

```

><><   TRICKS OF THE TRADE   ><><
><><><><><   TUTORIAL #11   ><><><><><
><><><><><><   MIKE WHITE   ><><><><><
><><><><><><   COUNTY LINE #2   ><><><><><
><><><><><><   R.D.#1 BOX 373   ><><><><><
><><><><><><   WAKEMAN , OHIO   ><><><><><
><><><><><><   44889   ><><><><><

```

LAST MONTH I SAID THAT I WOULD SHOW YOU THE MARTIAN AND THE MAZE, BOTH FROM THE 16K QUADRA, RUNNING IN AB! HERE'S THE MAZE:

```

10 CLEAR ;C=20253;%(C)=-32728;CALLA
20 GOSUB 90;%(C)=-32664;CLEAR ;CALLA
30 GOSUB 90;%(C)=-32600;CLEAR ;CALLA
40 GOSUB 90;%(C)=-32536;CLEAR
50 CY=24;CX=-10;PRINT "5:00
60 CX=-28;PRINT "FUZE TIMER
70 GOSUB 90;&(9)=-1;CALLA;GOSUB 90
80 GOSUB 90;&(9)=63;CLEAR ;STOP
90 FOR D=0TO 3000;NEXT D;RETURN
ENTER>A=20237;FOR B=A TO A+56STEP 2;INPUT %(B);NEXT B
ENTER>[NUMBER STRING] PRESSING [GO] AFTER EACH ENTRY
(NOTE) [READ EACH COLUMN DOWN WHEN ENTERING]

```

8669	10456	-32536	2563	-24574	1402	21856	96
20258	128	-12032	20265	876	5600	-24422	
-43	13311	-32567	-393	5984	-32518	27253	
-3789	2048	0	21728	-32514	25962	224	

[READ EACH COLUMN DOWN] MEANS THAT THE SECOND ENTRY MUST BE 20258, NOT 10456! THIS IS MACHINE CODE. IF IT'S NOT RIGHT WHEN YOU RUN IT, YOU WILL PROBABLY GET A [RESET]. IF YOU DO GET IT RUNNING, THE MAZE WILL APPEAR IN IT'S TRUE FORM, A SINGLE CHRDIS CHARACTER (SEE VOL.5 PG.14,15,37, AND 72 ARCADIAN). THE SCREEN IS THEN CLEARED, AND THE MAZE APPEARS AGAIN AT 2x SIZE. THEN, IT IS SHOWN AT 4x SIZE, AND, AS IT APPEARS IN THE GAME, AT 8x SIZE. THIS WAY, ALL THE MAZE BLOCKS ARE 8x8 PIXEL SQUARES, MAKING THE CHARACTER MOVEMENT AN EASY 8 PIXEL JUMP. HOWEVER, THE CHARACTER MUST BE 3 BYTES WIDE TO COVER A 20 BLOCK SCREEN (20x8=160), AND THERE ARE 24 PIXELS COVERED BY A 3 BYTE WIDE CHARACTER (8x3=24) (8 PIXELS PER BYTE). AT 8x SIZE THAT'S (24x8=192), WHICH WILL NEVER FIT ON THE SCREEN WITHOUT A "WRAPAROUND" OF 4 BLOCKS! THE PART THAT "WRAPS" IS KEPT ALL WHITE, AND THE MAZE IS XORED ONTO THE SCREEN, SO THE "WRAPAROUND" REMAINS INVISIBLE, AND DOESN'T INTERFERE WITH THE SCRATCHPAD. (IT "WRAPS" INTO THAT ALSO)!

AS FOR THE APPEARANCE OF THE MARTIAN, A FLYING SAUCER SENDS A BEAM DOWN, THEN THE BEAM GETS CUT OFF, AND THE MARTIAN APPEARS OUT OF IT HEAD FIRST! &(10) CANNOT BE USED, AS PART OF THE BEAM IS STILL VISIBLE, AND CONVENTIONAL ERASING GOES TOO SLOW TO LOOK RIGHT. THE FOLLOWING DEMONSTRATOR (IN AB) SHOWS HOW IT WAS DONE:

```

10 CLEAR ;NT=-1;FOR X=-60TO 0;GOSUB 20;NEXT X;GOTO 50
20 &(22)=-1;&(20)=150;&(17)=30;BOX X-1,31,31,8,2;BOX X,30,29,5,1
;BOX X,30,31,3,1;BOX X,33,9,3,1;RETURN
30 M=27;N=-40;P=-1;R=3
40 FOR Y=MTO NSTEP P;BOX 0,Y,7,1,R;NEXT Y;RETURN
50 &(17)=90;&(20)=0;GOSUB 30;N=-31;GOSUB 40;C=19276
60 FOR %(B)=128TO 136;%(B-3)=C;CALLA;C=C+256;NEXT %(B)
70 &(22)=0;FOR D=0TO 2000;NEXT D;&(22)=-1
80 GOSUB 30;M=N;N=27;P=1;R=2;GOSUB 40
90 FOR X=0TO 60;GOSUB 20;NEXT X;CLEAR ;NT=2
ENTER>A=20260;FOR B=A TO A+28STEP 2;INPUT %(B);NEXT B;B=A+10
ENTER>[NUMBER STRING] PRESSING [GO] AFTER EACH ENTRY
(NOTE) [READ EACH COLUMN DOWN WHEN ENTERING]

```

8669	-43	10331	-13871	256	32591	119	23915
20274	30771	128	128	14593	25443	30549	

THIS PROGRAM IS A SIMPLE CARTOON, WHERE A FLYING SAUCER FLYS ONTO THE SCREEN AND BEAMS DOWN OUR MARTIAN (FROM 16K QUADRA) RAY GUN IN EACH HAND. THEN HE BEAMS UP, AND THE SAUCER FLYS OFF. THE MARTIAN IS A "CHRDIS" CHARACTER THAT APPEARS ONE BYTE AT A TIME. LINE #60 DOES THE TRICK. THE CHARACTER FONT IS REDUCED TO A ONE BYTE CHARACTER, AND THE CHARACTER NUMBER INCREASES AS THE SCREEN LOCATION GOES DOWN (C=C+256). THE CHARACTER IS XORED ONTO THE SCREEN, SO IT HAD TO BE PROGRAMED IN AS A NEGATIVE CHARACTER. IN OTHER WORDS, WHEN FIGURING OUT MY CHARACTER PIXEL COLORS, I SET THE WHITE TO APPEAR BLACK AND THE BLACK TO APPEAR WHITE, EXCEPT THE LEFT COLUMN WHICH FALLS ON ALREADY WHITE BACKGROUND (MY BEAM IS 7 PIXELS WIDE, SEE LINE #40). THE CHARACTER COULD APPEAR IN THE OPPOSITE DIRECTION, FEET FIRST, IF THE LOOPS WERE REVERSED. THIS WOULD ALLOW FOR A WAY TO RAISE A STAGE CURTIN ON A PLAY, IF THAT IS WHAT YOU WANT. IF YOU ENTER> %(A+8)=2139, AND CHANGE THE LAST 5 "POKED IN" NUMBERS TO: 79, 7196, 32520, 2090, AND 8724, THE CHARACTER "PLOPS" ONTO THE SCREEN, AND COULD BECOME ANAMATED WITH MORE PROGRAMING. BE CAREFULL THOUGH! IN AB OR BB, "PLOPING" A CHARACTER DOWN LIKE THIS, WIPES OUT ANY TEXT THAT HAPPENS TO LAY "UNDERNEATH" THAT PART OF THE SCREEN THAT THE CHARACTER WILL OCCUPY. TO SEE THE TEXT, USE: &(0)=0;&(1)=90;&(2)=132;&(3)=6;&(9)=0. &(9)=63 RETURNS TO NORMAL.

I COULD GO ON TALKING ABOUT FANCY DISPLAY TRICKS, BUT THERE ARE TWO OTHER TOPICS OF GREATER IMPORTANCE THAT I HAVEN'T HARDLY MENTIONED YET, TIME SHARING, AND STRINGS! TIME SHARING IS A WAY OF MAKING 1800 BYTES GO FARTHER, BY USING THE SAME MEMORY TO DO MULTIPLE TASKS. VARIABLES ARE THE MOST READILY TIME SHARED, (SEE ABOVE PROGRAM). THE "B" VARIABLE STEPS BY TWO'S DURING THE TIME WE ARE INPUTING DATA, THEN BECOMES 20270 (A+10) DURING THE TIME THE PROGRAM GETS RUN. WE CAN DO THIS BECAUSE THESE TWO NUMBERS NEVER NEED TO BE KEPT AT THE SAME TIME! WE CAN ALSO TIME SHARE A STRING, OR EVEN THE WHOLE TEXT! HOW? YOU MAY ASK? SEE QUADRA, OR THIS COLUMN NEXT MONTH. KEEP BUGGIN' *