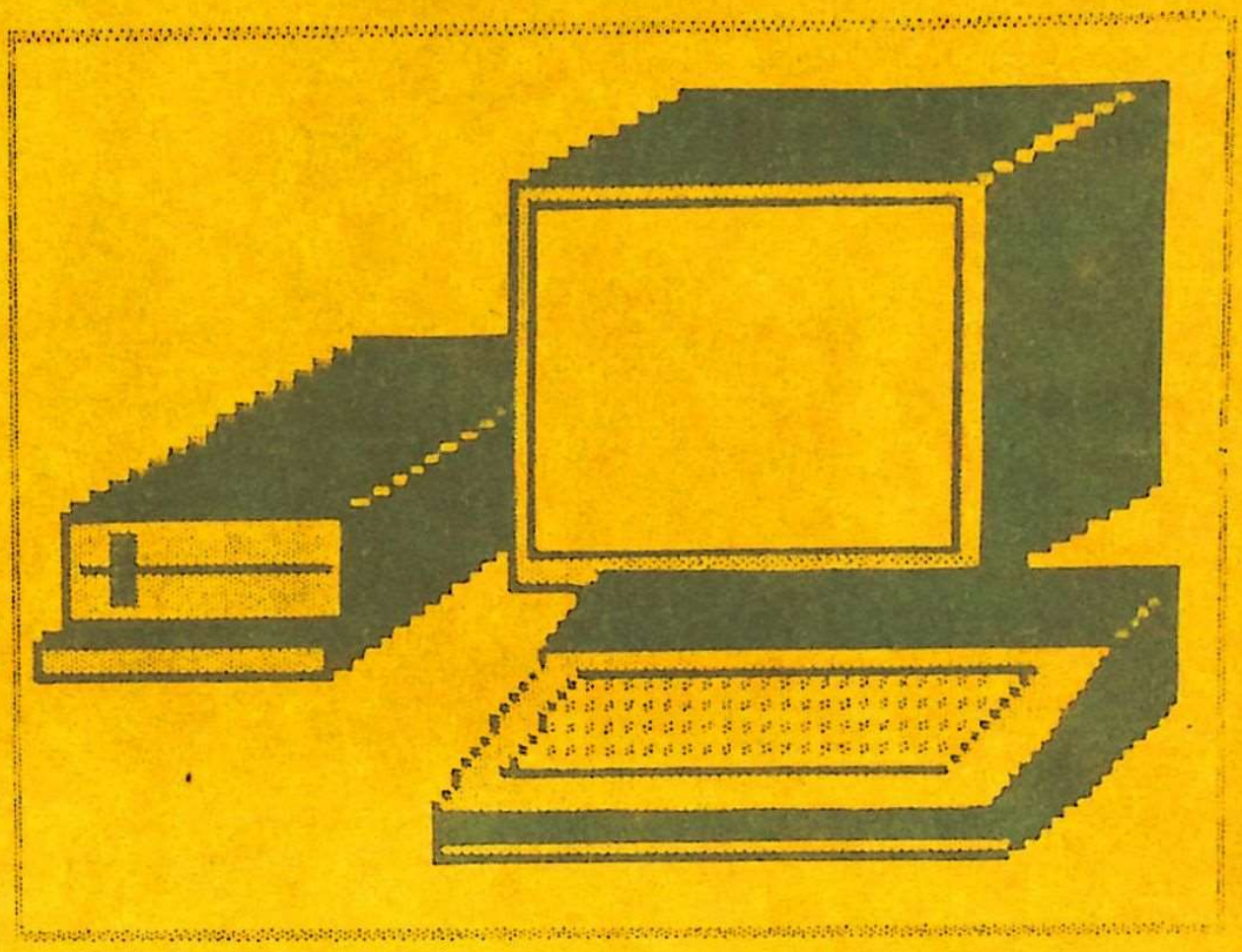


Co można zrobić z komputerem Atari XL,XE?

Podstawy programowania w Atari Basic'u
Podstawy grafiki znakowej
Rozbudowane techniki graficzne
Praktyczne wykorzystanie komputera

JERZY ZASKIEWICZ

CO MOŻNA ZROBIĆ Z ATARI XL, XE?

- *podstawy programowania w języku ATARI BASIC.*
- *podstawy grafiki znakowej.*
- *rozbudowane techniki graficzne*
- *kolorowa grafikia w trybie zerowym.*
- *program edukacyjny "Tabliczka mnożenia".*
- *praktyczne zastosowanie ATARI w pomiarach.*
- *interfejs do sterowania makietą kolejki.*

ELBLĄG 1991

SPIS TRESCI

1.LEKCJA 1

Nasz ATARI składa się,tak jak każdy komputer,z kilku podstawowych bloków.Są to:układy we-wy,jednostka centralna, pamięć zewnętrzna.

Układy wejścia-wyjścia służą do komunikowania się użytkownika z systemem komputerowym oraz systemu z użytkownikiem lub urządzeniami zewnętrznymi np.z drukarką lub modemem.W naszym przypadku układem wejścia jest klawiatura,a czasami joystick. Układem wyjścia-telewizor,monitor lub drukarka.Komputer wyświetla(lub drukuje) tam swoje komunikaty oraz wyniki działania programu.

Pamięć zewnętrzna służy do przechowywania danych i programów które są używane do pracy lub zabawy z komputerem.
Jednostka centralna jest sercem całego systemu komputerowego.Są tam wykonywane wszystkie operacje związane z przetwarzaniem dostarczonych danych oraz wszelkie obliczenia.W skład jednostki centralnej wchodzi mikroprocesor,pamięć oraz wiele innych układów odpowiedzialnych za prawidłową pracę komputera.W komputerze są dwa rodzaje pamięci:ROM(read only memory-pamięć tylko do odczytu) oraz RAM(random access memory-pamięć o swobodnym dostępie).W pamięci ROM przechowywane są dane,które są niezbędne do pracy komputera po jego włączeniu np.system operacyjny,zestaw znaków,interpreter BASIC'a.Są to dane,których nie można zmienić,służą tylko do odczytu.Pamięć RAM służy do przechowywania danych i programów wprowadzanych z pamięci zewnetrznej lub z klawiatury.Po wyłączeniu komputera wszystkie dane z pamięci RAM giną.
Nasz ATARI posiada 64kB RAM(800 XL,65XE) oraz 24 kB ROM.ATARI 130 XE posiada 128 kB RAM.

2.LEKCJA 2.

Język ATARI BASIC jest jednym z najlepszych interpreterów języka BASIC dostępnych dla małych komputerów.Posiada zestaw instrukcji graficznych i muzycznych i bez trudności można je wykorzystać. Instrukcje w ATARI BASIC'u piszemy zawsze dużymi literami.Linie programu numerujemy kolejnymi liczbami całkowitymi.Pisząc program należy numerować linie tak aby między dwie linie programu zawsze można było wstawić jeszcze jedną.
Nasz pierwszy program:

```
5 REM NASZ PIERWSZY PROGRAM
10 PRINT "TO JEST PIERWSZY PROGRAM"
20 GOTO 10
```

W linii 5 jest instrukcja REM.Służy ona do wprowadzania komentarza do programu.Po tej instrukcji możesz pisać co chcesz i jak chcesz.Można w ten sposób dokładnie opisać co robią poszczególne bloki programu.

W linii 10 jest instrukcja PRINT(zamiast pisać PRINT możesz używać znaku zapytania ?).Ta instrukcja wyprowadza na ekran to co napiszesz po niej w cudzysłowie.

W linii 20 mamy instrukcję skoku GOTO.Program napotkawszy GOTO wykonuje skok do linii programu podanej po instrukcji.W przypadku gdy taka linia nie istnieje jest zgłaszany błąd nr.12.

Uruchomienie programu następuje po napisaniu RUN (bez numeru linii) i wciśnięciu RETURN.Program wyświetli na ekranie komunikat,ale będzie to robił bez końca.Aby go przerwać musisz wcisnąć BREAK lub RESET.
Aby zobaczyć nasz program należy napisać instrukcję LIST.Program pojawi się na ekranie wg.kolejnych numerów linii.Jeżeli jest za długi i nie mieści się na ekranie możesz podać tylko te numery linii,które chcesz zobaczyć:

```
LIST 10
LIST 10,100
```
Jeżeli chcemy wprowadzić jakieś dane w czasie działania programu należy użyć instrukcji INPUT.
```
5 REM INSTRUKCJA INPUT
10 ?"PODAJ LICZBE:"
20 INPUT A
30 ?"WPROWADZILES LICZBE ";A
```
Po uruchomieniu programu pojawi się komunikat PODAJ LICZBE:,a poniżej znak zapytania.Komputer oczekuje na wprowadzenie liczby z klawiatury i wciśnięcie RETURN.Zauważ,że w linii 30 po komunikacie został napisany średnik.Spowodowało to wyświetlenie wprowadzonej liczby bezpośrednio po komunikacie,a nie w nowej linii.

ZAPIS I ODCZYT PROGRAMU.

Aby nie stracić programu po wyłączeniu komputera należy go zapisać na taśmie.Do tego celu służy instrukcja CSAVE.

---w pamięci jest program w BASIC'u,który chcesz zapisać
---sprawdź czy magnetofon jest podłączony do komputera
---do magnetofonu włóż kasetę przeznaczoną do zapisu programów
---jeżeli zapisujesz pierwszy program przewiń taśmę do końca w lewo i wyzeruj licznik
---przewiń taśmę w prawo do uzyskania na liczniku wartości 3,5 lub 10
---jeżeli zapisujesz kolejny program ustaw licznik taśmy wg.zapisanych danych
---zapisz stan licznika
---wciśnij PLAY i RECORD w magnetofonie
---napisz CSAVE i wciśnij RETURN
---komputer wyda dwa dźwięki

---wciśnij dowolny klawisz(z wyjątkiem BREAK)
---zapali się dioda w magnetofonie i rozpocznie się proces zapisu
---po skończeniu zapisz stan licznika:PROGRAM1 05-32
---przy zapisie następnego programu ustaw licznik taśmy o pewną stałą wartość dalej:PROGRAM2 37-48
---wyłącz magnetofon klawiszem STOP

Odczyt programu przebiega podobnie.Należy ustawić taśmę wg.zanotowanych danych,wcisnąć PLAY w magnetofonie,napisać CLOAD i wcisnąć RETURN.Po usłyszeniu jednego dźwięku wcisnąć dowolny klawisz(z wyjątkiem BREAK) i program zaczyna się ładować.Nie zawsze ładowanie przebiega bezbłędnie.Gdy coś się nie uda (komputer zgłosi błąd) należy sprawdzić ustawienie taśmy i spróbować jeszcze raz.

ZADANIE DOMOWE:

Napisać poniższy program,uruchomić go i zapisać na taśmę.

```
10 REM ZADANIE DOMOWE
20 ?"TEN PROGRAM JEST ZADANIEM DOMOWYM!"
30 ?"JAK CHCESZ GO PRZERWAC WCISNIJ"
40 ?"RESET LUB BREAK."
50 ?"********************************"
60 ?"PODAJ LICZBE:";
70 INPUT A
80 ?"MAM CIE.PODALES ";A
90 ?"********************************"
100 GOTO 50
```

POZNAKE INSTRUKCJE:
REM.,PRINT,GOTO,LIST,RUN,INPUT,CSAVE,CLOAD

3.LEKCJA 3

Możliwość wykonania skoku z zapamiętaniem miejsca powrotu daje instrukcja GOSUB...RETURN.

```
10 REM PROGRAM Z PODPROGRAMEM
20 ?"PODAJ LICZBE:";
30 GOSUB 100
40 ?"PODALES ";A
50 GOTO 20
90 REM PODPROGRAM
100 INPUT A
110 RETURN
```

W linii 20 program wyświetla komunikat,w linii 30 skacze do podprogramu w liniach 100,110.Po napotkaniu RETURN wraca do programu głównego(do następnej linii po tej,z której był wywołany podprogram).

Sprawdzanie określonych warunków i podejmowanie decyzji umożliwia instrukcja skoku warunkowego:

IF warunek THEN nr.linii,GOSUB nr.linii,ciąg instrukcji

```
10 REM PROGRAM SKOKU WARUNKOWEGO
20 ?"PODAJ LICZBE WIEKSZA OD 10 ";
30 INPUT A
40 IF A>10 THEN ?"BARDZO DOBRZE!!!":GOTO 20
50 IF A<=10 THEN ?"NIE OSZUKUJ!!!":GOTO 20
```

W liniach jest sprawdzana wartość wprowadzonej zmiennej A i w zależności od wyniku sprawdzenia wyświetlany jest odpowiedni komunikat.

POZNANE INSTRUKCJE:
GOSUB...RETURN,IF...THEN...

4.LEKCJA 4

W ATARI BASIC'u można wykonać wszystkie operacje matematyczne. Oznakowanie operacji jest trochę inne niż stosowane normalnie.Dodawanie i odejmowanie to + i -,mnożenie *, dzielenie /,potęgowanie ^(SHIFT i *).Relacje większości i mniejszości oznaczane są następująco:

A=B A jest równe B
A<B A jest mniejsze od B
A>B A jest większe od B
A<>B A nie jest równe B
A>=B A jest większe lub równe B
A<=B A jest mniejsze lub równe B

ZADANIE DOMOWE

Wprowadź poniższy program i zapisz go na kasecie.

```
5 REM OBLICZENIA MATEMATYCZNE
10 ?"PODAJ DWIE LICZBY A i B."
20 ?"B MUSI BYC WIEKSZE OD ZERA!"
30 INPUT A,B
40 IF B<=0 THEN ?"NIE OSZUKUJ!!!":GOTO 10
50 ?"SUMA A+B=";A+B
60 ?"ROZNICA A-B=";A-B
70 ?"ILOCZYN A*B=";A*B
80 ?"ILORAZ A/B=";A/B
90 GOTO 10
```

W linii 40 sprawdzana jest zmienna B (czy jest większa od zera-dalej jest dzielenie!).Jeżeli jest wszystko dobrze program wyświetla wyniki czterech działań na dwóch wprowadzonych liczbach.Instrukcja INPUT może wprowadzać więcej danych.

5.LEKCJA 5

Jeśli chcesz wykonać jakąś czynność pewną ilość razy to możesz zastosować instrukcję pętli:FOR K=n TO M:NEXT K

```
10 REM WYKORZYSTANIE PETLI
20 ?"ILE RAZY MAM NAPISAC KOMPUTER ";
30 INPUT A
40 FOR T=1 TO A
50 ?T;" KOMPUTER"
60 NEXT T
70 GOTO 20
```

Instrukcje zawarte między FOR...(linia 40) a NEXT...(linia 60) będą wykonywane tak długo,aż wartość zmiennej T będzie równa wartości zmiennej A.

Jeżeli chcemy aby program rozpoczynał działanie na czystym ekranie w początkowej linii musimy wprowadzić funkcję ?CHR$(n).Funkcja ta wyświetla na ekranie znaki,których kod podajesz jako n (n od 0 do 255).Niektóre z tych znaków mają specjalne znaczenie np.?CHR$(125)-czyści ekran,?CHR$(253)-wydaje pojedyńczy dźwięk.Funkcją odwrotną jest ASC.Wyświetla kod znaku np.? ASC("A").Dodaj do programu te trzy linie:

```
15 ?CHR$(125)
65 FOR K=1 TO 500:NEXT K
70 GOTO 15
```

Pusta instrukcja FOR...NEXT stanowi tzw.pętlę opóźniającą. Wykonanie 500 razy pętli trochę potrwa,a przez ten czas będzie można oglądać wyniki działania programu.Po tym czasie program skacze do linii 15 i czyści ekran.

POZNANE INSTRUKCJE:
FOR...TO...NEXT...,CHR$(n)

6.LEKCJA 6

Małe komputery ATARI posiadają cztery generatory dźwięku oznaczone numerami od 0 do 3.Uruchamia się je instrukcją SOUND:
SOUND nr.generatora,wysokość dźwięku,zniekształcenia,głośność
Każdy z parametrów może przyjmować wartości w pewnych granicach. Przekroczenie zakresu spowoduje błąd.
nr.generatora 0-3
wysokość dźwięku 0-255
zniekształcenia 0-15 (wartości parzyste),czysty dźwięk-10
głośność 0-15,0-cisza

Po wprowadzeniu instrukcji SOUND z prawidłowymi parametrami komputer zaczyna wydawać dźwięk.Oczywiście słychać go z głośnika monitora lub telewizora PAL.Wyłączenie generatora następuje po podaniu instrukcji SOUND z właściwym numerem i pozostałymi parametrami równymi 0:SOUND nr.,0,0,0

```
10 REM PIERWSZY DZWIEK
20 ?CHR$(125)
30 ?"PODAJ WYSOKOSC DZWIEKU (0-255):";
40 INPUT A
50 IF A<0 OR A>255 THEN 100
60 SOUND 0,A,10,10
70 FOR T=0 TO 200:NEXT T
80 SOUND 0,0,0,0
90 GOTO 20
95 REM KOMUNIKAT O BLEDNYCH DANYCH
100 ?CHR$(253)
110 ?"ZLE DANE !!!"
120 FOR T=0 TO 500:NEXT T
130 GOTO 20
```

W linii 50 przy sprawdzaniu warunku poprawności danych jest

zastosowana funkcja logiczna OR(LUB).Cały warunek wygląda więc tak:jeżeli A jest mniejsze od zera lub A jest większe od 255 to skocz do linii 100.
linia 60 uruchomienie generatora
linia 70 pętla opóźniająca(tyle czasu trwa dźwięk)
linia 80 wyłączenie generatora
linia 100-130 program obsługi błędu
Dane dla generatora można podawać także w programie:

```
5 REM GAMA
10 A=121:GOSUB 100
20 A=108:GOSUB 100
30 A=96:GOSUB 100
40 A=91:GOSUB 100
50 A=81:GOSUB 100
60 A=72:GOSUB 100
70 A=65:GOSUB 100
80 A=60:GOSUB 100
90 GOTO 10
95 REM PODPROGRAM DZWIEKOWY
100 SOUND 0,A,10,10
110 FOR T=0 TO 100:NEXT T
120 SOUND 0,0,0,0
130 RETURN
```

Trzeci parametr,zniekształcenia,umożliwia wydawanie dźwięków innych niż czyste.Są to różne szumy,warczenia,grzmoty.Wprowadź program i posłuchaj różnych efektów dźwiękowych.Możesz zapisać ciekawsze dane.Jak się znudzisz wciśnij RESET.

```
10 REM PELNY ZAKRES DZWIEKOW
20 FOR A=0 TO 255
30 FOR Z=0 TO 15
```

```
40 SOUND 0,A,Z,10
50 ?CHR$(125)
60 ?"SOUND 0,";A;",";Z;",10"
70 FOR T=0 TO 200:NEXT T
80 NEXT Z
90 NEXT A
```

Zobacz dokładnie jak są otwierane i zamykane pętle w instrukcjach FOR...NEXT.Zanim zamkniesz pętlę zewnętrzną (linia 90) musisz zamknąć pętlę,która jest w niej otwarta (linia 80).

POZNANE INSTRUKCJE:
SOUND,OR

7.LEKCJA 7.

Jeżeli chcemy umieścić dane w programie możemy zastosować instrukcję DATA.W czasie wykonywania programu komputer napotkawszy linie z instrukcją DATA ignoruje ją.Dopiero po napotkaniu instrukcji READ zaczyna poszukiwanie pierwszej linii danych.Dane są czytane kolejno.Za każdym wywołaniem instrukcja READ czyta następne dane.Aby powrócić na początek danych należy użyć instrukcję RESTORE.Spowoduje to odczyt danych od początku. Jeśli w programie jest kilka linii DATA można podać w instrukcji RESTORE numer linii,z której dane mają być odczytane.
Gdy danych zabraknie lub będziemy je chcieli odczytać ponownie bez użycia RESTORE,komputer zgłosi błąd numer 6.

Jeżeli chcemy wprowadzić instrukcją INPUT jakiś tekst należy zarezerwować dla tego tekstu miejsce w pamięci komputera, deklarując tzw.zmienną tekstową.Służy do tego instrukcja DIM. Deklarować możemy zmienne tekstowe:

```
DIM A$(20)
```

lub tablice numeryczne:

```
DIM TABLICA(5,10).
```

Zmienną tekstową stanowi ciąg znaków alfanumerycznych.Można na nim wykonywać różne operacje.Najpotrzebniejszą z nich jest możliwość ustalenia adresu zmiennej tekstowej.W tych zmiennych często umieszcza się podprogramy w języku maszynowym.Aby je uruchomić należy podać adres podprogramu.Służy do tego funkcja ADR(A$).
Programową obsługę błędów umożliwia instrukcja TRAP.Stosuje się ją przed instrukcją,w której może wystąpić błąd.Po instrukcji TRAP należy podać numer linii,do której ma skoczyć program po wykryciu błędu.

```
5 REM GAMA
10 DIM A$(1)
20 ?CHR$(125)
30 RESTORE
40 FOR T=1 TO 8
50 READ A
60 SOUND 0,A,10,10
70 NEXT T
80 SOUND 0,0,0,0
90 ?"CZY GRAMY JESZCZE RAZ T/N ";
100 INPUT A$
110 IF A$="T" THEN 20
200 DATA 121,108,96,91,81,72,65,60
```

POZNANE INSTRUKCJE:
DIM,DATA,READ,RESTORE,TRAP,ADR)

8.LEKCJA 8.

Niespodziankę w programie możemy zrobić stosując funkcję losową RND(0).Funkcja ta generuje liczbę w zakresie 0-0.9.Odpowiednie przetworzenie tej liczby daje możliwość utworzenia wartości w dowolnym zakresie:

INT(RND(0)*K) zakres losowania od 0 do K-1
INT(RND(0)*K)+N zakres losowania od N do K-1+N

Funkcja INT(n) przetwarza liczbę n na liczbę całkowitą.

Następną instrukcją skoku,o bardzo ciekawych możliwościach,jest instrukcja ON n GOTO nr,nr,nr,...lub ON n GOSUB nr,nr,nr,.... Wykonanie skoku do określonej linii uzależnione jest od wartości n.Dla n=1 skok do pierwszego nr.linii podanego po instrukcji,n=2 do drugiego numeru itd.

```
5 RZUT KOSTKA
10 DIM A$(1)
15 A=INT(RND(0)*6)+1
20 ON A GOTO 100,110,120,130,140,150,160,170
100 ?"JEDEN":GOTO 200
110 ?"DWA":GOTO 200
120 ?"TRZY":GOTO 200
130 ?"CZTERY":GOTO 200
140 ?"PIEC":GOTO 200
150 ?"SZESC":GOTO 200
200 ?"KONIEC T/N ";
210 INPUT A$
220 IF A$="N" THEN 15
```

9.LEKCJA 9.

Instrukcja POKE umożliwia wprowadzenie wartości z zakresu 0-255 do dowolnej komórki pamięci RAM.

POKE nr.komórki,wartość

Instrukcją odwrotną jest PEEK.Odczytuje ona wartość wybranej komórki RAM.

? PEEK(nr.komórki)

W komorce 764 komputer przechowuje kod ostatnio naciśniętego klawisza.Napisz krótki program i naciśnij kilka klawiszy.

```
5 REM KLAWIATURA
10 A= PEEK(764)
20 ? A
30 GOTO 10
```

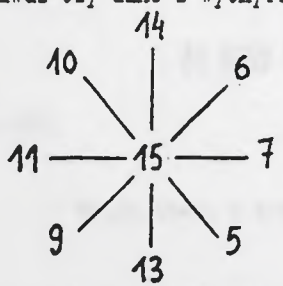
Kody klawiszy konsoli(START,SELECT,OPTION) odczytujemy z komórki 53279.Wprowadź poniższy program i naciskaj klawisze konsoli pojedynczo oraz ich kombinacje.Zapisz sobie wartości odpowiadające naciśniętym klawiszom.Może to się przydać w Twoich programach.

```
5 REM KONSOLA
10 A= PEEK(53279)
20 ? A
30 GOTO 10
```

Położenie joysticka odczytuje funkcja STICK(0) dla pierwszego portu i STICK(1) dla drugiego.

Napisz program poniżej i sprawdź czy dane z wychyleń zgadzają się z danymi na rysunku.

```
5 REM JOYSTICK
10 A= STICK(0)
20 ? A
30 GOTO 10
```



Naciśnięcie FIRE jest sprawdzane funkcją STRIG(0) i STRIG(1). FIRE pierwszego joysticka wciśnięty STRIG(0)=0,zwolniony STRIG(0)=1.

Wszystkie te instrukcje i komórki pamięci mogą być wykorzystane do sterowania w programie.Sterowanie z klawiatury może być wykonane za pomocą jednego klawisza (bez wciskania RETURN).

```
5 REM STEROWANIE DZWIEKIEM
10 W=100
15 ? "WLOZ JOYSTICK DO PIERWSZEGO PORTU."
20 ? "WYCHYLAJ W GORE I W DOL."
30 ST=STICK(0)
40 SOUND 0,W,10,10
50 IF ST=14 THEN W=W+1
60 IF ST=13 THEN W=W-1
70 IF W<0 THEN W=0
80 IF W>255 THEN W=255
90 ? W
100 GOTO 30
```

Powyższy program demonstruje możliwości sterowania dźwiękiem przy pomocy joysticka.

W linii 10 zmienna sterująca wysokością dźwięku przyjmuje wartość 100.W linii 30 zmiennej ST zostaje nadana wartość wychylenia joysticka.W liniach 50,60 następuje zmiana wartości zmiennej W w zależności od wychylenia joysticka.W.liniach 70,80 badana jest poprawność zmiennej W (czy mieści się w zakresie 0-255).

POZNANE INSTRUKCJE:
STICK,STRIG.POKE,PEEK

10.LEKCJA 10.

Komputer ATARI slynie z dobrej grafiki.Posiada 16 trybów graficznych.Możemy rozróżnić dwa podstawowe rodzaje trybów graficznych:bitowe i znakowe.Zajmiemy się tylko trzema trybami bitowymi 3,5 i 7.

Zmianę trybu możemy uzyskać stosując instrukcję GRAPHICS z odpowiednim numerem.Napisz GRAPHICS 3,wciśnij RETURN.Ekran zostanie podzielony na dwie części.W górnej będzie część graficzna w trybie 3,a w dolnej okienko tekstowe w trybie 0. Możemy tu wprowadzać instrukcje w trybie bezpośrednim.
Aby coś narysować musimy określić kolor punktu.Służy do tego instrukcja COLOR nr.W naszych trybach bitowych wartość nr może zawierać się w granicach od 0 do 3.Z tym,że COLOR 0 ustawi nam do rysowania kolor tła i rysunek będzie niewidoczny.

Po wybraniu koloru możemy narysować punkt.Umożliwi to nam instrukcja PLOT X,Y.Wartości X i Y muszą się mieścić w granicach danego trybu:

TRYB 3	X(0-39)	Y(0-23)(23)
TRYB 5	X(0-79)	Y(0-39)(47)
TRYB 7	X(0-159)	Y(0-79)(95)

W zastosowaniach programowych możemy wywolywać tryby bez okienka tekstowego dodając do numeru trybu wartość 16 (wartości Y w drugim nawiasie).Musimy pamiętać,że jako punkt zerowy został przyjęty punkt w lewym górnym rogu.

Instrukcją umożliwiającą kreślenie odcinków jest DRAWTO X,Y. Kreśli ona odcinek od ostatniego PLOT lub DRAWTO do podanych współrzędnych.Ograniczenia są takie same jak dla PLOT.Jaka będzie barwa rysowanego punktu zależy od ustawienia rejestrów kolorów instrukcją SETCOLOR.

SETCOLOR nr.rejestru,kod koloru,jasność

COLOR 1 nr.rejestru 0
COLOR 2 nr.rejestru 1
COLOR 3 nr.rejestru 2
COLOR 0 nr.rejestru 4

Kod koloru jest liczbą w zakresie 0-15.Wybieramy kolor z dostępnej palety ATARI.Jak ten kolor będzie wyglądał zależy od telewizora lub monitora.Np.8-niebieski,12-zielony,3-czerwony. Jasność -liczba parzysta w zakresie 0-14.Z tym,że 0 to jasność najmniejsza(prawie czarny),a 14 największa(prawie biały).
Kolory o tych samych jasnościach wyglądają na monitorze czarnobiałym tak samo.

```
5 REM KWADRAT
10 ?CHR$(125)
20 ? "PODAJ TRYB GRAFICZNY (3,5,7) ";
30 INPUT TR
40 IF TR<>3 AND TR<>5 AND TR<>7 THEN 10
50 GRAPHICS TR
60 ? CHR$(125)
70 ? "PODAJ X,Y LEWEGO,GORNEGO ROGU KWADRATU ";
80 INPUT X,Y
90 ? "PODAJ DLUGOSC BOKU ";
100 INPUT A
110 ? "PODAJ KOLOR ";
120 INPUT C
130 TRAP 200
140 COLOR C
150 PLOT X,Y:DRAWTO X+A,Y
160 DRAWTO X+A,Y+A:DRAWTO X,Y+A
170 DRAWTO X,Y
```

```
170 GOTO 60
200 ? CHR$(125):? CHR$(253)
210 ? "KURSOR PO ZA EKRANEM !!!"
220 FOR T=0 TO 300:NEXT T
230 GOTO 60
```

Rysowanie kwadratu następuje w liniach 150-170.Linie 200-230 obsługują sytuację błędną(gdy kwadrat nie zmieści się na ekranie).

POZNANE INSTRUKCJE:
GRAPHICS,COLOR,SETCOLOR,PLOT,DRAWTO

11.PODSUMOWANIE

W tych kilku rozdziałach zostały omówione podstawowe instrukcje ATARI BASIC'a oraz sposoby ich wykorzystania w programach.Stosując je możemy zacząć pisać programy własne. Należy stawiać sobie taki cel,z którym wiemy,że będziemy mieli kłopoty.Ale jak już program zacznie działać będzie duża frajda.

W rozdziale 13 zostało pokazanych kilka programów, które po uruchomieniu będą robiły coś ciekawego.Można się nimi pobawić,a następnie zanalizować co program robi i jak.Fragmenty tych programów można zastosować w swoich konstrukcjach.

W dalszych rozdziałach są omówione tryby znakowe ATARI i bardzo szerokie możliwości grafiki znakowej tego komputera.

Życzę wielu udanych,robiących niesamowite rzeczy programów. Pamiętaj żaden program nie zniszczy komputera!!!.Baw się i eksperymentuj.Bo po to masz komputer!

AUTOR

12.SYSTEM TURBO 2000

Wielu użytkowników ATARI XL,XE posiada magnetofony działające w systemie TURBO 2000.Jeżeli masz taki system i dodatkowo kartridż z oprogramowaniem,to masz właściwie wszystko co można wycisnąć z magnetofonu.Następnym krokiem jest zakup stacji dysków.

Niewątpliwą zaletą tego systemu jest znaczne zwiększenie szybkości transmisjii danych(stąd skrócenie czasu ładowania i zapisywania programów) oraz możliwość nadawania nazwy zapisywanym programom.Z tą ostatnią cechą wiąże się również umiejętność wyszukiwania przez system programu o podanej nazwie. Odpada konieczność ustawiania taśmy wg.wskazań licznika.

W systemie TURBO 2000 do współpracy z magnetofonem stosuje się instrukcje zapisu-odczytu normalnie używane przy pracy ze stacją dysków.

Zapis programu:

SAVE "D: NAZWA"

W cudzysłowie należy podać nazwę urządzenia z dwukropkiem D:,spację i dopiero potem nazwę programu.Nazwa może składać się z 10 znaków.

Odczyt programu:

LOAD "D: NAZWA" lub LOAD "D: *"

Jeżeli znamy dokładnie nazwę pod jaką zapisaliśmy nasz program to używamy instrukcji LOAD z pełną nazwą.W przeciwnym razie zastępujemy nazwę znakiem "*".System będzie kolejno sprawdzał nazwy programów,wyświetlał je na ekranie i prosił o decyzję, czy program załadować czy nie (T/N).

Załadowanie programu z uruchomieniem uzyskujemy stosując instrukcję:

RUN "D: NAZWA".

System załaduje program o podanej nazwie i uruchomi go. Instrukcje LIST i ENTER działają podobnie(LIST "D: NAZWA" i ENTER "D: NAZWA").

System TURBO 2000 jest bardzo wygodnym rozwiązaniem i może być polecany wszystkim użytkownikom magnetofonu.

13.PROGRAMY

Program INSTRUMENT pokazuje możliwości dźwiękowe ATARI. Z klawiatury można zagrać gamę(klawisze Q-I).Można wybierać różne barwy dźwięku(klawisze 1-7).Nową rzeczą jest sterowanie klawiszem HELP(komórka 732).Po włączeniu komputera jest w niej 0,a po wciśnięci HELP wartość 17.Z tym,że w tej komórce nie jest przywracana wartość początkowa.Wciśnięcie HELP+SHIFT daje wartość 81,a HELP+CONTROL 145.Aby przywrócić wartość początkową należy użyć instrukcji POKE 732,0.

```
10 REM INSTRUMENT
20 ?CHR$(125)
30 ?:?"------QWERTYUI------"
40 ?:?:?:?:?:?:?:?"START---WARTOSCI POCZATKOWE"
50 ?:?"OPTION--KROTKI DZWIEK"
60 ?:?"SELECT--DLUGI DZWIEK"
70 ?:?"1-7-----ZNIEKSZTALCENIA"
80 ?:?"HELP----DRUGI DZWIEK WLACZONY"
90 ?:?"HELP+SH-DRUGI DZWIEK WYLACZONY"
100 Z=10:S=1
110 K=PEEK(764)
120 IF K=47 THEN W=121:GOSUB 310
130 IF K=46 THEN W=108:GOSUB 310
140 IF K=42 THEN W=96:GOSUB 310
150 IF K=40 THEN W=91:GOSUB 310
160 IF K=45 THEN W=81:GOSUB 310
170 IF K=43 THEN W=72:GOSUB 310
180 IF K=11 THEN W=64:GOSUB 310
190 IF K=13 THEN W=60:GOSUB 310
200 IF K=31 THEN Z=2
210 IF K=30 THEN Z=4
```

```
220 IF K=26 THEN Z=6
230 IF K=24 THEN Z=8
240 IF K=29 THEN Z=10
250 IF K=27 THEN Z=12
260 IF K=51 THEN Z=14
270 IF PEEK(53279)=6 THEN Z=10:S=1:POKE 732,0
280 IF PEEK(53279)=5 THEN S=0.1
290 IF PEEK(53279)=3 THEN S=1
300 GOTO 110
305 REM DZWIEK
310 POKE 764,255
320 FOR G=15 TO 0 STEP -S
330 SOUND 0,W,Z,G
340 IF PEEK(732)=17 THEN SOUND 1,W/2,Z,G
350 NEXT G
360 SOUND 0,0,0,0:SOUND 1,0,0,0
370 RETURN
```

Program ZGADYWANKA jest prostą grą.Należy odgadnąć liczbę wylosowaną przez komputer.W programie zastosowano instrukcję END umożliwiającą zakończenie programu w każdym miejscu(linia 160).

```
5 REM ZGADYWANKA
10 REM LOSOWANIE
20 A=INT(RND(0)*10)+1
30 ?CHR$(125)
40 ?"ZGADNIJ MOJA LICZBA (1-10) ";
50 TRAP 30
60 INPUT B
70 IF A=B THEN 100
80 IF A<>B THEN 180
95 REM KOMUNIKAT PO TRAFIENIU
```

```
100 ?CHR$(125):GOSUB 250
110 ?"TRAFILES!!!"
120 ?:?:?"START---JESZCZE RAZ"
130 ?:?"SELECT-KONIEC"
140 S=PEEK(53279)
150 IF S=6 THEN 20
160 IF S=5 THEN END
170 GOTO 140
175 REM KOMUNIKAT PO NIETRAFIENIU
180 GOSUB 250
190 ?"NIETRAFILES!!!"
200 ?"ZGADUJ DALEJ!!!"
210 FOR T=0 TO 200:NEXT T
220 GOTO 50
245 REM DZWIEK
250 FOR T=0 TO 15
260 SOUND 0,50,10,T
270 NEXT T
280 SOUND 0,0,0,0
290 RETURN
```

Program KOSTKA pokazuje zastosowanie grafiki do prezentacji wyników losowania.Zwróć uwagę na sposób sterowania generatorem dźwięku(linia 340).

```
10 REM KOSTKA
20 GRAPHICS 3:GOSUB 140
30 FOR I=0 TO 10
40 K=INT(RND(0)*6)+1
50 GOSUB 340
60 COLOR 0:GOSUB 180:COLOR 1
70 ON K GOSUB 210,230,250,270,300,320
```

```
80 NEXT I
90 SOUND 0,0,0,0
100 ?CHR$(125):?"START JESZCZE RAZ"
110 IF PEEK(53279)=6 THEN 30
120 GOTO 110
130 REM RYSOWANIE KWADRATU
140 COLOR 2:PLOT 10,3:DRAWTO 24,3
150 DRAWTO 24,15:DRAWTO 10,15
160 DRAWTO 10,3:RETURN
170 REM KASOWANIE KOSTKI
180 COLOR 0:GOSUB 320:GOSUB 210
190 RETURN
200 REM JEDEN
210 PLOT 17,9:RETURN
220 REM DWA
230 PLOT 12,5:PLOT 22,13:RETURN
240 REM TRZY
250 GOSUB 230:GOSUB 210:RETURN
260 REM CZTERY
270 GOSUB 230:PLOT 12,13:PLOT 22,5
280 RETURN
290 REM PIEC
300 GOSUB 270:GOSUB 210:RETURN
310 REM SZESC
320 GOSUB 270:PLOT 12,9:PLOT 22,9
330 RETURN
340 SOUND 0,5*K,8,15:RETURN
```

Program OKRĄG pokazuje jeden ze sposobów kreślenia okręgu przy pomocy instrukcji PLOT.W ATARI BASIC'u nie ma instrukcji umożliwiającej wykreślenie na ekranie okręgu.Należy to wykonac

programowo.Linie 30-170 rysuje okrąg o środku w punkcie A,B i o promieniu R.

```
5 REM OKRAG
10 GRAPHICS 7:C=1:GOTO 200
20 TRAP 200:COLOR C
25 REM RYSOWANIE OKREGU
30 FI=0:Y1=0:X1=R
40 FIY=FI+Y1+Y1+1
50 FIXY=FIY-X1-X1+1
60 PLOT A+X1,B+Y1
70 PLOT A-X1,B+Y1
80 PLOT A+X1,B-Y1
90 PLOT A-X1,B-Y1
100 PLOT A+Y1,B+X1
110 PLOT A-Y1,B+X1
120 PLOT A+Y1,B-X1
130 PLOT A-Y1,B-X1
140 FI=FIY:Y1=Y1+1
150 IF ABS(FIXY)<ABS(FIY) THEN FI=FIXY:X1=X1-1
170 IF X1>=Y1 THEN 40
180 RETURN
195 REM WPROWADZANIE DANYCH
200 ?CHR$(125)
210 ?"PODAJ WSPOLRZEDNE SRODKA OKREGU ";
220 INPUT A,B
230 ?CHR$(125)
240 ?"PODAJ PROMIEN ";
250 INPUT R
260 ?CHR$(125)
270 ?OKRAG-PROMIEN ";R;"SRODEK ";A;",";B
```

```
280 GOSUB 20
290 IF PEEK(53279)<>6 THEN 290
300 C=C+1:IF C>3 THEN C=1
310 GOTO 200
```

Po podaniu danych wejściowych program kreśli okrąg.W przypadku podania danych,przy których kursor wyjdzie poza ekran instrukcja TRAP w linii 20 skieruje program do ponownego wprowadzania danych.Zwróć uwagę na sposób zatrzymania programu dla pokazania danych(linia 290).
Po drobnych modyfikacjach program może kreślić pełne koła,a właściwie koła wypełnione wzorkiem.Dodaj poniższe linie do programu:

```
15 FOR T=1 TO R
30 FI=0:Y1=0:X1=T
175 NEXT T
280 GOSUB 15
```

Dodatkowo można wprowadzić do linii 15 skok przy zmianie parametru:

```
15 FOR T=1 TO R STEP 2
```

Prosty PROGRAM GRAFICZNY umożliwia wykonanie rysunków w 5 trybie graficznym przy pomocy joysticka.Kolory wybiera się z klawiatury (1-4),z tym że 4 jest to kolor tła.W okienku tekstowym pojawia się aktualna wartość położenia kursora.Można tę wartość wprowadzić do własnych programów do instrukcji PLOT i DRAWTO.

```
10 REM PROGRAM GRAFICZNY
20 GRAPHICS 5:COLOR 1:X=10:Y=10:C=1:POKE 752,1
30 PLOT 0,0:DRAWTO 79,0:DRAWTO 79,39
40 DRAWTO 0,39:DRAWTO 0,0
50 ST=STICK(0)
60 K=PEEK(764)
```

```
70 COLOR 0:PLOT X,0:COLOR 1:PLOT X+1,0:PLOT X-1,0
80 COLOR 0:PLOT 0,Y:COLOR 1:PLOT 0,Y+1:PLOT 0,Y-1
90 IF ST=11 THEN X=X-1
100 IF ST=7 THEN X=X+1
110 IF ST=14 THEN Y=Y-1
120 IF ST=13 THEN Y=Y+1
140 IF K=31 THEN C=1
150 IF K=30 THEN C=2
160 IF K=26 THEN C=3
170 IF K=24 THEN C=0
180 IF ST=10 THEN X=X-1:Y=Y-1
190 IF ST=5 THEN X=X+1:Y=Y+1
200 IF ST=6 THEN X=X+1:Y=Y-1
210 IF ST=9 THEN X=X-1:Y=Y+1
230 IF Y>38 THEN Y=38
240 IF Y<1 THEN Y=1
250 IF X>78 THEN X=78
260 IF X<1 THEN X=1
270 IF STRIG(0)=0 THEN COLOR C:PLOT X,Y
280 ?CHR$(125):?"KURSOR X=;X;",Y=";Y
290 GOTO 50
```

Ten program pracuje w trybie 5.Można go oczywiście przerobić na pracę w innych trybach(3 i 7).Należy zmienić tryb graficzny w linii 20,rysowanie obrzeża ekranu w liniach 30,40 oraz ograniczenia wartości maksymalnych linie 230-260.

Kilka efektów dźwiękowych:

```
5 REM EFEKT 1
10 FOR CZ=4 TO 10
20 FOR GL=15 TO 0 STEP -1
30 SOUND 0,CZ,10,GL
```

```
40 NEXT GL
50 NEXT CZ
--------------------------
5 REM EFEKT 2
10 FOR GL=10 TO 0 STEP -0.1
20 SOUND 0,5,0,GL
30 NEXT GL
--------------------------
5 REM EFEKT 3
10 FOR CZ=10 TO 30
20 SOUND 0,CZ,10,10
30 NEXT CZ
40 FOR CZ=30 TO 10 STEP -1
50 SOUND 0,CZ,10,10
60 NEXT CZ
70 SOUND 0,0,0,0
--------------------------
5 REM EFEKT 4
10 FOR GL=0 TO 15
20 SOUND 0,2,0,GL
30 NEXT GL
40 FOR GL=15 TO 0 STEP -1
50 SOUND 0,2,4,GL
60 NEXT GL
```

Nasz ATARI posiada komórkę pamięci,w której losowo zmienia się wartość w granicach 0-255.Wartosc tę można zastosować tam gdzie jest potrzebne losowanie w tym zakresie.
Program poniżej pokazuje ciekawe zastosowanie tej komórki:

```
5 REM EFEKTY LOSOWE
10 GRAPHICS 7
```

```
20 FOR T=0 TO 30
30 POKE 712,PEEK(53770)
50 NEXT T
```

Kolor tła ekranu będzie się zmieniał w sposób losowy.Po dodaniu linii 40 i 60 będzie głośniej:

```
40 SOUND 0,PEEK(53770),10,10
60 SOUND 0,0,0,0
```

14.DLACZEGO GRAFIKA ZNAKOWA?

Nasz mały ATARI posiada bardzo rozbudowaną grafikę. Ma szesnaście trybów graficznych różniących się wielkością punktu(rozdzielczością) oraz ilością dostępnych kolorów.Tryby graficzne można podzielić na dwa podstawowe rodzaje:tryby znakowe i bitowe.

W trybach bitowych obraz na ekranie rysowany jest za pomocą pojedynczych punktów w określonym kolorze.W zależności od wielkości punktów ich ilość na ekranie może być różna.W tych trybach można wykonywać wielobarwne grafiki o dużej ilości szczegółów.Jeżeli zastosujemy odpowiednie oprogramowanie graficzne,jest to łatwe i przyjemne.Ale wykorzystanie rysunków w tych trybach we własnych programach napotyka na dwie trudności.Po pierwsze skomplikowany rysunek o wielu szczegółach jest rysowany na ekranie bardzo długo i potencjalny użytkownik programu straci do niego cierpliwość.Po drugie taki obrazek zajmuje bardzo dużo miejsca w pamięci RAM komputera.

W trybach znakowych informacja na ekranie pokazywana jest w postaci grupy punktów zwanych znakami.Mogą być to litery oraz inne znaki alfanumeryczne.Ale każdy z tych znaków może być dowolnie zmieniany przez użytkownika.W ten sposób można stworzyć nowe kroje pisma,znaki kartograficzne,symbole elektroniczne itp.

Zróbmy pierwszy eksperyment.W języku ATARI BASIC jest funkcja umożliwiająca sprawdzenie ilości wolnej pamięci RAM dostępnej dla programisty.Po włączeniu komputera piszemy w trybie bezpośrednim ? FRE(0).Otrzymana liczba(37902) jest ilością wolnych bajtów pamięci w trybie graficznym 0.Instrukcją GRAPHICS zmieniamy tryb graficzny na 7(piszemy GR.7).Otrzymujemy liczbę 34704,a w trybie 15 : 30782.W trybach znakowych odpowiednio: 1:38220,2:38470,12:37470,13:38230.Z naszego

eksperymentu wynika,że grafika całoekranowa w trybach znakowych zajmuje znacznie mniej pamięci niż grafika w odpowiednich im co do rozdzielczości trybach bitowych.

Pewną niedogodnością przy stosowaniu trybów znakowych do tworzenia grafiki jest konieczność starannego zaplanowania rysunku oraz stworzenia odpowiedniego zestawu znaków.Ale mając odpowiednie narzędzia można te prace wykonać bez większych problemów.I jeszcze mamy dodatkową premię.Jest nią możliwość stosowania jednego koloru więcej niż w trybach bitowych.

TRYBY ZNAKOWE

TRYB	KOLUMNY	WIERSZE(z oknem)	BEZ OKNA	KOLORY	RAM	GR.B
0	40	*	24	2	960	8
1	20	20	24	5	480	15
2	20	10	12	5	240	7
12	40	20	24	5	960	15
13	40	10	12	5	480	7

15.STEROWANIE W PROGRAMIE

Często istnieje potrzeba sterowania przebiegiem programu z klawiatury lub z klawiszy funkcyjnych.

Kod ostatnio wciśniętego klawisza możemy bardzo wygodnie odczytać z komórki o numerze 764.Napisz krótki program:

```
100 ? PEEK(764):GOTO 100
```

Po uruchomieniu programu(instrukcja RUN i wciśnięcie RETURN) na ekranie z góry na dół pojawi się wartość 255.Oznacza to,że nie był naciśnięty żaden klawisz.Wciskając interesujące Cię klawisze możesz poznać ich kod.Musisz zwrócić uwagę na to,że wartość w komórce 764 utrzymuje się do czasu wciśnięcia następnego klawisza.Należy po wykorzystaniu danego kodu wprowadzić do komórki wartość 255(instrukcją POKE 764,255).Wciśnięcie klawisza znakowego z klawiszem SHIFT podnosi wartość kodu o 64,z klawiszem CONTROL o 128 razem o 192.
Wykorzystanie w programie:

```
100 K=PEEK(764)
110 IF K=(kod1) THEN nr.linii1
120 IF K=(kod2) THEN nr.linii2
....
140 GOTO 100
```

Wciśnięcie odpowiedniego klawisza spowoduje skok do odpowiedniej linii programu bez wciskania RETURN.
Kody klawiszy konsoli odczytywane są w komórce 53279(START,OPTION,SELECT) i 732(HELP).

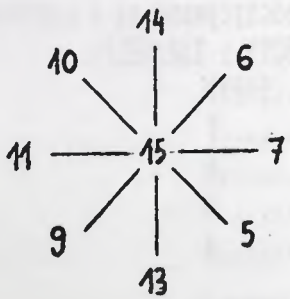
```
komórka.....53279
NIE WCISNIĘTY.....7
START.....6
SELECT.....5
OPTION.....3
```

START+SELECT.....4
START+OPTION.....2
SELECT+OPTION.....1
WSZYSTKIE.....0

Z komórką 732 sprawa ma się trochę inaczej.W komórce 53279 po zwolnieniu klawisza konsoli przywracana jest wartość początkowa 7.Natomiast w komórce 732 wartość początkowa ustawiana jest na 0 i po wciśnięciu i zwolnieniu klawisza HELP nie wraca do stanu początkowego.

WARTOSC POCZĄTKOWA.....0
HELP.....17
HELP+SHIFT.....81
HELP+CONTROL.....145

Wartość początkową można otrzymać używając instrukcji POKE 732,0
Sterowanie joystickiem jest wykorzystywane w programach graficznych,edytorach znaków no i oczywiście w grach.Nasz ATARI posiada dwa porty joysticków.Są to bardzo uniwersalne porty,ale na początku omówimy ich zastosowanie jako wejścia dla manipulatorów.Joystick można wychylać w ośmiu kierunkach.Funkcją,która odczytuje wartości przyporządkowane tym kierunkom jest STICK(0) dla joysticka nr.1 i STICK(1) dla joysticka nr.2.Wartości jakie przyjmuje ta funkcja są pokazane na rysunku.



Można też wykorzystać bezpośrednio instrukcję PEEK do odczytu stanu joysticków.Dla 1 632,dla 2 633.Napisz poniższy program,który pokaże Ci jak to działa:

```
100 ? STICK(0):GOTO 100
```

Po uruchomieniu programu pojawi się liczba 15 biegnąca z góry na dół.Jest to wartość przy położeniu spoczynkowym joysticka.Podłącz manipulator do 1 portu i wychylaj go na różne strony.Porównaj
wartości pojawiające się na ekranie do wartości na rysunku.Możesz zastąpić instrukcję STICK(0) instrukcją PEEK(632) i oczywiście zmienić numery joysticka.Nie zapomnij przełożyć go do drugiego portu.

Naciśnięcie FIRE można sprawdzić funkcją STRIG(0) lub STRIG(1):
FIRE wciśnięty.....STRIG(0)=0
FIRE zwolniony.....STRIG(0)=1

Jeżeli chcesz w programie zrobić jakąś niespodziankę,ale taką prawdziwą,bez możliwości przewidzenia kiedy ona wystąpi możesz skorzystać z generatora liczb losowych.Nasz ATARI słynie z bardzo dobrego generatora.Zastosowanie losowania nie wymaga chyba tłumaczenia.Losowanie zastępuje rzuty kostką,wybór zdarzenia,może symulować postępowanie przeciwnika.To oczywiście bardzo uatrakcyjnia każdy program.
ATARI BASIC dysponuje funkcją **RND(0)**,która odczytuje z generatora liczb losowych wartość w zakresie od 0 do prawie 1(0.9).Aby tą liczbę można było wykorzystać do sterowania należy ją przetworzyć na potrzebny nam zakres.Parametr funkcji,który podajemy w nawiasie nie ma wpływu na wynik losowania.
Chcesz wykonać komputerowy rzut kostką.Możemy napisać:

```
100 A=INT(RND(0)*6)+1:? A
```

W programie można to wykorzystać do sterowania w instrukcji

skoku warunkowego lub do losowania liczb w programie matematycznym.

```
100 A=INT(RND(0)*3)+1
110 IF A=1 THEN GOTO nr.linii
120 IF A=2 THEN GOTO nr.linii
130 IF A=3 THEN GOTO nr.linii
```

Ogólny wzór na zakres losowania:

```
A=INT(RND(0)*N)          A od 0 do N-1
A=INT(RND(0)*N)+K        A od K do N-1+K
```

Jednak wykonywane obliczenia w linii 100 trwają długo.Aby je uprościć można zastosować wartości z komórki 53770.Są tam losowo zmieniane liczby w zakresie od 0 do 255.Taki zakres pozwala na bezpośrednie sterowanie inną komórką pamięci.

Szybka,losowa zmiana kolorów tła:

```
100 POKE 712,PEEK(53770)
200 GOTO 100
```

Lub też generacja dźwięków losowych:

```
100 SOUND 0,PEEK(53770),10,10
200 GOTO 100
```

Możliwości zastosowania funkcji losowych w programach jest bardzo dużo.Dla przykładu można podać wszelkiego rodzaju gry karciane począwszy od pokera a skończywszy na brydżu.Jako prosty przykład wczytaj program ZGADYWANKA.Komputer losuje liczbę,a następnie prosi gracza o jej odgadnięcie.Podanie wartości za małej lub za dużej powoduje wyświetlenie odpowiedniego komunikatu.

16.PISZEMY KOLOROWO NA EKRANIE

Wywołanie trybu graficznego instrukcją GRAPHICS n umożliwi nam przeprowadzanie dalszych eksperymentów w trybie bezpośrednim.Wywołajmy tryb graficzny nr.2(pisząc GR.2).Ekran zostanie podzielony na dwie części.W dolnej części pojawia się okienko w trybie graficznym 0,a w części górnej mamy ekran w wywołanym przez nas trybie.

Aby napisać coś na ekranie musimy określić miejsce wyświetlania informacji instrukcją POSITION X,Y.Wartości X i Y nie mogą przekraczać maksymalnych wartości dla danego trybu;patrz tabelka na końcu rozdziału 2(i pamiętaj, że komputer liczy od 0).W przypadku przekroczenia tych wartości ATARI zgłosi błąd 141 kursor poza zakresem.

Następną czynnością jest wysłanie na ekran tego co chcemy na nim pokazać.W trybie 0 wystarczyła zwykła instrukcja PRINT(skrót PR. lub ?).Ale w trybach znakowych 1,2,12,i 13 należy posłużyć się bardziej rozbudowaną instrukcją print.Piszemy ? #6;" i to co chcemy pokazać".

np. POS.0,0:?#6;"A /A/ a /a/"

(litery /A/ i /a/ oznaczają wprowadzenie znaków w negatywie)

I teraz pytanie:co zobaczymy na ekranie po wprowadzeniu powyższej linii?.Na ekranie w pierwszej linii pojawią się cztery duże litery A,każda w innym kolorze.Piątym kolorem jest tło.

REJESTRY KOLOROW ORAZ SPOSOBY WPROWADZANIA ZNAKOW

Tło	rejestr	712
A (duża litera)	rejestr	708
/A/ (duża litera w negatywie)	rejestr	710(tło okienka)
a (mała litera)	rejestr	709(litery okienka)
/a/ (mała litera w negatywie)	rejestr	711

17.POROZMAWIAJMY O KOLORACH

Nasz ATARI dysponuje paletą 256 kolorów.Niestety praktyczne wykorzystanie tylu kolorów jest w prosty sposób niemożliwe.W trybach znakowych(z wyjątkiem trybu 0) możemy mieć na ekranie maksymalnie 5 kolorów.

ATARI BASIC posiada instrukcję do ustawiania odpowiedniej wartości w rejestrze kolorów.Jest to instrukcja

SETCOLOR nr.rejestru,kod koloru,jasność

Nr.rejestru : odpowiada mu odpowiednia komórka pamięci.
0 : komórka 708
1 : komórka 709
2 : komórka 710
3 : komórka 711
4 : komórka 712

Kod koloru : każdy kolor,który może być używany posiada odpowiedni kod.Jest to liczba w zakresie 0 do 15.

TABELA KODÓW KOLORÓW

szary.....0	zielony.....12
złoty.....1	żółtoniebieski.....13
pomarańczowy.....2	pomarańczowozielony.....14
czerwonopomarańczowy.....3	jasnopomarańczowy.....15
różowy.....4	
purpurowy.....5	
purpurowoniebieski.....6	
niebieski.....7	
niebieski.....8	
jasnoniebieski.....9	
turkusowy.....10	
zielononiebieski.....11	

Jasność jest wartością parzystą z przedziału od 0 do 14, z tym

że 0 jest to kolor najciemniejszy(prawie czarny),a 14 najjaśniejszy(prawie biały).

Instrukcję SETCOLOR można zastąpić instrukcją POKE. Instrukcja ta służy do wprowadzania danych bezpośrednio do komórki pamięci.Wartość wprowadzana może zawierać się w przedziale od 0 do 255.Instrukcją odwrotną jest instrukcja PEEK. Umożliwia ona sprawdzenie zawartości komórki pamięci o podanym numerze.Wywołaj tryb graficzny 2(pisząc GR.2) i napisz:POKE 712,14 .Tło ekranu zrobi się białe.Następnie:? PEEK(712).Pojawi się liczba 14.Jest to dana przechowywana w tej komórce pamięci.

Wartość jaką należy wprowadzić instrukcją POKE do rejestru koloru należy obliczyć z zależności:

POKE adres komórki,kod koloru*16+jasność

Należy pamiętać,że na ekranie telewizora zmiana jednego koloru wpływa na wszystkie inne barwy.Aby ustawić optymalne wartości w rejestrach kolorów skorzystaj z programu EDYTOR KOLORÓW.Po uruchomieniu na ekranie pojawią się numery rejestrów,wartości wprowadzone do tych rejestrów oraz rodzaj znaków,którymi opiekuje się dany rejestr.Sterowanie:joystick góra dół wybór rejestru,prawo lewo zmiana wartości w rejestrze.Jak już tak pomieszasz kolory,że nie będzie można nic zobaczyć na ekranie,naciśnij FIRE.Powrócisz do kolorów początkowych.

18.JAK WYGLĄDA ZNAK?

Znaki zbudowane są w matrycy 8x8 punktów.

1							
2	6	3	1				
8	4	2	6	8	4	2	1
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
1	1	1	1	1	1	1	1
0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0

16+8=24
16+8=24
128+64+32+16+8+4+2+1=255
32+4=36
32+4=36
32+16+8+4=60
16+8=24
0

Punkty znaku(bity) które są "zapalone" mają wartość 1.Wartości poszczególnych bitów od prawej do lewej wynoszą od 1 do 128.Sumując wartości bitów zapalonych z jednego rzędu otrzymujemy bajt.Osiem takich bajtów tworzy komplet danych jednego znaku.Tworzenie nowego znaku polega na odpowiednim ułożeniu bitów zapalonych i zgaszonych.

W trybach 1 i 2 każdy punkt w znaku może mieć tylko jeden kolor,z rejestru wybranego w zależności od sposobu zapisania znaku.Tryby 12 i 13 interpretują dane znakowe trochę inaczej.Czytają one bity parami i w zależności od wartości tej pary pobierają dane koloru z innego rejestru.W ten sposób w jednym znaku mogą być maksymalnie cztery kolory.

```
1 1 1 0 0 1 0 0
7 6 5 4 3 2 1 0
0 0                rejestr 712(kolor tła)
```

0 1 rejestr 708
1 0 rejestr 709
1 1 rejestr 710(znak w negatywie 711)

Przy wprowadzeniu znaku w negatywie punkt 1 1 odczytuje kolor z rejestru 711,co daje w sumie pięć kolorów na ekranie.Ale w jednym znaku mogą być tylko cztery kolory.

PROGRAM "PROSTY EDYTOR"

Program ten umożliwia rysowanie znaków w matrycy 8x8 punktów i przetwarzanie matrycy jednokolorowej na wielokolorową.Po narysowaniu można obejrzeć znak we wszystkich trybach znakowych oraz odczytać jego dane.

STEROWANIE W PROGRAMIE

N znaki w matrycy wielokolorowej normalnie
I znaki w matrycy wielokolorowej w negatywie
K kasowanie ekranu

Rysowanie joystick 1,kasowanie z fire.
Demonstracja znaków kolejno od góry w trybach:12,1,2,13.W okienku w trybie zerowym,a poniżej dane znaku.

Narysuj kilka znaków i zapisz ich dane.Będą nam potrzebne w następnych rozdziałach.

19.GDZIE SĄ ZNAKI I JAK JE ZMIENIC?

Dane znakowe są przechowywane w pamięci ROM od adresu 57344(224x256 tzn.od 224 strony pamięci).Danych w pamięci ROM nie damy rady zmienić.W związku z tym należy zestaw znaków przenieść w bezpieczne miejsce i tam dokonać potrzebnych zmian.Adres zestawu znaków ATARI przechowuje w komórce 756.Napisz ?PEEK(756) i otrzymasz liczbę 224.Jeżeli zestaw znaków przemieścimy w inne miejsce do komórki 756 musimy wprowadzić adres początku zestawu znaków.W komórce 106 ATARI przechowuje adres ostatniej dostępnej strony pamięci.Jeżeli wprowadzimy do tej komórki wartość mniejszą niż jest naprawdę,w miejsce które otrzymamy możemy bezpiecznie zapakować nasz zmieniony zestaw znaków.Cały zestaw zajmuje 1024 bajty(tzn.4 strony),ale miejsca należy zrobić trochę więcej.Tak na wszelki wypadek.Robimy miejsce na znaki:

```
CH:PEEK(106)-8:POKE 106,CH:GR.n
```

Po zrobieniu miejsca na znaki należy podać zawsze komendę GRAPHICS,aby komputer przepisał nowy adres do pamięci.CH jest to numer strony pamięci,od której będzie umieszczony nowy zestaw znaków.STARY:57344,NOWY:CHx256.

Przepisanie zestawu znaków można wykonać przy pomocy instrukcji PEEK i POKE,ale właśnie przepisywanie dużej ilości danych(1024 bity) jest słabą stroną ATARI BASIC'a.W programie "NOWE ZNAKI" przepisywanie zestawu znaków jest wykonywane przy pomocy procedury maszynowej umieszczonej w zmiennej tekstowej MM$(linia 15).Procedurę maszynową wywołujemy za pomocą instrukcji USR z odpowiednimi parametrami:

```
X=USR(ADR(MM$),STARY,NOWY,1024)
```

STARY.....adres starego zestawu znaków
NOWY.....adres nowego zestawu znaków

1024.....ilość bitów do przepisania
ADR(MM$)....funkcja podająca adres zmiennej MM$ w pamięci komputera

Aby zmienić znak w zestawie przepisanym do RAM należy znaleźć jego adres,a następnie wprowadzić nowe dane w miejsce starych.Przy zmianie kilku znaków najprostszym sposobem jest umieszczenie danych znakowych w liniach DATA,a znaki do przedefiniowania w zmiennej tekstowej.Na początku należy zadeklarować zmienną tekstową NZ$,a następnie wprowadzić do niej znaki,które mają być zmieniane(linia 10).Przedefiniowanie znaków linie 100,140.Dane znaków umieszczone są w liniach 150,156. Jeżeli masz przypadkowo dane znaków z poprzednich rozdziałów to możesz wprowadzić je do programu w liniach 150,153.Piszesz od nowa numer linii instrukcję DATA i wprowadzasz dane rozdzielone przecinkami:

150 DATA 255,255,255,255,255,255,255,255

Nie zapomnij,że danych musi być osiem.Po uruchomieniu programu zobaczysz na ekranie normalne znaki.Wciśnij START i zostaną pokazane znaki przedefiniowane,SELECT powrót do zestawu z ROM.Wciśnięciem START wprowadzasz do komórki 756 adres twojego zestawu znaków,a SELECT adres zestawu z ROM.Pierwsze cztery znaki mogą być takie jakie chcesz.Ale znaki w liniach 154,156 muszą być odpowiednio zaprojektowane.Służą one do prostej animacji.Polega ona na wyświetlaniu kolejno w tym samym miejscu znaków różniących się szczegółami.Daje to złudzenie ruchu.Jeżeli po wciśnięciu START wciśniesz A to trzy ostatnie znaki zostaną wykorzystane do animacji.Możesz stworzyć swój zestaw lub wykorzystać jeden z poniższych:

PIESEK

154 DATA 112,176,176,255,127,62,36,36

```
155 DATA 112,178,178,126,126,190,36,36
156 DATA 112,177,177,126,254,62,36,36
ROBOT 2
154 DATA 36,60,36,24,126,24,24,36
155 DATA 36,60,36,24,124,24,56,4
156 DATA 36,60,36,24,62,24,28,32
STEROWANIE W PROGRAMIE
START   zestaw znaków przedefiniowanych
SELECT  znaki z ROM
OPTION  zapis na taśmie zmiennej MM$ z deklaracją w linii 15
A       animacja trzech znaków(START  wyjście z animacji)
```

20.CO Z TYM MOZNA ZROBIC?

Teraz możesz spróbować zrobić jakiś ładny rysunek w grafice znakowej.Zapisz dane kilku znaków z PROSTEGO EDYTORA.Dobierz odpowiednie barwy stosując EDYTOR KOLORÓW.Nagraj procedurę przesuwu zestawu znaków z programu NOWE ZNAKI(wprowadzenie jej z klawiatury jest bardzo trudne).Jak przedefiniować znaki możesz dokładnie obejrzeć w programie NOWE ZNAKI.Do wprowadzenia znaków na ekran zastosuj instrukcję POSITION X,Y.Taki program może być stroną tytułową do większego programu lub ciekawym programem demonstracyjnym.Poniżej masz kilkanaście znaków opracowanych specjalnie do wykonania prostego rysunku.Dodatkowo możesz wykorzystać znaki PIESKA do animacji.

Pełna ściana:255,255,255,255,255,255,255,255
Sciana z oknem:255,255,195,195,195,195,255,255
Sciana z drzwiami:255,255,231,231,231,231,231,231
Góra dachu:24,24,60,60,126,126,255,255
Dach prawy:192,192,240,240,252,252,255,255
Dach lewy:3,3,15,15,63,63,255,255
Duża choinka:16,56,124,16,56,124,254,16
Mała choinka:0,0,0,16,56,124,254,16
Drzewko:16,56,92,116,44,24,16,16
Krzaczek:0,0,16,56,44,92,116,56
Płotek:0,0,85,255,85,85,255,85

Dość niezłe barwy możesz uzyskać wprowdzając do rejestrów kolorów następujące dane:708,20 709,198 710,150 711,36 712,162.
Nie zapomnij,że danych znakowych musi być koniecznie 8.

21.IDZIEMY KROK DALEJ.

Jeżeli chcemy w programie wykorzystać większą ilość przedefiniowanych znaków możemy mieć problem z wyświetlaniem komunikatów i danych liczbowych.Znaków wcale nie jest tak dużo.W trybach graficznych 1 i 2 wykorzystywane są tylko 64 znaki. Rozwiązaniem problemu jest przełączenie zestawu znaków. (program Przełączanie zestawu znaków).Rysunek w górnej części ekranu jest wykonany przy pomocy znaków przedefiniowanych (wg.danych z poprzedniego rozdziału).A w okienku tekstowym możesz zobaczyć ten sam rysunek,z tym że dane znaków są czytane z ROM(po przełączeniu zestawu w linii nad okienkiem tekstowym). Po wciśnięciu OPTION na taśmie zostanie zapisana procedura przełączania zestawu znaków(linie 175,285) oraz procedura przesuwu zestawu znaków z ROM do RAM.W linii 185 należy podać swoją zmienną NZ$,a w linii 255 dostosować ilość przedefiniowanych znaków do swoich potrzeb.Na początku programu należy zrobić miejsce na zestaw znaków i wywołać tryb graficzny(linia 15).Do praktycznego zastosowania najważniejsza jest możliwość sterowania miejscem przełączania zestawów.

Sterowanie następuje w linii 265.W komórkach pamięci o numerach 560,561 znajduje się adres programu wyświetlania.Jest to program,który mówi co,gdzie i jak ma być wyświetlane na ekranie.Modyfikując odpowiednie rozkazy programu wyświetlania możemy spowodować przełączenie zestawu znaków w odpowiedniej linii(wykonuje to program obsługi przerwań w zmiennej DL$). Linię 265 można odpowiednio zmodyfikować dostosowując do naszych wymagań:

```
265 DL=PEEK(560)+PEEK(561)*256:POKE DL+3+nr.linii,128+nr.trybu
```

Jako numer trybu należy podawać tryb graficzny procesora ANTIC(procesor graficzny ATARI).

TRYB BASICU	PRZEŁ.PRZY OKIENKU	TRYB ANTICA
0	24	2
1	24	6
2	14	7
12	24	4
13	14	5

Obserwując program "Przełączanie znaków,zmiana programu wyświetlania" można zauważyć,że na ekranie znajduje się kilka linii w GR.2 i kilka linii w GR.0.Jest to możliwe dzięki zmianie programu wyświetlania.

22.PROGRAM WYSWIETLANIA.

W komputerach ATARI XL,XE istnieje pewien problem.Jest to trudność z tworzeniem grafiki w różnych trybach graficznych na tym samym ekranie jednocześnie.Tworzeniem obrazu w ATARI zajmuje się specjalny procesor graficzny ANTIC.Ma swój zestaw rozkazów i odpowiednie programy,które są uruchamiane po wywołaniu każdego trybu graficznego.Adres programu wyświetlania jest zawarty w komórkach o numerach 560,561(młodszy i starszy bajt).Odpowiednia ingerencja w program wyświetlania umożliwia mieszanie trybów graficznych.Jeżeli masz na przykład pomysł na grę tekstowo graficzną to możesz stworzyć sobie ekran z pięcioma liniami w trybie 2 i tam pokazywać grafikę z przedefiniowanych znaków.oraz resztą w trybie 0.Tam będziesz umieszczał tekst po przełączeniu zestawu znaków na znaki z ROM.

Budowa programu wyświetlania na piechotę jest strasznie pracochłonna i wymaga wielu obliczeń.Ale program ZAPIS PROGRAMU WYSWIETLANIA umożliwi Ci dowolne zaprojektowanie ekranu.Są dwa ograniczenia:nie można stosować trybów graficznych 9,10,11 oraz nie można bardzo przekroczyć ilości linii stadardowego ekranu w trybie 0(24 linie).

ILOSC LINII ZAJMOWANA PRZEZ JEDNĄ LINIĘ DANEGO TRYBU GRAFICZNEO

GR.0	1linia
GR.1	1linia
GR.2	2linie
GR.3	1linia
GR.5	1/2linii
GR.7	1/4linii
GR.8	1/8linii
GR.12	1linia
GR.13	2linie

GR.15 1/8linii

Możliwe jest niewielkie powiększenie ekranu powyżej 24 linii GR.0.Np.5 linii GR.0,5 linii GR.1,10 linii GR.1,2 linie GR.2 i 3 linie GR.0.Razem daje to 27 linii GR.0 na całym ekranie.

Po uruchomieniu programu należy podać z ilu segmentów ma się składać ekran,a następnie wprowadzać kolejno tryby graficzne oraz ilości linii w tych trybach.Po podaniu wszystkich danych program wykona potrzebne obliczenia i po chwili możemy zobaczyć swoją kompozycję.Jeżeli jest zgodna z oczekiwaniami wciskamy START i nagrywamy na taśmę nasz własny program wyświetlania.Jeżeli coś nam nie odpowiada musimy zacząć od nowa wciskając SELECT.Takich programów można mieć kilka i używać wtedy kiedy są potrzebne.

Po zapisie programu wyświetlania na taśmie można skasować program w pamięci(instrukcj NEW) i załadować nagrany program(instrukcją ENTER "C:").

Co nagrało się na taśmę i jak to wykorzystać?Na taśmie nagranych jest sześć linii programu:

linia 95 deklaracja zmiennej tekstowej PR$ i umieszczenie w niej programu wyświetlania

linia 105 oblicza adres młodszego i starszego bajtu zmiennej PR$ i umieszcza go w komórkach 560,561

linia 115 deklaruje tablicę do przechowywania początków pamięci ekranów dla poszczególnych segmentów

linia 125 zatrzymuje program (z tej linii należy wykonać skok do dalszych części programu)

linia 135 dane dla linii 115

linia 155 podprogram obliczający młodszy i starszy bajt adresu segmentu i umieszczający go w komórkach 88,89

W pierwszych liniach programu musisz zrobić miejsce na zestaw

znaków,wywołać tryb graficzny i z linii 125 dokonać skoku omijając podprogram w linii 155.

```
15 CH=PEEK(106)-8:NOWY=CH*256:GR.2
125 GOTO 160
```

Aby wprowadzić informację do segmentu na ekranie należy:

1.Nadać segmentom kolejne numery zaczynając od zera.
2.Przypisać numer segmentu zmiennej X(X=nr.)
3.Wykonać skok do podprogramu w linii 155(GOSUB 155).
4.Wprowadzić do komórki 87 numer trybu graficznego danego segmentu(POKE 87,tryb).
5.Przy pomocy instrukcji POSITION wprowadzić informację do segmentu.

np.napis w segmencie nr.1 w trybie 0:

```
160 X=0:GOSUB 155:POKE 87,0
170 POSITION 0,0:?#6;"segment nr.1"
```

Dla wprawy proponuję zbudowanie krótkiego programu demonstracyjnego,zbudowanego w oparciu o posiadane przez nas informacje.Załaduj program "Zapis przełączania znaków" i zapisz na taśmie podprogram przełączenia znaków.Następnie załaduj program "Zapis programu wyświetlania" i zbuduj swój ekran np. 6 lini GR.2 i 10 lini GR.0.Nagraj program wyświetlania na taśmę za programem przełączania.Potem wyłącz komputer,włącz go ponownie i załaduj oba programy instrukcją ENTER"C:".
Z lini 125 wykonaj skok do lini 175.W linii 185 zadeklaruj swoją zmienną NZ$ oraz wprowadź tę zmienną.

```
185 DIM NZ$(2):NZ$:"QW"
```

W liniach 225-245 wprowadź przedefiniowanie znaków.

```
225  FOR I=1 TO 2:ADRES=CH+(ASC(NZ$(I))-32)*8
235  FOR J=0 TO 7:READ A:POKE ADRES+J,A
245  NEXT J:NEXT I
```

Dane znakowe wprowadź od linii 300.

```
300 DATA 255,255,255,255,255,255,255,255
302 DATA 255,129,129,129,129,129,129,255
```

Od linii 500 wprowadź swoje komunikaty.(patrz program "Przełączanie znaków,zmiana programu wyświetlania")

23.CO ZROBIC ABY BYLO COS SLYCHAC?

Do tej pory nasze programy byly ciche.Bardziej zajmowaliśmy się grafiką.Ale dźwięk jest też ważną częścią dobrego programu.Nasz ATARI ma cztery generatory dźwięku.

Generator uruchamia się instrukcją SOUND (SO.) z odpowiednimi parametrami:

```
SOUND nr.generatora,wysokość,barwa,głośność
nr.generatora       0,1,2,3
wysokość            od 0 do 255
barwa       wartości parzyste od 0 do 14(10 dźwięk czysty)
głośność            wartości od 0 do 15
np.SOUND 0,100,10,12
```

Generator po uruchomienie wytwarza dźwięk do czasu wyłączenia go instrukcją np.SOUND 0,0,0,0 lub instrukcją END.Można oczywiście w czasie pracy generatora zmieniać jego parametry.Spowoduje to zmianę dźwięku.Wprowadź programy podane dalej.Zobaczysz jakie efekty dźwiękowe może generować Twój komputer:

dźwięk rosnący szybko lub trochę wolniej

```
100 FOR T=0 TO 255        100 FOR T=0 TO 255 STEP 0.1
110 SOUND 0,T,10,10        pozostale linie bez zmian
120 NEXT T
130 SOUND 0,0,0,0
```

dźwięk malejący szybko lub trochę wolniej

```
100 FOR T=255 TO 0 STEP -1      100 FOR T=255 TO 0 STEP -0.1
```

pozostałe linie jak w programie powyżej

Kilka dźwięków:

```
100 FOR T=0 TO N:REM N ile razy chcemy usłyszeć
110 SOUND 0,100,10,10
120 FOR K=0 TO 20:NEXT K
130 SOUND 0,0,0,0
```

```
140 NEXT T
```

Jeżeli chcesz usłyszeć inne ciekawe możliwości uruchom program EFEKTY DZWIĘKOWE.Możesz wybierać z pośród dziesięciu różnych efektów.

Aby zagrać jakąś melodię należy kolejno podawać dane dla generatora zmieniając wysokość dźwięku.Dane można umieścić w liniach DATA i odczytywać instrukcją READ.

```
100 FOR T=0 TO 9
110 READ X,P
120 SOUND 0,X,10,10
125 FOR K=0 TO P:NEXT K
130 NEXT T
140 SOUND 0,0,0,0
150 DATA ....
```

W programie powyżej w linii 150 należy wprowadzić 10 par danych.Pierwsza liczba z pary steruje wysokością dźwięku,a druga czasem trwania.To już może być uznane za prostą melodię.

Dla większych znawców muzyki przeznaczony jest program MUZYKANT.Posiada on procedury maszynowe umożliwiające zagranie melodii,której dane wprowadza się do zmiennej tekstowej.Dokładny opis programu w menu.Dwukrotne wciśnięcie RETURN spowoduje zagranie melodii Smerfów.Po wprowadzeniu DEMO program zagra kawałek blusa.Procedury maszynowe zawarte w liniach 10,16 można dołączyć do własnych programów.Sposób wywołania procedury podany w linii 500.

Aby zobaczyć co można jeszcze wydobyć z Twojego ATARI uruchom program EDYTOR DZWIĘKÓW.Jest to program działający na zasadzie programu EDYTOR KOLORÓW.Na ekranie kolejno z góry na dół pojawiają się instrukcje SOUND dla poszczególnych generatorów.Sterowanie zmianami poszczególnych parametrów odbywa

się z klawiatury.Pierwsza linia(klawisze cyfrowe) generator 0: 1,2 częstotliwość,3,4 zniekształcenia,5,6 głośność.Odpowiednio klawisze z linii niższych dla pozostałych generatorów.START włączenie dźwięku SELECT wyłączenie.Zmiany parametrów można dokonywać w czasie trwania dźwięku.

Program ten umożliwi Ci sprawdzenie wpływów jednego dźwięku na drugi.Możesz dobierać sobie dowolne efekty dźwiękowe i ich dane zastosować w swoim programie.

Aby zagrać gamę C dur należy kolejno podać następujące wartości: C=121,D=108,E=96,F=91,G=81,A=72,H=64,C=60.

Trochę inny sposób na zagranie melodii pokazuje program poniżej.W liniach danych podawana jest wysokość i czas trwania dźwięku.W linii 150 wartość pauzy mnożona przez podany na początku podprogramu współczynnik steruje pętlą opóźniającą.Zmiana współczynnika umożliwia zmianę szybkości odtwarzania melodii bez zmiany rytmu.

```
100 T=10:RESTORE
110 READ A,B
120 IF A=-1 THEN SOUND 0,0,0,0:RETURN
130 SOUND 0,A,10,10
140 FOR P=1 TO B*T
150 GOTO 110
160 DATA 0,32,53,16,40,8,0,4,53,4,47,8,60,8,72,16,53,8,64,8
170 DATA 61,8,64,8,72,16,0,8,53,16,40,8,0,4,53,5,47,8,60,8
180 DATA 72,16,53,8,64,8,60,8,72,8,81,16,0,16,-1,-1
```

Odczytanie danych ujemnych powoduje wyjście z podprogramu w linii 120.

24.CO UMIEMY ZROBIC?

Przez te kilka rozdziałów nauczyliśmy się dużo ciekawych rzeczy.Umiemy pisać kolorowo na ekranie,budować własne znaki i komponować ekrany do ich wyświetlania.Dla urozmaicenia programu nauczyliśmy się tworzyć efekty dźwiękowe i proste melodyjki.

Dojście do tych wszystkich wiadomości zajęło mi dużo więcej czasu niż potrzeba na przeczytanie tych kilkunastu rozdziałów.

Teraz spróbujemy praktycznie wykorzystać nabyte umiejętności. Na początku zbuduj program wyświetlania składający się z siedmiu linii w GR.2,trzech linii w GR.1 i ośmiu linii w GR.0.Następnie opracuj kilkanaście znaków,z których będzie można zbudować zamek(w środku i na zewnątrz),jakieś drzewka,różne cegły,okna,wieże itp.Przygotuj program przedefiniowania tych znaków i przełączenia zestawu w pierwszej linii GR.1.Dodaj do tego jakeś efekty dźwiękowe i prostą melodyjkę na zakończenie(może być też inna na rozpoczęcie).Jak to wszystko przygotujesz to w ostatnim rozdziale tej części spróbujemy zrobić prostą grę przygodową.Wielkość i komplikacja tej gry będą zależały tylko od twojej fantazji.

W następnej części książki zajmiemy się budowaniem dużych plansz wieloekranowych,grafiką graczy i pocisków,animacją już w pełnym tego słowa znaczeniu oraz zbudujemy program graficzny(do rysowania w grafice znakowej).

25.PROSTA GRA

Mając narzędzia opisane w poprzednich rozdziałach możemy zbudować prostą grę tekstowo-graficzną.Musimy mieć jakiś pomysł na fabułę.Następnie zrobić plan pomieszczeń i zaludnić je odpowiednimi stworzeniami.Dalszą czynnością będzie zbudowanie programu sterowania w oparciu o plan pomieszczeń(skąd się wychodzi i dokąd trafia).Efekty graficzne i dźwiękowe to już sprawa dość prosta.Nasza gra będzie tylko prostym przykładem.

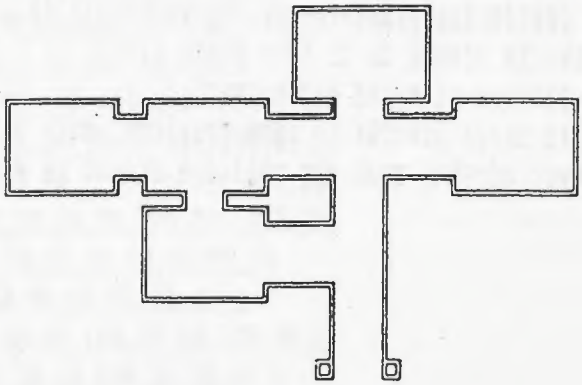
Można ją dowolnie rozbudować,wprowadzić wiele postaci,nowe pomieszczenia oraz dużo ciekawych zdarzeń losowych.To co zrobimy wystarczy do poznania możliwości naszego komputera,a rozbudową programu(a najlepiej budową nowej gry od początku) możesz zająć się po zrozumieniu działania poszczególnych elementów programu.

25.1.LEGENDA I PLAN PODZIEMI.

Zaczynamy od legendy(opisu gry).Można skorzystać z gotowych bajek,baśni,zdarzeń autentycznych lub stworzyć całkiem coś nowego.My napiszemy własną legendę o zamku na wzgórzu.

* Na wzgórzu stoją ruiny zamku.Podobno są w nich ukryte skarby.Pilnuje ich jednak straszny Diabeł Fujara z całą zgrają ohydnych potworów.Postanowiłeś zdobyć te skarby.W drodze spotkałeś starego człowieka,który wspomógł Cię dobrą radą.Powiedział:"Weź ze sobą gwizdek,lustro i butelkę z wodą." Skorzystałeś z tej rady i wyruszyłeś w drogę. *

To byłaby nasza legenda.Teraz należy opracować plan ruin zamku.Proponuję dość prosty układ pięciu komnat,z których dwie są przechodnie.



25.2.ZALUDNIAMY RUINY.

Jak już mamy gdzie chodzić możemy rozejrzeć się za mieszkańcami zamku.Będą oni stanowić główną atrakcję naszej wędrówki.Na początek proponuję tylko czterech.Są to:

1.Duch Wyjec jest to okropny hałaśnik.Przestraszyć go może tylko ktoś głośniejszy.Musisz użyć gwizdka.

2.Bazyliszek jak wiadomo poraża wzrokiem.Jako broni używasz lusterka.

3.Szkielet jest bardzo suchy i strasznie trzeszczy.Nie lubi wilgoci.Można go przegonić wodą z butelki.

4.Diabeł Fujara pilnuje skarbów(śpi na nich) Możesz go spotkać tylko wtedy gdy zdobędziesz przynajmniej sześć punktów.Obok Fujary leżą trzy przedmioty:worek złota,miecz i czapka.Musisz zabrać mu czapkę.

Używa jej do zakrywania rogów gdy wyskakuje do pobliskiej gospody na piwo.Za czapkę da Ci tyle złota ile zgromadziłeś punktów za walkę z potworami.

Jak widzisz nasze stworki to same przyjemniaczki.Dla każdego musisz opracować odrębny znak aby mógł się ukazać na ekranie.

25.3.Z CZEGO BUDOWAC?(ZESTAW ZNAKOW)

Przykladowy zestaw znaków,który można wykorzystać do budowania zamku(na zewnątrz i w środku) jest podany poniżej.Razem ze znakami z rozdziału CO Z TYM ZROBIC? umożliwiają wykonanie grafiki do naszej gry.Możesz oczywiście wykonać własny zestaw lub wykorzystać ich mniej.

cegła 0,223,223,223,0,251,251,251
cegła odwrotna 255,32,32,32,255,4,4,4
góra wieży 219,219,219,255,126,122,106,110
dół wieży 126,122,122,94,94,122,122,126
kwadrat1 126,189,219,231,231,219,189,126
kwadrat2 255,129,189,165,165,189,129,255
kwadrat3 255,153,189,231,231,189,153,255
mur 0,0,0,238,238,238,255,255
ściana z okienkami 255,255,143,143,255,241,241,255
gwizdek 0,0,30,255,62,0,0,0
lustro 0,60,126,126,126,126,126,60
butelka 102,36,36,66,66,126,126,60
czapka 0,24,60,126,126,126,255,255
worek 126,60,24,60,126,255,255,255
miecz 16,16,16,16,16,16,56,16
duch 14,56,84,40,56,28,14,3
szkielet 60,90,126,36,24,195,36,24
diabeł 36,36,36,126,90,36,24,0
bazyliszek 126,153,153,255,102,90,66,231

25.4.ZACZYNAMY.

Na początku zbuduj program wyświetlania:siedem linii GR.2,trzy linie GR.1,osiem linii GR.0.Nagraj go na taśmie a zaraz po nim program przełączania zastawu znaków.Skasuj program,który miałeś ostatnio w pamięci i wprowadź kolejno dwa nagrane programy(instrukcją ENTER"C:").W linii 265 zamiast POKE DL+14 napisz POKE DL+10.Spowoduje to przełączenie zestawów znaków nad pierwszą linią w GR.1.Teraz możesz wprowadzić deklarację zmiennej tekstowej NZ$(25) i wprowadzić znaki do przedefiniowania:NZ$:"QWERTYUIOPASDFGHJKLZXCVBN".(są do tego przeznaczone odpowiednie linie z instrukcją REM).Kolejność wprowadzania znaków jest ważna tylko dlatego,że grafikę,którą zrobimy,będzie można wykorzystać bez zmian w Twoim programie.

linia 15 miejsce na zestaw znaków
linia 35 wyłączenie ekranu
linia 55 kolory
linia 75 deklaracja zmiennych tekstowych,ustawianie zmiennych siły,wyposażenia,zerowanie punktów
linia 95-135 program wyświetlania
linia 155 adres pamięci ekranu(dla segmentu)
linia 175-205 przepisywanie znaków
linia 225-245 przedefiniowanie znaków
linia 265-285 przełączanie znaków
linia 300-348 dane znakowe
linie 500,585 rysowanie strony tytułowej
linia 595 wywołanie segmentu 1
linia 605 wywołanie segmentu 2
linia 615 wywołanie segmentu 3
linia 625 kasowanie segmentu 1
linia 645 kasowanie segmentu 2

linia 665 kasowanie segmentu 3
linia 685-695 sprawdzanie kodu klawiatury
linia 705 pętla opóźniająca(długa)
linia 710 pętla opóźniająca dźwiękowa
linia 730-1630 pętla główna(chodzenie po komnatach)
linia 1640-1710 spotkanie z potworami
linia 1720-1835 walka z potworami
linia 1845-1875 złe zakończenie
linia 1885-1925 rozprawa z Fujarą
linia 1935-2015 dobre zakończenie
linia 2020-2090 obliczanie ilości sprzętu
linia 2100-2135 efekty dźwiękowe Bazyliszka
linia 2140-2165 efekty dźwiękowe Szkieleta
linia 2170-2210 efekty dźwiękowe Wyjca
linia 2215-2550 efekty dźwiękowe Fujary
linia 2265 rysowanie ściany poziomej z drzwiami
linia 2275 rysowanie ściany poziomej bez drzwi
linia 2285 rysowanie ściany pionowej bez drzwi
linia 2305-2310 rysowanie ściany pionowej z drzwiami
linia 2320-2330 komnata z jednymi drzwiami
linia 2340 losowanie(rzut kostką)
Dane znakowe wprowadzimy od linii 300 ze skokiem 2:

1.pełna ściana	300	znak Q
2.dach prawy	302	znak W
3.dach lewy	304	znak E
4.duża choinka	306	znak R
5.mała choinka	308	znak T
6.drzewko	310	znak Y
7.cegła	312	znak U
8.cegła odwrócona	314	znak I

9.góra wieży	316	znak O
10.dół wieży	318	znak P
11.kwadrat 1	320	znak A
12 kwadrat 2	322	znak S
13 kwadrat 3	324	znak D
14 mur	326	znak F
15 ściana z okienkami	328	znak G
16 gwizdek	330	znak H
17 lustro	332	znak J
18 butelka	334	znak K
19 czapka	336	znak L
20 worek złota	338	znak Z
21 miecz	340	znak X
22 duch	342	znak C
23 szkielet	344	znak V
24 diabel	346	znak B
25 bazyliszek	348	znak N

25.5.PIERWSZY RYSUNEK.

Aby coś można było zobaczyć narysujemy teraz widok tytułowego Zamku na Wzgórzu.Rysowanie zaczniemy od linii 500.Będzie to strona początkowa naszej gry.Zawsze będzie pokazywana po uruchomieniu programu i dlatego nie umieścimy jej w podprogramie.Gwiazdki symbolizują spacje,a podkreślenia znaki wprowadzane w negatywie.

```
500 X=0:GOSUB 200:POKE 87,2
502 POS.0,0:?#6;"********O":POS.0,1:?#6;"********P***O"
504 POS.0,2:?#6;"********P***P":POS.0,3:?#6;"*r***FFQFFFQFF**r*y"
506 POS.0,4:?#6;"**r*QQQQaQQQQ":POS.0,5:?#6;"y**rEQQQQQQQQQWy"
508 POS.0,6:?#6;"**EQrQQrQQQrQQQQW*r":POKE 559,34
```

W linii 500 mamy ustalenie w którym segmencie ekranu będziemy pisać(bo to nasze rysowanie jest właściwie pisaniem).W pozostałych liniach mamy znaki ,które utworzą nasz zamek. Na początku programu,przed rozpoczęciem przedefiniowywania znaków możemy wprowadzić instrukcję POKE 559,0.Zostanie wyłączony ekran.Odciążony komputer będzie mógł znacznie szybciej wykonywać obliczenia związane z umieszczaniem naszych znaków.Po narysowaniu obrazka włączmy ekran(linia 508).

25.6.BUDUJEMY GRĘ DO KOŃCA.

Każde z pomieszczeń naszego zamku może składać się z czterech lub dwóch ścian.Każda ściana może mieć drzwi lub nie.

Z tego wynika,że do narysowania dowolnego pomieszczenia wystarczy umiejętność narysowania czterech ścian (pionowa z drzwiami i bez drzwi,pozioma z drzwiami i bez drzwi).Z tym,że należy podać komputerowi gdzie ma narysować te ściany.
Rysowanie ścian wykonują podprogramy w liniach 2265-2310.Dla ścian pionowych należy podać parametr SX (dla ściany prawej równy 6 dla lewej równy 13),a dla ścian poziomych SY (dla ściany górnej równy 0,a dla dolnej równy 6).Pomieszczenia,z wyjątkiem komnaty z jednym wejściem rysujemy przy pomocy tych podprogramów.Dla komnaty z jednym wejściem mamy osobny podprogram,bo są trzy jednakowe pomieszczenia i rysowanie każdego z nich oddzielnie byłoby zbędną stratą czasu i pamięci komputera(linie 2320-2330).

Do rysowania można zastosować dowolny znak.Tu zastosowana jest cegła pełna(ściany) U i kwadrat1 /a/(drzwi).
Kod wciśniętego klawisza jest badany podprogramem w linii 685,a wartość kodu przypisywana jest zmiennej ST.Każde wciśnięcie sygnalizowane jest dźwiękiem(czy wciśnięcie powoduje reakcję czy nie).

W czasie pisania programu warto wprowadzić w lini o wysokim numerze instrukcje sprawdzające kod wciśniętego klawisza.Każde odwołanie do tej linii(instrukcją GOTO nr.linii) umożliwi sprawdzenie kodu bez szukania w tabelach.
np.20000 ?PEEK(764):GOTO 20000

Po ukończeniu programu należy tę linię skasować.
Po lewej stronie pierwszego segmentu są pokazywane nasze zasoby oraz klawisz,który należy wcisnąć aby użyć danego

przedmiotu.Spotkanie z potworem jest sygnalizowane odpowiednim efektem dźwiękowym i pokazaniem potwora wewnątrz komnaty.

W drugim segmencie wprowadzana jest nazwa potwora oraz opis klawiatury.Po walce pojawia się komunikat oraz stan naszej siły i punkty.W przypadku gdy siła spadnie do zera kończymy grę("złe" zakończenie linia 1845).Gdy zbierzemy powyżej pięciu punktów możemy spotkać Diabła Fujarę.Musimy zabrać mu odpowiedni przedmiot(linia 1885-1925).Jak trafimy kończymy grę("dobre" zakończenie linia 1935-2015).W przypadku złego wyboru musimy szukać Diabła dalej,a w międzyczasie spotykać inne potwory.

Podstawą naszej gry jest pętla główna,w której poruszamy się odpowiednio po pomieszczeniach zamku.Przyjmujemy założenie,że zawsze stoimy twarzą do przodu i znajdujemy się w dolnej części ekranu.Powoduje to konieczność rysowania pomieszczenia w zależności od kierunku ruchu(np.poruszamy się korytarzem do przodu i mamy drzwi po lewej stronie,a gdy wracamy tym samym korytarzem drzwi są po stronie prawej).

Grafikę można oczywiście bardziej rozbudować(jest jeszcze kilka znaków do wykorzystania),można wprowadzić możliwość znajdowania przedmiotów w zamku.Ale to już są propozycje na nowy program.

26.KOLOROWA GRAFIKA W TRYBIE 0.

Tryb graficzny 0 jest trybem monochromatycznym.Ale stosując odpowiednie procedury maszynowe możemy wprowadzić kolory do tego trybu.Programy przedstawione poniżej demonstrują te możliwości.

Pierwszy program umieszcza w dolnej części ekranu 128 kolorów.

```
10 FOR T=1536 TO 1554
20 READ A:POKE T,A:NEXT T
30 POKE 39980,130:POKE 512,0:POKE 513,6:POKE 54286,192
40 DATA 72,138,72,162,0,142,10,212,142,24,208,232
50 DATA 232,208,246,104,170,104,64
```

Po umieszczeniu kolorów na ekranie komputer kontynuuje normalną pracę.Możesz program wylistować,dopisać dalszą część.Instrukcja GRAPHICS lub wciśnięcie RESET przerywa działanie procedury.

Program drugi umieszcza na całej powierzchni ekranu kolorową tęczę ATARI.

```
10 FOR T=1536 TO 1562
20 READ A:POKE T,A:NEXT A
30 POKE 39970,240:POKE 512,0:POKE 513,6:POKE 54286,192
50 DATA 72,138,72,152,72,166,200,160,192,141,10,212,142
60 DATA 24,208,232,136,208,246,230,200,104,168,104,170,104,64
```

Oba programy należą raczej do grupy programów prezentujących efekty upiększające stosowane głównie w stronach tytułowych programów lub programach demonstracyjnych.Program trzeci jest całkiem poważnym programem użytkowym.Umożliwia umieszczenie w każdej linii ekranu w grafice 0 dowolnie wybranego koloru znaków,tła i ramki ekranu.

```
10 GRAPHICS 0:DL=PEEK(560)+PEEK(561)*256
20 POKE DL+2,240:POKE DL+3,194:FOR T=DL+6 TO DL+28:POKE
   T,130:NEXT T
30 FOR T=1536 TO 1600:READ A:POKE T,A:NEXT T
```

```
40 DATA 8,72,138,72,166,203,189,65,6,141,10,212,141,23,208,189,
   66,6,141,24,208,189,67,6,141,26,208,232,232,232
50 DATA 224,75,208,2,162,0,134,203,104,170,104,40,64
60 DATA 169,138,141,36,2,169,194,141,37,2,169,0,133,203,169,192,
   141,14,212,76,138,194
69 REM KOLORY LITER
70 FOR T=1601 TO 1672 STEP 3:READ A:POKE T,A:NEXT T
79 REM KOLORY TLA
80 FOR T=1602 TO 1672 STEP 3:READ A:POKE T,A:NEXT T
89 REM KOLORY RAMKI
90 FOR T=1603 TO 1672 STEP 3:READ A:POKE T,A:NEXT T
100 DATA 14,0,0,0,0,0,14,14,14,14,0,0,0,0,0,8,8,8,8,8,8,8,8,0
110 DATA 0,198,198,198,198,198,50,50,50,50,14,134,134,134,134,
   0,0,0,0,0,0,0,0,14
120 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

W liniach 100,110,120 umieszczone są dane początkowe kolorów odpowiednio dla znaków,tła i ramki.Dane te są wprowadzane do odpowiednich komórek na stronie 6 pamięci.W czasie działania programu można te wartości zmieniać.Spowoduje to zmianę odpowiedniego koloru w wybranej linii.

	KOMÓRKI ODPOWIADAJĄCE ZA KOLORY		
LINIA	ZNAK	TŁO	RAMKA
1	1601	1602	1603
2	1604	1605	1606
3	1607	1608	1609
4	1610	1611	1612
5	1613	1614	1615
6	1616	1617	1618
7	1619	1620	1621
8	1622	1623	1624

9	1625	1626	1627
10	1628	1629	1630
11	1631	1632	1633
12	1634	1635	1636
13	1637	1638	1639
14	1640	1641	1642
15	1643	1644	1645
16	1646	1647	1648
17	1649	1650	1651
18	1652	1653	1654
19	1655	1656	1657
20	1658	1659	1660
21	1661	1662	1663
22	1664	1665	1666
23	1667	1668	1669
24	1670	1671	1672

EDYTOR KOLORÓW W TRYBIE ZEROWYM

Jest to program,który umożliwia ustawienie dowolnych kolorów w grafice zerowej.Można ustawić dowolny kolor tła w każdej linii,dowolny kolor ramki w każdej linii oraz dowolną jaskrawość znaków w linii w kolorze wybranego tła.W sumie w każdej linii można uzyskać trzy kolory co razem daje 24*3=72 kolory na całym ekranie.Zmiany linii uzyskuje się joystickiem góra-dół.Zmiana kolorów joystick prawo-lewo,po wciśnięci FIRE zmiany koloru są szybsze.Gdy kursor jest w pierwszej linii można dokonać wyboru zakresu zmiany koloru(z klawiatury):T-kolor tła,R-kolor ramki, L-kolor znaków.Po zmontowaniu ekranu wg.potrzeb lub odczuć estetycznych można zapisać go na taśmie dla całego ustawienia kolorów(w liniach DATA).Aby wykorzystać te dane we własnym programie należy dopisać linie z procedurą przełączania kolorów

(linie 10-60 z programu powyżej).Kolory ustalone w liniach DATA można oczywiście zmieniać w czasie działania programu wykorzystującego przełączanie kolorów,wprowadzając wartości bezpośrednio do odpowiednich komórek pamięci(1601-1672).

Wykorzystanie tych możliwości w praktyce możesz zaobserwować w programie CIĘŻARÓWKA.Program ten nie zawiera żadnych dodatkowych trików.Można dodać do niego przedefiniowane znaki i więcej dźwięku,ale chodziło głównie o pokazanie sposobu przełączania kolorów.

linia 100 skok do opisu i wyboru trudności
linia 110-290 wprowadzenie procedury przełączania kolorów
linia 300 losowanie miejsca rozpoczęcia gry
linia 350 deklaracja tablic
linia 360-400 wprowadzenie danych do tablicy odległości
linia 430-470 ładowanie magazynów "towarem" (losowanie)
linia 480-760 pętla główna
linia 770-810 zakończenie gry
linia 820-1040 rysowanie planszy
linia 1050-1210 opis i wybór trudności
linia 1220-1290 efekty dźwiękowe
linia 1300-1460 wyróżnienie magazynu, w którym jest gracz

W tym programie otrzymujesz pracę kierowcy wielkiego samochodu przeworzącego kontenery między magazynami. Rozmieszczenie magazynów pokazuje mapa w górnej części ekranu. Wg.niej musisz planować trasę,bo paliwo drogo kosztuje.Poniżej masz swój pojazd z ładunkiem.W części dolnej ekranu są pokazane magazyny z kontenerami.Musisz wybierać dokąd jedziesz i co zabierasz.Wynikiem gry jest zysk na jeden kurs.
Szerokiej drogi!!!

27.ROZBUDOWANY EDYTOR ZNAKOW.

Edytor znaków opisany w rozdziale osiemnastym miał pewną wadę:dane znaków należało ręcznie przepisać z monitora i wprowadzić do własnego programu.Ale tę czynność bardzo łatwo może za nas wykonać komputer.W tym edytorze znaki wpisywane są do programu od linii 300 dla znaków w trybach 1 i 2 i od linii 400 dla trybów 12 i 13.Dla wykorzystania znaków we własnych programach nie ma znaczenia w jakiej linii są ich dane. Ale na pewno wiesz,że ten sam znak będzie wyglądał zupełnie inaczej w trybach 1 i 2 i 12 i13.Program graficzny KOLOROWY BLOK RYSUNKOWY może pracować w każdym z tych trybów graficznych.Aby można było wykonywać rysunki posiada dwa zestawy znaków:od linii 300 dla gr.1 i 2 oraz od linii 400 dla gr.12 i 13.

STEROWANIE W PROGRAMIE

Rysowanie znaków---joystick 1(kasowanie z FIRE)
Z---zapis danych znaków w linii DATA na taśmie
1---demonstracja znaków w GR.1 i 2
2---demonstracja znaków w GR.12 i 13
K---kasowanie
D---druk danych(jeśli masz drukarkę)
W---wpis danych do programu

PRACA Z EDYTOREM

Po uruchomieniu programu należy podać,w którym trybie graficznym chcemy definiować znaki.Jest to ważne,jeśli tworzysz zestaw znaków dla programu graficznego.

Po narysowaniu znaku należy go obejrzeć w wybranym trybie graficznym wciskając 1 dla GR.1 i 2 lub 2 dla GR.12 i 13. Jak znak nam odpowiada wpisujemy go do programu , wciskając W. Jak nie-możemy go poprawić i ponownie wcisnąć 1 lub 2.A jeżeli całkiem nam nie pasuje kasujemy go wciskając K.Ilość znaków

wpisanych do programu pojawia się obok napisu EDYTOR ZNAKÓW.

Po zdefiniowaniu wszystkich potrzebnych znaków zapisujemy cały ich zbiór na taśmie wciskając Z.Na taśmie zostaną zapisane linie danych znakowych w zależności od trybu jaki zadeklarowaliśmy na początku tzn.dla grafiki 1 i 2 od linii 300,a dla grafiki 12 i 13 od linii 400.Zbiór znaków możemy dograć do programu graficznego (lub każdego innego programu) instrukcją ENTER"C:". Dlatego jest ważna decyzja w którym trybie graficznym będzie pracował nasz zestaw znaków.

28.PROSTA TECHNIKA GRACZY.

W programie NOWE ZNAKI była możliwość prostej animacji wykonanej przy pomocy odpowiednio zdefiniowanych znaków.Animację w miejscu można wykonać tą metodą dosyć prosto,ale ruch obiektu na ekranie nie wypada najlepiej.

Do celów animacji w ruchu zostały stworzone w ATARI specjalne możliwości.Jest to tzw.technika graczy i pocisków.My zajmiemy się prostszą odmianą techniki graczy.

CO TO JEST GRACZ?

Gracz jest to obiekt ekranowy o szerokości ośmiu punktów (jak znak) i wysokości maksymalnej do 256 punktów.Jest to bardzo dużo i w całości nie mieści się na ekranie.Dla celów praktycznych wystarczają gracze do dwudziestu punktów.Ale szerokość graczy może być dodatkowo zwiększona dwukrotnie i czterokrotnie.Samych graczy może być czterech.Jest wprawdzie sposób aby dodać piątego,ale tym już nie będziemy się zajmować.

Ruch w poziomie nie jest żadnym problemem ponieważ są odpowiednie rejestry ruchu dla każdego gracza i wprowadzenie tam wartości ustawia gracza w odpowiednim miejscu ekranu.Poziome pozycje graczy są ustawiane w rejestrach 53248-53251.Gorzej jest z ruchem w pionie.Tu,chyba przez niedopatrzenie konstruktorów ATARI,nie ma odpowiedniego rejestru.Problem ten obchodzi się stosując różne procedury w języku maszynowym.Potraktujemy sprawę trochę prościej,a efekty nie będą dużo gorsze.

Ale na początku załaduj program DEMO GRACZY.Będziesz miał okazję do zapoznania się z kilkoma rodzajami ważnych komórek służących do sterowania techniką graczy.Rejestry 53256-53259 odpowiadają za szerokość graczy.Wprowadzenie do nich 0-szerokość normalna, 1-szerokość podwójna,3-szerokość poczwórna.

Zmiany szerokości graczy dokonujemy z klawiatury.Pierwszy

rząd(od 1) steruje pierwszym graczem,drugi (od Q) drugim itd.
2,W,S,X-szerokość normalna
3,E,D,C-szerokość podwójna
4,R,F,V-szerokość poczwórna

Po uruchomieniu programu aktywnym graczem (tzn.mogącym się poruszać) jest gracz 1.Joystickiem możemy przemieszczać gracza po całym ekranie.W celu wykonania ruchu następnym graczem należy go uaktywnić wciskając pierwszy klawisz w jego rzędzie(1,Q,A,Z).

Uaktywnienie następnego gracza powoduje przeniesienie poprzedniego na jego miejsce początkowe.Do rejestrów pozycji poziomej można,tak jak do każdej komórki pamięci wprowadzić wartość od 0 do 255.Ale gracz jest widoczny całkowicie na ekranie w zakresie od 50 do 200(w zależności od szerokości). Następną komórką widoczną dla programisty jest komórka ustalenia priorytetu tzn. co będzie wyświetlane na pierwszym planie przy spotkaniu na ekranie.Jest to komórka 623.Sterowanie z klawiatury
5-1 wszyscy gracze,wszystkie kolory
6-2 gracze 1,2,wszystkie kolory,gracze 3,4
7-4 wszystkie kolory,wszyscy gracze
8-8 kolory 0,1,wszyscy gracze,kolory 2,3
9-32 tryb mieszania kolorów
0-49 tryb mieszania kolorów

Powprowadzaj różne wartości do tej komórki i poruszaj graczami po ekranie.W trybie mieszania kolorów można uzyskać dodatkowe kolory przy nałożeniu się na siebie dwóch graczy lub gracza i rysunku wykonanego znakami z określonego rejestru koloru.Kolory graczy ustawia się w rejestrach 704-707 wg.znanego wzoru:
POKE rejestr,kod koloru*16+jasność.

REJESTRY KOLIZJI

Gracze mogą być przemieszczani po całym ekranie.Mogą

nakładać się na rysunki i na innych graczy.Aby można było kontrolować te spotkania zostały wprowadzone rejestry kolizji.Po spotkaniu gracza z innym graczem lub nałożeniu się gracza na rysunek do odpowiedniego rejestru jest wprowadzana wartość umożliwiająca jednoznaczne określenie z czym spotkał się gracz. Gdy gracz spotkał się z kilkoma obiektami wartości są sumowane. Natomiast kilkakrotne spotkanie z tym samym obiektem nie wprowadzi większej wartości niż wynikająca z pojedynczej kolizji.Wartości w tych rejestrach nie powracają do stanu początkowego po przerwaniu kontaktu między obiektami.Aby je wyzerować należy wprowadzić do komórki 53278 jakąkolwiek wartość np.POKE 53278,255.

WARTOSCI W REJESTRACH KOLIZJI

KOLIZJE MIĘDZY GRACZAMI

	GRACZ1	GRACZ2	GRACZ3	GRACZ4
GRACZ1(53260)	0	2	4	8
GRACZ2(53261)	1	0	4	8
GRACZ3(53262)	1	2	0	8
GRACZ4(53263)	1	2	4	8

KOLIZJE MIĘDZY GRACZEM A RYSUNKIEM Z OKRESLONEGO REJ.KOLORU

	708	709	710	711
GRACZ1(53252)	1	2	4	8
GRACZ2(53253)	1	2	4	8
GRACZ3(53254)	1	2	4	8
GRACZ4(53255)	1	2	4	8

Poznaliśmy już komórkę pamięci 559.Służyła nam do wyłączania ekranu.Ale ma ona bardzo ważne zastosowania w technice graczy.

Każdy z graczy może mieć określoną wysokość punktu tzw.rozdzielczość:jednowierszową-punkt gracza ma wysokość punktu w grafice 15 i dwuwierszową-punkt jak w grafice 7.Wartość 46 w

tej komórce ustawia rozdzielczość dwu-,a 62-jednowierszową.

Rejestr 53277 steruje włączaniem techniki graczy 0-wyłącza 3-włącza.To jest wkład konstruktorów ATARI,a co my z tym zrobimy to już nasza sprawa.Zrobić można bardzo dużo.Jako pierwszy przykład obejrzyj program DEMO ZIEMIA-KSIĘŻYC.Jest tam pokazany ruch graczy bez animacji.Obiekty ruchome składają się z dwóch graczy każdy.

Po tym dość długim wstępie zajmiemy się zbudowaniem konstrukcji umożliwiającej umieszczenie na ekranie czterech graczy.Nasza technika będzie miała jedno ograniczenie - gracze mogą być tylko w rozdzielczości dwuwierszowej(POKE 559,46).

ZACZYNAMY!

Każdy gracz musi mieć zarezerwowane miejsce w pamięci komputera. W naszej prostej technice gracze zostaną umieszczeni w zmiennych tekstowych.Aby można było te zmienne łatwo znaleźć należy je zadeklarować na początku programu zaraz po włączeniu komputera. Jeżeli już coś pisałeś i chcesz zająć się techniką graczy musisz wyłączyć komputer,włączyć ponownie i wprowadzić pierwszą linię:

```
1 DIM GR1$(128),GR2$(128)
```

Następnie musisz ustalić jak wysoki będzie gracz i w jakich kierunkach będzie się poruszał i o ile punktów na jeden ruch:

ROZMIAR GRACZA=WYSOKOSC+PRĘDKOSC W GORĘ + PRĘDKOSC W DÓŁ

Przyjmijmy,że nasz gracz będzie miał wysokość 8 punktów i będzie się poruszał z prędkością dwóch punktów w każdą stronę. Deklarujemy zmienną tekstową o odpowiedniej wielkości:

```
2 DIM G$(12)
```

Zauważ różnicę:w linii 1 robiliśmy miejsce dla gracza w pamięci, a w linii 2 deklarujemy miejsce na dane gracza.Można oczywiście przygotować dowolną ilość danych graczy.Dodatkowo należy zadeklarować zmienną wypełnioną samymi zerami.Służy ona do

kasowania obrazu gracza na ekranie.Zwróć uwagę na szybki sposób wypełnienia zmiennej tekstowej stałą wartością(linia 3).

```
3 DIM C$(128):C$=CHR$(0):C$(128)=C$:C$(2)=C$
10 DATA 0,0,255,129,129,129,129,129,129,255,0,0
```

Teraz możemy wprowadzić dane gracza do odpowiedniej zmiennej:

```
30 FOR T=1 TO 12:READ D:G$(T,T)=CHR$(D):NEXT T
```

Następnie musimy wprowadzić trochę niezbędnych obliczeń w celu wskazania komputerowi gdzie i jak ma szukać danych dla graczy i gdzie ma je pokazywać.

```
100 A=4*(INT(PEEK(742)/4)-1)
110 POKE 54279,A
120 VSA=PEEK(134)+PEEK(135)*256
130 BOA=PEEK(140)+PEEK(141)*256
140 PM=256*A+512
150 OBR=PM-BOA
160 ADD=2
170 FOR T=1 TO 2
180 PMSTAR=INT(OBR/256)
190 PMMLOD=OBR-256*PMSTAR
200 POKE VSA+ADD,PMMLOD
210 POKE VSA+ADD+1,PMSTAR
220 OBR=OBR+128:ADD=ADD+8
230 NEXT T
240 POKE 559,46:POKE 53277,3
```

W linii 170 musisz wstawić wartość odpowiadającą używanej ilości graczy(w naszym przypadku 2).Jeżeli używasz tylko jednego gracza możesz zlikwidować linie 170,220 i 230.W linii 240 ustawiamy rozdzielczość techniki graczy i włączamy ją.

Jeszcze kilka linii i gracze ukażą się na ekranie:

```
400 GR1$=C$:GR2$=C$
```

```
410 POKE 704,14:POKE 705,0
420 POKE 53249,80:GR2$(80)=G$
430 X=80:Y=80:GOTO 540
440 ST=STICK(0):IF ST=15 THEN 440
450 IF ST=7 THEN X=X+2
460 IF ST=11 THEN X=X-2
470 IF ST=14 THEN Y=Y+2
480 IF ST=13 THEN Y=Y-2
490 IF X<50 THEN X=50
500 IF X>200 THEN X=200
510 IF Y<20 THEN Y=20
520 IF Y>110 THEN Y=110
530 GR1$(Y)=G$:POKE 53248,X
540 ? CHR$(125):?"X=";X,"Y=";Y
550 GOTO 440
```

W wyniku działania tego programu otrzymamy na ekranie nieruchomy,czarny kwadrat oraz biały kwadrat poruszający się we wszystkich kierunkach(joystick1).

29.ANIMACJA.

Ruch gracza na ekranie nie jest jeszcze animacją.Prawdziwa animacja polega na szybkim nakładaniu na siebie kilku obrazów różniących się między sobą szczegółami i przemieszczonych w stosunku do siebie w kierunku ruchu.W naszej prostej technice graczy jest możliwe wykonanie bardzo rozbudowanej animacji, składającej się z wielu faz.Ale dla celów praktycznych całkowicie wystarczy nam animacja czterofazowa.

Na początku załaduj program ANIMACJA NA JEDNYM EKRANIE.Po uruchomieniu zobaczysz uśmiechniętą buzię stworka PACKERA z gry M.U.L.E.(jednej z lepszych gier na małe ATARI).Stworek może poruszać się we wszystkich kierunkach i w każdym kierunku wygląda inaczej.Przebiera też szybko nóżkami.Poruszając stworkiem zauważysz na pewno,że przy ruchu w górę i w dół są 3 różne fazy,a w prawo i w lewo tylko dwie.Aby wykonać tego typu animację musisz przygotować wszystkie fazy ruchu swojej postaci: trzy w górę,trzy w dół i po dwie w lewo i w prawo.Razem daje to dziesięć postaci.Należy zadeklarować odpowiednią ilość zmiennych tekstowych od F1$ do F10$.Dane graczy są umieszczone bezpośrednio w zmiennych tekstowych przy pomocy EDYTORA GRACZY (opis w dalszych rozdziałach).Sterowaniem wyświetlania odpowiedniej fazy zajmuje się zmienna FAZA.Jest jeszcze potrzebna pomocnicza zmienna tekstowa AN$.Przy każdej zmianie współrzędnej X lub Y jest sprawdzana zmienna FAZA i w zależności od jej wartości zmienna pomocnicza AN$ przyjmuje wartość odpowiedniej fazy ruchu (linie 510-600).Zmienna FAZA jest zwiększana o jeden.Gdy jej wartość przekroczy 4 zostaje zmieniona na 1.W tym miejscu można zwiększyć wartość zmiennej FAZA i przygotować więcej faz ruchu.W ten sposób można uzyskać bardziej płynną animację.

Część programu sterowania animacją pokazana jest poniżej:

```
380 ST=STICK(0):IF ST=15 THEN 380
390 IF ST=11 THEN X=X-1:ON FAZA GOSUB 510,520,510,520
400 IF ST=7 THEN X=X+1:ON FAZA GOSUB 530,540,530,540
410 IF ST=13 THEN Y=Y-1:ON FAZA GOSUB 550,560,550,570
420 IF ST=14 THEN Y=Y+1:ON FAZA GOSUB 580,590,580,600
430 IF X<XI THEN X=XI:REM XI WARTOSC MIN X
440 IF X>XM THEN X=XM:REM XM WARTOSC MAX X
450 IF Y<YI THEN Y=YI:REM YI WARTOSC MIN Y
460 IF Y>YM THEN Y=YM:REM YM WARTOSC MAX Y
470 FAZA=FAZA+1:IF FAZA>4 THEN FAZA=1
480 POKE 53248,X:PM1$(Y)=AN$
490 FOR T=0 TO 10:NEXT T
500 GOTO 380
509 REM RUCH W LEWO
510 AN$=F7$:RETURN
520 AN$=F8$:RETURN
529 REM RUCH W PRAWO
530 AN$=F9$:RETURN
540 AN$=F10$:RETURN
549 REM RUCH W GORE
550 AN$=F1$:RETURN
560 AN$=F2$:RETURN
570 AN$=F3$:RETURN
579 REM RUCH W DOL
580 AN$=F4$:RETURN
590 AN$=F5$:RETURN
600 AN$=F6$:RETURN
```

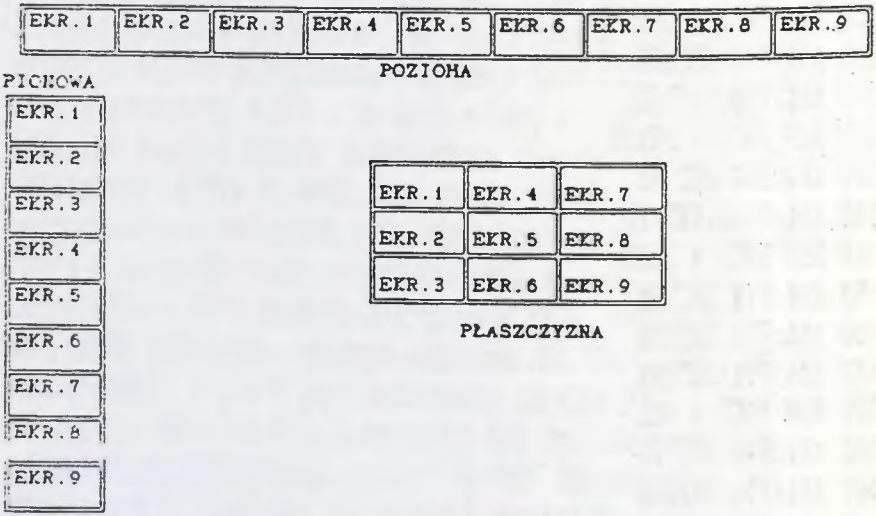
Jeśli chcesz zobaczyć jak szybka może być animacja zlikwiduj linię 490(pętla opóźniająca).

30.BUDUJEMY DUŻY EKRAN.

Masz pomysł na grę lub inny program,który nie mieści się na jednym ekranie.W ATARI to nie jest problem.W bardzo prosty sposób można zrobić miejsce na kilka ekranów graficznych w pamięci komputera i przełączać je w odpowiedniej chwili.Tu właśnie możemy wykorzystać zalety trybów znakowych.Zajmują bardzo mało pamięci RAM i takich ekranów można śmiało umieścić w programie kilkanaście.Zajmiemy się teraz budowaniem planszy wieloekranowej złożonej z dziewięciu ekranów.

W zależności od tego w jaki sposób można będzie poruszać się po naszej planszy będziemy ją nazywać :pionową,poziomą lub płaszczyzną.Dodatkowo musimy określić czy przy dojściu do ostatniego ekranu wraca się na pierwszy czy nie.

PRZYKŁADOWE KONFIGURACJE EKRANÓW



Aby zrobić miejsce w pamięci komputera na naszą planszę musimy znaleźć szczyt pamięci RAM:

MEM=PEEK(742)
oraz obliczyć adres programu wyświetlania:
DL=PEEK(560)+PEEK(561)*256
Adresy poszczególnych ekranów otrzymujemy odejmując od szczytu pamięci MEM liczbę odpowiadającą ilościom stron pamięci zajmowanej przez ekran w danym trybie graficznym.My będziemy odejmowali cztery strony co wystarczy dla każdego trybu znakowego:
EKR1=MEM-10:EKR2=MEM-14:EKR3=MEM-18:...EKR9=MEM-38
Tak to by ładnie wyglądało.Ale w praktyce często bywa tak,że po przełączeniu ekranu nie można na nim pisać (lub można tylko w kilku górnych wierszach),pojawiają się jakieś "śmieci",których usunięcie powoduje takie zawieszenie programu,że często nawet RESET nie pomaga.Oznacza to,że trafiliśmy naszym ekranem w obszar pamięci zajmowany przez nasz własny program lub system operacyjny komputera.Często więc wartości odejmowane od zmiennej MEM muszą być dobierane eksperymentalnie(szczególnie dla dużych programów wykorzystujących własne znaki i technikę graczy).
Przykładowa linia z programu KOLOROWY BLOK RYSUNKOWY określająca 10 ekranów(można je używać w trybach 1,2,12,13):
01=MEM-10:02=MEM-16:03=MEM-21:04=MEM-25:05=MEM-32:06=MEM-36:
07=MEM-40:08=MEM-44:09=MEM-48:010=MEM-52

Przełączenie ekranu otrzymujemy po przypisaniu zmiennej pomocniczej ADRES wartości adresu odpowiedniego ekranu i wykonanie prostego podprogramu:
POKE DL+5,ADRES:POKE 89,ADRES:POKE 106,ADRES+4
KIEDY PRZEŁĄCZAĆ EKRANY?
Podczas pruszania się stworek dochodzi do brzegu ekranu.Jedna ze współrzędnych osiąga wartość maksymalną lub minimalną.Jeżeli jest to animacja na jednym ekranie stworek staje i w miejscu

przebiera nogami. Przy planszy wieloekranowej musimy ustalić konfiguracje planszy i zastanowić się jak stworek może się przemieszczać.Np. przy planszy typu płaszczyzna gdy wartość X osiągnie maksimum(położenie skrajne prawe) to stworek może przejść na(patrz str.13 plansza płaszczyzna):

z ekranu 1 na 4
z ekranu 2 na 5
z ekranu 3 na 6
z ekranu 4 na 7
z ekranu 5 na 8
z ekranu 6 na 9
z ekranu 7 na 1
z ekranu 8 na 2
z ekranu 9 na 3

W tym [illegible] można zaplanować schemat ruchu dla każdego ekranu z odpowiedniej konfiguracji.Przy zmianie ekranów muszą być [illegible] z maximalnych na maksymalne i odwrotnie (tak aby stworek jest np.na ekranie 1,X przekracza wartość [illegible] po przejściu na ekran 4 X musi być zmienione na [illegible] aby stworek znalazł się w skrajnym lewym położeniu,a nie w [illegible] prawym).

Przykładowy blok przełączania ekranów:

```
500 GET:GOTOX(0)
510  IF  ST=7  THEN X=X+1:(ZMIANA FAZY):IF X>XMAX THEN X=XMIN:ON
ERR GOSUB 1040,1050,1060,1070,1080,1090,1010,1020,1030
520  IF  ST=11 THEN X=X-1:(ZMIANA FAZY):IF X<XMIN THEN X=XMAX:ON
ERR GOSUB 1070,1080,1090,1010,1020,1030,1040,1050,1060
530  IF  ST=14 THEN Y=Y-1:(ZMIANA FAZY):IF Y<YMIN THEN Y=YMAX:ON
ERR GOSUB 1030,1010,1020,1060,1040,1050,1090,1070,1080
540  IF  ST=13 THEN Y=Y+1:(ZMIANA FAZY):IF Y>YMAX THEN Y=YMIN:ON
```

```
EKR GOSUB 1020,1030,1010,1050,1060,1040,1080,1090,1070
550 GOTO 500
1010 EKR=1:ADRES=EKR1:GOSUB 1500:RETURN
1020 EKR=2:ADRES=EKR2:GOSUB 1500:RETURN
1030 EKR=3:ADRES=EKR3:GOSUB 1500:RETURN
1040 EKR=4:ADRES=EKR4:GOSUB 1500:RETURN
1050 EKR=5:ADRES=EKR5:GOSUB 1500:RETURN
1060 EKR=6:ADRES=EKR6:GOSUB 1500:RETURN
1070 EKR=7:ADRES=EKR7:GOSUB 1500:RETURN
1080 EKR=8:ADRES=EKR8:GOSUB 1500:RETURN
1090 EKR=9:ADRES=EKR9:GOSUB 1500:RETURN
1500 POKE DL+5,ADRES:POKE 89,ADRES:POKE 106,ADRES+4:RETURN
```

Na początku programu należy oczywiście obliczyć adresy poszczególnych ekranów,adres programu wyświetlania.Należy wykonać skok do jednej z linii 1010-1090,w zależności od którego ekranu chcemy zacząć ruch.Program powyższy jest uniwersalny. Zmiana linii skoków w liniach 510-540 umożliwia sterowanie przy różnych konfiguracjach ekranów.

Załaduj program ANIMACJA ZE ZMIANĄ KONFIGURACJI.Z klawiatury można ustawić odpowiednią konfigurację ekranów.1-płaszczyzna 2-pionowa,3-pozioma.Mapa,zaczynająca się od górnego,lewego rogu, pokazuje gdzie znajduje się siworek.

31.EDYTOR GRACZY.

Do wykonania ładnej animacji należy przygotować kilka faz ruchu.Do tego celu został przygotowany program EDYTOR GRACZY. Po uruchomieniu zobaczysz ekran podzielony na trzy części: nazwa programu,trzy pola rysunkowe i pole animacji,menu. W polach rysunkowych możesz przesuwać kursor joystickiem!. Rysowanie punktu po wciśnięciu FIRE.Ponowne wciśnięcie FIRE kasuje narysowany punki.Zmianę pola rysunkowego uzyskuje się wciskając odpowiednio 1,2,3.Trzy pola rysunkowe służą do rysowania poszczególnych faz animacji.Pierwsze pole dla 1 fazy (spoczynkowej),drugie dla 2-fazy,trzecie dla 3-fazy.Po narysowaniu wszystkich faz należy przetworzyć je na dane przechodząc na każde pole rysunkowe i wciskając D.Jeżeli rysujesz fazy ruchu różniące się niewielkimi szczegółami możesz skorzystać z możliwości przepisania jednej fazy na drugą. Wciskasz P i podajesz,na którą fazę chcesz przepisać daną fazę. Jeżeli nie odpowiada Ci wykonany rysunek wciskasz K dla skasowania.Po narysowaniu i przetworzeniu na dane trzech faz ruchu wciskamy A dla uzyskania obrazu w polu animacji. Nasz rysunek możemy przemieszczać joystickiem!.W czasie animacji możemy zmieniać szerokość gracza (1,2,3).Zakończenie animacji W.

Jeżeli chcesz zobaczyć jak wygląda animacja innych rysunków możesz wcisnąć jedną z liter FGHJBMERYU.Zapis danych w zmiennych tekstowych na taśmie po wciśnięciu Z.Należy podać numer linii.Zapis w liniach NR,NR+10,NR+20.Zmienne nazywają się P1$, P2$,P3$.Dla swoich celów można zmienić te nazwy po dograniu danych do własnego programu. *Należy uważać,aby żadna z danych gracza nie miała wartości 34 (kod cudzysłowu-nie można umieścić go w zmiennej tekstowej).Komputer nie sprawdza tego faktu i po dograniu takich danych zgłosi błąd.*

32.LOSOWE POJAWIENIE SIĘ GRACZA.

Graczem możemy poruszać za pomocą joysticka,ale możemy zapewnić mu także ruch samodzielny korzystając z funkcji RND. Musisz ustalić zmienną sterującą SKOK.Po liniach,w których jest sprawdzane położenie joysticka należy wprowadzić linię zwiększającą tę zmienną o jeden.Po osiągnięciu przez nią pewnej wartości np.100 należy wykonać skok do podprogramu wywołującego gracza poruszanego losowo.Ruchy tego gracza zależą głównie od generatora liczb losowych,ale częściowo mogą zależeć od położenia gracza sterowanego joystickiem(gracz losowy będzie "szukał" gracza sterowanego).Załaduj program LOSOWE POJAWINIE SIĘ GRACZA.Po uruchomieniu pojawi się czarny gracz poruszany joystickiem.Po pewnej chwili zobaczysz drugiego gracza,białego, poruszającego się losowo.Ale ten gracz preferuje wyraźnie kierunki poziome,przy których zbliża się do gracza czarnego. Spróbuj obejść białego gracza czarnym.Jak zbliżysz się do niego na pewną odległość w pionie,powiększy swoją wielkość dwukrotnie. W linii 420 zmienna SKOK jest zwiększana o jeden.
Gdy osiągnie wartość 100 następuje skok do podprogramu wywołania gracza losowego (linia 440).Podprogram wywołania gracza działa tylko dla wartości 100.Po wywołaniu gracza(lub gdy wartość zmiennej jest większa niż 100-tzn.gracz jest na ekranie) następuje losowanie kierunku,w którym ma się poruszać(linie 490-520).W liniach 490-500 dodatkowo sprawdzana jest wartość X gracza poruszanego joystickiem.Jak jest większa od X gracza losowego wartości dodawane do współrzędnej X tego gracza są podwajane.Gdy wartość jest mniejsza wartości odejmowane od współrzędnej X gracza losowego też są podwajane.Powoduje to szybkie poziome zbliżanie się gracza losowego do gracza poruszanego joystickiem.Gdy różnica współrzędnych Y jest

mniejsza od 10 gracz losowy uzyskuje szerokość podwójną(linie 565-568).Wciśnięcie FIRE usuwa gracza losowego z ekranu na jakiś czas(linia 410).

Wykorzystanie tego typu sterowania dodatkowym graczem powoduje zmniejszenie prędkości gracza sterowanego joystickiem.Jest to wyraźnie zauważalne.Powodem tego jest konieczność obsługi dodatkowych podprogramów.Ale wprowadzić może duże urozmaicenie w programie.Zbędna jest także linia opóźniająca ruch gracza sterowanego joystickiem.

33.KOLOROWY BLOK RYSUNKOWY.

Program KOLOROWY BLOK RYSUNKOWY umożliwia zbudowanie grafiki do twoich programów.Rysowanie odbywa się na dziewięciu "kartkach"-ekranach przy pomocy znaków.Zestaw znaków może być zbudowany przy pomocy ROZBUDOWANEGO EDYTORA ZNAKÓW lub możesz skorzystać z zestawów w programie.Po uruchomieniu program wymaga podania trybu graficznego,w którym chcesz tworzyć grafikę(patrz rozdział 27),a następnie zgłasza się na pierwszej "kartce".

Wybór znaków do rysowania odbywa się przez wciśnięcie spacji, najechanie kursorem na odpowiedni znak i wciśnięcie FIRE.Powrót na odpowiednią "kartkę" po wciśnięciu klawisza 1-9.Kolor znaków umieszczanych na ekranie wybiera się z klawiatury(QWER), zmieniając jednocześnie kolor kursora na taki sam jak kolor znaku:Litera T ustawia biały kolor kursora.Jest to wygodne przy przesuwaniu kursora przez znaki o tym samym kolorze.Kasowanie narysowanych znaków można wykonać dopiero po najechaniu na puste miejsce na palecie znaków i wciśnięcie FIRE.Po powrocie na ekran roboczy najechanie na znak i wciśnięcie FIRE spowoduje skasowanie tego znaku.Skasowanie całego ekranu jest dokonywane po wciśnięciu razem SHIFT,CONTROL i TAB.Uważaj---kasowanie jest bezpowrotne i komputer nie rząda potwierdzenie.Wciśnięcie K uruchamia tryb określania współrzędnych kursora."Kartki" zostają skonfigurowane w płaszczyznę i kursor poruszany jest po całej planszy przy pomocy joysticka.Jego współrzędne pojawiają się w okienku tekstowym.L-wyjście z tego trybu pracy i powrót na pierwszą kartkę.Te dane są potrzebne w programie przy sprawdzaniu czy gracz dojechał w prawidłowe miejsce na planszy. Dodatkową możliwością jest animacja.Po wciśnięciu A kolejne ekrany są przełączane z szybkością regulowaną przy pomocy joysticka.Wyjście z animacji -S.Po zbudowaniu odpowiedniego

zestawu znaków jest możliwa bardzo szybka animacja.Wzorując się na tej technice można wykonać animację na dużo większej ilości ekranów.Zapis danych rysunku na taśmie odbywa się po wciśnięciu Z.Należy podać numer linii,od której chcemy mieć dane w programie.W każdej linii będą dane dla dwóch linii ekranu,co daje dla:

GR.1 i 12 10 linii DATA
GR.2 i 13 5 linii DATA

Numery linii są zwiększane o jeden.

JAK WYKORZYSTAC DANE Z KBR?

Na taśmie znajdują się dane linii ekranu w liniach DATA.Na początku i na końcu każdej linii jest dodany znak *.Aby przenieść rysunek na ekran należy przedefiniować zestaw znaków, korzystając z tych samych danych znakowych i tej samej zmiennej NZ$.Jest to bardzo ważne.Przy innej kolejności danych znakowych lub znaków w zmiennej NZ$ rysunek może zmienić się w sposób trudny do przewidzenia.

Podprogram przenoszący dane z linii DATA na ekran dla wszystkich trybów tekstowych:

```
2000 DIM A$(X)
2010 FOR Y=0 TO L
2020 READ A$:A$=A$(2,X-1)
2030 POSITION 0,Y:?#6;A$
2040 NEXT Y
```

Zmienne X i L należy dobrać w zależności od stosowanego trybu graficznego.

```
GR.1     X=22:L=19
GR.2     X=22:L=9
GR.12    X=42:L=19
GR.13    X=42:L=9
```

Przy budowaniu programu z wykorzystaniem grafiki z KBR należy na początku programu zadeklarować zmienną A$(lub inną) o wielkości odpowiedniej dla stosowanego trybu graficznego.W celu przeniesienia rysunków na ekran trzeba kolejno przełączać ekrany i podprogramem wprowadzać dane obrazów na ekran.

34.CIEKAWE EFEKTY.

TRZĘSIENIE ZIEMI.

W ATARI często jest stosowane adresowanie przy pomocy pary rejestrów.W taki sposób oblicza się adres pamięci obrazu (komórki 88 i 89) lub programu wyświetlania (komórki 560 i 561).

```
DL=PEEK(560)+PEEK(561)*256
```

Wartość w drugiej komórce (starszy bajt) oznacza numer strony,a wartość w pierwszej komórce (młodszy bajt) numer bajtu na stronie pamięci,od którego dokładnie zaczyna się program.

Obliczenia odwrotne wykonuje się w następujący sposób:

```
STARSZY BAJT=INT(SLOWO/256)
MŁODSZY BAJT=SLOWO-INT(SLOWO/256)*256
```

Ale wróćmy do naszego trzęsienia ziemi.Pierwsze trzy rozkazy programu wyświetlania mają następujące znaczenie:umieść osiem pustych linii(wartość dziesiętna 112).Wkonaj obliczenia adresu programu wyświetlania na podstawie wzoru podanego na początku rozdziału i napisz:

```
?PEEK(DL),PEEK(DL+1),PEEK(DL+2)
```

Otrzymasz trzy liczby 112.Zmieniając losowo te wartości możemy otrzymać efekt skakania obrazu.Po wykonaniu efektu należy w pierwszych trzech rozkazach DL umieścić 112.

```
10 DL=PEEK(560)+PEEK(561)*256
20 FOR I=0 TO 30
30 FOR T=0 TO 2
40 POKE DL+T,16*INT(RND(0)*7)
50 NEXT T
60 NEXT I
70 FOR T=0 TO 2
80 POKE DL+T,112
90 NEXT T
```

Efekt ten można wykorzystać jako skutek zderzenia gracza z jakimś obiektem ekranowym.

TĘCZA ATARI

Najsłynniejszym efektem specjalnym ATARI jest tęcza. Wykorzystywana jest często (nawet z pewną przesadą) w wielu programach.Tęczę można zobaczyć uruchamiając prostą procedurę w języku maszynowym.

```
10 DIM A$(20):P=5
20 FOR T=1 TO 20:READ A:A$(T,T)=CHR$(A):NEXT T
30 Z=USR(ADR(A$),P)
40 END:REM TU SKOK DO DALSZEJ CZESCI PROGRAMU
50 DATA 104,104,104,168,232,142,10,212,138,153,18,208,169
60 DATA 6,205,31,208,208,241,96
```

Zasięg działania tęczy zależy od parametru P.

P=4--duże litery (rej.708)
P=5--małe litery (rej.709)
P=6--duże litery w negatywie (rej.710)
P=7--małe litery w negatywie (rej.711)
P=8--tło (rej.712)

Zwróć uwagę na linię 60 programu.Zawiera ona część danych procedury.Liczby 31 i 208 są adresem komórki sprawdzanej przez procedurę,a 6 wartością,po której wystąpieniu procedura kończy działanie.Adres podawany jest w postaci młodszy,starszy bajt. Po obliczeniu (208*256+31=53279) wynika,że procedura skończy działanie po wciśnięciu START.Wartość 6 można zmienić na 5 lub 3 w celu kończenia procedury innym klawiszem konsoli.Zmieniając adres sprawdzanej przez procedurę komórki możemy spowodować opuszczenie procedury po wciśnięciu określonego klawisza np.spacji:

zamień 31 na 252

208 na 2
6 na 33
2*256+252:764 komórka sprawdzająca klawiaturę
33 kod spacji

PRZESUW PIONOWY

Następnym ciekawym efektem możliwym do wykonania w prosty sposób jest przesuw pionowy linii ekranu.Aby umożliwić linii przesuw pionowy należy dodać do rozkazu tworzącego tę linię wartość 32. W programie wyświetlania rozkazy tworzące linie obrazu zaczynają się od numeru DL+4(pierwsze trzy rozkazy to tworzenie ośmiu pustych linii patrz TRZĘSIENIE ZIEMI).Umieśćmy pozwolenie na przesuw pionowy w linii 7 GR.2.

```
L=DL+3+7
POKE L,PEEK(L)+32
```

Wprowadzamy komunikat w linii,w której dozwolony jest przesuw:

```
POSITION 0,6:?#6;"KOMUNIKAT"
```

Aby dokonać przesuwu pionowego tej linii należy do rejestru przesuwu pionowego 54277 wartości od 0 do 15.Jak to wygląda w praktyce patrz program PRZESUW PIONOWY.

35.ZEGAR.

ATARI posiada wbudowany zegar systemowy.Jest to tzw.zegar czasu bieżącego.Liczy on od zera (od chwili włączenia komputera) i przechowuje wartści trzybajtowe.Znajduje się w pamięci RAM w komórkach 18,19,20.Zlicza on przerwania VBLANK występujące co 1/50 sekundy wg.standardu PAL.Wprowadź krótki program i uruchom go kilka razy:

```
10 CZAS=PEEK(18)*65536+PEEK(19)*256+PEEK(20)
20 CZAS=INT(CZAS/50)
30 ?"CZAS=";CZAS;" SEKUND"
```

Każda z kolejnych wartości będzie inna(większa).Można to wykorzystać w prosty sposób w programie sprawdzającym czas potrzebny na odpowiedź.Dodaj poniższe linie do programu pomiaru czasu:

```
5 GOTO 50
30 RETURN
40 GOSUB 10
50 CZ=CZAS
60  ?CHR$(125):?"ILE JEST ";INT(RND(0)*5)+1;"*";INT(RND(0)*6)+1;
:INPUT X
70 CZA=CZAS-CZ
80 ?"POTRZEBOWALES ";CZA;" SEKUND(Y)"
```

Program nie sprawdza czy podany wynik jest prawidłowy.

Prawdziwy zegar będzie jednak dopiero wtedy,gdy można będzie można wprowadzić aktualny czas (w celu ustawienia zegara) i wyświetlać czas bieżący na ekranie.Program poniższy ma te możliwości.Wprawdzie nie robi nic ponadto,ale połączenie zegara z wiadomościami zawartymi w rozdziale PRAKTYCZNE WYKORZYSTANIE KOMPUTERA umożliwi czasowe sterowanie urządzeniami zewnętrznymi.

```
30 GRAPHICS 2:POKE 752,1:? "PODAJ AKTUALNY CZAS"
```

```
40 ? "GODZINA ";:INPUT G:? "MINUTA ";:INPUT M:? "SEKUNDA ";:
   INPUT S
50 POKE 20,0:POKE 19,0:POKE 18,0
60 GOSUB 200
70 POSITION 5,5:? #6;GODZ;":";MIN;":";SEK;"   "
80 GOTO 60
200 C=(((PEEK(18)*256)+PEEK(19))*256+PEEK(20))/50
210 GODZ=INT(C/3600):C=C-(GODZ*3600)
220 MIN=INT(C/60):SEK=INT(C-(MIN*60))
230 SEK=SEK+S:IF SEK>59 THEN SEK=SEK-60:MIN=MIN+1
240 MIN=MIN+M:IF MIN>59 THEN MIN=MIN-60:GODZ=GODZ+1
250 GODZ=GODZ+G
260 GODZ=GODZ-INT(GODZ/24)*24
270 RETURN
```

36.OKIENKA.

Technikę okienek można wykorzystać do szybkiego pokazywania na ekranie nowej informacji,która nie musi być ciągle widoczna. Załaduj program OKIENKA.Po uruchomieniu masz możliwość umieszczenia na ekranie czterech okienek (klawisze 1,2,3,4) oraz zapis na taśmie procedury maszynowej obsługi iej techniki.Należy zwrócić uwagę na kolejność otwierania i zamykania okienek nachodzących na siebie.

OPTION---zapis procedury maszynowej tworzącej okienka na taśmie w zmiennej tekstowej W$.

START----ponowne uruchomienie programu(w przypadku gdy popłączesz kompletnie okienka)

SPOSOB WYKORZYSTANIA PROCEDURY "OKIENKA"

Dane do wyświetlania muszą być umieszczone w zmiennej tekstowej. Następnie znaki zmiennej muszą być przetworzone na kody wewnętrzne ATARI.Przetworzenia dokonuje procedura zawarta w zmiennej pomocniczej C$:

```
X=USR(ADR(C$),ADR(A$))
```

Przetworzenia zmiennych należy dokonać w takiej kolejności w jakiej były deklarowane w programie.Wymianę zawartości ekranu na zawartość zmiennej dokonuje procedura zawarta w zmiennej tekstowej W$:

```
X=USR(ADR(W$),ADR(A$),O,W,S,L)
```

O---miejsce umieszczenia lewego,górnego rogu okienka.Obliczamy je wg.danych z instrukcji POSITION:

```
O=X+Y*n
GR.0,12,13          n=40
GR.1,2              n=20
```

W---wysokość okna w liniach ekranu

S---szerokość okna w znakach

L---przesunięcie

	prostokąt	w prawo	w lewo
GR.0,12,13	40	39	41
GR.1,2	20	19	21

Pokazanie okna na ekranie uzyskuje się przez wywołanie procedury z odpowiednimi paramertrami.Powrót do danych ekranowych-ponowne wywołanie procedury z tymi samymi parametrami.

37.TABLICZKA MNOŻENIA

Aby praktycznie wykorzystać naszą wiedzę zbudujemy program edukacyjny do nauki tabliczki mnożenia.

Większość tego typu programów losuje dwie liczby, pokazuje je na ekranie i rząda wprowadzenia wyniku.My zrobimy całkowicie inny program.Zbudujemy grę wieloekranową z dość ciekawą grafiką w trybie 1.Posłużymy się zestawem znaków dla GR.1 i 2 z KBR.Oczywiście możesz zbudować własny zestaw znaków.Musi zawierać znaki wszystkich cyfr oraz znaki do stworzenia zabudowy miejskiej i wiejskiej.Na początku oczywiście scenariusz gry:

* Jesteś pracownikiem poczty.Do twoich obowiązków należy lokalizowanie na skomputeryzowanej mapie adresów doręczania przesyłek i dojazd tam kursorem w jak najkrótszym czasie.Poczta oczywiście przeszła już na nowy styl pracy i za poprawną lokalizację adresu w czasie krótszym od 60 sekund wypłaca Ci 50$.Ale gdy przekroczysz ten czas zapłata spada do 10$,by spaść do 2$ gdy czas przekroczy 90 sekund.Za nieprawidłową lokalizację potrącana jest kwota 25$.O doręczanie przesyłek nie musisz się martwić.Zadbają o to odpowiednio zaprogramowane roboty.Cała trudność polega na tym,że adres przesyłki podawany jest w sposób zakodowany-przy pomocy mnożenia dwóch liczb. *

Na początku musisz przygotować grafikę do programu. Dziesięć pełnych ekranów w trybie 1.Musisz rozplanować występowanie numerów domów wg.wyników mnożenia.Dla ułatwienia podaję wszystkie wyniki,które nie powtarzają się(z podziałem na grupy):

pełne dziesiątki
10,20,30,40,50,60,70,80,90,100(10)
wyniki do 25

1,2,3,4,5,6,7,8,9,12,14,15,16,18,21,24,25(17)
wyniki powyżej 25
27,28,32,35,36,42,45,48,49,54,56,63,64,72,81(15)
razem 42 wyniki.

Musimy też ustalić pewną ilosc wyników nieprawidłowych,których nie ma w tabliczce mnożenia:
11,13,17,19,22,31,33,39,41,43,47,52,55,61,62,65,74,75,76,77,82,84,85,92,93,96

Samą grafikę wykonujemy tak,aby mieć na ekranach rozmieszczone domki z numerami odpowiadającymi wynikom mnożenia.Między nie należy wstawić wyniki nieprawidłowe.Nagrywamy kolejno ekrany na taśmę i przechodzimy w tryb określania współrzędnych kursora (wciskamy K).Poruszając kursorem po całej planszy sprawdzamy wartosci współrzędnych dla wyników prawidłowych i zapisujemy te wartosci(razem z wynikiem).Oczywiście zapisujemy wartości w przedziale od do aby wystarczyło wskazać miejsce prawidłowego wyniku,bez konieczności dokładnego trafiania w cel.
Teraz należy zbudować szkielet programu zawierający podprogramy przełączania zestawów znaków,przedefiniowywania znaków, przełączania ekranów,ustalić kolory i wprowadzić je po ostatniej użytej instrukcji GRAPHICS.Aby ułatwić sobie pracę proponuję skorzystać z programu KBR po wykasowaniu niepotrzebnych linii.

Należy usunąć przełączanie ekranów z klawiatury,dane drugiego zestawu znaków,wprowadzanie parametrów początkowych,efekty dźwiękowe,animację,paletę znaków.Jako parametry początkowe należy wprowadzić parametry dla GR.1.Pozostawić trzeba tryb poruszania kursora po całym ekranie po skonfigurowaniu w płaszczyznę.Oczywiście możliwa jest inna konfiguracja.Należy dodać podprogram rysowania z danych i już można dograć naszą grafikę.Przełączając kolejno ekrany możemy narysować na nich

nasz plan miasta i okolic.Należy przygotować dodatkowe podprogramy:
1.Pomiaru czasu.
2.Losowania liczb i wyświetlania komunikatu o adresie przesyłki.
3.Sprawdzania poprawności odnalezionego adresu(wg.danych z położenia kursora zapisanych wcześniej).
4.Trzęsienie ziemi po zderzeniu gracza z elementem ekranu.
Jak to będzie nam dobrze działało możemy opracować swojego stworka w kilku fazach aby można było wykonać ciekawą animację.
Musisz jeszcze ustalić sposób zakończenia gry:
1.Zakończenie dobre-zarobiłeś np.1000$
2.Po dobrym zakończeniu możliwość kontynuacji z powiększaniem zdobytego kapitału.
3.Złe zakończenie-suma ujemna przekroczyła 100$.
4.Koniec gry lub rozpoczęcie od nowa z czystym kontem.

Oczywiście program może być całkiem inny.Jakikolwiek by on nie był myślę,że napiszesz go tak dobrze,że każdy chętnie będzie się nim bawił.Jak mi to wyszło sprawdź ładując program TABLICZKA MNOŻENIA.

Parametry początkowe:	linia 510,530
Zmiana ekranu:	linia 570
Rysowanie kolejnych ekranów:	linia 720,730
Podprogram rysowania:	linia 1020,1060
Dane rysunków:	linia 1070,1960
Dane gracza:	linia 1970,2080

Została określona tylko jedna zmienna dla wszystkich faz animacji.W zależności od fazy zmienia swoją zawartość w odpowiedniej linii.

Trzęsienie ziemi:	linia 2180,2300

Procedura OKIENKA: linia 2320,2360
Mapa: linia 2380,2520

Mapa została stworzona przy pomocy zmiennej tekstowej MP$.Zawiera ona przedefinowane znaki cyfr ułożone zgodnie z konfiguracją ekranu:
ADGBEHCFI
147258369
Po wciśnięciu spacji,w zależności od ekranu,na którym znajduje się gracz,zmienna przyjmuje odpowiednią wartość(numer wyróżnionego ekranu wprowadzony jest do zmiennej w negatywie).Okno mapy jest otwierane w lewym,górnym rogu,ma wysokość 3 linie i długość 3 znaki,bez przesunięcia(parametry 0,3,3,20).Mapa wyłącza się sama.

Pomiar czasu: linia 2540,2560
Losowanie: linia 2580,2610
Komunikat: linia 2630,2640
Sprawdzenie wyników: linia 2660,3080
Nie trafienie: linia 3090,3190
Trafienie: linia 3210,3340
Dźwięk: linia 3360,3390

Po wylosowaniu "adresu" należy dojść do odpowiedniego wyniku (numer domku) i wcisnąć FIRE.Poprawność "adresu" sprawdzana jest wg.nastepujących założeń:
czy jest prawidłowy ekran,czy współrzędne X i Y mieszczą się w zadanym zakresie.Gdy wynik jest prawidłowy,mierzony jest czas odpowiedzi i w zależności od niego obliczane wynagrodzenie.

38.PRAKTYCZNE ZASTOSOWANIE KOMPUTERA.

Do tej pory znajdowaliśmy się w zamkniętym świecie grafiki komputerowej,a co zdolniejsi z nas dodatkowo opanowali świat dźwięków.Ale cały czas poruszaliśmy się w świecie komputerowej iluzji,w świecie barw i dźwięków monitora.Każdy kto nie pasjonował się programowaniem,a komputery uważał za zbędny i nikomu nie potrzebny dodatek do życia mógł zawsze powiedzieć: "Po co to komu?".Aby mieć argumenty dla takich osobników zajmiemy się teraz zastosowaniem ATARI w praktyce,zajmiemy się współpracą naszego komputera ze światem realnym.

Na następnych stronach zostaną opisane różne układy elektroniczne .Niektóre bardzo proste,inne bardziej rozbudowane. W tym miejscu mam jedną prośbę.Osoby,które nie widzą różnicy między rezystorem a tranzystorem,a dodatkowo nie bardzo wiedzą, który koniec lutownicy robi się gorący,proszone są o przerwanie czytania tego rozdziału w tym miejscu.Konstruowane układy będą dołączane bezpośrednio do wejść komputera i brak podstawowych wiadomości może spowodować bardzo kosztowne uszkodzenie naszego ATARI.Braki w wiedzy można uzupełnić korzystając z odpowiedniej literatury lub doświadczeń kolegów.I dopiero wtedy zapraszam do dalszej lektury.

Do dalszych eksperymentów będą nam potrzebne gniazda ELTRA 881 (lub podobne) do podłączania do portów joysticków,lutownica, cyna,kalafonia,trochę giętkiego przewodu,narzędzia ręczne. Do montażu układów elektronicznych przydatne byłoby stanowisko do montażu próbnego bez lutowania np.zestaw do montażu na sprężynkach produkcji radzieckiej.Do budowy interfejsów na stałę należy wykonać płytkę drukowaną wg.załączonych rysunków.

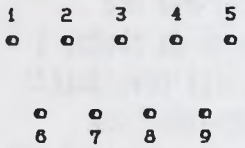
Wszystkie układy należy montować bardzo starannie,bo każde zwarcie lub niestaranne zmontowanie może spowodować poważne

uszkodzenie komputera.Opisane konstrukcje zostały wykonane przez autora.Są sprawdzone i działają.

PORTY JOYSTICKÓW.

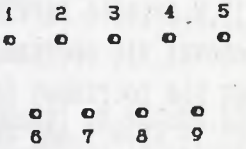
Przy omawianiu instrukcji STICK mówiliśmy,że wejścia joysticków są bardzo uniwersalnymi portami.Teraz omówimy je bardziej szczegółowo.

PORT 1



1---JOYSTICK DO GÓRY (A)(0)
2---JOJSTICK DO DOŁU (B)(1)
3---JOJSTICK W LEWO (C)(2)
4---JOYSTICK W PRAWO (D)(3)
5---PADDLE(1)
6---FIRE
7---+5V
8---MASA
9---PADDLE(0)

PORT 2



1---JOYSTICK DO GÓRY (E)(4)
2---JOJSTICK DO DOŁU (F)(5)
3---JOJSTICK W LEWO (G)(6)
4---JOJSTICK W PRAWO (H)(7)
5---PADDLE(3)
6---FIRE
7---+5V
8---MASA
9---PADDLE(2)

Do przyłączania jojsticka wykorzystywane są styki 1,2,3,4,6,8. Jeżeli jojstick jest z AUTOFIRE to dodatkowo 7.Wejście PADDLE służy do podłączenia manipulatora analogowego tzw.wioselek.Jest to potencjometr o wartości około 500 kom włączany między +5V(7), a masę(8).Suwak potencjometru podłącza się do wejścia 5 lub 9.

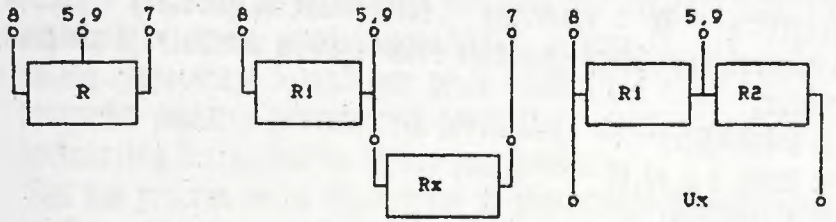
Położenie można odczytać przy pomocy instrukcji PADDLE z dpowiednim numerem (0,1,2,3).

np.10 ? PADDLE(0):GOTO 10

W zależności od położenia potencjometru wartość PADDLE będzie się zmieniać w granicach od 0 do 228.

Wykorzystanie wejść PADDLE może być dwojakiego rodzaju. Mogą służyć do pomiaru napięcia lub do pomiaru rezystancji (oporności).

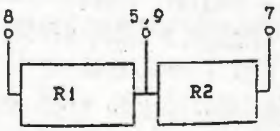
PADDLE(1) POMIAR REZYSTANCJI(2) POMIAR NAPIĘCIA(3)



Przygotuj potencjometr o wartości ok.500 kom (np.470 kom) i zmontuj prosty układ wg.schematu (1) lutując odprowadzenia potencjometru do odpowiednich końcówek wtyku.Wprowadź krótki program :

```
10 ? PADDLE(0):GOTO 10
```

Obracając potencjometrem obserwuj zmiany wartości wyświetlanej na ekranie.Powinna zmieniać się w zakresie od 0 do ponad 200. W celu uzyskania jakiejś stałej wartości funkcji PADDLE możemy zmontować prosty układ dwóch rezystorów(suma ich rezystancji powinna wynosić około 500 kom):



Tego typu układ może służyć np.jako identyfikator użytkownika komputera lub jako klucz zabezpieczający przed użytkowaniem nielegalnej kopii programu np.program graficzny dostarczany razem z odpowiednim joystickiem.Wprawdzie jest to bardzo proste zabezpieczenie programu,ale w odpowiednich warunkach może być

bardzo efektowne.

Do zmontowania następnego układu będzie potrzebny fotorezystor. Jest to element którego oporność zmienia się po wpływem oświetlenia.W stanie ciemnym jest największa.Układ montujemy wg.schematu (2).Rezystor R1 ma wartość ok.300 kom.W miejsce rezystora Rx wstawiamy fotoelement.Najprostszy sposób wykorzystania zmiany oporności fotorezystora demonstruje program poniżej:

```
10 A=PADDLE(1)
20 SOUND 0,A,10,10
30 ? PADDLE(1)
40 GOTO 10
```

Zmiany oporności spowodowane przysłanianiem fotoelementu powodują zmiany wysokości dźwięku.Jeżeli w miejsce fotorezystora podłączymy elektrody mocowane do palców jednej ręki otrzymamy prosty detektor kłamstwa.Dane z pomiarów można przechować w tablicy i po skończonym cyklu pomiarowym poddać dalszej obróbce. Graficzne przedstawienie danych otrzymanych z układu pomiaru rezystancji można uzyskać wykorzystując 8 tryb graficzny.Jest to tryb bitowy o najwyższej rozdzielczości:320 na 160 punktów (bez okienka tekstowego 192).Ale może mieć tylko jeden kolor punktu i zajmuje ponad 8 kilobajtów pamięci.Nadaje się do wszelkiego rodzaju wykresów i rysunków o dużej ilości szczegółów (schematy, rysunki techniczne).

```
10 GRAPHICS 8:COLOR 1
20 GOSUB 100
30 PLOT 0,A
40 FOR T=1 TO 319
50 GOSUB 100
60 DRAWTO T,A
```

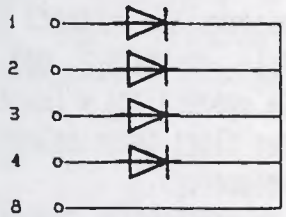
```
70 ? PADDLE(1)
80 FOR OP=0 TO 10:NEXT OP
90 NEXT T:GOTO 10
100 A=INT(PADDLE(1)/2):RETURN
```

W linii 100 są obliczenia dostosowujące wartości PADDLE do możliwości trybu graficznego.Jeżeli interesują nas szybkie zmiany rezystancji likwidujemy pętlę opóźniającą w linii 80.W przypadku pomiarów prowadzonych przez długi okres należy linię opóźniającą dostosować do okresu pomiarowego.

Ten sam program można wykorzystać do graficznego przedstawiania wyników pomiarów napięcia(program bez linii opóźniającej). Montujemy układ pomiarowy wg.schematu (3).Rezystor R1 ma wartość 300 kom,R2-200 kom.Do zacisków Ux dołączamy silniczek elektryczny prądu stałego.Kręcimy szybko ośką i na wykresie pojawiają się ostre piki napięciowe.Na tej zasadzie można zbudować wiatromierz.Przy pomocy instrukcji STICK ustalamy kierunek wiatru montując odpowiednio styki.Siłą wiatru mierzymy układem pomiarowym (3) z mikrosilnikiem prądu stałego.

Wejścia PADDLE można stosować do przyjmowania danych z zewnątrz. Aby sterowanie było pełne komputer musi mieć możliwość wysyłania danych sterujących.Porty joysticków mają również tę umiejętność. Przyjrzyj się dokładnie opisowi portów joysticków.Przy stykach 1,2,3,4 w każdym porcie są dodatkowo litery od A do H i liczby od 0 do 7.Są to kolejne bity ośmiobitowej szyny nadajnika - odbiornika PIA(układ MC 6820).W normalnych warunkach szyna pracuje jako wejście dla dwóch joysticków.Do sterowania kierunkiem przesyłania danych służą komórki 54018 i 54016. Możemy przełączyć na wyjścia (wysyłanie danych sterujących przez porty joysticków) oba porty lub tylko jeden.Do sprawdzenia jak to działa należy zmontować układ czterech diod LED.Nie

przestrasz się,że diody nie mają rezystorów ograniczających prąd. Rezystory te są wmontowane do wyjść układu PIA.Diody LED przylutuj bezpośrednio do gniazda 881.




```
10 P=PEEK(54018):POKE 54018,P-4
20 POKE 54016,255
30 POKE 54018,P
40 FOR T=1 TO 255
50 POKE 54016,T
60 FOR OP=0 TO 150:NEXT OP
70 NEXT T
```

linia 10 ustawienie PIA na wprowadzenie danych sterujących
linia 20 ustawienie dwóch portów na wyprowadzanie danych
(wpisanie 15 - pierwszy port wyjście drugi wejście)
linia 30 przywrócenie wartości początkowej (60) w rejestrze 54018
linia 50 wprowadzanie wartości do komórki 54016 powoduje przeniesienie ich na porty jojsticków w postaci binarnej

Jeżeli podłączysz zbudowany układ do pierwszego portu i uruchomisz program diody będą kolejno zapalały się wg.kodu dwójkowego.Jeżeli chcesz podłączyć diody do drugiego portu to aby coś zobaczyć musisz zmniejszyć opóźnienie w linii 60:

```
60 FOR OP=0 TO 30:NEXT OP
```

Jest to spowodowane tym,że na pierwszym porcie mamy pierwsze cztery bity,a na drugim cztery ostatnie,zmieniające się dużo wolniej.Zmień program wprowadzając poniższe linie:

```
20 POKE 54016,15
40 T=PEEK(633)
60 GOTO 40
70 zlikwiduj
```

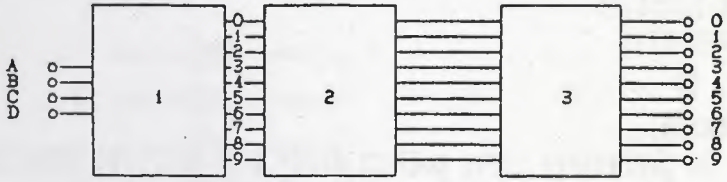
Do pierwszego portu podłącz diody,a do drugiego joystick. Port pierwszy mamy ustawiony na wyjście,a drugi na wejście. Wychylanie joysticka spowoduje wyświetlanie wartości funkcji STICK za pomocą diod w pierwszym porcie.Wprawdzie trudno uważać sterowanie diod za pomocą joysticka przez komputer za duże osiągnięcie,ale w miejsce diod można podłączyć dekoder kodu dwójkowego na dziesiętny i odpowiednie układy wykonawcze. Uzyskamy w ten sposób możliwośc sterowania nawet kilkunastoma urządzeniami zewnętrznymi.Dodatkowo mamy możliwość wprowadzenia danych wejściowych z czujników zewnętrznych (PADDLE).I to jest już prawdziwe sterowanie.Dodatkowe połączenie z zegarem da możliwości sterowania czasowego.

Najprostszy interfejs sterujący można zbudować stosując dekoder dwójkowo-dziesiętny typu 74141.W zasadzie dla interfejsów zasilanych bezpośrednio z komputera należy stosować cyfrowe układy scalone serii LS,aby nie obciążać dodatkowo zasilacza.

Układ 74141 daje możliwość sterowania dziesięcima urządzeniami zewnętrznymi.Stosując inny dekoder np.74154 zwiększamy ilość urządzeń do 16.My zostaniemy przy dekoderze 74141. Schemat blokowy interfejsu jest bardzo prosty.Składa się z trzech podstawowych bloków:

1-dekoder
2-układy pośredniczące

3-układy wyjściowe



DEKODER UKŁADY POSRED. UKŁADY WYJSCIOWE

Dekoder jest podłączony bezpośrednio do portu joysticka przełączonego na wyprowadzanie danych.Układy pośredniczące odwracają fazę sygnału otrzymanego z dekodera.Układy wyjściowe dostosowują sygnał z poziomu TTL do poziomu niezbędnego do wysterowania urządzenia zewnętrznego.

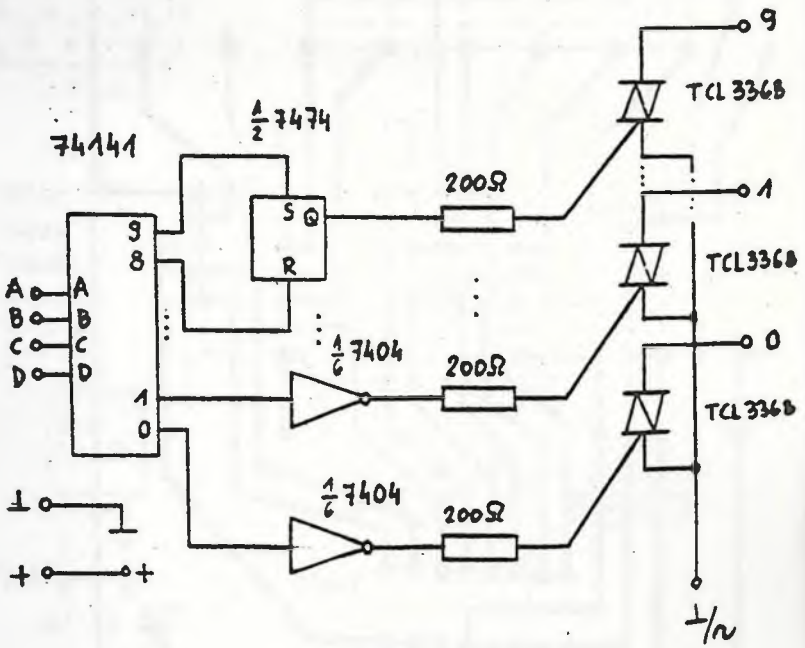
Jako dekodera możemy użyć dowolnego dekodera TTL przetwarzającego kod binarny na dziesiętny.Powinien być to układ scalony wykonany w technice LS,ale może być również zwykły. W egzemplarzu modelowym zastosowałem układ MH 74141.Przy konieczności sterowania większą ilością urządzeń zewnętrznych można zastosować dekoder 74154 (szesnaście wyjść).

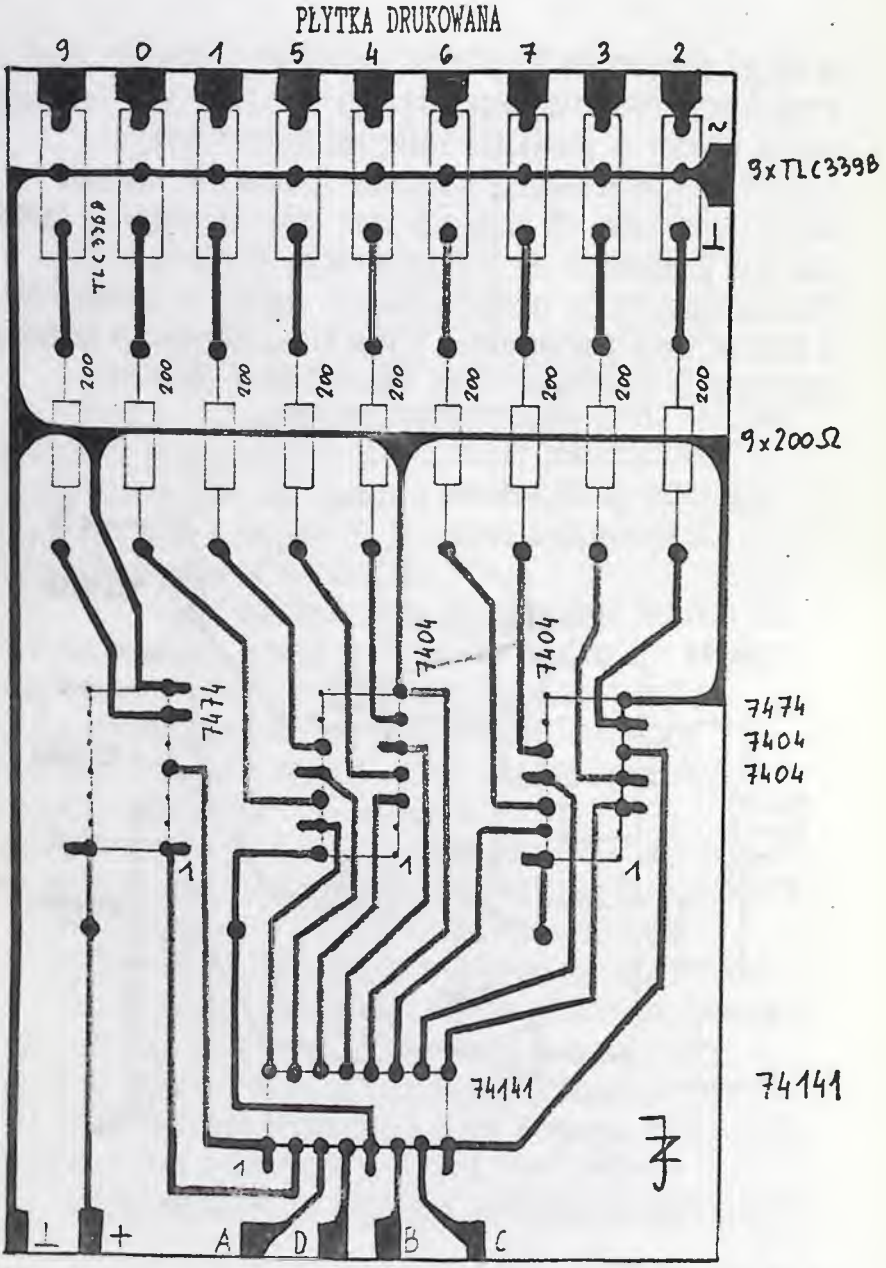
Układy pośredniczące służą głównie do odwrócenia fazy sygnału sterującego.Wyjście aktywne dekodera ma stan niski(0),a układy wyjściowe wymagają stanu wysokiego(1).Do tego celu służą inwertory typu 74LS04.Wszystkie sygnały sterujące będą aktywne tylko w czasie podania odpowiedniego kodu na wejście dekodera. Aby uzyskać możliwoś włączania i wyłączania na stałę jednego z urządzeń do wyjścia 9 i 8 został podłączony przerzutnik typu 74LS74.Sygnał 9 włącza urządzenie,a sygnał 8 wyłącza.Przerzutnik zmienia swój stan tylko w czasie podania "jego" sygnałów

Układy wyjściowe budowane są w zależności od sterowanych

urządzeń zewnętrznych.Mogą to być przekaźniki sterowane przez tranzystory,tranzystory mocy,tyrystory lub triaki.Nasz interfejs będzie używany do sterowania elementami makiety kolejki elektrycznej.Do sterowania w makiecie używane jest napięcie zmienne o wartości 16V.Prądy pobierane przez zwrotnice są bardzo duże,ale krótkotrwałe.Do sterowania takimi urządzeniami doskonale nadają się triaki.Są niewiele większe od tranzystorów, a mają możliwości przewodzenia prądów kilkuamperowych.W układzie modelowym są zastosowane triaki typu TLC 336 B (6A,600V).

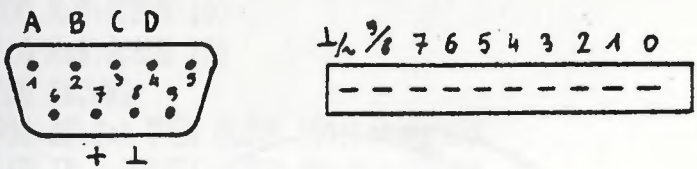
SCHEMAT IDEOWY INTERFEJSU.



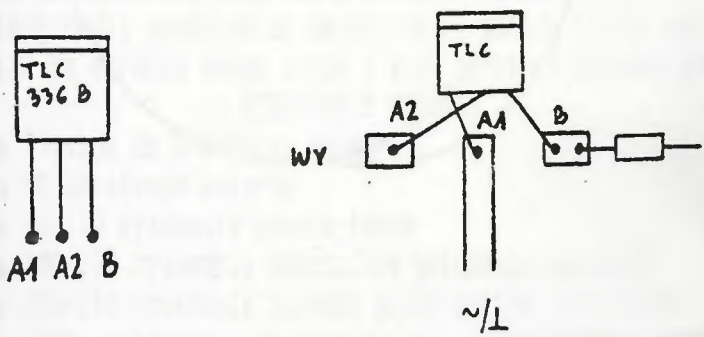

PŁYTKA DRUKOWANA
9 0 1 5 4 6 7 3 2
9xTLC339B
TLC339B
200
9x200Ω
7404
7474
7474
7404
7404
74141
74141
A
D
B
C

Płytka drukowana jest wykonana w postaci przystosowanej do montażu bez wiercenia otworów.Tzn.wyprowadzenia wkładów scalonych odginamy na zewnątrz i lutujemy bezpośrednio do płytki.To samo robimy z innymi elementami.Zwróć uwagę na lutowanie wyprowadzeń triaków.Gdy przylutujesz je inaczej interfejs będzie działał nieprawidłowo.Jako kabel połączeniowy z komputerem zastosuj przewód od starego joysticka.Wyjście sygnałów sterujących wyprowadź na złącze wielostykowe.Połączenie z makietą przez takie samo złącze przewodem taśmowym.

ZŁĄCZA INTERFEJSU



PODŁĄCZENIE TRIAKA



PROSTY UKŁAD MAKIETY

Aby wypróbować nasz interfejs w działaniu możemy zbudować prostą makietę składającą się z kółka,dwóch zwrotnic i dodatkowej bocznicy.Potrzebna będzie zwrotnica prawa i lewa.Makietę będziemy budowali z elementów torowych TT.Sposoby podłączania do interfejsu odnoszą się tylko do tego typu kolejki.Można oczywiście podłączać interfejs do kolejki HO,ale trzeba zwrócić uwagę na sposób sterowania zwrotnicami.

PROSTY PROGRAM STERUJĄCY

```
10 P=PEEK(54018):POKE 54018,P-4
20 POKE 54016,15:POKE 54018,P
30 S=1
40 K=PEEK(764)
50 IF K=33 THEN GOSUB 300
60 GOTO 40
100 M=6:GOSUB 400
110 M=4:GOSUB 400
120 RETURN
200 M=7:GOSUB 400
210 M=5:GOSUB 400
220 RETURN
300 IF S=1 THEN GOSUB 100:S=0:RETURN
310 IF S=0 THEN GOSUB 200:S=1:RETURN
400 POKE 54016,M
410 FOR T=0 TO 20:NEXT T
420 POKE 764,255
430 POKE 54016,0
440 RETURN
```

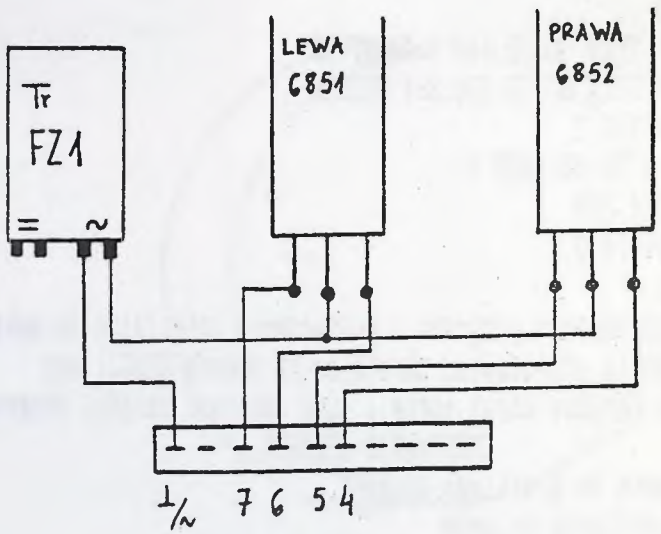
Po uruchomieniu programu i podłączeniu interfejsu do makiety będziemy mogli przestawiać zwrotnice za pomocą SPACJI. Aby widzieć na ekranie obraz torów i stan zwrotnic załaduj program

STEROWANIE KOLEJKĄ.

linia 70 skok do krótkiego opisu
linia 80 ustalenie kolorów
linia 90-140 rysowanie obrazu torów
linia 160-190 rysowanie znaczników położenia zwrotnic
linia 200-210 ustalenie sposobu pracy portów joysticków
linia 220 ustalenie parametrów na przełączenie zwrotnic w

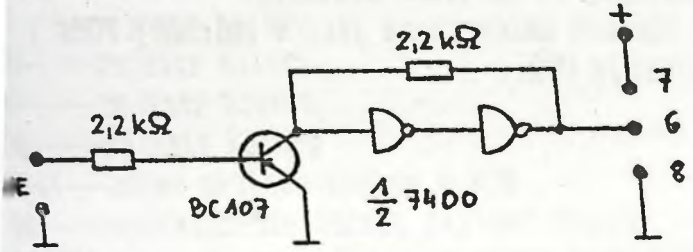
położenie początkowe
linia 360-400 przełączanie zwrotnic wg.podanych parametrów
linia 410-430 krótki opis
Oczywiście makieta może być dowolnie rozbudowana.Możesz wykonać interfejs o szesnastu wyjściach(a nawet dwa interfejsy podłączane do obu priów) i sterować odpowiednio większą ilością zwrotnic,przejazdów itp.Istnieje możliwośc wprowadzenia informacjii do komputera o położeniu pociągu i pokazania tego na rysunku.Ale to wymaga dość dużego nakładu pracy.Ale warto spróbować!

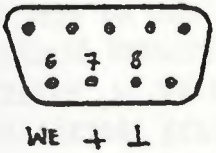
SPOSOB PODŁĄCZENIA ZWROTNIC DO INTERFEJSU



MIERNIK CZĘSTOTLIWOSCI

Ostatnim przykładem,jaki chciałem podać,w dziedzinie
cznego zastosowania ATARI jest miernik częstotliwości.
n mierzyć sygnały o poziomie TTL podane na wejści FIRE
szego portu.Maksymalna częstotliwość przebiegu badanego
wynosić 12 kHz.Nic nie stoi na przeszkodzie aby zastosować
iednie dzielniki (np.7490)i zwiększyć znacznie zakres
ządu.Aby umożliwić pomiar sygnałów innych niż TTL należy
wać prosty układ dopasowujący.Jest on zasilany bezpośrednio
mputera.





Zaladuj program POMIAR CZĘSTOTLIWOSCI.Po uruchomieniu należy podać czas pomiaru w zakresie 0.1 sek do 5.1.Dla częstotliwości większych powinien być mniejszy (powyżej 2kHz 0.5 sek),a dla małych większy(poniżej 100 Hz 5 sek).W czasie pracy programu wciśnięcie dowolnego klawisza spowoduje przejście do trybu wprowadzania czasu.Jednostkowe pomiary częstotliwości nie zawsze pokazują prawidłowy wynik.Dlatego program prowadzi ciągły pomiar i wyświetla wartość bieżącą częstotliwości oraz wartość średnią z trzech ostatnich pomiarów.Daje to możliwość porównania odchyłki wartości bieżącej od wartości średniej.
Procedura maszynowa pomiaru umieszczona jest w zmiennej POM$ i wywoływana jest instrukcją USR.

39.WAZNE KOMORKI PAMIĘCI.

764-----kod wciśniętego klawisza
53279---kod wciśniętego klawisza konsoli
732-----kod klawisza HELP
632-----joystick 0
633-----joystick 1
53770---wartości losowe
712-----rejestr koloru
711-----rejestr koloru
710-----rejestr koloru
709-----rejestr koloru
708-----rejestr koloru
57344---adres zestawu znaków w ROM
756-----wprowadzenie adresu zestawu znaków
560,561-wprowadzenie adresu programu wyświetlania
88,89---wprowadzenie adresu pamięci obrazu
106-----najwyższa dostępna strona pamięci
512.513-wprowadzenie adresu programu obsługi przerwań
54286---pozwolenie na obsługę przerwania(pozwolenie-192)
752-----widoczność kursora (0-widoczny 1-niewidoczny)
559-----włączanie i wyłączanie ekranu(0-wyłączony,34-włączony)
rozdzielczość graczy 46-dwuwierszowa(punkt jak GR.7)
62-jednowierszowa(punkt jak w GR.15)
53248-53251---pozycja pozioma graczy 1-4
53256-53259---szerokość graczy 1-4
0-normalna 1-podwójna 3-poczwórna
623----------priorytety wyświetlania
704-707-------rejestry kolorów graczy 1-4
53260-53263---rejestry kolizji między graczami

53252-53255---rejestry kolizji między graczami a rysunkiem
53277---------włączenie i wyłączenie techniki graczy
0-wyłączenie 3 włączenie
742-----------szczyt pamięci
54277---------rejestr przesuwu pionowego
18,19,20------zegar systemowy
53278---------kasowanie rejestrów kolizji
wpisanie dowolnej wartości(0-255)
54018---------sterowanie układem PIA(normalnie 60),sterowanie silnikiem magnetofonu (52-włączony)
54016---------sterowanie kierunkiem transmisji danych przez PIA oraz wprowadzanie danych przy ustawieniu portów joysticków jako wyjścia
77------------tryb losowej zmiany kolorów(samoczynny-po osiągnięciu w komórce wartości 128-kasowanie 0)
82,83---------lewy i prawy margines(normalnie 2 i 39)
87------------wartości z instrkcji GRAPHICS dla używanego trybu graficznego
755-----------odwrócenie znaków (4-odwrócone 0 normalne)

40.ZAKOŃCZENIE.

Co zrobić jak program nie chce działać tak jak chcemy lub działa inaczej niż powinien.Po pierwsze nie kopać psa,nie tłuc młodszego rodzeństwa,nie walić pięściami w komputer ,magnetofon nie mówiąc o monitorze.

Najlepiej zrobić przerwę.Przejść się na spacer,pograć w piłkę. Następnie zacząć od analizy programu.

1.Pamiętaj,że komputer liczy od 0.

2.Sprawdź ilość danych w liniach DATA.Jak jest za mało komputer zgłasza błąd,ale przy zmienionym programie wyświetlania komunikat może być niewidoczny.Jak danych jest za dużo,znaki będą miały inne kształty niż zaplanowałeś.

3.Sprawdź skoki GOTO i GOSUB.Może zmieniłeś numery linii i program skacze do linii nie istniejącej.

4.Sprawdź czy są instrukcję NEXT w pętlach FOR.

5.Sprawdź dokładnie dane w linii DATA czy nie ma kropki zamiast przecinka lub litery O zamiast zera.

6.Sprawdź czy jest prawidłowa ilość parametrów w instrukcjach USR(wywołanie podprogramów w języku maszynowym).

7.Nie kasuj nigdy wersji poprzedniej programu.Zawsze coś się może zdarzyć.A zaczynać od nowa nie zawsze się chce.

8.Zapisuj programy tylko na jednej stronie taśmy.

9.Ważne programy przechowuj na co najmniej dwóch taśmach(a nawet trzech).

10.Pamiętaj:komputer zawsze zrobi to co mu powiesz,ale sam nie domyśli się czego Ty od niego chcesz.

Na tym chciałbym zakończyć przegląd najprostszych technik grafiki znakowej.Są one dostępne z ATARI BASIC'a lub przy pomocy prostych procedur maszynowych.Nie omówiliśmy jeszcze wielu ciekawych rzeczy np.płynnego przesuwu ekranu.Ale są to już

techniki wymagające dość długich procedur w języku maszynowym. W zasadzie nie zrobiliśmy nic w trybach 12 i 13.A nadają się one szczególnie do tworzenia szczegółowych i bardzo kolorowych map. Spróbuj może przenieść na komputer którąś z gier fantazy np. ODKRYWCY NOWYCH SWIATOW,LABIRYNT SMIERCI lub jakąś inną firmy ENCORE.Oczywiście możesz opracować własny scenariusz.Jest też bardzo ciekawa seria książek WEHIKUŁ CZASU.Bardzo nadają się do przeniesienia na komputer w postaci gry tekstowo-graficznej.

Czeka również cała niewykorzystana dziedzina programów edukacyjnych,gdzie opisane techniki graficzne mogą oddać nieocenione usługi.Jeżeli masz zamiar używać często któregoś z programów z kasety załączonej do książki (np.EDYTOR ZNAKÓW i KBR),to skopiuj każdy z nich na oddzielną kasetę po kilka razy bez przerw.Będziesz mógł je wygodnie używać.Pisz programy i eksperymentuj.Pamiętaj: żaden program nie zniszczy komputera! To nie jest nasza gospodarka,tu możesz bezpiecznie eksperymentować .Życzę Ci wielu dobrych,ciekawych i szybko działających programów!!!

AUTOR

41.LITERATURA.

Zródla procedur i programów wykorzystanych w niektórych rozdzialach:

ROZDZIAŁ 13	IKS 5/1987(program okrąg)
ROZDZIAŁ 19	IKS 10/1988
ROZDZIAŁ 21	GRAFIKA ATARI
ROZDZIAŁ 22	IKS 7/8/1987
ROZDZIAŁ 23	IKS 12/1987
ROZDZIAŁ 26	KOMPUTER 12/1988,9/1987
	BAJTEK 4/1988
ROZDZIAŁ 28	GRAFIKA ATARI CZ.II
ROZDZIAŁ 34	KOMPUTER 9/1986,7/1988
ROZDZIAŁ 35	IKS 1/1987
ROZDZIAŁ 36	IKS 7/8/1988
ROZDZIAŁ 38	BAJTEK 2/1988,4/1988
	KOMPUTER 3/1988

UKŁADY SCALONE TTL SERII UCY74 I ICH ZASTOSOWANIE
---Jan Pieńkos,Janusz Turczyński