

# Лекция 3. Библиотеки графических функций

Определение, стандарты разработки, координатные представления и конвейер наблюдения, элементарные графические функции, OpenGL, DirectX и другие библиотеки.

# Определение

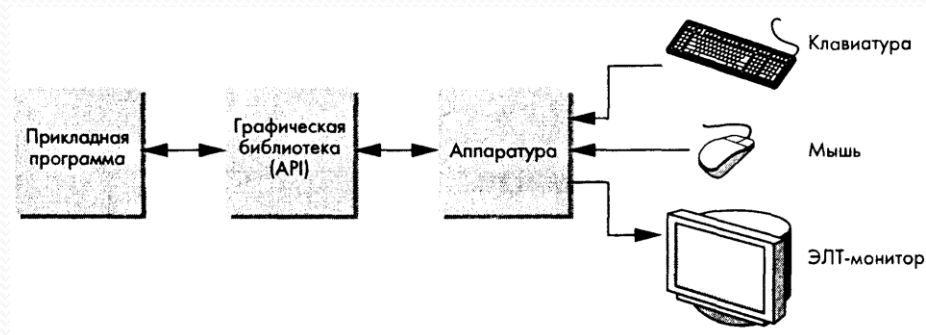


Рис. 1

- Графическая библиотека – это множество функций, которые образуют интерфейс между прикладной программой (или языком программирования, с помощью которого разрабатывается программа) и графической системой. Этот интерфейс называют программным интерфейсом приложения компьютерной графики (CG API - Computer-Graphics Application Programming Interface). Для программиста, занимающегося разработкой прикладной программы, существует только API, и он избавлен, таким образом, от необходимости вникать в подробности работы аппаратуры и программной реализации функций графической библиотеки.
- В схеме, приведенной на рис. 1, по одну сторону от API располагается прикладная программа, а по другую – комбинация программных и аппаратных средств, которая реализует функции, специфицированные в API, и формирует изображение на экране.
- Использование API позволяет разработчикам программ делать их универсальными, независимыми от низкоуровневых команд конкретного графического адаптера.

# Стандарты разработки библиотек

- Графические библиотеки должны обеспечивать универсальность работы прикладной программы на машинах с различным аппаратным и программным обеспечением. В противном случае, программу, разработанную для одной машины, скорее всего нельзя было бы перенести на другую систему без существенных изменений самой программы.
- Графические функции в любом пакете, как правило, задаются как набор описаний, которые не зависят от какого бы то ни было языка программирования. Затем задается привязка к языку программирования высокого уровня (C, C++, Java, Fortran, Pascal, Basic,... ). Эта привязка задает синтаксис, позволяющий пользоваться различными графическими функциями этого языка. Каждая привязка к языку задается так, чтобы можно было максимально использовать соответствующие возможности языка: типы данных, синтаксис задания параметров и обработки ошибок.

# Координатные представления

- При работе с простыми графическими библиотеками требуется, чтобы вся геометрическая информация задавалась в терминах системы координат экрана используемого устройства отображения. В этом случае, координаты точек задаются в системе координат с размерностями пикселей отдельного экрана или сантиметров (дюймов) от верхнего угла экрана (левосторонняя система координат). Но этот метод может быть неудобен для описания изображений, в которых расстояния измеряются в световых годах (при отображении данных, с которыми имеет дело астрономия), или таких, где расстояния измеряются микронами (при выводе изображений топологии интегральных микросхем). Кроме того, для разного оборудования эти системы координат могут быть различны как по направлениям координатных осей, так и по размерам области вывода. Совершенно очевидно, что для отображения трехмерных сцен такой метод прямо применить нельзя.
- Функции мощной графической библиотеки определяют координаты изображаемых объектов относительно начала отсчета произвольной декартовой системы координат. Геометрическое описание в этой системе координат может задаваться в любой удобной форме, как с использованием целых чисел, так и **чисел с плавающей запятой**, без учета ограничений для отдельных устройств вывода (единственным ограничением является диапазон представления целых или вещественных чисел в компьютере). Возможно связывать эти числа с любыми единицами измерения, выбираемыми в соответствии со спецификой решаемой задачи .
- В общем случае в процессе создания и изображения сцены используется несколько различных декартовых координатных систем.
  - Во-первых можно задавать формы отдельных объектов в отдельной системе координат для каждого объекта. Такие системы координат называются координатами моделирования, локальными или главными координатами. В качестве примера можно составить велосипед, задав каждую из его деталей (колеса, раму, седло, руль, шестеренки, цепь и педали) в отдельной системе координат моделирования. Если оба колеса велосипеда одинаковые по размеру, то в локальной системе координат можно описать только одно колесо.
  - Задав формы отдельных объектов, можно составить сцену («создать ее модель») путем расстановки объектов по соответствующим местам в системе координат сцены, которая называется внешней системой координат. Этот этап подразумевает преобразование отдельных систем координат моделирования в координаты с заданным положением и ориентацией относительно внешней системы координат.
  - Для описания не слишком важных сцен часть объектов можно вставлять непосредственно в общую структуру сцены во внешних координатах, пропуская этапы задания координат моделирования и преобразования координат моделирования во внешние координаты.

# Конвейер наблюдения

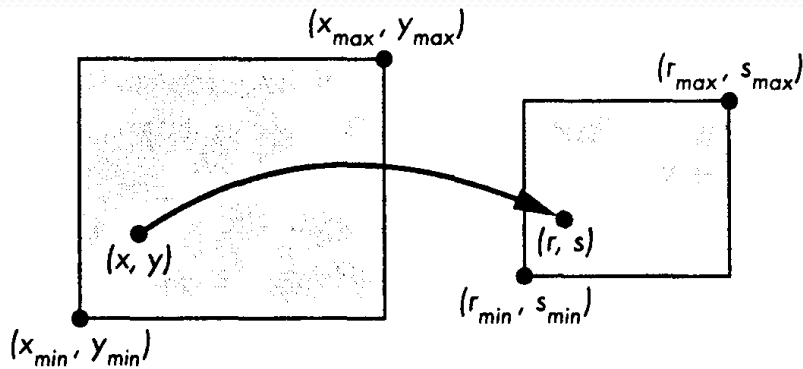


Рис. 2



Рис. 3

Конвейер наблюдения (viewing pipeline) – процесс преобразования элементов сцены, заданных во всех внешних координатах в изображение на экране в системе координат устройства вывода (рис.2 и 3). Рассмотрим подробнее этапы этого процесса:

- Вначале внешние координаты преобразуются в координаты наблюдения, соответствующие тому изображению сцены, которое мы хотим увидеть. В основе этой системы координат лежит положение и ориентация гипотетической камеры.
- После этого координаты объекта преобразуются в двумерную проекцию сцены, которая соответствует тому, что мы увидим на устройстве вывода.
- Затем эта сцена записывается в нормированных координатах, где значение каждой координаты попадает в диапазон от -1 до 1 или от 0 до 1.
- Затем графические драйверы преобразовывают нормированные координаты в целочисленные значения координат устройства вывода (экранные координаты в том случае, если это монитор). Т.о. нормированные координаты делают описание сцены независимой (device-independent graphics) от диапазона координат любого специального устройства вывода.
- Наконец, полученное описание попадает в буфер кадра растровой системы и превращается в изображение. Кроме двумерных экранных координат точки изображения в буфер кадра также может записываться информация о глубине точки, которая используется в различных алгоритмах визуализации и обработки поверхностей для определения видимых поверхностей или отсечения части рисунка, выходящие за пределы поля зрения (по глубине).

Конвейер наблюдения автоматически выполняется графической библиотекой в соответствии с проведенными настройками наблюдения сцены.

# Элементарные графические функции

- В состав мощного пакета API может входить несколько сотен функций, а потому желательно сразу же разделить их на категории:
  - Функции описания примитивов (primitive functions) определяют объекты нижнего уровня иерархии – примитивы, - которые способна отображать графическая система. В большинстве графических API имеются такие примитивы, как точки, отрезки прямых линий, многоугольники, пиксели, текст и различного рода криволинейные отрезки и участки криволинейных поверхностей. Для описания примитива требуется определить его положение. Например, прямоугольник задается через положения его углов, а сфера определяется через положение ее центра и радиус.
  - Функции задания атрибутов определяют, как будут выглядеть отображаемые объекты, т.е. атрибуты определяют способ вывода объектов на экран. Функции задания атрибутов (attribute functions) позволяют прикладному программисту выполнять широкий круг операций настройки изображения – от выбора цвета до указания образца заливки внутренней области многоугольника или шрифта для надписей на графике.
  - Функции визуализации (viewing functions) определяют параметры используемой модели синтезированной камеры, с помощью которой создается видимое изображение. От прикладного программиста требуется выбрать положение и ориентацию камеры во внешней системе координат и параметры объектива, в частности фокусное расстояние. Эти параметры позволяют правильно построить изображение и отсеять те объекты, которые оказываются вне поле зрения, чтобы не тратить времени на их обработку.
  - Функции источников света. Источник света характеризуется положением, интенсивностью и цветом излучения и его направленностью. В составе большинства API имеются функции для задания такого рода параметров, причем на сцене может присутствовать несколько источников света с разными характеристиками.
  - Функции задания свойств материалов объектов. Определяют оптические свойства материалов поверхностей объектов.
  - Функции геометрических преобразований (transformation functions) позволяют выполнять различные преобразования объектов – поворот, плоскопараллельный перенос, масштабирование и т.п.
  - Функции ввода графической информации (input functions) позволяют создавать интерактивные приложения. Эти функции играют роль промежуточного звена между устройствами ввода, такими как клавиатура, мышь, планшеты разного рода, дигитайзеры, и прикладной программой.
  - Специальные управляющие функции (control functions), которые позволяют управлять процессом выполнения программы, взаимодействовать с операционной системой, инициализировать приложение и обрабатывать ошибки других графических функций.

# OpenGL

- 1992 г. – разработана первая версия пакета OpenGL (Open Graphics Library - открытая графическая библиотека) как аппаратно-независимая версия пакета GL. Разработчиками OpenGL являются крупнейшие фирмы-разработчики как оборудования, так и программного обеспечения: Silicon Graphics, Inc., Microsoft, IBM Corporation, Sun Microsystems, Inc., Digital Equipment Corporation (DEC), Evans & Sutherland, Hewlett-Packard Corporation, Intel Corporation и Intergraph Corporation.
- Разработана специально для эффективной обработки трехмерных данных, но может работать и с описаниями двумерных сцен как с частным случаем трехмерного изображения, где все значения координаты z равны нулю.
- Программы, написанные с помощью OpenGL, можно переносить практически на любые аппаратные (персональный компьютер, графическая станция или суперкомпьютер) и программные платформы (Unix, Linux, SunOS, IRIX, Windows, MacOS, PlayStation3, Google Android и другие), получая при этом одинаковый результат.
- Существуют привязки пакета OpenGL к языкам C(C++), Ada и Fortran.
- Основная сфера использования – при разработке CAD-систем и научных приложений, а также компьютерных игр (Quake).
- Вспомогательные библиотеки, расширяющие возможности стандартной библиотеки OpenGL:
  - GLU (Graphics Utility Library) - библиотека графических утилит, которая включает функции формирования часто встречающихся сложных объектов (цилиндрических форм, сферических объектов, криволинейных поверхностей).
  - GLUT (GL Utility Toolkit) - библиотека, отвечающая за взаимодействие с системой окон. Содержит функции, обеспечивающие пользователя основными возможностями, характерными для большинства современных многооконных систем. Также поддерживает возможности ввода информации и взаимодействия с пользователем.
  - ...
- Стандарт OpenGL поддерживает функциональный интерфейс.
- Open Inventor – объектно-ориентированная библиотека классов для описания сцен, которые нужно изобразить с помощью функций OpenGL.

# DirectX

- 1995 г. - компания Microsoft представила первую версию библиотеки DirectX (тогда она называлась Game SDK). Все права на доработку/изменение DirectX принадлежат Microsoft.
- DirectX предназначен только для платформ под управлением ОС Windows.
- Стандарт DirectX основан на выборе интерфейсов Component Object Model (компонентная модель объектов), а объекты COM могут описываться практически любыми языками программирования, например C/C++, Delphi и даже Basic.
- Стандарт DirectX включает в себя модули поддержки:
  - программирования двумерной графики (модуль DirectDraw);
  - создания трехмерной графики (модуль Direct3D);
  - работы со звуками и музыкой (модули DirectSound и DirectMusic);
  - поддержки устройств ввода (модуль DirectInput);
  - разработки сетевых игр (модуль DirectPlay);
- Используется:
  - при разработке графических интерфейсов операционной системы Windows
  - разработчиками игр
  - ...

# OpenGL против DirectX

- Сегодня любая современная плата поддерживает и OpenGL и DirectX.
- В настоящее время подавляющее большинство прикладных программ, работающих с трехмерными объектами, опираются на одну из двух типовых библиотек – OpenGL или DirectX.
- OpenGL в основном используется в области профессиональных графических решений, но имеет все шансы занять главенствующее положение и в разработке обычных графических приложений.
- Большинство компьютерных игр (не говоря о графическом интерфейсе ОС Windows) используют исключительно библиотеку от Майкософт.

# Другие графические пакеты

- Были разработаны и многие другие библиотеки для программирования в области компьютерной графики.
- Некоторые из них предлагают общие графические стандартные функции, а некоторые предназначены для специальных приложений или отдельных аспектов компьютерной графики, таких как анимация, виртуальная реальность или графика в сети Internet.
- Язык моделирования виртуальной реальности – Virtual-reality Modeling Language (VRML), начинался как подмножество пакета Open Inventor, позволяет создавать трехмерные модели виртуальных миров в сети Internet.
- Изображения на Web-страницах можно создавать и с помощью графических библиотек, разработанных для языка Java:
  - Java 2D – для описания двухмерных сцен в рамках апплетов языка Java.
  - Java 3D – позволяет рисовать трехмерные Web-изображения.
- Библиотека Renderman Interface производства компании Pixar Corporation дает возможность генерировать сцены, используя разнообразные модели освещения.
- PHIGS.
- Графические библиотеки часто предлагаются как часть систем другого типа, таких как программы Mathematica, MatLab, Maple,...