

CLAIM JUMPER
1982

96 SHEETS • 5 x 5 QUAD
10 $\frac{1}{8}$ x 7 $\frac{1}{4}$ • 53-110



NATIONAL BLANK BOOK COMPANY, INC.
Holyoke, Massachusetts 01040-Made in USA

PRINT DISC LIST

10 FOR I = 0 TO 20
20 A = 15392 + I
30 PRINT A, PEEK(A)
40 NEXT I

GET:

15392	112
15393	112
15394	112
15395	66
15396	64
15397	60
15398	2
	2
	2

B = 15396
POKE B, 0 } DISK PAGES
POKE B+1, 0 } 0 and 1

IMPORTANT!

please return to

Groy Chang

~~2265 Deborah Dr~~

~~Santa Clara CA 95050~~

~~(408) 248-5228~~

~~(408) 738-4400 X312~~

Earl Brown

~~XXXXX~~

Hayward's

Synapse

220 Pearl St

~~Santa Clara CA 95060~~

~~(408) 423-2445~~

John

11/11/11

Debi

Box 1

425 25 16 1
 XXXX hex

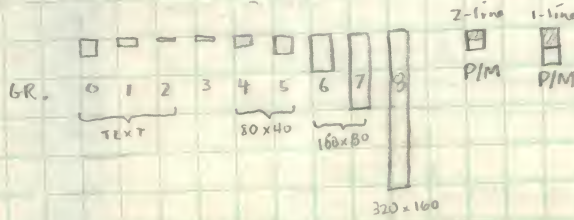
memory scan counter $\rightarrow$

$$\begin{array}{r} 3FFF \\ 3C40 \\ \hline 3BF \\ +1 \\ \hline 3C0 \end{array}$$

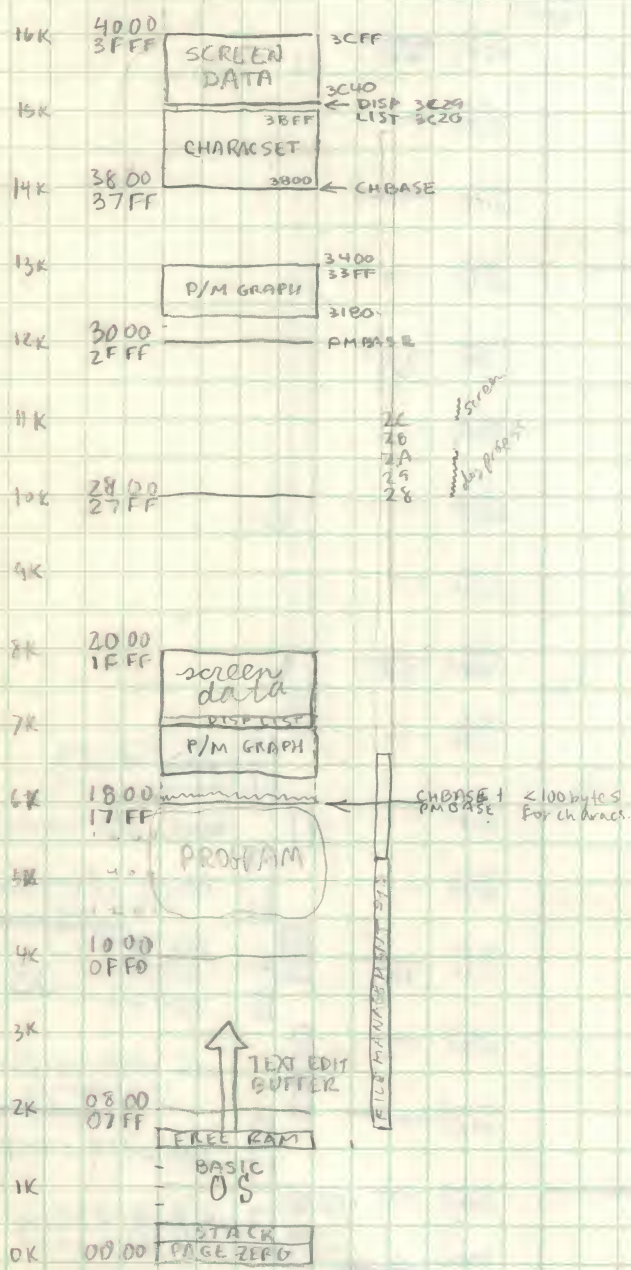
3C0 = 960 D.

locations
 for charac. names

Mov. 2000, 4000
 Display 2000, 4C 00 90
 chang 2000, 4C 00 90
 disasm 2000
 move dest < start. and (m)
 28C09
 registers
 step trace



3800
37FF
36FF

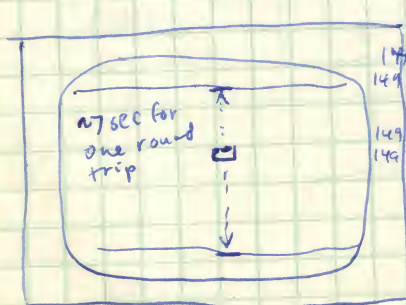


EXPERIMENT WITH PLAYER/MISSILE GRAPHICS UP/DOWN ONLY

Program to move a small rectangle player smoothly up and down screen.

10	BR.8	
20	POKE 559, 6	enable P/M DMA, single line
30	POKE 53248, 120	horizontal position, range 41-200
40	POKE 704, 88	set P0 color to pink
50	I = 56	page # for memory space data - why 56? = PEEK(106) - 8 (why?)
60	POKE 54279, I	store page # in BASE reg
70	POKE 53277, 3	Enable P/M DMA
80	J = I * 256 + 1024	Player loc BASE + offset
90	K = 120	BASIC routine, start
100	K = K + 1	line
110	POKE J + K, 255 : POKE J + K - 5, 0	blank
120	IF K = 223 THEN GOTO 200	
130	GOTO 100	
200	K = K - 1	
210	POKE J + K, 0 : POKE J + K - 5, 255	
220	IF K = 37 THEN GOTO 100	
230	GOTO 200	
93	D = J + 37	
94	C = J + 223	
95	B = J + K	
100	B = B + 1 : POKE B, 255 : POKE B - 5, 0	
	IF B >= C THEN GOTO 200	
130	GOTO 100	
200	B = B - 1 : POKE B, 0 : POKE B - 5, 255	
	IF B <= D THEN GOTO 100	
230	GOTO 200	

↓ "SPEEDUP" still ~ 7 sec per cycle



~7 sec.
round trip →

horizontal movement of space ship

```

90 FOR A = 1 TO 6
92 READ B
94 POKE J+128+A, B : NEXT A
100 I = I + 1 : POKE 53248, I : IF I = 200 THEN
    GOTO 300
110 GOTO 300
200 I = I - 1 : POKE 53248, I : IF I = 50 THEN
    GOTO 100
210 GOTO 300
300 IF STICK(0) = 11 THEN GOTO 200 (LEFT)
310 IF STICK(0) = 7 THEN GOTO 100 (RIGHT)
320 GOTO 300 (LOOP IF NO MOVE)
DATA 149, 149, 126, 126, 149, 149

```


1806



ANGLIA
1983

TIME TO EXECUTE BASIC STATEMENTS

```

10 FOR I=1 TO 1000
11 A=25      A=25
22 A=25      B=25
33 A=25      C=25
44 '
55 '
66 '
77 '
88 '
99 '
100 '
110 NEXT I
    
```

```

1 TO 100
PRINT
PRINT
PRINT
    
```

2 sec { FOR I=1 TO 1000
NEXT I

$$\begin{array}{r} 20 \text{ sec} \\ - 2 \text{ sec} \\ \hline 18 \text{ sec} \end{array}$$

$$\begin{array}{r} 20 \text{ sec} \\ - 2 \text{ sec} \\ \hline 18 \text{ sec} \div 10,000 \end{array}$$

$$\sim 34 \text{ sec} \div 1000 \Rightarrow \boxed{34 \text{ msec per PRINT}}$$

$$60 \text{ cycles (Hz)} = 16.7 \text{ msec/cycle}$$

$$1 \text{ MHz} = .001 \text{ msec/cycle}$$

$$\boxed{16.8 \text{ msec per } A=25}$$

```

FOR I=1 TO 500
10 A=STICK(0)
:
100 A=STICK(0)
    
```

$$27 \text{ sec} - 1 \text{ sec}$$

$$26 \text{ sec} \div 5000$$

$$\boxed{4.2 \text{ msec per } A=STICK(0)}$$

```

FOR I=1 TO 500
10 SOUND 1,111,111
20 " " " "
:
100
NEXT I
    
```

$$34 \text{ sec} \div 5000 \quad \boxed{6.8 \text{ msec per SOUND '1,111,111'}}$$

$$\text{POKE } 20 \text{ sec} \div 5000$$

$$\boxed{4.0 \text{ msec / POKE 53248,X}}$$

$$A=A+1 \quad 16 \text{ sec} \div 5000$$

$$A=A+1$$

$$:$$

$$A=A-1$$

$$A=A-1$$

$$\boxed{3.0 \text{ msec / increment or decrement}}$$

conclusion
BASIC IS TOO SLOW!

~~LDX YELST~~
~~LDX REDST~~

~~LDA~~
~~LDA YELST~~
~~AND REDST ANY?~~
~~BMI HH~~
~~LDA \$278~~
~~CMP #50F~~
~~BEQ TYEL~~
~~LDA \$279~~
~~CMP #50F~~
~~BEQ TYEL~~
~~HH JSR ENBST1~~
~~JSR ENBST2~~
~~RTS~~

add lines

TYEL
LDA YELST
CPX #01
CMP #1
BNE TRED
JSR ENBST1
LDA REDST
CMP #580
BNE RDED
LDX #2
JSR SELF

TRED
LDA REDST
CPY #1
CMP #1
BNE TST1
JSR ENBST2
LDA YELST
CMP #580
BNE YDED
LDX #0
JSR SELF

TST2
LDA \$279
CMP #50F
BEQ RTS2
LDA #1
STA REDST
LDA #0
STA YELST

TST1
LDA \$278
CMP #50F
BEQ RTS2
LDA #1
STA YELST
LDA #0
STA REDST
JMP SDF

SDF LDA FRAME
SEC
SBC #1
STA DFLAG
RTS2 RTS

RDED
LDA #50F
STA STIKZ
LDA FRAME
CMP DFLAG
BNE TST2
LDA #580
STA REDST
BNE TST2 / UNL

YDED
LDA #50F
STA STIK1
LDA FRAME
CMP DFLAG
BNE TST1
LDA #580
STA YELST
BNE TST1 / UNL

1870. The year was a very dry one
and the crops were much injured
by the drought. The wheat was
very poor and the corn was
also much injured. The
cattle and sheep were
also much injured by the
drought. The year was a
very hard one for the
people of the West.

ASY

DMA CTL	C022F<2E
COLPMØ	C02C0<58
PMBASE	CD407<30?
GRCTL	CD01D<03
Moiz pç	CD000<37-C8



3200

3210

326A

3280

PLAYER/MISSILE GRAPHICS, HEX LOADER FOR ASSEMBLY PROGRAMS, 7/18/81

MOVE PLAYER (or M) w/ STICK

SEE BASIC
MANUAL
FOR WHAT I DID

10 GR. 8

20 POKE 559, 46

2x line graphics

30 ~~HPφ = 150~~

40 POKE 704, 88

50 I = 56

60 POKE 54279, I

70 POKE 53277, 3

80 J = I * 256 + 512

82 ~~Vφ = 16~~

85 ~~POKE 53248, HPφ~~

88 K = J + 5

90 L = J - I

91 M = J + 4

92 ~~I = 56~~

93 ~~POKE 53248, I~~

94 ~~POKE 53248, 0~~

set color = PINK
page # for gr. - why 56??

set page #
enable DMA

starting address

because last 4 pages of
RAM - near top

384 for missiles

almost off screen

Horizontal Location Pφ, range 50-200

Vertical Position and player data

vertical range
data range

30-223 1-line 16-111 2-line

0-255 (φφ-FF)

L + Vφφ

J + Vφφ

K + Vφφ

M + Vφφ

K + Vφφ

100 Vφφ = 16 : HPφ = 60

110 PRINT HPφ, Vφφ

115 IF STICK(0) = 4 THEN GOTO 200

120 IF STICK(0) = 13 THEN GOTO 300

130 IF STICK(0) = 11 THEN GOTO 400

140 IF STICK(0) = 7 THEN GOTO 500

150 GOTO 110

200 Vφφ = Vφφ - 1

202 IF Vφφ ≤ 16 THEN Vφφ = 16 : GOTO 215

204 POKE J + Vφφ, 255

206 POKE K + Vφφ, φ

215 GOTO 110

300 IF Vφφ ≤ 16 THEN GOTO 110

302 Vφφ = Vφφ + 1

304 IF Vφφ ≥ 106 THEN Vφφ = 107 : GOTO 315

306 POKE L + Vφφ, φ

308 POKE M + Vφφ, 255

315 GOTO 110

move up

test for top of screen
redraw

next action

move down

test for bottom of screen
redraw

next action

400 HPφ = HPφ - 1 : IF HPφ ≤ 55 THEN HPφ = 55

402 POKE 53248, HPφ

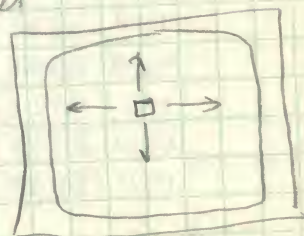
415 GOTO 110

500 HPφ = HPφ + 1 : IF HPφ ≥ 200 THEN HPφ = 200

502 POKE 53248, HPφ

515 GOTO 110

move left, test for edge
poke data
next



53252 for missileφ

7/23/81

BASIC DISPLAY LIST

PERK	15392	=	D.112	=	70	} 24 blank lines
DIS393	3G21		112	=	70	
94	3G22		112	=	70	
95	3G23		66	42		
96	3G24		64	40		} 3G40 is memory scan counter.
97	3G25		60	3G		
98	17		2	2		} 23 lines of GR.O
:	:		2	2		
:	:		:	:		
:	:		:	:		
15420	3G3G		2	2		} jump to 3G20 and wait for end of vert blank
21	3G3D		65	41		
22	3E		32	20		
23	3F		60	3G		
24	40					

WOW!

another method. use GR5 and peek locations to see what disp list BASIC has set up; modify.

DNACTL PERK(559) = D34 = 22H
 SDLSTH (56) = D60 = 3G } 3FFF
 SDLSTL (560) = D32 = 20 } 3G20 = 15392
 2DF

space for disp list + memory space

```

2 I=-1
5 REM LOAD DISP LIST
TRY 10 FOR I=0 TO 1200
20 READ IN
30 POKE 12320+I, IN
40 NEXT I
50 REM GENERATE DISP DATA
60 FOR J=1 TO 1200
70 POKE 14356+J, INT(255*RND(1))
80 NEXT J
100 DATA 112,112,112,74,0,56
110 DATA 10,10,0,0,10,10,10,10,10,10 (ten)
120 DATA 10,10 10 (ten)
130 DATA 65,32,48,999
150 REM LOAD DISP LIST (TR w/ 2020 hex
160 POKE 561,48 (hex30
170 POKE 560,32 (hex20)
180 REM END DMA not needed
190 POKE
  
```

DISPLAY LIST START
 3020 = 12,320

38,000 - DMA START = D.14356

IT WORKS!

23
 40
 560

QUICK PROGRAM DISPLAYS
 RANDOM CHARACTERS

```

10 FOR I=1 TO 1000
20 POKE 15424+I, PNT(255*RND(1))
30 NEXT I
  
```


Experiment with DIM 7/20/81

DIM A\$(1), B\$(1), C\$(1), D\$(1)
PRINT ADR(A\$), ... ADR(D\$)

2114 2115 2116 2117 $\nwarrow$ ADR for strings only

Maximum DIM possible w/ my memory is between

D. 13,300 and 13,350

try to peek at data
in these locations - doesn't
work - why?

DISPLAY LIST PROCEDURE

1. disable DMA - @ DMACTL

2. load Display List Counter DLSTH, DLSTL : OS shadow $\begin{matrix} \text{DLSTH} \\ \text{DLSTL} \end{matrix}$

3. counter points to Display List - enter Blank Jump Display
loads into Display Instr. Reg.

3-byte Instruction loads Memory Scan Ctr. : 2 bytes
counts, 4 MSB fixed, points to 4K space

4. Memory Scan Ctr points to display data! (base)

ex. 1GR.5 80 pixels/line $\frac{1}{4}$ pixel/line $\times \frac{120 \text{ lines}}{\text{screen}} \approx 40 \text{ pixel/screen}$
mode: A = 20 bytes $\times$ 4 pixels/byte
colors: 4 $\uparrow$ 2 bits/pixel = 4 colors
grid 80x4

try 10 FOR I=1,46
20 READ INS
30 POKE 12320 + I, INS
40 NEXT I

50 REM GENERATE DISP DATA

60 FOR I=1 TO 1200

70 POKE 12592 + I, INT(255 * RND(1))

80 NEXT I

100 DATA 112,112,112,74,0,56,10,10,10,10,10,
10,10,10

110 DATA 10,10,10

120 DATA 10,10,10, 65,32,48

$\frac{3020 \text{ Hex} = 12,320}{1}$

3800 Hex = 12552

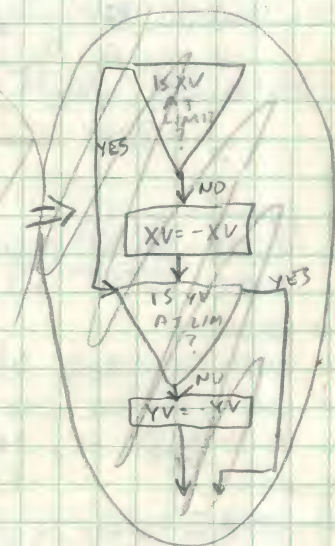
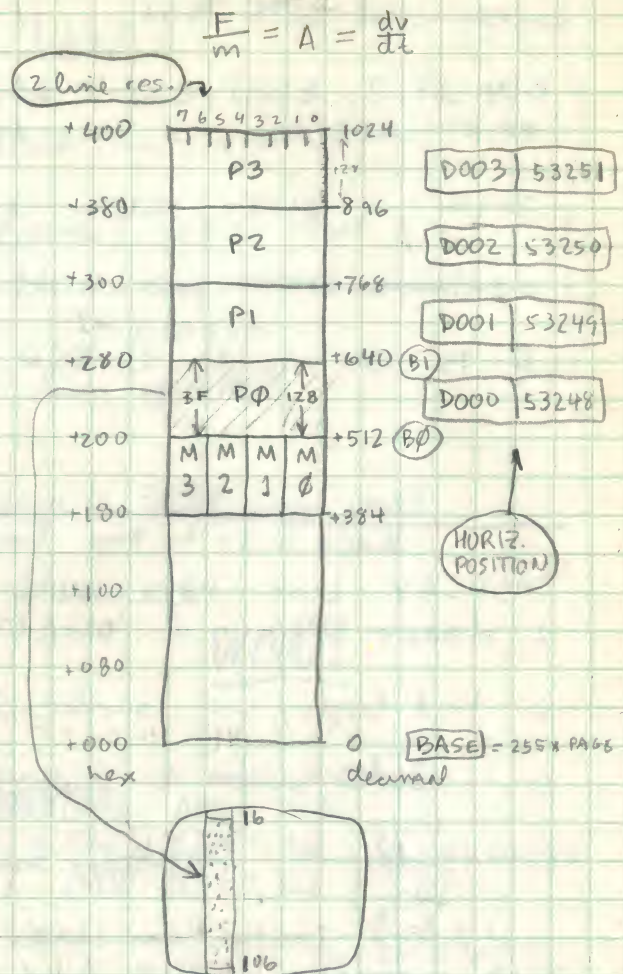
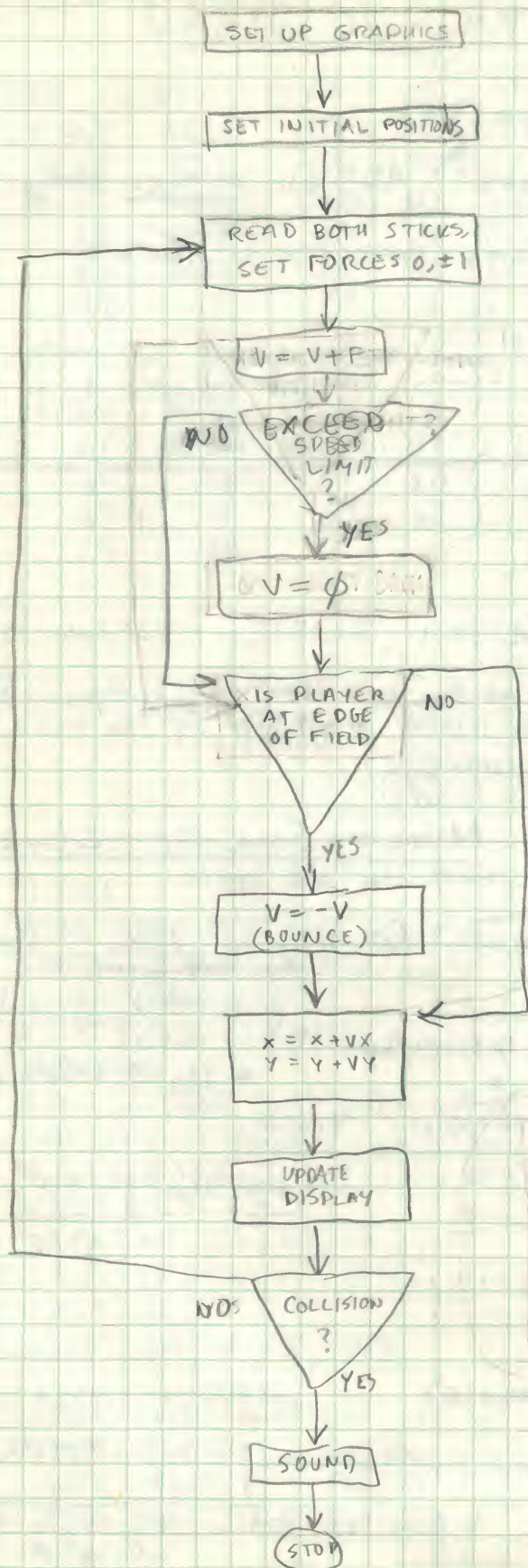
Hex 70,70,70,4A,00,38,0A,0A,
41,20,30

DISPLAY LIST CTR H.D. 54275 } 30 = 480 lines
L.O. 54274 } 20 = 320 lines

DMACTL: D.559

} 2

add 1?



TAG GAME - PHYSICS - Velocity, Force, etc.

X Pφ	X P1	X coordinate	55 to 200
Y Pφ	Y P1	Y coordinate	16 to 106
VX Pφ	VX P1	X-velocity	-XVMφ to XVMφ
VY Pφ	VY P1	Y-velocity	-YVM1 to YVM1
FX Pφ	FX P1	X-force	-1, 0, 1
FY Pφ	FY P1	Y-force	-1, 0, 1
VXMXφ	VX M1	X max velocity	5 ? & 3 ?
VYMXφ	VY M1	Y max velocity	5 ? & 3 ?
XMx		} coordinates of edge of field	
Y Mx			
XMN			
Y MN			

10 GR.5

20 POKE 559, 46 DMACTL, 2-line GR.

30 POKE 704, 88 Pφ pink

40 POKE 54279, 33 P/M base address (pg.)

50 POKE 53277, 3 GRCTL GR. control enb. DMA

60 BASE = 33 * 256

70 Bφφ = BASE + 512 : Bφ1 = BASE + 513 : Bφ2 = BASE + 514

80 Bφ3 = BASE + 515

72 XMN = 55 : XMx = 200 } fix edges
74 YMN = 16 : YMx = 106

100 FX PΦ = φ : FY PΦ = φ initial force : XMx

110 XPφ = 60 : YPφ = 60 : VX PΦ = 0 : VY PΦ = φ initial position and velocity

120 VXMXφ = 5 : VYMXφ = 8/10 for now!

130 IF STICK(0) = 14 THEN FY PΦ = -1 :

135 IF STICK(0) = 13 THEN FY PΦ = 1

140 IF STICK(0) = 11 THEN FX PΦ = -1 :

145 IF STICK(0) = 7 THEN FX PΦ = 1

150 VX PΦ = VX PΦ + FX PΦ : VY PΦ = VY PΦ + FY PΦ v = u + at

160 IF (VX PΦ) > VXMXφ OR (VY PΦ) > VYMXφ THEN VX PΦ = 0 : VY PΦ = φ stop if speed limit >

170 IF XPφ > XMx OR XPφ < XMN THEN VX PΦ = -VX PΦ } bounce

180 IF YPφ > YMx OR YPφ < YMN THEN VY PΦ = -VY PΦ

175 IF XPφ < XMN THEN VX PΦ = ABS()
185 IF YPφ < YMN THEN VY PΦ = ABS()

190 XPφ = XPφ + VX PΦ : POKE 53248, XPφ update X

200 POKE Bφφ + YPφ, φ : POKE Bφ1 + YPφ, φ } erase Y

210 POKE Bφ2 + YPφ, φ : POKE Bφ3 + YPφ, φ

220 YPφ = YPφ + VY PΦ

230 POKE Bφφ + YPφ, 255 : POKE Bφ1 + YPφ, 255 } new Y

240

250 GO TO 130

IT WORKS, but it's much too jerky

RSB

LDA #500

STA #3C24

A9
00
8D
24
3G
40

D

164
0
141
18 244
60 02
64

A9
07
8D
2F
3G
40

F4
02

H D
E7B3 59315 64 40
6 44 2C
7 15 0F
8 212 D4
9 16 10
20 3 03

RTI
BIT
D40F
BPL
03
etc

why would it work?

GR. X+16 doesn't work disp list pointers

investigate GR.0 peek(561) = 60 30 } 3020
 peek(560) = 32 20 }

GR3 561 = 62 3E } 3E4E = 15950
 560 = 78 4E }
GR5 59 3B } 3B6A
 106 6A }
GR7 47 2F } 2FAZ
 162 A2 }

52	179	C3
3	231	E7
4	178	B2
5	231	E7
6	178	B2
7	231	E7
8	178	B2
9	231	E7
520	190	BE
1	255	FF
2	17	11
3	235	EB
4	144	90
5	234	EA
6	209	D1
7	234	EA

LDX (XFF), Y

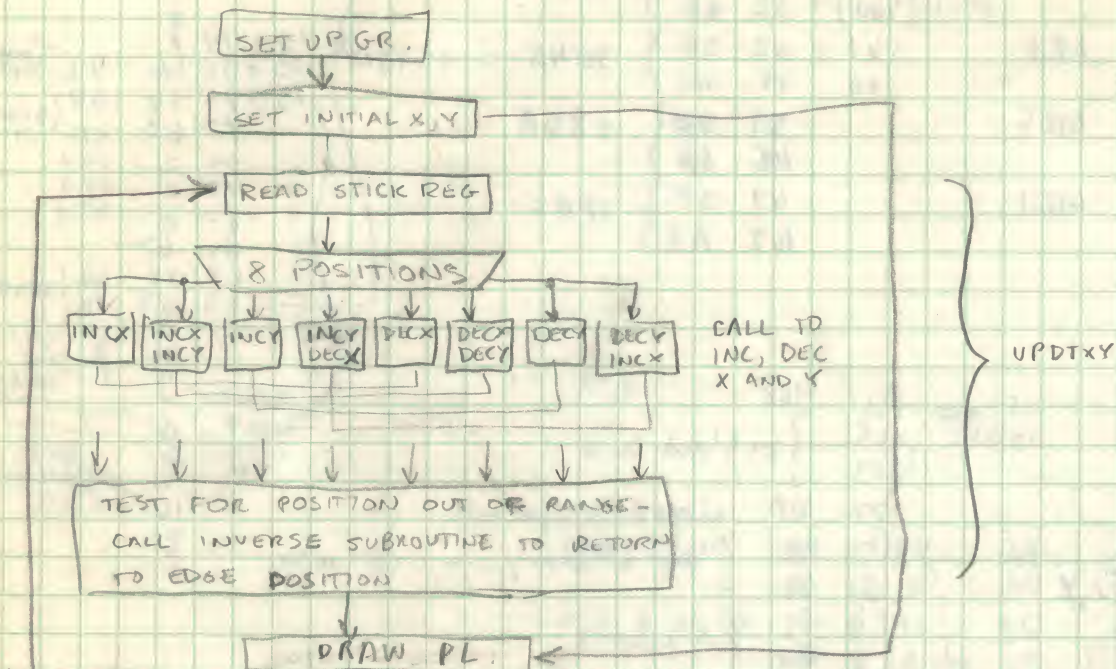
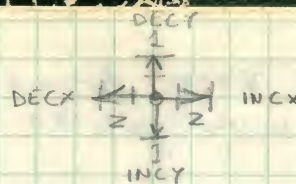
159950	D	HEX		
15952	112		} 24 black lines	
	112			
	112			
	72	48	mode 3, 3-byte inst.	
	112	70	} dipt memory scan ctr @ 3E70	
	63	3E		
	8	8	} i, mode 3	
	8	8		
15975	66	42	mode 2, 3-byte inst.	
	96	60	3F60 → mem. scan ctr	
	63	3F		
	2	2		
	2	2		
	2	2		
	65	41	} J, wait	
	78	4E		} J 15950
	62	3E		

	D	H
	8	
	8	
	8	
15974	8	int M6
15975	6b	42 347R
15976	9b	60}
77	b3	3F} 3F60 11
78	2 -	G.R.D
79	2 -	G.R.D
80	2 -	B.R.D
81	65-	
82	78	4E} 3e4E
83	62	3E} 11
	0	
	0	
	0	
memory	0	↑ 15950 JMD back S
16224 -	0	memory scan ctr.
scan -	0	G.R.P.
cds -	31	IF H30A
	16	10 11
	0	
	0	
E70	0	

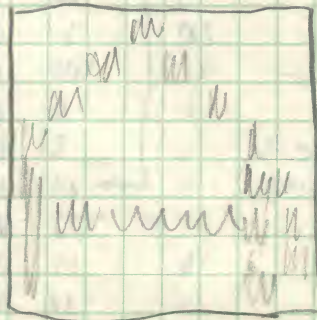
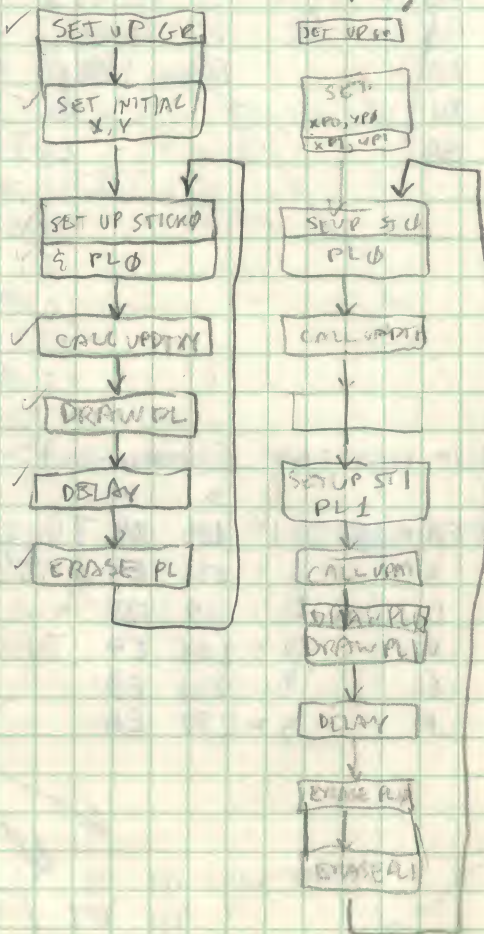
DISP LIST INT EXPERIMENT 8/8/81

NMI	(FFFA)	= (6553D) = 18B B4] E7B4 = (59316) = 4Y ZC BIT(D40F)			
B	1	= 231 E7] = 15 OF and (D40F)			
C	2	= 119 77 = 212 D4 S= bit 7 = MSB of (D40F)			
D	3	= 228 E4 = 16 10 BPL 03 branch if plus			
E	4	= 243 F3 = 3 03 " 6G JMP(0200) VDSLST			
F	5	= 230 E6 = 0 00 PHA,			
					48 LDA # \$D40F
					AD 0F
					D4 AND 20
					29 20
					FO BEQ

FLOWCHART 1 PLAYER, 4-DIRECTION MOVEMENT



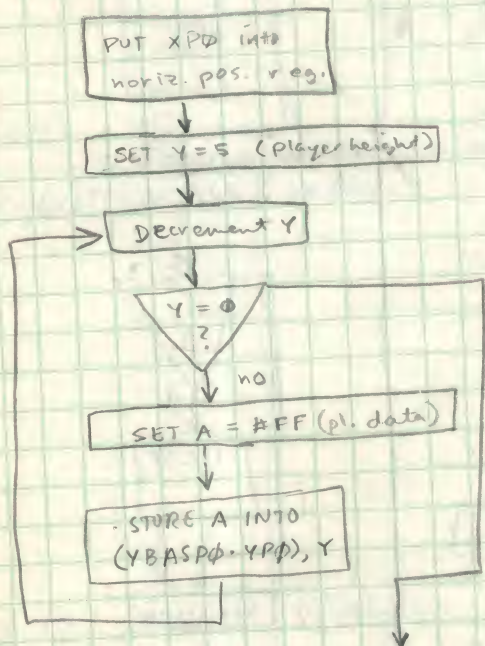
2 players



UPDATE DISPLAY

```

UPDATE X DISPLAY
LDA XPO
STA HPOSP
UPDATE Y DISPLAY
LDA #00
TAX
LDA #05 10-LINE PLAYER
TAY
DEY
BEQ
LDA #FF DISPLAY RECTAG. BLOCK
STA Y
  
```



COMMENT: ASSUME X,Y CURRENT
 ERASE LDA #04 PLAYER HEIGHT-1
 TAY INTO Y REG.
 ERLN LDA #00 ZEROS ERASE PL.
 ERLN STA (YBASP), Y WRITED TO R. LOC.
 STA (YBASP1), Y
 DEY
 BNE ERLN

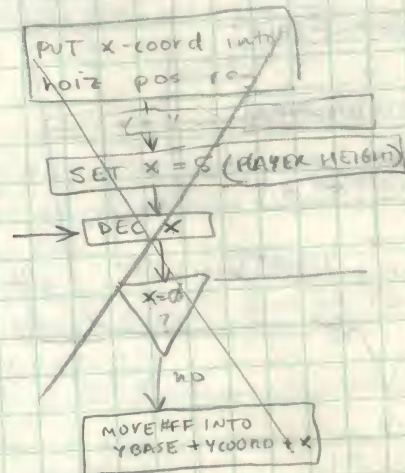
initial X, Y

```

30 LDA #XSTART
31 STA XPO
35 LDA #YSTART
40 STA YPO
  
```

D= DATA M= MEM LOC

M8 XPO X-coord
 M8 YPO Y-coord
 M16 HPOSP horiz pos. reg.
 -PL DATA player data
 M8 YBASP Y-position mem. base
 = YPO + 1



```

UPDATE X-COORD
LDA XPO
STA HPOSP
LDA #05
TAY
NEXT PLAYER LINE
DEY
BEQ
LDA #FF
STA (YBASP), Y
JMP NXTPLN
  
```

OR put decrement test at and save line, label

003E DRAW 100 LDA XPO
 110 STA HPOSP
 120 LDA #04
 130 TAY
 140 LDA #06, Y
 150 STA (YBASP), Y
 160 STA (YBASP1), Y
 170 DEY
 180 BNE NXTPLN

better

PLAYER HEIGHT -1
 INTO Y REG.

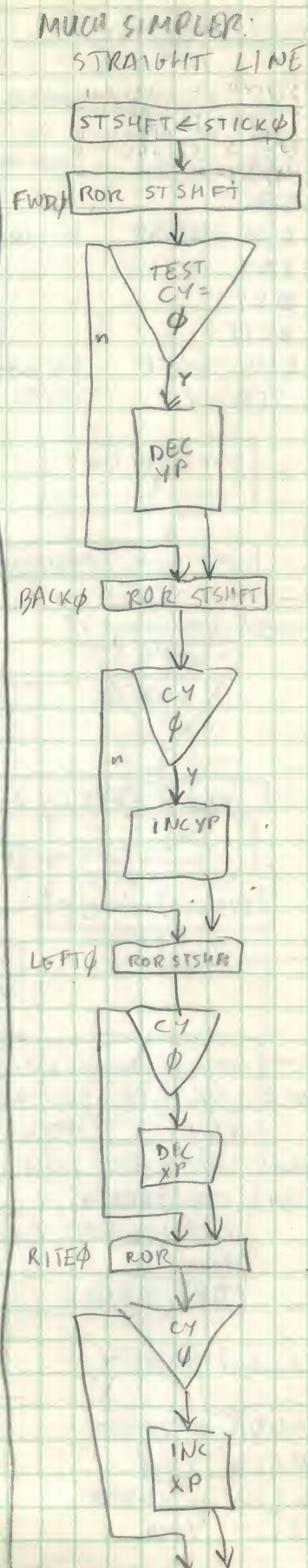
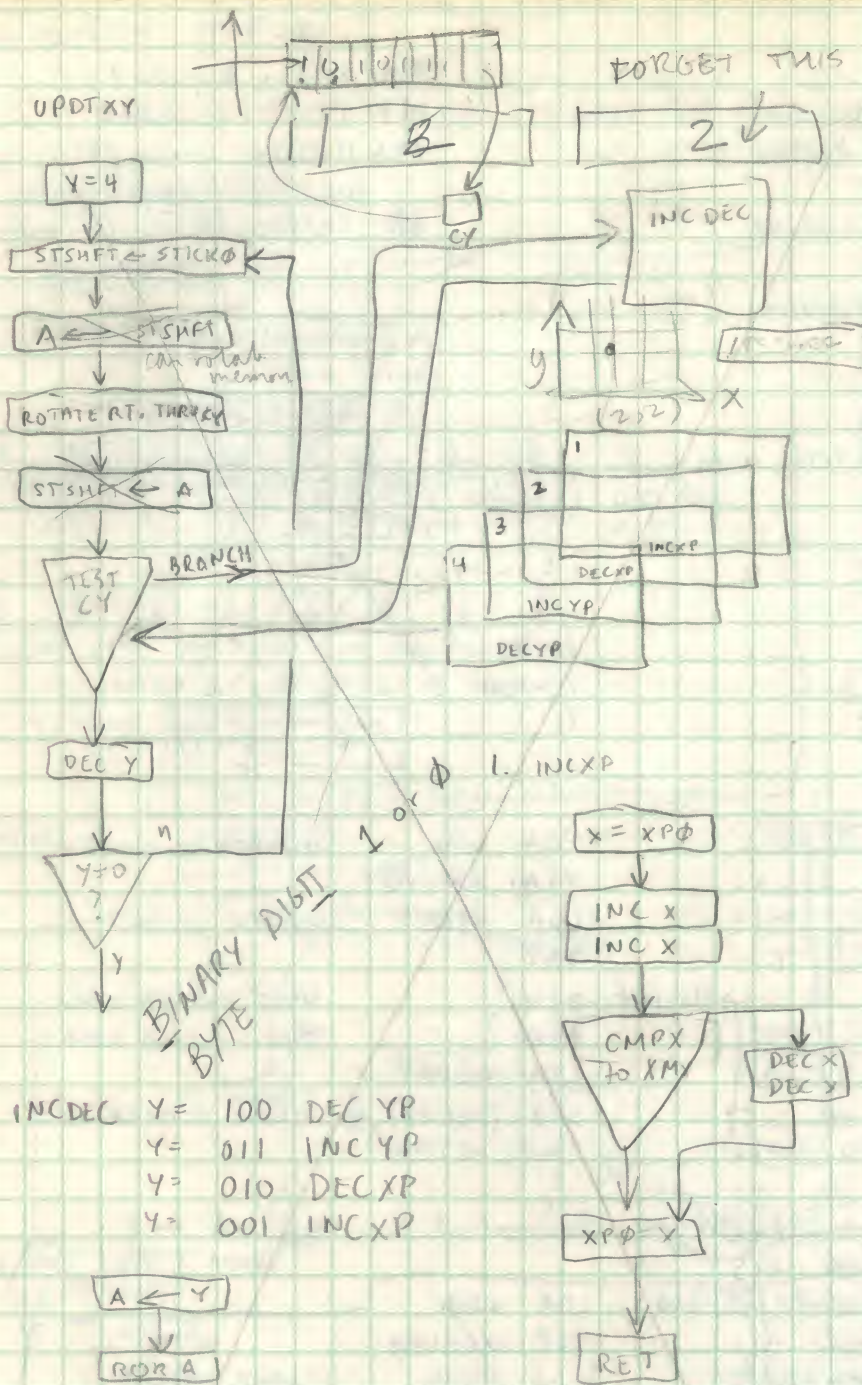
(must use YPO!)

WRITE TO PLAYER 0
 WRITE TO PLAYER 1

Y 4,3,2,1,0

YBASP
 YPO

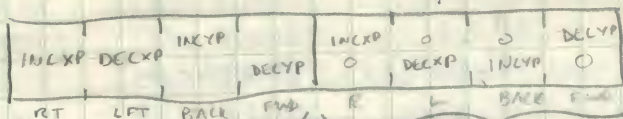
YBASE
 YCOORD



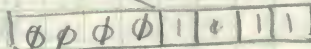
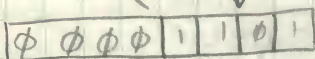
UPDTXY
STSHFT

PORTA

D300 H.
54,016 D.



H: 279
D: 633



278 H:
632 D.

0619

50 STUPP2 STUPP0

LDA YP0

PUT YP0

PREPARE FOR

FOR P1
add \$80

STA YP

INTD YP

UPDTXY SUBR.

LDA XP0

PUT XP0

STA XP

INTD XP

LDA STREG0

PUT STICK DATA

LDA YP
STA YP0
LDA XP
STA XP0

STA STSHFT

INTD STSHFT

JSR UPDTXY

UPDATE X AND Y

0628

78
80

0660

680

UPDTXY

NOP

UPDATE X AND Y COORDS.

700

FWD

ROR STSHFT

GET 'FWD' BIT

710

BCS BACK

GO ON IF STICK NOT FWD

720

LDA YP

FWD MEANS DEC YP

730

TAX

DEC YP BY 1

740

DEX

750

NOP

760

CPX #YMN

TEST FOR PL AT TOP EDGE

770

BCS EDJOKT

BRANCH IF OK

780

INX

OTHERWISE, CANCEL DEC

790

NOP

0672

800

EDJOKT

TXA

YP

NEW Y-COORD. ***

810

STA YP

820

BACK

ROR STSHFT

GET 'BACK' BIT

830

BCS LEFT

GO ON IF STICK NOT 'BACK'

840

LDA YP

BACK MEANS INC YP

850

TAX

INC YP BY 1

860

INX

870

NOP

880

CPX #YMX

TEST FOR PL AT BOT. EDGE

890

BCS EDJOKB

BRANCH IF OK

900

DEX

OTHERWISE, CANCEL INC.

910

NOP

920

EDJOKB

TXA

930

STA YP

NEW Y COORD. ***

940

rot

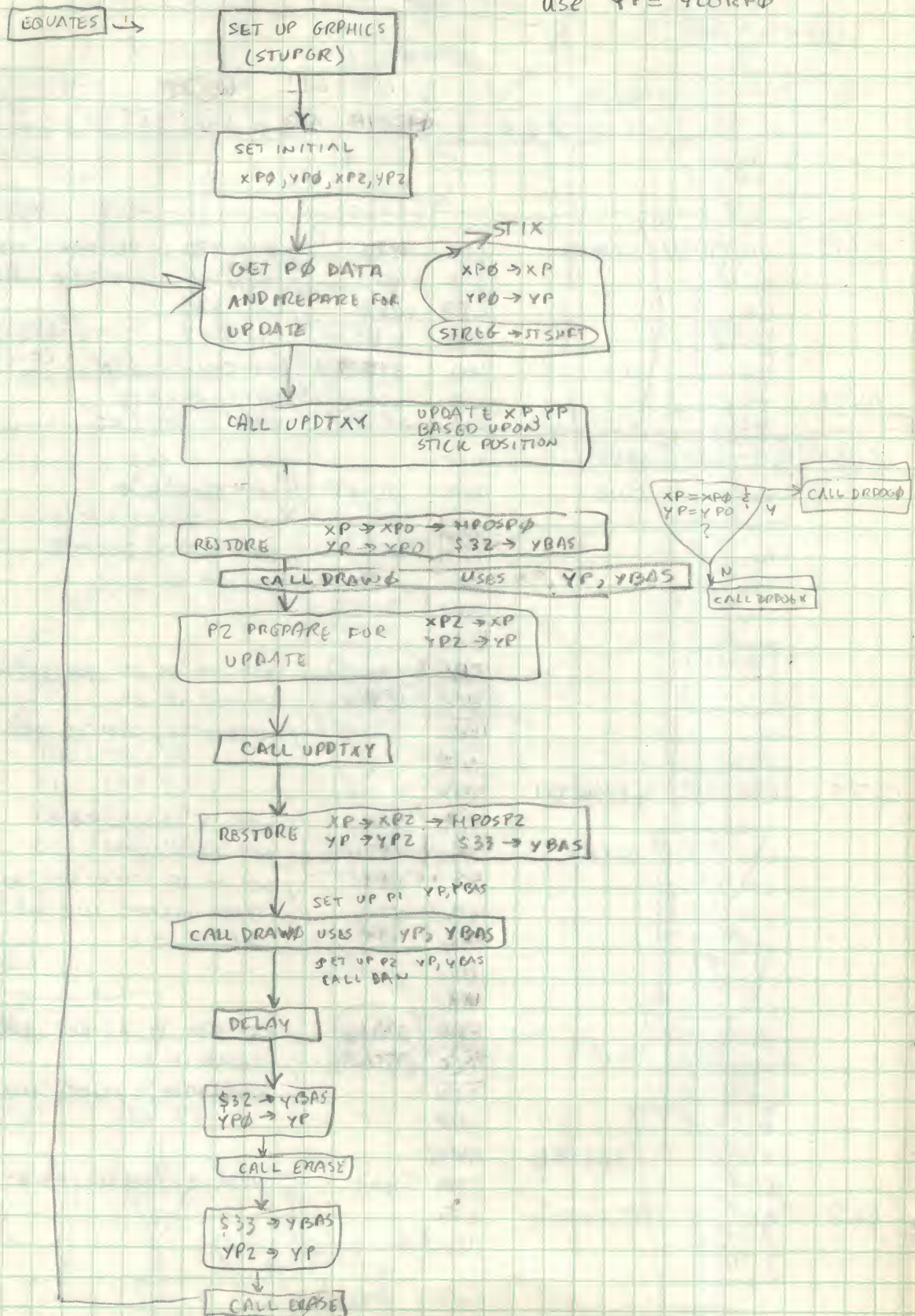
cont'd next page

68B

→ updone

2 players flowchart

USE YP = YCORP0



0640

200 DELAY LDX #02
 210 BOTSCR LDA VCOUNT
 220 CMP #D.112
 230 BCS BOTSCR
 240 MIDSCR LDA VCOUNT
 250 CMP #D.112
 260 BCC MIDSCR
 270 DEX
 280 BNE BOTSCR

0651

0653

290 ERASE LDA #04
 300 TAY
 310 LDA #00
 320 ERSLIN STA (XCOUNT),Y
 330 DEY
 340 BNE ERSLIN
 350 JMP STUPP0

65D

; NUMBER OF FRAMES DELAY, > 0
 ; ***
 ; STAY IN LOOP IF
 ; VCOUNT >= 112
 ; -----
 ; STAY IN LOOP UNTIL
 ; VCOUNT HITS 112 D.
 ; DECREMENT AND TEST
 ; SCREEN (33 MS) COUNTER
 ; PL HEIGHT - 1 ; ASSUME X,Y CURRENT

LEFT

ROR STSHFT
 BCS RIGHT
 LDA XP
 TAX
 DEX
~~DEX~~
 CPX #XMN
 BCS EDJOKL
 INX

; GET 'LEFT' BIT
 ; GO ON IF STICK NOT 'LEFT'
 ; LEFT MEANS DEC XP

EDJOKL

RIGHT

~~TRX NOP~~
 TXA
 STA (XP)
 ROR STSHFT
 BCS UPDONE
 LDA (XP)
 TAX
 INX
~~INX~~
 CPX #XMX
 BCC EDJOKR
 DEX
~~DEX NOP~~

; TEST FOR PL AT LEFT EDGE
 ; BRANCH IF OK
 ; OTHERWISE, CANCEL DEC

; NEW X COORD XXX
 ; GET 'RIGHT' BIT
 ; BRANCH IF STICK NOT 'RIGHT'
 ; RIGHT MEANS INC X

EDJOKR

UPDONE

TXA
 STA (XP)
 RTS

; TEST FOR PL AT RT. EDGE
 ; BRANCH IF OK
 ; OTHERWISE, CANCEL INC

; NEW X COORD. XXX

18

Sta. Loc.

VCOUNT

STREG

VMX

YMN

for 2 PL

HPOSP2 = \$

COLPM2 = \$

prod

STODPB

UPDTXY

FWD

EDJOKT

BACK

EDJOKB

LEFT

EDJOKL

RIGHT

EDJOKR

UPDONE

DRAW

NXTLIN

DELAY

BOTSCR

MIDSCR

ERSLIN

data

YP

XP

STSHFT

XP0

YP0

HPOSP0

DOG

.

.

.

.

Page 0

YBASP0

YBASP1

1 * = \$060F

2 YP = \$0600

4 XP = \$0601

6 STSHFT = \$0602

8 XP0 = \$0603

10 YP0 = \$0604

12 HPOSP0 = \$0605 DOOB

14 DOG = \$0606

7

8

9

A

16 YBASP0 = \$EC (16-bit#)

18 YBASP1 = \$CE (16-bit#)

20 VCOUNT = \$D40B

22 STREG = \$D300

X YCORP0 = \$CB

X YCORP1 = \$CE

YP0 = YCORP0 → CB X X YCORP0
 CC 3 Z YBASP0
 YP1 + \$80 = YCORP1 → CD X X YCORP1
 CE 3 Z YBASP1

DM

DMACTL

D400

2E

(PMDMA)

COLPM0

02C0

58

(P0 P1NK)

GRACTL

D01D

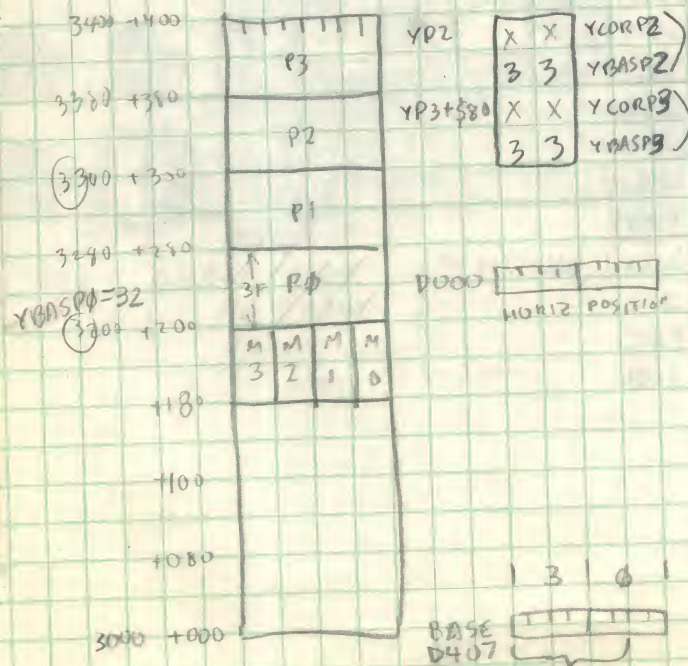
03

(ENB P1M)

STARTS AT 60F

DOG 0-37

47-



D407 = 30 PM BASE

0606-08 DOG 41

7 = 18 C1

8 = 38 FF

9 = 78 7F

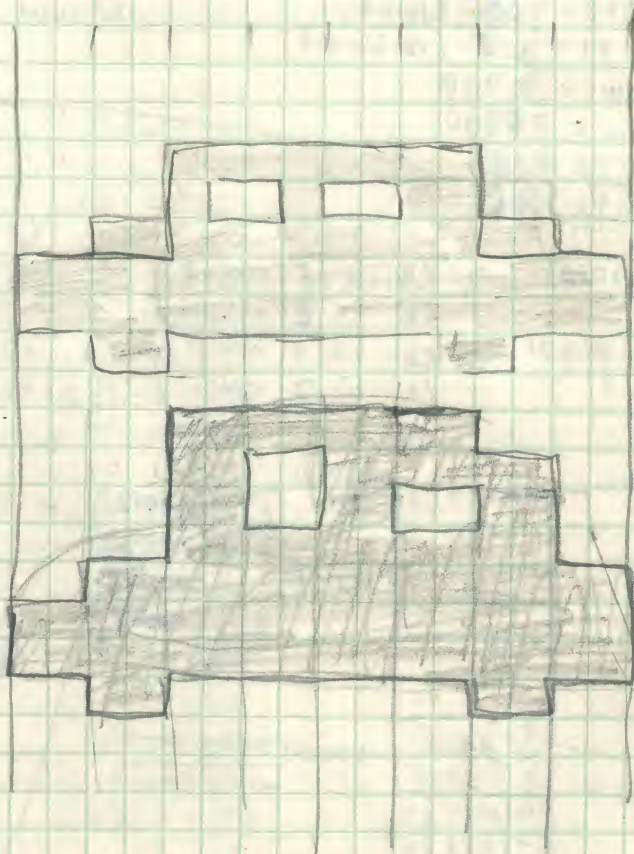
A = F8 63



SPOT



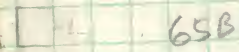
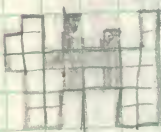
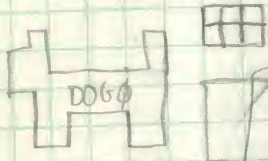
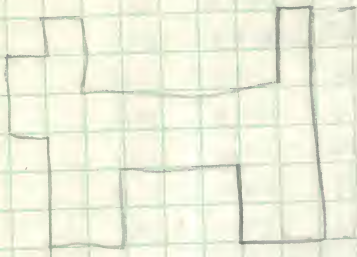
AND




```

STUPGR LDA #$2E
      STA DMACTL -
      LDA #$78
      STA HPOSP0 - < LDA #68
                      STA HOPSP2
      LDA #$58
      STA COLPM0 - < LDA #
                      STA COLPM2
      LDA #$30
      STA PMBASE -
      LDA #$3
      STA GRCTL -
      LDA #$32
      STA YBASP0 < LDA #533
                      STA YBASP2

```



65B

430
 440
 456
 460
 476
 480
 490
 506
 510
 511
 512

DRAWP0
 ↓
 DRAWP

NXTLN0

```

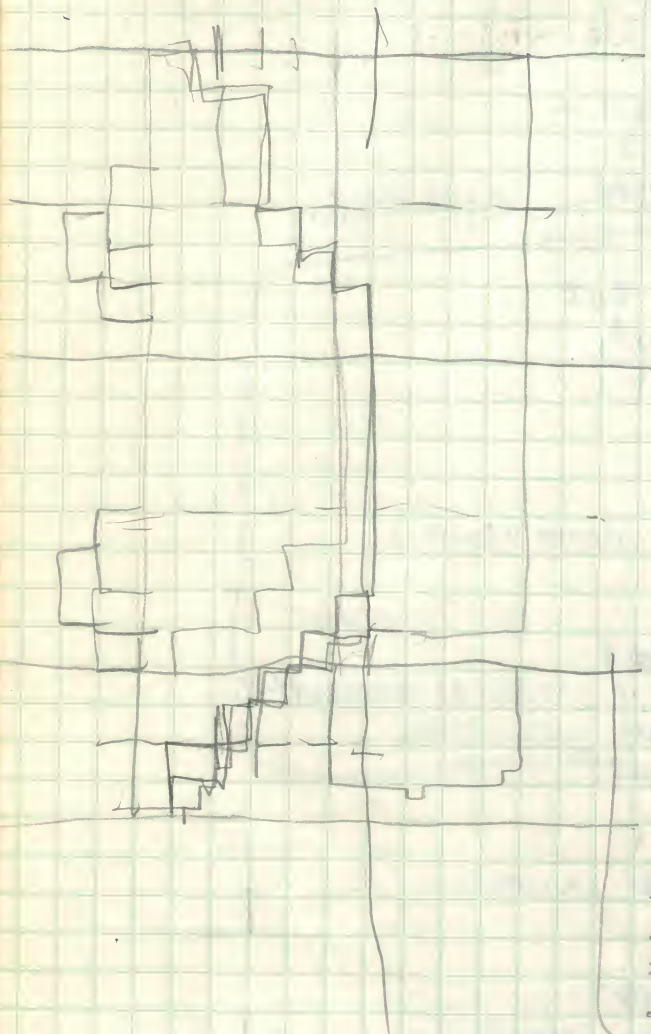
LDA XP      Dog2
STA HPOSP0
JSR DRAWP
LDA YP
NOP
STA YP
LDA #06 ; PLAYER HEIGHT
TAY
DEY
LDA $DOG,Y
STA (YP),Y ; STORE IN YBAS+Y,P+Y
BNE NXTLN
RTS

```

```

DRAWP0 LDA XP0
      STA HPOSP0 ; X COORD
      LDA YP0
      ORA #00
      STA #YCORP0
      LDA #$04 ; PLAYER HT. -1
      TAY
      LDA $DOG,Y ; DRAW DOG
      STA (YCORP0),Y
      DEY
      BNE NXTLN0

```

* = COLPM1
 .BYTE \$16
 * = COLPM3
 .BYTE \$36
 * = STA1
 .BYTE 00
 * = STA3
 .BYTE 00

PRIOR = \$26F

* = PRIOR

.BYTE \$31

* = XP0

.BYTE 100

* = XP2

.BYTE 180

* = DMACTL

.BYTE \$2E

* = HPSP0

.BYTE 100

* = HPSP2

.BYTE 180

* = COLPM0

.BYTE \$18

* = COLPM2

.BYTE \$38

* = PMBASE

.BYTE \$30

* = GRACL

.BYTE 3

* = YP0

.BYTE 50

* = YP2

.BYTE 80

Can I do
 this?
 probably
 not.

70 YP = \$CB
 80 YBAS = \$CC
 90 XP = \$0600
 100 XP0 = \$0601
 110 YP0 = \$0602
 120 XP1 = \$603
 130 YP1 = \$604
 140 XP2 = \$0605
 150 YP2 = \$0606
 160 XP3 = \$607
 170 YP3 = \$608

10 .OPT NOEJECT
 20 .TAB 4,8,12
 30 YMN = 16 ; VERT
 40 YMX = 106
 50 XMN = 55 ; HORIZ
 60 XMX = 200
 YMN1 = 144
 YMX1 = 234

180 HPOS P0 = \$D000
 190 HPOS P1 = \$D001
 200 HPOS P2 = \$D002
 210 HPOS P3 = \$D003
 222 YBAS01 = \$32
 230 YBAS23 = \$33

STAT = \$609
 STA1 = \$60A
 STA3 = \$60B
 JDIR = \$60C
 JDIR1 = \$60D
 JDIR3 = \$60E
 TRIG = \$610

240 VCOUNT = \$D40B
 250 STREG = \$D300
 260 STSHFT = \$060AF
 270 COLPM0 = \$2C0
 280 COLPM1 = \$2C1
 290 COLPM2 = \$2C2
 300 COLPM3 = \$2C3
 310 DMACTL = \$22F
 320 PMBASE = \$D407
 330 GRAC TL = \$D01D

YP 00CB 7 8
 YBAS 00CC 3 2
 ← add \$80 to Y coord of P1 or P3 before storing into YP

340 DOG0 = \$610
 350 * = \$610
 360 .BYTE 0, \$41, \$C1, \$FF, \$7F, \$63, \$63

STICK0 = \$278
 STICK1 = \$279
 TRIG0 = \$284
 TRIG1 = \$285

to draw one pixel line
 STA (YP), Y
 ← ranges 5 to 0
 0 = pressed
 1 = not pressed

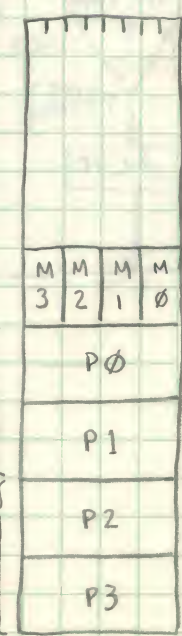
0622

370 * = \$0622
 380 LDA # \$2E
 390 STA DMACTL
 400 LDA #100
 410 STA HPOS P0
 420 LDA #200
 430 STA HPOS P2
 440 LDA # \$58
 450 STA COLPM0
 460 LDA # \$25
 470 STA COLPM2
 480 LDA # \$30
 490 STA PMBASE
 500 LDA #3
 510 STA GRAC TL
 520 LDA #50
 530 STA YP0
 540 LDA #80
 550 STA YP2
 560

JET = \$617
 * = \$617
 .BYTE 0, 0, \$14, \$1C, \$14

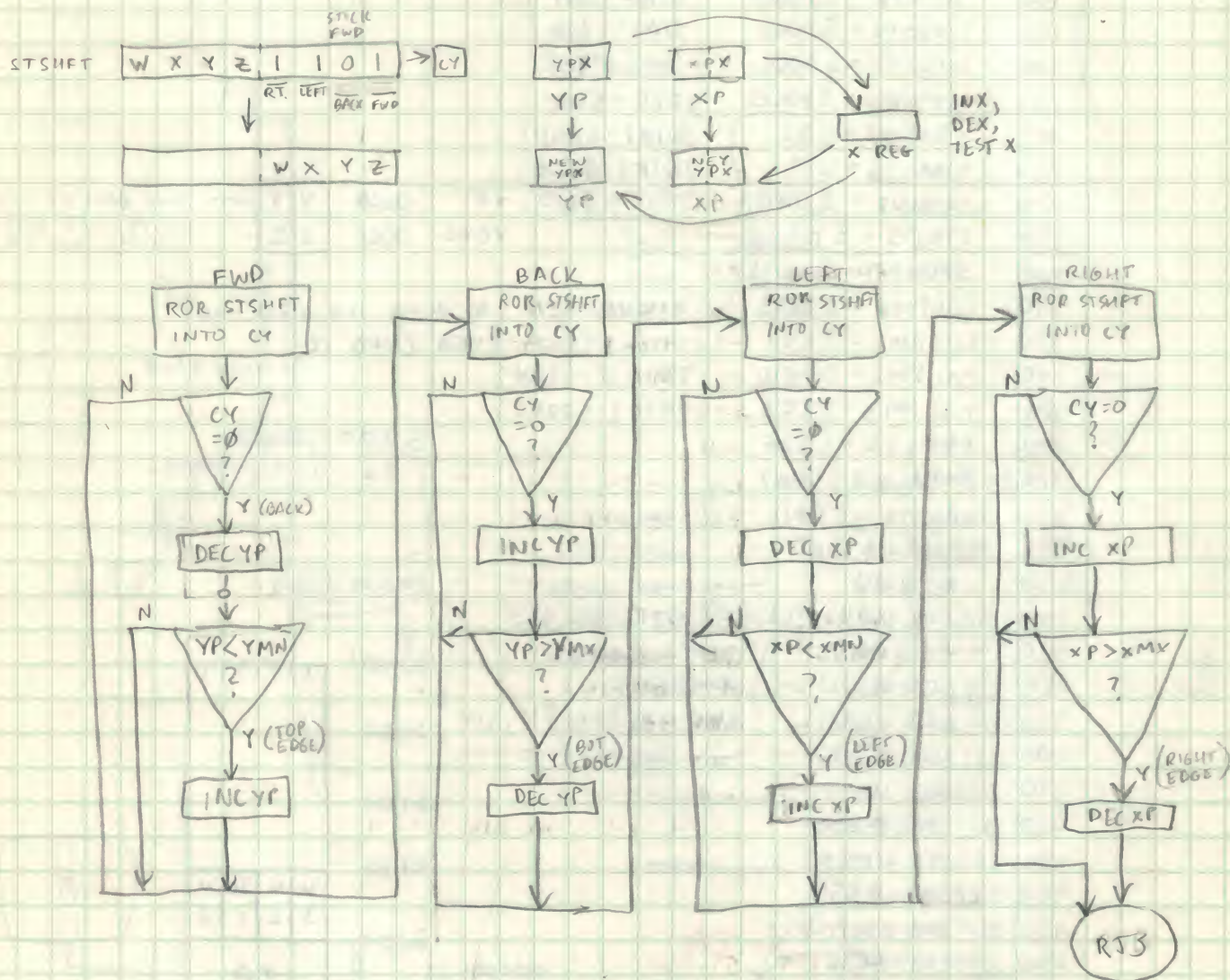
PMBASE = \$30

3000
 3080
 3100
 3180
 3200
 3280
 3300
 3380
 3400



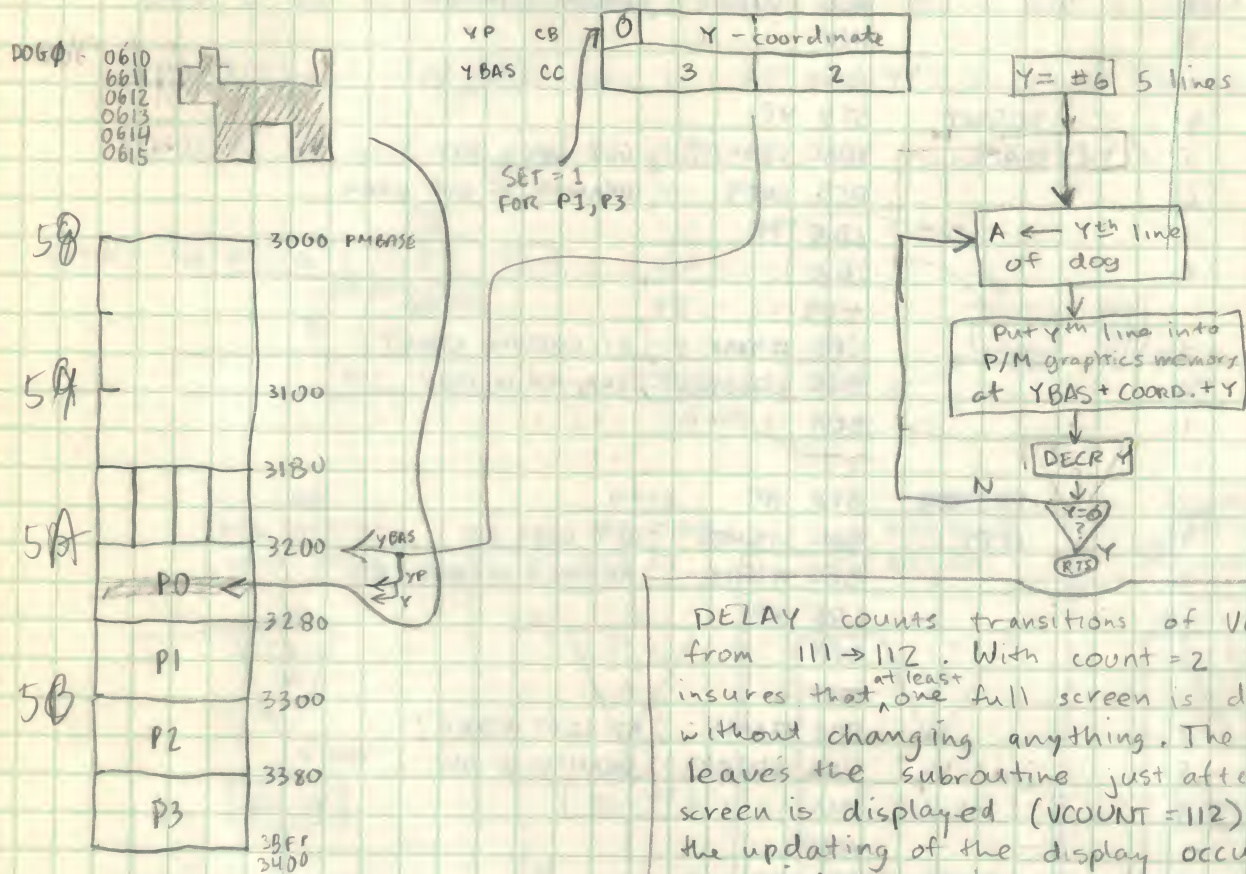
0652
0655

UPDTXY updates the X and Y coordinates of a player, based upon the position of the joystick. It assumes that the stick data is in STSHFT as shown below, and the X and Y coords $XP0$ ($XP2$) and $YP0$ ($YP2$) are in XP and YP . It places the new coords in XP and YP , and shifts STSHFT right 4 bits



0658	57	UPDTXY	ROR STSHFT	; GET FWD BIT
	58		BCS BACK	; BRANCH IF NOT FWD
	59		LDX YP	
	60		DEX	
	1		CPX #YMN	; AT TOP EDGE?
	2		BCS EDJOKT	; BRANCH IF OK
	3		INX	
0665	4	EDJOKT	STX YP	
0667	5	BACK	ROR STSHFT	; GET BACK BIT
A	6		BCS LEFT	; BRANCH IF NOT BACK
066C	7		LDX YP	
066E	8		INX	
F	9		CPX #YMX	; AT BOTTOM EDGE?
1	70		BCC EDJOKB	; BRANCH IF OK
3	1		DEX	
0674	2	EDJOKB	STX YP	
0676	3	LEFT	ROR STSHFT	; GET LEFT BIT
0679	4		BCS RIGHT	; BRANCH IF NOT LEFT
067B	5		LDX XP	
E	6		DEX	
F	7		CPX #XMN	; AT LEFT EDGE?
1	8		BCS EDJOKL	; BRANCH IF OK
3	9		INX	
0684	80	EDJOKL	STX XP	
0687	1	RIGHT	ROR STSHFT	; GET RIGHT BIT
	2		BCS UPDONE	; BRANCH IF NOT RIGHT
	3		LDX XP	
	4		INX	
	5		CPX #XMX	; AT RIGHT EDGE?
	6		BCC EDJOKR	; BRANCH IF OK
	7		DEX	
0695	880	EDJOKR	STX XP	
0698	890	UPDONE	RTS	

DRAWØ draws dogØ (6-byte data at DOGØ). DRAWØ assumes that YBAS (location 64) contains the page number for the particular player to be drawn, and that YP (location CB) contains the Y-coord. of the player. P1 and P3 are located in the 2nd half of a page in P/M graphics memory. Therefore, set the MSB of YP before entering DRAWØ with P1 or P3.



ERASE is the same as DRAWØ, except that it draws Ø's, or nothing, wiping out the player that was displayed for two frames (1/30 sec). Remember to load YP, YBAS with the Y-coord. and P/M graphics page no.

069A 900
 910
 920
 930
 940
 950
 960

DRAWØ
 → NXTLNØ

NØP
 LDY #6 ; PL HEIGHT
 LDA ØDOGØ, Y
 STA (YP), Y ; STORE IN P/M MEMORY
 DEY
 BNE NXTLNØ
 RTS

070C 1380
 1390
 1400
 1410
 2
 3
 4
 5
 071D 1460

DELAY
 → BOTSCR
 → MIDSCR

LDX #2 ; # FRAMES DELAY
 LDA VCOUNT
 CMP #112 ; STAY IN LOOP
 BCS BOTSCR ; IF VCOUNT >= 112
 LDA VCOUNT
 CMP #112 ; STAY IN LOOP UNTIL
 BCC MIDSCR ; VCOUNT HITS 112
 DEX ; TEST COUNTER
 BNE BOTSCR

06A5 970
 980
 990
 1000
 1010
 1020

ERASE
 → ERSLIN

LDY #6 ; PL HEIGHT + 1
 LDA #Ø
 STA (YP), Y ; ERASE LINE
 DEY
 BNE ERSLIN
 RTS

DRAWJ
 NXTLJ

LDY #4 ; JET HEIGHT
 LDA JET, Y
 STA (YP), Y ; DATA TO PM MEMORY
 DEY
 BNE NXTLJ
 RTS

DATA
EQUATES
SETUP GRAPHICS

SET INITIAL
COORDINATES
OF OBJECTS

STSHFT ← STREG
(2 STICKS DATA)

UPDATE P0
COORDS

XP0 → XP
YP0 → YP
XP → XP0
YP → YP0

XP2 → XP
YP2 → YP

UPDATE P2
COORDS

XP → XP2
YP → YP2

DRAW P0
IN MEM

XP0 → HDOSP0
YP0 → YP
\$32 → YBAS

DRAW 0

DRAW P2
IN MEM

XP2 → HDOSP2
YP2 → YP
\$32 → YBAS

DRAW 0

DELAY

ERASE P2

ERASE

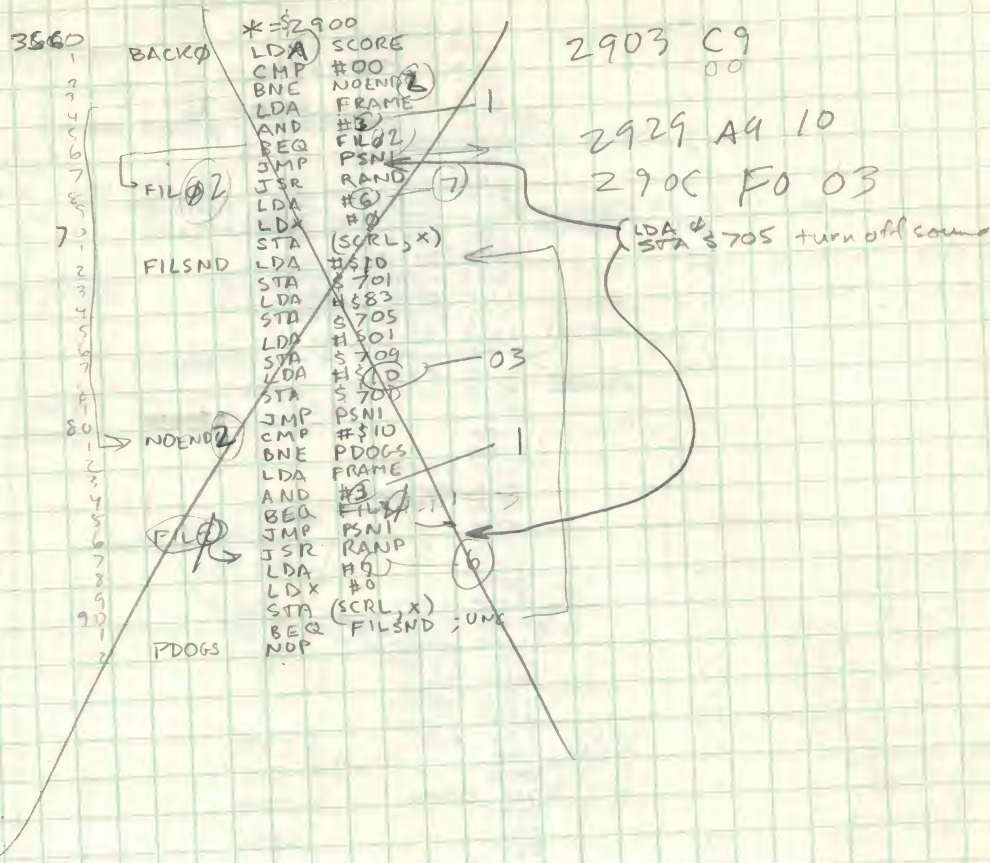
ERASE P0

ERASE

YP0 → YP
\$32 → YBAS

	1025	* = \$2000 ← STICK 0	
06AF	1030	STIX LDA STICK 0	; SET UP
	1046	STA STSHFT	; P0, P2 STSHFT
06BE	1050	UPDAT 0 LDA XP 0	; OLD COORDS
	1060	STA XP	; INTO XP, YP
	1070	LDA YP 0	
	1080	STA YP	
	1090	JSR UPDTXY	
	1100	LDA XP	; NEW COORDS ; FROM XP, YP
	2	STA XP 0	
	3	LDA YP	
	4	STA YP 0	
06C7	5	UPDAT 2 LDA XP 2	; OLD COORDS ; INTO XP, YP
	6	STA XP	
	7	LDA YP 2	
	8	STA YP	
	9	JSR UPDTXY	; NEW COORDS ; FROM XP, YP
	120	LDA XP	
	1	STA XP 2	
	2	LDA YP	
	3	STA YP 2	
06E8	4	DRWP 0 LDA XP 0	
	5	STA HPOS P 0	
	6	LDA YP 0	
	7	STA YP	
	8	LDA # \$32	
	9	STA YBAS	
	130	JSR DRAW 0	
06FA	AD0506	1 DRWP 2 LDA XP 2	
	8D02D0	2 STA HPOS P 2	
0700	AD0606	3 LDA YP 2 ←	79 22 06
	85CB	4 STA YP	79 36 40
	A933	5 LDA # \$33	
	6	STA YBAS	
	1370	JSR DRAW 0	
		DELAY	
071F	1470	ERS P 2 JSR ERASE	
	8	ERS P 0 LDA YP 0	
	9	STA YP	
	1500	LDA # \$32	
	1510	STA YBAS	
	1520	JSR DRAW 0 ERASE ←	
072E	1530	JMP STIX	

also need to poke PRIOR for PL color control



1. PUT IN ENDING ROUTINE
flash score
play music

2. PUT IN PITTER-PATTER

3. PUT IN START
DOG DAZE

(C) 1981 Gray Chang
PUSH STICK TO START

4. STICK → OUT OF ATTRACT MODE

5. DOG STOPS FLYING BONE X

change XMx

put \$14's in disp list; change CMP to 5

11030
11040

5
6
7
8
9
10
11
12
13
14
15
16
17

18
19
20
21
22
23
24
25
26

27
28
29
30
31
32
33
34
35
36
37
38

39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

62
63
64
65
66
67
68
69
70
71
72

73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500

GET SCOR DISP
LDY #10
LDA \$3CHC, Y
STA SCRDSP, Y
DEY
BNE SCRD
LDA #0
STA \$D208 AUDCTL

13
LDA #F0
STA FRAMX
LDX H0
STX NOTENO
INC NOTENO
LDX NOTENO
LDA NOTE, X
STA \$D200, AUDFI
BNE NOREST
LDA #A2, REST
STA \$D201, AUDCI

NOREST
LDA #A8, VOL
INC NOTENO
LDX #A
STA \$D201
PLNOT
TYA
PHA
JSR ENDFRX
JSR PLSCOR
PLA
TAY

DEY
BNE PLNOT
LDX NO 30
LDA NOTE, X
STA \$D200
BNE TIED
LDA #A0
STA \$D201
LDY #2
TIA
PHA
JSR ENDFRX
PLA
TAY
DEY
BNE PLTIE

PLTIE
LDA FRAMX
BEQ MUSCDN
INC NOTENO
JMP NXTNOT
BRK
BRK
BRK

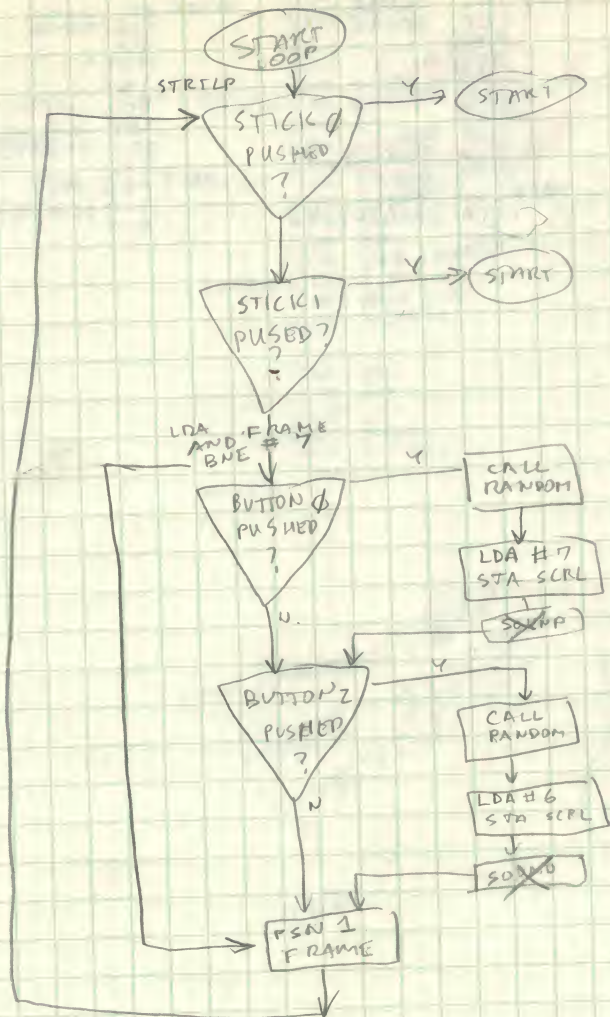
MUSCDN
ENDFRX
BOTSCX
LDA \$D40B
CMP #112
BCS BOTSCX
MIDSCX
LDA \$D40B
CMP #112
BCC MIDSCX
LDA FRAMX
AND #10
BEQ ON
BNE OFF
LDA SCRDSP, X
STA \$3CHC, X
DEX
BNE ON
BEQ XRET
LDA #0
STA \$3CHC, X
DEX
BNE OFF
RTS

ON
A
LDA SCRDSP, X
STA \$3CHC, X
DEX
BNE ON
BEQ XRET
LDA #0
STA \$3CHC, X
DEX
BNE OFF
RTS

OFF
XRET
RTS

LDX #10 (SCRDSP)

11900 SCRDSP = \$1C30
11901 FRAMX = \$1C2A
11902 NOTENO = \$1C2B
11903 NOTE = \$1C00
11904 X = \$1C00
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000



jscore

SCORE INC SCORE
LDA SCORE
CLC
ROR A
TAY
BCS HAF0
DEX
LDA #6
STA (\$3C4D), Y
RTS
HAF0 LDA #5D
STA (\$3C4D), Y
RTS

SCORE 2

SCORE2 DEC SCORE
LDA SCORE
CLC
ROR A
TAY
BCS HAF2
LDA #7
STA (\$3C4D), Y
RTS
HAF2 LDA #5D
STA (\$3C4D), Y
RTS

C3868<10,7C,5C,7F,5C,7C,
5C,
00

1. test for win
2. test for START
3. poke ATTRACT
4. test for time done

ENDFRAME

```

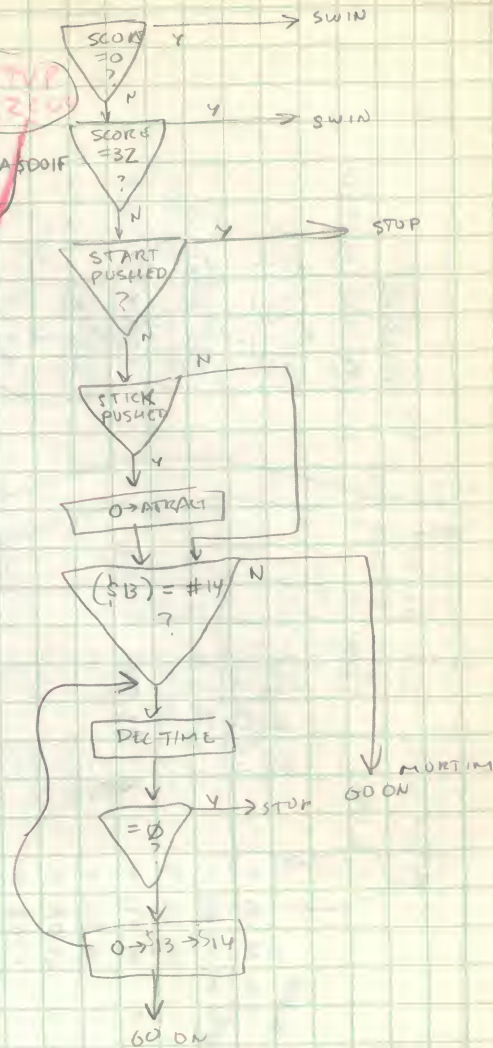
TIME = $
X = $1200
LDA SCORE
BEQ SWIN
CMP #20
BEQ SWIN
BNE NBWIN
JMP $1C50 ; MUSIC
NBWIN LDA #0 / STA $DDIF / LDA $DOIF
AND #1
BNE NOSTOP
JMP $1C50 ; MUSIC
NOSTOP LDA $D300 ; STICKS
EOR #1
BEQ UNTUCH
LDA #0
STA $4D ; ATTRACT
UNTUCH LDA $13
CMP #14 ; DEC
BNE MORTIM

LDA #0
STA $13
STA $14
DEC TIME
BNE MORTIM
JMP $
MORTIM

```

new RANGE = \$82F
 ENDFR = \$223D
 don't need to know

9320

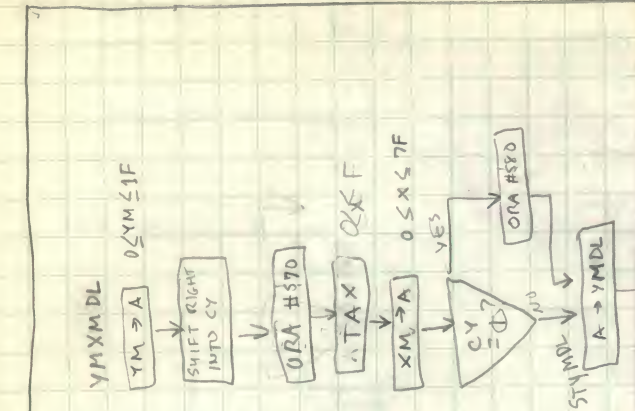


THINGS TO DO
 TO DOGS, JB75

PITTER-PATTER
 DEBUG STARTING
 SOUND

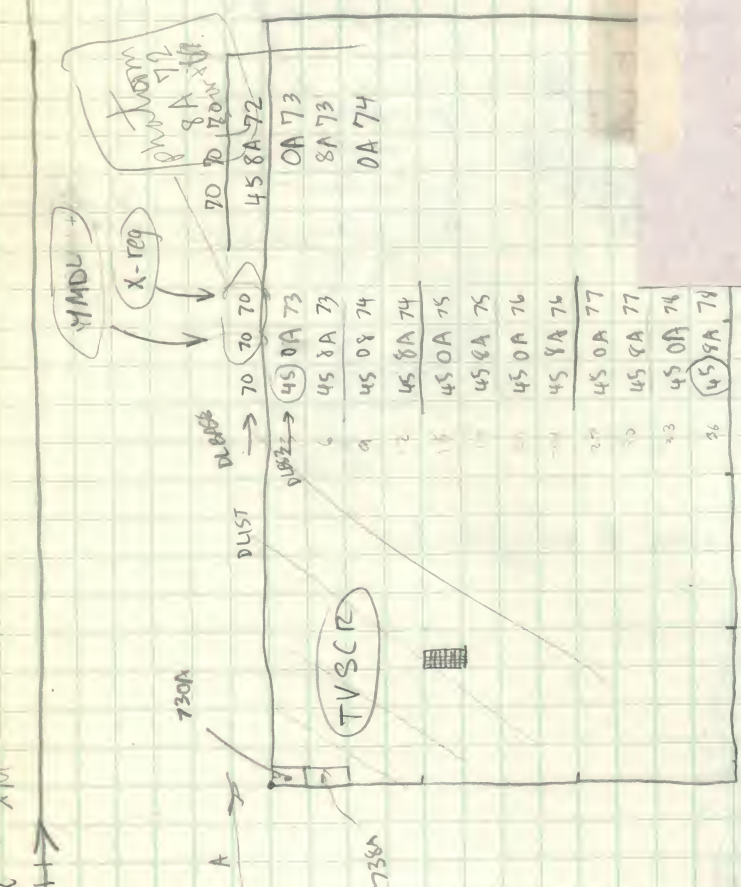
new page 0 assignment

- B0 FR1
- 1 RR1
- 2 FR3
- 3 FR4
- 4 AND1
- 5 AUD1
- 6 AUD2
- 7 AUD3
- 8 AUD4
- 9 VPD1
- A VPD2
- B VPD3
- C VPD4
- D BUR1
- E BUR2
- F BUR3
- C0 BUR4
- 1 XPD
- 2 XPD
- 3 XPD
- 4 XPD
- 5 YP1
- 6 YP2
- 7 YP3
- 8 YP3
- 9 XPD
- 0 YPD

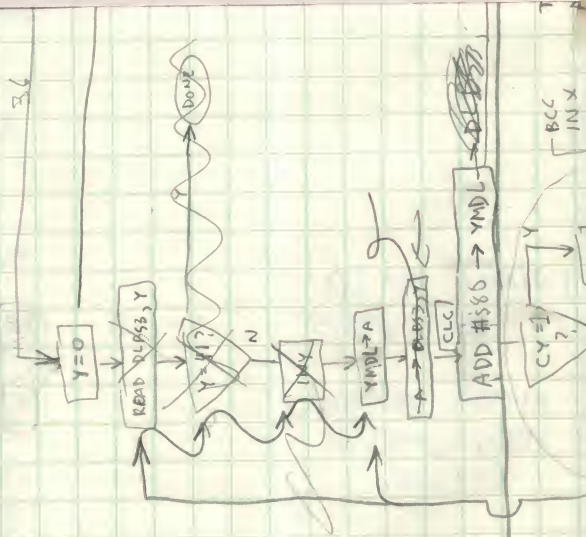


861
1.953

01 .THE 0,0,0
02 .OPT NOEJECT
0100 DLBASE=\$5000
0200 DLBS4=\$5004
0500 XM=\$600
0505 YM=\$601
0510 YMDL=\$603
1000 X=\$4000
1005 JSR YMXMDL
1006 JSR STUPDL
1010 BRK
8000 ;
8005 ;Coarse scroll
8006 ;XM,YM to display list
8007 ;
8010 YMXMDL
8015 LDA YM
8020 LSR A
8025 ORA #70
8030 TAX



YM	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
	7000	7086	7100	7180	7200	7280	7300	7386	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F00	7F80											





101
102
103

一寸山河一寸金

$R \rightarrow 0$

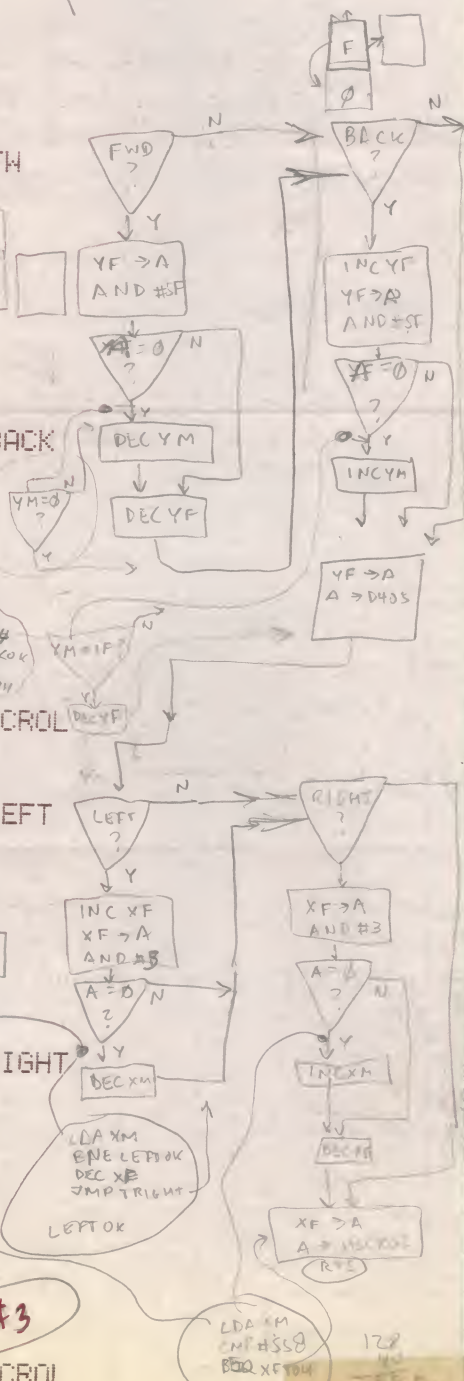
Fine scrolling



```

7500 ENDFR
      DEC FRAME
      BOTSER
      LDA $D40B
      CMP #112
      BCS BOTSER
      MIDSCR
      LDA $D40B
      CMP #112
      BCC MIDSCR
      RTS

```


$$\begin{array}{r} 128 \\ 140 \\ - 12 \\ \hline 258 \end{array}$$

0 1 2 3 4 5 6 7 8 9 A B

0 1 2 3 4 5 6 7 8 9 A B

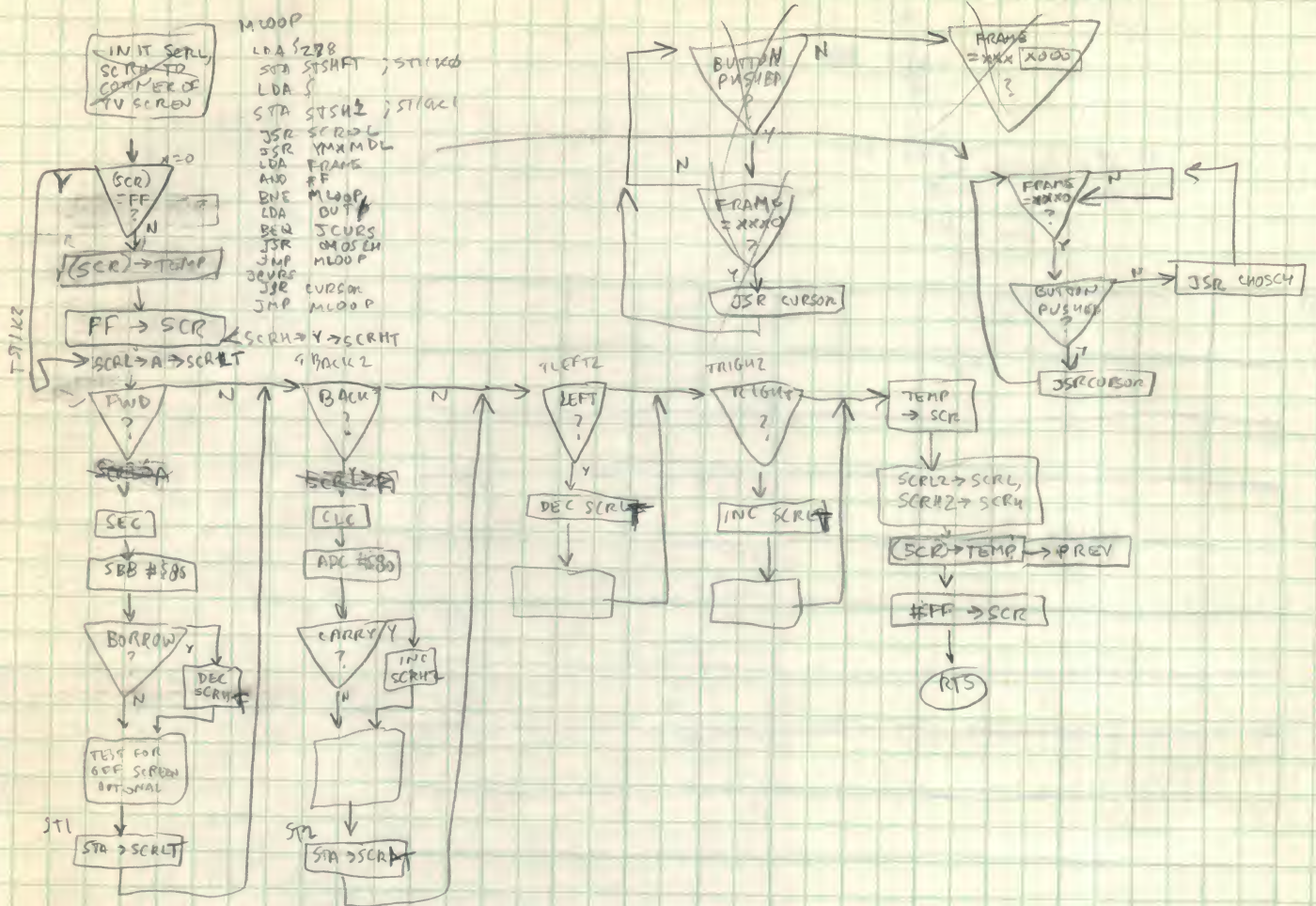
8:7
7:8

2 more
screens

$3 \times 3\frac{1}{2}$
 ≈ 10 screens

$2\frac{1}{2}$ wire
screen

SCROLLED SCREEN EDITOR



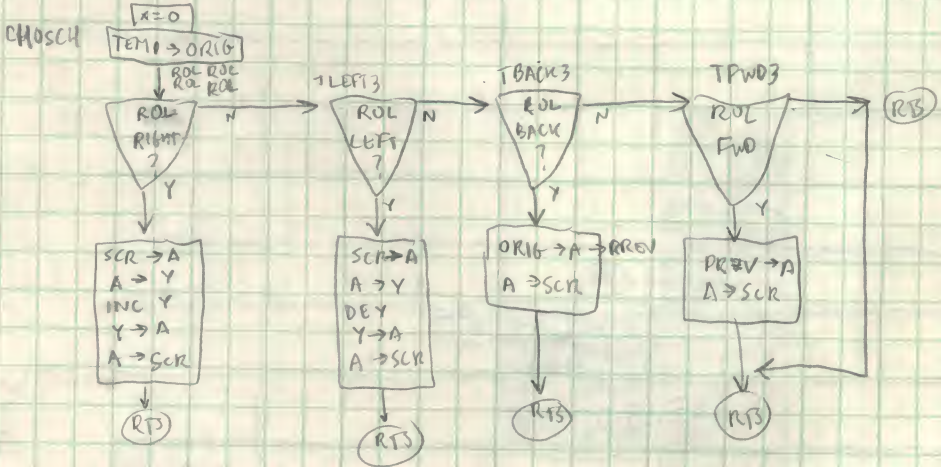
CHOSCH

STICK LEFT
DEC MEM CONTENTS
STICK IN TEMP

STICK RIGHT
INC MEM CONTENTS
STICK IN TEMP

STICK FWD
(PREV) -> MEM -> TEMP

STICK BACK
ORIG -> MEM




```

2000  %=$4400
2005  CURSOR
2010  LDW #0
2015  LDA (SCRL,X)
2020  CMP #FF
2025  BEQ TSTIK2
2030  STA TEMP
2035  LDA #FF
2040  STA (SCRL,X)
2045  TSTIK2
2046  LDY SCRH
2047  STY SCRHT
2050  LDA SCRL
2055  STA SCRLT
2060  ; Fwd bit
2065  ROR STSH1 ;FWD
2070  BCS TBACK2
2075  SEC
2080  SBC #80
2085  BCS ST1
2090  DEC SCRHT
2095  ST1
2100  STA SCRLT
2105  TBACK2
2106  ROR STSH1 ;BACK
2107  BCS TLEFT2
2110  LDA SCRL
2115  CLC
2120  ADC #80
2125  BCC ST2
2130  INC SCRHT
2132  ST2
2133  STA SCRLT
2135  TLEFT2
2140  ROR STSH1 ;LEFT
2145  BCS TRIGH2
2150  DEC SCRLT
2155  TRIGH2
2160  ROR STSH1 ;RIGHT
2165  BCS ENCUR
2170  INC SCRLT
2175  ENCUR
2180  LDA TEMP
2185  STA (SCRL,X)
2190  LDG SCRLT
2195  STG SCRLT
2200  LDA SCRHT
2205  STA SCRH
2210  LDA (SCRL,X)
2215  STA TEMP
2220  STA PREV
2225  LDA #FF
2230  STA (SCRL,X)
2235  RTS

```

```

2300  ;
2305  CHOSCH
2310  LDW #0
2315  LDA TEMP
2320  STA ORIG
2325  ROL STSH1
2330  ROL STSH1
2335  ROL STSH1
2340  ROL STSH1
2345  ;
2350  ROL STSH1
2355  BCS TLEFT3
2360  LDA (SCRL,X)
2365  TAY
2370  INY
2375  TYA
2380  STA (SCRL,X)
2383  RTS
2385  TLEFT3
2390  ROL STSH1
2395  BCS TBACK3
2400  LDA (SCRL,X)
2405  TAY
2410  DEY
2415  TYA
2420  STA (SCRL,X)
2425  RTS
2430  TBACK3
2435  ROL STSH1
2440  BCS TFWD3
2445  LDA ORIG
2450  STA (SCRL,X)
2455  RTS
2460  TFWD3
2465  ROL STSH1
2470  BCS CHRTS
2475  LDA PREV
2480  STA (SCRL,X)
2485  CHRTS
2490  RTS

```

```

2495  ;
1055  MLOOP
1057  JSR ENDFR
1060  LDA #278 ;STIK0
1065  STA STSHFT
1070  LDA #279 ;STIK1
1075  STA STSH1
1080  JSR SCROL
1085  JSR YMXNDL
1090  LDA FRAME
1095  AND #F
1100  BNE MLOOP
1105  LDA #0011 ;TRIG1
1110  SEC JCURS
1115  JSR CHOSCH
1130  JMP MLOOP
1135  JCURS
1140  JSR CURSOR
1145  JMP MLOOP

```

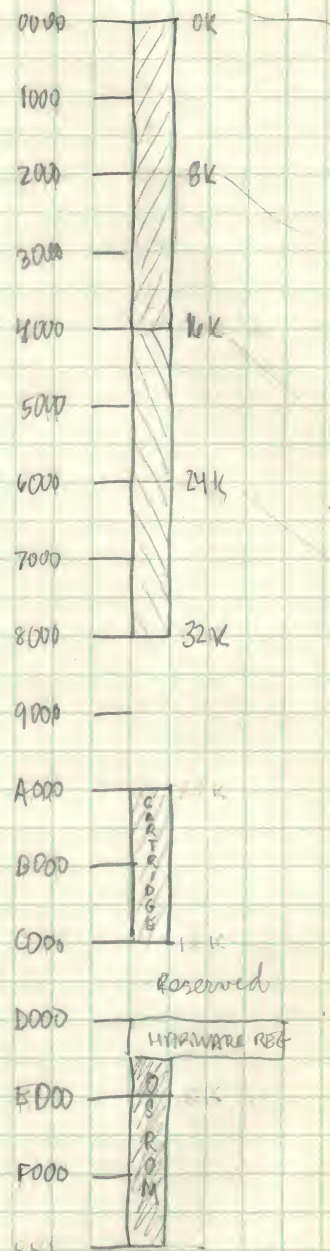
BNE
for
push button
to change
charac

```

0100  DLBASE=$5000
0200  DCBS4=$5004
0500  XM=$600
0505  YM=$601
0510  YMDL=$603
0515  XF=$604
0520  YF=$605
0525  STSHFT=$606
0530  FRAME=$607
0535  SCRLT=$608
0540  SCRHT=$609
0545  PREV=$60A
0550  TEMP=$60B
0555  STSH1=$60C
0560  ORIG=$60D
0800  SCRL=$B0
0805  SCRH=$B1
0899  ;

```

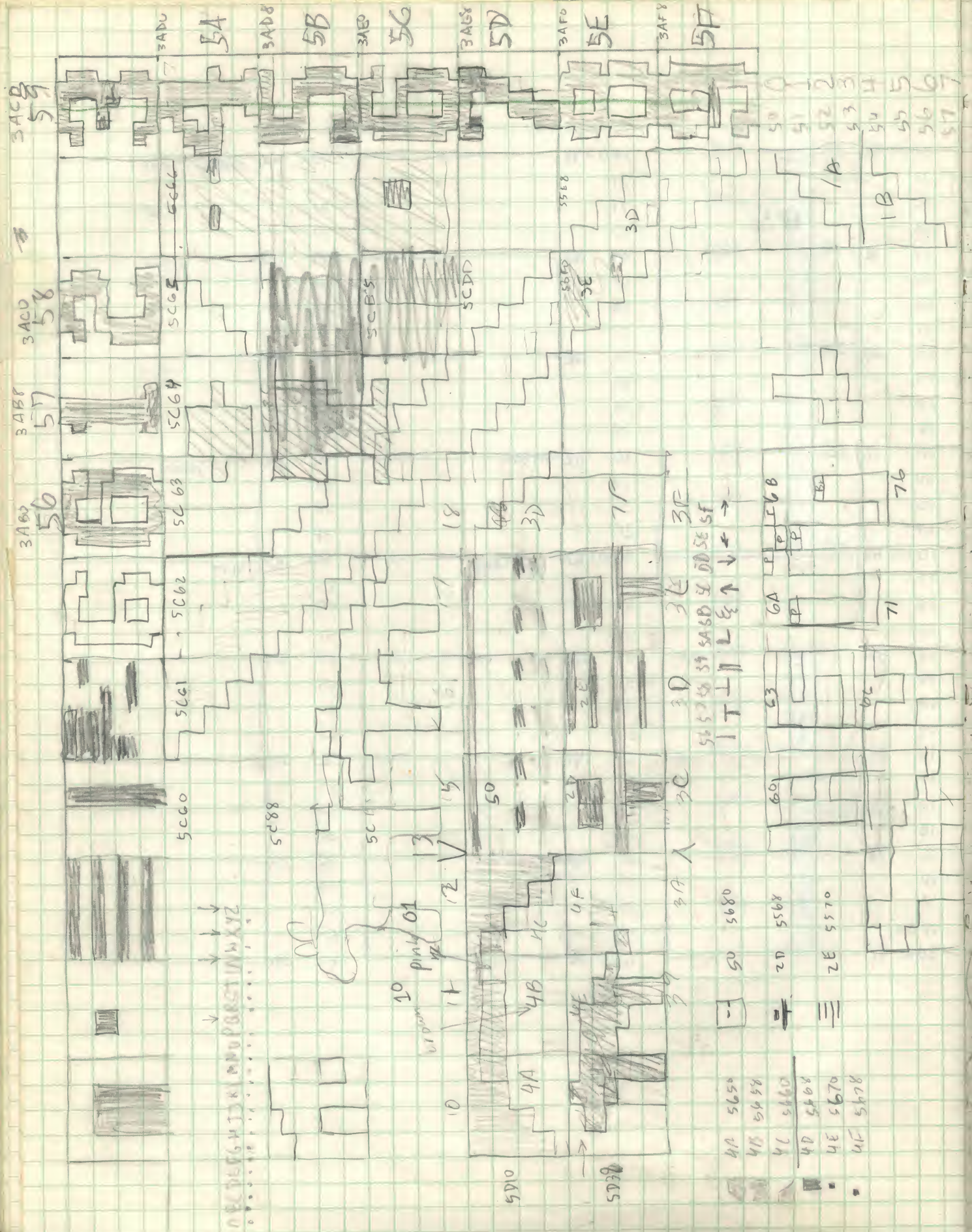
STA PREV



0000	0K
01	
02	
03	
04	1K
05	
06	
07	
08	2K
09	
0A	
0B	
0C	3K
0D	
0E	
0F	
10	00 4K
11	
12	
13	
14	5K
15	
16	
17	
18	6K
19	
1A	
1B	
1C	7K
1D	
1E	
1F	00
20	00 8K

20	00 8K
21	
22	
23	
24	9K
25	
26	
27	
28	10K
29	
2A	
2B	
2C	11K CHBASE
2D	
2E	
2F	
30	00 12K
31	
32	
33	
34	13K
35	
36	
37	
38	14K
39	
3A	
3B	
3C	15K
3D	
3E	
3F	
40	00 16K

4000	16K	SCROLL SETUP
41		
42		
43		
44	17K	CURSOR
45		
46		
47		
48	18K	SCROLL
49		
4A		
4B		
4C	19K	ENDER
4D		
4E		
4F		
5000	20K	DLIST
51		
52		
53		
54	21K	
55		
56		
57		
58	22K	
59		
5A		
5B		
5C	23K	
5D		
5E		
5F		
60	24K	



58 8
59 9
5A

53XX 33XX



FJKTG V X Z
GULD RUSM 1 2
SHOOT EM 1 2
SHOOT EM 2
YOU WIN
YOU LOSE
PINK WINS



58 8
59 9
5A

3AAB
55

4831

4832

4833

4834

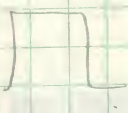
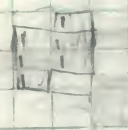
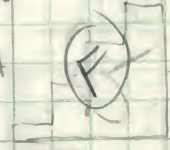
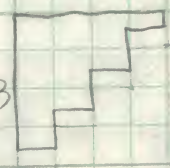
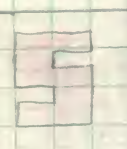
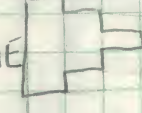
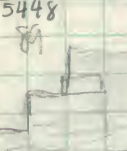
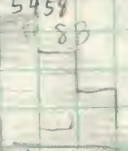
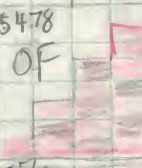
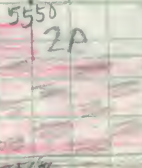
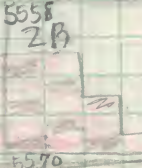
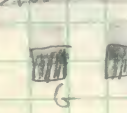
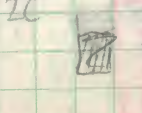
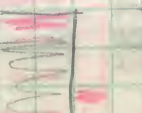
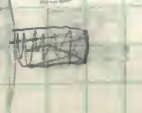
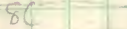
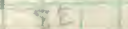
4835

4836

4837

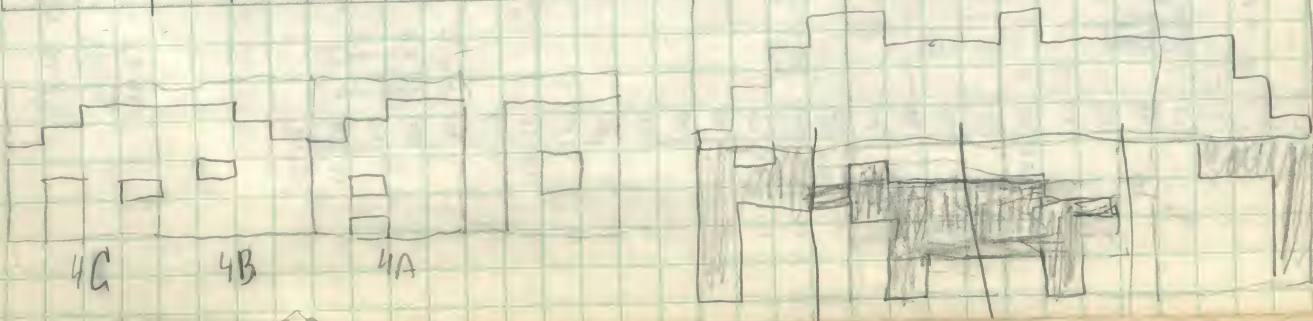
4838

4839

5400 5401 02 03 04 05 06 5407	5440 08 	5480 10 	54C0 18 	5500 20 	5540 28 
5408 09 0A 0B 0C 0D 0E 541F	09 	11 	19 	21 	29 
5410 02	5450 0A 	5490 12 	54D0 1A 	5510 2C 	5550 2A 
5418 03	5458 0B 	13 	1B 	23 	2B 
5420 04	5460 0C 	54A0 14 	54E0 1C 	5520 24 	5560 2C 
5428 05	0D 	15 	1D 	25 	2D 
5430 06	5470 0E 	54B0 16 	54F0 1E 	5530 26 	5570 2E 
5438 07	0F 	17 	1F 	27 	2F 
5448 08	5450 8A 	5458 8B 	5478 0F 	5550 2A 	5558 2B 
5460 09	5468 0G 	5470 0G 	5560 2C 	5568 2C 	5570 2E 
5480 0A	5488 0G 	5490 0G 	5498 0G 	5500 2C 	5508 2C 

56 → 52 → 32

5580 30	5580 38	5600 40	5640 48	5680 50	56C0 58
31	39	41	49	51	59
5590 32	55D0 3A	5610 42	5650 4A	5690 52	56D0 5A
33	3B	43	5658 4B	53	5B
55A0 34	55E0 3C	5620 44	5660 4C	56A0 54	56E0 5C
35	55E8 3D	45	5668 4D	55	5D
55B0 36	55F0 3E	5630 46	5670 4E	56B0 56	56F0 5E
37	3F	47	5678 4F	57	5F



$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

5930

42 D₈ B₁
for doing

[illegible]

MAIN 4000
44B7
5800
58EC
START 2C00
322C
5960
5A5B

4B7 } 574
BC }
62C } 728
FB }

BAL = BCC = BMG
BAGE = BCS = BMLE

speech
LM 3900N

heed disk 5-6-78-3-2

BUL 6200
65F8
POINTS 4D40
6000
60F8
CANROU 4E00
520B

5F8
F8
40B
24

SMO 4800
4AFD
CLAIM 5B00
5B9E
CREAT 4D11
SOUND 3900
398D
DLF ROUT ✓

2PD
9E
111
8D
13

ENDFR
INIT BOT
CLM JH PER

10
9F
5F

CHAR. LET 3AAB
3BFF
PB-M30BUYS
TTLES

157
7F
CB

NUG BIL 3AAB
3BFF
HOLY 3E80
3EAF
GUYS ✓

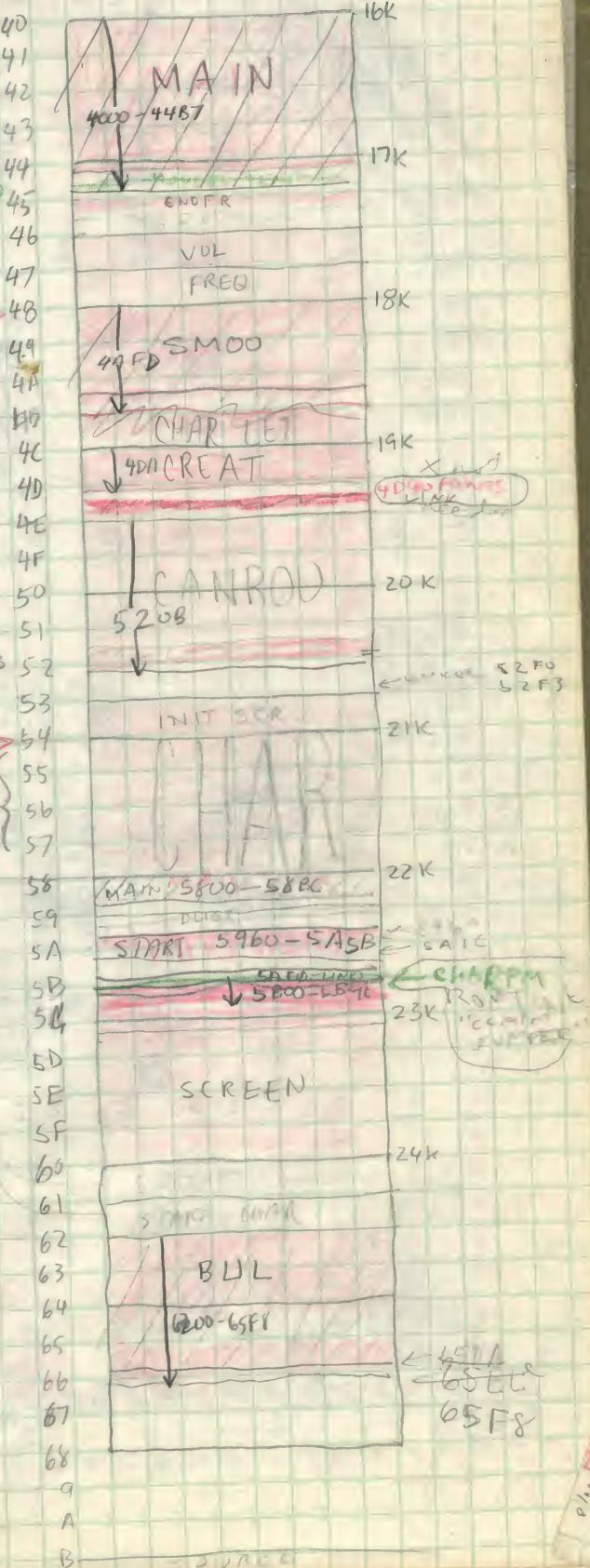
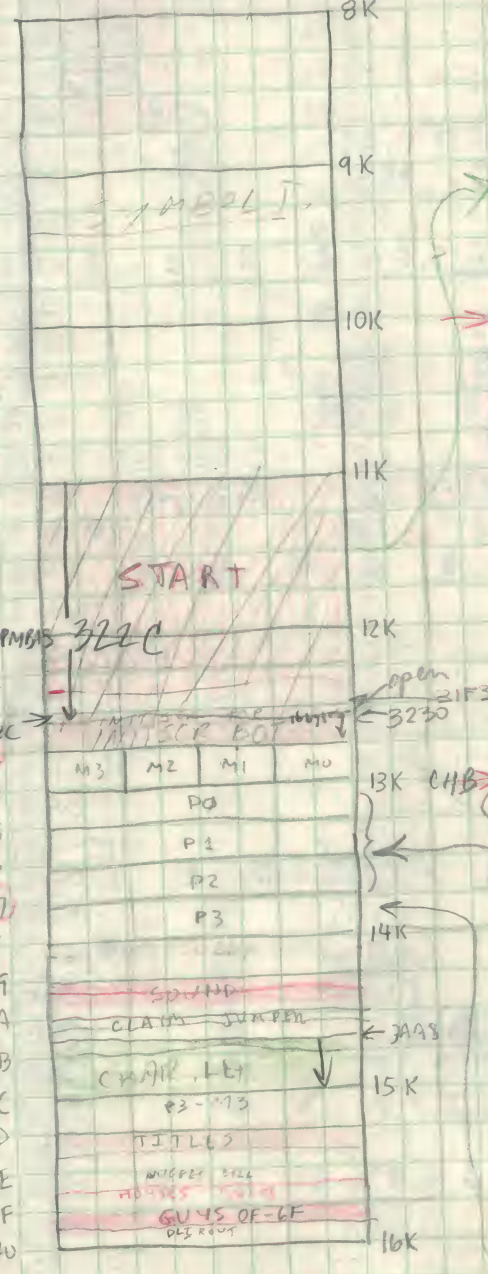
IF
21
7F

VOL ✓
FREQ ✓
INIT SCR
DLIST ✓

BE
BF
FF

CLAIM CAR
CACPOS, HSBAS

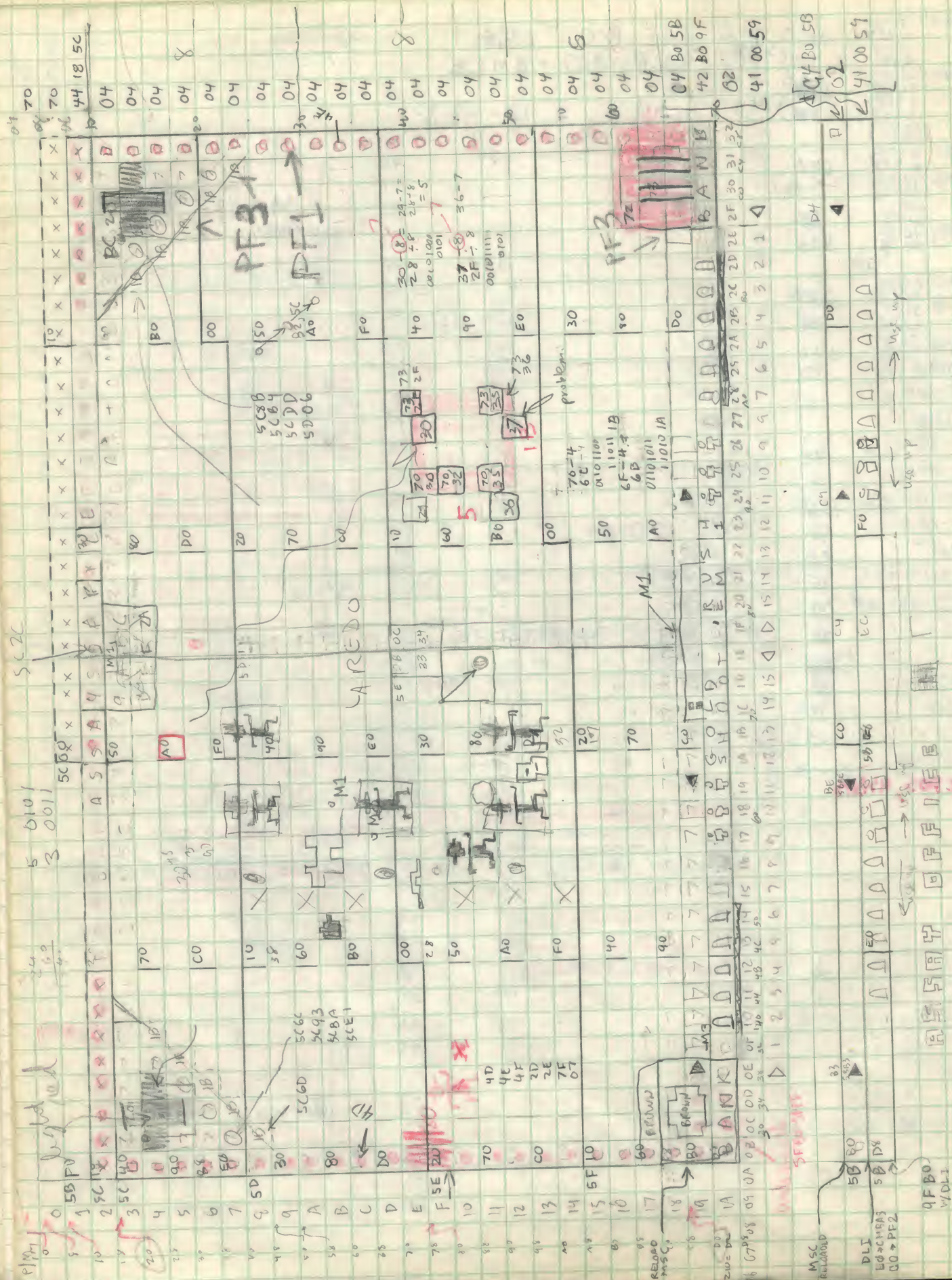
23
37
38



3628:70 7373 70
3650:70 7878 70

9090
9330
5100
9290

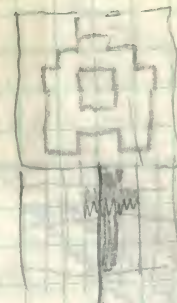
4800: AD OF 06 10 27 EA LA EP



BCC=BAL
BCS=BAGE

3F

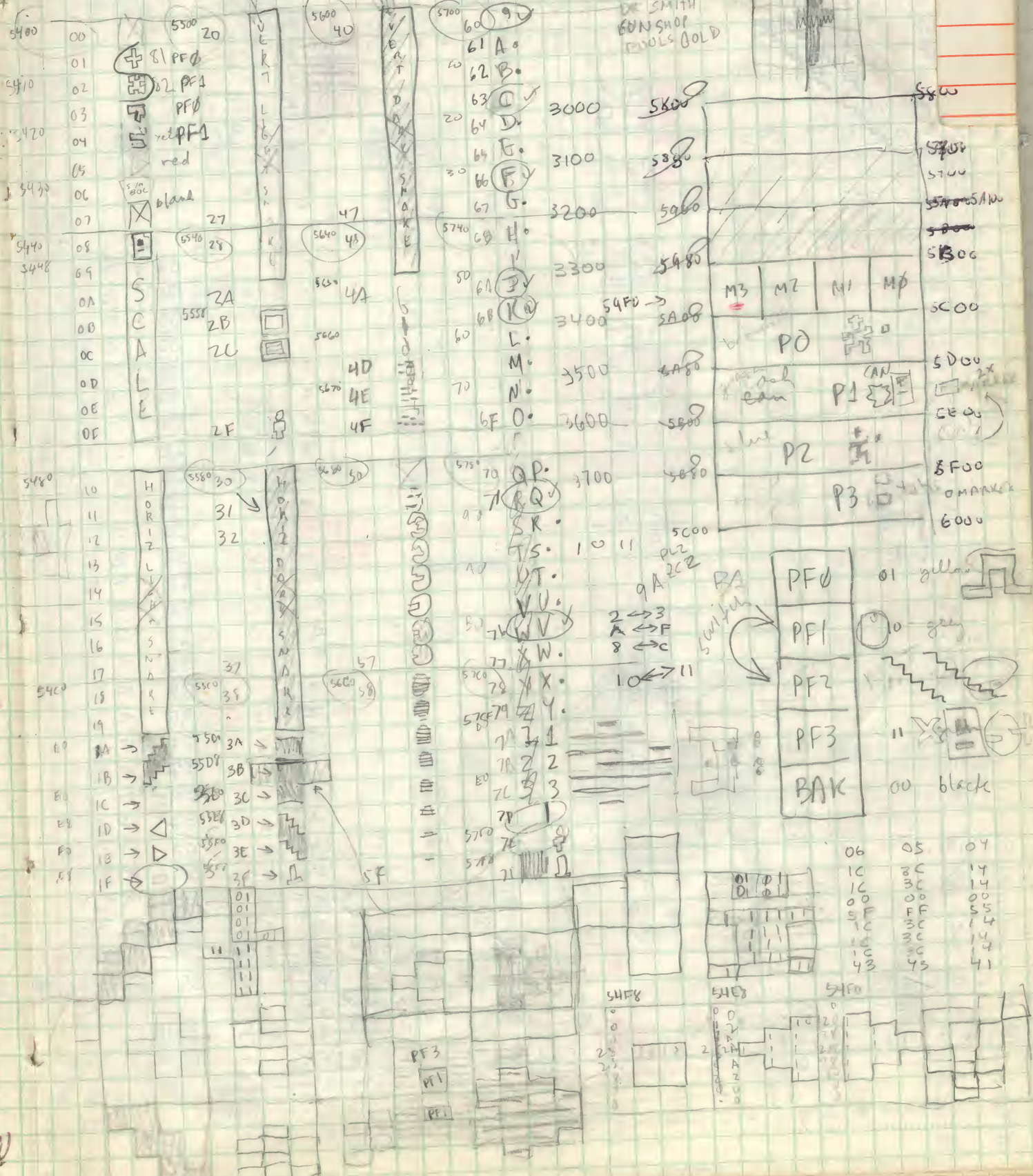
GOLD RUSH
EXTRA BULLETS
HEAD START
NORMAL GAME
PRESS START
ASSAY

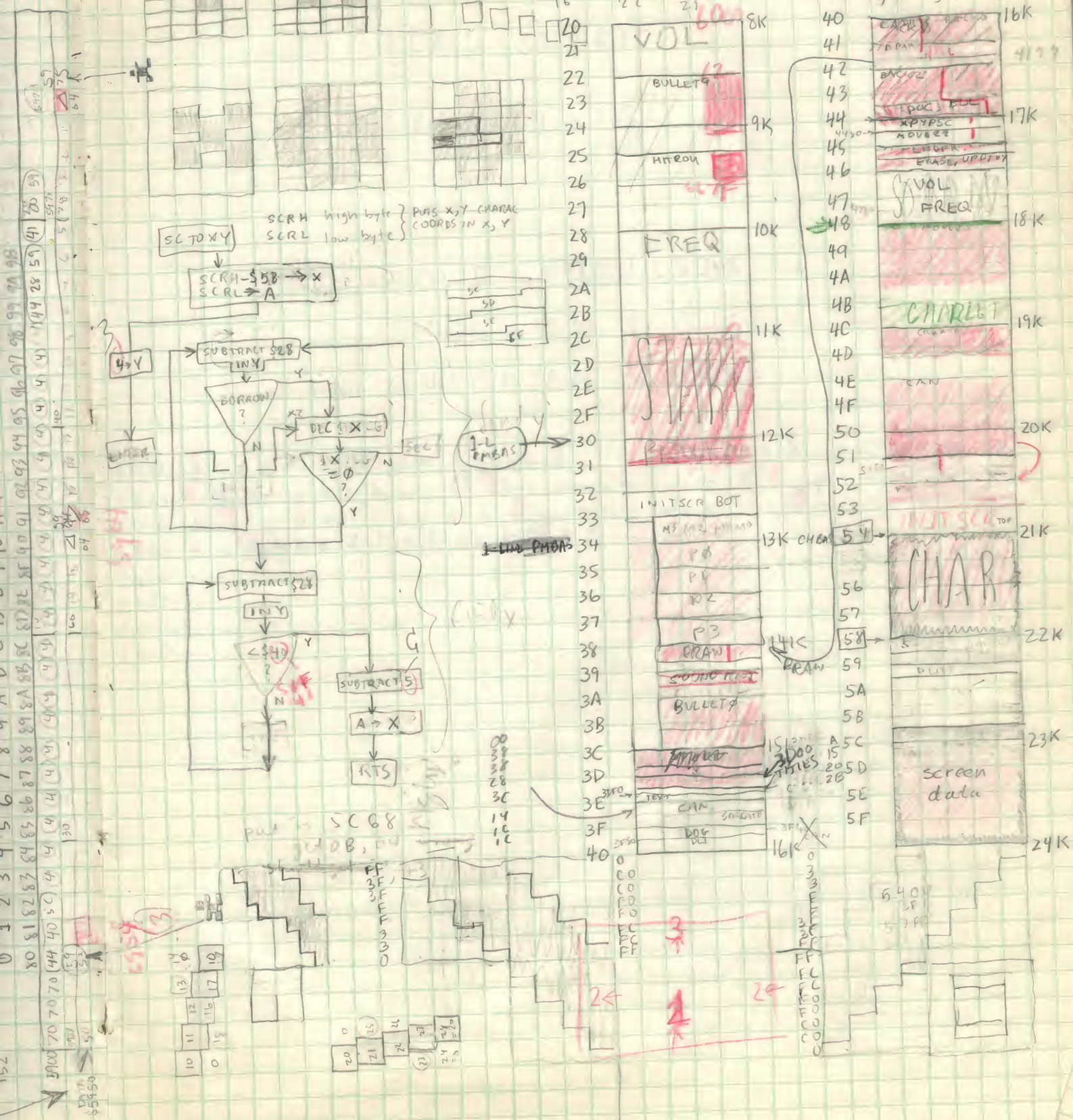


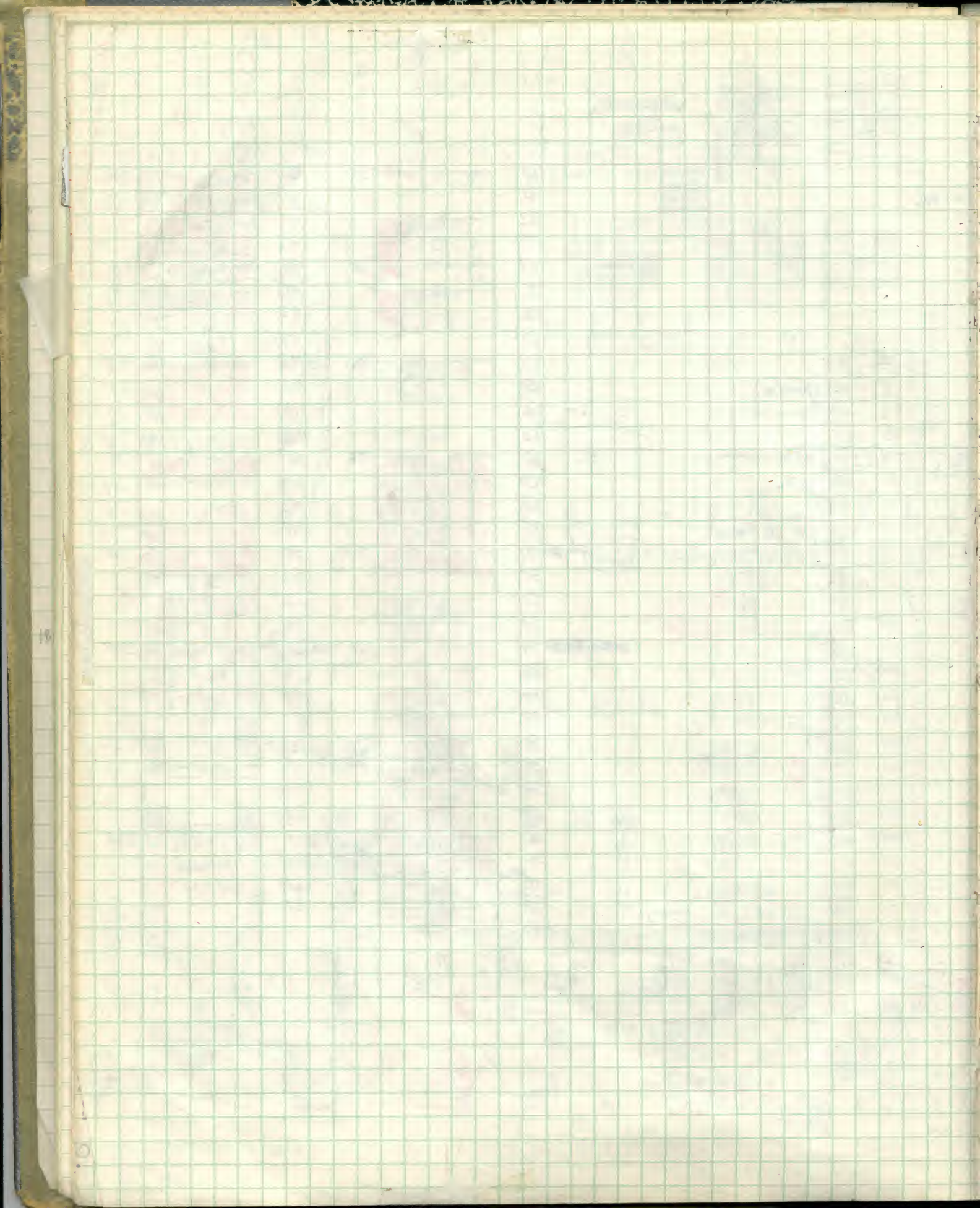
2
10

4977 = Randomness
4153

DR JONES
DR SMITH
GUN SHOP
BOWLS GOLD







☐ NORMAL GAME

☐ HEAD START

☐ 12 BULLETS

☐ GOLD RUSH \$0A

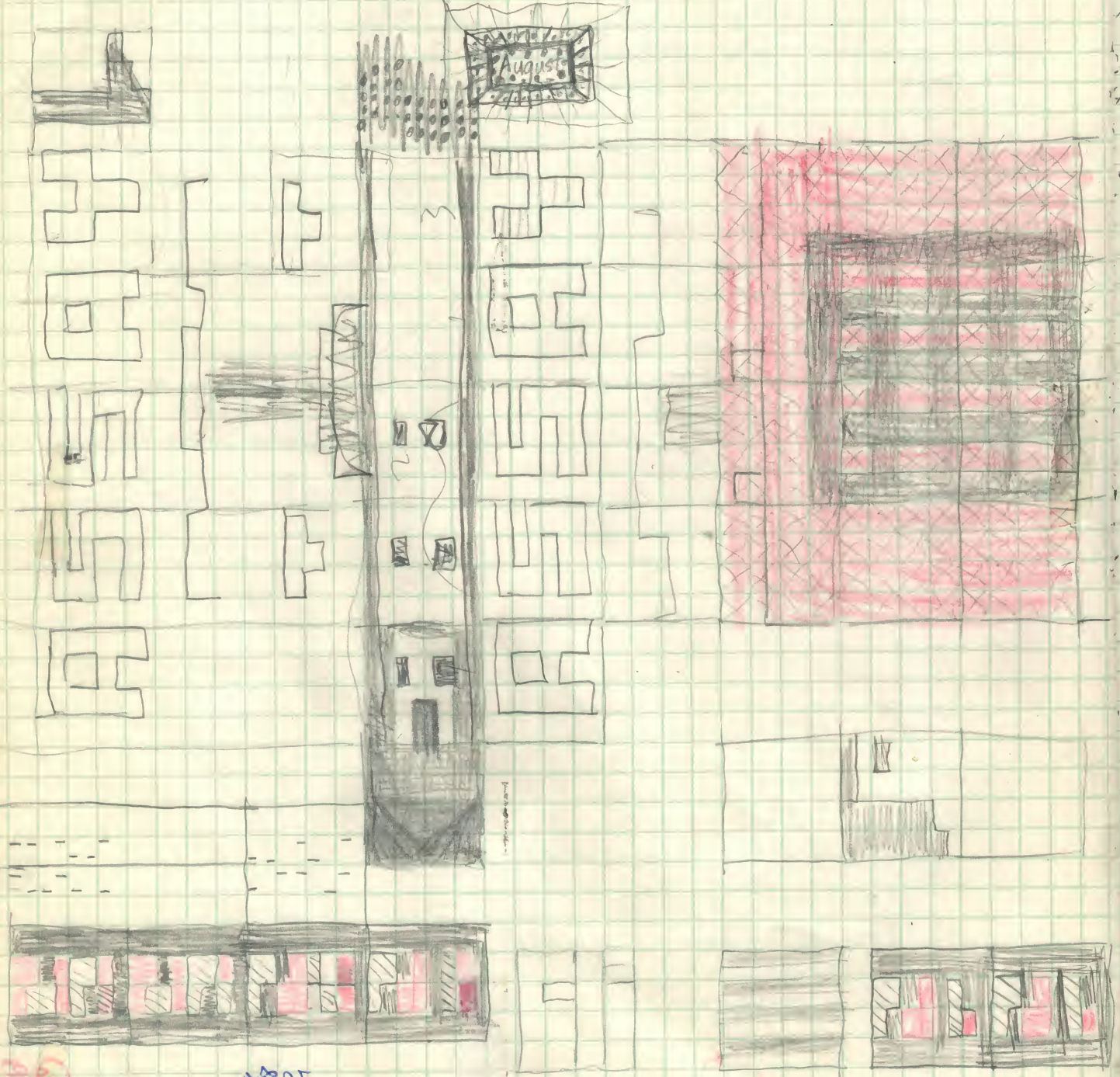
☐ SHOOT 'EM 1 \$15

☐ SHOOT 'EM 2 \$20

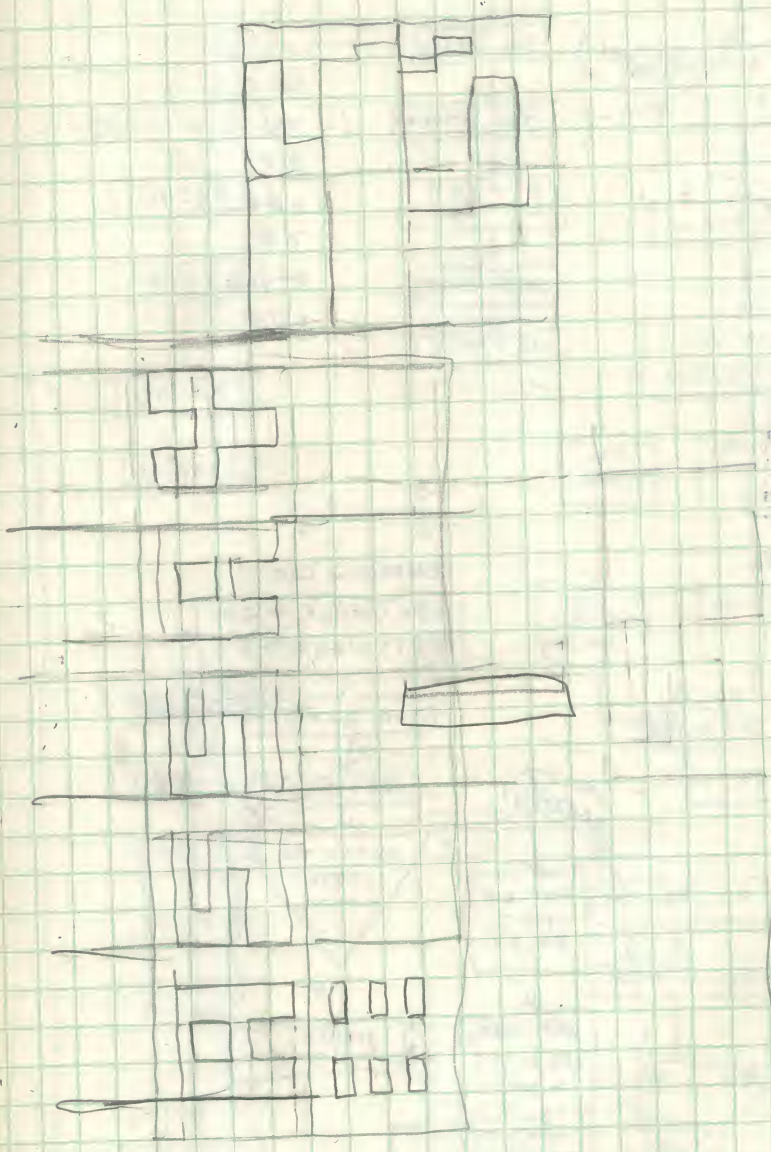
	press's	start	gold	rush	shoot	u'em	z
3D00	press's	start	gold	rush	shoot	u'em	z
3D20	shoot	vem	brown	wins	pink	win	
3D40	shoot	vem	brown	wins	pink	win	
3D60	shoot	vem	brown	wins	pink	win	
3D80	shoot	vem	brown	wins	pink	win	
3DA0	shoot	vem	brown	wins	pink	win	
3DC0	shoot	vem	brown	wins	pink	win	
3DE0	shoot	vem	brown	wins	pink	win	

11111111

August



88
105



USUAL STUFF

```

X = $4000
CREATE
JSR RAND
LDA #6
STA (SCRL, X)
JSR SCTOXY
STX ZPX
STY ZPY
BRK
    
```

```

RAND
LDX #0
XI LDA RANDOM
AND #$5F ; =
ORA #$5C
STA SCRH
LDA RANDOM
STA SCRL
LDA (SCRL, X)
CMP #50
BNE XI
RTS
    
```

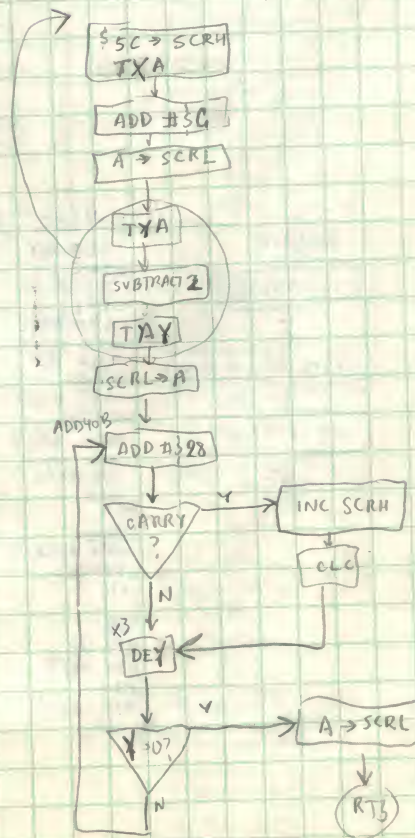
SCTOXY

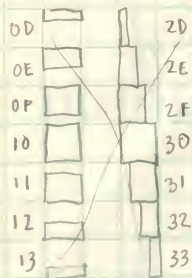
XY TO SC - CHANGES X, Y COORDS (IN X Y REGS) TO SCRH, SCRL BY DOING X, Y CONTENTS

XP YP
XP, YP COORDS TO SCRL, SCRH

```

XP → A
÷ 4
A → X
YP → A
÷ 4
A → Y
XY TO SC
    
```





422F

MOVEZ u_{000} XP, YP, ZX, ZY

GET XP, $\div 4$

SUBTRACT ZX

ABS X DIF

GET YP, $\div 4$

SUBTRACT ZY

ABS

COMPARE $YDIF \leq XDIF$

MOVE Y

GET YP, $\div 4$

SUBTRACT ZY

DIR

DECZY

INCZY

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

MOVEZ

XP $\rightarrow$ XP

YP $\rightarrow$ YP

ZX $\rightarrow$ ZX

ZY $\rightarrow$ ZY

JSR ERASEZ

JSR MOVEZ

ZX $\rightarrow$ ZPX

ZY $\rightarrow$ ZPY

SET ZTYPE

JSR DRAWZ

FRAME

STSH PT

XM

YM

TRQFLG

TRI FLG

ZX

ZY

ZPX

ZPY

ZTYPE

X DIF

generalize

INDEX = $\$0F$

ZPX + INDEX $\rightarrow$ ZX

ZPY + INDEX $\rightarrow$ ZY

JSR ZXYZSC

GET HYDRANT

READY TO CHOOSE NEXT DIRECTION?

GET DIR, UPDATE ZX, ZY

GET COLOR, XPO or YPZ $\rightarrow$ XP, YPO or YPZ $\rightarrow$ YP

ZDIRROUT.

UPDATE TWO SQUARES

DECZIN

DEL INDEX

MINUS ?

Y

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

ERASEZ

JSR ZXYZSC

$\phi \rightarrow A, \phi \rightarrow X$

STA (SCRL, X)

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

DRAWZ

JSR ZXYZSC

ZTYPE $\rightarrow A, \phi \rightarrow X$

STA (SCRL, X)

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

RTS

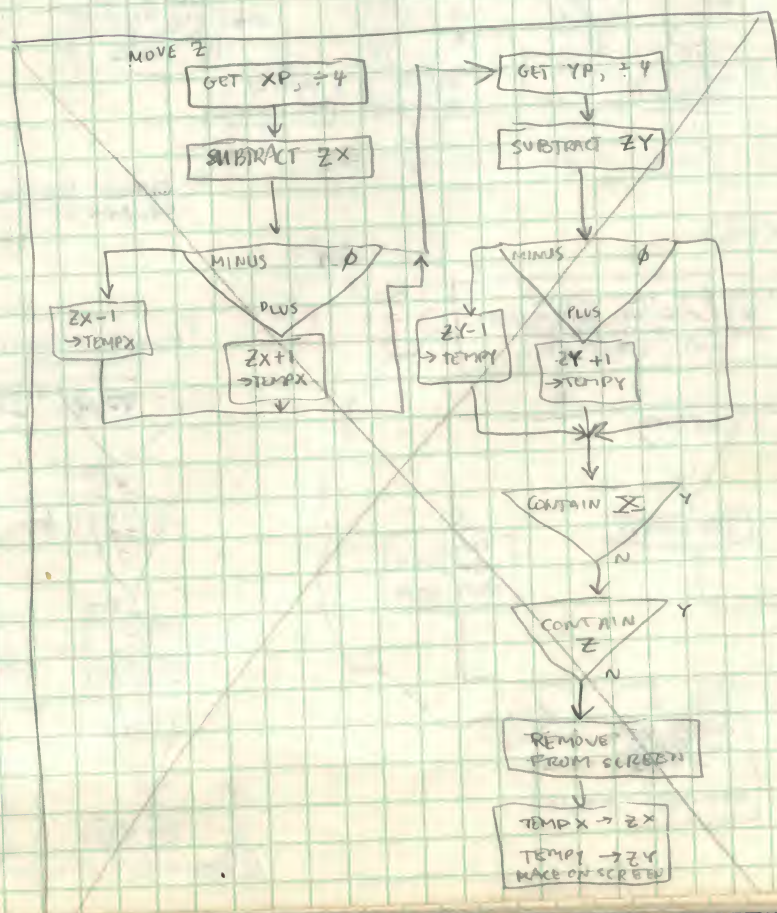
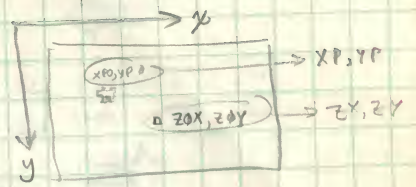
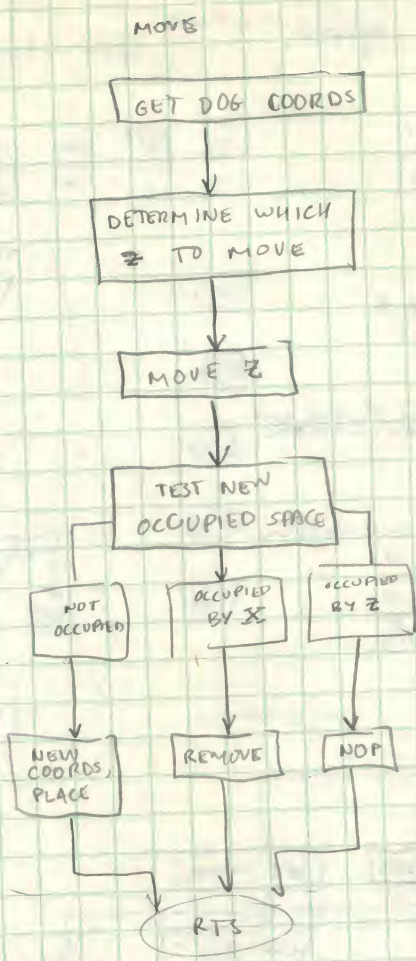
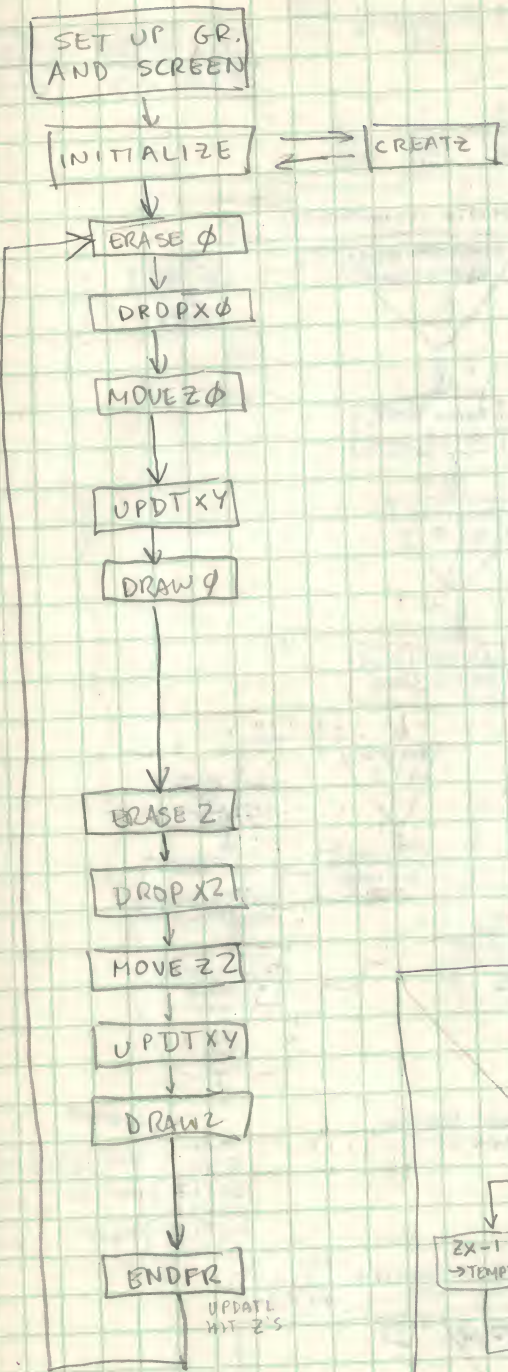
RTS

ZXYZSC
LDA ZY
SEC
SBCA $\$02$
TAY
LDA B $\$5C$
STA SCRL
LDA ZX
CLC
ADC B $\$0D$
ADD $40A$
ADC B $\$28$
BCC X4
JNC SCRL
CLC
X4
DEY
BNE ADD $40C$
STA SCRL
RTS
ABS X
BPL Q1
EDR $\$SPF$
CLC
ADC $\$01$
Q1 RTS

next page: SMOOTH!

Similar to MOVEZ, sets new dir flag

tests for occupied destination, no move if occupied



001
010

011
100

2/7/82
page: 3 hrs work

0101 0101 0010 0011

15

14
19

20
25

24
29

10 11 12 13 14
15 16 17 18 19

30 31 32 33 34
35 36 37 38 39

80

81
80 ← 00 X
01

20 25

81

01

21 26

22 27

23 28

24 29

DIRP direction of previous space
DIRF temp dir. for FINDSC
SCL2
SCL3
SCL4
SCL5
DIRN new square direction

14 0

19 10

0 15

40 45

41 46

42 47

43 48

44 49

45 50

46 51

47 52

48 53

49 54

50 55

51 56

52 57

53 58

54 59

55 60

56 61

57 62

58 63

59 64

60 65

61 66

62 67

63 68

64 69

65 70

66 71

67 72

SMOOTH (Summary)

TIME FOR NEW DIRECTION?

UPDATE

RTS

NEW DIRZ, SCL2, SCL3

OCCUPIED BY Z?

RTS

0 → SCL1, SCL1 (PREVIOUS SPACE)

OCCUPIED BY Z?

NEW BLANK SAME COLOR SCL2

NEW BLANK DIR COLOR SCL2

NEW BOX SAME COLOR SCL1

NEW COORDS → COORDS

RTS

SMOOTH

GET DIRP

POS

NEG

L.S. NYB = 9

L.S. NYB = 0

INC CONTENTS

DEC CONTENTS

JSR FNEWDIR MAKES DIRZ

4 → X JSR FINDSC

LDA (SCL1, X)

= Z

RTS

PLACE 0 IN PREV. POSITION: DIRZ → DIRN, DIRP → DIRF, Z → X, JSR FINDSC 0 → A, STA (SCL1, X)

Q5

= P1XXX

DEC NUMXX1

PLACE NEW BLANK, OPPOSITE COLOR

= P3XXX

DEC NUMXX3

PLACE NEW BLANK, SAME COLOR

INVERT MSB OF DIRN → DIRP

NEW COORDS → Z, ZY

see next page

Q → A, JSR FINDSC STA (SCL1, X)

FINDSC SUBROUTINE

USES 0, 2, 4 → X AND DIRF

DIRF

POS

LSB = 0

Y →

CLC ADC #1 (SCL1, X)

C = 1

INC SCL4

RTS

NEG

LSB = 0

Y ←

CLC ADC #4 (SCL1, X)

C = 0

DEC SCL4

RTS

FNEG

LSB = 0

Y ←

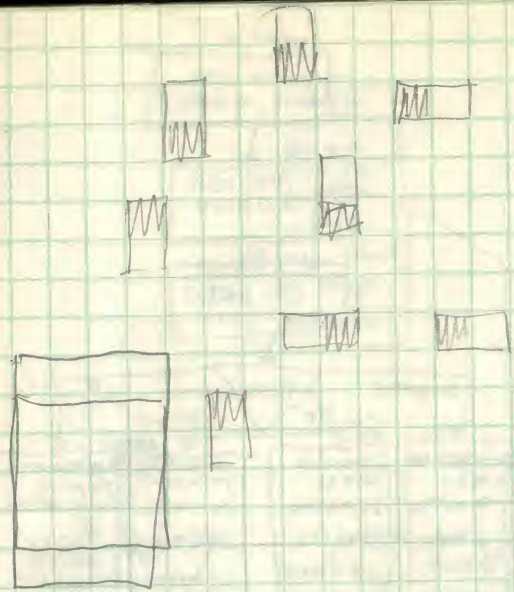
SEC SBC #4 (SCL1, X)

C = 0

SEC SBC #4 (SCL1, X)

RTS

SCL { present
SCL4 { previous x=0
SCL2 { previous x=2
SCL3 { next x=4
SCL4 { next x=4



6264 NEWDIR

JSR FNEWDR ; FIND NEW DIR, INTO DIRZF

TEST HERE

LDA DIRZF

STA DIRN

LDA DIRP

STA DIRZF

LDX #02

JSR FINDSC ; OLDSC

LDA #0

STA (SCRL,X) ; ERASE OLDZ

LDA DIRN

BMI Q6

BEQ Q7

LDX #0

LDA #510

STA (SCRL,X)

LDX #504

LDA #515

STA (SCRL,X)

INC ZY

LDA #81

STA DIRP

RTS

Q7

RTS

Q6

AND #501

BEQ Q8

RTS

Q8

RTS

Q7

RTS

Q6

AND #501

BEQ Q8

RTS

Q8

RTS

Q7

RTS

Q6

AND #501

BEQ Q8

RTS

Q8

RTS

Q7

RTS

occupancy

LDX #4
JSR FINDSC ; NEW SC

DOWN

conditional
add for
color change

conditional add

RIGHT

AND #501
BEQ Q8

Q7

Q6

Q8

Q7

Q6

Q8

Q7

Q6

Q8

Q7

Q6

Q8

Q7

Q6

Q8

X=0
X=2
X=4

NEWDIR

FIND NEW
DIR, → DIRZF

TEST FOR
OCCUPIED

0 → OLDSC
GET DIRN
→ NEWDIR

DOWN(01)

TEST Q6

UP(01)

LEFT(80)

RIGHT(81)

0 → OLDSC
10 → SC
15 → NEWSC
INC ZY
81 → DIRP

0 → OLDSC
20 → SC
25 → NEWSC
INC ZX
80 → DIRP

0 → OLDSC
29 → SC
34 → NEWSC
DEC ZX
00 → DIRP

0 → OLDSC
19 → SC
24 → NEWSC
DEC ZY
01 → DIRP

RTS

YELLOW 10-29

RED 30-49

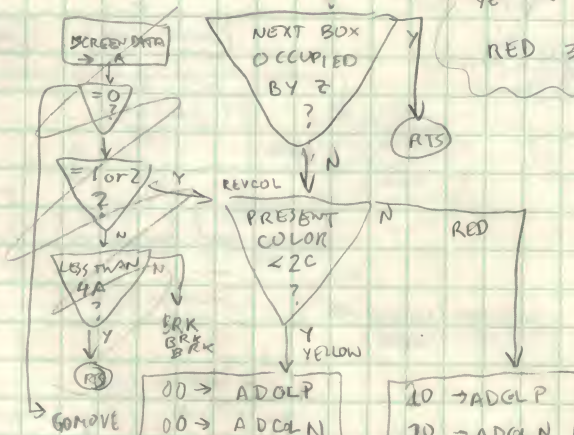
YELLOW < 29 < RED

ADCLP ADDCOLOR PRESENT BOX

ADCLN ADDCOLOR NEXT BOX

00 = YELLOW

10 = RED (add to charac)



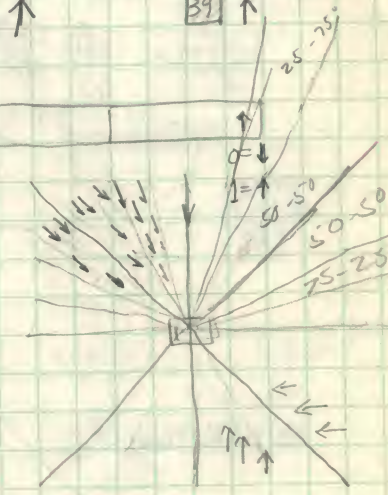
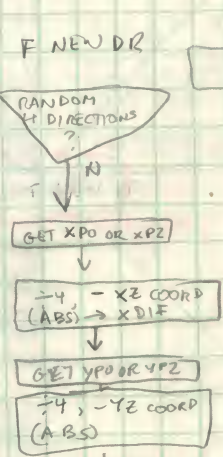
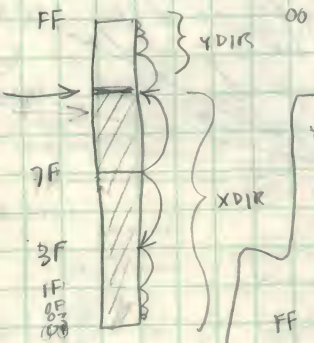
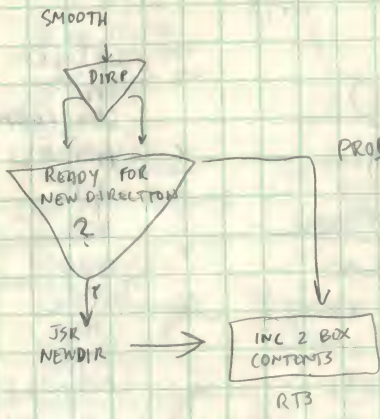
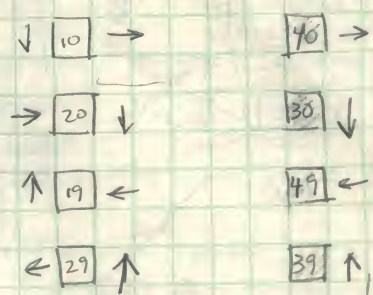
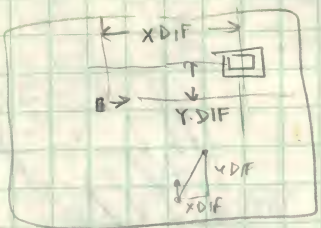
LDA (SCRL,X) ; X=4
CMP #509
BCS Q5
CMP #54A
BCS Q5
RTS ; OCCUPIED BY Z
Q5 CMP #52C
BCS RED
LDY #500
STY ADCLP
STY ADCLN
BCC Q12 ; UNCL
RED
LDY #520
STY ADCLP
STY ADCLN
Q12 CMP #500
BEQ GOMOVE
LDY #502
STY ADCLP
LDY #502
STY ADCLN
EOR #520
STA ADCLN
GOMOVE

NOT BLANK OR DIRP

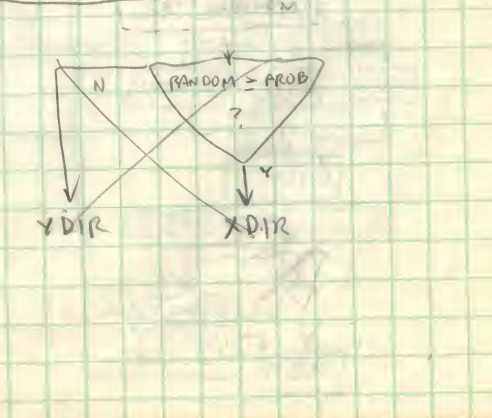
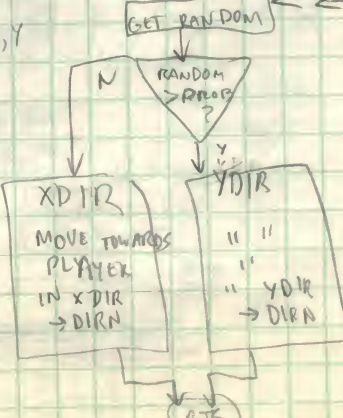
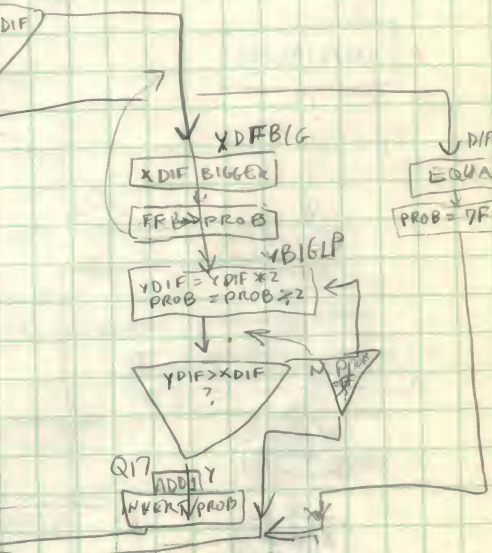
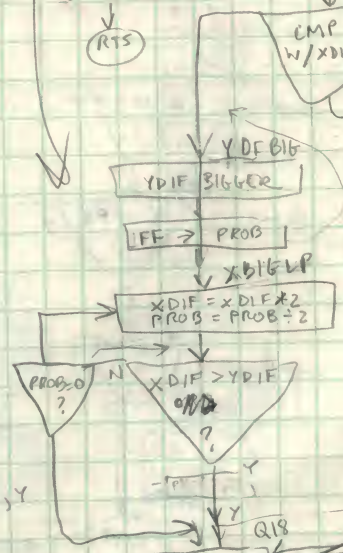
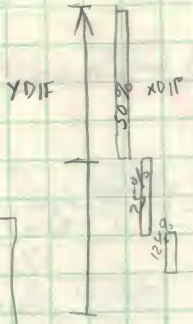
100 26

PLACE NEW BLANK

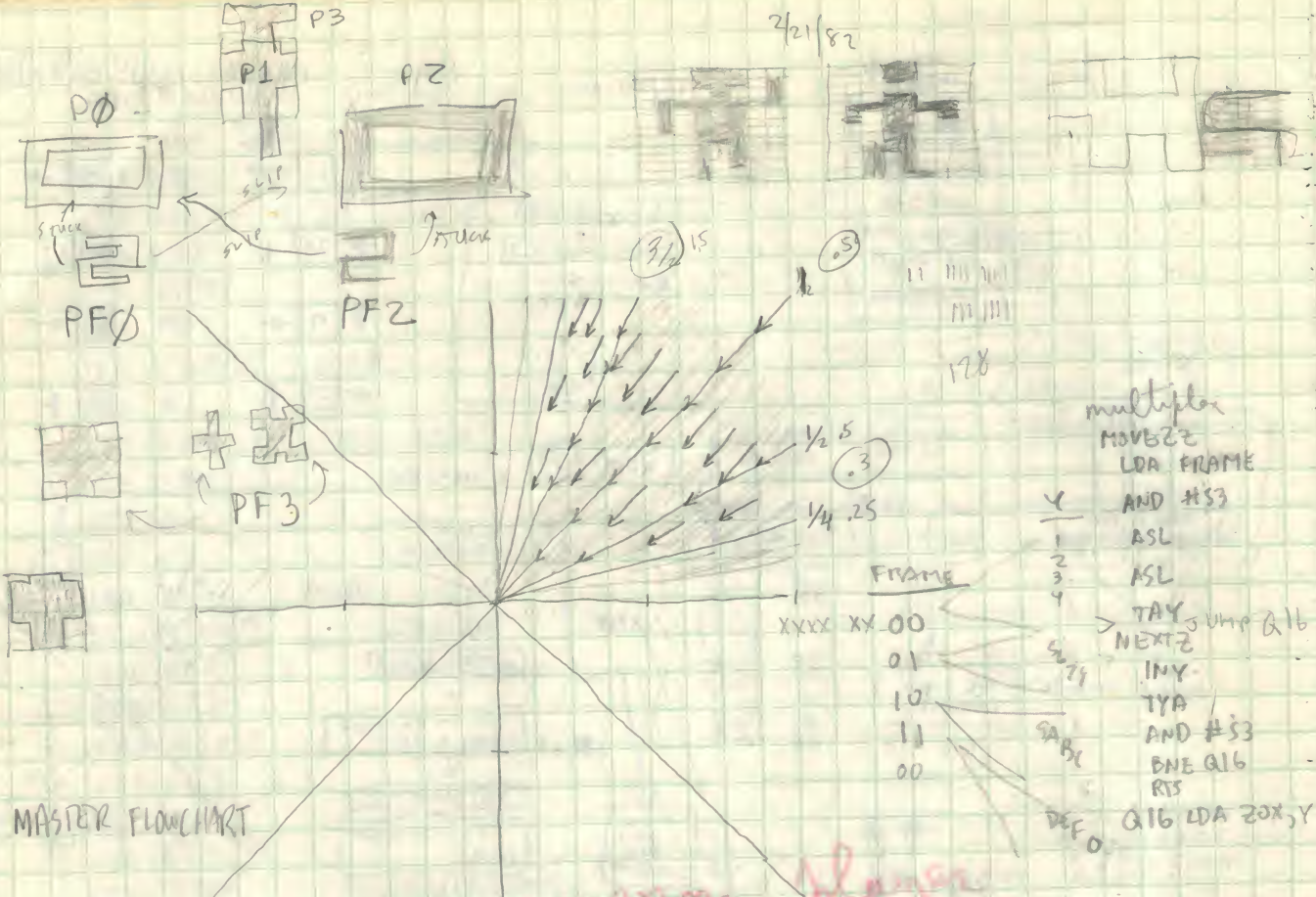
MOVE 20
RPO → AP
YPO → YP
ZOX → ZX
ZOY → ZY
(DIRPO → DIRP)
JSR SMOOTH
PARAMETERS ←



NEWDIR
SAVEZ = \$6
MOVEZ2 ; MOVE 16 Z5
LDY # \$10
NEXTZ
BCL Y DIR
BPL Q16
RTS
Q16 LDA ZOX, Y
BEQ NEXTZ
STA ZX
LDA ZOY, Y
STA ZY
STY SAVEZ
JSR SMOOTH
LDY SAVEZ
LDA ZX
STA ZOX, Y
LDA ZY
STA ZOY, Y
JMP NEXTZ

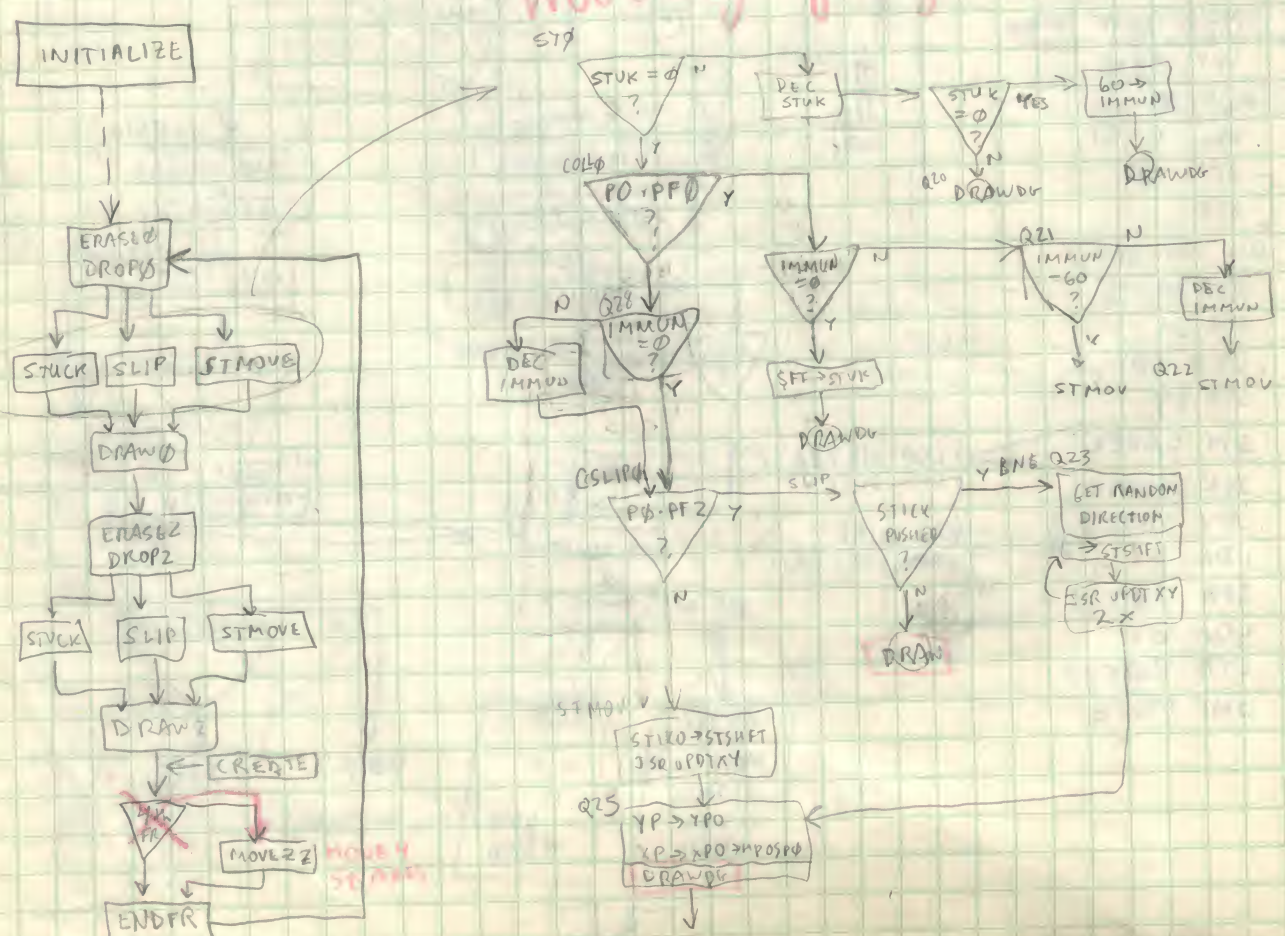


2/21/82



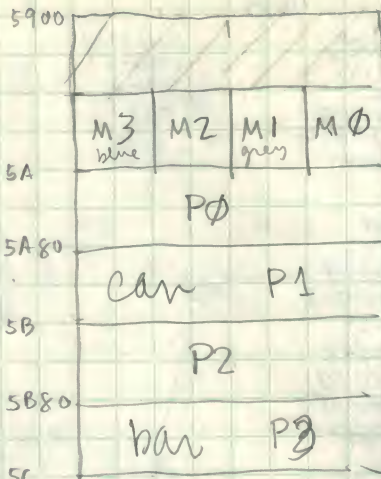
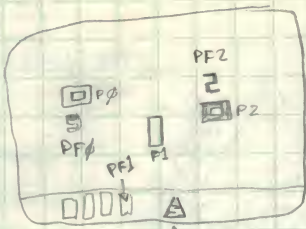
MASTER FLOWCHART

multiplier
 MOV B2Z
 LDA FRAME
 AND #53
 ASL
 ASL
 TAY
 NEXTZ
 INY
 TYA
 AND #53
 BNE Q16
 RTS
 Q16 LDA 20X,Y



RANCAN NOP 4F54
 LDA RANDOM
 AND #5F ; 5C40 TO ; 5EE7
 OR #5C
 CMP #5F
 BEQ RANCAN
 STA SCR4
 LDA RANDOM
 STA SCR4
 CMP #5E8
 BCS TESLW2
 RTS
 TESLW2
 LDA SCR4
 CMP #55E
 BEQ RANCAN
 RTS

ABSV



yellow

grey

red

blue

DRAWCN NOP 4F53
 LDY #6
 NYLNI
 LDA CAN,Y
 STA (Y),Y
 DEY
 BNE NYLNI
 RTS

SETCAN NOP 4F5F
~~LDY #5A~~
 LDA #5A
 STA YBAS
 LDA XPI
 STA XPI
 LDA YPI
 ORA #580
 STA YP

ABSV
 BPL QPL
 EOR #5FF
 CLC
 ADC #1
 QPL RTS

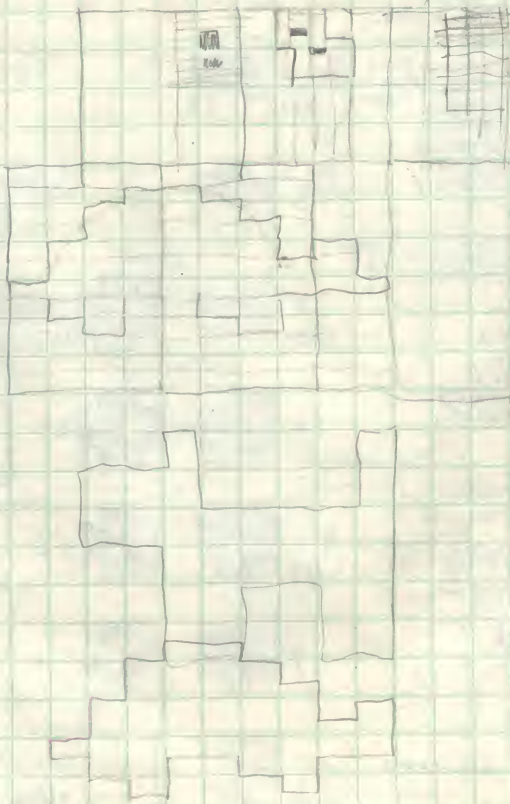
CPLMT #0

LDA XPI
 SEC
 SBC XPI
 BMI SETXPL
 SETXMI
 JSR ABSVAL
 STA CXMI
 LDA #0
 STA CXPL
 JMP YPLMT
 SETXPL
 LDA YPI
 SEC
 SBC YPI
 BMI SETYPL
 SETYMI
 JSR ABSVAL
 STA CYMI
 LDA #0
 STA CYPL
 RTS
 SETYPL
 STA CYPL
 LDA #0
 STA CYMI
 RTS

BARLEF
 LDA SCOR0
~~LDY #50F~~
 ASL A
 ASL A
 CLC
 ADC #550 40
 STA HP0SP3
 RTS

BARIT
 LDA SCOR2
 EOR #5FF
 AND #50F
 ASLA
 ASLA
 CLC
 ADC #580 78
 STA HP0SP3
 RTS

BAROFF
 LDY #50
 STY HP0SP3
 RTS



STATCN 670 — O1 PREBAT, UNMNNED
 — P0 - OWNED - 80
 CANCRI 671 — P2 - OWNED - 81
 — PROB CREATE CAL

SCORE 672 — EXPL 674
 — C XMS 675
 SCORE2 673 — CYDL 676
 — CYMS 677

ENDFR

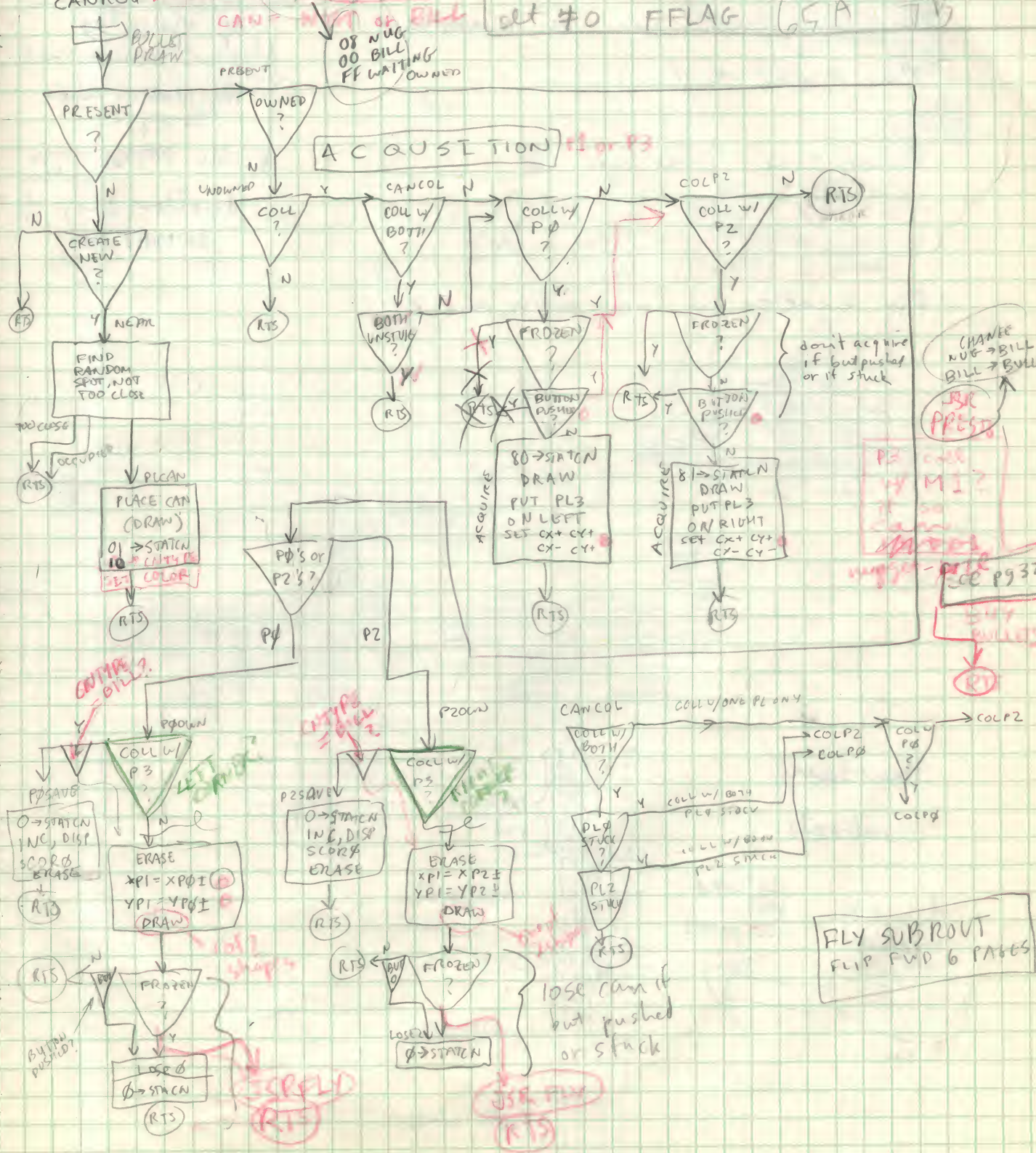
VCNMAX 658

VCNSM 654

FFLAG LGA

CNTYPE 674 10- mixed
-bal

set #0



Forbid car on left/right edge ✓

count bails >3 slow
>16 not allowed

~~BA100~~
~~BA100~~
NUMBT
NUMBT2

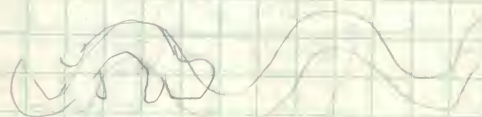
Slow when carrying car or NUMBT > 3 ✓

X min X max etc ✓

frozen time - place man 

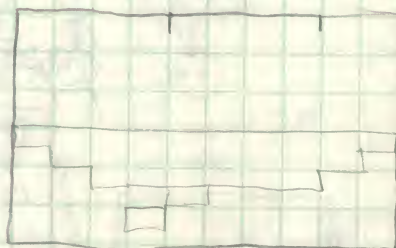
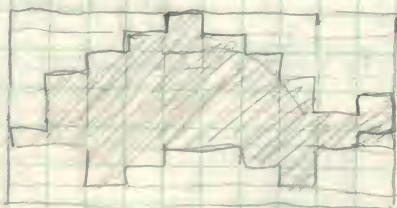
+ = red

x = white



alternate frames
if NUMBT > 3
or if car owned
inc NUMBT when
dropped
dec NUMBT when
eaten
disallow drop if
NUMBT > 15

fix barrier bug move scovy modale



1111 1001

0000 0001



dinosaur
rowboats

chance of
color of
new
snake

LDA SCORE
SEC
SBC SCORE2



15 11 11
14 11 11
13 11 01
12 11 00
11 10 10
10 10 10
9 10 10
8 10 10
7 10 10
6 10 10
5 10 10
4 10 10
3 10 10
2 10 10
1 10 10
0 10 10

100%
red

mostly red

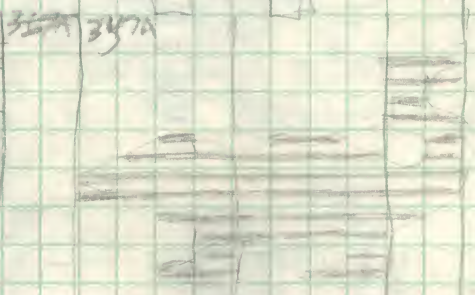
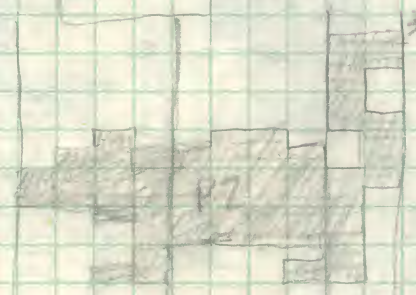
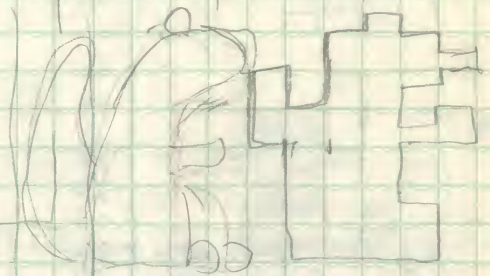
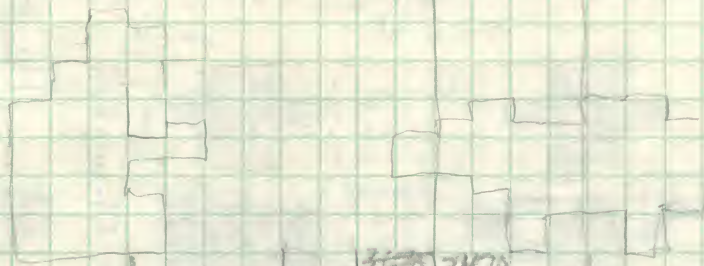
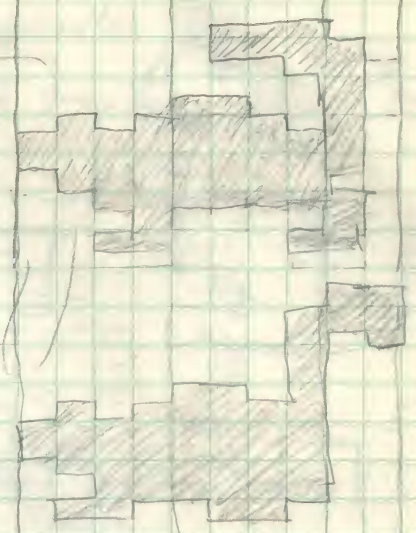
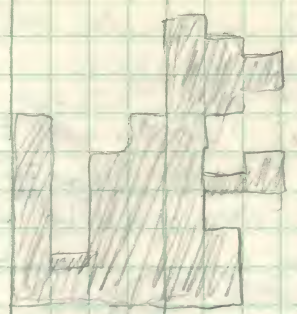
mixed

50-50

mostly white

100%
white

15 11 11
14 11 11
13 11 01
12 11 00
11 10 10
10 10 10
9 10 10
8 10 10
7 10 10
6 10 10
5 10 10
4 10 10
3 10 10
2 10 10
1 10 10
0 10 10

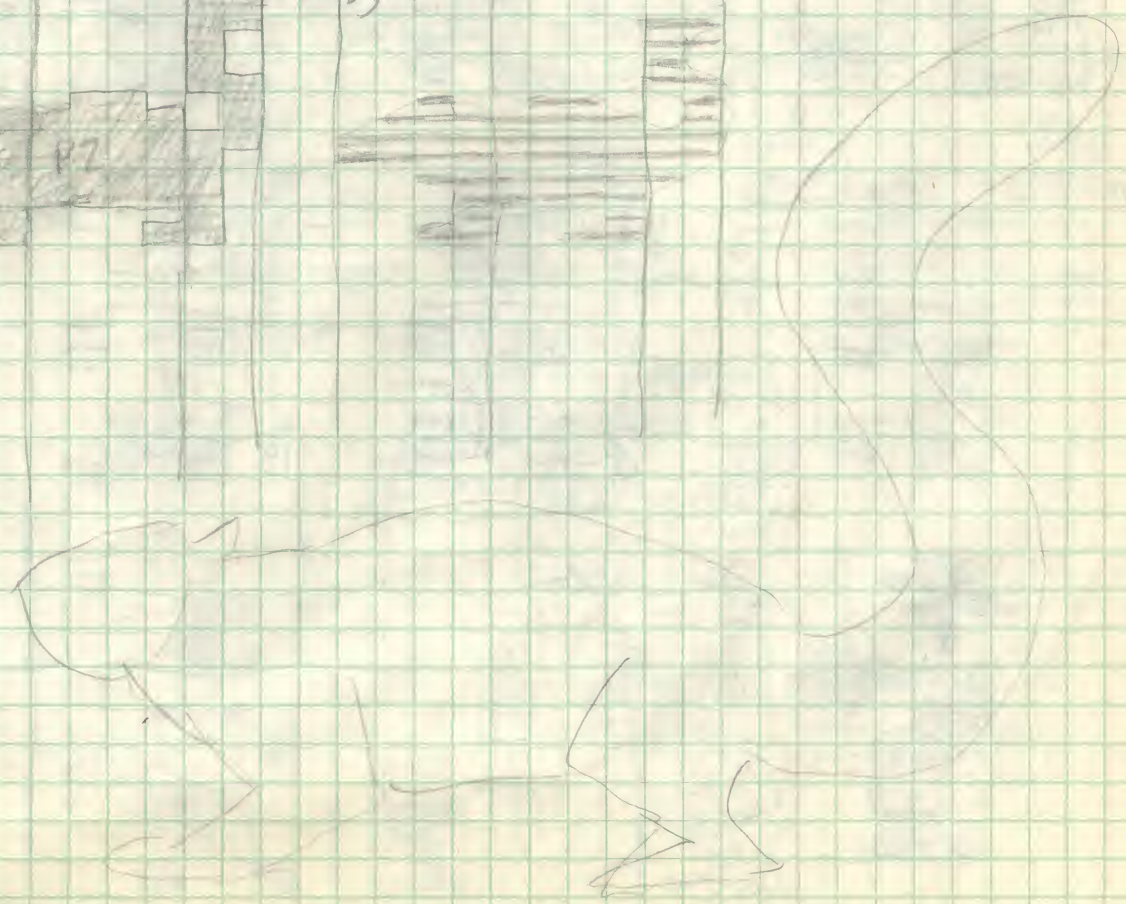


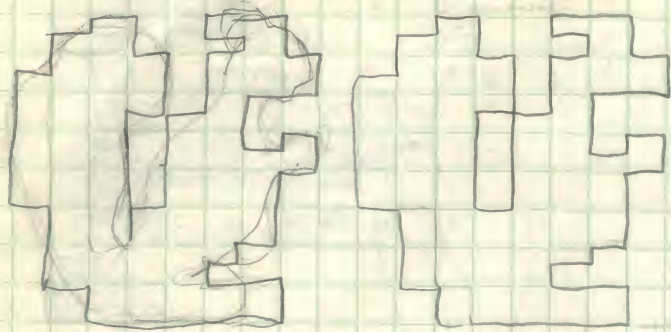
3675

347A 347B

3013

11 M2 3300
0006 3400
LAD

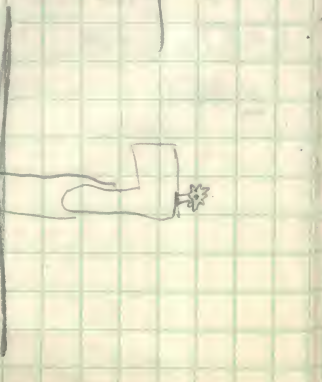
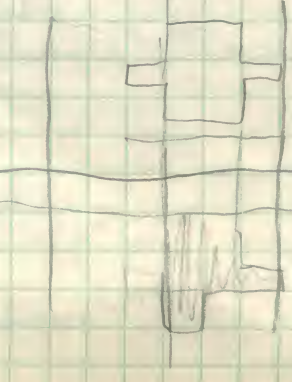
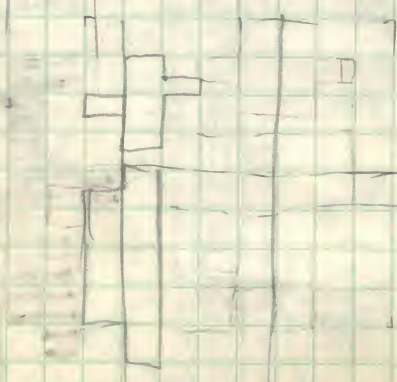
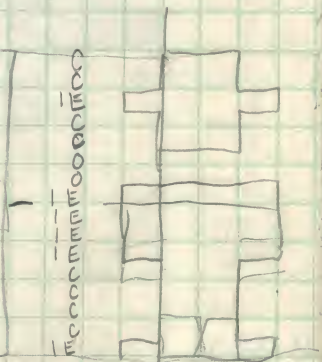
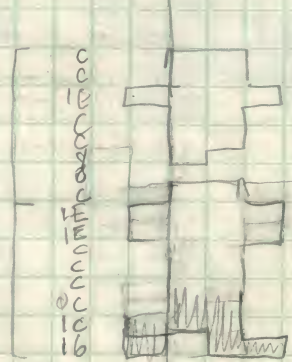
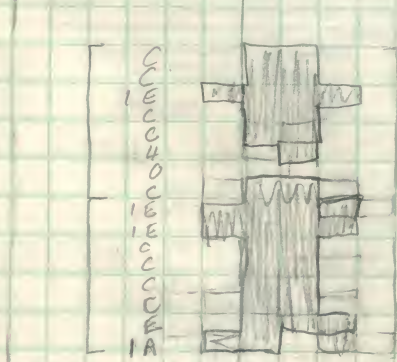
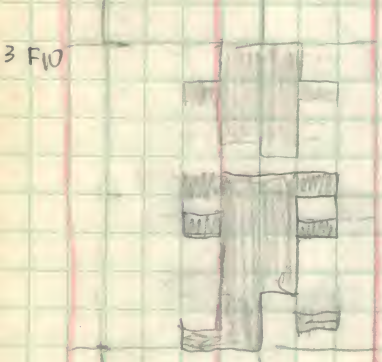
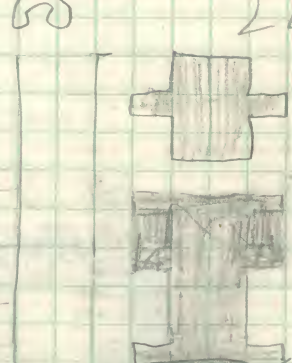
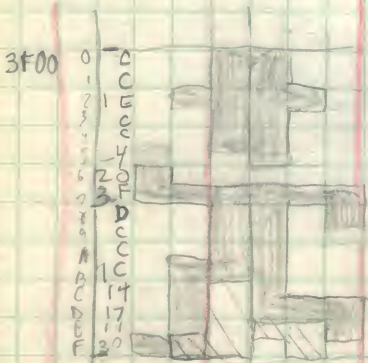
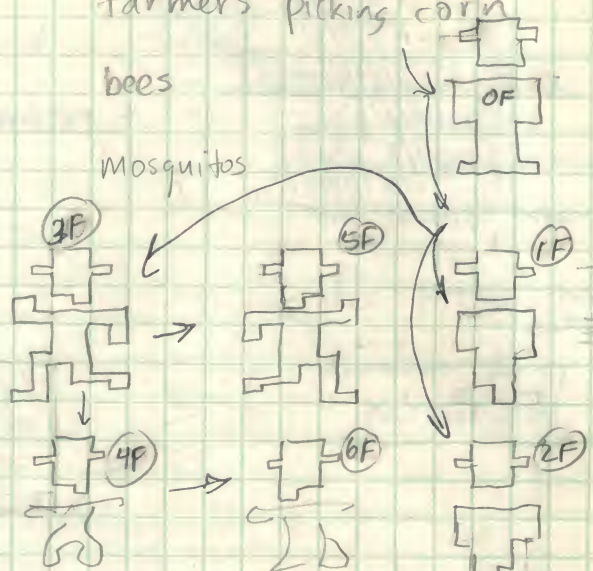




farmers picking corn

bees

Mosquitos



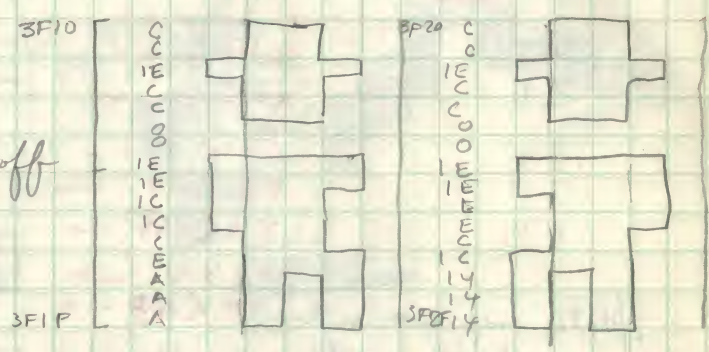
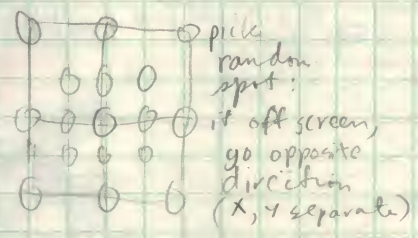
4/8/82 slow down farmers 50%
diag slowdown

limit X's to 5
change color X's

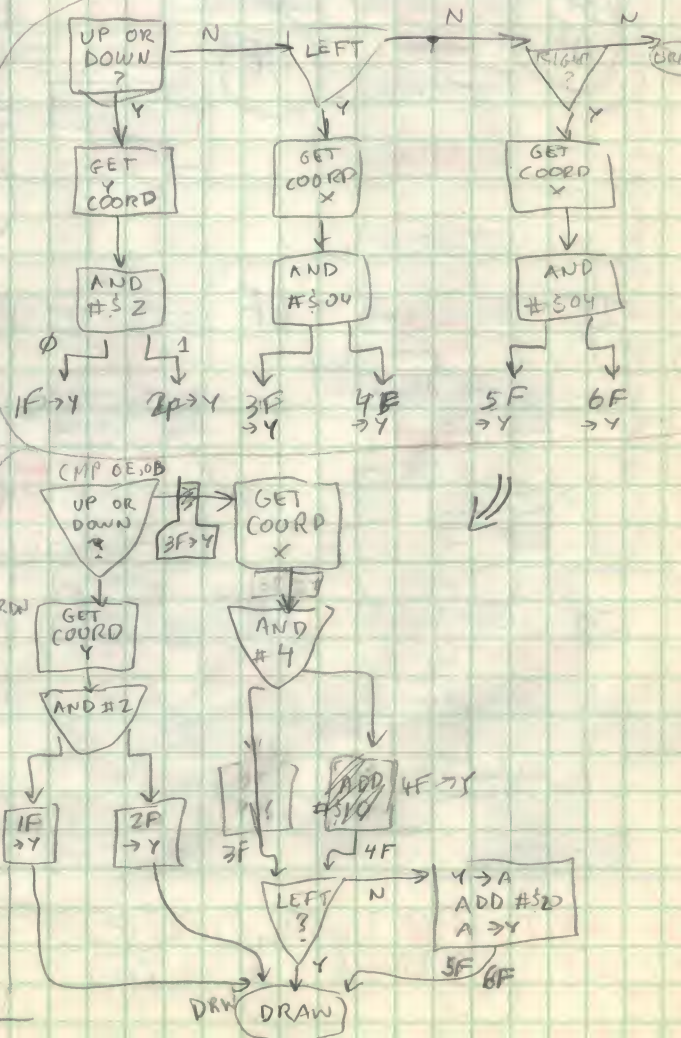
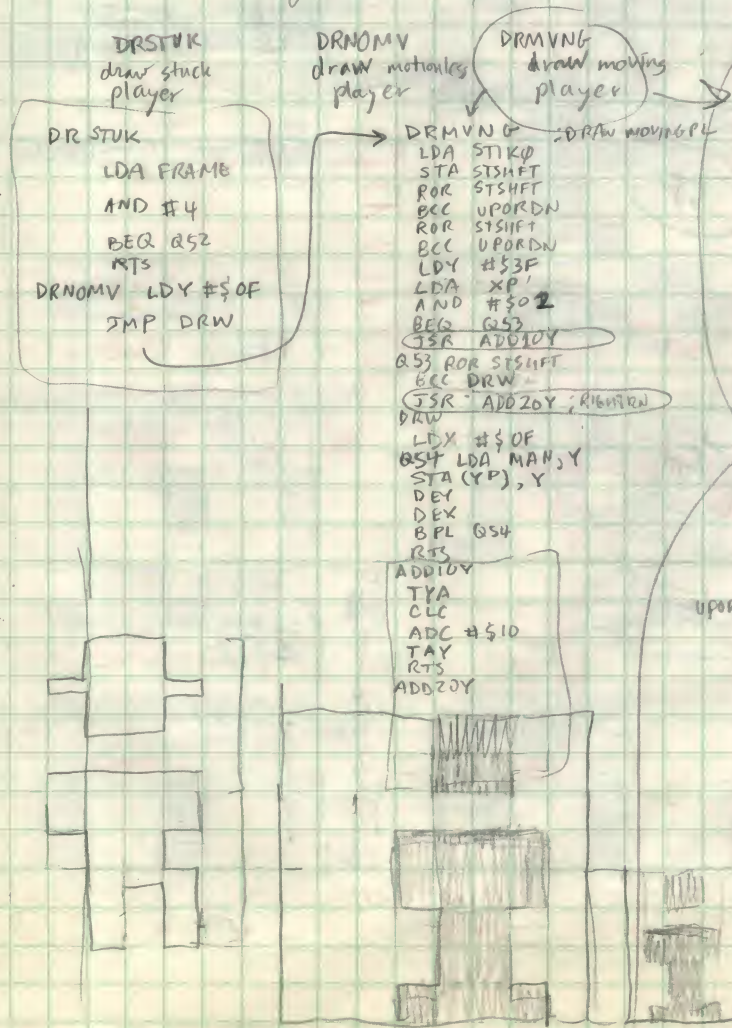
ANIMATE

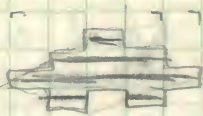
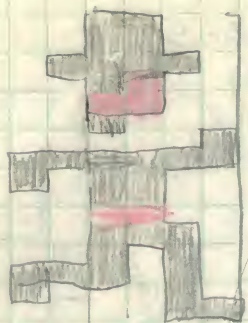
limit # snakes to 10

change can → corn
change snakes to ?
frozen w/ corn, corn flies off



don't drop X if owned
or if coll w/ can





test for Missile-Missile coll

PL X $\rightarrow$ or $\rightarrow$ X

PL Y $\rightarrow$ or $\rightarrow$ Y

X $\rightarrow$ X + 3 $\rightarrow$

NC X 6/3

INC Y 10/3

XM $\geq$ X $\rightarrow$

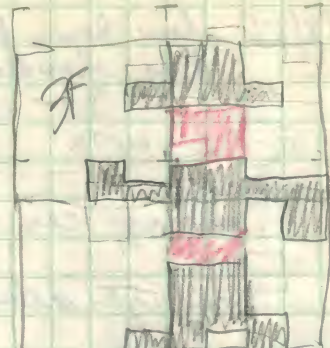
XM $\leftarrow$ X + 3 $\rightarrow$

YM $\geq$ Y $\rightarrow$

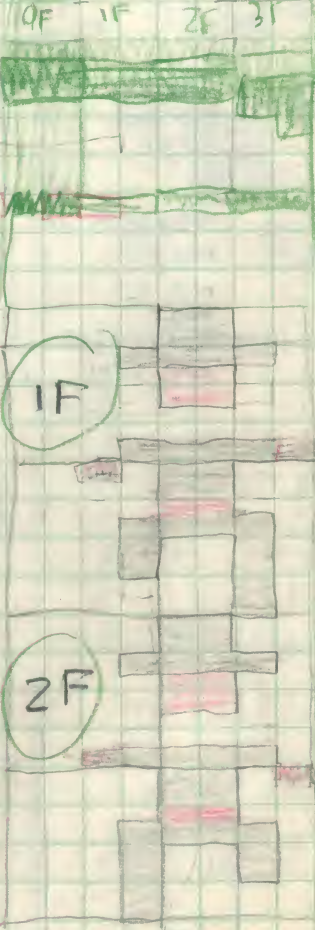
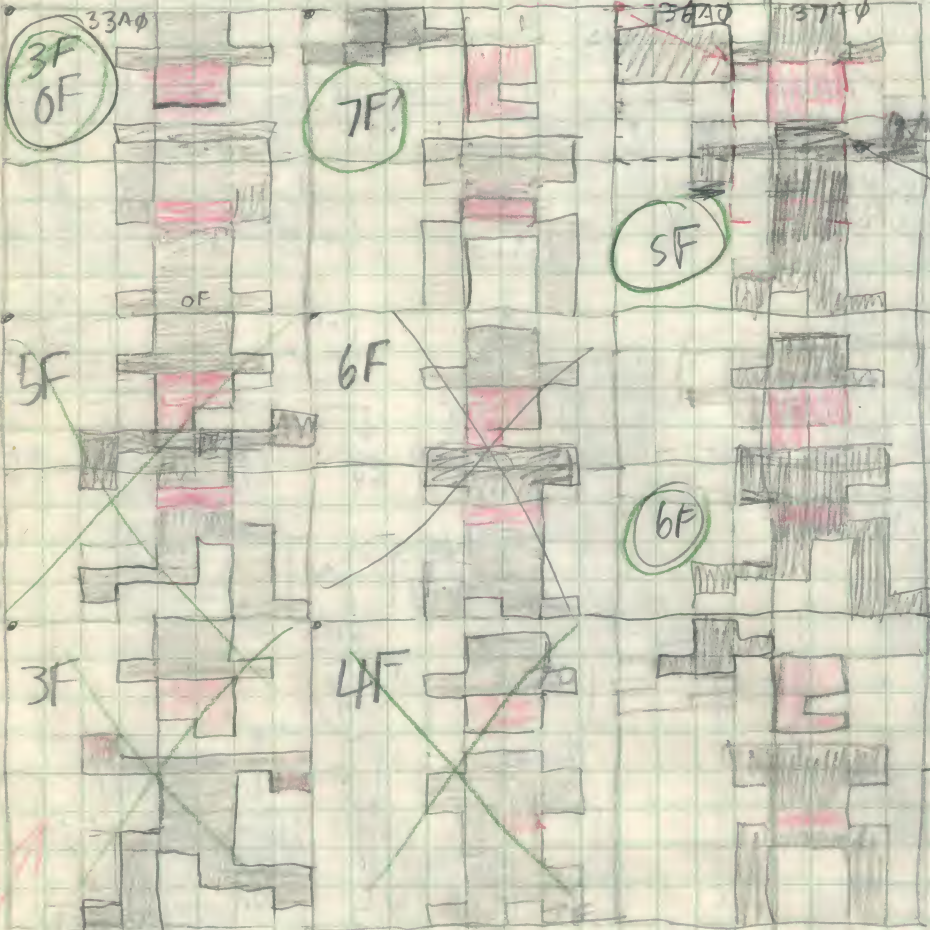
YM $\leftarrow$ Y + 3 $\rightarrow$

7/1/92 improvements to be made

MISSILES 3CXX vert. slowdown make bait disappear 10sec



34A
2
3
4
5
6
7
8
9
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z



Player 1 (P1) p1 player/P2
gets (M3)

LDA STOUT
BEQ Q65
LDA XPO
CLC #3
ADC XM2
CMP XM2
BCC BAL Q65
ADC #2
CMP XM2
BOS BAGE Q65
LDA YPO
ADC #3
CMP YM2
BCC BAL Q65
ADC #7
CMP YM2
BOS BAGE Q65
COWHTP
40 BYTES

LDA COLL
AND MASK
BEQ Q65

→ Q65 NO COLL BOT MT

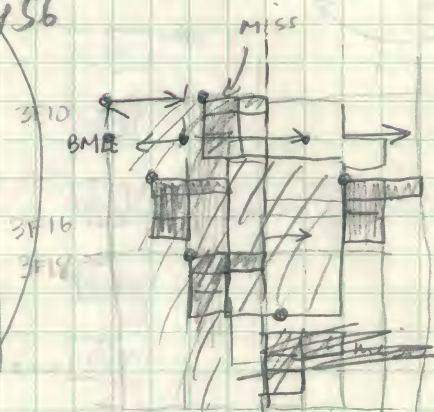
BMG = BAL = BCC
BMLE = BAGE = BOS

BNE COWHTP

LDA STOUT
BEQ Q65
LDA COLL
AND MASK
BNE COWHTP

1428
1456

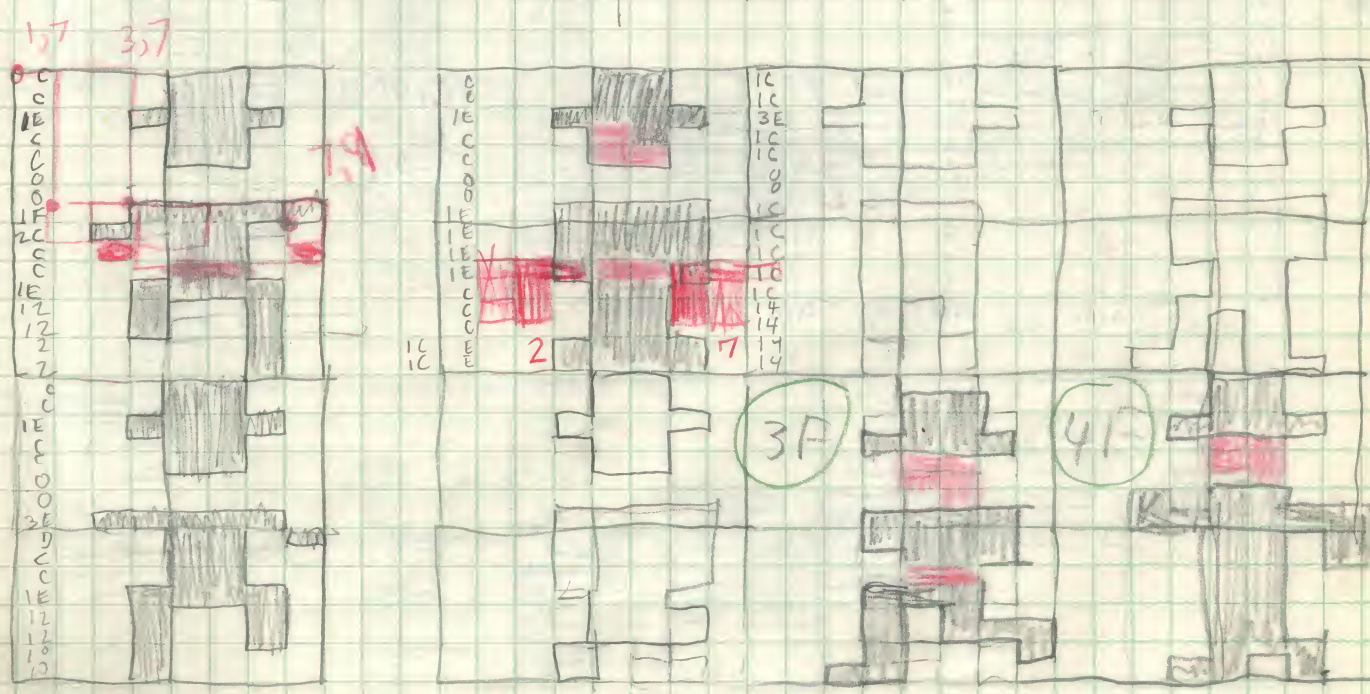
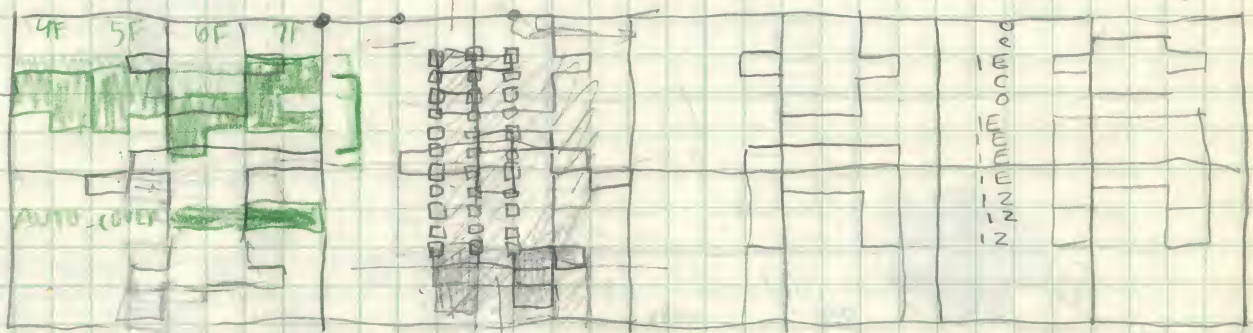
LDA XPO
CLC #2 ; CLR
ADC #2 ; CLR
CMP XM3
BMLE Q65
ADC #3 ; CLR
CMP XM2
BCC BMG Q65
LDA YPO
ADC #1 ; SETP
CMP YM2
BMLE Q65
ADC #5 ; CLR
CMP YM2
BCC BMG Q65
COWHTP

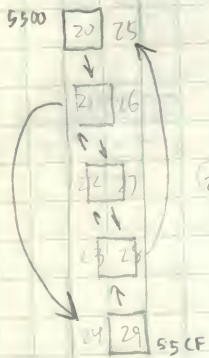
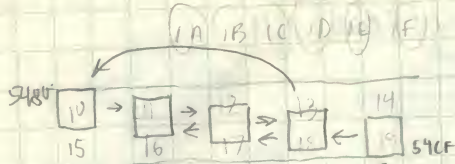


WTS

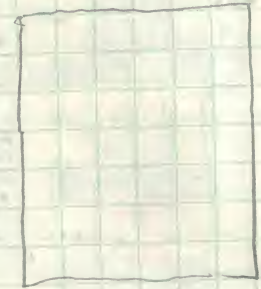
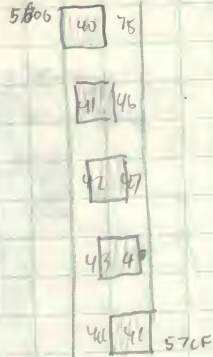
REDRMO 6324
REDRM2 6445

WIDEXD 6545
WIDEX2 6563



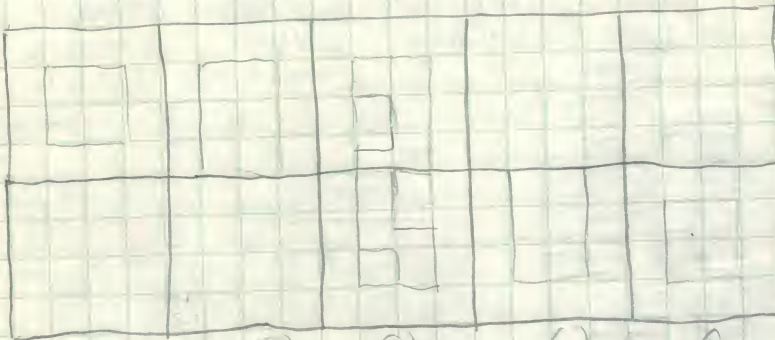


2A 2B 2C 2D 2E 2F

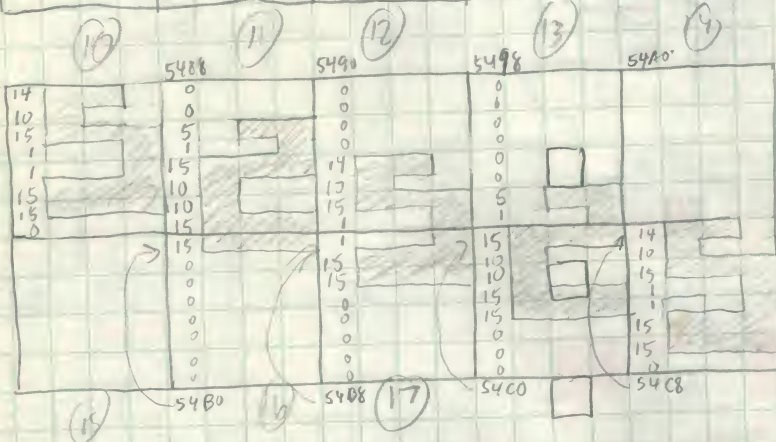


5510

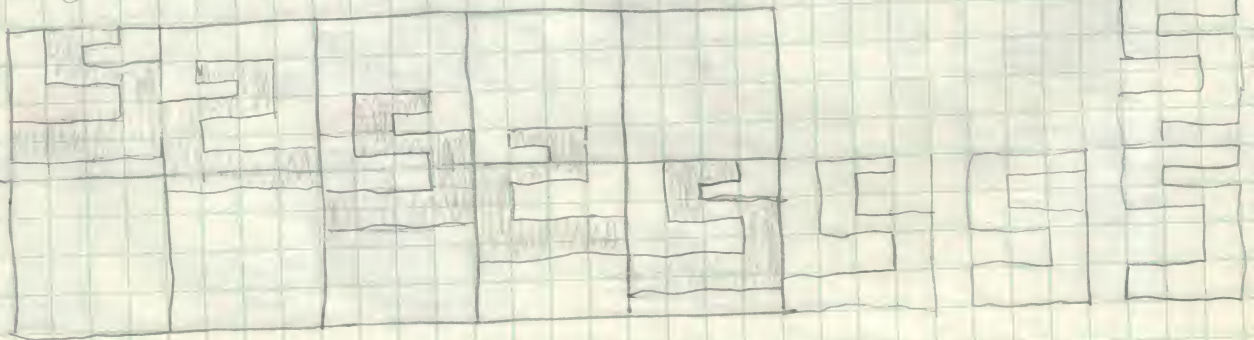
5570



5480



C.OBJ

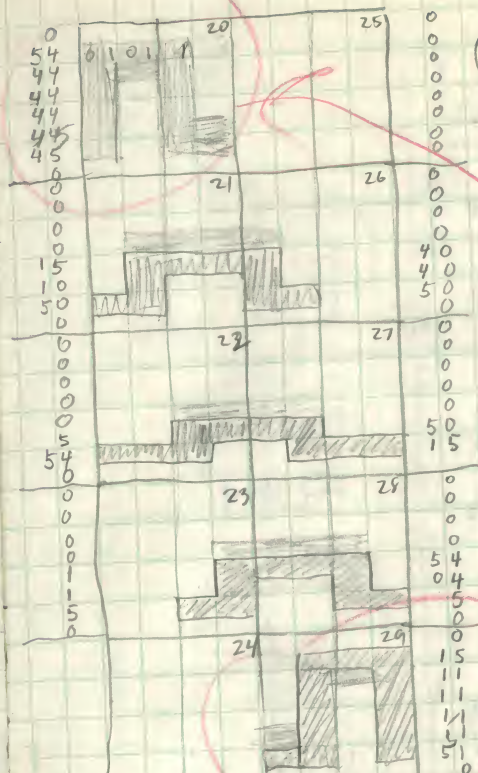


4/16/82

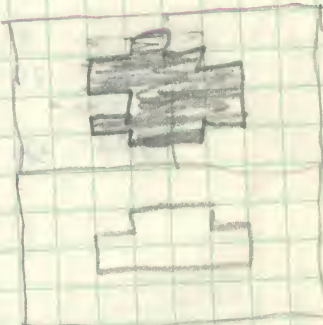
A, OBS

D, OBS
+ higher

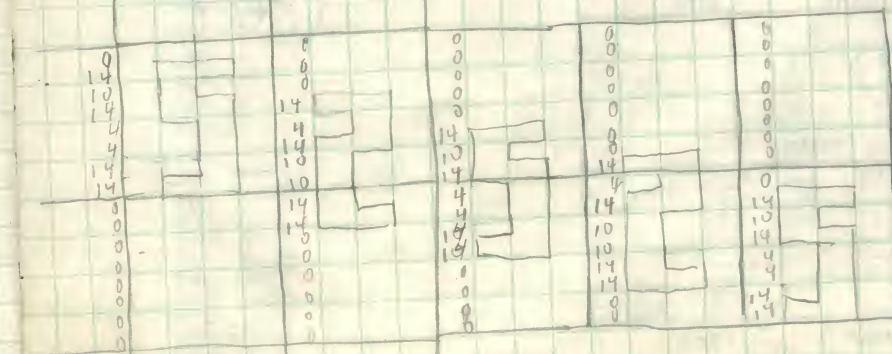
exposed



5530
5540

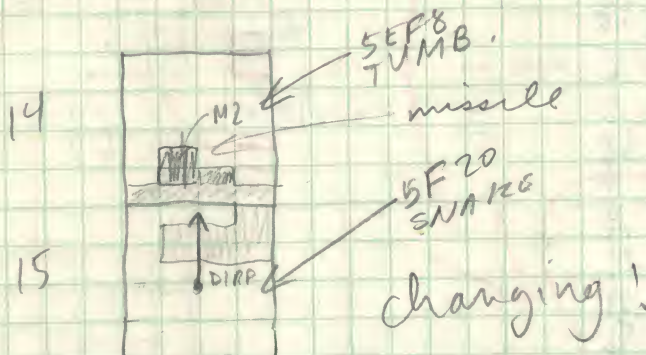
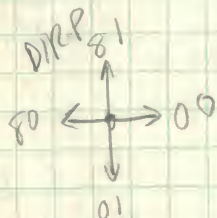


B, OBS

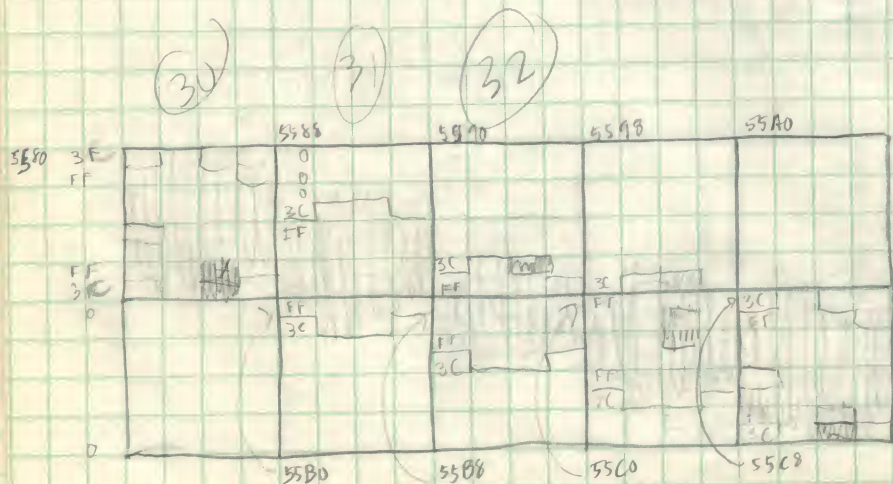


0 1 2 3 4 5 6 7

910	13
918	
920	15
988	
930	81



	0	1	2	3	4	5	6	7
690	20							
6A0	5F							



O.OBT whole set
cut down

corner
max off
cut off
P.OBT

JSR LOSEP
LDA RANDOM
BPL ONEDIR
JSR HORFLY
JMP VERFLY

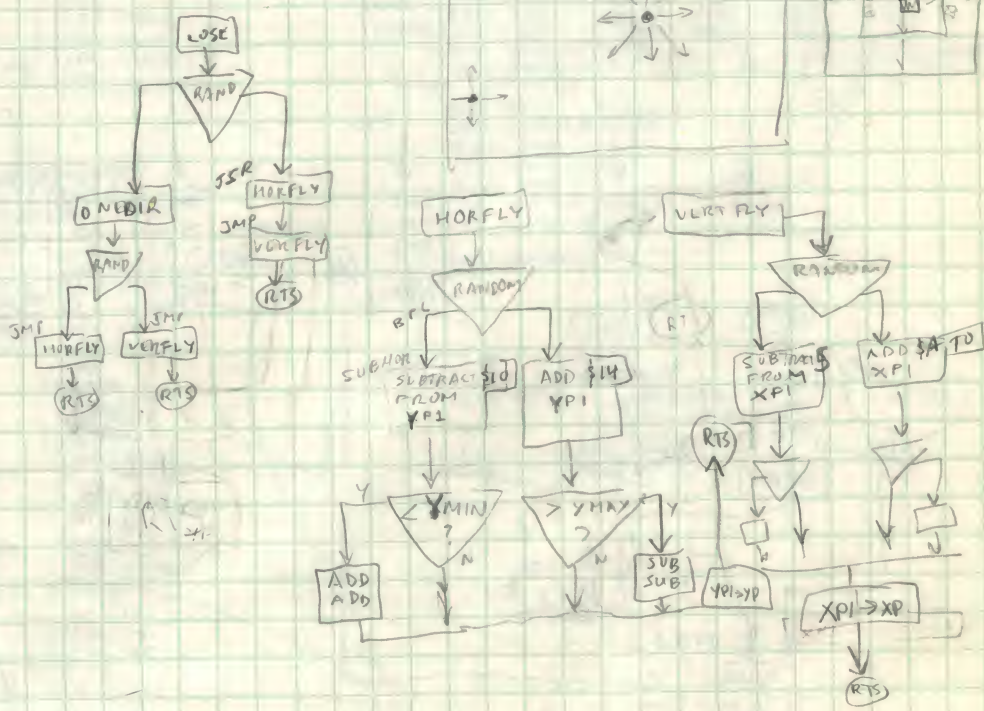
```

0 MEDIR
LDA RANDOM
BPL MORFLY
JMP VERFLY
MORFLY
LDA RANDOM
BPL SUBHOR < LDA
ADC # $14
CMP # $B8 ; YMAX
BCY YSTOR
SBC # $24
JMP YSTOR
SUBHOR
SBC # $10
CMP # $28 ; YMIN
BCY YSTOR
ADC # $24
YSTOR
STA YP1
STA YP
RTS

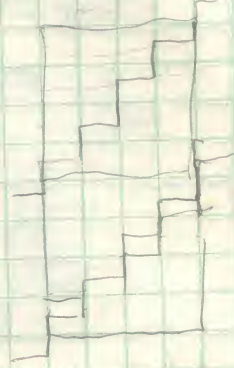
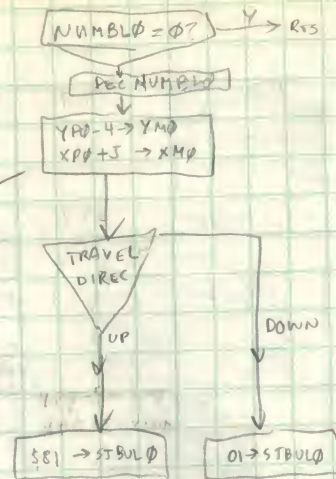
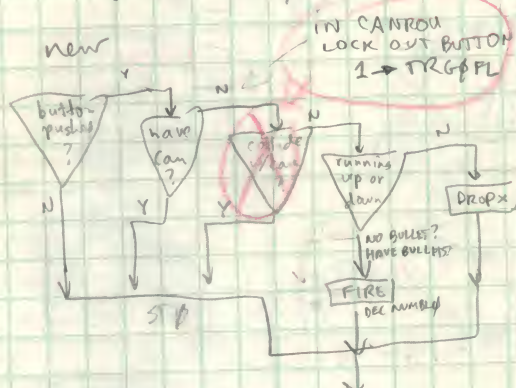
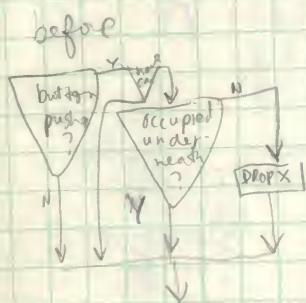
```

VERFLY

LDA	RANDOM	
BPL	SUBVER	← LTR
ADC	#50A	
CMP	#5C0	X MAX
BCC	XSTOR	
SBC	#50F	
IMP	YSTOR	
SUBVER		
SBC	#A05	
CMP	#534	X MIN
BCS	XSTOR	
ADC	#0F	
XSTOR		
STA	XPI	
STA	HPOSP1	
RIS		



canon - bag in hand, jump away when frozen ✓



BULLET STATUS { \$677 ST BUL0
\$677 ST BULZ

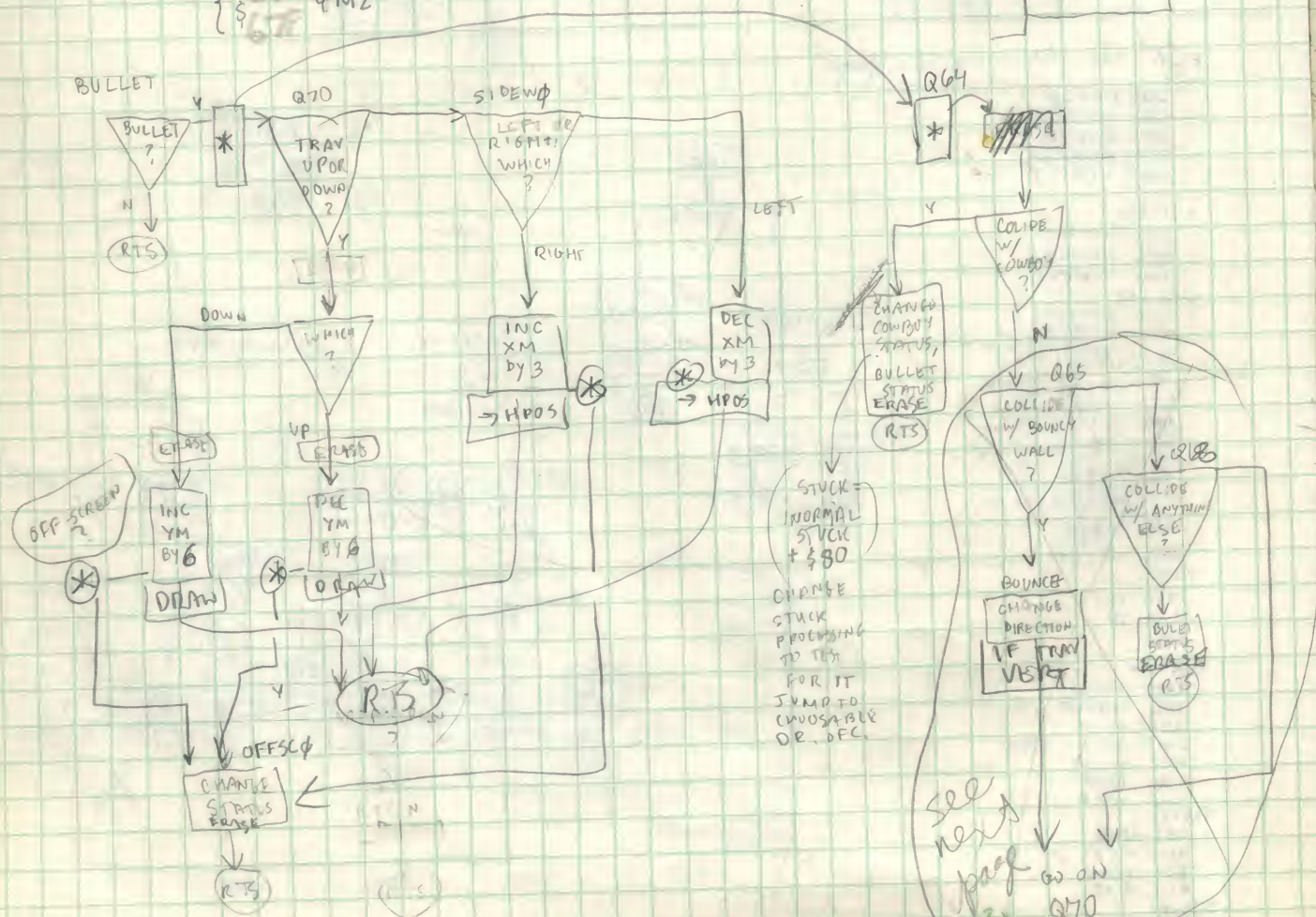
COORDS { \$678 XM0
\$678 YM0
\$678 XMZ
\$678 YMZ

NUMBER BUL { \$680 NUMBL0
\$681 NUMBLZ

00 = none
01 = trav down
02 = trav up

INC YM 81 = trav right
DEC YM 82 = trav left

INC XM
DEC XM



canon - nugget, cash, M1

odd collide w/ M1 at screen top: convert to cash

cash collide w/ M1 at screen bottom: buy bullets -6

CANON: LOCK OUT BUTTN

Q 64
65
66
67
68
70

MAIN MODULE, substitute lines

1204
1205
1206

with

LDA STATCH
CMP #580
BEQ ST0

LDA STIK0
CMP #50D
BNE Q60

LDA #501
STA STBUL0
Q60 CMP #50E
BNE Q61

LDA #581
STA STBUL0
JMP ST0

Q61 (DROP X)

2204
2205
2206

LDA STATCH
CMP #581
BEQ ST2

STIK1

Q62

STBUL2

Q63

STBUL2

ST2

insert 1228
→ 1230
2228
→ 2230

CMP #580
BNE Q66
LDA #0
STA STUK0
JMP GODOC

Q66 (Q67) ↑
+ cmp
NOP
NOP
NOP

MBASE = \$33

end:
JSR BUL0
JSR BUL2

3400 BUL0
* = \$3A00

LDA STBUL0
BNE Q64
RTS

LDA #MBASE
STA YBAS
LDA YMO
STA YP
LDA XPO
STA XP
JSR ERASE

LDA \$D008
AND #504
BEQ Q65

LDA #580 + \$3F
STA STUK2
HIT COWBOY

LDA #0
STA STBUL0
JSR ERSM0

Q65 LDA \$D000
AND #508
BEQ Q66
LDA #5Y0
CMP #550
BCC Q68
LDA #5XPO
CMP #57C
BCC BWRITE
LDA #582
STA STBUL0
JMP Q70

BWRITE LDA #502
STA STBUL0
BOUNCE RIGHT

Q70 LDA STBUL0

EMI SIDEM0
LSR A
BCC TRUP0
JSR ERSM0
LDA YMO
CLC
ADC #6
CMP #5DF
BCS OFFSC0
STA YMO
JSR REDRM0
RTS

ERASE FOR VERT MOTION

ERSM0
LDY YMO
LDA MBASE, Y
AND #5FC
STA MBASE, Y
INY
LDA MBASE, Y
AND #5FC
STA MBASE, Y
RTY

VERTICAL REDRAW

REDRM0
LDY YMO
LDA MBASE, Y
ORA #503
STA MBASE, Y
INY
LDA MBASE, Y
ORA #503
STA MBASE, Y
RTS

347E SIDEM0
LSR A
BCC TRLEFT

LDA XMO
CLC
ADC #503
CMP #5CF
BCS OFFSC1
STA XMO
STA HPOSM0
RTS
TRLEFT LDA XMO
SEC
SBC #03
CMP #528
BCC OFFSC1
STA XMO
STA HPOSM0
RTS
OFFSC1 JSR ERSM0

TRUP0

JSR ERSM0
LDA YMO
SEC
SBC #56
CMP #508
BCC OFFSC0
STA YMO
JSR REDRM0
RTS
OFFSC0 LDA #0
STA STBUL0
RTS

LDA STATCH
CMP #580
BEQ ST0

LDA STIK0
CMP #50D
BNE Q60

LDA STBUL0
BNE ST0
LDA NUMBL0
BEQ ST2

LDA #501
STA STBUL0
JMP GORDM0

Q60 CMP #50E
BNE Q61
LDA STBUL0
BNE ST0

LDA NUMBL0
BEQ ST0
LDA #502
STA STBUL0

JMP GORDM0

Q61

JMP Q70

Q68 LDA \$D000
BNE OFFSC1

NOP
NOP
NOP

INSERT OTHER
COLL. RESULTS

CORDM0 DEC NUMBL0
LDA YPO
SEL SBC #54
STA YMO
LDA XMO
ADC #56
STA XMO
JSR EIRSN0
JMP ST0

1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758

DOCTOR

```

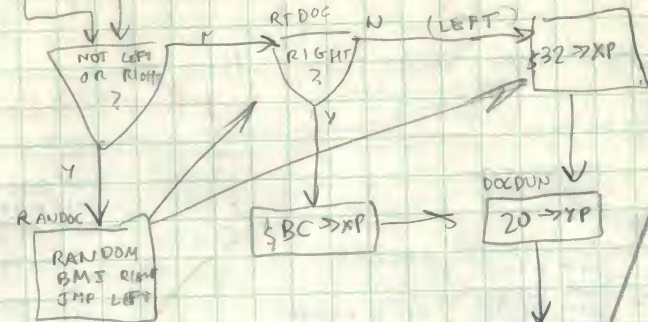
DOCT0
LDX #0
LDA STICK0
JMP DOCTX

DOCT2
LDX #2
LDA STICK1
JMP DOCTX

DOCTX
AND #500
CMP #500
BEQ RANDOC
CMP #508
BEQ RTDOC
LFD OC
LDA #52
STA XP
JMP DACDUN
RTDOC
LDA #5BC
STA XP
DACDUN
LDA #520
STA YP
RTS
RANDOC
LDA RANDOM
BMI RTDOC
JMP LFD OC
    
```

DOCT0
0 → X
STICK0 → A

DOCT2
2 → X
STICK1 → A

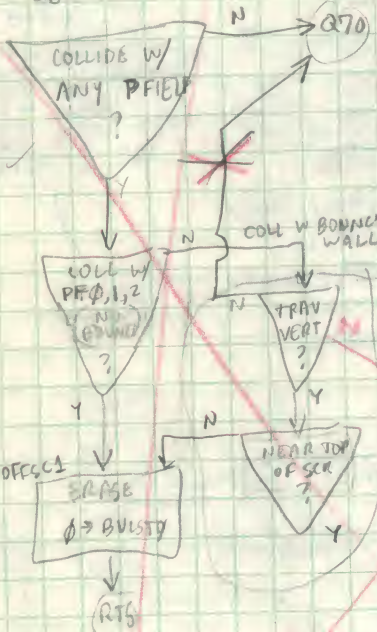


BCC BAL
BES BAGE

LDA \$D000
BEQ Q70
AND #5

PF 3 2 1 0
1 0 1 1

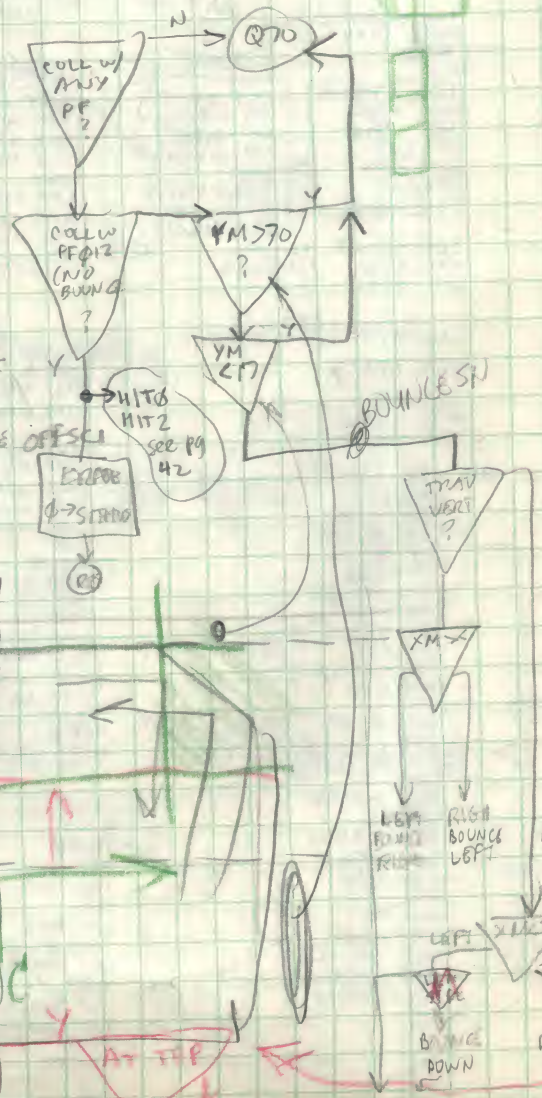
Q65



```

1140 RTS
1150 LDA $D000 ;MOPF
    BEQ Q70
    AND #507 ;MAPPF0,1,2
    BNE OFFSC1
    LDA BULST0
    BMI Q70
    LDA YP0
    CMP #550
    BCS OFFSC1
    LDA XP0
    CMP #570
    BCC BNRT0
    LDA #582
    STA STBUL0
    JMP Q70
    BNRT0
    LDA #581
    STA STBUL0
    CMP Q70
    
```

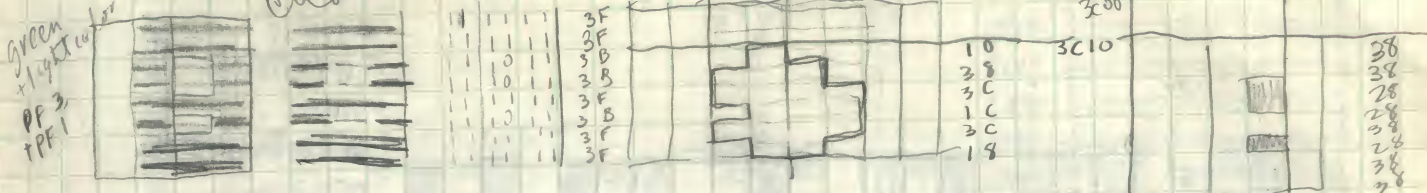
IF BOUNCED
50



CAR BOUNCED

AT TOP

Character



SOURCE
3E → SCRH
3EX → SCRL

DESTINATION
5C → 5CRH+2

```

NXTSQ
LDA (SURL), Y
STA (SURL+2), Y
DEY
BNE NXTSQ
RTS

```

LDA #7
LDY #526
QSI STA \$5C40,
BNE QSI

LDA #53E
STA SCR#
LDA #55C
STA SCR# + 2
LDA #580
STA SCR#
LDA #568
STA SCR# + 2
LDY #5
JSR NTSWQ
LDY #4
INC SCR#
LDA #590
STA SCR# + 2
JSA NTSWQ
LDY #3

LTA	#7
STA	\$569
STM	56A
STN	56B
STP	56C
	56D
	56E
	56F
	56G
	56H
	56I
	56J
	56K
	56L
	56M
	56N
	56O
	56P
	56Q
	56R
	56S
	56T
	56U
	56V
	56W
	56X
	56Y
	56Z

LDA #91B
STA \$5C6D
94
B3
E2
5D 09
5C 8A
B3
DC
5D 05
5D 2E

LDA #50
STA \$3318
STA \$3307
LDA #7A
STA

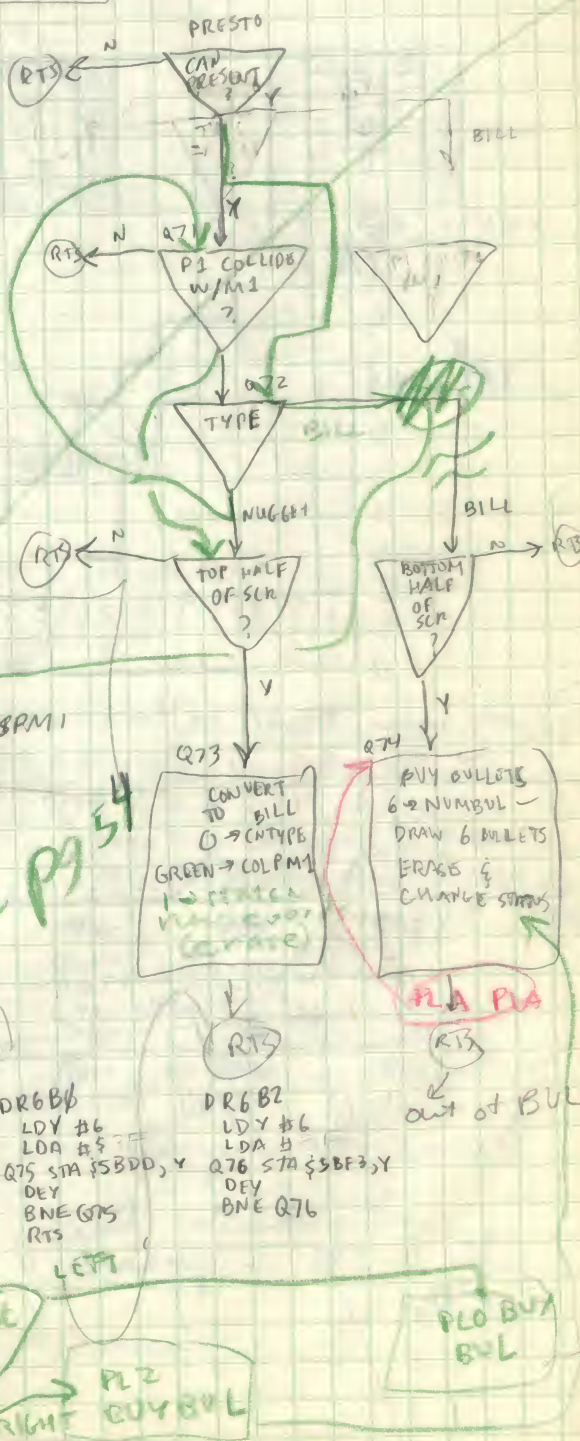
LDA STATCN
 BNE Q71
 RTS
 Q71 LDA \$DOOR - M1PL
 AND # \$02 ; M1PL COLL
 BNE Q72
 RTS
 Q72 LDA CNTYPE
 BNE NUGGET
 JMP BILL
 NUGGET
 LDA YM1
 CMP # \$80
 BCC Q73
 RTS
 Q73 LDA # 0
 STA CNTYPE
 LDA # \$RA ; BILL COLOR
 STA COLPM1
 RTS
 CMP #3, BCSQ71
 BILL
 LDA YM3
 CMP # \$80
 BCS Q74
 RTS
 Q74 LDA STATCN
 LSR A
 BCS P26BUL
 P06BUL
 LDA #306

BILL
LDA YMS
CMP #50
BCS Q74
RTS
Q74 LDA STATCN
LSR A
BCS P26BUL
P06BUL
LDA #506
STA NUMBL0
JSR ERASE
LDA #0
STA STATCN
JSR DR6B0
RTS

```
P26 BUL
LDA #506
STA NUMBL2
JSR ERASE
LDA #0
STA STATCN
JSR DR6B2'
RTS
```

```
DR6Bb
LDY #6
LOA #5
Q75 STA $BDD
DEY
BNE Q75
RTS
```

DR6B2
LDY #6
LDA H
Q76 STA \$5BF3, Y
DEY
BNE Q76



Voice PLO/PL2

✓ P ITPAT	0	1/3
✓ DROPX (FART)	1	1/3
✓ SHOOT	2	1/3
✓ BERZERK	3	1
✓ BERZERK2	4	3
✓ SNK → TUM	5	2
✓ TUM → SNK	6	4
✓ GOBBLE OWN X	7	2/4

SNK/TUM HIT 8 2/4

SNK PAT

TUM PAT

BOUNCE

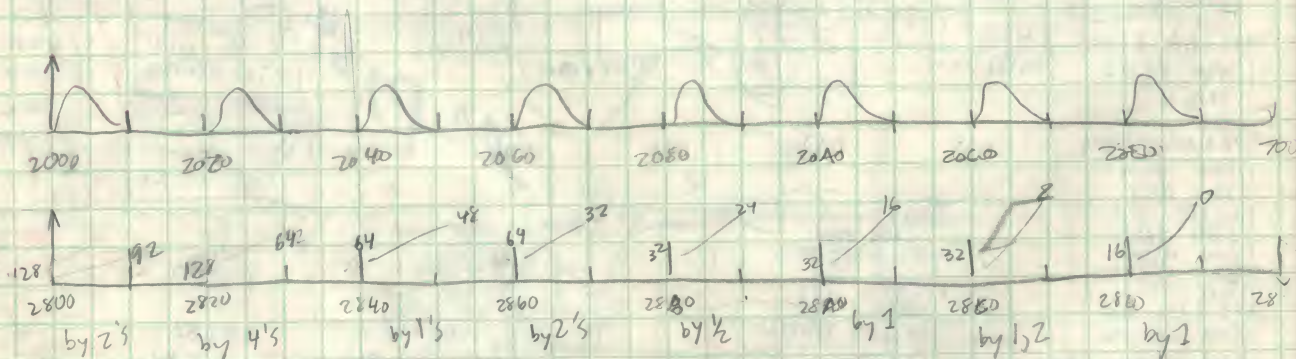
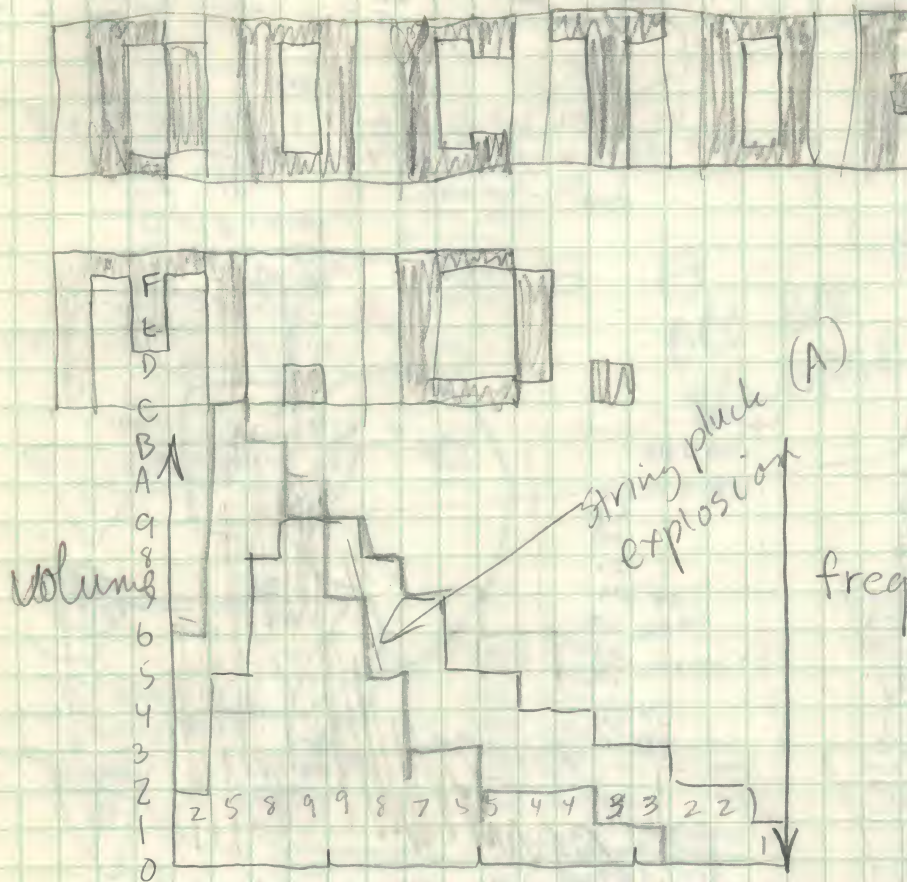
✓ OT IF X

MUSIC

TRADE /MUG →

~~BERZERK~~ →

SAVD



sounds

SW
①

PITPAT OF RUN ✓

CREATION - SNAKE, TMB ✓

(7) ~~AD886~~ ~~AD886~~ X

DIRECTION CHANGE - SNAKE, TMB
SNAKE → TMB
TMB → SNAKE

⑤ ⑥

③ ④

BERZERKED ✓

①

PROPX ✓

X DISAPPEAR

②

SHOOT ✓

BULLET BOUNCE ✓

③

BULLET HIT ✓

10 CAN APPEAR ✓

10 CAN → MONEY

10 MONEY → CLAIMED

VOICE
P1
1

P2
3

2

4

2

4

2

4

1

3

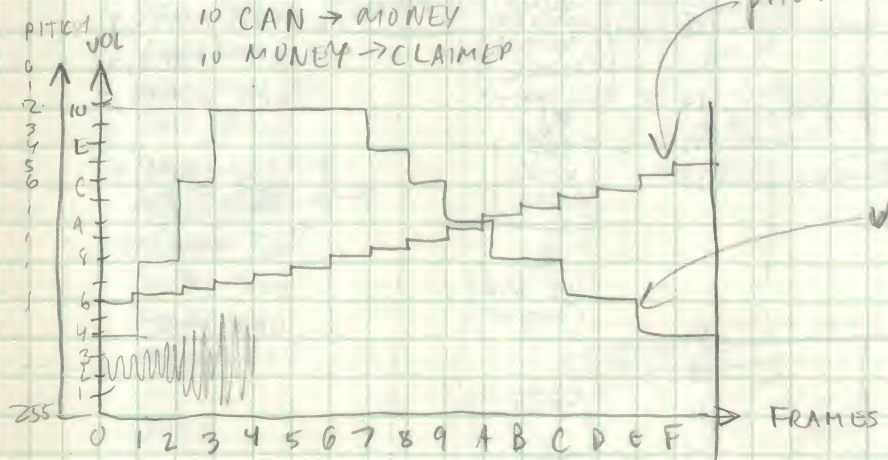
1

3

2

4

5800
5810
5820
5830
5840
5850
5860
5870



pitch

voice, pitch, type, volume

volume

X = \$3000

STA \$D01E
LDX #0
STX SNCNT1
STX AUD1

SN1 LDX SNCNT1
LDA AUD1
ORA \$2000,X
STA AUDC1
LDA \$2800,X
STA AUDF1

INC SNCNT1
BNE Q80

JSR ENDER
JMP SN1

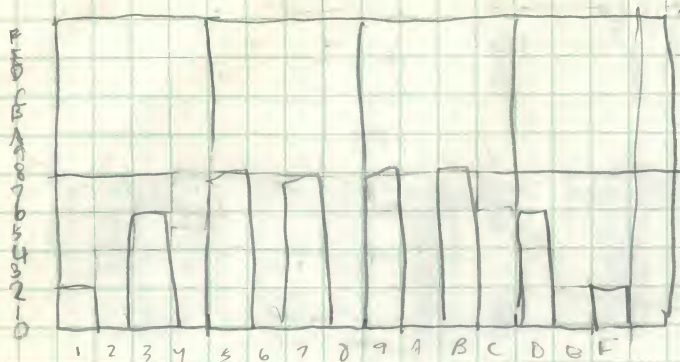
LDA AUD1
CLC
ADC #320
STA AUD1

6D0-6D7 P0X life
6D8-6DF P2X life
6E0-6E7 P0X pag
6E8-6EF P2X pag
6F0-6F7 P0X byte
6F8-6FF P2X byte

1. no shoot or drop X when stuck
2. different guy when hit ✓
3. give X's a lifetime?
requires 36 bytes of mem.
reserve 2 coord bytes use 48
1 lifetime byte bytes
4. at right edge, ignore right ✓
input, etc
5. can always within reach
6. can't shoot when
overlapping P2 or other
player
7. bounce fix

06D0
-06FF

6D0 XLP0
6D8 XHP0
6E0 XLP2
6E8 XHP2
6F0 XLIFE0
6F8 XLIFE2



2000 vol $\rightarrow$ 4600

2000 freq $\rightarrow$ 4700

FREQ
DUR
AUDC
UPDN

SNØV1
ØF $\rightarrow$ DUR1

SNØV3
ØF $\rightarrow$ DUR3

SN1V1
1F $\rightarrow$ DUR1

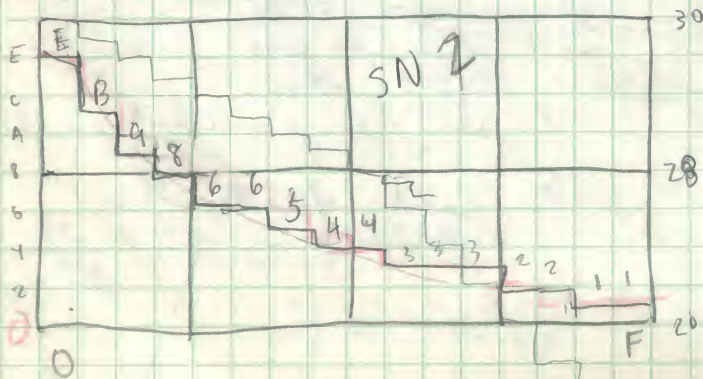
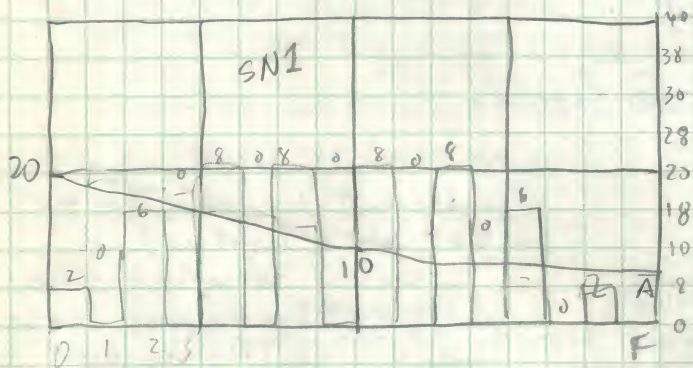
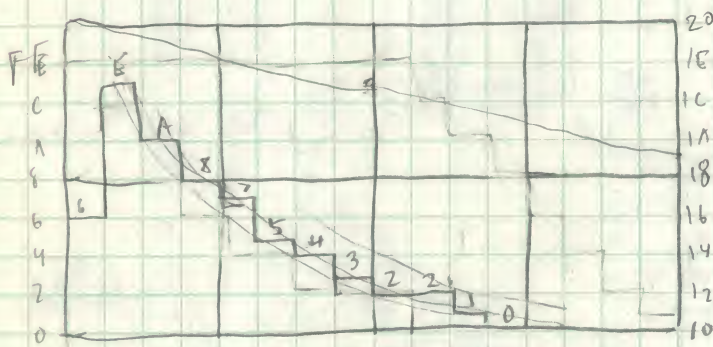
SN1V3
1F $\rightarrow$ DUR3

VOLUME = 54600
FREQ = 54700

VOICE1

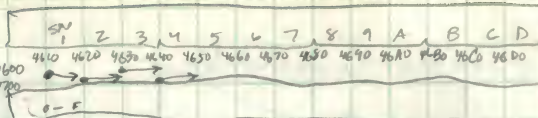


LDV #6, 1, 1 = VOLUME * 2
NXVØI
LDX DUR1, Y
BEQ QUIET
LDA VOLUME, X
ORA AUD1, X
STA AUDC1, Y
LDA FREQ, X
STA AUDF1, Y
LDA DUR1, Y
BNE STILL
BNE STILL
STILL
DEX
DEX
BNE NXVØI



DUR1 → AUD1
DUR2 → AUD2
DUR3 → AUD3
DUR4 → AUD4

VOL 4600
FREQ 4700

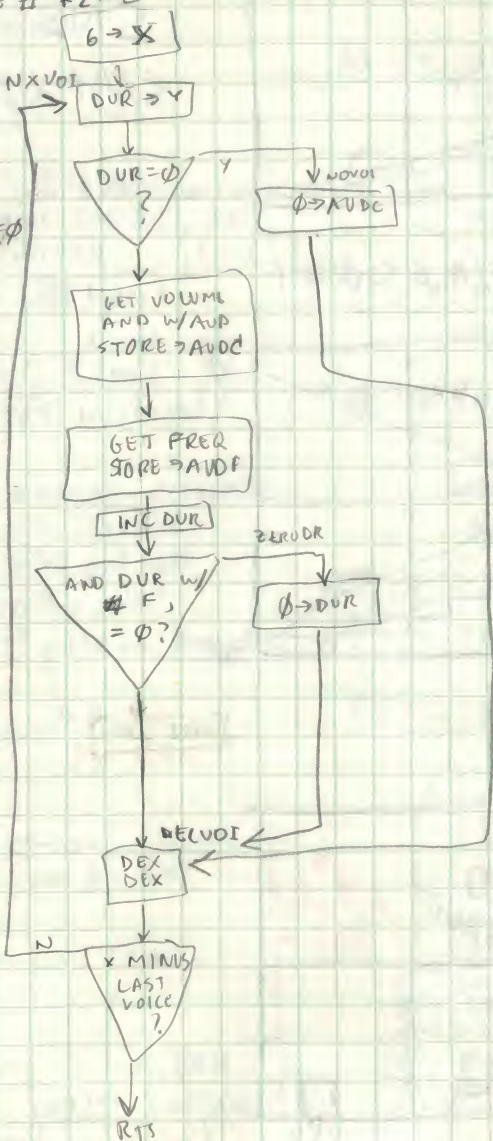


VOLUME = 4600
FREQ = 4700

X = 3960
SOUND HITCLR
STA HITCLR
LDX #6

; X = VOICE # * 2 - 2
; Y = DUR

NXVOI
LDY DUR1, X
BEQ NOVOT
LDA #0
STA AUDC1, Y
BEQ ENVOT, UNC
LDA VOLUME, Y
ORA AUD1, X
STA AUDC1, X
LDA FREQ, Y
STA AUDF1, X
INC DUR1, X
LDA DUR1, X
AND #50F
BEQ ZERODR
~~DEC DUR1, X~~
DECVOI
DEX
DEX
BPL NXVOI
RTS
NOVOT
LDA #0
STA AUDC1, X
BEQ DECVOI, UNC
ZERODR
LDA #0
STA DUR1, X
BEQ DECVOI, UNC

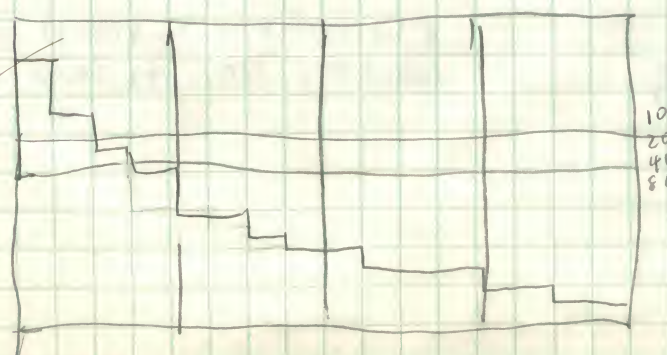


SN1V1; PL0 FART
LDA #50
STA DUR1
LDA #80
STA AUD1
RTS

SN1V3; PL2F

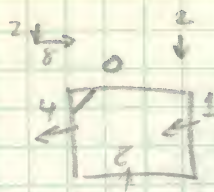
SN2V1; PL0 SHOOT
LDA #50
STA AUD3
LDA #70
STA DUR1

SN2V3; PL2 SH



5/4/82

progr. requirements
for shooting SNK, TUMB



additional changes

38

1 exact transformation of missile
coords to SCREEN MEM
(not too difficult)

M 1 change range of players
(4 sides)

2. New way to save SNK locations:
Screen memory, not charac coords, } saves
save both present & previous } time!
Screen locations (extra 32 bytes)

C 3. Make cans fly when unowned
and out-of-range

C 4. erase, redraw M1 based
on can status

C 5. move M3 offscreen upon ~~PLEN~~ ^{SAVGP}
(see line -235?) ^{ON}

3 New snk created: first ^{x coord.} 0 ^{coord}
location used (not simple increment)
(see next page (page 39)) ✓

4. Snk destroyed: search amongst present
snk locations, prev snk locations
until location found; zero out all
coords, dec NUMSNK, verify color?
zero out previous sq., put pile in present sq.

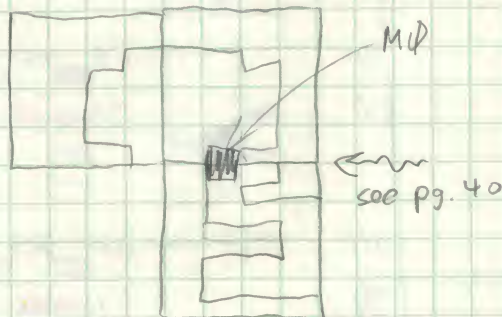
last; improve tumbleweed
appearance, draw scales,
gun store, put in bldgs.

M 6. Prohibit shooting while on PFZ

M 7. Hosp. coords

r. don't prohibit nugget
from appearing @ screen
bottom

\$C3



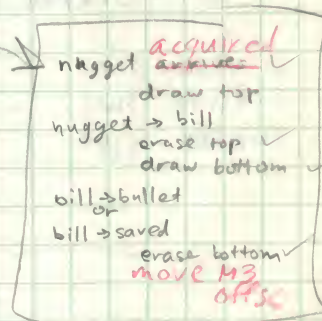
5. If it can't find SNK in list, BRK.
eventually, make it ignore collision
and continue (treat as miss)

6. Teach snakes to recognize piles as x's

OR, IF PLENTY OF PROCESSOR TIME,
SIMPLY SAVE PREVIOUS X-Y COORDS
OF SNAKE

```
LDA XPI
CMP #52D ; LEFT REACH
BNE RCHRTS
CMP #5CC
BNE RCHRTS
JSR EDGFLY
RCHRTS
```

~~RTS~~
RCHRTS RTS



39

NEW SNAKE
set words

OLD WAY

LDX NOMSNK

$$\left. \begin{array}{l} Z0X \rightarrow Z0X-1, X \\ Z1Y \rightarrow Z0Y-1, X \\ 1 \rightarrow DIRPZ0, X \end{array} \right\} \text{abs. index}$$

new way:

 find first $Z0X, X = \emptyset$
 use that space

Z0X =

Z1X =

Z2X =

⋮

ZFX

Z0Y

Z1Y

Z2Y

⋮

ZFY

DIRPZ0

DIRPZ1

DIRPZ2

⋮

DIRPZF

```

LDX #0
NEXTX0
LDA Z0X, X
BEQ SVZCOR

INX
TXA
AND #50F
BNE NEXTX0
BRK
  
```

```

SVZCOR
STX SNKIDX
JSR SCTOXY
LDX SNKIDX
STA Z0X, X
TXA
STA Z0Y, X
LDA #51
STA DIRPZ0, X
LDA EGGSC
STA SCRLZ0, X
LDA EGGSC
STA SCRHZ0, X
  
```

THEN $ZX \rightarrow Z0X-1, X$
 $ZY \rightarrow Z0Y-1, X$
 and $1 \rightarrow DIRPZ0, X$
 $SCRH \rightarrow SCRHZ0, X$

SCRLZ0 = \$690

SCRHZ0 = \$6A0

SNKIDX = \$684

make cars fly

56C0

D8



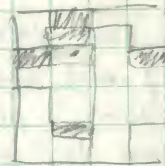
56B0

D7



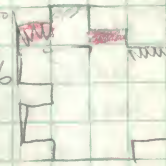
56A0

D9



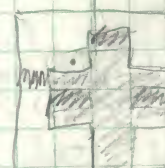
56B0

D6



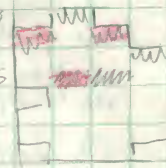
56D0

DA



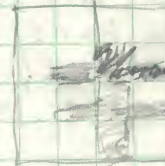
56A0

D5



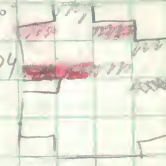
56D8

DB



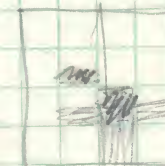
56B0

D4



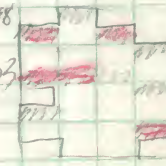
56E0

DC



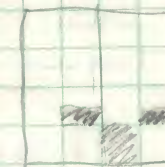
5680

D3



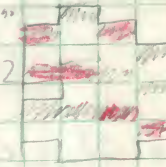
56E8

DD



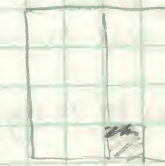
56C0

D2



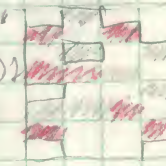
56F0

DE



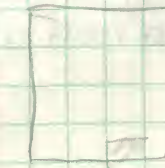
56E8

D1



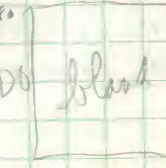
56F8

DF



56E0

D0

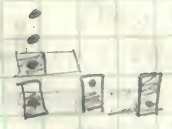


6320

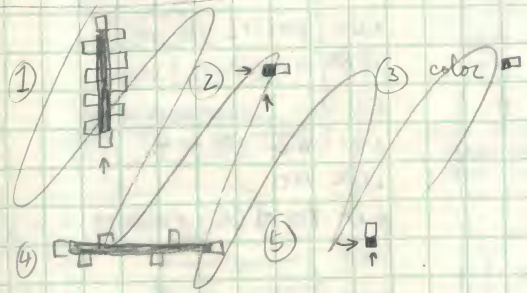
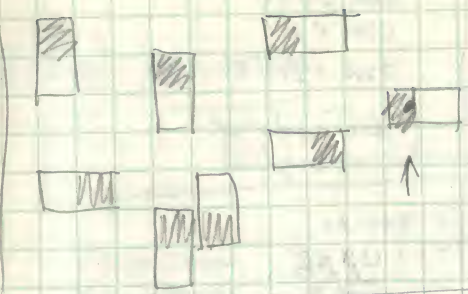
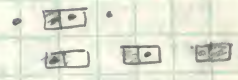
6355

Blank

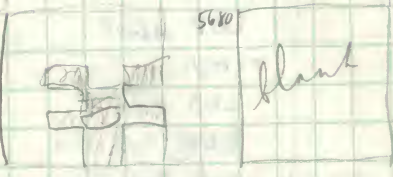
- 1 x cord match
- 2 y cord or y cord ± 1 match?
- 3 color match?



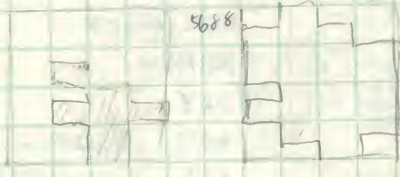
y cord match
x cord or x cord ± 1 match?
color match?



56C0

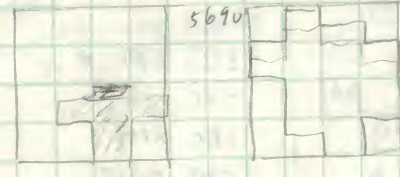


56B0

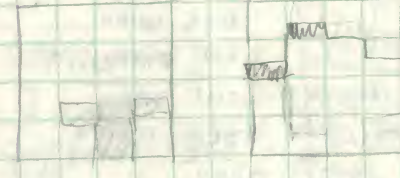


5688

56D0



5690



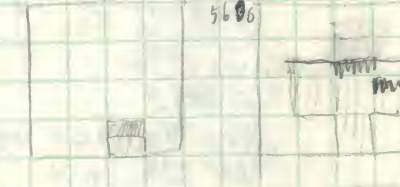
56E0



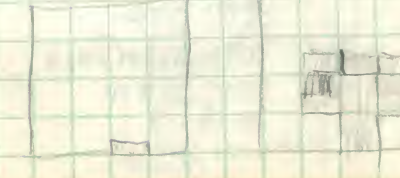
56A0



56F0



5606



* 2506
MITRON NOP

973 BHITFL = \$685
HITCOL = \$686

XMS = \$687
YMS = \$688

LINK
FINDSC = \$4960

AND #3 ; M00R M2 PF
BNE QHR1
JMP HITDUN
QHR1
CMP #3
BNE QHR2
BRK ; PF01 BOTH
NOP
NOP
QHR2

AND #1
EOR #1 ; 0=PF0
STA HITCOL ; 1=PF1
; TRANSFORM M COORDS

LDY BHITFL ; Y=0 or 2
LDA XMO,Y
SEC

SBC #4 ; OFFSET
LSR A
LSR A

STA XMS
LDA YMO,Y
SEC

SBC #9 ; OFFSET

LSR A
LSR A
LSR A

STA YMS
LDY \$50F

TXMCH
LDA ZOX,Y
BEQ NXSNKX
CMP XMS
BEQ CHEKY

NXSNKY
DEY
BMI TRY Y
BPL TXMCH ; UNC

CHEKY
LDA ZOX,Y
SEC
SBC YMS
BEQ TCOLOR
CMP #5FF
BNE QHR3
LDA DIRPZO,Y
CMP #1
BEQ TCOLOR
BNE NXSNKY ; UNC

5050
5090
5630
5710
5950
5970
6000



all those
missiles
belong
to this
character's
space

QHR3
CMP #501
BNE NXSNKY
LDA DIRPZO,Y
CMP #581
BEQ TCOLOR
BNE NXSNKY ; UNC

TRY Y
LDY \$50F
TYMCH

LDA ZOX,Y
BEQ NXSNKY
CMP YMS
BEQ CHEKY

NXSNKY
DEY
BMI FAILZF
BPL TYMCH ; UNC

CHEKY
LDA ZOX,Y
SEC

SBC XMS
CMP #5FF
BNE QHR4
LDA DIRPZO,Y

CMP #00
BEQ TCOLOR
BNE NXSNKY ; UNC

QHR4
CMP #501
BNE NXSNKY

LDA DIRPZO,Y
CMP #580
BEQ TCOLOR

BNE NXSNKY ; UNC

FAILZF
BRK ; RTS LATER
NOP ; (HIT "X")
BRK

TCOLOR
; NOW Y = SNK# HIT
LDA SCRLZO,Y
STA SCRZ

LDA SCRHZO,Y
STA SCR H

LDA DIRPZO,Y
STA DIRZF
LDX #2
JSR FINDSC

see code at
right

REPLAC
LDX #2
LDA #0
STA (SCRL,X) ; PREVIOUS

LDA BHITFL ; 0 or 2
LSR A ; 0 or 1
CLC

ADC #2 ; 3 or 4
LDX #0
STA (SCRL,X) ; PRESENT

NOP
NOP ; HIT SOUND
NOP

LDA BHITFL ; 0 or 2
BEQ ENO
ENZ JSR OFFSCI

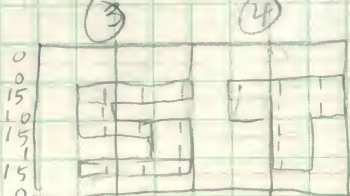
PLA
PLA
RTS ; LEAVE BUL0

ENO JSR OFFSC2
PLA
PLA
RTS ; LEAVE BUL2

LDA #0
STA ZOX,Y
DEC NUMSNK

HITDUN
PLA
LDA BHITFL
BEQ PL0DUN
JMP OFFSC3 ; LEAVE BUL2

PL0DUL JMP OFFSC1 ; LEAVE BUL0



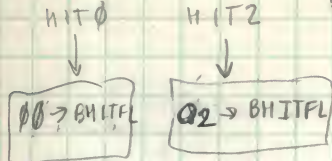
MISSILE COLLIDE W/ PF0, 1, 3

BHITFL = \$6

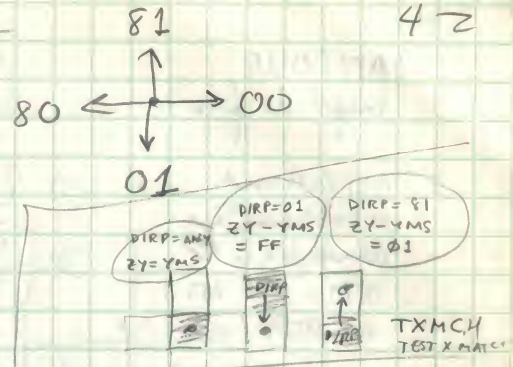
HITCOL = \$6

81

42



BULLET HIT
SNK ROUTING



TRANSFORM
BUL COORDS
TO X-Y
COORDS,
SCREEN LOCATION

→ SCRL, SERH
CAN GET SCRL, SERH FR. TABLE

AND #03
(PF0, PF1
COLL)

Y
HITDUP
IGNORE
(PASS THRU
CACTUS)

CMP #3
BOTH COLL.
=?

AND #1 PF0 COLL
EOR #1 INVERT
A → HITCOL
0 = PF0 COLL
1 = PF1 COLL

F → X (SNK INDEX)

TXMCH
X COORD MATCH?

CHECKY
Y COORD MATCH?

GET DIRP
+ or -
or 0

YCOORD
±1, 0
MATCH?

TCOLOR

COLOR
MATCH?

X'S
0-4
? → RTS

REPLAC
REPLACE
PREV SQ
W/ 0

REPLAC
REPLACE
PRESENT
SQ W/ 'X'

SOUND

HITDUP

ERASEM0
OR ERASEM2
0 → STBUL0
OR STBUL2

PLA RTS
PLA

F TRY

0 → X (SNK INDEX)

TYMCH
Y COORD MATCH?

CHECKX
X COORD MATCH?

GET DIRP
+ or -
or 0

XCOORD
±1, 0
MATCH?

Y

Y

Y

Y

Y

Y

Y

TCOLOR
LDX #0
LDA (SCRL, X)
CMP #10
BCS T1
BRK ; 0-7
T2 CMP #12A
BCS T3
JMP TESPFA
T3 CMP #14A
BCS TESPFI
BRK ; 8A-FF
TESPFA
LDA HITCOL
BEQ COLORX
BRK ; COLOR NOT ON
TESPFI
LDA HITCOL
CMP #1
BEQ COLORX
BRK ; COLOR NOT ON
REPLAC

FAILZ F
BRK
NOP
BRK

JMP IGNOR (LATER)

GET ZY
SUBTRACT
YMS

= 0 ?
Y → MATCH

= FF ?
Y → MATCH

DIRP = 1 ?
Y → MATCH

DIRP = 81 ?
Y → MATCH

NO MATCH

CANROU bugs to fix

1. HITROU → not a subroutine jump back to Q70, not RTS no PLA



2. push missile down
erase before pushing
redraw after!

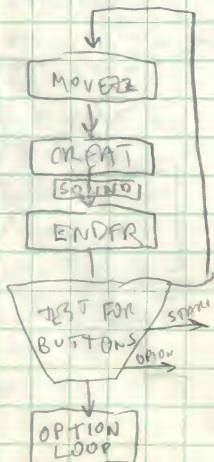
3. missile-bounce
don't bounce if goes down

4. change thrust

SET COL
CHANGE CO-ORD
CLEAR SCREEN
DRAW BORDER
DRAW SCREEN
CLEAR BOT
DRAW TITLE

ERASE BOTH M1
M3
CLEAR P.6

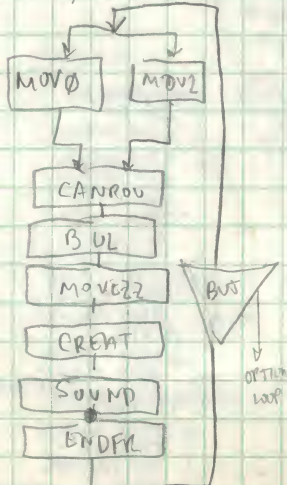
ATTRACT
LOOP



CLEAR SCREEN
DRAW BORDER

SET COL
SET CHANGE CO-ORD
DRAW HOSTIT
CANCRS
3F → CRECRS
CLEAR BOT
DRAW TRIANGLES
TITLE
DRAW M3
ENDFR HITLER ← CLEAR P.6

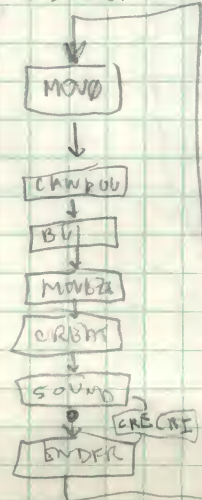
GAME1 LOOP



C S
D B
SET COL
SET CHANGE CO-ORD

CANCRS
FF → CRECRS
CLEAR BOT
DRAW TRIANGLES
TITLE
ERASE M3
ENDFR HITLER ← CLEAR P.6

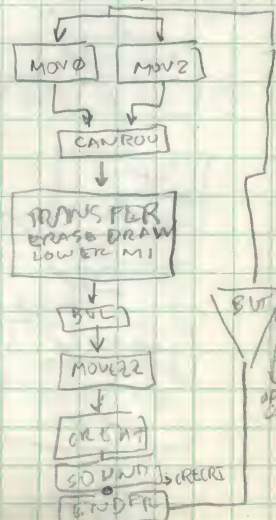
GAME 1 LOOP

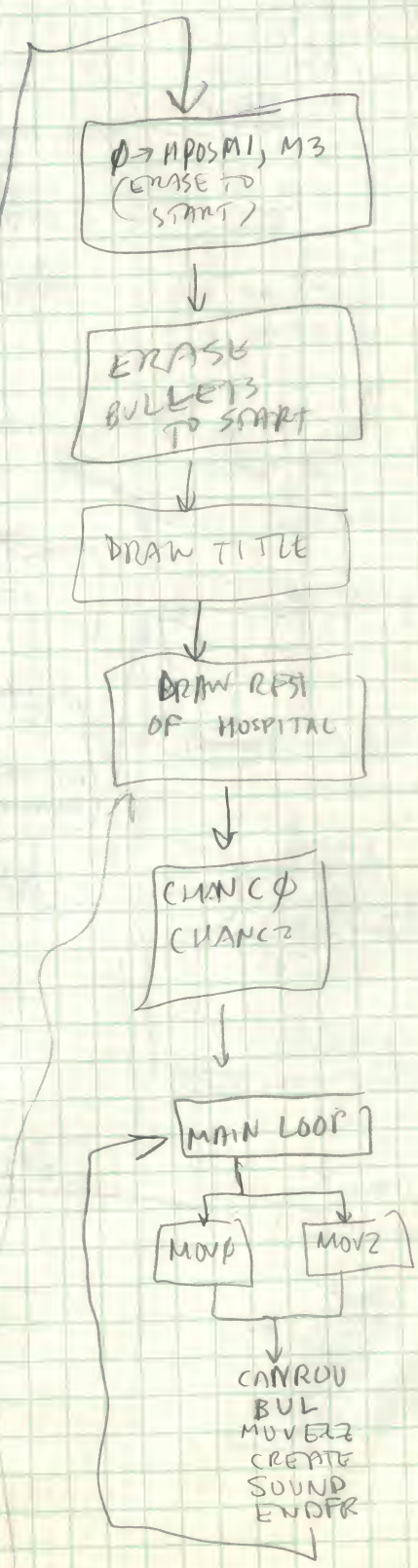
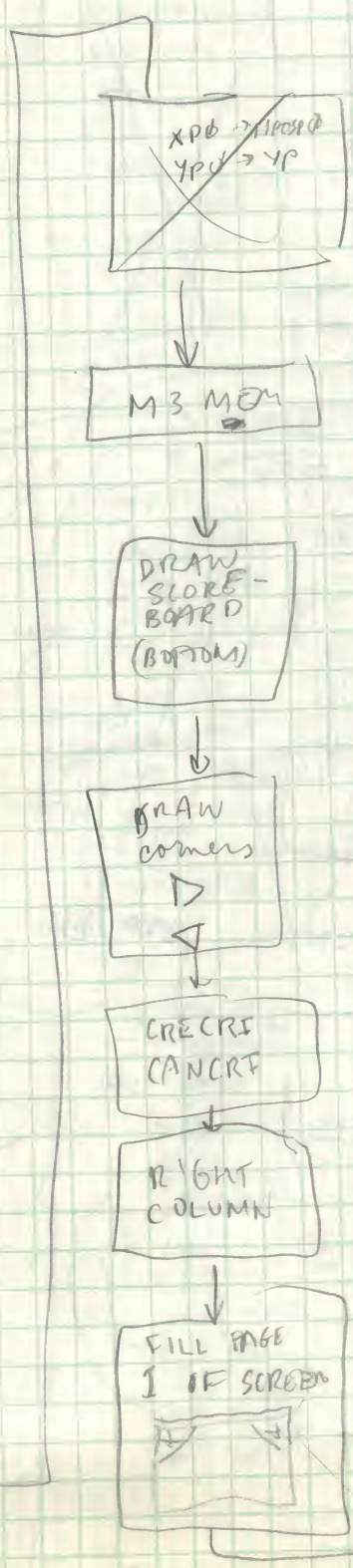
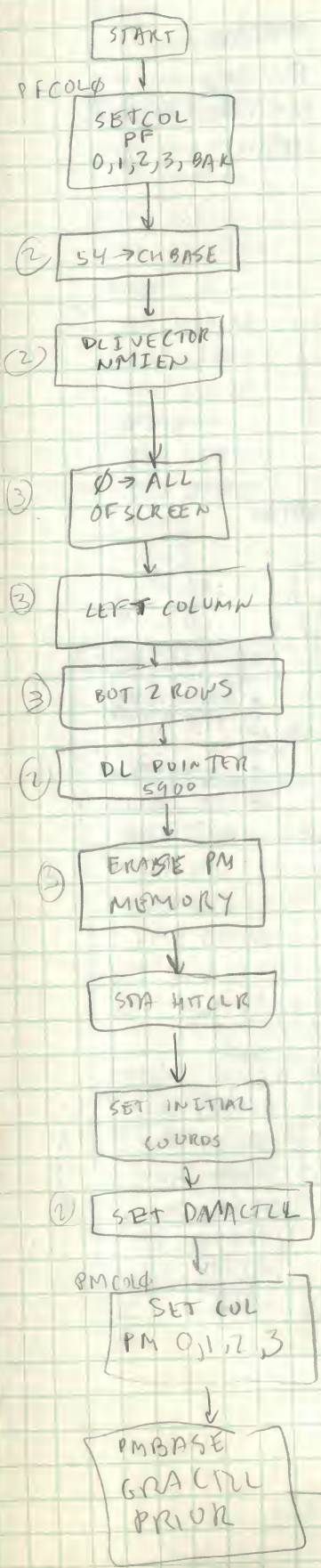


C S
D B
SET COL
SET CHANGE CO-ORD

CANCRS
FF → CRECRS
CLEAR BOT
DRAW TRIANGLES
TITLE
ERASE M3
ENDFR HITLER ← CLEAR P.6

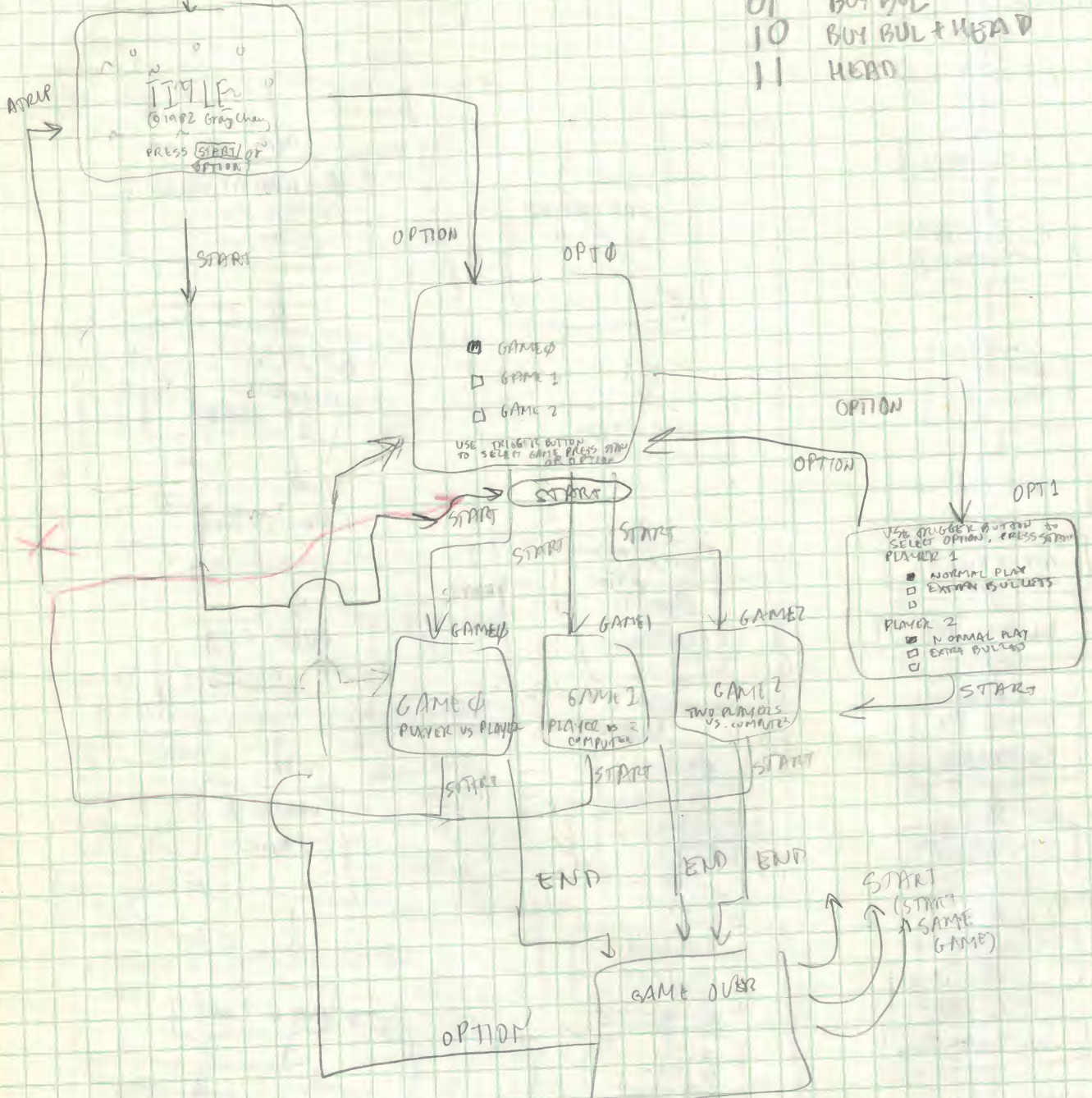
GAME 2 LOOP





LOAD PROGRAM

00 NORMAL
01 BUY BUL
10 BUY BUL + HEAD
11 HEAD



*3C06

JSR INIT → include all
JSR COLPF JSRCOLPM
LDA #0

STA CHANC0

STA CHANC2

JSR CLRSCR

JSR CLRBOT

JSR TITLEA

JSR PL12X

JSR CLRPG6

ATRLP

JSR MOVEZ

JSR CREAT

JSR SOUND

JSR STRTES

JSR OPTTES

JSR ENDER

JSR ENDER

JMP ATRLP

START

LDA STRFL

BEQ SS0

CMP #1

BEQ SS1

JMP GAME2

SS0 JMP GAME0

SS1 JMP GAME1

OPT0

JSR DROPT0

BOPT0

JSR POLL0

JSR STRTES

JSR OPTTES

JSR ENDPR

JSR ENDPR

JMP BOPT0

OPTTES

LDA #8

STA CONSOL

LDA CONSOL

LSR A

LSR A

LSR A

BCC OPT1DB

RTS

OPT1DB

LDA OPTFL

BEQ PLOP1

RTS

PLOP1

PLA

PLA

OPT1

JSR DROPT1

BOPT1

JSR POLL1

JSR STRTES

JSR OPTTES

JSR ENDER

JSR ENDER

JMP BOPT1

STRFL
OPTFL

STRTES

LDA #8

STA CONSOL

LDA CONSOL

LSR A

BCC STRDB ; DEBOUN

LDA #0

STA STRFL

RTS

STRDB

LDA STRFL

BEQ PLSTRT

RTS

PLSTRT

PLA

PLA

LDA #1

STA STRFL

STRT

OP0TES

LDA #8

STA CONSOL

LDA CONSOL

LSR A

LSR A

LSR A

BCC OP0DB

RTS

OP0DB

LDA OPTFL

BEQ PLOP0

RTS

PLOP0

PLA

PLA

LDA #1

STA OPTFL

OPT0

OP1TES

LDA #8

STA CONSOL

LDA CONSOL

LSR A

LSR A

BCC OP1DB

RTS

OP1DB

LDA OPTFL

BEQ PLOP1

RTS

PLOP1

PLA

PLA

LDA #1

STA OPTFL

OPT1

JSR DROPT1

BOPT1

JSR POLL1

JSR STRTES

JSR OPTTES

JSR ENDER

JSR ENDER

JMP BOPT1

INIT

SET

COLPF

COLPM

0 → CHANC0

0 → CHANC2

CLEAR SCR

CLEAR BOT

TITLE

0

DROP 12 X'S

CLR PG.6

MOVEZ

CREATE

SOUND

JSR

START

JSR

OPTION

ENDER

START

STARTFL → A

0 = 1 = 2

GAME0

GAME1

GAME2

PROPT0

DRAW OPT0

SCREEN

OP0

POLL STICKS,

UPDATE

STRFL

CONSOL

START

OPTION

PLORT1

PLA

PLA

DRAW OPT1 SCR

BOPT1

ENDER

POLL1

JSR STRTES

JSR OPTTES

JMP BOPT1

RTS

10

[illegible]

PRESS SELECT
E R BUT TON
CLAIM JUMPER

MULTI-GAME REFINEMENTS

non-crashed
variable

48

DROPT
JSR CLRSCR
LDA #20
STA \$5D13
STA \$5DDB
SAT \$5EA3
RTS

DROPT2
JSR CLRSCR
LDA #20
STA \$5D12
STA \$5D62
STA \$5DB2
STA \$5E52
STA \$5EA2
SAT \$5EF2
RTS

POLLO

~~LDA TRIGFL~~
~~BNE PLDRTS~~
~~LDA TRIGFL~~
~~BNE PLDRTS~~
~~LDA TRIGFL~~
~~BEQ PUSH10~~
~~LDA TRIG1~~

LDA TRIGFL
BEQ PUSH10
LDA TRIG1
BEQ PUSH10
LDA #0
STA TRIGFL
PUSH10
LDA TRIGFL
BEQ ADV0
RTS
ADV0
JSR DROPT+3
LDA #1
STA TRIGFL
INC GAMENO
LDA GAMENO
CMP #3
BNE GDISP
LDA #0
STA GAMENO
GDISP
BEQ DG0
CMP #1
BEQ DG1
STX \$5EA3 ; GAME3
RTS
DG0
STX \$5D13
RTS
DG1
STX \$5DD5
RTS

2FB0 6125
CF02 6170



GAME 0



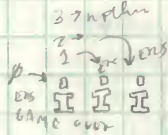
GAME 1



GAME 2

POLLI

2FA1 AD

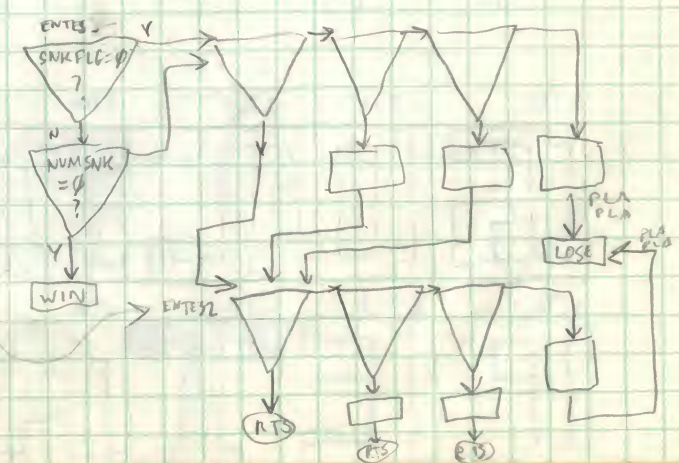
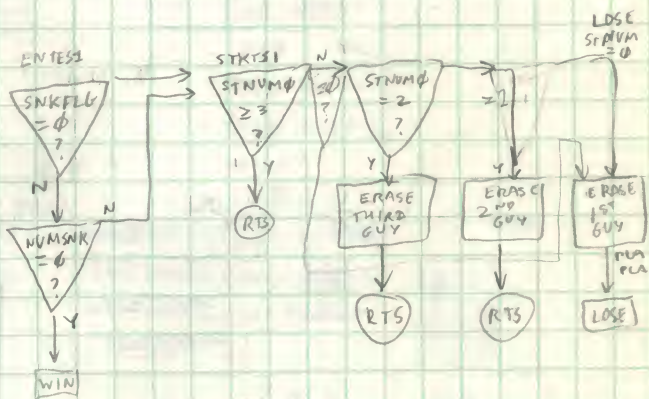
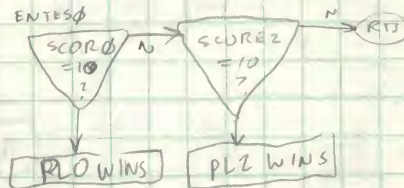


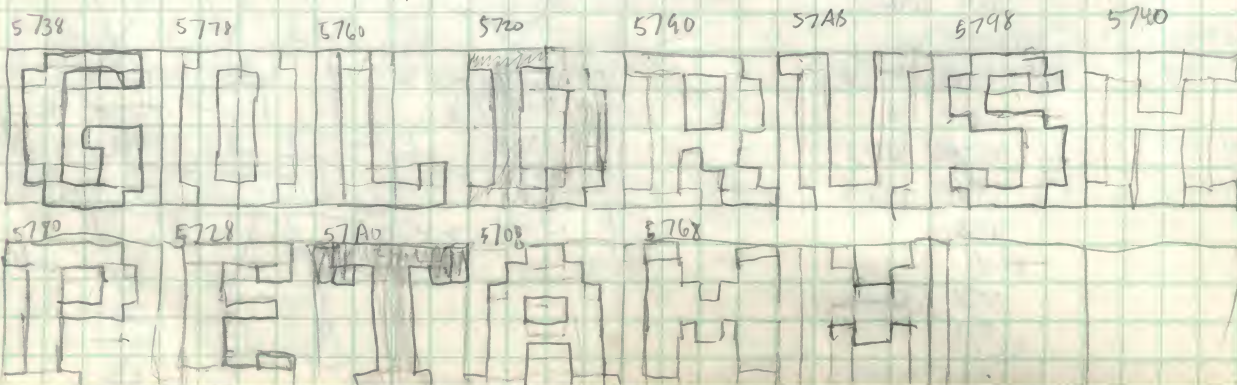
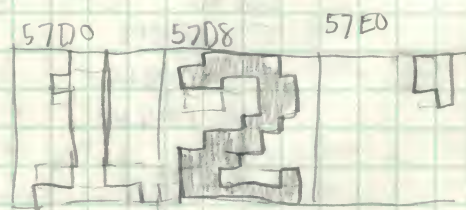
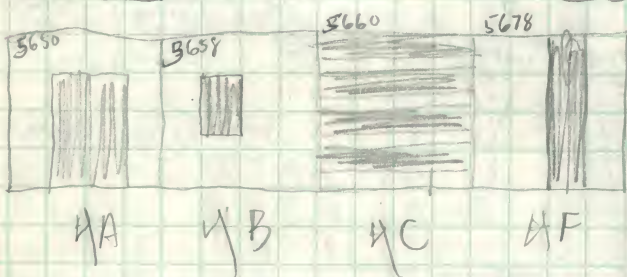
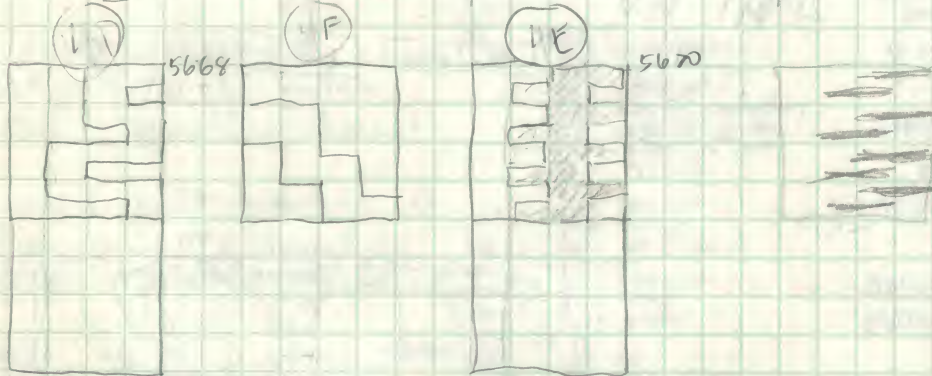
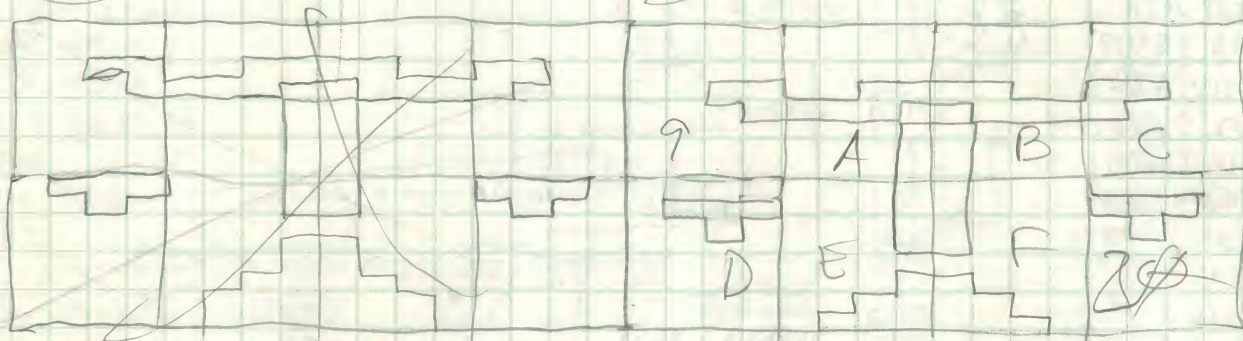
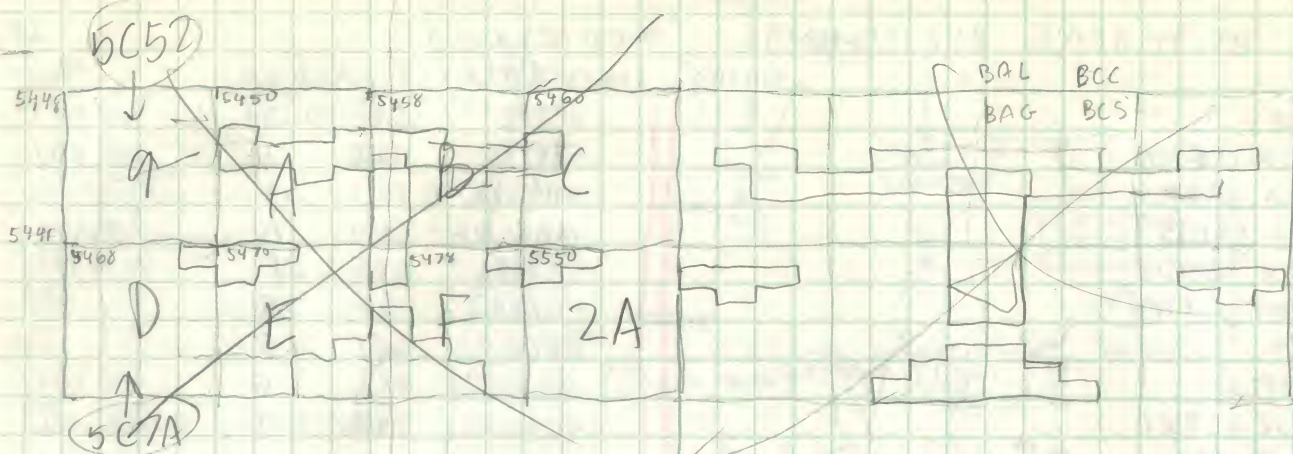
OPTFL = 6E1
STRTFL = 6E2
GAMENO = 6E3
MAXSNK = 6E4
CRECR1 = 6E5
CRECR2 = 6E6
MXBUL0 = 6E7
MXBULZ = 6E8
COLCRI = 6E9

0
0
0
A
OF
3F
6
6
1 7



END-OF-GAME ROUTINE





ENTES0

```

LDA SCOR0
CMP #50A
BCS P0WINS
LDA SCOR2
CMP #50A
BCS P2WINS
RTS
P0WINS JSR TITLE3
JMP ENLOOP
P2WINS JSR TITLE4
ENLOOP
JSR ENDFR
JSR STRTES
JSR OP0TES
JMP ENLOOP+2

```

ENTES1

```

LDA SNKFLG
BEQ STKTS0

```

```

LDA NUMSNK
BNE STKTS0

```

JSR TITLE6 : WIN
JMP ENLOOP

STKTS0

```

LDA STNUM0
BEQ ER3RD
CMP #3
BCS STRTS
CMP #2
BEQ ER3RD
LDA #0
STA $5B
RTS
ER3RD

```

; STNUM0 = 1

```

LDA STNUM0
CMP #3
BCS STBRK
TAX

```

```

LDA #0
STA $5B, X
TAX

```

```

BNE STRTS
JSR TITLE6 ;LOSE
JMP ENLOOP

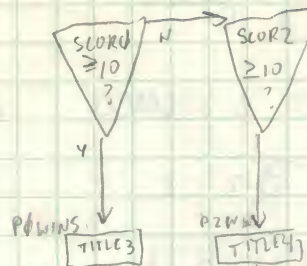
```

```

STRTS RTS
STBRK BRK

```

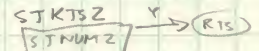
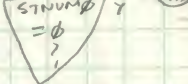
ENTES0



ENTES1



STKTS1



11 → 00
10 → 01
01 → 10
00 → 11

STKTS0

```

LDA STNUM0
BEQ STRTS
CMP #4

```

BCS STBRK (STRTS LATER)

```

TAX
LDA #0
STA $5BE3, X
TAX

```

```

CMP #3
BNE STRTS
JSR TITLES ;LOSE
JMP ENLOOP

```

```

STRTS RTS
STBRK BRK

```

```

LDA STNUM2
BEQ STRTS
CMP #4

```

BCS STBRK (STRTS LATER)

```

EOR #03
TAX
LDA #0
STA $5BF1, X
TAX

```

```

CMP #3
BNE STRTS
JSR TITLES ;LOSE
JMP ENLOOP

```

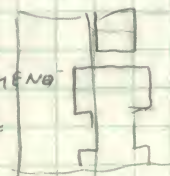
PLUGUYS

```

2 LDX #2
2 G1 LDA #ZF
3 STA $5BE4, X
3 STA $5BF1, X
1 G2 DEX
2 BPL G1
1 RTS

```

ZF @5578



5/21/82

Works quite well - 3 games
work still needed.

START

main title

options

GAME0 <

EXTRA BULBS

GAME1 <

DIFFICULTY
LEVELS

GAME2 <

DIFFICULTY
LEVELS

MAIN

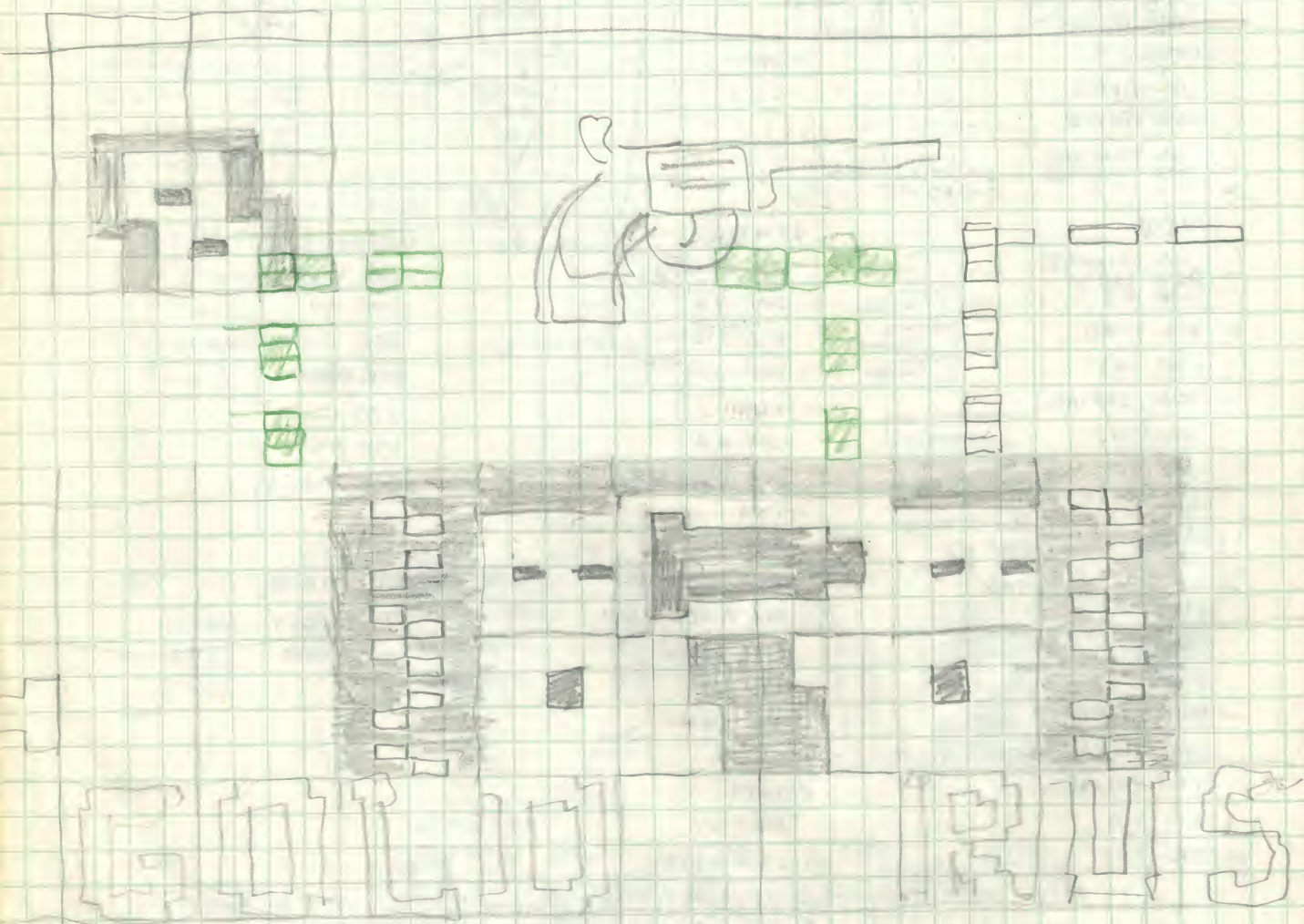
CREAT

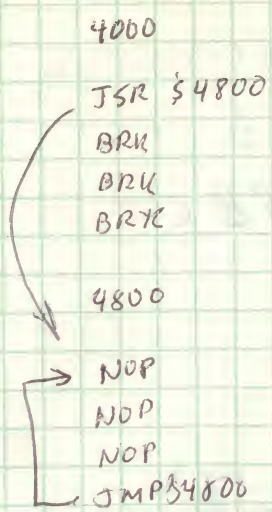
SMOD

CANRON

BUL

SCALE, TOWN
RICOCHET





4000 G (BRK)

S = F5

IF5, 1FF

9C FC / B9 00 / F8 56 /

- 2 LDX #
- 3 STX ABS
- 3 STX ABS
- 1 INX
- 2 STX ABS
- 3 STX ABS
- 1 INX
- 3 STX ABS
- 3 STX ABS

44

48 2 3

LDA
STA
STZ

LDX #2

drop x when not moving only
shoot in 4 directions

- CMP #50F
- BEQ Q61
- CMP #50E
- BEQ BUPQ
- CMP #50D
- BEQ BDNQ
- CMP #50B
- BEQ BLFQ
- CMP #507
- BEQ BRTQ
- BNE STQ

- CMP #50F
- BEQ Q61
- LDA STBUQ
- BNE STQ
- LDA NUMBLQ
- BEQ STQ
- LDA STIKQ
- CMP #50E
- BNE Q02
- LDA #2 ; UP
- STA STBUQ
- BNE CORDM0
- Q02 CMP #50D
- BNE Q03
- LDA #1 ; DOWN
- STA STBUQ
- BNE CORDM0
- Q03 CMP #50B
- BNE Q04
- LDA #502 ; LEFT
- BNE CORDM0 ; UNC
- Q04 CMP #507
- BNE STQ
- LDA #501 ; RIGHT
- BNE CORDM0
- CORDM0 STA STBUQ

- LDA (SCRL),Y
- STX (SCRL+2),Y
- STX (SCRL+4),Y
- DEX
- BPL
- RTS
- LDA SM
- STA SCRL
- LDA \$
- STA SCRL+2
- LDA \$
- STA SCRL+4
- LDA \$
- STA SCRL
- STA SCRL+2
- STA SCRL+4
- LDY #2
- JSR
- LDY #2

GOLD M. SRC

STARTM ~~A~~ ~~G~~

MAINM C

CANROUM ~~A~~ ~~B~~ D

BULM

MAINN ~~B~~

BAK L5 M

~~START~~ STARTN B

MAINM

CANROUM

BULM

STARTM

B

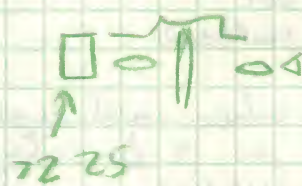
~~A~~ ~~G~~ E

D

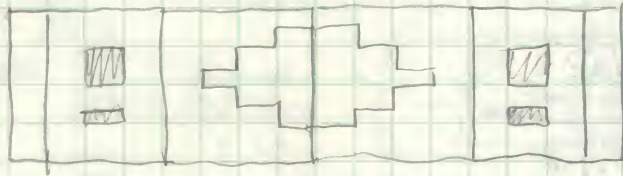


HEAD C9 2D 2L XMN

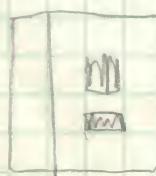
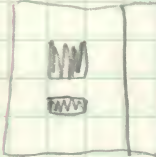
HEAD C9 CC



SANTES



bank
62 61 6E 6B



SOURCE FILE

7/1/82

MAINN SRC D

STARTN B

CANROUM F

STARTO A

BF

DB

BF

C3

BE

system confused
when trying to up date
GOLDRUSH
STARTM ~~A~~ F
STARTN C

6/28/82 to be done

sounds

start w/ bullets

expand bank sensitivity

2-color players

1st title

1-pl games 1 & 2 - trans

STARTN ~~defunct~~ ~~apl~~

GOLDRUSH OBJECT

STARTN B

MAINN A

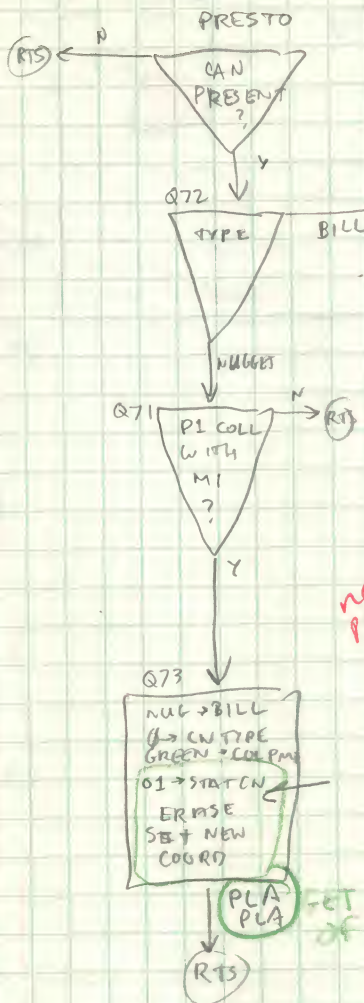
BULN A

7/1/82 ene.

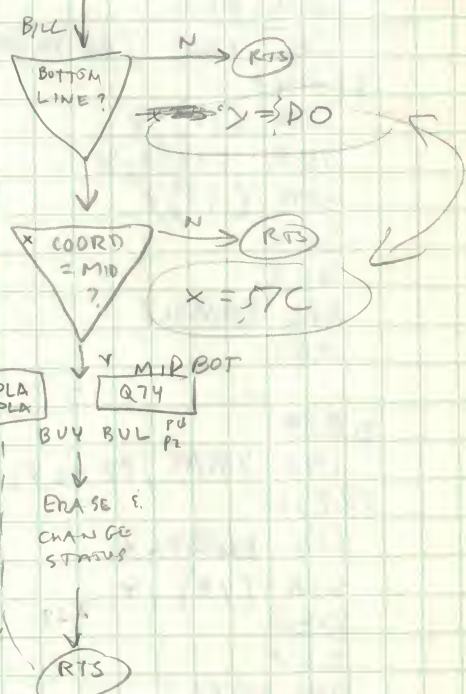
✓ 41F ~~EMOV2~~
✓ 43A8 ~~MOV2~~

5092 P06BUL

50AA P26BUL



new Presto sec p. 76



BKR 6435
A = 3 X = 0
Y = FF P = 35
S = F3

get out of Bul.

REDRMØ 631E 6308 } MAINA
REBRM2 643C 641E }
6418 }
V. C

7/1/82

408B JSR WIDENØ → 6542
427F JSR WIDEN2 → 656B
4289

408B 20 408C 10 63 JSR REDRMØ
4259 20 425A 2E 64 JSR REDRM2

WIDENØ 652A 6505
WIDEN2 6548 6523


```

DRW
LDY #0F
LDA MAN, X
STA (YP), Y
DEX
DEY
BPL

```

```

LDY #0F
LDA YBAS
CMP #P0BAS
BNE DRP3

```

DEC YBAS ; P0 TO M

```

DRM3LP
LDA (YP), Y
ORA FLESH, X
STA (YP), Y
DEX
DEY
BPL DRM3LP
RTS

```

```

DRP2
INC YBAS ; P2 TO P3
DRP3LP

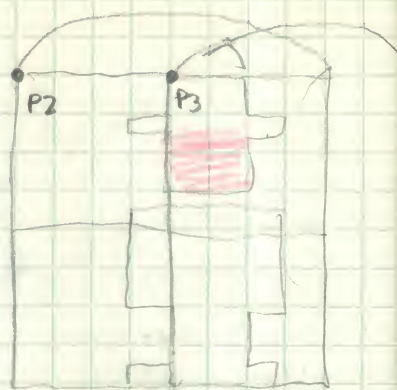
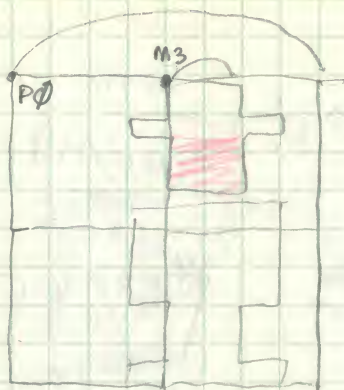
```

```

LDA FLESH, X
STA (YP), Y
DEX
DEY
BPL DRP3LP
RTS

```

FLESH = 3C00



STMOV0

```

Q25 LDA YP
STA YP0
LDA XP
STA XP0
STA HPOS P0
CLC
ADC #4
STA HPOS M3

```

W25

```

STA HPOS P2
CLC
APC #4
STA HPOS P3

```


ERASE LDY #50F
LDA #0
→ ERSLIN STA (YF),Y
DEY
BPL ERSLN
RTS

→ LDA YBAS LDY #50F
CMP #P0BAS
BNE ERS P3
DEC YBAS ; P2 to M3
~~LDA #0~~
ERM3LP
LDA (YF),Y
AND #\$3F ; M3=0
STA (YF),Y
DEY
BPL ERS M3
RTS

ERS P3
INC YBAS ; P2 to P3
~~LDY #0~~
LDA #0
ERP3LP
STA (YF),Y
DEY
BPL ERP3LP
RTS

GOLDM.SRC

STARTM G
MAINM C
CANROOM D
BULM
MAINN B

MAINO B (REN)
STARTO E

BAK LEM

STARTM B
MAINM B
CANROOM E
BULM D
STARTO B

MEM.SAV

GOLDRUSH - CAN'T WRITE
OVER - SYSTEM COMPUSED

STARTM F
STARTN C

GOLDRUSH OBJECT

STARTN B (messed up
line 2081)

MAINN A
BULN A
MANO

FULL

LINKS
P26BUL 52 F3
P26BUL 52 F3
• BAK 7/28
→ STARTO V.N
→ CLAIMA V.B
MAINO V.D
SMOON V.A
CANROOM V.F
BULN V.A
SOUNDB V.C

VERSION Q (REN)
CANROOM = M.M

CLAIMA = A.B

STARTO = 0.P

MAINQ = 0.D

CREATO = Kifen change aslo make

SMOON = SMOON

SOUNDB = B.P

7340
7950

7300

SOURCE FILE

MAINN D

STARTN B (messed up
line 2081)

CANROOM P

STARTO A

GOLDRUSH.3CL

SOUND DEV

MAINO C

DISK #1

MAINO D

SMOON A

STARTO D

GOLDRUSH.715

WIDEN0 6557

WIDEN2 6575

YMN=26

H DISK FULL
ERROR

DISK #2

CANROOM E

TEMP - OBS 715

STARTO F

STARTO M

SOUNDB A

DISK 3

CANROOM H

STARTO N

GOLDRUSH.715

CLAIMA B

SOUNDB B

CLAIMA A

GOLDRUSH

CANROOM E

DISK 4

SOUNDB C

STARTO E

CLAIMA A

GOLDRUSH

CANROOM E

DISK 5

MAINN A

DISK 6

MAINN B

forgot
shorten

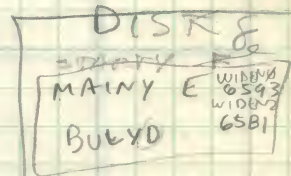
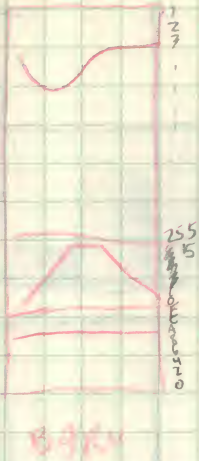
Attract

musical dropx

ending
1- player

colored house
speed up
shakes
tumbweed

random
houses + casti +
more
cack



ending
of houses

5-4-3
for speed

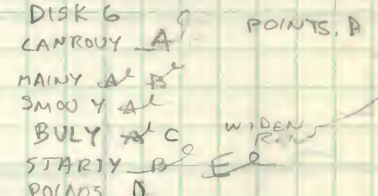
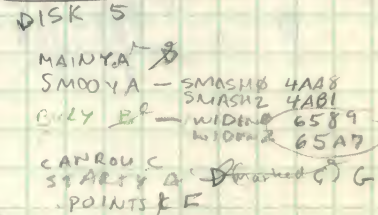
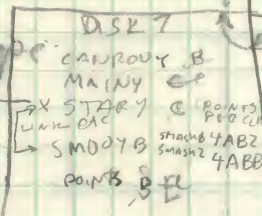
SOUND DEV

6000

XX
DUR

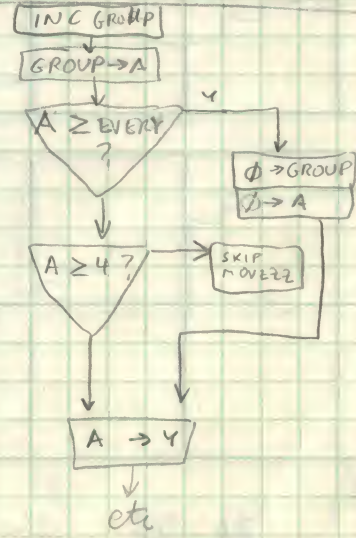
XX
AUD

6666G



MOVE Z
LDA FRAME
AND #3
ASL
ASL
TAY

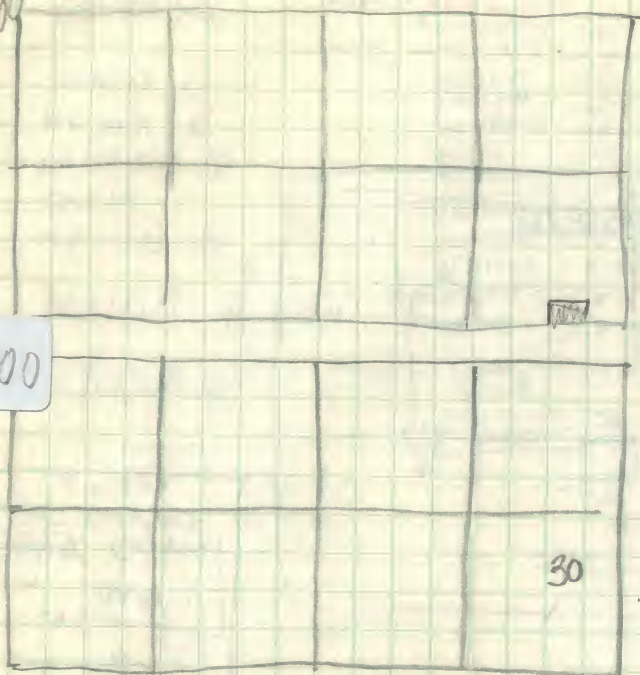
SPEED	20	25	33
EVERY	5	4	3
GROUP	0 000 1 001 2 010 3 011 4 100 5 101	000 001 010 011 100	000 001 010 011



WIDEN
713/12 6557 0
6575 2

4609

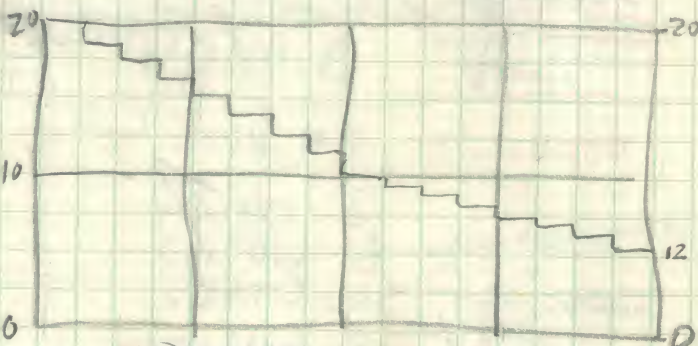
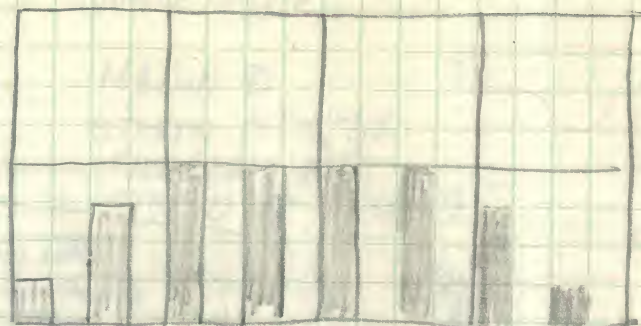
4700



V TYPE
1 0
3

00-0F
PITPAT

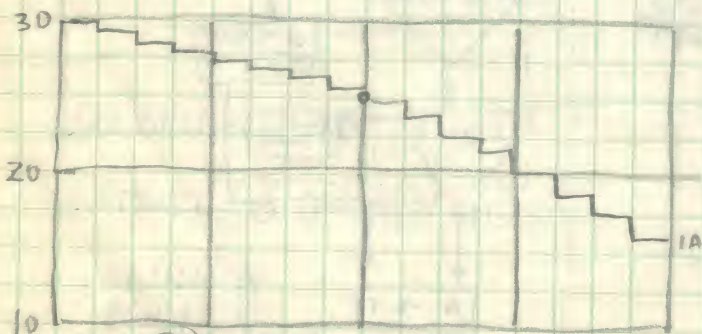
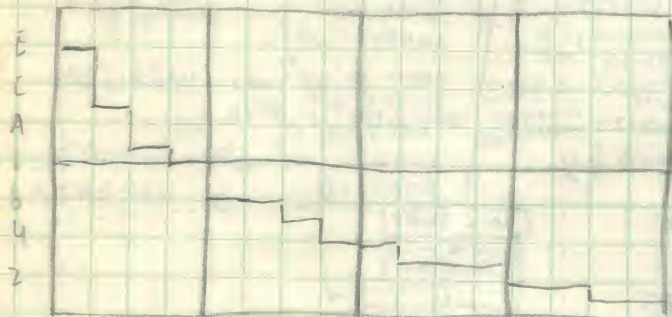
30



V TYPE
1 8
3 2

10-1F
FART

A buzz piano
y slow fart shuttle
C plucky strings

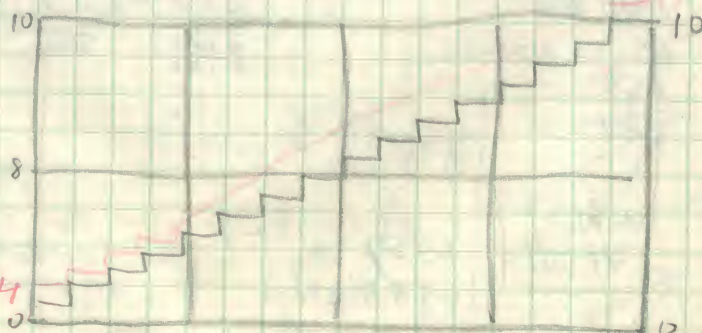
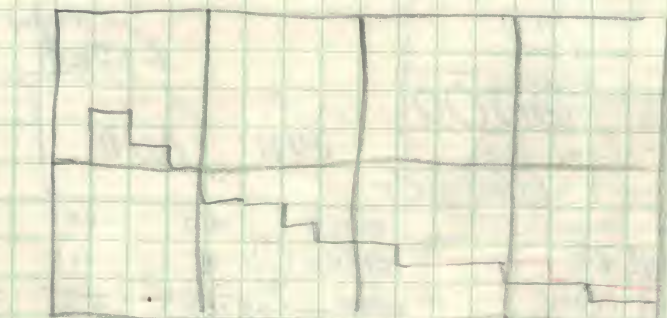


V TYPE
1 8
3

20-2F
SHOOT

TYPE A
PLUCK LONG
FUNNY

0.4
DRUM



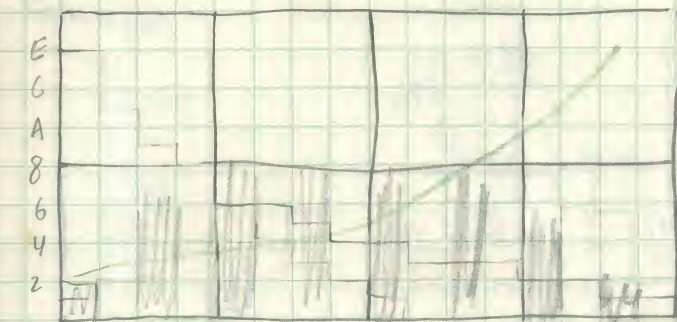
V TYPE
1 2
3

30-3F

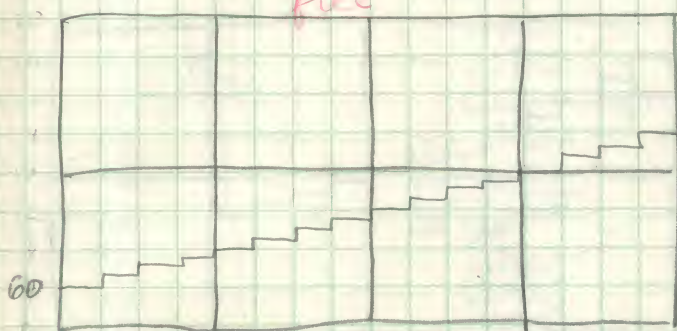
BERZERK

0 steam train
8 up hill

A click, bird chirp
G faint musical
berzerk



free

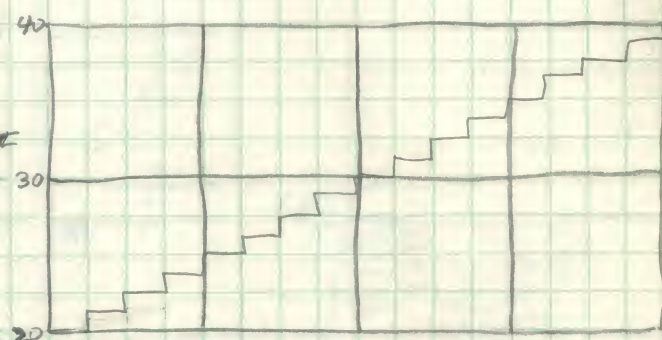
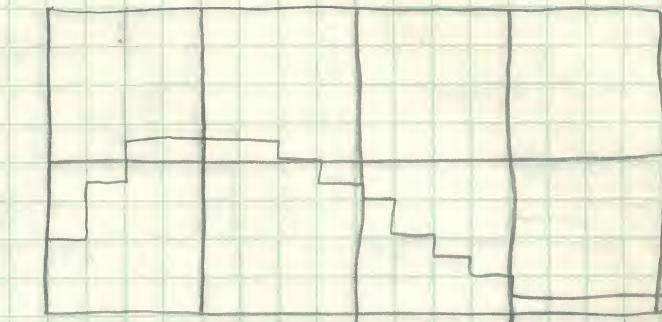


V
3

TYPE
2

40-4F
BERZERKZ
new
fant

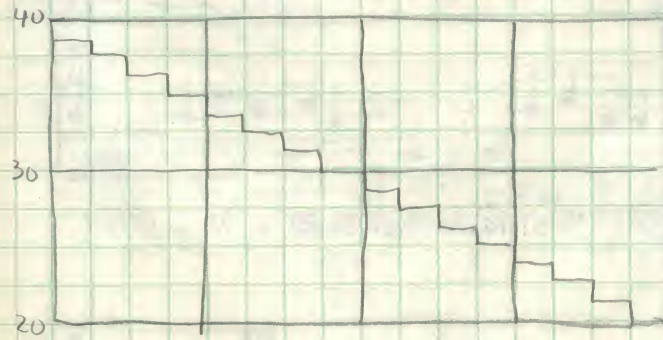
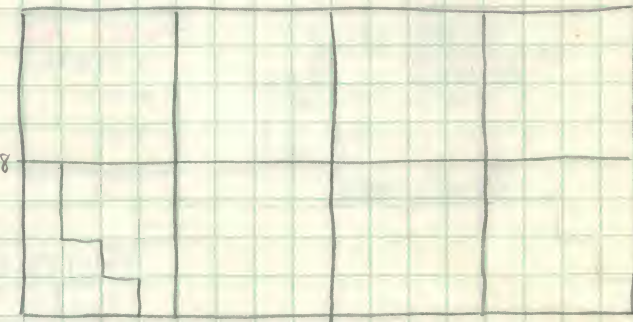
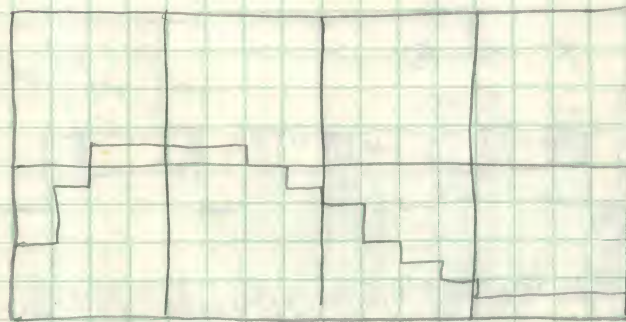
0 drum
4 berserk
8 hit track
can
A click,
chirp
G string
berserk



TYPE
A
(TONE)

50-5F
SNK → TUM

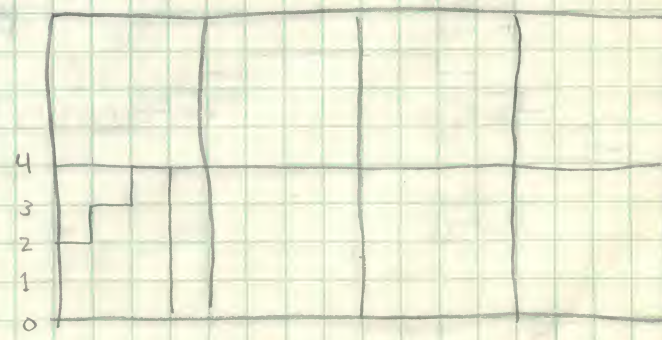
0 low stem
train
2 4 6



TYPE A
(TONE)

60-6F
TUM → SNK

others: train



0 foot stamp
8 brush on drum
G squeak

70-7F
EAT OWN X

(20)

4600

VOL

4700

freq
60

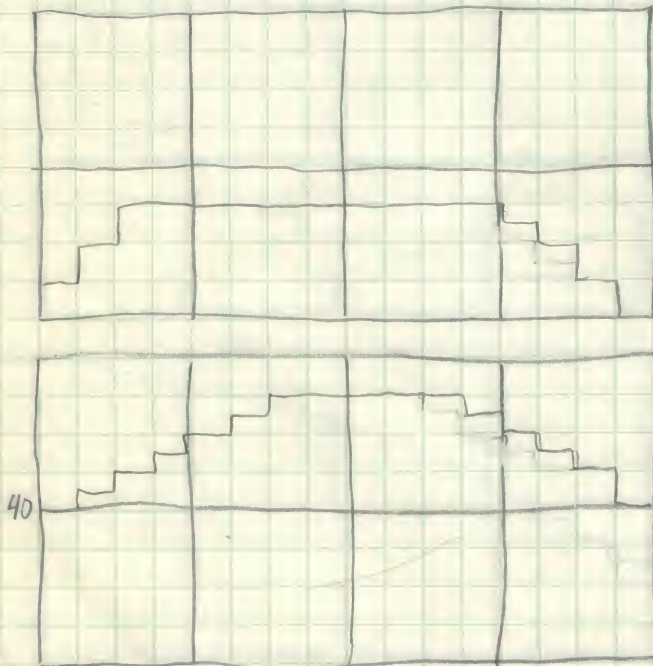
80 - 8 F

TYPE A
BUT PRESS
OUT OF X

5884

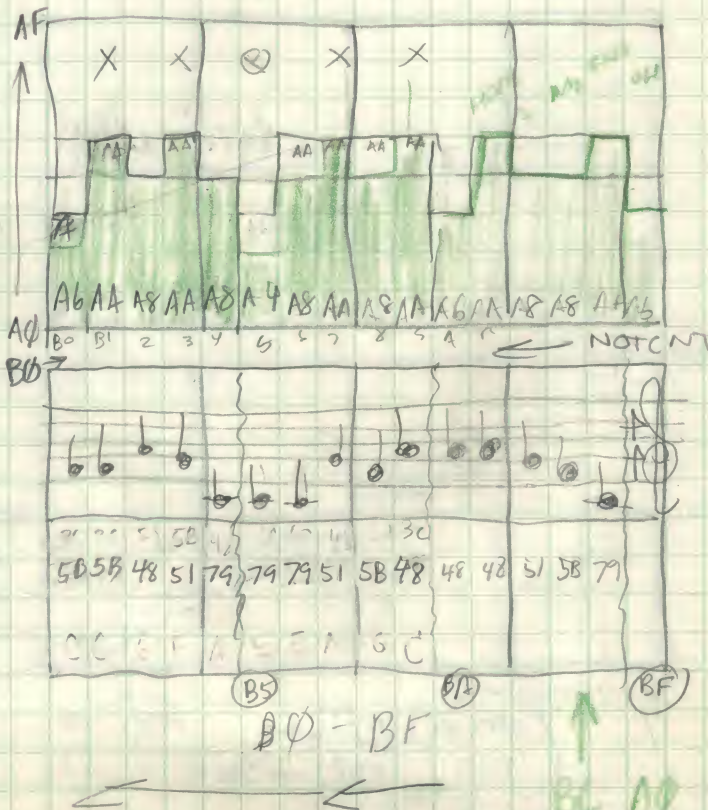
28
90 - 9 F (TYPE A)
"UH"

music



A0 - AF

TYPE A "OH"



B0 - BF

600 FRAME
601 SIGHT
602 XM
603 VM
604 TRG PFL
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

646 XPB
647 VPB
648 VOL1
649 MCNT1
650 VOL2
651 MCNT2
652 VOL3
653 MCNT3
654 VOL4
655 MCNT4
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

680 NUM90
681 CHANCE
682 DURI
683 AUDI
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

#1 6C1
2 6C2
3 6C3
4 6C4
5 6C5
6 6C6
7 6C7
1 6C9
2 6C0
3 6C1
4 6C2
5 6C3
6 6C4
7 6C5

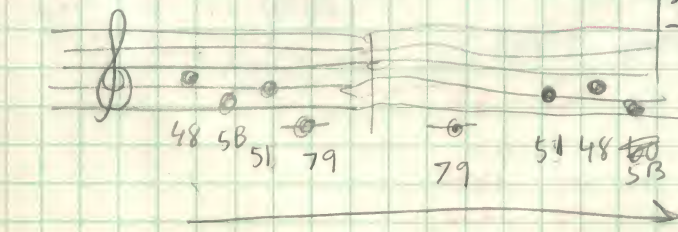
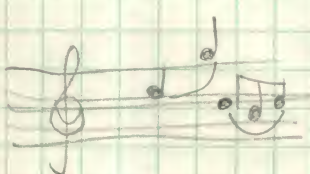
6C0 EMPTY
6C1
6C2
6C3
6C4
6C5
6C6
6C7
6C8
6C9
6D0
6D1
6D2
6D3
6D4
6D5
6D6
6D7
6D8
6D9
6E0
6E1
6E2
6E3
6E4
6E5
6E6
6E7
6E8
6E9
6F0
6F1
6F2
6F3
6F4
6F5
6F6
6F7
6F8
6F9
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

IMMUNO
STVK
STTEMP
VCNSMX
VCNMX
FFLAG
GROUP
EVERY
GROUP
XCOUNT
XFREE
SUBSNK
SNKIDX
WINCNT
WINNER
TEST
STATION
CANCRI
SCORE
CXPL
CYMI
CYPL
CYMI
CNTYPE
STBUL
XMO
XMO
XMI
XMI

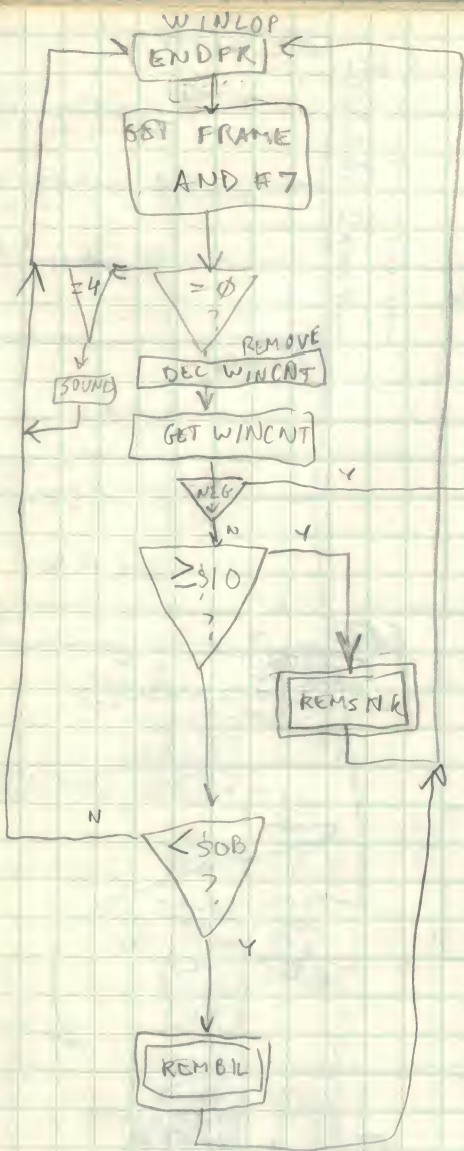
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

6F0
6F1
6F2
6F3
6F4
6F5
6F6
6F7
6F8
6F9
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800



25,000
15,000
0000
10 -
100 +
10,000 +
79 58 51 79
79 58 51 48
79 58 51 48

T 3
+ 2
X 1



JUMP TO WINLP

W/ WINCNT = 1

WINNER = 0 or 2

BUYHOU
LDX WINNER
BEQ GOBUY
LDX #6
GOBUY JSR PLHAUS
RTS ; OUT OF BOTES?
JMP!

WINLOP
JSR ENDFR
LDA FRAME
AND #7
BEQ REMOVE
CMP #4
BNE WINLOP
LDA #0
STA AUDCH
BEQ WINLOP ; UNC
REMOVE
DEC WINCNT
LDA WINCNT
BMI BUYHOU
CMP #510
BCS REMSNK ; BAGE
CMP #50B
BCS REMBIL ; BAGE
REMBIL NOP
AND #50F
LDX WINNER
BNE PZBIL
TAY
LDA #7
STA #7
STA #7
BNE SNBANG ; UNC
JWINLP JMP WINLOP

EDR #50F
TAY
LDA #7
STA #50F ; Y
BNE GRAY

REMSNK NOP
AND #50F ; WINCNT
TAX
LDA SCRL ; X
STA SCRL
LDA SCRLH ; X
STA SCRLH
LDA DIR20 ; X
STA DIR2F
LDX #2
JSR FINDSC
LDA #0
TAX
STA (SCRL, X)
STA (SCRL+2, X)

SNBANG
LDA #5
STA AUDCH
LDA #5
STA AUDFH
JMP WINLOP

1018

A	→	5
9	→	6
8	→	7
7	→	8
6	→	9
5	→	A
4	→	B
3	→	C
2	→	D
1	→	E
0	→	F

SCRL20

WINNER
WINCNT

5CDB

WINTIME

WINLOP

LET WINNER'S SCORE = 0 + BONUS

ERASE WINNING
PLAYER'S MONEY

STOP ACTION
PLAY MUSIC

ERASE SNK, TUM
W/ SOUND
1-by-1

ERASE WINNER'S
MONEY 1-by-1
W/ SOUND

LDX
JSR PLHAUS

INC NUM HOU
IF NUM HOU = 2,
NO END

GO BACK
TO MAIN
LOOP

NO STRTES!
NO PTES!

WIN CNT

1/8 sec

\$ IF

shake

IE
ID
IC
IB
IA
I9
I8
I7
I6
I5
I4
I3
I2
I1
I0

1 sec

nothing

I5
I4

every 8 frames

IF
IE
ID
IC
IB
IA
I9
I8
I7
I6
I5
I4
I3
I2
I1
I0

snk A
snk

9
8
7
6
5
4
3
2
1
0

1/8 sec

1/2 sec

1/8 sec

BILL
BILL 2

IDLE

10

XPO C1 FO

YPO C2 F1

XPI C3 F2

YPI C4 F3

XP2 C5 F4

YP2 C6 F5

XP3 C7 F6

YP3 C8 F7

XP4 C9

YP4 CA

YP CB

YBAS CC

F6

F7

SCDATA D4

F9

XP CF

F8

SCRL D6

FA

SCR4 D7

FB

+2 D8

FC

+2 D9

FD

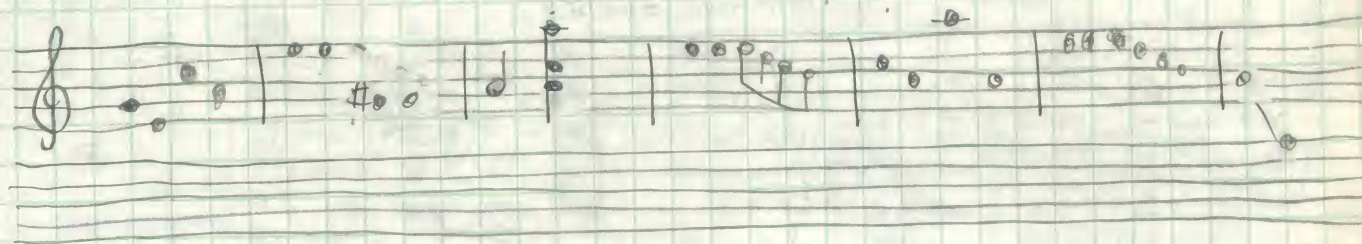
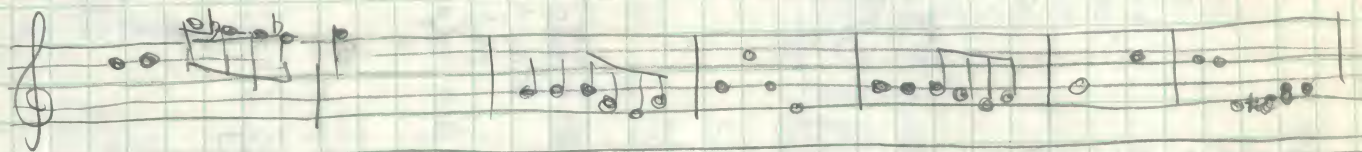
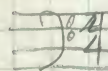
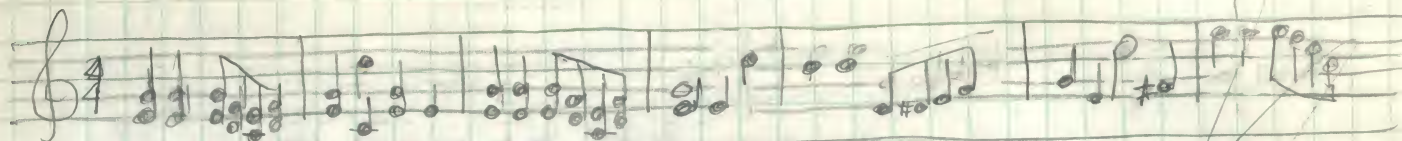
+1 DA

FE

+1 DB

FF

TMP SCRL NUM
DMP \$3A00



frequency of random
pink X

```

START
LFF → XFREQ
EO → XCOUNT

LDA RANDOM
AND XFREQ
BNE
LDR #0
JIR RANDSC
DRA (CRL, X)
BNE
LDR #2
STA (SCL, X)

SOUND

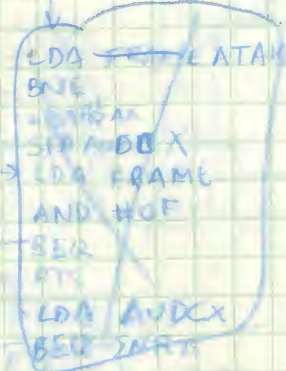
INC XCOUNT
LDA XCOUNT
SEC
SBC #SA
BNE
STA XCOUNT
LDA XFREQ
LIR
STB XFREQ
    
```

SYNAPSE SOFTWARE
RAY CHANG
BY PRESS START
1982

SOUND



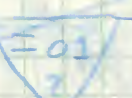
STANDARD



FRAME AND IF



COUNT AND 7

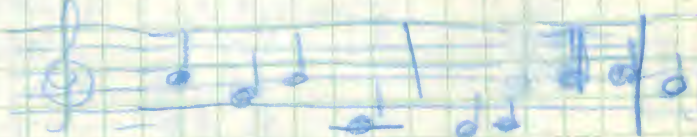


COUNT AND 7



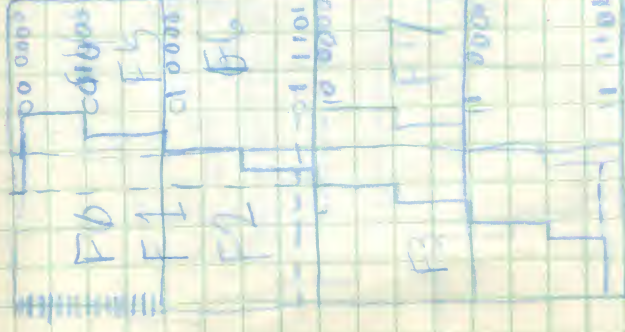
ALTF

$Q = 32$ for $H = 64$ Hz



	CH	OCTAVE	DURATION
C	0	06	00 H
D	2	01	01 Q
E	3	10	10 E
F	4	11	11 S
F#	6		
G	7		
G#	9		
A	9		
A#	A		
B	B		

List of notes
0 = end of 1/5



MCNT4
MCNT3

NOTES

BO
B1
B2
B3
B4

47B0
47B1
47B2
47B3
47B4

46B0
46B1
46B2
46B3
46B4

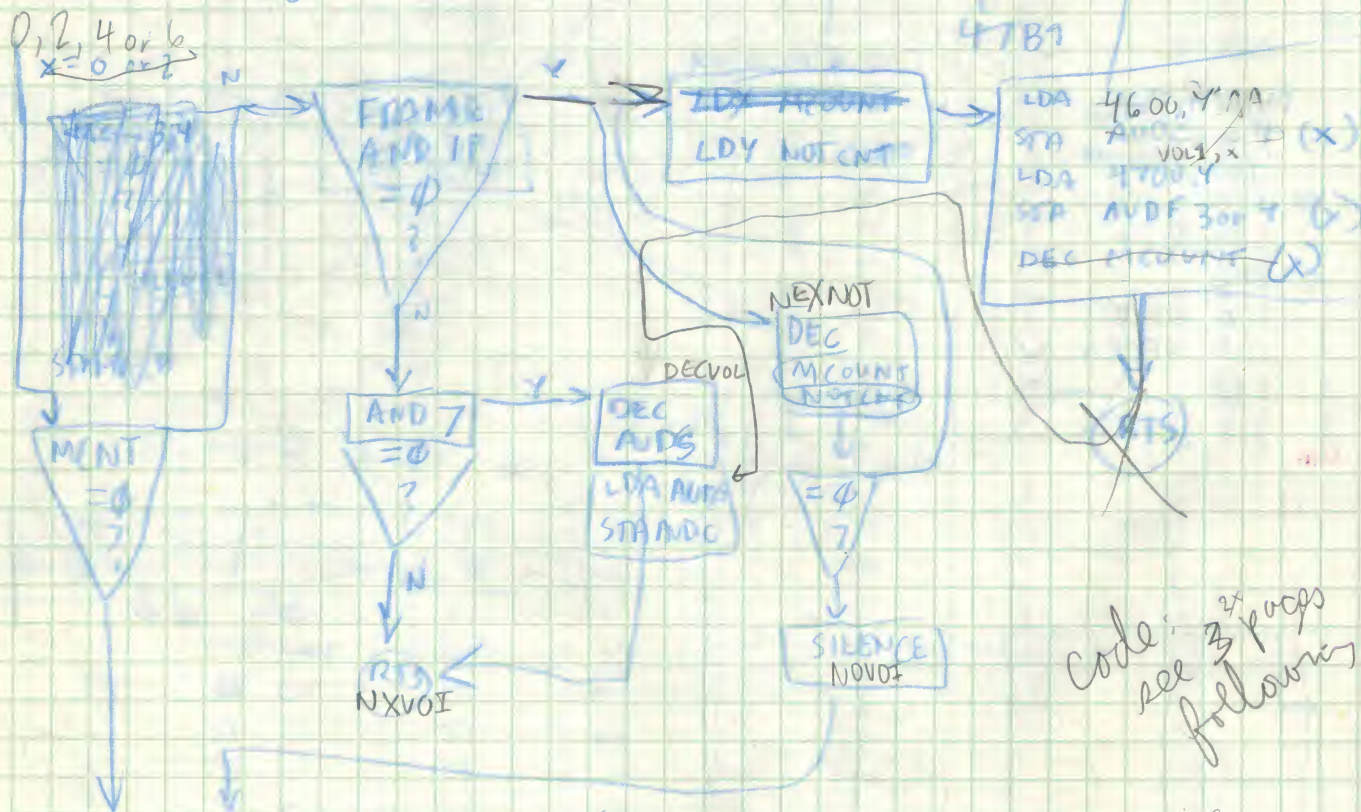
BS
B6
B7
B8
B9

E
E
A
G
C

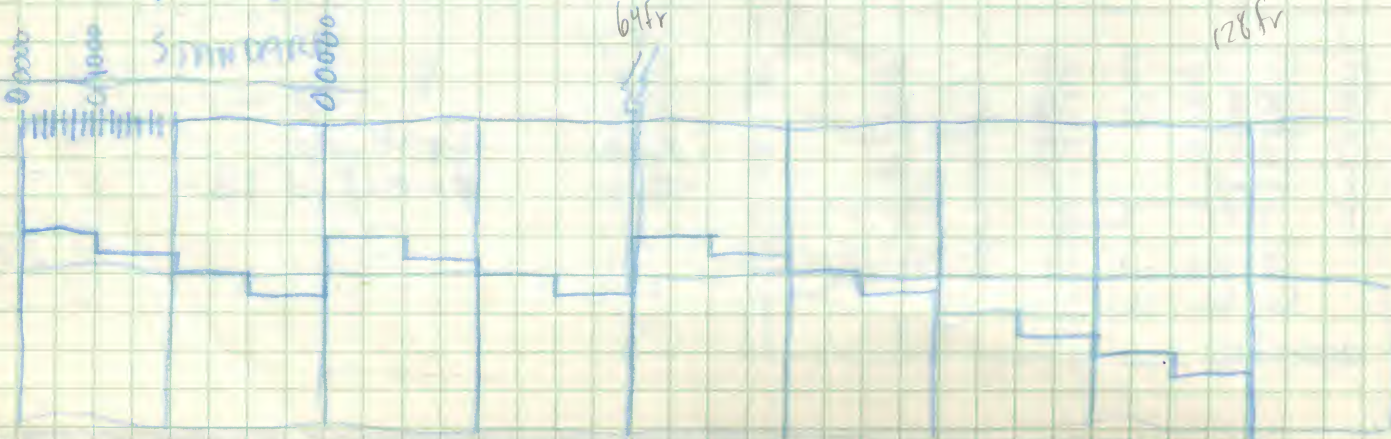
AG
AA
AA
AA
AA

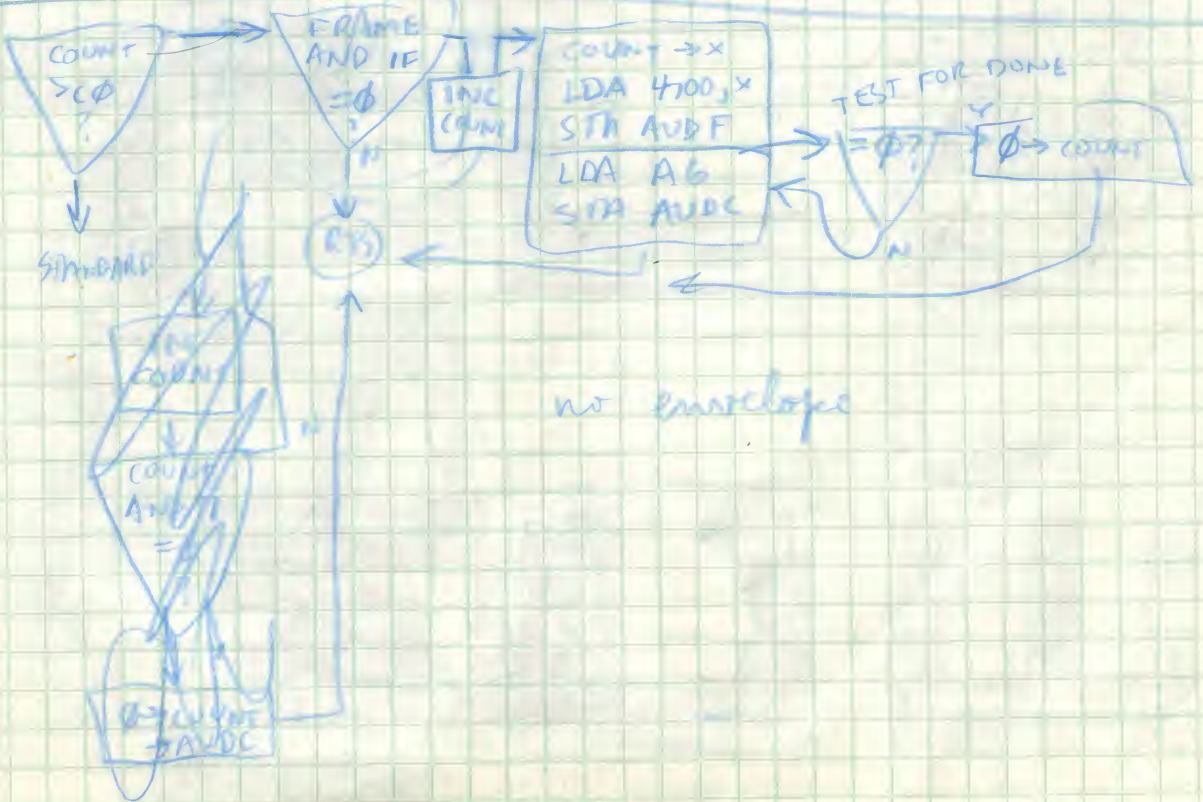
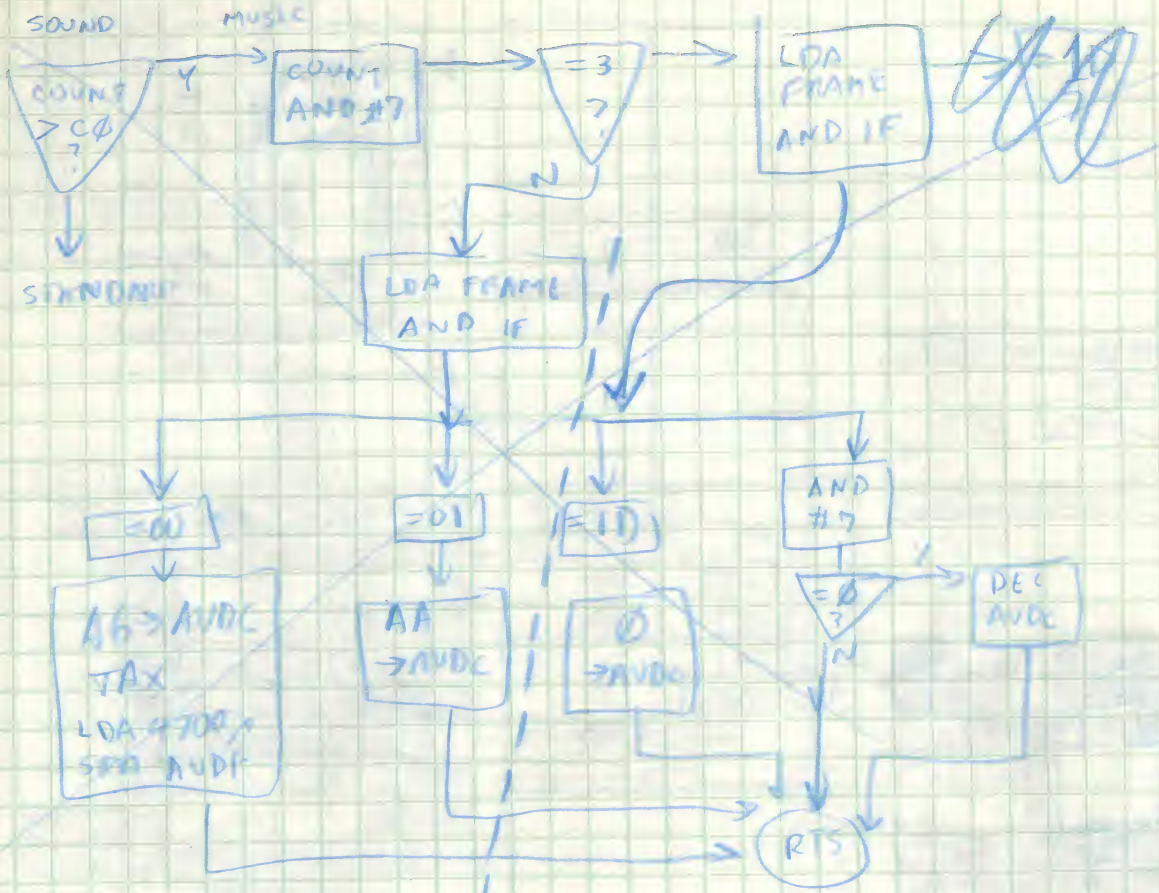
46B5
46B6
46B7
46B8
46B9

START MUSIC $\frac{B9 \rightarrow \text{NOTENT}}{05 \rightarrow \text{MCOUNT}}$

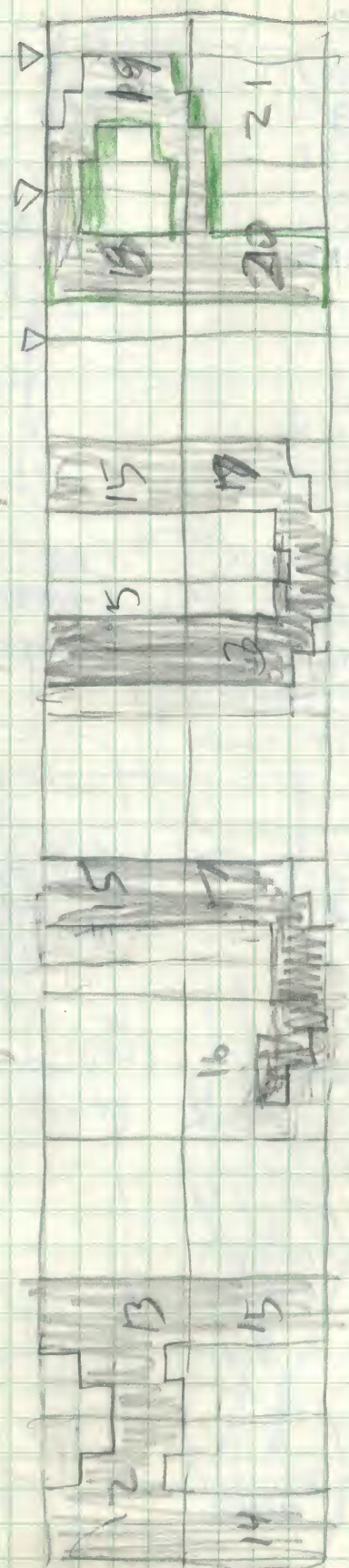
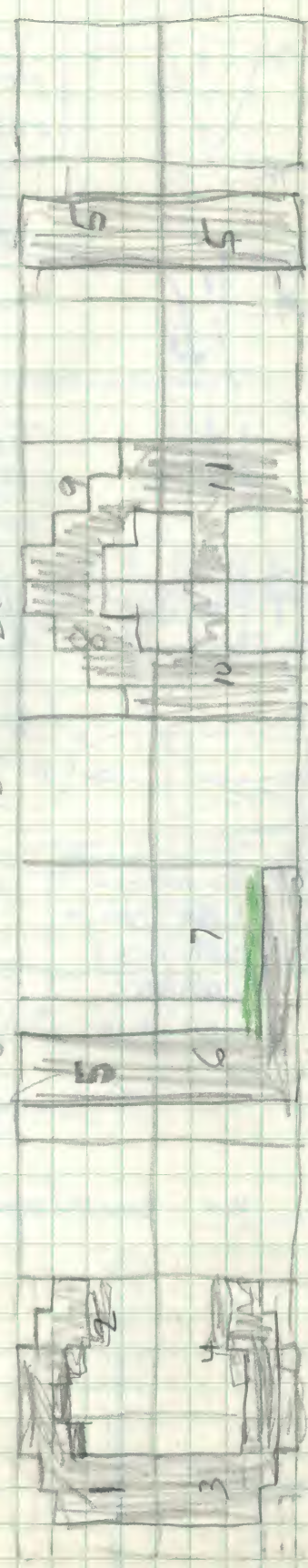
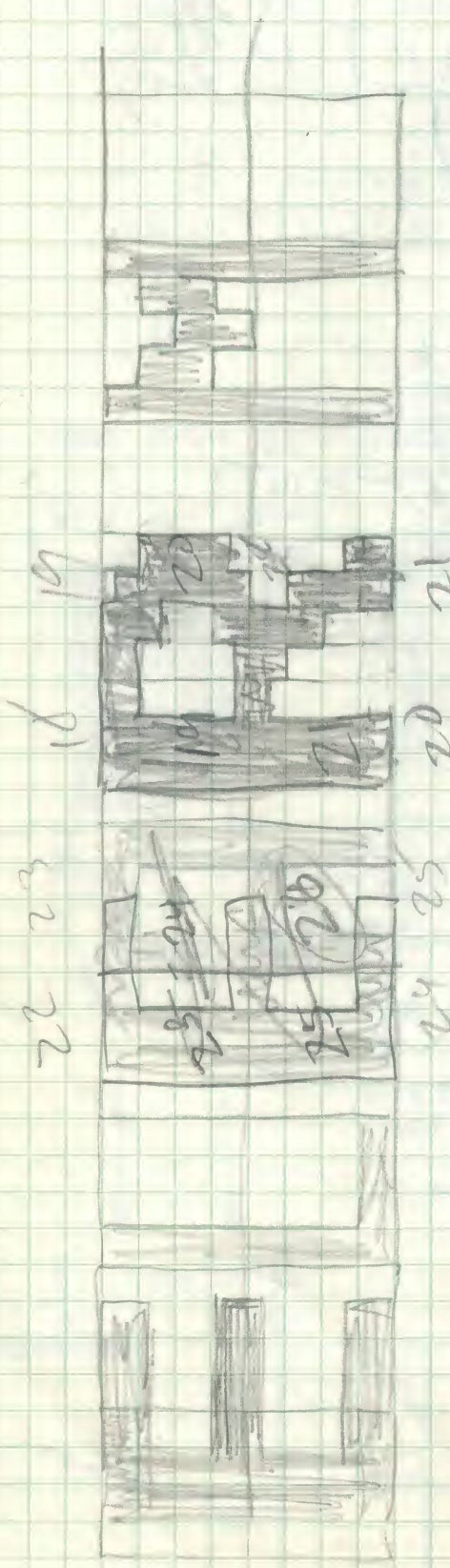


code: 3×10^{24} pops
see following





no envelope



Typing monitor command

5300, 53FF in edit mode
screens up source file!

TEST FOR EOL WHEN PLACING HOUSE

```
LDA CACPOS,Y
STA SCRL
LDA CACPOS+8,Y
STA SCR H
```

```
JSR TESOC
JSR ROW
JSR TESOC
```

```
LDY #2
L1 LDA (SCRL),Y
CMP #5
BCS FNLP
DEY
BPL L1
```

```
LDA SCR H
STA SCR H+2
LDA #2
CLC
ADC SCRL
STA SCRL+2
BCC HI OK
LDA SCR H
STA SCR H+2
INC SCR H+2
HI OK LDY #2
L2
```

```
TESOC LDY #2
OCLP LDA (SCRL),Y
CMP #5
BCS
```

TO SOLVE PROBLE
TEST FOR HERE

THEN IF HERE
SIMPLY SET B&P → 0



7/23 Things to do:

START

~~music after placing house~~ ←

MAIN

~~speedup (easy)~~

CREAT

SMOO

~~hit RANDSC~~

CANROU

~~start music w/ convert saved~~ ←
~~flashing missile destinations~~

SOUND

~~uh-oh capability~~

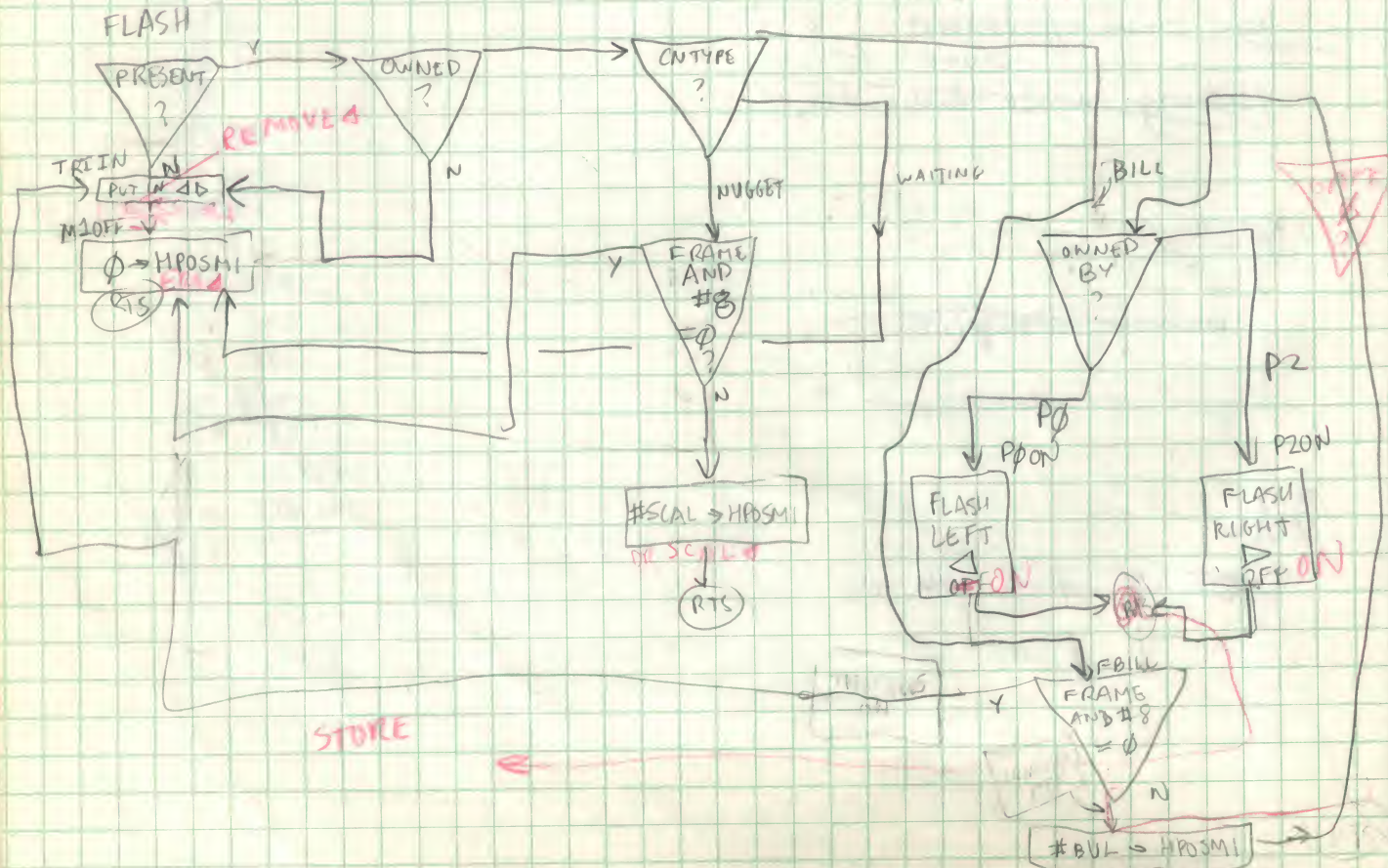
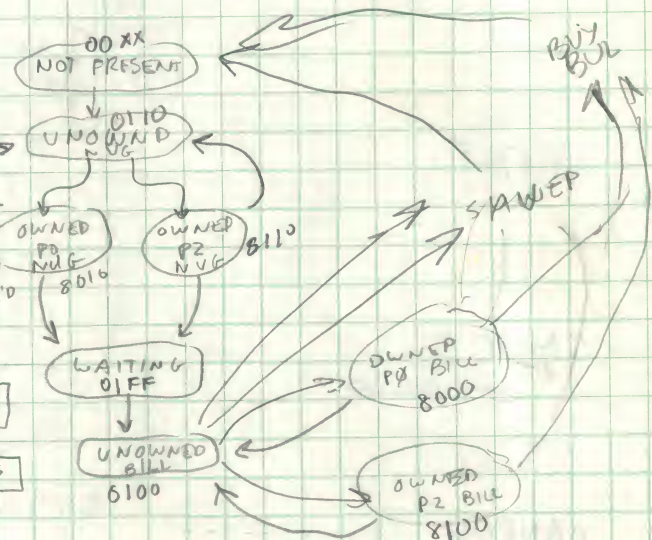
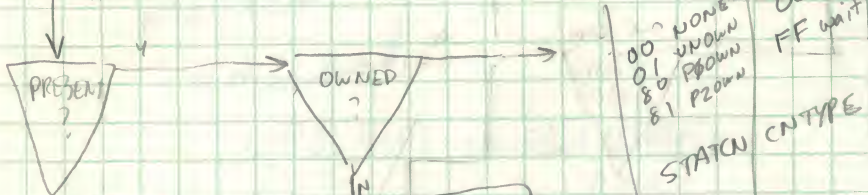
~~music capability~~

BUL

~~gun hit sound uh-oh~~

25 0 0 0
50 0 0 0

CANROV
FLASH




```
DEC NOTCN1,X
DEC MCNT1,X
BEQ NOV01
LDY NOTCN,X
LDA $4600,Y
STA VOL1,X
LDA $4700,Y
STA AVDF1,X
JMP STDVOL
```

EQ

NOTCN	1	B8B	B9 - B8 - B7 - B6 - B5
	2	A	
	3	E	
	4	(BE)	

EQ

MCNT1 = 64(6) - 5 - 4 - 3 - 2 - 1 - (0)

$$MCNT4 = 64F$$

EQ

VOL 1 = 648 ^AA987 ^AA987 ^AA987 ^AA987 ^A6543

2 64A

3 64C

4 = 64E

to initiate dataset

46 B0 - 46 BF
47 B0 - 47 BF

HUMB

10

7

BZ

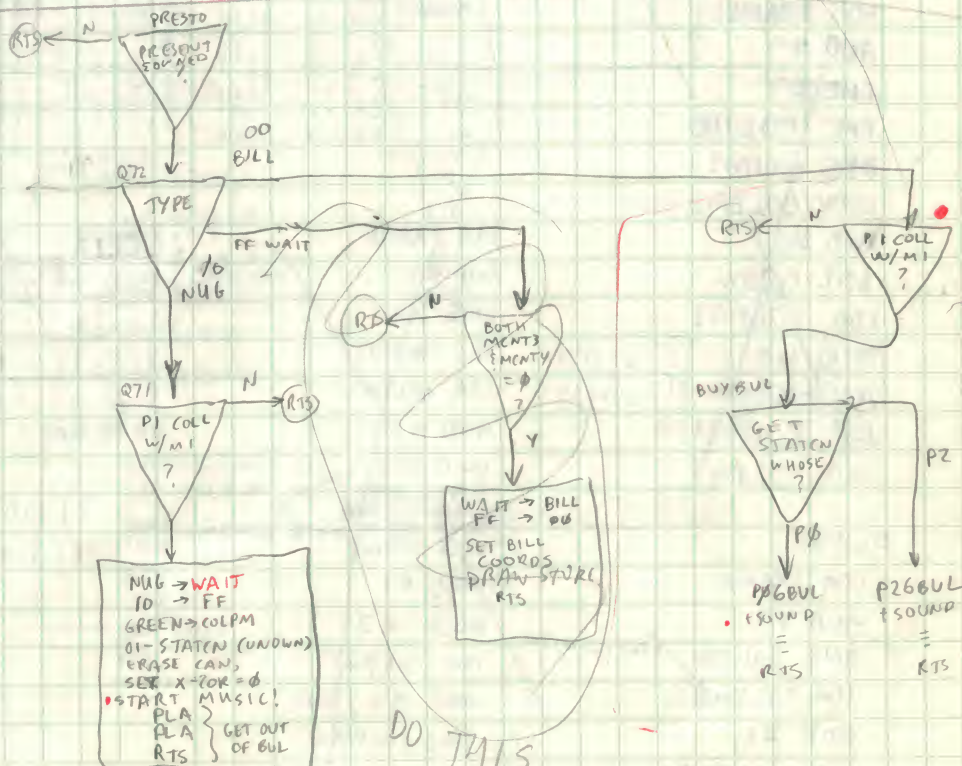
BA $\rightarrow$ NOT(N4) = ~~66~~6BE
6 $\rightarrow$ MCNT4 = 64F

$$6 \rightarrow MCNT4 = 64F$$

MUSIC

FLASH

LDA STATCH
 BEQ TRIIN ²
 BPL TRIIN
 LDA CNTYPE
 BEQ FBILL
 CMP #SF
 BEQ M1OFF ; WAITING
 LDA FRAME ; NUBBER
 AND #8
 BEQ M1OFF
 LDA # ; SCALE
 STA HPOS M1
 RTS
 FBILL
 LDA FRAME
 AND #8
 BEQ TRIIN
 LDA # ; STORE
 STA HPOS M1
 LDX #7 ; SCALE
 LDA STATCH / 80 OR 81
 CMP #81
 BEQ FP2
 STX
 RTS
 FP2 STX
 RTS
 TRIPOS
 TRI2 POS



DO THIS
EARLIER
NOT IN PRESTO
(N 11 UNKNOWN)

7/28/82 things to do

ATTRACT ✓ ENDFR

✗ Pulsating nugget & bill ✓

title - snakes appear fast, make 16 + full in gaps ✓ STARS

✗ FIFO X'S ✓

✗ SNK + TUM instead of S and T ✓

✗ limited bullets an option

SCORING!

~~MAIN - allow x on bot line~~ ✓

ATTRACT

LDA \$D300 ; STIX

EOR #\$FF

BEQ UNTUCH

LDA #0
STA \$4D ; ATTRACT

UNTUCH

~~PULSAT (CANROW)~~

~~LDA FRAME~~

~~AND #7~~

~~CMP #7~~

~~BNE COLOK~~

~~DEC COLPM1~~

~~LDA COLPM1~~

~~AND #8~~

~~BNE COLOK~~

~~LDA COLPM1~~

~~EOR #8~~

~~AND #8~~

~~EOR COLPM1~~

~~STA COLPM1~~

AND #0
OR #08

FLASH

LDA FRAME

AND #3

BNE COLOK

LDA COLPM1

EOR #2

STA COLPM1

COLOK

LDA STATEN

~~LDA FRAME~~

~~AND #3~~

~~BNE X1~~

~~DEC COLPM1~~

~~LDA COLPM1~~

~~AND #8~~

~~CMP #8~~

~~BNE X4~~

~~LDA COLPM1~~

~~AND #5 OF 7 STA COLPM1~~

~~X4 LDA FRAME~~

~~X1 AND #1~~

~~BEQ X2~~

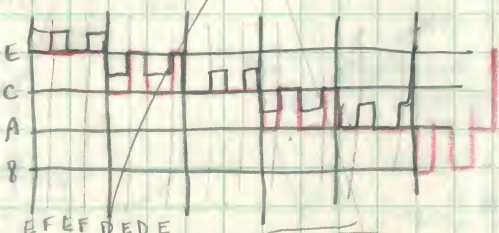
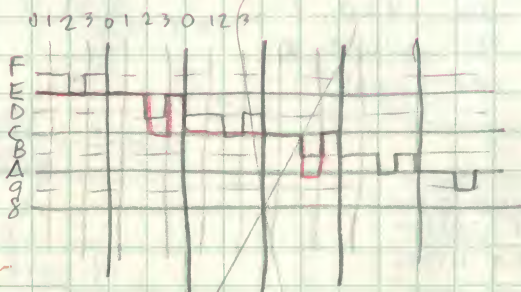
~~INC COLPM1~~

~~BNE X3 ; INC~~

~~X2 DEC COLPM1~~

~~X3~~

1111	1111	
COLOR	CF +7	1111
	CD +6	1110
	CB +5	1101
	CA +4	1100
	CG +3	1011
	CA +2	1010
	CG +1	1001
	CG +0	1000



1100
1110

FLASH

LDA FRAME

AND #3

BNE COLOK

LDA COLPM1

LDX STATEN

BPL OWN COL

EOR #2

BNE STCOL, INC

OWN COL

ORA #2

STCOL

STA COLPM1

COLOK

FIFO X'S

in MAIN module,
place coordinates
in 6C0, X } X = NUMBT
6C8, X } 1-2-3-4-5-6-7

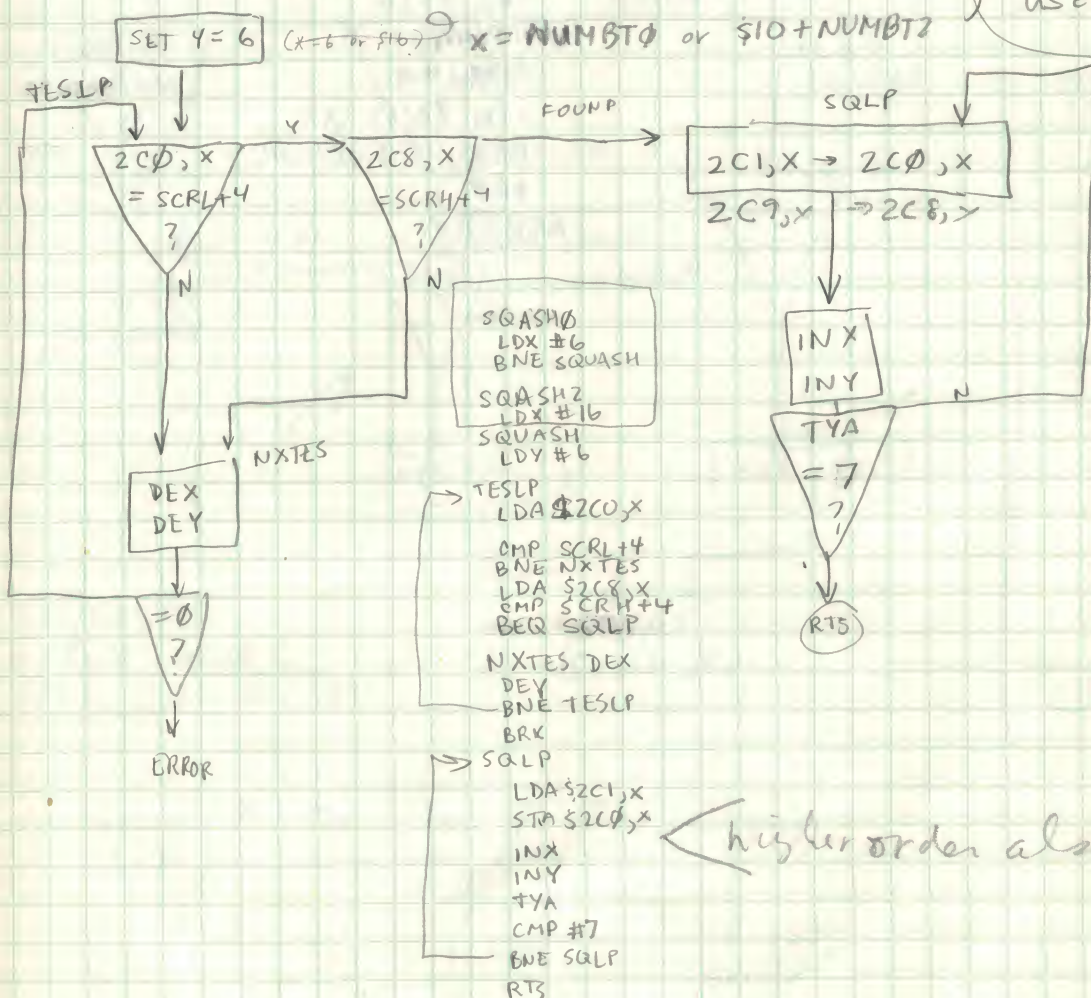
in MOVEZZ module
each time an X eaten,
search/find in stack
and compress. ~~but first~~
check for seven X's.
If so, remove X #1 and
compress down so that
there are six only - delete from screen

INITIALIZE X
TO TOP OF SEARCH
AREA X = 00 or 16

JSR SQUASH
DEC NUMBT ← after!

↓ SUBROUTINE
SQUASH

IN GAME 1
use 5 not +



LDA GAMEND

BNE ASBEF

TAX

LDA (SCRL,X) ; HEAD

NOP

CMP #\$2C

BCS TESTUM

LDA BHITFL

BNE ASBEF

CLC

ADC #\$20

BNE ENCONV "11C

TESTUM

LDX BHITFL

BEQ ASBEF

SEC

SBC #\$20

ENCONV

LDX #0

STA (SCRL,X)

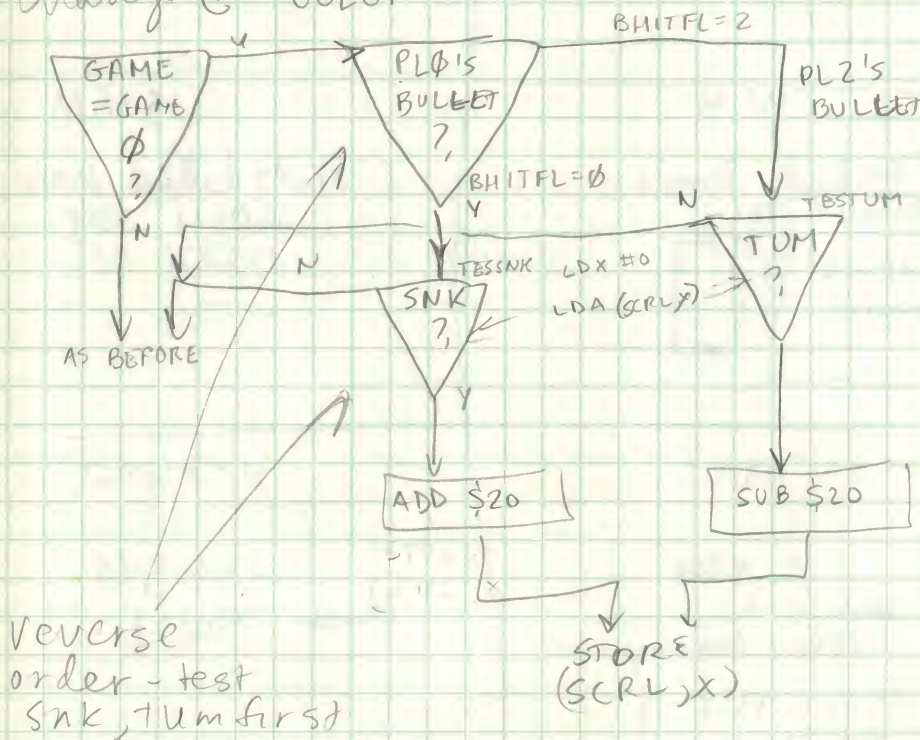
BEQ HITDUN ; UNC

BRK

ASBEF

bullet hit SKK - TUM	GAME 0	GAME 1
PL0 bullet hits TUM	$O \rightarrow T$	$O \rightarrow T$
PL0 bullet hits SNK	$\sqcap \rightarrow O$ convert sound	$\sqcap \rightarrow T$
PL2 bullet hit TUM	$O \rightarrow \sqcap$ convert sound	X
PL2 bullet hits SNK	$\sqcap \rightarrow S$	X

change @ COLOK



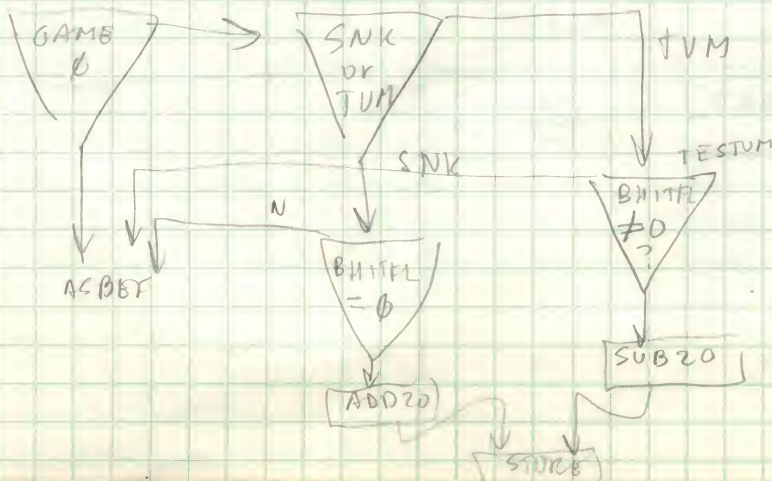
Reverse
order - test
SNK, TUM first

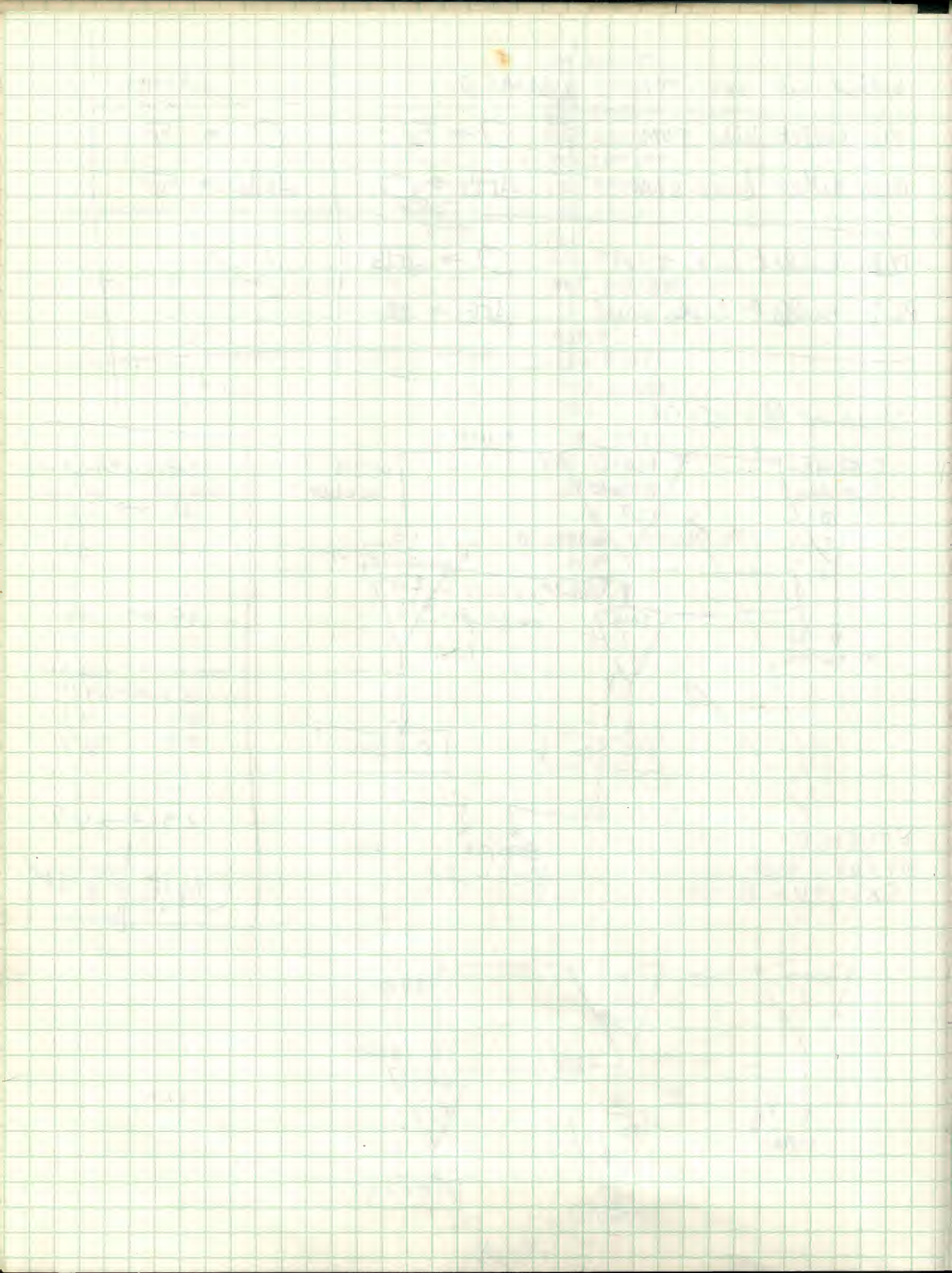
conversion

0001 xxxx	↔	0011 xxxx
10	↔	30
11	↔	31
12	↔	32
...		...
19	↔	39

0010 xxxxy	↔	0100 xxxxy
20	↔	40
21	↔	41
22	↔	42
...		...
29	↔	49

SNK | TUM

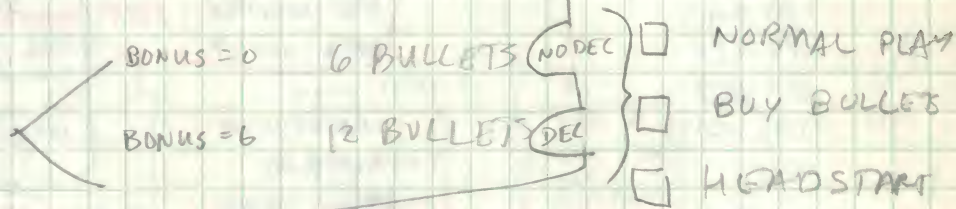




80-8F {debugging - monitor
90-9F}

BULLET HANDICAP JAZZ

- ☐ NORMAL PLAY
- ☐ 12 BULLETS
- ☐ HEAD START

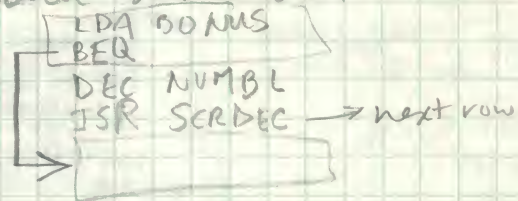


MAIN

each bullet shot,
DEC NUMBL
JSR SCRDEC

MAIN

each bullet shot



CANROW

BUY BUL
6 + BONUS → NUMBL

CANROW

BUY BUL
LDA BONUS
BEQ NOBUY

BUY EXACTLY 10 BUL
PLACE ON NEXT ROW

easy way
use FLASH
to put MI
or not put!
easy!

PUT BULLETS
ON NEXT
ROW

START

BONUS0
BNE
BONUS2
BNE
?

DRAW
BULLET
STORE

POINTS

```

LDA #5F
STA SCRL
LDA #DE
STA SCRL
LDX #0
JSR TESTDIS
IF WINLEY
LDA #5F5
STA SCRL
LDX #52
JSR TESTDIS
RTS

```

TESTDIS

```

LDA DIGH0
SEC
SBC #515
SBC WINLEY
BMI SEPAR

```

```

LDA FRAME
AND #40
BNE SEPAR

```

```

LDY #4
ZOUT STA (SCRL), Y
DEY
BPL ZOUT

```

```

SEPAR
LDA #0
LDY HFLAG0
BNE CHECK50
LDY HFLAG0+1
BNE CHECK50
LDA DIGH0, X
CHECK50
SEC
CLC
ADC DIGH0, X
CLD

```

```

LDA #0
LDA HFLAG0
ORA HFLAG0+2
BNE CHECK50
LDA DIGH0, X
CHECK50
SEC
CLC
ADC DIGH0, X
CLD

```

MAIN LOOP

ENTES
POINTS
TPOINT

IN "POINTS"

```

LDX #0
JSR TPOINT
LDX #2
JSR TPOINT

```

TPOINT

POINT WIN ROUTINE

```

LDA DIGH0, X
LDY HFLAG0
BNE CHECK50
SEC
CLC
ADC DIGH0, X
CLD
CHECK50

```

```

CMP #550
BCS PTWIN

```

RTS

PTWIN

PLA

PLA

PLA

PLA

LDA HFLAG0, X

BEG INFLAG

JMP ENLOOP

INFLAG

INC HFLAG, X

ENLOOP

JSR ENDR

JSR STRS

JSR OPITES

JSR SOUND

LDA HFLAG0

ORA HFLAG0+2

LSR

BEG ENLOOP

SELECT BUTTON TEST HERE

RTS -> MAIN LOOP

25000 ← Double
50000 ← Left for
750

ENLOOP

```

PLA
ENDR
STRS
OPITES
SOUND
FLWIN

```

STOPLP

```

ENDR
STRS
OPITES
SOUND
FLWIN

```

SELTES

PLA
PLA
RTS

```

LDA HFLAG0
ORA HFLAG0+2
LSR
BNE
JSR SELTES

```

```

LDA HFLAG
ORA HFLAG+2
BEQ ENLOOP

```

HFLAG

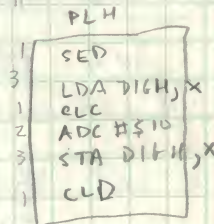
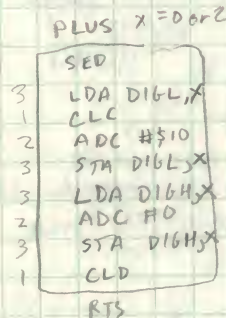
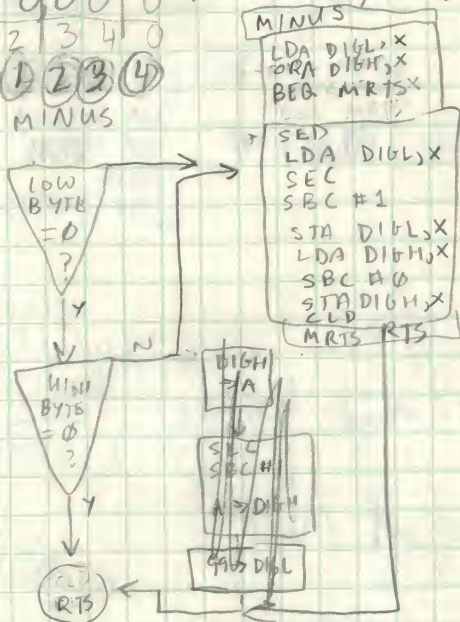
=0 25000 win
=1 50000 win
=2 no more (75000)

→

SCORING JAZZ (POINTS)

H	L
25,000	
15,000	
	0

1 0 - MINUS shoot or stuck
 1 0 0 +PLUS convert 1/2 x or shot SMOO & BUL
 1 0 0 0 PLH buy house
 Y = 0 1 2 3 4 MINUS



X = 0
 OR
 2

SCRL } screen loc
 SCRH }
~~SCRH+2~~ SEP
~~SCRH+2~~

Y = place 1-10-100-1000
 X = digit 0-1-2-3-4-5-6-7-8-9

IN ENTER TEST AND DISP SCORE

SEPAR
 LDA D1GH,0,X ; X=0
 LSR ; X=2
 LSR
 STA SEP+0
 LDA D1GH,0,X
 AND #5F
 STA SEP+1
 LDA D1GL,0,X
 LSR
 LSR
 LSR
 STA SEP+2
 LDA D1GL,0,X
 AND #5F
 STA SEP+3
 RTS

DISP
 LDY #0
 LDA SEP,Y
 BNE DESPLP
 INY
 CPY #4
 BNE TOLP

DESPLP
 LDA SEP,Y
 TAX
 CLC
 ADC #5
 STA (SCRL),Y
 INY
 CPY #5
 BNE DESPLP
 RTS

LDA SEP,Y

BNE DISPL

INY

=4?

RTS



POINTS
 LDA #0
 JSR SEPAR
 LDA #5F
 STA SCRH
 LDA #5F
 STA SCRL
 JSR DISP
 LDA #2
 JSR SEPAR
 LDA #3F5
 STA SCRL
 JSR DISP
 RTS

see

8/4/82

✓ 1 MUSIC BUL → HOUSE BRINGS SCORE → 25000
50000

LD A
CMP #5B-
BES
[MISC
]

2 OBJECT - guys 

✓ 3 DRAW HOUSE IF BOTH BONUS = 0

LD A BONUS0
ORA BONUS2
BEQ
JSR DRSTHS

✓ 4 ☐ CLAIM JUMPER
☐ SHOOT EM EASY ONE
☐ SHOOT EM TWO

✓ 5, 50000 PTS for second house

BUL 6. disallow repeat shoot / hit

~~7 10 pts for bul~~

MAIN 8. allow shoot in ASSAY

8/7/82

STCOL 5AF0 → 2EAC
STRTES 5AF3 → 2C3C
OPITES 5AF6 → 2C9A

SCDATA F9

1 strange track

1 sector fast less time to read

how long to read

test for bad sector

test for good sector

OS manual

resident disk handler
ROM

JSR disk handler

figure out how disk handler
how to detect bad sectors

2DP

sector 1

sector 720

cassette
make sure
no disk
attached

86

512
182
6
790

2CB
512
12x16
6

52 → 302
01 → 301

512
208
720

1 3
256 | 720
768

24 720
512
208

48 300

30
\$200 - \$2BF

304

308

sector 1

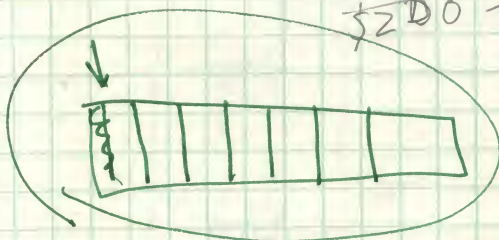
0 0
0 1
destination
Angchen

sector

starting
code

40

18 | 720



2C0-2CA
BAD SECTORS
CANT READ

02 ←
instruction
crashes

FUNNY TRACK

\$20-\$23

all sector #1
read 10 times

32 track #
18 sectors/track

256
32
576

← sector to read

20
12
40
20
\$0240

START
delete linker

STCOL
STRTES
OPITES

change
3290-3320

DATA

MODEY

needed
outside

STCOL 4B7B
STRTES 490C
OPITES 496A

BY CLAIM
POINTS

9/6/82 DISK
STCOL 2B74
STRTES 28FF
OPITES 2960

9/7/82 CASS
STCOL 2B77
STRTES 2902
OPITES 2963

need from
outside
link

SMASH0 ✓
SMASH2 ✓

MOV0 ✓
MOV2 ✓
MOVZ ✓

CREATE ✓
SMOOTH ✓

CANROW
BUL ✓

SOUND ✓

- RANDSC
ENDFR ✓

- P06BUL

- P26BUL

POINTS ✓

FINDSC ✓

PM
30 > 28

DUST
5900 →

CANROW

RANS
screen

needed
outside

P06BUL 3E90
P26BUL 3E82

supply

need
from outside

SMOO

FINDSC 4130
SMASH0 4282
SMASH2 428B

supply

CLAIM

SC DATA
STCOL
need

MAIN
DATA

"MAN"
"FLESH"

:=

needed
outside

SMOO
CANROW
WIDEN0
WIDEN2
CREAT

need

need from
outside
supply

MOUL22 59CF
MOV2 5830
JMOVZ
MOVE22 59CF

CREAT

RANDSC
3B77

BUL

WIDEN0 4660
WIDEN2 468

supply

FINDSC
need

POINTS

ENDFR
STRTES
OPITES
SOUND

need

2A91 498D
2A81

4834 33 0A
4895 pDES
488P

44 pages
88 tracks

49 pages = 86 sectors = 5 tracks
55 sectors

3200 →
4000 → 5600 ✓ MAIN × X 574
2008 → 48D0 ✓ START × 782
6200 → 42D0 ✓ BUL × 3F8
6006 → 54D0 ✓ POINTS × F8
4E00 → 3BC0 ✓ CANROM × 40B
4800 → 3FD0 ✓ SMOD × 2F3
5B00 → 3408 CLAIM × 9E
4C00 → 3AA0 ✓ CREAT × 171
3900 → 3A00 ✓ SOUND × 8D
3F80 → 35E0 ✓ DLIROU × 13
4500 → 47D0 ENDR ×
3600 → 369F ✓ INIT BUT × 9F
✓ INIT TOP OF -

3A00 → 34B0 CLAIM.DAT 4F
6100 → 3300 CLAIM.CHAR CF
3F00 → 5500 ✓ GUY'S "MAN" 7F
3C00 → 5580 ✓ BM3 "FLESH" 7F
3D00 → 3500 ✓ TITLES × CB
3E00 → × MAG-FILE X
3E80 → 39D0 ✓ CACTI (Mouses) × 29
4600 → 3900 ✓ VOL BF
4700 → 4700 ✓ FREQ BF
5300 → 3800 ✓ INTSK X FF
5906 → 5B80 ✓ DLIST 23
BAS 38 → 34 ✓ CHMR LES × 157
BAS 54 → 50 ✓ CHMR × 3FF

STCOL 487B
STATES 490C
OPITES 496A
Version C
4881
490C
496D

BUL [WIDEN0 4663
WIDEN2 4681

MAIN [× MOVZ 5830
MOVZ 59CF

SMOD [FINDSC 4130
SMASH0 4282
SMASH2 428B

CREAT [RANDSC 3B77 - BCC →
BNE →

CANROM [P06BUL 3E90
P26BUL 3E82

2C -
3638
2D -
3E
3F
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

5D screen direct

65 W DEN

88 track sector header

1	501 - 12
2	13 - 24
3	25 - 36
4	37 - 48
5	49 - 5A
6	5B - 6C
7	6D - 7E
8	7F - 90
9	81 - A2
10	A3 - B4
11	B5 - C6
12	C7 - D8
13	D9 - EA
14	EB - FC
15	FD - 10E
16	10F - 120
17	121 - 132
18	133 - 144
19	145 - 156
20	157 - 168
21	169 - 17A
22	17B - 18C
23	18D - 19E
24	19F - 1B0
25	1B1 - 1C2
26	1C3 - 1D4
27	1D5 - 1E6
28	1E7 - 1F8
29	1F9 - 20A
30	20B - 21C
31	21D - 22E
32	22F - 240
33	241 - 252
34	253 - 264
35	265 - 276
36	277 - 288
37	289 - 29A
38	29B - 2AC
39	2AD - 2BE
40	2BF - 2D0

read 10 X

or 5C

try to read

\$62 bad sector
4747474747474747
4747474747474747

DO bad sector

COPY
1-E0
F1-1ED
1E1-2D0

track 15 - all FF

2000.9BFF

OR \$47D0
ENDR

\$168 VOTC
\$169-170
DIR

LDA \$D40B
CMP #570
BCS FRAME
BS LDA \$D40B
CMP #570
BCC BS
RTS

OR \$35E0

PHA
LDA # \$A0
EOR \$4F
AND \$4E
STA \$D40A
STA \$D018
LDA # \$33 34
STA \$D409

PLA
RTI

DOF PAGE
V.2
42 27 00B4
42 25 0053
42 01 0050

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C
4	7	4	7	4	7	4	X	4	7	4	7	4	7	4	7	4	I
\$5E																	
\$61																	

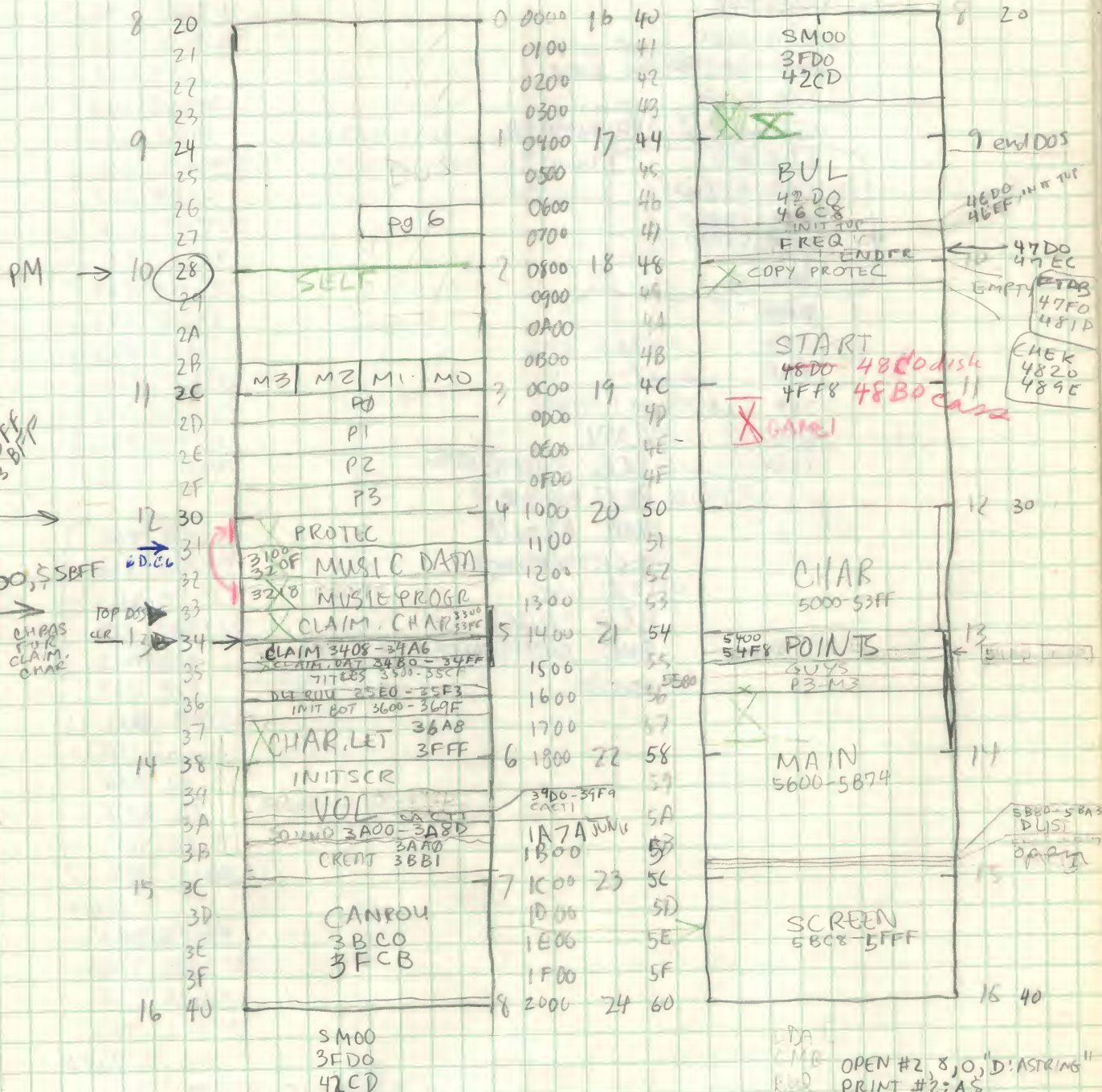
00-28 169
53-78
50-51

SECTION

creative computing

Sept 8 2
pg 90

April DOG DATA
Sandy Small



3100 < 6D, C6 (w)
6000 < 1.2 (w)

287D 07 06 40 35 FI

(4)
2E3E A5 06 LDA 6
2E40 05 07 ORA 7
2E42 DO DE BNE CRSH+1
2E44 60 RTS
↑
4096 2416 1

1E42 = 7746

NONSENSE

> LDA \$6FE
~~STA \$39FE~~ CMP #1
BNE DUMB+1

> LDA \$6FF
~~STA \$39FE~~ CMP #1
BEQ DUMB+1

LDX #0
Q LDA \$1200, X
STA \$2800, X
P: A \$3000, X
INX
BNE Q
PHA
TXA
PHA

JSR \$28D0

CRASH

BCC DUMB+1

BEQ DUMB+1

JSR MOV6

JSR XPYSC

BEQ CRASH

BNE NONSENSE

DATA

COUNT EQ 6FD

OR \$6000

HS 00 01 00 32 XXXX

FILE LOADS INTO \$3200

LDX #0

STX SECTHI

STX BUFLO

LDX #\$31

STA BUFHI

LDX #\$E7

STX DCOMND

LDX #\$52

STX SECTLO

JSR DISKOP

LDY STATUS

STY \$6FE

LDX #\$62

STX SECTLO

JSR DISKOP

LDY STATUS

STY \$6FF

STATUS SECT 52 → 6FE

STATUS SECT 62 → 6FF

INC BUFHB

INC SECTLB

JMP LDLP

DUMB LDA #0

STA \$6F8

START

JMP \$48D0

#2 later

LDY #10

STY COUNT

LDX #\$5E

STX SECTLO

LDY #0

STY \$14

LP JSR DISKOP

DEC COUNT

BNE LP

LDA \$14

CMP #\$7F

BCS DUMB+12

LOAD

LDY #\$6D

STY SECTLB

LDY #\$33

STY BUFHB

LDLP

LDA SECTLB < CMP #\$C7
BEQ START

STA SECTLO

LDA BUFHB

STA BUFHI

JSR DISKOP

LDA STATUS

CMP #1

BNE LOAD

OBJECT OF THE GAME

CLAIM JUMPER is a competitive two-player action game. The object of the game is to collect gold nuggets, trade them for cash, and ~~to~~ save cash in your bank for ^{buying} purchasing houses ~~and to prevent your opponent from doing the same things.~~ At the same time, you ~~must~~ ^{try to} avoid snakes or tumbleweeds ^{that are} pursuing you. Quite often, a gunfight determines who gets ~~the~~ the money safely to his bank. The winner is the first player to accumulate 25,000 points. ^{is the winner} Points are awarded for purchasing houses and for successfully fighting off ~~the~~ pursuing snakes or tumbleweeds.

LOADING CLAIM JUMPER

Make sure that you have at least 16K of memory, and have removed the BASIC or other cartridge from the cartridge slot(s). Plug joystick controllers into the first and second controller jacks. (Only the first joystick controller is needed for the optional one-player game). If the computer is on, turn it off.

FOR CASSETTE:

CLAIM JUMPER

Insert the cassette with SIDE A facing up. (This is the side with the label on it.) Press REWIND to make sure that the tape is completely rewound, and then press STOP. Press PLAY to prepare the program recorder for loading the program.

Press the START key and hold it down. Turn on the computer. After you hear the "beep", ^{release the START key and} press RETURN.

CLAIM JUMPER will take approximately three minutes to load.
If you should have any difficulty...

FOR DISK:

Turn on the disk drive and insert the CLAIM JUMPER disk. Turn on the computer. The program loads automatically.

> OBJECT OF THE GAME [

Collect gold nuggets

Trade for cash

Save money

} before opponent

> Start Buy house press start or option

At some time, Avoid hazards, theft

One player game is described under Options

> Acquiring Gold and money

> snakes and, Tumbleweeds and other hazards (paralysis)

> Trigger button  drop money
shoot
drop seed or egg

> ~~drop~~ seeds and eggs

> Shooting

direction of shooting

hit opponent - hospital

with snake or tumbleweed

> Scoring and Winning

Scoring

Winning

[continuation to 50000
pressing START or OPTION terminates the game]

→ (An assay office is a place that you have the purity of gold analyzed)
STARTING THE GAME

To start a new game, press START. You can also press OPTION to choose the options described later in these instructions.

PICKING UP GOLD AND MONEY

You control your cowboy with your joystick. The left joystick controls the dark (brownish green to green) cowboy, while the right joystick controls the light (pinkish to orange) cowboy. (The actual color depends on your TV ^{screen} adjustments.) The cowboy runs in the direction you push ~~move~~ your joystick.

A flickering gold nugget appears on the screen. To pick up the gold nugget, all you (the cowboy) have to do is run over and touch it. As long as you have the nugget, your opponent can't take it away from you. However, you'll soon see that you might want to drop the nugget. ^{or you might be forced to drop it.} ~~To drop it, simply press your trigger button.~~ ^{If you want to drop it, simply press your trigger button.}

A nugget is useless until it is traded for cash. To trade the nugget for cash, take it to the Assay office, located at the center top of the screen. A flashing dash shows you where to place the nugget. You needn't let go of the nugget; simply place it over the flashing dash. The nugget disappears and, after a few seconds, a ~~ten~~ dollar bill appears to the left of the assay office.

To pick up the ^{ten} \$-dollar bill, all you have to do is touch it. Just like the nugget, your opponent can't take the bill away from you as long as ~~you have it~~, unless you drop it, or you are forced to drop it.

Your goal is to save money in your bank, located in the lower left (dark bank) or lower right (light bank). To make a deposit, carry the bill to YOUR bank and drop it off. To let go of the bill, simply press your trigger button. The bill must be completely within the colored area of the bank to be deposited.

When you have saved ten bills, you will purchase an additional house ~~and a cash~~. The house purchase occurs automatically.

OPTIONS

There are two menus for selecting game play -- the CLAIM JUMPER and GAME SELECT MENU.

CLAIM JUMPER

Press OPTION key once to display the option menu. Press the OPTION key a second time to display the GAME SELECT MENU. The available options are for each player are

- ☐ NORMAL GAME
- ☐ BUY BULLETS
- ☐ HEAD START

Each player uses his joystick button to

The two players use their trigger buttons individually for themselves individually. Press your own

button once to select the BUY BULLETS option, once again to select BOTH BUY BULLETS and HEAD START, once again

to trigger button repeatedly until the desired option or combination of options are checked off.

describe

NORMAL GAME as "NORMAL GAME" is simply CLAIM JUMPER as described up to this point, you start the game w/ 10 bul.

BUY BULLETS

HEAD START

GAME

SELECT MENU

you have a limited number of bullets and must purchase them as you run out.

"BUY BULLETS" is a challenging game variation in which you must purchase bullets. When you run out of bullets, you may purchase 10 more bullets with a single 10-dollar bill. To make a purchase, carry a freshly-acquired 10-dollar bill to the door of the bullet store, located at the bottom center of the screen. Two flashing dashes show you where to take the money. When you place the money on the doorway, the money disappears and you get 10 bullets. The dashes under your score show how many bullets you have remaining as you use them up.

You can purchase more bullets before you run out of them, but you cannot accumulate more than 10.

When the OPTION menu is being displayed, pressing

The GAME SELECT MENU appears allows you to select three different games:

- CLAIM JUMPER
- SHOOT 'EM ONE
- SHOOT 'EM TWO

Use the SELECT key to select one of the three games. CLAIM JUMPER is the two-player game already described in these instructions. SHOOT 'EM ONE and SHOOT 'EM TWO are one-player shoot-em-up games. They allow you

You may not withdraw money from the bank; you must use a freshly acquired 10-dollar bill.

SNAKES, TUMBLEWEEDS, AND OTHER HAZARDS

Shortly after the game starts, two types of creatures start to appear on the screen -- snakes and tumbleweeds. The snakes, which are pink, are similar in color to the light cowboy. The tumbleweeds, which are brown, are closer in color to the dark cowboy. The snakes and tumbleweeds pursue the cowboy of the opposite color. In other words, the snakes chase after the dark cowboy and the tumbleweeds chase after the light cowboy. When a pursuing creature catches you, you are paralyzed for two seconds. If you have a gold nugget or ten-dollar bill, you will drop it, thereby allowing your opponent to steal it away from you.

After the two seconds are up, you are free to move again. You will have a brief period of immunity against paralysis. You should take advantage of this immunity by moving away from the pursuing creatures.

If you are the dark cowboy, you must avoid not only the pink snakes, but ~~stationary~~^{other} pink objects as well. They include:

- The pink and-blue house
- The boardwalk that forms the ^{right} screen border ~~on the right~~
- The pink bank
- ~~The two barbed-wire fences on either side of the store~~
- Pink snake eggs

The light cowboy must avoid not only the brown snakes, but also:

- The brown and-green house
- The boardwalk that forms the left screen border
- The brown bank
- Brown tumbleweed seeds

Both cowboys must avoid:

- The upper half of each cactus
- The building located at the bottom center of the screen
- The barbed-wire fence around this building

Touching any hazard causes you to be paralyzed for two seconds and to drop ~~the~~^{any} nugget or bill that you're carrying.

to improve your shooting and maneuvering skills when another ~~one~~ SHOOT 'EM ONE and SHOOT 'EM TWO ^{These are} identical to each other, except that SHOOT 'EM TWO is more difficult.

In both SHOOT 'EM games, the object of the game is to shoot and destroy all snakes and tumbleweeds on the screen. You can be paralyzed only twice; if you are paralyzed a third time, the game stops and you lose. Three small figures appear on the right at the bottom of the screen to... If you succeed in destroying all snakes and tumbleweeds, you win. ✓

In SHOOT 'EM, you are always the dark cowboy (left joystick). ^{The snakes crawl towards you, while} You are pursued by the snakes; the tumbleweeds wander at random. ^{The tumbleweeds} do not pose a hazard to you. ^{even more} To make the game challenging, the computer drops snake remnants at random on the screen. ^{You must avoid} running into them. In addition, the snake ^{remnants} cause tumbleweeds to turn into snakes.

They appear more and more frequently as the game progresses.

In shoot 'em, the BUY BULLETS option is always in effect, and you start the game with no bullets. ^{Before you can shoot} Therefore, you must first pick up the gold nugget, trade it for cash at the assay office, and use the cash to purchase bullets at the store. You get 10 bullets with each purchase, and

USING

YOUR TRIGGER BUTTON

pressing the button

You ^{can} use your trigger button for three different purposes. The result ^{of} depends on your situation when you press ~~the~~ button:

1. If you are carrying the nugget or bill, you drop what you are carrying.
2. ~~If you are not carrying~~ If you are standing still, you drop off a tumbleweed seed (dark cowboy) or a snake egg (light cowboy). Seeds and eggs are for defending yourself from pursuing creatures and for creating more hazards for your opponent. They are described in the next section. ~~At the~~
3. If you are running, you shoot a bullet. You shoot ~~the~~ the bullet in the direction you're running. Shooting bullets is described after the section on SEEDS AND EGGS.

DROPPING SEEDS AND EGGS

~~If you are not carrying a nugget or bill. To drop a tumbleweed seed or a pink snake egg, ^{first} releasing your joystick and ^{then} pressing your trigger button.~~

To drop a seed or egg, let go of your joystick, and then press your trigger button. (You can't drop a ^{seed or} egg while running or while carrying a nugget or bill.) The dark cowboy drops ~~brown~~ tumbleweed seeds, while the light cowboy drops pink snake eggs. When a snake eats a tumbleweed seed, the snake is ~~the~~ transformed into a tumbleweed; when a tumbleweed tumbles over a snake egg, the tumbleweed is transformed into a snake. In ~~other words~~, the seeds or eggs that you drop can convert a creature that is chasing after you into a creature that chases after your opponent instead.

Dropping seeds or eggs is perhaps your most effective defense against creatures (snakes or tumbleweeds) that are chasing after you. In addition, the seeds or eggs that you drop are an ~~an~~ additional hazard ^{for} that your opponent ~~must~~ ^{to} avoid.

⚠ If a tumbleweed tumbles ~~over~~ over a seed, or if a snake eats a snake egg, no transformation occurs.

You may not accumulate more than 10 bullets.

¶ You may have ~~not~~ more than six seeds or eggs on the screen. If you try to drop a seventh seed or egg, the oldest one on the screen is erased automatically. However, you can always drop a new seed or egg. If ~~you~~ ^{there are} already have six on the screen, the oldest one on the screen is erased.

¶ There can be no more than six seeds or six eggs on the screen. When the dark cowboy drops a seventh seed, ~~the~~ the oldest seed on the screen is removed automatically, so ~~there are~~ that only six remain ^{on the screen}. The same is true for eggs. The light cowboy dropping eggs.

¶ You can drop seeds & eggs in ~~the~~ clear areas only. You can not drop them on top of another seed or egg, on top of a building, or on top of a snake or tumbleweed. In addition, seeds and eggs are forbidden on the bottom line (below the level of the barbed-wire fence) and ~~inside~~ the hospitals (in the upper left and upper right corners). You can, however, drop a seed or egg directly on your opponent, thus causing him to be paralyzed.

SHOOTING BULLETS

Shooting bullets is a good way to steal (or steal back) your opponent's ~~gold~~ nugget or money. Also, you can quickly create more hazards for your opponent by shooting both the snakes and the tumbleweeds.

To shoot a bullet, press your trigger button while you are running. The bullet shoots ~~out~~ in the direction you're running. (You can't shoot while standing still or while carrying a nugget or bill.) You can't shoot again until your first bullet ~~hits something or~~ goes off the screen, or ~~hits either your opponent, a snake, or a tumbleweed.~~

(If you are running diagonally, the bullet shoots in a horizontal direction.)

GETTING HIT

If you are hit by a bullet, your hat flies off and you are transported automatically to one of the two hospitals, located in the upper left and upper right corners of the screen. You may choose ~~which~~ the hospital you go to by pushing your joystick left or right. If you don't designate one ^{hospital} or the other, the computer chooses one at random. You'll leave behind any nugget or bill you happened to be carrying. After a very brief stay at the hospital, you are fully

recovered and ready to resume playing. You will have a brief period of immunity against paralysis, just like after you have been paralyzed.

The hospital is a ^{peaceful} safe area where no hazards exist and you can not get shot. Conversely, you may not shoot any bullets while in the hospital or while touching any part of the blue hospital area. ~~If the cowboys shoot each other, they may be sent to the same hospital.~~

After a very brief stay at the hospital, you are fully recovered and ready to re-enter the hazardous play area.

HITTING SNAKES AND TUMBLEWEEDS

~~The result of shooting a snake or tumbleweed depends on who shot it. For example, if the dark cowboy shoots a snake, the snake is transformed into a tumbleweed.~~

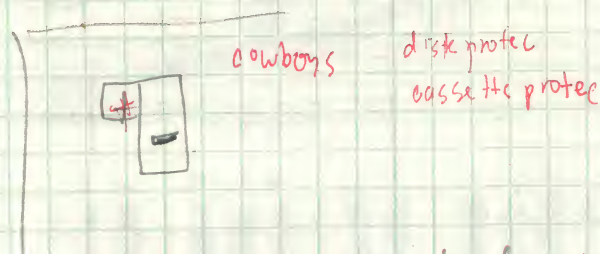
If you are the dark cowboy and you shoot a snake, the snake will be transformed into a tumbleweed. In other words, you can shoot a creature that is chasing you and change it into a creature that chases your opponent.

You can shoot a creature (snake or tumbleweed) ^{that is chasing after you} and change it into the opposite creature. In other words, if you are the dark cowboy, you can shoot snakes and turn them into tumbleweeds, which chase after the light cowboy; if you are the light cowboy, you can shoot tumbleweeds and turn them into snakes, which chase after the dark cowboy.

~~If you are the light cowboy and shoot a snake~~

You may also shoot a

~~If you shoot a~~ creature that is already chasing after your opponent. Doing so ^{doesn't} kills it, leaving only a remnant. For example, if you are the light cowboy and you shoot a snake, you kill the snake, leaving a snake remnant. A snake remnant looks like a pink "S". It behaves just like a snake egg; when a tumbleweed rolls over it, the tumbleweed is transformed into a snake. Similarly, a tumbleweed remnant looks like a brown "T". It behave just like a tumbleweed seed; when a snake eats it, the snake is transformed into a tumbleweed.



SCORING

You score points for buying houses and for fighting off the pursuing snakes or tumbleweeds. Your current score appears at the screen bottom near your bank.

You score points for buying houses and for converting creatures that are chasing after you into creatures that chase your opponent.

When the game starts, you own one house. your next house, and 30,000 for buying the next one.

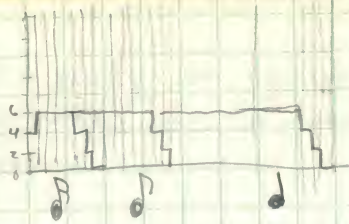
You score 20,000 points for buying a house. To buy a house, you need to save ten bills in your bank.

If you are the dark cowboy, you score 100 points for each snake you convert into a tumbleweed, either by using a tumbleweed seed or by shooting. Similarly, if you are the light cowboy, you score 100 points for each tumbleweed you convert into a snake.

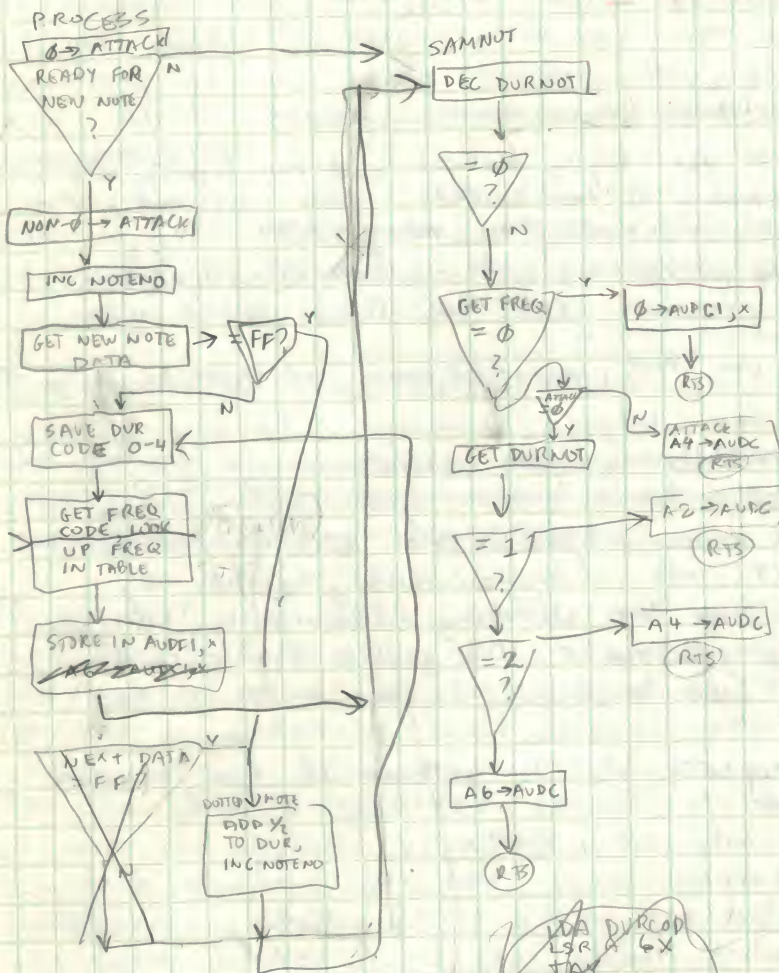
Your current score appears at the bottom of the screen near your bank.

WINNING

The first player to score 25,000 points is the winner. The game stops automatically when this score is reached. After the game stops, you may choose to continue the same game by pressing the SELECT key. The game will resume and continue until one player reaches 50,000 points.



2000	TABLE	20
2080	DURTAB	21
2100	DATA1	
2180	DATA2	22
2200	DATA3	
2280		23
	UBJFG	24



PROCES

```

    LDA #0
    STA ATTACK
    LDA DURNOT, X
    BNE SAMNOT
    LDA #1
    STA ATTACK, =0
    INC NOTENO, X
    LDY NOTENO, X
    LDA (SCRL+4), Y
    CMP #FFF
    BNE DOTOK
    INC NOTENO, X
    INC (SCRL+4), Y
    LDA #1
    STA DOTFL
    INC NOTENO, X
    LDA (SCRL+4), Y
    LSR A
    CLC
    ADC
    DOTOK
    AND #5C0
    STA DURCOD
    LDA (SCRL+4), Y
    AND #5BF
    TAY
    LDA FTABLE, Y
    STA AUDFI, X

```

SAMNOT

```

    LDA DURCOD
    LSR A, 6X
    TAY
    LDY DURCOD
    LDA DURTAB, Y
    STA DURNOT, X
    LDY DOTFL
    BEQ SAMNOT
    LSR A
    CLC
    ADC DURNOT, X
    STA DURNOT, X

```

DOTOK

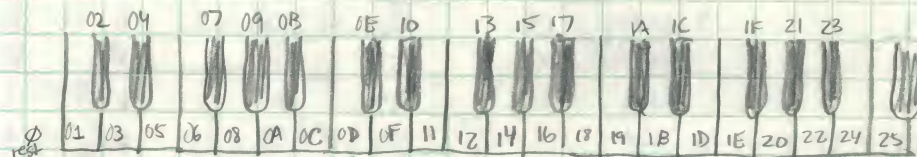
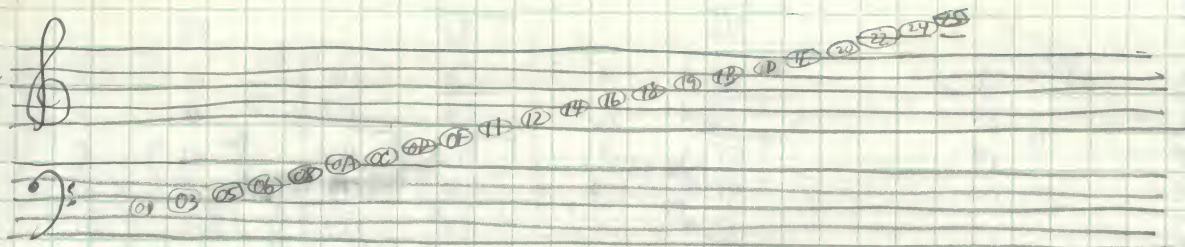
```

    AND #5C0
    STA DURCOD
    LDA (SCRL+4), Y
    AND #5BF
    TAY
    LDA FTABLE, Y
    STA AUDFI, X

```

SOURCE

SCREEN



XX XX XX XX (FF)
DUR PR dot

MID
C

X = VOICE 0, 2, 4

→ EQ NOTNO 6B8 } VOICE 1
→ EQ DURNOT 6B9 }

6BA } VOICE 2

6BB } VOICE 3

6BC } VOICE 3

6BD } VOICE 3

EQ DATAH 6BE

EQ FTABLE 6BF

EQ DURCOD

EQ DURTAB table { 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 }

CLEAR
LDX #5
LDA #0
MLP STA NOTNO, X } CLEAR
DEI MLP
RTS

PROCESS DATA STA ATTACK

LDA DURNOT, X

BNE SAMNOT

STA ATTACK

INC NOTEND, X

LDY NOTEND, X

LDA (SCRL+4), Y; DATA

AND #5C0

STA DURCOD

LDA FTABLE, Y

STA AUDFI, X

LDA #5A6

STA AUDCT, X

LDY DURCOD

LDA DURTAB, Y

STA DURNOT, X

LDY NOTEND, X

INY

LDA (SCRL+4), Y

CMP #5FF

BNE DOTOK

LDA DURNOT, X; DOT

LSR A

CLC

ADC DURNOT, X

STA DURNOT, X

DEC NOTEND, X

RTS

LDX #0
LDA #DATAH
STA SCRL+4
LDA #0
STA SCRL+4

SAMNOT DEC DURNOT, X BEQ REST

LDY NOTEND, X

LDA (SCRL+4), Y; DATA

AND #53F

BEQ REST

LDA DURNOT, X

BEQ ENDNOT

CMP #1

BEQ D1

CMP #2

BEQ D2

LDA #5A6

DUN STA AUDCT, X

RTS

REST

LDA #5A0

BEQ DUN

D1 LDA #5A6

BNE DUN

D2 LDA #5A6

BNE DUN

ATVOL

LDA #5A4

BNE DUN

LDA ATTACK

PNE ATVOL

OPTIONAL

BACK JSR CLEAR
LDX #0
LDA #DATAH
STA SCRL+4
LDA #DATAH
STA SCRL+4
JSR PROCESS
JSR ENDER
JMP BACK

AUDCTL.EQ D208
AUDFI.EQ D200
AUDCT.EQ D201
DATAH1 21
DATAH2 00
DATAH3 21
DATAH4 00
DATAH5 21
DATAH6 00

FTABLE 2000
DURTAB 2080

NOTEND 6B8
DURNOT 6B9

ATTACK 6BE
DURCOD 6BF

DOTFLG 6B6

1	<u>CO</u>	19
2	<u>CO</u>	25
	FF	19
3	<u>9E</u>	00
	<u>60</u>	19
	FF	25
4	<u>A2</u>	19
	<u>5E</u>	00
	5B	19
5	5E	25
	5D	14
	<u>60</u>	00
6	9E	19
	99	25
	9E	19
7	20	9E
	<u>21</u>	20
	A2	21
	22	A2
8	22	22
	20	22
	<u>1E</u>	20
10	<u>DE</u>	1E
	00	
	19	
	25	
11	19	
	00	
	19	
	25	
	<u>19</u>	
	00	
	19	
	25	
	<u>19</u>	
12	00	
	19	
	25	
	19	
	00	
	19	
	25	
	<u>19</u>	
13	00	
	19	
	25	
	19	
	00	

1	00 d	19
2	00 d	25
	FF	19
3	9E	00
	60	19
	FF	25
4	A2	19
	5E	00
	5B	19
5	5E	25
	5D	19
	60	00
6	9E	19
	99	25
	FF	19
7	20 A	00
	21	19
	A2 Q	25
	22 S	19
8	22 S	00
	20	19
	1E	25
	1B	19
	1A	9E
	1B	20
	1E	21
9	1D	A2
	1C	22
	1D	22
	20	20
10	DE	1E
	00	1B
	19	1A
	25	1B
11	19	1E
	00	1D
	19	1C
	25	1D
	19	20
	00	DE

2100 00
2152 DE

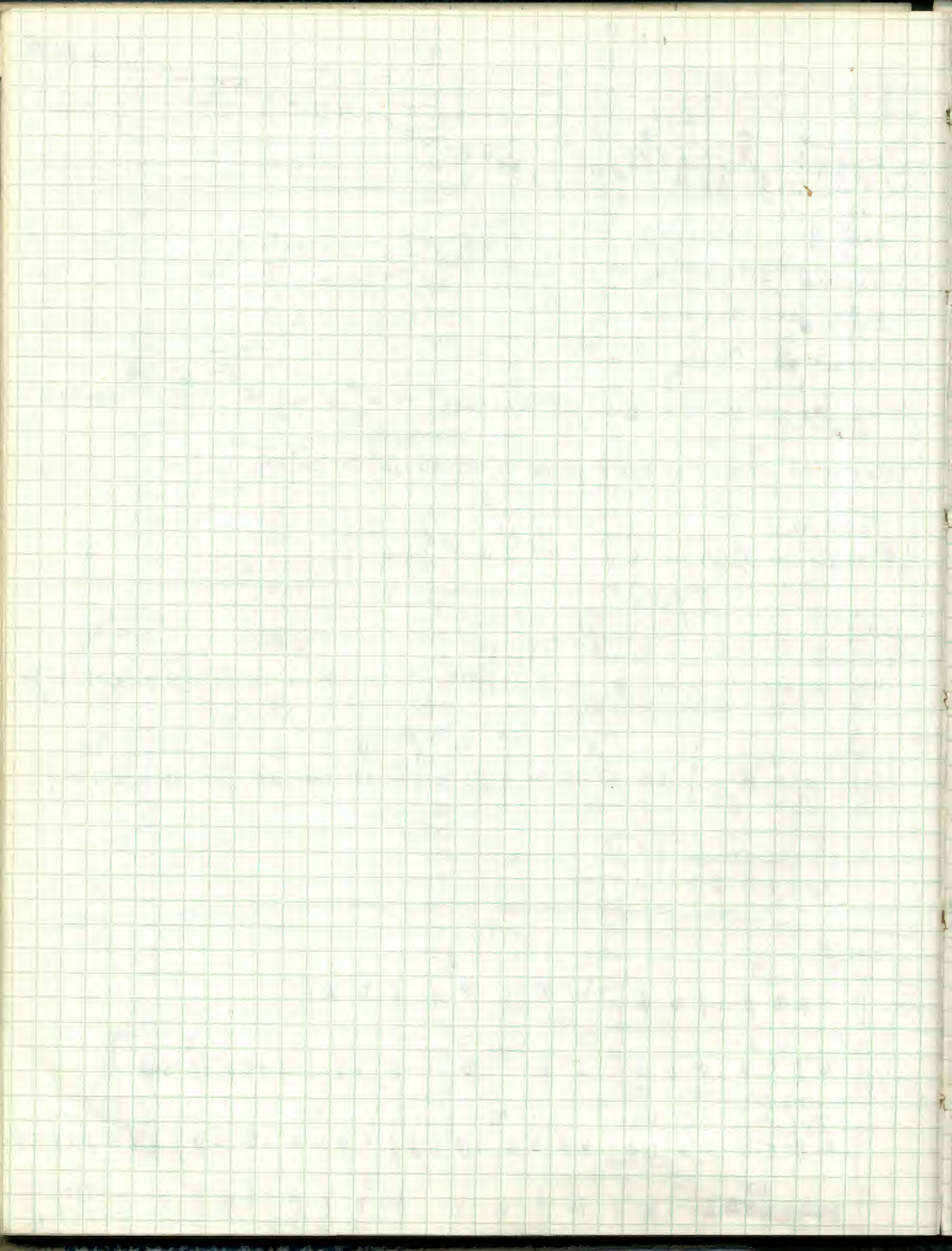
	1 C♯	2 C♯	3	4	5	6	7	8	9	10	11	12	13	14	15
	40	52	56	57	40	58	59	56	40	57	40	58	59	56	40
	52	56	57	40	58	59	56	40	57	40	58	59	56	40	52
	56	57	40	58	59	56	40	57	40	58	59	56	40	52	56
	57	40	58	59	56	40	57	40	58	59	56	40	52	56	57
	40	58	59	56	40	57	40	58	59	56	40	52	56	57	40
	58	59	56	40	57	40	58	59	56	40	52	56	57	40	58
	59	56	40	57	40	58	59	56	40	52	56	57	40	58	59
	56	40	57	40	58	59	56	40	52	56	57	40	58	59	56
	40	57	40	58	59	56	40	52	56	57	40	58	59	56	40
	57	40	58	59	56	40	52	56	57	40	58	59	56	40	57
	40	58	59	56	40	57	40	58	59	56	40	52	56	57	40
	58	59	56	40	57	40	58	59	56	40	52	56	57	40	58
	59	56	40	57	40	58	59	56	40	52	56	57	40	58	59
	56	40	57	40	58	59	56	40	52	56	57	40	58	59	56
	40	57	40	58	59	56	40	52	56	57	40	58	59	56	40
	57	40	58	59	56	40	52	56	57	40	58	59	56	40	57
	40	58	59	56	40	57	40	58	59	56	40	52	56	57	40
	58	59	56	40	57	40	58	59	56	40	52	56	57	40	58
	59	56	40	57	40	58	59	56	40	52	56	57	40	58	59
	56	40	57	40	58	59	56	40	52	56	57	40	58	59	56
	40	57	40	58	59	56	40	52	56	57	40	58	59	56	40
	57	40	58	59	56	40	52	56	57	40	58	59	56	40	57
	40	58	59	56	40	57	40	58	59	56	40	52	56	57	40
	58	59	56	40	57	40	58	59	56	40	52	56	57	40	58
	59	56	40	57	40	58	59	56	40	52	56	57	40	58	59
	56	40	57	40	58	59	56	40	52	56	57	40	58	59	56
	40	57	40	58	59	56	40	52	56	57	40	58	59	56	40
	57	40	58	59	56	40	52	56	57	40	58	59	56	40	57
	40	58	59	56	40	57	40	58	59	56	40	52	56	57	40
	58	59	56	40	57	40	58	59	56	40	52	56	57	40	5

2180 00
2184 D2

6	6	11	8D
0	0		8A
D	C	12	8B
0	6D		88
1	1		FF
0	0		4D
D	D	B	B
0	0		4A
6	6		4D
0	0		4B
C	D	14	4A
D	0		88
2	7		6
0	0		0
D	D		D
0	0		0
6	6	5	1
0	0		0
D	D		D
0	0		0
1	1		0
0	0		D
D	D	6	0
0	0		1
C	B		D
D	4		0
1	1		3
0	0		0
D	5		B
0	0		0
3	6		1
0	5		0
B	6		5
0	8		0
1	A		18CG
0	8		
5	A		
0	B		

2206 00
2275 C6

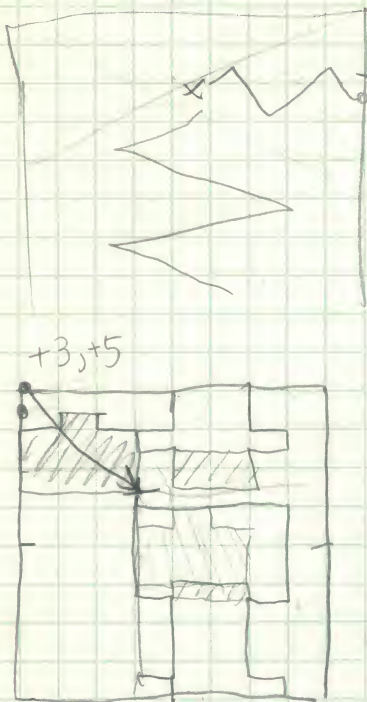
6568 ¹⁰ A8AB	Q ¹¹ D A	B ¹² 8	(FF) ^{E?} D ¹³ B A D	B ¹⁴ A 8	15 16 17	18
			(Q) (E) (E)	(E) (E) Q	789	6 H



COMPRESSION

MAIN

1330 MOVØ
LDA #Ø
LDY #PØBAS
BNE MOV
MOVZ
LDA #1
LDY #P2BAS
MOV
STY YBAS
STA IDX1
TAX
ASL
STA IDX2
ASL
STA IDX4
TAY
LDA YPØ, Y



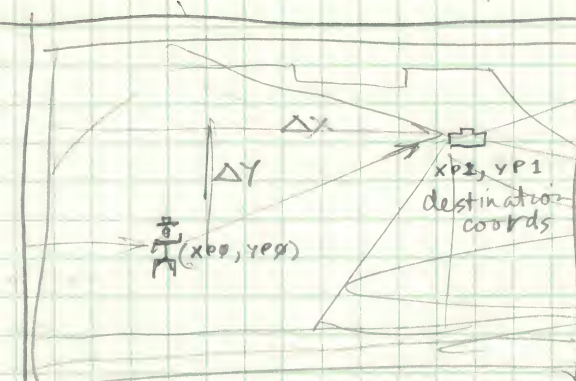
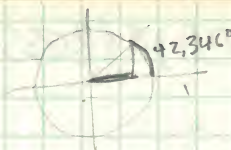
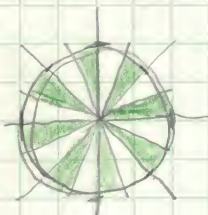
destination

(XPI-3, YPI-3)

DESTIN

00 RANDOM

STATION 01 CAN COORDS - 3



beeline zone
 $\Delta Y \leq \Delta X$ or $\Delta Y < \$10$

zig-zag zone
one second per zig

$\Delta Y > \Delta X$
and $> \$20$

in each 1-sec period,
press and release button
once. make it $\frac{1}{64}$
chance of press per
frame
CHANCE = $\frac{1}{64}$ in 1 frame

DISK 5

DISK 6

DISK 7

in 2 frame

22% chance in 16 frames

40%

32 Frames

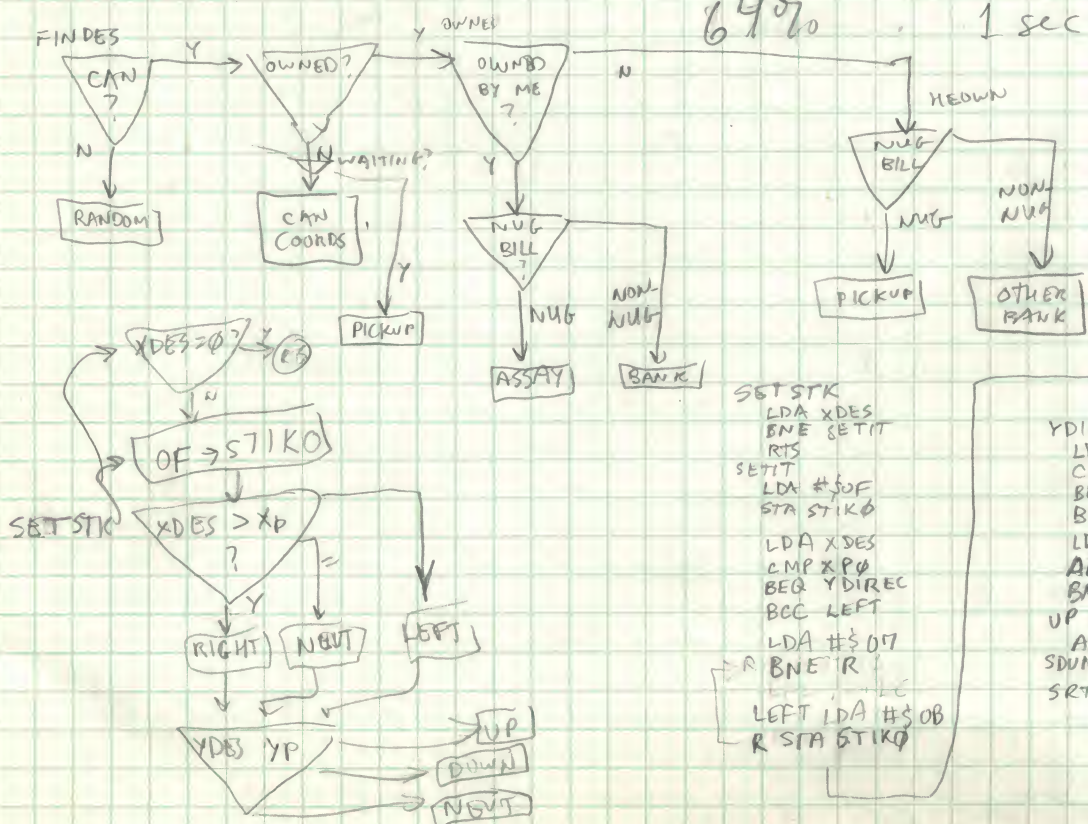
64%

1 sec

CLAIMX, OBS

SELF

SELF, OBS



SELF-PLAYING VERSION

STICK 700
STICK1 701

TRIG 702
TRIG1 703
FRAME 600

XDES 704
YDES 705
XDES 706
YDES 707

RANDOM

STATCN \$670

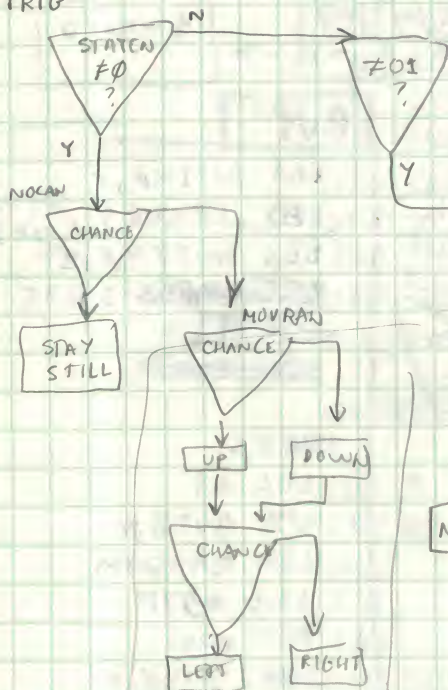
00 none
01 present
80 P0 owned
81 P2 owned

CNTYPE 678
08 nug
00 bill
FF waiting

MAIN:
STIKO = \$700
STIKI = \$701

START
ISR ENTESD
\$2800

TRIG



STK
RANDOM

LDA RANDOM
BPL STILL
AND #7
TAX
LDA TABLE,X
Q STA STICK
LDX #0
BEQ DUN
STILL #0F
STICK
BNE a

1111
1110
1101
1011
0111
0110
1010
0101
1001

OR \$2800
LDA 275
STA 701
LDA FRAME
AND #53F BNE QRTS
ISR FINDES
JSR SETSTK
QRTS RTS

FINDES
LDA STATCN
BEQ RANSTK
BPL OWNED
LDA CNTYPE
CMP #5FF
BNE CANCEC

PICKUP

LDA #
STA XDES
LDA #
STA
RTS

LDX #56Y
LDA #520
BNE DUN

CANCEC

LDA XPI
STA X
LDA YPI
STA
RTS

LDA YPI
SEC #3
TAX
LDA XPI
SEC #3
BNE DUN

OWNED

CMP #580
BNE HEOWN
LDA CNTYPE
CMP #510
BNE BANK

TASSAY

LDA #
STA
LDA #
STA
RTS

LDX #58Y
LDA #518
DUN
STX XDES
STA YDES
RTS

BANK

LDA #
STA X
LDA #
STA Y

LDX #534
LDA #5C0
BNE DUN

HEOWN

LDA CNTYPE
CMP #510
BEQ PICKUP

OT BANK

LDA #
STA X
LDA #
STA

LDX #5B0
LDA #5B3
BNE DUN

TABLE

.AS 0E0D0B07
.AS 060A0509


```

LDY #0
JSR GO
LDY #2
JSR GO
RTS

```

```

BUT0
BUT2
BUTIM0
BUTIM2

```

GO

```

LDA RANDM
AND #50F
BNE BU
JSR FINES
JSR SETK

```

BUT

```

LDA #0
LDX #0
CPX #50F
LDA RANDM

```

```

AND #57F
LDX STK0
CPX #50F
AND #5IF
BEG PRESS

```

BRTS RTS
PRESS

```

LDA BUT0
BEQ BRTS
LDA #0
STA BUT0
LDA #60
STA BUTIM0

```

```

LDA BUTIM0
BEQ TPRES
DEC BUTIM0
BNE BRTS

```

1 sec ave

2 sec ave

```

BEG PRESS
BNE BRTS

```

PRESS BUTTON

BUT0 ← 0

BUTIM0 ← 60

DEC BUTIM0
TO 0

1 → BUT0

BUT

LDA BUTIM0, Y

~~BEQ TPRES~~

~~DEC BUTIM0, X~~

~~BRTS RTS~~

~~LDA #1~~

~~STA BUT0~~

TPRES

LDA #1

STA BUT0, Y

LDA RANDM

AND #57F

BEQ PRESS

LDX STK0, Y

CPX #50F

BNE BRTS

AND #5IF

BEQ PRESS

BRTS RTS

PRESS

~~LDA #0~~

STA BUT0, Y = 0

LDA #60

STA BUTIM

RTS


```

INIT EQ
START EQ $28C0
FLAG, EQ $400B
; DONT FORGET TO
; CHANGE CHKSUM/CHKOR
OR $1000
TA $3000
JSR INIT $2A98
LDA $60A
BEQ ST
LDA #0

```

```

STA $600
ST STA $400B LDA #SPF
FLOOP JSR START STA $
JSR MOV0 3600
JSR CANON 1BC0
JSR XPYPS 2663
JSR PUL 22D0
JMP FLOOP CRASH

```

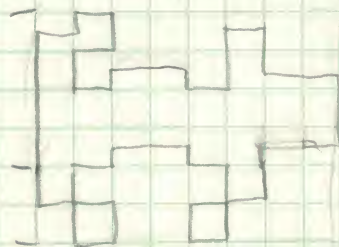
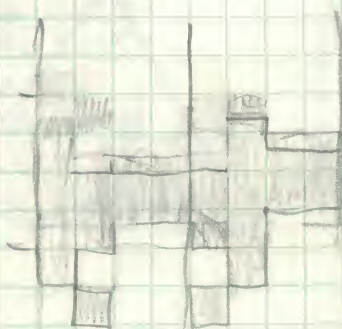
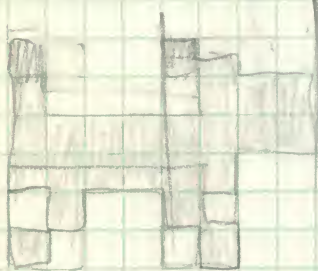
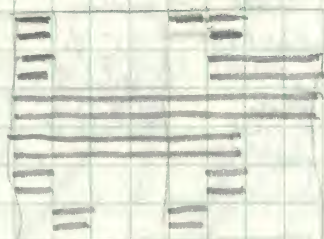
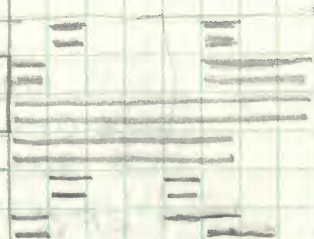
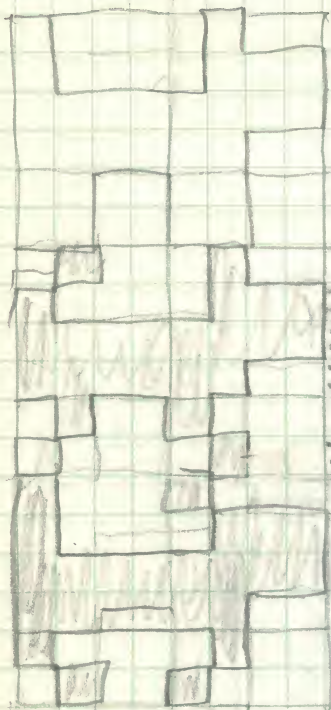
ZFC

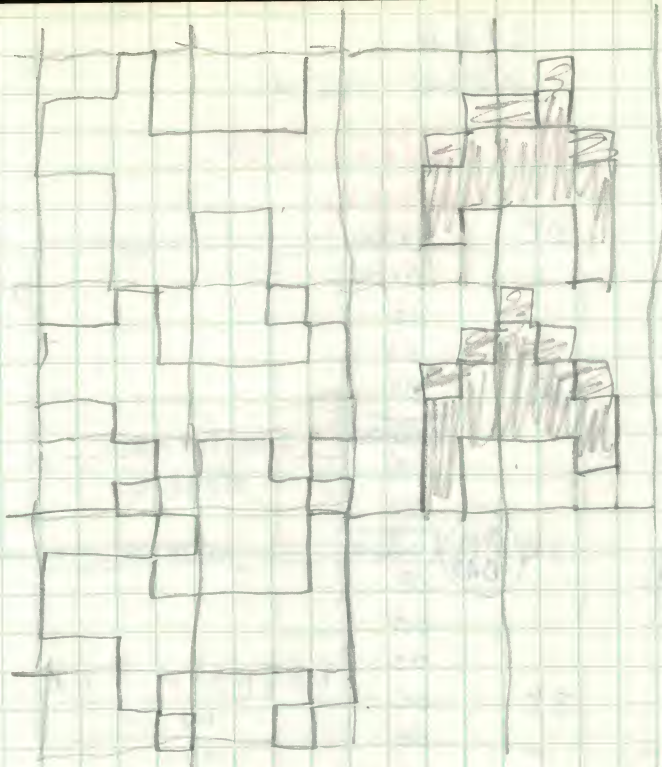
ENDFR

```

LDA $
CMP # $
BNE
PLA
PLA
JMP $0700

```



Handwritten notes in red ink, mostly illegible due to fading. Some visible words include "LARGE", "SMALL", "TOTAL", and "PERCENT".

⑦

0000 11010

IN MAIN

1 START-ENTER
 2 MUSIC LDX RANDOM
 3 START STX SCRL+4
 4 START LDA RANDOM
 5 CLAIM CMP #03
 6 SMO BCC G
 7 CTRM
 8 CTRM
 9 CTRM
 10 CANROW3
 11 SOUND-CHKOR
 12 MAIN Late
 13 START

ORA #3
 STX SCR4+4
 LDA RANDOM

IN CANROW

LDY RANDOM

STA (SCRL+4) X

USE 80-8F
 90-9F

IN SOUND

LDX #2
 LDA #5
 STA #5
 LDA #5
 STA #5

INC \$XXXX, X

DEC \$XXXX, X

ASL \$XXXX, X
 LSR

DEC \$XXXX, X

ROL \$XXXX, X
 ROR

TRAK

7 6D-7E
 8 7F-80
 9 81-A2
 10 A3-B4
 11 B5-C6
 12 C7-D8
 13 C7, C8, C9, CA, CB, CC

96 sectors

9000-5FFF

6D thru CC

96 sectors

OK

1K

2K

3K

4K

5K

6K

7K

8K

0000
 0100
 02
 03
 04
 0500
 06
 07
 0800
 09
 0A
 0B
 0C
 0D
 0E
 0F00
 1000
 1100
 1200
 13
 1400
 15
 16
 17
 1800
 19
 1A00
 1B
 1C
 1D
 1E
 1F00
 2000

FREQ.DAT-MUSIC

PAGE 6

DLIST-RAM

800-82F7

M3 M2 M1 M0

P0

P1

P2

P3

MUSIC FREQ. TAB.

DLIST-RAM

SCREEN
 1BC8-1FFF

AB4D
ENTER 4BDA

3XXX 4XXX 5XXX
→ 9XXX → AXXX → BXXX
add 6XXX

470E 30-40

20	
24	
28	
2C	
BAS 30/90	
31	MUSIC.DAT 3100 320F
32	MUSIC PROGR 3210 3210
33	CLAIM.CHAR 3300 33FC
34/94	CLAIM 3408-34A6
35	CLAIM.DAT 34B0-34FF
36	TITLES 3500-35CB
37	DLT ROW 35E0-35F3
38/98	INIT POT 3600-369F
39	CHAR.LET 36A8-36FF
3A	INIT SCR
3B	VOL 39D0-39F9
3C/9C	SOUND 3A00-3A8D
3D	3AA0 3BB1 CREAT 7AA0
3E	3BC0 9BC0 CANROM 9FCF
3F	header 9FD4
40/A0	

40/A0	
41	BUL 4000
42	43FB
43	+protec
44/A4	2 instr code
45	A400 SMOO 4400
46	46F3
47	4 instr + protec code
48/A8	FREQ 4700
49	START 4820
4A	→ 4FA2
4B	+ ~40 bytes
4C/AC	protec!
4D	
4E	
4F	
BAS 50/B0	
51	CHAR 5000 53FE
52	
53	
54/B4	POINTS 5400 54FE + 6 bytes
55/B5	ordtec
56	BUYS
57	PB-MB
58/B8	MAIN 5600
59	DO ANIMAT
5A	
5B	
5C/BC	removed 5663 NOPs
5D	B663
5E	5B80 5B80 (39 ms)
5F	
60/C0	INIT TOP 5F80 5F80
	NEW LOC B8D0 B8FF

FILES C: LIKE FILES X except updated
FILES D: put screen down low in memory
then put P/M
then put dlist & music table

insert copy protec code (disabled)
assemble high in memory - change links

each file.

check

\$3

\$4

\$5

=

SOUND

disk x — BLOAD "D: CLAIMW.OBJ"

startx .OR \$48D0 ^{MAIN} 1000<3000.5FFF m

mainx 5600

entry 42D0

18D0

pointx 5400

startx 48D0 ^{TSR} 2000

canonx 3BC0

entryx 47D0
35E0

smovx 3FD0

CLAIMW.OBJ

claimx 3408 — JMP 1411

creatx 3AA0 — JMP 1B32
1000

soundx 3A00

9820
START

SUPPLY

NEED

TBASE ✓
CACPOS ✓
DATA ✓

STCOL AADF

SMASHØ

(SCDATA) F9

SMASH2

STRTES A85C

MOVØ ✓

OPITES A8BD

MOV2 ✓

HSBAS ✓

CHKOK ABØZ

MOVZ ✓

CREATE ✓

SMOOTH ✓

CANROUV ✓

BUL ✓

SOUND ✓

RANDSC ✓

ENDFR ✓

PØ6BUL ✓

P26BUL ✓

POINTS ✓

FINDSC ✓

MUSIC etc ✓
CLAIM ✓
COPYRT ✓

MAIN ✓
5600

SUPPLY

NEED

DATA

MOVØ B600

SMOO

MAN

MOV2 B82C

CANROU

FLESH

JMOVZ

WIDENØ

:=

→ MOVZ B9C7

WIDEN2

CREAT

AØØØ
BUL ✓

SUPPLY

NEED

WIDENØ A393

FINDSC

WIDEN2 A3BI

POINTS ✓
B400

NEED

ENDFR

STRTES

OPITES ✓

SOUND ✓

?

9BCØ
CANROU ✓

SUPPLY

NEED

CANS

PØ6BUL 9E8F

P26BUL 9E81

SMOO ✓
A400

SUPPLY

FINDSC A560

SMASHØ A6B2

SMASH2 A6BB

9408

CLAIM

9408

NEED

SCDATA

STCOL

9AAØ
CREAT ✓

SUPPLY

RANDSC 9B71

9210
MUSIC

TABLE
DATAH1

L1

L2

L3

SOUND ✓

VOL ✓

FREQ ✓

9AØØ

OK.

05330 TSHT0
05340 LDA #000C ;P0PM
05350 AND #04 ;P0F2
05360 BNE NOSHT0
05370 LDA #0004 ;P0PF
05380 AND #04 ;P0PF2
05390 BNE NOSHT0
05400 LDA STUK0
05410 BEQ SHOOT0 ;UNC
05420 NOSHT0
05430 LDA #0
05440 STA STBUL0
05450 PLA
05460 PLA
05470 JMP NDRP0 ;NO SHT
05480 SHOOT0
05490 RTS
05500 TSHT2
OK.

LDA STUK0
BNE NDRP0 ;STUK
LDA #0004
AND #4 ;P0PF2
BEQ SHOOT0
LDA XP0
CMP #549
BCC NDRP0 ;LEFT HOS
CMP #5
BCS NDRP0 ;RT HOS
SHOOT0

00010 ;BULZ.SRC 7/01/82

OK.

00010 ;BULY.SRC 7/01/82

8527: AD F3 06	05662 COLOK
8529: D0 2B	05668 LDA #6F3 ;=GAMENO
8530: AA	05670 BNE ASBEF
852D: A1 FA	05672 TAX
852F: EA	05674 LDA (SCRL,X) ;HEAD
8530: C9 2C	05676 NOP
8532: B0 0F	05678 CMP #20 ;TUM/SNK
8534: AE 8C 06	05680 BCS TESTUM
8537: D0 1E	05682 LDX BHITFL
8539: 48	05684 BNE ASBEF
853A: 20 C5 65	05685 PHA
853D: 68	05686 JSR QSN1
853E: 18	05687 PLA
853F: 69 20	05688 CLC
8541: D0 0D	05689 ADC #20
	05690 BNE ENCONV ;UNC
8543: AE 8C 06	05692 TESTUM
8546: F0 0F	05694 LDX BHITFL
8548: 48	05696 BEQ ASBEF
8549: 20 D0 65	05697 PHA
854C: 68	05698 JSR QSN2
854D: 38	05699 PLA
854E: E9 20	05700 SEC
	05701 SBC #20

OK.

#6589

25993 WIDENØ

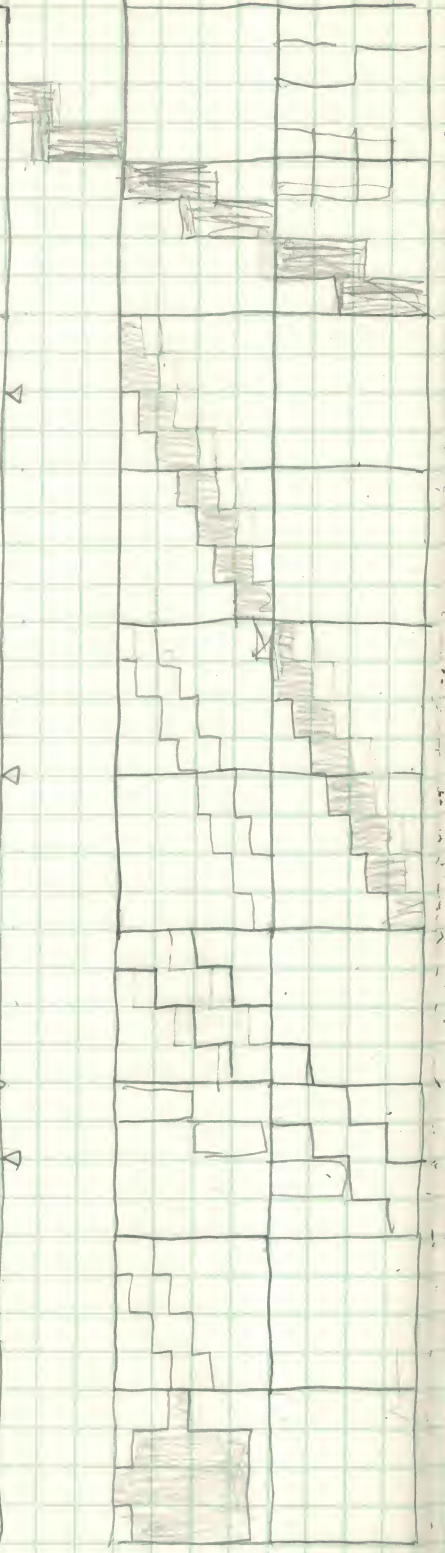
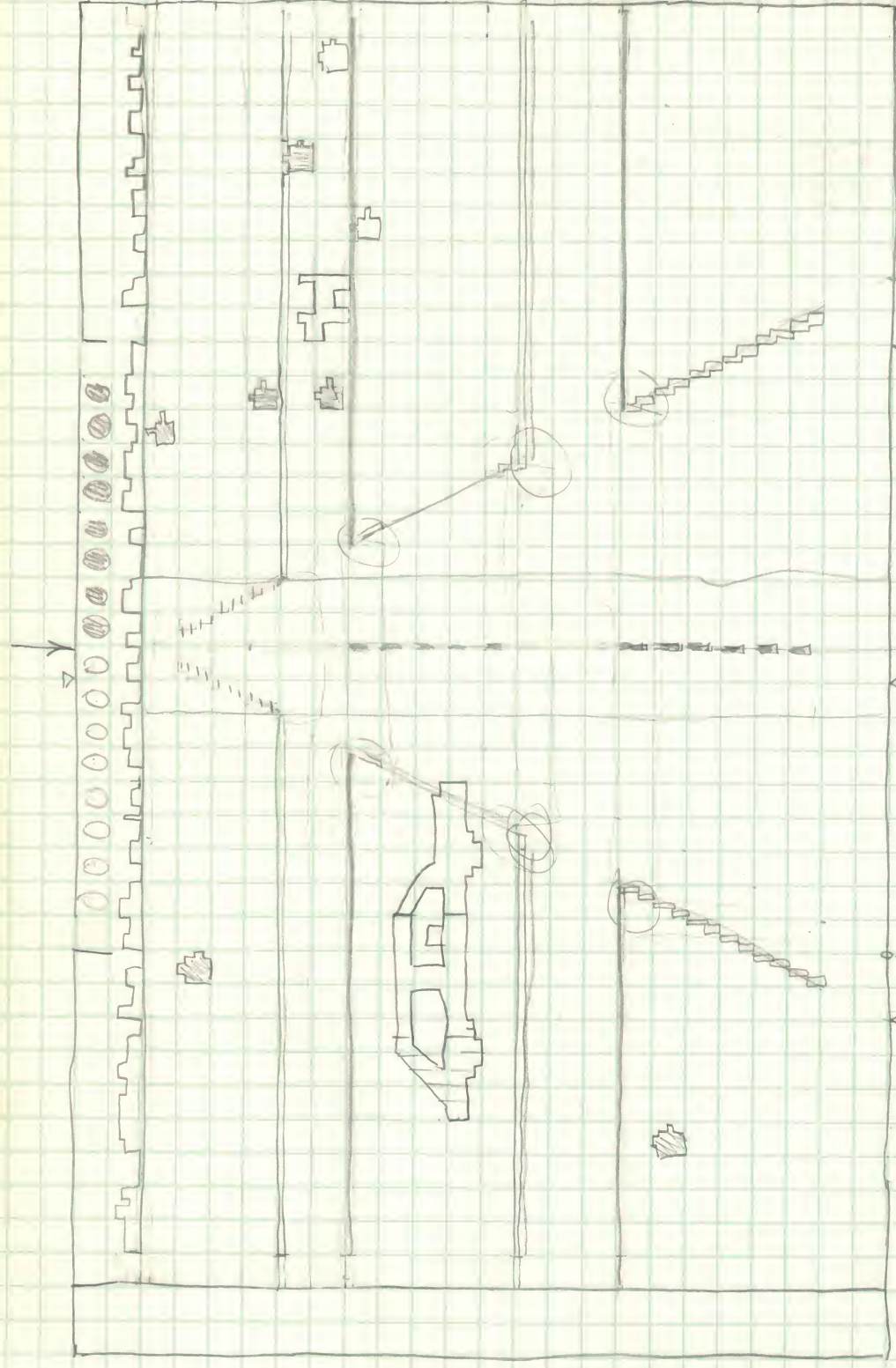
OK.

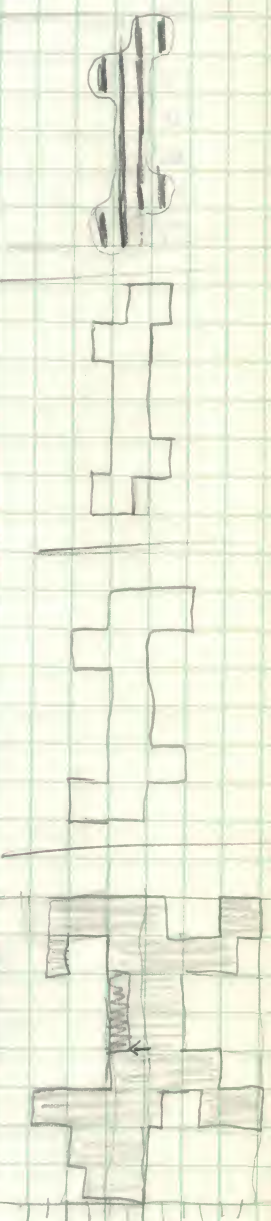
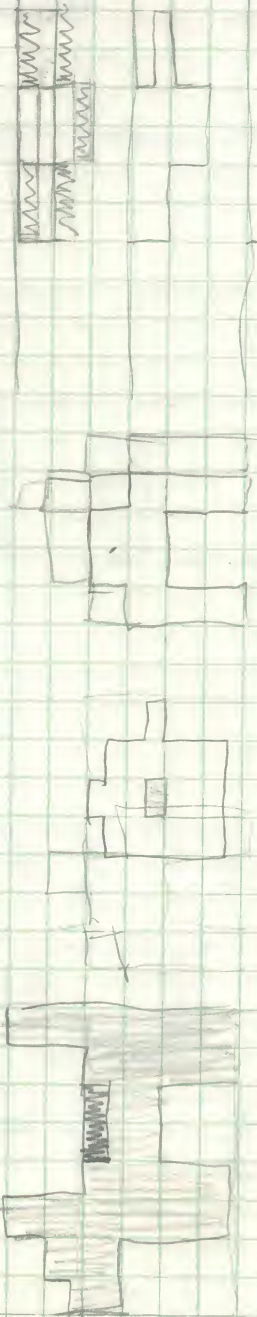
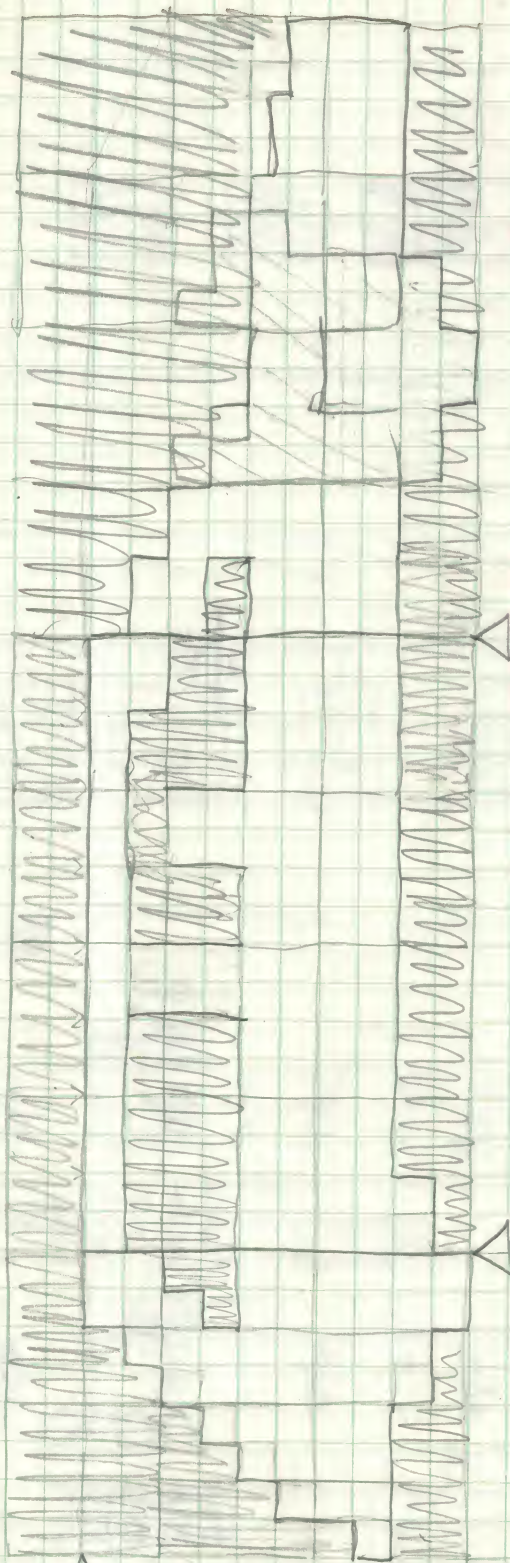
#65A7

26023 WIDENZ

OK.

70 70 70 41 20 30 41 20 30





XP0 } BONE
 YP0 } BONE
 XP1 } DOG1
 YP1 } DOG1
 XP2 } DOG2
 YP2 } DOG2
 XP3 } DOG3
 YP3 } DOG3

SCRL
 SCRL+2
 SCRL+4

TEST-BONE HYDRANT
 UPDATE DOG STATUS, POSITION
 UPDATE BONE STATUS, POSITION
 CAR ROW

P02 SPOT 00
 QUES 01
 BONE 80
 CAR FF

	2	3	4	5	6	7	8	9	10	11
10							3	1	-4	-4
11							6	4	3	1

12	-3	-2	0	2	1					
13	0	1	3	5	4					
14	3	4	5							
15	6							-4		
16							-5	0		

22										
33										
66										
77										
99										



11

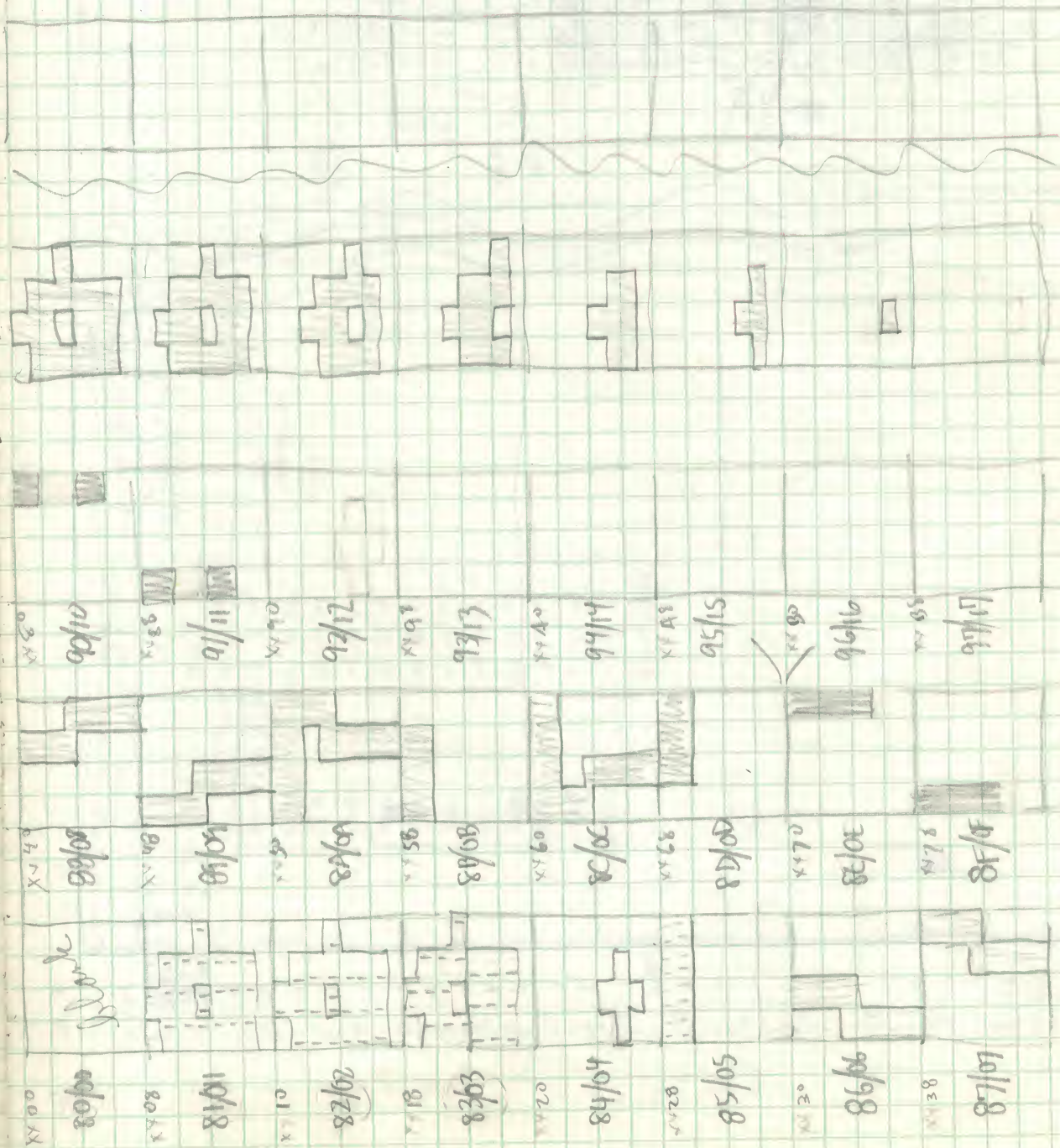
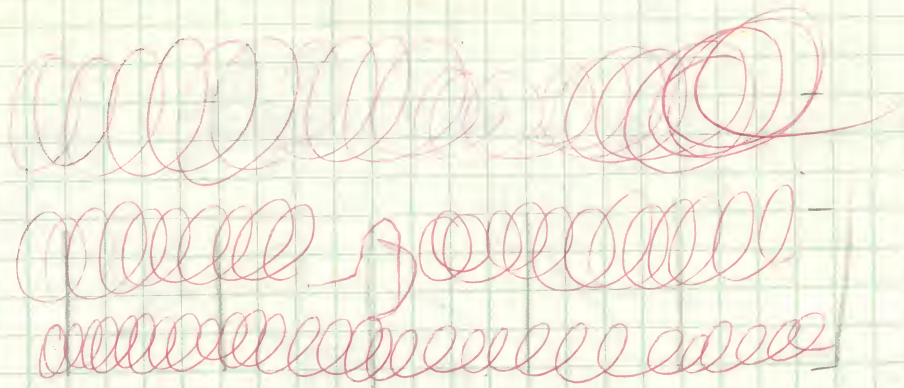
black PL1 - PL0
yellow PL3 - PL2
red
grey
blue

FF

FF

bak PF0 PF1 PF2 PF3

00 01 10 11 11



[illegible]

CD LOC
GE PAGE

LOAD DIS IN
ALL SCREEN MEM
2C40-3CFF

LOAD OF INTO
3C00 → 3C1F

LOAD H4 INTO
3C23

LOAD H4 INTO
3C26 → 3C3C

PUT 8 INTO
3C40

8 2 3
3C41, 3C42, 3C67

CALL
ADD 40 TO
LOC

CARRY TO
PAGE

PAGE OVERFLOW
TO 540?

PUT H1 INTO
3C42-3C67

PUT 4 INTO
3F0A-3FFF

PUT 4 INTO
3C42, 3C5D

PUT A INTO
3C4C, 3C67

PUT B INTO
3FDA

PUT C INTO
3FFF

6 → 3C4D
5H

7 → 3C95
5C

SET PF
COLORS

CHARAC.
DATA
AS PROGRAM

P = 3800
BYTE 0,0,0,0,0,0
BYTE

ENDPR ROUTINE

IS(SCLR) Y
? SD
N C=0

ENDPR
8 → DOIF
RET

FRAME
LSB
= 0

LDA(SCLR), X
TAX
DAX
TXG

STA(SCLR), X

LDA FRAME
MASA

X X

POPUP
X = 30C00

LDA FRAME
AND H1
BEQ ENDPOP
LDX #0
LDA(SCLR), X
BNE CMP #8E
CMP #8E
BEQ ENDPOP
BEQ POPDOWN

FRAME = \$641?

ENDPOP
STA(SCLR), X
LDA #8
STA \$DOIF
BNE BLINKH ; UNC

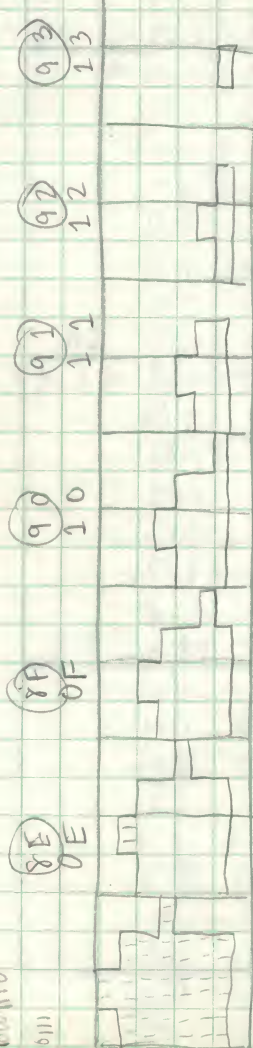
POPDOWN
LDA #87
STA(SCLR), X
BNE ENDPOP ; UNC

BLINKH
LDA FRAME
AND #53F, FREQ
CMP #5B, POPR.BLINK

BES RETPOP
LDA(SCLR), X
CMP #5A
BES RETPOP

LDA FRAME
AND #53F, FREQ
CMP #51F
BCC LIGHT

LDA #86
BNE COLH UNC
LIGHT LDA #5FA
COLH STA \$2C7
RETPOP RTS



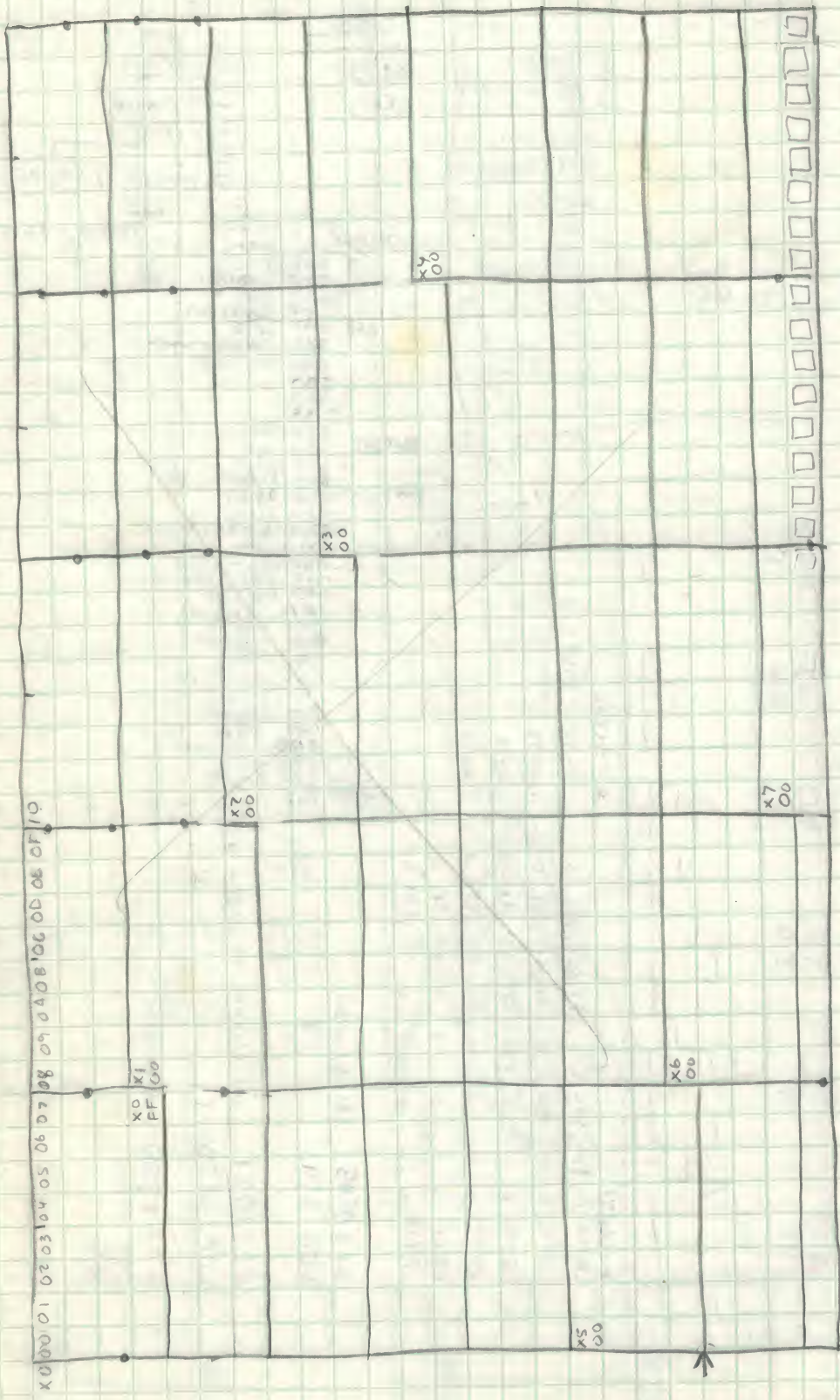
* = \$58B0
BYTE DISP LIST

* = \$4871
BYTE 0, \$30, \$FC, \$FF, \$FC, \$FC, 0
BYTE 0, 0, \$30, \$FC, \$FC, \$FF, \$FC, 0
BYTE 0, 0, 0, \$30, \$FC, \$FC, \$FF, 0
BYTE 0, 0, 0, 0, \$30, \$FC, \$FC, 0
BYTE 0, 0, 0, 0, 0, \$30, \$FC, 0
BYTE 0, 0, 0, 0, 0, 0, 0, \$30, 0

* = \$0F45 ; IN RANDOM
LDA #593

* = \$1021 ; IN CHANGE TO MODE 4 DISP
LDA \$18B0, Y

* = \$17BD ; IN ENDER
JSR POPUP
NOP
NOP



LP

A #1.79

B cheap red

E cheap black

H cheap black

K memory

L memory

O Maxwell

M

N memory

Q (CASS)

A BCD EFG HIJ K L

8/29/81 F2 screen setup - to append to main progr.
8/30/81 G2 DOG22 was must of dog program - no dog for jets, disp 0-99, 5min
F2 DOG22 record again for backup **BAD COPY**

8/31/81 F2 DOG23 try tape again - minor bugs fixed

9/3/81 D1 SCREEN DATA - works, to combine w/ 0-59 30

9/8/81 G2 DOG24 screen setup + dog PM every thing, DIAG DELAY 0-142
H1 DOG25 jet, (st all) must use 0-165 200

9/9/81 I2 DOG26 working jet + dog + random, debugger 0-165

9/10/81 J1 DOG27 2 working dogs, 1 jet, dog2 won't flash - printed

9/11/81 H1 DOG28 2 working, flashing dogs

9/12/81 F2 DOG2 JETS - Merge w/ DOG28 **lost!**
K1 DOG, 2 JETS, DIRECT CATCH, lines 0-3500 - minor goof
L1 same - backup - line 6280 - merge correctly; merge 0-165

9/13 G1 complete sound! append 0-58
F2 backup!

9/14 K1 DOG, JETS, SOUNDS 4 specs 0-18P
K2 flip side: backup!

H1 Object SAVE#C: <600,840 2000,20FF 3800,38FF
if sound trouble, C1FFD<8D;IE,2021 2-85 8-98

H2 - same - fix minor goofs

M1 toggle in window, flash - 1 line goof

N1 add DOG DARE SCREEN

M1 correct GOOFS

O1 STARTUP MUSIC HANDICAP, CLEAR ETC 0-16

P1 correct bugs, 0-16

9/23 N1 DOG DARE w/ music good 0-70

M1 ADD IN DISK LIST BYTES - 0-16

N1 MUSIC, TIME, HANDICAP (MAKE SETUP JMP -> 1C50)

M1 correct GOOFS

O1 STARTUP MUSIC HANDICAP, CLEAR ETC 0-16

P1 correct bugs, 0-16

7.7.7 K2 DOG, JETS SOUNDS w/ 16pt score, INTERRUPTING

N1 STARTUP FINAL HANDICAP, CLEAR ETC 0-16

M1 STARTUP NO SOUNDS - BO, BI - FINAL -> EXCEPT REP/SC/15C/4053

M1 STARTUP FINAL BACKUP w/ ALL CHANGES -> PROBLEM LOADING

O1 DOG, JETS NO LOGS, set

K2 SOUND, PAGE9

P1 DOG, JETS PITTER PATTEN

P2 DOG, JETS PITTER, DISK VERS.

M2 STARTUP DISK VERSION 6000 bugs

N1 STARTUP DISK GYCO bugs fixed

G2G1 OBJECT -> DISK both sides

M2 STARTUP, TEOTEXY CHANGES

M1 STARTUP, CASSETTE VERSION

TAPE RECORD

A1 DEMO

10: DEMO

35: CHINESE

CHARGE

8/23

DOG DISCS

B1

DIARIA

JSB

50: DOG2184

?

C1

DISQUANT MP

HARDIN

CO.

SIMPLE

WORTH

D1

DISK

DISK

DISK

DISK

E1

LOST!

LOST!

LOST!

LOST!

F2

DOG

DOG

DOG

DOG

G2

DOG

DOG

DOG

DOG

H1

DOG

DOG

DOG

DOG

I2

DOG

DOG

DOG

DOG

J1

DOG

DOG

DOG

DOG

K1

DOG

DOG

DOG

DOG

L1

DOG

DOG

DOG

DOG

M1

DOG

DOG

DOG

DOG

N1

DOG

DOG

DOG

DOG

O1

DOG

DOG

DOG

DOG

ASM	C32240	→ 40
DIAPY-	C36250	→ 30

Final

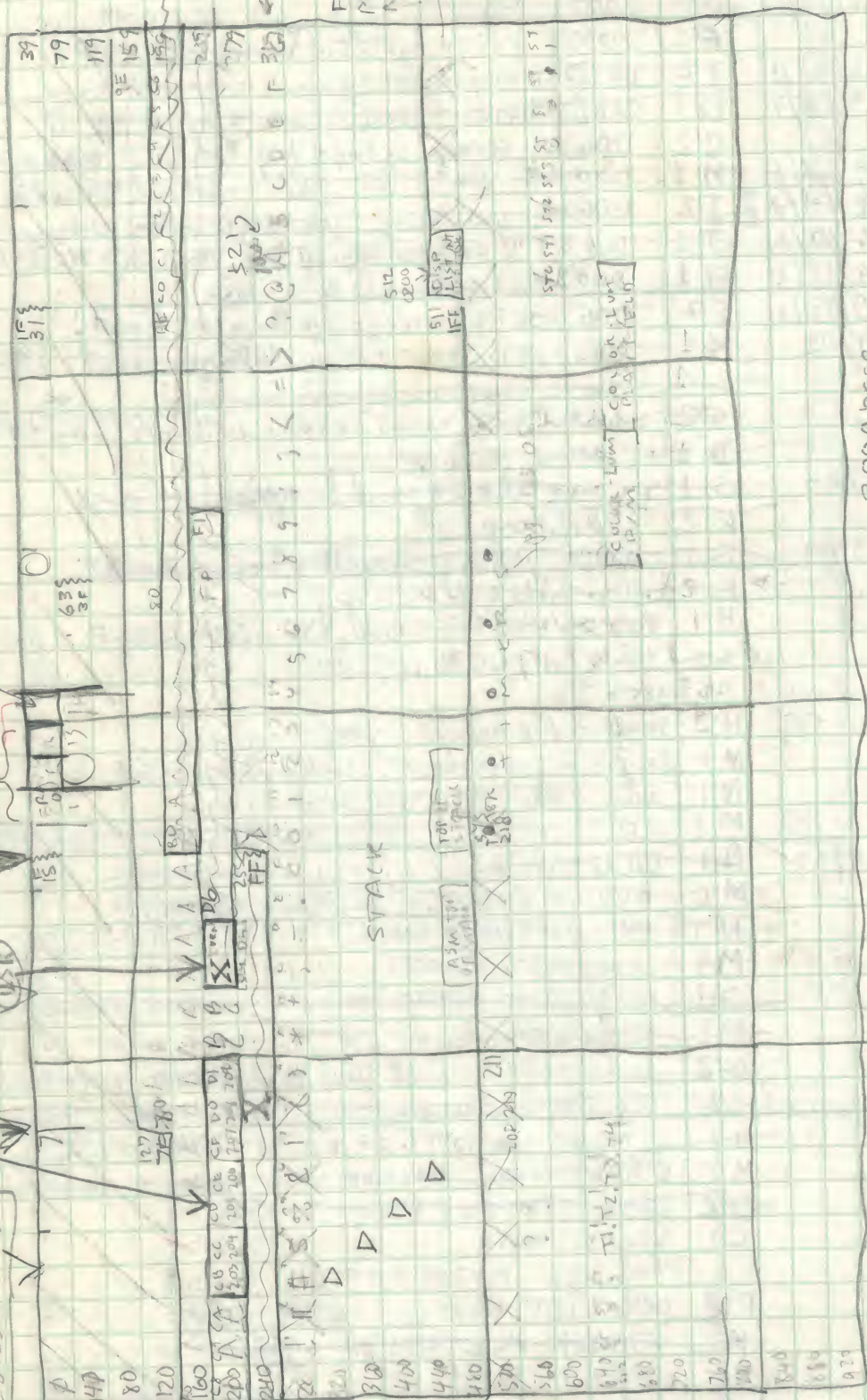
→ 40

2
④
③
②
①
40

12.5.17.67

41

$$d_{11} = \frac{1}{2} \left(\frac{1}{2} \sqrt{\frac{1}{2}} \right) \left(\frac{1}{2} \sqrt{\frac{1}{2}} \right) = \frac{1}{4}$$



FORI = 1704
POKE 2804, I
NEXT I

Sp. 7

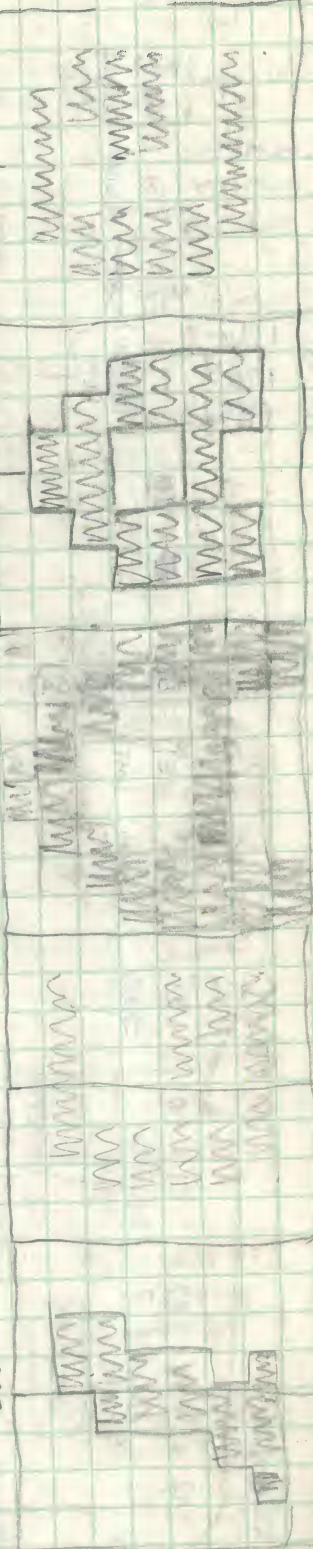
15 FEB 1951

11-15-25 145 725

color - lum color - lum
10/10 10/10

2009-09-28

2085



MEMLO 02E7 2
MEMTOP 02E5 2
APPMH# 000E 2
RAMTOP 006A 1
DAMSIZ 02E4 1

FILDAT
FILELG
NEWROW
HOLD4
ROWINC

1000 Paul Cabbage
E108 745-5525

~~3E00~~ 0 1 00 10 0001
~~10100~~ 0 1 00 10 0001
~~3E00~~ hy

C
CRSINH 02F0 1
ROWCRS 0054 1
COLCRS 0055 2
OLDROW 005A 1
OLDCOL 005B 2
TXTRON 0290 1
TXTCOL 2291 2
LMARON 0052 1
RMARON 0053 1

DOSTAR
DETAC
COUNTR
ROWAC
ENDPT

A 21 33
B 22 34
C 23 35
D 24 36
E 25 37
F 26 38
G 27 39
H 28 40
I 29 41
J 2A 42
K 2B 43
L 2C 44
M 2D 45
N 2E 46
O 2F 47
P 30 48
Q 31 49
R 32 50
S 33 51
T 34 52
U 35 53
V 36 54
W 37 55
X 38 56
Y 39 57
Z 3A 58

a 21 33
b 22 34
c 23 35
d 24 36
e 25 37
f 26 38
g 27 39
h 28 40
i 29 41
j 2A 42
k 2B 43
l 2C 44
m 2D 45
n 2E 46
o 2F 47
p 30 48
q 31 49
r 32 50
s 33 51
t 34 52
u 35 53
v 36 54
w 37 55
x 38 56
y 39 57
z 3A 58

PCOLRQ 02C0
PCOLRI 02C1
PCOLR2 02C2
PCOLR3 02C3

COLPM0 D012
COLPM1 D013
COLPM2 D014
COLPM3 D015

COLOR0 02C4
COLOR1 02C5
COLOR2 02C6
COLOR3 02C7
COLOR4 02C8

COLPF0 D016
COLPF1 D017
COLPF2 D018
COLPF3 D019
COLPF4 D01A

SCRFLG 02B8

ATTRACT 004D
COLRS4 004F
DRKMSK 004E

TABMAP
LOG MAP
LOG COL
BOTSCR

W	155	104	101	110	0	116	104	101	113
X	0	104	111	115	105	99	0	113	
Y	116	111	112	115	120	0	121	111	
Z	117	7	114	101	0	0	0	0	
A	114	101	97	100	121	0	116	111	
B	0	103	111	14	0	0	42	117	
C	115	116	0	109	111	118	101	0	
D	121	111	117	119	0	0	0	0	
E	106	111	121	115	116	105	99	107	
F	0	116	111	0	115	116	97	114	
G	116	0	116	104	101	0	97	99	
H	116	105	111	116	14	0	0	0	

01 23 45 67
0A BC DE FG
8 9 10 11
12 13 14 15
16 17 18 19
20 21 22 23
24 25 26 27
28 29 30 31
32 33 34 35
36 37 38 39
40 41 42 43
44 45 46 47
48 49 50 51
52 53 54 55
56 57 58 59
60 61 62 63
64 65 66 67
68 69 70 71
72 73 74 75
76 77 78 79
80 81 82 83
84 85 86 87
88 89 90 91
92 93 94 95
96 97 98 99
100 101 102 103
104 105 106 107
108 109 110 111
112 113 114 115
116 117 118 119
120 121 122 123
124 125 126 127
128 129 130 131
132 133 134 135
136 137 138 139
140 141 142 143
144 145 146 147
148 149 150 151
152 153 154 155
156 157 158 159
160 161 162 163
164 165 166 167
168 169 170 171
172 173 174 175
176 177 178 179
180 181 182 183
184 185 186 187
188 189 190 191
192 193 194 195
196 197 198 199
200 201 202 203
204 205 206 207
208 209 210 211
212 213 214 215
216 217 218 219
220 221 222 223
224 225 226 227
228 229 230 231
232 233 234 235
236 237 238 239
240 241 242 243
244 245 246 247
248 249 250 251
252 253 254 255
256 257 258 259
260 261 262 263
264 265 266 267
268 269 270 271
272 273 274 275
276 277 278 279
280 281 282 283
284 285 286 287
288 289 290 291
292 293 294 295
296 297 298 299
300 301 302 303
304 305 306 307
308 309 310 311
312 313 314 315
316 317 318 319
320 321 322 323
324 325 326 327
328 329 330 331
332 333 334 335
336 337 338 339
340 341 342 343
344 345 346 347
348 349 350 351
352 353 354 355
356 357 358 359
360 361 362 363
364 365 366 367
368 369 370 371
372 373 374 375
376 377 378 379
380 381 382 383
384 385 386 387
388 389 390 391
392 393 394 395
396 397 398 399
400 401 402 403
404 405 406 407
408 409 410 411
412 413 414 415
416 417 418 419
420 421 422 423
424 425 426 427
428 429 430 431
432 433 434 435
436 437 438 439
440 441 442 443
444 445 446 447
448 449 450 451
452 453 454 455
456 457 458 459
460 461 462 463
464 465 466 467
468 469 470 471
472 473 474 475
476 477 478 479
480 481 482 483
484 485 486 487
488 489 490 491
492 493 494 495
496 497 498 499
500 501 502 503
504 505 506 507
508 509 510 511
512 513 514 515
516 517 518 519
520 521 522 523
524 525 526 527
528 529 530 531
532 533 534 535
536 537 538 539
540 541 542 543
544 545 546 547
548 549 550 551
552 553 554 555
556 557 558 559
560 561 562 563
564 565 566 567
568 569 570 571
572 573 574 575
576 577 578 579
580 581 582 583
584 585 586 587
588 589 590 591
592 593 594 595
596 597 598 599
600 601 602 603
604 605 606 607
608 609 610 611
612 613 614 615
616 617 618 619
620 621 622 623
624 625 626 627
628 629 630 631
632 633 634 635
636 637 638 639
640 641 642 643
644 645 646 647
648 649 650 651
652 653 654 655
656 657 658 659
660 661 662 663
664 665 666 667
668 669 670 671
672 673 674 675
676 677 678 679
680 681 682 683
684 685 686 687
688 689 690 691
692 693 694 695
696 697 698 699
700 701 702 703
704 705 706 707
708 709 710 711
712 713 714 715
716 717 718 719
720 721 722 723
724 725 726 727
728 729 730 731
732 733 734 735
736 737 738 739
740 741 742 743
744 745 746 747
748 749 750 751
752 753 754 755
756 757 758 759
760 761 762 763
764 765 766 767
768 769 770 771
772 773 774 775
776 777 778 779
780 781 782 783
784 785 786 787
788 789 790 791
792 793 794 795
796 797 798 799
800 801 802 803
804 805 806 807
808 809 810 811
812 813 814 815
816 817 818 819
820 821 822 823
824 825 826 827
828 829 830 831
832 833 834 835
836 837 838 839
840 841 842 843
844 845 846 847
848 849 850 851
852 853 854 855
856 857 858 859
860 861 862 863
864 865 866 867
868 869 870 871
872 873 874 875
876 877 878 879
880 881 882 883
884 885 886 887
888 889 890 891
892 893 894 895
896 897 898 899
900 901 902 903
904 905 906 907
908 909 910 911
912 913 914 915
916 917 918 919
920 921 922 923
924 925 926 927
928 929 930 931
932 933 934 935
936 937 938 939
940 941 942 943
944 945 946 947
948 949 950 951
952 953 954 955
956 957 958 959
960 961 962 963
964 965 966 967
968 969 970 971
972 973 974 975
976 977 978 979
980 981 982 983
984 985 986 987
988 989 990 991
992 993 994 995
996 997 998 999
1000 1001 1002 1003
1004 1005 1006 1007
1008 1009 1010 1011
1012 1013 1014 1015
1016 1017 1018 1019
1020 1021 1022 1023
1024 1025 1026 1027
1028 1029 1030 1031
1032 1033 1034 1035
1036 1037 1038 1039
1040 1041 1042 1043
1044 1045 1046 1047
1048 1049 1050 1051
1052 1053 1054 1055
1056 1057 1058 1059
1060 1061 1062 1063
1064 1065 1066 1067
1068 1069 1070 1071
1072 1073 1074 1075
1076 1077 1078 1079
1080 1081 1082 1083
1084 1085 1086 1087
1088 1089 1090 1091
1092 1093 1094 1095
1096 1097 1098 1099
1100 1101 1102 1103
1104 1105 1106 1107
1108 1109 1110 1111
1112 1113 1114 1115
1116 1117 1118 1119
1120 1121 1122 1123
1124 1125 1126 1127
1128 1129 1130 1131
1132 1133 1134 1135
1136 1137 1138 1139
1140 1141 1142 1143
1144 1145 1146 1147
1148 1149 1150 1151
1152 1153 1154 1155
1156 1157 1158 1159
1160 1161 1162 1163
1164 1165 1166 1167
1168 1169 1170 1171
1172 1173 1174 1175
1176 1177 1178 1179
1180 1181 1182 1183
1184 1185 1186 1187
1188 1189 1190 1191
1192 1193 1194 1195
1196 1197 1198 1199
1200 1201 1202 1203
1204 1205 1206 1207
1208 1209 1210 1211
1212 1213 1214 1215
1216 1217 1218 1219
1220 1221 1222 1223
1224 1225 1226 1227
1228 1229 1230 1231
1232 1233 1234 1235
1236 1237 1238 1239
1240 1241 1242 1243
1244 1245 1246 1247
1248 1249 1250 1251
1252 1253 1254 1255
1256 1257 1258 1259
1260 1261 1262 1263
1264 1265 1266 1267
1268 1269 1270 1271
1272 1273 1274 1275
1276 1277 1278 1279
1280 1281 1282 1283
1284 1285 1286 1287
1288 1289 1290 1291
1292 1293 1294 1295
1296 1297 1298 1299
1300 1301 1302 1303
1304 1305 1306 1307
1308 1309 1310 1311
1312 1313 1314 1315
1316 1317 1318 1319
1320 1321 1322 1323
1324 1325 1326 1327
1328 1329 1330 1331
1332 1333 1334 1335
1336 1337 1338 1339
1340 1341 1342 1343
1344 1345 1346 1347
1348 1349 1350 1351
1352 1353 1354 1355
1356 1357 1358 1359
1360 1361 1362 1363
1364 1365 1366 1367
1368 1369 1370 1371
1372 1373 1374 1375
1376 1377 1378 1379
1380 1381 1382 1383
1384 1385 1386 1387
1388 1389 1390 1391
1392 1393 1394 1395
1396 1397 1398 1399
1400 1401 1402 1403
1404 1405 1406 1407
1408 1409 1410 1411
1412 1413 1414 1415
1416 1417 1418 1419
1420 1421 1422 1423
1424 1425 1426 1427
1428 1429 1430 1431
1432 1433 1434 1435
1436 1437 1438 1439
1440 1441 1442 1443
1444 1445 1446 1447
1448 1449 1450 1451
1452 1453 1454 1455
1456 1457 1458 1459
1460 1461 1462 1463
1464 1465 1466 1467
1468 1469 1470 1471
1472 1473 1474 1475
1476 1477 1478 1479
1480 1481 1482 1483
1484 1485 1486 1487
1488 1489 1490 1491
1492 1493 1494 1495
1496 1497 1498 1499
1500 1501 1502 1503
1504 1505 1506 1507
1508 1509 1510 1511
1512 1513 1514 1515
1516 1517 1518 1519
1520 1521 1522 1523
1524 1525 1526 1527
1528 1529 1530 1531
1532 1533 1534 1535
1536 1537 1538 1539
1540 1541 1542 1543
1544 1545 1546 1547
1548 1549 1550 1551
1552 1553 1554 1555
1556 1557 1558 1559
1560 1561 1562 1563
1564 1565 1566 1567
1568 1569 1570 1571
1572 1573 1574 1575
1576 1577 1578 1579
1580 1581 1582 1583
1584 1585 1586 1587
1588 1589 1590 1591
1592 1593 1594 1595
1596 1597 1598 1599
1600 1601 1602 1603
1604 1605 1606 1607
1608 1609 1610 1611
1612 1613 1614 1615
1616 1617 1618 1619
1620 1621 1622 1623
1624 1625 1626 1627
1628 1629 1630 1631
1632 1633 1634 1635
1636 1637 1638 1639
1640 1641 1642 1643
1644 1645 1646 1647
1648 1649 1650 1651
1652 1653 1654 1655
1656 1657 1658 1659
1660 1661 1662 1663
1664 1665 1666 1667
1668 1669 1670 1671
1672 1673 1674 1675
1676 1677 1678 1679
1680 1681 1682 1683
1684 1685 1686 1687
1688 1689 1690 1691
1692 1693 1694 1695
1696 1697 1698 1699
1700 1701 1702 1703
1704 1705 1706 1707
1708 1709 1710 1711
1712 1713 1714 1715
1716 1717 1718 1719
1720 1721 1722 1723
1724 1725 1726 1727
1728 1729 1730 1731
1732 1733 1734 1735
1736 1737 1738 1739
1740 1741 1742 1743
1744 1745 1746 1747
1748 1749 1750 1751
1752 1753 1754 1755
1756 1757 1758 1759
1760 1761 1762 1763
1764 1765 1766 1767
1768 1769 1770 1771
1772 1773 1774 1775
1776 1777 1778 1779
1780 1781 1782 1783
1784 1785 1786 1787
1788 1789 1790 1791
1792 1793 1794 1795
1796 1797 1798 1799
1800 1801 1802 1803
1804 1805 1806 1807
1808 1809 1810 1811
1812 1813 1814 1815
1816 1817 1818 1819
1820 1821 1822 1823
1824 1825 1826 1827
1828 1829 1830 1831
1832 1833 1834 1835
1836 1837 1838 1839
1840 1841 1842 1843
1844 1845 1846 1847
1848 1849 1850 1851
1852 1853 1854 1855
1856 1857 1858 1859
1860 1861 1862 1863
1864 1865 1866 1867
1868 1869 1870 1871
1872 1873 1874 1875
1876 1877 1878 1879
1880 1881 1882 1883
1884 1885 1886 1887
1888 1889 1890 1891
1892 1893 1894 1895
1896 1897 1898 1899
1900 1901 1902 1903
1904 1905 1906 1907
1908 1909 1910 1911
1912 1913 1914 1915
1916 1917 1918 1919
1920 1921 1922 1923
1924 1925 1926 1927
1928 1929 1930 1931
1932 1933 1934 1935
1936 1937 1938 1939
1940 1941 1942 1943
1944 1945 1946 1947
1948 1949 1950 1951
1952 1953 1954 1955
1956 1957 1958 1959
1960 1961 1962 1963
1964 1965 1966 1967
1968 1969 1970 1971
1972 1973 1974 1975
1976 1977 1978 1979
1980 1981 1982 1983
1984 1985 1986 1987
1988 1989 1990 1991
1992 1993 1994 1995
1996 1997 1998 1999
2000 2001 2002 2003
2004 2005 2006 2007
2008 2009 2010 2011
2012 2013 2014 2015
2016 2017 2018 2019
2020 2021 2022 2023
2024 2025 2026 2027
2028 2029 2030 2031
2032 2033 2034 2035
2036 2037 2038 2039
2040 2041 2042 2043
2044 2045 2046 2047
2048 2049 2050 2051
2052 2053 2054 2055
2056 2057 2058 2059
2060 2061 2062 2063
2064 2065 2066 2067
2068 2069 2070 2071
2072 2073 2074 2075
2076 2077 2078 2079
2080 2081 2082 2083
2084 2085 2086 2087
2088 2089 2090 2091
20

с 990 <

160) sek

50

69

1. a

2

1

(4)

2

11

Y

1

7

1

10

72

100

2

1

Align

left CTRL [

right]

center Y

Blank line J (not removed during formatting)

O (removed)

Block (define) N (start
M end)

change case C

delete line X

CURSOR beg. line A

end line E

l. margin G

r. margin H

top page T

bot page B

end ;

DELETE

next char shift
backsp

prev char backsp

word W

prev word P

line X

rest of line F

fixed line terminator J

WINDOW

MOVE left L

right R

down D

up U

SCROLL

up 20 V

down 20 Q

left 20 TAB

right 20 SHIFT TAB

CANCEL

command Z

NUMBER	SHEET COUNT	RULING
53-008	60	WIDE
53-108	60	5 x 5 QUAD
53-208	60	PLAIN
53-010	96	WIDE
53-110	96	5 x 5 QUAD
53-210	96	PLAIN
53-012	96	WIDE & MARGIN