Урок 6. Первый графический движок, график функции.

Настало время освоить первый графический движок для нашей будущей игры. Конечно его возможности не самые совершенные, но зато знаний первых пяти уроков хватить, чтобы построить на его базе небольшую игру.

Как и любой графический движок, он содержит много функций. Для удобства работы с ними их объединяют в единый **модуль.**  Чтобы подключить *модуль* работы с графикой в начале нашей программы мы должны добавить строчку:

**uses** GraphABC;

После оператора uses идёт имя *модуля*. Вот список необходимых нам для работы программы стандартный функций данного *модуля*.

SetWindowWidth(w: integer) – устанавливает ширину окна в точках(пикселах), равную w. Стандартная ширина экрана составляет 1980 точек. Поэтому если ширина окна будет равна 960 точек, то окно займёт половину экрана компьютера.

SetWindowHeight(h: integer) – устанавливает высоту окна в точках, равную h. У стандартного экрана высота составляет 1080 точек.

ClearWindow() – очищает экран.

MoveTo(x, y: integer) – устанавливает позицию кисти в точку с координатами (x, y). При установки позиции надо помнить, что в компьютерной графике позиция отсчитывается от **верха**  окна **вниз**. Верхняя левая точка имеет координаты (0,0).

LineTo(x, y: integer) – рисует линию от текущей позиции кисти до указанных координат x и у.

SetPenColor(c: Color) – устанавливает в качестве текущего цвета пера цвет с. Обратите внимание, что функция ожидает от нас данный до этого незнакомого нам типа Color. Для простоты работы мы будем использовать стандартный константы модуля GraphABC : сlRed для задания красного цвета, clGreen для зелёного и clBlue для синего.

Также нам понадобиться использовании функции Round. Данная функция не входит в модуль GraphABC, а занимается округлением дробного числа(Double) до целого(Integer).

Откройте программу lesson6.pas .

Как видно, сначала мы устанавливаем размер экрана, очищаем экран . Затем устанавливаем две линии координат. Их пересечении будет точкой 0,0 на нашем отображении графикока функций. Затем мы устанавливаем цвет нашего линии – красный.

Как понятно, мы будет отображать графики по традиционной системе. То есть положительное значение верху, а отрицательное – внизу. Также мы должны пересчитывать точки в математические значения и обратно. Для этого надо написать соотвествующие функции.

После этого мы используем цикл, где последовательно переводим координату положение точки в координату x, потом вычисляем значения нашей функции(в данном случае это квадратное уравнение вида y=a(x\*x)+bx+c ) . После этого опять переводим в положение точек и отображаем. Как видим математические вычисления нашей программы отделены от построения. Они выполняются в разных системах координат. Как правило, в играх и других программах так делается всегда, чтобы при смене графического движка на более совершенный все математические операции в программе оставались бы неизменными.

Запустите программу. Как видим, она верно отображает график функции.

К сожалению, график не очень информативен. Мы видим не можем понять, в какой точке наша функция принимает значение 0. Чтобы сделать программу удобнее, выполните следующие задания.

1. Перед отображением графика функции зелёным цветом нарисуйте сетку, по которой были бы видны единицы. Для этого вам надо написать функцию XToPixels , а также использовать цикл for.
2. Попробуйте поменять константу Scale для изменения масштаба графика.
3. Синим цветом выведите другой график функции, например y=kx+b

Вывод вы можете сделать в цикле, проверяя, например, если l=0, то устанавливается красный цвет и вычисляется функция Square , а если l=1, то устанавливается синий цвет и вычисляется ваша функция . Всё остальное в цикле остаётся неизменным.