Содержание:

Введение

Представление данных на мониторе компьютера в графическом виде впервые было реализовано в середине 50-х годов для больших ЭВМ, применявшихся в научных и военных исследованиях. С тех пор графический способ отображения данных стал неотъемлемой принадлежностью подавляющего числа компьютерных систем, в особенности персональных. Графический интерфейс пользователя сегодня является стандартом “де-факто” для программного обеспечения разных классов, начиная с операционных систем.

Существует специальная область информатики, изучающая методы и средства создания и обработки изображений с помощью программно-аппаратных вычислительных комплексов, – компьютерная графика. Она охватывает все виды и формы представления изображений, доступных для восприятия человеком либо на экране монитора, либо в виде копии на внешнем носителе (бумага, кинопленка, ткань и прочее). Без компьютерной графики невозможно представить себе не только компьютерный, но и обычный, вполне материальный мир. Визуализация данных находит применение в самых разных сферах человеческой деятельности. Для примера назовем медицину (компьютерная томография), научные исследования (визуализация строения вещества, векторных полей и других данных), моделирование тканей и одежды, опытно-конструкторские разработки.

В зависимости от способа формирования изображений компьютерную графику принято подразделять на растровую, векторную и фрактальную.

Отдельным предметом считается трехмерная (3D) графика, изучающая приемы и методы построения объемных моделей объектов в виртуальном пространстве. Как правило, в ней сочетаются векторный и растровый способы формирования изображений.

На стыке компьютерных, телевизионных и кинотехнологий зародилась и стремительно развивается сравнительно новая область компьютерной графики и

анимации.

Заметное место в компьютерной графике отведено развлечениям. Появилось даже такое понятие, как механизм графического представления данных (Graphics Engine). Рынок игровых программ имеет оборот в десятки миллиардов долларов и часто инициализирует очередной этап совершенствования графики и анимации.

Хотя компьютерная графика служит всего лишь инструментом, ее структура и методы основаны на передовых достижениях фундаментальных и прикладных наук: математики, физики, химии, биологии, статистики, программирования и множества других. Это замечание справедливо как для программных, так и для аппаратных средств создания и обработки изображений на компьютере. Поэтому компьютерная графика является одной из наиболее бурно развивающихся отраслей информатики и во многих случаях выступает “локомотивом”, тянущим за собой всю компьютерную индустрию.

1. Понятие виртуальности

С экранов телевизоров, со страниц компьютерной и некомпьютерной прессы все чаще слышится словосочетание '' виртуальная реальность ''. Что же скрывается под этим модным сегодня словом?

В первую очередь имеется в виду трехмерное, объемное изображение (в отличие от псевдотрехмерной графики на плоскости) и трехмерный звук.

Однако в полной мере ощутить всю прелесть виртуальной реальности можно только при наличии таких элементов, как детекторы перемещения, позволяющие отслеживать изменения положения пользователя в увязке с изображением на экране монитора и датчики, фиксирующие действия пользователя. До недавнего времени такие системы можно было увидеть лишь в крупнейших в мире игровых центрах, а стоимость их превышала все мыслимые значения. Но все меняется в современном мире компьютерной техники, и в результате постоянного снижения цен на компьютерное оборудование такие системы становятся доступными и рядовым пользователям настольных компьютеров. Более того, все чаще производители и игрового программного обеспечения, и бизнес – приложений встраивают поддержку виртуальной реальности в свои системы. А с середины 1995 года появилось несколько фирм, специализирующихся на выпуске такого программного обеспечения.

Системы виртуальной реальности в сочетании с ПК широко используются сейчас для развлечений. Они представляют собой различные более или менее сложные устройства, реагирующие на движения пользователя. Если несколько работающих систем виртуальной реальности соединить, образуется так называемое общее киберпространство, где пользователи могут встретить друг друга. Система отслеживания движений головы позволяет вам бросить взгляд в любую сторону киберпространства. А что в этом пространстве можно делать и что с вами произойдет - зависит от используемой прикладной программы.

Некоторые высококачественные системы виртуальной реальности используют специальные манипуляторы, подобные мыши и джойстику, передвижения которого вверх и вниз интерпретируется датчиками как движение пользователя вперед и назад. Это устройства дают дополнительную возможность передвижения в виртуальной реальности. Элитарные системы виртуальной реальности предлагают также стереоскопические 3D - изображения и стереозвук, а также возможность общаться с другими пользователями в едином киберпространстве с помощью встроенных микрофонов. Сегодня лучшие аркадные видеоигры поддерживают виртуальную реальность, что позволяет игрокам бороться не с генерируемым программой противником, а друг с другом.

Предметный мир, окружающий нас - трехмерный. Наши глаза воспринимают объекты под разными углами: два независимых изображения анализируются мозгом, и в результате их сопоставления формируется образ предмета, его признаки и глубина изображения. Расстояние между глазами человека обычно составляет 6-7 см, и когда зрачки сосредотачиваются на предмете, левый и правый глаз фокусируются в этом направлении. В зависимости от расстояния до объекта угол обзора изменяется. Наши глаза и мозг анализируют расстояние, основываясь на различии между изображениями, получаемыми левым и правым глазом. Это различие называют параллаксом зрения. Именно с помощью этого эффекта и создаются трехмерные объемные изображения.

2. История развития игровой графики

У игровой 3D графики, по меркам развития компьютерных технологий, долгая история. Она уходит корнями в те времена, когда программисты лишь пытались создать третье измерение для игр. На самом же деле такое 3D больше было похоже на очень сложное 2D. Простой пример принципов такого "движка". Ребра

всего окружающего изображаются линиями одного цвета. Боевая машина предстает в виде нескольких зеленых граней. Горы, представляющие собой далекий ландшафт прорисованы с той же тщательностью и тем же цветом. Перед игроком - перекрестье прицела столь же потрясающего качества, но красного цвета. В небе - Земля: круг в верхней части экрана, разделенный вогнутой линией, отсекающей воображаемую освещенную часть от не освещенной (они не отличаются по цвету друг от друга) и обращающей нашу планету в месяц. Да, да! Это она! Величайшая война на Луне во всей истории компьютерных игр! Оригинальная Battlezone, римейк которой был выпущен недавно. Подлинник же увидел свет в 1980 году. Он был создан для компьютеров Atari, и является первой настоящей 3D-игрой с видом от первого лица, первым танковым симулятором и первым симулятором какой бы то ни было боевой техники вообще.

3. Игровая графика и примеры ее применения

Создание игровой графики, художественное оформление игры – один из важнейших моментов процесса разработки. На это уходит львиная доля бюджета игры, а сама графика в значительной степени определяет то, что называется «атмосферой игры». Кроме того, хорошая графика – это еще и одно из условий успешных продаж: вспомните броскую рекламу в журналах, построенную на реальной внутриигровой графике, скриншоты в Интернете и на коробках с играми.

Вообще говоря, понятие «игровая графика» включает в себя так называемый концепт-арт, то есть эскизы и наброски, во многом определяющие то, как игра будет выглядеть, и собственно компьютерную – внутриигровую – графику. Как правило, художники, занимающиеся эскизами, работают в тесном сотрудничестве с дизайнерами игры. Они помогают конкретизировать замысел, создавая наброски героев и декораций. Иногда при этом выполняется и трехмерное моделирование. Далее в дело вступают компьютерные художники, непосредственно занимающиеся созданием персонажей (точнее говоря, текстурами, как для двухмерных спрайтов, так и для полигонов, из которых состоят 3D-объекты) и прорисовку задних планов (так называемого окружения), и художники-аниматоры (о них речь пойдет в следующей главе). В некоторых компаниях на помощь к ним приходят еще и специалисты, отвечающие за встраивание графики в движок игры.

3.1 Пространство

Трехмерное пространство в играх имеет координаты и соответствующие оси. Все, что мы видим или не видим: объекты стены источники света, основные элементы (спрайты, воксели, полигоны) - обладает координатами разного рода.

Самая главная система координат (почти всегда одинаковая)- это система координат, берущая отчет от виртуальной камеры, то есть относительно экрана. Чаще всего используется левосторонняя система координат. В таком случае точка пересечения всех осей (в которой все координаты нулевые) будет в левом нижнем углу экрана. Ось X будет уходить вправо по нижнему краю. Ось Y будет уходить вверх по правому краю. Ось Z будет уходить как бы вглубь. В случае правосторонней системой координат, точка пересечения осей, соответственно, будет справа.

Поскольку игровые объекты могут находиться в любой точке трехмерного пространства, вычислительная машина определяет, что, собственно, видно наблюдателю. Здесь определяется направление камеры и угол обзора. Для того чтобы не прорисовывать все, что находится в направлении взгляда (для повышения производительности и во избежание исчерпания ресурсов Z-буфера) задаются передняя и задняя отсекающие плоскости. Не выводится на экран то, что находится к виртуальной камере ближе передней отсекающей плоскости и дальше задней отсекающей плоскости. С этим явлением все встречались не раз. В старых коридорных войнах времен Doom это делалось при помощи тьмы. Устанавливалась граница, дальше которой все было погружено в кромешный мрак. Присутствует это и сейчас. Например, в Star Wars: Rogue Squadron наличествует туман. В дневных миссиях прекрасно видно, как, во время полета, ландшафт выступает из бежевой пелены нам навстречу.

Таким образом, на экране появляется только то, что находится в зоне, подобной пирамиде. Эта зона определяется четырьмя прямыми и двумя плоскостями. Из углов экрана в глубь сцены уходят прямые, которые по мере удаления от его поверхности расходятся в разные стороны. Так определяется, до какой отметки вправо, влево, вниз и вверх "видит" виртуальная камера. Задняя отсекающая плоскость есть строго установленное расстояние, далее которого объекты не выводятся на экран. Она определяет пределы "видимости" в глубину. Наш взгляд перпендикулярен этой плоскости. Она оказывается в основании пирамиды. Передняя отсекающая плоскость находится прямо перед камерой и отсекает

объекты, находящиеся ближе нее к экрану. Все, что есть в этой "пирамидоподобной" зоне, проецируется на экран. Что бы определить координаты объекта на экране, к его вершинам применяется преобразование, которое отражает координаты трехмерного пространства на координаты экрана. Преобразование осуществляется с помощью матрицы размером 4x4. в обычном варианте, для получения двумерных вершины на экране, умножается вектор трехмерных координат в пространстве на матрицу преобразования. До недавнего времени эти вычисления выполнялись только на программном уровне. Компания AMD разработала технологию 3Dnow!, суть которой в том, что процессор может выполнять команды матричной математики, производя вычисления с плавающей точкой по принципу SIMD (Single Instruction Multiple Data, одна команда много данных), что существенно увеличило скорость преобразований в программах, использующих эти команды. Такие игры "взметнулись" на новый уровень производительности в расчетах с 3D. Вспомним Unreal! "Программка" бегала на AMD K6-II побыстрее, чем на "втором пне" с той же частотой. Для точности надо заметить, что не намного быстрее, поскольку операции с плавающей точкой из набора x86 у этого "камня" от Advanced Micro Devices исполнялись существенно медленнее, чем у Intel'овских "мыслящих кристаллов". Ответным шагом Intel, стало создание аналога 3Dnow! - 50 команд в составе Streaming SIMD Extentions от Intel, примененных в Pentium III, которые подняли производительность программных преобразований на более высокий уровень, поскольку Intel, в отличии от AMD, не стала торопиться и сделала куда более "мощную" технологию, "обгоняющую" 3Dnow! По всем возможностям. Жаль, что команды SSE пока еще не где не реализованы в играх, если не считать графических тестов, имитирующих реальную игру. Но, это все - программные расчеты. На них ориентировались до тех пор, пока корпорация nVIDIA не разработала графический процессор GeForce 256, выполняющий такие преобразования на аппаратном уровне. Это - наиболее эффективный способ на сегодняшний день, оставляющий позади все "софтверные" вычисления. Надо просто поиграть в Q3 на "карточке" с таким "камушком", и тогда быстро и без лишних слов "доходит" насколько хороша аппаратная трансформация.

Преобразования не происходят сами по себе. Для этого, а также для управления объектами почти всегда используются фреймы. Фрейм - это управляющие границы объекта с преобразованием, применяемым ко всем его потомкам (несколько заумно, но проще, извините, не получается). Если он представляется наглядно, то это "делается" в форме параллелепипеда. Куб, как известно, тоже является параллелепипедом, но с равными сторонами. Впрочем, многие из вас, наверное,

видели изображение фрейма. Те, кто пользовался программами для рисования 3D (3D Studio Max, Ray Dream STUDIO, Light Wave и т.п.) видели наглядное представление этого "явления". Оно выполняется для упрощения работы художника с объектом и изображается всякий раз, когда объект выделяется. Выделенный объект оказывается, как бы заключенным в параллелепипеде или кубе, у которого есть только тонкие грани, которые не мешают созерцать объект.

Управляющими фреймы являются потому, что объект, находящийся внутри, неподвижен относительно своего "обрамления" и движется только вместе с ним. То есть, проще говоря, для перемещения объекта надо переместить его фрейм. Возникает вопрос: а как же человек будет шевелить руками? В случае с полигонной моделью это происходит следующим образом. Фреймы присоединяются друг к другу по иерархии. Модель человека имеет свой фрейм, его рука - свой, привязанный к большему, а кисть - свой. Что бы модель шевелила кистью, должно измениться положение соответствующего фрейма относительно того, к которому он привязан. При этом происходит трансформация объекта, то есть внесение изменений в сцену от кадра к кадру при его перемещении, масштабировании и вращении.

Присоединенные к фрейму объекты могут трансформироваться относительно других фреймов. Для того что бы вычислить двумерные координаты объекта на экране, "машина" совмещает результаты преобразований всех фреймов, расположенных выше этого объекта по иерархии, и определяет окончательные преобразования объекта.

Большинство графических механизмов (или, просто "движков") позволяют избегать излишних преобразований, задерживающих процесс определения координат. Обычно сохраняется копия матрицы итогового преобразования каждого фрейма. Она получается умножением матриц всех его преобразований. В случае, если все вышестоящие по иерархии фреймы не изменились, итоговое преобразование можно не пересчитывать.

Чаще всего, камера, так же, как и другие объекты, имеет свой фрейм. Она тоже может быть объектом трансформации. Все фреймы должны быть прикреплены к какому либо старшему фрейму. На вершине иерархии находится единственный фрейм, который ни к чему не прикреплен (но все прикреплено к нему) - фрейм сцены.

3.2 Спрайты

Во времена спрайтовых "движков" фреймы практически не применялись. Хотя сейчас, при использовании спрайтов в полигонных "движках", фреймы у них могут быть.

Представить себе спрайт можно как плоскую картинку в трехмерном пространстве, повернутую лицевой стороной к наблюдателю. Монстр изображается на бором спрайтов, на которых он "запечатлен" в разных положениях. Спрайты перемещаются, меняя координаты в трехмерном пространстве, меняются картинки, изображающие монстра в нужной точке. При приближении к наблюдателю, пространственные координаты монстра меняются в соответствии системой координат "движка", приближая персонажа. Меняются и экранные координаты по оси Z. С приближением, изображение противника масштабируется в большую сторону, заполняя одной точек спрайта несколько точек экрана, а с удалением - в меньшую, так что на одну экранную точку приходится несколько "с поверхности персонажа" (их цвета смешиваются). При очень сильном приближении появляется эффект при котором монстр пугает нас не только клыками, шипами и прочими атрибутами его персоны, но и гигантскими прямоугольниками на своей поверхности.

Кроме того, у спрайтовых объектов наблюдается движение рывками. Это происходит оттого, что все построено по принципу мультфильма: одна фаза (положение персонажа, изображенное одной плоской картинкой) сменяется другой фазой, как сменяются рисованные кадры. Между двумя фазами есть участо движения, который просто не отображается.

Родоначальник жанра 3D Action (если не считать Wolfstein3D, который был не достаточно популярен для такого ярлыка), Doom был спрайтовым. Однако стены в его "движке" отображались обычным текстурным методом, о котором будет рассказано позже. Спрайтовыми были все 3D-игры выпущенные до Quake.

3.3 Воксели

Пиксель - это точка на плоскости, а воксель - в пространстве. В этом их геометрическое различие. Воксель имеет свой цвет и, обычно (если движок

неплохой), реагирует на освещение его изменением.

На основе вокселей разработано много "движков". Самые запоминающиеся - серия Voxel Space, разработанная NovaLogic. Новейший из них поддерживает аппаратные ускорители при наложении полигонов, то есть имеет полигонно-воксельную основу. Ничто не мешает "движку" совмещать полигоны, воксели и спрайты. Более того в большинстве воксельных механизмов есть и текстуры(например, на стенах, на боевых машинах и т.д.). NovaLogic гордится этим, потому что воксели накладываются программно, а полигоны с отрисованными плоскими картинками воксельных участков кадра - аппаратно.

Обычно на основе вокселей принято строить движки авиасимуляторов и квестов. В случае с авиасимуляторами воксели помогают смоделировать ландшафт земной поверхности. Накладывать полигоны в количестве, достаточном для гладкости поверхности было бы очень расточительно по отношению к вычислительным ресурсам "машины", так как у каждого полигона имеются не только свои пространственные координаты, но и вершины, грани, текстуры, на которых есть точки (тексели). Все это требует огромного количества вычислений. Сравнительно проще применять полигоны, там, где можно делать довольно плоские, в нужных местах поверхности, объекты и накладывать на них текстуры, на которых уже есть много точек. Это уменьшает количество затрачиваемых вычислительных ресурсов, которые при использовании вокселей требуются на пересчет координат каждой точки в пространстве.

Результат таков, что при моделировании гладких поверхностей используются полигоны, а при моделировании изгибающихся (например, ландшафтов) - воксели.

Воксели есть не только в симуляторах, но и в квестах. Например, "движки" от Blade Runner и Sanitarium (в русской версии Шизариум) могут похвастаться воксельными персонажами.

У вокселей есть один существенный недостаток. Но заключается в том, он заключается в том, что они не могут обрабатываться аппаратно. При всех тенденция перехода от программной реализации графики к аппаратной, до сих пор не придумано аппаратуры, которая могла бы работать с вокселями. Никто из производителей вычислительной техники даже не пытается заниматься этим вопросом. Аппаратная реализация полигонной трехмерной графики дошла до передачи видеоаппаратуре таких функций как трансформация и освещение, а в воксельной графике все, так же как и раньше, обрабатывается центральным

процессором за счет его универсальных вычислительных команд.

Проблема в том, что для обработки вокселей требуется много операций с плавающими точками, гораздо больше, чем при расчетах с полигонами. А, изготавливать дорогой микропроцессор для графики было бы не целесообразно. Поэтому, целочисленные расчеты по наложению текстур у центрального процессора "отняли" еще давно. Позже это сделали со многими другими функциями. В 1999 году появились такие технологии, используя которые совсем недорого сделать устройство для вычеслений с плавающей точкой в графическом процессоре. Его и сделали в nVIDIA GeForce 256. но, к сожалению, даже мощности такого устройства, какое создала эта корпорация в "видеокамне", недостаточно для расчетов с вокселями. Потому и не создана до сих пор конструкция, производящая операции с ними на аппаратном уровне.

3.4 Полигоны

Спрайты и воксели - это неотъемлемая часть современной 3D-графики, но чаще всего она основывается на полигонах. Полигоны впервые появились в том виде, в котором мы привыкли их воспринимать, в "хитовейшей" игре, одном из лучших 3D-Action своего времени (а, возможно и всех времен), великом Quake. Его "движок" мог накладывать невероятное по тем временам количество текстур. Все "геймеры" были "в шоке". Какой рывок в графике! Раньше текстуры накладывались только на стены в упомянутых выше коридорных войнах времен Doom. А в Quake каждый монстр оказался покрыт множеством таких рисунков. "Монстрятник" стал трехмерным, и его участники больше не "дергались" при резкой смене заранее прорисованных фаз (с этого момента персонажи игр с хорошими движками обрели собственные поверхности). Противник в этом хите мог плавно поворачиваться перед нами, потому что в каждом новом кадре можно было пересчитать его текущую позицию, которая не была заранее отрисованным "мультипликационным" спрайтом, а была результатом динамического перемещения, не ограниченного тесными рамками фаз. Плавное движение из кадра в кадр радовало глаз. Но ведь индустрия электронных развлечений не стоит на месте. Каждый день придумывается что-то новое, реализуется в одном из грядущих "движков", после чего мы сможем наблюдать эту свежепридуманную технологическую особенность в реально воплощенных в жизнь играх.

Полигонная технология создания трехмерной графики на сегодняшний день является самой развитой и продолжает прогрессировать с невероятной скоростью. "Движки" сменяются один другим, и, каждый раз, мы можем созерцать богатые возможности нового графического механизма. По этой причине в терминологии, связанной с наложением полигонов появилось (и появляется постоянно) много новых слов. Большинство из них, кроме функций графических механизмов, обозначают аналогичные возможности аппаратуры для 3D-графики.

До разъяснения таких терминов как билинейная, трилинейная, анизотропная фильтрация, мип-мэппинг и многих иных, следует объяснить три первоначальных слова, касающихся полигонной 3D-графики. Это - текстура, тексель и, собственно, полигон. Не зная значений этих терминов, бесполезно понять что-либо иное.

Полигон - это абстрактная геометрическая фигура, на которой основывается обсуждаемая разновидность графики. Полигон, выражаясь "абсолютно русскими" словами, - это многоугольник. То есть у этой фигуры может быть три и более углов. В графике у полигона - три угла по той причине, что большее количество углов не приносит пользы, но "пожирает" вычислительные ресурсы нашего "железного друга". Конечно, если графика обрабатывается на аппаратном уровне, то никакие распределяемые ресурсы не затрачиваются, но графический процессор (или их набор) должен быть более совершенным и мощным. Реально же в терминах программистов, полигон - это геометрический примитив, то есть простейшая фигура.

Полигон, как сказано выше, абстрактен. Он не имеет внешнего вида - это факт, но тогда сам собой напрашивается вопрос, а что же мы видим на экране? Просто, на каждый полигон "натягивается" текстура. Текстура - ни что иное, как обычный растровый (т.е. точечный) рисунок, покрывающий поверхность полигона и придающий ему определенный вид. Рисунок масштабируется, поворачивается и цвет его точек (текселей) переносится на экран. При удалении рисунка от экрана (то есть при увеличении значения по оси Z) происходит уменьшение значения растра. При приближении происходит увеличение растра, и он, так же, как и в случае со старыми спрайтовыми "движками", пугает нас огромными текселями, "закрывающими" по несколько пикселей (точек на плоской поверхности экрана).

Но Quake не остался в таком виде - появилась его особая версия GLQuake, основывающаяся на работе 3D-ускорителей "через" интерфейс прикладного программирования OpenGL. В ней были применены новейшие (в те времена) технологии: альфа смешение, билинейная фильтрация, мип-мэппинг, дымка и

многое другое. Конечно, в великом Q (точнее сказать - GLQ) было реализовано много "красивостей", но еще больше их оказалось в Q2.

Заключение

Все области применения – будь то инженерная и научная, бизнес и искусство – являются сферой применения компьютерной графики. Возрастающий потенциал ПК и их громадное число - порядка 100 миллионов - обеспечивает соблазнительную базу для капиталовложений и роста. Неизвестно как долго продлиться тенденция удвоения капиталовложений, особенно под воздействием цен, однако ожидается устойчивое 10% ежегодное повышение в последующие 5 лет.

Сегодня особенно привлекательны для инвесторов компании, специализирующиеся на графических интерфейсах пользователя, объектно-ориентированных программах, виртуальной реальности и программном обеспечении параллельных процессов. Мы вступаем в новую эпоху расширения полномочий графических систем при движении по информационной супермагистрали.

Список использованной литературы

1. Информатика: Базовый курс/С.В. Симонович и др. – СПб.: «Питер», 2001.
2. http://tl78.net/publ/8-1-0-10
3. http://www.ito.su/2003/tezis/II-4-2672-Publication.html
4. http://blitz-3d.narod.ru/teoria/zalc/glava10/
5. http://gameconstructor.narod.ru/clauses/games_as_it_is_done/game_the_schedule.htm