Система автоматизированного создания интерактивных параметрических чертежей для электронных учебников геометрии

С.С. Крылов, И.А. Моргунов
Московский Авиационный Институт
(государственный технический университет)
Москва, Россия
pmf_priem@mai.ru, merl@pisem.net

Аннотация

В данной статье описаны разработанные авторами язык программирования с геометрической семантикой, интерпретатор, среда программирования и средства визуализации.
В статье рассматриваются представление геометрической сцены, возможности языка, среды программирования и средств визуализации.

***Ключевые слова**: визуализация, язык программирования, интерпретатор, среда программирования, геометрический чертеж.*

1. ВВЕДЕНИЕ

При обучении математике, особенно геометрии, одним из основных видов учебной деятельности является решение задач, для которых существенны наглядные представления. Таким образом, как только мы применяем дистанционное и вообще компьютерное обучение, появляется необходимость в удобном инструменте для создания наглядных чертежей и механизме визуализации построений.
Обычный чертеж дает недостаточное представление о структуре объектов и сцены, тогда как поэтапное построение чертежа наглядно демонстрирует объекты, их взаимосвязи и свойства. Кроме того, для стереометрии весьма полезно иметь возможность «облета» сцены, для получения наилучшего представления о взаимном расположении объектов в пространстве.

2. ТРЕБОВАНИЯ К СИСТЕМЕ

Рассмотрим основные требования, которые мы выдвигаем к системе создания и отображения интерактивных чертежей.
1. **Возможность процедурного создания параметризованных геометрических сцен с заданной топологией.** Язык должен располагать необходимым набором встроенных функций. Этот набор функций должен давать возможность не только отображения объектов, но и манипулирования ими, а также предоставлять средства для оформления сцен (названия объектов, метки, надписи).
Параметризация геометрических объектов относительно друг друга позволяет описать связи между объектами сцены. При изменении положения объектов или их геометрических размеров, чертеж должен измениться, сохранив наложенные связи.
2. **Возможность аффинных преобразований чертежа.** Особенно это важно для стереометрии, где часто необходимо иметь возможность смотреть на чертеж с разных точек, чтобы получить более полное представление о взаимном расположении объектов в пространстве.
3. **Простота языка.** Основная группа людей, кто будет заниматься написанием программ на данном языке – методисты и преподаватели, т.е. люди часто далекие от программирования. Поэтому синтаксис и семантика языка должны быть максимально просты, чтобы облегчить им процесс создания программ.
4. **Возможность выполнения программы в интернет-браузере на компьютере клиента.** Транслятор данного языка должен функционировать в интернет-браузере. Благодаря этому упрощается использование языка в процессе обучения. Система не должна требовать специализированных интерфейсов, чтобы облегчить переносимость.
5. **Возможность интеграции в систему дистанционного обучения.** Транслятор должен легко встраиваться в оболочку обучающей системы. Таким образом, должна быть реализована удобная и простая среда программирования.
6. **Возможность пошагового выполнения программы.** Для лучшего понимания материала учащимся часто полезно наблюдать процесс пошагового создания чертежа. Это важно не только для задач на построение, но и для чертежей с большим количеством вспомогательных построений.
7. **Возможность расширения языка за счет написания и использования новых функций и организации их в библиотеки.**
8. **Режим отладки.** Для прикладного программиста необходим режим отладки, в котором он мог бы проследить процесс построения, найти и исправить логические ошибки в коде программы.

3. СУЩЕСТВУЮЩИЕ ГЕОМЕТРИЧЕСКИЕ СИСТЕМЫ

Рассмотрим несколько типичных представителей семейств систем с развитой поддержкой геометрического моделирования

3.1 Системы обучения геометрии

В данный момент число разработанных систем обучения геометрии невелико, и мы рассмотрим наиболее популярные из них.

3.1.1 «Живая геометрия», «Geometer's Sketchpad».

Программа «Живая Геометрия» [1] — эффективное средство для широкого спектра пользователей от учеников 5-го класса до студентов вуза. В основном она рассчитана на поддержку школьного курса геометрии и алгебры. «Живая Геометрия» обеспечивает работу с динамическими геометрическими сценами, поддержку алгебры, тригонометрии, приближенных вычислений и расчетов. Визуальный метод «Живой Геометрии» позволяет младшим ученикам приобретать необходимый опыт манипуляции математическими объектами. С помощью графического редактора Sketchpad учащиеся могут создать объект, а затем изучить его

математические свойства. «Живая геометрия» – русская версия популярной американской обучающей программы по геометрии «Geometer's Sketchpad» [2], разработанной фирмой Key Curriculum Press.

В программе существует возможность написания геометрических скриптов. Скрипты предоставляют механизм выделения построений в подпрограммы. Язык описания близок к языку геометрических текстов.

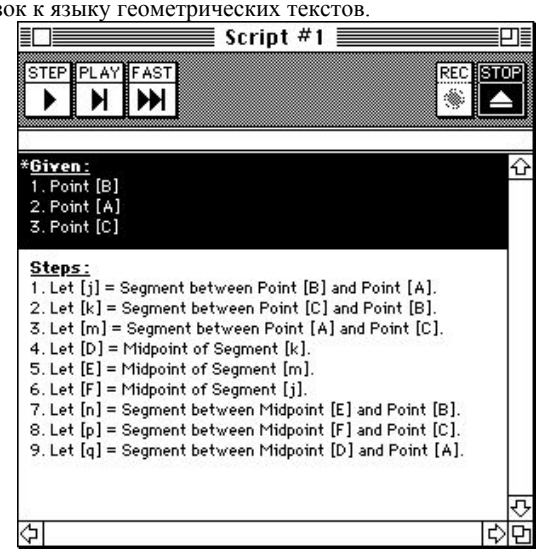

Рис. 1. Пример скрипта в «Geometer's Sketchpad»

Из приведенного примера видно, что язык для написания скриптов достаточно громоздкий. Такое описание конечно более понятно для учащихся, но для разработчика такой язык синтаксически избыточен и недостаточно строго формализован.

Главным недостатком этого продукта является то, что он реализует только планиметрические построения. Программа не работает со стереометрическими чертежами, то есть не охватывает существенную часть курса геометрии.

Также следует отметить, что использование чертежей данного продукта в обучающих системах существенно затруднено, так он не предоставляет открытых интерфейсов.

3.1.2 «Конструктивная геометрия»

«Конструктивная Геометрия» – это учебная программа, охватывающая большую часть курса геометрии, преподаваемой в средних школах, классах с углубленным изучением математики и высших учебных заведениях.

В программе реализованы средства для аффинных преобразований чертежа, а также реализована возможность создания и планиметрических, и стереометрических чертежей.

Основной недостаток этой системы в том, что в ней отсутствует возможность прикладного программирования. Система позволяет только интерактивное взаимодействие с пользователем, что существенно осложняет процесс построения сложного чертежа и делает невозможным автоматизированное создание чертежей. Таким образом, эта программа не обеспечивает должного уровня автоматизации подготовки учебно-методических материалов.

3.2 CAD/CAM системы

Мощными средствами геометрического моделирования обладают CAD/CAM системы, такие как Pro/Engineer, Unigraphics [5], КОМПАС, SolidWorks [4] и т.д. Современные программы автоматизированного проектирования предназначены не только для создания чертежей – они обладают гораздо более широкими возможностями и используются практически во всех отраслях техники. С их помощью можно создавать модели различных конструкций и проверять свойства модели на компьютере до начала производства, что существенно снижает число ошибок проектирования.

Данные системы очень дороги и обладают излишними, для системы обучения, возможностями. Кроме того, они крайне сложны и требуют определенных навыков у разработчика. Использование их в обучении геометрии нецелесообразно.

3.2.1 Язык GDL. Система автоматизированного проектирования «ArchiCAD».

Язык программирования GDL [3] разработан фирмой Graphisoft.

GDL(Geometric Description Language — Язык описания геометрических форм) — это встроенный в «ArchiCAD» язык программирования, подобный BASIC. Он позволяет писать программы, описывающие трехмерные объекты и соответствующие им на плане двумерные символы. В зависимости от указанных пользователем параметров программа может создавать различные объекты и символы.

Созданные с помощью GDL объекты можно размещать как на двумерных чертежах планов и разрезов, так и в трехмерном пространстве. В дальнейшем эти GDL-объекты автоматически отобразятся на всех видах.

Язык GDL, несомненно, мощный инструмент для создания как двух мерных, так и трехмерных сцен, но он является проблемно-ориентированным прикладным языком программирования в области архитектурного проектирования, далекой от обучения геометрии.

3.2.2 Язык MAXScript. Система трехмерного моделирования 3D Studio Max.

3D Studio Max дает возможность конструирования сложных трехмерных фотореалистичных сцен из большого набора примитивов и анимированных последовательностей.

Язык «MAXScript», встроенный в 3DSMax, позволяет создавать геометрические сцены, но большим недостатком этого языка, с точки зрения поставленной проблемы является его сложность для потенциальных пользователей.

Прикладные языки систем проектирования обладают геометрической семантикой, что делает их близкими к рассматриваемой проблеме. В связи с этим опыт их изучения полезен, но они являются составной частью дорогостоящих программных продуктов, ориентированных на решение совершенно других задач.

Рассмотрев данные языки, следует заметить, что в языке должна присутствовать возможность указания имени управляющего объекта при создании зависимого.

3.3 X3D

Существуют технологии для описания трехмерных сцен в среде интернет. Одной из таких технологий является X3D [6]. X3D является развитием языка моделирования виртуальной реальности (VRML) первоначального ISO стандарта для трехмерной графики используемой в Интернете (ISO/IEC 14772).

X3D – наиболее близкое решение из существующих аналогов. Но для получения изображений отвечающих требованиям, предъявляемым к стереометрическим чертежам, например, автоматическая прорисовка пунктиром невидимых линий потребовалось бы реализовывать собственную систему воспроизведения X3D-файлов, что весьма трудоемко. В скриптах X3D нет прямой возможности

параметризации создаваемых объектов сцены. Кроме того, сам язык описания достаточно громоздок и непрост, что существенно усложняет процесс создания интерактивных чертежей.

Существуют и другие системы для создания интерактивных чертежей, но мы ограничимся рассмотренными, так как другие системы близки какой-то из рассмотренных и обладают каким-то из недостатков:

- Отсутствие языка прикладного программирования, что очень сильно усложняет процесс автоматизации создания учебно-методических материалов, или сложность языка, что требует от методиста-разработчика особых знаний и навыков.
- Принципиальная невозможность поддержки стереометрии.
- Трудности интеграции с системой обучения (осложнено или не возможно встраивание продукта в систему обучения).

4. ОПИСАНИЕ ПРЕДЛАГАЕМОЙ СИСТЕМЫ

4.1 Архитектура системы

Система визуализации и интерпретатор языка объединены в едином модуле (далее «интерпретатор»). Интерпретатор представляет собой ActiveX объект, что позволяет легко подключать его системам обучения, а также отображать чертежи в интернет-браузере. Программы и библиотеки хранятся независимо от интерпретатора в системе обучения и передаются интерпретатору средствами системы обучения, что делает сам интерпретатор независимым от конкретной реализации системы обучения.

4.2 Модель представления сцены

Разработанная модель представления сцены позволяет хранить информацию не только об объектах сцены, но и об их связях, о необходимости чего мы говорили в разделе 2.

Обозначим множество объектов сцены $\mathcal{A}$.

Определим функцию параметризации объектов.

$y, x_1, ..., x_n \in \mathcal{A}$, $y = F(x_1, ..., x_n)$, тогда F – функция параметризации. Таким образом, накладываются связи на объекты сцены.

Пусть теперь множество $\mathcal{F}$ – множество всех функций параметризации (множество связей) для заданной сцены.

Тогда определим отношение предшествования следующим образом: пусть существуют объекты x и y из множества $\mathcal{A}$ и функция F из множества $\mathcal{F}$ и выполнено соотношение: $y = F(x)$ (подразумевается, что функция F может зависеть и от других объектов, но для краткости описания прочие аргументы опущены в записи), тогда будем говорить, что x предшествует y. То есть:

Пусть $\exists x, y \in \mathcal{A}$ и $\exists F \in \mathcal{F}$ такие, что $y = F(x)$, тогда $x \to y$. Будем называть объект $x \in \mathcal{A}$ предшествующим $y \in \mathcal{A}$, если $x \to y$.

Построим транзитивное замыкание множества $\mathcal{F}$. Пусть $\exists x, y, z \in \mathcal{A}$ и $\exists F, G \in \mathcal{F}$ такие, что $y = F(x)$ и $z = G(y)$, тогда функция $P(x) = G(F(x))$ принадлежит транзитивному замыканию множества $\mathcal{F}$, то есть $P \in \overline{\mathcal{F}}$. Таким образом, для транзитивного замыкания множества $\mathcal{F}$ отношение предшествования становится транзитивным. Определим отношение зависимости.

Пусть $\exists x, y \in \mathcal{A}$ и $\exists F \in \overline{\mathcal{F}}$ такие, что $y = F(x)$, тогда $x \Rightarrow y$.

Будем называть объект $y \in \mathcal{A}$ зависимым, если $\exists x \in \mathcal{A}$ такой, что $x \Rightarrow y$.

Аналогично определим независимый объект: объект $x \in \mathcal{A}$ называется независимым, если $\forall a \in \mathcal{A}$ имеем $a \not\Rightarrow x$ (x не зависит от a).

Тогда определим множество $\mathcal{U}$ следующим образом: $\forall x \in \mathcal{A}$, если x – независимый, то $x \in \mathcal{U}$. Множество $\mathcal{U}$ назовем управляющим множеством.

Определим множество $\mathcal{Z}_x$ – множество объектов сцены зависящих от x. Пусть $x \in \mathcal{A}$, тогда $\forall y \in \mathcal{A}$ такого, что $x \Rightarrow y$, получаем $y \in \mathcal{Z}_x$.

Следует обратить внимание на основное свойство объектов из множества $\mathcal{U}$: если $x \in \mathcal{U}$, то при изменении положения x, поменяется положение всех объектов множества $\mathcal{Z}_x$.

Для $\forall x \in \mathcal{Z}_x$ объект x не может произвольно изменять свое положение, тогда как $\forall x \in \mathcal{U}$, x может изменять свое положение.

Определим $\mathcal{Z}$ – множество зависимых объектов сцены. $\forall x \in \mathcal{U}$, $\mathcal{Z}_x \subseteq \mathcal{Z}$ и $\forall y \in \mathcal{Z}$, $\exists x \in \mathcal{U}$ такой, что $y \in \mathcal{Z}_x$.

Заметим, что $\mathcal{A} = \mathcal{U} \cup \mathcal{Z}$.

Связи, описанные при построении чертежа – это множество $\mathcal{F}$. Данное множество функций дает возможность реализовать интерактивность чертежа. При изменении некоторого объекта происходит перерисовка всех зависимых объектов (зависимость задана функцией) с сохранением связей. Таким образом, множество $\mathcal{U}$ (управляющее множество) описывает те объекты сцены, которые могут быть изменены учащимися. Кроме того, это множество играет ключевую роль при создании анимации, так как функции преобразования чертежа, зависящие от времени, могут быть наложены только на объекты из управляющего множества.

4.3 Система визуализации

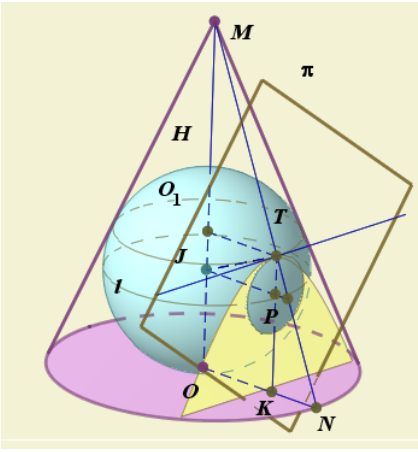

Рис. 2. Пример стереометрического чертежа, созданного в системе

4.3.1 Интерактивность

Графическая компонента предоставляет обучаемому два вида работы с иллюстрацией – ее изучение и выбор некоторых объектов для ответа на поставленный вопрос. Ученику дается возможность вращения, сдвига и масштабирования иллюстрации для ее детального изучения с различных точек зрения, что особенно полезно при работе со стереометрическими чертежами.

При выборе объекты выделяются толщиной и цветом линии.

4.3.2 Анимация

Анимация реализована с помощью системы эффектов, применяемых к примитивам иллюстрации (точкам, прямым, многогранникам и т.д.), и содержащих параметры их преобразования как функцию времени. Такой подход позволяет интерактивно редактировать содержание анимации, что является существенным отличием от простого "видеоролика". Создание анимаций упрощается за счет наличия математических связей между примитивами сцены, что позволяет приводить в движение сложные конструкции, задавая параметры анимации только одной или нескольким точкам. Например, для деформации тетраэдра достаточно задать сдвиг одной из его вершин. При анимации графиков при необходимости используются их точные математические представления.

К основным анимационным эффектам относятся:

- поворот, сдвиг и масштабирование отдельных объектов или всей сцены в целом;
- появление и исчезновение объектов, в том числе отдельных граней, поверхностей и линий контура;
- постепенная прорисовка линий (конец линии движется по экрану, рисуя линию);
- постепенная закраска граней;
- выделение объектов цветом, толщиной линии и/или миганием;
- плавная трансформация графика одной функции в график другой функции.

4.3.3 Реализация

Графическая компонента [7] состоит из следующих основных частей:

Рис. 3. Структура графической компоненты

Графическая сцена. Модуль, отвечающий за текущее состояние сцены. Включает в себя хранилище входящих в нее объектов и математических соотношений между ними (множества $\mathcal{A}$ и $\mathcal{F}$).

Компоновка изображения. Модуль, отвечающий за создание изображения по объектам сцены. В данном модуле происходит отрисовка двухмерных чертежей и формирование команд к подсистеме OpenGL для трехмерной визуализации.

OpenGL. Модуль, отвечающий за прорисовку трехмерного графического изображения пользователю посредством библиотеки OpenGL.

Анимация. Модуль, отвечающий за отображение интерактивной анимации. Включает в себя набор эффектов, применяемых к объектам из управляющего множества $\mathcal{U}$ сцены по заложенному преподавателем сценарию.

Внешнее хранилище. Объектное хранилище, позволяющее сохранять как объекты сцены со связывающими их соотношениями, так и анимационные эффекты.

4.4 Система программирования

Разработанный язык является прикладным, процедурным, интерпретируемым языком. Синтаксис и семантика языка достаточно просты.

В языке реализован механизм описания функций с возможность использования параметров-переменных. Функции создания объектов задают геометрические связи – функции множества $\mathcal{F}$. Аргументами функции $F \in \mathcal{F}$ являются объекты представленные параметрами-переменными.

Пример. Программа нахождения середины отрезка.

```
function Divide(A: point, B: point): point {
  c1=Circle(A,B);
  c2=Circle(B,A);
  E,F=c1^c2;
  b=Line(E,F);
  G=a^b;
  RESULT=G;
}
main {
  Set2DMode();
  A=Point(-0.2,0);
  B=Point(0.2,0);
  G=Divide(A,B);
}
```

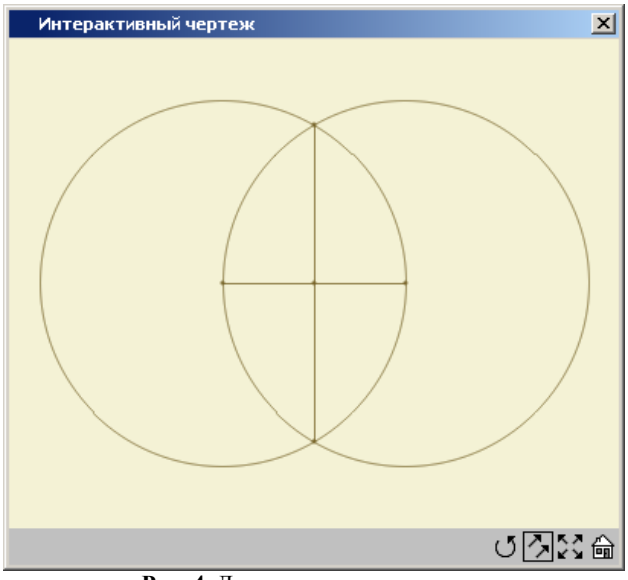

Рис. 4. Деление отрезка пополам.

Интерпретатор реализован на языке Microsoft Visual C/C++ и представляет собой объект ActiveX, работающий в интернет-браузере. Среда программирования реализована на языке JavaScript.

Особенности интерпретации управляющих структур программы – текст некоторой части программы просматривается, анализируется и интерпретируется столько раз, сколько раз выполняется эта программная ветвь.

Для удобства написания программ реализован отладчик с пошаговым выполнением программы.

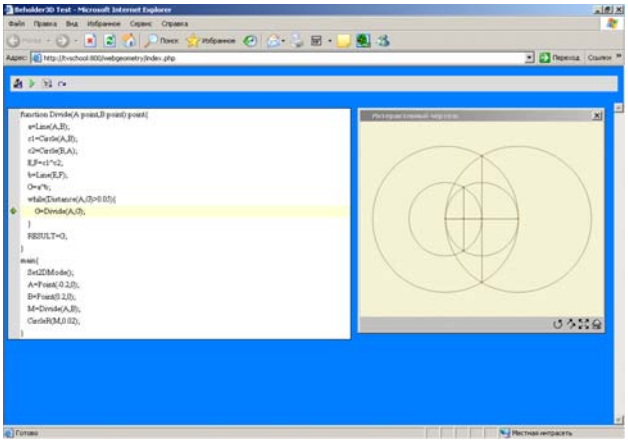

Рис. 5. Пошаговое выполнение.

Создаваемые чертежи могут быть полностью интерактивными. Меняя положение управляющих точек (объекты множества $\mathcal{U}$), можно проследить за изменением всего чертежа (элементов множества $\mathcal{Z}$). На следующем рисунке показаны 3 последовательных положения центра окружности. При перемещении центра окружности B (центр B движется от линии AC), изменяются положения окружности, радиусы BA и BC проведенные в точки пересечения линии AC и окружности и соответствующая хорда AC.

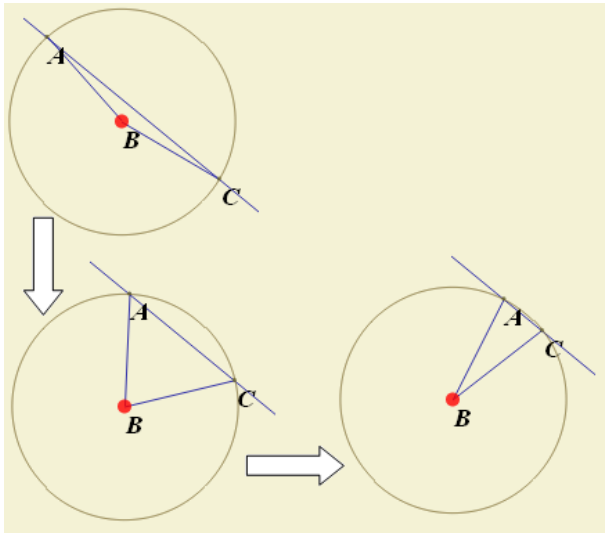

Рис. 6. Интерактивность.

В языке реализованы функции изменения атрибутов объектов (толщина, цвет, видимость). Кроме того, реализованы операторы ветвления, цикла и массивы.

Пример. Построение гиперболического параболоида

```
main{
   for(i,0,10,1){
      for(j,0,10,1){
         x=i/10-0.5;
         y=j/10-0.5;
         A[i][j]=inv Point3D(x,y,x*x-y*y);
      }
   }
   for(i,0,10,1){
      for(j,0,10,1){
         if(i>0){
            Line3D(A[i][j],A[i-1][j]);
         }
         if(j>0){
            Line3D(A[i][j],A[i][j-1]);
         }
      }
   }
}
```

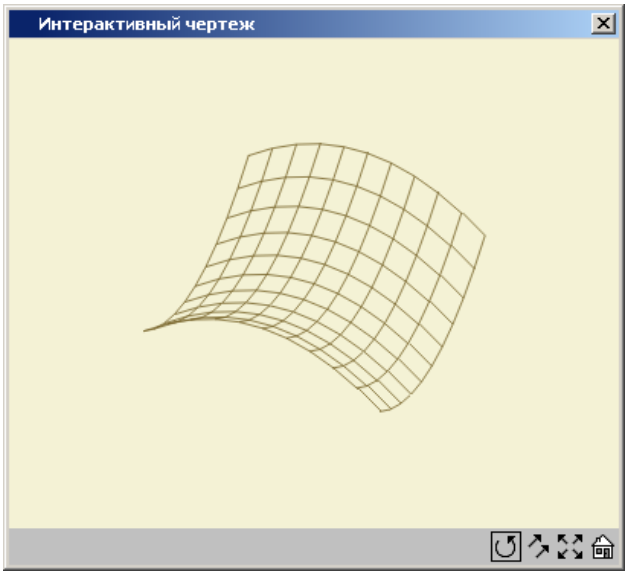

Рис. 7. Гиперболический параболоид.

Реализована возможность создания и подключения библиотек функций для повышения эффективности создания материалов.

5. ЗАКЛЮЧЕНИЕ

Описанная система полностью удовлетворяет требованиям, описанным в разделе 2 и, кроме того, данная система не содержит недостатков обнаруженных в других системах и описанных в разделе 3.

Интерактивные чертежи хранят информацию не только об объектах сцены, но и об их связях – таким образом увеличивается наполненность учебно-методических материалов.

Разработанная модель представления сцены описывает и объекты, и связи, позволяя реализовать весь необходимый набор функций для работы с чертежами.

Система визуализации реализует как планиметрические, так и стереометрические построения – охвачен весь курс геометрии. Следует отдельно отметить оптимизацию работы визуализатора – при пересчете сцены происходит только пересчет объектов в соответствии с наложенными связями. Интерпретатор не зависит от конкретной реализации системы обучения и может быть легко встроен в любую из них.

6. БИБЛИОГРАФИЯ

[1] Институт новых технологий, - http://www.int-edu.ru/
[2] Key Curriculum Press, - http://www.keypress.com/
[3] Graphisoft, - http://www.graphisoft.com/
[4] SolidWorks Corporation, - http://solidworks.com/
[5] UGS Corporation, - http://www.ugs.com/
[6] Web 3D Consortium, - http://www.web3d.org/x3d/
[7] Д.В. Дзюба, С.С. Крылов. Графическая компонента интерактивных систем обучения математике. Электронный журнал «Труды МАИ» №11, 11.04.2003г. http://www.mai.ru/projects/mai_works/articles/num11/article1/auther.htm
[8] Б.И. Аргунов, М.Б. Балк. Геометрические построения на плоскости, - М.: Учпедгиз, 1956г.
[9] Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. - М.: Мир, 1978. - Т. 1,2.
[10] Кауфман В. Ш. Языки программирования. Концепции и принципы. - М.: Радио и связь, 1993г.
[11] Фомичев В. С. Формальные языки, грамматики и автоматы, СПбГЭУ "ЛЭТИ"
[12] Страуструп Б. Язык программирования С++ (2-ред)./Пер. с англ.-М.: Радио и связь, 1995.
[13] Н. Вирт "Язык программирования Паскаль", -Л.: Изд-во ЛГУ, 1974.

Об авторах

Крылов Сергей Сергеевич – доцент, кандидат физ.-мат. наук, доцент кафедры вычислительной математики и программирования Московского Авиационного института.
Адрес: Московский авиационный институт (государственный технический университет), 125993, г. Москва, ГСП-3, Волоколамске шоссе, 4.
Телефон: 158-4090
Email: pmf_priem@mai.ru
Моргунов Игорь Анатольевич – студент 6 курса факультета прикладной математики и физики Московского Авиационного Института.
Адрес: Московский авиационный институт (государственный технический университет), 125993, г. Москва, ГСП-3, Волоколамске шоссе, 4.
Телефон: 158-4983
Email: merl@pisem.net

Computer aided system for designing interactive parametric drawings for geometrical e-learning books

Abstract

This paper is devoted to the geometrical programming language, interpreter, programming environment and mechanism of visualization.

The paper contains information about the geometrical scene representation, capabilities of language, capabilities of programming environment and mechanism of visualization.

Keywords: *visualization, programming language, interpreter, programming environment, geometrical drawing.*

About the author(s)

Krylov Sergey is a docent at Moscow Aviation Institute, Department of Applied Mathematics and Physics. His contact email is pmf_priem@mai.ru.
Morgunov Igor is a student at Moscow Aviation Institute, Department of Applied Mathematics and Physics. His contact email is merl@pisem.net.