

УДК 681.3

## Визуальная среда разработки систем геометрического моделирования

Карабчевский В.В., Хлепитько И.В.  
Донецкий национальный технический университет  
[karabch@pmi.dgtu.donetsk.ua](mailto:karabch@pmi.dgtu.donetsk.ua)

### Abstract

*Karabchevsky V., Hlepitko I. Visual environment for geometric modeling systems development. Basic methods and principles of building visual development environment oriented to simplify creation graphic systems for manipulation geometric objects are described.*

### Постановка проблемы

На данном этапе развития компьютерных технологий системы геометрического моделирования являются незаменимым компонентом во многих отраслях производственной деятельности человека. В последние годы в инженерной практике широко применяются вычислительные возможности персональных компьютеров для выполнения расчетов, автоматизации проектирования, организации и планирования экспериментальных исследований, для обработки результатов испытания машин, механизмов, аппаратов. Кроме того, сейчас необходима комплексная компьютеризация инженерной деятельности на всех этапах жизненного цикла изделий. Широкое признание получили такие системы как AutoCAD, 3D MAX, SolidWorks и т.д. Все они обладают собственными достоинствами и недостатками, каждая система имеет огромный функционал, позволяющий решить большинство задач в той или иной отрасли, но и за лицензионную копию такого продукта придется выложить солидную сумму. При решении широкого спектра задач затраты скорее всего окупятся, но приобретать такую мощную систему ради решения однотипных задач зачастую является нецелесообразным.

В большинстве случаев при покупке одной из перечисленных систем используется лишь малая часть огромного функционала, предоставляемого разработчиками данной системы. И тогда на предприятии может быть принято решение о написании собственной системы, обладающей весьма ограниченным набором функций, который, однако, позволяет решить поставленную задачу.

Такая проблема также возникает в большинстве высших учебных заведений нашей страны. В настоящее время для инженера любой специальности навыки и умения решения производственных и научных задач с помощью вычислительных машин являются необходимыми. Однако покупка лицензионных копий большинства графических систем учебным заведениям не по карману и тогда

единственным выходом является создание собственных систем геометрического моделирования.

В этом случае много времени уходит на разработку собственных классов и их связывание, написание функций для работы с графическими объектами и т.д. За счет необходимости написания системы «с нуля» значительно увеличивается время разработки системы и возникает необходимость иметь некоторую библиотеку классов и функций, которая позволит легко и быстро создавать простые графические системы для решения конкретной задачи. Кроме того, для быстрого и удобного использования этой библиотеки целесообразно создать некоторую визуальную среду, которая позволит максимально ускорить и упростить работу программиста при создании графической системы. Данная среда может быть легко освоена даже неопытным программистом и является связующим звеном между функциями ядра и программистом проектировщиком системы геометрического моделирования.

### Анализ достижений и публикаций

На данный момент средствам разработки систем геометрического моделирования уделяется огромное внимание. Выделяют три типа ядер: лицензируемые ядра, частные и ядра, поставляемые в исходном коде [1].

Лицензируемые ядра геометрического моделирования разрабатываются одной компанией, которая лицензирует их другим компаниям для их CAD-систем. Использование таких ядер в данном случае является нецелесообразным, так как за это придется выложить немалую сумму.

Частные ядра геометрического моделирования разрабатываются исключительно в собственных целях, и использование таких ядер не является возможным.

Ядра, доступные в исходном коде подобны лицензированным ядрам. Они также разрабатываются и поддерживаются одной компанией и затем лицензируются другим компаниям для использования в CAD-приложениях. Отличие стоит в том, что эти

разработчики обеспечивают исходный код ядра. Для пользователей, которые имеют группы разработки и хотят сами настраивать ядро системы, очень удобно иметь возможности настройки, поскольку исходный код доступен. Использование таких ядер позволяет получить достаточно большой функционал для решения большинства задач, поставленных перед программистом проектировщиком. Однако структура таких ядер достаточно сложна и для эффективной работы разработчику требуются знания и навыки работы с таким инструментарием.

Таким образом, возникла необходимость в создании визуальной среды разработки систем геометрического моделирования, удобной и простой в использовании.

В [2] были рассмотрены проблемы, связанные с построением обучающих систем геометрического моделирования. Позднее было принято решение о создании системы, описанной в [3]. В ходе работы были сделаны выводы о необходимости создания инструментария, позволяющего сократить временные затраты при построении схожих систем, используя накопленные навыки и знания. В [4] был описан функционал разрабатываемого ядра, сформулированы проблемы, возникающие при построении ядра. На основе данного ядра была разработана система, описанная в [5]. Ее основной задачей являлось построение трехмерного представления по комплексному чертежу в учебных целях для решения задач прикладной геометрии, а также демонстрация возможностей ядра. Визуальная среда разработки систем геометрического моделирования предназначена для упрощения использования данного ядра, а также для сокращения временных затрат на изучение возможностей ядра.

### **Постановка задачи**

Целью данной работы является построение визуальной среды, удовлетворяющей вышеописанным требованиям. Можно выделить две составные части данной системы: графическое ядро, содержащее основной набор функций и классов и, собственно, сама визуальная среда, позволяющая использовать ядро для вставки блоков кода в проект пользователя.

### **Создание визуальной среды**

На данном этапе разработки реализовано графическое ядро, содержащее классы для работы с такими геометрическими объектами как точка, линия, окружность, эллипс, сплайн, а также функции для работы с такими объектами.

К таким функциям относятся функции поворота объекта на произвольный угол относительно произвольной точки, перенос объекта, для отрезков реализованы функции выравнивания (align) и удлинения (lengthen), аналогичные одноименным функциям системы AutoCAD. При разработке данного ядра была использована структура, приближенная к структуре ядра Open CASCADE, описанного в [6].

Сама визуальная среда в значительной мере напоминает среду Delphi. Данная среда содержит дизайнер форм, редактора исходного текста, палитру компонентов и инспектор объектов.

Основой разрабатываемого в данной среде проекта является главная форма будущей системы геометрического моделирования. На рисунке 1 показана главная форма в процессе разработки:

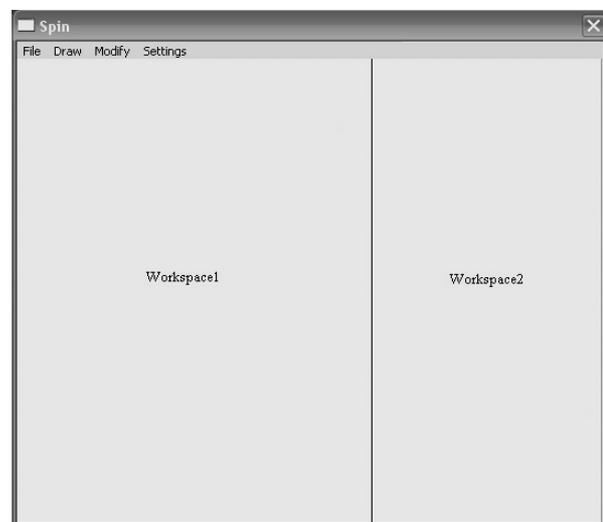


Рисунок 1 – Главная форма будущей среды геометрического моделирования

Как видно из рисунка, на форму добавлены такие компоненты как рабочая область и главное меню. За модификацию главной формы отвечает дизайнер форм. Программист проектировщик в ходе работы с проектом может создавать на главной форме элементы управления из палитры компонентов, связывая их с пользовательскими функциями. К основным элементам управления можно отнести такие компоненты, как рабочая область, главное меню, командная строка, кнопка, поле для ввода, поле для отображения текста, строка состояния.

Также в среде существуют невидимые для пользователя компоненты. К числу таких компонентов, например, относится список графических объектов. Данный компонент виден на главной форме проекта в режиме

проектирования, но в режиме выполнения он, естественно, скрыт от глаз пользователя.

При создании нового проекта в рассматриваемой среде, создается файл проекта формата xml, который содержит описание разрабатываемого приложения. В файле проекта хранятся основные свойства главной формы (например, положение, цвет фона, тип рамки) и компонентов, добавленных проектировщиком на форму. Соответственно, содержимое файла проекта обновляется при добавлении, модификации или удалении объектов на форме. Кроме того, создается файл с кодом программы на языке C++, который отображается в редакторе исходного текста и может быть при необходимости изменён пользователем. Редактор исходного текста и сгенерированный код для обработчиков кнопок главного меню показаны на рисунке 2:

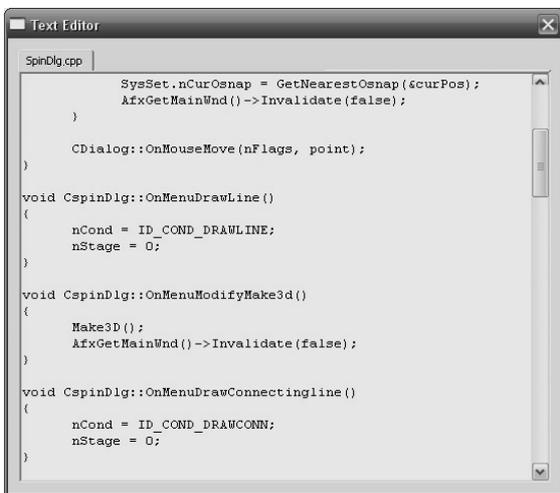


Рисунок 2 – Редактор исходного текста

Добавление компонента происходит путем выбора его из палитры компонентов и последующего указания его положения на главной форме проекта. Сама палитра компонентов показана на рисунке 3. При этом на форме создается заданный компонент с установленными по умолчанию значениями основных свойств.

Программно это реализуется следующим образом: класс с описанием диалогового окна, соответствующего главной форме проекта, содержит в качестве свойств массивы основных компонент. При создании проекта вышеупомянутые массивы пусты и заполняются по мере добавления пользователем компонент. Таким образом, при добавлении на главную форму проекта, например, элемента управления типа «кнопка», размерность массива кнопок увеличивается, в него заносится новый элемент,

и свойства этого элемента заполняются по умолчанию.

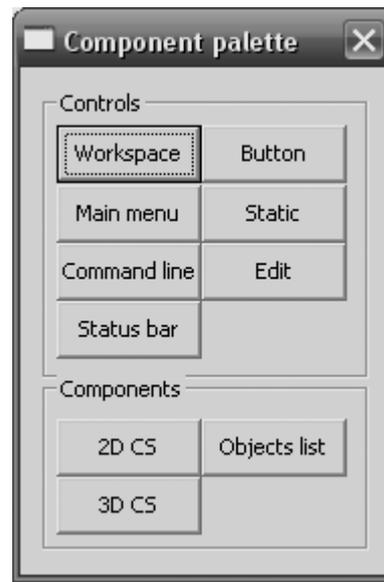


Рисунок 3 – Палитра компонентов

Корректировать значения этих свойств пользователь может при помощи инспектора объектов. На рисунке 4 показаны инспектор объектов и свойства главной формы проектируемой системы, редактируемые с его помощью.

В дальнейшем можно установить обработчики для созданных компонент, используя стандартные функции ядра или пользовательские функции.

Особенностью данной среды разработки является упрощенное связывание обработчиков элементов управления со стандартными функциями ядра. Например, для таких элементов управления как кнопка и пункт меню можно связать обработчик на нажатие со стандартной функцией создания одного из графических объектов. Для этого на главной форме должен быть создан такой компонент, как список графических объектов. Этот компонент должен быть создан в единственном экземпляре для любого из создаваемых приложений и после связывания с главной формой проекта выполнение любой операции с графическими объектами подразумевает модификацию объектов, принадлежащих именно этому списку.

Кроме того, в функции данного компонента входит сохранение и загрузка объектов из файлов формата dxf [7].

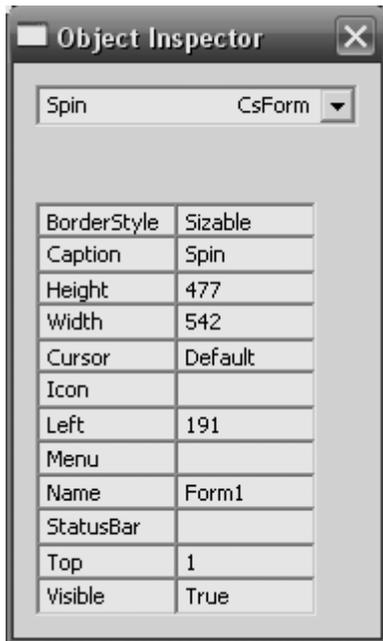


Рисунок 4 – Использование инспектора объектов для редактирования свойств главной формы проекта

Данный формат поддерживается всеми достаточно мощными системами геометрического моделирования и таким образом программист проектировщик может обеспечить переносимость проектов, создаваемых в разрабатываемой им системе, в более крупные системы.

Для выполнения видовых операций, на главную форму проекта необходимо добавить компонент «система координат». Данный компонент связывается с элементом управления «рабочая область». Любая рабочая область может содержать только одну систему координат. При привязке стандартной функции ядра, отвечающей за видовые операции (панорамирование, зуммирование) необходимо указать систему координат, для которой будет выполняться данная операция [8]. В зависимости от поставленной задачи, можно использовать двумерную или трехмерную систему координат. При использовании стандартной функции ядра Make3D, которая по двумерным проекциям объекта строит его трехмерное представление, необходимо указать и двумерную систему координат, в которой находятся проекции объекта, и трехмерную систему координат, в которой будет построено трехмерное представление этого объекта [9].

Элемент управления «командная строка» предназначен для ввода команд пользователем разрабатываемой системы геометрического моделирования, а также для задания числовых характеристик создаваемых и модифицируемых

объектов. Программист проектировщик имеет возможность задавать стандартные команды, которые будут распознаваться командной строкой, и устанавливать обработчики для этих команд.

### Применение разработанных средств

В ходе работы было проведено также сравнение временных затрат на построение различных геометрических систем. Как было упомянуто ранее, геометрическая система, описанная в [3], была разработана «с нуля», без использования вышеописанных средств. Некоторое время спустя на основе данной среды разработки была построена система, описанная в [5], и обладающая более мощным функционалом по сравнению с предыдущей.

Использование данных средств разработки позволило сократить время разработки второй системы примерно в 4 раза по сравнению с первой. Таким образом, можно сделать вывод о значительном выигрыше по времени создания и функциональности разрабатываемых систем при использовании данной визуальной среды разработки систем геометрического моделирования.

Для более наглядной демонстрации возможностей данной визуальной среды рассмотрим графическую систему для построения трехмерного представления по комплексному чертежу, упомянутую выше.

На рисунке 5 показано решение задачи о нахождении вертикали и горизонтали плоскости, заданной тремя точками, с помощью данной системы:

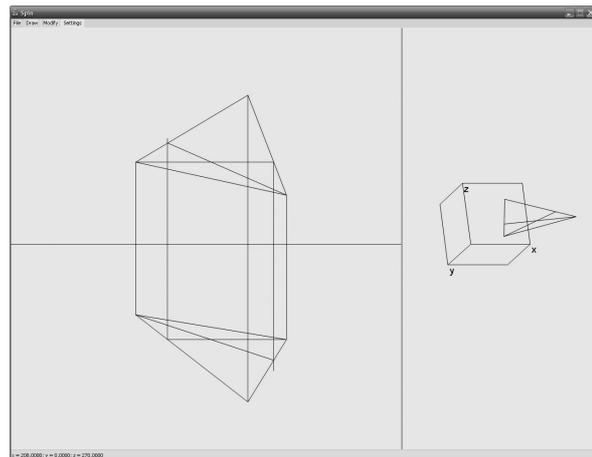


Рисунок 5 – Нахождения горизонтали и фронталы плоскости, заданной треугольным отсеком

В данной системе использованы такие компоненты как главное меню, строка состояния,

рабочая область, список объектов и система координат.

При создании главного меню программист указывает наименование и положение кнопок в меню, а также устанавливает обработчики на их нажатие.

При создании строки состояния программист устанавливает свойство «Show cursor position» для отображения координат курсора в строке состояния. На программном уровне происходит следующее: при установке этого свойства в обработчик перемещения курсора записывается вызов функции для вывода координат в строку состояния.

Для работы с графическими объектами создается две рабочие области. Слева располагается область для работы с двумерными объектами. С ней связывается двумерная система координат, для которой указывается её центр и направление осей. Справа создается область для работы с трехмерными объектами, указывается для нее трехмерная система координат. После этого на форму помещается скрытый компонент «список объектов». На нажатие кнопки главного меню Draw->Line устанавливается стандартная функция рисования отрезка, которая определяет рабочую область, на которой необходимо рисовать объект, и после задания двух точек отрезка заносит новый объект в список графических объектов.

Для рисования вертикальных и горизонтальных линий используется режим Orto, который включается при нажатии кнопки главного меню Settings->Orto. В данном обработчике изменяется системный флаг bIsOrto. В каждый проект автоматически добавляется переменная SysSet структуры SystemFlags. Системные флаги предназначены для фиксации состояния системы в некоторый конкретный момент времени. Таким же образом происходит перевод системы в режимы Snap и Grid. Стандартная функция рисования объектов определяет состояние системы по системным флагам и в дальнейшем выполняет необходимые для данного режима операции.

Для создания трехмерного объекта используется функция Make3D, упомянутая ранее. Она использует первые два выделенных объекта из списка объектов в качестве горизонтальной и фронтальной проекции создаваемого трехмерного объекта. Для упрощения задания объектов используется режим привязок, с помощью которого можно указывать концы отрезков и точки их пересечения при задании нового отрезка. Для этого используется стандартная функция ядра, которая находит ближайшую привязку в некоторой области заданного радиуса.

Стоит также упомянуть о таком виде графических объектов как соединительная линия (connection line). Эта линия связывает две проекции трехмерного объекта. Модификация таких объектов напрямую не возможна. Модификация соединительной линии происходит в том случае, когда модифицируется сам объект, к которому она относится. При удалении объекта также удаляются и все его соединительные линии. Удаление соединительной линии между двумя проекциями приводит к разрушению трехмерного объекта, построенного по данным проекциям.

В пункте меню Modify можно выбрать функцию для модификации множества выбранных объектов. Все эти функции работают с компонентом «список объектов». Для списка объектов существуют стандартные функции поворота и переноса объектов, реализованные на основе аффинных преобразований [10]. Для объектов типа «отрезок» также предусмотрены функции выравнивания по образцу и удлинения. Реализация функции поворота проиллюстрирована на рисунке 6:

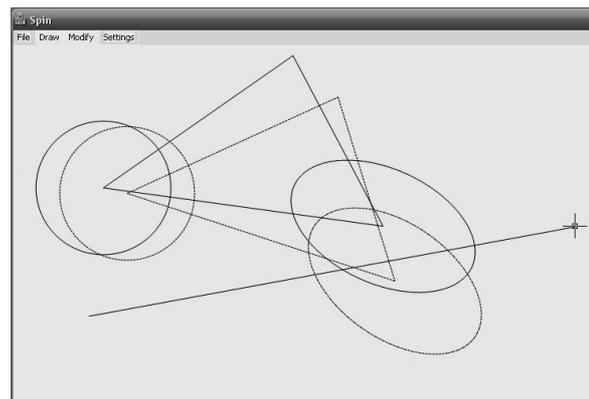


Рисунок 6 – Реализация функции поворота объектов

С помощью данной системы можно выполнять более сложные задачи. Найдем точку пересечения плоскости с перпендикуляром, опущенным из произвольной точки. Для этого сначала построим плоскость, проведем для нее фронталь и горизонталь и опустим на них перпендикуляр из произвольной точки. Сделать это можно при помощи привязок. После чего при помощи команды Lengthen в меню Modify продлим фронтальную проекцию перпендикуляра до пересечения со второй гранью треугольного отсека, определяющего плоскость.

Построим горизонтальную проекцию этой линии и удлиним горизонтальную проекцию перпендикуляра до пересечения с этой линией. Это и будет точка пересечения перпендикуляра с

плоскостю. Решение задачи нахождения точки пересечения плоскости с перпендикуляром, опущенным на плоскость из произвольной точки, и ее трехмерное представление показано на рисунке 7:

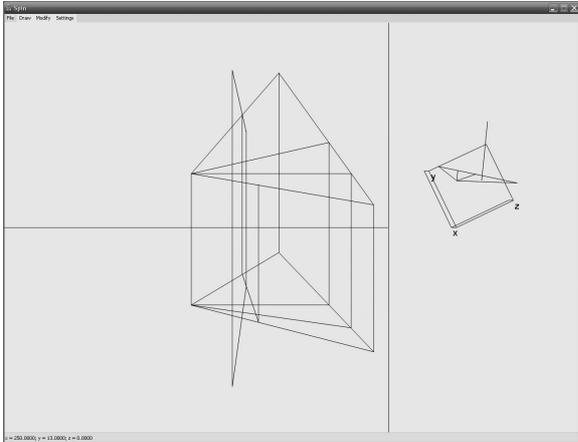


Рисунок 7 – Точка пересечения перпендикуляра с плоскостью

### Выводы

Использование сочетания визуальных компонентов и функций ядра позволит разработчику графической системы избавиться от долгой рутинной работы при создании интерфейса графической системы и сократит временные затраты на построение системы в целом. Разрабатываемая система имеет удобный интуитивно понятный интерфейс и легко осваивается пользователем.

Система позволяет пользователю использовать стандартные функции для создания геометрических объектов, выполнения модификации таких объектов, а также выполнения видовых операций над всеми объектами в целом. Область применения разрабатываемой визуальной среды достаточно велика. Данная система может использоваться как в учебных целях, для построения простых систем геометрического моделирования, предназначенных для демонстрации задач начертательной геометрии, так и на производстве, для создания узкоспециализированных графических систем с минимальными временными затратами.

В ближайшем будущем планируется усовершенствование данной среды, наращивание её функциональных возможностей, модификация интерфейса для максимального удобства пользователя среды. Построение трехмерного представления на данном этапе возможно лишь в случае, когда проекциями являются отрезки. В ближайшем будущем также планируется устранить этот недостаток, позволить

пользователю создавать поверхности большинством возможных средств, реализовать функции для работы с твердотельными объектами.

### Литература

1. Обзор ядер геометрического моделирования – <http://www.cad.dp.ua/obzors/karnel.php>
2. Карабчевський В.В. Засоби розробки навчальних систем для курсу "Інженерна графіка" // Прикладна геометрія та інженерна графіка. Міжвідомчий науково-технічний збірник. Випуск 71. Київ: КНУБА. – 2002. – С. 195-200.
3. Карабчевский В.В., Хлепиться И.В. Графический редактор для работы с параметризованными двумерными объектами // Матеріали III наукової конференції молодих учених та студентів «Інформатика та комп'ютерні технології». Донецьк: ДонНТУ. – 2007. – С. 278-280.
4. Карабчевский В.В., Хлепиться И.В. Разработка проблемно-ориентированных систем геометрического моделирования Наукові праці Донецького національного технічного університету. Серія "Проблеми моделювання та автоматизації проектування динамічних систем" (МАП-2007). Випуск: 6 (127) - Донецьк: ДонНТУ. - 2007. – С. 234-239.
5. Карабчевський В.В., Хлепиться І.В. Засоби розробки систем геометричного моделювання. Наукові нотатки. Міжвузівський збірник (за напрямом «Інженерна механіка»). Випуск 22. Частина 1. «Сучасні проблеми геометричного моделювання (квітень, 2008). – Луцьк, 2008. – С. 133-137.
6. Использование Open CASCADE для создания приложений – <http://www.sapr.ru/article.aspx?id=7878&iid=319>
7. Зоммер В. AutoCAD – 2007. Руководство чертежника, конструктора, архитектора. – М.: ООО «Бином – Пресс», 2007. – 816 с.
8. Л. Аммерал Машинная графика на персональных компьютерах. Пер. с англ. – М.: "Сол Систем", 1992. – 232 с.
9. Л. Аммерал Интерактивная трехмерная машинная графика. Пер. с англ. – М.: "Сол Систем", 1992. – 317 с.
10. Е.В. Шикин, А.В. Боресков – Компьютерная графика. Полигональные модели. – М.: ДИАЛОГ – МИФИ, 2001. – 464с.

Поступила в редколлегию 04.03.2009