УДК 004.92

ИНТЕРПРЕТИРУЮЩИЙ OPENGL ДЛЯ КОМПЬЮТЕРНОЙ ГРАФИКИ

Матросов В. А.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Амельченко Н. П. – канд. техн. наук, доцент, доцент кафедры ИКГ

Аннотация. *OpenGL* – это ведущий в отрасли кроссплатформенный интерфейс программирования графических приложений (*API*) и единственный основной *API* с поддержкой практически всех операционных систем. Многие языки, такие как *Fortran, Java, Tcl/Tk* и *Python*, имеют привязки *OpenGL* для использования возможностей визуализации *OpenGL*. В этой статье мы представляем *Ch OpenGL Toolkit*, действительно платформо-независимую привязку *Ch к OpenGL* для компьютерной графики. *Ch* – это встраиваемый интерпретатор *C/C++* для кроссплатформенного написания сценариев, программирования оболочки, численных вычислений и встроенных сценариев.

Ключевые слова: методология и методы взаимодействия, графические утилиты, программная поддержка, графические системы, распределенная/сетевая графика

Введение. Область компьютерной графики продолжает быстро развиваться с постоянно растущим числом применений в различных областях, таких как развлечения, бизнес, искусство, образование, медицина, инженерия и промышленность. Появилось множество пакетов программного обеспечения для создания и обработки двумерной (2*D*)/трехмерной (3*D*)-графики. *OpenGL* – это интерфейс программирования графических приложений (*API*) для языка программирования *C/C++*. Основной мотивацией разработки *OpenGL API* является создание независимого от операционной системы, оконной системы и аппаратной платформы *API* для разработки 2*D*/3*D* графики. С момента появления *OpenGL API* в 1992 году многие приложения, такие как *CAD, CAM* и разработка игр, выиграли от его кроссплатформенной доступности. *OpenGL* стал основной средой для разработки портативных приложений 2*D*/3*D* графики. Она также широко используется для преподавания и изучения компьютерной графики. Такие характеристики, как независимость от устройства и портативность, делают *OpenGL* стратегическим интерфейсом для курсов по компьютерной графике. Компьютерные платформы различаются у разных преподавателей и студентов, а также в разных школах и домах. Используя *OpenGL*, программы, разработанные на одной машине, можно отлаживать и оценивать на других машинах с разными платформами, и в результате графика будет одинаковой [1].

Поскольку *OpenGL* является одним из самых популярных пакетов графического программного обеспечения, многие языки, такие как *Fortran, Java, Tcl/Tk* и *Python*, имеют привязки к *OpenGL* для использования возможностей визуализации *OpenGL*. *Ch* – это встраиваемый интерпретатор *C/C++*. *Ch OpenGL Toolkit* еще больше повышает переносимость *OpenGL API*. Обычно прикладные программы *OpenGL* необходимо компилировать и компоновать, прежде чем запускать их на разных платформах. *Ch OpenGL Toolkit* делает приложения *OpenGL* действительно переносимыми на различные платформы. *C Ch OpenGL* исходный код приложения *OpenGL* может легко запускаться на различных платформах без процессов компиляции и компоновки.

Основная часть. *Ch,* первоначально разработанный Ченгом [2, 3], является встраиваемым интерпретатором *C/C++* для кроссплатформенного создания сценариев, программирования оболочки, численных вычислений и встроенных сценариев. Он поддерживает все возможности стандарта языка *C*, ратифицированного в 1990 году. Многие новые возможности, такие как комплексные числа, массивы переменной длины, арифметика с плавающей точкой *IEEE* и родовые математические функции, впервые реализованные в *Ch*, были приняты в *C*99, новом стандарте языка *C*, утвержденном в 1999 году. Кроме того, *Ch* поддерживает классы в *C++* для объектно-ориентированного программирования. Как и другие

математические программные пакеты, такие как *MATLAB, Ch* имеет встроенную поддержку двух- и трехмерного графического черчения, вычислительных массивов для векторных и матричных вычислений, а также анализа линейных систем с помощью расширенных функций численного анализа на основе *LAPACK*. Благодаря возможностям вычислительных массивов мы можем задавать векторные и матричные операции непосредственно в выражениях так же, как скаляры, как в интерактивном исполнении, так и в программах. На рисунке 1 показано интерактивное выполнение программных операторов в оболочке *Ch*. Декларатор типов array и *int* объявляют переменные *A* и *B* как вычислительные массивы типа int. Заголовочный файл *array.h* должен быть включен в программу, чтобы использовать вычислительные массивы.

Рисунок 1 – Векторные и матричные операции в *Ch*

Ch OpenGL – это привязка *Ch к OpenGL*. Она обеспечивает доступ к полной функциональности *OpenGL, GLU, GLUT и GLAUX. Ch OpenGL* имеет много новых возможностей. Во-первых, *Ch* – это кроссплатформенный интерпретатор *C/C++*. Как часть дистрибутива *Ch, Ch OpenGL* позволяет разработчикам приложений *OpenGL* писать приложения в кроссплатформенной среде. Весь исходный код приложения *OpenGL* может легко запускаться на различных платформах без процессов компиляции и компоновки, как показано на рисунке 2. Во-вторых, синтаксис *Ch OpenGL* точно такой же, как и интерфейс *C* для *OpenGL*. Нет необходимости изучать новый синтаксис. В-третьих, *Ch OpenGL* является встраиваемым. Встраивание *Ch* в графические приложения позволяет разработчикам или пользователям динамически генерировать и манипулировать графикой во время выполнения программы. Наконец, вычислительный массив в *Ch* делает векторные и матричные операции, выполняемые в трехмерной графике, более лаконичными. Мы считаем, что вычислительная эффективность графических приложений может быть значительно повышена за счет использования численных возможностей *Ch*. Интерфейс *Ch* к *OpenGL* намного проще, чем интерфейс любого другого языка к *OpenGL*, потому что и *Ch*, и *OpenGL* имеют одинаковый синтаксис и типы данных. Скрипты *Ch* могут получать доступ к функциям в статических или динамических бинарных библиотеках C/ C++ через *Ch SDK*. На рисунке 3 показано, как скрипт *Ch OpenGL* взаимодействует с библиотекой *OpenGL C.* Когда запускается сценарий *Ch OpenGL (*исходный код *C/C++ OpenGL), Ch* ищет файлы функций, которые соответствуют функциям в сценарии, в библиотеке функций *Ch*. Эти функциональные файлы передают параметры функций в пространстве *Ch* соответствующим функциям в динамически загружаемой библиотеке *Ch (CDLL*) в пространстве *C* через *Ch SDK.*

Функции в *CDLL* вызывают функции *OpenGL C* и возвращают результаты обратно в функциональные файлы *Ch. Ch OpenGL Toolkit* состоит из библиотеки функций *Ch* и динамически загружаемой библиотеки *Ch для OpenGL*. Реализация *Ch OpenGL Toolkit* проста для простых функций *OpenGL C*. Для некоторых функций, имеющих в качестве аргументов

указатели на функции обратного вызова, требуется специальная обработка, поскольку эти функции в пространстве *C* имеют обратные вызовы в пространстве *Ch*. Стратегия решения этой проблемы в *Ch OpenGL Toolkit* заключается в создании аналога *Ch callback* в пространстве *C*. В качестве примера, программа для системы, показанной на рисунке 4 [4], использует функцию *glutDisplayFunc* для установки обратного вызова дисплея для каждого окна в *GLUT*. Когда *GLUT* определяет, что окно экранного пространства нуждается в повторном отображении, вызывается обратный вызов дисплея для окна экранного пространства в пространстве C, а этот обратный вызов *C*, в свою очередь, вызывает обратный вызов *Ch* через динамически загружаемой библиотеки.

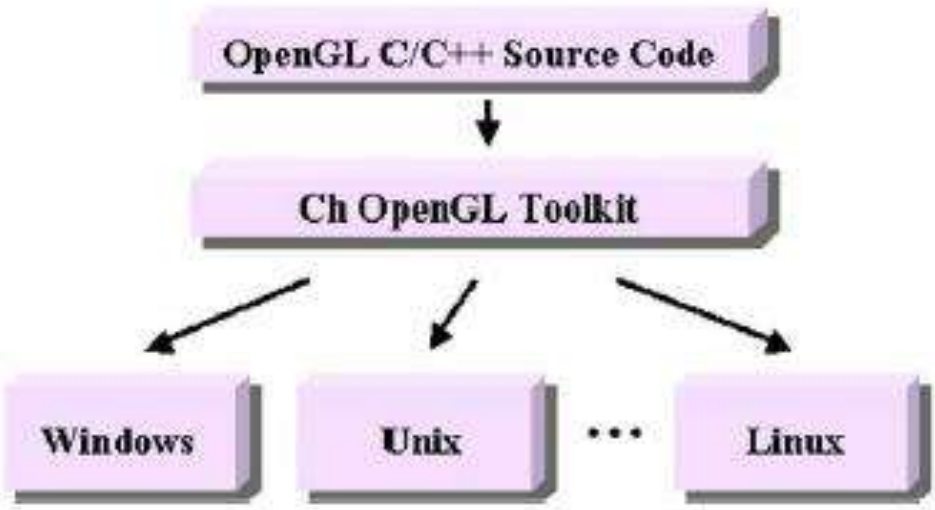

Рисунок 2 – Запуск исходного кода *C/C++ OpenGL* на различных платформах

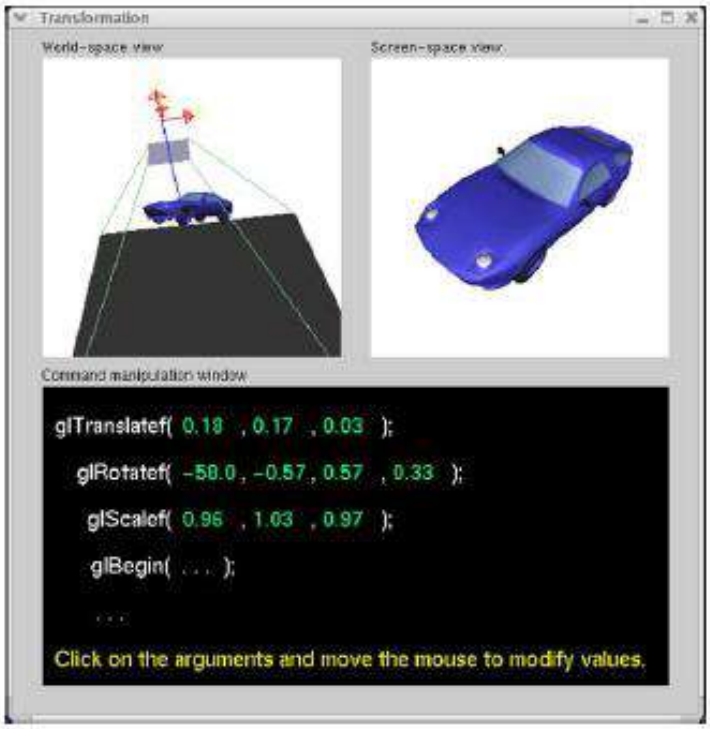

Рисунок 3 – *Ch SDK* позволяет *Ch* скриптам взаимодействовать с *C/C++* бинарными библиотеками

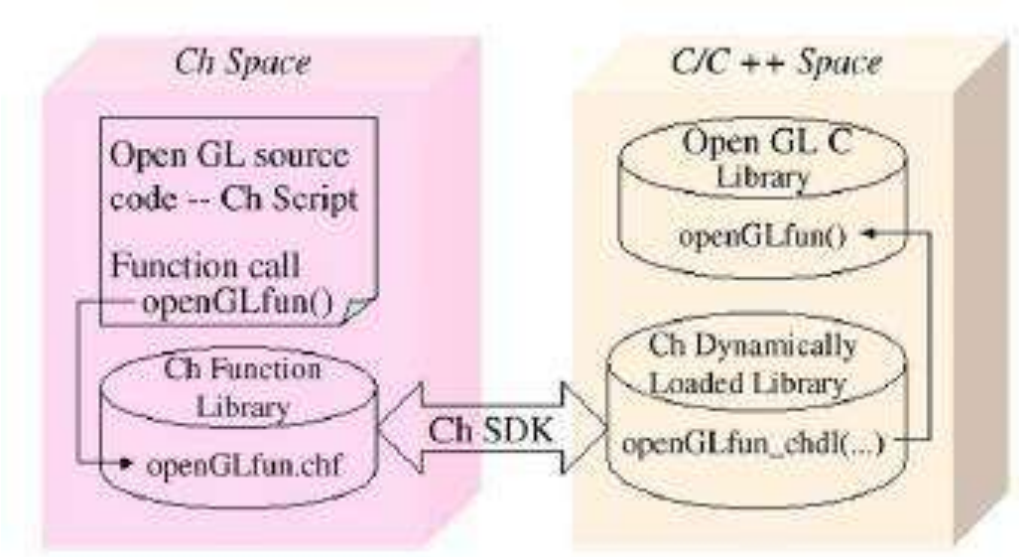

Рисунок 4 – Запуск программы *OpenGL* в интерпретируемом и интерактивном режимах

Заключение. В статье была представлена методология, которую можно использовать для реализации визуализации на основе *Ch OpenGL*. С помощью *Ch* программы визуализации на языке *C/C++* можно напрямую использовать в системах визуализации. Система визуализации на основе *Ch OpenGL* предоставляет возможность для новых форм обслуживания клиентов, таких как интерактивная 3*D* конфигурация продукта, заказ и обучение клиентов. Это также идеальная среда для преподавания и изучения компьютерной графики.

Список литературы

1. *Woo M, Neider J. OpenGL programming guide: the official guide to learning OpenGL / Woo M, Neider J, Davis T, Shreiner D. — version 1.2, 3 ed. — Reading: Addison-Wesley, 1999.*
2. *Cheng HH. Scientific computing in the Ch programming language / Cheng HH. — Scientific Programming, 1993.*
3. *Ch—an embeddable C/C++ interpreter [Electronic resource] / Softintegration, Inc. — Mode of access: http://www.softintegration.com.*
4. *Robins N. OpenGL tutorial [Electronic resource] / Mode of access: http://www.xmission.com/~nate/tutors.html.*

UDC 004.92

INTERPRETIVE OPENGL FOR COMPUTER GRAPHICS

Matrosov V. A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Amelchenko N. P. – PhD, assistant professor, associate professor of the department of ECG

Annotation. OpenGL is the industry-leading, cross-platform graphics application programming interface (API), and the only major API with support for virtually all operating systems. Many languages, such as Fortran, Java, Tcl/Tk, and Python, have OpenGL bindings to take advantage of OpenGL visualization power. In this article, we present Ch OpenGL Toolkit, a truly platform-independent Ch binding to OpenGL for computer graphics. Ch is an embeddable C/C++ interpreter for cross platform scripting, shell programming, numerical computing, and embedded scripting.

Keywords: methodology and techniques, interaction techniques, graphics utilities, software support, graphics systems, distributed/network graphics