

УДК 681.3.01

Д.Ю. Вишнякова, Н.А. Надолинский**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ТРЕХМЕРНЫХ СЦЕН**

В настоящее время практически в любой сфере программирования используются графические изображения. В перспективных тренажерных комплексах, а также комплексах управления подвижными объектами широко используются принципы виртуальной реальности или, другими словами, трехмерной графики. В настоящей работе предлагается к обсуждению разработанная нами программа, создающая трехмерную сцену, которая может послужить основой для создания систем виртуальной реальности.

В понятие трехмерной сцены входят:

- ◆ Изображение местности (ландшафт, небо).
- ◆ Статические (неподвижные) объекты.
- ◆ Динамические (подвижные) объекты.

Каждый из этих пунктов включает в себя текстурирование полигонов, прозрачность, и для удаленных объектов – туман.

К изображению местности можно отнести, во-первых, ландшафт. Ландшафт генерируется при помощи фракталов. Как известно, фрактальное изображение создается применением некоторой геометрической операции к геометрической фигуре и последующим многократным применением той же операции к полученному результату. Чтобы сгенерировать фрактальный ландшафт, достаточно присвоить случайные высоты трем вершинам равностороннего треугольника, а затем изогнуть каждое ребро, поднимая или опуская его середину на случайную высоту. Соединив линиями середины трех сторон, мы получим четыре треугольника. Если теперь применить операции изгиба и деления к каждому из получившихся треугольников, то вскоре у нас получится нечто, невероятно похожее на реальный ландшафт. Конечно же в действительности генерация фрактальных ландшафтов не сводится к примитивному алгоритму “изогнуть, разделить, рекурсивно повторить”, ибо существует достаточно проблем, связанных с хранением, делением, а также изображением ландшафта.

Касаясь изображения ландшафта, а также всех статических и динамических объектов сцены возникает проблема удаления невидимых поверхностей. Вся сцена со всеми ее компонентами состоит из более чем нескольких десятков тысяч полигонов. От алгоритма удаления невидимых поверхностей будет напрямую зависеть производительность всей системы (под производительностью понимается количество кадров в секунду при выводе изображения), т.к. зачастую мы обзораем не всю сцену сразу, а лишь отдельную ее часть, включающую в себя несколько сотен или тысяч полигонов. В данном случае наиболее целесообразно использовать метод двоичного разбиения пространства (Binary Space Partitioning Tree), т.е. построить BSP-дерево для всего статического мира. Вся суть метода состоит в разбиении всей сцены на так называемые блоки (кластеры). Причем при разбиении выбирается плоскость, которая делит пространство на два полупространства (кластера). Рекурсивно повторяя описанный процесс для каждого из полученных кластеров, по-

лучаем дерево, ветвями которого являются разбивающие плоскости, а листьями полигоны, составляющие сцену.

Одним из основных преимуществ предложенного метода является полная независимость дерева от параметров проектирования (положения центра проектирования, направление проектирования и др.), что делает его весьма удобным для построения серий изображений одной и той же сцены из разных точек наблюдения. Это обстоятельство привело к тому, что BSP-деревья стали широко использоваться в ряде систем виртуальной реальности, в частности, широко известны в играх типа Doom, Quake, Quake II, Quake III.

Для динамических объектов также можно построить свое BSP-дерево, которое при проецировании будет вклеиваться в общее BSP-дерево.

Итак, после создания всей сцены, а также удаления невидимых поверхностей следовало бы отметить способ изображения всей картины на экране.

При написании программ для операционной системы Windows нецелесообразно было бы использовать стандартный интерфейс GDI (Graphic Device Interface) для вывода на поверхность окна, т.к. GDI является простым и достаточно медленным интерфейсом, что не приемлемо при работе с 3D-графикой. Наиболее эффективно было бы использовать уже ставшие стандартом в области графики библиотеки OpenGL (SGI) и Direct3D (Microsoft). Также сюда можно было бы отнести и библиотеку Glide (3dfx), но она поддерживает только чипсеты одной фирмы – 3dfx, что делает ее непопулярной. Direct3D при программировании является исключительно неудобной библиотекой, нежели OpenGL, однако эффективно (если не наиболее эффективно) использует ресурсы видеоадаптеров нижнего ценового диапазона (непрофессиональных). OpenGL является стандартом де-факто для всех профессиональных графических адаптеров. Именно поэтому предпочтительнее было бы использовать ту или иную библиотеку, в зависимости от того, на каком компьютере использующая ее программа будет выполняться.

Предлагаемая программа создания сцен написана на языке Delphi 5.0, с использованием OpenGL для вывода графических объектов. При каждом запуске программы осуществляется генерация ландшафта, как следствие построение BSP-дерева, а затем вывод изображения в реальном режиме времени с обязательным использованием текстурирования, освещенности, тумана и прозрачности.

Программа разработана в рамках работ по мультимедийным системам, проводимых международной лабораторией ELDIC.

УДК 007.681.518.2

И.В. Гречин

НОВЫЙ ПОДХОД К ЭКСПЕРТНОЙ СИСТЕМЕ В ТЕХНОЛОГИИ ОБУЧЕНИЯ

В настоящее время большое внимание уделяется разработке обучающих программ для нужд образования. Создано множество систем, сделано много попыток создать «электронного учителя». Можно уже смело говорить, что опыт в этом направлении достаточно накоплен и не стоит тратить много времени на перечисление всех достоинств этих программ, остановимся на некоторых немаловажных моментах.