«ИНФОРКОМ»

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР «ZX – SPECTRUM»

Динамическая графика

Москва – 1995

<< ИНФОРКОМ >>

ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР

<<ZX-SPECTRUM>>

Динамическая графика

Москва - 1995

Книга является третьей частью четырехтомной серии, посвя
щенной вопросам исполнения графики на персональном компьютере
Spectrum. Рассмотрены вопросы анимации изображений в растро-
вой и векторной графике. Приведены практические приемы и полез-
ные советы для тех, кто занимается разработкой компьютерных игр.

К читателю!

Уважаемый читатель! В Ваших руках книга, посвященная динамической графике компьютера ZX-Spectrum. Это третий том нашей графической серии.

Широкое распространение компьютера **ZX-Spectrum** по просторам России, Украины, Белоруссии, республик Прибалтики и других, вызвало вполне естественное желание многих способных программистов писать для него программы. Наличие миллионов экземпляров этой домашней ЭВМ на руках у пользователей делает вполне осмысленной и рентабельной самостоятельную разработку игровых, обучающих и прикладных программ. Хорошие программы отечественных авторов уже сейчас могут расходиться тиражами от нескольких сотен и даже до нескольких тысяч экземпляров. И в наших республиках уже сложились авторские коллективы, работающие над программным обеспечением не просто ради интереса, но и с коммерческим успехом.

Одно из основных требований к хорошей коммерческой программе - это привлекательная графика. Встречают, как известно, "по одежке" и только провожают "по уму". Вот и получается, что сколь бы ни была умна и содержательна Ваша программа, ей надо делать "красивую одежку", то есть использовать в ней приемы и методы работы с графикой. Этому и посвящены книги нашего четырехтомного сериала.

В первом томе ("Элементарная графика") мы рассказали о том, как исполняется графика на экране из БЕЙСИКа и из машинного кода. Мы показали, что основная задача, стоящая перед программистом - найти соответствие между адресом в экранной памяти компьютера и координатами изображения на экране. Мы рассказали о том, как это делается с использованием стандартных процедур ПЗУ или с помощью собственных процедур.

Опираясь на эти сведения, как на базовые, во втором томе серии ("Прикладная графика") мы привели массу полезных советов, приемов и примеров по тому, как исполняется плоская и трехмерная графика в программах.

"Динамическая графика" - наиболее сложная часть работы. Здесь необходимо не просто уметь создавать изображения на экране, но и изменять их и перемещать. Все это должно делаться быстро и требует безусловно хорошего знания машинного кода и навыков в написании процедур на АССЕМБЛЕРе. Если Вы пока этим не обладаете, Вам целесообразно приобрести и прочитать первые две книги серии.

Все, что касается графики "Спектрума", невозможно изложить ни в какой книге, как бы велика она ни была. При подготовке этого издания мы стояли также и перед проблемой необходимого "упрощения" материала до минимально возможного уровня и, естественно, многое при этом потеряли. Компенсировать нехватку информации по тем или иным вопросам Вы всегда можете, если будете читать наше регулярное периодическое издание "ZX-РЕВЮ", которое выходит с 1991 года с периодичностью 6 раз в год.

И последнее. Знание машинного кода, в отличие от БЕЙСИКа, не может прийти само по себе. Для этого надо иметь информацию и самому начинать экспериментировать с разными командами. Если Вы только собираетесь этим заниматься, Вам может быть непонятной большая часть этой книги. Но не пугайтесь. У нас есть книга "Программирование в машинных кодах", по которой уже десятки тысяч человек самостоятельно, без учителей и курсов освоили машинный код и АССЕМБЛЕР Z80. Наши читатели называют ее "Библией для Синклеристов".

Все, что Вам может потребоваться в работе, изучении, освоении новых материалов, Вы всегда можете приобрести у нас. Практика нашей работы организована так, что мы постоянно допечатываем все свои труды. "Программирование в машинных кодах", например, уже выдержало более 5-ти переизданий, "Элементарная графика" - три и т.д.

Обращайтесь. Все, что нами сделано за более чем пять лет работы, всегда в Вашем распоряжении.

Ваш "Инфорком".

Адрес для обращений: 121019, Москва, а/я 16, "Инфорком"

1. ВВЕДЕНИЕ

Анимация - это тот прием, с помощью которого в компьютерные игры вдыхается жизнь. Возможность компьютера создавать, хранить, изображать на экране и перемещать изображения едва ли не стала наиболее популярной чертой персональных ЭВМ.

Термин "компьютерная графика" для разных людей может означать различные вещи, но обычно под ним понимается именно создание изменяющихся изображений. Для бизнесмена понятие "компьютерная графика" может заключать в себе изображение графиков, диаграмм, гистограмм и других видов визуального отображения финансовой и деловой информации. Диспетчер на железной дороге, может быть удовольствуется в качестве "компьютерной графики" мигающими квадратиками на дисплее, изображающими приближающиеся составы, а для курсанта летного училища за понятием "компьютерная графика" скрываются авиатренажеры, способные как можно более реалистично имитировать все этапы полета от взлета до посадки с непрерывно меняющимся видом из окна пилотской кабины. Еще более широкий смысл в понятие "компьютерная графика" вкладывают те, кто имеют дело с системами автоматизированного проектирования, например при разработке дизайна легковых автомобилей.

В значительной степени "компьютерная графика" может быть статичной, когда неподвижные изображения, отражающие необходимую информацию, выдаются на экран по требованию пользователя, но в тех случаях, когда нужно максимальное приближение к действительности, когда нужно создать пользователю эффект присутствия и сопричастности, когда нужно вдохнуть элемент реальной жизни в программу, на арену выходит анимация изображений.

Искусство анимации (а это именно искусство) появилось задолго до появления компьютеров. Наиболее известно оно по муль-

типлнкационным фильмам, которыми наслаждаются миллиарды людей. Попробуем вдуматься в то, какой огромный труд, вдохновенный талант и практический опыт стоят за каждой минутой одного мультфильма. Ведь в каждую секунду на экране изображаются двадцать четыре кадра, а каждый кадр - это отдельное изображение, показывающее различные стадии движения анимафицируемого объекта. Не требуется больших способностей в арифметике, чтобы подсчитать, сколько труда вкладывают художники в один десятиминутный мультфильм.

Мультфильм - это пример рисованной анимации. Художник готовит серию отдельных изображений, которые будучи засняты на пленку в нужной последовательности, создают впечатление движущегося объекта. Но существуют и другие приемы анимации, при которых эффекта движения объекта не возникает, тем не менее, это тоже анимация. Например, в кукольной мультипликации часто применяют такой прием, когда камера как бы показывает объект с разных сторон. Обратите внимание на то, что сам объект остается неподвижным, но, тем не менее, это тоже анимация. Нередко применяют "наплыв", когда камера как бы "наезжает" на неподвижный объект и он увеличивается на экране, при этом могут проявляться новые его детали, и это тоже анимация. Так что, как видите, в анимации можно управлять не только изменением формы и местоположения объекта, можно управлять и камерой, которая этот объект снимает.

И, кстати сказать, именно эта техника позволяет наиболее полно использовать вычислительные способности компьютеров и именно здесь в максимальной степени компьютеры могут помочь художникам-аниматорам, освободив их от рутинной работы многократного перерисовывания один раз созданного изображения.

Художники-аниматоры, работающие в киноиндустрии - это очень квалифицированные люди, успех которых во-многом зависит не только от художественного дара, но и еще от двух факторов, которые освоены многими поколениями мультипликаторов, но, тем не менее, требуют многочисленных проб и ошибок и огромного

практического опыта.

Первый фактор - это соблюдение режима реального времени. Имеется в виду, что последовательность кадров, изображающих отдельные элементы движения, должна быть продумана так, чтобы движущиеся объекты на экране перемещались с реальной скоростью.

Второй фактор - натуралистичность. Это означает, что движущиеся объекты должны подчиняться элементарным законам физики, описывающим движение объектов, и, в частности, закону всемирного тяготения.

Здесь также на помощь может прийти компьютер, поскольку трудный путь проб и ошибок легче проходится в компьютерном эксперименте.

Итак, компьютер может найти самое широкое применение в киноиндустрии, и это применение уже началось. Многие, наверное, видели захватывающие кадры межзвездных битв в кинозпопее Стивена Спилберга "Звездные Войны", в подготовку которых вложен труд многих сотен программистов. Уже появились и первые мультипликационные фильмы, подготовленные полностью с помощью компьютера, но нас сейчас интересует несколько другой вопрос. Не использование компьютеров в киноиндустрии, а программные приемы, с помощью которых достигается анимация изображений - вот основная тема этой книги.

Вернемся к компьютерной графике. В предыдущих книгах нашей серии "Элементарная графика" и "Прикладная графика" мы позволили себе высказать соображение, что для "Спектрума" можно выделить три основные вида графики - растровую графику, векторную графику и блочную графику. В основу такой классификации положено не то, как выглядит экран при том или ином методе, а то, какие программистские приемы при этом задействованы.

Анимация включает в себя три основных этапа - подготовку изображения, переработку изображения и печать изображения. Пол

готавливается изображение с помощью утилитных программ типа графических редакторов или с помощью специально для этого написанных процедур. Вопросы печати изображения на экране мы рассмотрели для всех трех видов графики в предыдущей книге, а в данной книге остановимся на втором этапе - переработке изображения.

Блочная графика предполагает печать из БЕЙСИКа по команде PRINT AT или из машинного кода по команде RST 10H. При этом печатаемые символы (а они могут быть заданы пользователем в виде графических примитивов) могут сами по себе представлять некоторые графические образы, а могут и входить составной частью в более крупные изображения. Нижеприведенная БЕЙСИК-программа демонстрирует принципы анимации на блочной графике и она достаточно показательна для понимания сути процесса анимации. Сначала изображение строится в заданном месте экрана, затем оно стирается и, наконец, строится новое изображение, несколько смещенное относительно предыдущего, после чего цикл повторяется. Так возникает иллюзия движения изображения по экрану.

Листинг 1
Пример простейшей анимации

```
10 CLS
20 LET A$="▀▄"
30 LET B$="  "
40 FOR k=0 TO 21
50 PRINT AT k,12; A$
60 PAUSE 5
70 PRINT AT k,12; B$
80 NEXT k
90 GO TO 40
```

Одной из модификаций этой техники является оконтуривание двигающегося изображения пустыми знакоместами для "затирания" ранее имевшегося на экране содержания. Итак, техника анимации включает в себя рисование, задержку, стирание и новое рисование. Блочная графика проста и понятна, она иногда находит

тойное применение, но по выразительности все-таки значительно уступает растровой графике.

Растровая графика. Если в блочной графике основным элементом является блок 8*8, равный одному знакоместу, то в растровой графике основным элементом изображения является точка. Основную роль в растровой графике имеет разрешающая способность экрана, т.е. сколько точек может быть уложено в единицу площади. "Спектрум" имеет вполне пристойное разрешение - 256 точек по горизонтали и 192 точки по вертикали. Каждая точка (пиксел) может индивидуально включаться, переключаться или проверяться.

Если бы нужно было по экрану двигать только одну точку, то техника программирования была бы той же, что и для блочной графики. Включение пиксела - пауза - выключение пиксела - включение в новом месте со смещением. К сожалению, это неинтересно и пикселы объединяют в группы для получения какого-либо изображения. С этим шаблоном, состоящим из пикселов, работают точно так же, просто имеют дело с каждым индивидуальным пикселом по очереди. Такие растровые шаблоны, представляющие из себя независимые управляемые объекты, называют **спрайтами**.

Итак, спрайтами можно управлять, организуя их перемещение по экрану по заданному закону. Но в концепции спрайтов есть еще один очень важный момент - это **детекция коллизии** спрайтов. Когда один спрайт сталкивается на экране с другим, или когда он сталкивается с каким-то участком фоновой картинки экрана, программа, управляющая движением спрайтов должна определить, с чем столкнулся спрайт, принять решение и передать управление процедуре, которая обработает возникшее столкновение (коллизию). Примеров можно приводить сколько угодно. Если Вы играете в "Arkanoid", то в момент коллизии шарика и ракетки должна включиться процедура, которая изменит направление движения шарика так, чтобы согласно законам физики угол падения равиялся углу отражения. Точно так же при коллизии шарика и кирпичной стенки должна включиться процедура, изображающая "взрыв" кирпича и разлетание осколков. В качестве примера мы привели небольшую

программу на БЕЙСИКе, которая проиллюстрирует концепцию анимации в растровой графике и технику детекции коллизии.

Листинг 2
Пример детекции коллизии

```
 10 CLS
 20 FOR k=1 TO 20: PRINT AT k,1;"■":
    PRINT AT k,30 "■"
 30 NEXT k
 40 FOR k=1 TO 30: PRINT AT 1,k;"■":
    PRINT AT 20,k "■"
 50 NEXT k
 60 GO TO 150

 69 REM Подпрограмма печати точки
 70 PLOT x,y
 80 PAUSE 4
 90 RETURN

 99 REM Подпрограмма стирания точки
100 PLOT OVER1; x,y
110 RETURN

119 REM Подпрограмма детекции коллизии
120 IF POINT (x+xdir,y+ydir)=1
    AND POINT (x+xdir,y+ydir+1)=1
    AND POINT (x+xdir,y+ydir-1)=1
    THEN LET xdir=-xdir
130 IF POINT (x+xdir,y+ydir)=1
    AND POINT (x+xdir+1,y+ydir)=1
    AND POINT (x+xdir-1,y+ydir)=1
    THEN LET ydir=-ydir
140 RETURN

149 REM Головной цикл
150 LET x=70: LET y=100: LET xdir=-3.1: LET ydir=1
160 GO SUB 120: REM Проверка коллизии.
```

```
170 GO SUB 70: REM Печать точки.
180 GO SUB 100: REM Стирание точки.
190 LET x=x+xdir
200 LET y=y+ydir
210 GO TO 160
```

Векторная графика. Как в блочной графике основным элементом изображения является элементарный блок 8*8, а в растровой - точка, так в векторной графике основным элементом изображения является линия, а точнее - отрезок прямой. Все изображения, в том числе и объемные, строятся из совокупности отрезков. Такие изображения могут как двигаться по экрану, так и вращаться вокруг собственных осей. Векторная графика - вычисляемая и при всех манипуляциях с изображением программа рассчитывает новые координаты начала и конца каждого отрезка, из которых изображение составлено. Основной вычислительный аппарат при этом - чисто математический, основанный на формулах, известных из геометрии и тригонометрии. Любители игровых программ сталкиваются с подобным методом воспроизведения графики довольно часто. Поскольку это самый "быстрый" вид графики, его широко применяют в программах-имитаторах, в том числе и в имитаторах космических кораблей. Нижеприведенная БЕЙСИК-программа демонстрирует изображение куба средствами векторной графики.

Листинг 3
Простейший пример векторной графики.

```
10 LET ox=50: LET oy=40
20 LET a=30: LET b=40: LET c=1: LET d=20
30 PLOT ox,oy
40 DRAW 0,a: DRAW b,0: DRAW 0,-a: DRAW -b,0
50 DRAW c,d: DRAW b,0: DRAW -c,-d
60 PLOT ox+0,oy+a: DRAW c,d: DRAW b,0: DRAW -c,-d
70 PLOT ox+c,oy+d: DRAW 0,a: DRAW b,0: DRAW 0,-a
80 LET a=a+1: LET b=b+1: LET c=c+1: LET d=d+1
90 STOP
```

2. АНИМАЦИЯ В РАСТРОВОЙ ГРАФИКЕ

2.1 Анимация переднего плана. Спрайты.

Итак, **СПРАЙТ** - это некоторый графический объект, который может перемещаться по экрану без искажения заднего плана. Спрайты всегда прямоугольны (в частном случае - квадратны). Размеры спрайтов могут быть любыми Вам удобными. Нередко их выбирают так, чтобы ширина и высота были кратны восьми, поскольку одно знакоместо на экране имеет размер 8x8 пикселов. С другой стороны, поскольку для обеспечения плавности перемещения спрайтов в разные стороны, их двигают не на знакоместо, а только на пикселы, нет никакой особенной необходимости привязывать их размеры к знакоместам.

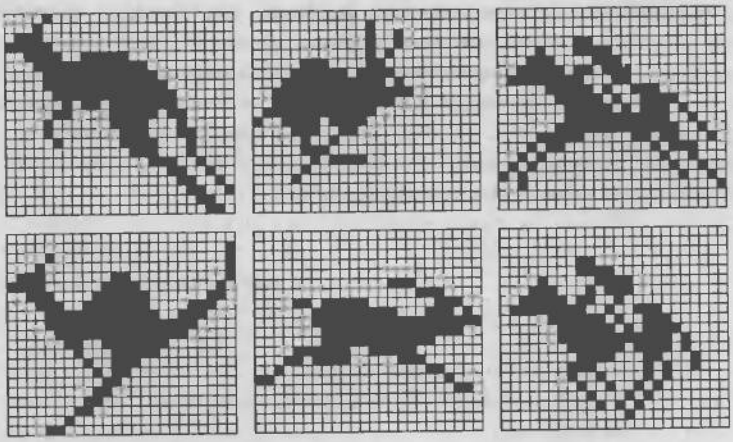


Рис. 1 Примеры спрайтов.

В рамках данной книги мы будем иметь дело с общим случаем, когда размеры спрайта не кратны восьми и примем за базовый размер спрайта 24 x 21 пиксел. Этот размер широко применяется, он

достаточно велик для того, чтобы реализовать Ваши художественные способности, но не слишком велик, чтобы скорость операций со спрайтом стала критичной для программы. Выше, на рис.1 даны образцы иекоторых спрайтов (кенгуру, кролик, наездник), выполненных в данном формате.

2.1.1. Создание спрайтов.

Для создания спрайтов применяют специальные программы - **генераторы спрайтов**. Обычно генератор спрайтов - это упрощенная разновидность графического редактора. Он должен обеспечивать возможности создания, редактирования и сохранения спрайтов. Как правило, генератор спрайтов имеет в оперативиой памяти буфер, в котором можно разместить сразу несколько спрайтов для одновременной работы с ними. После того, как работа закончена, буфер сохраняется на ленте или диске. Широко известен генератор спрайтов, входящий в пакет **Laser-Basic**, а также сегодня существует немало адаптаций графического редактора **Artstudio**, позволяющих не только рисовать спрайты, но и "вырезать" их из готовых рисунков.

Если у Вас нет такой "специализированной" версии **Artstudio**, то кое-какнх успехов можно добиться с использованием режима *Capture Font* в обычной версии **Artstudio**. Это делается так: войдя в режим *Magnify X 8* в этом редакторе Вы можете изобразить любой желаемый спрайт.

Второй этап - "выделение" этого спрайта в виде окна, после чего можно войти в меню *Font Editor* и дать команду на "захват" шрифта - *Capture Font*. Все, что содержится в этом окне, будет расценено программой, как шаблоны символов 8X8. Предварительно надо только установить указатель текущего символа символьного набора на первый символ и проследить за тем, чтобы окно было изображено точно шириной 24 пиксела (ни больше, ни меньше).

Мы не описываем здесь эту операцию более подробно, пос-

кольку Вы можете об этом прочитать во втором томе нашего сериала - *"Прикладная графика"* в том разделе, где говорилось о создании шрифтов нестандартного размера.

В результате этой операции у Вас образуется как бы новый символьный набор, в котором шаблоны первых 9-ти символов будут содержать конструкцию спрайта. Этот символьный набор можно сохранить. Длина сохраненного блока будет 768 байтов. Из них только 63 байта определяют конструкцию нашего спрайта. "Вырезать" из символьного набора нужные 63 байта уже совсем просто, хотя надо сделать маленькую перекодировку, связанную с тем, что шаблон символа и шаблон спрайта несколько отличаются по порядку следования байтов (рис. 2,3). В шаблонах символов байты следуют сверху вниз, а сами символы - слева направо через восемь, в то время как в шаблоне спрайта - байты следуют сверху вниз, а слева направо - через двадцать один. Посмотрите на рис. 2,3 и Вам все станет ясно.

Нижние три строки в шаблонах символов - нулевые и их нужно "отрезать". Это происходит потому, что три символа по вертикали занимают 24 строки, а в нашем спрайте нам нужна только 21 строка.

Листинг 4
Программа-перекодировщик

```
10 LET adr1 = 30000          :REM Адрес загрузки симв. набора
20 LET adr2 = 54600          :REM адрес буфера создаваемых
                                  спрайтов
30 LET shift=0               :Специальный счетчик байтов,
                              которые не надо копировать
40 LOAD "" CODE adr1,768     :REM загрузка симв. набора
50 FOR a=1 TO 72             :REM количество байтов в
                                  шаблонах символов
60 IF a=54 OR a=55 OR a=56 OR a=62 OR a=63 OR a=64
   OR a=70 OR a=71 OR a=72
   GO TO 70                  :REM обход для последних строк
```

```
 70 POKE (adr2+a-1-shift),PEEK (adr1+a-1)
 80 LET shift=shift+1
 90 NEXT a
100 STOP
```

Шаблоны символов

Байт 1	Байт 9	Байт 17
Байт 2	Байт 10	Байт 18
.....	.....	.....
Байт 8	Байт 16	Байт 24
Байт 25	Байт 33	Байт 41
Байт 26	Байт 34	Байт 42
.....	.....	.....
Байт 32	Байт 40	Байт 48
Байт 49	Байт 57	Байт 65
Байт 50	Байт 58	Байт 66
---------- Нижние три строки необходимо "отрезать" ----------		
Байт 56	Байт 64	Байт 72

Рис. 2 Раскладка шаблонов 9-ти символов, определяемых пользователем.

Шаблон спрайта

1-ый байт	22-ой байт	43-ий байт
2-ой байт	23-ий байт	44-ый байт
.....	.....	.....
20-й байт	41-й байт	62-й байт
21-й байт	42-й байт	63-й байт

Рис. 3 Структура шаблона спрайта.

2.1.2. Формат спрайта.

Прежде, чем мы приступим к разбору процедур, управляющих спрайтами, нам надо договориться о том, в каком формате хранится информация о спрайте в памяти компьютера. Предположим, что мы остановились на спрайтах размером 24 X 21 пиксел. Всего такому спрайту необходимы 504 бита, т.е. 63 байта оперативной памяти. Расположение спрайта в оперативной памяти показано на рис. 4. на примере спрайта, изображающего краба. Нули соответствуют выключенным пикселам, а единицы, соответственно, - включенным.

Раскладка битов

```
0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 0
0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0
0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0
0 1 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0
0 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0
1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1
1 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1
1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1
0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0
1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1
1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0 0 1
1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
```

Рис. 4 Побитовая раскладка в спрайте 24 X 21.

2.1.3 Печать спрайта.

Прежде, чем заниматься анимацией спрайта, попробуем его хотя бы просто распечатать.

В этой книге мы будем придерживаться тех же принципов объединения БЕЙСИКа и машинного кода, которые были нами применены в книгах "*Элементарная графика*" и "*Прикладная графика*" и которые осиованы на передаче параметров в машинный код из БЕЙСИКа с помощью функций пользователя FN (), - см. "*Элементарная графика*", с. 109-111.

Итак, предположим, что где-то в оперативной памяти, например начиная с адреса **SPADR** (в наших примерах принято 54600) и выше, у Вас хранится таблица из нескольких спрайтов. Предположим для определенности, что в этой таблице всего 10 спрайтов и вся таблица занимает 63*10=630 байтов.

Процедура **FN f(x,y,n)** предназначена для печати спрайта с номером **n** из таблицы спрайтов таким образом, чтобы его верхний левый угол (позиция печати) приходился на знакоместо экрана с координатами **x** и **y**.

x - горизонтальная координата. Задается в знакоместах. Поскольку спрайт сам по себе имеет шнрину в три знакоместа, не следует задавать параметр **x** больше, чем 28.

y - вертикальная координата. Задается в знакоместах. Не может быть более, чем 20.

Листинг 5

Пример использования процедуры печати спрайта.

```
10 LET  adr=54100:      REM адрес загрузки процедуры
20 LET long=75:         REM длина процедуры
30 LET check=60377:     REM контрольная сумма
```

```
 40 LET sum=0              REM счетчик контрольной суммы
 30 FOR k=0 TO long-1:
    READ a:                REM ввод данных в компьютер
 40 POKE (adr+k),a:        REM формирование машинного кода
 50 LET sum=sum+a          REM накопление контрольной суммы
 60 NEXT k
 70 IF sum<>check THEN PRINT "??": STOP
 80 REM

100 REM *** Пример использования процедуры
110 DEF FN f(x,y,n)=USR 54100
120 BORDER 4: PAPER 4: INK 0: CLS
130 FOR p=1 TO 10
140 FOR k=1 TO 5
150 RANDOMIZE FN f(k*3-3, (k*4)-3, p)
160 NEXT k
170 NEXT p
180 PAUSE 0
190 CLS
200 RANDOMIZE FN f(10,10,9)
210 PAUSE 10
220 RANDOMIZE FN f(10,10,10)
230 PAUSE 0
240 GO TO 200

300 REM *** Данные для машинного кода
310 DATA 042,011,092,001,004
320 DATA 000,009,086,001,008
330 DATA 000,009,094,237,083
340 DATA 148,211,009,126,050
350 DATA 150,211,123,230,024
360 DATA 246,064,103,123,230
370 DATA 007,183,031,031,031
380 DATA 031,130,111,034,146
390 DATA 211,058,150,211,071
400 DATA 017,063,000,033,009
```

```
410 DATA 213,025,016,253,237
420 DATA 091,146,211,195,001
430 DATA 212,201,174,072,013
440 DATA 014,003,000,000,000
450 DATA 060,000,000,000,000
```

Если Вы предварительно потрудитесь и "закачаете" в буфер спрайтов, начиная с адреса 54600 и далее десяток шаблонов спрайтов, что займет 630 байтов, то сможете немного поэкспериментировать с приведенной здесь программой. Для того, чтобы было с чем "экспериментировать", мы привели несколько шаблонов спрайтов. Можете также воспользоваться теми спрайтами, которые есть в Приложении к данной книге.

Строка 10 определяет адрес процедуры печатн спрайта. Она передает три параметра в машинный код - координаты **x** и **y**, а также номер спрайта (от 1 до 10) из набора, находящегося в буфере. Строки 130 и 140 обеспечивают циклы, в которых происходит печать спрайтов.

Строки 200 и 220 демонстрируют пример простейшей анимации. Спрайт не двигается по экрану, а видоизменяется стоя на месте. Для того, чтобы Вам увидеть этот эффект, поместите в свой буфер в качестве девятого и десятого спрайтов какую-либо пару из приведенных на рис. 1, например "наездника".

Управлять скоростью анимации Вы можете изменяя параметр в операторе PAUSE. Поэкспериментируйте с ним, добиваясь наилучшего впечатления.

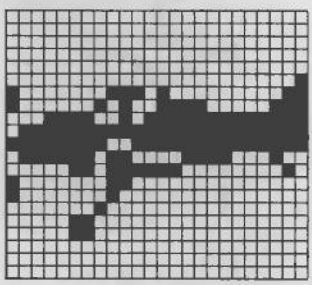


Рис. 5 Моноплан

Данные для ввода спрайта

```
000 000 000 000 000 000 128 129
030 127 255 126 006 128 128 001
006 006 000 000 000 000 000 000
000 000 000 232 079 095 255 063
195 252 192 128 000 000 000 000
000 000 000 000 000 000 000 001
003 007 255 255 255 196 002 000
000 000 000 000 000 000 000
```

Данные для ввода спрайта

```
000 255 255 036 038 035 033 033
033 255 255 001 001 001 003 003
003 002 002 000 000 036 255 255
000 024 126 231 255 255 255 195
126 024 036 066 129 000 000 000
000 000 000 255 255 036 100 196
132 004 004 255 255 128 128 128
192 192 192 064 064 000 000
```

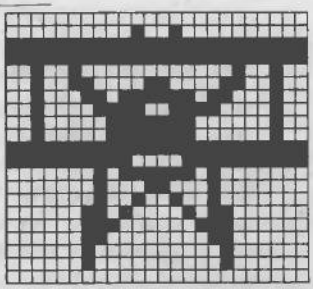


Рис. 6 Биплан-1

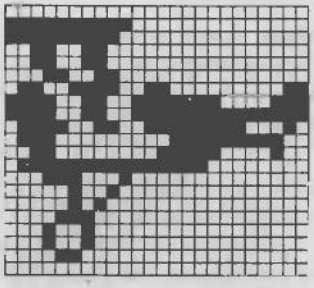


Рис. 7 Биплан-2

Данные для ввода спрайта

```
000 255 255 051 051 025 103 243
243 121 112 048 127 028 004 005
014 018 018 012 000 000 192 128
000 000 128 152 063 063 159 007
015 255 064 128 000 000 000 000
000 000 000 000 000 000 000 000
003 135 255 226 252 132 000 000
000 000 000 000 000 000 000
```

Данные для ввода спрайта

```
000 255 000 000 000 000 004 007
004 000 001 001 003 005 029 128
002 000 002 002 000 000 255 016
016 016 056 056 199 124 254 017
017 017 017 147 254 000 000 000
000 000 000 254 000 000 000 000
064 192 064 000 000 000 128 064
112 112 128 000 128 128 000
```

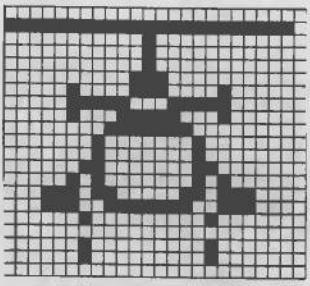


Рис. 8 Вертолет-1

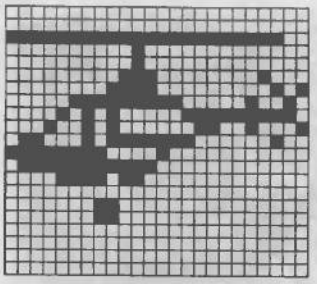


Рис. 9 Вертолет-2

Данные для ввода спрайта

```
000 000 000 255 000 000 000 000
007 010 018 098 255 127 015 000
001 001 000 000 000 000 000 000
255 032 032 112 240 252 131 129
254 008 240 096 000 128 128 000
000 000 000 000 000 252 000 000
008 005 018 254 145 004 000 000
000 000 000 000 000 000 000
```

Данные для ввода спрайта

```
000 000 031 033 127 036 036 036
036 036 036 063 055 060 063 191
255 191 019 019 012 060 024 255
000 255 036 036 036 036 036 036
255 247 060 255 255 255 255 036
036 195 000 000 248 132 254 036
036 036 036 036 036 252 244 060
252 253 255 253 200 200 048
```

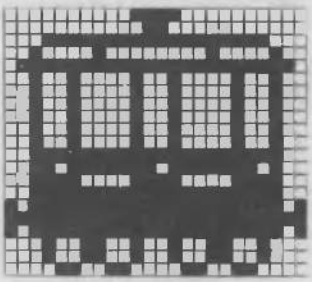


Рис. 10 Вагон

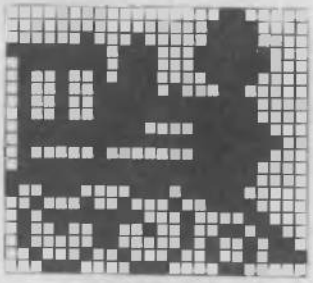


Рис. 11 Паровоз

Данные для ввода спрайта

```
000 000 002 255 127 073 073 073
073 127 127 065 255 255 156 034
095 073 065 034 028 000 000 032
113 113 112 244 255 255 225 255
001 255 255 059 068 226 146 130
068 056 096 240 240 248 248 240
096 240 240 248 252 248 240 240
224 240 024 108 150 151 096
```

Данные для ввода спрайта

```
000 000 062 062 062 062 062 000
120 123 123 123 003 171 255 255
126 063 030 015 007 120 120 000
252 252 252 252 252 000 254 254
254 254 254 254 254 000 255 219
255 255 048 000 120 120 120 000
252 252 252 252 000 085 127 127
127 127 127 255 126 254 252
```

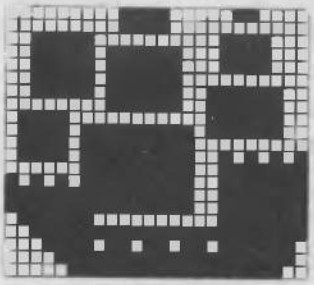


Рис. 12 Военный корабль

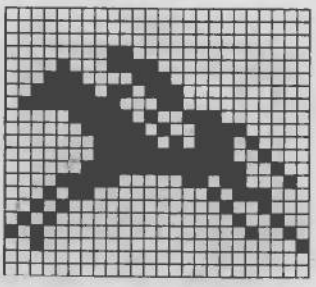


Рис. 13 Наездник-1

Данные для ввода спрайта

```
000 000 000 000 008 028 062 127
007 003 001 001 003 015 022 040
080 160 032 000 000 000 000 000
192 240 056 220 140 211 233 245
255 255 131 000 000 000 000 000
000 000 000 000 000 000 000 000
000 000 128 224 208 200 228 098
184 072 036 018 001 000 000
```

Данные для ввода спрайта

```
000 000 000 000 008 028 062 127
007 003 003 001 000 001 001 000
000 000 000 000 000 000 000 192
240 060 092 200 147 233 245 255
251 241 193 033 146 069 042 004
000 000 000 000 000 000 000 000
000 128 192 224 208 208 136 072
064 128 000 000 000 000 000
```

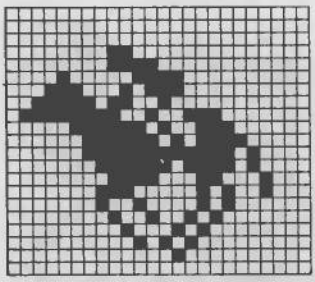


Рис. 14 Наездник-2

Теперь рассмотрим подробно, как работает процедура печати спрайтов на экране. Итак, входными параметрами являются **n**- номер спрайта и **x,y** - координаты левого верхнего угла, выраженные в знакоместах.

Листинг 6
Процедура печати спрайта

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D354  2A0B5C          LD HL,(#5C0B)    ;DEFADD - системная пере-
                                       ;менная, указывающая на
                                       ;то, где находятся пара-
                                       ;метры функции пользова-
                                       ;теля. Ее адрес: #5C0B
                                       ;(23563).
D357  010400          LD   BC,#0004    ;Сдвиг от DEFADD на 4
D35A  09              ADD  HL,BC       ;байта, переход к перво-
                                       ;му параметру.
D35B  56              LD   D,(HL)      ;Координата x.
D35C  010800          LD   BC,#0008    ;Сдвиг еще на 8 байтов ко
D35F  09              ADD  HL,BC       ;второму параметру
D360  5E              LD   E,(HL)      ;Координата y.
D361  ED5394D3        LD   (COORD),DE  ;Запомнили координаты ле-
                                       ;вого верхнего угла спра-
                                       ;йта в программной пере-
                                       ;менной.
D365  09              ADD  HL,BC       ;Сдвиг к 3-му параметру.
D366  7E              LD   A,(HL)      ;Номер спрайта.
D367  3296D3          LD   (NUMB),A    ;Запомнили его в програм-
                                       ;мной переменной.
D36A  7B              LD   A,E         | Алгоритм расчета адреса
D36B  E618            AND  #18         | в экранном файле, когда
D36D  F640            OR   #40         | координаты заданы в
D36F  67              LD   H,A         | знакоместах и выставле-
D370  7B              LD   A,E         | ны в регистровой паре
D371  E607            AND  #07         | DE.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D373   B7             OR   A           Адрес образуется в ре-
D374   1F             RRA              гистровой паре HL.
D375   1F             RRA
D376   1F             RRA
D377   1F             RRA
D378   82             ADD  A,D
D379   6F             LD   L,A
D37A   2292D3         LD   (S_ADR),HL  ;Запомнили экранный адрес
                                        ;в переменной.
D37D   3A96D3         LD   A,(NUMB)    ;Взяли номер спрайта.
D380   47             LD   B,A         ;В регистре B создаем
                                        ;счетчик.
D381   113F00         LD   DE,#003F    ;#3F = 63 - это длина
                                        ;одного шаблона спрайта.
D384   2109D5         LD   HL,SPADR    ;Базовый адрес таблицы
                                        ;спрайтов.
D387   19       LOOP  ADD  HL,DE       ;Смещение на длину одно-
                                        ;го спрайта.
D388   10FD           DJNZ LOOP        ;Возврат к началу цикла.
D38A   ED5B92D3       LD   DE,(S_ADDR);Адрес в экранном файле.
D38E   C301D4         JP   S_PRT       ;Переход на печать
                                        ;спрайта.
D391   C9             RET              ;Возврат.

D392   3B       S_ADR DEFW #48AE       ;Адрес в экранном файле,
                                        ;соответствующий коорди-
                                        ;натам, заданным в знако-
                                        ;местах.
D393   111B     COORD DEFW #0D0E       ;Координаты левого верх-
                                        ;него угла спрайта.
D395            NUMB  DEFB 03          ;Номер спрайта.
```

Процедура B_PRT.

```
Адрес Маш.код Метка   Мнемоника      Комментарий.

D401  0603            LD B,#03       ;Создаем счетчик столбцов.
D403  D5      LOOP-S  PUSH DE        ;Сохранение адреса в эк-
D404  C5              PUSH BC        ;ранном файле и параметра
                                     ;внешнего цикла на стеке.
D405  0E03            LD   C,#03     ;Счетчик рядов.
D407  0608            LD B,#08       ;Счетчик линий в знако-
                                     ;месте
D409  D5      LOOP-R  PUSH DE
D40A  7E      LOOP    LD   A,(HL)    ;Переброска линии
D40B  12              LD   (DE),A    ;из буфера спрайта на
                                     ;экран.
D40C  14              INC  D         ;Приращение старшего бай-
                                     ;та экранного адреса оз-
                                     ;начает, что следующая
                                     ;линия рисуется не рядом
                                     ;с предыдущей, а под
                                     ;ней.
D40D  23              INC  HL        ;Следующая линия.
D40E  10FA            DJNZ LOOP      ;Конец цикла для 8-ми
                                     ;линий одного знакоместа.
D410  D1              POP  DE        ;Восстановление адреса
                                     ;левого верхнего угла.
D411  3E20            LD   A,#20     ;Переход к соседнему зна-
D413  83              ADD  A,E       ;кому месту снизу.
D414  5F              LD   E,A       ;
D415  79              LD   A,C       ;Уменьшение счетчика зна-
D416  3D              DEC  A         ;комест по вертикали.
D417  FE01            CP   #01       ;Если это не третий
D419  2005            JR   NZ,NEXT   ;ряд, то переход.
D41B  0605            LD   B,#05     ;Если же он - третий,
                                     ;то в нем надо печатать
                                     ;только пять линий.
D41D  4F              LD   C,A       ;Запомнили счетчик.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D41E  18E9            JR    LOOP-R       ;Возврат для печати зна-
                                         ;коместа третьего ряда.
D420  0608    NEXT    LD    B,#08        ;Счетчик восьми линий.
D422  4F              LD    C,A          ;Запомнили счетчик.
D423  FE00            CP    #00          ;Все ряды напечатаны?
D425  20E2            JR    NZ,LOOP-R    ;Если нет, то возврат.
D427  C1              POP   BC           ;Все ряды в столбце на-
D428  D1              POP   DE           ;печатаны. Надо восста-
                                         ;новить счетчики и адрес
                                         ;в экранном файле, прежде
                                         ;чем перейти к очередному
                                         ;столбцу.
D429  13              INC   DE           ;Приращенне младшего бай-
                                         ;та адреса дисплейного
                                         ;файла означает, что оче-
                                         ;редная линия рисуется
                                         ;справа от предыдущей.
D42A  10D7            DJNZ  LOOP-S       ;Если счетчик столбцов
                                         ;еще не обнулился, следу-
                                         ;ет возврат к началу вне-
                                         ;шнего цикла.
D42C  C9              RET                ;Возврат.
```

2.1.4. Перемещение спрайтов.

Теперь, когда мы научились создавать спрайты и печатать их на экране в нужной нам позиции, настало время подумать о самом интересном моменте - о перемещении спрайтов. Для того, чтобы начать эксперименты с анимацией спрайтов, нам потребуется специальная процедура, которую мы будем использовать еще не раз. Назовем эту процедуру драйвером спрайта **SPRDRV** и рассмотрим здесь поподробнее.

В качестве первого приближения поставим задачу простейшего

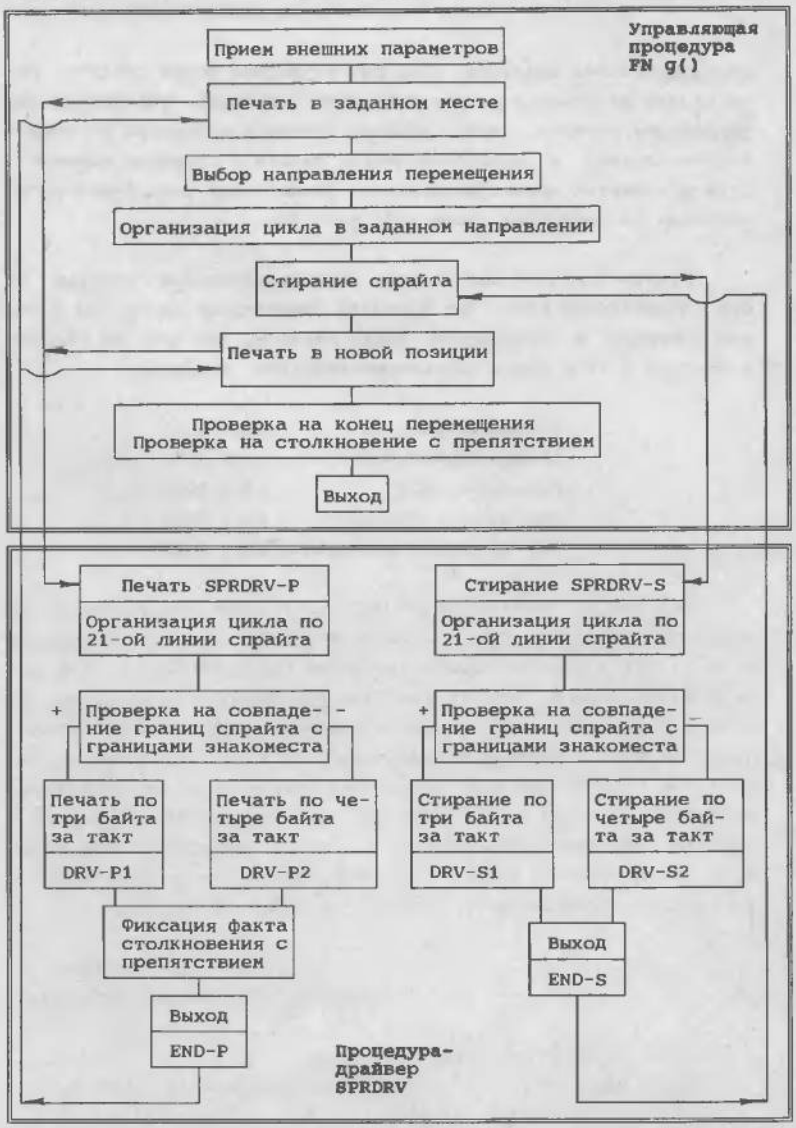


Рис. 15 Блок-схема процедуры простого перемещения спрайта.

перемещения без анимации, т.е. без изменения формы спрайта. Наша задача разобьется на две подзадачи. Первая - управление перемещением спрайта, этим займется головная процедура и вторая задача состоит в непосредственно печати и стирании спрайта - этим и займется процедура-драйвер. Связь между ними будет организована по следующей схеме (см. рис. 15).

Печать/стирание выполняются последовательной печатью по XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ). Это наиболее эффективный прием. Мы о нем уже говорили в предыдущих томах сериала, но если Вы желаете убедиться в этом сами, поэкспериментируйте, например:

```
Исходный байт:              1100 1010
Накладываемый байт:         1111 0000
Результат (XOR)    :        0011 1010
Повторное наложение:        1111 0000
Окончательный результат:    1100 1010
```

Как видите, окончательный результат ничем не отличается от исходного байта. В этом и состоит основной смысл использования функции XOR. То есть первое наложение байта на байт по XOR эквивалентно печати спрайта на заранее подготовленном фоне, а второе наложение в то же место обеспечивает стирание и восстановление фона. Во втором томе нашего сериала (*"Прикладная графика"*, с. 46-51) мы даже вывели ряд правил, которых необходимо придерживаться для того, чтобы при такой печати по XOR свести к минимуму проблемы, связанные с "клэшингом" атрибутов. Вам также может быть полезна работа *С.*Суркова "Вывод спрайтов на экран с маской", опубликованная в сборнике *ZX-FORUM* (с.126-131).

Листинг 7
Загрузчик управляющей процедуры.

```
10 REM *** Загрузчик машинного кода
20 LET  adr=53500: LET long=170: LET check=70909: LET sum=0
30 FOR k=0 TO long-1: READ a
40 POKE (adr+k),a: LET sum=sum+a
```

```
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 042,011,092,017,004
120 DATA 000,025,078,030,008
130 DATA 025,070,025,126,230
140 DATA 003,050,158,209,025
150 DATA 126,050,159,209,025
160 DATA 126,230,001,050,160
170 DATA 209,025 126,230,001
180 DATA 050,161,209,025,126
190 DATA 017,063,000,033,009
200 DATA 213,025,061,032,252

210 DATA 205,093,210,254,000
220 DATA 040,019,058,161,209
230 DATA 254,000,040,012,058
240 DATA 160,209,198,001,230
250 DATA 001,050,160,209,024
260 DATA 071,118,205,196,209
270 DATA 058,158,209,254,000
280 DATA 040,019,254,001,040
290 DATA 026,254,002,040,033
300 DATA 005,005,005,120,230

310 DATA 252,040,044,195,132
320 DATA 209,013,013,013,121
330 DATA 230,252,040,033,195
340 DATA 132,209,012,012,012
350 DATA 121,214,231,048,022
360 DATA 195,132,209,004,004
370 DATA 004,120,214,150,048
380 DATA 011,058,159,209,061
390 DATA 050,159,209,254,000
400 DATA 032,159,058,160,209
```

```
410 DATA 237,067,250,208,254
420 DATA 000,200,118,205,093
430 DATA 210,201,001,063,000
440 DATA 001,000,000,000,000
```

Формат управляющей процедуры.

DEF FNg (x,y,d,l,s,c,n)

x,y - координаты исходной позиции спрайта. Они относятся к левому верхнему углу и задаются в пикселах.

d - direction - параметр, определяющий направление перемещения.
d=0 - перемещение справа налево.
d=1 - перемещение слева направо.
d=2 - перемещение снизу вверх.
d=1 - перемещение сверху вниз.

l - параметр, определяющий дистанцию, на которую перемещается спрайт. За единицу принято перемещение на три пиксела. Таким образом, если установить здесь например число 20, то спрайт продвинется на 60 пикселов и остановится.

s - switch - параметр, определяющий остается спрайт на экране после окончания перемещения или нет. Если s=0, то он останется на экране. Если s=1, то спрайт исчезнет.

c - collision - флаг детекции коллизии. Если его включить (c=1), то при столкновении с любым препятствием (со включенными пикселами экрана) спрайт остановится после того, как три пиксела спрайта "наедут" на препятствие. Если его отключить (c=0), то спрайт будет проходить через любое препятствие.

n - номер спрайта. Если у Вас в таблице спрайтов задано не-

сколько шаблонов, то здесь достаточно указать на нужный Вам номер спрайта.

Листинг 8
Процедура, управляющая перемещением спрайта.

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D0FA  0000    YXLAST  DEFW #0000       ;Когда спрайт закончит
                                       ;перемещение, в эту пере-
                                       ;менную запишутся послед-
                                       ;ние координаты y и x.
                                       ;По ним можно определить,
                                       ;например, где произошла
                                       ;коллизия с препятствием.
D0FC  2A0B5C          LD HL,(#5C0B)    ;  Точка входа.
                                       ;DEFADD - системная пере-
                                       ;менная, указывающая на
                                       ;то, где находятся пара-
                                       ;метры функции пользова-
                                       ;теля. Ее адрес: #5C0B
                                       ;(23563).
D0FF  110400          LD   DE,#0004    ;Сдвиг от DEFADD на 4
D102  19              ADD  HL,DE       ;байта, переход к перво-
                                       ;му параметру.
D103  4E              LD   C,(HL)      ;Координата x левого вер-
                                       ;хнего угла спрайта.
D104  1E08            LD   E,#08       ;Сдвиг еще на 8 байтов ко
D106  19              ADD  HL,DE       ;второму параметру.
D107  46              LD   B,(HL)      ;Координата y.
D108  19              ADD  HL,DE       ;Переход к 3-му пар-ру.
D109  7E              LD   A,(HL)      ;Направление движения
                                       ;спрайта - d.
D10A  E603            AND  #03         ;Маскирование шести стар-
                                       ;ших битов (за ненадобно-
                                       ;стью).
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D10C  329ED1          LD    (DIREC),A  ;Запомнили направление в
                                       ;программной переменной.
D10F  19              ADD   HL,DE      ;Переход к 4-му пар-ру.
D110  7E              LD    A,(HL)     ;Дистанция перемещения.
D111  329FD1          LD    (DIST),A   ;Запомнили в переменной.
D114  19              ADD   HL,DE      ;Переход к 5-му пар-ру.
D115  7E              LD    A,(HL)     ;Переключатель в.
D116  E601            AND   #01        ;Семь старших битов не
                                       ;нужны.
D118  32A0D1          LD    (SWTCH),A  ;Запомнили в переменной.
D11B  19              ADD   HL,DE      ;Переход к 6-му пар-ру.
D11C  7E              LD    A,(HL)     ;Флаг детекции коллизии.
D11D  E601            AND   #01        ;Маскирование старших
                                       ;битов.
D11F  32A1D1          LD    (FLAG),A   ;Сохранили в переменной.
D122  19              ADD   HL,DE      ;Переход к 7-му пар-ру.
D123  7E              LD    A,(HL)     ;Номер спрайта (n).
D124  113F00          LD    DE,#003F   ; Определение начала
D127  2109D5          LD    HL,#D509   ; шаблона спрайта в таб-
D12A  19        LOOP  ADD   HL,DE      ; лице по его номеру (см.
D12B  3D              DEC   A          ; комментарий на с.26).
D12C  20FC            JR    NZ,LOOP    ;
D12E  CD5DD2    BEGIN CALL  SPRDRV-P   ;Вызов драйвера для печа-
                                       ;ти спрайта.
D131  FE00            CP    #00        ;По окончании печати
                                       ;драйвер выдает в аккуму-
                                       ;ляторе единицу, если
                                       ;произошло наложение пик-
                                       ;селов спрайта на вклю-
                                       ;ченные пикселы фонового
                                       ;экрана Это и есть колли-
                                       ;зии с препятствием.
D133  2813            JR    Z,BYPASS   ;Если ноль, то обход.
D135  3AA1D1          LD    A,(FLAG)   ;Проверяется флаг детек-
D138  FE00            CP    #00        ;ции коллизии.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D13A  280C            JR Z,BYPASS        ;Если нет детекции колли-
                                         ;зии, то обход.
D13C  3AA0D1          LD   A,(SWTCH)     | Переключение пере-
D13F  C601            ADD  A,#01         | ключателя SWTCH в
D141  E601            AND  #01           | противоположное
D143  32A0D1          LD   (SWTCH),A     | положение.
D146  1847            JR   END           ;Завершение работы по
                                         ;причине коллизии с пре-
                                         ;пятствием.
D148  76      BYPASS  HALT               ;Краткая пауза.
D149  CDC4D1          CALL SPRDRV-S      ;Вызов процедуры стирания
                                         ;спрайта.
D14C  3A9ED1          LD   A,(DIREC)     ;Направление перемещения
D14F  FE00            CP   #00           ;влево?
D151  2813            JR   Z,LEFT        ;Если так, переход.
D153  FE01            CP   #01           ;Направление вправо?
D155  281A            JR   Z,RIGHT       ;Если так, переход.
D157  FE02            CP   #02           ;Направление вверх?
D159  2821            JR   Z,UP          ;Если так, переход.
D15B  05              DEC  B             ;Остается только направ-
D15C  05              DEC  B             ;ление вниз. Уменьшаем
D15D  05              DEC  B             ;координату y в регистре
                                         ;"B" на три единицы.
D15E  78              LD   A,B           ;Проверка на выход за
D15F  E6FC            AND  #FC           ;пределы нижней границы
                                         ;экрана. #FC=1111 1100.
                                         ;Команда AND FC может
                                         ;дать ноль только если
                                         ;число в аккумуляторе
                                         ;от 0 до 3.
D161  282C            JR   Z,END         ;Конец работы.
D163  C384D1          JP   CONT          ;Иначе продолжаем.
D166  0D      LEFT    DEC  C             ;Уменьшение
D167  0D              DEC  C             ;координаты x.
D168  0D              DEC  C  .          ;
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D169  79              LD    A,C          ;Проверка на выход за
D16A  E6FC            AND   #FC          ;левую границу экрана.
D16C  2821            JR    Z,END        ;Если так, то конец.
D16E  C384D1          JP    CONT         ;Иначе продолжаем.
D171  0C      RIGHT   INC   C            ;Увеличение
D172  0C              INC   C            ;координаты x.
D173  0C              INC   C            ;
D174  79              LD    A,C          ;Проверка на выход за
D175  D6E7            SUB   #E7          ;правый край экрана.
D177  3016            JR    NC,END       ;Если так, то конец.
D179  C384D1          JP    CONT         ;Иначе продолжаем.
D17C  04      UP      INC   B            ;Увеличение координаты y.
D17D  04              INC   B            ;
D17E  04              INC   B            ;
D17F  78              LD    A,B          ;Проверка на выход за
D180  D696            SUB   #96          ;верхний край экрана.
D182  300B            JR    NC,END       ;Если так, то конец.
D184  3A9FD1  CONT    LD    A,(DIST)     ;Уменьшение дистанции
D187  3D              DEC   A            ;движения
D188  329FD1          LD    (DIST),A     ;на 1 пункт.
D18B  FE00            CP    #00          ;Проверка на конец
                                         ;перемещения.
D18D  209F            JR    NZ,BEGIN     ;Если не конец, то воз-
                                         ;врат к началу движения.
D18F  3AA0D1  END     LD    A,(SWTCH)    ;Иначе загрузка SWTCH.
D192  ED43FAD0        LD    (YXLAST),BC:После окончания переме-
                                         ;щения спрайта его пос-
                                         ;ледние координаты за-
                                         ;поминаются в этой ячейке
                                         ;памяти. Это полезно, ес-
                                         ;ли спрайт с чем-то стол-
                                         ;кнулся. Можно узнать,
                                         ;где произошло столкнове-
                                         ;ние и использовать это в
                                         ;программе.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D196  FE00            CP   #00         ;Проверка SWTCH.
D198  C8              RET  Z           ;Если 0, то спрайт рисо-
                                       ;вать больше не надо и
                                       ;возврат.
D199  76              HALT             ;Краткая пауза.
D19A  CD5DD2          CALL SPRDRV-P    ;Последнее рисование
                                       ;спрайта.
D19D  C9              RET              ;Выход.
D19E  01        DIREC DEFB 01          ;Направление движения.
D19F  3F        DIST  DEFB 3F          ;Дистанция перемещения.
D1A0  00        SWTCH DEFB 00          ;Переключатель ■.
D1A1  01        FLAG  DEFB 01          ;Флаг детекции коллизии.
D1A2  0000            DEFW 0000
D1A4  0000            DEFW 0000
```

Листинг 9

Загрузчик драйвера спрайтов **SPRDRV**

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=53700: LET long=365: LET check=94117: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 229,213,197,245,229
120 DATA 221,225,120,205,177
130 DATA 034,040,082,237,067
140 DATA 015,210,050,229,209
150 DATA 062,021,008,221,094
160 DATA 000,221,086,021,221
170 DATA 078,042,006,002,175
180 DATA 203,027,203,026,203
190 DATA 025,031,016,247,035
```

```
200 DATA 035,035,174,119,043
210 DATA 126,169,119,043,126
220 DATA 170,119,043,126,171
230 DATA 119,008,061,040,025
240 DATA 008,221,035,036,124
250 DATA 230,007,032,205,001
260 DATA 202,140,008,095,008
270 DATA 062,021,147,128,071
280 DATA 205,177,034,024,189
290 DATA 241,193,209,225,201
300 DATA 237,067,081,210,062

310 DATA 021,008,221,094,000
320 DATA 221,086,021,221,078
330 DATA 042,035,035,126,169
340 DATA 119,043,126,170,119
350 DATA 043,126,171,119,008
360 DATA 061,040,218,008,221
370 DATA 035,036,124,230,007
380 DATA 032,221,008,095,008
390 DATA 001,184,140,062,021
400 DATA 147,128,071,205,177

410 DATA 034,024,205,229,213
420 DATA 197,229,221,225,151
430 DATA 050,041,211,120,205
440 DATA 177,034,202,216,210
450 DATA 237,067,194,210,050
460 DATA 130,210,062,021,008
470 DATA 221,094,000,221,086
480 DATA 021,221,078,042,006
490 DATA 005,175,203,027,203
500 DATA 026,203,025,031,016

510 DATA 247,035,035,035,071
520 DATA 126,160,196,033,211
530 DATA 126,168,119,043,126
```

```
540 DATA 161,196,033,211,126
550 DATA 169,119,043,126,162
560 DATA 196,033,211,126,170
570 DATA 119,043,126,163,196
580 DATA 033,211,126,171,119
590 DATA 008,061,040,025,008
600 DATA 221,035,036,124,230

610 DATA 007,032,183,001,205
620 DATA 140,008,095,008,062
630 DATA 021,147,128,071,205
640 DATA 177,034,024,167,193
650 DATA 209,225,058,041,211
660 DATA 201,237,067,021,211
670 DATA 062,021,008,221,094
680 DATA 000,221,086,021,221
690 DATA 078,042,035,035,126
700 DATA 161,196,033.211,126

710 DATA 169,119,043,126,162
720 DATA 196,033,211,126,170
730 DATA 119,043,126,163,196
740 DATA 033,211,126,171,119
750 DATA 008,061,040,201,008
760 DATA 221,035,036,124,230
770 DATA 007,032,206,008,095
780 DATA 008,001,184,140,062
790 DATA 021,147,128,071,205
800 DATA 177,034,024,190,245

810 DATA 062,001,050,041,211
820 DATA 241,201,001,000,000
830 DATA 083,000,000,000,000
```

Листинг 10
Драйвер спрайтов **SPRDRV**

Примечание: во время работы процедура в некоторых случаях модифицирует саму себя. Ячейки памяти, изменяющиеся динамически по ходу работы, отмечены в листинге символами *****.

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

D1C4  E5      SPRDRV-S PUSH HL            ;Это точка входа для
                                          ;стирания спрайта.
                                          ;Сохранение всех
D1C5  D5               PUSH DE            ;основных регистровых
D1C6  C5               PUSH BC            ;пар на машинном
D1C7  F5               PUSH AF            ;стеке.
D1C8  E5               PUSH HL            ;Переброска адреса из
D1C9  DDE1             POP  IX            ;HL в IX через стек.
D1CB  78               LD   A,B           ;Запомнили регистр B,
                                          ;т.к. вызов #22B1 его
                                          ;портит.
D1CC  CDB122           CALL #22B1
```

;Это вызов процедуры ПЗУ PIXEL-ADD, которая расположена ;начиная с адреса #22AA. Процедура служит для того, чтобы по ко-;ординатам пиксела, заданным в паре BC (**x** - в C, а **y** - в B) най-;ти адрес, соответствующий этому пикселу в дисплейном файле. ;Этот адрес выдается процедурой в регистровую пару HL. Поскольку ;по этому адресу находится байт, позиция включенного бита кото-;рого также важна, эта процедура выдает в регистре A позицию ;пиксела внутри байта.

;Обычно в "высокой" графике принято отсчитывать коорди-;наты X и Y, начиная с левого нижнего угла экрана. Здесь же этот ;порядок изменен. Все процедуры, которые в своей работе обраща-;ются к процедуре MAIN, задают координаты положения спрайта в ;системе отсчета, в которой за ноль приняты координаты левого ;верхнего угла. Именно поэтому вызов процедуры PIXELL-ADD проис-

```
;ходит не по обычной точке входа #22AA, а через точку #22B1.

Адрес Маш.код Метка   Мнемоника        Комментарий.

D1CF  2852            JR    Z,FIT-S    ;Переход, если пиксел на-
                                       ;ходится в  нулевой пози-
                                       ;ции, т.е. левая граница
                                       ;спрайта совпадает с гра-
                                       ;ницей знакоместа.
D1D1  ED430FD2        LD    (#D20F),BC ;Модифицируем команду
                                       ;#D20E координатами y,x.
D1D5  32E5D1          LD    (#D1E5),A  ;Настраиваем счетчик сме-
                                       ;щений битов. См. команду
                                       ;#D1E4.
D1D8  3E15            LD    A,#15      ;#15=21 - число строк в
                                       ;спрайте.
D1DA  08              EX    AF,AF'     ;Сохранение регистра A.
D1DB  DD5E00  BEGIN1  LD    E,(IX+0)   ;Адреса трех байтов, об-
D1DE  DD5615          LD    D,(IX+21)  ;разующих строку,  (см.
D1E1  DD4E2A          LD    C,(IX+42)  ;рис. 7).
D1E4  0602    *****   LD    B,#02      ;Инициализация счетчика
                                       ;ротаций (см. #D1D5)
D1E6  AF              XOR   A          ;Обнуление аккумулятора.
D1E7  CB1B    LOOP-1  RR    E          ;Ротация вправо (пере-
D1E9  CB1A            RR    D          ;мещение пиксела вправо)
D1EB  CB19            RR    C          ;регистров C,D,E,A. Вы-
D1ED  1F              RRA              ;полняется столько раз,
D1EE  10F7            DJNZ  LOOP-1     ;сколько выставлено в B.
D1F0  23              INC   HL         ;Смещение
D1F1  23              INC   HL         ;на четвертое
D1F2  23              INC   HL         ;знакоиесто вправо.
D1F3  AE              XOR   (HL)       ;Стирание линии в
                                       ;крайнем правом знако-
                                       ;месте повторной печатью
D1F4  77              LD    (HL),A     ;по XOR.
D1F5  2B              DEC   HL         ;Переход ко 2-му
                                       ;знакоместу справа.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D1F6  7E              LD   A,(HL)      ; Стирание линии во
D1F7  A9              XOR  C           ; 2-м справа
D1F8  77              LD   (HL),A      ; знакоместе.
D1F9  2B              DEC  HL          ;Переход ко 2-му
                                       ;знакоместу слева.
D1FA  7E              LD   A,(HL)      ; Стирание линии во
D1FB  AA              XOR  D           ; 2-м слева
D1FC  77              LD   (HL),A      ; знакоместе.
D1FD  2B              DEC  HL          ;Переход к крайнему
                                       ;левому знакоместу.
D1FE  7E              LD   A,(HL)      ; Стирание линии в
D1FF  AB              XOR  E           ; крайнем левом
D200  77              LD   (HL),A      ; знакоместе.
D201  08              EX   AF,AF'      ;Восстановили счетчик
                                       ;линий.
D202  3D              DEC  A           ;Уменьшили его.
D203  2819            JR   Z,END-S     ;Если весь спрайт
                                       ;стерт, то выход.
D205  08              EX   AF,AF'      ;Запомнили счетчик
                                       ;линий.
D206  DD23            INC  IX          ;Переход к очередному
                                       ;байту в шаблоне
                                       ;спрайта.
D208  24              INC  H           ;Переход к соседней сни-
                                       ;зу линии на экране.
D209  7C              LD   A,H         ;Если очередной экран-
D20A  E607            AND  #07         ;ный ряд не закончен,
D20C  20CD            JR   NZ,BEGIN1   ;возврат к началу цикла.
D20E  01CA8C  *****   LD BC,#8CCA      ;Взяли счетчик линий.
D211  08              EX   AF,AF'      ;
D212  5F              LD   E,A         ;Расчет по формуле:
D213  08              EX   AF,AF'      ;B=y+21-E, где E -
D214  3E15            LD   A,#15       ;количество ненапеча-
D216  93              SUB  E           ;танных строк спрайта.
D217  80              ADD  A,B         ;
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D218  47              LD   B,A
D219  CDB122          CALL #22B1        ;Вызов PIXEL-ADD (см.
                                        ;выше).
D21C  18BD            JR   BEGIN1       ;Возврат к началу цикла.
D21E  F1      END-S   POP  AF           Финишные операции,
D21F  C1              POP  BC           связанные с балан-
D220  D1              POP  DE           сировкой стека.
D221  E1              POP  HL
D222  C9              RET               ; Выход из процедуры.

D223  ED4351D2 FIT-S  LD   (#D251),BC ;Ввели координату в
                                        ;команду #D251.
D227  3E15            LD   A,#15        ;21 строка в спрайте.
D229  08              EX   AF,AF'       ;Времению сохранили.
                                        ;Это начало цикла по
                                        ;строкам.
D22A  DD5E00  BEGIN2  LD   E,(IX+0)     ;Три байта, образующие
D22D  DD5615          LD   D,(IX+21)    ;одну линию в спрайте.
D230  DD4E2A          LD   C,(IX+42)    ;
D233  23              INC  HL           ;Расчет адреса
D234  23              INC  HL           ;правого знакоместа.
D235  7E              LD   A,(HL)       ;Сняли с экрана.
D236  A9              XOR  C            ;Наложили спрайт по
                                        ;XOR.
D237  77              LD   (HL),A       ;Поместили результат на
                                        ;экран.
D238  2B              DEC  HL           ;Соседнее знакоместо
                                        ;слева.
D239  7E              LD   A,(HL)       ;Стирание линии в
D23A  AA              XOR  D            ;этом знакоместе.
D23B  77              LD   (HL),A       ;
D23C  2B              DEC  HL           ;Крайнее левое
                                        ;знакоместо.
D23D  7E              LD   A,(HL)       ;Стирание линии в
D23E  AB              XOR  E            ;крайнем левом
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

D23F  77              LD   (HL),A     ;знакоместе.
D240  08              EX   AF,AF'      ;Восстановили
                                       ;счетчик линий.
D241  3D              DEC  A           ;Уменьшили его.
D242  28DA            JR   Z,END-8     ;Выход, если все
                                       ;линии удалены.
D244  08              EX   AF,AF'      ;Сохранили счетчик.
D245  DD23            INC  IX          ;Переход к следующему
                                       ;байту в шаблоне
                                       ;спрайта.
D247  24              INC  H           ;Переход к соседней
                                       ;(снизу) линии экрана.
D248  7C              LD   A,H         ;Если очередной экран-
D249  E607            AND  #07         ;ный ряд не закончен,
D24B  20DD            JR   NZ,BEGIN2   ;возврат к началу цикла.
D24D  08              EX   AF,AF'      | Запомнили счетчик
D24E  5F              LD   E,A         | линий.
D24F  08              EX   AF,AF'      |
D250  01B88C  *****   LD   BC,#8CB8   ;Ввод текущей координ-
                                       ;наты y (см. также
                                       ;команду #D223).
D253  3E15            LD   A,#15       ;Расчет по формуле:
D255  93              SUB  E           ;B=y+21-E, где E -
D256  80              ADD  A,B         ;количество ненапечатан-
D257  47              LD   B,A         ;ных строк спрайта.
D258  CDB122          CALL #22B1       ;Вызов PIXEL-ADD
D25B  18CD            JR   BEGIN2      ;Возврат к началу

D25D  E5    SPRDRV-P  PUSH HL          ;Точка входа для печати.
                                       ;Сохранение всех основных
D25E  D5              PUSH DE          ;регистровых пар на ма-
D25F  C5              PUSH BC          ;шинном стеке.
D260  E5              PUSH HL          ;Переброска адреса из
D261  DDE1            POP  IX          ;HL в IX через стек.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D263  97              SUB  A           ;Обнуленне аккумулятора.
D264  3229D3          LD   (COLLIZ),A  ;Инициализация COLLIZ.
D267  78              LD   A,B         ;Сохраняем B (см. выше).
D268  CDB122          CALL #22B1       ;Вызов процедуры ПЗУ PI-
                                       ;XELL-ADD. Определяем ад-
                                       ;рес в дисплейном файле,
                                       ;соответствующий x и y.
                                       ;Подробнее см. выше.
D26B  CAD8D2          JP   Z,FIT-P     ;Переход, если пиксел на-
                                       ;ходится в  нулевой пози-
                                       ;ции знакоместа.
D26E  ED43C2D2        LD   (#D2C2),BC  ;Модифицируем команду
                                       ;#D2C1 координатами y,x.
D272  3282D2          LD   (#D282),A   ;Модифицируем команду
                                       ;#D281 номером позиции
                                       ;пиксела, к которому от-
                                       ;носится координата.
D275  3E15            LD   A,#15       ;Количество линий в
                                       ;спрайте = 21.
D277  08              EX   AF,AF'      ;Оставили A в качестве
                                       ;счетчика линий.
D278  DD5E00  BEGIN3  LD   E,(IX+0)    ;Три линии из трех
D27B  DD5615          LD   D,(IX+21)   ;соседних по горизонтали
D27E  DD4E2A          LD   C,(IX+42)   ;знакомест.
D281  0605    *****   LD   B,#05       ;Здесь стоит номер пози-
                                       ;ции включенного пиксела
                                       ;в байте (см. коианду по
                                       ;адресу #D272).
D283  AF      LOOP-3  XOR  A           ;Обнуление аккумулятора.
D284  CB1B            RR   E           ;Ротация вправо.
D286  CB1A            RR   D           ;
D288  CB19            RR   C           ;
D28A  1F              RRA              ;
D28B  10F7            DJNZ LOOP-3      ;Конец цикла ротации.
D28D  23              INC  HL          ;Смещение
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D28E  23              INC  HL            ;на четвертое
D28F  23              INC  HL            ;знакоместо вправо.
D290  47              LD   B,A           | Печать линии в
D291  7E              LD   A,(HL)        | крайнем правои
D292  A0              AND  B             | знакоместе.
D293  C421D3          CALL NZ,COL-ON     ; При наличии колли-
                                         ;зии включили перемен-
                                         ;ную COLLIZ.
D296  7E              LD   A,(HL)        |
D297  A8              XOR  B             |
D298  77              LD   (HL),A        |
D299  2B              DEC  HL            ;Переход ко 2-му
                                         ;знакоместу справа.
D29A  7E              LD   A,(HL)        | Печать линии во
D29B  A1              AND  C             | 2-м справа
D29C  C421D3          CALL NZ,COL-ON     | знакоместе.
D29F  7E              LD   A,(HL)        |
D2A0  A9              XOR  C             |
D2A1  77              LD   (HL),A        |
D2A2  2B              DEC  HL            ;Переход ко 2-му
                                         ;знакоместу слева.
D2A3  7E              LD   A,(HL)        | Печать линии во
D2A4  A2              AND  D             | 2-м слева
D2A5  C421D3          CALL NZ,COL-ON     | знакоместе.
D2A8  7E              LD   A,(HL)        |
D2A9  AA              XOR  D             |
D2AA  77              LD   (HL),A        |
D2AB  2B              DEC  HL            ;Переход к крайнему
                                         ;левому знакоместу.
D2AC  7E              LD   A,(HL)        | Печать линии в
D2AD  A3              AND  E             | крайнеи левом
D2AE  C421D3          CALL NZ,COL-ON     | знакоместе.
D2B1  7E              LD   A,(HL)        |
D2B2  AB              XOR  E             |
D2B3  77              LD   (HL),A        |
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

D2B4  08              EX   AF,AF'      ;Взяли счетчик строк
                                       ;спрайта.
D2B5  3D              DEC  A           ;Уменьшили его на 1.
D2B6  2819            JR   Z,END-P     ;Если все строки спрайта
                                       ;напечатаны, то конец.
D2B8  08              EX   AF,AF'      ;Запомнили результат.
D2B9  DD23            INC  IX          ;Переход к следующему
                                       ;байту в шаблоне спрайта
D2BB  24              INC  H           ;Переход к соседней сни-
                                       ;зу линии на экране.
D2BC  7C              LD   A,H         ;Если это ненулевая ли-
D2BD  E607            AND  #07         ;ния в знакоместе, то
D2BF  20B7            JR   NZ,BEGIN3   ;возврат в начало цикла.
D2C1  01CD8C  *****   LD   BC,#8CCD    ;Ввод координат y,x, см.
                                       ;также команду #D26E.
D2C4  08              EX   AF,AF'      ;Взяли счетчик линий.
D2C5  5F              LD   E,A         ||Расчет по формуле:
D2C6  08              EX   AF,AF'      ||B=y+21-E, где E -
D2C7  3E15            LD   A,#15       ||количество ненапеча-
D2C9  93              SUB  E           ||танных строк спрайта.
D2CA  80              ADD  A,B         ||
D2CB  47              LD   B,A         ||
D2CC  CDB122          CALL #22B1       ;Вызов PIXEL-ADD (см.
                                       ;выше).
D2CF  18A7            JR   BEGIN3      ;Возврат к началу.
D2D1  C1      END-P   POP  BC          ||Финишные операции,
D2D2  D1              POP  DE          ||связанные с балан-
D2D3  E1              POP  HL          ||сировкой стека.
D2D4  3A29D3          LD   A,(COLLIZ)  ||
D2D7  C9              RET              ||

D2D8  ED4315D3 FIT-P  LD   (D315),BC   ;Ввели координату в
                                       ;команду #D314.
D2DC  3E15            LD   A,#15       ;21 строка в спрайте.
```

```
Адрес Маш.код Метка   Мнемоника        Комиентарий.

D2DE  08              EX   AF,AF'      ;Временно сохранили.
                                       ;Это начало цикла по
                                       ;строкам.
D2DF  DD5E00  BEGIN4  LD   E,(IX+0)    ;Три байта, образующие
D2E2  DD5615          LD   D,(IX+21)   ;одну линию в спрайте.
D2E5  DD4E2A          LD   C,(IX+42)   ;
D2E8  23              INC  HL          ;Расчет адреса
D2E9  23              INC  HL          ;правого знакоместа.
D2EA  7E              LD   A,(HL)      ;Сняли с экрана.
D2EB  A1              AND  C           ;Сравнили с тем, что
                                       ;есть в спрайте.
D2EC  C421D3          CALL NZ,COL-ON   ;Включили COLLIZ.
D2EF  7E              LD   A,(HL)      ;Сняли линию с экрана.
D2F0  A9              XOR  C           ;Наложили на нее по XOR
                                       ;линию из спрайта.
D2F1  77              LD   (HL),A      ;Поместили рез-т на
                                       ;экран.
D2F2  2B              DEC  HL          ;Соседнее знакоместо
                                       ;слева.
D2F3  7E              LD   A,(HL)      ||
D2F4  A2              AND  D           ||  С ним все то же
D2F5  C421D3          CALL NZ,COL-ON   ||  самое.
D2F8  7E              LD   A,(HL)      ||
D2F9  AA              XOR  D           ||
D2FA  77              LD   (HL),A      ||
D2FB  2B              DEC  HL          ;Самое левое знакоместо.
D2FC  7E              LD   A,(HL)      ||
D2FD  A3              AND  E           ||  С ним все то же
D2FE  C421D3          CALL NZ,COL-ON   ||  самое.
D301  7E              LD   A,(HL)      ||
D302  AB              XOR  E           ||
D303  77              LD   (HL),A      ||
D304  08              EX   AF,AF'      ;Восстановили счетчик
                                       ;линий в спрайте.
D305  3D              DEC  A           ;Уменьшили счетчик линий.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D306  28C9            JR   Z,END-P       ;Если конец, то на выход
D308  08              EX   AF,AF'        ;Запомнили счетчик линий.
D309  DD23            INC  IX            ;Переход к следующему
                                         ;байту в шаблоне спрайта
D30B  24              INC  H             ;Переход к соседней сни-
                                         ;зу линии на экране.
D30C  7C              LD   A,H           ;Если это ненулевая ли-
D30D  E607            AND  #07           ;ния в знакоместе, то
D30F  20CE            JR   NZ,BEGIN4     ;возврат в начало цикла.
D311  08              EX   AF,AF'        ;Взяли счетчик линий.
D312  5F              LD   E,A           ;Поместили его в E.
D313  08              EX   AF,AF'        ;Сохранили счетчик линий
D314  01B88C  *****   LD   BC,#8CB8      ;Ввод координат y,x, см.
                                         ;также команду #D2D8.
D317  3E15            LD   A,#15         ;Расчет по формуле:
D319  93              SUB  E             ;B=y+21-E, где E -
D31A  80              ADD  A,B           ;количество ненапечатан-
D31B  47              LD   B,A           ;ных строк спрайта.
D31C  CDB122          CALL #22B1         ;Вызов PIXEL-ADD (см.
                                         ;выше).
D31F  18BE            JR   BEGIN4        ;Возврат к началу.

D321  F5      COL-ON  PUSH AF            Включение переменной
D322  3E01            LD   A,#01         COLLIZ.
D324  3229D3          LD   (COLLIZ),A
D327  F1              POP  AF
D328  C9              RET
D329  00      COLLIZ  DEFB 00            ;Программная переменная,
                                         ;в которой  фиксируется
                                         ;факт коллизии с пре-
                                         ;пятствием.

D32A  00              NOP
D32B  00              NOP
D32C  00              NOP
```

2.1.5. Управление спрайтами от клавиатуры с использованием прерываний 2-го рода.

Выше мы уже говорили, что существуют два основных метода управления движением спрайтов. Первый - это управление от клавиатуры или джойстика, т.е. управление от пользователя. Второй - автоматическое управление из программы. Обычно герои игры (пользовательские персонажи) управляются от клавиатуры, а монстры (непользовательские персонажи) управляются автоматически по заранее заданному алгоритму. При этом алгоритм управления ими может быть достаточно сложным и использовать методы искусственного интеллекта.

Листинг 11
Загрузчик процедуры, управляющей спрайтом от клавиатуры.

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=53100: LET long=250: LET check=81326: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 042,011,092,017,004
120 DATA 000,025,126,230,001
130 DATA 050,095,208,032,011
140 DATA 042,096,208,237,075
150 DATA 106,207,205,196,209
160 DATA 201,030,008,025,078
170 DATA 025,070,025,126,230
180 DATA 001,050,093,208,025
190 DATA 126,033,009,213,017
200 DATA 063,000,025,061,032
```

```
210 DATA 252,034,096,208,205
220 DATA 093,210,237,067,106
230 DATA 207,243,175,071,050
240 DATA 094,208,062,207,033
250 DATA 002,206,119,035,016
260 DATA 252,062,206,237,071
270 DATA 237,094,251,201,000
280 DATA 000,000,000,000,000
290 DATA 000,000,000,000,000
300 DATA 000,000,000,000,255

310 DATA 243,245,197,213,229
320 DATA 008,245,008,221,229
330 DATA 058,095,208,254,000
340 DATA 040,024,205,007,208
350 DATA 058,094,208,254,000
360 DATA 040,014,058,093,208
370 DATA 254,000,040,007,205
380 DATA 123,207,175,050,095
390 DATA 208,195,252,207,221
400 DATA 225,008,241,008,225

410 DATA 209,193,241,251,201
420 DATA 237,075,106,207,042
430 DATA 096,208,062,239,219
440 DATA 254,203,103,040,044
450 DATA 203,095,040,031,203
460 DATA 087,040,018,062,247
470 DATA 219,254,203,103,040
480 DATA 001,201,013,013,013
490 DATA 121,230,252,200,024
500 DATA 025,012,012,012,062

510 DATA 231,145,216,024,016
520 DATA 005,005,005,120,230
530 DATA 252,200,024,007,004
540 DATA 004,004,062,153,144
```

```
550 DATA 216,197,237,075,106
560 DATA 207,205,196,209,193
570 DATA 205,093,210,050,094
580 DATA 208,237,067,106,207
590 DATA 201,000,000,000,064
600 DATA 215,000,000,000,000
```

Формат управляющей процедуры.

DEF FNh (s,x,y,c,n)

s - переключатель (SWITCH). Если **s=1**, процедура должна быть включена. Если **s=0** - процедура выключается. Необходимость в специальном способе отключения процедуры связана с тем, что процедура размещается и работает в памяти резидентно, от прерываний. Поэтому ее нетрудно включить, но выключить ее из БЕЙСИКа не удастся, если не предусмотреть для этого специальные меры. Интересно, что после того, как процедура отработает и Вы в БЕЙСИКе войдете в режим редактирования программы, спрайт может продолжать бегать по экрану, подчиняясь нажатиям управляющих клавиш. Для отключения процедуры и предусмотрен параметр **s=0**. Вызов процедуры для отключения можно делать с одним этим параметром, не указывая все прочие. В этом случае процедуру выключения спрайта можно определить, как **Fn z(s)**.

x,y - координаты исходной позиции спрайта. За начало отсчета принимается левый верхний угол экрана.

0<=x<=231

0<=y<=154

c - флаг детекции коллизии. 1 - включен. 0 - выключен.

n - номер спрайта (1...10).

Листинг 12

Процедура, управляющая спрайтом от клавиатуры

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CF6A  0000    YXLAST  DEFW #0000         ;Последние координаты
                                         ;y и x. ;По ним можно
                                         ;определить, например,
                                         ;где произошла коллизия
                                         ;с препятствием.
CF6C  2A0B5C          LD   HL,(#5C0B)    ;Установление адресов па-
                                         ;раметров функции поль-
                                         ;зователя Fn ().
CF6F  110400          LD   DE,#0004      ;Сдвиг от DEFADD на 4
CF72  19              ADD  HL,DE         ;байта, переход к перво-
                                         ;му параметру.
CF73  7E              LD   A,(HL)        ;Прием параметра s.
CF74  E601            AND  #01           ;Маскирование старших би-
                                         ;тов.
CF76  325FD0          LD   (SWTCH),A     ;Запомнили в переменной.
CF79  200B            JR   NZ,CONT       ;Если s=1, то выключаться
                                         ;не надо и работу продол-
                                         ;жаем.
CF7B  2A60D0  S-LAST  LD   HL,(ADSPR)    ;Адрес шаблона спрайта.
CF7E  ED4B6ACF        LD   BC,(#CF6A)    ;Последние координаты y,x
CF82  CDC4D1          CALL SPRDRV-S      ;Стирание спрайта вызовом
                                         ;драйвера SPRDRV-S.
CF85  C9              RET                ;Выключение процедуры.
CF86  1E08    CONT    LD   E,#08         ;Сдвиг еще на 8 байтов ко
CF88  19              ADD  HL,DE         ;второму параметру.
CF89  4E              LD   C,(HL)        ;Координата x.
CF8A  19              ADD  HL,DE         ;Переход к 3-му пар-ру.
CF8B  46              LD   B,(HL)        ;Координата y.
CF8C  19              ADD  HL,DE         ;Переход к пар-ру c.
CF8D  7E              LD   A,(HL)        ;Приняли параметр.
CF8E  E601            AND  #01           ;Маскирование старших би-
                                         ;тов.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CF90  325DD0          LD    (#D05D),A    ;Запомнили состояние фла-
                                         ;га детекции коллизии в
                                         ;программной переменной.
CF93  19              ADD   HL,DE        ;Переход к пар-ру n.
CF94  7E              LD    A,(HL)       ;Приняли номер спрайта.
CF95  2109D5          LD    HL,#D509     | Определение начала
CF98  113F00          LD    DE,#003F     | шаблона спрайта в таб-
CF9B  19        LOOP  ADD   HL,DE        | лице по его номеру (см.
CF9C  3D              DEC   A            | комментарий на с.26).
CF9D  20FC            JR    NZ,LOOP      |
CF9F  2260D0          LD    (ADSPR),HL   ;Запомнили начало шаблона
CFA2  CD5DD2          CALL  SPRDRV-P     ;Печать спрайта (см. с.
                                         ;44).
CFA5  ED436ACF        LD    (YXLAST),BC  ;Текущие координаты y,x.
CFA9  F3              DI                 ;Отключение прерываний.
                                         ;на время организации
                                         ;процедуры обработки IM2.
CFAA  AF              XOR   A            ;Обнуление аккумулятора.
CFAB  47              LD    B,A          ;Нуль в счетчике "B".
CFAC  325ED0          LD    (COLLIZ),A   ;Инициализация переменной
CFAF  3ECF            LD    A,#CF        ;Формирование таблицы
CFB1  2102CE          LD    HL,#CE02     ;для IM 2. Начиная с ад-
CFB4  77        LOOP  LD    (HL),A       ;реса #CE02 256 байтов
CFB5  23              INC   HL           ;памяти заполняются кодом
CFB6  10FC            DJNZ  LOOP         ;#CF. Теперь если мы выс-
CFB8  3ECE            LD    A,#CE        ;тавим в регистре I код
CFBA  ED47            LD    I,A          ;#CE, то независимо от
                                         ;того, что будет на шине
                                         ;данных в момент прихода
                                         ;прерывания, в качестве
                                         ;процедуры для его обра-
                                         ;ботки будет назначена
                                         ;процедура, начинающаяся
                                         ;с адреса #CECF.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CFBC  ED5E            IM   2           ;Включение прерываний
                                       ;второго рода.
CFBE  FB              EI                ;Разрешение прерываний.
CFBF  C9              RET               ;Возврат.
CFC0  00              NOP
CFC1  00              NOP
CFC2  00              NOP
CFC3  00              NOP
CFC4  00              NOP
CFC5  00              NOP
CFC6  00              NOP
CFC7  00              NOP
CFC8  00              NOP
CFC9  00              NOP
CFCA  00              NOP
CFCB  00              NOP
CFCC  00              NOP
CFCD  00              NOP
CFCE  00              NOP

CFCF  FF              RST  #38          ;Начало процедуры обра-
                                        ;ботки прерываний 2-го
                                        ;рода.  RST #38 - это
                                        ;сканирование клавиатуры
                                        ;в поисках нажатой клави
                                        ;ши.
CFD0  F3              DI                ;На время обработки IM 2
                                        ;прерывания надо отклю-
                                        ;чать, иначе возможно
                                        ;зависание компьютера.
CFD1  F5              PUSH AF           |
CFD2  C5              PUSH BC           |  Сохранение всех
CFD3  D5              PUSH DE           |  основных регистров на
CFD4  E5              PUSH HL           |  машинном стеке.
CFD5  08              EX   AF,AF'       |
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CFD6  F5              PUSH AF
CFD7  08              EX   AF,AF'
CFD8  DDE5            PUSH IX
CFDA  3A5FD0          LD   A,(SWTCH)     ;Переменная SWTCH несет
                                         ;в себе параметр в, опре-
                                         ;деляющий режим Вкл/Выкл.
CFDD  FE00            CP   #00           ;Если выставлено "Выкл",
CFDF  2818            JR   Z,EXIT        ;то ничего делать не надо
                                         ;и проследуем на выход.
CFE1  CD07D0          CALL KEYB          ;Вызов процедуры, обраба-
                                         ;тывающей прерывания
                                         ;2-го рода.
CFE4  3A5ED0          LD   A,(COLLIZ)    ;Была коллизия при по-
CFE7  FE00            CP   #00           ;следней печати спрайта?
CFE9  280E            JR   Z,EXIT        ;Если нет, пройти на
                                         ;выход.
CFEB  3A5DD0          LD   A,(FLAG)      ;Если коллизия была,
                                         ;проверяем флаг детекции
CFEE  FE00            CP   #00           ;коллизии.
CFF0  2807            JR   Z,EXIT        ;Если он отключен, пере-
                                         ;ход на выход.
CFF2  CD7BCF          CALL S-LAST        ;Если он включен, стира-
                                         ;ем спрайт последний раз.
CFF5  AF              XOR  A             ;Обнуление аккумулятора.
CFF6  325FD0          LD (SWTCH),A       ;Засылка нуля в эту пере-
                                         ;менную обеспечит прекра-
                                         ;щение обработки прерыва-
                                         ;ний на следующем проходе
CFF9  C3FCCF  EXIT    JP   END           ;Шаг вперед.
CFFC  DDE1    END     POP  IX
CFFE  08              EX   AF,AF'
CFFF  F1              POP  AF            Финишные операции.
D000  08              EX   AF,AF'        Балансировка стека.
D001  E1              POP  HL            Разрешение преры-
D002  D1              POP  DE                ваний.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

D003  C1              POP  BC
D004  F1              POP  AF
D005  FB              EI
D006  C9              RET                ;Возврат

D007  ED4B6ACF  KEYB  LD   BC,(YXLAST);Координаты y,x.
D00B  2A60D0          LD   HL,(ADSPR) ;Адрес шаблона спрайта.
D00E  3EEF            LD   A,#EF      ;Старший байт адреса
                                      ;порта, равный #EF, ука-
                                      ;зывает на правый полу-
                                      ;ряд верхнего (цифрового)
                                      ;ряда клавиатуры.
D010  DBFE            IN   A,(#FE)    ;Порт клавиатуры.
D012  CB67            BIT  4,A        ;Бит 4 соответствует
                                      ;клавише "6" (вниз).
                                      ;Если эта клавиша нажа-
                                      ;та, бит - выключен.
                                      ;Подробности см. в кни-
                                      ;ге "Программирование в
                                      ;"машинном коде",
                                      ;с.92-94.
D014  282C            JR   Z,DOWN     ;Движение "вниз".
D016  CB5F            BIT  3,A        ;Проверка клавиши "7"
D018  281F            JR   Z,UP       ;Движение "вверх".
D01A  CB57            BIT  2,A        ;Проверка клавиши "8"
D01C  2812            JR   Z,RIGHT    ;Движение "вправо".
D01E  3EF7            LD   A,#F7      ;Старший байт адреса
                                      ;порта, равный #F7,
                                      ;указывает на левый
                                      ;верхний полуряд клавиа-
                                      ;туры.
D020  DBFE            IN   A,(#FE)    ;Порт клавиатуры.
D022  CB67            BIT  4,A        ;Проверка клавиши "5"
                                      ;("влево").
D024  2801            JR   Z,#D027    ;Движение влево.
```

```
Адрес Маш.код Метка   Мнемоиика        Комментарий.

D026  C9              RET              ;Выход из процедуры.

D027  0D      LEFT    DEC  C           ;Уменьшение координаты
D028  0D              DEC  C           ;х на три единицы.
D029  0D              DEC  C           ;
D02A  79              LD   A,C         ;
D02B  E6FC            AND  #FC         ;Проверка на выход за
                                       ;пределы экрана.
D02D  C8              RET  Z           ;Выход, если так.
D02E  1819            JR   NEXT        ;Иначе продолжить работу
D030  0C      RIGHT   INC  C           ;Увеличение координаты
D031  0C              INC  C           ;х на три единицы.
D032  0C              INC  C           ;
D033  3EE7            LD   A,#E7       ;Проверка на выход за
D035  91              SUB  C           ;пределы экрана.
D036  D8              RET  C           ;Возврат, если так.
D037  1810            JR   NEXT        ;Иначе продолжить работу
D039  05      UP      DEC  B           ;Уменьшение координаты
D03A  05              DEC  B           ;у на три единицы.
D03B  05              DEC  B           ;(Напомним, что у нас
                                       ;за начало отсчета принят
                                       ;левый верхний угол, по-
                                       ;этому при движении
                                       ;вверх координата у не
                                       ;увеличивается, как
                                       ;обычно, а уменьшается.)
D03C  78              LD   A,B         ;Проверка на выход за
D03D  E6FC            AND  #FC         ;пределы экрана.
D03F  C8              RET  Z           ;Возврат, если так.
D040  1807            JR   NEXT        ;Иначе продолжить работу
D042  04      DOWN    INC  B           ;Увеличение координаты
D043  04              INC  B           ;у на три единицы.
D044  04              INC  B           ;
D045  3E99            LD   A,#99       ;Проверка на выход за
D047  90              SUB  B           ;пределы экрана.
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

D048  D8              RET  C             ;Возврат, если так.
D049  C5        NEXT  PUSH BC            ;Запомнили новые коорди-
                                         ;наты.
D04A  ED4B6ACF        LD   BC,(YXLAST);Прежде чем печатать
                                         ;спрайт в новом месте,
                                         ;нужно стереть его в
                                         ;старом, для чего восста-
                                         ;новили предыдущие x,y.
D04E  CDC4D1          CALL SPRDRV-S    ;Стирание спрайта (см.
                                         ;с. 40)
D051  C1              POP  BC          ;Восстановление новых
                                         ;координат.
D052  CD5DD2          CALL SPRDRV-P    ;Печать спрайта (см.
                                         ;с. 44)
D055  325ED0          LD   (COLLIZ),A ;После выхода из проце-
                                         ;дуры печати спрайта
                                         ;аккумулятор несет ин-
                                         ;формацию о том, прои-
                                         ;зошла коллизия (1) или
                                         ;нет (0).
D058  ED436ACF        LD   (YXLAST),BC;Запомнили новые коорди-
D05C  C9              RET              ;наты в качестве текущих.
D05D  00        FLAG  DEFB 00          ;Флаг детекции коллизии
D05E  00        COLLIZ DEFB 00         ;Если коллизия при печати
                                         ;спрайта была, эта пере-
                                         ;менная включается.
D05F  00        SWTCH DEFB 00          ;Параметр s.
D060  40        ADSPR DEFW 0000        ;Адрес начала шаблона
                                         ;спрайта.
D062  00              NOP
D063  00              NOP
D064  00              NOP
D065  00              NOP
```

2.1.6. Перемещение спрайтов двойной ширины.

Итак, Вы уже убедились, чего можно достичь путем перемещения по экрану спрайтов размером 24 X 21 пиксел. В целом это более 500 пикселов и в этом объеме можно выразить очень многое (см. например некоторые образцы спрайтов в Приложении), но все же легко представить себе случай, когда этого недостаточно.

К сожалению, "Спектрум" не та машина, на которой можно работать со спрайтами очень больших размеров (недостаточное быстродействие приводит к неплавности перемещений), но немного увеличить размеры спрайтов все-таки можно. Так, например, можно представить себе составной спрайт, состоящий как бы из двух спрайтов. На рис. 16 представлен составной спрайт паровоза, состоящий из двух отдельных спрайтов.

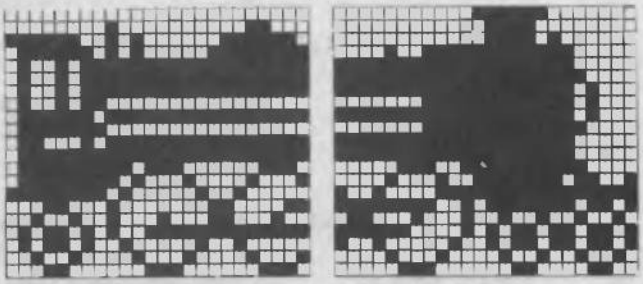


Рис. 16 Образец спрайта двойной ширины.

Спрайт 1 000 000 252 126 075 075 075 075 126 127 098 127 127
127 255 024 036 090 090 036 024 000 128 160 160 255
255 255 000 255 000 255 255 220 162 065 128 136 159
065 034 028 000 024 126 255 255 255 255 000 255 000
255 255 014 017 032 195 196 255 032 017 014

Спрайт 2 000 000 000 009 255 255 255 001 255 001 255 255 007
008 144 255 098 252 144 008 007 031 015 015 255 255
255 255 255 255 255 255 255 063 159 255 047 036 043
075 132 003 128 000 064 224 224 240 232 232 232 240
224 224 240 217 255 237 146 109 109 146 012

Для перемещения по экрану такого спрайта подготовим процедуру FN i(x,y,d,l,s,c,n). Все параметры, задаваемые при вызове процедуры, те же самые, что и в процедуре FN g(), см. с.32. Единственное отличие связано с параметром n. Если раньше он определял номер спрайта из таблицы спрайтов, то здесь речь идет уже о двух спрайтах, а не об одном и потому он определяет только номер *первого* спрайта. Предполагается, что шаблон второго спрайта расположен непосредственно за первым.

Листинг 13
Загрузчик машинного кода процедуры, управляющей перемещением спрайтов двойной ширины.

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=52400: LET long=235: LET check=77522: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 042,011,092,017,004
120 DATA 000,025,078,030,008
130 DATA 025,070,025,126,230
140 DATA 003,050,148,205,025
150 DATA 126,050,149,205,025
160 DATA 126,230,001,050,150
170 DATA 205,025,126,230,001
180 DATA 050,146,205,025,126
190 DATA 017,063,000,033,009
200 DATA 213,025,061,032,252

210 DATA 175,050,147,205,205
220 DATA 093,210,205,137,205
230 DATA 197,229,017,063,000
240 DATA 025,062,024,129,079
```

```
250 DATA 205,093,210,205,137
260 DATA 205,225,193,197,229
270 DATA 006,001,118,016,253
280 DATA 225,193,000,000,118
290 DATA 205,196,209,197,229
300 DATA 017,063,000,025,062

310 DATA 024,129,079,205,196
320 DATA 209,225,193,058,148
330 DATA 205,254,000,040,019
340 DATA 254,001,040,026,254
350 DATA 002,040,033,005,005
360 DATA 005,120,230,252,040
370 DATA 059,195,084,205,013
380 DATA 013,013,121,230,252
390 DATA 040,048,195,084,205
400 DATA 012,012,012,121,214

410 DATA 207,048,037,195,084
420 DATA 205,004,004,004,120
430 DATA 214,150,048,026,058
440 DATA 147,205,254,000,040
450 DATA 007,058.146,205,254
460 DATA 000,032,012,058,149
470 DATA 205,061,050,149,205
480 DATA 254,000,194,230,204
490 DATA 058,150,205,237,067
500 DATA 174,204,254,000,200

510 DATA 118,205,093,210,017
520 DATA 063,000,025,062,024
530 DATA 129,079,118,205,093
540 DATA 210,201,254,000,200
550 DATA 062,001,050,147,205
560 DATA 201,000,000,000,000
570 DATA 000,000,000,000,000
```

Листинг14

Процедура, управляющая перемещением спрайтов двойной ширины.

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CCAE  0000    YXLAST  DEFW #0000       ;Последние координаты
                                       ;y и x. ;По ним можно
                                       ;определить, например,
                                       ;где произошла коллизия
                                       ;с препятствием.
CCB0  2A0B5C          LD   HL,(#5C0B)
CCB3  110400          LD   DE,#0004
CCB6  19              ADD  HL,DE
CCB7  4E              LD   C,(HL)      Ввод параметров
CCB8  1E08            LD   E,#08
CCBA  19              ADD  HL,DE       из БЕЙСИКа.
CCBB  46              LD   B,(HL)
CCBC  19              ADD  HL,DE
CCBD  7E              LD   A,(HL)      Настройка
CCBE  E603            AND  #03
CCC0  3294CD          LD   (DIREC),A   программных
CCC3  19              ADD  HL,DE
CCC4  7E              LD   A,(HL)      переменных
CCC5  3295CD          LD   (DISTN),A
CCC8  19              ADD  HL,DE
CCC9  7E              LD   A,(HL)
CCCA  E601            AND  #01
CCCC  3296CD          LD   (SWTCH),A
CCCF  19              ADD  HL,DE
CCD0  7E              LD   A,(HL)
CCD1  E601            AND  #01
CCD3  3292CD          LD   (FLAG),A
CCD6  19              ADD  HL,DE
CCD7  7E              LD   A,(HL)
CCD8  113F00          LD   DE,#003F    Отыскание начала шабло-
CCDB  2109D5          LD   HL,#D509    на спрайта по его
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CCDE  19        LOOP  ADD  HL,DE         номеру.
CCDF  3D              DEC  A
CCE0  20FC            JR   NZ,LOOP
CCE2  AF              XOR  A              ;Инициализация програм-
CCE3  3293CD          LD   (COLLIZ),A     ;мной переменной.
CCE6  CD5DD2  BEGIN   CALL SPRDRV-P       ;Печать спрайта.
CCE9  CD89CD          CALL COL-ON         ;Включение переменной
                                          ;COLLIZ, если произошло
                                          ;столкновение спрайта с
                                          ;каким-либо объектом.
CCEC  C5              PUSH BC             ;Временное сохранение
CCED  E5              PUSH HL             ;регистров.
CCEE  113F00          LD   DE,#003F       ;Определение адреса на-
CCF1  19              ADD  HL,DE          ;чала шаблона второго
                                          ;спрайта.
CCF2  3E18            LD   A,#18          ;Увеличение координаты
CCF4  81              ADD A,C             ;x на 24 пункта для печа-
CCF5  4F              LD   C,A            ;ти второго спрайта.
CCF6  CD5DD2          CALL SPRDRV-P       ;Печать второго спрайта.
CCF9  CD89CD          CALL COL-ON         ;Выставление переменной
                                          ;COLLIZ.
CCFC  E1              POP HL              ;Восстановление регистров
CCFD  C1              POP  BC             ;со стека без очищения
CCFE  C5              PUSH BC             ;стека.
CCFF  E5              PUSH HL             ;
CD00  0601            LD   B,#01
CD02  76      LOOP-1  HALT
CD03  10FD            DJNZ LOOP-1          Краткая
CD05  E1              POP  HL
CD06  C1              POP  BC              пауза и восстановление
CD07  00              NOP
CD08  00              NOP                  координат 1-го спрайта
CD09  76              HALT
CD0A  CDC4D1          CALL SPRDRV-S       ;Стирание первого спрайта
CD0D  C5              PUSH BC             ;Сохранение регистров
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CD0E  E5              PUSH HL          ;на стеке.
CD0F  113F00          LD   DE,#003F    ;Определение адреса нача-
CD12  19              ADD  HL,DE       ;ла шаблона 2-го спрайта.
CD13  3E18            LD   A,#18       ;Смещение на экране
CD15  81              ADD  A,C         ;на три знакоместа
CD16  4F              LD   C.A         ;вправо.
CD17  CDC4D1          CALL SPRDRV-S    ;Стирание второго спрайта
CD1A  E1              POP  HL          :Восстановление регистров
CD1B  C1              POP  BC          ;
CD1C  3A94CD          LD   A,(DIREC)   ;Направление движения.
CD1F  FE00            CP   #00         ;Влево?
CD21  2813            JR   Z,LEFT      ;Переход, если так.
CD23  FE01            CP   #01         ;Вправо?
CD25  281A            JR   Z,RIGHT     ;Переход, если так.
CD27  FE02            CP   #02         ;Вверх?
CD29  2821            JR   Z,UP        ;Переход, если так.
CD2B  05              DEC  B           ;Последний вариант
CD2C  05              DEC  B           ;вниз. Уменьшаем на
CD2D  05              DEC  B           ;3 пункта координату y.
CD2E  78              LD   A,B         ;Проверка на выход за
CD2F  E6FC            AND  #FC         :пределы экрана.
CD31  283B            JR   Z,END       ;Если выход произошел,
                                       ;заканчиваем работу.
CD33  C354CD          JP   CONT        ;Ииаче продолжаем.
CD36  0D        LEFT  DEC  C           ;Уменьшаем на 3 пункта
CD37  0D              DEC  C           ;координату x.
CD38  0D              DEC  C           ;
CD39  79              LD   A,C         ;Проверка на выход за
CD3A  E6FC            AND  #FC         ;пределы экрана.
CD3C  2830            JR   Z,END       ;Если выход произошел,
                                       ;заканчиваем работу.
CD3E  C354CD          JP   CONT        ;Иначе продолжаем.
CD41  0C        RIGHT INC  C           ;Увеличиваем на 3 пункта
CD42  0C              INC  C           ;координату x.
CD43  0C              INC  C           ;
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CD44  79              LD   A,C         ;Проверка на выход за
CD45  D6CF            SUB  #CF         ;пределы экрана.
CD47  3025            JR   NC,END      ;Если выход произошел,
                                       ;заканчиваем работу.
CD49  C354CD          JP   CONT        ;Иначе продолжаем.
CD4C  04        UP    INC  B           ;Увеличиваем на 3 пункта
CD4D  04              INC  B           ;координату y.
CD4E  04              INC  B           ;
CD4F  78              LD   A,B         ;Проверка на выход за
CD50  D696            SUB  #96         ;пределы экрана.
CD52  301A            JR   NC,END      ;Если выход произошел,
                                       ;заканчиваем работу.
CD54  3A93CD    CONT  LD   A,(COLLIZ)  ;Проверяем, была колли-
CD57  FE00            CP   #00         ;зия или нет.
CD59  2807            JR   Z,NEXT      ;Обход, если нет.
CD5B  3A92CD          LD   A,(FLAG)    ;Если была, то надо про-
CD5E  FE00            CP   #00         ;верить состояние флага
                                       ;детекции коллизии.
CD60  200C            JR   NZ,END      ;Если он был включен,
                                       ;то коллизия приводит к
                                       ;окончанию работы.
CD62  3A95CD    NEXT  LD   A,(DISTN)   ;Иначе продолжаем работу.
CD65  3D              DEC  A           ;Уменьшаем дистанцию
CD66  3295CD          LD   (DISTN),A   ;перемещения спрайта.
CD69  FE00            CP   #00         ;
CD6B  C2E6CC          JP   NZ,BEGIN    ;Если не весь путь пока
                                       ;пройден - возврат.
CD6E  3A96CD    END   LD   A,(SWTCH)   ;Проверяем параметр s.
CD71  ED43AECC        LD   (YXLAST),BC;Запомнили последние
                                       ;координаты.
CD75  FE00            CP   #00         ;Если s=0, то можно
CD77  C8              RET  Z           ;выходить из процедуры.
CD78  76              HALT             ;
CD79  CD5DD2          CALL SPRDRV-P    ;Печать левой половины
                                       ;спрайта в последний раз
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

CD7C  113F00           LD   DE,#003F      ;Переход к правой
CD7F  19               ADD  HL,DE         ;половине
CD80  3E18             LD   A,#18         ;спрайта.
CD82  81               ADD  A,C           ;
CD83  4F               LD   C,A           ;
CD84  76               HALT               ;
CD85  CD5DD2           CALL SPRDRV-P      ;Печать правой половины
                                          ;спрайта.
CD88  C9               RET                ;Окончательный выход.

CD89  FE00     COL-ON  CP #00             | Если произошла колли-
CD8B  C8               RET  Z             | зия, эта процедура
CD8C  3E01             LD   A,#01         | включает переменную
CD8E  3293CD           LD   (COLLIZ),A    | COLLIZ.
CD91  C9               RET                |

CD92  00       FLAG    DEFB 00            ;Флаг детекции коллизии.
CD93  00       COLLIZ  DEFB 00            ;Переменная, фиксирующая
                                          ;факт прохождения колли-
                                          ;зии.
CD94  00       DIREC   DEFB 00            ;Направление перемещения.
CD95  00       DISTN   DEFB 00            ;Дистанция перемещения.
CD96  00       SWTCH   DEFB 00            ;Переключатель s.
CD97  00               NOP
CD98  00               NOP
CD99  00               NOP
CD9A  00               NOP
```

2.1.7. Перемещение спрайтов двойной высоты.

Для перемещения по экрану такого спрайта подготовим процедуру FN j(x,y,d,l,s,c,n). Все параметры, задаваемые при вызове процедуры, те же самые, что и в процедуре FN i(), см. с.61.

Листинг 15.
Загрузчик машинного кода процедуры, управляющей перемещением спрайтов двойной высоты.

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=52100: LET long=230: LET check=76012: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 000,042,011,092,017
120 DATA 004,000,025,078,030
130 DATA 008,025,070,025,126
140 DATA 230,003,050,103,204
150 DATA 025,126,050,104,204
160 DATA 025,126,230,001,050
170 DATA 105,204,025,126,230
180 DATA 001,050,101,204,025
190 DATA 126,017,063,000,033
200 DATA 009,213,025,061,032

210 DATA 252,175,050,102,204
220 DATA 205,093,210,205,086
230 DATA 204,197,229,017,063
240 DATA 000,025,062,021,128
250 DATA 071,205,093,210,205
260 DATA 086,204,225,193,197
270 DATA 229,006,001,118,016
280 DATA 253,225,193,000,000
290 DATA 118,205,196,209,197
300 DATA 229,017,063,000,025

310 DATA 062,021,128,071,205
320 DATA 196,209,225,193,058
```

```
330 DATA 103,204,254,000,040
340 DATA 019,254,001,040,026
350 DATA 254,002,040,033,005
360 DATA 005,005,120,230,252
370 DATA 040,051,195,041,204
380 DATA 013,013,013,121,230
390 DATA 252,040,040,195,041
400 DATA 204,012,012,012,121

410 DATA 214,231,048,029,195
420 DATA 041,204,004,004,004
430 DATA 120,214,132,048,018
440 DATA 058,102,204,254,000
450 DATA 032,011,058,104,204
460 DATA 061,050,104,204,254
470 DATA 000,032,128,058,105
480 DATA 204,237,067,130,203
490 DATA 254,000,200,118,205
500 DATA 093,210,017,063,000

510 DATA 025,062,021,128,071
520 DATA 118,205,093,210,201
530 DATA 254,000,200,058,101
540 DATA 204,254,000,200,062
550 DATA 001,050,102,204,201
560 DATA 000,000,001,000,000
```

Листинг 16

Процедура, управляющая перемещением спрайтов двойной высоты.

```
Адрес Маш.код Метка   Мнемоника       Комментарий.

CB82  0000    YXLAST  DEFW #0000      ;Последние координаты
                                      ;y и x. ;По ним можно
                                      ;определить, например,
                                      ;где произошла коллизия
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CB84  00              NOP
CB85  2A0B5C          LD   HL,(#5C0B)
CB88  110400          LD   DE,#0004
CB8B  19              ADD  HL,DE          Ввод параметров
CB8C  4E              LD   C,(HL)
CB8D  1E08            LD   E,#08          из БЕЙСИКа.
CB8F  19              ADD  HL,DE
CB90  46              LD   B,(HL)
CB91  19              ADD  HL,DE          Настройка
CB92  7E              LD   A,(HL)
CB93  E603            AND  #03            программных
CB95  3267CC          LD   (DIREC),A
CB98  19              ADD  HL,DE          переменных
CB99  7E              LD   A,(HL)
CB9A  3268CC          LD   (DISTN),A
CB9D  19              ADD  HL,DE
CB9E  7E              LD   A,(HL)
CB9F  E601            AND  #01
CBA1  3269CC          LD   (SWTCH),A
CBA4  19              ADD  HL,DE
CBA5  7E              LD   A,(HL)
CBA6  E601            AND  #01
CBA8  3265CC          LD   (FLAG),A
CBAB  19              ADD  HL,DE
CBAC  7E              LD   A,(HL)         ;
CBAD  113F00          LD   DE,#003F       Отыскание начала шабло-
CBB0  2109D5          LD   HL,#D509       на спрайта по его
CBB3  19        LOOP  ADD  HL,DE          номеру.
CBB4  3D              DEC  A
CBB5  20FC            JR   NZ,LOOP
CBB7  AF              XOR  A              ;Инициализация програм-
CBB8  3266CC          LD   (COLLIZ),A     ;мной переменной.
CBBB  CD5DD2    BEGIN CALL SPRDRV-P       ;Печать спрайта.
CBBE  CD56CC          CALL COLTST         ;Проверка была коллизия
                                          ;или нет.
```

```
Адрес  Маш.код  Метка   Мнемоника        Комментарий.

CBC1   C5                PUSH BC          ;Временное сохранение
CBC2   E5                PUSH HL          ;регистров.
CBC3   113F00            LD   DE,#003F    ;Определение адреса на-
CBC6   19                ADD  HL,DE       ;чала шаблона второго
                                          ;спрайта.
CBC7   3E15              LD   A,#15       ;Увеличение координаты
CBC9   80                ADD  A,B         ;У на 21 пункт для опре-
                                          ;деления координат пе-
CBCA   47                LD   B,A         ;чати второго спрайта.
CBCB   CD5DD2            CALL SPRDRV-P    ;Печать 2-го спрайта.
CBCE   CD56CC            CALL COLTEST     ;Проверка коллизии.
CBD1   E1                POP  HL          ;Восстановление регистров
CBD2   C1                POP  BC          ;со стека без очищения
CBD3   C5                PUSH BC          ;стека.
CBD4   E5                PUSH HL          ;
CBD5   0601              LD   B,#01       ||
CBD7   76                HALT             ||
CBD8   10FD              DJNZ #CBD7       ||  Краткая
CBDA   E1                POP  HL          ||
CBDB   C1                POP  BC          ||  пауза и восстановление
CBDC   00                NOP              ||
CBDD   00                NOP              ||  координат 1-го спрайта
CBDE   76                HALT             ||
CBDF   CDC4D1            CALL SPRDRV-S    ;Стирание первого спрайта
CBE2   C5                PUSH BC          ;Сохранение регистров
CBE3   E5                PUSH HL          ;на стеке.
CBE4   113F00            LD   DE,#003F    ;Определение адреса начала
CBE7   19                ADD  HL,DE       ;шаблона 2-го спрайта.
CBE8   3E15              LD   A,#15       ;Смещение на экране
CBEA   80                ADD  A,B         ;на 21 пиксел
CBEB   47                LD   B,A         ;вниз.
CBEC   CDC4D1            CALL SPR-DRV-S   ;Стирание второго спрайта.
CBEF   E1                POP  HL          ;Восстановление регистров
CBF0   C1                POP  BC          ;
CBF1   3A67CC            LD   A,(DIREC)   ;Направление движения.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CBF4  FE00            CP    #00        ;Влево?
CBF6  2813            JR    Z,LEFT     ;Переход, если так.
CBF8  FE01            CP    #01        ;Вправо?
CBFA  281A            JR    Z,RIGHT    ;Переход, если так.
CBFC  FE02            CP    #02        ;Вверх?
CBFE  2821            JR    Z,UP       ;Переход, если так.
CC00  05              DEC   B          ;Последний вариант -
CC01  05              DEC   B          ;вниз. Уменьшаем на
CC02  05              DEC   B          ;3 пункта координату y.
CC03  78              LD    A,B        ;Проверка на выход за
CC04  E6FC            AND   #FC        ;пределы экрана.
CC06  2833            JR    Z,END      ;Если выход произошел,
                                       ;заканчиваем работу.
CC08  C329CC          JP    CONT       ;Иначе продолжаем.
CC0B  0D        LEFT  DEC   C          ;Уменьшаем на 3 пункта
CC0C  0D              DEC   C          ;координату x.
CC0D  0D              DEC   C          ;
CC0E  79              LD    A,C        ;Проверка на выход за
CC0F  E6FC            AND   #FC        ;пределы экрана.
CC11  2828            JR    Z,END      ;Если выход произошел,
                                       ;заканчиваем работу.
CC13  C329CC          JP    CONT       ;Иначе продолжаем.
CC16  0C        RIGHT INC   C          ;Увеличиваем на 3 пункта
CC17  0C              INC   C          ;координату x.
CC18  0C              INC   C          ;
CC19  79              LD    A,C        ;Проверка на выход за
CC1A  D6E7            SUB   #E7        ;пределы экрана.
CC1C  301D            JR    NC,END     ;Если выход произошел,
                                       ;заканчиваем работу.
CC1E  C329CC          JP    CONT       ;Иначе продолжаем.
CC21  04        UP    INC   B          ;Увеличиваем на 3 пункта
CC22  04              INC   B          ;координату y.
CC23  04              INC   B          ;
CC24  78              LD    A,B        ;Проверка на выход за
CC25  D684            SUB   #84        ;пределы экрана.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CC27  3012            JR   NC,END        ;Если выход произошел,
                                         ;заканчиваем работу.
CC29  3A66CC    CONT  LD   A,(COLLIZ)    ;Проверяем, была колли-
CC2C  FE00            CP   #00           ;зия или нет.
CC2E  200B            JR   NZ,END        ;Если была, то на выход
CC30  3A68CC          LD   A,(DISTN)     ;Иначе продолжаем работу.
CC33  3D              DEC  A             ;Уменьшаем дистанцию
CC34  3268CC          LD   (DISTN),A     ;перемещения спрайта.
CC37  FE00            CP   #00           ;Если не весь путь пока
CC39  2080            JR   NZ,BEGIN      ;пройден - возврат.
CC3B  3A69CC    END   LD   A,(SWTCH)     ;Проверяем параметр s.
CC3E  ED4382CB        LD   (YXLAST),BC;Запомнили последние
                                         ;координаты.
CC42  FE00            CP   #00           ;Если s=0, то можно
CC44  C8              RET  Z             ;выходить из процедуры.
CC45  76              HALT
CC46  CD5DD2          CALL SPRDRV-P      ;Иначе надо напечатать
                                         ;спрайт последний раз.
                                         ;Печать верхней половины.
CC49  113F00          LD   DE,#003F      ;Переход к нижней
CC4C  19              ADD  HL,DE         ;половине
CC4D  3E15            LD   A,#15         ;спрайта.
CC4F  80              ADD  A,B           ;
CC50  47              LD   B,A           ;
CC51  76              HALT               ;
CC52  CD5DD2          CALL SPRDRV-P      ;Печать нижней половины
                                         ;спрайта.
CC55  C9              RET                ;Окончательный выход.

CC56  FE00      COLTST CP #00            ;Проверка на ноль и воз-
CC58  C8              RET  Z             ;врат, если коллизии не
                                         ;было.
CC59  3A65CC          LD   A,(FLAG)      ;Если была, проверяем
                                         ;флаг детекции коллизии.
CC5C  FE00            CP   #00           ;Если он выключен, нет
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

CC5E  C8              RET  Z             ;вопросов и возврат.
CC5F  3E01            LD   A,#01         ;Иначе включаем перемен-
CC61  3266CC          LD   (COLLIZ),A    ;ную, фиксирующую факт
                                         ;коллизии.
CC64  C9              RET                ;Возврат.

CC65  00       FLAG   DEFB 00
CC66  00       COLLIZ DEFB 00
CC67  00       DIREC  DEFB 00
CC68  00       DISTN  DEFB 00
CC69  00       SWTCH  DEFB 00
```

2.1.8. Полная анимация спрайтов.

До сих пор мы занимались только перемещением спрайтов. Это в какой-то степени является динамической графикой, но все-таки не совсем, поскольку так можно изображать далеко не все объекты. Этот метод был хорош для движения самолетов, кораблей, автомобилей, локомотивов и пр. технических устройств. Большинство же объектов живой природы в процессе перемещения меняют не только свои координаты, но и свою форму. Поэтому для того, чтобы, например, изобразить скачущего всадника, нам надо не только обеспечить перемещение спрайта по экрану, но и выполнить переключение нескольких спрайтов, в которых изображены разные фазы движения.

Анимация, как и большинство прочих технологий в компьютерной графике, связана с обманом зрения. Быстрая смена нескольких статичных изображений создает иллюзию движения. Но анимация может быть исполнена и без движения, представьте себе, например человечка, подпрыгивающего на одном месте.

Как только Вы начнете ставить свои эксперименты, Вы увидите, что даже очень малые изменения в спрайтах дают весьма ощу-

тимые результаты. Посмотрите на кадры наездника, приведенные на рис. 17. Практически все тело лошади и сам наездник остаются неподвижны. Изменения касаются только ног и хвоста. Если бы мы попробовали внести изменения и в положение тела, то результирующая последовательность кадров была бы очень неплавной.

Во многих случаях Вам вполне достаточно иметь два кадра, изображающих две фазы движения, и переключать их с тем, чтобы получить анимационный эффект. Но мы здесь рассмотрим более общий случай и подготовим процедуру, позволяющую иметь столько кадров, сколько нам понадобится, правда не более, чем есть в таблице спрайтов. Кроме прочих параметров, рассмотренных в вышеприведенных процедурах, здесь появятся два новых параметра - **f** - количество кадров **(frames)** в анимационной последовательности и **v** **(velocity)** - скорость анимации. Обратите внимание на этот параметр. Он интересен для постановки собственных экспериментов. Изменяя его, можно достигать самых разных результатов.

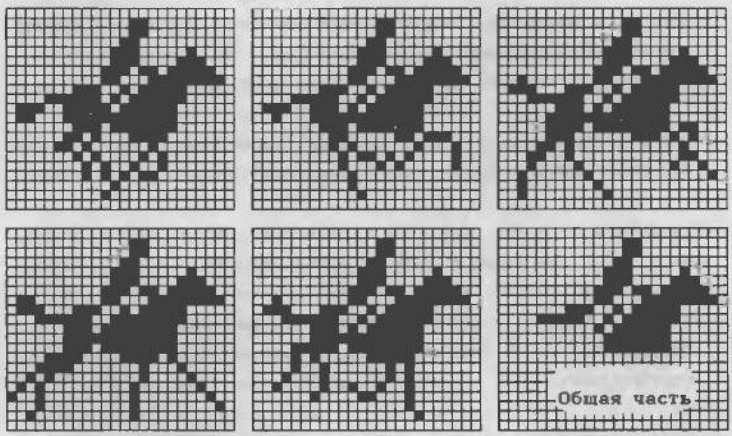


Рис. 17 Пять фаз движения спрайта

Формат процедуры:

FN k(x,y,d,l,s,f,c,v,n)

x,y - координаты исходной позиции спрайта. Они относятся к левому верхнему углу и задаются в пикселах.

d - direction - параметр, определяющий направление перемещения.
d=0 - перемещение справа налево.
d=1 - перемещение слева направо.
d=2 - перемещение снизу вверх.
d=1 - перемещение сверху вниз.

l - параметр, определяющий дистанцию, на которую перемещается спрайт. За единицу принято перемещение на три пиксела. Таким образом, если устаиовить здесь иапример число 20, то спрайт продвинется на 60 пикселов и остановится.

s - switch - параметр, определяющий остается спрайт на экране после окоичаиия перемещения или иет. Если s=0, то он останется на экране. Если s=1, то спрайт исчезнет.

f - количество кадров в анимациоиной последовательности. Оно не может быть больше, чем число спрайтов в таблице (=<10)

c - collision - флаг детекции коллизии. Если его включить (c=1), то при столкновении с любым препятствием (со включенными пикселами экрана) спрайт остановится после того, как три пиксела спрайта "наедут" на препятствие. Если его отключить (c=0), то спрайт будет проходить через любое препятствие.

v - скорость анимации (1<=v<=255, самая медлениая).

n - номер первого спрайта из Вашей кадровой последовательности. Если у Вас в таблице спрайтов задано иесколько шабло-

нов, то здесь достаточно указать на нужный Вам номер спрайта.

Листинг 17
Загрузчик машинного кода процедуры аиимации спрайта.

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=51700: LET long=275: LET check=82463: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машиного кода

110 DATA 042,011,092,017,004
120 DATA 000,025,078,030,008
130 DATA 025,070,025,126,230
140 DATA 003,050,209,202,025
150 DATA 126,050,211,202,025
160 DATA 126,230,001,050,212
170 DATA 202,025,126,050,216
180 DATA 202,050,215,202,025
190 DATA 126,230,001,050,210
200 DATA 202,025,126,050,217

210 DATA 202,025,126,017,063
220 DATA 000,033,202,212,025
230 DATA 061,032,252,034,213
240 DATA 202,058,215,202,061
250 DATA 050,215,202,254,000
260 DATA 032,009,058,216,202
270 DATA 050,215,202,042,213
280 DATA 202,017,063,000,025
290 DATA 205,093,210,254,000
300 DATA 040,019,058,210,202
```

```
310 DATA 254,000,040,012,058
320 DATA 212,202,198,001,230
330 DATA 001,050,212,202,024
340 DATA 090,229,213,197,058
350 DATA 217,202,071,017,000
360 DATA 000,033,000,000,237
370 DATA 176,193,209,225,118
380 DATA 205,196,209,058,209
390 DATA 202,254,000,040,019
400 DATA 254,001,040,026,254

410 DATA 002,040,033,005,005
420 DATA 005,120,230,252,040
430 DATA 045,195,182,202,013
440 DATA 013,013,121,230,252
450 DATA 040,034,195,182,202
460 DATA 012,012,012,121,214
470 DATA 231,048,023,195,182
480 DATA 202,004,004,004,120
490 DATA 214,150,048,012,058
500 DATA 211,202,061,050,211

510 DATA 202,254,000,194,054
520 DATA 202,058,212,202,237
530 DATA 067,242,201,254,000
540 DATA 200,118,205,093,210
550 DATA 201,001,000,000,001
560 DATA 194,214,001,003,050
570 DATA 231,048,022,195,232
580 DATA 202,004,004,004,120
590 DATA 214,150,048,011,058
600 DATA 003,203,061,050,003

610 DATA 203,254,000,032,155
620 DATA 058,004,203,237,067
630 DATA 086,202,254,000,200
640 DATA 118,205,093,210,201
650 DATA 000,000,000,000,000
```

Листинг 18
Процедура, управляющая
аиимацией спрайтов.

```
Адрес Маш.код Метка   Мнемоника        Коммеитарий.

C9F2  0000    YXLAST  DEFW #0000       ;Последние координаты
                                       ;y и x. ;По ним можно
                                       ;определить, иапример,
                                       ;где произошла коллизия
                                       ;с препятствием.
C9F4  2A0B5C          LD   HL,(#5C0B)  ;Указание на адрес, в ко-
                                       ;тором располагаются па-
                                       ;раметры функции пользо-
                                       ;вателя FN k().
C9F7  110400          LD   DE,#0004    ;Вычисление адреса пер-
C9FA  19              ADD  HL,DE       ;вого параметра.
C9FB  4E              LD   C,(HL)      ;Прием параметра x.
C9FC  1E08            LD   E,#08       ;Смещение ко второму
C9FE  19              ADD  HL,DE       ;параметру.
C9FF  46              LD   B,(HL)      ;Прием параметра y.
CA00  19              ADD  HL,DE       ;Смещение к третьему
                                       ;параметру.
CA01  7E              LD   A,(HL)      ;Прием параметра d.
CA02  E603            AND  #03         ;Маскирование старших
                                       ;битов.
CA04  32D1CA          LD   (DIREC),A   ;Выставление программной
                                       ;переменной.
CA07  19              ADD  HL,DE       ;Смещеиие к четвертому
                                       ;параметру.
CA08  7E              LD   A,(HL)      ;Прием параметра 1.
CA09  32D3CA          LD   (DISTN),A   ;Выставление программной
                                       ;переменной.
CA0C  19              ADD  HL,DE       ;Смещение к пятому
                                       ;параметру.
CA0D  7E              LD   A,(HL)      ;Прием параметра s.
CA0E  E601            AND  #01         ;Маскирование старших
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

                                         ;битов.
CA10  32D4CA          LD    (SWTCH).A    :Выставление программной
                                         ;переменной.
CA13  19              ADD   HL,DE        ;Смещение к шестому
                                         ;параметру.
CA14  7E              LD    A,(HL)       ;Прием параметра f.
CA15  32D8CA          LD    (FRAMES),A   ;Выставление программных
CA18  32D7CA          LD    (FRAM-1),A   ;переменных.
CA1B  19              ADD   HL,DE        ;Смещение к седьмому
                                         ;параметру.
CA1C  7E              LD    A,(HL)       ;Прием параметра c.
CA1D  E601            AND   #01          :Маскирование старших
                                         ;битов.
CA1F  32D2CA          LD    (FLAG).A     ;Выставление переменной.
CA22  19              ADD HL,DE          ;Переход к восьмому
                                         ;параметру.
CA23  7E              LD    A,(HL)       ;Прием параметра v.
CA24  32D9CA          LD    (RATE),A     ;Выставление программной
                                         ;переменной.
CA27  19              ADD   HL,DE        ;Переход к девятому
                                         :параметру.
CA28  7E              LD    A,(HL)       ;Прием номера первого
                                         ;спрайта из последова-
                                         ;тельности.
CA29  113F00          LD    DE,#003F     ;Длина шаблона спрайта.
CA2C  21CAD4          LD    HL,#D4CA     ;Адрес указывает на 126
                                         ;байтов ниже, чем начало
                                         ;таблицы спрайтов. Каж-
                                         ;дый спрайт занимает 63
                                         ;байта, поэтому адрес
                                         ;указывает на две длины
                                         ;спрайта ниже, чем начало
                                         ;таблицы.
CA2F  19        LOOP  ADD   HL,DE        Отыскание адреса нача-
CA30  3D              DEC   A            ла шаблона спрайта по
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CA31  20FC            JR   NZ,LOOP     его номеру. В результате
                                       HL указывает на одну
                                       длину спрайта ниже, чем
                                       его истинное положение
                                       в таблице.
CA33  22D5CA          LD   (ADRES),HL  ;Выставление программной
                                       ;переменной.
CA36  3AD7CA  BEGIN   LD   A,(FRAM-1)  ;Сколько кадров осталось
                                       ;напечатать?
CA39  3D              DEC  A           ;Уменьшить на 1 и
CA3A  32D7CA          LD   (FRAM-1),A  ;запомнить.
CA3D  FE00            CP   #00         ;Если ие все кадры напеча-
CA3F  2009            JR   NZ,BYPASS   ;таиы, то обход.
CA41  3AD8CA          LD   A,(FRAMES)  ;Если все, то надо осве-
CA44  32D7CA          LD   (FRAM-1),A  ;жить перемеиную FRAM-1.
CA47  2AD5CA          LD   HL,(ADRES)  ;
CA4A  113F00  BYPASS  LD   DE,#003F    ;
CA4D  19              ADD  HL,DE       ;Теперь HL указывает точ-
                                       ;но на начало нужного
                                       ;спрайта в таблице шабло-
                                       ;нов.
CA4E  CD5DD2          CALL SPRDRV-P    ;Вызов процедуры печати
                                       ;спрайта (см. с.44).
CA51  FE00            CP   #00         ;При печати спрайта была
                                       ;коллизия с каким-либо
                                       ;объектом иа экране?
CA53  2813            JR   Z,CONT      ;Если нет, переход.
CA55  3AD2CA          LD   A,(FLAG)    ;Если была, надо проверить
                                       ;флаг детекции коллизии.
CA58  FE00            CP   #00         ;Если он выключен, про-
CA5A  280C            JR   Z,CONT      ;должить работу.
CA5C  3AD4CA          LD   A,(SWTCH)   ;Если он включен, подго-
CA5F  C601            ADD  A,#01       ;тавливаем выход из
CA61  E601            AND  #01         ;процедуры.
CA63  32D4CA          LD   (SWTCH),A   ;
```

```
Адрес Маш.код Метка   Мнемоника        Коммеитарий.

CA66  185A            JR   END         ;
CA68  E5      CONT    PUSH HL          ;Сохранение основных
CA69  D5              PUSH DE          ;регистров от поврежде-
CA6A  C5              PUSH BC          ;ния командой LDIR.
CA6B  3AD9CA          LD   A,(RATE)    ;Скорость анимации.
CA6E  47              LD   B,A         ;
CA6F  110000          LD   DE,#0000    | Это не блочная перебро-
CA72  210000          LD   HL,#0000    | ска по команде LDIR, а
CA75  EDB0            LDIR             | обыкновенная затяжка
                                       | времени. Чем больше
                                       | число в регистре B, тем
                                       | длительнее пауза.
CA77  C1              POP  BC          ;Восстаиовление регист-
CA78  D1              POP  DE          ;ров со стека.
CA79  E1              POP  HL          ;
CA7A  76              HALT             ;Краткая пауза.
CA7B  CDC4D1          CALL SPRDRV-S    ;Стирание спрайта (см.
                                       ;стр. 40)
CA7E  3AD1CA          LD   A,(DIREC)   ;В каком направлении
                                       ;движется спрайт?
CA81  FE00            CP   #00         ;Влево?
CA83  2813            JR   Z,LEFT      ;Переход, если так.
CA85  FE01            CP   #01         ;Вправо?
CA87  281A            JR   Z,RIGHT     ;Переход, если так.
CA89  FE02            CP   #02         ;Вверх?
CA8B  2821            JR   Z,UP        ;Переход, если так.
                                       ;Во всех прочих слу-
                                       ;чаях остается только
                                       ;движение вниз.
CA8D  05              DEC  B           ;Уменьшение координаты
CA8E  05              DEC  B           ;У на три пункта.
CA8F  05              DEC  B           ;
CA90  78              LD   A,B         ;Проверка на выход за
CA91  E6FC            AND  #FC         ;пределы экрана.
CA93  282D            JR   Z,END       ;Конец работы, если так.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

CA95  C3B6CA          JP    NEXT        ;Иначе продолжаем.
CA98  0D      LEFT    DEC   C           ;Уменьшение координаты
CA99  0D              DEC   C           ;x на три пункта.
CA9A  0D              DEC   C           ;
CA9B  79              LD    A,C         ;Проверка на выход за
CA9C  E6FC            AND   #FC         ;пределы экрана.
CA9E  2822            JR    Z,END       ;Конец работы, если так.
CAA0  C3B6CA          JP    NEXT        ;Иначе продолжаем.
CAA3  0C      RIGHT   INC   C           ;Увеличение координаты
CAA4  0C              INC   C           ;x на три пункта.
CAA5  0C              INC   C           ;
CAA6  79              LD    A,C         ;Проверка на выход за
CAA7  D6E7            SUB   #E7         ;пределы экрана.
CAA9  3017            JR    NC,END      ;Конец работы, если так.
CAAB  C3B6CA          JP    NEXT        ;Иначе продолжаем.
CAAE  04      UP      INC   B           ;Увеличение координаты
CAAF  04              INC   B           ;y на три пункта.
CAB0  04              INC   B           ;
CAB1  78              LD    A,B         ;Проверка на выход за
CAB2  D696            SUB   #96         ;пределы экрана.
CAB4  300C            JR    NC,END      ;Конец работы, если так
CAB6  3AD3CA  NEXT    LD    A,(DISTN)   ;Уменьшаем счетчик
CAB9  3D              DEC   A           ;пройденного расстояния.
CABA  32D3CA          LD    (DISTN),A   ;
CABD  FE00            CP    #00         ;Проверяем его на конец.
CABF  C236CA          JP    NZ,BEGIN    ;Если не конец, то
                                        ;возврат.
CAC2  3AD4CA  END     LD    A,(SWTCH)   ;Проверка параметра s.
CAC5  ED43F2C9        LD    (YXLAST),BC;Запомнили координаты
                                        ;последней точки в про-
                                        ;граммной переменной.
CAC9  FE00            CP    #00         ;Если s=0, то ничего
                                        ;больше делать не надо
CACB  C8              RET   Z           ;Окончательный выход.
CACC  76              HALT              ;Если s=1, напечатаем
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

                                       ;спрайт в последний раз.
CACD  CD5DD2          CALL SPRDRV-P    ;Вызов процедуры печати.
CAD0  C9              RET              ;Выход из процедуры.

CAD1  00      DIREC   DEFB 00          ;Программная переменная,
                                       ;указывающая направление
                                       ;перемещения спрайта.
CAD2          FLAG    DEFB 00          ;Флаговая переменная -
                                       ;флаг детекции коллизии.
CAD3          DISTN   DEFB 00          ;Программная переменная,
                                       ;указывающая дистанцию
                                       ;перемещения спрайта.
CAD4  00      SWTCH   DEFB 00          ;Параметр s.
CAD5  0000    ADRES   DEFW 0000        ;Программная переменная,
                                       ;в которой хранится ад-
                                       ;рес шаблона спрайта в
                                       ;таблице спрайтов.
CAD7  00      FRAM-1  DEFB 00          ;Программная переменная,
                                       ;хранящая количество не-
                                       ;показанных спрайтов из
                                       ;анимационной последова-
                                       ;тельности.
CAD8  00      FRAMES  DEFB 00          ;Программная переменная,
                                       ;хранящая количество
                                       ;кадров (спрайтов) в ани-
                                       ;мационной последователь-
                                       ;ности.
CAD9  00      RATE    DEFB 00          ;Программная переменная,
                                       ;хранящая частоту смены
                                       ;кадров. Чем она больше,
                                       ;тем медленнее анимация.
```

Как же работает эта процедура? Отдельные спрайты, которых столько же, сколько кадров в анимационной последовательности, берутся из таблицы спрайтов и по-очереди печатаются на экране.

Сначала печатается первый спрайт. Далее он стирается и, вместо того, чтобы напечатать его же со смещением в три пиксела в заданном направлении, печатается второй спрайт, затем третий и т.д. Процесс повторяется до тех пор, пока не будет напечатан последний спрайт из кадровой последовательности, после чего вновь печатается первый.

2.1.9 Некоторые рекомендации.

Для того, чтобы получить благоприятный результат, надо придерживаться нескольких основополагающих принципов.

Во-первых, помните, что для получения гладкой и плавной анимации, напоминающей движение реальных тел в живой природе, надо стараться делать отдельные фазы движения с минимальными изменениями. Чем проще, тем лучше. Глаз сам поможет обмануть наблюдателя.

Во-вторых, необходимо помнить, что если у Вас есть последовательность из N кадров, то за N-ным кадром идет первый кадр. И потому изменения при переходе от кадра N к кадру 1 должны быть столь же минимальны, как при переходе от кадра 1 к кадру 2. То есть последовательность смены кадров должна быть ЗАМКНУТА. Наиболее часто встречающаяся ошибка начинающих - составление ОТКРЫТОЙ последовательности спрайтов, в которой последний спрайт плохо согласован с первым.

В-третьих, если Вы приступили к работе над последовательностью кадров, то на первых порах не перемещайте спрайты по экрану, а ограничьтесь печатью всей последовательности попеременно в одной позиции. Это даст Вам возможность легко обнаружить ошибки и поправить их. Даже очень малая несогласованность в движении отдельных точек изображения может привести к очень плохим конечным результатам, а причины плохого результата очень трудно обнаружить на двигающихся спрайтах.

В заключение, надо сказать несколько слов и о работе с цветом. До сих пор все, что мы писали об анимации спрайтов, относилось к черно-белой (или одноцветной) печати. Для анимации цветных спрайтов надо понять еще и о проблемах, связанных с экраном цветовых атрибутов "Спектрума".

По содержанию наших предыдущих книг Вы знаете, что цветовые атрибуты "Спектрума" привязаны не к пикселам, а к знакоместам и разрешение экрана цветовых атрибутов составляет 32X24 знакоместа (вместо 256X192 пиксела в черно-белой графике). Каждое знакоместо - это квадрат размером 8X8 пикселов.

Что же делать, чтобы двигать цветной спрайт? Прежде всего, здесь от Вас потребуется не только написать процедуру для перемещения черно-белого шаблона, но и процедуру для перемещения цветных атрибутов спрайта. Второй вопрос - как их двигать? Очевидно, что они могут быть двинуты только на величину знакоместа, т.е. не менее, чем на восемь пикселов. Но если Вы попробуете и черно-белую составляющую двигать тоже на 8 пикселов за один такт, то увидите, что результат будет очень грубым. Фактически это уже будет не растровая анимация, а блочная, причем такого сорта, какую можно получить и из БЕЙСИКа с помощью оператора PRINT AT... . Поэтому здесь делают так, что на один такт движения атрибутов приходятся несколько тактов движения черно-белого шаблона.

Если черно-белый шаблон двигать на 4 пиксела за такт, то на два такта движения шаблона придется один такт движения атрибутов. Если двигать ч/б шаблон на 2 пиксела за такт, то на четыре такта движения шаблона придется один такт движения атрибутов.

Но и здесь возможны интересные решения. Так, например, в программе Everyone's a Wally (фирма Mikrogen) герои двигаются по более хитрому закону, когда на шесть тактов движения шаблона приходится два такта движения атрибутов. Как это сделано, показано на рис.18. Шаблон спрайта в этой игре имеет размер 16X16,

т.е. в ширину занимает 2 знакоместа. Отдельные кадры печатаются со смещением в 2 пиксела и поэтому бывают моменты, когда шаблон растягивается на три знакоместа. Можно было бы атрибуты спрайта всегда печатать с размером 3 знакоместа в ширину, но для большей плавности перемещения в игре сделано так, что атрибуты иногда занимают два знакоместа, а иногда три.

```
Условные обозначения: XXXXX - шаблон
                      AAAAA - атрибуты
                      00000 - пустые места

Кадр 1       XXXXXXXX XXXXXXXX
             AAAAAAAA AAAAAAAA

Кадр 2       00XXXXXX XXXXXXXX XX000000
             AAAAAAAA AAAAAAAA AAAAAAAA

Кадр 3       0000XXXX XXXXXXXX XXXX0000
             AAAAAAAA AAAAAAAA AAAAAAAA

Кадр 4       000000XX XXXXXXXX XXXXXX00
             AAAAAAAA AAAAAAAA AAAAAAAA

Кадр 5                XXXXXXXX XXXXXXXX
                      AAAAAAAA AAAAAAAA

Кадр 6                00XXXXXX XXXXXXXX XX000000
                      AAAAAAAA AAAAAAAA AAAAAAAA

Кадр 7 = Кадру 1      0000XXXX XXXXXXXX XXXX0000
                      AAAAAAAA AAAAAAAA AAAAAAAA
```

Рис. 18 Организация движения шаблона и цветовых атрибутов в программе **Everyone's a Wally.**

Как видите, здесь нет полной цикличности в смене кадров. Так, кадр 1 равен кадру 7 только с точки зрения шаблона. С точки зрения атрибутов они различаются. Если бы нашу последовательность продолжить, то оказалось бы, что полное соответствие наступит только на кадре 13. Такое решение программистов обеспечило довольно неплохую гладкость анимации не только шаблона, но и атрибутов при достаточно экономном расходе памяти.

Конечно, существуют и другие приемы. Внимательно смотрите на движение спрайтов в Ваших любимых играх и Вы сможете узнать много интересного и, возможно, почерпнете что-то и для себя.

2.2 Анимация заднего плана. Окна.

Для обеспечения самой плавной и гладкой анимации требуется еще более тонкое управление движением графики на экране. Самое же плавное перемещение можно получить скроллингом экрана или части экрана на 1 пиксел. Здесь мы рассмотрим две процедуры, обеспечивающие скроллинг по горизонтали и вертикали.

2.2.1. Горизонтальный скроллинг экрана.

Формат процедуры для выполнения горизонтального скроллинга экрана - следующий: **FN l(l,d)**.

l - length - параметр, указывающий на какую величину (в пикселах) следует исполнить скроллинг.

d - direction - параметр, определяющий направление перемещения.
d=1 - скроллинг слева направо
d=0 - скроллинг справа налево.

Листинг 19.
Загрузчик машинного кода процедуры горизонтального скроллинга.

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=51500: LET long=85: LET check=7863: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода
```

```
110 DATA 243,042,011,092,017
120 DATA 004,000,025,070,030
130 DATA 008,025,126,254,000
140 DATA 032,034,014,191,197
150 DATA 033,000,064,229,006
160 DATA 032,167,203,030,035
170 DATA 016,251,225,048,004
180 DATA 062,128,182,119,017
190 DATA 032,000,025,013,032
200 DATA 233,193,016,224,251

210 DATA 201,014,191,197,033
220 DATA 031,064,229,006,032
230 DATA 167,203,022,043,016
240 DATA 251,225,048,004,062
250 DATA 001.182,119,017,032
260 DATA 000,025,013,032,233
270 DATA 193,016,224,251,201
```

Листинг 20
Процедура, управляющая
горизонтальным скроллингом

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C92C  F3              DI                 ;Запрет прерываний
C92D  2A0B5C          LD   HL,(#5C0B)    ;Указание на адрес, в ко-
                                         ;тором располагаются па-
                                         ;раметры функции пользо-
                                         ;вателя FN 1().
C930  110400          LD   DE,#0004      ;Вычисление адреса пер-
C933  19              ADD  HL,DE         ;вого параметра.
C934  46              LD   B,(HL)        ;Прием параметра 1.
C935  1E08            LD   E,#08         ;Переход ко второму
C937  19              ADD  HL,DE         ;параметру.
 938  7E              LD   A,(HL)        ;Прием параметра d.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C939  FE00            CP   #00         ;Определяем направление
                                       ;скроллинга.
C93B  2022            JR   NZ,LEFT     ;Если влево, то
                                       ;переход.
C93D  0EBF    DIST    LD   C,#BF       ;#BF=192 DEC - количест-
                                       ;во горизонтальных линий
                                       ;экрана.
C93F  C5              PUSH BC          ;Сохранение на стеке
C940  210040          LD   HL,#4000    ;Адрес начала экранной
                                       ;области.
C943  E5      VERT    PUSH HL          ;Сохранили на стеке.
C944  0620            LD   B,#20       ;32 байта - длина од-
                                       ;ной линии.
C946  A7              AND  A           ;Сброс флага переноса.
C947  CB1E    HORIZ   RR   (HL)        ;Вращение вправо пикселов
                                       ;в одной линии знакоместа.
                                       ;Пиксел, вышедший за пре-
                                       ;делы знакоместа отклады-
                                       ;вается во флаг переноса.
C949  23              INC  HL          ;Переход к следующему зна-
                                       ;коместу справа.
C94A  10FB            DJNZ HORIZ       ;Если не вся линия сдвину-
                                       ;та, возврат в начало
                                       ;цикла.
C94C  E1              POP  HL          ;Восстановление адреса
                                       ;самого первого знакоместа
                                       ;в текущей  линии для ре-
                                       ;шения вопроса о "прокру-
                                       ;тке" (когда рисунок ухо-
                                       ;дящий вправо выплывает
                                       ;слева).
C94D  3004            JR   NC,BYPASS   ;Если последний уходящий
                                       ;вправо за пределы экрана
                                       ;пиксел был выключен, то
                                       ;все OK, ничего делать
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

                                       ;не надо, обход на BYPASS
C94F  3E80            LD   A,#80       ;Включение старшего бита.
C951  B6              OR   (HL)        ;Включение левого пиксела
C952  77              LD   (HL),A      ;в первом знакоместе.
C953  112000  BYPASS  LD   DE,#0020    ;Переход к очередной ли-
C956  19              ADD  HL,DE       ;нии экрана.
C957  0D              DEC  C           ;Уменьшили счетчик линий.
C958  20E9            JR NZ,VERT       ;Если не все линии смеще-
                                       ;ны, то возврат.
C95A  C1              POP  BC          ;Проверка на исчерпание
C95B  10E0            DJNZ DIST        ;дистанции перемещения 1.
                                       ;Если не весь путь прой-
                                       ;ден, возврат на DIST.
C95D  FB              EI               ;Включение прерываний и
C95E  C9              RET              ;выход.


C95F  0EBF      LEFT  LD   C,#BF       ;#BF=192 DEC - количест-
                                       ;во горизонтальных линий
                                       ;экрана.
C961  C5              PUSH BC          ;Сохранение на стеке
C962  211F40          LD   HL,#401F    ;Адрес первой линии в
                                       ;последнем знакоместе
                                       ;первой экранной строки.
C965  E5        VERT1 PUSH HL          ;Сохранили на стеке.
C966  0620            LD   B,#20       ;32 байта - длина од-
                                       ;ной линии.
C968  A7              AND  A           ;Сброс флага переноса.
C969  CB16      HORIZ1 RL  (HL)        ;Вращение влево пикселов
                                       ;в одной линии знакоместа.
                                       ;Пиксел, вышедший за пре-
                                       ;делы знакоместа отклады-
                                       ;вается во флаг переноса.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C96B  2B              DEC  HL          ;Переход к соседнему зна-
                                       ;коместу слева.
C96C  10FB            DJNZ HORIZ1      ;Если не вся линия сдвину-
                                       ;та, возврат в начало
                                       ;цикла.
C96E  E1              POP  HL          ;Восстановление адреса
                                       ;последнего знакоместа
                                       ;в текущей линии для ре-
                                       ;шения вопроса о "прокру-
                                       ;тке" (когда рисунок ухо-
                                       ;дящий влево выплывает
                                       ;справа).
C96F  3004            JR   NC,PASS-1   ;Если последний уходящий
                                       ;влево за пределы экрана
                                       ;пиксел был выключен, то
                                       ;все OK, ничего делать
                                       ;не надо, обход на PASS-1
C971  3E01            LD   A,#01       ;Включение младшего бита.
C973  B6              OR   (HL)        ;Включение правого пиксела
C974  77              LD   (HL),A      ;в последнем знакоместе.
C975  112000  PASS-1  LD   DE,#0020    ;Переход к очередной ли-
C978  19              ADD  HL,DE       ;нии экрана.
C979  0D              DEC  C           ;Уменьшили счетчик линий.
C97A  20E9            JR   NZ,VERT1    ;Если не все линии смеще-
                                       ;ны, то возврат.
C97C  C1              POP  BC          ;Проверка на исчерпание
C97D  10E0            DJNZ LEFT        ;дистанции перемещения 1.
                                       ;Если не весь путь прой-
                                       ;ден, возврат.
C97F  FB              EI               ;Разрешение прерываний.
C980  C9              RET              ;Выход.
```

2.2.2 Вертикальный скроллинг экрана.

Формат процедуры для выполнения вертикального скроллинга экрана - следующий: FN m(l,d).

l - length - параметр, указывающий на какую величину (в пикселах) следует исполнить скроллинг.

d - direction - параметр, определяющий направление перемещения.
d=0 - скроллинг вниз.
d=1 - скроллинг вверх.

Листинг 21
Загрузчик машинного кода процедуры вертикального скроллинга

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=50900: LET long=215: LET check=68798: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 243,042,011,092,017
120 DATA 004,000,025,070,030
130 DATA 008,025,126,230,001
140 DATA 040,091,197,001,032
150 DATA 000,033,000,064,017
160 DATA 162,199,126,018,035
170 DATA 019,016,250,001,175
180 DATA 000,033,000,064,017
190 DATA 000,065,213,229,006
200 DATA 032,026,119,035,019
```

```
210 DATA 016,250,225,209,036
220 DATA 062,007,164,032,010
230 DATA 062,032,133,111,040
240 DATA 004,124,214,008,103
250 DATA 020,062,007,162,032
260 DATA 010,062,032,131,095
270 DATA 040,004,122,214,008
280 DATA 087,013,032,209,006
290 DATA 032,033,162,199,017
300 DATA 160,087,126,018,035

310 DATA 019,016,250,193,016
320 DATA 167,251,201,197,033
330 DATA 160,087,017,162,199
340 DATA 006,032,126,018,035
350 DATA 019,016,250,001,175
360 DATA 000,033,160,086,017
370 DATA 160,087,213,229,006
380 DATA 032,126,018,035,019
390 DATA 016,250,225,209,037
400 DATA 124,230,007,254,007

410 DATA 032,012,125,214,032
420 DATA 111,254,224,040,004
430 DATA 062,008,132,103,021
440 DATA 122,230,007,254,007
450 DATA 032,012,123,214,032
460 DATA 095,254,224,040,004
470 DATA 062,008,130,087,013
480 DATA 032,201,017,000,064
490 DATA 033,162,199,006,032
500 DATA 126,018,035,019,016

510 DATA 250,193,016,160,251
520 DATA 201,000,000,000,000
530 DATA 000,000,000,000,000
```

Листинг 22
Процедура, управляющая вертикальным скроллингом

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C6D4  F3              DI                 ;Запрет прерываний
C6D5  2A0B5C          LD   HL,(#5C0B)    ;Указание на адрес, в ко-
                                         ;тором располагаются па-
                                         ;раметры функции пользо-
                                         ;вателя FN m().
C6D8  110400          LD   DE,#0004      ;Вычисление адреса пер-
C6DB  19              ADD  HL,DE         ;вого параметра.
C6DC  46              LD   B,(HL)        ;Прием параметра 1.
C6DD  1E08            LD   E,#08         ;Переход ко второму
C6DF  19              ADD  HL,DE         ;параметру.
C6E0  7E              LD   A,(HL)        ;Прием параметра d.
C6E1  E601            AND  #01           ;Определяем направление
                                         ;скроллинга.
C6E3  285B            JR   Z,DOWN        ;Переход, если вверх.
C6E5  C5        MAIN  PUSH BC            ;Сохранение на стеке
C6E6  012000          LD   BC,#0020      ;32 байта - длина одной
                                         ;линии экрана.
C6E9  210040          LD   HL,#4000      ;Начало экранной области
                                         ;в оперативной памяти.
C6EC  11A2C7          LD   DE,#C7A2      ;Адрес временного буфера.
C6EF  7E        LINE1 LD   A,(HL)        ;Переброска первой
C6F0  12              LD   (DE),A        ;линии
C6F1  23              INC  HL            ;экрана
C6F2  13              INC  DE            ;во временный
C6F3  10FA            DJNZ LINE1         ;буфер
C6F5  01AF00          LD   BC,#00AF      ;Высота экрана.
C6F8  210040          LD   HL,#4000      ;Адрес начала первой
                                         ;линии экрана.
C6FB  110041          LD   DE,#4100      ;Адрес начала второй
                                         ;линии экрана.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C6FE  D5      AGAIN   PUSH DE          ;Сохранили на
C6FF  E5              PUSH HL          ;стеке.
C700  0620            LD   B,#20       ;Переброска
C702  1A      LINE2   LD   A,(DE)      ;второй
C703  77              LD   (HL),A      ;линии
C704  23              INC  HL          ;экрана
C705  13              INC  DE          ;в
C706  10FA            DJNZ LINE2       ;первую.
C708  E1              POP  HL          ;Восстановление
C709  D1              POP  DE          ;данных со стека.
C70A  24              INC  H           ;Переход к соседней
                                       ;линии снизу.
C70B  3E07            LD   A,#07       ;Если три младших бита
C70D  A4              AND  H           ;регистра H не равны
C70E  200A            JR   NZ,NOSEG    ;нулю, то переход через
                                       ;границу экранного сег-
                                       ;мента не произошел.
C710  3E20            LD   A,#20       ;Если же они равны нулю,
C712  85              ADD  A,L         ;то надо проверить, то
C713  6F              LD   L,A         ;ли нам нужен переход
C714  2804            JR   Z,NOSEG     ;в очередной сегмент,
                                       ;то ли достаточно
                                       ;перейти в очередной
                                       ;ряд из восьми линий.
C716  7C              LD   A,H         ;Переход в очередной
C717  D608            SUB  #08         ;сегмент не нужен -
C719  67              LD   H,A         ;вычитаем 08 из H и
                                       ;остаемся в старом сег-
                                       ;менте, но в следующем
                                       ;ряду.
C71A  14      NOSEG   INC  D           Все то же самое для
C71B  3E07            LD   A,#07       строки, из которой
C71D  A2              AND  D           происходит копирова-
C71E  200A            JR   NZ,NEXT     ние.
C720  3E20            LD   A,#20
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C722  83              ADD  A,E
C723  5F              LD   E,A
C724  2804            JR   Z,NEXT
C726  7A              LD   A,D
C727  D608            SUB  #08
C729  57              LD   D,A

C72A  0D        NEXT  DEC  C           ;Уменьшаем счетчик строк.
C72B  20D1            JR   NZ,AGAIN    ;Если не исчерпан,то
                                       ;возврат
C72D  0620            LD   B,#20       ;Счетчик на 20 байтов.
C72F  21A2C7          LD   HL,#C7A2    ;Адрес временного буфера,
                                       ;в который  отправлялась
                                       ;самая верхняя строка
                                       ;экрана.
C732  11A057          LD   DE,#57A0    ;Адрес начала последней
                                       ;строки экрана.
C735  7E        LOOP  LD   A,(HL)      ;Замена последней
C736  12              LD   (DE),A      ;строки экрана
C737  23              INC  HL          ;на первую путем
C738  13              INC  DE          ;копирования
C739  10FA            DJNZ LOOP        ;через буфер.
C73B  C1              POP  BC          ;Восстановление счетчика
                                       ;дистанции.
C73C  10A7            DJNZ MAIN        ;Если не весь путь
                                       ;пройден, то возврат
                                       ;к началу самого внешне-
                                       ;го цикла.
C73E  FB              EI               ;Разрешение прерываний.
C73F  C9              RET              ;Выход.

C740  C5        DOWN  PUSH BC          ;Сохранили на стеке
                                       ;путь, который осталось
                                       ;пройти.
```

```
Адрес Маш.код Метка    Мнемоника        Комментарий.

C741  21A057          LD   HL,#57A0    ;Адрес последней линии
                                       ;экрана.
C744  11A2C7          LD   DE,#C7A2    ;Адрес временного буфера
                                       ;для сохранения послед-
                                       ;ней линии.
C747  0620            LD   B,#20       ;32 байта - длина одной
                                       ;линии экрана.
C749  7E       LINE3  LD   A,(HL)      ;Переброска
C74A  12              LD   (DE),A      ;последней
C74B  23              INC  HL          ;линии экрана во
C74C  13              INC  DE          ;временный
C74D  10FA            DJNZ LINE3       ;буфер.
C74F  01AF00          LD   BC,#00AF    ;Высота экрана.
C752  21A056          LD   HL,#56A0    ;Адрес предпоследней
                                       ;линии.
C755  11A057          LD   DE,#57A0    ;Адрес последней линии.
C758  D5       AGAIN2 PUSH DE          ;
C759  E5              PUSH HL          ;
C75A  0620            LD   B,#20       ;Переброска
C75C  7E       LINE4  LD   A,(HL)      ;предпоследней
C75D  12              LD   (DE),A      ;линии
C75E  23              INC  HL          ;экрана
C75F  13              INC  DE          ;в последнюю.
C760  10FA            DJNZ LINE4       ;
C762  E1              POP  HL          ;
C763  D1              POP  DE          ;
C764  25              DEC  H           ;Переход к соседней
                                       ;линии сверху.
C765  7C              LD   A,H         ;Если три младших бита
C766  E607            AND  #07         ;регистра H не равны
C768  FE07            CP   #07         ;семи, то переход через
C76A  200C            JR   NZ,NOSEG2   ;границу экранного сег-
                                       ;мента не произошел.
C76C  7D              LD   A,L         ;Если же они равны нулю,
C76D  D620            SUB  #20         ;то надо проверить, то
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C76F  6F              LD   L,A          ;ли нам нужен переход
C770  FEE0            CP   #E0          ;в предыдущий сегмент,
C772  2804            JR   Z,NOSEG2     ;то ли достаточно
                                        ;перейти в предыдущий
                                        ;ряд из восьми линий.
C774  3E08            LD   A,#08        ;Переход в предыдущий
C776  84              ADD  A,H          ;сегмент не нужен -
C777  67              LD   H,A          ;прибавляем 08 к H и
                                        ;остаемся в старом сег-
                                        ;менте, но в предыдущем
                                        ;ряду.
C778  15      NOSEG2  DEC  D            Все то же самое для
C779  7A              LD   A,D          строки, из которой
C77A  E607            AND  #07          происходит копирова-
C77C  FE07            CP   #07          ние.
C77E  200C            JR   NZ,NEXT2
C780  7B              LD   A,E
C781  D620            SUB  #20
C783  5F              LD   E,A
C784  FEE0            CP   #E0
C786  2804            JR   Z,NEXT2
C788  3E08            LD   A,#08
C78A  82              ADD  A,D
C78B  57              LD   D,A
C78C  0D      NEXT2   DEC  C            ;Уменьшаем счетчик
                                        ;строк.
C78D  20C9            JR   NZ,AGAIN2    ;Если не исчерпан, то
                                        ;возврат.
C78F  110040          LD   DE,#4000     ;Начало экранного файла
C792  21A2C7          LD   HL,#C7A2     ;Адрес временного буфера
C795  0620            LD   B,#20        ;Длина одной линии
C797  7E              LD   A,(HL)       ;Замена первой
C798  12              LD   (DE),A       ;строки экрана
C799  23              INC  HL           ;на последнюю путем
C79A  13              INC  DE           ;копирования
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C79B  10FA            DJNZ LOOP1         ;через буфер.
C79D  C1              POP  BC            ;Восстановление счетчика
                                         ;дистанции.
C79E  10A0            DJNZ DOWN          ;Если не весь путь
                                         ;пройден, то возврат
                                         ;к началу внешнего
                                         ;цикла.
C7A0  FB              EI                 ;Разрешение прерываний
C7A1  C9              RET                ;Выход.
C7A2  00000000 BUFFER DEFM 00 00 00 00
C7A6  00000000        DEFM 00 00 00 00
C7AA  00000000        DEFM 00 00 00 00
C7AE  00000000        DEFM 00 00 00 00
C7B2  00000000        DEFM 00 00 00 00
C7B6  00000000        DEFM 00 00 00 00
C7BA  00000000        DEFM 00 00 00 00
C7BE  00000000        DEFM 00 00 00 00
```

2.2.3 Движение спрайта в окне.

В предыдущих параграфах мы ознакомились с тем, как выполняется скроллинг экрана и убедились в том, что его можно использовать для оживления заднего плана. Но двигать весь экран - занятие довольно расточительное как с точки зрения быстродействия, так и в смысле расходуемой памяти. Гораздо практичнее было бы рассмотреть скроллинг в рамках небольшого окна, содержащего спрайт. Полезность такой процедуры очевидна уже тем, что повторяя раз за разом прокрутку спрайта (спрайтов) через окно, Вы можете создать ощущение непрерывной последовательности и динамики движения. Сравните эту ситуацию с тем, что Вы видите из окна быстродвижущегося поезда. С Вашей точки зрения поезд стоит на месте, а мимо окна мелькают деревья, столбы и придорожные постройки. Но полезность заключается не только в этом.

Можно представить систему экранных окон, например такую, как на рис. 19. Спрайт постепенно "выползает" в первом окне, проходит его и "скрывается" за рамой. После непродолжительной задержки "выплывает" во втором окне, проходит его и повторяет то же самое в третьем окне.

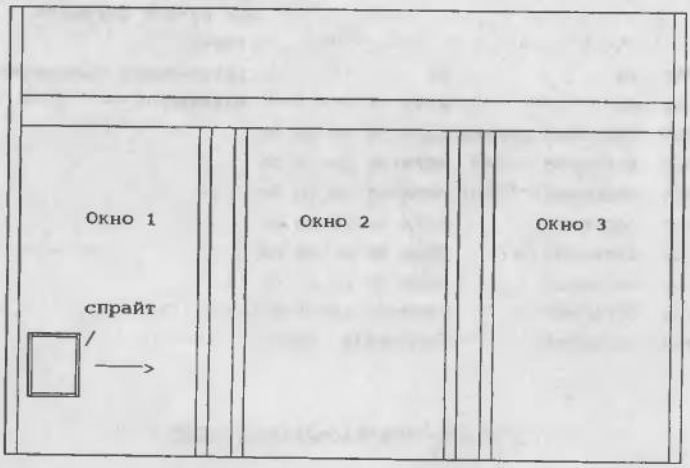


Рис. 19. Скроллинг в системе нескольких окон дает полное ощущение того, что спрайт проходит за объектами переднего плана (за рамой)

Формат процедуры для выполнения скроллинга в окне заданного размера - следующий: **FN n(x,y,l,n,d,r).**

Процедура скроллирует все то, что имеется в пределах окна, заданного параметрами **x,y,l.**

x,y - координаты окна. Они задаются для левого верхнего угла окна. Способ задания координат здесь принят в знакоместах, а не в пикселах.

0<=x<=31; 0<=y<=21

l - ширина окна (в знакоместах 0...31)

n - номер спрайта, скроллируемого в окне.

d - direction - параметр, определяющий направление перемещения.
d=0 - скроллинг вправо.
d=1 - скроллинг влево.

r - флаг повтора: r=0 - есть повтор; r=1 - нет повтора.

Здесь ничего не сказано о высоте окна. Предполагается, что высота его равна высоте спрайта, т.е. 21 пикселу.

Элементарные основы техники скроллирования

Задача скроллирования спрайта в окне - весьма интересна и поучительна, поскольку дает наиболее полное художественное впечатление для наблюдателя. Поэтому мы рассмотрим ее здесь подробнее, чем прочие процедуры.

В основу техники гладкого попиксельного скроллирования шаблона изображения здесь (как и в большинстве подобных случаев) положены команды ротации байтов **RRA** (ротация аккумулятора вправо) и **RLA** (ротация аккумулятора влево). Выбор той или другой зависит от того, куда движется спрайт. Это то, что касается шаблона спрайта, находящегося где-то в оперативной памяти. На экране же скроллинг организован тоже командами ротации байта, но на этот раз **RR (HL)** (ротация вправо) и **RL (HL)** (ротация влево). При этом в регистровой паре HL содержится адрес восьмипик-

сельной линии экрана, принадлежащей текущему знакоместу.

Весь прием основан на том, что прн ротации байта его крайний бит "откладывается" во флаг переноса (флаг С регистра F) и потом "подхватывается" при ротации соседнего байта. Рассмотрим, как это происходит для одной строки спрайта. Она имеет в ширину 24 пиксела, т.е. задана тремя байтами. Флаг переноса (С) в начале операции должен быть обязательно обнулен. Это выполняется какой-либо логической командой, например AND A.

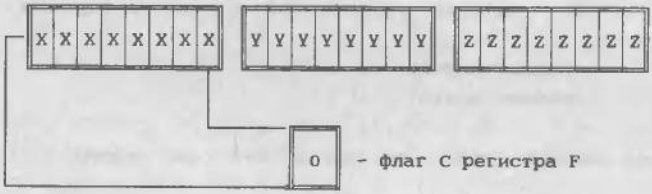


Рис. 20. Исходное состояние.

После первой операции ротации для байта XXXX XXXX его младший (крайний правый) бит перейдет во флаг С, а содержимое флага С перейдет в старший (крайний левый) бит этого байта, т.е. он обнулится. Так происходит ротация (скроллинг) вправо. Если бы нам надо было скроллировать спрайт влево, то мы начинали бы не с байта XXXX XXXX, а с байта ZZZZ ZZZZ.

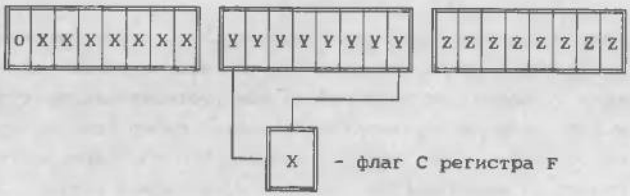


Рис. 21 после первой операции RRA

Следующую операцию ротации вправо проводим для соседнего справа байта (YYYY YYYY). Он "подхватывает" из флага C младший бит от XXXX XXXX и помещает в свой старший бит. И так далее.

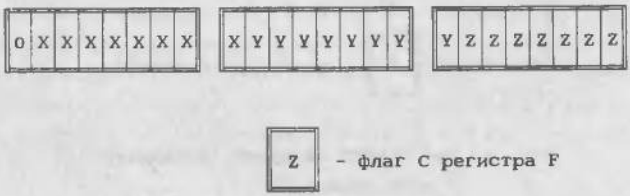


Рис. 22 Результат трех операций RRA

Как видите, в шаблоне спрайта у нас вся линия сместилась на один пиксел вправо, а во флаг C отложился правый бит самого правого байта. Возникает вопрос: что же с ним делать? Может быть, он не нужен?

Напротив, он очень даже нужен, ведь именно ради него и затевалась вся эта "кухня". Его теперь надо поместить на экран. Так возникнет впечатление плавного, попиксельного "выплывания" спрайта в окне.

Если спрайт движется вправо, то для этого мы воспользуемся командой **RR (HL)**, которая вращает биты в том байте, на который указывает адрес, установленный в паре HL. В книге "Элементарная графика" мы говорили о том, что для графических работ принято в этой паре хранит экранные адреса. И немалую роль в таком подходе программистов сыграл и тот факт, что в системе команд процессора Z-80 команды **RR (HL)** и **RL (HL)** имеются, а команды типа *RR (BC), RL (BC), RR (DE)* и *RL (DE)* - отсутствуют.

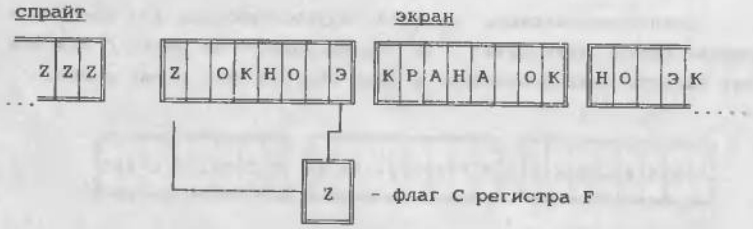


Рис. 23 Так спрайт начинает "выплывать" в окне экрана

Операция скроллирования в окне RR (HL) повторяется столько раз, сколько знакомест имеет окно по ширине.

Как работает процедура.

1. Как обычно, работа процедуры начинается с того, что принимаются из БЕЙСИКа параметры процедуры (x,y,1... и др.).

2. Затем организуются пять вложенных циклов. Как обычно, параметр цикла выставляется в регистре B. Вершинами циклов являются команды

```
PUSH HL
PUSH BC,
```

а окончанием цикла - команды

```
DJNZ adress
POP BC
POP HL
```

Структура этих циклов связана со структурой спрайта и раскладкой экрана.

Спрайт имеет в ширину три знакоместа. В высоту - три ряда.

Каждое знакоместо имеет восемь горизонтальных линий, кроме нижнего ряда - в нем только пять линий, поскольку высота спрайта у нас принята равной 21 пикселу (8+8+5). Каждая линия имеет восемь точек (битов).

Первый цикл (самый внешний) связан с шириной окна и имеет размерность, равную ширине окна в знакоместах + 3, т.к. "прогнать" спрайт надо не только по ширине окна, но и по ширине самого спрайта, которая равна трем знакоместам. Он начинается в адресе #C210 и заканчивается в #C263.

Второй цикл связан с тем, что ширина каждого знакоместа равна восьми пикселам. Его размерность - 8. Он начинается в адресе #C218 и заканчивается в #C25F.

Третий цикл связан тремя рядами знакомест в спрайте (последний ряд неполный) и имеет размерность 3. Он начинается в адресе #C229 и заканчивается в #C25B.

Четвертый цикл связан с тем, что в знакоместе содержатся восемь пиксельных линий (в нижнем ряду - пять). Его размерность - 8 или 5. Он начинается в адресе #C236 и заканчивается в адресе #C24D.

Пятый цикл связан с тем, что в одном ряду спрайта есть три знакоместа. Он имеет размерность, равную трем. Этот цикл вынесен из главной процедуры во вспомогательные подпрограммы **LEFT** и **RIGHT**, которые осуществляют скроллинг соответственно влево или вправо, в зависимости от того, какое направление задано пользователем в параметре **d**. Для движения влево он начинается в адресе #C2C1 и заканчивается в #C2D7. Для движения вправо он начинается в #C2A4 и заканчивается в #C2B0.

Структура организации циклов приведена на рис.24.

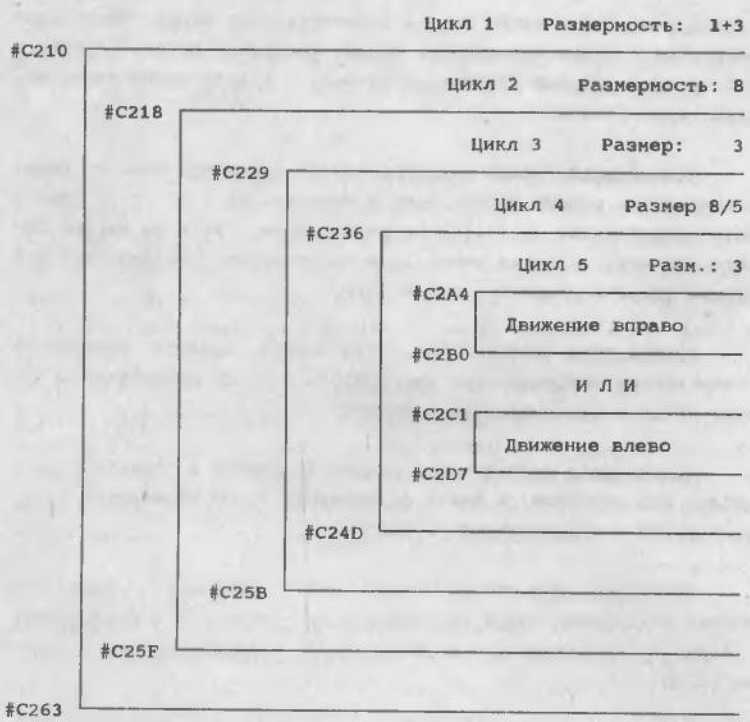


Рис. 24 Структура процедуры.

3. Поскольку нам нужно, чтобы спрайт "выплывал" в окне постепенно, пиксел за пикселом, нам приходится "вращать" байты в спрайте, а это значит, что исходный оригинальный спрайт "портится". Для того, чтобы не уродовать спрайты в таблице спрайтов (начинается с адреса #D548 = 54600 DEC), мы должны создать временный буфер, в котором и будем манипулировать со спрайтом. Этот буфер создаем после нашей процедуры, начиная с адреса #C2E1. Он помечен меткой BUFFER.

Для "перетаскивания" спрайта из таблицы шаблонов в этот буфер применяется специальная процедура **GETSPR** (Get Sprite). Но в ее работе имеются некоторые тонкости, которые следует здесь рассмотреть.

Во-первых, в этой процедуре организован троичный счетчик, который служит для того, чтобы не каждый раз при обращении к этой процедуре происходило "перетаскивание" шаблона спрайта из таблицы спрайтов в буфер, а только один раз из четырех. Так как спрайт имеет в ширину три знакоместа, то новый спрайт принимать в буфер, пока все 3 знакоместа не вышли на экран, не надо.

Этот троичный счетчик организован в программной переменной **COUNT**. При инициализации программы в нем выставляется 1. При обращении к процедуре **GETSPR** наличие там нуля говорит о том, что буфер надо заполнить. Далее происходит декремент счетчика, он обнуляется, после чего в нем выставляется число 3. Теперь при трех ближайших обращениях к процедуре GETSPR буфер не будет переписываться.

При четвертом обращении появляется возможность переписать буфер новым шаблоном спрайта, но будет ли она реализована зависит от того, нужна ли нам повторная "прогонка" этого спрайта, т.е. зависит от того, что задано в параметре r. Если Нам такой повтор не нужен (r=1), то выключается флаговая переменная **FLAG** (изначально она включена), после чего новый спрайт в буфер не помещается, а его место в буфере забивается нулями.

В общих чертах это все сложности, связанные с организацией скроллинга спрайтов в заданном окне.

Листинг 23
Загрузчик машинного кода процедуры скроллирования спрайта в окне

```
10 REM *** Загрузчик машинного кода
20 LET  adr=49600: LET long=290: LET check=81010: LET sum=0
```

```
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 042,011,092,001,004
120 DATA 000,009,086,014,008
130 DATA 009,094,009,126,050
140 DATA 218,194,009,126,050
150 DATA 223,194,009,126,230
160 DATA 001,050,222,194,009
170 DATA 126,230,001,050,221
180 DATA 194,062,001,050,219
190 DATA 194,050,224,194,123
200 DATA 230,024,246,064,103

210 DATA 123,230,007,183,031
220 DATA 031,031,031,130,111
230 DATA 058,222.194,254,000
240 DATA 040,006,058,218,194
250 DATA 133,061,111,058,218
260 DATA 194,060,060,060,071
270 DATA 197,229,205,102,194
280 DATA 225,006,008,229,197
290 DATA 017,225,194,058,222
300 DATA 194,254,000,040,003

310 DATA 017,011,195,006,003
320 DATA 229,197,120,254,001
330 DATA 032,004,006,005,024
340 DATA 002,006,008,197,213
350 DATA 229,058,222,194,254
360 DATA 000,032,005,205,160
370 DATA 194,024,003,205,189
380 DATA 194,225,209,193,036
390 DATA 019,016,231,193,225
```

```
400 DATA 062,032,133,111,048

410 DATA 004,062,008,132,103
420 DATA 016,204,193,225,016
430 DATA 183,193,118,016,171
440 DATA 201,058,224,194,061
450 DATA 050,224,194,192,062
460 DATA 003,050,224,194,058
470 DATA 223,194,017,063,000
480 DATA 071,033,072,213,167
490 DATA 237,082,025,016,253
500 DATA 067,017,225,194,058

510 DATA 219,194,254,000,040
520 DATA 001,126,018,035,019
530 DATA 016,243,058,221,194
540 DATA 254,000,200,062,000
550 DATA 050,219,194,201,167
560 DATA 006,003,213,026,031
570 DATA 018,245,062,021,131
580 DATA 095,048,001,020,241
590 DATA 016,242,209,058,218
600 DATA 194,071,203,030,035

610 DATA 016,251,201,167,213
620 DATA 006,003,026,023,018
630 DATA 245,123,214,021,095
640 DATA 048,001,021,241,016
650 DATA 242,209,058,218,194
660 DATA 071,203,022,043,016
670 DATA 251,201,013,000,000
680 DATA 001,000,010,003,000
```

Листинг 24
Процедура, управляющая
скроллингом спрайта в окне

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C1C0  2A0B5C          LD   HL,(#5C0B)  ;Указание на адрес, в
                                       ;котором располагаются
                                       ;параметры функции поль-
                                       ;зователя FN n().
C1C3  010400          LD   BC,#0004    ;Вычисление адреса пер-
C1C6  09              ADD  HL,BC       ;вого параметра.
C1C7  56              LD   D,(HL)      ;Прием параметра x.
C1C8  0E08            LD   C,#08       ;Переход ко второму
C1CA  09              ADD  HL,BC       ;параметру.
C1CB  5E              LD   E,(HL)      ;Прием параметра y.
C1CC  09              ADD  HL,BC       ;Переход к третьему
                                       ;параметру.
C1CD  7E              LD   A,(HL)      ;Прием параметра l.
C1CE  32DAC2          LD   (WIDTH),A   ;Сохранили ширину окна
                                       ;в программной пере-
                                       ;менной.
C1D1  09              ADD  HL,BC       :Переход к четвертому
                                       ;параметру.
C1D2  7E              LD   A,(HL)      ;Прием параметра n.
C1D3  32DFC2          LD   (NUMBER),A  ;Запомнили номер спрайта
                                       ;в программной перемен-
                                       ;ной,
C1D6  09              ADD  HL,BC       ;Переход к пятому пара-
                                       ;метру.
C1D7  7E              LD   A,(HL)      ;Прием параметра d.
C1D8  E601            AND  #01         ;Гашение ненужных
                                       ;старших битов.
C1DA  32DEC2          LD   (DIRECT),A  ;Запомнили в программной
                                       ;переменной.
C1DD  09              ADD  HL,BC       :Переход к шестому пара-
                                       ;метру.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C1DE  7E              LD   A,(HL)        ;Прием параметра x.
C1DF  E601            AND  #01           ;Оставили младший бит.
C1E1  32DDC2          LD   (REPIT),A     ;Сохранили в переменной.
C1E4  3E01            LD   A,#01         ;
C1E6  32DBC2          LD   (FLAG),A      ;Инициализация флаговой
                                         ;переменной.
C1E9  32E0C2          LD   (COUNT),A     ;Инициализация троичного
                                         ;счетчика.
C1EC  7B              LD   A,E           |
C1ED  E618            AND  #18           |
C1EF  F640            OR   #40           |
C1F1  67              LD   H,A           | Поиск в дисплейном
C1F2  7B              LD   A,E           | файле адреса, соот-
C1F3  E607            AND  #07           | ветствующего коор-
C1F5  B7              OR   A             | динатам, заданным в
C1F6  1F              RRA                | знакоместах.
C1F7  1F              RRA                |
C1F8  1F              RRA                | Адрес выставляется
C1F9  1F              RRA                | в регистровой паре
C1FA  82              ADD  A,D           |      HL
C1FB  6F              LD   L,A           |
C1FC  3ADEC2          LD   A,(DIREC)     ;Проверяем, в каком на-
C1FF  FE00            CP   #00           ;правлении идет скроллинг
C201  2806            JR   Z,LBL-1       ;Если ноль, то вправо.
                                         ;Иначе - влево.
C203  3ADAC2          LD   A,(WIDTH)     ;Ширина окна.
C206  85              ADD  A,L           ;Получили x+1
C207  3D              DEC  A             ;Получили x+1-1,
C208  6F              LD   L,A           ;что равно координате x
                                         ;последнего (правого)
                                         ;столбца окна. Если
                                         ;скроллинг идет влево,
                                         ;то эта координата нужна
                                         ;как исходная.
C209  3ADAC2  LBL-1   LD   A,(WIDTH)     ;Ширина окна.
```

```
Адрес Маш.код Метка    Мнемоника        Комментарий.

C20C  3C              INC  A            ;
C20D  3C              INC  A            ;
C20E  3C              INC  A            ;Рассчитали 1+3 - длину
                                        ;пути, который должен
                                        ;пройти спрайт.
C20F  47              LD   B,A          ;Организовали счетчик
                                        ;первого, самого внешнего
                                        ;цикла. (Счетчик пути).
C210  C5       LOOP-1 PUSH BC           ;Сохранили его на стеке.
C211  E5              PUSH HL           ;Запомнили на стеке
                                        ;адрес в экранной па-
                                        ;мяти, соответствую-
                                        ;щий правому или левому
                                        ;верхнему углу окна.
C212  CD66C2          CALL GETSPR       ;Прием спрайта в буфер.
C215  E1              POP  HL           ;Экранный адрес
C216  0608            LD   B,#08        ;Ширина каждого знако-
                                        ;места равна 8 пикселам.
C218  E5       LOOP-2 PUSH HL           ;Сохранили на стеке перед
C219  C5              PUSH BC           ;организацией второго
                                        ;цикла.
C21A  11E1C2          LD   DE,BUFFER    ;Адрес буфера. Там на-
                                        ;ходятся данные о шаб-
                                        ;лоне левого столбца
                                        ;спрайта.
C21D  3ADEC2          LD   A,(DIREC)    ;Направление скроллинга.
C220  FE00            CP   #00          ;Проверка направления.
C222  2803            JR   Z,LBL2       ;Если движение вправо,
                                        ;то обход следующей опе-
                                        ;рации.
C224  110BC3          LD   DE,#C30B     ;Если же движение идет
                                        ;влево, то вместо данных
                                        ;о левом столбце спрайта
                                        ;шаблона спрайта нам нуж-
                                        ;ны данные о его правом
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

                                       ;столбце. Они смещены на
                                       ;#2A байтов (42 байта) от
                                       ;начала буфера -
                                       ;#C2E1 + #2A = #C30B.
C227  0603    LBL-2   LD    B,#03      ;Цикл по трем вертикаль-
                                       ;ным рядам спрайта.
C229  E5      LOOP-3  PUSH  HL         ;Сохранение регистров на
C22A  C5              PUSH  BC         ;стеке перед организацией
                                       ;третьего цикла.
C22B  78              LD    A,B        ;Проверка, с каким рядом
C22C  FE01            CP    #01        ;мы имеем дело.
C22E  2004            JR    NZ,LBL-3   :Если не с последним,
                                       ;то обход следующей
                                       ;операции.
C230  0605            LD    B,#05      ;В последнем ряду спрай-
                                       ;та только пять линий,
                                       ;т.к. высота спрайта
                                       :равна 21 пикселу.
C232  1802            JR    LOOP-4     ;Обход.
C234  0608    LBL-3   LD    B,#08      ;В двух верхних рядах
                                       ;спрайта по восемь ли-
C236  C5      LOOP-4  PUSH  BC         ;ний. Это организация
C237  D5              PUSH  DE         ;счетчика четвертого
C238  E5              PUSH  HL         ;цикла.
C239  3ADEC2          LD    A,(DIREC)  :Направление скроллинга.
C23C  FE00            CP    #00
C23E  2005            JR    NZ,LBL-4   ;Если влево. то обход.
C240  CDA0C2          CALL  RIGHT      ;Вызов процедуры скрол-
                                       ;линга вправо.
C243  1803            JR    LBL-5      ;Обход.
C245  CDBDC2  LBL-4   CALL  LEFT       ;Вызов процедуры скрол-
                                       ;линга влево.
C248  E1      LBL-5   POP   HL         :Восстановление регистров
C249  D1              POP   DE         ;со стека перед оконча-
C24A  C1              POP   BC         ;нием цикла.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C24B  24              INC  H           ;Переход к соседней снизу
                                       ;линии знакоместа на эк-
                                       ;ране.
C24C  13              INC  DE          ;Переход к соседней снизу
                                       ;линии в шаблоне спрайта.
C24D  10E7            DJNZ LOOP-4      ;Конец четвертого цикла.
C24F  C1              POP  BC          ;Восстановление регистров
C250  E1              POP  HL          ;со стека перед оконча-
                                       ;нием цикла.
C251  3E20            LD   A,#20       ;Переход к соседнему сни-
C253  85              ADD  A,L         ;зу знакоместу на экране.
C254  6F              LD   L,A
C255  3004            JR   NC,LBL-6    ;Если не произошло пере-
                                       ;полнение регистра L, то
                                       ;обход следующих опера-
                                       ;ций.
C257  3E08            LD   A,#08       ;Если же такое перепол-
C259  84              ADD  A,H         ;нение произошло, то это
C25A  67              LD   H,A         ;означает, что мы "пере-
                                       ;валили" за границу эк-
                                       ;ранного сегмента и в
                                       ;этом случае регистр H
                                       ;увеличивается скачком
                                       ;на 8 пунктов.
C25B  10CC    LBL-6   DJNZ LOOP-3      ;Конец третьего цикла.
C25D  C1              POP  BC          ;Восстановление регистров
C25E  E1              POP  HL          ;со стека перед оконча-
                                       ;нием цикла.
C25F  10B7            DJNZ LOOP-2      ;Конец второго цикла.
C261  C1              POP  BC          ;Восстановление счетчика
                                       ;первого (внешнего) цик-
                                       ;ла.
C262  76              HALT             ;Краткая пауза.
C263  10AB            DJNZ LOOP-1      ;Конец внешнего цикла.
C265  C9              RET              ;Конец работы.
```

Процедура **GETSPR**.

Осуществляет переброску шаблона спрайта во временный буфер.

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C266  3AE0C2  GETSPR  LD    A,(COUNT)   ;Приняли данные из
                                        ;троичного счетчика.
C269  3D              DEC   A           ;Сняли с него единицу.
C26A  32E0C2          LD    (COUNT),A   ;И снова его запомнили.
C26D  C0              RET   NZ          ;Если не ноль, значит
                                        ;наш спрайт еще не весь
                                        ;"выехал" на экран и при-
                                        ;нимать спрайт в буфер
                                        ;пока не надо.
C26E  3E03            LD    A,#03       ;Выставляем число 3 в
C270  32E0C2          LD    (COUNT),A   ;троичном счетчике,
C273  3ADFC2          LD    A,(NUMBER)  ;Номер спрайта.
C276  113F00          LD    DE,#003F    ;Длина одного шаблона
                                        ;спрайта.
C279  47              LD    B,A         ;Создали счетчик.
C27A  2148D5          LD    HL,#D548    ;Адрес начала таблицы,
                                        ;в которой размещаются
                                        ;спрайты (54600).
C27D  A7              AND   A           ;Сброс флага переноса,
                                        ;чтобы он не влиял на
                                        ;результат операции SBC.
C27E  ED52            SBC   HL,DE       ;Выставление начального
C280  19      NEXT    ADD   HL,DE       ;адреса для первого
                                        ;спрайта.
C281  10FD            DJNZ  NEXT        ;Возврат за отысканием
                                        ;адреса очередного
                                        ;спрайта.
C283  43              LD    B,E         ;Длина шаблона спрайта.
C284  11E1C2          LD    DE,BUFFER   ;Приняли адрес временного
                                        ;буфера для размещения
                                        ;спрайта.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C287  3ADBC2  AGAIN    LD A,(FLAG)      ;Проверка флага, связан-
                                         ;ного с необходимостью
                                         ;повторного "прогона"
                                         ;спрайта,
C28A  FE00              CP   #00         ;Проверка на ноль.
C28C  2801              JR   Z,BYPASS    ;Если ноль, то спрайт
                                         ;принимать не надо, идем
                                         ;в обход и обнуляем бу-
                                         ;фер.
C28E  7E                LD   A,(HL)      ;Байт шаблона спрайта.
C28F  12        BYPASS  LD   (DE),A      ;Запомнили его в буфере.
C290  23                INC  HL          ;Переход к очередному
                                         ;байту шаблона.
C291  13                INC DE           ;Переход к очередному
                                         ;байту буфера.
C292  10F3              DJNZ AGAIN       ;Если не все байты шабло-
                                         ;на скопированы, то воз-
                                         ;врат.
C294  3ADDC2            LD   A,(REPIT)   ;Проверка необходимости
                                         ;повтора.
C297  FE00              CP   #00         ;Если есть необходимость,
C299  C8                RET  Z           ;то выход, в переменной
                                         ;FLAG остается единица.
C29A  3E00              LD   A,#00       ;В противном случае об-
C29C  32DBC2            LD   (FLAG),A    ;нуляем переменную FLAG.
C29F  C9                RET              ;Возврат в вызывающую
                                         ;процедуру.
```

Процедура **RIGHT**.

Осуществляет "прокрутку" окна вправо.

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C2A0  A7      RIGHT   AND  A           ;Сброс флага переноса.
C2A1  0603            LD   B,#03       ;Начало внутреннего,
                                       ;пятого цикла. Установка
                                       ;параметра цикла.
C2A3  D5              PUSH DE          ;Сохранение регистра
                                       ;на время работы подпро-
                                       ;граммы.
C2A4  1A      LOOP-5A LD   A,(DE)      ;Взяли в аккумулятор
                                       ;байт из левого верти-
                                       ;кального столбца в шаб-
                                       ;лоне спрайта.
C2A5  1F              RRA              ;Ротация вправо.
C2A6  12              LD   (DE),A      ;Заменили байт в шабло-
                                       ;не.
C2A7  F5              PUSH AF          ;Сохранили флаговый ре-
                                       ;гистр на стеке. Важен
                                       ;флаг переноса.
C2A8  3E15            LD   A,#15       ;Сдвиг на 21 байт для
                                       ;перехода к соседнему
C2AA  83              ADD  A,E         ;справа столбцу.
C2AB  5F              LD   E,A         ;Указатель в буфере
                                       ;спрайта переместили.
C2AC  3001            JR   NC,LBL-7    :Если не возникло пе-
                                       ;реполнения, то обход.
C2AE  14              INC  D           ;Иначе наращиваем
                                       ;старший регистр.
C2AF  F1      LBL-7   POP  AF          ;Восстановили флаг C.
C2B0  10F2            DJNZ LOOP-5A     ;Конец пятого цикла.
C2B2  D1              POP  DE          ;Восстановили указатель
                                       ;на буфер.
C2B3  3ADAC2          LD   A,(WIDTH)   ;Ширина окна.
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

C2B6  47              LD   B,A           ;Ввели ее в качестве
                                         ;счетчика.
C2B7  CB1E    LOOP-6A RR   (HL)          ;Ротация вправо байтов
                                         ;в окне.
C2B9  23              INC  HL            ;Переход в окне к сосед-
                                         ;нему знакоместу справа.
C2BA  10FB            DJNZ LOOP-6A       ;Прокручиваем линию по
                                         ;всей ширине окна.
C2BC  C9              RET                ;Возврат в вызывающую
                                         ;процедуру.
```

Процедура **LEFT**.

Осуществляет "прокрутку" окна влево.

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

C2BD  A7      LEFT    AND  A             ;Сброс флага переноса.
C2BE  D5              PUSH DE            ;Сохранение регистра
                                         ;на время работы подпро-
                                         ;граммы.
C2BF  0603            LD   B,#03         ;Начало внутреннего,
                                         :пятого цикла. Установка
                                         ;параметра цикла.
C2C1  1A      LOOP-5B LD   A,(DE)        ;Взяли в аккумулятор
                                         ;байт из правого верти-
                                         ;кального столбца в шаб-
                                         ;лоне спрайта.
C2C2  17              RLA                ;Ротация влево.
C2C3  12              LD   (DE),A        ;Заменили байт в шабло-
                                         ;не.
C2C4  F5              PUSH AF            ;Сохранили флаговый ре-
                                         ;гистр на стеке. Важен
                                         ;флаг переноса.
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

C2C5  7B              LD   A,E           ;Сдвиг назад на 21 байт
C2C6  D615            SUB  #15           ;для перехода к соседнему
C2C8  5F              LD   E,A           ;слева столбцу.
C2C9  3001            JR   NC,LBL-8      ;Если не возникло двоич-
                                         ;ного займа, то обход.
C2CB  15              DEC  D             ;Уменьшаем старший байт.
C2CC  F1       LBL-8  POP  AF            ;Восстановили флаг C.
C2CD  10F2            DJNZ LOOP-5B       ;Конец внутреннего,
                                         ;пятого цикла.
C2CF  D1              POP  DE            ;Восстановили указатель
                                         ;на буфер.
C2D0  3ADAC2          LD   A,(WIDTH)     ;Ширина окна.
C2D3  47              LD   B,A           ;Ввели ее в качестве
                                         ;счетчика.
C2D4  CB16     LOOP-6B RL   (HL)          ;Ротация влево байтов
                                         ;в окне.
C2D6  2B              DEC  HL            ;Переход к соседнему
                                         ;знакоместу слева.
C2D7  10FB            DJNZ LOOP-6B       ;Прогон на всю ширину
                                         ;окна.
C2D9  C9              RET                ;Возврат в вызывающую
                                         ;процедуру.


C2DA  0D       WIDTH  DEFB 0D            ;Ширина окна.
C2DB  00       FLAG   DEFB 00            ;Флаг, указывающий на
                                         ;необходимость повтора
                                         ;"прогона" спрайта.
C2DC  00              NOP                ;
C2DD  00       REPIT  DEFB 00            ;Параметр повтора.
C2DE  00       DIRECT DEFB 00            ;Направление скроллинга.
C2DF  0A       NUMBER DEFB 0A            ;Номер спрайта.
C2E0  03       COUNT  DEFB 03            ;Троичный счетчик для
                                         ;процедуры GETSPR.
C2E1  00       BUFFER DEFB 00            ;Временный буфер для
                                         ;хранения шаблона
                                         ;спрайта.
....  00       Здесь и далее расположены временные данные
                о шаблоне спрайта.
```

2.2.4 Управление окном от прерываний 2-го рода.

Выше, в разделе *2.1.5* мы рассмотрели процедуру **FN h()**, предназначенную для управления спрайтами от клавиатуры с использованием прерываний второго рода. Как мы только что убедились, "оконная" технология, связанная со скроллингом позволяет в некоторых случаях получать более интересные результаты, чем технология работы со спрайтами. Было бы интересно объединить процедуру, управления от клавиатуры с использованием прерываний второго рода с процедурой скроллирования спрайта в окне, чем мы здесь и займемся на примере процедуры **FN o(s,x,y,l,n,d,r)**.

Эта процедура уже может стать основным приводом (engine) Вашей будущей самостоятельной игры типа Space Raders, в которой Вы управляете движением своей пушки "влево-вправо" и расстреливаете проплывающих над Вами инопланетян. При этом движением (скроллингом) "инопланетян" как бы управляет от прерываний 2-го рода процедура **FN o()**, а движением спрайта "пушки" - процедура **FN h()**.

Параметры процедуры.

s - параметр старт/стоп. Поскольку процедура работает от прерываний 2-го рода, то ее надо не только уметь запустить, но и уметь остановить. Средствами БЕЙСИКа она не остановится. Для этого и введен дополнительный параметр **s**. При задании **s=0** процедура стартует, а при **s=1** - останавливается.

x,y - координаты окна. Они задаются для левого верхнего угла окна. Способ задания координат здесь принят в знакомстах.

```
0<=x<=31;          0<=y<=21
```

l - ширина окна (в знакоместах 0...31)

n - номер спрайта, скроллируемого в окне.

d - direction - параметр, определяющий направление перемещения.
d=0 - скроллинг вправо.
d=1 - скроллинг влево.

r - флаг повтора: r=0 - есть повтор; r=1 - нет повтора.

Листинг 25

Загрузчик машинного кода процедуры скроллирования спрайта в окне с использованием IM2.

```
 10 REM *** Загрузчик машинного кода
 20 LET  adr=49200: LET long=315: LET check=81594: LET sum=0
 30 FOR k=0 TO long-1: READ a
 40 POKE (adr+k),a: LET sum=sum+a
 50 NEXT k
 60 IF sum<>check THEN PRINT "??": STOP
100 REM *** Данные для машинного кода

110 DATA 243,042,011,092,001
120 DATA 004,000,009,126,254
130 DATA 000,032,009,017,252
140 DATA 207,237,083,250,207
150 DATA 251,201,017,139,192
160 DATA 237,083,250,207,014
170 DATA 008,009,086,009,094
180 DATA 009,126,050,100,193
190 DATA 009,126,050,108,193
200 DATA 009,126,230,001,050
```

```
210 DATA 107,193,009,126,230
220 DATA 001,050,106,193,062
230 DATA 001,050,104,193,050
240 DATA 109,193,050,101,193
250 DATA 123,230,024,246,064
260 DATA 103,123,230,007,183
270 DATA 031,031,031,031,130
280 DATA 111,034,102,193,251
290 DATA 201,058,107,193,254
300 DATA 000,040,006,058,100

310 DATA 193,133,061,111,058
320 DATA 101,193,071,016,005
330 DATA 205,240,192,006,008
340 DATA 120,050,101,193,017
350 DATA 110,193,042,102,193
360 DATA 058,107,193,254,000
370 DATA 040,003,017,152,193
380 DATA 006,003,229,197,120
390 DATA 254,001,032,004,006
400 DATA 005,024,002,006,008

410 DATA 197,213,229,058,107
420 DATA 193,254,000,032,005
430 DATA 205,042,193,024,003
440 DATA 205,071,193,225,209
450 DATA 193,036,019,016,231
460 DATA 193,225,062,032,133
470 DATA 111,048,004,062,008
480 DATA 132,103,016,204,195
490 DATA 252,207,058,109,193
500 DATA 061,050,109,193,192

510 DATA 062,003,050,109,193
520 DATA 058,108,193,017,063
530 DATA 000,071,033,072,213
540 DATA 167,237,082,025,016
```

```
550 DATA 253,067,017,110,193
560 DATA 058,104,193,254,000
570 DATA 040,001,126,018,035
580 DATA 019,016,243,058,106
590 DATA 193,254,000,200,062
600 DATA 000,050,104,193,201

610 DATA 167,006,003,213,026
620 DATA 031,018,245,062,021
630 DATA 131,095,048,001,020
640 DATA 241,016,242,209,058
650 DATA 100,193,071,203,030
660 DATA 035,016,251,201,167
670 DATA 213,006,003,026,023
680 DATA 018,245,123,214,021
690 DATA 095,048,001,021,241
700 DATA 016,242,209,058,100

710 DATA 193,071,203,022,043
720 DATA 016,251,201,005,004
730 DATA 000,000,000,000,000
```

Листинг 26

Дисассемблер процедуры, управляющей скроллингом спрайта в окне при использовании прерываний 2-го рода.

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C030  F3              DI               ;Отключение прерываний.
C031  2A0B5C          LD   HL,(#5C0B)  ;Указание на адрес, в
                                       ;котором располагаются
                                       ;параметры функции поль-
                                       ;зователя FN n().
C034  010400          LD   BC,#0004    ;Вычисление адреса пер-
C037  09              ADD  HL,BC       ;вого параметра.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C038  7E              LD    A,(HL)       ;Проверка параметра
C039  FE00            CP    #00          ;s.
C03B  2009            JR    NZ,START     ;Если не 0, то стартуем
C03D  11FCCF          LD    DE,#CFFC     ;Иначе надо прекратить
C040  ED53FACF        LD    (#CFFA),DE   ;обработку прерываний
                                         ;2-го рода. Оформляем
                                         ;выход из процедуры
                                         ;FN h() - см.с. 56.
C044  FB              EI                 ;Разрешение прерываний.
C045  C9              RET                ;Возврат.

C046  118BC0  START   LD    DE,IM2       ;Включение процедуры.
C049  ED53FACF        LD    (#CFFA),DE   ;В процедуре FN h( )
                                         ;выставляется новый
                                         ;адрес для обработки
                                         ;прерываний 2-го рода.
C04D  0E08            LD    C,#08        ;Переход ко 2-му пара-
C04F  09              ADD   HL,BC        ;метру.
C050  56              LD    D,(HL)       ;Прием параметра x.
C051  09              ADD   HL,BC        ;Переход к 3-му пара-
                                         ;метру.
C052  5E              LD    E,(HL)       ;Прием параметра y.
C053  09              ADD   HL,BC        ;Переход к 4-му пара-
                                         ;метру.
C054  7E              LD    A,(HL)       ;Прием параметра l.
C055  3264C1          LD    (WIDTH),A    ;Сохранили ширину окна
                                         ;в программной перемен-
                                         ;ной.
C058  09              ADD   HL,BC        ;Переход к 5-му пара-
                                         ;метру.
C059  7E              LD    A,(HL)       ;Прием параметра n.
C05A  326CC1          LD    (NUMBER),A   ;Сохранили в переменной.
C05D  09              ADD   HL,BC        ;Переход к 6-му пара-
                                         ;метру.
C05E  7E              LD    A,(HL)       ;Прием параметра d.
```

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C05F  E601            AND  #01           ;Маскирование старших
                                         ;битов.
C061  326BC1          LD   (DIREC),A     ;Сохранили в переменной.
C064  09              ADD  HL,BC         ;Переход к 7-му пара-
                                         ;метру.
C065  7E              LD   A,(HL)        ;Прием параметра r.
C066  E601            AND  #01           ;Маскирование старших
                                         ;битов.
C068  326AC1          LD   (REPIT),A     ;Сохранили в переменной.
C06B  3E01            LD   A,#01
C06D  3268C1          LD   (FLAG),A      ;инициализация флаговой
                                         ;переменной.
C070  326DC1          LD   (COUNT),A     ;Инициализация троично-
                                         ;го счетчика.
C073  3265C1          LD   (COUNT8),A    ;Инициализация восьмерич-
                                         ;ного счетчика.
C076  7B              LD   A,E           |
C077  E618            AND  #18           |
C079  F640            OR   #40           |
C07B  67              LD   H,A           | Поиск в дисплейном
C07C  7B              LD   A,E           | файле адреса, соот-
C07D  E607            AND  #07           | ветствующего коор-
C07F  B7              OR   A             | динатам, заданным в
C080  1F              RRA                | знакоместах.
C081  1F              RRA                |
C082  1F              RRA                | Адрес выставляется
C083  1F              RRA                | в регистровой паре
C084  82              ADD  A,D           |      HL
C085  6F              LD   L,A           |

C086  2266C1          LD   (SCRAD),HL    ;Сохранили его в про-
                                         ;граммной переменной.
C089  FB              EI                 ;Разрешение прерываний.
C08A  C9              RET                ;Возврат.
```

Резидентная процедура, обрабатывающая прерывания 2-го рода. Мы попадаем сюда из процедуры Fn h(), см с.56, поскольку операцией #C049 там был выставлен именно этот адрес.

```
Адрес Маш.код Метка   Мнемоника          Комментарий.

C08B  3A6BC1  IM2     LD   A,(DIREC)     ;Направление скроллинга.
C08E  FE00            CP   #00           ;Проверка направления.
C090  2806            JR   Z,LBL-1       ;Если вправо, то переход
C092  3A64C1          LD   A,(WIDTH)     ;Ширина окна.
C095  85              ADD  A,L           ;Получили x+1
C096  3D              DEC  A             ;Получили x+1-1.
C097  6F              LD   L,A           ;что равно координате x
                                         ;последнего (правого)
                                         ;столбца окна. Если
                                         ;скроллинг идет влево,
                                         ;то эта координата нужна
                                         ;как исходная.
C098  3A65C1  LBL-1   LD   A,(COUNT8)    ;Проверка восьмиразряд-
                                         ;ного счетчика,
C09B  47              LD   B,A           ;Выставляется как пара-
                                         ;метр цикла.
C09C  1005            DJNZ BRANCH        ;Элегантный пример того,
                                         ;как инструкция DJNZ ис-
                                         ;пользуется не для орга-
                                         ;низации цикла, а для
                                         ;организации периодичес-
                                         ;кого ветвления.
C09E  CDF0C0          CALL GETSPR        ;Прием спрайта в буфер.
C0A1  0608            LD   B,#08         ;Установка значения в
C0A3  78      BRANCH  LD   A,B           ;восьмиразрядном счет-
C0A4  3265C1          LD   (COUNT8),A    ;чике.
C0A7  116EC1          LD   DE,BUFFER     ;Адрес в буфере, указы-
                                         ;вающий на начало левого
                                         ;столбца спрайта.
C0AA  2A66C1          LD   HL,(SCRADD)   ;Экранный адрес.
C0AD  3A6BC1          LD   A,(DIREC)     ;Направление скроллинга.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C0B0  FE00            CP   #00         ;Вправо?
C0B2  2803            JR   Z,LBL-2     ;Если движение вправо,
                                       ;то обход следующей опе-
                                       ;рации.
C0B4  1198C1          LD   DE,#C198    ;Адрес, с которого в бу-
                                       ;фере начинается правый
                                       ;столбец спрайта.
C0B7  0603    LBL-2   LD   B,#03       ;Цикл по трем вертикаль-
                                       ;ным рядам спрайта.
C0B9  E5      LOOP-1  PUSH HL          ;Сохранение регистров на
C0BA  C5              PUSH BC          ;стеке перед организацией
                                       ;цикла.
C0BB  78              LD   A,B         ;Проверка, с каким рядом
C0BC  FE01            CP   #01         ;мы имеем дело.
C0BE  2004            JR   NZ,LBL-3    ;Если не с последним,
                                       ;то обход следующей
                                       ;операции.
C0C0  0605            LD   B,#05       ;В последнем ряду спрай-
                                       ;та только пять линий,
C0C2  1802            JR   LOOP-2      ;Обход.
C0C4  0608    LBL-3   LD   B,#08       ;В двух верхних рядах
                                       ;спрайта по восемь ли-
C0C6  C5      LOOP-2  PUSH BC          ;ний. Это организация
C0C7  D5              PUSH DE          ;счетчика очередного
C0C8  E5              PUSH HL          ;цикла.
C0C9  3A6BC1          LD   A,(DIREC)   ;Направление скроллинга.
C0CC  FE00            CP   #00
C0CE  2005            JR   NZ,LBL-4    ;Если влево, то обход.
C0D0  CD2AC1          CALL RIGHT       ;Вызов процедуры
                                       ;скроллинга вправо.
C0D3  1803            JR   LBL-5       ;Обход.
C0D5  CD47C1  LBL-4   CALL LEFT        ;Вызов процедуры
                                       ;скроллинга влево.
C0D8  E1      LBL-5   POP  HL          ;Восстановление регистров
C0D9  D1              POP  DE          :со стека перед оконча-
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C0DA  C1              POP  BC          ;нием цикла.
C0DB  24              INC  H           ;Переход к соседней снизу
                                       ;линии знакоместа на эк-
                                       ;ране.
C0DC  13              INC  DE          ;Переход к соседней снизу
                                       ;линии в шаблоне спрайта.
C0DD  10E7            DJNZ LOOP-2      ;Конец цикла.
C0DF  C1              POP  BC          ;Восстановление регистров
C0E0  E1              POP  HL          ;со стека перед оконча-
                                       ;нием цикла.
C0E1  3E20            LD   A,#20       ;Переход к соседнему сни-
C0E3  85              ADD  A,L         ;зу знакоместу на экране.
C0E4  6F              LD   L,A
C0E5  3004            JR   NC,LBL-6    ;Если не произошло пере-
                                       ;полнение регистра L, то
                                       ;обход следующих опера-
                                       ;ций.
C0E7  3E08            LD   A,#08       ;Если же такое переполь-
C0E9  84              ADD  A,H         ;нение произошло, то это
C0EA  67              LD   H,A         ;означает, что мы вошли
                                       ;в новый экранный сегмент
                                       ;и регистр  H увеличива-
                                       ;ется на 8 пунктов.
C0EB  10CC    LBL-6   DJNZ LOOP-1      ;Конец цикла.
C0ED  C3FCCF          JP   #CFFC       ;Выход через процедуру
                                       ;Fn h(), см.с.56.
```

Процедура **GETSPR.**

```
C0F0  3A6DC1  GETSPR  LD   A,(COUNT)   ;Приняли данные из
                                       ;троичного счетчика.
C0F3  3D              DEC  A           ;Сняли с него единицу.
C0F4  326DC1          LD   (COUNT),A   ;И снова его запомнили.
C0F7  C0              RET  NZ          ;Если не ноль, значит
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

                                       ;наш спрайт еще не весь
                                       ;"выехал" на экран и при
                                       ;нимать спрайт в буфер
                                       ;пока не надо.
C0F8  3E03            LD    A,#03      ;Выставляем число 3 в
C0FA  326DC1          LD    (COUNT),A  ;троичном счетчике,
C0FD  3A6CC1          LD    A,(NUMBER) ;Номер спрайта.
C100  113F00          LD    DE,#003F   ;Длина шаблона одного
                                       ;спрайта.
C103  47              LD    B,A        ;Создали счетчик.
C104  2148D5          LD    HL,#D548   ;Адрес начала таблицы,
                                       ;в которой размещаются
                                       ;спрайты (54600).
C107  A7              AND   A          ;Сброс флага переноса,
                                       ;чтобы он не влиял на
                                       ;результат операции SBC.
C108  ED52            SBC   HL,DE      ;Выставление начального
C10A  19        NEXT  ADD   HL,DE      ;адреса для первого
                                       ;спрайта.
C10B  10FD            DJNZ  NEXT       ;Возврат за отысканием
                                       ;адреса очередного
                                       ;спрайта.
C10D  43              LD    B,E        ;Длина шаблона спрайта.
C10E  116EC1          LD    DE,BUFFER  ;Приняли адрес временного
                                       ;буфера для размещения
                                       ;спрайта.
C111  3A68C1  AGAIN   LD    A,(FLAG)   ;Проверка флага, связан-
                                       ;ного с необходимостью
                                       ;повторного "прогона"
                                       ;спрайта,
C114  FE00            CP    #00        ;Проверка на ноль.
C116  2801            JR    Z,BYPASS   ;Если ноль, то спрайт
                                       ;принимать не надо, идем
                                       ;в обход и обнуляем бу-
                                       ;фер.
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

C118  7E              LD   A,(HL)         ;Байт шаблона спрайта.
C119  12      BYPASS  LD   (DE),A         ;Запомнили его в буфере.
C11A  23              INC  HL             ;Переход к очередному
                                          ;байту шаблона.
C11B  13              INC  DE             ;Переход к очередному
                                          ;байту буфера.
C11C  10F3            DJNZ AGAIN          ;Если не все байты шабло-
                                          ;на скопированы, то воз-
                                          ;врат.
C11E  3A6AC1          LD   A,(REPIT)      ;Проверка необходиимости
                                          ;повтора.
C121  FE00            CP   #00            ;Если есть необходимость,
C123  C8              RET  Z              ;то выход, в переменной
                                          ;FLAG остается единица.
C124  3E00            LD   A,#00          ;В противном случае об-
C126  3268C1          LD   (FLAG),A       ;нуляем переменную FLAG.
C129  C9              RET                 ;Возврат в вызывающую
                                          ;процедуру,

                    Процедура RIGHT.

C12A  A7      RIGHT   AND  A              ;Сброс флага переноса.
C12B  0603            LD   B,#03          ;Начало внутреннего
                                          ;цикла. Установка
                                          ;параметра цикла.
C12D  D5              PUSH DE             ;Сохранение регистра
                                          ;на время работы подпро-
                                          ;граммы.
C12E  1A      LOOP-3A LD   A,(DE)         ;Взяли в аккумулятор
                                          ;байт из левого верти-
                                          ;кального столбца в шаб-
                                          ;лоне спрайта.
C12F  1F              RRA                 :Ротация вправо.
C130  12              LD   (DE),A         ;Заменили байт в шаблоне
C131  F5              PUSH AF             ;Сохранили флаговый ре-
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

                                          ;гистр на стеке. Важен
                                          ;флаг переноса.
C132  3E15            LD    A,#15         ;Сдвиг на 21 байт для
                                          ;перехода к соседнему
C134  83              ADD   A,E           ;справа столбцу.
C135  5F              LD    E,A           ;Указатель в буфере
                                          ;спрайта переместили.
C136  3001            JR    NC,LBL-7      ;Если не возникло пе-
                                          ;реполнения, то обход.
C138  14              INC   D             ;Иначе наращиваем
                                          ;старший регистр.
C139  F1       LBL-7  POP   AF            ;Восстановили флаг С.
C13A  10F2            DJNZ  LOOP-3A       ;Конец цикла.
C13C  D1              POP   DE            ;Восстановили указатель
                                          ;на буфер.
C13D  3A64C1          LD    A,(WIDTH)     ;Ширина окна.
C140  47              LD    B,A           ;Ввели ее в качестве
                                          ;счетчика.
C141  CB1E     LOOP-4A RR   (HL)          ;Ротация вправо байтов
                                          ;в окне.
C143  23              INC   HL            ;Переход в окне к сосед-
                                          ;нему знакоместу справа.
C144  10FB            DJNZ  LOOP-4A       ;Прокручиваем линию по
                                          ;всей ширине окна.
C146  C9              RET                 ;Возврат в вызывающую
                                          ;процедуру.

                    Процедура LEFT.

C147  A7       LEFT   AND   A             ;Сброс флага переноса.
C148  D5              PUSH  DE            ;Сохранение регистра
                                          ;на время работы подпро-
                                          ;граммы.
C149  0603            LD    B,#03         ;Начало внутреннего
```

```
Адрес Маш.код Метка    Мнемоника          Комментарий.

                                          ;цикла. Установка
                                          ;параметра цикла.
C14B  1A      LOOP-3B  LD   A,(DE)        ;Взяли в аккумулятор
                                          ;байт из правого верти-
                                          ;кального столбца в шаб-
                                          ;лоне спрайта.
C14C  17                RLA               ;Ротация влево.
C14D  12                LD   (DE),A       ;Заменили байт в шабло-
                                          ;не.
C14E  F5                PUSH AF           ;Сохранили флаговый ре-
                                          ;гистр на стеке. Важен
                                          ;флаг переноса.
C14F  7B                LD   A,E          ;Сдвиг назад на 21 байт
C150  D615              SUB  #15          ;для перехода к соседнему
C152  5F                LD   E,A          ;слева столбцу.
C153  3001              JR   NC,LBL-8     ;Если не возникло двоич-
                                          ;ного займа, то обход.
C155  15                DEC  D            ;Уменьшаем старший байт.
C156  F1      LBL-8     POP  AF           ;Восстановили флаг С.
C157  10F2              DJNZ LOOP-3B      ;Конец внутреннего цикла.
C159  D1                POP  DE           ;Восстановили указатель
                                          ;на буфер.
C15A  3A64C1            LD   A,(WIDTH)    ;Ширина окна.
C15D  47                LD   B,A          ;Ввели ее в качестве
                                          ;счетчика.
C15E  CB16    LOOP-4B   RL   (HL)         ;Ротация влево байтов
                                          :в окне.
C160  2B                DEC  HL           ;Переход к соседнему
                                          ;знакоместу слева.
C161  10FB              DJNZ LOOP-4B      ;Прогон на всю ширину
                                          ;окна.
C163  C9                RET               ;Возврат в вызывающую
                                          ;процедуру.
```

```
Адрес Маш.код Метка   Мнемоника        Комментарий.

C164  05      WIDTH   DEFB #05         ;Ширина окна.
C165  04      COUNT8  DEFB #00         ;Восьмеричный счетчик.
C166  00      SCRAD   DEFW #0000       ;Экранные координаты
C168  00      FLAG    DEFB #00         ;Флаг, указывающий на
                                       ;необходимость повтора
                                       ;"прогона" спрайта.
C169  00              NOP
C16A  00      REPIT   DEFB #00         ;Параметр повтора.
C16B  00      DIREC   DEFB #00         ;Направление скроллинга
C16C  00      NUMBER  DEFB #00         ;Номер спрайта.
C16D  00      COUNT   DEFB #00         ;Троичный счетчик для
                                       ;процедуры GETSPR.
C16E  0000..  BUFFER  DEFM #0000...    ;Буфер для временного
                                       ;хранения шаблона
                                       ;спрайта.
```

3. АНИМАЦИЯ В ВЕКТОРНОЙ ГРАФИКЕ

3.1 Вместо вступления.

Для того, чтобы ЭФФЕКТИВНО использовать работу с векторной графикой, от программиста, в принципе, требуется знание некоторых наук (хотя бы двух) - аналитической геометрии и матричной алгебры. Невредно также еще знать основы начертательной геометрии. С другой стороны, мы понимаем, что далеко не все наши читатели имеют высшее образование, а многие из тех, кто его и имеют, либо не изучали этих дисциплин, либо успели их прочно подзабыть.

Поэтому мы постараемся обойтись без них, оставаясь в рамках того, что Вам известно из курса средней школы, но при этом нам придется пойти на значительные упрощения. У этих упрощений есть неприятная особенность. С одной стороны, они (упрощения) облегчают понимание основных принципов и идей (и это хорошо), но с другой стороны они же усложняют практическую реализацию тех же самых принципов и идей (а это уже плохо).

Чтобы совместить несовместимое, мы подойдем к векторной графике поэтапно. Во-первых, простыми словами мы изложим основные понятия и приемы, не выходя в высшую математику, а во-вторых, мы подскажем, как и для чего впоследствии Вы сможете применить эту высшую математику, если все же решитесь заняться этими вопросами более глубоко.

Итак, обратите внимание на следующее:

Знание НАЧЕРТАТЕЛЬНОЙ ГЕОМЕТРИИ Вам может пригодиться для того, чтобы образно представлять, как ведут себя точки, линии, плоские геометрические фигуры и объемные тела в пространстве. Чтобы представлять, как они могут располагаться относительно

друг друга, что образуется в результате пересечения одних объектов с другими и т.п. Это знание не является необходимым, а только полезным.

Знание АНАЛИТИЧЕСКОЙ ГЕОМЕТРИИ необходимо для того, чтобы представлять линии, геометрические фигуры и тела в виде уравнений и систем уравнений. Тогда определить, например, пересекает ли данная линия данное тело или нет (попал в Вас луч лазера, выпущенный противником или нет?), можно путем решения системы уравнений. Это легко поручается компьютеру. При этом будут получены и координаты точек пересечения (можно узнать, в какой отсек Вашего корабля попал враг). Можно делать и такие вещи, как строить в виде функции перпендикуляр к криволинейной поверхности, определять, находится ли данное дерево в тени данного холма или нет, если солнце расположено там-то и там-то. Одним словом, аналитическая геометрия нужна, чтобы использовать ФУНКЦИОНАЛЬНОЕ ПРЕДСТАВЛЕНИЕ для всевозможных геометрических объектов. Она как бы помогает забыть о том, что в программе летают корабли, стреляют лазеры, светят звезды. Вместо всего этого есть только уравнения, уравнения и еще раз уравнения. А все перипетии космических баталий - это решения этих уравнений. Решаем уравнения, проверяем результат. Если он такой-то или такой-то, значит цель поражена - получите премию за уничтоженного пирата.

Для кого-то луч лазера - это смертоносный заряд, а с точки зрения аналитической геометрии это всего лишь уравнение прямой. Орбитальная станция - вовсе и не станция, а набор граней, каждая из которых вовсе и не грань, а набор отрезков, каждый из которых тоже описывается уравнением прямой. Так что, вопрос поражения станции лазерным лучом - это решение системы полученных уравнений.

Можно ли обойтись без такого функционального представления? На простых задачах, по-видимому да. Но чтобы не запутаться в сложных ситуациях, к нему придется все-таки обратиться.

Знание МАТРИЧНОЙ АЛГЕБРЫ тоже необходимо, но по другой причине. Оно необходимо, чтобы программы были ЭФФЕКТИВНЫМИ. Когда корабль летит в пространстве и при этом кувыркается, а окружающие его объекты тоже не стоят на месте, постоянно приходится пересчитывать координаты всех точек и концов отрезков, изображаемых на экране. Можно написать сотни процедур, охватывающих всевозможные комбинации преобразований, но на это не хватит ни компьютерной памяти, ни скорости работы процессора, ни собственного ума.

Поэтому программисты стараются во-первых разбить всевозможные преобразования координат на простые составляющие, каждое из них описывают простейшим алгоритмом, а потом оказывается, что любые самые сложные операции можно "собирать" из простейших, как из набора кубиков детского конструктора, если сделать так, чтобы эти "кубики" были похожи друг на друга (то есть принадлежали бы одному набору). Для этого и служит матричная алгебра. Сначала все уравнения, описывающие поведение объектов в пространстве представляют в матричной форме (в виде однотипных кубиков), а затем из этих кубиков собирают необходимую подпрограмму. "Кубики" при этом "стыкуются" друг с другом по правилам операций с матрицами (сложение матриц, умножение матриц на постоянный коэффициент, перемножение матриц между собой, обращение матриц). Обратите внимание на то, что полученная в результате подпрограмма может работать по-разному, в зависимости от того, какие "кубики" в нее вставили. А "вставляются" всякий раз разные.

Процедуру, которая "собирает" другие процедуры ("кубики") в конструкцию для решения задачи называют ПОРОЖДАЮЩЕЙ ПРОЦЕДУРОЙ. Когда программист устраивает головоломные сражения на экране, он фактически работает только с ней и всякий раз "подсовывает" ей нужные параметры. А она просмотрев эти параметры "собирает" в одну конструкцию множество прочих процедур и запускает их.

В рамках нашей книги мы не будем вторгаться ни в аналити-

ческую геометрию, ни в алгебру матриц, иначе нам пришлось бы увеличить размеры этой книги в несколько раз и потерять при этом основную массу читателей. Так что, предоставим тем, кому это нужно, развивать свои знания в этом направлении самостоятельно. Самое же общее понятие о том, зачем все это нужно, мы Вам дали.

3.2 Системы координат.

Анимация в векторной графике фактически происходит за счет перемещения тела в пространстве, т.е. за счет изменения его положения относительно некоторой системы координат. Разбираясь с этим вопросом, мы никак не сможем обойтись без того, чтобы не уделить некоторое внимание обзору основных координатных систем. Мы понимаем, что эти вопросы известны нашим читателям из программы средней школы и потому затронем их очень кратко.

Прежде всего, скажем, что задать систему координат - это значит указать, где находится ее начало и как заданы основные направления.

3.2.1. Основная задача векторной графики.

Давайте посмотрим, с какими объектами Вам приходится иметь дело, когда Вы играете в какую-либо игру, работающую в векторной графике. Возьмем, для примера, широкоизвестную программу ELITE.

Итак, самый первый объект, без которого никак не обойтись - это Вы сами, хотя это и не объект, а субъект. У Вас очень важная роль в игре. Вы - ***наблюдатель*** и с Вами связана своя система координат, началом которой является та точка, в которой Вы находитесь. Так ее и назовем - ***система координат, связанная с наблюдателем.***

Второй объект, без которого тоже не проходит ни одна игра - это экран Вашего телевизора (монитора). У него своя система координат, очень нужная и важная. Она двумерная. Как вы знаете, экран имеет разрешающую способность по горизонтали 256 пикселов и по вертикали - 192. Начало этой системы координат в левом нижнем углу экрана. Назовем ее ***экранной системой координат.***

Третий объект - Ваш корабль. Поскольку Вы в полете крепко пристегнуты ремнями к креслу, то мы можем считать, что ***система координат, связанная с кораблем*** - это та же ***система, которая связана с наблюдателем***. Начало координат этой системы - геометрический центр Вашего корабля, но это сложно. Проще считать, что ее начало - Вы сами. Ось X направлена вдоль главной оси Вашего корабля вперед. Ось Y перпендикулярна к ней и находится в главной плоскости корабля, а ось Z перпендикулярна им обеим.

Четвертый объект - корабль противника, находящийся от Вас на определенном расстоянии и совершающий в пространстве маневры. У него есть своя система координат, связанная с ним. Она определяется конструкцией корабля точно так же, как и та система, которая связана с Вашим кораблем.

Ну и, наконец, могут быть (а могут и не быть) в программе еще несколько систем координат, связанных с какими-либо небесными телами, например со звездой, планетой или, скажем, с орбитальной станцией.

Итак, Ваша задача при исполнении динамической векторной графики сводится к тому, чтобы строить и перестраивать на экране изображения космических объектов в то время, как эти объекты вращаются в пространстве (изменяют положение осей своей системы координат относительно принятой Вами системы), плюс к этому летят по какой-то траектории (изменяет положение начала своей системы координат относительно принятой) плюс к этому Вы тоже не стоите на месте, а двигаетесь и вращаетесь, то есть тоже постоянно изменяете направление своих осей. А поскольку и строить и перестраивать изображение надо на плоском экране монито-

ра, то дополнительно возникает задача по пересчету всех этих пространственных изменений в плоскую систему координат, связанную с экраном.

Если все это выглядит до некоторой степени запутанным, то не пугайтесь, сейчас мы начнем этот клубок распутывать.

3.2.2. Плоские системы координат.

Системы координат на плоскости - двумерные и потому, чтобы определить положение точки достаточно двух параметров.

Прямоугольная декартова система.

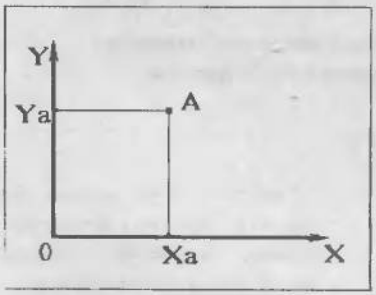


Рис. 25. Плоская прямоугольная система координат.

Эта система является простейшей, см. рис.25. В ней координаты каждой точки задаются двумя числами - проекциями этой точки на горизонтальную и вертикальную оси. Простейшим примером является экран Вашего монитора (телевизора).

Если вернуться к программе ELITE, то однозначно можно предсказать, что в этой системе координат организована глобальная карта галактики (рис.26). Координаты X,Y каждой звезды программа самостоятельно рассчитывает по какому-то своему алгоритму. Чтобы не тратить память на хранение всей этой карты (а таких галактик в игре бесконечное количество) программа генерирует карту с помощью статического датчика случайных чисел, используя при этом в качестве входных параметров шесть байтов

(читатели нашего периодического издания ZX-РЕВЮ называют их "галактическими байтами").

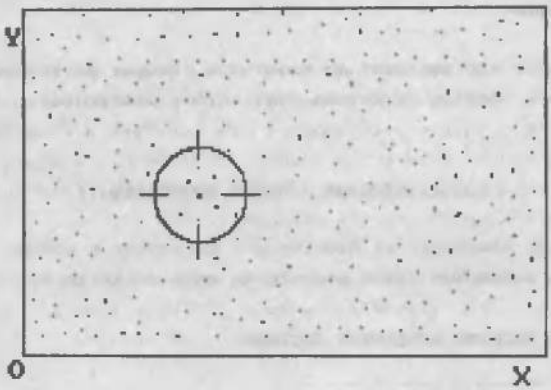


Рис. 26 Глобальная карта галактики задается в плоской прямоугольной системе

Полярная система координат.

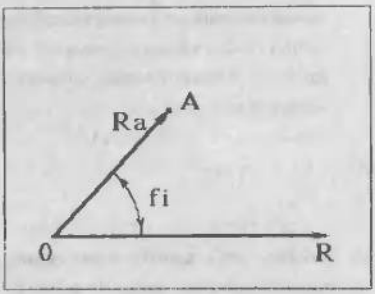


Рис. 27 Полярная система координат

Система представлена на рис.27. В этой системе координат положение каждой точки задается величиной ее радиуса-вектора относительно начала координат и углом поворота этого радиуса-вектора.

По-видимому, в этой системе координат организована локальная карта галактики. В ней неважны точные координаты звезд, а важно только расстояние (длина радиуса-вектора) от места Вашего нахождения до цели. Оно нужно для того, чтобы понять, хватит ли Вам

топлива на борту корабля для перелета. Ваш запас ограничен 7-ю световыми годами. Начало системы координат "привязано" к Вашему местоположению.

В этой же системе организована такая операция, как заправка топливом от звезды. Поскольку при этом основным фактором является только расстояние между Вашим кораблем и поверхностью светила, то такой системы там вполне достаточно. Важно не подлететь к звезде слишком близко, т.к. можно перегреть корабль, но и нельзя находиться от нее слишком далеко.

В этой же системе координат Вы работаете, когда приближаетесь к орбитальной станции с помощью JUMP-двигателя. Но при этом начало системы координат "привязано" к планете.

3.2.3. Пространственные системы координат.

Трехмерная прямоугольная система.

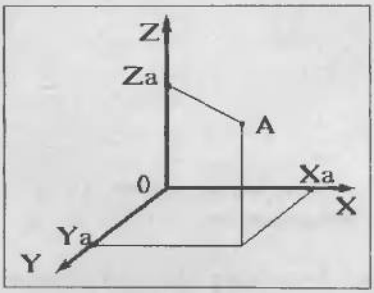


Рис. 28 Трехмерная прямоугольная система координат

Она образуется тремя взаимноперпендикулярными осями. Положение точки в пространстве в этой системе может быть задано тремя координатами - проекциями на эти оси (рис.28).

В играх в этой системе координат задают конструкцию кораблей. Возьмите лист бумаги в клеточку, вычертите на ией конструкцию своего корабля и для каждой вершиины проставьте ее координаты (см. рис.29). Всякий раз, когда в поле Вашего зрения попадает посторонний объект (корабль, станция, астероид), он вычерчивается на экране по координатам,

взятым из таблиц. А в таблицах они заданы именно в этой системе. За начало координат принимается центр тяжести или геометрический центр корабля. Сколько бы кораблей Вы ни использовали в своей программе, задать координаты конструкции каждого Вам придется.

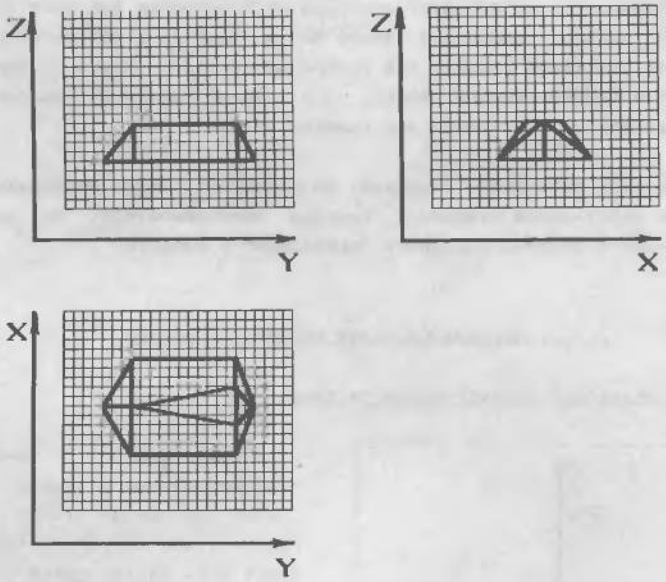


Рис. 29 Изображение кораблей в пространственной прямоугольной системе координат

Сферическая система координат. Это аналог полярной системы координат, но в пространстве. Положение объекта задается во-первых радиусом-вектором, направленным от начала координат к объекту, а во-вторых, двумя углами, которые образуются между этим радиусом-вектором и главными осями (см. рис. 30).

Если в программе ELITE Вы ведете карьеру бойца, то проводите в этой системе координат основную долю времени. В этой

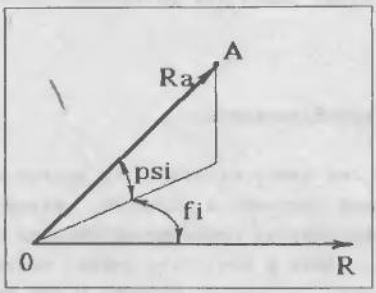


Рис. 30 Сферическая система координат

системе координат задают положение объектов относительно наблюдателя в пространстве. Так, например, в этой системе координат происходят сцены космических боев и операции причаливания к космической станции. Предположим, Вам надо изобразить на экране вражеский корабль. Вопрос, **как его изображать**, решается по его конструкции, заданной в прямоугольной системе, а вот вопрос, **где его изображать**, уже решается в трехмерной сферической системе. За начало системы координат принимается положение наблюдателя.

Те, кто играют в игру ELITE, знают, что использование JUMP-двигателя невозможно, если в поле зрения локатора корабля есть какие-либо объекты (астероиды, корабли и т.п.). Это очень мешает двигаться быстро и вынуждает даже миролюбивых пилотов уничтожать все то, что находится в зоне видимости. И наложено ограничение на использование JUMP-двигателя отнюдь не случайно. Так устроена программа именно потому, что процедуры, имитирующие JUMP-двигатель, обрабатывают координаты Вашего корабля в полярной системе, начало которой "привязано" к плаиете. Когда же появляется какой-либо объект, то надо переключаться на процедуры, работающие в сферической системе координат, "привязанной" к Вам. Это и делает программа, а Вам же сообщают, что использовать JUMP-переход нельзя, потому что якобы он может пройти ненормально из-за помех, вносимых посторонним телом в геометрию пространства. Кстати, это пример того, как тонко обыгрываются программные ограничения в сценарной поддержке. Игра от этого становится даже интереснее.

Прочие системы. Существует бесконечное количество и иных

видов систем координат, но для наших задач уже достаточно вышерассмотренных четырех.

3.3 Координатные преобразования.

Это важная задача, причем она имеет отношение не только и не столько к пересчету из полярной системы в плоскую прямоугольную и наоборот или к пересчету из сферической системы в прямоугольную трехмерную и т.п.. Дело в том, что очень часто приходится делать преобразования, оставаясь в одной и той же системе, например при изменении координат точки начала отсчета или при повороте координатных осей. Так что задача разбивается на две: пересчет координат между системами и преобразования систем координат.

3.3.1. Пересчет координат между системами.

Перевод координат из плоской прямоугольной системы в полярную показан на рис.31. Уравнения для пересчета очень просты:

```
Ra = SQRT (Xa*Xa + Ya*Ya)          (1)
fi = arctg (Ya/Xa)
```

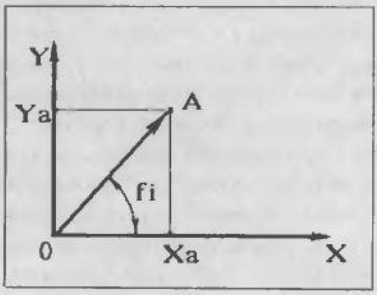


Рис.31 Перевод координат из плоской прямоугольной системы в полярную

Перевод координат из полярной системы в плоскую прямоугольную - это задача обратная к только что рассмотренной:

```
Xa = Ra*COS(fi)                                    (2)
Ya = Ra*SIN(fi)
```

Перевод координат из прямоугольной трехмерной системы в сферическую показан на рис.32 . Уравнения имеют вид:

```
R = SQRT (Xa*Xa + Ya*Ya + Za*Za)
psi = arctg (Za/SQRT(Xa*Xa + Ya*Ya))               (3)
fi = arctg (Ya/Xa)
```

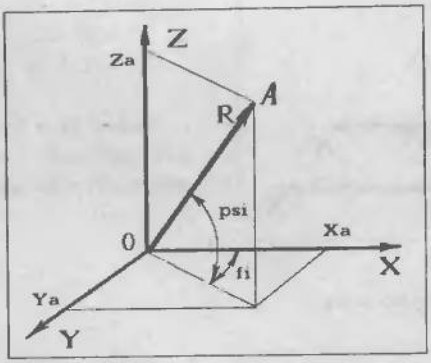


Рис. 32 Перевод координат из прямоугольной системы координат в сферическую.

Обратный перевод координат из сферической системы в прямоугольную описывается следующими уравнениями:

```
Xa = R*COS(psi)*COS(fi)
Ya = R*COS(psi)*SIN(fi)                            (4)
Za = R*SIN(psi)
```

3.3.2 Преобразования координат внутри системы.

В рамках данного раздела мы будем обозначать старые координаты символами X,Y,Z, а новые координаты - со штрихом - X', Y', Z' и т.п.

Преобразования плоской прямоугольной системы. Это самый простой случай. Рассмотрим сначала параллельный перенос осей (см. рис.33).

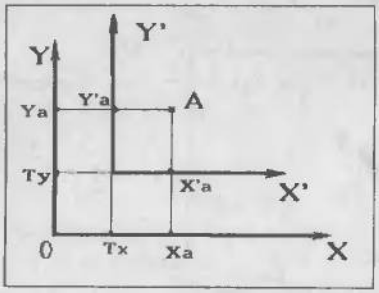


Рис. 33 Перенос осей

$$X' = X + Tx \qquad (5)$$
$$Y' = Y + Ty$$

Здесь Tx и Ty - координаты вектора, порождающего новую систему координат.

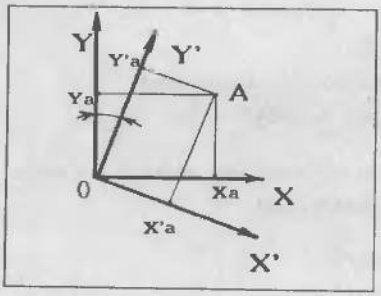


Рис. 34 Поворот осей

Поворот осей (см. рис. 34) на произвольный угол "teta" описывается следующей системой уравнений.

$$X' = X*COS(teta) - Y*SIN(teta) \qquad (6)$$
$$Y' = X*SIN(teta) + Y*COS(teta)$$

Принцип композиции: если происходит сложное перемещение, выражающееся одновременно в изменении положения начала координат и в повороте координатных осей, то итоговое перемещение можно рассматривать как результат двух независимых и несвязанных между собой перемещений. То есть, пересчет координат можно выполнять по частям, разбивая одно сложное преобразование на два или более простых.

Таким образом, если нам надо одновременно повернуть оси и перенести их в новое место, то разбив задачу на две, мы получим:

```
X' = Tx + X*COS(teta) - Y*SIN(teta)          (7)
Y' = Ty + X*SIN(teta) + Y*COS(teta)
```

Принцип композиции справедлив и для более сложных преобразований: включающих в себя переводы из одних систем в другие и внутрисистемные преобразования.

Преобразования полярной системы координат. Согласно принципу композиции перемещений здесь мы также можем раздельно рассмотреть перенос начала системы координат на радиус-вектор T и поворот координатной оси на угол "teta".

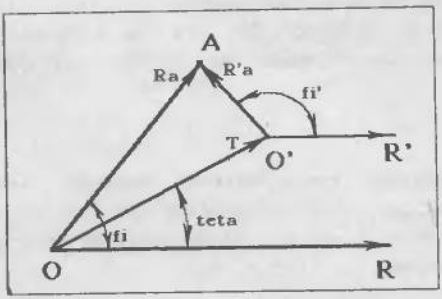


Рис. 35 Перенос начала полярной системы координат

Чтобы не путаться с векторами и обойтись без аналитической геометрии, мы воспользуемся принципом композиции перемещений и выполним эту операцию посредством нескольких преобразований, например так:

- переведем исходный вектор R в прямоугольную систему координат по формулам (2). Обозначим эту операцию [P1].

- переведем преобразующий вектор T тоже в прямоугольную систему координат по тем же формулам (2). Это операция [P2].

- в прямоугольной системе координат перейдем от старых координат к новым по формулам (5). Пусть это будет операция [P3].

- вернемся к полярной системе координат по формулам (1). Пусть это будет операция [P4].

Тогда вся операция переноса начала системы координат (обозначим ее T) будет представлять из себя как бы комбинацию из более простых стандартных операций.

$$[T] = ([P1],[P2],[P3],[P4])$$

Вот тут-то нам бы очень пригодилась матричная алгебра. Вы ведь видите, что из "кубиков" [P1...P4] мы "собрали" операцию [T], которая может быть "кубиком" для других, еще более сложных операций.

Поворот координатной оси в полярной системе. Эта задача совсем проста. поскольку начало координат при этом не меняется, то радиус-вектор остается тем же. А координатный угол изменяется на величину поворота оси "teta".

$$\begin{aligned} R' &= R \\ fi' &= fi + teta \end{aligned} \qquad (8)$$

Преобразования трехмерной прямоугольной системы координат.

На рис. 36 показан перенос осей в пространстве. Ситуация здесь аналогично рассмотренной выше для плоской прямоугольной системы и формулы для пересчета имеют следующий вид:

```
X' = X + Tx
Y' = Y + Ty                                    (9)
Z' = Z + Tz
```

Здесь Tx, Ty и Tz - координаты вектора, порождающего новую систему координат.

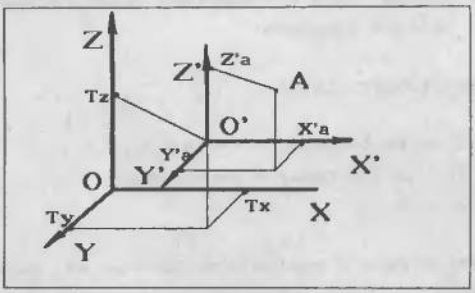


Рис. 36 Перенос осей в трехмерной прямоугольной системе координат

На рис.37 показан поворот координатных осей на угол "teta" вокруг оси X:

```
X' = X
Y' = Y*COS(teta) - Z*SIN(teta)                 (10)
Z' = Y*SIN(teta) + Z*COS(teta)
```

При повороте вокруг оси Y:

```
X' = X*COS(teta) + Z*SIN(teta)
Y' = Y                                         (11)
Z' = -X*SIN(teta) + Z*COS(teta)
```

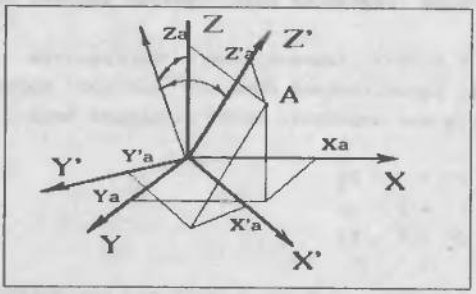


Рис. 37 Поворот осей в трехмерной прямоугольной системе координат

При повороте вокруг оси Z:

```
X' = X*COS(teta) - Y*SIN(teta)
Y' = X*SIN(teta) + Y*COS(teta)          (12)
Z' = Z
```

Поворот вокруг некоей произвольной оси мы не рассматриваем, т.к. согласно принципу композиции перемещений можно вместо него рассмотреть два поворота вокруг фиксированных осей и результат будет тот же.

Преобразования сферической системы координат. Чем дальше мы двигаемся, тем больше уравнений нам приходится выписывать и, если мы захотим дать все необходимые уравнения для изменения начала начала системы координат и для поворота радиуса-вектора, то получим очень громоздкий раздел.

С другой стороны, на примере преобразования полярной системы Вы, уважаемый читатель, уже поняли, что все можно сделать более просто, если учесть принцип композиции перемещений.

- переводим сферические координаты исходного радиуса-век-

тора R в трехмерную прямоугольную систему по формулам (4);

- переводим сферические координаты преобразующего радиус-вектора R в ту же систему по тем же формулам (4);

- в новой системе по формулам (9-12) выполняем переносы и повороты осей;

- делаем обратное преобразование по формулам (3) и возвращаемся назад в сферическую систему координат.

3.4 Представление геометрических объектов в памяти компьютера.

3.4.1 Представление точки.

Это простейший геометрический объект. Для того, чтобы описать точку, нам достаточно выделить несколько ячеек памяти (от двух до шести) для хранения ее координат. Прежде всего, надо определиться, в какой системе координат мы работаем. Если это плоская система, то для задания точки достаточно двух координат, а если пространственная, то нужно уже три.

Теперь нам нужно определиться с тем, сколько байтов мы расходуем на одну координату. Во многих случаях достаточно по одному байту (от 0 до 255), но если этого диапазона значений Вам недостаточно, то придется использовать по два байта (от 0 до 65535).

Какие координаты Вы запоминаете, зависит от избранной системы координат. Так, например, в трехмерной прямоугольной системе Вы запоминаете, а впоследствии обрабатываете три проекции точки на оси X,Y,Z. В сферической же системе координат Вы запоминаете длину радиус-вектора и пару углов.

3.4.2 Представление отрезка.

Для представления отрезка прямой в памяти компьютера нужно вдвое больше ячеек памяти, чем для представления точки. Это и не удивительно, ведь у отрезка два конца. Неважно, какой системой координат Вы при этом пользуетесь, надо только, чтобы координаты обоих концов отрезка были записаны в одной системе.

3.4.3. Представление плоской фигуры.

Будем считать такую фигуру n-угольником. Даже если это и не так, например если мы имеем дело с окружностью, то ее можно заменить n-угольником с большим количеством вершин и малым размером сторон.

Прежде всего, нам конечно придется занести в память координаты всех n вершин этой фигуры. Но этого еще, увы, недостаточно. Дело в том, что мы еще ничего не знаем о сторонах. На рис.38 показаны две совершенно непохожие друг на друга пятиугольные фигуры, имеющие одинаковые координаты вершин.

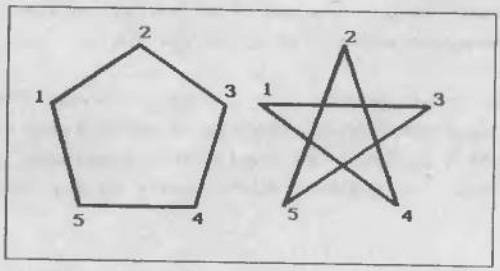


Рис. 38 Две различные фигуры с одинаковыми координатами вершин

Таким образом, для задания плоской фигуры кроме массива

координиат вершин необходимо организовать еще массив, описывающий стороны фигуры. Так, на рисунке 38 сторона 1-2 у первой фигуры существует, а у второй - нет, зато там существует сторона 1-3, которой нет у первой фигуры.

Самый простой способ создать такой массив - записать порядок обхода всех вершин многоугольника, начиная с любой. При этом имеет смысл руководствоваться определенным порядком обхода, например обходить вершины всегда только против часовой стрелки (где такое правило может оказаться полезным, мы покажем далее).

Тогда для первой фигуры (рис. 38) массив описания сторон будет иметь вид: 1,2,2,3,3,4,4,5,5,1; а для второй фигуры - соответственно: 1,3,3,5,5,2,2,4,4,1.

Написать программу (процедуру) для построения фигуры, у которой заданы экранные координаты всех вершин и описаны стороны, совсем несложно. Это можно сделать как на БЕЙСИКе, так и в кодах. Способ реализации неважен. Важно то, как это организовано.

Для этой книги мы подготовили такую программу в машинном коде, но при редактировании вынуждены были от нее отказаться, т.к. за большим размером самой программы теряется ее суть. Нам остается только ограничиться демонстрационным примером на БЕЙСИКе, который естественно работает далеко не плавно, но зато демонстрирует сам принцип того, как задаются плоские фигуры.

Демонстрационный пример.

Эта программа демонстрирует простейшую векторную анимацию на БЕЙСИКе. Фигура 1 (рис.38) переходит в фигуру 2 (рис. 38) за несколько шагов, количество которых задано переменной steps.

```
10 CLS: OVER 1: REM задавая режим OVER 1, мы обеспечиваем
                   стирание ранее нарисованного изображе-
                   ния путем повторной печати в том же
                   месте. Режим OVER 1 на БЕЙСИКе работа-
                   ет так же, как наложение по XOR в ма-
                   шинном коде.
20 DIM a(10):    REM массив описания сторон
30 DIM x(5): DIM y(5):  REM массивы координат исходной фи-
                           гуры
40 DIM p(5): DIM q(5):  REM массивы координат вершин ко-
                           нечной фигуры
50 DIM d(5): DIM e(5):  REM вспомогательные массивы, в ко-
                           торых хранятся приращения ко-
                           ординат x,y при переходе от
                           точки N к вершине N+1. Нужен
                           для команды DRAW.
60 FOR i=1 TO 5
70 READ x(i): READ y(i):REM Ввод координат вершин исходной
                           фигуры.
80 NEXT i
90 FOR i=1 TO 5
100 READ p(i): READ q(i):REM Ввод координат вершин конечной
                           фигуры.
110 NEXT i
120 FOR i=1 TO 10
130 READ a(i)            :REM Ввод описания сторон фигуры.
140 NEXT i
150 LET steps=10         :REM Можете поменять это число по
                           своему вкусу.
160 FOR k=1 TO steps
170 FOR j=1 TO 5
180 LET x(j)=x(j)+(p(j)-x(j))/steps*k
190 LET y(j)=y(j)+(q(j)-y(j))/steps*k
200 NEXT j
210 GO SUB 500: PAUSE 50:REM Печать фигуры.
220 GO SUB 500           :REM Стирание фигуры.
230 NEXT k
```

```
240 OVER 0                  :REM Восстановление режима перед
                                 выходом
250 STOP

500 PLOT x(1),y(1)
510 FOR i=1 TO 5
520 LET d(i)=x(a(2*i))-x(a(2*i-1))
530 LET e(i)=y(a(2*i))-y(a(2*i-1))
540 DRAW d(i),e(i)
550 NEXT i
560 RETURN

900 DATA 128,150            :REM координаты вершин
910 DATA 192,100                 исходной фигуры
920 DATA 168,40
930 DATA 88,40
940 DATA 64,100

1000 DATA 128,150           :REM координаты вершин
1010 DATA 168,40                 конечной фигуры
1020 DATA 64,100
1030 DATA 192,100
1040 DATA 88,40

1100 DATA 1,2,2,3,3,4,4,5,5,1 :REM массив описания сторон
```

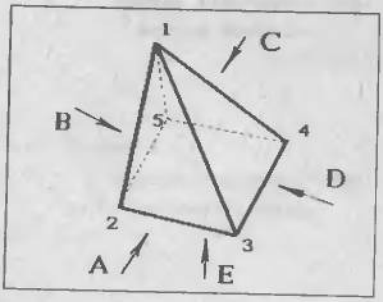
3.4.4. Представление объемных тел.

Трехмерные тела имеют вершины, которые связаны ребрами. Плоские геометрические фигуры, ограниченные этими ребрами, образуют грани.

Разумеется, задание тела начинается с того, что задаются трехмерные координаты всех его вершин. Однако, как и для случая с плоской геометрической фигурой, этого совершенно недостаточ-

но. Необходимо еще задать дополиительное описаиие. В принципе, здесь есть две возможности. Во-первых, можно описать все ребра, связывающие вершины друг с другом, как это делалось для плоской фигуры. Во-вторых, можно попробовать описать все грани тела. Второй путь для практической работы удобнее. Так, он удобнее хотя бы потому, что объемное тело удобно строить как совокупность граней, а описывать грани, как плоские фигуры, мы уже умеем. Не забудем при этом руководствоваться правилом описания сторон каждой грани "против часовой стрелки".

На рис.39 показана пирамида. У нее пять граней и массив описанния граней будет иметь, например такой вид:




```
1,2,2,3,3,1       - грань A;
1,5,5,2,2,1       - грань B;
1,4,4,5,5,1       - грань C;
1,3,3,4,4,1       - грань D;
2,5,5,4,4,3,3,2   - грань E.
```

Рис. 39 Описание граней объемного тела

Обратите внимание на то, что когда мы говорим о том, что вершины в гранн "должны обходиться против часовой стрелки", то полагаем, что мы смотрим на эту грань как бы "снаружи", а не "изнутри". Поэтому если Вам непонятно, как образовался порядок обхода вершин для граней B,C и E, то посмотрите на них с того направления, которое указано стрелкой.

И еще одно небольшое замечанне. Для конкретной программы можно упростить описание граней, если соблюдается условие, что каждая грань представляет выпуклую, связную и замкнутую об-

ласть, ограниченную сторонами. Т.е., например, стороны ограничивают область, а не проходят сквозь нее так, как это происходит в "звезде", показанной на рис. 38.

Тогда описание граней той же пирамиды может быть упрощено до следующего (в два раза уменьшается расход памяти):

```
1,2,3     - грань A;
1,5,2     - грань B;
1,4,5     - грань C;
1,3,4     - грань D;
2,5,4,3   - грань E.
```

3.5 Плоское преобразование.

До сих пор мы говорили о перемещениях и вращениях тел в пространстве, имея в виду некоторые преобразования систем координат, которые являются как бы "реальными", в которых эти объекты заданы, существуют и взаимодействуют. Такие координаты называют "мировыми" или фактическими. Но кроме них есть еще так называемые "наблюдаемые" координаты.

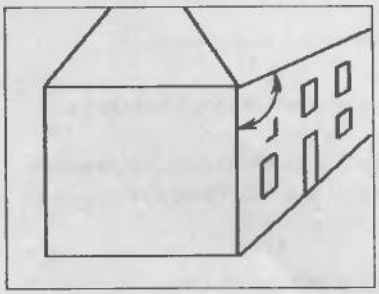


Рис. 40. Искажение углов при плоском преобразовании

Все, что Вас окружает в реальной жизни является трехмерным, но попробуйте закрыть один глаз и Вы увидите только плоское изображение окружающей действительности. Тем не менее Вы все равно сможете отличить куб от сферы. Тот факт, что "на глаз" угол между крышей и углом соседнего дома равен 120-ти градусам (рис. 40) никак не введет Вас в заблуждение. Всем своим

жизненным опытом Вы подготовлены к тому, что этот угол - прямой, а то, что он таковым не выглядит, так это просто обман зрения.

"Наблюдаемые" координаты, как следует из их названия, связаны еще и с наблюдателем. Понятно, что для наблюдателя объект является как бы двумерным, плоским, поэтому для отыскания этих наблюдаемых координат надо еще определиться с типом проектирования трехмерного тела на двумерную визуальную плоскость. Способы же такого проектирования (типы) бывают разными.

Из курса черчения средней школы Вы знаете, что наиболее часто применяются два типа аксонометрических проекций трехмерных тел на плоскость - это диметрическая проекция и изометрическая проекция. Они показаны на рис. 41.

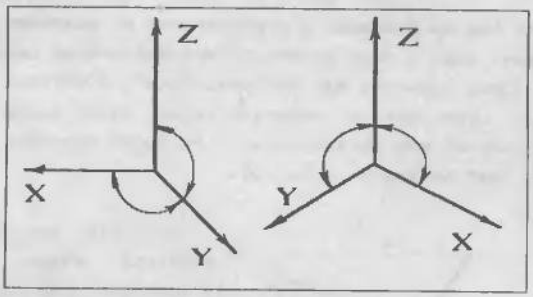


Рис. 41 Диметрическая и изометрическая проекция

Как видите, в диметрической проекции фиксируют направления осей Y и Z, а ось X направляют под углом 135 градусов по отношению к ним.

В изометрической проекции фиксируют направление оси Z, а оси X и Y располагают так, чтобы между всеми осями образовывался угол в 120 градусов.

Таким образом, при пересчете координат из трехмерных фактических в наблюдаемые координаты на визуальной плоскости важную роль играют углы наклона координатных осей "alfa" и "beta".

Для диметрической проекции мы получаем:

```
X' = Y - X*COS(alfa)          (13)
Y' = Z - X*SIN(alfa)
```

Поскольку угол "alfa" известен и равен 45 градусам, то получаем следующие формулы преобразования:

```
X' = Y - X*SQR(2)/2           (14)
Y' = Z - X*SQR(2)/2
```

Для изометрической проекции мы получаем:

```
X' = Y*COS(beta) - X*COS(alfa)          (15)
Y' = Z - X*SIN(alfa) - Y*SIN(beta)
```

Здесь углы "alfa" и "beta" тоже известны и равны 30 градусам. Таким образом:

```
X' = (Y-X)*SQR(3)/2           (16)
Y' = Z-(X+Y)/2
```

Если Вам из каких-то соображений захочется использовать иные типы проекций с другими значениями углов "alfa" и "beta", то Вы сами сможете рассчитать необходимые формулы, но при всех условиях постарайтесь зафиксировать хотя бы одну ось. Принято всегда фиксировать вертикальную ось, чтобы линии, являющиеся вертикальными в трехмерном объекте, оставались вертикальными и в наблюдаемом изображении, иначе изображение выглядит неестественно.

3.6 Проблемы линий невидимого контура.

Сокрытие линий невидимого контура - одна из важнейших задач в компьютерной графике. Сразу оговоримся, что она имеет множество решений. Самое общее решение - весьма сложное. Оно требует от компьютера высокого быстродействия, а от программиста большого опыта и потому мы его рассматривать не будем, а только наметим подходы, чтобы те, кому это потребуется в реальной практике, знали с чего начинать.

Во втором томе нашей графической серии мы рассмотрели один из возможных приемов сокрытия невидимых линий при изображении сложной трехмерной поверхности. Метод довольно интересный и, в определенной степени универсальный, но требующий большого объема компьютерных вычислений. Для построения графиков и при обработке результатов натурных или вычислительных экспериментов он подходит и вполне соответствует задачам тома "Прикладная графика", но если мы собираемся вести речь о графике динамической, то этот метод, увы, не годится. Здесь нам нужна скорость. Мы даже согласны смириться с тем, что само изображение может быть примитивным (например, космический корабль из программы "ELITE"), но изображаться на экране оно должно не только правильно, но и обязательно быстро.

По ходу игры корабль может совершать сложные пространственные маневры, мы можем видеть его с разных сторон, но всегда невидимые линии его контура не должны быть изображены. Проблема состоит в том, что компьютер должен "решить сам", видима та или иная линия или нет. Это интересная задача, которая предоставляет нашим читателям широкие возможности для самостоятельных исследований, но с чего-то надо начинать, и мы дадим здесь один несложный традиционный алгоритм.

Давайте рассмотрим один из космических кораблей. Его проекции показаны на рис.42 . Обратите внимание на то, что для то-

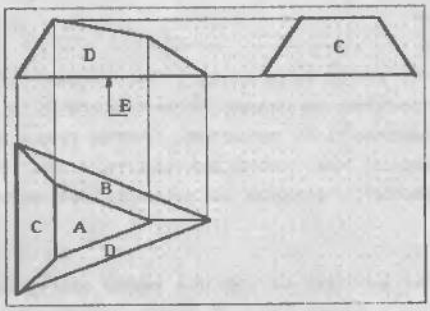


Рис. 42 Проекции "космического корабля"

го, чтобы изобразить на экране его трехмерное изображение (рис.43), нам было бы достаточно знать трехмерные координаты его вершин (6 вершин по три байта), а также условия связи их между собой, т.е. данные о том, какие вершины связаны между собой, а какие нет (так, как мы это делали для плоского многоугольника) - 9 ребер по 2 байта. Таким образом, для изображения этого корабля в любой его проекции нам бы хватило всего лишь 36-ти байтов, которые можно задать один раз и всегда ими пользоваться.

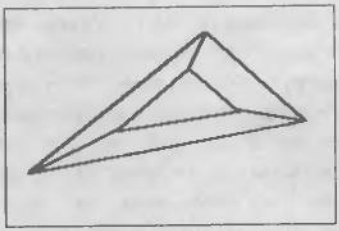


Рис. 43. "Космический корабль"

Тем не менее, в разделе, посвященном представлению трехмерных тел, мы почему-то пошли по иному пути. Мы представили каждый корабль не как набор вершин и связывающих их ребер, а как набор плоских граней, каждая из которых имеет по нескольку плоских вершин. Для данного "корабля" у нас получается 5 граней

(A,B,C,D,E), из которых три грани - четырехугольные, а две грани - треугольные.

Оказывается, такой подход может быть использован для того, чтобы при произвольном положении космического тела в пространстве, всегда можно было бы вычислить, какие грани видимы, а какие - нет. И сделать это тоже очень просто - нам поможет правило "часовой стрелки", которым мы руководствовались при нумерации вершин.

Взгляните на рисунок 44. На нем видны две грани нашего корабля - грань "A" и грань "E". А теперь посмотрите, как пронумерованы вершины при этих гранях. Если идти по этим вершинам в порядке возрастания (A: 1,2,3), (E: 4,5,6), то Вы увидите, что по грани "A" обход совершается против часовой стрелки, а по грани "E" - наоборот - по часовой стрелке. Интересно, не правда ли? Почему же так происходит?

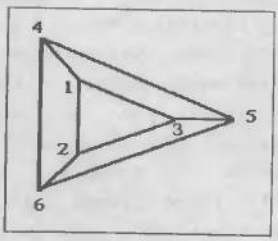


Рис. 44

А происходит это потому, что на грань "A" мы смотрим как бы "снаружи", а на грань "E" - как бы "изнутри". Вывод однозначный - грань A - видима для нас и ее надо изображать, а грань "E" - невидима и ее изображать не надо. Такое решение принять очень просто и все это исследование по каждой грани можно поручить самому компьютеру. Надо только договориться, каким образом компьютер узнает, что одна грань нумеруется против часовой стрелки, а другая - по ней.

К счастью, и это тоже сделать нетрудно. Если у Вас есть экранные координаты хотя бы трех вершин, принадлежащих данной грани (а у Вас есть экранные координаты всех ее вершин), то дальше логика простая. Возьмем любые три соседние вершины, при-

надлежащие одной грани. Запишем их координаты x1,y1; x2,y2; x3,y3. Рассмотрим приращения координат x и y при переходе от точки x1,y1 к двум другим:

$$\begin{vmatrix} x3 - x1 & x2 - x1 \\ y3 - y1 & y2 - y1 \end{vmatrix} \qquad (17)$$

А теперь вспомним из школьного курса простую формулу для расчета определителя системы двух линейных уравнений:

$$(x3-x1)*(y2-y1)-(x2-x1)*(y3-y1) = \begin{vmatrix} x3 - x1 & x2 - x1 \\ y3 - y1 & y2 - y1 \end{vmatrix} \qquad (18)$$

Так вот, если результат получится меньше нуля, то это значит, что вершины пронумерованы "против часовой стрелки" и изображать эту грань надо. Если бы результат оказался больше нуля, то тогда вершины были бы пронумерованы "по часам", значит грань невидима и изображению не подлежит.

Почему именно эта формула определяет, как пронумерованы вершины, мы объяснять не будем. Это доказательство входит в курс аналитической геометрии и нам оно ни к чему. Если есть сомнения, возьмите карандаш и бумажку в клеточку и через пять минут Вы убедитесь в правильности формулы и сами.

Пример в машинных кодах мы приводить не будем. Опытные "синклеристы" уже и так все поняли, и у них уже есть свои идеи в голове, как применить этот алгоритм на практике. А для начинающих мы "повращаем" на экране космический корабль с помощью обыкновенного БЕЙСИКа.

Возьмите лист бумаги в клетку и нарисуйте проекции нашего "корабля". За начало координат примем вершину 6 (0,0,0).

Всего вершин - 6, а граней - 5.

Тогда координаты вершин будут:

```
1 - (2,6,3)      2 - (2,2,3)      3 - (7,4,2)
4 - (0,8,0)      5 - (10,4,0)     6 - (0,0,0)
```

Теперь для всех пяти граней запишем условие обхода против часовой стрелки (если смотреть на грань снаружи):

```
A - 1,2,3        B - 1,3,5,4      C - 1,4,6,2
D - 2,6,5,3      E - 4,5,6
```

Вот мы и задали космический корабль. Можно приступать к программированию, но когда будем описывать грани, введем еще одно число - количество вершин в данной грани.

Описание массивов:

w(6,3) - массив трехмерных координат для всех вершин;
g(5,4) - массив, описывающий грани против часовой стрелки;
p(5) - массив, в котором хранится число вершин в каждой грани;
x(6) - массив плоских (экранных) горизонтальных координат для каждой вершины;
y(6) - массив плоских (экранных) вертикальных координат для каждой вершины.

Демонстрационная программа

```
 1  DIM w(6,3): DIM g(5,4): DIM p(5)
 2  DIM x(6): DIM y(6)
 3  LET teta = PI/18: REM Шаг углового поворота
 4  LET st = SIN (teta): LET ct = COS (teta)
 5  LET mx=2: LET my=2: LET mz=2
 6  LET dmx = 1: REM шаг приближения/удаления
10 DATA 6,5 : REM количество вершин и граней
20 REM координаты вершин
```

```
 21 DATA 2,6,3
 22 DATA 2,2,3
 23 DATA 7,4,2
 24 DATA 0,8,0
 25 DATA 10,4,0
 26 DATA 0,0,0
 30 REM описание граней (против часовой стрелки);
       первое число - количество вершин в грани
 31 DATA 3,1,2,3
 32 DATA 4,1,3,5,4
 33 DATA 4,1,4,6,2
 34 DATA 4,2,6,5,3
 35 DATA 3,4,5,6
 40 REM ввод исходных данных
 50 READ m,n
 60 FOR k=1 TO m : Ввод координат вершин
 62 FOR j=1 TO 3
 64 READ w (k,j)
 66 NEXT j: NEXT k
 70 FOR k=1 TO n: Ввод описания граней
 72 READ p (k)
 73 LET s = p(k)
 74 FOR j=1 TO s
 76 READ g (k,j)
 78 NEXT j: NEXT k
 99 GO SUB 100: GO TO 300
100 REM пересчет трехмерных координат вершин
       в двумерные экранные координаты по формулам
       для изометрической проекции (15)
105 CLS
110 FOR k = 1 TO m
120 LET x(k) = mx*SQR(3)*(w(k,2)-w(k,1))/2 + 127
130 LET y(k) = my*(w(k,3) - (w(k,2)+w(k,1))/2) + 87
140 NEXT k
200 REM решение вопроса о том, надо ли изображать
       данную грань
210 FOR k = 1 TO n
```

```
220 LET n1 = g(k,1): LET n2 = g(k,2): LET n3 = g(k,3):
    REM взяли номера трех соседних вершин
230 LET x1 = x(n1): LET x2 = x(n2): LET x3 = x(n3)
240 LET y1 = y(n1): LET y2 = y(n2): LET y3 = y(n3)
250 LET test = (x3-x1)*(y2-y1)-(x2-x1)*(y3-y1)
260 IF test < 0 THEN GO SUB 1000: REM грань изображаем
270 NEXT k
280 RETURN
300 REM блок управления от клавиатуры
310 LET a$=INKEY$
320 IF a$="" THEN GO TO 300
330 IF a$ =  "1" THEN GO SUB 400: REM уменьшение
340 IF a$ =  "5" THEN GO SUB 500: REM крен влево
350 IF a$ =  "6" THEN GO SUB 600: REM тангаж
360 IF a$ =  "7" THEN GO SUB 700: REM тангаж
370 IF a$ =  "8" THEN GO SUB 800: REM крен вправо
380 IF a$ =  "0" THEN GO SUB 900: REM увеличение
390 IF a$ =  "q" THEN GO SUB 9000:REM конец
399 GO TO 300

400 LET newmx=mx-dmx: IF newmx<1 THEN RETURN
410 LET newmy=my-dmx
420 LET newmz=mz-dmx
440 LET mx=newmx: LET my=newmy: LET mz=newmz
450 GO SUB 100: REM рисуем новую картинку
460 RETURN

500 REM Вращение вокруг оси X против часов
520 FOR k=1 TO m
530 LET ynew = w(k,2)*ct - w(k,3)*st
540 LET znew = w(k,2)*st + w(k,3)*ct
550 LET w(k,2)=ynew: LET w(k,3)=znew
560 NEXT k
570 GO SUB 100: REM Рисуем новую картинку
580 RETURN
```

```
600 REM Вращение вокруг оси Y по часам
620 FOR k=1 TO m
630 LET xnew = w(k,1)*ct - w(k,3)*st
640 LET znew = w(k,1)*st + w(k,3)*ct
650 LET w(k,1)=xnew: LET w(k,3)=znew
660 NEXT k
670 GO SUB 100: REM Рисуем новую картинку
680 RETURN

700 REM Вращение вокруг оси Y против часов
710 LET st=-st
720 GO SUB 600  :REM мы воспользуемся той же процедурой,
                    что и для вращения "по часам",
                    только изменив синус угла teta
                    на противоположный. Косинус менять
                    не надо, т.к. это функция четная.
730 LET st=-st
740 RETURN

800 REM Вращение вокруг оси X по часам
810 LET st=-st
820 GO SUB 500
830 LET st=-st
840 RETURN

900 LET newmx=mx+dmx: IF newmx>10 THEN RETURN
910 LET newmy=my+dmx
920 LET newmz=mz+dmx
940 LET mx=newmx: LET my=newmy: LET mz=newmz
950 GO SUB 100: REM рисуем новую картинку
960 RETURN

1000 REM изображение грани
1020 LET s = p(k)     : REM количество вершин в грани
1030 LET nf = g(k,1): REM номер первой вершины
1040 PLOT x(nf),y(nf)
1050 FOR f = 2 TO s  : REM цикл по остальным вершинам
```

```
1060 LET nv = g(k,f): REM номер вершины
1070 LET dx=x(nv)-x(nf): LET dy=y(nv)-y(nf)
1080 DRAW dx,dy
1090 LET nf=nv
1100 NEXT f
1110 REM строим последнее (замыкающее) ребро к 1-ой вершине.
1120 LET nv=g(k,1)
1130 LET dx=x(nv)-x(nf): LET dy=y(nv)-y(nf)
1140 DRAW dx,dy
1150 RETURN
```

3.7 Полезные приемы.

Предложенный здесь метод настолько прост, что требовать от него большого совершенства не приходится, Так, например, он не может четко работать с полыми объектами, он "не любит", когда встречаются грани, представляющие из себя невыпуклый многоугольник и т.п. Зато он работает настолько быстро, что небольшие погрешности могут быть и не очень критичными, а в динамичной игре и не очень заметными. Тем не менее, на несколько несложных, но важных моментов мы здесь все-таки укажем.

Принцип декомпозиции.

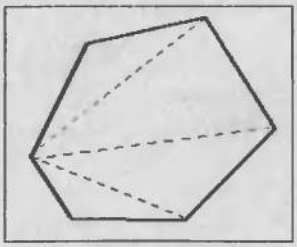


Рис. 45 Декомпозиция на треугольники

Этот принцип состоит в том, что грани трехмерного тела упрощаются до треугольников и вместо описания сложной грани мы вводим несколько описаний треугольников (см. рис.45).

Треугольники - достаточно удобные фигуры уже хотя бы тем, что они всегда выпуклы. Рутинную работу по декомпозиции граней своего объекта можно произвести

вручную, при подготовке данных для программы, а можно ее и не производить, а поручить компьютеру, ограничившись описанием граней в виде многоугольников. Правда, при таком подходе компьютер может оказаться иногда в тупиковой ситуации, например, на рис. 46 показано, как он может "выдумать" несуществующий треугольник. Методика борьбы с этим явлением состоит в том, что надо взять себе за правило проводить разбиение всегда из одной и той же вершины, например из той, которая значится в списке первой, а первой ставить ту вершину, которая Вам удобна. Для случая, представленного на рис.47 такое описание будет иметь вид: 6,1,2,3,4,5.

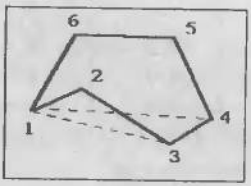


Рис. 46 Неправильная декомпозиция

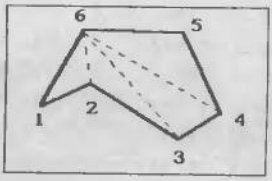


Рис. 47 Правильная декомпозиция

Для варианта, представленного на рис.48, Вам ни вручную, ни автоматически не удастся найти приемлемого способа декомпозиции из одной вершины. Здесь нужен специальный алгоритм. И он существует. Суть его следующая:

- берутся две соседние вершины многоугольника, например 1 и 8 (рис.48).

- берутся их соседние вершины. Для т.8 - это вершина 7, а для т.1 - это вершина 2.

- проводятся возможные диагонали. Для т.8 - это 8-2, а для т. 1 - это 1-7.

- из этих двух диагоналей выбирается кратчайшая. "Побеждает" диагональ 1-7, проведенная из вершина 1, а вершина 8 из дальнейшего рассмотрения выпадает.

- вместо выпавшей вершины "встает" ее соседка (вершина 7) и алгоритм повторяется для другой пары вершин 1 и 7.

- так далее, пока все вершины не будут исчерпаны.

Шаг	Рассматриваемые вершины	Рассматриваемые диагонали	Кратчайшая диагональ	Выпадающая вершина
1	8,1	1-7, 8-2	1-7	8
2	7,1	7-2, 1-6	7-2	1
3	7,2	7-3, 2-6	2-6	7
4	6,2	6-3, 2-5	6-3	2
5	6,3	6-4, 3-5	6-4	3

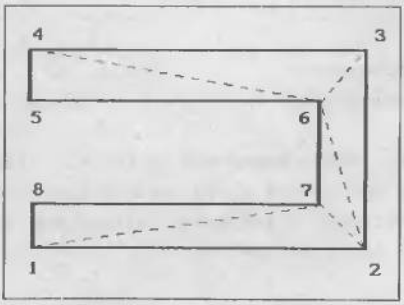


Рис. 48 Алгоритм декомпозиции сложной фигуры

Как видите, этот путь позволяет проводить автоматическую декомпозицию сложных фигур на треугольники. Если в программе очень много плоских фигур (граней тел) и они непрерывно меняют свою форму, то вручную делать декомпозицию на треугольники не

всегда возможно и пользуются таким или подобным алгоритмом.

Принцип фиктивных ребер. Это еще один полезный прием, который часто применяют в векторной графике.

С полыми телами при описанной выше технологии лучше дел не иметь, но иногда все-таки это может быть необходимым. Как же поступать, если такая задача стоит и обойти ее не удается, см. например объект на рис. 49.

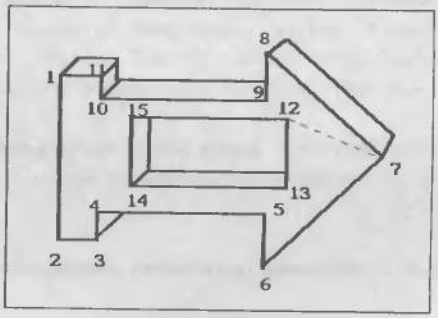


Рис. 49 Описание полого объекта.

Сложная грань имеет внутренний вырез. В этом случае вводится понятие фиктивных ребер. Они соединяют точку внешнего контура с точкой внутреннего. Как и настоящие ребра, они включаются в описание грани, но при работе программы на экране не изображаются. Чтобы в описании грани сразу было видно, какое ребро является фиктивным, а какое - нет, то при задании фиктивных ребер ставится знак "минус" перед вершиной. Тогда, например, верхняя грань тела, показанного на рис.49, будет описываться следующим образом:

```
1,2,3,4,5,6,7,-12,13,14,15,12,-7,8,9,10,11
```

Программа должна сама разобраться, что если стоит "минус",

то сторону рисовать не надо, а надо просто перейти к новой координате.

С помощью такого приема можно изображать не только внутренние полости, но и просто отрезки прямых, например декоративные. Обратите также внимание на особенность порядка обхода точек во внутреннем отверстии: 12,13,14,15. Как Вы видите, они обходятся по часовой стрелке, хоть и лежат на видимой поверхности. Дело в том, что наше правило обхода "против часов", которое мы ввели выше, было неполным. *Движение должно осуществляться таким образом, чтобы при движении от предыдущей точки к последующей, "тело" фигуры оставалось бы слева. Для внешних контуров это и дает обход против часовой стрелки, но для внутренних (хоть и видимых) движение оказывается "по часам".*

Использование фиктивных ребер может значительно расширить Ваши возможности по изображению трехмерных тел.

3.8 Изображение затененных поверхностей.

Итак, мы составили простой и быстро работающий алгоритм для решения вопроса видима грань или невидима. Но с этой задачей непосредственно связана еще одна задача. Очень интересно выглядят трехмерные тела, если у них есть грани, которые видимы, но находятся в тени. То есть, если в пространстве имеется источник света (а он в общем случае может перемещаться так же, как и Ваши объекты), то встает задача определения, а какие же грани находятся в тени? Если такие грани заштриховать, программа получит очень сильный толчок вперед.

Если алгоритм для определения видима грань или нет, Вам понятен, то задачу эту решить можно. Представьте себе, что глаз наблюдателя расположен там, где в данный момент находится источник света. Тогда те грани, которые видимы для такого "глаза" и будут освещены, а те, которые нет, - окажутся в тени и должны штриховаться.

Вернемся к нашей трехмерной проекции и предположим, что источник света находится справа от нас (рис.50).

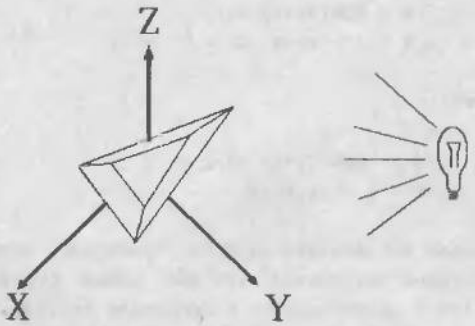


Рис. 50 Расположение источника света

Теперь, если мы будем смотреть на наш объект так, как будто бы наш глаз находится там же, где расположен источник света, то увидим следующую картину (рис. 51).

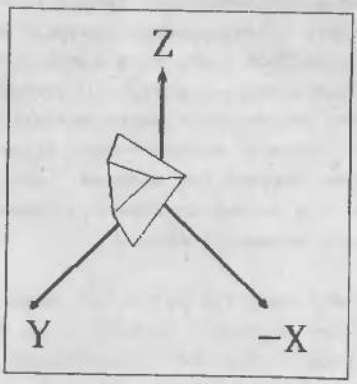


Рис. 51 Взгляд на объект со стороны источника света

Картина очень похожа на ту, что была и раньше, только как бы ось Y заняла место оси X, а ось "-X" заняла место оси Y. Это, конечно, очень частный случай, но для понимания сути вопроса нам годится и он. Теперь мы можем пересмотреть формулы (16), с помощью которых мы по трехмерным координатам объекта x,y,z находили плоские экранные координаты X,Y и переписать их для случая, когда глаз находится там же, где источник света.

Тогда вместо:

```
X = SQR(3)*(y-x)/2
Y = z - (y+x)/2
```

Мы получим:

```
X = -SQR(3)*(y+x)/2
Y = z - (y-x)/2                                    (17)
```

Теперь, если Вы захотите написать процедуру, которая определит, какие грани затеняются, то Вам можно будет заменить строки 120 и 130 в вышеприведенной программе (с.164), учитывая те изменения, которые произошли в формуле (17) и по той же самой методике найти грани, которые являются "невидимыми" с точки зрения источника света. ***Если грань одновременно видима для Вас и невидима для источника света, то она закрашивается (заполняется).***

Для "заполнения" грани можно воспользоваться технологией, которую мы рассматривали в книге "Элементарная графика" на стр.173. Единственной проблемой при этом может быть выбор точки, которая находится внутри заполняемого контура, с которой начинается процесс заполнения. Она обязательно должна оказаться внутри контура, иначе процесс "заливки" испортит весь экран. Вручную, конечно такую точку можно выбрать без проблем, но в динамической графике программа сама должна правильно находить такую точку, невзирая на то, какова сложность фигуры.

Если все грани у Вашего объекта представляют собой выпуклые многоугольники (а так и должно бы быть, особенно если Вы все грани упростили до треугольников с помощью декомпозиции), то это делается очень просто (см. рис. 52). Берем любую вершину, например A (xa,ya). Берем две соседние вершины, например B с координатами xb и yb и C (xc,yc). Кстати, найти их несложно, поскольку у нас имеется массив, описывающий каждую грань. Теперь мысленно соединяем B и C и на отрезке BC находим среднюю

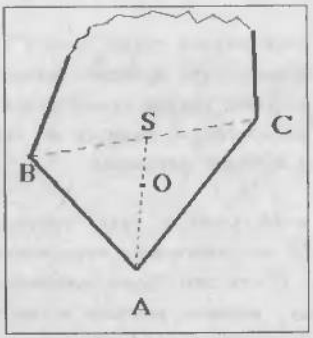


Рис. 52 Поиск точки внутри контура

точку S. Ее координаты:

```
xs=(xb+xc)/2;
ys=(yb+yc)/2
```

А теперь мысленно соединяем точку A с точкой S и на полученном отрезке тоже найдем середину, точку *O*. Ее координаты:

```
xo=xa/2+xb/4+xc/4;
yo=ya/2+yb/4+yc/4
```

Эта точка всегда будет лежать внутри нашего треугольника ABC, и, соответственно, внутри и выпуклого многоугольника ABCD... . Если же Ваш исходный многоугольник невыпуклый, то надо сначала исполнить декомпозицию его на треугольники.

В заключение отметим, что технология затушевывания замкнутых контуров работает довольно медленно и поэтому в быстроработающих программах применяется нечасто. Так, например, в авиаимитаторах и "звездных войнах" на "Спектруме" как правило не до этого, хотя в программе Starglider программистам удалось достичь таких скоростей, что кое-что они затушевали. В то же время, в имитаторах колесных и гусеничных машин теневые стороны холмов и сооружений затушевываются (заливаются) очень часто.

3.9 Самый общий подход к изображению невидимых линий.

Наиболее общий подход при решении проблемы сокрытия невидимых линий мы только наметим. Мало шансов, что кому-то удастся реализовать его на "Спектруме" в коммерческой программе, разве что только в виде эксперимента. Но, как известно, мир персональных компьютеров "Спектрумом" открывается, но не заканчивается и знать принципы общего подхода достаточно полезно.

Этот подход состоит в том, что каждая грань нашего тела разбивается на воображаемые треугольники (это принцип декомпозиции), после чего для каждой стороны каждой грани решается вопрос о том, как она расположена относительно каждого из таких треугольников. При этом применяется **принцип пирамиды**.

Предположим, что из какой-то k-ой грани мы взяли треугольник ABC и теперь решаем вопрос, как он расположен относительно j-ой стороны какой-то m-ой грани. Пусть это будет отрезок KL. Теперь построим бесконечную пирамиду, вершина которой находится в точке O, из которой производится наблюденне, а ребра которой проходят через вершины нашего треугольника (OA, OB и OC) - см. рис. 53.

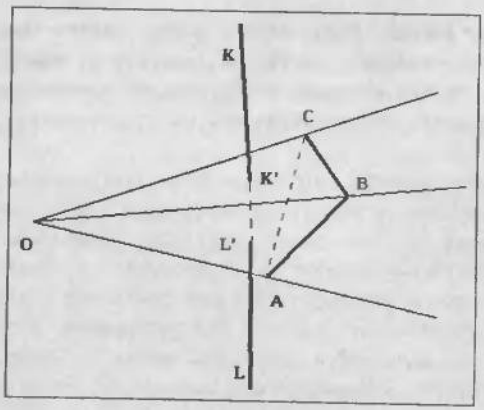


Рис. 53 Принцип пирамиды

Вопрос видимости отрезка KL, таким образом, сводится к тому, чтобы определить его положение относительно пирамиды OABC. И здесь возможны следующие случаи:

1. KL не проходит через пирамиду, значит для данного треугольника этот отрезок видим.

2. KL полностью лежит внутри пирамиды за треугольником (невидим);

3. KL полностью лежит внутри пирамиды перед треугольником (видим).

4. KL пересекает пирамиду перед треугольником (видим).

5. KL пересекает пирамиду за треугольником (видим, но частично). Зная координаты всех точек A,B,C,K,L,O несложно рассчитать, какая часть отрезка KL видима, а какая - нет.

6. KL пересекает и пирамиду и треугольник (видим частично).

Остается только выяснить, какой же случай из шести возможных произошел у нас. Это делается с помощью серии из нескольких тестов. Они проводятся с помощью математического аппарата из аналитической геометрии. Тесты эти стандартные, все исходные данные для них в программе уже есть (это координаты вершин тела) и потому оперативную память они почти не расходуют, но каковы затраты времени, Вы можете догадаться. В принципе, считают, что зависимость здесь квадратичная, т.е. если количество ребер в Вашем объекте увеличить в два раза, то скорость работы уменьшится в четыре раза. Конечно, существуют разнообразные приемы для повышения производительности, но это уже ухищрения программистов, свои для каждого конкретного случая.

4. ПОЛЕЗНЫЕ СОВЕТЫ

Если Вы приступаете к созданию собственной аннмации на компьютере, то Вам не помешают несколько полезных советов. Конечно все те хитрости и приемы, которые отработали профессионалы за более чем полвека мультипликации, нельзя вместить на страницы никакой книги, но кое-что мы Вам все-таки подскажем. И прежде всего порекомендуем повнимательнее смотреть мультфильмы по телевизору и компьютерную графику в игровых программах. Вас ожидают удивительные открытия, которые Вы сможете с успехом использовать в собственных работах.

1. Значение головы.

Голова, как известно, очень важная часть тела и не только потому, что в ней размещаются мозги. Обратите вниманне на персонажей практически любого мультика. Голова у них почти всегда преувеличена. Если обычно у человека ширина головы равна примерно половине ширины плеч (посмотрите на себя в зеркало), то в мультфильмах и в компьютерных играх она достигает ширины плеч, а бывает и еще больше.

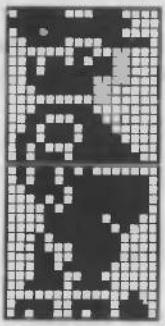


Рис. 54 Фрагмент из программы Everyone's a Wally

Это сделано не просто так, а для того, чтобы дать возможность выразить массу чувств и настроений через мимику лица. Это же помогает и отличать персонажей друг от друга. Посмотрите на Wally, героя игр "Three Weeks in Paradise", "Pijamarama" и др (рис. 54). У него голова занимает почти половину всего изображения.

А про всенародно любимого DIZZY мы и говорить не будем. Он такой забавный и веселый в какой-то степени потому, что по форме похож на яйцо и размер головы у него практически равен размеру всего тела.

2. Значение рук.

Руки после головы по важности идут на втором месте. Это тоже очень выразительный объект. Разведя их в стороны, можно выразить недоумение. Подняв их вверх, можно выразить радость. "Опустить руки" - это примерно то же самое, что упасть духом, согласно народному присловью.

Быстрыми движениями согнутых в локтях рук выражают спешку. Руки вытягивают вперед при столкновении с препятствием, руки отбрасывают назад при падении на спину.

3. Значение туловища.

Туловище для анимации - самый малоперспективный объект. При скромных возможностях нашего экранного разрешения из него в смысле динамики и выразительности ничего путного не выжмешь. Поэтому его надо разрабатывать как статический элемент, не изменяющий своей формы в процессе перемещений. Зато на нем могут быть использованы какие-либо декоративные и орнаментальные украшения.

4. Значение ног.

Ноги, в отличие от туловища, все-таки динамический объект, но, в отличие от рук и головы, маловыразительный. Парадоксально, но факт, что хоть все живое и перемещается с помощью ног, тем не менее им уделяется гораздо меньше внимания, чем рукам и голове.

Единственный случай, когда ноги становятся выразительными, это как раз тогда, когда герой - неподвижен. Вспомните, как Рокфорд нетерпеливо потопывает ногой в программе Boulder Dash.

С ногами связан ряд профессиональных приемов. Во-первых, ноги часто вообще не изображают, пряча их под длинным баллахоном. Сравните, например, игры "Knight Lore" и "Pentagram" одной и той же фирмы Ultimate. Вы увидите, что в первой игре ноги у героя хоть и слабо, но работают, а во второй они не видны. Упрощение позволило чуть приподнять быстродействие и в результате монстров во второй игре стало побольше и ведут они себя агрессивнее (динамичнее).

Во-вторых, если ноги и изображают, то довольно простыми средствами. Все-равно выразительности в них мало и при перемещении героев по экрану глаз пользователя за ними не следит. Ему достаточно, что они там есть. Практически, что Вы нарисуете дюжину спрайтов, описывающих фазы движения ног, что ограничитесь только парой, на конечный результат это мало повлияет.

В третьих, от ног иногда отказываются совсем для того, чтобы передать особую динамику движения. Американская студия "Ханна-Барбера" впервые применила трюк, который Вы все безусловно видели - когда герой начинает бежать, у него вместо ног рисуется облачко пыли.

С ногами же связан еще один трюк. Быстродвижущимся персонажам пририсовывают... третью ногу. Это также создает иллюзию динамичности и скорости.

5. Задний план.

Вопрос выбора заднего плана очень непрост. Хочется, чтобы он был красивым и, в то же время, ему необходимо быть простым. Если уж его не сделать красивым, то что же тогда будет красивым, ведь разукрашивать спрайты не очень-то удается? Если его

сделать сложным, то во-первых могут быть проблемы с "клэшингом" цветовых атрибутов, а во-вторых, он начнет перетягивать на себя внимание пользователя, а это понизит выразительность динамических персонажей.

Таким образом, Вам нужна золотая середина. К счастью, на "Спектруме" далеко не разбежишься и слишком сложный задний план нарисовать не получится. Но Вам важно для себя понять, что к этому и не надо стремиться, так как это серьезно усложняет Вашу задачу и при том не ссобо нужно. Здесь можно порекомендовать во-первых, использовать для заднего плана крупную графику и, во-вторых, сделать ее регулярной (черепица крыш, кирпичные стены, деревянные и бетонные заборы, деревья и кусты могут изображаться на разных экранах из одинаковых элементов). Поскольку глаз быстро привыкает к таким регулярным формам, то перестает на них реагировать, остается только общее благоприятное художественное впечатление. Например, после того, как Вы выйдете из парка, у Вас останется ощущение того, что Вы действительно были в парке, а сколько там было деревьев, каких и как они были нарисованы, Вы и не вспомните. Это особенность компьютерной анимации - все внимание уделяется только движущемуся объекту. Прочее - антураж.

Если же Вы начнете по-разному изображать деревья разных пород и размеров, то при переходе от экрана к экрану Ваше внимание начнет отвлекаться на задний план и пострадает художественное впечатление от анимации. Одним словом, чем проще средства, тем лучше результат.

6. Двигать только то, что необходимо и не больше.

Представьте себе, что Ваш автобус едет по дороге. Спросим себя: "Что нам надо двигать на экране?" Вроде бы, естественно, автобус. Но не тут-то было. Очень и очень часто двигают на экране совсем не то, что движется на самом деле, а нечто другое. Посмотрите на рис.55 . Можно, конечно, здесь было бы двигать и

автобус, но что с ним делать, когда он дойдет до края экрана? Это не вполне понятно. Во-первых, можно в этот момент подзагрузить новый экран и продолжить в том же духе. Но это неудобно, поскольку подгрузка нового экрана занимает большое время, возникает срыв в динамике игры. А потом снова та же проблема, когда мы дойдем до края нового экрана.

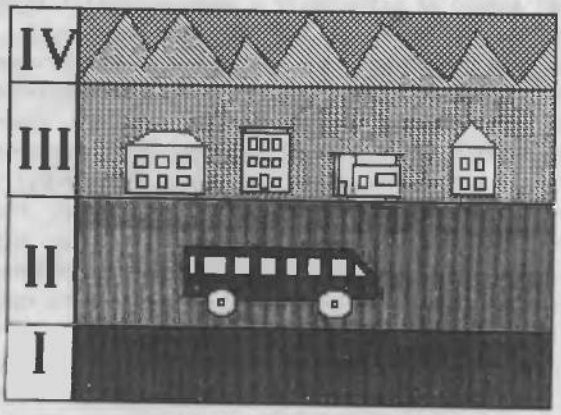


Рис. 55 Скроллинг в окнах I и III

Во-вторых, можно бы и не подгружать новый экран, но тогда пришлось бы делать горизонтальный скроллинг экрана, когда справа из-за границы выступает новый пейзаж, а слева он скрывается за кадром. Это, конечно возможно, но скроллировать целый экран - работа очень медленная во-первых, а во-вторых эти экраны страшно расходуют оперативную память Вашего компьютера.

Так что же делать? Ответ очень прост. Всегда надо подумать, а нельзя ли двигать по экрану как можно меньше объектов. Если вернуться к нашему примеру с автобусом на рис. 55, то можно увидеть, что картина как бы представляет из себя четыре плана. Задний план - горы (IV). Средний план - придорожные постройки (III). Ближний план - газон и автомобиль на шоссе (I,II). Так вот, для того, чтобы передать движение автобуса, достаточно

двигать только средний план - полосу экрана (окно III) высотой в 3-5 знакомест, на которой изображены придорожные постройки и полосу I - газон. Если их еще запустить с разными скоростями, т.е. полосу I двигать в два раза быстрее, чем полосу III - возникнет совершенно устойчивый эффект глубины пространства.

Все для этого у Вас есть. Окна III и I можно запустить на скроллинг под управлением от прерываний 2-го рода (процедура FN o(....), рассмотренная выше), а спрайт автобуса - перемещать в небольших пределах вправо-влево под управлением от клавиатуры (процедура FN k(...), тоже рассмотренная выше).

Это намного эффективнее в смысле скорости и экономичнее в смысле памяти по сравнению с передвижениями всего экрана.

А если теперь вместо автобуса поставить бронетранспортер с крупнокалиберным пулеметом, а в придорожных постройках на среднем плане разместить засевших боевиков, обозначив их вспышками огнестрельного оружия (самих их рисовать не надо), то теперь можно не просто ехать по шоссе, а еще и лихо постреливать по сторонам - готовая игра.

7. Движение к наблюдателю.

До сих пор мы говорили о движении объектов в плоскости экрана. Но, возможно, Вам захочется сделать такую анимацию, когда объекты двигаются к наблюдателю (от наблюдателя) и при этом увеличиваются (уменьшаются) в размерах. Должны Вас сразу разочаровать - средствами блочной графики это сделать невозможно, а в растровой графике - очень сложно. Решение своей задачи ищите в средствах векторной графики, а от применения цвета воздержитесь.

8. Преувеличение.

Утрирование изображений - основной прием в анимации. Мы уже говорили, что утрируют размеры головы. Но можно и нужно утрировать и другие важные в смысле выразительности детали.

Так, например, для транспортных средств (автомобили, поезда, тележки) практически всегда утрируют две вещи - колеса и фары (в зависимости откуда Вы смотрите). Автомобиль с колесами от трактора конечно меньше похож на настоящий автомобиль, но зато лучше привлекает взгляд и лучше производит впечатление автомобиля.

9. Динамика в статике.

Нечасто удается прибегать к высокой динамичности своей графики. Это требует сложной работы художника, быстродействия от компьютера и рассеивает внимание играющего. Поэтому всегда, в любой сцене ищите как можно заменить динамику статикой. А делается это просто. В соответствии с законами физики динамические нагрузки, действующие на упругие тела, вызывают в них деформации. Вот утрируя (преувеличивая эти деформации) и передают динамику игры.

Рассмотрим пример. Вам нужно показать, как футболист наносит удар по мячу, после чего мяч летит, ударяется о кирпичную стену и отскакивает назад.

Вы можете сколько хотите прорисовывать разбег футболиста, показывать кадры удара и полета мяча, но все это будет очень сложно и все равно неестественно. Вам больше даст крупный план, в котором Вы покажете, как мяч прогибается от удара ногой, как он сплющивается при ударе о стену и потом отлетает. Именно момент неестественного сжатия мяча и передаст всю динамику ситуации

Вы не раз видели в мультфильмах, как автомобили растягиваются в момеит старта и как они сжимаются при торможении. Если на экране у Вас джентльмен прогуливается с собачкой, то динамику этого процесса можно показать неестественным растяжением поводка. Если на Вашего героя обрушился удар дубинкой по голове, можете показать, как он сминается в лепешку, а потом снова из нее возникает.

Фазы "смятия" различных предметов и других деформаций - одни из самых любимых объектов компьютерных аниматоров.

П Р И Л О Ж И Н И Е

Образцы спрайтов для самостоятельного набора

Для того, чтобы Вам было с чем позэкспериментировать мы привели в этой книге несколько десятков образцов шаблонов спрайтов для самостоятельного набора. Используя эти образцы, Вы и сами сможете подготовить свои спрайты для своих собственных программ.

Для ввода шаблонов в память можете воспользоваться следующей нехитрой программой (Загрузчик-1). Если же Вы перейдете от первых экспериментов к интенсивному программированию в динамической графике, то Вам потребуется более мощное средство для создания, редактирования и сохранения спрайтов. Думаем, что Вы без труда найдете как его создать или воспользуетесь более развитым загрузчиком, приведенным здесь же (Загрузчик-2). Он позволит не только вводить данные по спрайту, но и просматривать, что же у Вас получилось и вносить небольшие исправления.

Загрузчик спрайтов-1.

Загрузчик загружает Ваши спрайты, начиная с адреса 54600, принятого в данной книге в качестве базового адреса таблицы спрайтов.

```
10 LET adr=54600
20 FOR i=1 TO 63
30 INPUT "Give me a byte. ";b
40 POKE adr,b
50 PRINT adr,b
60 LET adr=adr+1
70 NEXT i
80 INPUT "Is it all? y/n? ";n$
```

```
 90 IF n$="n" THEN GO TO 20
100 IF n$<>"y" THEN GO TO 80
110 STOP
```

Загрузчик спрайтов-2.

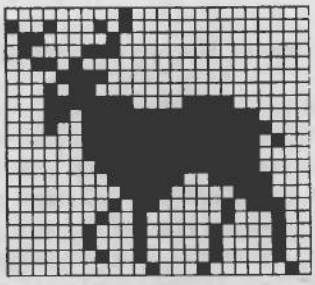
Примечание: подчеркнутые символы A,B,C...I вводятся в графическом режиме (курсор G).

```
 10 LET adr=54600
 20 FOR n=1 TO 10              :REM номера спрайтов
 30 GO SUB 400
 40 FOR k=1 TO 63              :REM 63 байта в спрайте
 50 INPUT "Give me a byte ";b$
 60 IF b$="e" THEN GO TO 160 :REM если Вы при вводе спрайта
                               ошибетесь и захотите внести
                               изменение, нажмите клавишу E.
 70 IF b$="f" THEN STOP        :REM чтобы закончить работу,
                               нажмите клавишу F.
 80 IF b$="" THEN GO TO 50
 90 IF CODE b$(1)<48 OR CODE b$(1)>57 THEN GO TO 50
100 LET b=VAL(b$)
110 IF b>255 OR b<0 THEN GO TO 50
120 LET address=adr+63*(n-1)+k-1
130 POKE address,b
140 GO SUB 500
150 GO SUB 600
160 IF b$="e" THEN GO SUB 700
170 NEXT k
180 NEXT n

400 REM  Подпрограмма очищает первые 9 символов графики
         пользователя от A до I.
410 FOR i=1 TO 72
420 POKE USR "A"+i-1,0
```

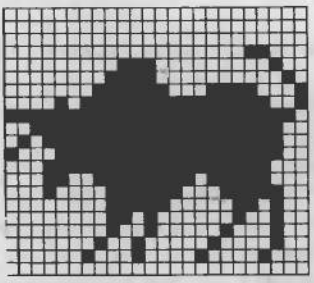
```
430 NEXT i
440 RETURN

500 REM  Подпрограмма печатает на экране результат Вашего
        ввода.
510 PRINT AT 0,0;"          <32 пробела>                  "
520 PRINT AT 0,0;"SPRITE N ";n
530 PRINT AT 0,14;"BYTE N ";k
540 PRINT AT 0,24;"="
550 PRINT AT 0,26;b
560 RETURN

600 REM Подпрограмма рисует Ваш спрайт с помощью символов
        графики пользователя.
610 IF k<22 THEN POKE USR "A"+k-1,b: GO TO 640
620 IF k<43 THEN POKE USR "D"+k-22,b: GO TO 640
630 POKE USR "G"+k-43,b
640 PRINT AT 10,15;"ADG"
650 PRINT AT 11,15;"BEH"
660 PRINT AT 12,15;"CFI"
670 RETURN

700 REM Подпрограмма позволяет вернуться к ранее введенному
        байту в случае ошибки и повторить ввод.
710 INPUT "Give me a number of byte (1...63) ";nb
720 LET k=nb-1
730 LET b$="0"
740 RETURN
```

Данные для ввода спрайта

```
128,145,096,067,052,015,022,031
027,019,003,003,001,000,000,000
001,002,002,001,000,064,064,128
128,000,000,000,195,255,255,255
255,255,252,120,176,032,032,032
032,033,000,000,000,000,000,000
000,192,240,248,244,240,240,248
056,092,068,132,132,132,002
```



Олень



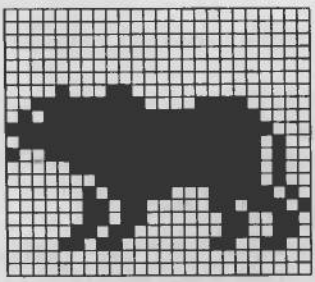
Буйвол

Данные для ввода спрайта

```
000,000,000,000,000,000,001,011
255,063,095,143,031,025,016,000
000,000,001,002,000,000,000,000
000,112,240,248,255,255,255,255
255,255,254,252,248,208,144,032
033,000,000,000,000,024,004,002
241,249,254,252,252,252,248,248
124,052,036,068,132,008,000
```

Данные для ввода спрайта

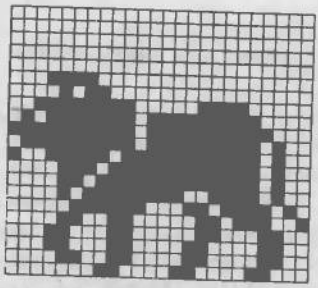
```
000,000,000,000,000,000,008,063
095,255,127,159,007,001,002,003
003,006,013,000,000,000,000,000
000,000,000,192,225,255,255,255
025,255,255,248,096,096,192,128
000,000,000,000,000,000,000,000
000,224,240,248,244,244,244,244
250,221,102,102,204,000,000
```



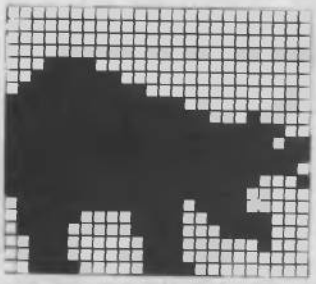
Тигр

Данные для ввода спрайта

```
000,000,000,000,000,030,011,063
095,255,127,159,014,013,011,007
012,024,024,012,001,000,000,000
000,000,000,000,193,223,223,223
012,255,255,252,227,193,193,195
195,134,000,000,000,000,000,000
000,224,240,248,244,244,244,244
244,122,185,140,012,012,024
```



Лев



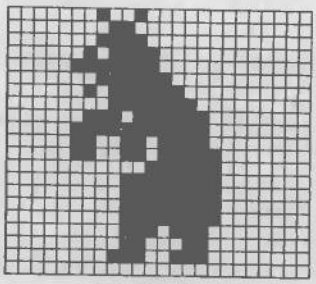
Белый медведь

Данные для ввода спрайта

```
000,000,000,000,030,063,127,255
255,255,255,255,255,255,255,127
121,120,056,056,060,000,000,000
000,000,128,224,252,255,255,255
251,255,255,255,253,060,060,028
028,030,000,000,000,000,000,000
000,000,016,252,251,255,254,241
224,224,248,120,008,000,000
```

Данные для ввода спрайта

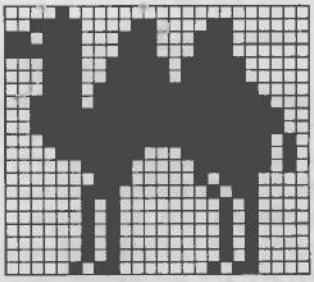
```
001,000,001,002,007,001,002,000
003,007,006,006,000,000,000,000
000,000,000,000,000,032,192,224
240,240,248,252,254,191,255,111
111,031,127,127,127,127,119,099
099,231,000,000,000,000,000,000
000,000,000,000,000,128,128,128
128,128,128,000,000,000,000
```



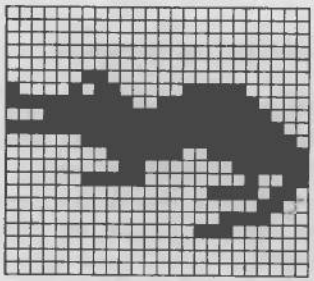
Бурый медведь

Данные для ввода спрайта

```
008,060,220,252,028,028,029,061
063,063,031,015,003,001,003,002
002,002,002,002,005,000,048,113
113,251,251,255,255,255,255,255
227,193,193,128,128,128,128,128
128,128,000,192,192,192,224,224
240,248,252,252,250,250,250,112
176,080,080,080,080,080,176
```



Верблюд



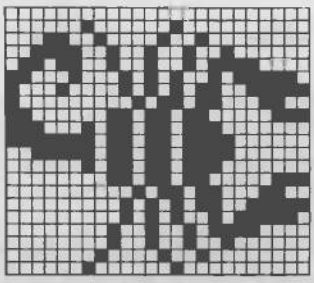
Крокодил

Данные для ввода спрайта

```
000,000,000,000,000,003,133,255
031,255,003,000,000,003,000,000
000,000,000,000,000,000,000,000
000,000,000,131,207,255,255,255
252,096,224,000,000,000,001,000
000,000,000,000,000,000,000,000
224,240,248,252,254,255,063,019
246,012,120,192,000,000,000
```

Данные для ьвода спрайта

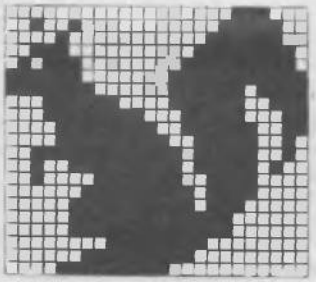
```
000,002,025,060,108,196,136,128
192,226,126,062,002,000,000,000
000,000,000,001,002,000,004,009
146,146,084,085,045,219,219,219
219,219,219,045,085,084,146,146
009,004,000,000,000,064,240,190
159,140,007,128,192,192,128,007
140,159,190,240,064,000,000
```



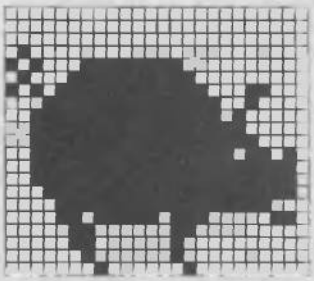
Скорпион

Данные для ввода спрайта

```
000,036,008,056,092,252,254,031
031,015,015,063,039,033,003,007
007,007,000,003,015,000,000,000
001,003,007,007,135,227,243,249
253,253,252,254,254,255,255,255
254,248,056,124,252,254,254,255
239,231,227,243,242,242,244,240
240,224,192,192,000,000,000
```



Белка



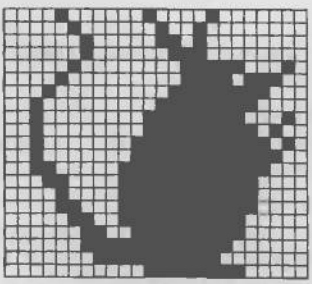
Поросенок

Данные для ввода спрайта

```
000,000,000,064,163,071,175,031
063,063,063,063,063,063,031,015
013,006,006,002,001,000,000,000
000,252,254,255,255,255,255,255
255,255,255,255,255,247,006,004
004,002,000,000,000,000,000,000
024,144,160,224,240,218,254,254
254,248,006,000,000,000,000
```

Данные для ввода спрайта

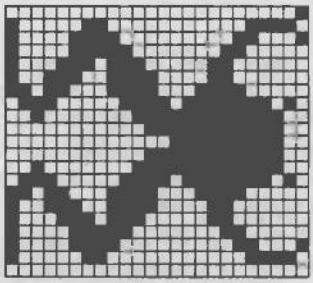
```
008,004,002,002,004,008,016,032
032,032,032,032,048,024,008,012
006,003,003,001,000,016,009,013
007,003,003,007,015,031,031,063
063,127,127,127,127,127,063,255
255,031,128,000,000,000,226,220
248,240,226,244,250,252,248,240
240,240,224,224,192,128,224
```



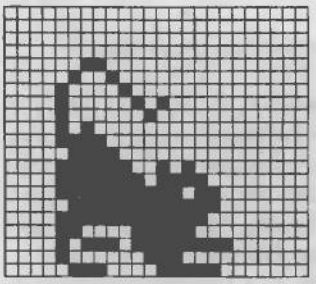
Мышка

Данные для ввода спрайта

```
000,131,132,135,141,156,216,112
032,000,000,000,032,112,216,156
000,135,135,131,000,000,000,128
192,060,240,241,123,063,031,007
031,063,123,241,240,224,192,001
000,000,001,030,049,096,192,192
013,250,252,252,252,252,252,025
224,192,192,096,049,030,001
```



Лягушка



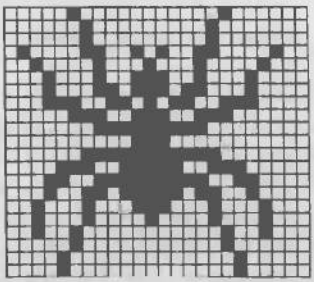
Котенок

Данные для ввода спрайта

```
000,000,000,000,003,004,008,008
008,010,015,007,015,015,015,007
015,012,011,008,007,000,000,000
000,000,128,064,040,016,000,000
128,202,239,253,255,255,062,191
028,015,000,000,000,000,000,000
000,000,000,000,000,000,000,000
128,128,000,000,192,000,128
```

Данные для ввода спрайта

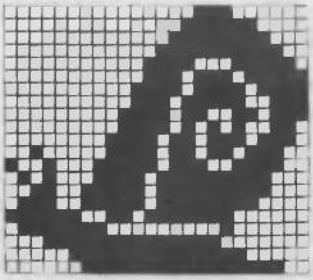
```
004,002,002,067,033,017,017,028
007,001,000,003,014,025,019,034
034,036,004,008,008,000,000,000
041,017,057,187,214,057,255,056
255,124,255,125,056,016,000,000
000,000,064,128,128,132,008,016
016,112,192,000,000,128,224,048
144,136,136,072,064,032,032
```



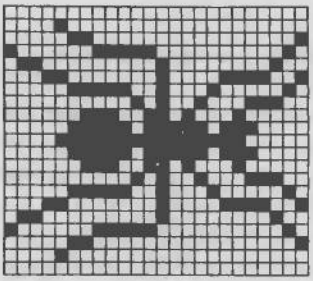
Паук

Данные для ввода спрайта

```
000,000,000,000,000,000,000,000
000,000,000,032,145,081,051,115
252,127,031,007,003,001,003,007
015,030,029,061,059,123,122,246
246,247,239,239,239,223,000,255
255,255,192,240,248,252,062,223
239,231,055,166,238,220,060,248
248,240,192,064,128,224,252
```



Улитка



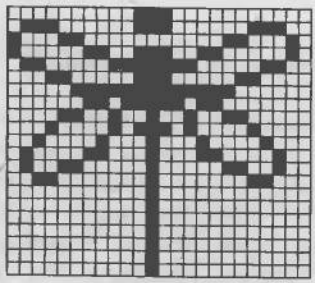
Муравей

Данные для ввода спрайта

```
000,008,006,129,096,024,007,000
003,007,015,007,003,000,007,024
096,129,006,008,000,000,000,000
240,008,008,200,041,154,223,255
223,154,041,200,008,008,240,000
000,000,000,000,001,002,004,121
130,028,032,112,224,112,032,028
130,121,004,002,001,000,000
```

Данные для ввода спрайта

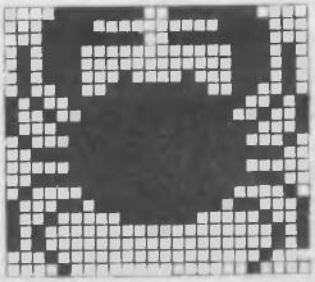
```
000,112,140,131,096,024,007,003
012,016,033,067,076,048,000,000
000,000,000,000,000,056,056,000
057,254,056,255,125,146,186,017
017,016,016,016,016,016,016,016
016,016,000,028,098,130,012,048
192,128,096,016,008,132,100,024
000,000,000,000,000,000,000
```



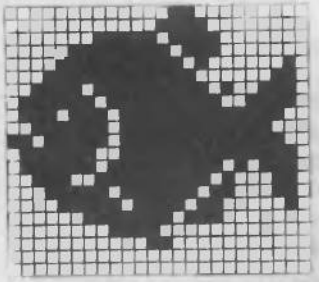
Стрекоза

Данные для ввода спрайта

```
000,001,001,001,000,000,000,015
031,063,063,127,127,127,255,255
255,195,129,001,007,224,223,240
255,227,112,056,158,207,247,255
255,255,255,255,255,207,135,000
024,244,000.255,000,255,254,254
028,000,128,192,224,240,240,240
240,240,224,192,000,000,000
```



Краб



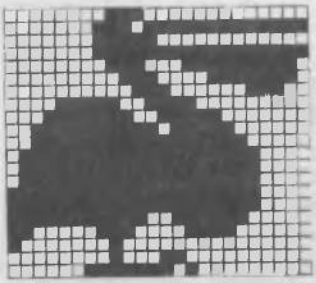
Пиранья

Данные для ввода спрайта

```
000,000,000,000,000,000,000,000
000,000,000,016,040,048,016,008
121,138,243,032,081,000,000,000
000,000,000,000,000,000,000,000
067,172,200,079,036,227,033,195
130,067,000,000,000,048,108,112
112,056,028,014,254,018,034,066
132,008,240,176,016,024,012
```

Данные для ввода спрайта

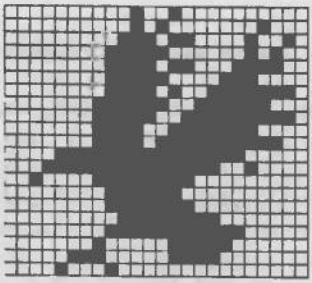
```
015,030,031,092,092,092,076,167
147,143,071,063,007,063,067,141
144,160,160,016,008,231,000,231
000,000,066,255,255,255,255,255
255,255,255,255,255,126,000,000
000,000,240,120,248,058,058,050
034,229,201,241,226,252,224,252
194,177,009,005,005,008,016
```



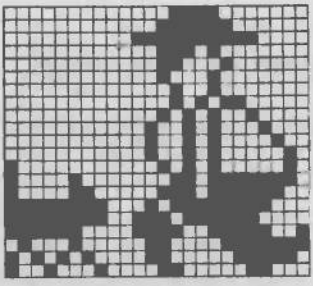
Пеликан

Данные для ввода спрайта

```
000,000,000,000,001,002,098,224
240,127,127,127,240,224,098,002
001,000,000,000,000,004,009,018
148,085,085,086,046,127,255,255
255,127,046,086,085,085,148,018
009,004,192,000,096,158,048,062
113,198,152,224,192,224,152,198
113,062,048,158,096,000,192
```



Орел



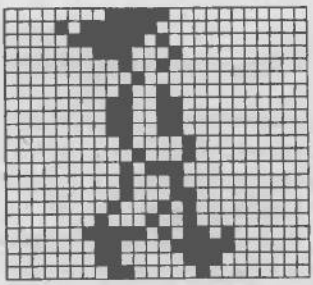
Шериф

Данные для ввода спрайта

```
000,000,000,000,000,000,000,000
000,000,000,000,128,132,134,255
255,252,068,170,145,007,015,063
014,012,008,004,005,010,018,018
018,010,006,014,023,059,060,024
024,012,128,192,240,128,064,128
128,096,144,136,132,130,130,254
252,248,240,249,127,062,024
```

Данные для ввода спрайта

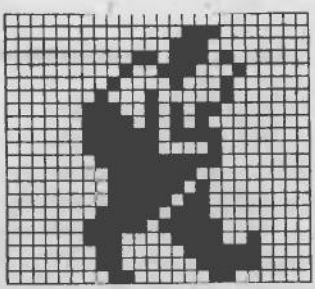
```
000,011,007,001,000,000,000,000
000,000,000,000,000,000,000,000
001,003,001,001,000,248,240,232
196,072,032,072,204,204,204,066
034,092,068,070,138,018,235,135
131,193,000,000,000,000,000,000
000,000,000,000,000,000,000,000
000,000,000,064,192,128,000
```



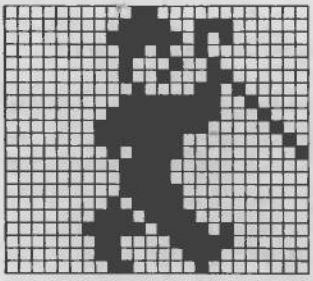
Гном

Данные для ввода спрайта

```
000,000,000,000,000,000,001,003
003,003,003,001,000,000,000,001
003,003,001,001,000,000,003,007
055,078,132,016,149,213,247,240
255,254,125,251,247,239,135,131
129,192,000,128,000,064,032,064
064,128,000,128,128,128,000,000
000,000,128,144,240,240,192
```



Горбун



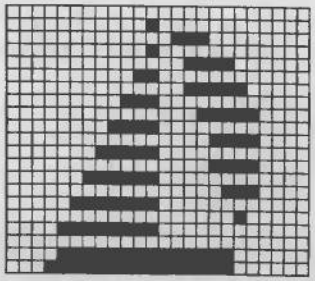
Чарли Чаплин

Данные для ввода спрайта

```
000,000,000,000,000,000,000,000
000,001,001,000,000,000,000,000
001,003,001,001,000,048,121,253
105,068,072.033,127,255,255,252
156,120,120.124,188,222,239,135
131,193,000,192,064,000,128,192
160,144,136,004,002,001,000,000
000,000,000,064,192,128,000
```

Данные для ввода спрайта

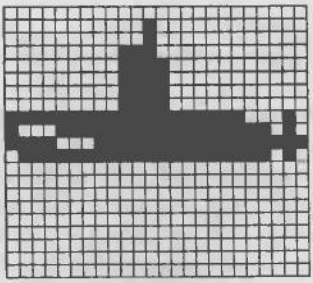
```
000,000,000,000,000,000,000,000
000,000,000,001,000,003,000,007
000,015,000,015,031,000,016,007
016,003,048,003,112,001,240,000
240,000,240,000,240,000,240,000
255,255,000,000,000,000,192,000
224,000,240,000,240,000,240,000
112,000,032,000,000,192,192
```



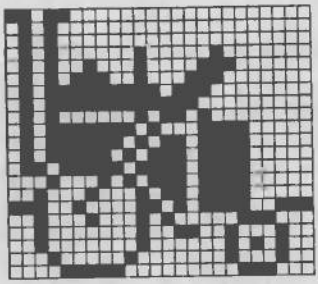
Яхта

Данные для ввода спрайта

```
000,000,000,000,000,000,000,000
255,143,241,127,000,000,000,000
000,000,000,000,000,000,016,016
112,120,120,120,120,255,255,255
255,000,000,000,000,000,000,000
000,000,000,000,000,000,000,000
000,000,226,250,255,250,000,000
000,000,000,000,000,000,000
```



Подводная лодка



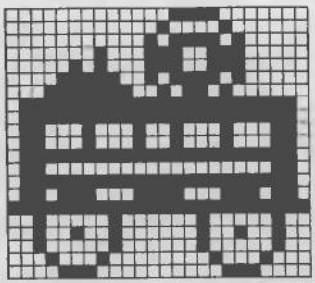
Паровоз

Данные для ввода спрайта

```
216,080,080,080,082,087,095,095
088,095,095,095,111,017,034,228
034,032,032,016,015,000,000,000
032,033,035,247,254,041,213,173
093,189,093,045,053,040,038,032
064,128,000,000,000,128,192,128
000,000,000,224,224,224,224,224
224,231,056,068,084,068,056
```

Данные для ввода спрайта

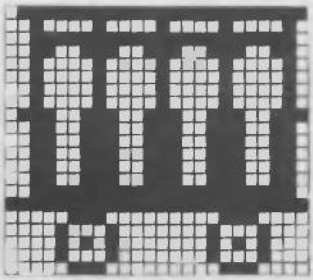
```
000,000,000,001,005,015,031,127
127,098,098,127,096,127,110,255
032,036,032,017,014,007,011,023
028,028,151,203,255,255,036,036
255,000,255,060,255,129,129,129
000,000,128,064,160,224,224,160
064,254,254,070,070,254,006,254
118,255,004,036,004,136,112
```



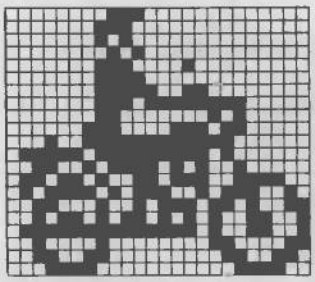
Тендер

Данные для ввода спрайта

```
127,033,063,051,033,033,033,033
115,051,051,051,051,051,063,255
007,008,010,008,007,255,008,255
156,008,008,008,008,156,156,156
156,156,156,255,255,000,128,128
128,000,255,066,254,230,066,066
066,066,231,230,230,230,230,230
254,255,112,136,168,136,112
```



Вагон



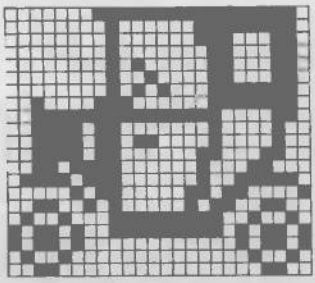
Мотоциклист

Данные для ввода спрайта

```
000,001,001,000,001,001,001,001
001,003,019,125,050,109,094,049
033,127,033,051,030,224,192,064
160,194,224,243,223,128,135,255
255,245,055,185,046,106,254,000
000,000,000,000,000,000,000,000
000,224,160,096,192,064,032,060
114,211,153,153,195,102,060
```

Данные для ввода спрайта

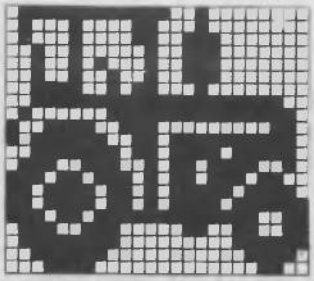
```
001,000,000,000,000,000,000,032
063,061,061,061,055,123,005,050
072,180,180,072,048,255,129,129
128,160,144,168,192,255,128,176
129,129,129,131,130,255,255,000
000,000,254,254,198,198,198,198
254,254,134,132,132,140,024,062
097.204,146,173,045,018,012
```



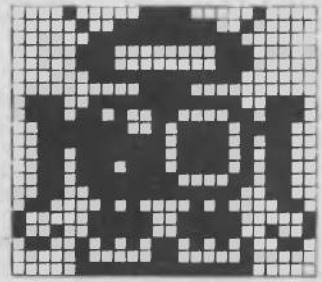
Старинный автомобиль

Данные для ввода спрайта

```
127,068,068,036,037,036,060,063
064,158,191,063,115,237,222,222
237,115,127,063,030,249,049,051
019,019,147,177,255,255,112,055
183,150,214,215,215,255,131,128
000,000,000,000,128,128,128,128
032,252,254,006,254,190,242,236
222,191,243,051,063,030,012
```



Трактор



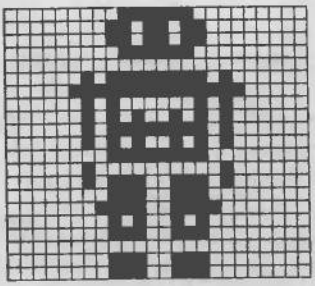
Робот-1

Данные для ввода спрайта

```
003,004,003,007,007,003,000,115
118,063,063,055,055,055,051,097
131,147,101,004,007,060,255,255
000,000,255,060,255,152,151,247
247,119,248,255,066,231,231,090
024,255,016,032,192,224,224,192
000,206,110,188,188,172,172,108
204,134,193,201,166,032,224
```

Данные для ввода спрайта

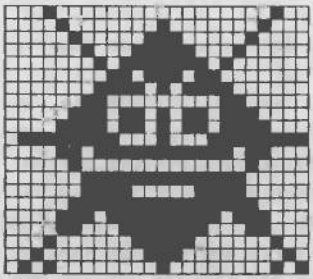
```
000,000,000,000,000,002,007,002
002,002,002,000,002,002,000,001
000,000,000,000,000,126,219,219
126,000,255,255,129,165,255,165
255,000,231,231,231,165,231,000
231,231,000,000,000,000,000,064
224,064,064,064,064,000,064,064
000,128,000,000,000,000,000
```



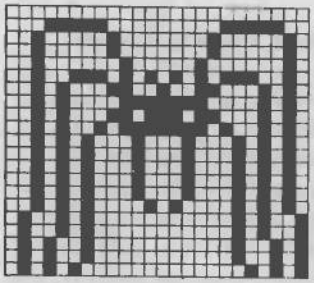
Робот-2

Данные для ввода спрайта

```
016,008,004,002,001,000,001,003
007,015,127,013,004,003,003,007
014,012,016,032,080,000,008,028
062,127,221,235,000,107,107,136
255,000,255,193,255,255,127,062
028,008,004,008,016,032,064,128
192,096,112,120,255,216,016,224
224,240,184,024,004,002,005
```



Инопланетянин



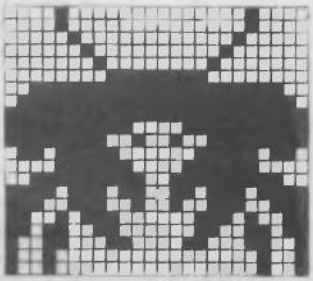
Паук-привидение

Данные для ввода спрайта

```
000,031,032,032,032,039,040,040
040,041,042,042,042,042,042,042
074,074,082,082,084,000,000,128
128,065,085,201,127,221,127,034
034,034,034,020,000,000,000,000
000,000,000,124,130,130,002,114
138,010,138,074,042,042,042,042
042,042,041,041,037,037,021
```

Данные для ввода спрайта

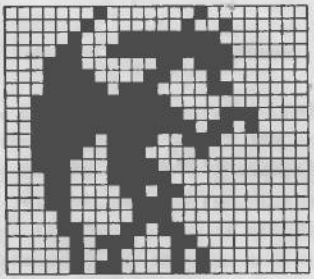
```
008,008,004,002,001,000,063,127
255,255,255,111,015,063,247,231
203,217,152,144,144,000,000,000
000,000,255,255,255,255,195,036
129,195,231,102,036,129,129,195
000,000,016,016,032,064,128,000
252,254,255,255,255,246,240,252
239,231,211,155,025,009,009
```



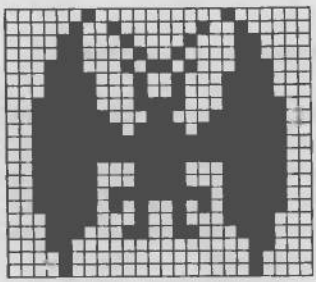
Космокраб

Данные для ввода спрайта

```
001,002,004,012,028,030,031,063
063,063,062,060,056,024,024,024
008,012,005,004,005,001,002,127
255,013,024,112,248,252,254,243
225,240,248,108,056,056,108,198
133,133,000,064,128,192,192,128
192,000,112,208,128,000,000,000
000,000,000,000,000,000,000
```



Вампир-1



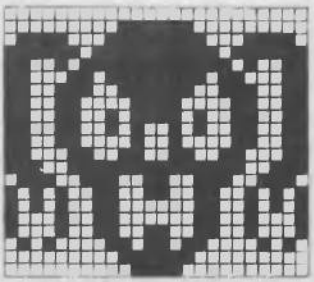
Вампир-2

Данные для ввода спрайта

```
002,004,012,012,028,030,030,062
063,063,063,063,062,062,062,062
030,028,024,008,008,000,129,066
036,090,036,036,060,024,189,255
255,060,060,255,165,036,102,000
000,000,064,032,048,048,056.120
120,124,252,252,252,252,124,124
124,124,120,056,024,016,016
```

Данные для ввода спрайта

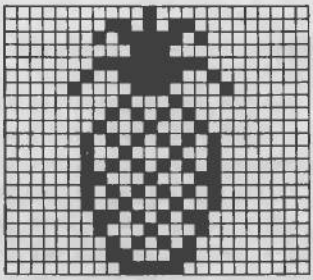
```
000,000,120,253,203,198,206,204
204,204,204,198,231,115,169,169
137,137,080,000,000,000,126,255
255,255,255,126,060,189,165,036
102,255,153,153,129,129,153,219
126,060,000,000,030,191,211,099
115,051,051,051,051,099,231,206
149,149,145,145,010,000,000
```



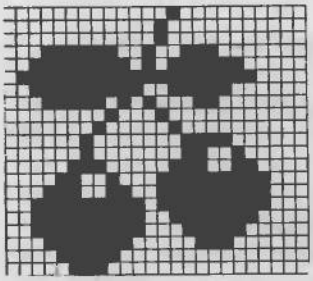
Привиденне

Данные для ввода спрайта

```
000,000,001,000,001,002,004,000
000,001,002,002,002,003,002,002
001,001,000,000,000,016,214,125
056,255,124,056,068,170,017,170
068,170,017,170,068,171,017,170
198,124,000,000,000,000,000,128
064,000,000,000,128,128,128,128
128,128,000,000,000,000,000
```



Ананас



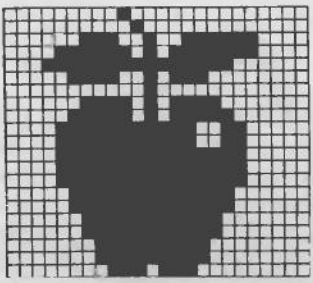
Вишня

Данные для ввода спрайта

```
000,000,000,031,063,127,063,031
000,001,002,002,007,012,028,063
063,063,031,015,007,008,016,016
039,175,255,167,080,136,004,003
003,007,143,207,239,231,227,193
128,000,000,000,000,128,224,240
192,000,000,000,192,032,048,248
248,248,240,224,192,000,000
```

Данные для ввода спрайта

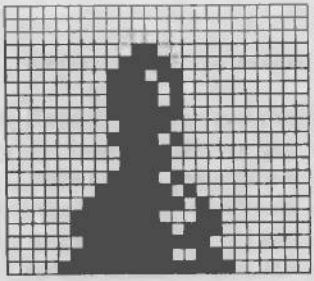
```
000,000,003,015,031,007,000,003
007,015,015,015,015,015,007,007
003,003,001,001,000,064,032,147
215,255,151,016,215,215,254,254
255,255,255,255,255,255,255,255
255,238,000,000,224,240,192,000
000,128,192,096,096,224,224,224
192,192,128,128,000,000,000
```



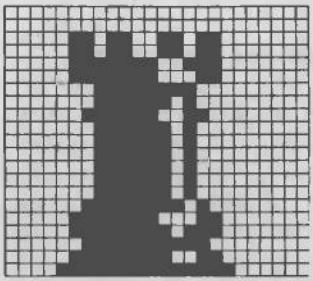
Яблоко

Данные для ввода спрайта

```
000,000,000,000,000,000,000,000
000,000,000,000,000,000,001,003
007,007,003,007,015,000,048,120
236,244,244,252,120,048,120,244
120,120,244,250,253,241,251,255
249,255,000,000,000,000,000,000
000,000,000,000,000,000,000,000
000,000,128,128,000,128,192
```



Пешка



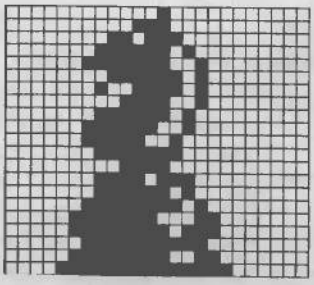
Ладья

Данные для ввода спрайта

```
000,000,006,006,007,007,001,001
003,001,001,001,001,001,001,003
007,007,003,007,015,000,000,205
205,243,241,254,250,243,250,250
250,250,250,250,253,241,251,255
249,255,000,000,128,128,128,128
000,000,000,000,000,000,000,000
000,000,128,128,000,128,192
```

Данные для ввода спрайта

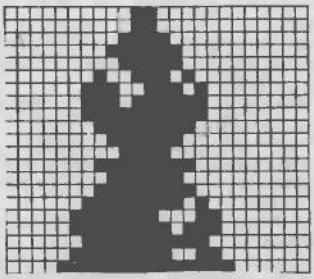
```
000,000,000,001,003,000,001,000
001,003,003,001,000,001,001,003
007,007,003,007,015,008,056,220
250,249,253,061,121,250,242,228
252,120,238,250,253,241,251,255
249,255,000,000,000,000,000,000
000,000,000,000,000,000,000,000
000,000,128,128,000,128,192
```



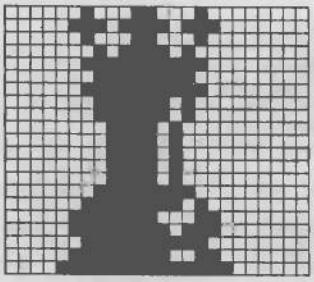
Конь

Данные для ввода спрайта

```
000,000,000,000,000,001,003,003
003,001,000,000,001,000,001,003
007,007,003,007,015,048,048,120
252,124,058,157,191,255,254,252
120,254,252,254,253,241,251,255
249,255,000,000,000,000,000,000
000,000,000,000,000,000,000,000
000,000,128,128,000,128,192
```



Слон



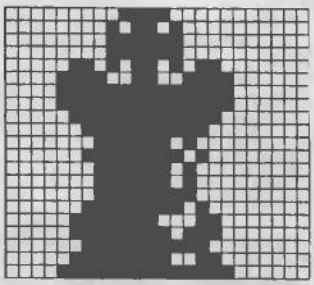
Фарзь

Данные для ввода спрайта

```
002,007,002,001,003,001,003,001
001,000,000,000,000,000,001,003
007,007,003,007,015,049,123,049
122,255,254,255,250,250,244,244
244,244,244,254,253,241,251,255
249,255,000,128,000,000,000,000
000,000,000,000,000,000,000,000
000,000,128,128,000,128,192
```

Данные для ввода спрайта

```
000,000,000,000,006,015,015,015
007,003,001,003,001,001,001,003
007,007,003,007,015,120,180,252
252,181,051,255,255,255,255,250
253,250,250,250,253,241,251,255
249,255,000,000,000,000,128,192
192,192,128,000,000,000,000,000
000,000,128,128,000,128,192
```



Король