



Implosia FX

Ультрабыстрый, стабильный, нодовый шаттер под XSI

Разработано Mr.Core

Содержание

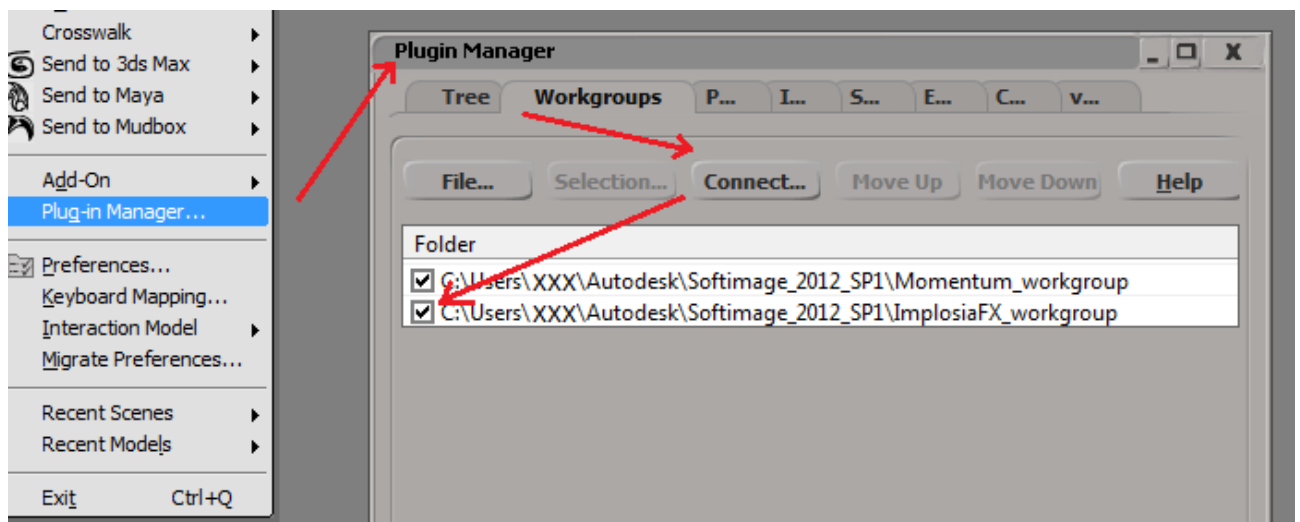
1. Введение.....	3
2. Быстрый старт.....	4
3. Комплексный шаттеринг.....	6
4. Центральный компаунд.....	9
5. Вспомогательные компаунды.....	12
6. Перечень нод.....	13
7. Существующие ограничения и баги.....	14
8. Обратная связь.....	15

Введение

На данный момент под XSI не существует адекватных шаттеров, все решения имеют либо низкую скорость, либо низкую стабильность, либо вообще ничего не имеют кроме головной боли и вызова острого желания шаттерить в другом пакете. **Implosia** разрабатывалась как эксперимент по резке геометрии на айсе, булеан-нода с возможностью прикрутить базовую резку хаотично покоруженными плоскостями. В процессе работы **Implosia** показала хорошие временные характеристики и возможность резать очень высоко-полигональные модели (>100K треугольников), этому способствовало в первую очередь булеан-ядро Carve CSG, написанное на с++ и изначально разработанное для процессинга результатов сканирования реальных объектов. В результате, **Implosia** превратилась в полноценный шаттер, которым можно управлять прямо из ICE. Текущая версия работает под 2012 64-битной версией Softimage, так же необходим редистрибутив VC++ 2010 (может быть скачан [ТУТ](#) или установлен из комплекта поставки).

Для установки необходимо подключить рабочую группу «**ImplosiaFX_workgroup**» из плагин менеджера Softimage.

File-> Plug-in Manager->Workgroups ->connect



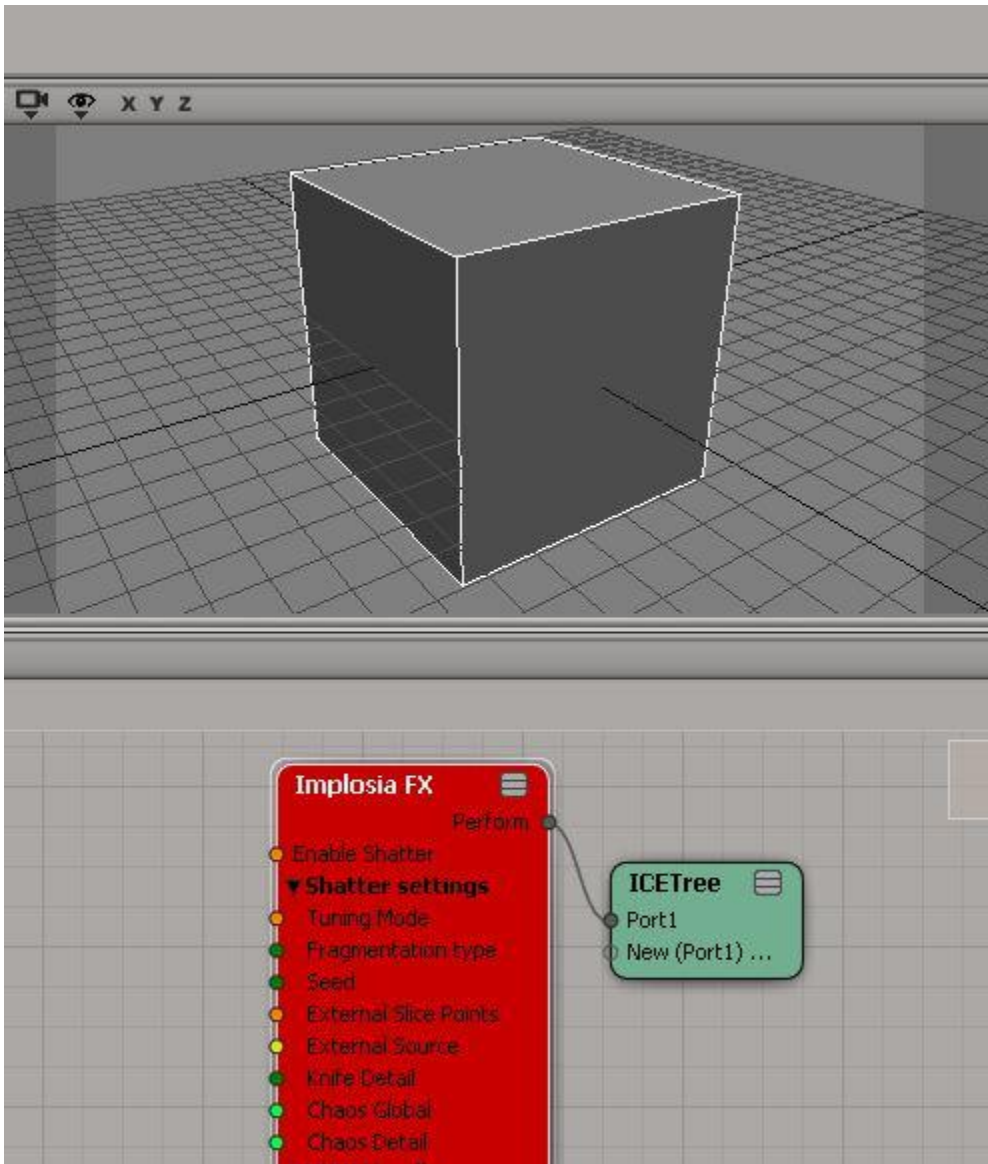
Перезапустите программу.

Быстрый старт

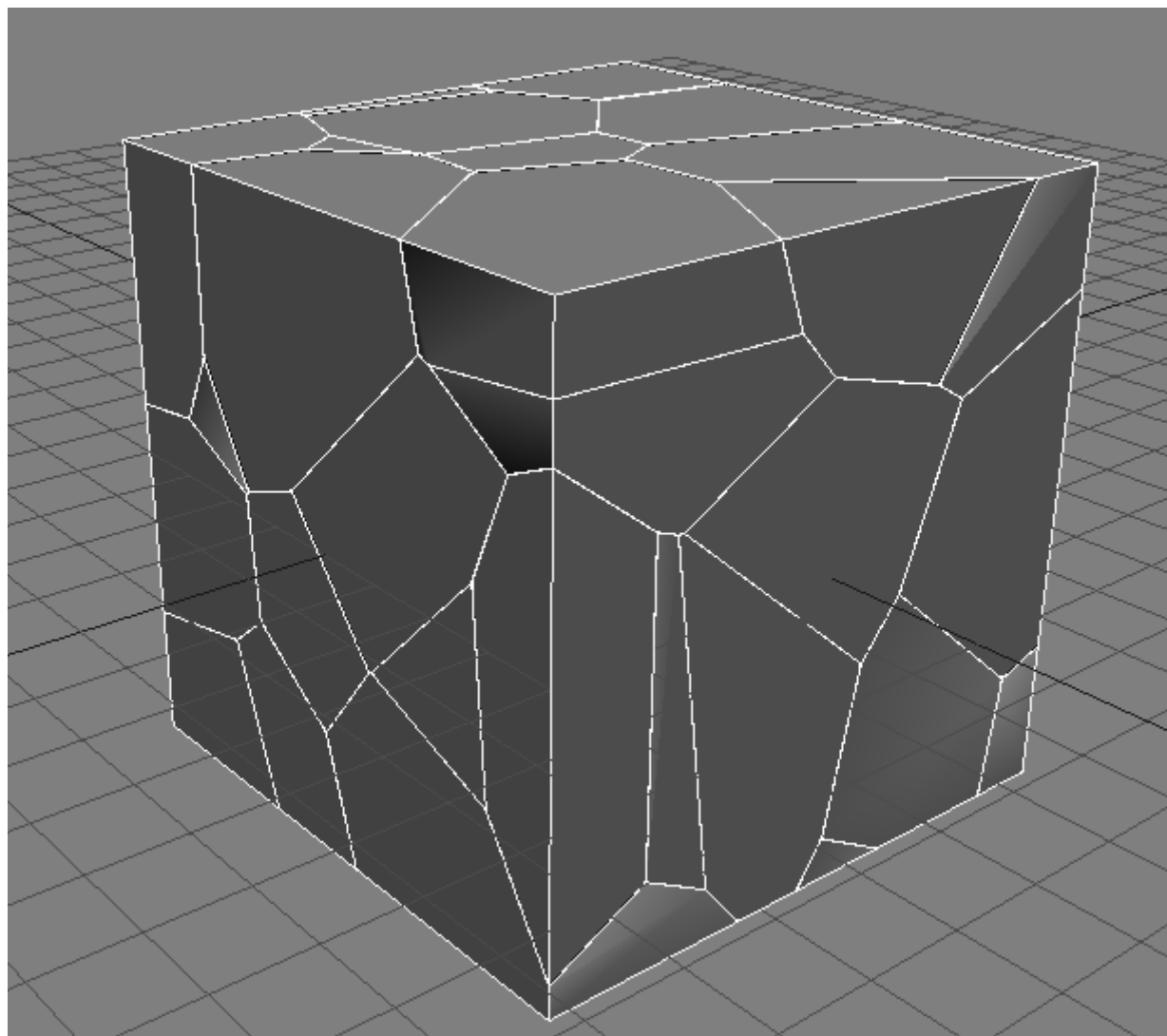
Крошение простого кубика в качестве примера.

- 1) Создайте полигональный куб.
- 2) Создайте на нем не симулированное ICE дерево в моделинг стеке.
- 3) Перетащите центральный компаунд **Implosia FX** из пресет менеджера.

Компаунд должен быть красным, так как содержит внутри ненайденный UV-атрибут.



- 4) Откройте PPG компаунда и поставьте галочку в порте **ENABLE SHATTER**, после этого снимите галочку в **TUNING MODE** порте.



5) Зафризьте геометрию.

Комплексный шаттеринг

На данный момент **Implosia** предоставляет два алгоритма резки – Вороной и Континиус.

Разбиение Вороного. Способ основан на разрезании геометрии на основе входных точек, вокруг которых на выходе будут выпуклые куски. Принцип резки основан на обтесывании каждого куска так, что бы каждая сторона данного куска была равноудаленно расположена между двумя соседними точками. Способ является самым быстрым из существующих в шаттере алгоритмов, но дает так же не слишком реалистичный результат. Путём хаотизации положения входных точек можно варьировать размер кусков.

Разбиение Континиус. В данном режиме геометрия разрезается набором входных ножей – хаотично покореженных триангулированных боксов, сплюснутых по оси Y. Способ является чуть более медленным чем Вороной, но при этом дает очень реалистичные результаты. Имеет гибкий контроль, позволяющий располагать плоскости в пространстве практически как угодно, получая эффекты начиная от обычных неравномерных обломков бетона, и заканчивая нарезанными щепками\кольцами дерева.

Перед шаттерингом необходимо:

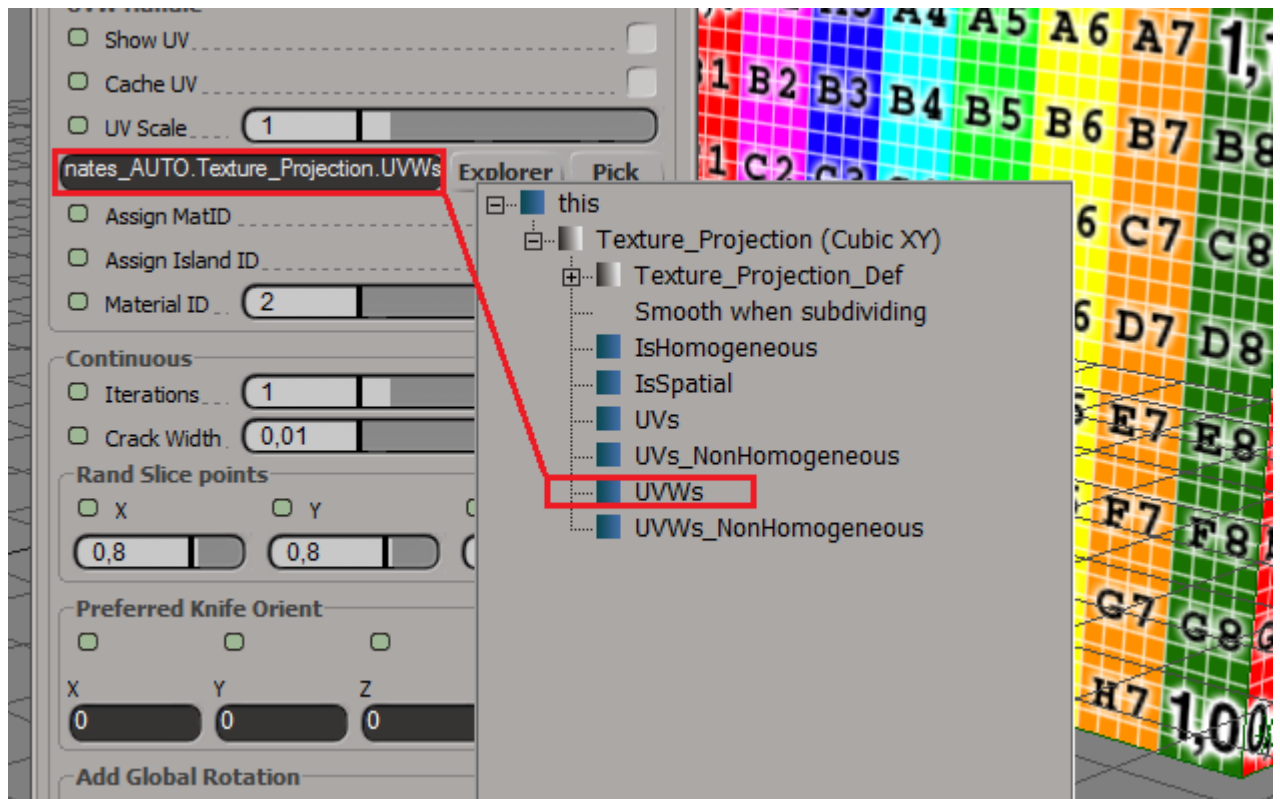
- 1) Зафризить текстурные проекции
- 2) Закрыть все дыры в геометрии
- 3) Убрать само-пересечения геометрии
- 4) Зафризить масштабирование геометрии
- 5) Если объект сложный (хай-поли\присутствуют многосторонние полигоны\объект имеет очень неравномерную топологию) -необходима триангуляция

После подготовки геометрии, на ней создается моделинг-стек с ICE деревом, и подсоединяется центральный компаунд **Implosia FX**. Он содержит все необходимые компоненты шаттера, включая процедурную генерацию ножей с ЮВ-координатами и кеширование некоторых атрибутов. На данный момент реализация поддержки UV и кластеров на ICE очень плохая, именно для этого и необходимо кешировать связанные с этим атрибуты.

Подробнее настройки различных режимов фрагментации рассмотрены в параграфе «**Центральный компаунд**».

После того, как был настроен и применен компаунд, перед финальной резкой необходимо отметить следующие галочки –

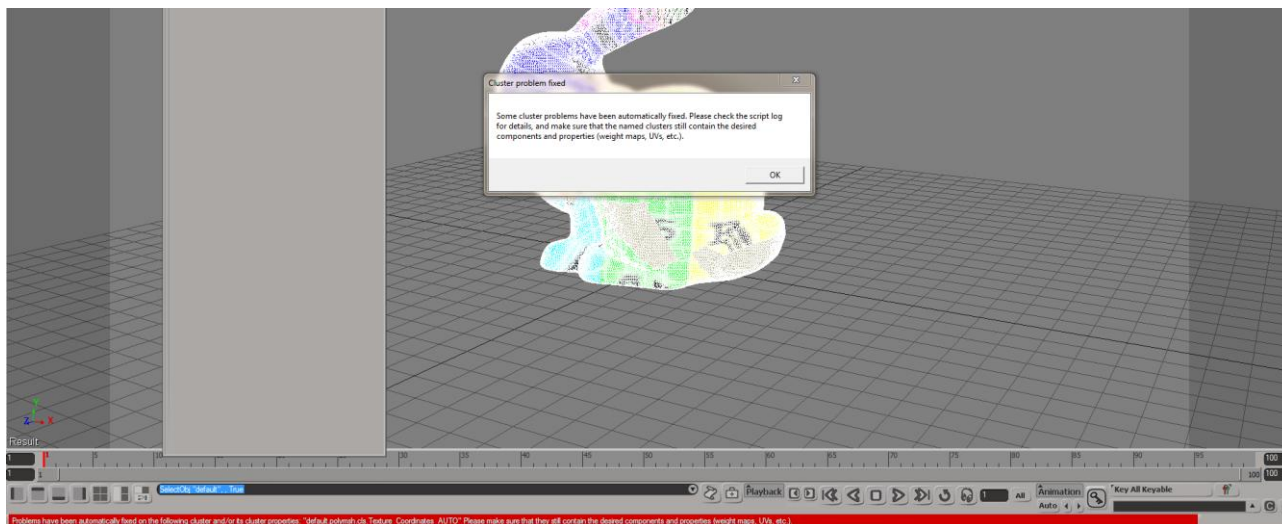
- 1) **Cache UV** – перенос ЮВ-координат на порезанную геометрию в обход глюков ICE путем кэширования атрибута. Необходимо так же указать путь к атрибуту ЮВ-координат (следствием правильного пути будет смена цвета компаунда на фиолетовый)



- 2) **Assign MatID** – назначение индексов полигонам исходной части геометрии и вырезанной, для последующего применения материалов. Так же кэшируется.
- 3) **Assign Island ID** – назначение и кэширование индекса полигон-острова группам полигонов. Может быть в дальнейшем использовано для быстрого экстракта осколков в отдельную геометрию с помощью скриптов.

После того, как отмечены все галочки, отключается TUNING MODE и результат фризится.

При этом возможно появление различных ошибок, наподобие этой:



Это является, как уже говорилось выше, следствием недоработки ICE в области моделинга.

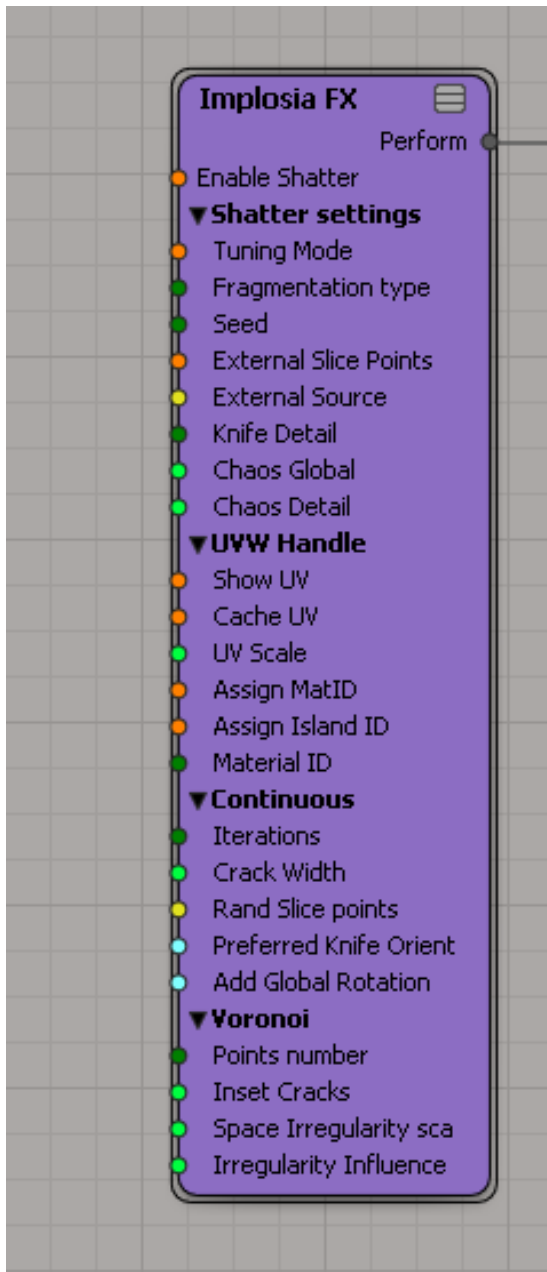
Далее на пофриженую геометрию в дерево на моделинг стеке добавляется компаунд Apply Material Data, где отмечаются соответствующие галочки для считывания атрибутов из кеша.

Примечание – на данный момент, атрибут Island ID требует для корректного просчета включение атрибута Face ID.

Окончательно зафризьте геометрию.

Центральный компаунд

Центральный компаунд – **Implosia FX** - содержит несколько свитков, отвечающих за определенную ступень шаттеринга.



1) Свиток **Shatter settings** определяет основные настройки шаттера.

- **Tuning mode** – позволяет настраивать шаттеринг путем визуального наблюдения основных детлей, для Вороного это отображение точек-центров, для континууса – отображение положения ножей и их основных характеристик.
- **Fragmentation type** – выбор типа разбиения.
- **Seed** – значение опорной выборки (номер уникальности разбиения)

- **External Slice points** – определяет, использовать ли в качестве точек интернальные значения компаунда, или внешние данные (работает для обоих типов разбиения).
- **External Source** – позволяет задать точки для резки, в случае вороного данные будут использованы как центры диаграммы вороного, для континиуса – центры ножей (использовать с осторожностью, если выбран тип континиус, при значениях >50 количество осколков на объектах, имеющих баундинг бокс приближенный по форме к кубу может быть очень большим, равно как и время резки).
- **Knife detail** – на данный момент определяет детализацию ножей для континиус типа, вынесен в глобальные настройки в связи с потенциальным использованием в других типах фрагментации, основанных на ножах.
- **Chaos Global** – определяет силу больших вмятин ножа.
- **Chaos Detail** – определяет силу маленьких вмятин ножа.

2) Свиток **UVW Handle** определяет обработку ЮВ-координат\материалов и прочих утилитарных данных.

- **Show UV** – Отображение текстуры на порезанной геометрии до фриза. Может некорректно отображать данные из-за вышеописанных глюков ICE. Кеширование снимает эти проблемы.
- **Cache UV** –кеширует ЮВ атрибут.
- **UV Scale** – масштаб ЮВ-проекции на порезанных частях.
- **Assign MatID** – назначает исходным полигонам уникальный индекс и кеширует его.
- **Assign Island ID** – назначает группам полигонов, образующих отдельный цельный суб-объект (polygon island) уникальный индекс и кеширует его.
- **Material ID** – уникальный индекс для исходных полигонов. Не может быть отрицательным в служебных целях.

3) Свиток **Continuous** определяет специфические настройки континиус-разбиения.

- **Iterations** – количество режущих ножей.
- **Crack Width** – Толщина ножей.
- **Rand Slice points** – определяет отклонение центров ножей от центра баундинг-бокса объекта до его границ. (0 – нету отклонения \ 1 – максимальное отклонение).

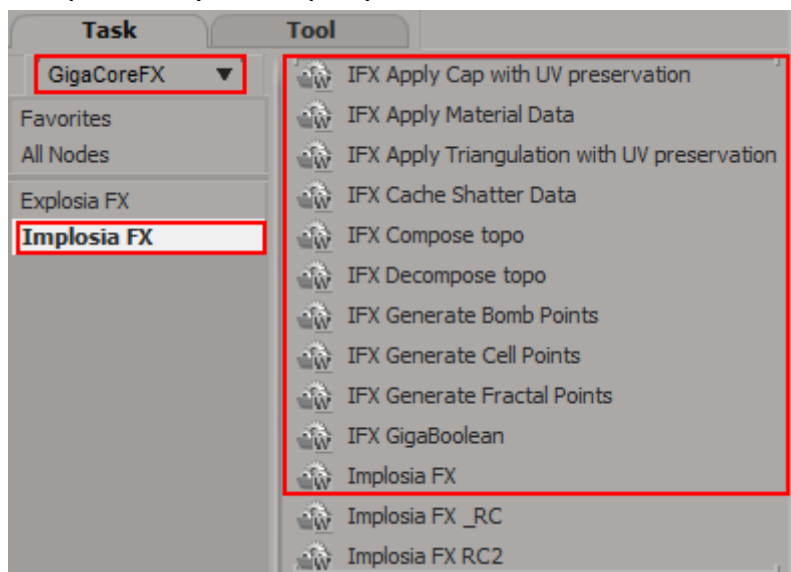
- **Preferred Knife orient** – первые три поля определяют ось, по которой ножи будут иметь выравнивание (большие значения ведут к большему выравниванию ориентации), четвертый параметр определяет допустимый диапазон случайного вращения по заданной оси.
- **Add Global Rotation** – поворачивает все ножи на данное вращение.

4) Свиток **Voronoi** определяет специфические для данного типа разбиения параметры.

- **Points Number** – количество кусков.
- **Inset Cracks** – толщина зазоров между кусками.
- **Space Irregularity Scale** – определяет размер неравномерностей распределения точек, генерируемых интернально.
- **Irregularity Influence** – принимает значения от 0 до 1, регулирует неравномерность заполнения точек. Базируется не на простой фильтрации, поэтому сохраняет исходно заданное количество точек.

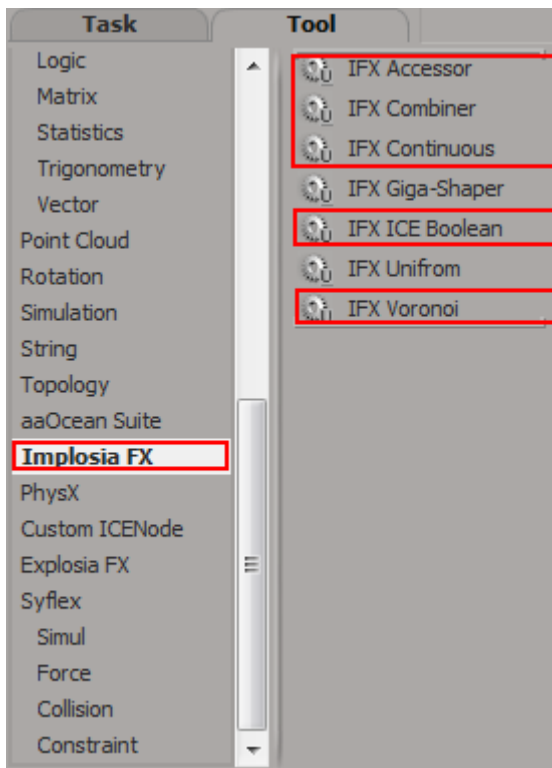
Вспомогательные компаунды

- 1) **IFX Apply Material Data** – служит для переноса ЮВ-данных и соответствия материалов разным частям порезанной геометрии в обход глюков ICE.
- 2) **IFX Cache Shatter Data** – используется для кэширования второстепенных топоданных, таких как ЮВ\ индекс материалов\ индекс полигональных островов. Используется интернально центральным компаундом, может быть использован при сборке кастомных шаттеров.
- 3) **IFX Compose topo \ IFX Decompose topo** – используются для конвертации данных о геометрии включая основные и вторичные атрибуты. Могут быть использованы в кастомных шаттерах.
- 4) **IFX GigaBoolean** – булеан модуль, заменяет встроенный устаревший.
- 5) **IFX Generate Points** – набор компаундов, предназначенных для генерации внешних точек.
- 6) Остальные компаунды являются сугубо служебными и могут давать непредсказуемый результат, использование на свой страх и риск.



Перечень нод

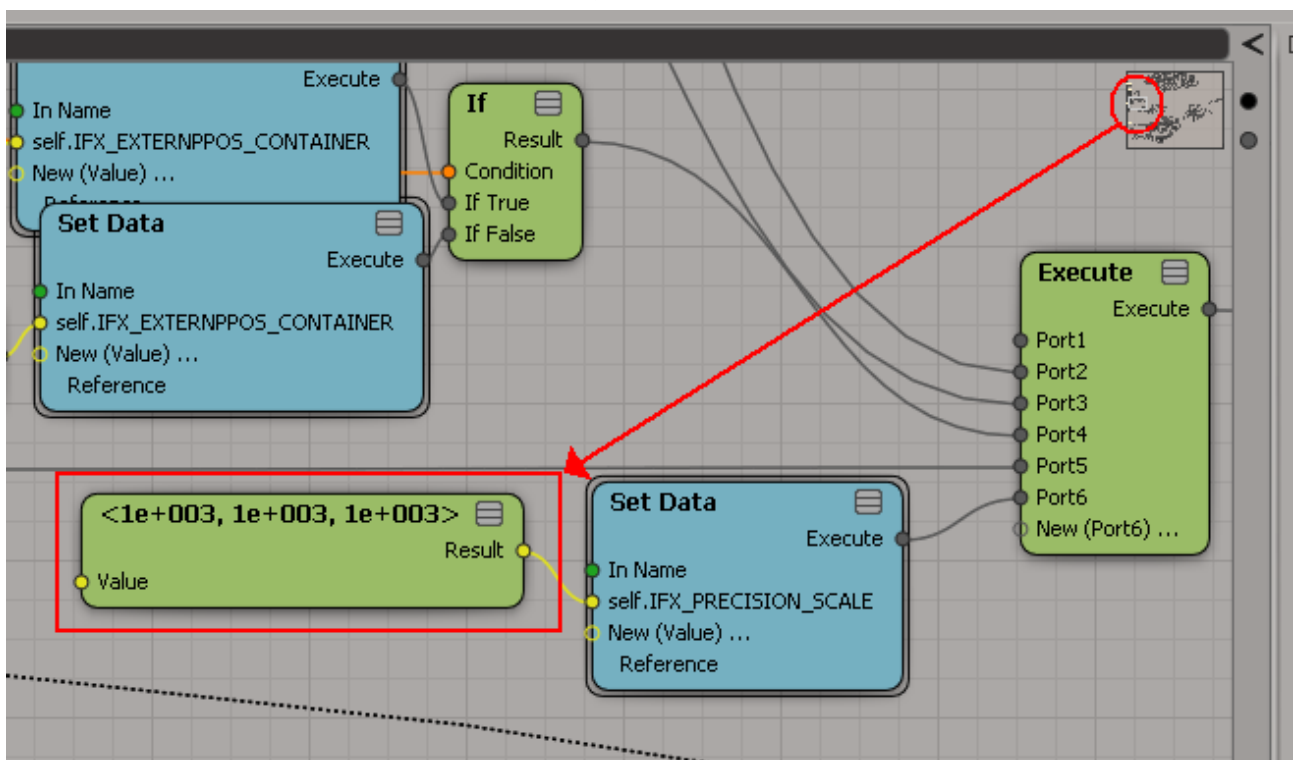
- 1) **IFX ICE Boolean** – имплементация булевой библиотеки Carve CSG под ICE, используется компаундом булеан-модуля, также может быть использована как ядро кастомного шаттера.
- 2) **IFX Accessor** – конвертор, извлекает данные из типа gtopo.
- 3) **IFX Combiner** – комбинирует исходные данные в тип gtopo, возможна фильтрация на основе входного значения порта Face ID.
- 4) **IFX Voronoi** – зашитый в ноду алгоритм разбиения по вороному, используется центральным компаундом шаттера, ни для чего более не предназначен.
- 5) **IFX Continuous** – зашитый алгоритм континуус, использование так же ограничено центральным компаундом.



Существующие ограничения и баги

Implosia FX является некоммерческим экспериментом в области RD\FX, написанная автором без глубокого понимания с++ и программирования в целом☺. Так же используются некоторые сторонние библиотеки без сопутствующей к ним документации, что в целом обуславливает наличие мелких и не очень багов.

При некорректных результатах в первую очередь необходимо **ТРИАНГУЛИРОВАТЬ** геометрию, это снимает 90% артефактов, связанных непосредственно с топологией. Если геометрия маленьких размеров, или же имеет очень мелкие детали, имеет смысл отмасштабировать её вручную в 10...10000 раз перед шаттерингом и после него соответственно в 0.1...0.0001. Это в некоторых случаях снимает неточности машинного вычисления с числами с плавающей запятой. Вместо ручного масштабирования так же можно воспользоваться интернальным атрибутом в центральном компаунде, и увеличить его значение в соответствующее количество раз.



Возможные баги с неправильным переносом UV пока что неразрешимы, но как показывает практика, в 90% случаях просто забывают указать нужный ЮВ атрибут\ забывают поставить галку кеширования.

Обратная связь

Вопросы и предложения присылайте на почту – GbotFx@gmail.com.

Если вы находите Имплотию полезной, или она приносит вам деньги – вы можете продонейтить её будущее развитие, а так же другие некоммерческие проекты.

WebMoney кошельки:

Российские рубли - **R366923999702**

Гривны Украины - **U201684264764**

Доллары США - **Z140468531908**

Так же просьба присылать на почту ссылку на ваши работы, выполненные с привлечением Имплотии, если только они не находятся под DNA. ☺