



"ИНФОРКОМ" 121019, Москва, Г-19, а/я 16

Вниманию читателей!

Мы продолжаем публиковать адреса пунктов, в которых можно приобрести наши материалы.

- г. МОСКВА, ул. Новый Арбат, д.2. 19-е отделение связи, 1-ый этаж операционного зала.
- г. МОСКВА, радиорынок в Митино, проезд до ст. метро Тушинская или поездом до пл. Трикотажная (Рижское направл.). Суббота, воскр, 9 - 14.
- г. МОСКВА, радиорынок в Царицыно, проезд до ст. метро Царицыно. Суббота, воскр, 9 - 14.
- г. БЕЛГОРОД, магазин "РАДИОТОВАРЫ", ул. Ленина, 32.
- г. ВОРОНЕЖ, Студия компьютерных игр SAN-SAN. Магазин-салон "ЭЛЕКТРОНИКА", тел. 14-00-73.
- г. ДНЕПРОПЕТРОВСК, ул. Шевченко, 34. Фирма "ЭКОС".
- г. ЕКАТЕРИНБУРГ, магазин "СПЕКТРУМ", Главный проспект, 99.
- г. КЕМЕРОВО, магазин "ТЕХНИЧЕСКАЯ КНИГА", ул. Весенняя, 24.
- г. КЕМЕРОВО, магазин "ОРБИТА", пр. Ленина, 133.
- г. КИРОВ, "Дом науки и техн.", Магазин-салон "МАРС", ул. Производственная, д. 27.
- г. КРАСНОЯРСК, радиорынок, проезд до ост. "Затон", субб., вскр.
- г. НИЖ. НОВГОРОД, ИМА "Ф-ПЛЮС", магазин "ФОТОЛЮБИТЕЛЬ", ул. Горького, 146.
- г. ОРЕНБУРГ, магазин "ВОЕННАЯ КНИГА", ул. Советская.
- г. РЫБИНСК, ул. Гоголя, 1. ТТЦ "ГНОМ".
- г. ЧЕБОКСАРЫ, маг. "ЭКСПРЕСС", НПК Фирма "НОВА", ул. Привокзальная, д.6.

Вниманию дистрибуторов! Адреса наших оптовых покупателей публикуются бесплатно.

СПЕКТРУМ В ШКОЛЕ

Дорогие друзья!

В прошлом месяце мы "играючи" научились отыскивать решение сложных алгебраических уравнений численным методом простой итерации.

Это, как Вы понимаете, не единственный возможный численный метод и сегодня мы попробуем с Вами сыграть в другую компьютерную игру, которая тоже позволяет находить корни алгебраических уравнений. Назовем эту игру "Вилка". Имеется в виду "артиллерийская вилка", хотя можно было бы назвать игру и "Перелет - Недолет".

Если Вы не знаете, что такое артиллерийская вилка, то лучше бы Вам этого и не знать, потому что ситуация, когда Вы в нее попадаете, очень неприятна. Представьте себе теплое летнее утро в полевом блиндаже. Хорошо позавтракав, противник начинает лениво высевать дневную норму снарядов и мин по вашим позициям. Бах!!! Снаряд разорвался где-то далеко впереди. "Недолет", - ухмыляетесь Вы и продолжаете спокойно уплетать консервы. Бах!!! Одинокий снаряд разрывается где-то в сотне метров сзади. "Так, теперь перелет".

Вот теперь Вы должны понимать, что попали в "вилку" и, если у противника есть хоть какие-то мозги в голове, то следующим выстрелом он накроет Ваш блиндаж, будьте покойны. Двумя выстрелами он "пристрелял" орудие и теперь у него есть математические таблицы, прицельные приспособления и большой опыт, чтобы в третий раз не промахнуться. Более того, пристреляв одно орудие и использовав эти данные и для других орудий, он может ударить теперь и всей батареей. Бросайте свою тушенку и двигайте в окоп, - хорошо,

если Вы не пожалели сил и заранее откопали запасную линию, где и отсидитесь во время арналета.

Вот так же можно попасть "выстрелом" и в корень алгебраического уравнения, если предварительно сделать несколько "выстрелов".

Причем Вам совершенно все равно, если первые "выстрелы" будут очень неточными. Единственное, что должно быть сделано - в одном выстреле должен быть "перелет", а в другом "недолет", иначе "вилка" не получится.

Рассмотрим простое уравнение:

$$3x^2 + 5x - 22 = 0$$

Сделаем первый выстрел - пусть $x=0$. Подставим x в уравнение, оно естественно не сойдется и возникнет "невязка" $d = -22$.

Раз невязка имеет знак "минус", будем считать, что это "недолет".

Сделаем второй выстрел $x=5$. Теперь $d=78$ - это большой перелет. То, что мы хотели, произошло. Образовалась артиллерийская вилка:

$x=0$ - мало

$x=5$ - много.

Если бы она не образовалась, нам пришлось бы еще "пострелять". А теперь все просто - ударим посередине $x=2.5$. Подсчитаем невязку $d = 9.25$ - это опять перелет, но уже ближе к цели, мы получили новую вилку

$x=0$ - мало

$x=2.5$ - много.

Вновь ударим посередине $x=1.25$. Невязка d равна примерно -12 - это недолет.

$x=1.25$ - мало

$x=2.5$ - много.

Следующий удар нанесем по пункту 1.875. И так далее. Точное решение $x=2$ и с каждым "выстрелом" мы все ближе к нему подбираемся.

Вы, конечно поняли, что на каждом шаге мы берем "недолет" и "перелет" и стреляем посередине, потом опеределаем, что это было - "недолет" или "перелет" и от предыдущего шага берем противоположное значение, чтобы всегда оставалась "вилка".

Теперь нам осталось только поручить все эти дела компьютеру.

1 REM здесь Вы введете свое уравнение.

10 DEF FN A(X) -

20 LET b=.01 : REM допустимое значение "невязки"

29 REM теперь вводим пару значений x так, чтобы получить "вилку"

30 INPUT "Xmin?":x0

40 INPUT "Xmax?":x1

49 REM рассчитаем невязку

50 LET d0 = FN A(x0)

60 LET d1 = FN A(x1)

70 LET print = 500 80 GO SUB print

90 IF SIGN (d0) = SIGN (d1) THEN GO TO 30: REM "вилка" не получилась"

100 LET x= (x0 + x1)/2: REM "вилка получилась"

110 LET d = FN A(x)

120 IF ABS(d)<b THEN GO TO 200

130 IF SGN (d) = SGN (d0) THEN GO TO 170

140 LET x1 = x: LET d1 = d

150 GO SUB print

160 GO TO 100

170 LET x0 = x: LET d0 = d

180 GO SUB print

190 GO TO 100

200 PRINT "!!! x= ";x,"d= ";d

210 STOP

500 PRINT "x0= ";x0,"d0= ";d0

510 PRINT "x1= ";x1,"d1= ";d1

520 RETURN

Этот метод определения корней в сложных алгебраических уравнениях тоже известен

не одну сотню лет и называется методом Ньютона. Нам с Вами просто очень повезло, что ко времени появления первых ЭВМ уже были известны десятки и сотни подобных численных методов, легко адаптируемых для компьютеров. Впрочем, может быть ЭВМ именно потому и появились, что потребовалось упростить работу сотен расчетчиков, сидевших месяцами с арифмометром над баллистическими таблицами для расчета траекторий полета первых ракет.

SINCLAIR LOGO

(Продолжение) Начало см. на с. 69-74, 90-96.

ГЛАВА 5 ПОВТОРЯЮЩИЕСЯ ОПЕРАЦИИ

Условия.

Под "условием" в ЛОГО понимается операция, которая дает в результате логическое значение TRUE ("истина") или FALSE ("ложь"), в ответ на вопрос типа: "Это число равно другому?"

В ЛОГИКЕ, а также в языках программирования, имеющих дело с обработкой списков (ЛИСП, ПРОЛОГ, ЛОГО), такие операции называют ПРЕДИКАТами. Может быть поэтому в ЛОГО все примитивы, относящиеся к этой категории, оканчиваются на "-Р"

Попробуйте набрать:

```
PRINT EQUALP 5 5
```

- получите в результате:

```
TRUE
```

Здесь условие EQUALP проверяет, равны ли два параметра. Если да, то выдается результат TRUE, иначе выдается результат FALSE. Например:

```
PRINT EQUALP 5 7
```

```
FALSE
```

Математические условия.

Как и в других математических операциях здесь используют символы, которые размещают между двумя аргументами (так называемая индексная форма записи). Это следующие символы:

> - больше

< - меньше

= - равно.

Примеры:

```
PRINT 1 > 2
```

```
FALSE
```

```
PRINT 5 < 8
```

```
TRUE
```

```
MAKE "FACT 5-5
```

```
PRINT :FACT
```

```
TRUE
```

Знак "=" используется для сравнения двух объектов любого типа. Это могут быть числа, слова, списки. Эта операция дает значение TRUE тогда и только тогда, когда оба объекта одинаковы. Эта операция тождественна операции EQUALP, которую мы рассмотрели ранее. Разница, как видите, только в форме записи, поскольку EQUALP записывается не "между", а "перед" двумя параметрами (префиксная форма записи).

```
MAKE "SHOPPING[BREAD BUTTER MILK STAMPS]
```

```
PRINT :SHOPPING =[BREAD BUTTER MILK STAMPS]
```

```
TRUE
```

```
MAKE "GIRL "MARY
```

```
PRINT :GIRL=MARY
```

```
TRUE
```

```
MAKE "NAME "MARY
```

```
PRINT :GIRL = :NAME
```

```
TRUE
```

```
PRINT (7-5)-2 TRUE
PRINT EQUALP(7-5) 2
TRUE
```

Условия очень часто используют совместно с оператором IF. IF после себя требует два, а иногда и три операнда. Первый - это условие, а второй - список инструкций, который должен быть выполнен, если условие справедливо. Например:

```
MAKE "TODAY "FRIDAY
IF :TODAY = "FRIDAY [PRINT "TGIF]
TGIF
```

Если оператор IF имеет три операнда, то в третьем операнде записывается список инструкций, которые должны выполняться, если условие "ложно".

```
MAKE "TODAY "MONDAY
IF :TODAY= "FRIDAY [PRINT "TGIF]
[PRINT [NOT FRIDAY YET]]
NOT FRIDAY YET
```

Как и конструкции REPEAT, команда IF может иметь достаточно длинную запись и ее приходится переносить со строки на строку.

А теперь мы рассмотрим некоторые примитивы, являющиеся условиями, и обсудим, в каких случаях они могут применяться.

MEMBERP

Проверяет, является ли первый операнд (а он может быть числом, словом или списком) частью второго (который должен быть списком).

```
TO CHECKSHOP: STUFF
MAKE "SHOPPING[BREAD BUTTER MILK STAMPS]
IF MEMBERP: STAFF:SHOPPING
[PRINT [GOT IT]] [PRINT [OH DEAR FORGOTTEN IT]]
END
```

```
CHECKSHOP "MILK
GOT IT
CHECKSHOP "BACON
OH DEAR FORGOTTEN IT
```

NUMBERP

Эта программа выдает значение TRUE, если ее аргумент является числом. В противном случае выдается FALSE.

Вам нередко приходится проверять, является ли то, что ввел пользователь, числом, прежде, чем приступить к обработке этого ввода. Согласитесь, нелепо задать вопрос о том, сколько Вам лет ("How old are you?") и пытаться как-то обсчитывать полученный результат, если в ответе было:

NOT VERY MUCH ("не очень много")

```
TO CHECKNUM
MAKE "ANSWER FIRST READLIST
IN NUMBERP: ANSWER
[PRINT [[THANK YOU]]]
[PRINT [[I WANTED A NUMBER]]]
END
```

Обратите внимание на то, что мы используем FIRST READLIST. Если бы мы использовали только READLIST, то результат был бы списком. Даже если в нем было бы

всего одно число, например 5, он все равно распознавался бы только как список, а не как число.

WORDP

Эта операция дает в результате TRUE, если ее операндом является слово. Вспомните, что числа тоже являются словами, поэтому они тоже дают в результате TRUE.

```
PRINT WORDP 3
TRUE
```

```
PRINT WORDP "WISE
TRUE
```

```
PRINT WORDP [MEN]
FALSE
```

В последнем случае операнд является списком, в который входит только одно слово, но все же это список, а не слово и потому в результате выдано FALSE.

LISTP

Дает в результате TRUE, если операнд является списком.

```
PRINT LISTP:SHOPPING
TRUE
```

В практической работе Вам придется много раз писать процедуры для обработки списков. Очень хорошей практикой будет устраивать в начале каждой процедуры проверку: "А является ли входной параметр списком?"

EMPTYP

Проверяет, не пустой ли операнд. Ищет пустое слово " или пустой список [] и если то, либо другое найдено, выдает TRUE. Этот оператор имеет очень широкое применение. Длинные списки анализируются путем разбиения их на элементы и последовательной обработки элементов одного за другим, пока список не станет пустым. Точно также слова анализируют, разбивая их на символы. Мы в дальнейшем будем очень часто использовать выражение вида:

```
IF EMPTYP : ALIST[STOP]
```

для того, чтобы останавливать процесс.

NAMEP

Дает в результате TRUE, если операндом является имя чего-либо, т.е. этому имени соответствует некоторое содержание.

```
MAKE "MONTH "APRIL
PRINT NAMEP "MONTH
TRUE
PRINT NAMEP "FRED
FALSE
```

Рекурсия.

Мы уже рассматривали такие процедуры, как

```
TO TRI
  FORWARD 50
  RIGHT 120
  TRI
END
```

Запустите эту процедуру. Сама она не остановится. Придется для этого использовать

BREAK (CAPS SHIFT + SPACE). Самое важное в этой процедуре - это то, что она входит в бесконечный цикл повторов без использования повтора REPEAT, а вызовом самой себя TRI. Итак, TRI исполняет первых две инструкции - FORWARD 50 и RIGHT 120, после чего снова вызывается TRI, которая исполняет первые две инструкции и т.д. и т.п.

Рекуррентные процедуры могут иметь входные параметры, как и любые другие процедуры. В простейшем виде эти входные параметры всегда одни и те же при каждом вызове процедуры. Мы можем легко изменить процедуру TRI и задать любой многоугольник.

```
TO POLY :SIDE :ANGLE
  FORWARD :SIDE
  RIGHT :ANGLE
  POLY :SIDE :ANGLE
END
```

Эта процедура отличается от той, которую мы рассматривали в главе 2 (с инструкцией REPEAT). Она может изображать некоторые такие вещи, которые той не под силу. Попробуйте:

```
POLY 50 144
```

Когда процедура вызывает саму себя, это называется рекурсией. В ЛОГО это основной способ организации повторяющихся процессов.

В процедурах TRI и POLY мы использовали очень простой вид рекурсии, когда процедура не знает, где же следует остановиться.

Рекурсия используется не только для того, чтобы организовать повторяющиеся действия. Здесь могут быть достигнуты и некоторые дополнительные цели.

Предположим, что Вы покупаете тюбик зубной пасты и, вскрыв его, находите внутри ваучер, дающий право на снижение цены на 20 центов при покупке очередного тюбика, по сравнению с этим. Спрашивается: "Когда Вам надо идти за следующим тюбиком?" Ответ: "Немедленно". Ведь и в нем будет такой же ваучер и т.д. и скоро Вы завалите квартиру почти бесплатной зубной пастой.

Аналогичный прием может быть использован и в ЛОГО:

```
TO ADDON :NUMBER
  PRINT :NUMBER
  ADDON :NUMBER+1
END
```

```
ADDON 1
1
2
3
...
и т. д.
```

При первом вызове ADDON печатается 1 и во второй вызов передается параметр 2 и т. д.

Рассмотрим еще один пример. Предположим, что нам надо найти сумму чисел от 1 до N, где число N начинается с единицы и непрерывно возрастает. Может быть, мы примем решение, что процесс надо остановить, когда сумма превысит одну тысячу. И тогда мы вводим команду STOP, служащую для того, чтобы останавливать текущую процедуру. В принципе, STOP действует так же, как и END, но может размещаться в любом месте процедуры.

Обратите внимание на то, что STOP останавливает только ту процедуру, внутри которой находится. Если эта процедура вызывалась другой, то та не останавливается и продолжает работать.

```
TO ADDUP :NUMBER
```

```

IF :TOTAL >1000 [STOP]
MAKE "TOTAL :TOTAL + :NUMBER
PRINT :TOTAL
ADDUP :NUMBER+1
END
MAKE "TOTAL 0

```

```

ADDUP 1

```

```

1

```

```

3

```

```

6

```

```

10

```

```

15

```

```

...

```

```

и т. д.

```

Обратите внимание на то, что нам приходится выставлять исходное значение TOTAL, равное нулю, вне процедуры, поскольку нам не надо, чтобы TOTAL принимало бы нулевое значение всякий раз, как происходит исполнение процедуры.

Рекурсия очень часто оказывается очень удобной при работе со списками. Например, поэлементная печать списка, которую мы рассматривали в гл. 1, может быть сделана более элегантно с помощью рекурсии. Например, так:

```

TO LBL :ALIST
PRINT FIRST :ALIST
LBL BUTFIRST :ALIST
END

```

Все очень просто. Процедура распечатывает содержимое начала списка, а затем точно так же поступает с его остатком. Когда же она закончит свою работу? Когда список станет пустым.

```

TO LBL :ALIST
IF EMPTY? :ALIST [STOP]
PRINT FIRST :ALIST
LBL BUTFURST :ALIST
END

```

Проверьте эту процедуру на каком-нибудь списке, а затем попробуйте ее видоизменить так, чтобы она печатала список в обратном порядке.

Конечно, в некоторых случаях рекуррентные выражения могут быть весьма сложными, но простейшие применения рекурсии поистине просты. Кроме бесконечных рекурсий (которые на самом деле тоже бесконечными не являются, ведь при каждом вызове процедуры ЛОГО фиксирует факт этого вызова в оперативной памяти и рано или поздно эта память исчерпывается) существуют еще две возможности. В приведенных примерах процедура начинается с проверки. В языках, предназначенных для структурного программирования, таких как Паскаль или скажем, в развитой версии БЕЙСИКа Бета-БЕЙСИК, эта конструкция называется циклом WHILE. В ЛОГО это выглядит так:

```

TO WHILE <параметр>
IF <условие> [STOP]
... <тело процедуры> ...
WHILE <параметр>
END

```

В этой процедуре <тело> не исполняется, если <условие> принимает значение TRUE. В частности, эта процедура вообще не исполняется, если исходное значение параметра таково, что <условие> справедливо.

Другой прием состоит в проверке условия в конце серии повторяющихся инструкций. Это соответствует команде UNTIL Паскаля или БЕТА БЕЙСИКа.

```

TO UNTIL <параметр>
... <тело процедуры>
IF <условие> [UNTIL<параметр>]
END

```

Здесь тело процедуры всегда выполняется до проверки условия, так что по крайней мере один раз процедура будет исполнена.

Очень часто, когда мы используем рекурсию для организации повторяющихся вычислений, нам необходимо создать стартовую процедуру, которая задаст начальные значения переменных, сделает первый вызов рекуррентной процедуры и сможет обработать полученный из нее результат. В нижеприведенном примере, который должен отыскать сумму, последовательности впечатанных с клавиатуры чисел, процедура DOTOTAL сначала устанавливает ноль в качестве текущей суммы, а затем вызывает рекуррентную процедуру TOTALUP. Когда работа закончена, DOTOTAL печатает итоговый результат.

```

TO DOTOTAL
MAKE "TOTAL 0
TOTALUP
PRINT SENTENCE [THE TOTAL IS] :TOTAL
END

```

Процедура TOTALUP демонстрирует использование процедуры NUMBERP для того, чтобы проверять те числа, которые пользователь вводит с клавиатуры. Она позволяет пользователю ввести числовую последовательность и закончить ее словом END или любым другим нечисловым вводом. Процедура организована по схеме UNTIL, рассмотренной выше. Единственное отличие состоит в том, что вводится дополнительная инструкция

```
MAKE "TOTAL :TOTAL + :NUM
```

если условие справедливо (TRUE). Эту инструкцию можно было бы поместить и в начало процедуры, но только в том случае, если бы первое значение NUM задавалось бы вышележащей процедурой DOTOTAL.

```

TO TOTALUP
PRINT [TYPE IN A NUMBER]
MAKE "NUM FIRST READLIST
IF NUMBERP :NUM[MAKE "TOTAL :TOTAL +NUM TOTALUP]
END

```

Когда с клавиатуры будет введено какое-то слово, не являющееся числом, вызов TOTALUP будет прерван. В итоге, будет остановлен и весь процесс и управление будет передано в вышележащую процедуру в точку после первого вызова TOTALUP.

Генератор случайной последовательности.

Идеи, изложенные в этой главе, можно применить к генерации случайных последовательностей. Рассматривайте это как первый шаг к программам, с помощью которых компьютер будет писать стихи.

Давайте рассмотрим сначала как ЛОГО может выдавать случайные числа. Вообще-то эти числа рассчитываются и потому не являются вполне случайными, но маловероятно, что Вы заметите их неслучайность.

Так, RANDOM 5, например, выдаст одно число из последовательности 0,1,2,3,4. Точно также RANDOM 6 даст число от 0 до 5.

Попробуйте:

```
REPEAT 50 [PRINT RANDOM 6]
```

- и посмотрите, что из этого выйдет.

Если Вам надо иметь случайное число не из диапазона [0;4], а из диапазона [1;5], то Вы тоже можете воспользоваться оператором RANDOM 5, но к полученному результату надо прибавить единицу. Это делается так:

```
1 + RANDOM 5.
```

Обратите внимание на то, что запись RANDOM 5+1, будет неправильной, поскольку сначала будет исполнено сложение 5 + 1, а затем процедура RANDOM с параметром 6.

Давайте используем эту процедуру для того, чтобы извлечь из списка случайный элемент. Прежде всего введем переменную ITEMNO - случайное целое число от 1 до значения COUNT, а затем используем эту переменную для выбора элемента списка.

```
TO CHOOSE :ALIST
  MAKE "ITEMNO 1 + RANDOM COUNT :ALIST
  MAKE "CHOICE ITEM :ITEMNO :ALIST
END
```

Попробуйте эту процедуру с каким-либо списком. Например, для списка PRESENT:

```
CHOOSE :PRESENT
```

Для того, чтобы генерировать предложения, нам нужна сначала какая-то канва, какая-то скелетная схема, например:

```
THE SMALL DOG QUICKLY BITES THE CARELESS MAN
```

Рассмотрим структуру этого предложения, учитывая, что не все наши читатели могут быть знакомы с английской грамматикой.

- Артикль "the" может иметь в качестве альтернативы артикль "a".
- Прилагательное SMALL (маленький, -ая) может быть заменено на иное, например: [BIG BROWN SPOTTED CARELESS]
- Существительное "DOG" (собака) является таким же объектом, как и [MAN TREE TABLE KENNEL]
- Наречие QUICKLY (быстро) описывает, как происходит событие. Вместо него можно употребить [VICIOUSLY SLOWLY CAREFULLY HAPPILY]
- Глагол BITES (кусает) - указывает на происходящее действие точно так же, как и глаголы: [LIKES CHASES CARRIES HIDES].
- Еще один артикль "THE".
- Еще одно прилагательное "CARELESS" (беспечный)
- Еще одно существительное "MAN" (человек, мужчина).

Итак, если мы введем сокращения:

AST - артикль;

ADJ - прилагательное;

ADV - наречие;

VER - глагол;

KOU - существительное, то получим следующую скелетную схему:

```
[AKT ADJ NOU ADV VER ART ADJ SOU]
```

Приравняем этот список переменной PATTERN, а с именами AKT, ADJ, NOU и пр. свяжем свои списки глаголов, наречий, существительных и т.п. Так, например, список с именем "ADJ" будет содержать:

```
[SMALL BIG BROWN SPOTTED CARELESS]
```

Теперь мы можем конструировать случайные предложения.

```

TO RANDSENTENCE
  MAKE "PATTERN [ART ADJ NOU VER ART ADJ NOU]
  MAKE "ART [SMALL BIG BROWN SPOTTED CARELESS]
  MAKE "KOU [MAK TREE TABLE KENNEL DOG]
  MAKE "ADV [QUICKLY VICIOUSLY SLOWLY CAREFULLY HAPPILY]
  MAKE "VER [BITES LIKES CHASES CARRIES HIDES]
  REPEAT 50 [DORANDOM]
END

TO DORANDOM
  MAKE "OUTLINE []
  FOLLOW :PATTERN
  PRINT "OUTLINE
  PRINT "
END

```

Процедура FOLLOW сначала проверяет, не является ли ее параметр (а это список, в котором закодирована скелетная схема нашей фразы) пустым. Если это так, происходит выход из процедуры. Иначе, берется первый элемент скелета, а он является именем списка слов и из этого списка делается случайный выбор. Выбранное слово добавляется к собираемому предложению OUTLINE.

```

TO FOLLOW :APATTERN
  IF EMPTY? :APATTERN [STOP]
  MAKE "THISWORD FIRST :APATTERN
  CHOOSE THING :THISWORD
  MAKE "OUTLINE SENTENCE :OUTLINE :CHOICE
  FOLLOW BUTFIRST :APATTERN
END

```

ГЛАВА 6. ЧЕРЕПАШЬЯ ГРАФИКА-2.

Идеи, рассмотренные в двух предыдущих главах, могут быть проиллюстрированы с помощью "черепашьей графики". Возьмем процедуру RESPI из гл. 3 и модифицируем ее так, чтобы "черепашка" в одних случаях поворачивала бы направо, а в других - налево. Сделаем это, обеспечив дополнительный входной параметр, который будет списком чисел, значений COUNT, для которых "черепашка" должна поворачивать влево. Когда COUNT является числом, входящим в этот список, выполняется поворот налево, иначе - направо. Вставлять всю эту логику в оператор REPEAT, по-видимому, слишком длинно, поэтому проще создать отдельную процедуру:

```

TO RESPI2 :SIZE :ANGLE :MAX :ALIST
  MAKE "COUNT 1
  REPEAT :MAX [SUBSPI]
  RESPI2 :SIZE :ANGLE :MAX :ALIST
END

TO SUBSPI
  FORWARD :SIZE * :COUNT
  IF MEMBERP :COUNT :ALIST [LEFT :ANGLE] [RIGHT :ANGLE]
  MAKE "COUNT :COUNT+1
END

```

Проверьте эту процедуру со следующими параметрами:

```

RESPI2 5 120 6 [1 3]
RESPI2 5 90 11 [3 4 5]

```

На рис. 1 показан пример работы процедуры.

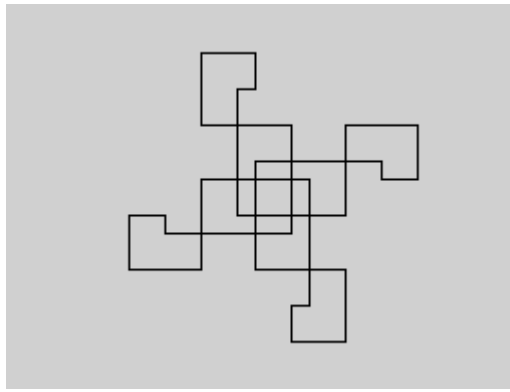


Рис.1 RESPI2 5 90 7 [1 2 3 4]

Мы рассмотрели выше способ остановки рекуррентных процедур. Теперь интересно было бы рассмотреть, как можно остановить рекуррентную процедуру, изображающую замкнутую диаграмму (такую, в которой "черепашка" возвращается в ту точку, с которой начинала работу. Прежде всего, нам надо узнать, вернулась ли "черепашка" в точку старта. В ЛОГО есть процедура-примитив, которая вычисляет координаты "черепашки" в любой момент времени. Это процедура `position`, она выдает список из двух чисел. Первое число - горизонтальная координата, отмеренная от центра экрана; второе число - вертикальная координата, она тоже измеряется от центра экрана и для верхней его половины является положительной, а для нижней - отрицательной.

ЛОГО имеет три команды для перемещения "черепашки" в указанное положение:

`SETX 50` - переводит ее на вертикаль, имеющую координату X, равную 50.

`SETY -50` - переводит ее на горизонталь, имеющую координату Y, равную -50.

`SETPOS [-20 30]` - устанавливает "черепашку" в точку, с указанными в списке координатами.

Если перо "черепашки" опущено, то при исполнении каждой из этих трех операций будет прорисована линия от исходной точки к конечной.

В главе 3 мы говорили о том, что знак "минус" перед числом имеет двойственное значение. ЛОГО применяет простое правило к оценке знака "минус", если он находится в списке. Если перед знаком есть пробел или нет ничего и число следует непосредственно за знаком (без пробела), то знак «минус» рассматривается, как часть числа и это означает, что число отрицательное. Во всех прочих случаях этот знак интерпретируется, как самостоятельное слово.

Итак, `[- 10 10]` - это три слова, точно так же и `[10 - 10]` и `[10-10]` и ни одна из этих записей не может быть использована в качестве входного параметра для процедуры `SETPOS`.

С другой стороны, `[10 -10]` и `[-10 10]` - состоят из двух слов и могут быть использованы при вызове `SETPOS`.

Следует внимательно относиться и к другим процедурам, принимающим список входных параметров в виде чисел, среди которых могут быть и отрицательные.

Кроме процедур, позволяющих выставить X и Y-координаты "черепашки", существуют две функции `XCOR` и `YCOR`, позволяющие определить и выдать текущие координаты X и Y. Эти функции не могут выдать эти координаты на экран, но могут предоставлять их другим процедурам для обработки.

Если мы хотим определить момент, когда "черепашка" сделает замкнутый контур, то кроме координат нас должно интересовать еще и направление, в котором она смотрит. Этот результат может быть получен с помощью функции `HEADING`. - она выдает результат от 0 до 360 (можете рассматривать его, как показание стрелки компаса).

Север - 0

Восток - 90

Юг - 180

Запад - 270

Таким образом, поворачивая "черепашку" направо, Вы увеличиваете значение HEADING, а поворачивая ее налево - уменьшаете его. Направление, в котором "смотрит" "черепашка" можно не только измерить, но и задать с помощью команды SETHEADING, после которой должно идти число от 0 до 360.

Вернемся к нашей проблеме определения момента, когда "черепашка" закончит замкнутый контур и пойдет по второму разу.

Нам надо зафиксировать координаты "черепашки" в исходной точке и направление ее движения. Для удобства мы можем собрать эти три числа в один список:

```
MAKE "STATE LPUT HEADING POSITION
```

А после этого мы можем регулярно проверять положение "черепашки", дабы уловить момент, когда она будет находиться в той же точке и смотреть при этом в том же направлении.

```
IF EQUALP :STATE LPUT HEADING POSITION [STOP]
```

Конечно, не следует делать такую проверку до того, как "черепашка" сделает первый шаг.

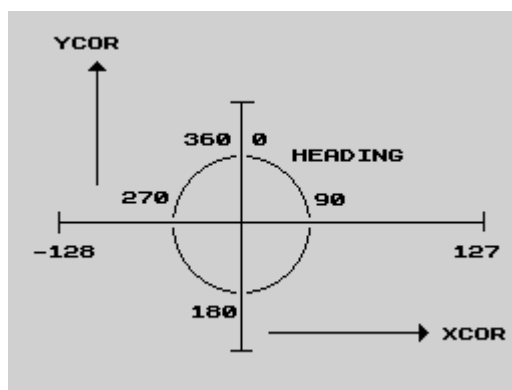


Рис. 2 Координаты "черепашки"

Можно использовать процедуры, задающие координаты X и Y и для того, чтобы рисовать на экране графики. Предположим, что у Вас есть список из двенадцати чисел (ежемесячные расходы) и его надо передать в качестве входного параметра в процедуру, которая нарисует на экране график. Пока для простоты изложения мы предположим, что все числа не превосходят 160 и не являются очень малыми, по сравнению с этим числом.

Сначала надо нарисовать пару осей. Начиная из левого нижнего угла из координаты [-125 -80] нарисуем горизонтальную линию с двенадцатью вертикальными метками, соответствующими 12 месяцам.

```
TO XAXIS
  PENUP
  SETPOS [-125 -80]
  PENDOWN
  SETHEADING 90
  FORWARD 5
  REPEAT 12 [FORWARD 20 SETY -82 SETY -80]
END
```

Вертикальные метки идут через каждые 20 шагов "черепашки", т.е. 20 шагов соответствуют одному календарному месяцу.

Аналогично рисуется и вторая ось. Ее тоже следовало бы разметить делениями шкалы, но поскольку они зависят от тех количеств, которые мы измеряем и тех единиц, которые мы применяем, это лучше делать для каждого конкретного случая по-своему и наш читатель сможет сделать это самостоятельно, по вкусу.

```
TO YAXIS
```

```

PENUP
SETPOS [120 -85]
PENDOWN
SETY 80
END

```

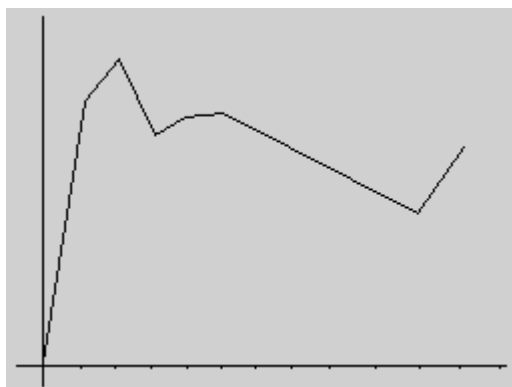


Рис. 3 График ежемесячных расходов.

Оси пересекаются в т. $[-120 -80]$, поэтому это и будет исходная точка нашего графика. Просмотрим список, используя переменную COUNT и для каждого значения нарисую линию, соединяющую предыдущую точку с текущей.

Поскольку исходная точка у нас находится не в нуле, а отстоит от него на 120 единиц влево и на 80 единиц вниз, мы должны будем соответственно вычесть 120 из горизонтальной координаты и 80 из вертикальной координаты каждой текущей точки.

```

TO PLOT
  MAKE "COUNT :COUNT + 1
  MAKE "XCOR :COUNT*20-120
  MAKE "YCOR ITEM :COUNT :VALUelist-80
  SETPOS LIST :XCOR :YCOR
END

```

COUNT необходимо в начале работы выставлять в нуль с тем, чтобы когда процедура выполнялась бы в первый раз, это значение было бы равно единице. Необходимо также подготовить список VALUelist значений, для которых должен строиться график.

В итоге мы получим главную процедуру:

```

TO GRAPH :VALUelist
  XAXIS
  YAXIS
  SETPOS [-120 -80]
  PENDOWN
  MAKE "COUNT 0
  REPEAT 12[PLOT]
  PENUP
  SETPOS [-120 -80]
END

```

Если среди данных есть значения, большие чем 160, или весьма малые значения, то либо "черепашка" выйдет за пределы экрана, либо график будет плохо различим. В этих случаях возникает необходимость масштабировать данные. Для этого их умножают на какое-либо число, в результате чего получают график приемлемых размеров.

Итак, научились использовать рекурсию при построении графических фигур. Приведем в качестве примеров еще несколько фигур, при построении которых тоже используется рекурсия.

Во-первых, рассмотрим бесконечную кривую, которую называют "снежинкой".

Построим равносторонний треугольник. Разделим каждую сторону на три части и на среднем отрезке построим еще по равностороннему треугольнику и т.д. (Рис. 4)

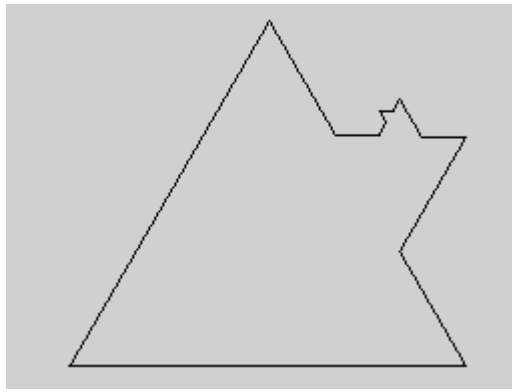


Рис. 4 "Снежинка"

Для рисования треугольника мы поступаем просто - три раза:

```
FORWARD :LENGTH
RIGHT 120
```

Для того же, чтобы изобразить "снежинку", мы должны заменить процедуру FORWARD процедурой изображения треугольника с размером 1/3 предыдущего.

```
TO SNOWFLAKE :LENGTH
  REPEAT 3[SNOWLINE :LENGTH RIGHT 120]
END
```

Для того, чтобы нарисовать сторону "снежинки", служит процедура SNOWLINE. "Черепашка" проходит треть текущей стороны, поворачивается влево на 60 градусов, рисует линию длиной 1/3 от предыдущей, поворачивается направо на 120 градусов и т.д.

```
TO SNOWLINE :LENGTH
  FORWARD :LENGTH/3
  LEFT 60
  FORWARD :LENGTH/3
  RIGHT 120
  FORWARD :LENGTH/3
  LEFT 60
  FORWARD :LENGTH/3
END
```

Это создает тот эффект, который нам нужен, но СТОП! Вместо каждой из сторон треугольника мы получили по четыре отрезка. Но ведь на каждом из них должен быть построен свой треугольник. Поэтому всякая команды FORWARD в процедуре SNOWLINE должна быть фактически заменена вызовом самой этой процедуры SNOWLINE.

Очевидно, что с каждым шагом стороны этих треугольников будут все меньше и меньше и их трудно будет рассмотреть.

Мы можем заранее установить какой-то предел на этот размер и когда длина стороны становится меньше этого предела, прекращаем эту рекурсию и идем дальше. Если этот предел включить в качестве параметра в процедуру SNOWLINE, то можно начинать с довольно больших треугольников и постепенно вычерчивать "снежинку" все детальнее и детальнее.

```
TO SNOWFLAKE :LENGTH :LIMIT
  REPEAT 3 [SNOWLINE :LENGTH RIGHT 120]
END

TO SNOWLINE :LENGTH
  IF :LENGTH < :LIMIT [FORWARD :LENGTH STOP]
  SNOWLINE :LENGTH/3
  LEFT 60
  SNOWLINE :LENGTH/3
```

```

RIGHT 120
SNOWLINE :LENGTH/3
LEFT 60
SNOWLINE :LENGTH/3
END

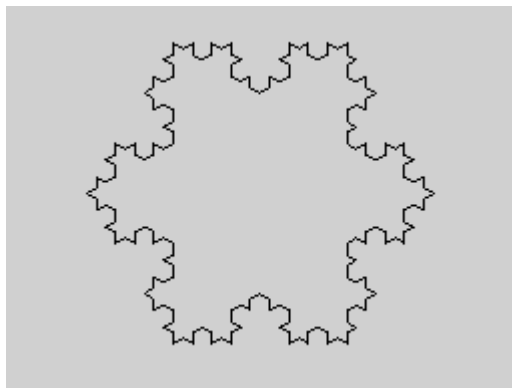
```

Хорошим демонстрационным примером будет:

```

PENUP
SETY -30
PENDOWN
SNOWFLAKE 100 5

```



И напоследок кривая, которую называют "драконом". Она рисуется двумя процедурами, которые вызывают друг друга по-очереди. Между вызовами одна процедура обеспечивает поворот направо, а другая - налево. Длина шага в рекуррентной последовательности одна и та же, поэтому здесь нам нужен иной метод для того, чтобы знать, когда остановиться. Можно, например, сделать так, что когда процедура дойдет до определенного уровня глубины рекурсии, то процесс должен прерываться. Для этого введем счетчик, который назовем LEVL. Он начнет свою работу, например со значения 5 и при каждой рекурсии будет убывать на единицу, пока не обнулится.

Вот эти процедуры:

```

TO LEFTDRAGON :SIZE :LEVL
  IF :LEVL = 0 [FORWARD :SIZE STOP]
  LEFTDRAGON :SIZE :LEVL-1
  LEFT 90
  RIGHTDRAGON :SIZE :LEVL-1
END

TO RIGHTDRAGON :SIZE :LEVL
  IF :LEVL = 0 [FORWARD :SIZE STOP]
  LEFTDRAGON :SIZE :LEVL-1
  RIGHT 90
  RIGHTDRAGON :SIZE :LEVL-1
END

```

Не имеет значения, с какой процедуры Вы начнете работу. Рекомендуется начинать с небольших значений SIZE и LEVL. Начинать работу можно, например, так:

```

CLEARSCREEN
PENUP
SETH 90
SETX 100
PENDOWN
LEFTDRAGON 2 20

```

(Продолжение следует)

ПРИМЕНЕНИЕ АСSEMBЛЕРА ДЛЯ СОЗДАНИЯ БЫСТРОРАБОТАЮЩИХ ПРОГРАММ

Перевод с английского Пашорина В.И.
Продолжение. Начало см. с. 9-12, 48-61, 97-103

7. ДВИЖЕНИЕ

Теперь, когда мы познакомились со способами вывода информации на экран и способами считывания состояния клавиатуры, мы можем рассмотреть способы задания перемещения объектов на экране дисплея. Это, безусловно, важная часть любой игровой программы. В этой главе мы рассмотрим все возможные способы задания перемещения.

Для организации движения изображения по экрану большинство программистов используют знакоместа, что позволяет ускорить выполнение программы и оперировать с любым из возможных цветов. Весь экран представляет собой прямоугольную матрицу 24x32, т.е. всего 768 знакомест и поэтому перемещение символа из одного знакоместа в другое, соседнее с ним, выглядит не совсем плавным. При этом, если необходимо показать перемещение только одного пиксела, приходится переписывать весь символ из одного знакоместа в другое. Однако при этом, в отличие от BASIC-команды PLOT, нет ограничений на выбор цвета.

Перемещение знакоместа переднего плана экрана

Для начала необходимо определить стартовую позицию шаблона и затем, используя способы, рассмотренные в главе 5, установить шаблон в это знакоместо. Обычно начальное положение дублируется, поскольку полностью задать положение шаблона - это значит установить его текущее и последующее положения на экране дисплея. Для исключения мигания при перемещении шаблона необходимо сразу же выводить его в новой позиции целиком. Такой способ используется для задания режима OVER 1, при котором перемещение шаблона осуществляется на фоне заднего плана без изменения последнего.

Программа 7.1 демонстрирует этот способ. Здесь шаблон в виде космического корабля перемещается по экрану вверх/вниз при нажатии клавиш 1/Q и вправо /влево при нажатии клавиш 9/0. Клавиша "SPACE" используется для возврата в БЕЙСИК.

Листинг 7. 1.

АДРЕС	МЕТКА	МАШ. КОД	АСSEMBLER	КОММЕНТАРИЙ
			ORG 23760	
23760		F3	DI	; Отключение прерываний.
23761		18 18	JR START	; Переход на начало работы.
23763	DATA_1	0 0 15 24 49 226 225 224		; Шаблон символа CHR\$ 144.
23771		24 60 255 255 153 255 231 60		; Шаблон символа CHR\$ 145.
23779		0 0 240 24 140 71 135 7		; Шаблон символа CHR\$ 145.
23787	START	ED 5B 7B 5C	LD DE, (23675)	; Начало области UDG.
23791		21 D3 5C	LD HL, DATA_1	
23794		01 18 00	LD BC, 24	
23797		ED B0	LDIR	; Задание трех первых символов ; графики UDG.
23799		18 0A	JR BEGIN	; Переход на начало работы.
23801	DATA_2	22 0 0 16 8 17 8 144 145 146		; Эта последовательность ко- ; дов является символьной стро- ; кой, которую можно трактовать ; как AT 0,0; PAPER 8; INK 8; ; CHR\$ 144; CHR\$ 145; CHR\$ 146.
23811	BEGIN	FD 36 0E 28	LD (IY+14), 40	; Выставляем цвет бордюра.
23815		FD 36 53 0F	LD (IY+83), 15	; Выставляем INK и PAPER.

23819	3E 02	LD A, 02	
23821	CD 01 16	CALL 5633	; Открыли канал печати на экран.
23824	CD 6B 0D	CALL 3435	; Очистка экрана.
23827	3E 02	LD A, 02	
23829	CD 01 16	CALL 5633	
23832	11 F9 5C	LD DE, DATA_2	; Начало символьной строки.
23835	01 0A 00	LD BC, 10	; Длина символьной строки.
23838	CD 3C 20	CALL 8252	; Печать символьной строки.
23841 L7	11 00 5B	LD DE, 23296	; Начало буфера принтера.
23844	21 F9 5C	LD HL, DATA_2	; Начало символьной строки.
23847	01 0A 00	LD BC, 10	; Длина символьной строки.
23850	ED B0	LDIR	; Переброска символьной строки
			; DATA_2 в начало буфера прин-
			; тера.
23852	2A 01 5B	LD HL, (23297)	; Координаты текущей позиции
			; печати.
23855	01 FE EF	LD BC, 61438	
23858	ED 78	IN A, (C)	; Ввод данных с порта клавиату-
			; ры. Согласно тому числу, ко-
			; торое установлено в BC, здесь
			; принимается сигнал с правой
			; половины верхнего ряда клавиш
			; (клавиши 6...0).
23860	CB 47	BIT 0, A	; Проверка клавиши "0".
23862	20 01	JR NZ, L1	; Переход, если не нажата.
23864	24	INC H	; Если нажата.
23865 L1	CB 4F	BIT 1, A	; Проверка клавиши "9".
23867	20 02	JR NZ, L2	; Переход, если не нажата.
23869	25	DEC H	; Если нажата.
23870 L2	01 FE F7	LD BC, 63486	
23873	ED 78	IN A, (C)	; Опрос полуряда (1...5).
23875	CB 47	BIT 0, A	; Проверка клавиши "1".
23877	20 01	JR NZ, L3	; Переход, если не нажата.
23879	2D	DEC L	; Если нажата.
23880 L3	01 FE FB	LD BC, 64510	
23883	ED 78	IN A, (C)	; Опрос полуряда (Q...T).
23885	CB 47	BIT 0, A	; Проверка клавиши "Q".
23887	20 01	JR NZ, L4	; Переход, если не нажата.
23889	2C	INC L	; Если нажата.
23890 L4	7D	LD A, L	; Координата по вертикали.
23891	FE 16	CP 22	; Проверка на выход за пре-
			; делы экрана снизу.
23893	20 03	JR NZ, L5	; Переход, если все в порядке.
23895	2D	DEC L	; Иначе уменьшаем L
23896	18 05	JR L6	; и переходим на метку L6.
23898 L5	FE FF	CP 255	; Проверка на выход за пре-
			; делы экрана сверху.
23900	20 01	JR NZ, L6	; Переход, если все в порядке.
23902	2C	INC L	; Иначе увеличиваем L.
23903 L6	7C	LD A, H	
23904	FE 1E	CP 30	; Проверка на выход за пре-
			; делы экрана справа.
23906	20 03	JR NZ, L8	; Переход, если все в порядке.
23908	25	DEC H	; Иначе уменьшаем H.
23909	18 05	JR L9	; и переходим на метку L9.
23911 L8	FE FF	CP 255	; Проверка на выход за пре-
			; делы экрана слева.
23913	20 01	JR NZ, L9	; Переход, если все в порядке.
23915	24	INC H	; Иначе увеличиваем H.
23916 L9	ED 5B 01 5B	LD DE, (23297)	; Координаты старой позиции
			; печати.
23920	AF	XOR A	; Обнуление флага переноса (это
			; важный момент при подготовке
			; к исполнению команды SBC).
23921	E5	PUSH HL	; Запомнили координаты новой

23922	ED 52	SBC HL, DE	; позиции печати на стеке. ; Проверяем изменилась ли по- ; зиция печати.
23924	E1	POP HL	; Восстановили новую позицию.
23925	28 2C	JR Z, L12	; Если позиция печати не не- ; нялась, делаем обход на L12.
23927	22 01 5B	LD (23297), HL	; Запомнили координаты новой ; позиции печати в буфере ; принтера.
23930	FD 36 57 03	LD (IY+87), 3	; Включение OVER 1.
23934	11 F9 5C	LD DE, DATA_2	
23937	01 0A 00	LD BC, 10	
23940	CD 3C 20	CALL 8252	; Печать в старых координатах. ; а поскольку включен режим ; OVER 1, то это на самом деле ; не печать, а стирание.
23943	11 00 5B	LD DE, 23296	
23946	01 0A 00	LD BC, 10	
23949	CD 3C 20	CALL 8252	; Печать в новой позиции.
23952	11 F9 5C	LD DE, DATA_2	
23955	21 00 5B	LD HL, 23296	
23958	01 0A 00	LD BC, 10	
23961	LD B0	LDIR	; Копирование новой позиции ; в старую.
23963	21 88 13	LD HL, 5000	
23966 L11	2B	DEC HL	
23967	7C	LD A, H	Цикл задержки.
23968	B5	OR L	
23969	20 FB	JR NZ, L11	
23971 L12	01 FE 7F	LD BC, 32766	Опрос нажатия
23974	ED 78	IN A, (C)	клавиши
23976	CB 47	BIT 0, A	"ПРОБЕЛ".
23978	C2 21 5D	JP NZ, L7	; Если не нажата, переход на L7
23981	FD 36 57	00 LD (IY+87), 0	; Восстановление режима OVER 0.
23985	FB	EI	; Разрешение прерываний.
23986	C9	RET	; Возврат в БЕЙСИК.

Космический корабль может перемещаться одновременно в двух направлениях без изменения изображения заднего плана. Корабль выводится в цветах PAPER 8 и INK 8, чтобы оставаться контрастным на любом фоне заднего плана изображения. Скорость перемещения корабля регулируется с помощью цикла задержки. Если же корабль остается неподвижным, то процедура стирающая старое изображение и печатающая новое обходится, чтобы избежать мерцания.

Процедура хоть и выглядит немалой по своим размерам, тем не менее очень проста. Просмотрите ее, в ней все должно быть понятно и доступно.

Вы можете усовершенствовать эту программу так, чтобы космический корабль при выходе его за пределы экрана, появлялся с противоположной стороны.

Процедура в машинных кодах выполняется довольно быстро и если в цикле задержки записать LD HL, 1, то за время перемещения корабля его изображение не успеет полностью восстановиться на новой позиции. Поэтому для большинства программ необходимы циклы задержки, чтобы сохранить изображение на время перемещения.

Перемещение изображения заднего плана экрана

Для задания перемещения заднего плана необходимо организовать перемещение не одного знакоместа, а всего экрана или его части. Это перемещение обычно выполняется в одном из 4 направлений: вверх, вниз, вправо, влево; причем перемещаются одновременно и символы и их атрибуты. Если смещение всего экрана происходит на одну символьную позицию, то необходимо решить, как поступить с символами, вышедшими за пределы экрана. Мы можем либо потерять их и выводить на противоположной стороне экрана пустые

знакоместа, либо запомнить их вместе с атрибутами и затем выводить на экран в соответствующем месте на противоположной стороне. Такой эффект заикливания изображения весьма полезен для игровых программ.

Программы с 7.2 по 7.5 демонстрируют смещение атрибутов без возврата с противоположной стороны для всех четырех направлений (такое движение принято называть "протяжкой" - SCROLL). Программы с 7.6 по 7.9 демонстрируют смещение атрибутов по всем четырем направлениям с возвратом (назовем это движение "прокруткой" - ROLL).

"Прокрутку" и "протяжку" символов для всех четырех направлений демонстрируют программы 7.10 ... 7.18.

Листинг 7.2

SCROLL вверх для атрибутов.

АДРЕС	МЕТКА	МАШ. КОД	АССЕМБЛЕР	КОММЕНТАРИЙ
			ORG 23760	
23760		11 00 58	LD DE, 22528	; Начало области экранных ; атрибутов.
23763		21 20 58	LD HL, 22560	; Начало области атрибутов ; для второго ряда знакомест.
23766		01 E0 02	LD BC, 736	; Длина перебрасываемого блока ; (23 нижних строки по 32 зна- ; коместа).
23769		ED B0	LDIR	; Перенос атрибутов на одну ; строку вверх.
23771		06 20	LD B, 32	Заполнение нижней строки
23773		3A 8D 5C	LD A, (23693)	экрана постоянными атрибута-
23776 L2		12	LD (DE), A	ми, взятыми из системной пе-
23777		13	INC DE	ременной ATTR_P(23693)
23778		10 F2	DJNZ L2	
23780		C9	RET	; Возврат.

Эти программы легко усовершенствовать для организации ROLL и SCROLL только части экрана. Весь экран условно разбит на 3 сегмента, по 8 рядов в каждом. Каждый ряд, в свою очередь, состоит из 8 линий. Каждая линия в дисплейной области памяти занимает 32 байта. Причем записывается в эту область памяти сначала первая линия первого ряда, затем первая линия второго ряда и т.д. последовательно до первой линии восьмого ряда, включительно.

Затем записываются вторые линии первых восьми рядов, затем третьи и так последовательно весь сегмент, после чего записывается второй и за ним последовательно третий сегмент экрана. Каждый сегмент занимает 2046 байтов в дисплейной области памяти.

Обратите внимание, что в процедурах 7.3...7.18 используются команды передачи блока байтов - LDIR и LDDR. После выполнения этих команд в регистровых парах HL и DE остается записанный конечный адрес блока +1 (для LDIR) и конечный адрес - 1 (для LDDR), Это можно использовать для задания адреса продолжения выполнения программ процедуры, а не перезаписывать адреса в HL и DE.

Программы 7.2...7.18 написаны так, чтобы не вносить в них изменения при смене адреса загрузки. Программа 7.19 является усовершенствованной версией программы SCREEN UP SCROLL. Если Вам все понятно в ней, попробуйте сами усовершенствовать программу SCREEN DOWN SCROLL.

Сравнивая программы 7.15 и 7.19 Вы заметите что вторая занимает уже не 99, а только 87 байтов, хотя и это не абсолютно компактная программа. Весьма полезные детали искусства программирования, а также способы уменьшения объема памяти и времени работы показаны в программе 7.15. Эта программа перемещает вверх каждый символ ряда, линия за линией.

Листинг 7.3

SCROLL вниз для атрибутов

```
L7 LD DE, 23295
LD HL, 23253
LD BC, 736
LDDR
LD B, 32
LD A, (23593)
LD (BE), A
INC DE
DJNZ L7
RET
```

Листинг 7.4

SCROLL влево для атрибутов

```
L12 LD B, 24
LD HL, 22528
PUSH HL
POP DE
PUSH BC
INC HL
LD BC, 31
LDIR
LD A, (23693)
LD (DE), A
POP BC
DJNZ
L12 RET
```

Листинг 7.5

SCROLL вправо для атрибутов

```
L14 LD B, 24
LD HL, 23295
PUSH HL
POP DE
PUSH BC
DEC HL
LD BC, 31
LDDR
LD A, (23693)
LD (DE), A
POP BC
DJNZ, L14
RET
```

Листинг 7.6

ROLL вверх для атрибутов.

```
LD HL, 22528 ; Начало ат-
; рибутов.
PUSH HL ; Копирование
POP DE ; через стек.
LD B, 32 ; 32 знако-
; места ряда.
L16 LD A, (HL) ; Запомнили
PUSH AF ; атрибуты на
; стеке.
INC HL ; Знакомес-
; то справа.
DJNZ L16 ; Повтор для
; всего ряда.
LD BC, 736 ; Протяжка для
LDIR ; 23-х нижних
; рядов.
LD B, 32 ; Восстанавли-
L17 POP AF ; ваем атрибуты
DEC HL ; верхнего ряда
LD (HL), A ; в нижнем ряду
DJNZ L17 ; экрана.
RET ; Возврат.
```

Листинг 7.7

ROLL вниз для атрибутов

```
LD HL, 23295
PUSH HL
POP DE
LD B, 32
L24 LD A, (HL)
PUSH AF
DEC HL
DJNZ L24
LD BC, 736
LDDR
LD B, 32
L25 POP AF
INC HL
LD (HL), A
DJNZ L25
RET
```

Листинг 7.8

ROLL влево для атрибутов

```
L32 LD B, 24
LD HL, 22528
PUSH HL
POP DE
PUSH BC
LD A, (HL)
INC HL
LD BC, 31
LDIR
LD (DE), A
POP BC
DJNZ L32
RET
```

Листинг 7.9

ROLL вправо для атрибутов

```
L33 LD B, 24
LD HL, 23295
PUSH HL
POP DE
PUSH BC
LD A, (HL)
DEC HL
LD BC, 31
LDDR
LD (DE), A
POP BC
DJNZ L33
RET
```

Листинг 7.10

ROLL влево для символов.

```
L1 LD B, 192 ; Высота экрана.
LD HL, 16384 ; Начало экран-
; ной области.
PUSH HL ; Копирование
POP DE ; через стек.
PUSH BC ; Запомнили BC.
LD A, (HL) ; Запомнили те-
; кущую линию
; самого левого
; знакоместа.
INC HL ; Адрес линии
; справа.
LD BC, 31 ; 31 знакоместо
; в ряду.
LDIR ; Копирование.
LD (DE, A) ; Копирование
; линии самого
; левого знако-
; места в самое
; правое.
POP BC ; Восстановление
; счетчика линий
DJNZ L1 ; Если не все
; линии скопиро-
; ваны, то воз-
; врат на L1.
RET ; Выход.
```

Листинг 7.11

ROLL вправо для символов

```
L1 LD B, 192
LD HL, 22527
PUSH HL
POP DE
PUSH BC
LD A, (HL)
DEC HL
LD BC, 31
LDDR
LD (DE), A
POP BC
DJNZ L1
RET
```

Листинг 7.12

SCROLL влево для символов.

```
L13 LD HL, 16384
XOR A
LD B, 192
PUSH BC
PUSH HL
POP DE
INC HL
LD BC, 31
LDIR
LD (DE), A
POP BC
DJNZ L13
RET
```

Листинг 7.13

SCROLL влево для символов. (второй вариант)

```
L1 LD HL, 16384
LD DE, 32
LD B, 192
LD (HL), 0
ADD HL, DE
DJNZ L1
LD HL, 16385
LD DE, 16384
LD BC, 6143
LDIR
EX DE, HL
LD (HL), 0
RET
```

Листинг 7.14

SCROLL вправо для символов.

```
L14 LD HL, 22527
XOR A
LD B, 192
PUSH BC
PUSH HL
POP DE
DEC HL
LD BC, 31
LDDR
LD (DE), A
POP BC
DJNZ L14
RET
```

Листинг 7.15

SCROLL вверх для символов.

```
L3 LD DE, 16384
LD HL, 16416
LD BC, 2016
LDIR
LD B, 8
LD HL, 18432
LD DE, 16608
PUSH BC
PUSH HL
PUSH DE
LD BC, 32
LDIR
POP DE
POP HL
POP BC
INC D
INC H
DJNZ L3
LD DE, 18432
LD HL, 18464
LD BC, 2016
LDIR
LD B, 8
LD HL, 20460
LD DE, 18656
L4 PUSH BC
PUSH HL
PUSH DE
LD BC, 32
LDIR
POP DE
POP HL
POP BC
INC D
INC H
DJNZ L4
LD DE, 20480
LD HL, 20512
LD BC, 2016
LDIR
LD B, 8
LD HL, 20704
L5 PUSH BC
PUSH HL
LD B, 32
L6 LD(HL), 0
INC HL
DJNZ L6
POP HL
POP BC
INC H
DJNZ L5
RET
```

Листинг 7.16

ROLL вверх для символов.

```
LD HL, 16384
L18 LD B, 32
L19 LD A, (HL)
PUSH AF
INC HL
DJNZ L19
XOR A
LD L, A
INC H
LD A, H
CP 72
JR NZ, L18
LD DE, 16384
LD HL, 16415
LD BC, 2016
LDIR
LD B, 8
LD HL, 18342
LD DE, 16608
L20 PUSH BC
PUSH HL
PUSH DE
LD BC, 32
LDIR
POP DE
POP HL
POP BC
INC D
INC H
DJNZ L20
LD DE, 18432
LD HL, 18464
LD BC, 2016
LDIR
LD B, 8
LD HL, 20480
LD DE 18656
L21 PUSH BC
PUSH HL
PUSH DE
LD BC, 32
LDIR
POP DE
POP HL
POP BC
INC D
INC H
DJNZ L21
LD DE, 20480
LD HL, 20512
LD BC, 2016
LDIR
LD HL, 22527
L22 LD B, 32
L23 POP AF
LD (HL), A
DEC HL
DJNZ L23
LD A, 255
LD L, A
DEC H
LD A, H
CP 79
JR NZ, L22
RET
```

Листинг 7.17 SCROLL вниз для символов.

L8 LD DE, 22527
LD HL, 22495
LD BC, 2016
LDDR
LD B, 8
LD HL, 18656
LD DE, 20480
PUSH BC
PUSH HL
PUSH DE
LD BC, 32
LDIR
POP DE
POP HL
POP BC
INC D
INC H
DJNZ L6
LD DE, 20479
LD HL, 20447
LD BC, 2016
LDDR
LD B, 8
LD HL, 16608
LD DE, 18432

L9 PUSH BC
PUSH HL
PUSH DE
LD BC, 32
LDIR
POP DE
POP HL
POP BC
INC D
INC H
DJNZ L9
LD DE, 18431
LD HL, 18399
LD BC, 2016
LDDR
LD HL, 16384
LD B, 32
L10 LD(HL), 0
L11 INC HL
DJNZ L11
XOR A
LD L, A
INC H
LD A, H
CP 72
JR NZ, L10
RET

Листинг 7.18 ROLL вниз для символов

L26	LD HL, 22527 LD B, 32	; Конец дис- ; плейной об- ; ласти. ; 32 знакоместа ; в строке, ; Запомнили ; нижнюю линию ; текущего зна- ; коместа. ; Знакоместо ; слева.
L27	LD A, (HL) PUSH AF DEC HL DJNZ L27 LD A, 255 LD L, A DEC H LD A, H CP 79 JR NZ, L26 LD DE, 22527 LD HL, 22495 LD BC, 2016 LDDR LD B, 8 LD HL, 18656	; И так 8 раз. ; Возврат к ; предыдущей ; линии послед- ; него знако - ; места ряда. ; Проверка на ; выход в 23-ю ; строку экрана ; Если нет, то ; возврат назад ; Копирование ; информации в ; третьем сег- ; менте на ряд ; вниз. ; Восемь линий. ; Адрес начала ; первой линии ; последнего ; ряда второго

	LD DE, 20480	; сегмента.
		; Адрес начала
		; первой линии
		; первого ряда
		; третьего сег-
		; мента.
L28	PUSH BC	; Сохранили PP-
	PUSH HL	; гистры от
	PUSH DE	; порчи коман-
		; дой LDIR.
	LD BC, 32	; 32 байта в
		; каждой линии.
	LDIR	; Перенос одной
		; линии через
		; границу сег-
		; ментов.
	POP DE	; восстановле-
	POP HL	; ние регистров
	POP BC	
	INC D	; Переход к
	INC H	; очередной ли-
		; нии ряда.
	DJNZ L28	
		; Возврат для
		; переноса 8-ми
		; линий через
		; границу сег-
		; мента.
	LD DE, 20479	; Движение ин-
	LD HL, 20447	
		; формации во
	LD BC, 2016	; втором сег-
	LDDR	; менте на ряд
		; вниз.
	LD B, 8	
	LD HL, 16608	
	LD DE, 18432	Перенос
L29	PUSH BC	информации
	PUSH HL	через
	PUSH DE	границу
	LD BC, 32	между
	LDIR	первым
	POP DE	и вторым
	POP HL	сегментами.
	POP BC	
	INC D	
	INC H	
	DJNZ L29	
	LD DE, 18431	; Движение ин-
	LD HL, 18399	; формации в
	LD BC, 2016	; первом сег-
	LDDR	; менте на ряд
		; вниз.
	LD HL, 16384	; Начало экрана
L30	LD B, 32	
L31	POP AF	Восстановле-
	LD (HL), A	ние со стека
	INC HL	информации
	DJNZ L31	для восьми
	XOR A	линий первого
	LD L, A	ряда (на стек
	INC H	она поступила
	LD A, H	из восьми
	CP 72	нижних линий

JR NZ, L30	экрана).
RET	; Выход.

Листинг 7.19

SCROLL вверх для символов (2-ой вариант)

	LD HL, 16384	; Начало дис-
		; плейной об-
		; ласти памяти.
	CALL SEGM	; Движение пер-
		; вого сегмента
	EX DE, HL	Перенос
	LD HL, 18438	ряда через
	PUSH HL	границу
	CALL UP	сегментов.
	POP HL	
	CALL SEGM	; Движение вто-
		; рого сегмента
	EX DE, HL	Перенос ряда
	LD HL, 20480	через границу
	PUSH HL	сегментов.
	CALL UP	
	POP HL	
	CALL SEGM	; Движение тре-
		; тьего сегмен-
		; та.
L4	LD HL, 20704	Обнуление
	PUSH HL	(стирание)
L3	LD 8, 32	информации
	LD (HL), 0	в последнем
	INC HL	ряду
	DJNZ L3	третьего
	POP HL	сегмента.
	INC H	
	LD A, H	
	CP 88	
	JR NZ, L4	
	RET	; Выход

Продолжение Л. 7.19.

Нижеследующая процедура копирует все ряды (кроме верхнего) в текущем сегменте на один ряд выше.

SEGM	LD B, 7	; Первый ряд сег-
		; мента не пере-
		; носим, потому 7
L1	PUSH BC	; Сохранили от
		; "порчи".
	LD BC, 32	; Соответствующая
		; линия следующе-
		; го ряда отстоит
		; на 32 байта от
		; текущей.
	PUSH HL	; Копирование ад-
	POP DE	; реса через стек
	ADD HL, BC	; HL указывает на
		; ряд ниже
	PUSH HL	; Сохранили HL и
		; DE от "порчи"
	PUSH DE	; командой LDIR
	CALL UP	; Копирование ря-

		; да вверх.
POP DE		
		Восстановление
POP HL		регистров.
POP BC		
DJNZ L1		; Если не все 7
		; рядов скопиро-
		; ваны, возврат.
RET		; Выход.

Нижеследующая процедура копирует содержимое экранного ряда на один ряд выше.

UP	LD B, 8	; 8 линий в ряду.
L2	PUSH BC	; Сохранили регис-
	PUSH HL	; тры от "порчи"
	PUSH DE	; командой LDIR.
	LD BC, 32	; 32 байта в линии
	LDIR	; Перенос на один
		; ряд вверх.
	POP DE	
	POP HL	; Восстановление
	POP BC	; регистров.
	INC D	; Переход к оче-
		; редной линии
	INC H	; текущего ряда.
	DJNZ L2	; Если не все 8
		; линий скопирова-
		; ны, возврат.
	RET	; Выход.

Вместо программы, выполняющей перемещение сразу же целого ряда, можно написать программу, выполняющую перемещение только одного столбца. Чтобы переместить весь ряд, эта программа должна повториться 32 раза. В то же время, такое решение позволит перемещать не весь ряд, а только заданное количество столбцов и, более того, при таком решении нет необходимости хранить в памяти все 256 байтов верхнего ряда экрана (как, например, в программе 7.18), а только 8 байтов одного столбца ряда. Программа 7.20 показывает, как достигается такое уменьшение объема памяти. Здесь адрес первого байта смещаемого столбца записывается в регистровую пару DE, а общее число смещаемых столбцов записывается в регистр B. Эта программа может быть размещена в любом участке памяти.

Листинг 7.20.

SCROLL вверх по столбцам.

	XOR A	; Сброс ак-ра.
	LD DE, 16388	; Адрес верхней
		; линии 4-го
		; столбца.
	LD B, 10	; Двигаем вверх
		; 10 столбцов.
L5	PUSH DE	; Сохранение ре-
		; гистров
	PUSH BC	; от корроупции.
	PUSH DE	; Копирование из
	POP HL	; DE в HL.
	LD B, 8	; 8 линий ряда.
L6	LD A, (HL)	; Запоминаем на
	PUSH AF	; стеке восемь
	INC H	; линий верхнего
	DJNZ L6	; ряда.
L4	LD B, 7	
		; 7 рядов сег-
		; мента.

L2	PUSH BC LD HL, 32 ADD HL, DE	; Указание на ; следующую ли- ; нию ряда.
	PUSH HL LD B, 8	; Восемь линий ; в ряду.
LS	LD A, (HL) LD (DE), A INC H INC D DJNZ L1 POP DE POP BC DJNZ L2	; Копирование ; через акк-р ; для одного ; ряда ; Восстановление ; регистров. ; Повтор для ; очередного ря- ; да в сегменте.
	LD HL, 1824 ADD HL, DE LD A, H CP 88 JR Z, END PUSH HL LD B, 8	; Переход через ; границу сегм. ; Проверка на ; конец экран- ; ной области.
L3	LD A, (HL) LD (DE), A INC H INC D DJNZ L3 POP DE JR L4	; Перенос восьми ; линий через ; границу сег- ; мента. ; Возврат для ; работы со сле- ; дующий сегм.
END	LD DE, 32 SBC HL, DE LD B, 8	; Восстановле- ; ние со стека в ; нижнем ряду
L7	POP AF LD (HL), A DEC HL DJNZ L7 POP BC POP DE INC DE	; столбца того, ; что запомнили ; в самом верх- ; нем ряду. ; Восстановление ; регистров. ; Переход к оче- ; редному столб- ; цу экрана.
	DJNZ L5	; Если не все 10 ; столбцов об- ; служены, воз- ; врат на L5.
	RET	; Выход.

Дальнейшее уменьшение объема памяти, занимаемого программой, может быть достигнуто, если выполнять перемещение одновременно не всех рядов, а только заданного их количества. При этом можно реализовать смещение на экране определенного окна с заданными размерами. Программа 7.31, занимающая всего 58 байтов, демонстрирует это решение. Для проверки выхода на границу сегмента экрана здесь используется команда BIT 0, H. Если нулевой бит регистра H не равен 0, то мы находимся на границе сегмента экрана и вместо добавления числа 32 к числу, записанному в регистровой паре DE, необходимо добавить число 1824 для перевода "первого" ряда в следующий сегмент. Число смещаемых рядов определяется числом шагов цикла L4. Адрес ячейки верхнего левого угла смещаемого окна может быть определен по формуле:

$$16384 + (1824 * \text{INT}(\text{ROW}/8)) + ((\text{ROW} - \text{INT}(\text{ROW}/8)) * 32 + \text{COLUMN})$$

Здесь ROW - номер экранной строки (в знакоместах).

COLUMN - номер экранного столбца (в знакоместах).

Для смещения всего экрана в регистровой паре DE должно быть записано число 16384, в регистре В перед циклом L6 число 32 и здесь же, но перед циклом L4 - число 23. Эту программу можно переделать для организации смещения вверх/вниз.

Листинг 7.21.

```

XOR A
LD DE, 16515      ; Левый верхний
                  ; угол прокручи-
                  ; ваемого "окна"
                  ; имеет коорди-
                  ; наты X=3;
                  ; Y=4.
LD B, 7           ; Ширина "окна"
                  ; - 7 знакомест.
L5    PUSH DE
      PUSH BC
      PUSH DE
      POP HL
      LD B, 8
L1    LD A, (HL)
      PUSH AF
      INC H
      DJNZ L1
      LD B, 13    ; Высота "окна"
                  ; - 13 знакомест
L4    PUSH BC
      LD HL, 32
      ADD HL, DE
      BIT 0, H
      JR Z, L2
      LD HL, 1824
      ADD HL, DE
L2    PUSH HL
      LD B, 8
L3    LD A, (HL)
      LD (DE), A
      INC D
      INC H
      DJNZ L3
      POP DE
      POP BC
      DJNZ L4
      LD B, 8
L5    DEC H
      POP AF
      LD (HL), A
      DJNZ L5
      POP BC
      POP DE
      POP DE
      INC DE
      DJNZ L6
      RET
```

В ПЗУ компьютера ZX Spectrum имеется одна из наиболее полезных процедур - SCROLL n, позволяющая смещать n строк вместе с их атрибутами. Число n должно быть больше или равным 2 (но, конечно, не больше, чем 24).

Программа 7.22 показывает, как можно использовать эту процедуру для очистки строк с 16-ой по 24-ю включительно. Число строк отсчитывается от 24-й, при этом в регистр В записывается число строк-1 и девять раз вызывается процедура SCROLL n, находящаяся по

адресу 3584.

Листинг 7.22

```
LD B, 9
L1  PUSH BC
    LD B, 8
    CALL 3584
    POP BC
    DJNZ L1
    RET
```

Ну, и в конце этой главы разберем процедуру, позволяющую одновременно смещать влево (или вправо) сразу же половину символа. Эта процедура хоть и короткая, но довольно сложная, т.к. здесь используются новые команды. Например, команда RLD перемещает биты с 0-го по 3-й в биты с 7-го по 4-й в ячейке, адрес которой находится в регистровой паре HL. Биты же с 7-го по 4-й из этой ячейки переписываются в биты с 0-го по 3-й в регистр A. А данные, которые были в этом регистре в битах с 0-го по 3-й переписываются в ячейку, адрес которой находится в регистровой паре HL, и занимают здесь соответственно 0...3 биты. Эта команда не влияет на флаги состояния процессора.

Листинг 7.23

```
DI
LD HL, 22527
LD B, 196
L1  PUSH BC
    XOR A
    LD B, 32
    PUSH HL
L2  RLD
    DEC HL
    DJNZ L2
    POP DE
    EX DE, HL
    ADD A, (HL)
    LD (HL), A
    EX DE, HL
    POP BC
    DJNZ L1
    EI
    RET
```

В программе 7.23 для каждой строки экрана используется команда RLD для перемещения правой половины каждого символа в левую часть. Левая половина первого символа при этом остается пустой, т.к. для каждой строки изначально в регистре A записан 0. Чтобы организовать циркулярное смещение, необходимо временно запоминать значение левой половины первого символа, а затем выводить ее, как правую половину последнего символа.

НАША ПРЕЗЕНТАЦИЯ

Сегодня мы представляем Вам нашу новую книгу "30 часов БЕЙСИКа для начинающих". Надо сказать, что это наш первый опыт в том смысле, что до сих пор мы издавали только книги, написанные собственными руками. Сейчас, увы, дело обстоит так, что Вы, уважаемые друзья, читаете быстрее, чем мы пишем и потому нам пришлось обратиться к классике. Поводом для этого стали многочисленные обращения дать что-то для начинающих или хотя бы повторить наше самое первое издание, с которым мы дебютировали в 1989 г. - "Большие возможности Вашего "Спектрума".

Мы сразу отказались от идеи повтора нашей первой книги, ведь за эти годы она была перепечатана многими организациями, где слово в слово, где с добавлениями и, вполне возможно, она Вам встречалась под другим наименованием. Вместо этого мы сделали перевод с книги Клайва Пригмора (Clive Prigmore) и Пола Шрива (Paul Shreeve), которая является методическим пособием по "Спектруму" для Британского Национального колледжа заочного повышения квалификации (National Extension College).

Книга готова и ее оригинал-макет на дискете предлагается всем желающим самостоятельно заняться печатью по нашей лицензии и распространением. Мы же ждем ее выход из типографии не раньше ноября текущего года.

В этой книге нам больше всего понравился ход мысли автора. Все основные операторы БЕЙСИКа даются читателю на основе конкретных практических задач. Сегодня в качестве презентации новой книги мы даем одну маленькую главу, которая, как нам кажется, хорошо иллюстрирует методический подход автора. Глава посвящена членению стрингов оператором (...ТО...).

5.6 Расчленение стрингов.

Мы никогда не задумывались о том, что такое дата? Например, как компьютер представляет себе 23 июня 1973 года?

Мы знаем, что компьютер умеет хранить данные. И делает это двумя способами - в виде чисел или в виде символьных строк (стрингов). В виде чисел он уделяет по 5 байтов памяти на каждое число, а для стрингов - по одному байту на каждый символ. Так что же такое дата? Число или стринг?

23 июня 1973 года - вроде бы стринг, а если мы запишем то же самое как 230673 - то вроде бы это число. Давайте разбираться.

Если 230573 это число, то почему 110593 (11 мая 1993 года) меньше, ведь 93-ий год был позже 73-го? А если это стринг, то спрашивается, как компьютер сможет рассчитать, сколько лет будет в 93-м году молодому человеку, родившемуся в 73-м, да и вообще как тогда оперировать с календарем?

Ответ очень прост. Даты - это особый вид информации и работать с ними надо особо. Их надо вводить, как стринги, а хранить в памяти и обрабатывать надо как числа. И вводят их именно как стринги еще и потому, что не все элементы даты нам одинаково важны.

Например, если наша организация ежегодно в декабре месяце повышает заработную плату тем, кто отработал у нас более десяти лет, то для этого важен только год, в котором работник был зачислен. Если же наша организация занимается настройкой пианино и роялей, то каждые три месяца мы рассылаем всем своим зарегистрированным клиентам письмо-напоминание о том, что надо бы обслужить инструмент. В этом случае нас интересует только месяц, когда был последний раз обслужен данный клиент, а год и день не имеют никакого значения.

Наконец, для отдела жалоб и претензий, которому полагается разбираться с каждым заявлением в 30-ти дневный срок, важен день и месяц поступления заявления.

Итак, мы видим, что стринг, содержащий дату, имеет важное значение как единое целое, но есть серьезные причины для того, чтобы вырезать из него отдельные части и обрабатывать их отдельно. Таким образом, после того, как дата была введена в формате:

ДДММГГ или

ДД.ММ.ГГ или

ММ.ДД.ГГ или как угодно, нам нужно иметь возможность "вырезать" из нее отдельно дни, месяцы и годы и работать с ними порознь, но теперь уже как с числами.

Для этого нужен специальный оператор, и он есть. На "Спектруме" это делает оператор

(...TO...).

Его полная форма, например, такая:

x\$ (a TO b)

Здесь из строки x\$ вырезается его часть, начинающаяся с символа, номер которого равен a и до символа с номером b.

Если x\$ = "СПЕКТРУМ", то

x\$ (5 TO 7) = "ТРУ"

Вы можете посмотреть, как это происходит, на примере следующей программы:

```
12 INPUT "Введите исходное слово"; m$
16 PRINT m$
20 INPUT "Начало вырезки"; a
40 PRINT a
50 INPUT "Конец вырезки"; b
60 PRINT b
80 LET n$=m$(a TO b)
90 PRINT m$;"(";"a;" TO ";"b;"")="; n$
100 STOP
```

В некоторых других языках программирования, а также в некоторых других версиях БЕЙСИКа Вы найдете еще несколько операторов для вычленения субстроки из строки. Нам бы не стоило на них и останавливаться, но Вы должны знать, как Вам реализовывать те же возможности, раз этих операторов у Вас нет.

LEFT\$ (x\$,k) - дает Вам k левых символов строки. На "Спектруме" его аналогом будет:

x\$(1 TO k).

RIGHT\$ (x\$,k) - дает k правых символов строки. На "Спектруме" аналог подобрать труднее, приходится пользоваться функцией LEN, которую мы рассмотрели ранее:

x\$(LEN (x\$)-k+1 TO LEN (x\$))

MID\$(x\$,k,m) - дает Вам m символов, начиная с k-го символа. С этим оператором дело обстоит проще:

x\$(k TO k+m)

* * *

Книга написана в стиле дружеского диалога, имеет большое количество вопросов для самоконтроля и упражнений с решениями. Она доставит удовольствие не только тем, кто сам осваивает БЕЙСИК, но и тем, кто передает свой опыт другим.

Общий объем книги довольно велик, но мы "уложили" его в 208 стр. стандартного формата.

Спешите приобрести лицензию, а мы в это время планируем издание собственной книги "БЕЙСИК для подготовленного пользователя".

Ваш "ИНФОРКОМ".

FORUM

В прошлом выпуске "ZX-РЕВЮ" мы объявили о создании совместно с Государственным Политехническим музеем АССОЦИАЦИИ ПОЛЬЗОВАТЕЛЕЙ ДОМАШНИХ КОМПЬЮТЕРОВ и Фонда "ФЛОП" (фонд литературы, описаний и программ) и читатели уже с интересом откликнулись на это начинание.

Нам пишут Д. Палтусов и В. Кутин из г. Екатеринбурга:

Здравствуйте, господа!

Мы хотим выразить вам глубокую признательность. Дело, начатое Вами позволит сформировать наконец-то в России рынок программ, периферии и литературы для машин класса "Spectrum". Теперь люди, использующие компьютер не только для игр, смогут рассчитывать, что "Родина их не забудет".

ИФК: Большое спасибо на добром слове. Мы передаем все письма с пометкой "ФЛОП" в Политехнический музей. Надо сказать, что это их инициатива в создании такого фонда. Мы же только оказываем посильную помощь в его становлении, пользуясь тем авторитетом, который "ZX РЕВЮ" имеет по стране.

И главное, что мы хотели бы видеть в итоге это не совсем рынок программ и периферии. Скорее нам видится будущее, как свободный рынок талантливых людей. А программы, устройства и прочее - это вроде как визитная карточка этих талантов, способных делать большие дела и получать достойную оплату за свой труд.

KOPP: Мы готовы предложить некоторые свои работы. Это годы работы и килобайты кода:

1. Base - игровая программа (92).
2. Constructor - конструктор для создания игр (92).
3. Generator - редактор спрайтов (91).
4. Instal - дисассемблер (93).
5. Mega-gen - редактор спрайтов (93).
6. Multiedit+Utility - редактор мультфильмов (92).
7. Shareholder - игра (92).
8. Turbo-Copy - копировщик (93).

ИФК: Вашими разработками мы открываем раздел "Авторская программа" (см. соответствующий раздел).

KOPP: Вместе с тем, у нас есть несколько разработок, которые нельзя считать авторскими, поскольку мы просто адаптировали или доработали фирменные программы. Мы полагаем, что авторские права на них мы заявлять не можем. Может быть Вы что-то подскажете?

Это русифицированная и доработанная версия графического редактора "ARTSTUDIO", русифицированные аналоги игр серии DIZZY Оливера Твинса, доработанная версия АССЕМБЛЕРА "ZEUS".

ИФК: Дорогие друзья, мы не сильны в юриспруденции, но знаем, что кроме имущественного авторского права есть еще и неимущественное моральное. А это, как нам кажется, то, что всегда останется с Вами, если Вы знаете, что сделали большое и нужное дело, то почему бы об этом не знать и другим?

Никто не претендует на авторские права Шекспира, когда переводит "Ромео и Джульетту", но разве мы бы познакомились с его произведениями, если бы Маршак и Пастернак не приложили к делу свой талант и переводчика и поэта? И разве сегодня мы не чтим память и того и другого?

А знаете, сколько добавляют и отнимают режиссеры, когда ставят пьесу или снимают фильм по тому же Шекспиру? Конечно, имя Шекспира всегда стоит на афише, но выдающихся режиссеров народ тоже знает и помнит.

Нам кажется, что адаптация программ вещь безусловно нужная, полезная и

стимулировать ее расширение желательно. Тем более, что большинство программистов начинают именно с этого и занимаются этим до тех пор, пока наберутся достаточного опыта для создания собственных программ.

Если создать вокруг адаптации ореол "второсортности", то можно сильно подорвать желание будущих творцов двигаться вперед, что-то изучать, создавать и творить. Интересы нашего общего дела от этого только пострадают. Поэтому мы думаем, что для авторских адаптаций надо ввести свой раздел, что мы и сделаем.

Поэтому пожалуйста пришлите "объективки" на Ваши адаптации (название, автор, дата и т.п.) и мы включим их в раздел.

* * *

Нам пишет Скворцов Владимир Николаевич из города Чебоксары:

Ассоциация пользователей домашних компьютеров - это как раз то, что надо, это здорово! Ведь сколько любителей компьютеров предоставлены сами себе. И у каждого есть свои находки, свои открытия в любимых программах и компьютерах. И как часто хочется поделиться со своими единомышленниками находками и открытиями. Поэтому готов сотрудничать по любым вопросам, связанным с деятельностью Ассоциации.

* * *

Нам пишут из г. Ташкента:

Привет, ИНФОРКОМ!

Мы, Сметанин Юрий и Ильичев Андрей, проведя некоторые исследования, хотели бы внести свой небольшой вклад в досье игры ELITE. К сожалению, мы не получаем "ZX-РЕВЮ" за 1993 год, а хотелось бы, поэтому не знаем, может быть эта работа кем-то уже сделана.

ИФК: Сейчас, когда работа почты стабилизировалась, мы вновь принимаем на себя ответственность отправлять "ZX РЕВЮ" за пределы России и подписку от читателей из республик СНГ принимаем и на 93-ий год и на 94-ый, хотя, правда, для них это стоит существенно дороже. Мы надеемся, что наши читатели скооперируются, ведь "синклеристы" не водятся поодиночке, а ведут социально-активный образ жизни и выписывать "РЕВЮ" они смогут.

КОРР: ... Итак, мы выяснили значение почти всех кодов в символьном наборе программы. (Список кодов прилагается).

ИФК: Сведений о полном исследовании кодов к нам не поступало, хотя Вы вероятно знаете о том, что Д.П. Шилин проводил исследование кодировки слогов текстовых сообщений (ZX-РЕВЮ N5,6 1992г.)

КОРР: ... Значение некоторых кодов, а именно 2,3,4,127 выяснить не удалось. При попытке их распечатать программа "зависает", а некоторые коды 7,10,95... не имеют печатного символа. Возможно это какие-то управляющие коды.

Интересны коды 163,167,192. Их значения непостоянны. На место знака (#) вставляется имя планеты, на которой Вы находитесь. В коде 199 - Galactic Chart - дальше печатается номер галактики, в которой Вы находитесь.

Как видно из анализа набора кодов, абсолютно все сообщения состоят из 1-2 кодов, отсюда становится возможной русификация программы. Над этим мы сейчас и работаем.

Код	Значение	115	DOCKING COMPUTER
1	LAVE	116	GALACTIC HYPERSPACE
2'		117	MILITARY LASER
3'		118	MINING LASER
4'		119	
5	FUEL:0.0 LIGHT YEARS	120	INCOMING MISSILE
6	CASH 7	121	ENERGY
8	0.0	122	GALACTIC
9	CR	123	DOCKING COMPUTER ON
10		124	ALL
11		125	CLOAKING DEVICE
12	LL	126	COMMANDER
13		127	
14		128	AL
15	MAY DAY! INVADED	129	LE
16	ON	130	XE
17	EQUIPMENT:	131	GE
18	: CLEAN	132	ZA
19	: OFFENDER	133	CE
20	: FUGITIVE	134	BI
21	HARMLESS	135	SO
22	MOSTLY HARMLESS	136	US
23	POOR	137	ES
24	AVERAGE	138	AR
25	ABOVE AVERAGE	139	MA
26	COMPETENT	140	IN
27	DANGEROUS	141	DI
28	DEADLY	142	RE
29	--- ELITE ---	143	A
30	PRESENT	144	ER
31	FUEL LEAK	145	AT
32		146	EN
33	32 38	147	BE
34		148	RA
35	символы	149	LA
36		150	VE
37	псевдографики	151	TI
38		152	ED
39-58	Совпадает со стандартным набором	153	OR
59-62	Графика	154	QU
63	? (вопросит. знак)	155	AV
64	_ (подчеркивание)	156	TE
65-90	Совпадает со стандартным алфавитом	157	IS
91-94	Графика	158	RI
95		159	DN
96	FRONT	160	FUEL SCOOPS ON
97	REAR	161	CHART
98	LEFT	162	GOVERNMENT
99	RIGHT	163	DATA ON (#)
100	ENERGY LOW	164	INVENTORY
101	RIGHT ON COMMANDER	165	SYSTEM
102	EXTRA	166	PRICE
103	PULSE LASER	167	(#) MARKET PRICES
104	BEAM LASER	168	INDUSTRIAL
105	FUEL	169	AGRICULTURAL
106	MISSILE	170	RICH
107	LARGE CARGO BAG	171	AVERAGE
108	E.C.M. SYSTEM	172	POOR
109	EXTRA PULSE LASERS	173	MAINLY
110	EXTRA BEAM LASERS	174	UNIT
111	FUEL SCOOPS	175	VIEW
112	ESCAPE POD	176	QUANTITY
113	ENERGY BOMB	177	ANARCHY
114	ENERGY UNIT	178	FEUDAL
		179	MULTI-GOVERNMENT

180	Dictatorship	218	Firearms
181	Communist	219	Furs
182	Confederacy	220	Minerals
183	Democracy	221	Gold
184	Corporate State	222	Platinum
185	Ship	223	
186	Product	224	Gem Stones
187	Laser	225	Alien Items
188	Human Colonial	226	Cr
189	Hyperspace	227	Large
190	Short Range Chart	228	Fierce
191	Distance	229	Small
192	Hyperspace (#)	230	Green
193	Hyperspace Range!	231	Red
194	May Day! Coriolis in Danger !	232	Yellow
195	Light Years	233	Blue
196	Tech. Level	234	Black
197	Cash	235	Harmless
198	Billion	236	Slimy
199	Galactic Chart	237	Bug-eyed
200	Target Lost	238	Horned
201	Missile Jammed	239	Bony
202	Range	240	Fat
203	St	241	Furry
204	Quantity of	242	Rodent
205	Sell	243	Frog
206	Cargo	244	Lizard
207	EQUIPT	245	LOBSTER
208	Food	246	BIRD
209	Textiles	247	HUMANOID
210	RADIOACTIVES	248	FELINE
211	Slaves	249	INSECT
212	Liquor/Wines	250	Average Radius
213	Luxuries	251	COM
214	Narcotics	252	COMMANDER
215	Computers	253	DESTROYED
216	Machinery	254	RO
217	Alloys	255	UNIT QUANTITY

Коды, отмеченные знаком ('), вызывают "зависание" программы при попытке их вывести на печать. В символах 163,167,192 на место знака (#) вписывается имя планеты, на которой Вы находитесь.

Символы 7, 10, 11, 13, 14, 35, 119 не имеют печатного обозначения и печатаются как пробел.

До свидания! Успехов Вам!

* * *

Нам пишет Сороченко Сергей Владимирович из г. Норильска:

Здравствуйте, редакция "ZX РЕВЮ"! Я большой поклонник компьютера "SPECTRUM" и Вашего журнала. Я занимаюсь исследованием сложных имитаторов, адвентюрных и стратегических игр. В этом письме я хочу предложить мое описание игры FIGHTER BOMBER, сделанной фирмами Vector Graphics и Activision.

FIGHTER BOMBER - это имитатор самолета-штурмовика. В этой игре прекрасная графика, идентичная графике программы HARD DRIVING, сделанной фирмой DOMARK.

ИФК: Огромное спасибо Сергей Владимирович и за теплые слова и за Ваш труд. Вашу работу мы напечатали в разделе "Советы экспертов". Вы знаете, что мы очень любим программы именно этих жанров и попали в самое сердце.

В Вашей работе нам особенно понравилось то, что Вы не просто "раскрутили" порядок работы с программой, но и критически обсудили некоторые особенности ее работы. Всегда приятно иметь дело с творческим подходом.

KOPP: ... В заключение примите, пожалуйста, мои поправки к Вашему описанию игры ТОМАНАВК. Я не знаю, что на карте обозначено цифрами, но это не Ваши базы, так как при выходе на них я не обнаружил на земле ничего, а при посадке по координатам никакого ремонта не было. Ремонт, заправку и перевооружение производят только на вертолетных площадках, которых в каждом квадрате четыре.

И еще. Повреждение TADS - это не повреждения хвостового оперения, а повреждение прибора, показывающего, какая цель находится в перекрестке прицела.

ИФК: Ну вот, Вы нас и прокололи. А может, мы специально такую пенку сделали? Вдруг они там в Пентагоне тоже "ZX-РЕВЮ" читают? Может, мы хотели оставить полки "Томагавков" без ремонтной базы, а теперь они обо всем догадаются. Ну да ладно, сдерем с них пару сотен миллионов ихних папуасских долларов за это "ноу-хау" и купим себе новый джойстик.

KOPP:... В следующем письме я хочу предложить описание игры OCEAN CONQUEROR фирмы Digital Integration.

ИФК: Огромное спасибо. От души поиграем, как раз и новый джойстик опробуем.

Военнослужащий Похилук М.П. из г. Славгород Алтайского края задает следующий вопрос:

"В нашем небольшом городе образовалось нечто вроде клуба любителей "Спектрума" и у некоторых его членов имеются компьютеры с ПЗУ "ТУРБО-90", у большинства же, как и у меня, ПЗУ стандартное. Преимущества "ТУРБО-90" налицо, и мы хотели бы иметь такое у себя, но специалистов в этой области у нас в городе мы не знаем, а вот сможем набрать программу, которая делала бы то же самое, что ПЗУ "ТУРБО-90", но работала бы где-нибудь в свободной области ОЗУ мы смогли бы".

ИФК: Сделать то же самое, что делает встроенный монитор этой прошивки в принципе можно. Можно "вытащить" монитор из ПЗУ и расположить его в ОЗУ, изменив адресацию в машинных кодах. Но это теряет смысл, так как любая программа-монитор, расположенная в ОЗУ, может находиться как раз в той же области памяти, в которой находится исследуемая программа. Ведь заранее совершенно неизвестно, какая область памяти останется "свободной" после загрузки интересующей программы. Прелесть "ТУРБО-90" в том и состоит, что монитор не в ОЗУ, а в ПЗУ. При этом все ОЗУ свободно и при помощи монитора, находящегося в ПЗУ Вы можете "забираться" в любые уголки памяти и исследовать любые программы, загруженные в ОЗУ. Извлекать же монитор из ПЗУ для размещения его в ОЗУ нет необходимости, ведь это не самый лучший монитор. Существует достаточно много программ-мониторов, выполняющих все те же функции, что и монитор "ТУРБО-90" плюс еще много других. Например, "MONS-4" ("MONS-3") из пакета программ "DEVPAC". Это довольно старая программа, но она до сих пор является непревзойденной по своим возможностям. "MONS", кстати, устроен так, что может работать в любых адресах. То есть, если Ваша исследуемая программа расположена в верхней части памяти, Вы можете загрузить "MONS" в нижнюю часть и наоборот. К тому же, при помощи монитора "ТУРБО-90" Вы не имеете возможности дизассемблирования программы, а это очень немаловажная функция при исследовании кодов программ.

К сказанному следует добавить еще один любопытный факт. В свое время мы сообщали о мониторе, работающем в ОЗУ, который находится в игре "EQUINOX". По своим функциям и управляющим клавишам он поразительно похож на монитор "ТУРБО-90" (или монитор "ТУРБО-90" похож на монитор "EQUINOX"). Если Вас интересует эта информация, то подробности смотрите "ZX-РЕВЮ" N11-12, за 1991 г. на стр. 254.

И, наконец, последнее. Мы очень рады, что в Вашем городке образовалось "нечто вроде клуба любителей "Спектрума". Знаем мы, что то же самое есть и в сотнях других больших и малых городов. Мы много раз уже писали, что мы приветствуем неформальное объединение синклеристов, что это одна из наших главных целей, что мы всегда готовы абсолютно бесплатно и без очереди давать объявления о создании и работе клубов, но почему-то никто этим не пользуется. Разве наши читатели не хотят наладить связь между клубами напрямую? Разве это не нужно? Конечно, кое-какую информацию о своем компьютере Вы получаете из "ZX-РЕВЮ", но мы ведь не можем объять необъятное.

Объединяйтесь, обменивайтесь информацией напрямую, мы не боимся конкуренции, а наоборот окажем любую помощь для свободной циркуляции информации между клубами.

* * *

По поводу ПЗУ "ТУРБО-90" пришло еще одно письмо от нашего читателя из Ульяновска - Силантьева М.

Он исследовал эту прошивку методом "теневого ПЗУ". Таким способом это делать очень удобно, если в Вашем компьютере предусмотрена такая возможность - теневое ПЗУ, загружаемое в ОЗУ. Для тех, кто не знаком с этим, сообщаем, что такая возможность есть у компьютеров, имеющих две "линейки" микросхем памяти - отдельно основное ОЗУ и отдельно экранная память. При этом, если после рестарта компьютера, сделать PEEK ADR, где ADR - 0...16383, то читаться будет ПЗУ, но если подать команду POKE ADR, то будет сделана запись в область памяти ОЗУ, являющейся "теневой" для ПЗУ (в данный момент она неактивна и никак себя не проявляет). Можно даже сделать SAVE "SYS"CODE 0,16384 на компьютере, имеющем, например, монитор "ТУРБО-90". А потом загрузить его в свой компьютер, (имеющий стандартный монитор) командой LOAD "SYS"CODE 0. Для того, чтобы переключиться на эту "теневую" прошивку, наш читатель Силантьев М. пользуется тумблером. В "Новосибирском" варианте такое переключение выполняется командой OUT 15,0. При этом переключается триггер, отключающий основное ПЗУ и подключающий теневую область ОЗУ. При этом активным становится загруженный монитор, а основное ПЗУ - пассивным. Теперь можно воспользоваться дополнительными возможностями, заложенными в этой прошивке. Для того, чтобы переключиться обратно на основное ПЗУ, надо опять выполнить команду OUT 15,0. Определить, какое ПЗУ в данный момент является активным, можно, например, по начертанию символов: В "ТУРБО-90" оно несколько отличается. Можно подать команду NEW, тогда будет выведено фирменное сообщение об используемой версии.

После этого отступления мы возвращаемся к письму нашего корреспондента.

КОРР: "У меня обнаружилась одна неприятность. Первые 5 байтов теневого ПЗУ самопроизвольно изменяются после первого же обращения к ним. Если попытаться вывести значение этих ячеек командой:

```
FOR f=0 TO 10: PRINT PEEK f: NEXT f
```

то первые 5 байтов имеют нулевые значения. А если поочередно давать команды:

```
PRINT PEEK 0
```

```
PRINT PEEK 1
```

то печатаются неизвестно откуда взявшиеся числа. При попытке изменить их они самопроизвольно восстанавливаются. Дизассемблирование этих первых 5 байтов дает какую-то абракадабру. Из за этого неправильно обрабатывается подпрограмма RESET (RANDOMIZE USR 0)

ИФК: Мы проверили то, о чем пишет наш читатель на "Новосибирском" варианте Спектрума и убедились, что этот компьютер ведет себя в точности так же. В чем же тут дело? Может, кто-нибудь из читателей сталкивался уже с подобной проблемой? Или хотя бы знает причину этого явления? Если такие есть, откликнитесь, пожалуйста.

* * *

Дорогие читатели. Вы помните заметку "МАЛЕНЬКИЕ ХИТРОСТИ" в 5-6 выпуске "РЕВЮ" за этот год, где на стр.96 было опубликовано сообщение Радзевича А. А. из г.Нальчика? Оно касалось любопытных эффектов, возникающих при исполнении оператора DRAW. Пока никто не ответил на вопрос "Как это происходит?" Но мы со своей стороны провели небольшой розыск по зарубежным журналам и оказалось, что этой проблемой занимаемся не только мы. Вот что мы нашли в июньском номере журнала "SINCLAIR USER" за 1986 г. (No. 51) на стр. 15. Там приводится несколько иной вариант программы, воспроизводящей такие эффекты.

```
5 LET n=1
```

```
10 PAPER 0: INK 4: BORDER 0: CLS
20 FOR a=51 TO 10000 STEP 100
25 LET n=n+0
30 PLOT 65, 30
40 DRAW INVERSE n; 120, 120, PI*a
50 NEXT a
```

Были и другие публикации в этом журнале, например N5,1986, но суть не меняется - все это разные модификации той же идеи. К сожалению, намеков на причины происходящих на экране эффектов там нет. Итак, проблема остается открытой, хоть и приобрела международный характер.

* * *

Нам пишет Лярский С.С. из пос. Березовский Свердловской обл:
Уважаемые коллеги!

Разрешите мне поделиться с Вами своими соображениями по поводу дешифрации портов в "Ленинграде". Если человек впервые собирает такую конструкцию как компьютер и еще если он его сумеет довести до того состояния, в котором он мало-мальски работает, - этот человек, конечно, радуется тому, что у него есть компьютер, что игрушки играют, что программы работают и т. д. Но это только первое время. Дальше наступает непреодолимое желание усовершенствовать свою машины, расширить ее возможности, а если еще у Вас такая "сырая" штукovina, как первый выпуск "Ленинграда" - бери в руки паяльник, напрягай мозги и "вперед с песней".

Усовершенствований хватит на всю оставшуюся жизнь. Это, конечно же, не упрек авторам Ленинградской версии.

Очевидно, их задачей было сделать наиболее простую, дешевую и более-менее "Спектр"-совместимую машину. Возможно, им и удалось решить задачу простоты, но что касается дешевизны и совместимости - это большой вопрос. Вот и мне, после сборки и наладки такого "Ленинграда", пришлось столкнуться с тем, что не все игры, а конкретно "BRUCE LEE" работают от джойстика. Я стал советоваться с друзьями и те, кто раньше меня собрали и немного освоились с этим компьютером, посоветовали подать логический "0" на три старших разряда порта джойстика. Но после того, как я это проделал, некоторые игры, например, "EXOLON", зависали после загрузки и запуска Бейсик-загрузчика. Тогда я стал разбираться с Монитором, зашитым в ПЗУ и понял, что при существующей дешифрации портов в "Ленинграде" таких накладок мне не избежать. Исходя из этих соображений и еще прочитав разработку НТК "ПЛЮС" по интерфейсам "Спектрума", родилась схема, которую я Вам и предлагаю (приведена на следующей странице).

В своем "Спектруме" я исключил две микросхемы 555КП1, которые использовались для чтения клавиатуры, магнитофона и джойстика. Микросхему 555ТМ9 я оставил как выводной порт с адресом "FE", добавив две микросхемы 555АП5 для чтения клавиатуры и магнитофона и с учетом дальнейшего расширения возможностей моего компьютера добавил параллельный порт 580ВВ55, включив его по рекомендациям НТК "ПЛЮС", только с более жесткой дешифрацией портов. В этой схеме я использовал свободные инверторы "Ленинграда", а также элементы, которые выделяли сигналы IORD' и IOWR' (D14.1, D14.4).

Сигналы по схеме:

C2 - запись в порт "FE", C3 - чтение порта "FE", сигнал C1 - вспомогательный.

Если не устраивает жесткая дешифрация, можно исключить элементы D41.1 и D42.1 и тогда вместо сигнала C1 на входы 4 - D42.2 и 12 - D42.4 подать сигнал IORQ'. Адреса внешних портов MC 580ВВ55 будут следующие:

"1F" - чтение, запись порта "А"(порт джойстика);

"3F" - чтение, запись порта "В";

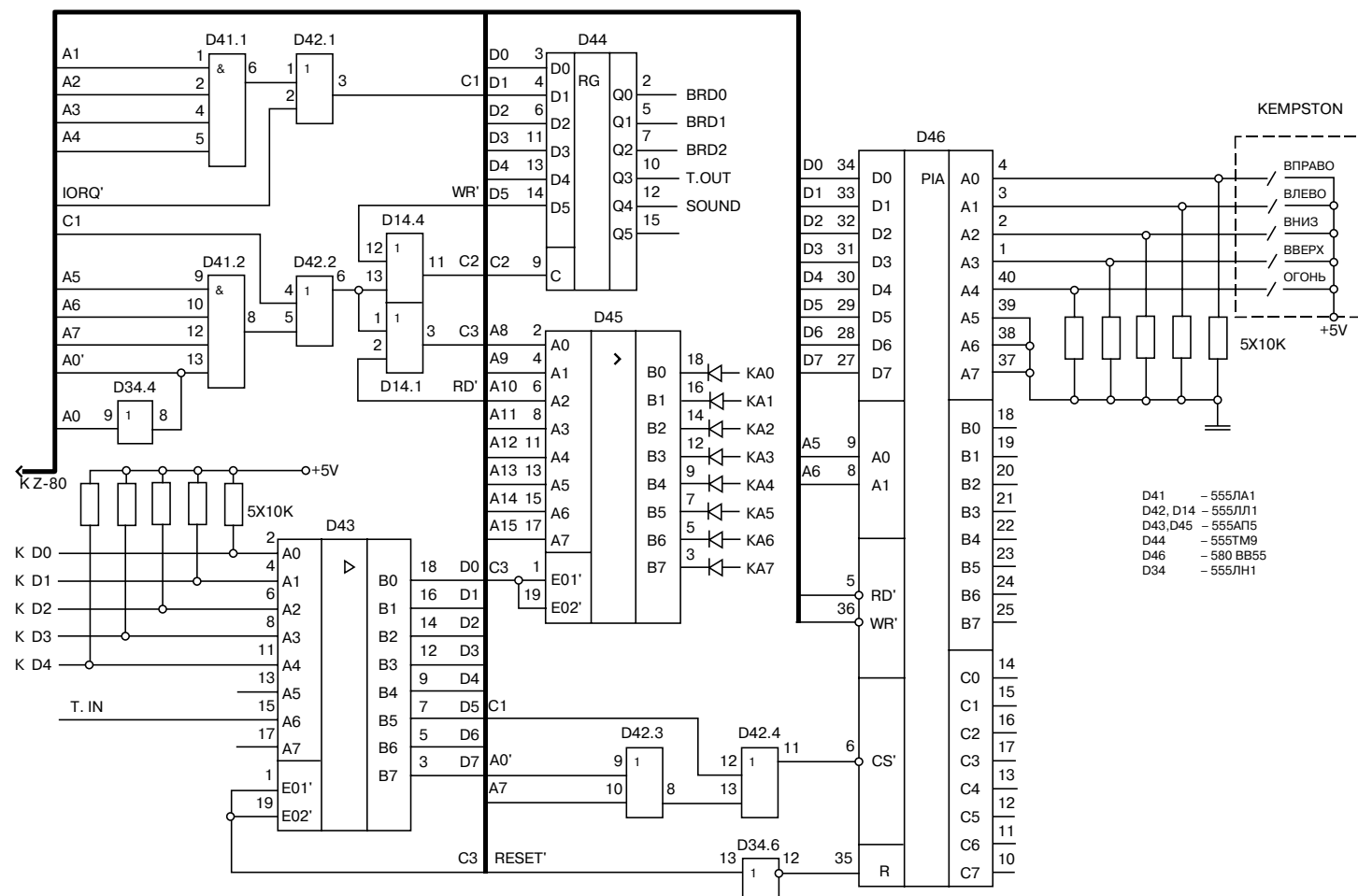
"5F" - чтение, запись порта "С";

"7F" - регистр управляющего слова.

Вот, пожалуй, и все по поводу этой схемы.

ИФК: Нас, как и большинство наших читателей, очень интересует все, что касается совершенствования компьютеров в плане повышения их совместимости с фирменными моделями. Таким работам мы даем "зеленую улицу" и принимаем их с особой благодарностью.

Приносим свои извинения нашим читателям за нестандартный вид схемы (мы уже говорили о сложностях исполнения графических работ в текстовом редакторе), а для работы в графической среде у нас не хватает кадров и времени. Так, например, инвертированные сигналы вместо общепринятой черты над обозначением сигнала у нас отмечены знаком (') - апостроф.



* * *

Нам пишет Стас Рубцов из г. Улан-Удэ.

Здравствуй, Инфорком!

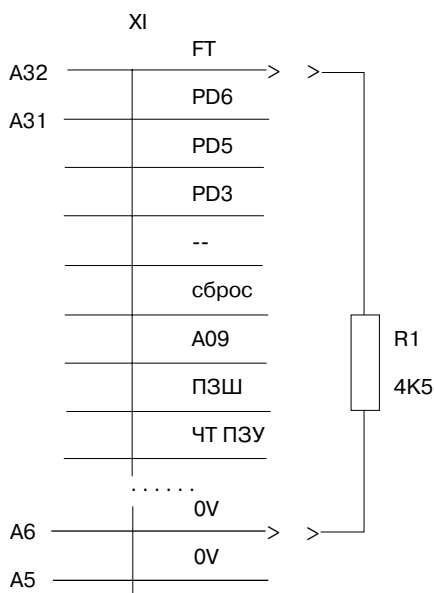
Большое спасибо за очень нужный и интересный журнал. Подписчиком я не являюсь, но регулярно пользуюсь Вашими разработками и работаю с журналом. Беру я его у знакомого, "товарища по Спектруму", если можно так сказать. Он является Вашим подписчиком с самого первого выпуска. Так что мне, можно сказать, повезло.

ИФК: Спасибо на добром слове, Стас. Мы очень рады, что у Вас есть хороший друг. На Вашем примере еще раз посоветуем всем, кто нас читает, объединяться и еще раз объединяться. Новые повышения цен на печать в конце этого года поставят Вас перед необходимостью существенных затрат на вздорожавшее "РЕВЮ" и только единение поможет выжить и Вам и нам.

КОРР: ... Работаю я на ZX совместимой машинке "Дельта-С". У нас в городе "Дельты" разошлись по рукам за две недели - по самым предварительным сведениям - три партии по 800 штук. Были ли еще партии, я не знаю. Во всех Ваших журналах нет ни одного слова о компьютере "Дельта-С" и это может говорить либо об ограниченном его распространении

либо о его полной совместимости с фирменной машиной. Ни то ни другое не соответствует действительности. "Дельта-С" распространена и в Томске, и в Иркутске и в других городах, в Москве я видел в фирменном "Московском" (на площади трех вокзалов). Работаю я с "Дельтой-С" год, и хочу сказать несколько слов об этой машинке. По моим сведениям, "Дельта-С" выпускается по меньшей мере на трех заводах. У меня вариант Курского завода (305040, Курск, ПО "Прибор"). Оформление машины не очень хорошее, в частности, "жесткая" контактная клавиатура, не очень качественно сделан вывод звука, неважный внешний вид. Однако технически машина прекрасная. Имеет встроенный ZX-интерфейс II, буферизированный выход на ZX-принтер. Несовместима "Дельта-С" с фирменным "Спектрумом" только в одном. При наличии в программе вариаций на тему "POKE 23570,16" - вывод заголовков типа "BYTES..." и др. на принтер, из-за ошибки в схеме не проходит номер с подпрограммой ПЗУ "COPY-LINE" по адресу 0F0C-0F13 (COPY-L-2), т.е. если принтер не подключен, программа "виснет".

Выход очень прост. Надо "подключить" принтер, дать ответ из принтера по линии "BUSY" - "занят", что принтер не готов к приему информации. Но так как не у всех есть фирменный ZX-принтер, советую всем столкнувшимся с этой проблемой просто подать на вход "BUSY" уровень "ЛОГ 0". Я поступаю так: соединяю на разъеме вывод A32 "ГТ" (готовность принтера), соответствующий сигналу "BUSY" с выводом A5(A6) т.е. с общим проводом, через резистор 4.5 кОм, при этом не надо разбирать компьютер. После такой переделки все программы идут, как на фирменной машине.



Краевой разъем платы.

Звук можно вывести на усилитель, для этого на свободном месте (на задней стенке компьютера) монтируем гнездо СГ-5, вскрываем компьютер, находим единственный проводник, ведущий к звукоизлучателю (от транзистора на плате) и в тоже самое место припаиваем минусовый вывод конденсатора К50-3 10 мкФ х 12 V, плюсовой вывод припаиваем к контактам "1" и "4" на СГ-5, а к контакту "2" на том же разъеме припаиваем "массу" компьютера (перемычку от контакта "2" разъема "+5В" либо от контакта "2" разъема RGB).

Хотелось бы, чтобы изготовители "Дельты-С" учли эти вопросы в своих разработках.

ИФК: От всей души присоединяемся к этому пожеланию и думаем, что независимо ни от чего Ваши рекомендации помогут многим любителям повысить совместимость своих машин.

КОРР: ... Я прошу извинить меня за то, что получилось так, как будто у "Дельты-С" есть существенные недостатки. Это прекрасная машина и это видно даже по печатной плате, жалко только, что к нему не прилагается принципиальная схема. Может быть его производители учтут и этот момент?

По вопросам совместимости различных версий "Спектрума" нам пишет из Йошкар-Олы Москвин Денис Егорович:

Адресую свое письмо в раздел "Форум" журнала "ZX-РЕВЮ", так как хочу поделиться своим открытием в области совместимости компьютера "Пентагон 48" с фирменным "ZX-Spectrum".

В последнее время 48-е машины вытесняются 128-ми, но мне еще раз хотелось бы остановиться на 48-м "Пентагоне".

В ваших журналах опубликовано множество статей, посвященных совместимости "Пентагона", но среди них не оказалось статьи о том, почему не работают на "Пентагоне" такие всем известные программы, как "Cobra Stalone", "Renegade-1", "Str. Hawk" и т.п. Мне бы хотелось, чтобы этот пробел был восполнен.

Эти программы и им подобные не "идут" потому, что в "Пентагоне" не коммутируется сигнал WAIT (ожидание). Этот сигнал, как оказалось, в фирменном компьютере используется для разрешения конфликтов памяти, вызванных регенерацией нижней области ОЗУ, где хранится содержимое экрана.

Скоммутировать сигнал WAIT можно с помощью триггера. В "Пентагоне" это микросхема TH2 (DD11), расположенная в нижней части платы компьютера рядом с микросхемой ЛП5, с которой выводятся цвета (R, G, B) и яркость (Y).

Если сигнал WAIT подключить согласно приведенной ниже схеме, то можно будет смело загружать "Renegade", т.к. она будет работать!

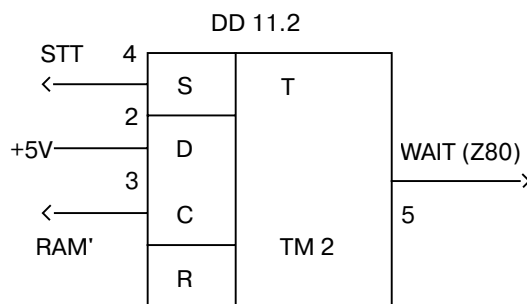


Рис. 2

Здесь на схеме:

STT - это вывод 12 м/с DD1.3 (ЛН1 - на плате она одна);

RAM' - это вывод 11 DD18.1 (ЛА3 - ее легко найти, т.к. именно к выводу 11 подключена перемычка, уходящая под процессор);

WAIT - это вывод 24 микропроцессора.

Таким образом, если подытожить все исправления "Пентагона", опубликованные в Ваших журналах, то "пентагоновская" версия 48-го станет полным аналогом фирменного SPECTRUMa.

Данная доработка опробована на 5-ти компьютерах. Опыт показал, что при подключении сигнала WAIT к процессорам КР 1858 ВМ1 (советский аналог Z80) программы Cobra и Renegade все равно не работают. Видимо, это происходит из-за неправильной системы прерываний у данного процессора.

Нам пишет Гасин Руслан Владимирович из Магадана.

Здравствуйте, уважаемая редакция!

Я являюсь недавним Вашим читателем (подписчиком). Основной целью подписаться на Ваш журнал было желание научиться работать с машинными кодами, описание программ (в основном системных), взлом (а вернее техника защиты) программ.

Ну что ж. В принципе я удовлетворен. Мне понравилась информация, представленная Михаиленко В.С. Все это хорошо. Но я думаю, что многие согласятся со мной.

Я предлагаю хоть чуть-чуть рассмотреть технику защиты программ на диске (в системе TR-DOS). Конечно, можно заявить, что с дисководом работают не так уж и много пользователей ZX-SPECTRUM. Я сам только подумываю о нем (дисковом). Но я и не требую доскональных данных о защите (на дискетах). Отталкиваясь от Ваших же принципов (дать общую методику, идею, направление, а люди сами додумают), можно дать лишь общие сведения. Например, такую. На диске есть директорий, он "говорит" о длине, расположении файлов и т.д. Так почему бы не сделать так, чтобы на диске был файл, но операционная система не могла ни вывести его название, длину в директорий, ни запустить на выполнение без разрешения пользователя. Конечно же, все это выполнимо, и что-то можно сделать? Но не хватает сведений о TR-DOS (ее переменных и т. д.)

Естественно, это не совсем защита от взлома, а скорее от несанкционированного доступа к программе.

Очень прошу, напечатайте мое письмо. Вдруг, кто-то откликнется и расскажет, что знает. А от Вас, если это произойдет, зависит только вопрос - напечатать или нет. Но я думаю, что одну страницу не грех будет отдать.

Надеюсь, что мое письмо не оставит Вас равнодушными.

ИФК: Уважаемый Руслан Владимирович, здесь двух мнений быть не может. Голосуем "За" единогласно и безусловно дадим все, что сочтут нужным сообщить наши читатели. Но, кажется на горизонте появилось кое-что поинтереснее. Мы имеем в виду операционную систему IS-DOS. Мнение знакомых специалистов, опробовавших ее, единодушное - ЭТО БЛЕСК, который действительно превращает "Спектрум" в полупрофессиональную ЭВМ. Впрочем, пока СТОП... мы находимся в начале пути, информация пока скудна, но не исключено, что следующий год пройдет под знаком именно этой системы.

* * *

Нам пишет Алексей Жильцов из Самары:

Я хочу сообщить Вам об особенностях игры "Death Wish-3". Если встать в доме под окно и нажать "W", то на экране появится план улицы и прицел, двигая которым и стреляя, можно расстреливать пробегающих бандитов. Только нужно следить, чтобы сзади никто не подошел, иначе будут проблемы.

* * *

Многие наши читатели жалуются, что мы мало внимания уделяем аппаратной части. Это так, но если Вы вспомните, мы уже писали, что у истоков "ИНФОРКОМа" и "ZX-РЕВЮ" стоял коллектив по своей квалификации скорее преподавательский, чем технический. Поэтому аппаратные вопросы мы всегда отдавали другим. Сначала нам помогал в этом НТК "ПЛЮС", а сейчас, когда эта фирма увы больше не работает, мы можем только опираться на Ваши письма, уважаемые читатели. Писем с программами или программными вопросами намного больше, чем с аппаратными. И тем ценнее для нас каждое письмо, в котором так или иначе затрагиваются аппаратные проблемы. И коль уж сегодня мы дали некоторые рекомендации по возможным доработкам "Дельты-С", то очень кстати будет и сообщение нашего читателя из г.Норильска - Галушака Александра. Вот что он пишет:

"В Норильске много владельцев ПК "Дельта-С", но многим из них не нравится, что в "Дельте" нет порта KEMPSTON-джойстика. Я подозреваю, что в СНГ существует немало типов компьютеров без KEMFSTON-джойстика.

Я подключил джойстик к "Дельта-С" своему товарищу и хочу поделиться опытом с читателями "ZX-РЕВЮ".

Собран интерфейс KEMPSTON-джойстика на двух микросхемах, установленных на небольшой печатной плате, и соединяется с Z80 тринадцатью проводами, включая питание.

Я обратил внимание, что на "Дельте-С" вместо двух микросхем 2764 используется

одна микросхема 27128, а место для второй микросхемы - свободно. Поскольку на ПЗУ выводятся почти все необходимые для интерфейса джойстика сигналы (кроме IORQ и RD), то я разработал печатную плату в расчете на то, что она будет установлена на место неиспользуемой микросхемы 2764 (ПЗУ-1) и соединяться с основной платой компьютера короткими гибкими проводами. Расположенный таким образом интерфейс размещается внутри корпуса компьютера ничему не мешая. С разъемом KEMPSTON-джойстика дополнительная плата соединяется шлейфом (6 жил) необходимой длины.

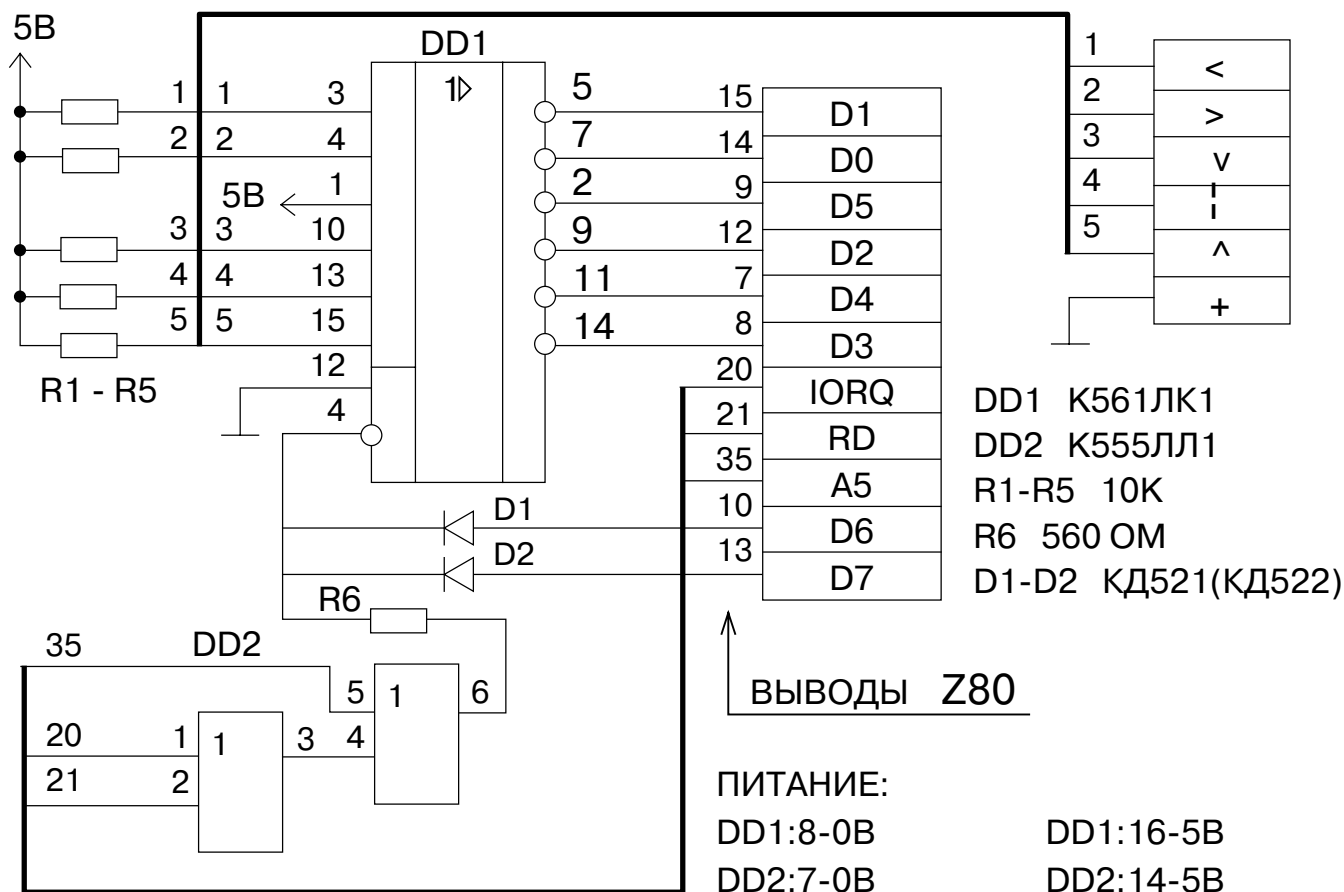


Рис. 1

Возможны и другие решения. В частности, если в ПК используются 2 микросхемы 2764, то можно использовать эту же печатную плату, расположив ее над процессором Z80, и соединив непосредственно с выводами Z80.

ИФК: Александр готов ответить на любые вопросы, возникшие при изготовлении интерфейса и дает свой адрес:

Просьба вкладывать конверт с обратным адресом.

Еще, если есть такие вопросы, могу выслать рисунок печатной платы для ZX-LPRINT-3 (описание которого было в "РЕВЮ-91" N 7-8). Этот интерфейс принтера мною собран и благополучно работает уже год".

В работе с письмами наших читателей нередко один вопрос вызывает другой. Вот и сейчас так. Один наш читатель успешно решил проблему подключения джойстика, другой пытается найти подходы к подключению "мыши". Матвеев Ю.Н. из Кемеровской обл. просит откликнуться тех, кто может что-то подсказать по этой проблеме. Он также был бы рад познакомиться с целью переписки с "синклеристами", любящими заниматься доработками своих машин.

652692, Кемеровская обл. , Гурьевский р-н, с. Горскино, Матвееву Ю.Н.

В заключение передаем уважаемому корреспонденту пламенный привет от редактора раздела "К.Н." Матвеева Ю.А.

Как добро и зло, как плюс и минус не могут существовать друг без друга, так не могут существовать отдельно друг от друга вопросы защиты и взлома программ. Ситуация складывается примерно так. Появляются новые разновидности защиты, существующие какое-то время, затем они раскалываются, появляется литература на тему о защите и ее снятии. Новые варианты защиты разрабатываются уже на основе систематизированного материала по снятию защиты. Появляются новые, более изощренные способы взлома и процесс продолжается дальше, замыкая очередной виток эволюционной спирали. Кто сделает следующий ход в этой бесконечной игре?

Мы уже печатали материалы по защите программ, хорошо осознавая тот факт, что этим материалом пользуются как те, кто защищает свои программы, так и те, кто их после этого взламывает. Сегодня мы предлагаем материал, присланный Григорием Борисовичем Лупповым из города Кирова. На наш взгляд им как раз могут результативно воспользоваться разработчики защиты, в то же время для "взломщика" - здесь всего лишь общие фразы.

Вот что пишет наш корреспондент: "Я в общем-то серьезно "Спектрумом" не занимаюсь - больше спец по "Вектору 06Ц". Однако имею "Пентагон 48К" и с удовольствием читаю "ZX-РЕВЮ", когда удастся выпросить его у знакомых "спектрумовщиков". В номере 1-2 за 1992 г. увидел статью по защите, а защита программ - дело одинаково актуальное для всех бытовых персональных ЭВМ в конфигурации с магнитофоном. Хочу предложить Вашему вниманию свой взгляд на этот вопрос.

В ходе долгих размышлений и поисков вариантов защиты программ для "Вектора", я пришел к выводу, что действительно, для магнитофонных версий ПЭВМ надежной защиты нет! Все способы защиты, которые существуют на "Спектруме", это своего рода искусные, красивые трюки авторов, крайне недолговечные. Эти трюки в целом проблемы не решают. Внешне это выглядит так: автор, петляя и изворачиваясь, как заяц-русак, старается сбить со следа волка-взломщика. Результат один - заяц погибает.

Нужно применять другой подход. Защита должна быть простая, относительно несложно ставиться, но самое главное - она должна быть трудоемкой для снятия и легко модифицируемой (на случай, если одну модификацию сняли).

Предлагаю классическую схему: основная программа в своем формате (отличном от стандартного), впереди идет закодированный загрузчик с декодировщиком в формате "Спектрума" и не обязательно с автозапуском.

Слабость схемы очевидна - декодер всегда можно просмотреть и разобраться в принципе его работы. Поэтому процесс декодирования должен включать в себя несколько основных положений.

1. Декодирование с использованием всего адресного пространства компьютера.
2. Декодирование с подвязкой по времени.
3. Перевод на декодированный загрузчик должен быть неочевидным (чтобы нельзя было вставить переход на взламывающую подпрограмму после декодировки), он должен быть включен в закодированный текст и декодироваться последним, то есть декодирование идет от старших адресов к младшим.

4. При декодировании используются псевдослучайные числа. В начале работы декодер забивает все свободное пространство ОЗУ (и видео-ОЗУ в том числе) псевдослучайными числами. Генератор таких чисел должен быть:

а) последовательным - это значит, что n -ое число можно получить только после получения всех предыдущих чисел;

б) параметрическим - это значит, что получаемая последовательность будет зависеть от параметров генератора, то есть генератор легко перенастраивается.

Числа в память должны заноситься не подряд, а либо через какое-то смещение (не равное 1) либо по случайным адресам (это лучше), например, первые два байта из генератора определяют адрес, третий байт - данные. Естественно декодер и загрузчик меняться не должны.

Затем декодер начинает непосредственно свою работу. Декодирование идет сверху - вниз. Для получения очередного байта над ним нужно проделать ряд операций с использованием данных из ОЗУ, адресуемых несколькими маркерами. Маркер - циклически бегающий по адресу декодера; маркер - циклически бегающий по всему ОЗУ и ПЗУ (кроме декодера и загрузчика); случайный маркер - обеспечивающий определенную защиту от резидентных взломщиков. Время-зависимые данные (регистр R или ячейка-таймер) обеспечивают защиту от эмулирования декодера пиратским взломщиком.

После декодирования управление передается на загрузчик. Так как содержимое памяти загрузчику известно, то он может загружать (декодировать) программу, используя содержимое памяти (однако снижается при этом технологичность и модифицируемость защиты). Загрузчик должен иметь гибкую структуру, чтобы загружаемая программа легко перехватывала управление в любое время. Лучше всего использовать блочный формат: вся программа разбивается на блоки по 32, 64 или 256 байтов; в начале каждого блока идет заголовок с адресом загрузки, в конце - байт-паритет KC.

Программа может иметь и внутреннюю защиту, включающую проверку наличия в памяти меток о работе декодера и загрузчика и т. д. Это можно сделать на авторских программах, но не на чужих.

Для большей модифицируемости можно применять плавающую константу чтения, которая либо изменяется по определенному алгоритму, либо содержится в таблице.

Технологичность защиты обеспечивается тем, что автору нужно только раз получить закодированный особым образом загрузчик и использовать его затем на разных программах. Как только защиту вскроют (ничто не вечно под Луной), можно изменить параметры генератора случайных чисел и таблицу констант чтения. "Пирату" придется повторять весь трудоемкий процесс вскрытия. Трудоемкость определяется тем, что нужно эмулировать ОЗУ так, чтобы там был и набор псевдослучайных чисел и пират-эмулятор декодера. Это можно сделать с использованием дисководов или дополнительного ОЗУ (в 128K версии), однако при этом будут изменяться время-зависимые данные и процесс декодировки нарушится. Если программа рассчитана на версию компьютера с музыкальным процессором, то последний можно использовать в качестве генератора время-зависимых данных.

Не могу ничего посоветовать относительно борьбы с "MAGIC", т.к. не знаю тонкостей работы в этом режиме. Однако многие программы после обработки кнопкой "MAGIC" не работают, значит придумать что-то можно".

Материал этот чисто теоретический, у многих может вызвать вопрос типа: "Это все складно звучит, но как же все это практически реализовать? Где конкретные примеры, демонстрирующие изложенные принципы?" Примеров действительно нет, так как любой практический пример приведет к снижению во много раз долговечности защиты, сведя на нет все усилия автора программы. Хорошая теория - это уже больше половины дела. А как это конкретно воплотить в жизнь - как говорится, думайте сами, решайте сами...

ИФК: Но Григорий Борисович, видимо предвидя подобные вопросы, все же продолжает письмо конкретным примером, предлагая защитный трюк, связанный с блоками без заголовка.

KOPP... Копии определяют конец такого блока, если при выходе из подпрограммы

LOAD_BYTES установлен флаг Z. Подпрограмма загрузки бита имеет вид:

```
05E3 CALL #05E7
05E6 RET NC
05E7 LD A,#16
05E9 DEC A
05EA JR NZ, #05E9
05EC AND A
05ED INC B
05EE RET Z
....
05F8 JR Z, #05ED
```

Если долго не появляется смена уровней (фронт сигнала), то подпрограмма досрочно возвращает управление. Такая ситуация возникает как раз при загрузке файлов без заголовка, если задана фиктивная (большая) длина. Трюк заключается в том, чтобы выгружать такой блок как можно более длинным. То есть сначала мы рассчитываем байт-паритет (КС), идущий в конце каждого блока, и записываем его в конец файла в виде данных, а выгружаем файл с максимальной длиной. При загрузке все будет нормально, так как мы знаем реальную длину блока и загрузим его без проблем. У тех же утилит, которые не знают реальную длину (программы-копировщики), возникнут проблемы со свободной памятью. Конечно, можно обрезать такой файл, однако нет гарантии, что обрезали только ненужные данные, или что загрузчик не проверяет целостность файла.

ИФК: Но и на этом письмо Григория Борисовича не заканчивается. Он предлагает читателям еще несколько слов насчет формата записи байта на магнитофон.

Он пишет: "В "Векторе" используется следующий формат: 1 заменяется на 01, а 0 - на 10 (так называемый каскадный код). При этом длина битов одинаковая, кроме того, не надо так точно выдерживать междубитовую и междубайтовую длину. Я уже пробовал эмулировать такой формат на "Спектруме". Результаты - положительные. Они хороши еще и тем, что на "Спектруме" можно при этом формате достичь в 2-3 раза большей скорости записи, по сравнению со стандартной, без каких-либо особых проблем (нужен только загрузчик). Это готовый турбо-формат, причем с изменяемой константой записи, то есть Вы можете регулировать ее в широких пределах. Константа чтения автоматически определяется при загрузке. Для защиты от помех используется дублирование. Каждый блок (по 256 байт) задублирован. И помехоустойчивость можно улучшить, если дублировать не отдельный блок, как сейчас (по принципу: блок 1, дубль 1, блок 2, дубль 2, ...), а всю программу (блок 1, блок 2, блок n, дубль 1, дубль 2, ... дубль n). Кроме того, возможно сделать (а на ВЕКТОРЕ уже сделано), что сбой в каком-то блоке не вызывает сброс всей программы (на "Спектруме" одна помеха вызывает сбой всего файла), так как программа разбита на мелкие блоки (как отсеки в подводной лодке). Вы можете остановить магнитофон, отмотать чуть назад и повторно загрузить сбойное место. Процесс загрузки отображается на экране в виде карты загрузки. Очень наглядно видно, где произошел сбой. Кроме того, есть возможность, пожертвовав сбойным участком в 32 байта, загрузить оставшуюся информацию.

На "Векторе" загрузчик занимает 256*2 байт, на "Спектруме" его можно сделать короче".

ИФК: Если у читателей есть интерес к тому, что говорилось об этом формате, то Григорий Борисович предлагает выслать текст загрузчика и текст подпрограммы записи в таком формате на магнитофон.

Уважаемый Григорий Борисович! Мы искренне благодарим Вас за предоставленный материал. Думаем, что это мнение разделят наши читатели.

ПРОФЕССИОНАЛЬНЫЙ ПОДХОД

(С) Збитнев В. А., 1992, 1993. (г. Новосибирск)

Применение статического генератора случайных чисел на примере программы "ELITE"

Вероятно, уже все знают о том, что такое RND. Во многих игровых и учебных программах этот оператор используется для того, чтобы поведение компьютера в некоторых ситуациях было непредсказуемым. Эту роль прекрасно исполняет генератор случайных чисел на ZX-SPECTRUMе. Он относится к типу динамических, то есть таких, которые изменяют свое состояние независимо от того, когда вызывается этот генератор. Свое значение "спектрумовский" RND вычисляет, используя переменную SEED (23670 DEC), которая изменяется с течением времени.

Но так бывает не во всех компьютерах. Многие, наверное, знают работу RND на других версиях Бейсика для компьютеров типа "Ямаха" (MSX), IBM. Сколько бы раз Вы не вызывали бы RND в своей программе (например для рисования звездного неба), он выдает одни и те же числа после команды RUN, RANDOMIZE (в случае рисования звездного неба одна и та же картина будет на экране, если Вы остановите программу и снова запустите ее через RUN).

Такой вид RND-генератора называется статическим. Он плох для непредсказуемых действий, но он прекрасно может работать в качестве хранилища огромного объема данных, не требующих в памяти места. Но самое главное - эта информация будет постоянной, сколько бы раз эти данные не были бы затребованы. Простейший пример такого RND - генератора на ассемблере

```
RND: LD     A, (R1)
      LD     D, A
      LD     A, (R2)
      LD     (R1), A
      ADD    A, D
      LD     D, A
      LD     A, (R3)
      LD     (R2), A
      ADD    A, D
      RLCA
      LD     (R3), A
      RET
```

R1, R2, R3 - переменные

В регистре A будет содержаться случайное число (0-255). Но подобный генератор случайных чисел все время будет генерировать очередные случайные числа. Теперь создадим программу, подобную RANDOMIZE в Бейсике.

```
RAND: LD     A, 5
      LD     (R1), A
      LD     A, 90
      LD     (R2), A
      LD     A, 37
      LD     (R3), A
      RET
```

Теперь, после вызова подпрограммы RAND некий указатель встаёт на начало некой таблицы, в которой хранится бесконечное количество байтов, которые и извлекаются из неё

подпрограммой RND. Где это можно использовать? К примеру, в хорошо известной игре "ЖИЗНЬ" ("LIFE") нужна матрица со случайно расположенными в ней клетками. Вводить через DATA весь массив неудобно, медленно и не практично. Если заполнять массив через обыкновенный RND, то возникает проблема с тем, как снова воспроизвести картину, если это понадобится. Опять прибегаем к DATA. Но есть более практичный способ: задание массива через статический RND, который будет выдавать те же самые значения, что и в первый раз, стоит только вызвать RAND.

В игровых программах с большим количеством комнат таким путем можно запоминать все комнаты, как комбинацию трех (или другого числа байтов), в данном случае - 5, 90, 37. Стоит их только изменить и Вы имеете опять новую комнату. Программа "Demons" использует как раз такой способ, она чем-то напоминает "ЖИЗНЬ", но выглядит в 7 раз красивее и эффектнее. Попробуйте, не пожалеете!

Программа 1 Demons

	ORG	30000	
	CALL	RAND	; Сброс генератора случайных чисел в исходное положение
	CALL	RNDSC	; Заполнение экрана атрибутами через RND
NXTPG	LD	IX, 22528	; Старый массив
	LD	HL, 31000	; Новый массив
	LD	BC, 768	; 768 байтов - размер области экранных атрибутов
LOOKS	LD	A, (HL)	
	INC	A	
	AND	7	
	CP	(IX+1)	; Если слева, справа, сверху или снизу может кто-то
	JP	Z, SEAT	; "съесть" этот атрибут, то пусть "съедает".
	CP	(IX-1)	
	JP	Z, SEAT	
	CP	(IX+32)	
	JP	Z, SEAT	
	CP	(IX-32)	
	JP	Z, SEAT	
EATED	INC	HL	; Продолжить обработку массива
	INC	IX	
	DEC	BC	
	LD	A, B	
	OR	C	
	JP	NZ, LOOKS	
	CALL	COPY	
	RET		
SEAT	LD	(HL), A	; "Съедаем" текущий атрибут
	JR	EATED	
COPY	HALT		
	LD	HL, 31000	; Копирование нового массива в старый
	LD	DE, 22528	
	LD	BC, 768	
	LDIR		
	RET		
RNDSC	LD	HL, 31000	; Заполняем новый массив байтами с числовым значением
	LD	BC, 768	; от 0 до 7
FILL	CALL	RND1	
	AND	7	
	LD	(HL), A	
	INC	HL	
	DEC	BC	
	LD	A, B	
	OR	C	
	JP	NZ, FILL	
	CALL	COPY	; Копирование массива в старый массив
	RET		

```

RAND      LD      A, 5           ; 5, 90, 37 - байты, означающие параметры экрана
          LD      (R1), A
          LD      A, 90
          LD      (R2), A
          LD      A, 37
          LD      (R3), A
          LD      HL, 16384
          LD      DE, 16385
          LD      BC, 6144
          LD      (HL), 255
          LDIR
          RET

RND1      LD      A, (R1)       ; Генератор случайных чисел
          LD      D, A
          LD      A, (R2)
          LD      (R1), A
          ADD     A, D
          LD      D, A
          LD      A, (R3)
          LD      (R2), A
          ADD     A, D
          RLCA
          LD      (R3), A
          RET

R1         DEFB  0             ; Исходные значения параметров
R2         DEFB  0
R3         DEFB  0

```

Следующая область применения - формирование статичной (неменяющейся) информации. Возьмем к примеру игру "ELITE". Откуда компьютер с таким небольшим объемом памяти берет названия и координаты всех 256 звезд, в каждой из восьми галактик, да ещё и неизвестно сколько всего этих галактик. Где хранятся данные о ценах на каждой из планет, её техническом уровне, общественном строе и населении.

Разумеется, это работа статического RND. Если Вы наберете программу 2 на Ассемблере или 2.1 на БЕЙСИKe, то сможете получить данные о всех названиях планет и их координатах. Остальная информация каким-то образом хранится в 6 основных байтах и может быть образована одним из 4 RND, вызываемых последовательно для каждой звезды. Может быть, удастся расшифровать и остальные данные.

Итак, статический RND может хранить в себе информацию. Можно, к примеру, сделать слоги русскими, задействовать остальные байты для русских имен, для пола, возраста, адреса и пожалуйста: у Вас есть картотека на 10000 человек, работающих на космическом корабле. Статический RND может быть применим и для динамической информации. Примером может служить игра SENTINEL, в которой все ландшафты имеют свой номер, а задание их в памяти невозможно - слишком расточительно. Но в процессе игры Вы можете некоторым образом изменять эту информацию уже в матрице, созданной этим RND.

Программа 2.1

Заполняет массив данными о планете, начиная с адреса 40000 в следующем формате:

```

| Имя звезды | X | Y |
| 8 байтов  |16.|16.| = 10 байтов на планету

```

```

          ORG     30000
          CALL    RAND
          LD      B, 0           ; 256 звезд
STARS     PUSH    BC
          CALL    LDNAME        ; Считать имя в NAME

```

	CALL SAVNM	; Записать имя из NAME в массив, на который
	POP BC	; указывает MASSV
	DJNZ STARS	
	RET	
LDNAME	LD HL, NAME	; Позиция в имени указывается переменной POS
	LD (POS), HL	
	LD B, 9	; Очистили 9 байтов
CLEAR	LD (HL), 0	; Имя будет равно 8 символам
	INC HL	
	DJNZ CLEAR	
	LD A, (D0+1)	
	LD (D1), A	; Запишем координату Y
	LD A, (D2+1)	
	LD (D3), A	; Координата X звезды
	LD A, (D0)	; Условие, при котором 6-ой бит указывает на то,
	AND 64	; в слове нет последнего слога
	PUSH AF	
	CALL LDSLG	; Три слога
	CALL LDSLG	
	CALL LDSLG	
	POP AF	
	JR Z, RND2	; Если нет последнего слога, то уход на холостой RND
LDSLG	LD A, (D4+1)	; Условие отсутствия слога:
	AND 31	; номер слога = 0, т.е. слог "AL" - невозможен
	JR Z, RND2	
	LD HL, SLOGS	
	ADD A, A	; Номер слога*2
	LD E, A	
	LD D, 0	
	LD HL, DE	; Находим позицию слога в SLOGS
	LD A, (HL)	; Считываем первый символ слога
	INC HL	
	PUSH HL	
	LD HL, (POS)	
	LD (HL), A	; Записываем первый символ по адресу POS
	INC HL	; Следующий символ слога
	EX DE, HL	
	POP HL	
	LD A, (HL)	; Если символ равен "?", то не записывать
	CP 63	; урезанный слог
	JP Z, SAVPS	
	LD (DE), A	; Записываем 2-й символ слога
	INC DE	
SAVPS	LD (POS), DE	; Сохраняем позицию
RND2	LD HL, (D0)	; Сама процедура RND
	LD DE, (D2)	
	ADD HL, DE	
	EX DE, HL	
	LD HL, (D2)	
	LD (D0), HL	
	LD HL, (D4)	
	LD (D2), HL	
	ADD HL, DE	
	LD (D4), HL	; D0<---D2<---D4<---D0+D2+D4
	RET	
SAVENM	LD HL, NAME	; Записываем имя в массив
	LD DE, (MASSV)	
	LD BC, 8	
	LDIR	
	LD A, (D1)	; Считываем Y
	SRL A	; Преобразуем в нормальный вид
	XOR 127	

```

LD      (DE), A      ; Записываем в массив
INC     DE
LD      A, (D3)      ; Считываем X и записываем в массив
LD      (DE), A
INC     DE
LD      (MASSV), DE  ; Запоминаем позицию в массиве
RET

RAND    LD      HL, RNDDAT ; Копируем данные о галактике в массив RND
LD      DE, D0
LD      BC, 6
LDIR    LD      HL, 40000
LD      (MASSV), HL
RET

D0      DEFW 00      ;
D1      DEFW 00      ;
D2      DEFW 00      ; Переменные
D3      DEFW 00      ;
D4      DEFW 00      ;
MASSV   DEFW 00      ;
NAME    DEFW "      "
SLOGS   DEFM "ALLEXEGEZA"
        DEFM "CEBISOUSES"
        DEFM "ARMAINDIRE"
        DEFM "A?ERATENBE"
        DEFM "RALAVETIED"
        DEFM "ORQUANTEIS"
        DEFM "RION"

RNDDAT  DEFB 74      ; 61...66 байты в отгружаемом блоке состояния ELITE
        DEFB 90
        DEFB 72
        DEFB 2
        DEFB 83
        DEFB 183

```

Программа 2.1

Печать имён планет всех галактик ELITE.

```

10  PAPER 0: INK 6: BORDER 0
20  CLS: PRINT "ELITE SPACE""1-STAR NAMES""2-PLOTS OF STARS"
30  LET k$=INKEY$: IF k$="1" THEN GO TO 60
40  IF k$="2" THEN GO TO 200
50  GO TO 30

```

Подпрограмма вывода названий.

```

60  LET n=0: GO SUB 1070: GO SUB 1010: REM установка генератора случайных чисел в исходное
    положение и установка стринга n$, содержащего слоги всех названий планет.
70  CLS: PRINT INK 5; "NO NAME"
80  FOR f=1 TO 20
90  LET r=d0-INT (d0/256)*256
100 GO SUB 2000: REM В строках 100...120 и 150 содержатся четыре вызова генератора для одной
    звезды.
110 GO SUB 2000
120 GO SUB 2000
130 IF r>127 THEN LET r=r-128
140 IF r-64<0 THEN GO SUB 1030: REM Если 6-ой бит равен нулю, то производится холостой
    вызов RND и слово становится короче на один слог.
150 GO SUB 2000
170 LET n=n+1: IF n=256 THEN STOP: REM Если число звёзд равно 256, то конец работы.
180 NEXT f: BEEP .1, 30: PAUSE 0: GO TO 70: REM Формируется следующее имя

```

Вывод позиций звёзд

```
200 CLS: PLOT 0,23: DRAW 255,0: DRAW 0,129
205 DRAW -255,0: DRAW 0,129
210 GO SUB 1010
220 FOR f=0 TO 256
230 LET y=INT (d0/256): LET y=INT (y/2)+24
240 LET x=INT (d2/256)
250 PLOT OVER 1;x,175-y: REM Вывод звезды
260 GO SUB 1030: GO SUB 1030
270 GO SUB 1030: GO SUB 1030
280 NEXT f: STOP

1000 REM *Подпрограмма инициализации статического генератора*
1010 RESTORE: READ d0, d2, d4: RETURN

1020 REM *Статический генератор случайных чисел*
1030 LET s=d0+d2: IF s>65535 THEN LET s=s-65536
1040 LET s=s+d4: IF s>65535 THEN LET s=s-65536: REM s=d0+d2+d4 (двухбайтное логическое
    сложение)
1050 LET d0=d2: LET d2=d4
1060 LET d4=s: RETURN

1065 REM *Генерация списка возможных слогов*
1070 LET n$="ALLEXEGEZACEBISOUSESARMAINDIREA?"
1080 LET n$=n$+"ERATENBERALAVETIEDORQUANTEISRION"
1090 RETURN

2000 REM *Выбор случайного слога*
2010 LET h=INT (d4/256): LET h=h-INT (h/32)*32
2020 IF h=0 THEN GO SUB 1030: RETURN
2030 LET h=h*2: PRINT n$(h+1)
2040 LET h=h+1
2050 IF n$(h+1)="?" THEN GO SUB 1030: RETURN
2060 PRINT n$(h+1): GO SUB 1030: RETURN

3000 DATA 74+90*256
3010 DATA 72+2*256
3020 DATA 83+183*256
3030 REM этот блок данных соответствует галактике с параметрами 74, 90, 72, 2, 83, 183
    (байты 61...66 в блоке состояния программы ELITE)
```

ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ

В прошлом выпуске "Форума" наши читатели обращались за помощью по поводу программ "Swords & Sorcery" и "Dun Darach". Пока никто не откликнулся на эти письма, мы на скорую руку пробежались по своим архивам, выудили кое-что из импортных журналов и составили нечто вроде эссе из разных источников. Это, конечно, капля в море, на серьезный поиск пока не было времени, в дальнейшем может быть дадим больше информации.

Мы настолько рады, что начинает разгораться интерес к жанру ADVENTURE, что спешим раздуть первую искру, надеясь, что скоро вспыхнет и костер.

Swords & Sorcery.



Вот несколько подсказок по прохождению первого квадранта (Quadrant 1).

Когда Вы проходите тренировку, то проведите один день с мастером Yama, шесть дней с мастером Bog и 7 дней с Мерлоком (Merlock). Это даст Вам 22 пункта магии.

Обязательно съешьте все пирожки (pies), сколько сможете себе позволить (до 5 штук) и во время своего путешествия прежде чем исследовать что-либо (Explore) включайте режим Attack на случай неожиданного нападения.

В первом квадранте самым первым делом пойдите в комнату 6 и возьмите там меч (sword). Комнаты мы нумеруем слева направо таким образом, что левая верхняя комната имеет номер 1. Меч позволит Вам наносить удары (режим ATTACK HACK). Все имущество, какое соберете, относите в эту комнату. Капкан (Room Trap) здесь лучше не разряжать (disarm), т.к. после этого на Вас нападают 5...6 монстров.

Мы приносим извинения за возможную неадекватность перевода, т.к. эту программу, мы сами не пускали, а в английском языке такие слова, как Room Trap могут иметь много разных значений и при переводе адекватное выбирается только по смыслу, другого пути нет.

После комнаты 6 идите в комнату 18 и там разрядите (disarm) капкан (trap). Тогда Вы сможете достать staff (волшебную палочку) из мешка (sack). Это даст Вам дополнительно 16 пунктов магии и, кроме того, два новых заклятья, но работают они только в том случае, если Ваши магические способности (magic skill) выше, чем боевые (fighting skill).

Владеть этой волшебной палочкой хорошо, но дорого, т.к. при этом быстро расходуется много магической энергии. Есть оригинальный выход - ее надо проглотить (eat). Тогда и энергия не расходуется, т.к. ее у Вас с собой больше нет, и магические способности сохраняются, ведь она всегда при Вас.

Соберите все мечи, которые Вам встретятся, и отнесите их в комнату 6. В комнате 15 есть Кольцо Прыгучести (Ring of Jumping), а ключ N1 (Key 1) Вы найдете в комнате 14. Вам следует также взять Sabaton в комнате 22 (что это такое мы не знаем, может быть какое-то оружие. В английском языке с давних пор принято давать имена собственные оружию. Вспомните меч Эскалибур короля Артура). Если у Вас к этому времени еще нет магического меча (Magic sword), убедитесь, что Вы выбрали ATTACK FOR A DEFENCE, иначе с монстром, который там обитает, Вам не справиться, его не берет заклятье FIREBOLT, не берет его и обычное оружие. Выход из этого квадранта находится в комнате 4.

Теперь "Dun Darach".

В этой игре полно потайных дверей. Их можно обнаружить тщательной проверкой стен. Обычно на наличие такой двери указывает факел на стене.

Первым делом Вам надо приобрести книгу (book) в магазине (shop) на улице King Street. Ее надо отдать леди Mgor (Ledy Mhor), а она Вам даст взамен другую. С ее помощью откроется потайная дверь на улице Myre Street. Внимательно поищите здесь. Если Вам удастся найти щит (shield), то его можно посвятить (offer) богине Дану (Goddess Danu).

На Древнем Холме (Old Hill) Вы найдете лиру (Lyre). С ней идите к Дейну (Dan), которой обитает за потайной дверью на улице Park Row. Он даст Вам взамен за эту лиру щит, посвященный богу Мидиру (Midir). Отнесите его богу и он даст Вам ключ.

В этой игре можно неплохо зарабатывать. Например так. Возьмите золото у Ассэйера (Assayer) на улице West Way и продайте его брокерам на улице Cross Street. Существует еще очень много разных способов, но этот наипростейший. Но опасайтесь воров, они часто похищают то, на чем находится Ваша "звезда". Поэтому никогда не помещайте ее на предмет, которым дорожите, например держите ее на деньгах (Iridi), которые сравнительно легко восстанавливаются.

В игре множество головоломок. В частности, многие встают в тупик перед вопросом, как взять жемчужину (Pearl) из комнаты Strong Room, что на северной стене и куда потратить свои деньги.

Обе проблемы можно решить одновременно. Набрав 10 000 Iridi, отнесите деньги в гильдию воров на улице Silver Street. Они продадут Вам лицензию (Licence), с помощью которой Вы и похитите эту жемчужину.

А вот пример еще более крутой головоломки. Чтобы войти в замок, Вам необходим свиток (Scroll of Scar) и специальный ключ (B-Key). В замок Вы войдете, но первая же дверь не откроется никакими стараниями. Вам надо иметь при себе еще какой-нибудь предмет (любой) и пройти с ними в комнату, где есть три вращающихся колеса - это система кодового замка. Ваша задача выложить предметы на стол (по одному) так, чтобы колеса остановились в определенной комбинации. Код этой комбинации - DPE, хотя чтобы его получить, надо бы решить головоломку, выцарапанную на столе. После того, как колеса остановятся, надо взять свиток со стола и идти в открывшуюся дверь.

И после этого Вас еще будет ожидать очень много разных загадок и непонятных подсказок. Так, например, $2^{**}25$ - означает 2 в двадцать пятой степени. Если с помощью калькулятора Вы найдете, чему равно это число, то результат поможет выработать систему, с помощью которой можно будет пройти через длинные ряды дверей в замке.

В общем, друзья, это серьезная игра. Не удивляйтесь, если Вам придется поблуждать по улицам старинного города год - полтора, не меньше.

* * *

И, в заключение, обращение ко всем! В той части, которая касается адвентюрных программ, у нас большие архивы и возможности. Если Вы где-то "застряли" и Вам нужна оперативная помощь, пишите, не стесняйтесь. Может быть мы сами дадим полезный совет, может быть кто-то из наших читателей пройдет Вашим путем и решит проблему, но ведь под лежащий камень вода не потечет, давайте вместе решать эти задачи.

И предложение к тем, кто досконально разобрался с какими-либо адвентюрами. Если Вы можете стать внештатными консультантами, сообщайте нам, мы дадим Ваш адрес и напишем, по каким программам к Вам можно обращаться за помощью. Если будет достаточное количество писем, этот раздел может стать и постоянным.

HEAVY ON THE MAGIC



(с) Троеглазовы П. и Г. 1993
Хабаровский край, с. Гайтер.

Здравствуйте!

В N6 за 1992 год Вы давали описание программы "HEAVY ON THE MAGIC". Каракашев А.Г. неплохо поработал с программой, но раскрывая листинг программы, он не догадался раскрыть и проанализировать отгружаемый блок, а именно здесь можно было найти путь к завершению программы или, по крайней мере, приблизиться вплотную к решению главной задачи - найти выход из замка.

Сначала о неточностях в описании. Дух в котле не AU, а AI. Магических званий - 9, но о них позже. В программе 188 активных предметов. Корреспондент пишет, что побывал во всех комнатах. Это не так.

Составив карту, мы заметили, что на первом этаже угловые комнаты справа не имеют входов. Ясно, что одна из них FURNACE. Мы быстро выяснили, что эта "пекарня" находится в верхнем правом углу, под лестницей. А вот в комнату рядом с Циклопом можно попасть, видимо, только при определенных условиях. Может, нужно разрушить стену или здесь поможет пресловутый ERLSTONE? Во всяком случае, нам удалось побывать в этой комнате не с игры, а искусственно задав соответствующие координаты Аксилу. В комнате - абсолютная темнота. Ее можно осветить на долю секунды, используя любое из заклятий и при вспышке можно увидеть 2 предмета - прах от убитого монстра и книгу Гримуар. Выход из комнаты один - "W" и он выводит Вас на 4-й этаж в район PIT (?). Комната полна странностей, например, все вызванные предметы исчезают из нее, как в любой другой комнате. Апекс, вызванный ранее, не уходит, сколько бы Вы ему не говорили "APEX, THANKS", он отвечает - "GLAD TO HELP" и остается в комнате. Мы можем поднять Гримуар, у нас будет два Гримуара, а на том же месте будет стоять еще один, поднимем его у нас три книги, а на полу еще одна, - и так без конца. Каждый из поднятых Гримуаров содержит столько же заклинаний, сколько у Вас было до появления в этой комнате. Лишние предметы выкладываются и стоят во всех углах комнаты, кроме ее центра, где лежит прах. Если выложить что-либо на прах, предмет исчезает. Заклятье BLAST в этой комнате не действует на предметы. Выйдя из этой комнаты, Аксил уже не может вернуться назад - позади стена.

Всего комнат 256. Счет ведется с первого этажа, слева направо - левая верхняя - N1, FURNACE - N8 и т.д. Счет комнат правого нижнего угла замка сдвинут на этаж вверх -

2 этаж - 64,
3 этаж - 128,
4 этаж - 192,
1 этаж - 0.

Видимо, в нулевую комнату можно попытаться войти с помощью заклинаний или каких-либо других действий из комнаты N255 (или N63?).

Теперь об отгружаемом блоке. Рассказывать о каждом байте долго, а их 1342, но вот основные:

Первые 27 байтов - сведений об Аксиле, их можно отгружать и загружать отдельно, используя опции главного меню 4 и 5.

2-й байт - стойкость (STAMINA) 3-й байт - ила магии (MAGIC) 4-й байт - удача (LUCK) 5-й байт - звание (GRADE) В этом байт такое соответствие цифр:

- 1 - NEOPHYTE (новичок)
- 2 - ZELATOR (соискатель)
- 3 - FRACTICUS (практикант)
- 4 - PHILOSOPHUS (философ)
- 5 - ADEPTUS MINOR (младший эксперт)
- 6 - ADEPTUS MAJOR (старший эксперт)
- 7 - ADEPTUS EXEPT (высший эксперт)
- 8 - MAGISTR TEMPLI (магистр храма)
- 9 - MAGUS (маг, чародей)

6-й байт - набранные очки (points)

Далее идут пять блоков по 4 байта - предметы, которые находятся у Аксила. И 27-й байт - наличие заклинаний:

- 224 - INVOKE, FREEZE, BLAST
- 240 - INVOKE, FREEZE, BLAST, CALL
- 248 - INVOKE, FREEZE, BLAST, CALL, TRANSFUSION

94-й байт - комната, в которой находится Аксил. Например, 8 - FURNAGE, 43 - STINGS, 61 - CLAWS, 108 - MISERI (старт) и т.д.

Блок байтов со 188 по 217 - монстры, постоянно обитающие в 30-ти комнатах, в эти комнаты не может заходить никто, кроме Аксила. Пока не будет убит монстр, в эту комнату нельзя вызывать никого, даже Апекса, иначе Аксил погибнет. Два монстра в экране - мгновенная смерть. Убив монстра, можно спокойно, не боясь, что кто-нибудь затопчет Аксила у входа, задать дальнейший маршрут или обдумать дальнейшие действия. В комнату не заходит никто даже после смерти монстра, но теперь сюда можно вызвать любого из обитателей замка. Пока монстры живы, соответствующие байты равны нулю. Смерть монстра обозначается номером комнаты, в которой он обитает, например:

- 189-й байт - 150 (тролль)
- 193-й байт - 138 (призрак)
- 197-й байт - 18 (оборотень)
- 200-й байт - 49 (вампир)
- 205-й байт - 63 (циклоп)
- 210-й байт - 82 (ленивец)
- 217-й байт - 253 (медуза) и т.д.

С 541-го до 1292-го идут 188 четырехкилобайтных блока - это "активные" предметы. Активными мы называем предметы, непосредственно участвующие в игре, их можно брать, переносить, в крайнем случае - осматривать и изучать или уносить. Пассивные - фон. Все активные предметы, как правило, участвуют в отгрузках.

Из 4-х байтов, обозначающих какой-либо предмет, важны 2-й и 4-й: 2-й - комната, в которой находится предмет, 4-й - его координаты в комнате.

В игре 19 монстров. Каракашев, видимо, забыл упомянуть об одном из них - это голова дракона "RABAK". Она не пропустит Вас, пока Вы не скажете - "RABAK, WATER". На все вопросы у "дракоши" один ответ, но вот на фразу - "RABAK, DRAGON" он ответил нам "Мне горячо?". "Ах, тебе горячо!!!" - подумали мы и предложили ему водички и RABAK открыл нам дорогу к последнему "танцевальному залу" в "LIGHTGATE".

Апекс не то, чтобы "ходячая энциклопедия", но все-таки знает немало. У него в запасе около 40 вариантов ответов и, заметьте, в разных местах замка он может ответить по-разному на один и тот же вопрос. В некоторых случаях он прямо подсказывает, как вызвать птицу Феникс, как пройти воду или как пройти через пропасть. Подсказывает, как подобрать ключи к дверям, ведь над каждым ключом нарисован знак и на вопрос "APEX, SIGN", он

отвечает - "Это звездный знак", ключей 12, знаков Зодиака тоже 12. Остается сравнить знаки, нарисованные в звездах над ключами с названиями комнат и станет ясно, каким ключом открывать какую дверь (смотри рис. 1).



Рис.1

Про знак с двумя кольцами Апекс говорит - "Это знак пошлости". К этим дверям не нужны ключи, просто нужно поставить на стол мешочек с золотом "BAG". Про знак, у дверей с односторонним проходом, он говорит - "Знак - хода нет".

Остальных обитателей замка он явно недолюбливает:

WYVERN и TROLL - "Лучше их убить!"

CYCLOPS - "Один глаз и полный идиот!" и т.п.

Апекс, даже ничего не говоря, подсказывает слово SORONOROS. Обратите внимание на его живот, видите знак? Теперь прикиньте его к надписи на стене...

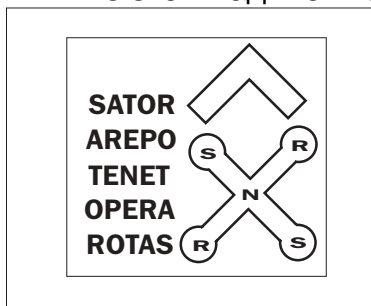


Рис.2

А самое главное - он подсказывает, где искать выход - "APEX, EXIT", ответ - "PILE TOMBS AND PARADISE". Такие комнаты есть на 2-ом и 4-ом этажах - N65 и N242.

Но нам все-таки кажется, что та самая странная "темница" на первом этаже должна дать ключ к выходу на свободу, ведь попасть в нее еще пока не удавалось никому (естественным путем).

У нас еще осталось 10 неиспользованных предметов. Каракашев пишет о 4-х, но кроме 3 костей (BONE) и одной "RIB" куда-то нужно пристроить "FOOT", "GRIMOIRE" (в нулевой комнате), 2 оставшиеся сумочки с золотом (BAG) и 2 золотых камня, похожих на SNAKE - "ROCK". Камни тяжелые, когда Аксил их поднимает, он теряет 10 единиц энергии, но не вздыхает "О-Н-Н...", как если бы он брал отравленный предмет.

Волчья лапа "FOOT" поднимает на несколько единиц цифру в графе "Удача". (Это же происходит, когда мы берем FLASK).

Может быть, какие-то из этих вещей нужно принести в комнату PILE TOMBS и тогда откроется вход в эту странную комнату с прахом в центре и бесконечным Гримуаром рядом с прахом, а уже потом идти в PARADISE. Но для того, чтобы проверить эти догадки нужно еще найти или хотя бы узнать, что такое EARLSTONE, ведь он должен подсказать, как

открыть дверь в LIGHTGATE. Тот путь, что подсказывает Каракашев А.Г., видимо неправильный, т.к. попав в комнату с рубином с помощью Астарота, разрушив дверь с помощью Асмодея, мы входим в дверь. Аксил исполняет танцевальный номер, но очки... убывают. Значит, недостаточно этого, чтобы найти вход - необходимо открыть эту последнюю дверь.

У "темницы" есть еще одна особенность - принесенные с собой предметы можно множить до бесконечности (эффект Гримуара), нужно только поставить на пол необходимый предмет, затем поднять его и т.д. Мы только не проверили, что будет, если набрать этих "фантомов", войти в обычную комнату, выставить их, затем выйти из комнаты и снова войти - скорее всего они исчезнут - проверьте.

Недостаточно ясна и роль FURNAGE. Обычно туда забрасывают Аксила демоны, если он им надоест, отбирая у него все предметы и магию. Но... Возьмите с собой CLASP, другие необходимые предметы и попросите Астарота отправить Вас туда. CLASP предохранит Аксила от огня, он может ходить по комнате, исследовать ее, вызывать предметы и демонов и, если они будут недовольны, то дальше этой комнаты не бросят Аксила. Вот только загвоздка - как оттуда выбраться, может вызвать "фантом" SWORD и снова использовать Астарота?...

Да, кстати, нулевая комната приготовила для нас еще один сюрприз - если мы будем брать лежащий там Гримуар и снова ставить его, у нас будет накапливаться сила магии. Таким образом, ее можно довести до 99 единиц.

В предыдущем описании был неправильно назван один из трех предметов, необходимых для вызова духа - AI. Это - SKULL, а не HEAD. Оба черепа HEAD непригодны к использованию. Только поняв, что на предметах, имеющих важное значение, как правило, есть какие-то знаки или цифры, нам удалось найти третий необходимый предмет для вызова AI.

Каракашев пишет, что он набрал 99 единиц в графе "очки" и, мол, странно, что Аксил не получил свободы - игра то сделана! Дело в том, что очки начисляются не только за прохождение маршрута и выполнение необходимых условий. За каждого убитого монстра начисляются очки, например, за Тролля - 1, за Змея - 2, за Медузу - 8 и т.д. Чем сильнее противник, тем больше очков Вы набираете. Можно набрать 99 очков не выходя из какой-либо комнаты, где часто появляются монстры. Количество очков важно для применения TRANSFUSION. Если у Аксила кончается энергия, он может воспользоваться этим заклинанием и десять единиц перекочают из графы "очки" в графу "стойкость". Вы можете повысить стойкость Аксила до 99-ти, но в этом случае у Вас может оказаться 0 очков. Поэтому пользуйтесь "TRANSFUSION" экономно, ведь для выхода на свободу Аксилу несомненно нужен полный комплект и выполненных действий и очков.

В начале игры Вы можете изменить (перетасовать) цифры в графах "стойкость", "магия", "удача". Лучший вариант, когда меньшая цифра в графе "стойкость", большая в "магии" и средняя в "удаче". Просто после старта нужно быстренько пробежать через 3 комнаты и подкрепиться хлебом (10 единиц), а затем возвратиться к старту и взять Гримуар. После того, как Вы подберете недостающие заклинания CALL и TRANSFUSION, Ваша сила магии повышается до 50 единиц и Вы будете уничтожать слабых и средних противников с первого удара. В этом случае Ваше путешествие значительно облегчается. Кроме этого, можно повысить силу магии, переноса с собой магические предметы - MANTIS, SWORD, SNAKE и т.п.

Мы подозреваем, что повысить звание выше PHILOSOPHUS в этой программе невозможно, хотя искусственным путем (изменив 5-й байт в отгружаемом блоке) мы смогли их получить. Но, возможно, каким-то путем их можно получить в комнатах PILE COLLODON, PILE TOMBS, PARADISE и в той же самой темнице на первом этаже, а уж звание Мага - только на выходе на свободу. Возможно...

К тому же, в программе есть еще масса не используемых слов - зачем они?

BRONZE, BIRD, CRYPT, CHROMA, COPPER, COBALT, FOUNTAIN, IPSISSIMUS, MAGNAM, MONSTER, NICKEL, NEARBY, OGRE, PASSWORD и т.д.

ЧИТАТЕЛЬ ЧИТАТЕЛЮ

Своими успехами в области программирования на Бейсике делится с нами т. Дронов из Читы. Его опыт пригодится начинающим программистам.

"Высылаю Вам на суд свою первую программу и хочу сказать Вам большое спасибо за "ZX-РЕВЮ". Благодаря Вам я, изучив за месяц два годовых комплекта "ZX-РЕВЮ", узнал очень много хитростей и секретов Бейсика. Подтверждением этого является эта программа, написанная за один вечер. Я знаю, Бейсик - легкий язык, но без хорошей книги его не ПРОСТО освоить. А Ваша литература лучшая из всех, которую я видел. Информация излагается очень понятно и доступно. "Сейчас я, с помощью Вашей книги, приступаю к изучению Ассемблера. Надеюсь овладеть и этим языком, хотя он намного сложнее.

Теперь несколько слов о программе. Это аналог широко известной настольной игры "Морской бой", в которую мы частенько играли в детстве. Маленькое отличие заключается в том, что в моей игре корабли можно располагать как угодно и где угодно в пределах своего квадрата. Так же делает и компьютер."

ИФК: Спасибо за теплые слова. От себя хотим добавить несколько слов. Корабли в этой игре - только одноклеточные, их можно ставить и поодиночке, и вплотную друг к другу как угодно. После попадания корабль сразу же считается потопленным. Всего их 20 у Вас и столько же у компьютера. Посмотрите на листинг программы: действительно, это занятие на один вечер. Если организовать в программе ограничения на расположение кораблей, на их конфигурацию (одно- или многоклеточные и т.д.), программа получилась бы как минимум, раз в пять длиннее. Но можем Вас уверить, что набрав самостоятельно эту программу, Вы с не меньшим азартом будете вести сражения с компьютером. А может быть, вскоре у Вас появится желание внести в программу ограничения на конфигурацию и расположение кораблей - что же, пробуйте смело. Лучшее обучение программированию - это обучение "в бою".

Автор предлагает русифицировать программу при помощи символов UDG-графики. Однако, поскольку мы уже привыкли к постоянно используемому нами методу русификации, мы приводим, как обычно, вариант для загружаемого русско-латинского символьного набора. Думаем, что автор простит нам такое вторжение в его программу и еще несколько непринципиальных штрихов. Итак, предлагаем программу всем читателям.



Рис. 1

```
1 GO TO 10
2 BORDER 7: PAPER 7: INK 0: CLEAR : LOAD "chr" CODE 64600
4 GO TO 0
5 SAVE "SEA" LINE 2: SAVE "chr"CODE 64600,768: STOP
8 POKE 23606,88: POKE 23607,251: RETURN
9 POKE 23606,0: POKE 23607,60: RETURN
10 BORDER 5: PAPER 7: INK 0: CLS : POKE 23658,8: GO SUB 8
15 REM РИСУЕМ ИГРОВОЕ ПОЛЕ ===
20 FOR N=0 TO 9
30 LET Z$=CHR$ (N+65)
```

```

40 PRINT AT 7,N+4;N;AT 7,N+19;N;AT N+8,3;Z$;TAB 18; Z$
50 NEXT N
60 PRINT AT 0,1; PAPER 4;" DRONOV * TCHITA * 1993 "
70 PRINT AT 3,10; PAPER 7; INK 1; " МОРСКОЙ БОЙ "
80 DRAW 255,0: DRAW 0,175: DRAW -255,0: DRAW 0,-175
90 PLOT 112,112: GO SUB 630: PLOT 232,112: GO SUB 630
100 REM ФОРМИРУЕМ КООРДИНАТЫ КОРАБЛЕЙ КОМПЬЮТЕРА =====
110 PRINT #0;"ПОДОЖДИТЕ МИНУТОЧКУ, ПОЖАЛУЙСТА."
120 DIM A$(20,4): DIM B$(20,4)
130 FOR N=1 TO 10
140 LET Y=8+INT (RND*10): LET X=4+INT (RND*10)
150 LET A$(N)=CHR$ 22+CHR$ Y+CHR$ X
160 IF N=1 THEN GO TO 200
170 FOR M=1 TO N-1
180 IF A$(N)=A$(M) THEN GO TO 140
190 NEXT M
200 NEXT N
210 INPUT ;: PRINT #0;"Я ГОТОВ! РАСПОЛОЖИМ ВАШИ КОРАБЛИ": BEEP .5,40: PAUSE 50
220 REM РИСУЕМ КОРАБЛИ =====
230 FOR N=1 TO 20
240 GO SUB 690: LET Y=Y-57: LET X=X-29
250 LET G$=CHR$ 22+CHR$ Y+CHR$ X: LET B$(N)=G$
260 IF N=1 THEN GO TO 300
270 FOR M=1 TO N-1
280 IF B$(N)=B$(M) THEN GO SUB 740: GO TO 240
290 NEXT M
300 PRINT INK 1; G$+CHR$ 143
310 NEXT N
320 REM НАЧАЛО ИГРЫ =====
330 LET W=0: LET Q=0: BEEP .5,20
340 PRINT AT 20,4;20;TAB 19;20
350 LET N=INT (RND*2): REM РОЗЫГРЫШ ПЕРВОГО ХОДА =====
360 IF N THEN GO TO 520
370 PRINT #0;TAB 10;"ВАШ ВЫСТРЕЛ"
380 PAUSE 40
390 IF Q=20 THEN INPUT ;: PRINT #0;TAB 4;"ПОЗДРАВЛЯЮ! ВЫ ПОБЕДИЛИ": FOR K=10 TO 15: BEEP
.1,K: NEXT K: GO TO 790
400 GO SUB 690: LET Y=Y-57: LET X=X-44
410 IF SCREEN$ (Y,X)="+" THEN GO TO 620
420 IF SCREEN$ (Y,X)="X" THEN GO TO 620
430 GO SUB 680
440 FOR N=1 TO 20
450 IF F$=A$(N) THEN PRINT #0; TAB 4;"БРАВО! ПРЯМОЕ ПОПАДАНИЕ": LET A$(N)=C$+"X": PRINT
PAPER 2;A$(N): LET Q=Q+1: PRINT AT 20,4;20-Q;" ": BEEP .5,1: PAUSE 20:GO TO 390
460 NEXT N
470 PRINT C$+"+": BEEP .1,40
480 PRINT #0;TAB 14;"МИМО!"
490 REM ПЕРЕХОД ХОДА =====
500 PAUSE 40: INPUT ;
510 IF W=20 THEN PRINT #0;TAB 6; "ВАША ФЛОТИЛИЯ РАЗБИТА": BEEP .8,2: BEEP .6,2: BEEP .2,2:
BEEP .8,2: GO TO 760
520 PRINT #0;TAB 10; "МОЙ ВЫСТРЕЛ": PAUSE 20
530 LET Y=8+INT (RND*10): LET X=19+INT (RND*10)
540 IF SCREEN$ (Y,X)="+" THEN GO TO 530
550 IF SCREEN$ (Y,X)="X" THEN GO TO 530
560 GO SUB 680
570 FOR N=1 TO 20
580 IF F$=B$(N) THEN LET B$(N)=C$+"X": INPUT ;: PRINT #0; TAB 6;"ВАШ КОРАБЛЬ ПОТОПЛЕН":
PRINT INK 1;B$(N): LET W=W+1: PRINT AT 20,19;20-W;" ": BEEP .5,1: GO TO 500
590 NEXT N
600 INPUT ;: PRINT #0;TAB 5;"ВАМ ПОВЕЗЛО, Я НЕ ПОПАЛ": PRINT C$+"+"
610 BEEP .1,40: GO TO 380
620 PRINT #0; "ВЫ СЮДА УЖЕ СТРЕЛЯЛИ": GO TO 380
630 RESTORE
640 FOR N=0 TO 8

```

```
650 READ Y,X: DRAW Y,X
660 NEXT N
670 RETURN
680 LET C$=CHR$ 22+CHR$ Y+CHR$ X: LET F$=C$+" ": RETURN
690 INPUT "ВВЕДИТЕ КООРДИНАТЫ " ;D$
700 IF LEN D$<2 THEN GO SUB 740: GO TO 690
710 LET Y=CODE D$(1): LET X=CODE D$(2)
720 IF Y<65 OR Y>74 OR X<48 OR X>57 THEN GO SUB 740: GO TO 690
730 RETURN
740 PRINT #0;TAB 12; "ОШИБКА! "
750 BEEP .5,20: PAUSE 40: RETURN
760 FOR N=1 TO 20
770 IF A$(N,4)=" " THEN PRINT PAPER 2;A$(N)
780 NEXT N
790 PAUSE 70: INPUT ;: PRINT #0;TAB 4;"ХОТИТЕ ЕЩЕ РАЗОК? (Y/N)": PAUSE 0
800 IF INKEY$="N" THEN GO SUB 9: STOP
810 RUN
820 DATA 0,-81,-81,0,0,81,81,0,0,-2,2,0,0,-81,-81,0,0,2
```

СОВЕТЫ ЭКСПЕРТОВ

Fighter Bomber

* VECTOR GRAPHICS *

* ACTIVISION *

Эксперт: Сороченко С. В.

г. Норильск



Загрузка игры

После загрузки первого блока Вам предложат ввести имя пилота и выбрать один из четырех самолетов:

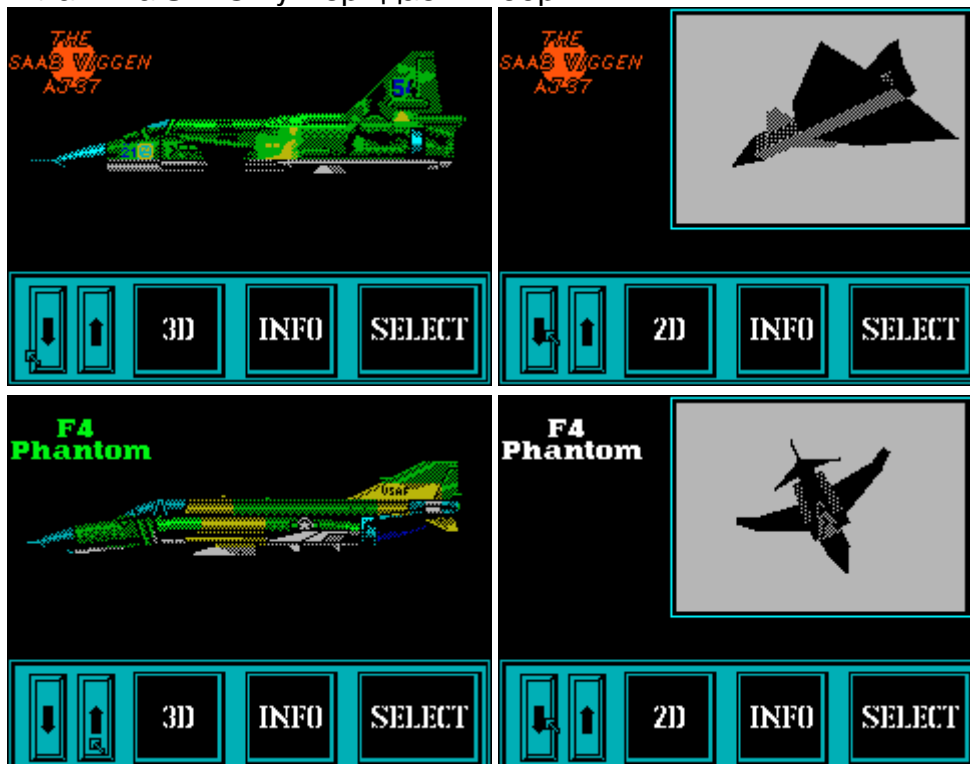
SAAB VIGGEN AJ 37 - самолет совместного производства шведских и британских авиастроителей.

F-4 PHANTOM - ВВС США

TORNADO - Британские ВВС

MIG-27 ("FLOGGER") - советские ВВС. (Кстати, на рисунке изображен вовсе не MIG-27 FLOGGER, а MIG-23 FLOGGER - это я проверил в справочнике. Информация в программе также дана на MIG-23, а не на MIG-27).

Итак, на экране показан SAAB. Клавишами A,Q,O,P можно перемещать курсор по пиктограммам. Клавиша SPACE утверждает выбор.





Пиктограммы "ВВЕРХ" и "ВНИЗ" - это просмотр самолетов.

3D - это самолет в трехмерном виде.

INFO - информация об этом самолете.

SELECT - выбор штурмовика. После нажатия SELECT загружаются миссии, которые Вы можете выбрать. На экране появятся надписи:

Free Flight - свободный полет

Covert - военная миссия.

Ниже размытым шрифтом написаны остальные военные миссии, которые Вы сможете выбрать только после выполнения предыдущей миссии. В каждой миссии по два задания.

Сначала для ознакомления с аспектами полета следует выбрать Free Flight - появляется новое меню:

On Runway - на взлетной полосе.

At 30000 ft - на высоте 30000 футов.

Above Bridge - над мостом.

Lined Up - приземление.

ENTER выход в предыдущее меню.

Выберем On Runway и загружаем собственно саму игру. Внимание! В зависимости от типа выбранного самолета, после загрузки этого блока нужно загрузить еще два маленьких блока. Например:

"TORNADO" - игра запускается сразу без подзагрузки маленьких блоков.

"SAAB" - первые два блока.

"MIG" - вторые два блока.

"FANTOM" - третья пара блоков.

После этого на экране появится вид из кабины самолета.

Управление самолетом

Клавиши:

Q,A - тангаж.

O,P - крен.

W - торможение с помощью шасси.

B - воздушные тормоза.

CAPS SHIFT + 1...0 - изменение тяги двигателя (1 - наименьшая, 0 - наибольшая), при двукратном нажатии CAPS + 0 двигатели переходят в форсированный режим.

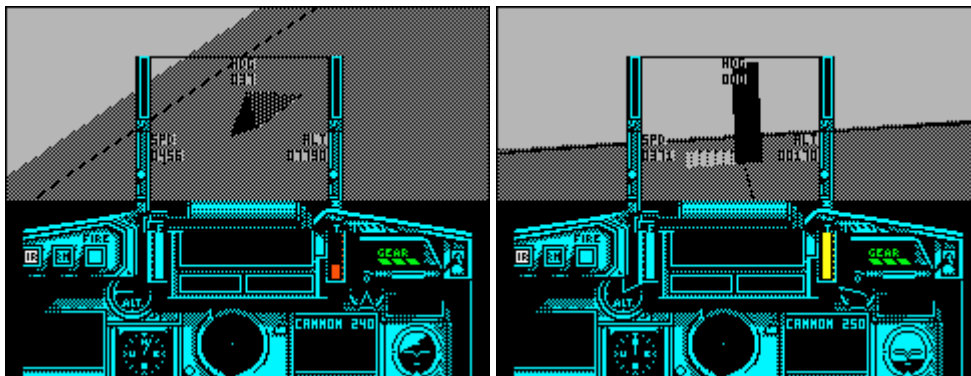
ENTER - производит переключение на один из трех видов вооружения.

CANNON - пушка;

MAVERICK - ракеты "воздух-земля" с наведением по лучу радара.

SEWINDER - ракеты "воздух-воздух" с инфракрасным наведением.

Клавиша G убирает и выпускает шасси (только в воздухе).



Приборная доска

В центре - радар с цифрой, указывающей дальность его действия в милях. Эта цифра изменяется нажатием клавиши R.

Так как в разных самолетах расположение остальных приборов разное, то описывать их расположение не буду, а назову только функции.

Квадратик с надписью IR становится красным, если по Вам выпустили ракету (только в боевых миссиях).

Прибор с буквой N в верхней части - компас.

Прибор, показывающий тангаж и крен выглядит характерно для всех имитаторов.

Столбик с буквой T рядом - указатель тяги.

Столбик с буквой F - горючее.

Также имеются окно вооружения, которое показывает, сколько и какого у Вас боезапаса осталось, и окно бортового компьютера, предупреждающего об опасности и других важных вещах.

Прибор с надписью ALT - альтиметр (Ваша примерная высота).

Прибор с цифрами 0, 1, 2, 3 - Ваша скорость в MAX-ах.

Ваши высота, скорость, а также направление показаны в центре основного экрана в точных цифрах.

Основная прелесть этой игры, на мой взгляд - это возможность обозревать свой самолет во время игры с самых различных позиций.

Итак,

SYMBOL SHIFT+1 - обзор из кабины самолета.

SS+2 и SS+3 - неизвестно зачем.

SS+4 - посмотреть на самолет с вышки, находящейся на ВПП. Скорее всего, Вы увидите просто точку в небе, но клавишами K и J можно увеличить или уменьшить изображение.

SS+5 - посмотреть назад.

SS+6 - посмотреть влево.

SS+7 - посмотреть вправо.

SS+8 - посмотреть сверху, работают клавиши K и J

SS+9 - посмотреть сзади, клавиши K и J функционируют.

SS+0 - посмотреть на самолет с южной стороны, то есть с той стороны, в которую он направлен в начале игры. Это полезно, чтобы оценить Ваше положение относительно ВПП. Этот режим позволяет кроме увеличения и уменьшения изображения, вращать изображение по горизонтали клавишами I и U, а также по вертикали, - D и E.

Полет

Полет в принципе ничем не отличается от других имитаторов, но маневрирование, при взгляде со стороны, производит большое впечатление.

Однако, наряду с достоинствами, есть несколько досадных недоработок, которые

снижают впечатление, как об имитаторе.

Так, например, даже на самой большой тяге можно лететь с выпущенными шасси и ничего плохого, кроме снижения скорости не случится.

Самолет почему-то может садиться в любом месте, а не только на ВПП, поэтому посадка не представляет никакой сложности. Можно сесть где-то в районе ВПП, а потом подрулить к ней.

Но самый, на мой взгляд, грубый просчет состоит в том, что можно рулить хвостовым оперением (клавиши Z и X) не только на маленькой скорости на ВПП, но и при скорости 1200 миль/час в воздухе. Где это видано, чтобы самолет разворачивался в воздухе без крена?! Такого не выдержит никакое хвостовое оперение.

Военные миссии

Я не стану описывать все миссии - это займет слишком много места, а описание получилось и так слишком длинным.

Первая миссия, которую Вы можете выбрать - это COVERT. После этого на экране появятся две подмиссии. Вы можете выбрать только первую, так как вторая для Вас пока закрыта.

Первая подмиссия - SLEEPER.

Вам покажут участок карты США. Обязательно надо ознакомиться с текстом миссии, иначе не будете знать полетного задания.

Пиктограмма INFO предложит Вам выбрать один из пунктов на карте. По нажатии "FIRE", в окне появится надпись, что это за объект.

Пиктограмма "OK" означает, что Вы ознакомились и решили начать действия.

В первой миссии SLEEPER Вам нужно уничтожить лагерь террористов, а затем вернуться на базу.

Это самая простая миссия. В ней нет самолетов противника, но если Вы пролетите слишком близко к цели, по Вам могут выпустить ракеты.

Целей в этой миссии две, хотя уничтожить нужно только одну. Вторая цель во внимание не принимается.

Итак, Вы стоите на аэродроме.

В "окне" показаны координаты цели. Клавишей N можно переключать на координаты Вашей базы.

HDG - направление,

RGE - расстояние в милях.

Переключите радар на 25 миль.

Справа появится точка - это цель. Взлетайте и ложитесь на курс. Когда до цели останется меньше 20 миль, переключайтесь на VP "Мейверик". На экране появится крест. Нажмите S и головка наведения ракеты захватит цель. При этом на экране бортового компьютера появится надпись LOCKED. Не рекомендую захватывать цель на расстоянии больше 20 миль - велика вероятность промаха. После захвата цели нажмите "пробел" - ракета попадет в цель. Следить за ней можно по радару. Если ракета попала, компьютер выдаст TARGET DESTROYED - цель уничтожена. Если компьютер выдал MISS TARGET DESTROYED - значит Вы уничтожили главную или одну из главных целей. После того, как Вы уничтожите все главные цели, можете вернуться на базу, приземлиться на взлетную полосу и выключить двигатели.

После этого Вас спросят, хотите ли Вы повторить миссию. Если нет, то нужно загрузить блок, после загрузки которого Вам снова покажут карту и курс, которым Вы летали, а также скажут, сколько целей Вы подбили и засчитывается Вам эта миссия или нет. После этого запросят, желаете ли Вы принять следующую миссию или начать игру с самого начала. Если Вы хотите сыграть в следующую миссию, снова загружайте блок миссий и выбирайте COVERT. Вы увидите, что открылась вторая подмиссия - BRIDGE END. Это уже намного более сложное задание, так как появляются самолеты противника. Самолеты можно сбить пушкой, что очень сложно. Но можно и с помощью ракет "Сайдуиндер". Если на радаре появился самолет - переключайтесь на "Сайдуиндер". При этом в экране

вооружения появятся расстояние до самолета, его высота, направление и скорость. Теперь нужно поймать его в центральный экран. После появления сообщения LOCKED - пускайте ракету и вражескому самолету не миновать уничтожения.

Ваш самолет в отличие от вражеских имеет возможность выпускать радиолокационные и тепловые ловушки против вражеских ракет. Радиолокационная - клавиша С, тепловая - клавиша F.

Если то, что Вы прочитали, Вам интересно, смело стартуйте в небо и пусть не отвернется от Вас Удача!

Mercenary "NOVAGEN" 1987

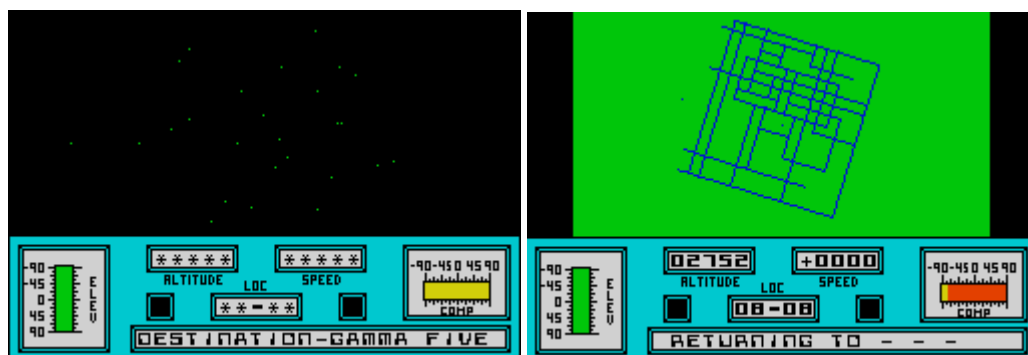


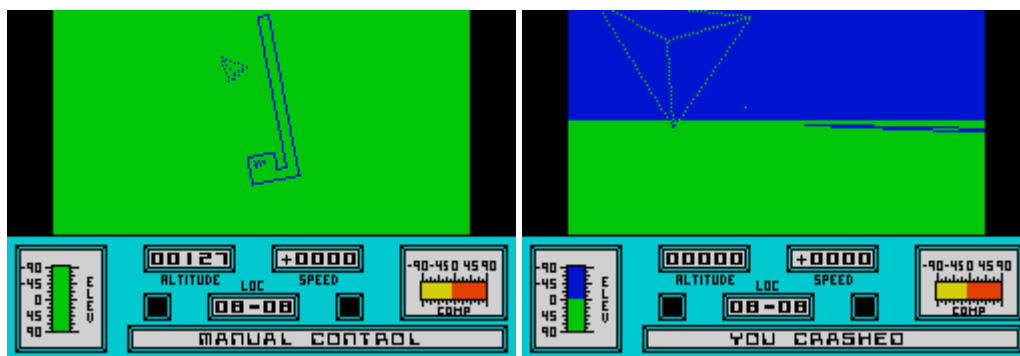
Эксперт: Кустов Г.В.
Г. Львов

Я хотел бы выступить в качестве неофициального эксперта и рассказать про очень интересную, на мой взгляд, программу, которую можно было бы поставить рядом с "ELITE". Называется она "MERCENARY". Про эту программу я нигде ничего не слышал и в Вашем журнале она тоже не затрагивалась. Думаю, мои исследования пригодятся любителям этой программы.

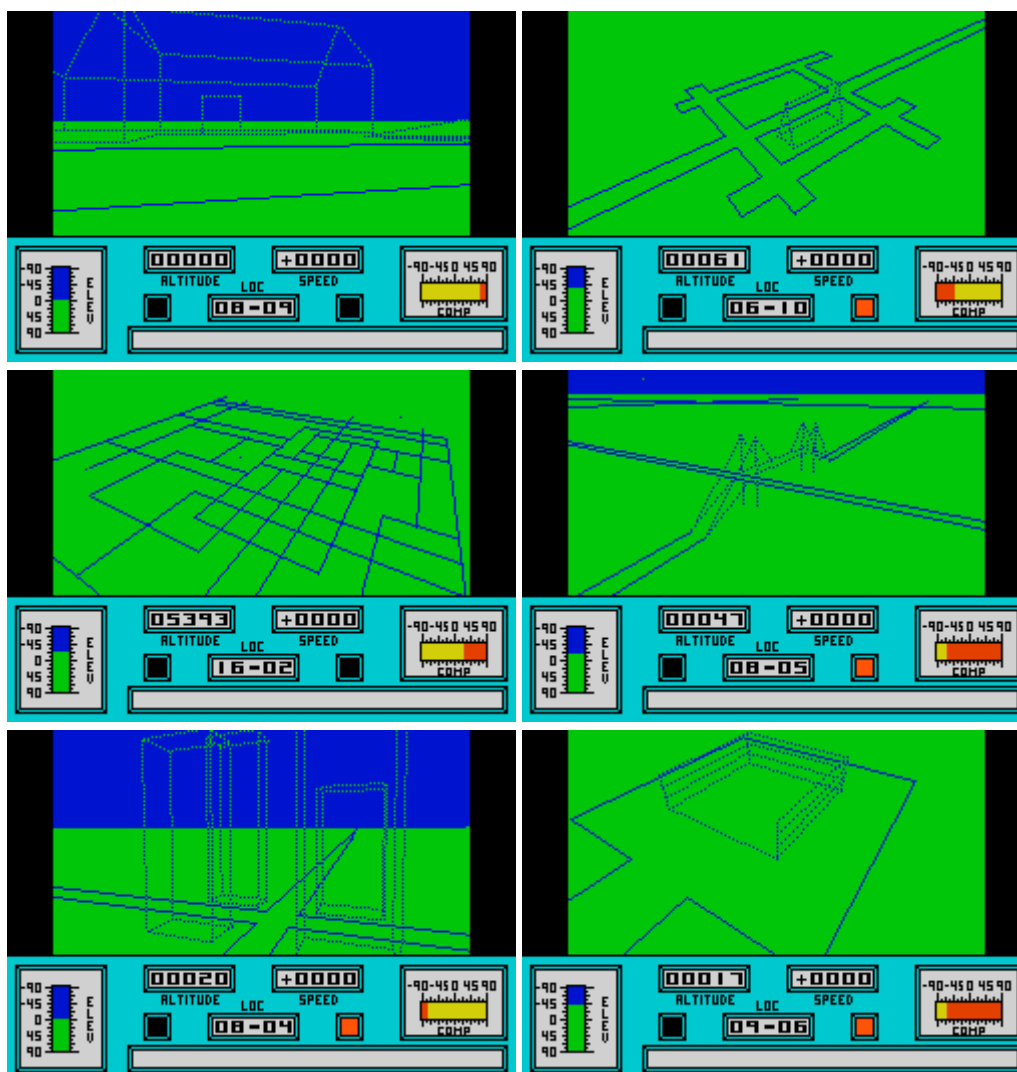
Итак, "MERCENARY ESCAPE FROM TARG" можно отнести к жанру ARCADE/ADVENTURE с элементами PUZZLE, так как главное - это умение ориентироваться в подземных лабиринтах и даже в надземных. В этой программе использована очень необычная объемная графика и для простоты видимые объекты нарисованы только из ребер, т.е. есть уникальная возможность проходить сквозь объекты и рассматривать их с любой из шести сторон света. Этот эффект настолько сильно захватывает, что забывается отсутствие геометрических плоскостей у нарисованных объектов. В программе возможно пользование несколькими видами транспортных средств. Управление транспортом имеет разные особенности для каждого вида транспорта.

Действие начинается с того, что Ваш звездный корабль стартует и берет курс на систему "GAMMA FIVE". После этого бортовой компьютер сообщает о повреждении системы управления и корабль сбивается с курса, который невозможно откорректировать. Вы летите прямо на какую-то планету, катастрофа неминуема и... Вы разбиваетесь.





Оказавшись на поверхности, Вы получите сообщение - о ходе войны между PLYAR и MECHANOID (PLYAR - это люди, MECHANOID - роботы). Вы находитесь возле аэробазы, где стоит самолет. У Вас с собой 9000 кредитов, а самолет стоит 5000. Все же, его желательно купить, иначе вряд ли Вы далеко уйдете пешком.



Итак, Ваша миссия: как наемник, работая на людей (хотя можно работать и на роботов) Вы должны найти главного робота и продать его людям для предотвращения войны, а также найти и доставить различные предметы. В виде премии за них Вы должны получить более 1000000 кредитов, на которые можно купить новый корабль и покинуть эту негостеприимную планету, завершив тем самым свою миссию.

Управление игрой.

Выбор управления возможен один раз, в начале игры - Вы выбираете один из трех возможных типов джойстика. Кроме выбранного джойстика есть управляющие клавиши на клавиатуре:

L - выход из транспорта,

B - вход в транспорт,

T - взять предмет,

D - положить предмет,

E - подъем/опускание лифта,

1,2,3,4 - переключение скоростей соответственно 1,2,3 и 4 скорости,

BREAK - сброс скорости до 0,

K - плавное увеличение скорости,

J - плавное уменьшение скорости,

CS+S (CAPS SHIFT + S) - вывод на ленту игровой ситуации,

CS+L - ввод с ленты отгруженной игры,

CS+Q - экстренный вызов самолета.

В случае управления героем, джойстик действует так:

"ВВЕРХ" - вперед,

"ВНИЗ" - назад,

"ВПРАВО" - вправо,

"ВЛЕВО" - влево,

"ОГОНЬ" - не действует. Если Вы управляете автомобилем или TURBO-автомобилем, джойстик действует так:

"ВВЕРХ" - не действует,

"ВНИЗ" - не действует,

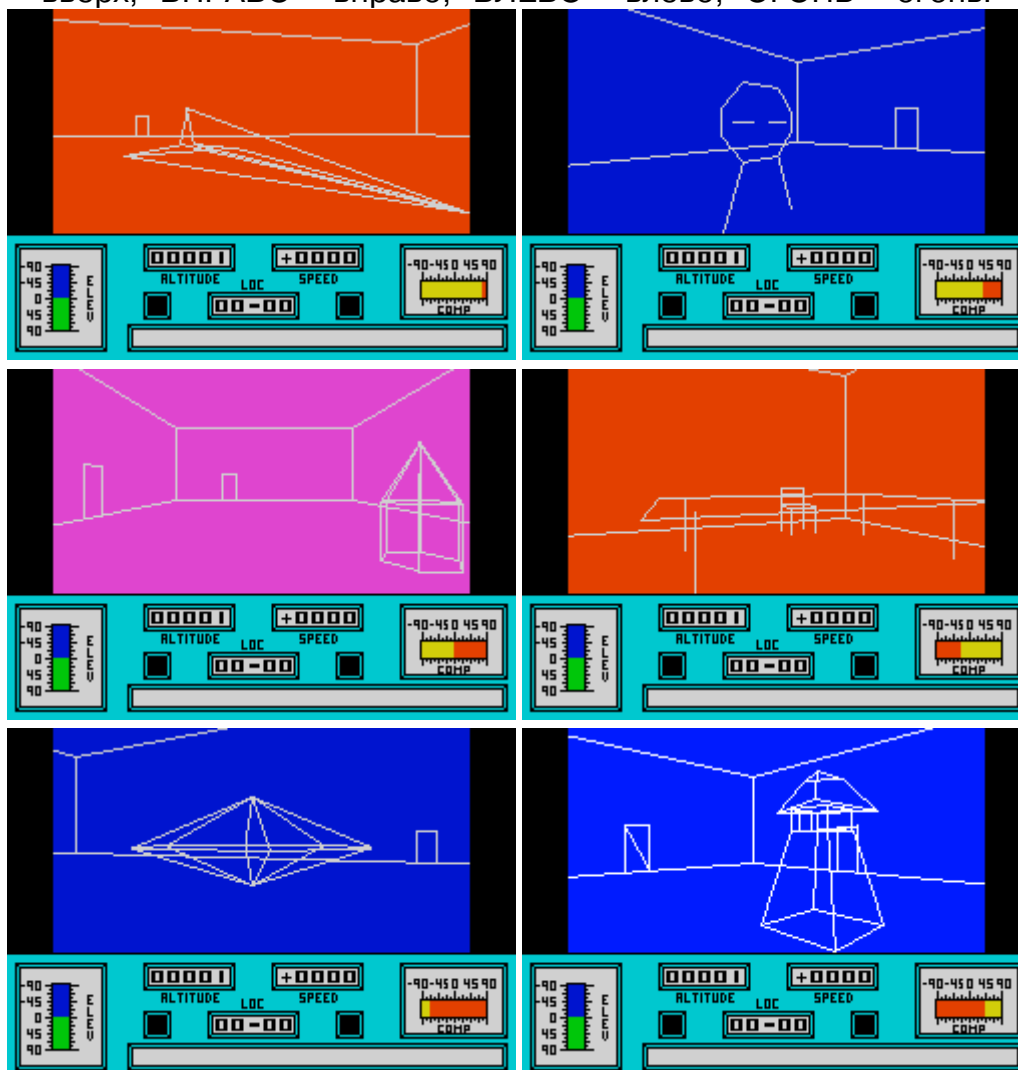
"ВПРАВО" - вправо,

"ВЛЕВО" - влево,

"ОГОНЬ" - огонь.

В самолетах и в летающей тарелке - следующее соответствие: "ВВЕРХ" - вниз,

"ВНИЗ" - вверх, "ВПРАВО" - вправо, "ВЛЕВО" - влево, "ОГОНЬ" - огонь.



Работа с лентой.

Выгрузку игровой ситуации осуществляют нажатием CS+S. После этого в строке для сообщений выдается запрос, какой блок выгружать 1...4. Это позволяет вывести на ленту четыре разных игровых ситуации. У каждого блока свой номер, который проверяется в момент загрузки. Пригодиться выгрузка под несколькими номерами может, когда Вы идете, к примеру, на рискованную операцию. Тогда Вы выгружаете игровую ситуацию до операции под одним номером, а после операции - под другим. Загрузка игровой ситуации происходит так же, как и выгрузка, только в начале надо нажать CS+L вместо CS+S.

Экран программы

После запуска программы в нижней части экрана появится таблица, отражающая текущую информацию.

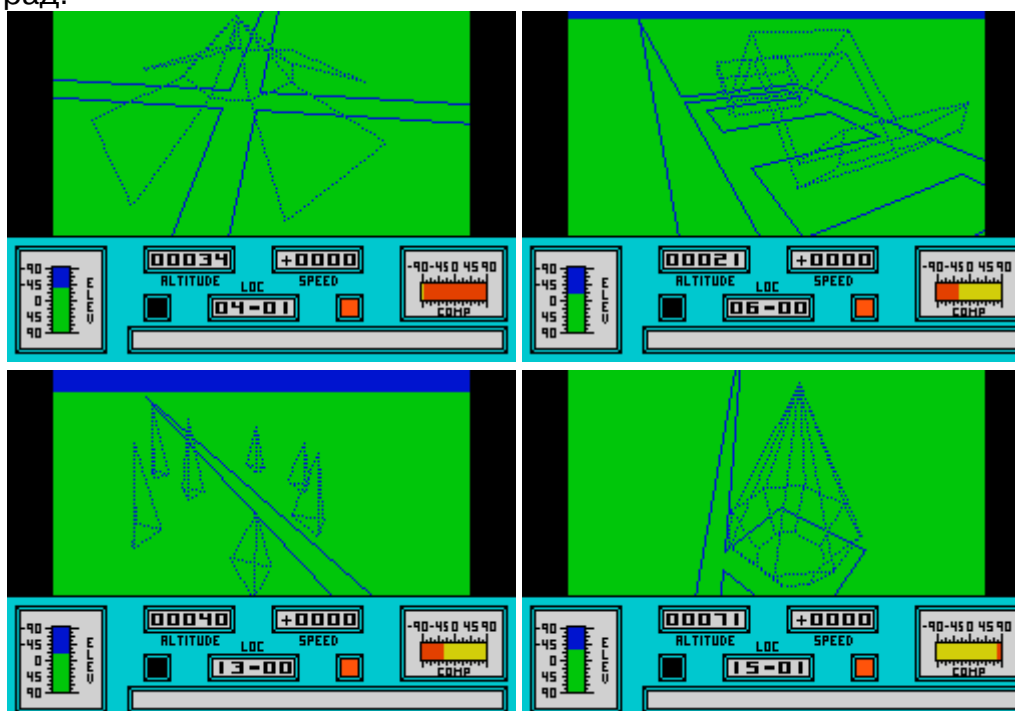


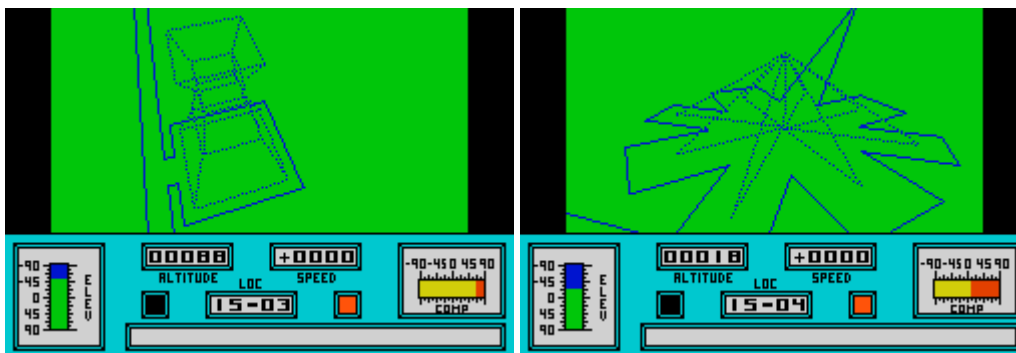
Рис. 1

- 1 - прибор, показывающий угол тангажа
- 2 - высотомер
- 3 - локатор
- 4 - спидометр
- 5 - компас
- 6 - строка информационных сообщений
- 7 - индикатор снижения высоты (действует, когда высота меньше 300 метров).
- 8 - индикатор металлоискателя.

Описание транспортных средств и предметов.

Итак, очутившись на поверхности незнакомой планеты, Вы получили предложение купить мобильный самолет, что и делаете нажатием клавиши Y. Этот мобильный самолет имеет фантастические возможности. Он за секунду разгоняется до 1500 км/час (!) и за столько же сбрасывает скорость до нуля. Можно плавным увеличением 4-ой скорости разогнаться и до 4900 км/час. Можно даже зависнуть на любой высоте и, кувыряясь во все стороны, как внутри шара, осматривать местность под любыми углами. Управление очень простое. Единственно, при поездке нужно выбирать 1-ую или 2-ую скорость с углом тангажа не более 15 град.





Имеется еще один самолет. Он находится в ангаре с координатами по локатору "***-***". Внешне он похож на современный истребитель. Его отличительная особенность - большая скорость (более 7000 км/час) и он позволяет взлетать на высоту до 99999.

В квадрате 03-00 в ангаре находится летающая тарелка. Основное отличие от самолетов - низкая скорость (1800 км/час), отсутствие крена при поворотах и почти полное отсутствие инерции.

В квадрате (координаты по локатору 09-05) в ангаре находится TURBO-автомобиль, который развивает скорость по земле до 1800 км/час.

В квадрате 11-13 возле ангара у дома стоит автомобиль. Он может разгоняться до 300 км/час.

Всего в игре имеется 7 ангаров с лабиринтами. Шесть подземных ангаров в квадратах: 09-06, 09-05, 03-00, 11-13, 81-35, **-* и один ангар находится на высоте 64997 метров над квадратом 06-08.

Метод отыскания ангаров. Вход в ангар выглядит, как маленькая площадка, обнесенная с трех сторон забором. Если ангар в черте города, то находится он возле дороги.

Примерно через две минуты после покупки самолета Вам сообщают, что нужно лететь в квадрат 09-06. Наберите высоту примерно 100 метров и, маневрируя самолетом, войдите в этот квадрат. Следите за высотой, можно разбиться. Попад в нужный квадрат, погасите скорость до нуля и осмотритесь, в поисках входа в ангар. Осторожно садитесь на землю и подкатите на колесах ко входу. Можно оставить самолет (или другой транспорт) на поверхности в удобном месте, а можно вместе с самолетом заехать на лифтовую площадку и опуститься в ангар. В ангаре и в лабиринтах транспорт не действует, поэтому нужно из него выйти и дальше идти пешком.

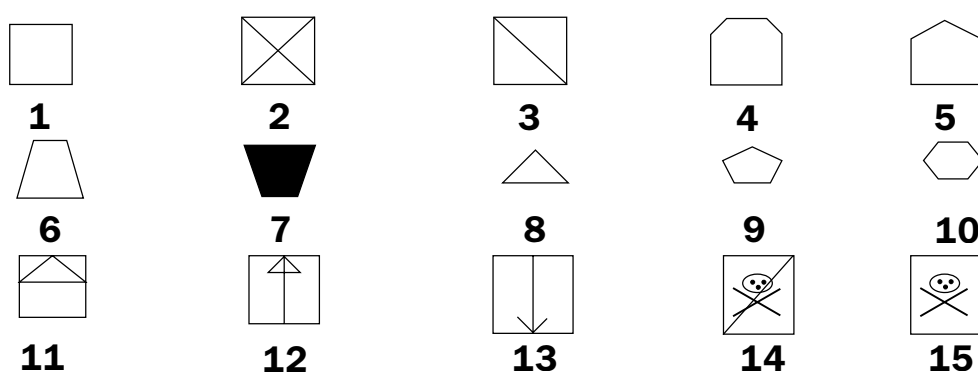


Рис.2

Обозначение	Назначение
1	Прямоугольная дверь. Можно войти без ключа.
2	Телепортатор двухстороннего действия (туда и обратно).
3	Телепортатор одностороннего действия (туда)
4	При попытке в нее войти будет сообщение LOCKED (закрыто). Чтобы пройти нужно иметь с собой кристалл такой же формы как и дверь.
5	то же
6	то же

7	то же
8	то же
9	то же
10	то же
11	При входе в нее будет очень темно, поэтому нужно выйти задом. Необходимо иметь с собой PHOTON EMITER (фотонный прожектор) для прохождения в эту дверь
12	Вход в лифт, поднимающий вверх.
13	Вход в лифт, опускающий вниз.
14	Войдя в нее Вы телепортируетесь и перепутается структура лабиринта относительно компаса (запад на восток), войдя во 2-ой раз, ситуацию можно исправить.
15	Если эта дверь находится в лабиринте, который находится на высоте 64997, то Вы вывалитесь и упадете на землю и растеряете все предметы которые были с Вами. Если эта дверь находится в подземном лабиринте, тогда из этой комнаты нет обратного пути. И придется вызывать самолет с потерей всех предметов или заново вводить игру.

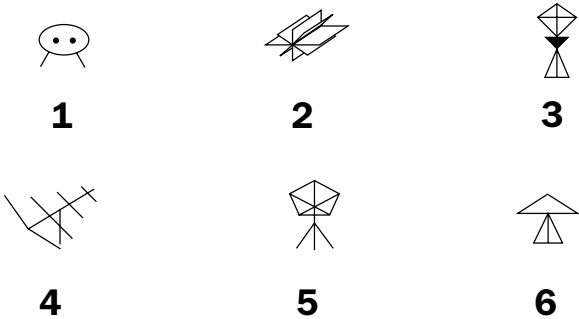
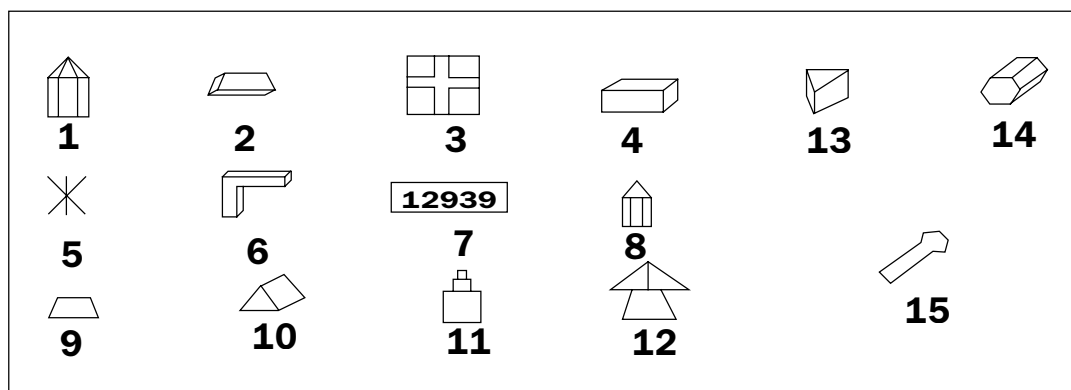


Рис. 3

Обозначение	Название	Назначение	Где найти
1	SIGHTS зрение	Включает прицел для точного наведения на объект	09-06
2	POWERAMP турбина	Дает возможность мобильному самолету развивать скорость до 9500 км/час и взлетать на высоту 99999 м.	09-06
3	ANTIGRAV антигравитатор	С его помощью можно брать с собой тяжелые предметы: такие как "нейтронное топливо" и все виды транспорта.	09-05
4	ANTENNA антенна	Имея ее в COMMUNICATION ROOM (квадрат 09-06) Вы можете купить ракету за 999999 кредитов Вам дадут информацию о ее местонахождении для того, чтобы на этой ракете завершить миссию.	09-05
5	METAL DETECTOR металлоискатель	Обнаруживает над землей вход в любой ангар	09-05
6	PHOTON EMITER фотонный прожектор	Необходим для прохождения комнат с дверями (11)	09-05



Обозначение	Название	Где взять	Где продать
1	ENERGY CRYSTAL энергетический кристалл	09-06	либо в 03-00, либо в POWER ROOM (08-08)
2	CATERING PROVISION провиант	09-06	в KETCHIN (08-08)
3	MEDICAL SUPPLIES медикаменты	09-06	в INFIRMARI (08-08)
4	LARGE BOX большой ящик	09-06	в INFIRMARI (08-08)
5	NEUTRON FUEL нейтронное топливо	09-05	в ENGINE ROOM (08-08)
6	USEFUL ARMAMENT оружие	11-13	в ARMOUERY (03-00) или (08-08)
7	ESSENTIAL 12939 SUPPLY PEPSI	03-00	в CONFERENCE ROOM (08-08)
8	MECHANOID робот	03-00	в INTERVIEW ROOM (08-08)
9	GOLD золото	81-35	в EXCHEQVER (08-08)
10	DATA BANK банк данных	**_**	в CONTROL ROOM (08-08)
11	WINCHESTER винчестер	**_**	в LABORATORY (03-03) или (08-08)
12	ANTI TIME BOMB антибомба с часовым механизмом	08-08	?
13	CHEESE сыр	11-13	?
14	NOVADRIVE управление	**_**	?
15	COFFIN гроб	09-05	?

Ангар в квадрате **_** обнаруживается так. Сначала надо долететь до квадрата 99-99, потом, при появлении квадрата **_** сбросить скорость и с высоты около 4000 метров посмотреть, где внизу вход в ангар.

Точка, висящая над городом это тоже ангар. Он находится на высоте 64997 метров. Нужно взлететь на эту высоту. Вы увидите в воздухе большую плиту, садитесь на нее. В одном из углов плиты есть квадратная площадка, являющаяся входом.

Теперь о дверях, которые встречаются в лабиринтах. В таблице (рис. 2) даны краткие сведения о дверях:

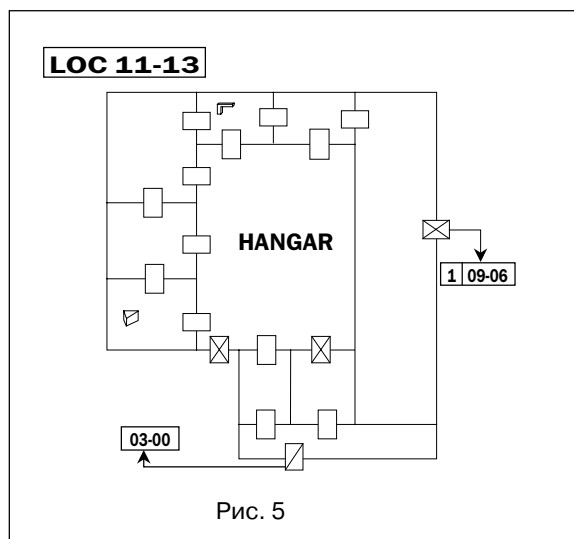
Все предметы, с которыми Вы столкнетесь по ходу игры, можно условно разделить на два класса. В одном классе предмет - товар, в другом классе предмет - навигационное средство.

В таблицу (рис. 3) внесены предметы, необходимые для навигации.

А теперь рассмотрим предметы, интересующие покупателей. Их можно использовать для продажи. В следующей таблице (рис. 4) дана информация о том, где взять товар и где его продать.

О предметах можно сообщить еще такую особенность. Если на базе (квадрат 03-00) взять MECHANOID (робота), то после этого нельзя здесь продавать предметы. С собой можно одновременно иметь не более десяти предметов. В таблице против некоторых предметов стоят знаки "?". Пока значение этих предметов невыяснено. Может быть, кто-нибудь выяснит этот вопрос?

В заключение прилагаю карты лабиринтов (рис. 5 ... 12).



На картах лабиринтов есть односторонние и двусторонние телепортаторы, не отмеченные куда они телепортируют. Некоторые из них телепортируют каждый раз в другое место а некоторые еще не изучены. Есть и другие неисследованные вопросы, например: почему в городе все строения можно сбивать выстрелами своего оружия? Что за самолет, постоянно летающий над городом по одному и тому же курсу на высоте 500 метров? Можно ли увидеть самого себя?

Успеха Вам! GOOD LUCK!

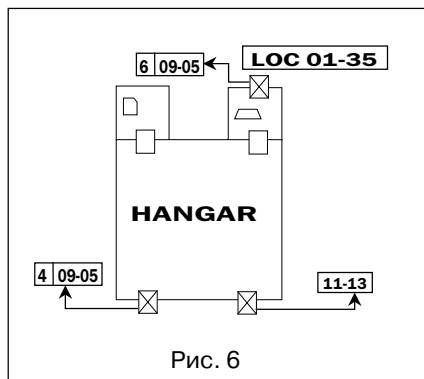


Рис. 6

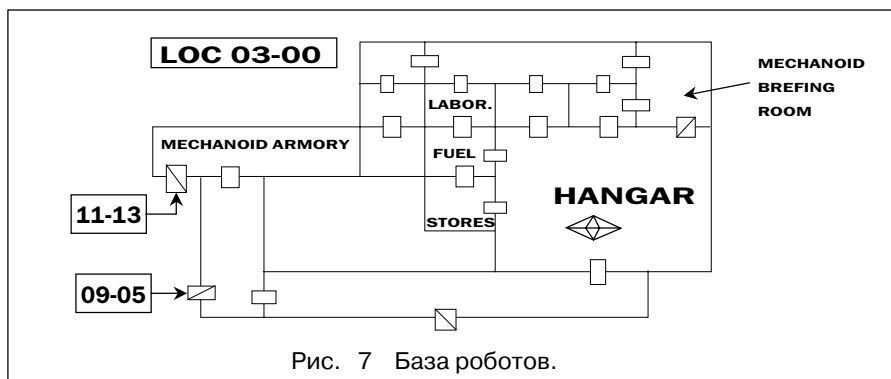


Рис. 7 База роботов.

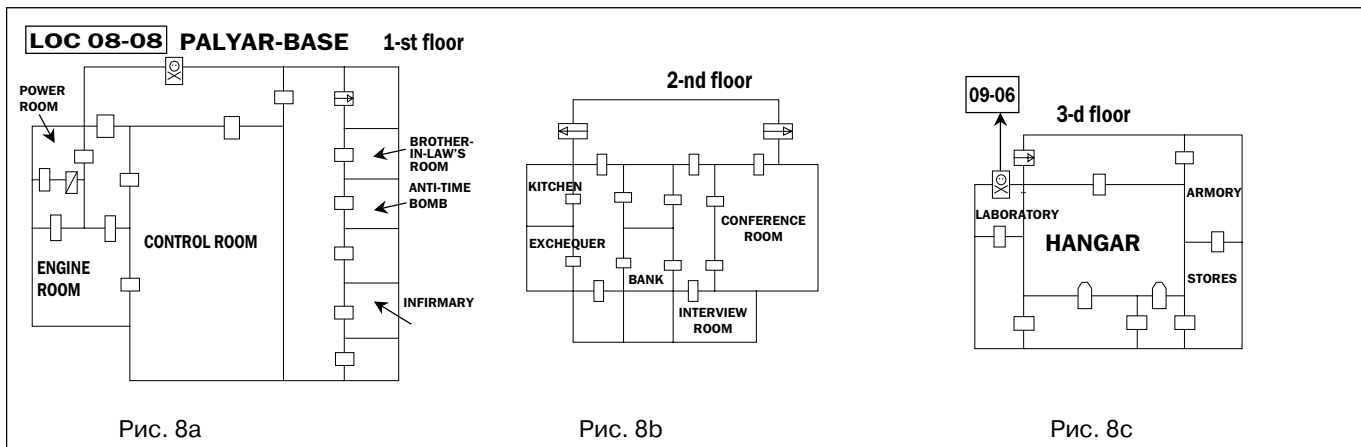


Рис. 8a

Рис. 8b

Рис. 8c

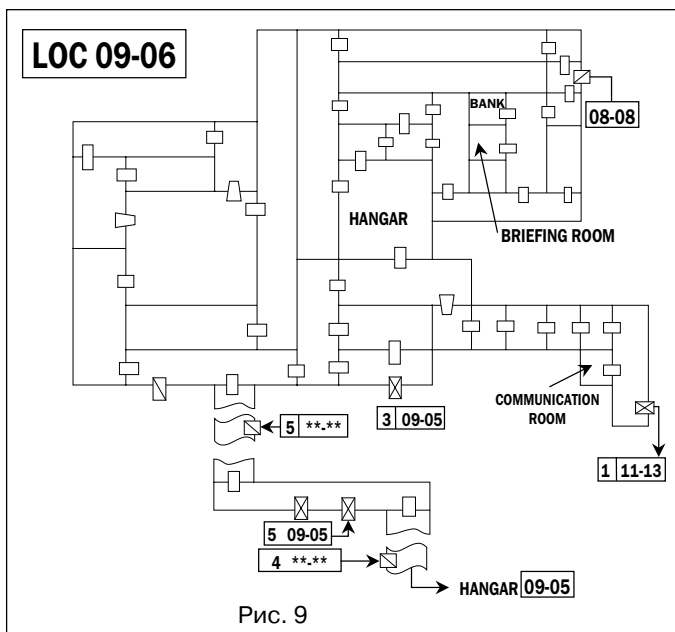


Рис. 9

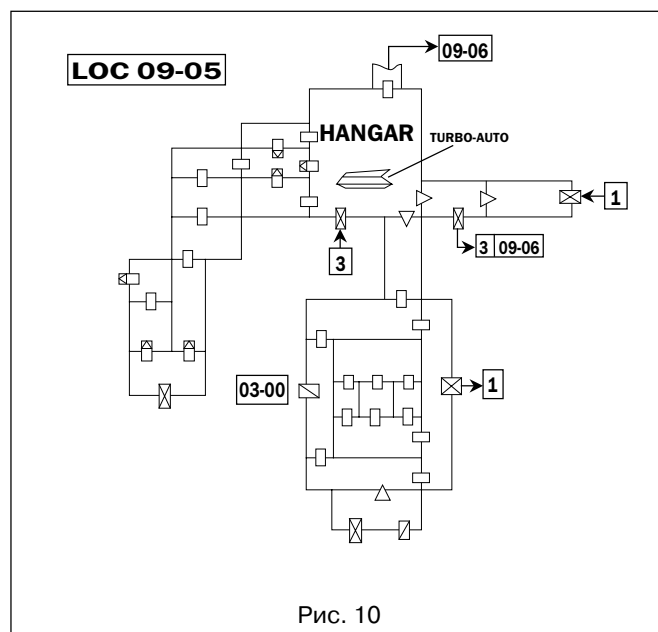


Рис. 10

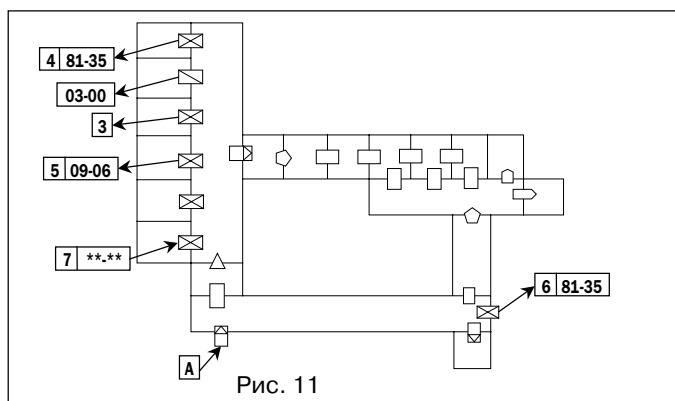


Рис. 11

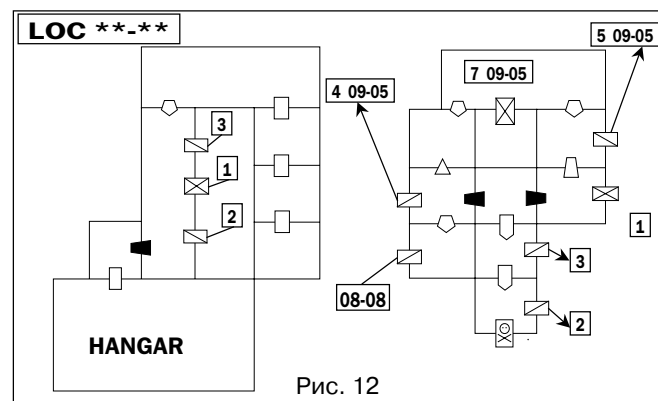


Рис. 12

Сделайте сами

DICEY

Сегодня в рубрике "Сделайте сами" мы предлагаем интересную настольную игру в кости. Это вариация на тему известного "покера". Тем, кто знает, что это такое, наверное ничего объяснять не надо, для тех же, кто с ним не знаком, поясним, что бросая пять игровых костей, нужно стараться создать на них одну из комбинаций выпавших очков. Для того, чтобы это было возможно, после первого броска разрешается перебросить те кости, которые Вас не устраивают. О том, что это за комбинации и как они оцениваются, мы говорить не будем, в листинге программы эта информация есть.

Покер требует в определенной степени умения мыслить комбинаторно, принимать правильные решения. Известно, что против теории вероятности не пойдешь, но тем не менее, есть игроки, которые играют лучше, а есть играющие хуже. Очевидно, первые придерживаются какой-то выигрышной стратегии, а вторые ее еще не выработали.

Данная игра дает двойную возможность для реализации Вашей стратегии. Дело в том, что за набранные комбинации Вы не просто получаете очки, а имеете возможность передвигать по результатам броска фишки на игровом поле. Надо еще хорошо подумать, прежде чем добиваться той или иной комбинации, к чему Вам следует стремиться. Это зависит от конкретной ситуации на игровом поле.

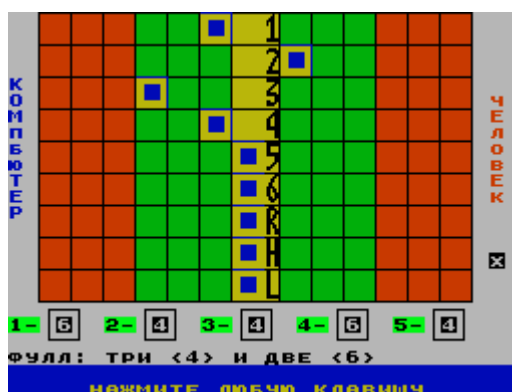
Игра нравится даже тем, кому вообще игровые программы не нравятся, и мы надеемся, что и Вам она доставит много радости.

Хочется сказать несколько слов о проблеме "ошибок в листингах". Это бич всех компьютерных журналов и у нас в "ZX-РЕВЮ" это случалось. Трудоемкость подготовки статьи с игрой для Вас, уважаемые читатели, примерно в три раза выше, чем любой другой аналогичной статьи в связи с необходимостью многочисленных проверок, перепроверок и отладок и, тем не менее, Вы сами знаете, что ошибки проникали в окончательный текст. Это досадно, хотя и не вредно для тех, кто хочет не просто набрать программу, но при этом еще и освоить программирование.

Сейчас мы изменили технологию подготовки программ для самостоятельного набора и ошибки практически исключены. Можете смело набирать и запускать программу. Если что-то не работает, проверяйте правильность ввода.

Наша технология такова: сначала программа набирается, отлаживается, русифицируется на "Спектруме". Когда она полностью работоспособна, она "сбрасывается" на диск (TR-DOS), после чего программными средствами перекодируется "символика" и конвертируется формат записи в MSDOS. В виде файла MSDOS она верстается и распечатывается на IBM-компьютере. Поскольку все операции автоматические, то листинг, который Вы видите перед собой, точно соответствует тому, что было в памяти "Спектрума", когда программа была отлажена и работала, и ошибок здесь просто не должно быть.

Несколько слов о русификации. Мы придерживаемся ранее принятого нами способа русификации. Если Вы с ним не знакомы, Вам понадобятся выпуски ZX-РЕВЮ N3, 4 за этот год и, может быть, N 1,2 тоже за этот год.



```

1 GO TO 3
2 CLEAR 64599: LOAD "chr"CODE 64600
3 BORDER 1: PAPER 1: CLS : POKE 23658,8: POKE 23606,88: POKE 23607,251:
4 PRINT AT 10,5; INK 3; BRIGHT 1;"ВАМ НУЖНА ИНСТРУКЦИЯ?";AT 12,6;"НАЖМИТЕ <Y> ИЛИ <N>": BEEP
   .5,3: BEEP .35,0
5 IF INKEY$="" THEN GO TO 5
6 IF INKEY$="Y" THEN GO SUB 9100: GO TO 8
7 GO SUB 9270
8 CLS : BORDER 1: INK 0: PAPER 7
9 PRINT AT 10,0; FLASH 1; INK 1; PAPER 6; "СЕЙЧАС Я ПОДГОТОВЛЮ ИГРОВОЕ ПОЛЕ"
10 GO SUB 9900: REM U.D.G. ====
15 RANDOMIZE
20 DIM T(5):DIM R(5):DIM Z$(6) :LET T$=CHR$ 132+CHR$ 136:LET B$ =CHR$ 129+CHR$ 130
30 DIM D(5,2,1): FOR I=1 TO 5: FOR N=1 TO 2: READ P: LET D(I,N ,1)=P: NEXT N: NEXT I
34 REM ПОДГОТОВКА КООРДИНАТ ИГРОВОГО ПОЛЯ И ИЗОБРАЖЕНИЯ ФИШЕК=
35 DIM Y(9): DIM X(13)
40 FOR I=0 TO 8: LET Y(I+1)=2*I: NEXT I
50 FOR I=1 TO 7: LET X(I)=2*I: NEXT I: FOR I=8 TO 13: LET X(I) =2*I+1: NEXT I
60 LET E1=0: LET E2=0: DIM F(9): FOR I=1 TO 9: LET F(I)=7: NEXT I
65 CLS : IF PLAY=2 THEN LET GO=1: PRINT AT 10,6; INK 1;"ПЕРВЫЙ ИГРОК НАЧИНАЕТ": PAUSE 75
66 IF PLAY=1 AND GO=1 THEN PRINT AT 10,7; INK 1;"КОМПЬЮТЕР НАЧИНАЕТ": PAUSE 75
67 IF PLAY=1 AND GO=-1 THEN PRINT AT 10,4; INK 1;"О.К. - ВЫ ХОДИТЕ ПЕРВЫМ": PAUSE 75
70 GO SUB 9000: REM ИГРОВОЕ ПОЛЕ =====
80 GO SUB 2000: REM ИЗОБРАЖЕНИЕ КОСТЕЙ И ФИШЕК =====
90 GO SUB 4500: REM ПРОВЕРКА НА ПОБЕДУ =====
100 LET GO=GO*-1
200 PRINT AT 21,0; INK 1; PAPER 6; "                ПЕРВЫЙ БРОСОК                "
202 FOR N=1 TO 10: FOR I=1 TO 5: LET R(I)=1+INT (RND*6): NEXT I
205 BEEP .01,INT (RND*10)
210 GO SUB 3500: NEXT N
230 IF PLAY=2 OR GO=1 THEN GO SUB 3000
231 IF PLAY=1 AND GO=-1 THEN GO SUB 3600: GO SUB 5000: GO SUB 5500
234 IF M=0 THEN GO TO 310
235 PRINT AT 21,0; INK 1; PAPER 6;"                ВТОРОЙ БРОСОК                "
236 FOR N=1 TO 10
240 GO SUB 2400
250 GO SUB 3500
255 NEXT N
270 IF PLAY=2 OR GO=1 THEN GO SUB 3000
271 IF PLAY=1 AND GO=-1 THEN GO SUB 3600: GO SUB 5000: GO SUB 5500
274 IF M=0 THEN GO TO 310
275 PRINT AT 21,0; INK 1; PAPER 6;"                ТРЕТИЙ БРОСОК                "
276 FOR N=1 TO 10
280 GO SUB 2400
290 GO SUB 3500
295 NEXT N
310 GO SUB 3600
315 IF PLAY=1 AND GO=-1 THEN GO SUB 6300
320 PRINT AT 21,0;X$
330 PRINT #1; INK 6; "                НАЖМИТЕ ЛЮБУЮ КЛАВИШУ                "
340 PAUSE 0: INPUT ;
350 IF GO=-1 THEN GO TO 370
360 IF PLAY=2 THEN PRINT INK 1; AT 21,0;"                БРОСАЕТ ПЕРВЫЙ ИГРОК                "
361 IF PLAY=1 THEN PRINT INK 1; AT 21,0;"                БРОСАЕТ КОМПЬЮТЕР                "
365 GO TO 70
370 IF PLAY=2 THEN PRINT INK 2; AT 21,0;"                БРОСАЕТ ВТОРОЙ ИГРОК                "
371 IF PLAY=1 THEN PRINT INK 2; AT 21,0;"                БРОСАЕТ ЧЕЛОВЕК                "
380 GO TO 70
2000 REM ИЗОБРАЖЕНИЕ КОСТЕЙ И ФИШЕК
2010 FOR I=1 TO 5: PLOT D(I,1,1),D(I,2,1) : DRAW 15,0: DRAW 0,-15: DRAW -15,0: DRAW 0,15:
   PRINT BRIGHT 1; INK 0; PAPER 4;AT 19,6*(I-1);I;"-": NEXT I
2020 FOR I=1 TO 9
2025 IF I=E1 OR I=E2 THEN NEXT I
2026 IF I=10 THEN GO TO 2040
2030 PRINT OVER 1; INK 1; PAPER 6;AT Y(I),X(F(I));T$;AT Y(I)+1, X(F(I));B$: NEXT I

```

```

2040 IF NOT E1 AND NOT E2 THEN RETURN
2050 LET DISP1=DISP1*G0: LET DISP2=DISP2*G0
2060 LET F(E1)=F(E1)+DISP1
2070 IF F(E1)<1 THEN LET F(E1)=1
2080 IF F(E1)>13 THEN LET F(E1)=13
2090 PRINT OVER 1; INK 1; PAPER 6; AT Y(E1),X(F(E1));T$;AT Y(E1)+1,X(F(E1));B$
2100 IF NOT E2 THEN RETURN
2110 LET F(E2)=F(E2)+DISP2
2120 IF F(E2)<1 THEN LET F(E2)=1
2130 IF F(E2)>13 THEN LET F(E2)=13
2140 PRINT OVER 1; INK 1; PAPER 6;AT Y(E2),X(F(E2));T$;AT Y(E2)+1,X(F(E2));B$
2150 RETURN
2400 REM СБРОС ЗНАЧЕНИИ КОСТЕЙ==
2410 FOR I=1 TO M: LET R(T(I))=1+INT (RND*6): NEXT I
2420 BEEP .01,INT (RND*10)
2430 RETURN
2999 REM ПОВТОРНЫЙ БРОСОК =====
3000 LET R$="": INPUT AT 0,0;"ВВЕДИ Y ЕСЛИ О.К. ИЛИ N ЕСЛИ НЕТ";LINE R$
3010 IF R$<>"Y" AND R$<>"N" THEN GO TO 3000
3020 IF R$="Y" THEN GO TO 310
3030 INPUT "СКОЛЬКО КОСТЕЙ ПЕРЕБРОСИТЕ? ";M
3040 LET M=INT M: IF M<1 OR M>5 THEN GO TO 3030
3045 PRINT #1; FLASH 1; INK 2; PAPER 7;"БУДУТ ПЕРЕБРАСЫВАТЬСЯ КОСТИ ... ": PAUSE 50
3050 FOR I=1 TO M: INPUT "КОСТЬ НОМЕР : ";T(I): LET T(I)=INT T(I)
3055 IF T(I)<1 OR T(I)>5 THEN GO TO 3050
3060 NEXT I: RETURN
3499 REM ПЕЧАТЬ ЗНАЧЕНИЙ КОСТЕЙ=
3500 FOR I=1 TO 5: PRINT AT 19,6*I-3; INK 7; PAPER 0;R(I): NEXT I
3510 RETURN
3599 REM СОРТИРОВКА ЗНАЧЕНИЙ КОСТЕЙ =====
3600 FOR N=1 TO 4: FOR I=1 TO 5-N
3610 IF R(I)<R(I+1) THEN LET A=R(I): LET R(I)=R(I+1): LET R(I+1)=A
3620 NEXT I: NEXT N
3629 REM HOLD HOW MANY OF EACH==
3630 DIM V(5,2): LET COUNT=1: LET A=0
3640 FOR N=1 TO 5
3650 IF N=5 THEN GO TO 3700
3660 FOR I=N TO 4
3670 IF R(I)=R(I+1) THEN LET COUNT=COUNT+1: GO TO 3690
3680 GO TO 3700
3690 NEXT I
3700 LET A=A+1: LET V(A,2)=COUNT: LET V(A,1)=R(N)
3710 LET N=N+COUNT-1: LET COUNT=1
3720 NEXT N
3769 REM НАЗНАЧАЕМ ПЕРЕМЕННОЙ X$ КОМБИНАЦИЮ КОСТЕЙ И УСТАНОВЛИВАЕМ СЧЕТЧИКИ ПЕРЕМЕЩЕНИЯ
      ФИШЕК ==
3790 DIM X$(32): LET E1=0: LET E2=0: LET DISP1=0: LET DISP2=0
3800 IF V(1,2)<>5 THEN GO TO 3820
3801 REM ПЯТЕРКИ =====
3802 LET X$="ПЯТЕРКА <"+STR$ V(1,1)+"> !!!"
3804 LET DISP1=V(1,2)-1: LET E1=V(1,1): GO TO 3930
3820 IF V(1,2)<>4 AND V(2,2)<>4 THEN GO TO 3830
3821 REM ЧЕТВЕРКИ =====
3822 LET DISP1=3: IF V(1,2)=4 THEN LET E1=V(1,1)
3823 IF V(2,2)=4 THEN LET E1=V(2,1)
3824 LET X$="ЧЕТВЕРКА <"+(STR$ V(1,1) AND V(1,2)=4)+(STR$ V(2,1) AND V(2,2)=4)+">": GO TO
      3930
3830 IF V(1,2)<>3 AND V(1,2)<>2 OR V(2,2)<>3 AND V(2,2)<>2 OR V(1,2)=V(2,2) THEN GO TO 3850
3834 REM ФУЛЛ =====
3835 LET X$="ФУЛЛ: ТРИ <"+(STR$ V(1,1) AND V(1,2)=3)+(STR$ V(2,1) AND V(2,2)=3)+"> И ДВЕ
      <"+(STR$ V(1,1) AND V(1,2)=2)+(STR$ V(2,1) AND V(2,2)=2)+">"
3838 FOR I=1 TO 2
3840 IF V(I,2)=3 THEN LET DISP1= V(I,2)-1: LET E1=V(I,1)
3843 IF V(I,2)=2 THEN LET DISP2= V(I,2)-1: LET E2=V(I,1)
3845 NEXT I: GO TO 3930

```

```

3849 REM ТРОЙКИ =====
3850 FOR I=1 TO 3: IF V(I,2)=3 THEN LET DISP1=V(I,2)-1: LET E1=V(I,1): LET X$="ТРОЙКА
    <" + STR$ V(I,1) + ">"
3851 NEXT I
3854 REM ПАРА ИЛИ ДВЕ ПАРЫ =====
3855 LET A=0: FOR I=1 TO 5: IF V(I,2)=2 THEN LET A = A+1: LET B=I
3856 NEXT I
3860 IF A=0 THEN GO TO 3930
3870 IF A=2 THEN GO TO 3900
3880 LET E1=V(B,1): LET DISP1=V(B,2)-1
3895 LET X$="ПАРА <" + STR$ V(B,1) + ">": GO TO 3930
3900 LET A=0: FOR I=1 TO 5
3905 IF V(I,2)=2 AND NOT A THEN LET A=I
3910 IF V(I,2)=2 AND A THEN LET B=I
3915 NEXT I
3920 LET X$="ДВЕ ПАРЫ, <" + STR$ V(A,1) + "> И <" + STR$ V(B,1) + ">"
3925 LET DISP1=V(A,2)-1: LET E1=V(A,1): LET DISP2=V(B,2)-1: LET E2=V(B,1)
3929 REM ИНИЦИАЛИЗАЦИЯ СУММЫ И МАРКЕРОВ БОЛЬШОЙ И МАЛОЙ СУММЫ==
3930 LET TOTAL=0: LET HIGH=0: LET LOW=0: FOR I=1 TO 5
3940 LET TOTAL=TOTAL+R(I): NEXT I
3950 IF TOTAL>12 AND TOTAL<23 THEN GO TO 3995
3955 IF TOTAL>25 THEN LET HIGH=3
3960 IF TOTAL>22 AND TOTAL<26 THEN LET HIGH=2
3965 IF TOTAL<10 THEN LET LOW=3
3970 IF TOTAL>9 AND TOTAL<13 THEN LET LOW=2
3975 IF CODE X$(1)<>32 THEN GO TO 3985
3979 REM БОЛЬШАЯ ИЛИ МАЛАЯ СУММА
3980 GO SUB 4010: RETURN
3984 REM ВЫБОР КОМБИНАЦИИ =====
3985 IF PLAY=1 AND GO=-1 THEN GO TO 3994
3986 PRINT AT 21,0;"ВЫБИРАЙТЕ, НА ЧТО ВЫ ИДЕТЕ..... ": PAUSE 100 : PRINT AT 21,0;"СУММА /
    ";X$(1 TO 22);"?
3990 INPUT "S" ЕСЛИ СУММА, ИНАЧЕ "X": "; LINE R$
3991 IF R$<>"S" AND R$<>"X" THEN GO TO 3990
3993 IF R$="S" THEN GO SUB 4010
3994 REM СТРИТ =====
3995 FOR I=1 TO 4: IF R(I)<>R(I+1)+1 THEN GO TO 3998
3996 NEXT I
3997 LET DISP1=(3 AND R(1)=6)+(2 AND R(1)=5): LET E1=7: LET X$=("БОЛЬШОЙ" AND
    R(1)=6)+("МАЛЫЙ" AND R(1)=5)+" СТРИТ"
3998 IF CODE X$(1)=32 THEN LET X$="          Н И Ч Е Г О !"
4000 RETURN
4010 LET X$=("БОЛЬШАЯ СУММА =" AND HIGH)+("МАЛАЯ СУММА =" AND LOW)+STR$ TOTAL
4020 LET DISP1=(HIGH AND HIGH)+(LOW AND LOW): LET DISP2=0
4030 LET E1=(8 AND HIGH)+(9 AND LOW): LET E2=0
4040 RETURN
4500 REM ПРОВЕРКА ВЫИГРЫША =====
4510 LET A=0: LET B=0
4520 FOR I=1 TO 9
4530 IF F(I)<4 THEN LET A=A+1
4540 IF F(I)>10 THEN LET B=B+1
4550 NEXT I
4560 IF A>2 OR B>2 THEN GO TO 4575
4570 RETURN
4575 FOR I=1 TO 30
4580 IF PLAY=2 THEN PRINT AT 21,5; FLASH 1; BRIGHT 1; INK 1; PAPER 6; "ПОБЕДИЛ "+("ПЕРВЫЙ"
    AND A>2)+("ВТОРОЙ" AND B>2)+" ИГРОК"
4581 IF PLAY=1 THEN PRINT AT 21,5; FLASH 1; BRIGHT 1; INK 1; PAPER 6; ("ПОБЕДИЛ
    КОМПЬЮТЕР!!!" AND A>2)+("ВЫ ВЫИГРАЛИ! ПОЗДРАВЛЯЮ." AND B>2)
4585 BEEP .01,I: NEXT I
4590 LET R$="": INPUT "СЫГРАЕМ ЕЩЕ? ('Y' ИЛИ 'N') "; LINE R$
4600 IF R$="" THEN GO TO 4590
4610 IF R$="Y" THEN GO TO 4660
4620 LET R$="": INPUT "КОНЕЦ? ('Y' ИЛИ 'N') "; LINE R$
4630 IF R$="" THEN GO TO 4620

```

```

4640 IF R$="Y" THEN GO TO 9998
4650 GO TO 4590
4660 CLS : PRINT ""ВЫ МОЖЕТЕ ИЗ
МЕНИТЬ РЕЖИМ ИГРЫ.""СНАЧАЛА ВВ
ЕДИТЕ ""1"" ИЛИ ""2""""ПО ЧИСЛ
У ИГРАЮЩИХ; ЗАТЕМ""(ЕСЛИ ВЫ ИГ
РАЕТЕ С КОМПЬЮТЕРОМ)""ВВЕДИТЕ
""-1"" ЕСЛИ ХОТИТЕ ХОДИТЬ""ПЕР
ВЫЙ. ИНАЧЕ ВВЕДИТЕ ""1""
4670 INPUT "ОДИН ИЛИ ДВА-ИГРОКА?"; PLAY
4680 LET PLAY=INT PLAY: IF PLAY<1 OR PLAY>2 THEN GO TO 4670
4690 IF PLAY=2 THEN GO TO 60
4700 INPUT " -1, ЕСЛИ ХОДИТЕ ПЕРВЫМ, ИНАЧЕ 1:"; GO
4710 LET GO=INT GO: IF GO=0 OR GO>1 OR GO<-1 THEN GO TO 4700
4720 GO TO 60
4999 REM УСТАНОВКА МАРКЕРОВ ПРИОРИТЕТОВ =====
5000 DIM P(9): LET A1=0: LET A2=0
5010 FOR I=1 TO 9: REM ЕСТЬ ЛИ ФИШКИ В КРАСНОЙ ЗОНЕ? =====
5020 IF F(I)<4 THEN LET A1=A1+1
5030 IF F(I)>10 THEN LET A2=A2+1
5040 NEXT I
5050 IF A1<>2 OR A2<>2 THEN GO TO 5210
5060 LET B1=7
5070 FOR I=1 TO 9
5080 IF F(I)<B1 AND F(I)>3 THEN LET B1=F(I): LET B2=I
5090 NEXT I
5100 IF B1=7 THEN GO TO 5130
5110 IF B2<7 THEN LET P(B2)=1: RETURN
5120 LET P(B2)=B2: RETURN
5130 LET B1=14
5140 FOR I=1 TO 6
5150 IF F(I)<B1 AND F(I)>10 THEN LET B1=F(I): LET B2=I
5160 NEXT I: IF B1=14 THEN GO TO 5180
5170 LET P(B2)=1: RETURN
5180 FOR I=7 TO 9
5190 IF F(I)<B1 AND F(I)>10 THEN LET B1=F(I): LET B2=I
5200 NEXT I: LET P(B2)=B2: RETURN
5210 IF A1<>2 THEN GO TO 5280
5220 LET B1=14
5230 FOR I=1 TO 9
5240 IF F(I)<B1 AND F(I)>3 THEN LET B1=F(I): LET B2=I
5250 NEXT I
5260 IF B2>=7 THEN LET P(B2)=B2: RETURN
5270 LET P(B2)=1: RETURN
5280 IF A2<>2 THEN GO TO 5380
5290 LET A2=0: FOR I=7 TO 9
5300 IF F(I)>10 THEN LET A2=A2+1
5310 NEXT I: IF A2<2 THEN GO TO 5350
5320 FOR I=7 TO 9
5330 IF F(I)>10 THEN LET P(I) = 1
5340 NEXT I: RETURN
5350 LET B1=14: FOR I=1 TO 6
5360 IF F(I)>10 AND F(I)<B1 THEN LET B1=F(I): LET B2=I
5370 NEXT I: LET P(B2)=1: RETURN
5380 FOR I=1 TO 9
5390 IF F(I)=1 THEN LET P(I)=3
5400 IF F(I)=2 OR F(I)=3 THEN LET P(I)=2
5410 NEXT I: RETURN
5499 REM РАСЧЕТ ПРИОРИТЕТА=1 ===
5500 LET A=0: LET M=0: FOR I=1 TO 6
5510 IF P(I)=1 THEN LET A=I
5520 NEXT I: IF A=0 THEN GO TO 5570
5530 FOR I=1 TO 5
5540 IF R(I)<>A THEN GO SUB 6210
5550 NEXT I: IF M=0 THEN GO TO 310

```

```

5560 RETURN
5570 IF P(7)>I THEN GO TO 5640: REM НЕТ СМЫСЛА ДЕЛАТЬ СТРИТ ===
5580 IF E1=7 THEN GO TO 310
5590 IF E1=0 AND E2=0 THEN LET M=1: LET T(1)=5: RETURN
5600 IF P(7)<>7 THEN GO TO 5640 : REM СТРИТ НЕ РЕАЛЕН =====
5610 FOR I=1 TO 4
5620 IF R(I)=E1 THEN LET M=1: LET T(1)=1: RETURN
5630 NEXT I
5640 IF P(8)<>8 OR TOTAL<20 THEN GO TO 5720: REM БОЛЬШАЯ СУММА СОМНИТЕЛЬНА
=====
5650 IF TOTAL>23 THEN GO TO 5690
5660 FOR I=3 TO 5
5670 IF R(I)<4 THEN GO SUB 6210
5680 RETURN
5690 FOR I=4 TO 5
5700 IF R(I)<3 THEN GO SUB 6210
5710 RETURN
5720 IF P(9)<>9 OR TOTAL>15 THEN GO TO 5800: REM МАЛАЯ СУММА СОМНИТЕЛЬНА
=====
5730 IF TOTAL<12 THEN GO TO 5770
5740 FOR I=1 TO 3
5750 IF R(I)>3 THEN GO SUB 6210
5760 NEXT I: RETURN
5770 FOR I=1 TO 2
5780 IF R(I)>2 THEN GO SUB 6210
5790 RETURN
5800 IF DISP1=1 AND DISP2=1 THEN GO TO 5860
5810 IF DISP1<>1 THEN GO TO 5980 : REM НЕТ ПАР =====
5820 FOR I=1 TO 5
5830 IF R(I)<>E1 AND P(E1)=0 THEN GO SUB 6210
5840 IF P(E1)>1 THEN LET M=4: IF I>1 THEN LET T(I)=I
5850 NEXT I: RETURN
5860 IF P(E1)>1 AND P(E2)>1 THEN GO TO 5910: REM ДВУХ ПАР НЕТ ==
5870 IF P(E1)=0 AND P(E2)=0 THEN GO TO 5940: REM ПРИНИМАЕТСЯ ДВЕ ПАРЫ И ПЕРЕБРАСЫВАЕТСЯ
НЕПАРНАЯ КОСТЬ =====
5879 REM 2 ПАРЫ, ПЕРЕБРАСЫВАЕТСЯ 1 ПАРА И НЕПАРНАЯ КОСТЬ =====
5880 FOR I=1 TO 5
5890 IF R(I)<>E2 AND P(E2)=0 OR R(I)<>E1 AND P(E1)=0 THEN GO SUB 6210
5900 NEXT I: RETURN
5910 FOR I=1 TO 5
5920 IF R(I)=E1 OR R(I)=E2 THEN GO SUB 6210
5930 NEXT I: RETURN
5940 FOR I=1 TO 5
5950 IF R(I)<>E1 AND R(I)<>E2 THEN GO SUB 6210
5970 NEXT I: RETURN
5980 IF DISP1<>2 AND DISP2<>2 THEN GO TO 6110: REM НЕ ТРОЙКА ===
5990 IF DISP1=1 OR DISP2=1 THEN GO TO 6050
6000 FOR I=1 TO 5
6005 IF E1=0 THEN GO TO 6025
6010 IF DISP1=2 AND F(E1)>5 AND R(I)>E1 THEN GO SUB 6216
6015 IF P(E1)=3 AND R(I)=E1 THEN GO SUB 6210
6020 IF F(E1)<6 AND P(E1)<3 AND R(I)<>E1 THEN LET M=1: LET T(1) = I
6025 IF E2=0 THEN GO TO 6045
6030 IF DISP2=2 AND F(E2)>5 AND R(I)<>E2 THEN GO SUB 6210
6035 IF P(E2)=3 AND R(I)=E2 THEN GO SUB 6210
6040 IF F(E2)<6 AND P(E2)<3 AND R(I)<>E2 THEN LET M=1: LET T(1)=I
6045 NEXT I: RETURN
6049 REM ФУЛЛ =====
6050 IF E1=0 AND E2 = 0 THEN GO TO 310
6060 FOR I=1 TO 5
6070 IF P(E1)>1 AND P(E2)>1 THEN GO SUB 6210
6080 IF P(E1)>1 AND P(E2)=0 AND R(I)<>E2 THEN GO SUB 6210
6090 IF P(E1)=0 AND P(E2)>1 AND R(I)<>E1 THEN GO SUB 6210
6100 NEXT I: RETURN
6110 IF DISP1<>3 AND DISP2<>3 OR E1>6 THEN GO TO 6150: REM НЕ ЧЕТВЕРКА =====

```

```

6120 FOR I=1 TO 5
6130 IF R(I)<>E1 AND DISP1=3 OR R(I)<>E2 AND DISP2=3 THEN GO SUB 6210
6140 NEXT I: RETURN
6150 IF HIGH=3 OR LOW=3 OR DISP1=4 OR E1=7 AND P(7)=0 THEN GO TO 310: REM БЕЗ ПЕРЕБРОСКИ
      =====
6160 IF HIGH=2 AND P(8)=0 THEN LET M=1: LET T(1)=5: RETURN
6170 IF LOW=2 AND P(9)=0 THEN LET M=1: LET T(1)=1: RETURN
6180 LET M=5: FOR I=1 TO 5
6190 LET T(I)=I: NEXT I: RETURN
6210 LET M=M+1: LET T(M)=I
6220 RETURN
6300 IF NOT HIGH AND NOT LOW THEN RETURN
6310 IF F(8)<4 AND HIGH OR F(9)<4 AND LOW THEN RETURN
6320 IF DISP1=2 AND DISP2=1 AND (F(8)-HIGH)>3 AND HIGH THEN RETURN
6321 IF DISP1=2 AND DISP2=1 AND (F(9)-LOW)>3 AND LOW THEN RETURN
6322 IF DISP1=1 AND DISP2=2 AND (F(8)-HIGH)>3 AND HIGH THEN RETURN
6323 IF DISP1=1 AND DISP2=2 AND (F(9)-LOW)>3 AND LOW THEN RETURN
6330 IF DISP1>HIGH AND HIGH OR DISP1>LOW AND LOW THEN RETURN
6340 IF (F(E1)-DISP1)<4 THEN RETURN
6344 IF E2=0 THEN GO TO 6350
6345 IF (F(E2)-DISP2)<4 THEN RETURN
6350 IF F(E1)>10 AND F(8)<10 AND HIGH THEN RETURN
6354 IF E2=0 THEN GO TO 6360
6355 IF F(E2)>10 AND F(8)<10 AND HIGH THEN RETURN
6360 IF F(E1)>10 AND F(9)<10 AND LOW THEN RETURN
6364 IF E2=0 THEN GO TO 6370
6365 IF F(E2)>10 AND F(9)<10 AND LOW THEN RETURN
6370 GO SUB 4010: RETURN
8999 REM ГРАФИКА = = = = =
9000 FOR I=0 TO 17: PRINT PAPER 6; AT I,14; " ": NEXT I
9005 FOR I=0 TO 17: PRINT PAPER 6; AT I,16;CHR$(144+I): NEXT I
9010 FOR I=0 TO 17: PRINT PAPER 2;AT I,2;Z$;AT I,23;Z$: NEXT I
9020 FOR I=0 TO 17: PRINT PAPER 4;AT I,8;Z$;AT I,17;Z$: NEXT I
9040 PLOT 15,31: FOR I=1 TO 4: DRAW 216,0: DRAW 0,16: DRAW -216, 0: DRAW 0,16: NEXT I: DRAW
      216,0 : DRAW 0,16: DRAW -216,0
9050 FOR I=2 TO 14 STEP 2: PLOT I*8-1,175: DRAW 0,-143: NEXT I:FOR I=17 TO 29 STEP 2: PLOT
      I*8-1,175: DRAW 0,-143: NEXT I
9055 IF PLAY=1 THEN GO TO 9070
9060 PRINT INK 1;AT 2,0;"П";AT 3,0;"Е";AT 4,0;"Р";AT 5,0;"В";AT 6,0;"Ы";AT 7,0; "Й";AT
      8,0;" ":AT 9,0;"И";AT 10,0; "Г"; AT 11,0;"Р";AT 12,0;"О"; AT 13,0; "К"
9065 PRINT INK 2; AT 2,30;"В";AT 3,30;"Т";AT 4,30;"О";AT 5,30;"Р";AT 6,30;"О";AT
      7,30;"Й";AT 8,30;" ":AT 9, 30; "Й"; AT 10,30; "Г"; AT 11,30; "Р"; AT 12,30; "О"; AT
      13,30;"К": GO TO 9080
9070 PRINT INK 1;AT 4,0;"К";AT 5,0;"О";AT 6,0;"М";AT 7,0;"П";AT 8,0;"б";AT 9,0;"Ю";AT
      10,0;"Т"; AT 11,0; "Е";AT 12,0; "Р"
9075 PRINT INK 2; AT 5,30;"Ч";AT 6,30; "Е"; AT 7,30; "Л";AT 8,30;"О";AT 9,30;"В";AT
      10,30;"Е";AT 11,30;"К"
9080 IF GO=1 THEN PRINT AT 15,0; INK 7; PAPER 0; FLASH 1;"X";AT 15,30; PAPER 7; " "
9085 IF GO=1 THEN PRINT AT 15,0; PAPER 7;" ":AT 15,30; INK 7;PAPER 0; FLASH 1;"X"
9090 RETURN
9100 REM ИНСТРУКЦИЯ =====
9110 BORDER 1: INK 6: PAPER 1: PRINT : CLS
9120 PRINT AT 3,0;
      " ЦЕЛЬ ЭТОЙ ИГРЫ СОСТОИТ В ТОМ,
      ЧТОБЫ ПРОВЕСТИ ТРИ ФИШКИ В КРАС-
      НУЮ ЗОНУ НА СВОЕЙ ПОЛОВИНЕ ПОЛЯ
      ДО ТОГО КАК ЭТО СДЕЛАЕТ ВАШ СО-
      ПЕРНИК."
9130 PRINT '
      ИГРА ВЕДЕТСЯ ДЕВЯТЬЮ ФИШКАМИ.
      СНАЧАЛА ОНИ РАСПОЛОЖЕНЫ В ЦЕНТРЕ
      ПОЛЯ НА ДОРОЖКАХ, ОБОЗНАЧЕННЫХ
      ""1"", ""2"",...""6"", ""R"", ""
      H"", ""L"".
      " ЗА ОДИН ХОД ВЫ МОЖЕТЕ ПЕРЕ-

```

ДВИГАТЬ ДО ДВУХ ФИШЕК. ВЕЛИЧИНА
ХОДА ОПРЕДЕЛЯЕТСЯ РЕЗУЛЬТАТОМ
БРОСАНИЯ ПЯТИ КОСТЕЙ."

9140 GO SUB 9300

9150 PRINT AT 5,0;

" ПОСЛЕ ПЕРВОГО БРОСКА ВЫ МОЖЕ-
ТЕ ДВАЖДЫ ПЕРЕБРОСИТЬ СТОЛЬКО
КОСТЕЙ, СКОЛЬКО ХОТИТЕ. ПРИ ВТО-
РОМ И ТРЕТЬЕМ БРОСКАХ РАЗРЕШАЕТ-
СЯ ИСПОЛЬЗОВАТЬ ЛЮБЫЕ КОСТИ."

9160 PRINT "

" ВАША ЦЕЛЬ ПРИ ПЕРЕБРОСКЕ КОС-
ТЕЙ - СОСТАВИТЬ ТАКУЮ КОМБИНА-
ЦИЮ, КОТОРАЯ ДАСТ ВАМ ВОЗМОЖ-
НОСТЬ ЛУЧШЕГО ХОДА ФИШКАМИ."

9170 GO SUB 9300

9180 PRINT AT 1,0;

" ПРАВО НА ХОД ДАЮТ СЛЕДУЮЩИЕ
КОМБИНАЦИИ:"

"- ДВЕ ИЛИ БОЛЬШЕ КОСТЕЙ ОДИНАКО-
ВОГО ЗНАЧЕНИЯ 1...6 (ПЕРЕДВИ-
ГАЮТСЯ ФИШКИ НА ДОРОЖКЕ С СО-
ОТВЕТСТВУЮЩИМ НОМЕРОМ);"

"- ""СТРИТ"" - ПОСЛЕДОВАТЕЛЬНОСТЬ
ИЗ ПЯТИ ИДУЩИХ ПОДРЯД ЗНАЧЕНИЙ
КОСТЕЙ (ПЕРЕДВИГАЕТСЯ ФИШКА ПО
ДОРОЖКЕ ""R"" - ""RUN"");"

9190 PRINT "

"- БОЛЬШАЯ СУММА КОСТЕЙ - БОЛЬШЕ
22 (ПЕРЕДВИГАЕТСЯ ФИШКА ПО ДО-
РОЖКЕ ""H"" - ""HIGH"");"

"- МАЛАЯ СУММА - СУММА ВСЕХ КОС-
ТЕЙ МЕНЬШЕ 13 (ПЕРЕДВИГАЕТСЯ
ФИШКА ПО ДОРОЖКЕ ""L"" - ""LOW"")."

9200 GO SUB 9300

9205 PRINT AT 2,0;

" КОМБИНАЦИИ ПОЗВОЛЯЮТ СДЕЛАТЬ
ХОД НА ОПРЕДЕЛЕННУЮ ВЕЛИЧИНУ:"

9210 PRINT TAB 17;"ПАРА - 1 ШАГ" TAB 15;"ТРОЙКА - 2 ШАГА" TAB 13;"ЧЕТВЕРКА - 3 ШАГА" TAB 14;
"ПЯТЕРКА - 4 ШАГА" TAB 2;"БОЛЬШОЙ СТРИТ (2-6) - 3 ШАГА"

9220 PRINT TAB 4;"МАЛЫЙ СТРИТ (1-5) - 2 ШАГА" TAB 2;"БОЛЬШАЯ СУММА (>25) - 3 ШАГА" TAB 14;"БОЛЬШАЯ
СУММА (23-25) - 2 ШАГА" TAB 4;"МАЛАЯ СУММА (<10) - 3 ШАГА" TAB 2;"МАЛАЯ СУММА (10-12)
- 2 ШАГА"

" ЗАМЕЬТЕ, ЧТО ЕСЛИ У ВАС ЕСТЬ
ДВЕ ПАРЫ, ИЛИ ФУЛЛ (ПАРА+ТРОЙКА)
ТО ВЫ ИМЕЕТЕ ПРАВО ПЕРЕДВИНУТЬ
ДВЕ ФИШКИ."

9230 GO SUB 9300

9240 PRINT "" TAB 11;"ЗАМЕЧАНИЯ" ;AT 2,11; OVER 1;"_____"

"1) ВЫ НЕ ИМЕЕТЕ ПРАВА ХОДИТЬ ОД-
НОВРЕМЕННО ПО ДОРОЖКАМ ""H"" (""L"") И ""1""...""6""."

"2) ФИШКА НЕ МОЖЕТ ВЫХОДИТЬ ЗА
ВНЕШНЮЮ ГРАНИЦУ КРАСНОЙ ЗОНЫ
- ПРОПУЩЕННЫЙ ХОД ПРОПАДАЕТ."

9245 PRINT "

"3) ИГРОК, ДЕЛАЮЩИЙ ХОД, ОТМЕЧА-
ЕТСЯ КУРСОРОМ "; FLASH 1; INK 7; PAPER 0;"X"; FLASH 0; INK 6; PAPER 1;"."

9250 PRINT "

"4) НОМЕРА КОСТЕЙ ИЗОБРАЖАЮТСЯ
ЧЕРНЫМ ПО ЗЕЛЕНОМУ, НАПРИМЕР:

"" TAB 16; INK 0; PAPER 4;"5-":

GO SUB 9300

9255 CLS : PRINT AT 5,0;

" В НАЧАЛЕ ИГРЫ ФИШКИ УСТАНОВ-
ЛЕНЫ НА НЕЙТРАЛЬНЫХ ОТМЕТКАХ."

```

ОНИ ИЗОБРАЖАЮТСЯ ТАК:``TAB 15; INK 0; PAPER 6;CHR$ 132;CHR$ 136; PAPER 1 ` TAB 15;
PAPER 6;CHR$ 129;CHR$ 130
9260 PRINT `
"-----
    ЕСЛИ ВЫ ХОТИТЕ ПРОЧИТАТЬ ЭТУ
    ИНСТРУКЦИЮ ЕЩЕ РАЗ, НАЖМИТЕ <I>,
    ИНАЧЕ НАЖМИТЕ ЛЮБУЮ КЛАВИШУ...": GO SUB 9300: IF INKEY$="I" THEN GO TO 9110
9270 BORDER 1: PAPER 1: INK 6: CLS : PRINT AT 10,0;
"    ЕСЛИ ВЫ ИГРАЕТЕ С КОМПЬЮТЕРОМ,
    ВВЕДИТЕ ""1"", А ЕСЛИ С ПАРТНЕРОМ,
    ТО ВВЕДИТЕ ""2"". "
9271 INPUT "ВВЕДИТЕ 1 ИЛИ 2 : ";PLAY
9272 LET PLAY=INT PLAY: IF PLAY>2 OR PLAY<1 THEN GO TO 9271
9275 IF PLAY=2 THEN GO TO 9290
9280 CLS : PRINT AT 7,0;
"    ПОЖАЛУЙСТА ИМЕЙТЕ ВВИДУ, ЧТО
    КОГДА КОМПЬЮТЕР ПРИНИМАЕТ РЕШЕ-
    НИЕ О ТОМ, КАКУЮ КОСТЬ ЕМУ ПЕРЕ-
    БРОСИТЬ, ОН РАСПОЛАГАЕТ КОСТИ В
    ПОРЯДКЕ ИХ ЗНАЧЕНИЙ.````
9285 PRINT ``
"    ЕСЛИ ВЫ ХОТИТЕ ХОДИТЬ ПЕРВЫМ,
    ТО ВЕДИТЕ ""-1"" ИНАЧЕ ВЕДИТЕ ""1""
9286 INPUT "ВВЕДИТЕ 1 ИЛИ -1 : ";GO
9287 LET GO=INT GO: IF GO=0 OR GO>1 OR GO<-1 THEN GO TO 9286
9290 BEEP .25,3: BEEP .15,0: BEEP .15,0: BEEP .25,3: CLS : RETURN
9300 PRINT #1;
"    НАЖМИТЕ ЛЮБУЮ КЛАВИШУ...": PAUSE 0: BEEP .25,3: BEEP .15,0:CLS : RETURN
9900 POKE 23675,88: POKE 23676,255: RESTORE 9900
9901 FOR I=0 TO (18*8-1): READ A
9902 POKE USR "A"+I,A: NEXT I
9904 RETURN
9905 DATA 0,8,24,56,120,88,88,24,24,24,24,24,126,126,0: REM A&B=1
9910 DATA 0,0,60,126,70,2,6,12,12,24,24,48,48,126,126,0: REM C&D=2
9915 DATA 0,126,126,6,12,24,48,96,124,126,6,6,102,124,120,0: REM E&F=3
9920 DATA 0,6,14,30,54,102,102,102,102,127,127,6,6,6,6,0: REM G&H=4
9925 DATA 0,126,126,96,96,96,126,126,6,6,6,12,24,112,96,0: REM I&J=5
9930 DATA 0,6,12,24,24,48,48,120,108,108,102,102,54,60,24,0: REM K&L=6
9935 DATA 0,56,124,102,102,108,108,120,112,120,108,108,102,102,102,0: REM M&N=R
9940 DATA 0,102,102,102,102,102,102,126,126,102,102,102,102,102,102,0: REM O&P=H
9945 DATA 0,96,96,96,96,96,96,96,96,96,96,126,126,0: REM Q&R=L
9950 DATA 20,27,68,27,116,27,164,27,212,27: REM DICE PLOT CO-ORDS
9980 SAVE "DICEY" LINE 2: SAVE "chr" CODE 64600,768: VERIFY "DICEY": VERIFY "chr" CODE: GO
    TO 9998
9990 RANDOMIZE USR 15619: REM :ERASE "DICEY"
9992 RANDOMIZE USR 15619: REM :SAVE "DICEY" LINE 2
9994 LET ERR=USR 15619: REM : VERIFY "DICEY"
9996 IF ERR<>0 THEN PRINT FLASH 1; "ERROR!": BEEP 1,0: STOP
9998 BEEP .1,26: BEEP .1,20
9999 BORDER 7: PAPER 7: INK 0: BRIGHT 0

```

Автостарт программы происходит со строки 2, в которой загружается символьный набор.

В строке 3 происходит переключение курсора с [L] на [C] для упрощения опроса клавишей, а также переключение на загруженный русско-латинский символьный набор.

В строках 4...6 организован вывод инструкции при нажатии клавиши "Y". Инструкция выводится при помощи подпрограммы в строках с 9100. Прочитайте ее, Вам понятнее будут многие моменты при отладке программы.

После этого при помощи подпрограммы 9270 происходит задание числа играющих. Возможна игра "ЧЕЛОВЕК-КОМПЬЮТЕР" или "ИГРОК 1-ИГРОК 2". Это определяет переменная PLAY, соответственно имеющая значение 1 или 2. В том случае, если Вы играете с компьютером, можно предоставить право первого хода Вам или компьютеру. Это

определяется переменной GO в строке 9286. В том случае, если играют два партнера, первым всегда ходит игрок 1.

В строках 8...10 происходит задание UDG-графики. Это выполняет подпрограмма со строк 9100.

Со строки 15 начинается инициализация переменных, участвующих в программе. Переменные B\$ и B\$ в строке 20 определяют изображение фишек на игровом поле.

Основной цикл игры представлен в строках 70...380. Остальные строки программы (с 2000) являются различными фрагментами и подпрограммами, необходимыми для игры.

После прорисовки игрового поля в строках 70 и 80 и проверки условия победы в игре (строка 90) происходит передача хода противнику-другому игроку в строке 100.

В строках 200... 210 организован первый бросок. Значение костей изображается при помощи подпрограммы со строки 3500. Далее (строка 230), если бросок выполнил человек, выполняется подпрограмма со строки 3000 - это пауза для оценки результата и принятия решения о том, нужна ли переброска. Если не нужна, то программа делает скачок на строку 310. А если нужна, то следует запрос о перебрасываемых костях: сколько их и какие будут перебрасываться. Далее - возврат из подпрограммы.

В строке 231 если был ход компьютера, им выполняются действия по оценке хода и определению приоритетного варианта при помощи подпрограмм со строк 3600, 5000, 5500. Если в результате выполнения этих проверок определится, что переброска не нужна, то произойдет скачок со строки 234 на строку 310. Иначе программу продолжает группа строк 235...255 это - выполнение второго броска для тех костей, которые требуют переброски.

Строки 270...274 - это копия строк 230...234 - здесь выполняются те же действия по оценке результатов броска.

Аналогично выполняется и третий бросок при помощи строк 275...295, после чего программа продолжается со строки 310. При помощи подпрограммы со строки 3600 происходит выявление результативной комбинации костей, то есть той, которая приводит к выполнению хода игроком.

В строке 320 выявленная результативная комбинация выводится на экран. После нажатия любой клавиши (строки 330,340) происходит передача хода противоположной играющей стороне (строки 350...380), после чего программа за циклируется на строку 70, то есть на повторение всех игровых действий.

Строки 9980 и 9990 служат для самозаписи программы соответственно для магнитофонной и дисковой версии. Не забудьте также о строке 2, которая в случае дисковой версии должна быть следующей:

```
2 CLEAR 64599: RANDOMIZE USR 15619: REM : LOAD "chr"CODE 64600
```

На этом мы прощаемся с Вами, желаем с пользой и интересом провести время за этой игрой, а мы пока готовим крупный сюрприз для ближайших выпусков - программный комплекс для самостоятельного создания адвентюрных игровых программ - Adventure Building System, сокращенно - ABS.

Вы увидите, что с его помощью можно разрабатывать и увлекательные обучающие программы для детей.

Идет длительный процесс по адаптации и отладке этого пакета. Мы надеемся, что к следующему выпуску все работа будут закончены и мы вынесем его для широкого применения.

Искренне Ваш, "ИНФОРКОМ".

Компьютерная новелла

(С) Матвеев Ю. А., 1993

Стратегия капитана Кренона.

(По мотивам программы Rebel Star)

Часть 1. Лунная база "Дельта".

Скоростной космический лайнер "Викинг", следующий рейсом Земля-Венера, вышел на основную трассу в 0.15 по Гринвичу, когда пассажиры уже готовились ко сну. В двухместной каюте номер 29 при тусклом свете ночника возле обзорного экрана, заменявшего иллюминатор, в креслах сидели двое: седой крупный мужчина лет шестидесяти, одетый в серый строгий костюм, и светловолосый юноша в форме курсанта Высшей Космической школы. Оба молча смотрели на желто-серую горбушку удаляющейся Луны.

- Дед, а дед, - тихо прошептал юноша, не отрывая глаз от экрана. - Ты помнишь свою первую высадку на Луну?

- Конечно. Это было еще в конце прошлого века.

- Интересно было наверное, правда?

Бывший командир Группы космических десантников Элмер Кренон медленно перевел взгляд на внука:

- Интересно? Вряд ли... - и, чуть помолчав, добавил: - Скорее страшно.

Юноша внимательно посмотрел на деда и легонько тронул его за рукав:

- Ты обещал рассказать.

Кренон ничего не ответил, а только поднялся с кресла и, неспеша расстегивая пуговицы на пиджаке, тяжело вздохнул.

Ущербная Луна сместила в левую часть экрана и постепенно пряталась за его краем. Кренон снял пиджак и аккуратно повесил на спинку кресла. Потом подошел к внуку и положил тяжелые теплые ладони ему на плечи.

- Ну, пожалуйста, дед, - протянул юноша. - Я давно хотел услышать эту историю.

Кренон посмотрел на экран и тихо сказал:

- Хорошо, только потом сразу спать.

Юноша встал, потянулся, разминая мышцы, и с легкостью начал стаскивать с себя курсантскую форму. Тем временем Кренон плеснул в стакан вишневого сока из прозрачной пластиковой бутылки и залпом осушил его.

Вспоминать историю высадки спасательной бригады было тяжело. Тогда он только принял командование десантной группой патрульного корабля "Радий" и, конечно, не подозревал, что судьба уготовит ему такую жестокую проверку на зрелость. Потом многие его хвалили за удачно проведенную операцию, но он-то знал, что, будь в его распоряжении хотя бы два часа лишнего времени, все могло оказаться по-другому. Не пришлось бы жертвовать людьми. Многие из подчиненных были его друзьями. Они заканчивали одну учебку, все вместе пошли служить в одно десантное подразделение, и район им достался, казалось бы, тихий - Луна. Экваториальная зона, сектор А-3. Некоторые были даже недовольны и пытались перевестись в другие десантные группы. Да и сам, молодой тогда еще, Кренон хоть и убеждал иных остаться, понимал, что Луна - это не огнедышащий Меркурий и не коварный Марс.

Служба шла размеренно и спокойно, пока серьезно не заболел командир десантников Петр Ковальски. В срочном порядке его отправили на землю, а командиром патрульного корабля был назначен Элмер Кренон. Неприятности начались на следующий день...

Он разминался в спортивном зале, когда неожиданно по внутреннему телеселектору его вызвал радист Билли Брэг. Освободив кисти рук от ремней тренажера, Кренон поднялся и неспеша подошел к пульту.

- Ну, что там у тебя?

- Только что получили сигнал SOS. - хрипло отозвался Брэг.

- Сейчас буду. - Кренон шлепнул ладонью по выключателю и тихо выругался. Без Ковальски он чувствовал себя достаточно беспокоино и неудобно, словно цыпленок без матери курицы.

Накинув поверх спортивного костюма китель, Кренон направился в командный отсек. Билли Брэга он застал возле дисплея бортового компьютера. На экране светилась карта-сетка района бедствия. Краем глаза Кренон заметил в верхнем правом углу экрана название лунной базы, с которой пришел сигнал.

- "Дельта?" - удивленно спросил он.

Брэг молча кивнул, переключая масштаб изображения.

- Что могло случиться у селенологов? - вслух подумал Кренон.

Брэг оторвался от экрана и повернулся к командиру:

- Они не отвечают, передатчик постоянно посылает сигнал SOS и больше ничего.

Кренон почувствовал, как лоб покрывается испариной:

- Объявляй общую тревогу. Да, и еще: свяжись с "Селеной" и доложи по форме.

В то время, как Билли Брэг выполнял поручения командира, Кренон опустился в кресло перед компьютером и запросил файл данных на лунную базу "Дельта".

Изучая колонки цифр, он думал о том, что могло стать причиной бедствия на этой тихой, забытой Богом базе. Построенная во времена второй волны колонизации и с самого начала отданная селенологам, база имела мощную буровую установку, небольшую, по нынешним временам, исследовательскую лабораторию, атомный реактор, сад с фонтаном под прозрачным центральным куполом и много всякого рода мелочей, необходимых для работы и отдыха персонала. Конструкция базы и ее основные элементы были сделаны на совесть. Малейшее отклонение от нормы в работе важнейших агрегатов четко отслеживалось центральным компьютером. Автоматизированная система контроля во главе с огромным вычислительным комплексом "Исаак", заслужившая самые высокие оценки на всех базах подобного типа, на этот раз, видимо, не сумела предотвратить бедствие. Вероятно, все произошло внезапно и последствия оказались необратимыми.

Командный отсек быстро заполнили созванные по тревоге десантники. Весь экипаж патрульного корабля "Радий" собрался у центрального пульта управления, чтобы послушать указания орбитального комплекса "Селена". Выслушав сообщение Билли Брэга, оператор "Селены" некоторое время не выходил на связь.

- Как всегда, - заметил Джаспер Прюн, - Теперь дебаты устроят...

- Симпозиум соберут, - согласился с ними Леон Троцкий, который еще в школе космодесантников получил прозвище "Революционер" из-за своей известной фамилии.

Однако ответ с "Селены" пришел быстро. Уверенный голос оператора внушал оптимизм:

- Это Ваш район, так что действуйте по инструкции.

К каналу связи подключился Кренон:

- Специальных указаний нет?

- Нет, - ответил оператор.

- Понял, - задумчиво отозвался командир "Радия". - Отбой.

Он медленно встал с кресла. Все в ожидании повернулись к нему.

- Готовность номер один и по полной форме, - выдохнул Кренон.

Десантники удивленно переглянулись. Все слышали, что "Селена" не дала специальных указаний. Следовательно, можно было бы не брать с собой боевых роботов, полный боекомплект вооружений и не надевать военные скафандры. В таких случаях обычно ограничивались медицинской аппаратурой и легкими скафандрами.

- Береженого Бог бережет, так что ли? - нарушил молчание Леон Троцкий.

- Особенно революционеров, - иронично отозвался Джим Дигриз.

- Выполняйте приказ, - отрезал Кренон.

Десантники стали молча расходиться. Курт Левин, - заместитель командира патрульного корабля, пилот первого класса, - расположился за пультом управления. Только сейчас Кренон посмотрел на главный экран. Лунное Море Дождей без единой капельки воды медленно выплывало навстречу, открывая бесконечные пространства застывшей миллионы лет назад магмы. До лунной базы "Дельта" оставалось около двухсот километров.

- Включаю орбитальные двигатели, - доложил Курт Левин.

Патрульный корабль лег на правый борт и медленно стал приближаться к поверхности Луны, удерживая курс на ""Дельту".

Корабль они посадили недалеко от базы селенологов. Прозрачный центральный купол переливался в ослепительных лучах не по-земному яркого солнца. Служебные постройки, примыкавшие к куполу, выглядели обыкновенно. Никаких следов разрушений замечено не было. На связь по-прежнему никто не выходил.

Кренон построил отряд в шлюзовом отсеке "Радия". Двенадцать вооруженных десантников в скафандрах со снятыми гермошлемами в ожидании стояли напротив него.

- Разбиваемся на четыре группы, - начал Кренон. - Первая группа - Нельсон Смит и Джаспер Прюн. Ваша задача осмотреть цех переработки. Обо всех обнаруженных неисправностях сразу докладывать.

Механик Джаспер Прюн утвердительно кивнул, но тут же, сняв с плеча лазерное ружье, продемонстрировал его на всеобщее обозрение:

- А это тогда для чего? Гвозди забивать?

Все засмеялись, лишь один Кренон с каменным лицом молча посмотрел на Прюна.

- Вторая группа, - продолжил Кренон, - Вилли Брэг, Леон Троцкий и Илэйн Фрейх. Вы осмотрите центральный вход, ангар и прилегающие помещения. Третья группа - Дон Кейлони, Сларти Бартфаст и Рита Рампол. Ваша задача проверить жилые отсеки. Пойдете через служебный вход. Старший группы Дон Кейлони.

Непалец Рита Рампол хотел что-то сказать, но Кренон жестом остановил его.

- Четвертая группа - Курт Левин, Джим Дигриз и курсанты Джил и Джой. Проверьте запасной выход и присоединяйтесь ко второй группе. Старший - Курт Левин. Задача ясна? - спросил Кренон, придирчиво осматривая каждого десантника.

Вопросов ни у кого не было. Кренон посмотрел на Нельсона Смита:

- Боевые дроиды готовы?

- Да.

- Приступайте к разгрузке. Каждой группе - по дроиду.

Смит подошел к автоматической двери, ведущей в дроидный отсек и нажал на панели электронного замка несколько кнопок. С мягким гудением, дверь отъехала в сторону, освободив проход. Из глубины помещения бесшумно вышли один за другим четыре боевых дроида. Десанникам пришлось потесниться: боевые машины цилиндрической формы на массивных стальных ногах-шарнирах заняли почти все свободное пространство. Кренон ободряюще кивнул на вопросительный взгляд Нельсона Смита и махнул рукой.

- Выходим, - сказал он и закинул на плечо фотонный излучатель.

Застегивая колпак гермошлема, Кренон прокрутил в голове все параграфы инструкций для спасательных отрядов и остался доволен.

Они оставили корабль на дне небольшого древнего кратера и, разделившись на четыре группы, быстро добрались до базы.

Первые опасения появились, когда они остановились напротив наглухо запертых шлюзовых ворот. Автоматика полностью заблокировала вход и открыть двери не было никакой возможности. Командиры групп доложили по общей связи о том, что и у них ничего не получается. База молчала, сигнал SOS больше не поступал. В эфире воцарилась тишина.

- Похоже, нас здесь совсем не ждут, - прозвучал в шлемофонах голос Курта Левина. - Командир, придется применить дронов.

- Возможно, - задумчиво отозвался Кренон, - но лучше не тратить их заряды. У нас

есть светосабли, давайте их и попробуем.

Светосабля взрывного контактного действия позволяла прорубать проходы в любых завалах и поэтому давно была на вооружении спасательных отрядов. Практически любой, даже самый прочный, металл сдавался под натиском мощных электро-лазерных разрядов.

- А что нам прикажете делать? - прозвучал недовольный голос Джаспера Прюна. - У нас нет с собой ни одной светосабли.

- Я иду к Вам, - ответил Кренон. Он посмотрел на отливающий солнцем стеклянный колпак гермошлем Леона Троцкого. - За старшего - Троцкий.

- Есть, - откликнулся Леон. - Билли, где будем рубить?

- Революция, не засоряй эфир, - пошутил кто-то из десантников.

Кренон легко преодолел несколько метров до входа в цех переработки, где его ждали Нельсон Смит и Джаспер Прюн.

- А что, если у них отключена герметизационная защита - увидев приближающегося Кренона спросил Смит. Нас же сдует, как пылинки!

Кренон ничего не ответил. Он вплотную приблизился к шлюзу и, отметив середину двери, достал из ножен светосаблю. В этот момент на связь вышел Дон Кейлони, старший группы, которая шла через служебный вход:

- Командир, у нас ничего не выходит. Прошу разрешить использовать дроида.

- Разрешаю.

В шлемофоне что-то затрещало, затем радался звук, напоминающий взрыв хлопушки, а следом прогремел бас Курта Левина:

- У меня получилось, я в тамбуре. Гермозащита работает. Можно смело входить.

Кренону хватило четырех ударов, чтобы прорубить внешнюю шлюзовую дверь, ведущую в цех переработки. Он вошел в довольно просторное помещение шлюзового тамбура. Сработала система герметизации и прорубленный вход закрылся прозрачной световой пленкой, временно защищающей базу от вакуума. Следом вошли Прюн и Смит, затем медленно вошел дROID. Кренон остановился перед внутренней шлюзовой дверью, чтобы отдышаться. Надо сказать, что работать светосаблей было нелегко и приходилось прилагать большие усилия для хорошего контакта сабли с металлом.

- Старшие групп, - сказал Кренон, - доложите обстановку.

Первым откликнулся Леон:

- Вошли в предбанник.

- Аналогично, - отозвался Курт Левин.

Дон Кейлони после недолгого молчания поведал всем о том, что его дROID разрезал вторую перегородку и они вошли в гардеробную.

- Что-нибудь наблюдаете? поинтересовался Кренон.

- Ничего подозрительного, ответил Кейлони, - хотя... Черт!!!

Послышался треск и на время все стихло. Потом все услышали дрожащий голос Рампола, перекрываемый шумом:

- Внимание! Нападение на группу... - Опять треск и шум.

- Что там у вас?

- Дон, ответь!

- Говорит Дон Кейлони. Нас обстреляли какие-то дроиды. Мы отступаем.

- Говорит Кренон. Сколько их?

- Не знаю, много. Очень много!

- Дон, без паники... - Кренон, замахнувшись саблей, рубанул твердый металл перегородки. Ничего не получилось. Перегородка устояла.

- Говорит Леон. Мы вскрыли центральный вход, я вижу с десяток дроидов, которые движутся к служебному входу. Меня пока не замечают. Впрочем, подождите...

Они открыли о... - речь Троцкого оборвалась на полуслове. Эфир наполнился непрекращающимся треском.

- Всем назад! - кричал в микрофон Кренон, нанося светосаблей сильнейшие удары упрямой перегородке. Наконец, она подалась.

В образовавшемся проходе он увидел высокого дроида с круглой головой и

остроносой лазерной пушкой, который очень быстро двигался в сторону служебного входа. Затаив дыхание, Кренон смотрел на удалявшегося робота.

- Ого! - выглядывая из-за плеча командира, воскликнул Нельсон Смит, - откуда они здесь?

Кренон отступил назад:

- Дон, отзовись!

- На связи, - тяжело дыша сказал Кейлони.

- Я надеюсь, все обошлось?

- Сларт ранен, а у нас нет даже медицинского зонда.

- Дон, они все движутся к вам.

- Мы отошли, - успокоил всех Кейлони. - Прячемся за своим железным болваном.

Командир, какие будут указания?

Кренон на секунду замешкался. Такого поворота событий он, конечно, не ожидал. Хорошо, что внутреннее чутье его не подвело и он полностью вооружил отряд. Но кто мог знать, что безобидная захолустная база селенологов буквально кишит военными дроидами. Конечно, на базах подобного типа всегда есть пять или шесть дроидов. Это, как правило, механические роботы, которые занимаются погрузкой и разгрузкой кораблей, а также добычей полезных ископаемых. Что здесь делают военные машины? Сомнительно, что их привезли с Земли. Зачем? Да и контрольную проверку ни один корабль с подобным грузом не смог бы пройти, а если бы это произошло, то на всех постах стало бы известно о существовании такой странной базы селенологов. С "Селены" никаких специальных указаний не было. Ясно, что там ничего не знали. Правда, военные дроиды это не их профиль. Вопросы нужно задавать Комитету Обороны. Тогда, какого дьявола нас не предупредили? Остается одно: дроидов собирали на этой базе. Причем, быстро и в тайне от всех. Видимо, выполняли заказ какой-нибудь могущественной военной корпорации.

И вдруг он почувствовал, что знает, где лежит ключ к разгадке тайны лунной базы "Дельта".

- Смит, - позвал Кренон. - А как управляются эти дроиды?

Из тени развороченной шлюзовой двери появился роботехник Нельсон Смит:

- Обычно по радиосвязи. Но, в принципе, программа может быть занесена и в оперативную память.

Смита слышали все, но слово снова взял Кренон:

- Дело в том, что этих дроидов собирали здесь. Для каких целей - не важно. Но я хочу сказать, что выполняя спецзаказ какой-то корпорации, селенологи вряд ли были посвящены в детали и не знали, как и где дроиды будут применяться, так? Поэтому готовый продукт, скорее всего, был без рабочей программы.

- Возможно, - протянул Смит.

- Кто может управлять роботами с базы?

- Что? - Нельсон вздрогнул и посмотрел сквозь стекло гермовлема на Кренона. - Хочешь сказать, сами селенологи?

- Допустим...

- А кто тогда посылал сигнал бедствия?

- Вот именно, - выдохнул Кренон, - я думаю, этих селенологов дроиды и убрали.

- Выходит, бунт?

- Да.

- Кто же во главе?

- Тебе видней... Мне, например, кажется, что это "Исаак" - компьютер. Больше никому.

У Нельсона округлились глаза:

- Похоже, очень похоже на правду. В комплексе "Исаак" есть передающие и принимающие устройства. Ему вполне под силу управлять сотней таких дроидов.

- Хорошо, - Кренон снял с плеча фотонный излучатель. - Эту догадку мы можем проверить?

- Конечно, нужно только настроиться на их частоту.

- Займись этим, Смит. Билли Брэг, ты меня слышишь?

- Я на связи.

- Что с Леоном?

- Легкое ранение, Илэйч им сейчас занимается.

- Хорошо. Дон Кейлони!

- Слышу тебя.

- Пока ничего не предпринимайте. Ждите указаний.

- Мы за границей шлюза. Нас не трогают.

- Обращение к старшим групп! До моего особого указания всем находиться за пределами базы.

- Я поймал сигнал "Исаака"! закричал Смит, работавший с личным передатчиком. - Все сигналы закодированы, ничего понять невозможно. Но это точно он!

Кренон доверял Нельсону Смиту как самому себе.

- Судя по карте, - сказал он, - через цех переработки по тоннелю мы сможем попасть в операторскую, а там и компьютерный центр недалеко.

- Верно, - подтвердил Смит.

- Так он вас туда и пустит, вмешался Джаспер Прюн, поглаживая блестящую поверхность военного дроида. - У него там наверняка охрана.

- Мы их отвлечем, - уверенно сказал Кренон. - Ты видел, как они все помчались к служебному входу?

Джаспер Прюн пожал плечами. Кренон снова вышел на связь:

- Дон, ты меня слышишь?

- Я все понял. Берем огонь на себя, - вздохнул он.

- Дон, прошу, держитесь как можно дольше. Успеха! - заключил Кренон, а для остальных добавил: - группы Курта Левина и Леоны Троцкого ждите указаний. Я с Прюном и Смитом попробую пробраться к "Исааку".

- Мы выдвигаемся, - доложил Дон Кейлони.

Кренон выглянул в проем и увидел, как еще несколько дроидов, словно по приказу, направились к служебному входу.

- Какое у них может быть оружие? - шепотом поинтересовался Кренон у Смита.

- Какое угодно: от слабеньких зикеров до мощных лазерных установок.

В эфире опять что-то затрещало и послышались легкие хлопки. Стало ясно, что у служебного входа началась схватка. Кренон внимательно осмотрел цех переработки, который предстояло пересечь, чтобы войти в тоннель. По центру цеха располагался огромный конвейер, который непрерывно работал. Дальше он видел мощную буровую установку, а за ней вход в заветный тоннель, ведущий к цели. Вдоль конвейера располагались процессоры, которые перерабатывали поступающий с конвейерной линии грунт. Повсюду были разбросаны груды щебня и остатки обработанных ископаемых.

Кренон махнул рукой и, пробежав вперед несколько шагов, остановился под прикрытием конвейера. Следом примчались Прюн и Смит. Военный дROID немного поотстал, зато надежно защищал их с тыла. Шум в эфире прекратился и десантники отчетливо услышали чей-то стон.

- Докладывает Сларты Бартфаст. Первая атака отбита. Кейлони ранен. Рита Рампол убит.

- Группы Троцкого и Левина вперед! - срывающимся голосом командовал Кренон.

- Осторожней, командир!

Кто это сказал, Кренон так и не понял: ослепительный яркий, чуть голубоватого цвета луч пронесся слева и ударился в процессор, рассыпавшись фонтаном искр. Стреляли из тоннеля. Конечно, не все дроиды противника ушли к служебному входу. Прюн был прав. "Исаак" обеспечил себе надежную защиту.

Совсем рядом прошипел еще один луч. Кренон рванулся вперед и, преодолев несколько метров свободного пространства, спрятался за грудой щебня.

В эфире все трещало, стреляло и взрывалось. Были слышны чьи-то крики и стоны. Кренон заметил, что справа от него в коридоре, ведущем к нейтральному входу, где

находилась группа Леона Троцкого, скопилось с десятков неприятельских дроидов. Часть роботов отделилась от остальных и двигалась теперь к буровой установке

- По-моему, это к нам, - заметил Джаспер Прюн, который взобрался на ленту конвейерной линии, чтобы оценить обстановку.

Кренон не видел, что произошло у конвейера - ему мешала гора щебня, за которой он прятался, но он отчетливо услышал крик Смита:

- Ну, иди сюда, гад!

Яркие вспышки озарили весь цех. Металл и камни плавились под натиском смертоносных лучей. Этот кошмар продолжался несколько секунд, но Кренону показалось, что прошла целая вечность. Когда все стихло, он увидел андроида двухметрового роста с лазерным пистолетом в железной руке. "Здесь еще и андроиды есть", - только и успел подумать Кренон, когда луч лазера расплавил добрую половину груды щебня в полуметре от него. Он несколько раз выстрелил в сторону андроида, но, похоже, промахнулся. Необходимо было сменить укрытие, иначе в следующий раз, вместо этой бурой корки оплавленного камня, будет лежать он сам. Кренон бросился к тоннелю, который казался таким близким и ему было уже все равно, что совсем недавно оттуда стреляли. В глазах черным пятном стояла вспышка луча, бившего в щебень. Кто-то уже бежал за ним. Кренон обернулся. Это был Джаспер Прюн. В эфирной многоголосице было невозможно разобрать, что кричал ему механик. Из глубины тоннеля под ноги ударил луч, но на этот раз Кренон успел выстрелить ответной очередью. Где-то в глубине коридора вспыхнул голубым пламенем патрульный дROID и с грохотом разлетелся на куски.

В мимолетном затишье Кренон вдруг разобрал слова Прюна:

- Смита убили. Но он положил трех дроидов.

Комок подкатил к горлу. Пытаясь справиться с неприятным ощущением, Кренон прокричал в микрофон:

- Все группы к служебному тоннелю!

Отозвался, почему-то, только один Леон Троцкий:

- Я у входа в цех переработки.

- А остальные?

- Билли Крэг убит, а Илэйн Фрейх тяжело ранен и уже не может идти. Он отстреливается у входа в ангар.

- Где Курт Левин? - спросил Кренон.

- Я его не видел, - ответил Леон.

Сквозь шум и свист в шлемофоне отчетливо прозвучал голос Курта:

- Я не могу пересечь ангар. Веду круговую оборону.

- Где твои люди?

- Курсанты у соседнего лунохода, а Джим Дигриз с военным дроидом направился к вам.

- Посмотри, там где-то Илэйн Фрейх. Он ранен. Помоги ему, - подключился к разговору Леон Троцкий.

- Ладно. А, черт! Ах ты, электронный выродок!...

Курт Левин больше не отвечал. Кренон и Прюн уже шли по тоннелю. Сзади, принимая на себя основной огонь, двигался военный дROID.

Сквозь перекрестный огонь в тоннель проник Леон Троцкий, неизвестно как сумевший преодолеть этот страшный путь.

Кренон понимал, что основной удар приняли на себя группы Курта Левина и Дона Кейлони и им удалось оттянуть часть дроидов.

Дверь операторской была заперта изнутри. Кренон решил использовать один из четырех мощнейших зарядов своего боевого робота. Вспышка, фейерверк огня и двери как не бывало.

ДROID въехал в образовавшийся проход. Следом вошел Кренон, а Джаспер Прюн не успел. С противоположной стороны раздался выстрел. Шипящий луч ударил в грудь Прюну и тот успел лишь вскинуть лазерное ружье, когда прозвучал еще один выстрел. Джаспер Прюн упал навзничь с прожженной насквозь грудью.

Кренон едва успел отскочить в сторону, как из бокового коридора, ведущего в компьютерный центр, ударил встречный луч. Заряд лазера приняла на себя броня боевого дроида десантников. Узкий коридор наполнился дымом.

В этой сумасшедшей свалке трудно было разобраться. Их зажали со всех сторон. Сразу три огромных дроида вели непрерывный огонь и Кренону пришлось прятаться в операторской. Улучшив момент, он попытался связаться с остальными, но ему никто не ответил. Лишь один раз, откуда-то, словно с далекой Земли, до него донесся слабый голос Дона Кейлони, которого он уже давно мысленно похоронил:

- Командир, мертвые не простят, если мы не победим.

Послышалось ему это или нет, но он почувствовал второе дыхание. Потратив всю обойму на дроида, охранявшего вход в компьютерный центр, Кренон ворвался внутрь. Посередине зала, мигая лампочками и переливаясь всеми цветами радуги, мирно гудел компьютер "Исаак". Кто бы мог подумать, что эта машина сумеет принести столько бед. Кренон перезарядил фотонный излучатель и начал методично расстреливать управляющие консоли компьютера. С противоположного конца зала к нему приближался дроид с лазерной головой. Кренону показалось, что "Исаак" буквально раскалился, предчувствуя свой близкий конец. И теперь командир десантников понимал, что этот единственный дроид последняя надежда компьютера. Он представил себе, как электронная душа "Исаака" переселяется в военного дроида. И вот уже компьютер - не беспомощная железяка, набитая микросхемами, а грозный и опасный противник. Да вот же он: совсем рядом!

"О чем это я?" - одернул себя Кренон, уворачиваясь от лазерного луча, - по-моему, начинаю сходить с ума". Он с удивлением отметил, что вот уже несколько минут его не покидает желание прекратить стрельбу, да и вообще бросить фотонный излучатель на пол. Ему стало страшно. Дроид был совсем близко. Но прятаться за "Исааком" Кренону почему-то совсем не хотелось. Он хорошо видел, как шарообразная голова дроида поворачивается в его сторону, лазерная пушка смотрела прямо в лицо.

- Не-е-т! - закричал Кренон и в последнее мгновение успел отпрыгнуть в сторону.

Дроид дернулся следом и выстрелил. Лазерный луч, шипя, стал вгрызаться в непрочную обшивку "Исаака". Кренон видел, как плавились электронные мозги агрессора и как тут же застыл без движений вражеский дроид. Это была победа! Раб сам убил своего хозяина. Мгновение спустя в зал ворвался Леон Троцкий:

- Неужели конец?

- Конец, - выдохнул Кренон.

В гермошлемах затрещало и на связь вышел Курт Левин:

- Леон, ты еще спрашиваешь? Иди и посмотри, сколько вокруг меня собралось застывших истуканов.

У Кренона не было сил даже улыбнуться.

* * *

Когда Элмер Кренон закончил свой рассказ, его внук еще долго лежал в постели, смотрел в потолок и молчал. Потом он спросил:

- Дед, а сколько человек вас осталось тогда в живых?

- Пять человек: я, Леон Троцкий, Курт Левин, Джим Дигриз и Дон Кейлони.

- А вы узнали, почему компьютер "Исаак" взбунтовался?

- Когда вскрыли черный ящик, то выяснилось, что переполненный военной информацией, компьютер просто не выдержал экспериментов этих, так называемых, "селенологов". Почему это произошло, мы так и не узнали, потому что дроид очень сильно повредил даже черный ящик. Мы можем лишь догадываться, какая каша заварилась в недрах базы перед нашим прибытием, только с той поры компьютеры "Исаак" были сняты с производства.

- А вы еще летали вместе?

- Со времен "Дельты" мы стали неразлучными друзьями и побывали в таких передрягах, что события на лунной базе покажутся детской забавой.

- Это когда? - У внука загорелись глаза.

- Ну-у, - протянул Кренон, - например, в системе Альфа Центавра, когда мы наткнулись на поселение совершенно чуждых нам существ.

- Дед, расскажи!

- В следующий раз, - усмехнулся Кренон. - Давай спать...

АВТОРСКАЯ ПРОГРАММА

Развивая инициативу, объявленную в прошлом выпуске "РЕВЮ", мы начинаем публикацию заявок на авторское право по программным и аппаратным разработкам для домашних персональных компьютеров системы "Синклер-Спектрум".

"ИНФОРКОМ" рассчитывает, что кроме чисто "заявочного" характера этот раздел будет служить еще и справочным для тех, кто хочет и может сам что-то создать. Здесь Вы можете увидеть над чем работают Ваши коллеги, может быть найдете полезные идеи и для себя.

Мы надеемся, что публикации этого раздела явятся своеобразным генератором новых идей.

Раздел открыт для Вас, уважаемые авторы.

1. СЕРВИСНЫЕ ПРОГРАММЫ (УТИЛИТЫ).

ЗАЯВКА: У001.

ПРОГРАММА: INSTALL (инсталлятор).

АВТОР: В. Кутин, г.Екатеринбург.

ДАТА РАЗРАБОТКИ: март 1993 г.

ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.

НАЛИЧИЕ ЗАЩИТЫ: нет.

ДОКУМЕНТАЦИЯ: - не указано.

КРАТКОЕ ОПИСАНИЕ:

Программа позволяет создать дисассемблер под тот адрес, который нужен, при необходимости встроит в дисассемблер и монитор. Полученным дисассемблером можно пользоваться независимо от инсталлятора. Он обладает следующими особенностями:

- особая компактность;
- цветность;
- озвученность;
- многооконность (4 окна для одновременной работы с разными фрагментами);
- наличие DISPLACE-системы для мониторинга программ, написанных в области экрана или в области системных переменных;
- 16 функций монитора;
- 16 функций дисассемблера;
- имеются процедуры, написанные специально для хаккеров.

* * *

ЗАЯВКА: У002.

ПРОГРАММА: TURBO-COPY (копировщик)

АВТОР: В. Кутин, г.Екатеринбург.

ДАТА РАЗРАБОТКИ: июль 1993 Г.

ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.

НАЛИЧИЕ ЗАЩИТЫ: нет.

ДОКУМЕНТАЦИЯ: - не указано.

КРАТКОЕ ОПИСАНИЕ:

Внешне выглядит, как TURBO COMP, ZX-COPY, TF-COPY, но имеет ряд преимуществ:

- работа с двумя скоростями - стандартной и аналогичной ПЗУ-90.
- в отличие от TURBO-COMP имеет функцию VERIFY;
- выдает информацию о компрессии блоков в процентах;
- имеет современное оформление (стилизованный шрифт);
- работает на тех машинах, на которых TURBO COMP не работает;
- особо полезна для машин с ПЗУ-90.

* * *

ЗАЯВКА: У003.

ПРОГРАММА: MEGA-GEN (генератор/редактор спрайтов)

АВТОР: Д. Палтусов, г.Екатеринбург

ДАТА РАЗРАБОТКИ: апрель 1993 г.

ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.

НАЛИЧИЕ ЗАЩИТЫ: нет.

ДОКУМЕНТАЦИЯ: - не указано.

КРАТКОЕ ОПИСАНИЕ:

Предназначена для создания спрайтов любого размера (по вертикали - в пикселах, по горизонтали - в знаках). Имеет развитую систему меню, удобна в работе, возможно управление джойстиком (Sinclair и Kempston), работает с группой спрайтов, результаты сводит в файл, удобный для использования в пользовательской программе ("подшивает" файл-указатель размещения спрайтов).

* * *

ЗАЯВКА: У004.

ПРОГРАММА: GENERATOR (генератор/редактор спрайтов)

АВТОР: Д.Палтусов, г.Екатеринбург.

ДАТА РАЗРАБОТКИ: март 1992 г.

ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер, БЕЙСИК

НАЛИЧИЕ ЗАЩИТЫ: есть.

ДОКУМЕНТАЦИЯ: не указано.

КРАТКОЕ ОПИСАНИЕ:

Рассчитан на начинающих программистов, позволяет создавать спрайты 16 X 16. Готовые спрайты можно использовать как из машинного кода, так и из Бейсика ("зашиваются" в область UDG). Возможно управление джойстиком (Sinclair).

* * *

ЗАЯВКА: У005.

ПАКЕТ: CONSTRUCTOR

АВТОР: Д.Палтусов, г.Екатеринбург.

ДАТА РАЗРАБОТКИ: сентябрь 1992 г.

ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.

НАЛИЧИЕ ЗАЩИТЫ: нет.

ДОКУМЕНТАЦИЯ: - не указано.

КРАТКОЕ ОПИСАНИЕ:

Пакет процедур, достаточных для создания игровой программы - карточной игры.

Состав пакета:

OPENDAUM - демонстрационная карточная игра, написанная с помощью "конструктора".

CONSTRUCTOR - блок подпрограмм в машинном коде.

TXT.FIL - текстовый файл программы OPENDAUM, написанный под ассемблер "ZEUS". Позволяет изучать логику работы игры и технологию применения "конструктора".

* * *

ЗАЯВКА: У006.

ПАКЕТ: MULTIEDIT UTILITY1

АВТОРЫ: Д.Палтусов, В.Кутин, г.Екатеринбург.

ДАТА РАЗРАБОТКИ: апрель 1992 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.
НАЛИЧИЕ ЗАЩИТЫ: есть.
ДОКУМЕНТАЦИЯ: - не указано.
КРАТКОЕ ОПИСАНИЕ:

Аналог аниматора FANTA VISION (для IBM совместимых машин). Служит для создания мультфильмов в четыре этапа:

- определение формата кадра;
- создание фаз анимации;
- моделирование экранного изображения;
- компиляция мультфильма.

В пакет входит мощный компилятор, позволяющий сгенерировать файл, независимый от материнской программы под любой адрес загрузки и программа UTILITY1 для "вырезания" исходного спрайта из заставок, созданных в других графических редакторах.

* * *

2. ПРИКЛАДНЫЕ ПРОГРАММЫ

Заявок нет.

3. ИГРОВЫЕ ПРОГРАММЫ

ЗАЯВКА: И001.
ПРОГРАММА: SHAREHOLDER
ЖАНР: TRADITIONAL/MANAGEMENT
АВТОРЫ: Д.Палтусов, В.Кутин, г.Екатеринбург
ДАТА РАЗРАБОТКИ: август 1992 г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.
НАЛИЧИЕ ЗАЩИТЫ: есть.
КРАТКОЕ ОПИСАНИЕ:

Компьютерный аналог известной за рубежом настольной игры (освещалась в журнале "Наука и жизнь").

Количество играющих - до 4-х человек, за них может играть компьютер по одному из двух возможных алгоритмов.

Цель игры - повышение состояния за счет рациональной игры с акциями на бирже.

Управление: клавиатура, CURSOR, SINCLAIR, KEMPSTON.

Экранный интерфейс: типа "ARTSTUDIO".

* * *

ЗАЯВКА: И002.
ПРОГРАММА: BASE
ЖАНР: PUZZLE/STRATEGY
АВТОРЫ: Д.Палтусов, В.Кутин, Р.Мельников, г.Екатеринбург.
ДАТА РАЗРАБОТКИ: Июнь 1992 Г.
ЯЗЫК ПРОГРАММИРОВАНИЯ: Ассемблер.
НАЛИЧИЕ ЗАЩИТЫ: есть.
КРАТКОЕ ОПИСАНИЕ:

Тема игры: освобождение космической базы, захваченной пришельцами.

Логика работы: противник уничтожается, будучи запертым в безвыходном пространстве.

Экранный интерфейс - типа "BOULDER DASH" и "ROCKFORD".

Игра многоуровневая и имеет встроенный редактор уровней, позволяющий развивать ее далее.

Управление: клавиатура, CURSOR, SINCLAIR, KEMPSTON.

Внимание, СИНКЛЕРИСТЫ!

То, о чем многие месяцы просят некоторые наши читатели, наконец свершилось! Мы объявляем о дне рождения нового журнала, посвященного играм для IBM-совместимых компьютеров, а также всему, что с ними связано.

1 января 1994 года выйдет в свет
первый номер журнала "PC-РЕВЮ"

Папой этого издания будет "ИНФОРКОМ", а мамой - новая фирма, которую мы сейчас регистрируем - акционерное общество закрытого типа "ИНФОРКОМ - ПРЕСС".

Наших старых, добрых и верных читателей и почитателей просим не беспокоиться. День рождения "PC-РЕВЮ" не станет похоронным для "ZX-РЕВЮ", мы работали и будем работать над улучшением его содержания и не только не собираемся сворачивать это дело, но и будем развивать его дальше. Более того, мы рассчитываем на то, что и многие из Вас подключатся к созданию и распространению этого журнала.

ЧТО ТАКОЕ "PC-РЕВЮ"?	СОСТАВ "PC-РЕВЮ"
<p>1. "PC-РЕВЮ" будет выпускаться и распространяться электронным способом. Уход от полиграфии снимет массу преград с его свободного распространения.</p> <p>2. Планируется выпускать по одному номеру ежемесячно. Объем каждого выпуска - 2,0...2,5 мегабайта. В заархивированном виде объем выпуска составит 1080 К в расчете на то, что он будет поставляться на одной дискете HD (1,2М) или на трех дискетах DD (3X360K) по выбору заказчика.</p> <p>3. "PC-РЕВЮ" будет поставляться без защиты от копирования. Мировая практика показала, что самая лучшая "защита" - это отсутствие защиты, но при оперативном обновлении информации. При таком подходе выигрывает и издатель и потребитель.</p> <p>4. Журнал готовится так, чтобы с ним можно было работать на IBM-совместимых машинах любого класса. Единственное ограничение - графические иллюстрации рассчитаны на мониторы VGA. Пользователи, не имеющие их, смогут работать с журналом, но не смогут просматривать сопровождающие картинки.</p> <p>5. У читателя будет возможность распечатать любые понравившиеся ему статьи журнала на имеющемся у него оборудовании.</p> <p>6. "PC-РЕВЮ" будет распространяться через сети дистрибуторов. Порядок распространения следующий: "ИНФОРКОМ-ПРЕСС" направляет ограниченное количество мастердисков своим дистрибуторам по регионам с лицензией на неограниченное копирование и распространение. Локальные дистрибуторы могут сами продавать выпуски "PC-РЕВЮ" на своих дискетах или копировать журнал на дискету заказчика, а могут создать свою дистрибуторскую сеть, в этом случае они станут Генеральными дистрибуторами.</p>	<p>1. Обзоры новейших игровых программ с иллюстрациями.</p> <p>2. Исследование и анализ известных и широкораспространенных игровых программ.</p> <p>3. Переписка с читателями. Каждый, кто любит игровые программы, сможет написать о своих открытиях и проблемах. Любое грамотно и интересно написанное письмо будет опубликовано для обсуждения.</p> <p>4. Обзоры зарубежной прессы: интервью с выдающимися программистами и издателями, обзоры новых аппаратных средств, расширяющих возможности стандартной машины, рассказы об известных фирмах.</p> <p>5. Раздел "Компьютерная новелла".</p> <p>6. Игровые этюды. К каждому номеру журнала мы будем прилагать отгруженные модули состояния тех или иных известных игр, представляющие собой некий этюд (по типу шахматного), решение которого доставит читателю удовольствие. Лучшие решения будут поощряться ценными призами.</p> <p>7. Рекламный блок. Этот блок не займет более 10% объема журнала и, если получит достойное развитие, позволит значительно снизить Ваши затраты на получение отдельных выпусков.</p> <p>6. Справочный блок. В него войдут:</p> <ul style="list-style-type: none">- указатель наших дистрибуторов (люди во всех уголках страны должны знать, к кому им надо обращаться для приобретения очередного выпуска);- указатель игр (для быстрого поиска, где и когда освещалась та или иная игра);- указатель авторов (для фиксации авторских прав наших внештатных корреспондентов, статьи которых будут публиковаться в журнале);- указатель самых низких цен по фирмам г. Москвы на радиоэлектронную продукцию на текущий месяц.

Итак, для создания собственной первичной сети Генеральных дистрибуторов мы объявляем ТЕНДЕР на распространение "РС-РЕВЮ". Тендер - это как бы закрытый аукцион и он применяется в тех случаях, когда открытый аукцион провести невозможно.

1. ТЕНДЕР завершится к 15.12.93

2. Подать заявку каждый желающий может уже сейчас. Заявка подается в произвольной форме, в заявке Вы должны указать МАКСИМАЛЬНО ВОЗМОЖНУЮ ДЛЯ ВАС цену, по которой Вы готовы принимать мастердиск одного выпуска журнала для лицензированного копирования и распространения, а также форму оплаты (нал., б/нал.). При определении экономически обоснованной максимальной цены ориентируйтесь на свои возможности по распространению копий, а также на возможность организации Вами региональной дистрибуторской сети.

Обращаем Ваше внимание на то, что указание вами предельно допустимой цены НИ К ЧЕМУ ВАС НЕ ОБЯЗЫВАЕТ. В 99 случаях из 100 Вам придется платить значительно меньше, в данном случае Ваша цифра явится только свидетельством серьезности намерений и наличия потенциальных возможностей как будущего дистрибутора, что нам и необходимо знать при создании первичной сети.

3. 15.12.93 мы отберем 100 наиболее весомых заявок и установим единую цену мастердиска. Она будет равна размеру самой слабой из ста отобранных заявок.

4. Все победители ТЕНДЕРА получают извещение с указанием установленной цены и дальнейшие указания.

5. Итоги ТЕНДЕРА будут объявлены в последнем выпуске "ZX-РЕВЮ" этого года.

ПРИМЕЧАНИЕ:

1. Если на ТЕНДЕР поступит менее 100 заявок, ТЕНДЕР считается несостоявшимся и мы примем решение об изменении порядка распространения "РС-РЕВЮ".

2. Всех держателей "Зеленых пакетов", заинтересованных в распространении "РС-РЕВЮ", просим обратиться к нам по телефону или письмом. Им, как уже зарегистрированным дистрибуторам, будут предоставлены некоторые льготы.

Содержание

СПЕКТРУМ В ШКОЛЕ	1
SINCLAIR LOGO	4
ГЛАВА 5 ПОВТОРЯЮЩИЕСЯ ОПЕРАЦИИ	4
Условия.	4
Математические условия.	4
Рекурсия.....	6
Генератор случайной последовательности.....	9
ГЛАВА 6. ЧЕРЕПАШЬЯ ГРАФИКА-2.....	11
ПРИМЕНЕНИЕ АССЕМБЛЕРА ДЛЯ СОЗДАНИЯ БЫСТРОРАБОТАЮЩИХ ПРОГРАММ	17
7. ДВИЖЕНИЕ	17
Перемещение знакоместа переднего плана экрана	17
Перемещение изображения заднего плана экрана	19
НАША ПРЕЗЕНТАЦИЯ.....	31
5.6 Расчленение стрингов.	31
FORUM	33
ПРОФЕССИОНАЛЬНЫЙ ПОДХОД	48
ПРИМЕНЕНИЕ СТАТИЧЕСКОГО ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ НА ПРИМЕРЕ ПРОГРАММЫ "ELITE"	48
ВОЗВРАЩАЯСЬ К НАПЕЧАТАННОМУ	54
SWORDS & SORCERY.	54
HEAVY ON THE MAGIC	56
ЧИТАТЕЛЬ ЧИТАТЕЛЮ	60
СОВЕТЫ ЭКСПЕРТОВ.....	63
FIGHTER BOMBER	63
Загрузка игры.....	63
Управление самолетом	64
Приборная доска	65
Полет.....	65
Военные миссии	66
MERCENARY	67
Управление игрой.	68
Работа с лентой.	70
Экран программы	70
Описание транспортных средств и предметов.....	70
СДЕЛАЙТЕ САМИ	76
DICEY	76
КОМПЬЮТЕРНАЯ НОВЕЛЛА	86
СТРАТЕГИЯ КАПИТАНА КРЕНОНА.	86
Часть 1. Лунная база "Дельта".	86
АВТОРСКАЯ ПРОГРАММА	95