

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ ALGORITHMIC AND PROGRAMMING

УДК 004.925

А. В. Харланович, О. А. Новосельская
Белорусский государственный технологический университет

ПОСТРОЕНИЕ ФРАКТАЛЬНЫХ ДЕРЕВЬЕВ И ИХ ПРОГРАММНАЯ РЕАЛИЗАЦИЯ В 3DSMAX

В статье рассмотрены области применения теории фракталов, особенности их построения, а также современные программные средства, реализующие автоматизированное построение фрактальных структур. Основным направлением данного исследования является рассмотрение возможности внедрения теории фракталов в существующие программы трехмерной графики на примере наиболее популярной – 3DsMax. Особенность теории фракталов – свойство самоподобия – позволяет задавать топологическую структуру базового трехмерного объекта, что важно для внедрения сгенерированного элемента, например, в игровую графику и игровые движки. В статье показана возможность внедрения теории фракталов в процесс моделирования такого самоподобного объекта, как дерево Пифагора. Приведены алгоритмы создания фрактальных моделей средствами 3DSMax и внедренный плагин на основе скрипта, написанного на языке MaxScript. Отличительной особенностью алгоритма является возможность выбора в качестве базового элемента как стандартных объектов 3DsMax, так и сплайновых форм с наложенными модификаторами. Это позволяет расширить вариативность построенных моделей и снизить трудоемкость их разработки.

Ключевые слова: фрактальная графика, множество, сплайн, 3DsMax, MaxScript, дерево Пифагора.

Для цитирования: Харланович А. В., Новосельская О. А. Построение фрактальных деревьев и их программная реализация в 3DsMax // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. 2022. № 2 (260) С. 1–130.

A. V. Kharlanovich, O. A. Novoselskaya
Belarusian State Technological University

CONSTRUCTION OF FRACTAL TREES AND ITS SOFTWARE IMPLEMENTATION IN 3DSMAX

The article describes the areas of application of fractals theory. Main features of fractals construction are shown, as well as modern software tools that implement its automated construction. The main direction of this study is to consider the possibility of introducing the fractals theory into existing 3D graphics programs using the most popular one 3DsMax as an example. An attribute of fractals theory, i. e. the property of self-similarity, allows one to set the topological structure of the basic three-dimensional object. This fact is very important for the implementation of the generated element, for example, in game graphics and game engines. The article shows the possibility of involving the fractal theory into the process of modeling such a self-similar object as the Pythagorean tree. Algorithms for creating fractal models using 3DSMax and embedded Plugin with the help of Script written on the basis of MaxScript language are given. Before the algorithm is running, users can choose both standard 3DsMax objects and spline shapes with modifiers as a basic element of the final shape. This allows to expand the variability of the constructed models and reduce the complexity of its development.

Key words: Fractal Graphics, Set, Spline, 3DsMax, MaxScript, Pythagorean tree.

For citation: Kharlanovich A. V., Novoselskaya O. A. Construction of Fractal Trees and its Software Implementation in 3DsMax. *Proceedings of BSTU, issue 3, Physics and Mathematics. Informatics*, 2022, no. 2 (260), pp. 121–130 (In Russian).

Введение. Теория фракталов является одной из наиболее актуальных и стремительно развивающихся теорий, которые находят самое широкое применение в разных областях деятельности

человека. Использование фрактальных моделей позволило значительно продвинуться в решении различных практически значимых задач [1–3]. Интенсивное развитие теории фракталов не

только выдвигает новые вопросы, но и позволяет искать подходы к задачам, ранее сформулированным, но не получившим до настоящего времени удовлетворительных решений.

Фракталы – это сложные структуры, которые являются самоподобными и поэтому имеют одинаковые паттерны в любом масштабе. Фракталы могут быть узорами и фигурами. Располагая только внешним видом, оценка фрактальных свойств затруднена, а в большинстве случаев невозможна [4].

Основные свойства фрактальных множеств:

- имеют тонкую структуру, т. е. содержит произвольно малые масштабы;
- имеют форму самоподобия (приближенную или статистическую);
- фрактальная размерность больше топологической размерности;
- в большинстве случаев фрактал определяется рекурсивно.

Фракталы применяются для отображения объектов реального мира, которые имеют сложную форму, не поддающуюся описанию через привычную геометрию [5]. Имеются определенные ограничения на применение к природным объектам некоторых математических методов, работающих на абстрактных фракталах. Небольшая глубина иерархии природных фракталов затрудняет процедуру скейлинга и налагает ограничения на точность определения размерности. Это обстоятельство потребовало разработки таких робастных методов оценки размерности по данным наблюдений, которые корректно работают на конечной глубине иерархии масштабов и на ограниченном объеме данных. Разработанные методы имеют достаточно широкую область применения или работают с каким-либо одним классом фрактальных объектов. Например, это позволяет анализировать данные, так как заранее неизвестно, есть скейлинг или нет, и если есть, то к какому типу фракталов относится исследуемый объект [6].

Размерность объекта показывает, по какому закону растет его внутренняя область. Аналогичным образом возрастает «объем» фрактала с ростом его размеров, но его размерность – величина не целая, а дробная [7]. Поэтому граница фрактальной фигуры не линия: при большом увеличении становится видно, что она размыта и состоит из спиралей и завитков, повторяющихся в малом масштабе саму фигуру. В природных фракталах отсутствует точное самоподобие – вместо строго геометрического самоподобия элементов фрактала наблюдается лишь их топологическое или статистическое самоподобие. Топологическое самоподобие позволяет рассматривать фрактал как элемент структуры изображения.

К преимуществам использования фракталов для построения изображений относится то, что они занимают меньше места в памяти, их можно легко

масштабировать вплоть до бесконечности [8], избегая при этом пикселизации, а также, наоборот, можно сжимать готовые файлы.

Фрактальная графика, так же как векторная и трехмерная, является вычисляемой. Ее главное отличие в том, что изображение строится по уравнению или системе уравнений [9, 10]. Поэтому в памяти компьютера для выполнения всех вычислений ничего кроме формулы хранить не требуется. Для того чтобы представить все многообразие фракталов, удобно прибегнуть к их классификации. Существуют геометрические, динамические и стохастические фракталы [6].

Построение геометрических фракталов происходит поэтапно. Сначала изображается основа. Затем некоторые части основы заменяются на фрагмент. На каждом следующем этапе части уже построенной фигуры, аналогичные заменяемым частям основы, вновь заменяются на фрагмент, взятый в подходящем масштабе. Всякий раз масштаб уменьшается. Когда изменения становятся визуально незаметными, считают, что построенная фигура хорошо приближает фрактал и дает представление о его форме. Для получения самого фрактала нужно бесконечное число этапов. Меняя основу и фрагмент, можно получить много разных геометрических фракталов.

Основная часть. Фракталы находят все большее применение в науке по причине того, что они описывают реальный мир иногда даже лучше, чем традиционная физика или математика.

Сферы применения фракталов [11]:

- в радиотехнике используется фрактальная геометрия при проектировании антенных устройств;
- в телекоммуникации в сфере сетевых технологий проведено множество исследований, представляющих самоподобие трафика, передаваемого по разного рода сетям;
- децентрализованные сети. Система назначения IP-адресов в сети Netsukuku использует принцип фрактального сжатия информации для компактного сохранения информации об узлах сети;
- в физике и других естественных науках фракталы возникают при моделировании нелинейных процессов, таких как пламя, турбулентное течение жидкости, облака, сложные процессы диффузии – адсорбции и т. п. При моделировании пористых материалов (в нефтехимии) также применяются фракталы;
- в трейдинге они используются для анализа состояния биржевых рынков. Фракталы рынка являются одним из индикаторов в торговой системе Била Вильямса;
- фракталы как элементы визуализации и спецэффектов [12]. Они притягивают и завораживают своей красотой и бесконечностью. Именно поэтому (но и не только) их очень часто

используют для создания различного рода визуализаций, видеоинсталляций, спецэффектов в компьютерной графике и т. д.

Область применения фракталов обширна и не ограничивается лишь их прямыми природными аналогами. Ими можно описывать сосуды кровеносной системы, ветвление бронхов, нейронную сеть и т. д.

Фрактальные технологии применяются в 3D-моделировании [13] и в 3D-печати, поскольку с их помощью можно создавать сложные объемные структуры.

В компьютерной графике фракталы применяются для построения реалистичных изображений природных объектов, таких как поверхности морей, деревья, кусты, горные ландшафты и т. д. [14, 15]. Применять фрактальные изображения можно в самых разных сферах, начиная от создания обычных текстур и фоновых изображений и заканчивая фантастическими ландшафтами для компьютерных игр или книжных иллюстраций. Создаются подобные фрактальные шедевры (как и векторные) путем математических расчетов, но, в отличие от векторной графики, базовым элементом фрактальной графики является сама математическая формула – это означает, что никаких объектов в памяти компьютера не хранится, и изображение (как бы ни было оно замысловато) строится исключительно на основе уравнений.

В настоящей статье рассматривается область применения фракталов – компьютерная графика. Фракталы нашли применение в компьютерном дизайне, в мультфильмах и кино. Многие крупные достижения науки о фракталах стали возможны только с использованием методов вычислительной математики. Компьютерная графика воссоздает на экране монитора бесконечное разнообразие фрактальных форм – красоту пейзажей, ландшафтов, облаков.

Для создания 3D-моделей фракталов можно использовать различные программное обеспечение. Существуют специализированные программы [16–24], имеющие встроенные алгоритмы, реализующие основные этапы построения. Продукты разноплановы. Одни представляют собой генераторы фракталов в чистом виде, другие, помимо построения фракталов, обеспечивают более широкие возможности построения и визуализации изображений, включая внедрение трехмерных объектов, расширенный контроль настроек освещения, внедрение теней, размытие краев и пр. Имеющиеся программные средства направлены на построение фрактальных структур абстрактной формы, не позволяют корректно управлять топологией трехмерных объектов. Основной функционал включает выбор алгоритма генерации фрактала, увеличение / уменьшение фрагмента изображения, изменение цветовой гаммы, сохранение

полученного изображения. Также некоторые программы включают возможность вводить собственные формулы и осуществлять дополнительный контроль, такой как фильтрация полученного изображения и генерирование фрактальной анимации. Однако ошибки топологической структуры могут сказаться при отображении анимации, приводят к проблемам при создании видеофильмов, компьютерных игр и т. п.

Одной из задач проводимого исследования является рассмотрение математической основы теории фракталов и изучение способов их внедрения в программу трехмерной графики 3DsMax с последующим управлением топологической структурой. Практические особенности задачи построения динамических трехмерных объектов определяют такие требования к алгоритмам формирования фракталов, как высокое быстродействие и низкая ресурсоемкость, а также возможность отображения многообразия фрактальной структуры, свойствами которой являются: нерегулярность, квазипериодичность и высокая масштабная инвариантность. Имеется практическая необходимость построения фрактала, удовлетворяющая следующим функциональным критериям, таким как приемлемая алгоритмическая сложность, древовидная структура, высокая фрактальная размерность, регулируемая детализация.

В программе 3DsMax имеется несколько вариантов построения фракталов [23]:

1) построение фракталов «в ручном режиме». То есть данный метод не предусматривает автоматизации действий. Например, для создания пирамиды Серпинского необходимо построить пирамиду и поэтапно вырезать из нее лишние части;

2) разработка специального скрипта, позволяющего полностью автоматизировать процесс;

3) третий способ представляет собой «полуавтоматический режим» – основа будущего фрактала создается вручную, а дальнейшие действия выполняет программа (скрипт), составленная с помощью специального плагина Para 3D.

Существуют подходы для генерации 2D-деревьев [25, 26]. В них применяют так называемые L-системы [27], суть которых заключается в итерационном повторении простого алгоритма рисования узлов. В итоге получается 2D-фрактал. Однако 3D-деревья в рамках вышеперечисленных подходов не рассматриваются. В 3DsMax существует возможность строить деревья и растительность из расширенной библиотеки AES Extended. Деревья представляют собой сгенерированные объекты, полигональная структура которых не поддается модификации. Деревья занимают большой объем памяти, впоследствии они отображаются в виде обобщенных форм. Поэтому требуется разработка собственного решения.

Для генерации простейшего фрактала и переноса алгоритма на построение фрактального дерева рассмотрена кривая Коха как типичный геометрический фрактал. Процесс ее построения выглядит следующим образом: берем единичный отрезок, разделяем на три равные части и заменяем средний интервал равносторонним треугольником без этого сегмента. В результате образуется ломаная, состоящая из четырех звеньев длиной $1/3$. На следующем шаге повторяем операцию для каждого из четырех получившихся звеньев и т. д. Предельная кривая и есть кривая Коха.

Алгоритм построения фрактала заключается в следующем:

1) выбор аксиомы (начального набора символов). Например, для снежинки Коха это будет $f++f++f$;

2) создание правил для собственно алгоритма генерации, в данном случае это будет закон $f \rightarrow f-f++f-f$, а также скрипта, выполняющего построение фрактала по этим правилам.

В данном алгоритме обозначения включают:

- F (переместиться вперед на один шаг, прорисовывая след);
- f (переместиться вперед на один шаг, не прорисовывая след);
- $[$ (открыть ветвь (команда ветвления));
- $]$ (закрыть ветвь (команда ветвления));
- $+$ (увеличить угол α на величину θ);
- $-$ (уменьшить угол α на величину θ).

Поскольку фракталы позволяют представить сложные неевклидовы объекты, образы которых похожи на природные, в данном исследовании было принято решение о построении дерева Пифагора с помощью внедренного в программу трехмерной графики скрипта (листинг 1). Дерево Пифагора – разновидность фрактала, основанная на фигуре, известной как «Пифагоровы штаны» [28–30]. Если в классическом дереве Пифагора угол равен 45° , то также можно построить и обобщенное дерево Пифагора при использовании других углов. Такое дерево часто называют обдуваемое ветром дерево Пифагора. Если изображать только отрезки, соединяющие каким-либо образом выбранные «центры» треугольников, то получается обнаженное дерево Пифагора.

Алгоритм, представленный на рис. 1, состоит из формирования вертикального отрезка, рекурсивного построения из верхнего конца этого отрезка еще двух под определенными углами и вызывания функции создания двух последующих отрезков для каждой ветви дерева.

Реализуется функция (листинг 1) замены строки для создания правил собственно алгоритма генерации и функция (листинг 2) построения фрактала по вышеуказанному правилу.

Реализация приведенного в листингах 1, 2 скрипта показана на рис. 2.

Можно реализовать построение дерева и другим способом. Для этого скрипт будет содержать функцию с передаваемыми параметрами высоты и процент разветвления дерева, выбора родителя для каждого последующего элемента дерева (листинг 3). При этом есть возможность непосредственно в скрипте указать формы, которые будут включать деревья, задавать их свойства и применять модификаторы.

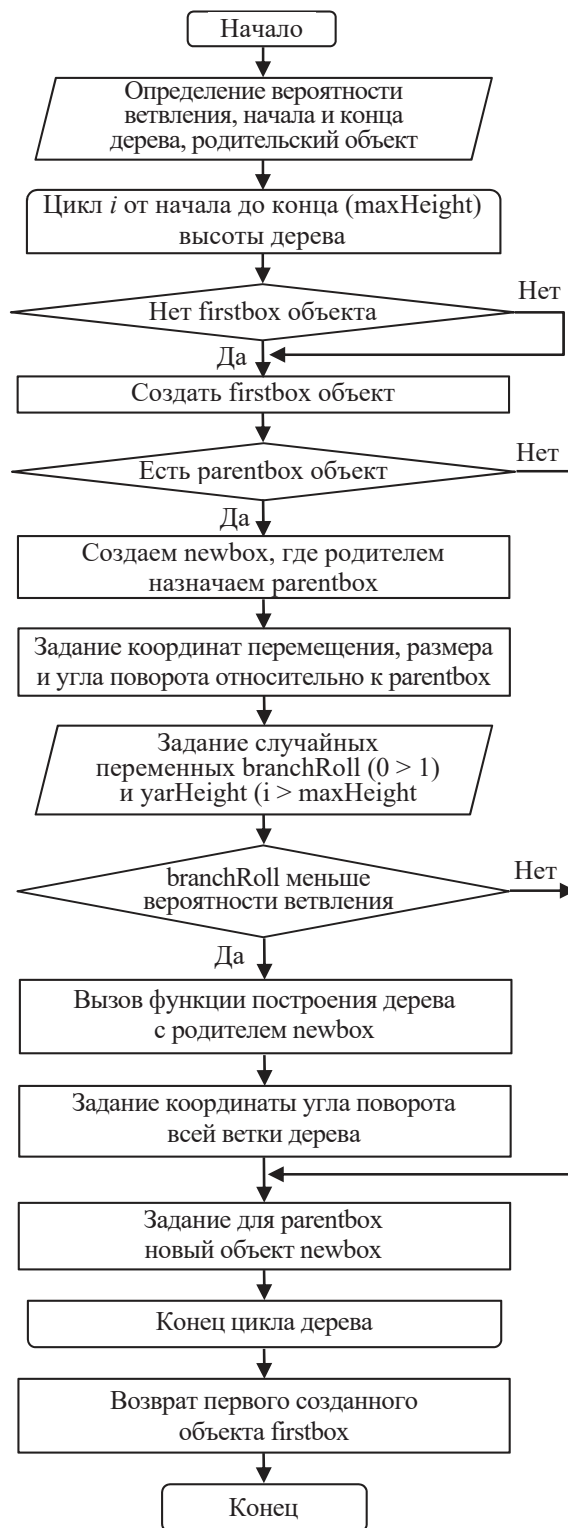


Рис. 1. Алгоритм построения дерева

```

fn gens str s n= ( str1=copy str  count=0;
  for j=1 to n do ( for i=1 to str.count
do ( count+=1
  if str[i]=="f" then ( str1=replace
str1 count 1 s
  count+=s.count-1 ) )
  str = copy str1  count=0; )
return str1 )

```

Листинг 1. Код функции замены строки

```

fn CreateFrac str fi theta a= (
  VertexArray=#()
  ArrayC=#()
  struct p(fi,a,theta,X,Y,Z)
  append VertexArray [0,0,0]
  X0=0 Y0=0 X=0 Y=0 Z=0 count=1
  for i=1 to str.count do ( if str[i]=="f"
then (
  ss=SplineShape pos:[X,Y,Z]
  s11 = AddNewSpline ss
  addKnot ss 1 #corner #line [X,Y,Z]
  X+=a*(cos theta)*(cos fi)
  Y+=a*(cos theta)*(sin fi)
  Z+=a*(sin theta)
  addKnot ss 1 #corner #line [X,Y,Z]
  updateshape ss  count+=1 )
  if str[i]=="+" then ( fi+=random 5 30 )
  if str[i]=="-" then ( fi-=random 5 30 )
  if str[i]=="&" then ( theta+=random 5 30
)
  if str[i]=="^" then ( theta-=random 5 30
)
  if str[i]=="[" then ( f1=p fi a theta X
Y Z  append ArrayC f1 )
  if str[i]=="]" then ( count1=Ar-
rayC.count
  if count1>0 then (
  f1=ArrayC[count1]
  fi=f1.fi a=f1.a
  X=f1.X Y=f1.Y Z=f1.Z
  theta=f1.theta;
  deleteItem ArrayC count1
) ) ) )

```

Листинг 2. Код функции генерации фрактала

В результате выполнения скрипта строится дерево из самоподобных элементов, которое можно использовать для стилизованных сцен, в анимации.

Для учета всех параметров скрипта и работы с ним на уровне пользователя целесообразным является внедрение плагина как расширения 3Ds Max.

Существует огромное количество плагинов для 3DsMax. Их функционал определяется разработчиками и предназначен для автоматизации процесса моделирования сложных 3D-форм. Разработка собственного плагина для генерации фрактального дерева включала в себя следующие этапы.

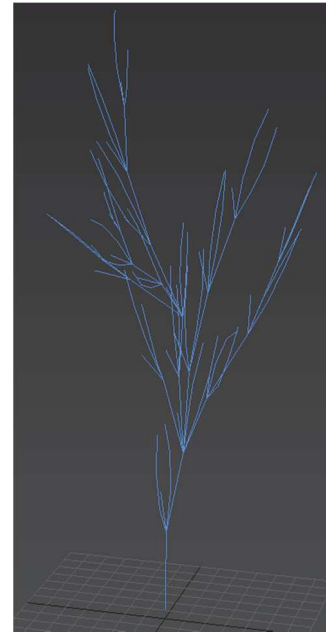


Рис. 2. Результат выполнения скрипта построения дерева Пифагора

```

myTree = tree 0 15 branchingChance:0.35
/* Функция построения */
function tree currentHeight maxHeight
parentbox:undefined branchingChance:0.15 =
(
  firstbox = undefined
  for i = currentHeight to maxHeight do (
    newbox = Sphere Segments:20
    ...
    if parentbox != undefined then (
      newbox.parent = parentbox
      in coordsys parent new-
box.scale=[0.9,0.9,0.9]
      in coordsys parent newbox.pos=[0, 0,
40]
      ...
      about newbox.parent rotate newbox(
        eulerAngles rotX rotY rotZ )
      branchRoll = random 0.0 1.0
      if branchRoll <= branchingChance
then (
        branchbox = tree i maxHeight
parentbox:newbox ... ) )
      parentbox = newbox )
  firstbox )

```

Листинг 3. Код функции генерации дерева

Для того чтобы разработанный скрипт добавить в данный список, надо определить в скрипте утилиту. Во вкладке Utilities параметр Utilities отображает список доступных скриптовых утилит. Сценарная служебная панель создается с использованием конструкции определения «utility» в MaxScript. Синтаксис верхнего уровня представлен в листинге 4.

```
utility <var_name> <description_string>
(
  <utility_body>
)
```

Листинг 4. Синтаксис плагина

Далее оформляется визуальная часть, т. е. вставляются элементы управления пользовательского интерфейса (radioButtons, spinner, button и т. д.) для управления значениями построения фрактальной фигуры. В качестве передачи значений и обработки построения берутся обработчики событий, которые также пишутся в теле utility. В результате разрабатывается один файл с вышеперечисленным содержанием и сохраняется с расширением «.ms». Для того чтобы разработанный плагин появился в списке, требуется сохранить в формате «.ms» и папке установки 3DsMax / Plugins. Выбранный файл сценария открывается в модальном окне командной панели. Файлы дополнительных модулей обязательно должны находиться в папке Plugins, иначе 3DsMax их не увидит.

Разработанный плагин сформирован в виде модального окна (рис. 3), в котором пользователь может выбрать форму родительской фигуры, ее размер и вероятность ветвления. Наложение модификатора обеспечивает видоизменение родительского объекта, а параметры фрактала позволяют управлять итоговой формой всего фрактального дерева.

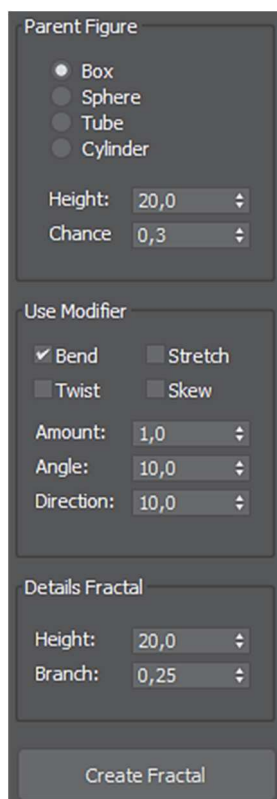


Рис. 3. Разработанное окно с плагином Fractal

Результат выполнения плагина показан на рис. 4. Поскольку за основу взят объект Box, не содержащий внутренних подразбиений структуры, то результат наложения модификатора Bend не оказывает влияния на его структуру. Наличие подразбиений позволит видоизменить базовый объект и построенное на его основе дерево.

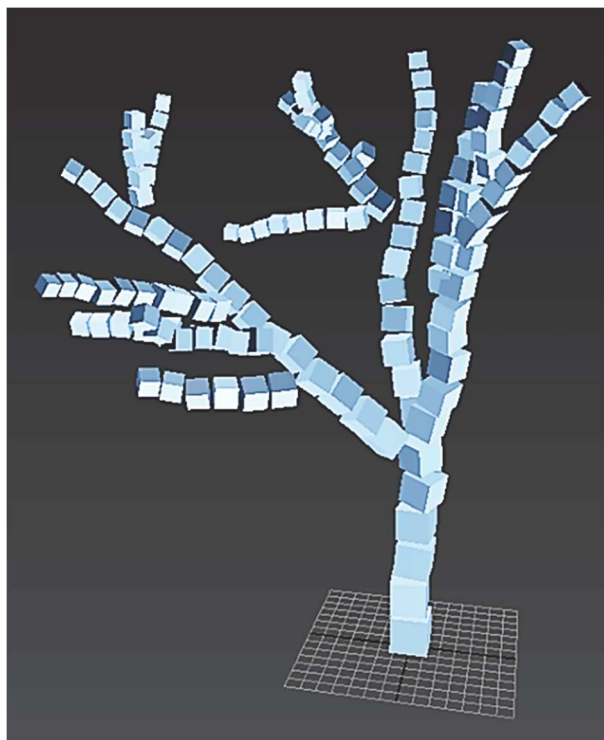


Рис. 4. Результат построения фрактального дерева

С позиции пользователя плагин, в частности, решает следующую задачу: создает фрактальный объект заданного размера, формы, уровня разбиения и применяет выбранный модификатор после нажатия на кнопку «Create Fractal». В результате можно управлять всеми необходимыми параметрами результирующих объектов, независимо от их форм и типа (геометрические формы или плоские кривые), что особенно важно для построения 3D-моделей с заданным уровнем детализации. Пользователь самостоятельно может выбрать основной элемент, из которого будет строиться фрактальное дерево, задавать его топологию и размеры.

Использование модификаторов в сочетании с алгоритмами построения фрактальных деревьев позволяет добиться управляемой реалистичностью изображений с возможностью изменения топологической структуры. В результате появляется большое количество разных инструментов, опций, приемов и способов моделирования, создания и редактирования текстур, работы над реалистичностью изображения.

Заключение. Фрактал можно взять за основу для формирования разветвленных пространственных структур, например деревьев, так как фрактальные структуры при различном увеличении не претерпевают в среднем значительных изменений, имеют ветвления, рассматривают нерегулярные, сильно изломанные, изрезанные объекты. Для подразделения геометрической структуры объекта целесообразно использовать алгебраические фракталы на основе L-систем. Для построения сложных трехмерных объектов требуется разработка собственных решений, интегрируемых в программное обеспечение для трехмерной графики с возможностью управления топологией.

После исследования готовых программных средств было решено внедрить фрактальную структуру в наиболее часто используемую программу 3D-графики – 3DsMax – с использованием языка программирования MaxScript и написать код по построению простейшего фрактала. Так как фракталы самоподобны, то наилучшим способом их построения является использование рекурсивного метода. Сравнение реализуемых фрактальных деревьев с помощью AEC Extended Foliage и разработанного плагина Fractal показано на рис. 5.

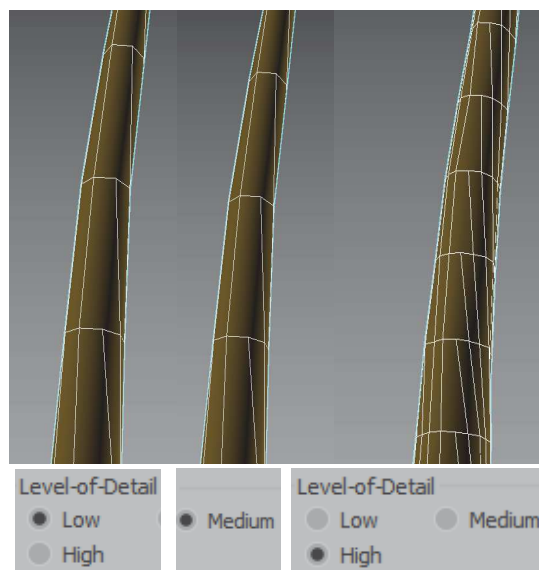
Проанализировав топологию полученных объектов, можно заметить, что при создании фрактальных деревьев встроенными возможностями 3DsMax не имеет различий при изменении уровня детализации дерева от Low к Medium, сами полигоны строятся по принципу четырехгранной формы, но при этом имеют непропорциональное соотношение, которым невозможно управлять без дополнительного наложения топологического модификатора, что является существенным недостатком в моделировании сцен для игровой графики.

Разработанный плагин дает возможность непосредственно управлять количеством подразбиений и соотношением сторон в полигональной сетке. Плагин также позволяет выбирать самоподобный элемент и управлять его топологией в зависимости от топологической размерности фрактала.

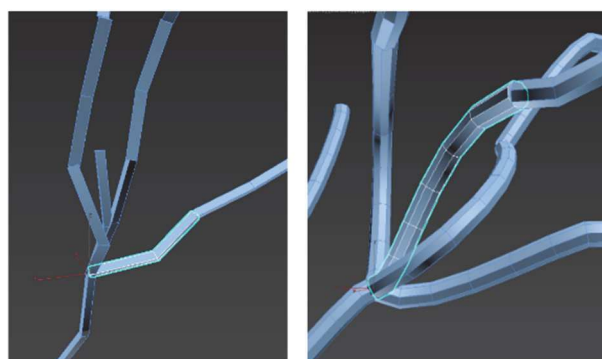
Таким образом, можно сделать вывод, что для оптимизации процесса моделирования построение сложных природных 3D-структур целесообразным является внедрение фрактальной графики совместно со специальными плагинами. При этом важно, чтобы построенный фрактал имел возможность трансформации, стилизации для низкополигональных сцен.

Список литературы

1. Щербаков Н. В., Гершова В. В. Удивительный фрактал // Россия и ВТО: экономические, правовые и социальные аспекты: сб. ст. участников IV Междунар. науч. студенческого конгресса. М., 2013. С. 1755–1758.
2. Качарава А. С. Живая математика: практическое применение фракталов в жизни // Международный школьный научный вестник. 2019. № 5. С. 59–67.



a



б

Рис. 5. Сравнение топологии фрактальных деревьев:
a – в программе 3DsMax;
б – с помощью разработанного плагина Fractal

Для разработки собственного плагина потребовалось задействовать три различные функции, определить три блока группировки `groupBox`, которые отвечают за каждую функцию. Одна функция – для кнопки, по нажатию на которую вызывается вторая функция. Она очищает рабочее пространство, определяет требуемые переменные до вызова и выполняет функцию построения фрактала. Третья функция отвечает за само построение. В итоге предложен способ представления сложных неевклидовых 3D-объектов с управляемой топологией, образы которых похожи на природные.