее необходимо верно прочесть материал и повторить его в PBR.

Карты

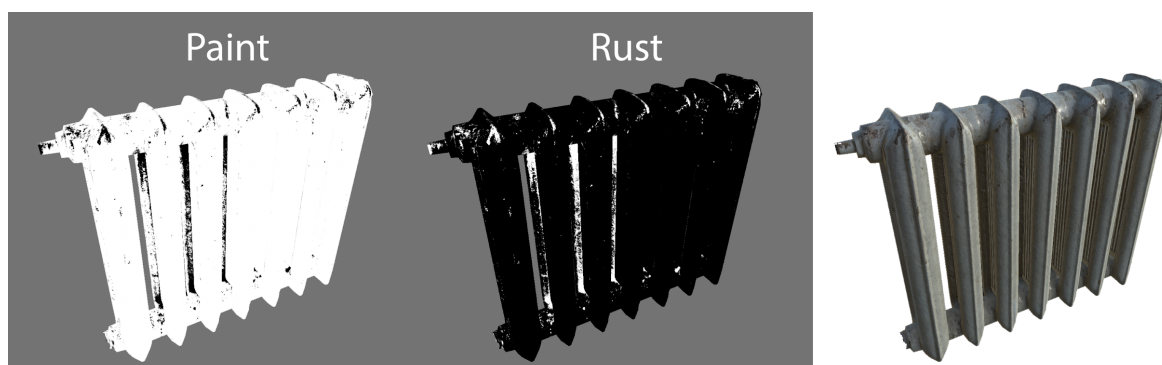
Далее необходимо верно прочесть материал и повторить его в PBR. Сначала нужно понять сколько на ассете материалов. Например, простой ассет вроде топора состоит из двух материалов: дерева и металла.



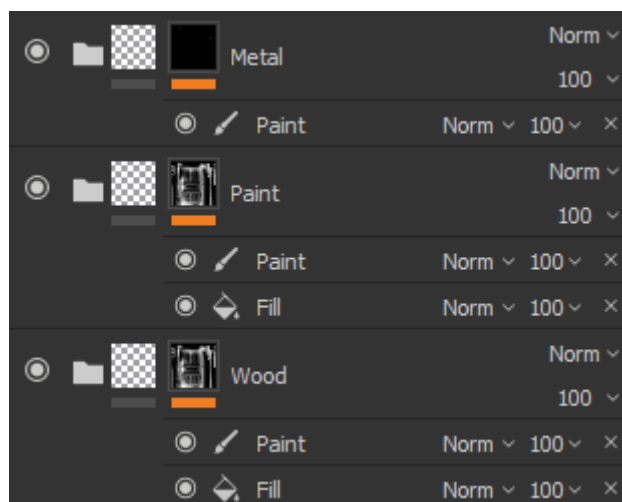
Для каждого материала нужна своя маска. Ее необходимо сделать самостоятельно.
Создаем black mask и для нее создаем **fill**.



Hint: маску лучше изначально создавать из карты **diffuse** с последующей доработкой. В photoshop можно использовать threshold (**Image** → **adjustments** → **threshold**), после чего ее нужно довести в сабстенсе (или в том же photoshop) руками.



Каждый материал должен называться соответственно и иметь собственную папку. Например, дверь состоит из трех материалов: краска, дерево и металл.



У каждой папки есть **blackmask**, к которой создается слой **fill**, куда, в свою очередь, кладется сделанная в **photoshop** маска, поверх накидывается **paint**, где дорисовываем нужные элементы (если они не были сделаны заранее в **photoshop**)