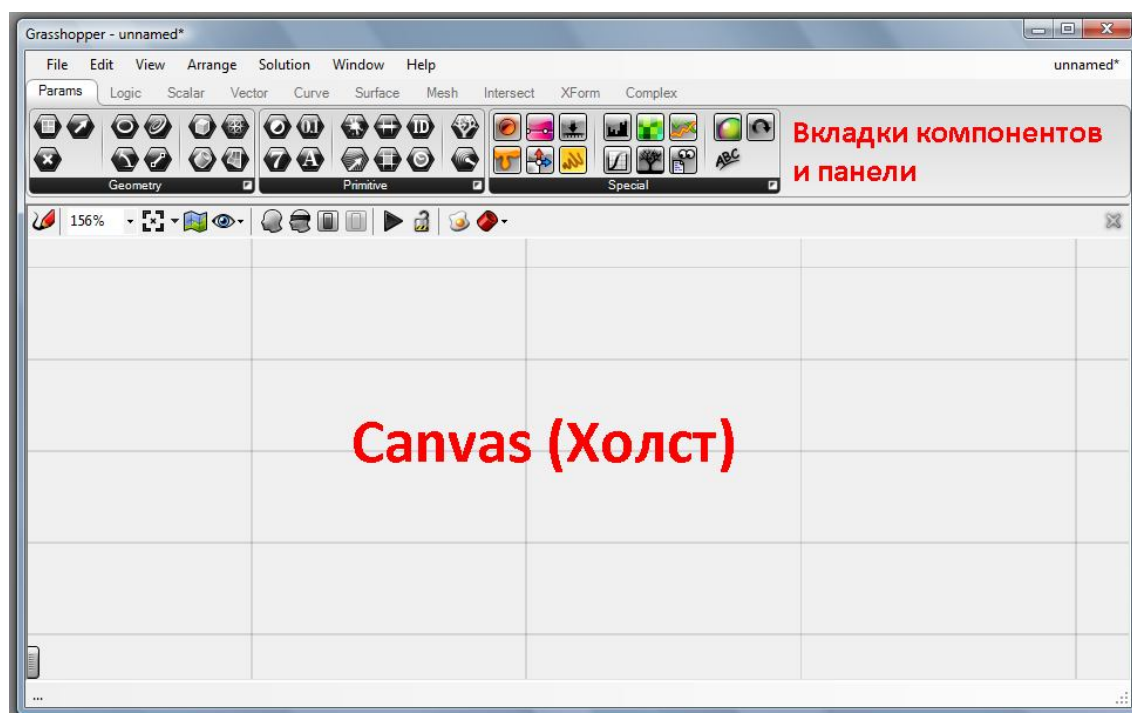


2. Основы

2.1. Интерфейс, рабочая область

Интерфейс Grasshopper состоит из двух основных частей: Панели компонентов и Холста. Панель компонентов содержит все элементы, которые необходимы нам для создания нашего дизайна. Холст – это рабочее пространство, куда помещаются компоненты и где задаются связи между ними, определяющие наш алгоритм построения. Для добавления компонента на холст Вы можете щелкнуть на нем мышью, а затем щелкнуть на холсте, чтобы компонент появился на нем. Или вы можете просто перетащить компонент на холст. Другие части интерфейса также просты в использовании, и мы познакомимся с ними в дальнейшем. Больше информации по этому вопросу доступно по ссылке:

<http://en.wiki.mcneel.com/default.aspx/McNeel/ExplicitHistoryPluginInterfaceExplained.html>



2.2. Компоненты

Различные типы компонентов в Grasshopper расположены на вкладках (Tabs) в верхней части окна. В пределах вкладок однотипные компоненты дополнительно сгруппированы в панели. Компоненты можно найти на вкладках: Params, Logic, Scalar, Vector, Curve, Surface, Mesh, Intersect, XForm and Complex.

Params: компоненты импорта готовой геометрии из Rhino; числовые, строковые и др. примитивы; специальные компоненты (числовой слайдер, панель, градиент, чтение из файла и др.)

Logic (логические компоненты): булевы функции, операторы сравнения, работа со списками данных, создание функций, сценариев, работа с диапазонами и деревьями данных.

Scalar (скалярные): константы, числовые интервалы, математические операции, степенные, логарифмические и тригонометрические функции.

Vector (векторные): компоненты работы с цветом, векторами, точками и сетями точек.

Curves (кривые): компоненты создания, редактирования, разбиения и анализа кривых.

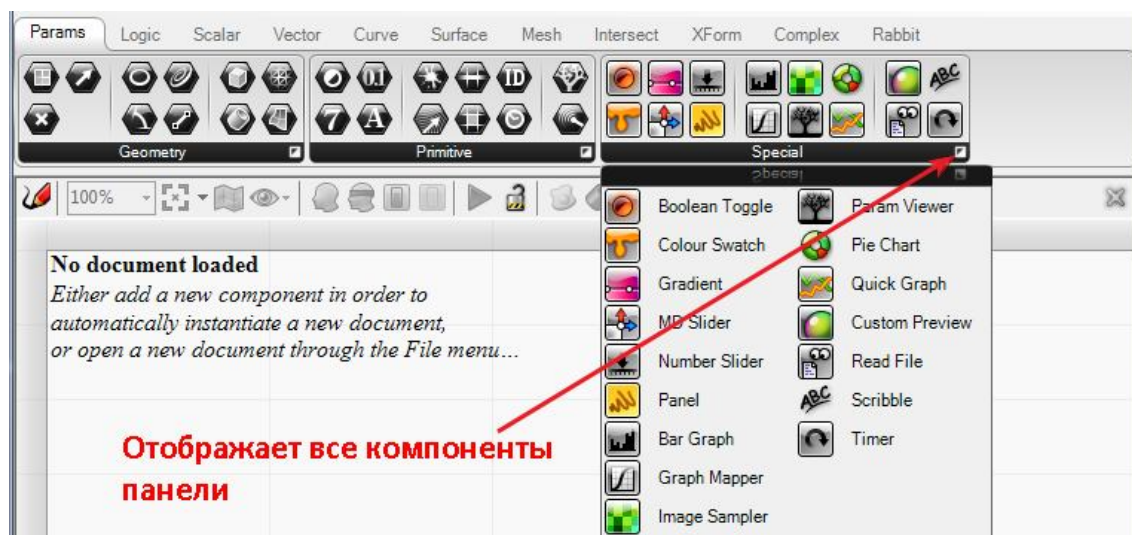
Surface (поверхности): создание поверхностей и объектов, анализ и редактирование.

Mesh (сеть): работа с полигональными сетками, инструменты триангуляции.

Intersect (пересечение): булевы операции над объектами, вычисление пересечений кривых и поверхностей.

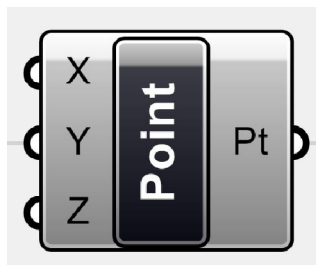
XForm (трансформации): компоненты морфинга объектов, выполнения аффинных и евклидовых трансформаций.

Complex (комплексные): работа с комплексными числами.



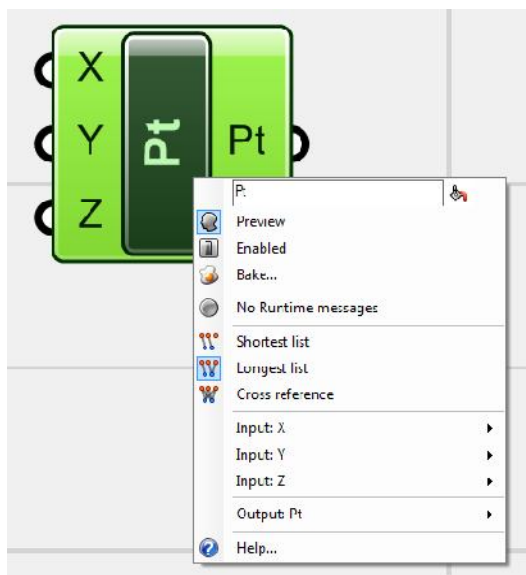
Некоторые компоненты генерируют объекты и отображают данные, некоторые предназначены для манипулирования существующей геометрией или данными. Параметрами называются объекты, которые представляют собой данные. Например: точка или линия. Вы можете задать их, используя соответствующие параметры (например: координаты) или из начерченных вручную объектов с рабочего поля Rhino. Компоненты – это объекты, которые выполняют какие-то действия (например: перемещение, задание ориентации, разбиение объекта на части). Для того чтобы компонент смог выполнить заложенные в него действия, ему необходимы соответствующие исходные данные.

В данном руководстве будет использоваться термин компонент, когда речь идет о любых объектах Панели компонентов. А для выделения в тексте названий компонентов будут использованы <>, например <Point>.



Компонент <Point> (точка)

Если щелкнуть правой кнопкой мыши на компоненте, то появится меню, которое содержит основные свойства и настройки компонента. Данное меню называется **“Context pop-up menu”** («Контекстное всплывающее меню»).



«Контекстное всплывающее меню»

Начиная с данного момента, нам нужно будет отыскать подходящие компоненты на Панели компонентов и задать соединения между ними для генерации вашего алгоритма построения геометрии и просмотра результатов в окне Rhino. Если при написании программ и сценариев кодируется абстрактная версия алгоритма на языке программирования, то здесь в Grasshopper холст содержит визуальную версию алгоритма наподобие блок-схемы, которая более понятна и гибка для дизайнера.

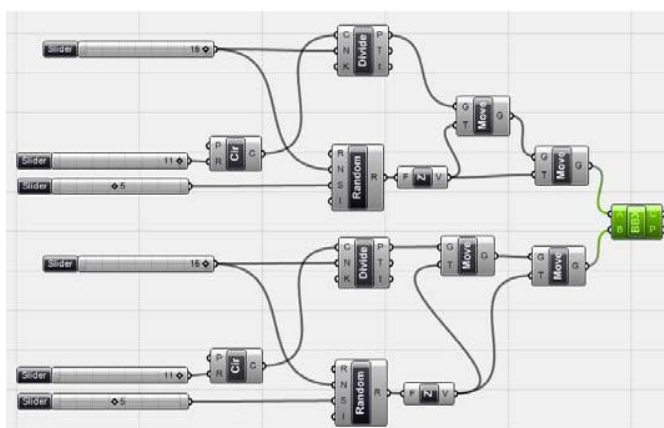
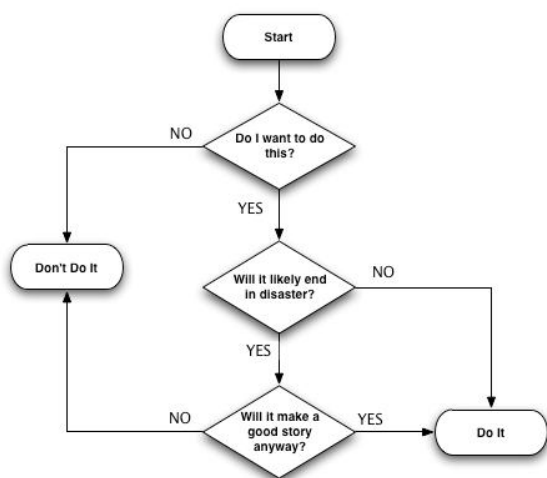


Рис. 1. Блок-схема и алгоритм Grasshopper

Задание внешней (существующей) геометрии

В большинстве случаев мы начинаем наш проект дизайна чего либо, с передачи нарисованных в Rhino объектов на холст Grasshopper. Это могут быть точки, кривые, поверхности и даже множество сложных объектов. Это означает, что вы можете использовать созданные вручную в Rhino или сгенерированные с помощью сценария объекты как внешние исходные данные в Grasshopper. Поскольку любая геометрия в Grasshopper требует наличия компонента на холсте для того чтобы иметь возможность работы с ним, то мы должны определить внешнюю геометрию с помощью соответствующих компонентов. Для этого необходимо обратиться к панели Geometry на вкладке Params. Она содержит перечень компонентов различных типов, которые позволяют передать внешнюю геометрию из рабочей области Rhino в Grasshopper.

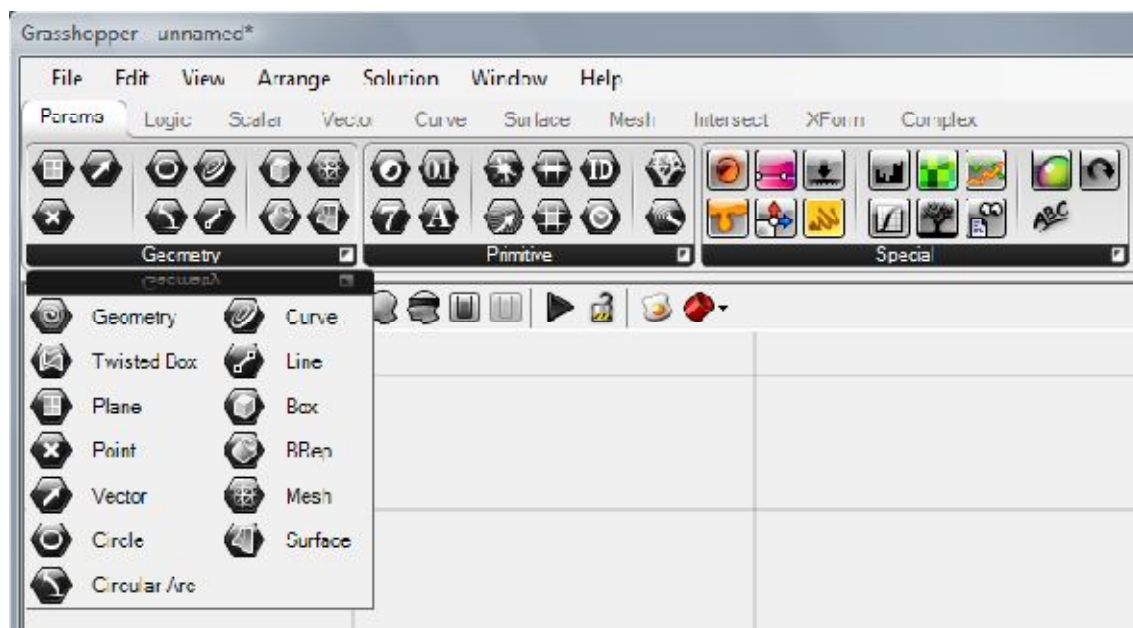


Рис. 2. Различные типы геометрии на панели Params > Geometry

После добавления компонента на холст, необходимо задать объект Rhino, вызвав контекстное меню и выбрав пункт «set one ... / set multiple ...» для задания объекта, соответствующего компоненту. Затем нужно выделить вашу геометрию в Rhino. С момента назначения одного/нескольких объектов компоненту они представлены объектами Grasshopper и могут быть использованы для любых целей разработки дизайна.

Выполним простой пример.

Мы имеем три точки, нарисованные в Rhino, и хотим из них построить треугольник в Grasshopper. Первым делом мы должны передать эти точки в Grasshopper. Для этого нам понадобятся три компонента <Point> из Params > Geometry > Point. У каждого из них нужно зайти в контекстное меню (правой кнопкой), выбрать 'set one point' и затем выбрать точку в окне Rhino.

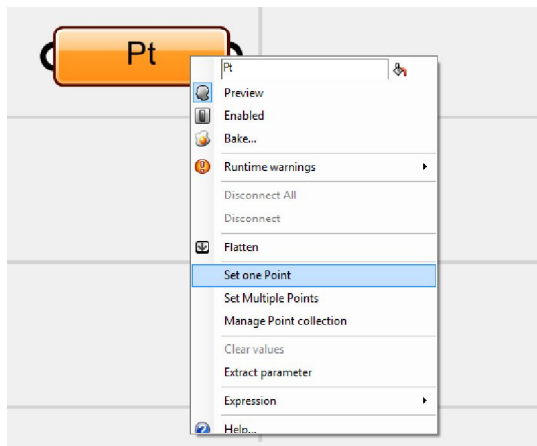


Рис. 3. Задание точки из Rhino в компоненте Grasshopper

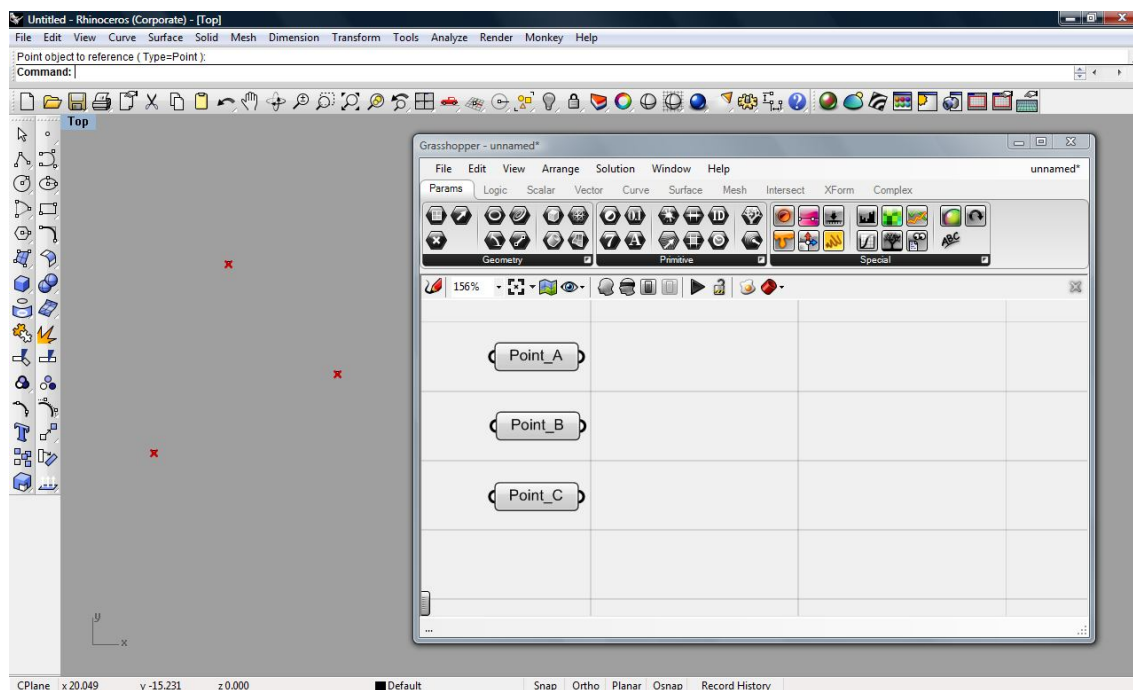


Рис. 4. Холст Grasshopper и три точки, которые заданы в Grasshopper (отображаются красными крестиками в рабочем окне Rhino). Компоненты переименованы в Point A/B/C с помощью первой опции их контекстного меню, для распознавания их на холсте.

Соединение компонентов

Существует много действий, которые можно производить с компонентами. Основное то, что компоненты могут брать некоторые данные из одного или нескольких источников и возвращать обратно результат. Нам нужно соединить компоненты, которые содержат исходные данные с компонентом, который эти данные обрабатывает. Результат обработки мы можем передать другим компонентам, для дальнейшей обработки и так далее.

Вернемся к нашему примеру, и перейдем на вкладку Curve и на панели Primitive отыщем компонент <line>. Перетащим его на холст. Затем присоединим компонент <point A> к порту (входу) 'A' компонента <line>, а <point B> к порту 'B' (для соединения компонентов просто нажмите мышкой на полуокружности с правой стороны компонента <point> и перетащите

курсор на другую полуокружность целевого объекта (A/B вход компонента <line>). Вы видите, что Rhino нарисовал отрезок между этими двумя точками.

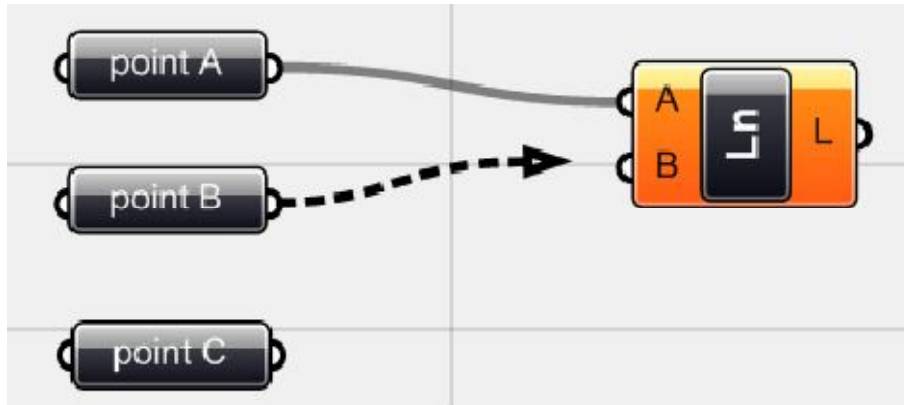


Рис. 5. Соединение компонента <point> с компонентом <line>, перетаскиванием от выхода <point B> ко входу <line>.

Теперь добавьте еще один компонент <line> для точек <point B> и <point C>. И еще раз для точек <point C> и <point A> добавьте компонент <line>. Да! Треугольник появился в окне просмотра Rhino.

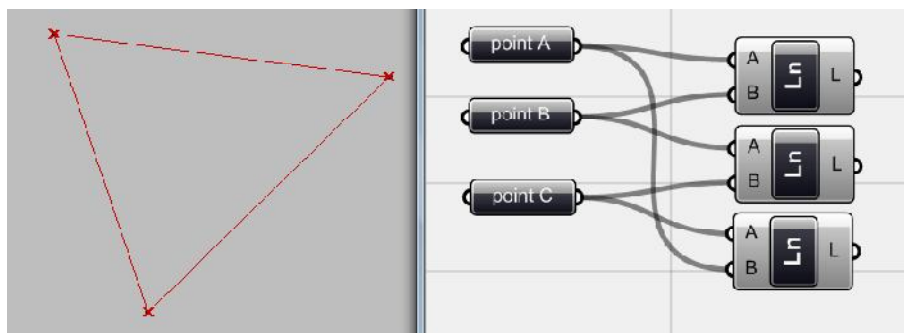


Рис. 6. Компонент <line> рисует линию между компонентами <point>. Как вы видите, любой компонент может быть использован несколько раз как источник данных для других действий.

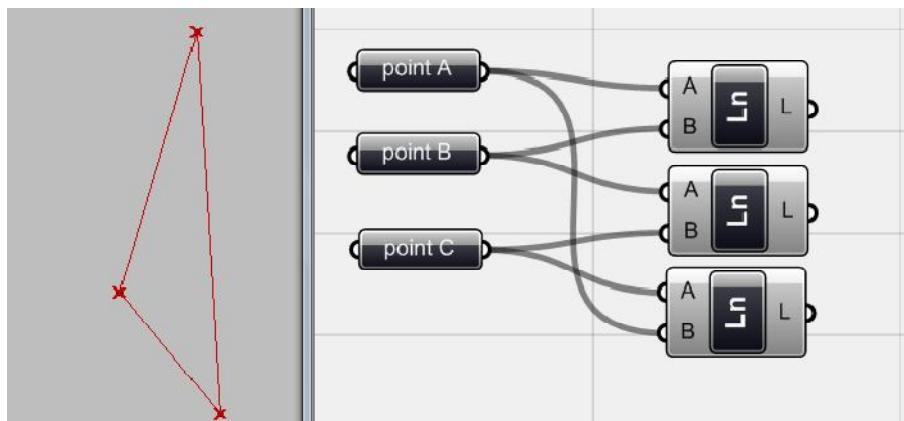


Рис. 7. Если теперь вы вручную измените расположение точек в окне Rhino, то расположение точек в Grasshopper (крестики - x) и результирующий треугольник тоже соответствующим образом изменятся, но линии между точками сохранятся.

Как видно из этого простого примера, ассоциативная (параметрическая) техника дает возможность манипулировать точками и треугольником, построенным по ним, без дальнейшей необходимости корректировать его. Таким образом, идея заключается в подготовке объектов (исходных данных алгоритма), задании отношений между объектами и добавлении других манипуляций применяемых к ним (функция алгоритма) и генерации дизайна (результат алгоритма). В дальнейшем мы будем придерживаться этой концепции при разработке нашего понимания алгоритмов.

Input/Output (Вход/ Выход)

Как упоминалось ранее, любой компонент в Grasshopper имеет вход и выход. Это означает, что компонент обрабатывает получаемые данные и возвращает обработанные данные обратно. Входы располагаются с левой стороны компонента, а выходы с правой. Данные, поступающие из любого источника, подключаются к входной секции компонента, а выход компонента является результатом его специфической функции.

Некоторые особенности этого вопроса вы можете узнать по адресу:

<http://en.wiki.mcneel.com/default.aspx/McNeel/ExplicitHistoryVolatileDataInheritance.html>

Вы должны знать, какого вида исходные данные вам нужны для какой-либо конкретной функции и что вы получите в результате. В дальнейшем мы будем говорить о разных видах данных, которые нужны для каждого компонента. Если задержать указатель мыши над любым входом/выходом компонента, то появится всплывающее окно, где вы можете увидеть имя данного порта, тип данных необходимых для компонента, перечень предопределенных данные или их отсутствие, и даже для чего он нужен.



Рис. 8. Всплывающее окно-подсказка появляется, если вы задержите указатель мыши над портом входа/выхода компонента.

Несколько соединений

Иногда вам нужно указать компоненту более чем один источник данных. Представьте себе, что в приведенном выше примере вы хотите нарисовать два отрезка из <point A> к <point B> и <point C>. Вы можете использовать два различных компонента <line> или вы можете использовать один компонент <line> и прикрепить обе точки B и C ко второй точке компонента <line>. Для этого необходимо зажать клавишу Shift, если вы хотите подключить второй источник данных к компоненту, в противном случае Grasshopper заменит его. Когда вы держите Shift,

стрелка мыши появляются в зеленых фигурных скобках с крошечным значком (+), в то время как обычно стрелка серая. Вы можете также использовать клавишу Ctrl, чтобы отключить один компонент из другого (как правило, вы можете отключить компонент от другого из контекстного меню). В этом случае фигурные скобки вокруг мыши будут красными с крошечным значком (-).

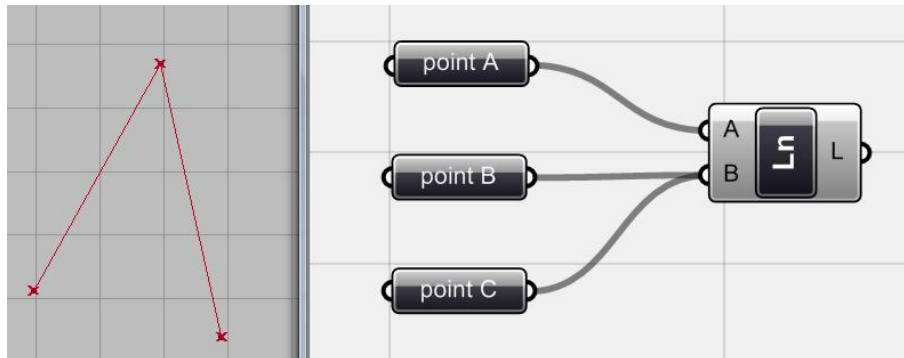


Рис. 9. Несколько соединений для одного компонента, при нажатой клавише Shift.

Цветовое кодирование

Существует система цветового кодирования внутри Grasshopper, которая показывает текущий рабочий статус компонента.

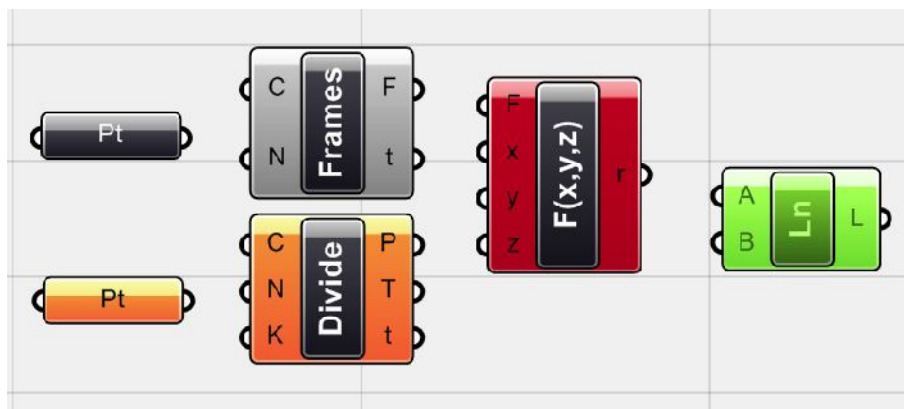


Рис. 10. Цветовое кодирование компонентов

Серый цвет компонента означает, что проблем нет, и данные введены правильно / компонент работает правильно. Оранжевый показывает предупреждение, и это означает, что есть, по крайней мере, одна проблема, которая должна быть решена, но компонент по-прежнему работает. Красный цвет компонента означает ошибку, и компонент не работает в этой ситуации. Источник ошибки должен быть найден и устранен для того, чтобы компонент работал должным образом. Вы можете найти первую помощь об источнике ошибки в контекстном меню компонента (контекстное меню > Runtime warning/error), а затем отыскивать исходные данные, чтобы найти причину ошибки. Зеленый цвет означает, что этот компонент выбран. Геометрия, связанная с этим компонентом также выделяется в окне просмотра Rhino (в противном случае вся геометрия, генерируемая в Grasshopper, отображается красным цветом).

Предпросмотр

У всех компонентов, которые создают объекты в Rhino, есть опция 'Preview' ("Предварительный просмотр") в меню. Мы можем использовать ее, чтобы скрыть или отобразить геометрию в рабочем окне. Если отключить предпросмотр (Hidden output), то компонент становится темно-серым. Опцию предварительного просмотра обычно используют, чтобы скрыть нежелательную геометрию, такую как базисные точки и линии в сложных моделях, чтобы избежать загромождения сцены. Данная опция в сложных моделях помогает обрабатывать данные быстрее, поэтому, пожалуйста, скрывайте свою базовую геометрию, когда не нужно чтобы она была видна.

2.3. Сопоставление данных

Для многих компонентов Grasshopper можно использовать перечень данных, а не только одно значение. Таким образом, вы можете задать список точек для компонента <line>, и по этому списку он отобразит несколько линий вместо одной. Таким образом, можно отрисовать сотни объектов с помощью всего одного компонента, если мы зададим необходимую информацию.

Посмотрите на этот пример:

Мы имеем два различных набора, в каждом по семь точек. Используем два компонента <point> и выберем 'set multiple points' для задания всех верхних точек в одном компоненте, а все нижние точки задаются вторым компонентом. Как видно, присоединив эти два набора точек к компоненту <line>, между ними появляются семь линий. Таким образом, вы можете генерировать несколько объектов с помощью каждого компонента.

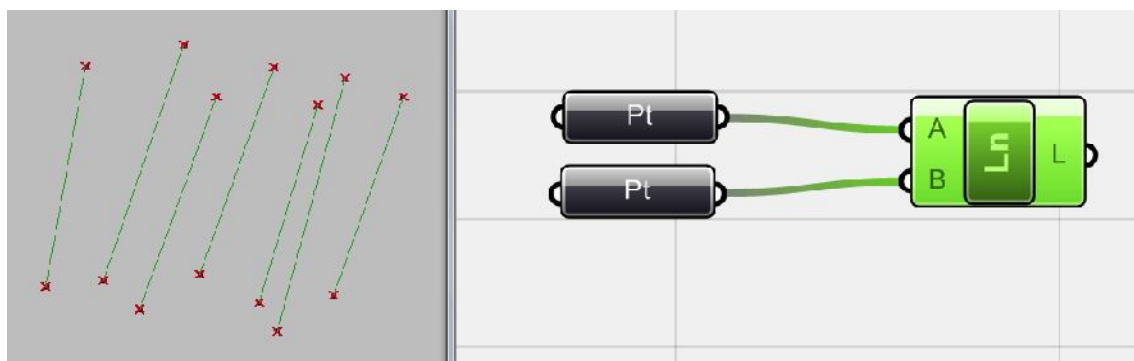


Рис. 11. Несколько наборов точек и генерация линий между ними.

Но что если количество точек (данных) в наборах будет разным?

В примере ниже имеем 7 точек в верхнем ряду и 10 точек в нижнем. Здесь мы должны разобраться в концепции управления данными в Grasshopper, которая называется 'Data matching' (Сопоставление данных). Если вы посмотрите на контекстное меню компонента, то увидите три опции называемые:

Shortest list

Longest list

Cross reference

Обратите внимание на различия между ними на рисунке (см. ниже).

Ясно, что **Shortest list** (кратчайший список) использует кратчайшие набор данных, чтобы отобразить линии. **Longest list** (длинный список) использует длинный набор данных, при этом элементы более короткого списка используются более одного раза. **Cross reference** (перекрестные ссылки) соединяет любые возможные пары точек из обоих списков друг с другом. Этот вариант требует очень много памяти и иногда для этого требуется значительное время на обновление изменений в сцене. Поскольку на рисунках все показано, то не будем описывать данные параметры более подробно. За дополнительной информацией обратитесь к следующей ссылке:

<http://en.wiki.mcneel.com/default.aspx/McNeel/ExplicitHistoryDataStreamMatchingAlgorithms.html>

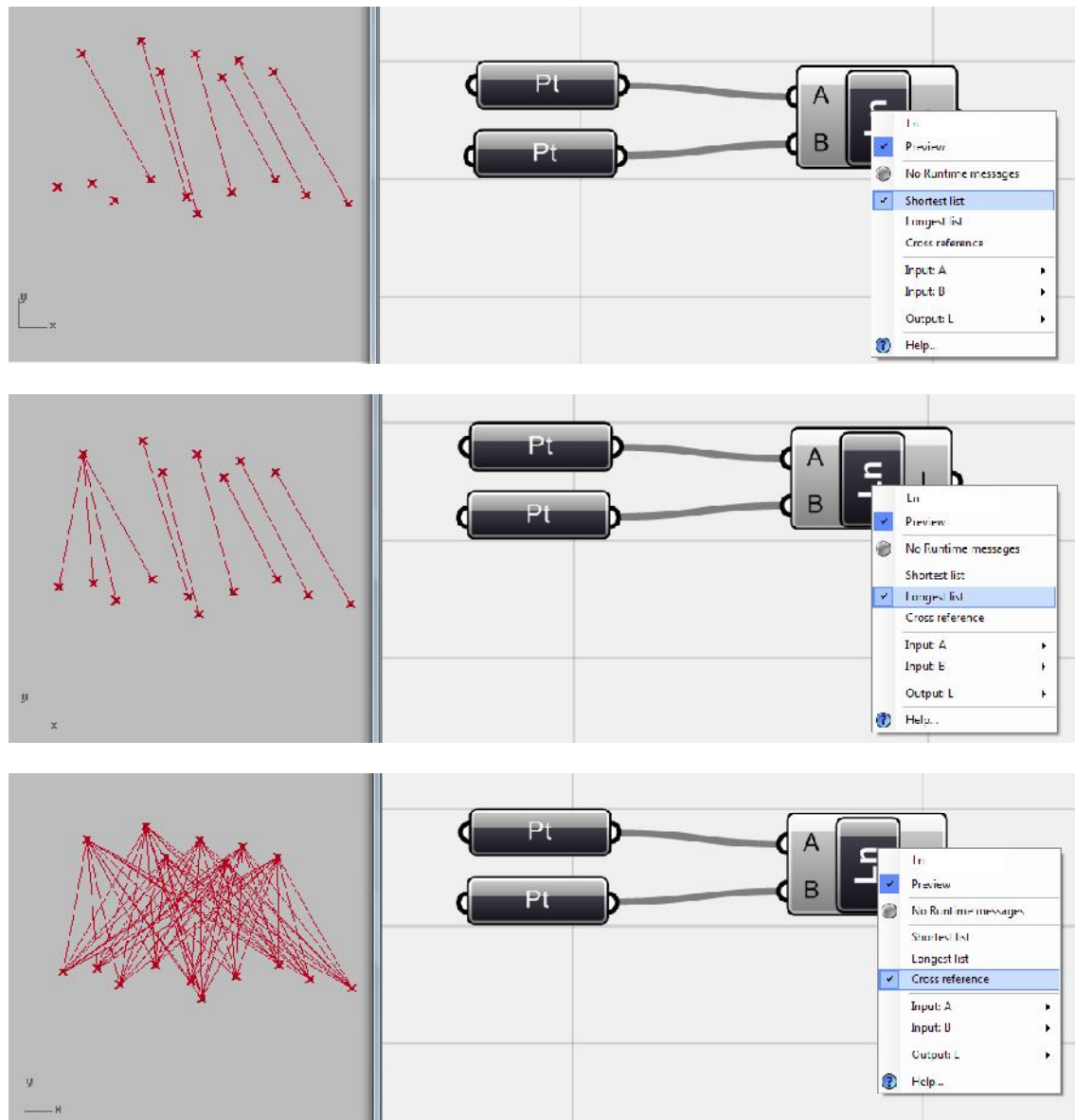


Рис. 12. Сопоставление данных A: shortest list, B: longest list and C: cross reference

2.4. Справка по компонентам (Контекстное меню)

Поскольку компонентов в Grasshopper очень много, то мы не сможем здесь рассказать про все компоненты сразу. Вы будете находить их и узнаете как их использовать постепенно, в

экспериментах. Я рекомендую вам поиграть, выбрать некоторые компоненты, перейти в контекстное меню компонента (щелкните правой кнопкой мыши) и почитать их Help (Справка), которую всегда полезно посмотреть. В справке обычно указано как этот компонент работает, какие данные ему требуются и что получается на выходе. Есть и другие полезные функции, в этом контекстном меню, которое мы будем обсуждать позже.

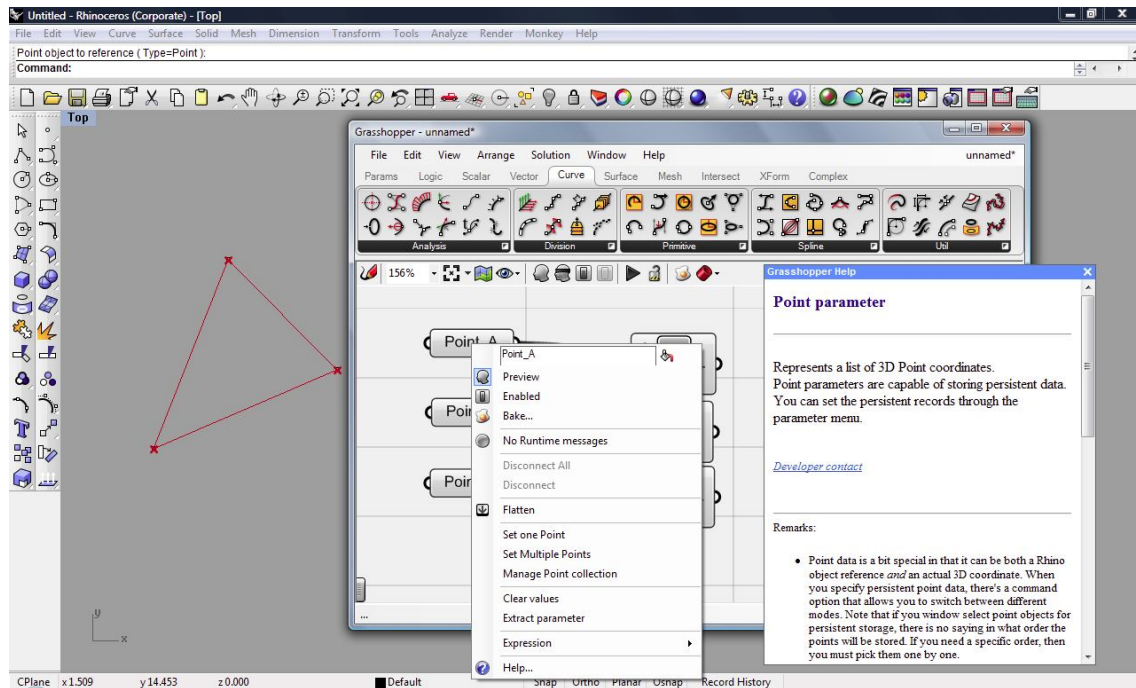


Рис. 13. Контекстное меню и раздел Справка компонента

2.5. Поиск/добавление компонентов

Если вы знаете название компонента, который вы хотите использовать, или если вы хотите найти его быстрее, чем переключая вкладки компонентов и панели, вы можете дважды щелкнуть на холсте и начать вводить имя компонента для добавления его на холст. Для тех, кто привык работать с клавиатурными сочетаниями, это хороший трюк!

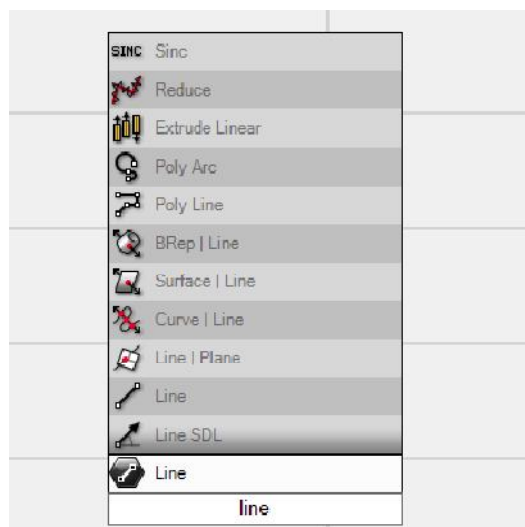


Рис. 14. Поиск компонента <line> во всплывающем меню компонентов, при двойном щелчке мышью на холсте и вводе имени компонента. Компонент будет добавлен на холст.