

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
25-Jul-83/Ksk

!!!!!!  
DUNGEONS OF DAGGORATH  
!!!!!!

Copyright (c) 1983 - Unified Technologies, Inc.  
12479 Walsh Avenue  
Los Angeles, Calif. 90066  
Tel. (213) 822-9989

### WARNING !!!

No part of this work - documents, schematic diagrams, drawings, printed circuit board artwork, program source listings or object code, may be reproduced by any mechanical, photographic, or electronic process, or in the form of a phonographic recording, nor may it be stored in a retrieval system, transmitted, or otherwise copied for public or private use, without permission from Unified Technologies, Inc.

This software [and manual] are both protected by U.S. Copyright Law (Title 17 United States Code). Unauthorized reproduction and/or sales may result in imprisonment of up to 1 year and fines of up to \$10,000 (17 USC 506). Copyright infringers may also be subject to civil liability.

All graphic designs and displays, technical manuals and accompanying schematic diagrams are protected by the U.S. Copyright Act of 1976.

This Act provides for increased penalties for violating federal copyright laws. Courts CAN IMPOUND infringing articles while legal action is pending. If infringers are convicted, courts can ORDER DESTRUCTION of the infringing articles.

In addition, the Act provides for payment of statutory damages of up to \$50,000 in certain cases. Infringers may also have to pay costs and attorney's fees, fines of up to \$25,000, and face an imprisonment of up to one year.

Unified Technologies, Inc. will aggressively enforce its copyrights against any infringers. WE WILL USE ALL LEGAL MEANS to immediately halt the manufacture, distribution, illegal use or operation of a copy of a hardware or software system made by us. Anyone who purchases such copies risks forfeiting such a copy.

HAVE FUN!

PAGE 1

# FINAL R.S. PRODUCTION ARCHIVE

A non-exclusive perpetual license is hereby granted to Michael J. Spencer Jr. to copy, replicate or emulate the source code contained herein, for the purposes of replicating the Dungeons of Daggorath game.

Mar. 7, 2002

*Douglas J. Morgan*  
Douglas J. Morgan  
President, DynaMicro Inc.  
Chairman, Unified Technologies

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
25-Jul-83/Ksk

PAGE 2

Acknowledgements:

This program would not have been possible without the help of Doug, Phil, April, and Jim. After almost two years, it may finally get out in cartridge form...

Thank you, thank you, thank you!

Keith S. Kiyohara  
July 20, 1983

# DUNGEONS OF DAGGORATH

---

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
25-Jul-83/Ksk

PAGE 3

Operator synonyms

FCB	OPSYN	DB
FDB	OPSYN	DW
FCC	OPSYN	DB
RMB	OPSYN	DS

include all game files

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 4

```
INCLUD B:CD.ASM
!!!!!! COMMON DEFINITIONS !!!!!!
"CMFX" opcode
008C SKIP2 EQU $8C
      Bit Position Mnemonics
0001 BIT0 EQU Z00000001
0002 BIT1 EQU Z00000010
0004 BIT2 EQU Z00000100
0008 BIT3 EQU Z00001000
0010 BIT4 EQU Z00010000
0020 BIT5 EQU Z00100000
0040 BIT6 EQU Z01000000
0080 BIT7 EQU Z10000000
0000 DEBFLG EQU 0           $debug flag.
```

## ; ASCII Character Mnemonics

0000	C,NUL	EQU	\$00
0001	C,SOH	EQU	\$01
0002	C,STX	EQU	\$02
0003	C,ETX	EQU	\$03
0004	C,EOT	EQU	\$04
0005	C,ENQ	EQU	\$05
0006	C,ACK	EQU	\$06
0007	C,BEL	EQU	\$07
0008	C,BS	EQU	\$08
0009	C,HT	EQU	\$09
000A	C,LF	EQU	\$0A
000B	C,VT	EQU	\$0B
000C	C,FF	EQU	\$0C
000D	C,CR	EQU	\$0D
000E	C,SO	EQU	\$0E
000F	C,SI	EQU	\$0F
0010	C,DLE	EQU	\$10
0011	C,DC1	EQU	\$11
0012	C,DC2	EQU	\$12
0013	C,DC3	EQU	\$13
0014	C,DC4	EQU	\$14
0015	C,NAK	EQU	\$15
0016	C,SYN	EQU	\$16
0017	C,ETB	EQU	\$17
0018	C,CAN	EQU	\$18
0019	C,EM	EQU	\$19
001A	C,SUB	EQU	\$1A
001B	C,ESC	EQU	\$1B
001C	C,FS	EQU	\$1C
001D	C,GS	EQU	\$1D
001E	C,RS	EQU	\$1E
001F	C,US	EQU	\$1F
0020	C,SP	EQU	\$20
007F	C,DEL	EQU	\$7F

## Internal Character Codes

0000	I.SP	EQU	\$00	; <SP> space
001B	I.EXCL	EQU	\$1B	; exclamation point
001C	I.BAR	EQU	\$1C	; underline
001D	I.QUES	EQU	\$1D	; question mark
001E	I.DOT	EQU	\$1E	; period
001F	I.CR	EQU	\$1F	; carriage return
0020	I.SHL	EQU	\$20	; small heart (left)
0021	I.SHR	EQU	\$21	; small heart (right)
0022	I.LHL	EQU	\$22	; large heart (left)
0023	I.LHR	EQU	\$23	; large heart (right)
0024	I.BS	EQU	\$24	; <BS> backspace

## Display Area RAM Definitions

1800	G6.LEN	EQU	6144	; length of Graphics-6 display buffers
1000	D0\$BAS	EQU	\$1000	; Resolution Graphics-6 Display #0
1300	D0.LEN	EQU	32*19*8	; 32 bytes * 19 lines * 8 scan lines
2300	D0\$END	EQU	D0\$BAS+D0.LEN	; ending addr
2046	D0.SAM	EQU	\$2046	; magic SAM bits
2800	D1\$BAS	EQU	\$2800	; Resolution Graphics-6 display buffer
1300	D1.LEN	EQU	32*19*8	; 32 bytes * 19 lines * 8 scan lines
3800	D1\$END	EQU	D1\$BAS+D1.LEN	; ending addr
20A6	D1.SAM	EQU	\$20A6	; magic SAM bits!

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 7

!!!!!!  
V E C T O R   L I S T   C O M M A N D S  
!!!!!!

These are special command codes to the vector list driver:

00FF	V\$NEW	EQU	\$FF	;start new sequence
00FE	V\$END	EQU	\$FE	;end sequence
00FD	V\$JMP	EQU	\$FD	;chain to addr specified by next word
00FC	V\$REL	EQU	\$FC	;enter short vector mode
00FB	V\$JSR	EQU	\$FB	;jump to vector list subroutine
00FA	V\$RTS	EQU	\$FA	;return from vector list subroutine
0000	V\$ABS	EQU	\$00	;return to absolute mode

```
!!!!!!!!!!!!!! COLOR COMPUTER DEFINITIONS !!!!!!!
!!!!!!
```

```
, Condition Code Mnemonics
```

0080	CC.E	EQU	BIT7
0040	CC.F	EQU	BIT6
0020	CC.H	EQU	BIT5
0010	CC.I	EQU	BIT4
0008	CC.N	EQU	BIT3
0004	CC.Z	EQU	BIT2
0002	CC.V	EQU	BIT1
0001	CC.C	EQU	BIT0

```
, Interrupt Vector Addresses
```

0100	V\$SWI3	EQU	\$0100
0103	V\$SWI2	EQU	\$0103
0106	V\$SWI	EQU	\$0106
0109	V\$NMI	EQU	\$0109
010C	V\$IRQ	EQU	\$010C
010F	V\$FIRQ	EQU	\$010F

```
, BASIC ROM Routines
```

A000	POLCAT	EQU	\$A000
A002	CHRROUT	EQU	\$A002
A004	CSRDON	EQU	\$A004
A006	BLKIN	EQU	\$A006
A008	BLKOUT	EQU	\$A008
A00A	JOYIN	EQU	\$A00A
A00C	WRTLDR	EQU	\$A00C

```
, PIA Definitions
```

FF00	PIA\$0	EQU	\$FF00
FF20	PIA\$1	EQU	\$FF20
0000	P.PI1DA	EQU	0
0001	P.PICRA	EQU	1
0002	P.PI1DB	EQU	2
0003	P.PICRB	EQU	3

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 9

!!!!!! D U N G E O N D E F I N I T I O N S !!!!!!!

Horizontal Maze Features

0000	HF.PAS	EQU	0	open passage
0001	HF.DOR	EQU	1	regular door
0002	HF.SDR	EQU	2	secret door
0003	HF.WAL	EQU	3	solid wall

Vertical Maze Features

0000	VF.HUP	EQU	0	hole up
0001	VF.LUP	EQU	1	ladder up
0002	VF.HDN	EQU	2	hole down
0003	VF.LDN	EQU	3	ladder down

Object Classes

0000	K.FLAS	EQU	0	flask
0001	K.RING	EQU	1	ring
0002	K.SCRO	EQU	2	scroll
0003	K.SHIE	EQU	3	shield
0004	K.SWOR	EQU	4	sword
0005	K.TORC	EQU	5	torch

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 10

!!!!!! TASK CONTROL BLOCKS !!!!!!!

0000	OFF	SET	0	
0000	P.TCPTR	EQU	OFF	linked list pointer
0002	P.TCTIM	EQU	OFF+2	countdown timer
0003	P.TCRIN	EQU	OFF+3	task address
0005	P.TCIDA	EQU	OFF+5	data block pointer
0007	TC.LEN	EQU	OFF+7	length of TCB

Queue Disposition Codes

0000	Q.NUL	EQU	0	null queue
0002	Q.JIF	EQU	2	jiffy queue
0004	Q.TEN	EQU	4	tenth queue
0006	Q.SEC	EQU	6	second queue
0008	Q.MIN	EQU	8	minute queue
000A	Q.HOUR	EQU	10	hour queue
000C	Q.SCD	EQU	12	scheduler queue

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 11

```
!!!!!! TEXT DISPLAY CONTROL BLOCKS
!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!!
0000    OFF    SET    0
0000    P.TXBAS EQU    OFF
0002    P.TXCNT EQU    OFF+2
0004    P.TXCUR EQU    OFF+4
0006    P.TXINV EQU    OFF+6
0007    P.TXSGL EQU    OFF+7
0008    TX.LEN EQU    OFF+8      ,length of TXB
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 12

```
!!!!!!!!!!!!!! V E C T O R D I S P L A Y C O N T R O L B L O C K S !!!!!!!
0000    OFF      SET    0
0000    P.VDBAS EQU    OFF          ;base address
0002    P.VDEND EQU    OFF+2        ;end address
0004    P.VDSAM EQU    OFF+4        ;magic SAM bits
0006    VD.LEN   EQU    OFF+6        ;length of VDB
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 13

```
!!!!!!ATTACK PARAMETER BLOCK!!!!!!
;ATTACK PARAMETER BLOCK
0000    OFF    SET    0
0000    P.ATPOW EQU    OFF      ;power
0002    P.ATMGO EQU    OFF+2    ;magic offense
0003    P.ATMGD EQU    OFF+3    ;magic defense
0004    P.ATPHO EQU    OFF+4    ;physical offense
0005    P.ATPHD EQU    OFF+5    ;physical defense
0006    P.ATXXX EQU    OFF+6    ;filler
000A    P.ATDAM EQU    OFF+10   ;damage
000C    AT.LEN  EQU    OFF+12   ;length of ATB
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 14

!!!!!! CREATURE CONTROL BLOCKS !!!!!!!

000C	C.TYPES	EQU	12	number of creature types
0000	OFF	SET	0	
0000	P.CCPOW	EQU	OFF	power
0002	P.CCMGO	EQU	OFF+2	magic offense
0003	P.CCMGD	EQU	OFF+3	magic defense
0004	P.CCPHO	EQU	OFF+4	physical offense
0005	P.CCPHD	EQU	OFF+5	physical defense
0006	P.CCTMV	EQU	OFF+6	movement delay time
0007	P.CCTAT	EQU	OFF+7	attack delay time
0008	P.CCOBJ	EQU	OFF+8	object list pointer
000A	P.CCDAM	EQU	OFF+10	damage
000C	P.CCUSE	EQU	OFF+12	"in-use" flag
000D	P.CCTYP	EQU	OFF+13	creature type
000E	P.CCIDIR	EQU	OFF+14	direction of travel
000F	P.CCROW	EQU	OFF+15	row
0010	P.CCCOL	EQU	OFF+16	column
0011	CC.LEN	EQU	OFF+17	length of CCB

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 15

```
;!!!!!! C R E A T O R E D E F I N I T I O N B L O C K S !!!!!!
;OFF      SET    0
0000     P.CDPOW EQU    OFF          ;power
0002     P.CDMGO EQU    OFF+2        ;magic offense
0003     P.CDMGD EQU    OFF+3        ;magic defense
0004     P.CDPHO EQU    OFF+4        ;physical offense
0005     P.CDPHD EQU    OFF+5        ;physical defense
0006     P.CDTMV EQU    OFF+6        ;movement delay time
0007     P.CDTAT EQU    OFF+7        ;attack delay time
0008     CD.LEN   EQU    OFF+8        ;length of CDB
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.0B

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 16

```
!!!!!! OBJECT CONTROL BLOCKS !!!!!!!
    OFF    SET    0
    P.OCPTR EQU    OFF      linked list pointer
    P.OCRROW EQU    OFF+2    row
    P.OCCOL  EQU    OFF+3    column
    P.OCLVL  EQU    OFF+4    level
    P.OCOWN  EQU    OFF+5    ownership byte
    P.OCXXX  EQU    OFF+6    temp bytes
    P.OCTYP   EQU    OFF+9    object type
    P.OCCLS   EQU    OFF+10   object class
    P.OCREV   EQU    OFF+11   revelation power requirement
    P.OCMGO   EQU    OFF+12   magic offense
    P.OCPHO   EQU    OFF+13   physical offense
    OC.LEN   EQU    OFF+14   length of OCB
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 17

```
!!!!!!  
!! OBJECT DEFINITION BLOCKS  
!!!!!!  
0000    OFF    SET    0  
0000    P.ODCLS EQU    OFF      ;object class  
0001    P.ODREV EQU    OFF+1    ;revelation power requirement  
0002    P.ODMGO EQU    OFF+2    ;magic offense  
0003    P.ODPHO EQU    OFF+3    ;physical offense  
0004    OD.LEN EQU    OFF+4    ;length of ODB
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 18

```
!!!!!! DIRECT PAGE DEFINITIONS !!!!!!!
!!!!!! !!!!!!!
```

0200	ORG	\$0200	;where our DP resides
0002	SETDP	\$02	;tell the assembler
0200	DP.BEG	EQU	\$ ;first location in DP
	RAM	Initialized by ONCE	
0200	BIGZER	RMB	1 ;overlapped 16-bit zero
0201	BIGONE	RMB	2 ;overlapped 16-bit one
0203	NEGONE	RMB	3 ;16-bit negative one
0205	UCNTRX	RMB	4 ;X-coordinate of centroid
0207	VCNTRY	RMB	2 ;Y-coordinate of centroid
0209	FLIP	RMB	2 ;pointer to VDB on display
020B	FLOP	RMB	2 ;pointer to VDB being updated
020D	AUTPTR	RMB	2 ;index into AUTTAB
020F	OCBPTR	RMB	2 ;index into OCBLND
0211	LINPTR	RMB	2 ;index into LINBUF
0213	PCROW	RMB	1 ;player position
0214	PCCOL	RMB	1
0215	POBJWT	RMB	2 ;weight of all objects
0217	PLRBLK	EQU	\$ ;player ATB base addr
0217	PPOW	RMB	2 ;power
0219	PMGO	RMB	1 ;magic offense
021A	PMGD	RMB	1 ;magic defense
021B	PPHO	RMB	1 ;physical offense
021C	PPHD	RMB	1 ;physical defense
021D	PLHAND	RMB	2 ;left hand
021F	PRHAND	RMB	2 ;right hand
0221	PDAM	RMB	2 ;damage
0223	PDIR	RMB	1 ;player facing direction
0224	PTORCH	RMB	2 ;torch in use
0226	PRLITE	RMB	1 ;regular light
0227	PMLITE	RMB	1 ;magic light
0228	FAINT	RMB	1 ;non-zero means we have fainted
0229	BAGPTR	RMB	2 ;what we are carrying
022E	FRZFLG	RMB	1 ;"freeze" all creatures flag

VDG (MC6847) Control RAM			
022C	VGINV	RMB	1
; Inverse Video Mode ; 0 = Light on Dark ; -1 = Dark on Light			
VECTOR Generator RAM			
022D	OCFTAD	RMB	1
022E	FADCNT	RMB	1
; amount of fade (\$FF = invisible) ; fade count (used by VECTOR)			
022F	Y0	RMB	2
0231	X0	RMB	2
0233	Y1	RMB	2
0235	X1	RMB	2
; starting Y ; starting X ; ending Y ; ending X			
0237	XX	RMB	3
023A	YY	RMB	3
023D	DX	RMB	3
0240	DY	RMB	3
; running-X coordinate ; running-Y coordinate ; running-X increment ; running-Y increment			
0243	LENGTH	RMB	2
0245	DELTAX	RMB	1
0246	DELTAY	RMB	1
0247	VBAS	RMB	2
0249	VBEND	RMB	2
; dot count ; display-X address change ; display-Y address change ; display base addr ; display end addr			
VCTLST (Vector List Processor) RAM			
024B	VXOFF	RMB	2
024D	VYOFF	RMB	2
024F	VXSCAL	RMB	1
0250	VYSCAL	RMB	1
; X-offset value ; Y-offset value ; multiplicative scale factor in X ; multiplicative scale factor in Y			
0251	DRWFLG	RMB	1
0252	TX	RMB	2
0254	TY	RMB	2
; flag indicates if starting (Y,X) set ; true X ; true Y			
Sound Generator RAM			
0256	SNDRAM	EQU	\$
0258	SNDRND	RMB	3
0259	SNSAV	RMB	3
025B	SNENV\$	RMB	2
025D	SNENV!	RMB	2
025F	SNSAVA	RMB	2
0261	SNVOL	RMB	1
0262	SNDELAY	RMB	1
0263	FREQ1	RMB	1
0265	FREQ2	RMB	1
0267	SNOTE1	RMB	2
0269	SNOTE2	RMB	2
; where sound RAM starts ; sound random number generator ; scratch area ; envelope generator start ; envelope generator delta ; alternate save location ; volume level ; RC circuit delay ; storage ; storage ; music note 1 ; music note 2			
Random Number Generator RAM			

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 20

026B        SEED        RMB        3                      ;General Purpose Random Number

## ; Dungeon Viewer RAM definitions

026E	RLIGHT	RMB	1	; regular light
026F	MLIGHT	RMB	1	; magic light
0270	DLIGHT	RMB	1	; old lighting value (fainting)
0271	DROW	RMB	1	; row
0272	DCOL	RMB	1	; column
0273	HLFSTP	RMB	1	; half-step forward flag
0274	BAKSTP	RMB	1	; half-step backward flag
0275	MAGFLG	RMB	1	; magic light flag
0276	PASFLG	RMB	1	; passage flag used for peek-a-boo
0277	AUTFLG	RMB	1	; auto-play flag

## ; Parser Ram

0278	PARFLG	RMB	1	; successful match flag
0279	PARNCT	RMB	1	; loop counter
027A	VERIFY	RMB	1	; verify on/off flag
027B	FULFLG	RMB	1	; full word match

## ; Dungeon Generator RAM definitions

027C	DROW	RMB	1	; row
027D	DCOL	RMB	1	; column
027E	DST	RMB	1	; distance to travel
027F	CELCNT	RMB	2	; number of cells created
0281	LEVEL	RMB	1	; maze level value
0282	CMXFTR	RMB	2	; index into CMXLND
0284	LVLPTR	RMB	2	; index into LVLVDG
0286	VFTPTR	RMB	2	; index into VFTTAB

## ; Dungeon Subroutine Parameters

0288	ROW	RMB	1	; row position
0289	COL	RMB	1	; column position
028A	DIR	RMB	1	; temp dir holder
028B	RANGE	RMB	1	; temp sight range holder

## ; Object Subroutine Parameters

028C	OBJCNT	RMB	1	; used by ONCE (object distribution)
028D	OBJLVL	RMB	1	; used by ONCE (object distribution)
028E	OBJTYP	RMB	1	; object type
028F	OBJCLS	RMB	1	; object class
0290	SPEFLG	RMB	1	; specific object flag
0291	OFINDF	RMB	1	; OCB table ptr "rewind" flag
0292	OFINDP	RMB	2	; OCB table ptr
0294	MAPFLG	RMB	1	; used by scrolls

```

; CLOCK RAM
0295  TIMBEG EQU   $      ; where the timers live
0295  JIFFY    RMB   1      ; JIFFY counter
0296  TENTH   RMB   1      ; TENTH counter
0297  SECOND  RMB   1      ; SECOND counter
0298  MINUTE  RMB   1      ; MINUTE counter
0299  HOUR    RMB   1      ; HOUR counter
029A  DAY     RMB   1      ; DAY counter
029B  TIMEND EQU   $      ; end of timers

029B  SLEEP    RMB   1      ; system SLEEP counter
029C  NOISEF   RMB   1      ; noise flag
029D  NOISEV   RMB   1      ; noise value
029E  FADFLG   RMB   1      ; wizard fade-in flag

; Queue Pointers
029F  QUEBEG  EQU   $      ; where the queue ptrs live
029F  NULQUE  RMB   2      ; NULL queue
02A1  JIFQUE  RMB   2      ; JIFFY queue
02A3  TENQUE  RMB   2      ; TENTH of a second queue
02A5  SECQUE  RMB   2      ; SECOND queue
02A7  MINQUE  RMB   2      ; MINUTE queue
02A9  HOUQUE  RMB   2      ; HOUR queue
02AB  SCRQUE  RMB   2      ; scheduler queue
02AD  QUEEND  EQU   $      ; end of queue ptrs

; Heart Ram
02AD  HEARTF  RMB   1      ; heart beat (on/off)
02AE  HEARTC  RMB   1      ; visual heartbeat countdown counter
02AF  HEARTR   RMB   1      ; current heart rate
02B0  HEARTS   RMB   1      ; heart size flag
02B1  HBEATF   RMB   1      ; audio heartbeat flag

; Display Mode RAM
02B2  DSPMOD   RMB   2      ; display routine in use
02B4  UPDATE   RMB   1      ; "need to update" flag
02B5  NEWLUK   RMB   1      ; delayed update flag
02B6  NEWLIN   RMB   1      ; used by EXAMINE function
02B7  TXBFLG   RMB   1      ; standard/non-standard i/o flag

; LOAD/SAVE Cassette Function Flags
02B8  ZFLAG    RMB   1      ; load/save (-1/1) flag

; SCHEduler RAM
02B9  TCBPTR   RMB   2      ; next available TCB
02BB  RSTART   RMB   1      ; scheduler restart flag
02BC  KBDHDR   RMB   1      ; "put" pointer
02BD  KBDTAL   RMB   1      ; "get" pointer

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 23

```
; Line Buffer Routine RAM
02BE    BUFFLG RMB    1          ;line buffering flag
; COMSWI RAM
02BF    SWIJMP RMB    2          ;dispatch addr
; Temporary Storage
02C1    T0     RMB    1
02C2    T1     RMB    1
02C3    T2     RMB    1
02C4    T3     RMB    1
02C5    T4     RMB    1
02C6    T5     RMB    1
02C7    T6     RMB    1
02C8    T7     RMB    1
; Debugging Storage
02C9    D0     RMB    1
02CA    D1     RMB    1
02CB    D2     RMB    1
02CC    D3     RMB    1
02CD    D4     RMB    1
02CE    D5     RMB    1
02CF    D6     RMB    1
02D0    D7     RMB    1
; Insure Base Page RAM Size
02D1    DP-END EQU    $          ;last location of direct page
00D1    DP.LEN  EQU    DP-END-DP.BEG ;length of DP definitions
```

```
!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!!
! C O M M O N   R A M   L O C A T I O N S
!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!!
```

02D1 MM,BEG EQU \$ ;where memory begins

; ASCII Character Buffers

02D1	KBDBUF	RMB	32	Keyboard circular buffer
02F1	LINBUF	RMB	32	line input buffer
0311	LINEND	RMB	2	line buffer end
0313	TOKEN	RMB	32	token buffer
0333	TOKEND	RMB	1	token buffer end
0334	STRING	RMB	34	ASCII string buffer
0356	SWCHAR	RMB	10	S/W character expansion buffer
0360	OBJSTR	RMB	32	object string names

; Text Control Blocks (TCB)

0380	TXTEXA	RMB	TX.LEN	EXAMINE command text screen
0388	TXTSTS	RMB	TX.LEN	Status Line
0390	TXTPRI	RMB	TX.LEN	Primary Text Screen

; Creature Matrix/Control Block (CCB) Storage

0398	CMXLND	RMB	CTYPES*5	creature matrix
03D4	CMXEND	EQU	\$	,
03D4	CCBLND	RMB	CC.LEN*32	Land of the Creatures
05F4	CCBEND	EQU	\$	,

; Dungeon Maze RAM

05F4	MAZLND	RMB	32*32	The "Maze"
09F4	MAZEND	EQU	\$	,
09F4	NEIBOR	RMB	9	map of surrounding cells

; Task Control Block Storage

09FI	TCBLND	RMB	TC.LEN*38	Land of the TCBs
0B07	TCBEND	EQU	\$	,

; Object Control Block (OCB) Storage

OB07	EMPHND	RMB	OC.LEN	empty hand holder
OB15	OCBLND	RMB	OC.LEN*72	Land of the OCBs
OF05	OCBEND	EQU	\$	,
OF05	MM-END	EQU	\$	last location of memory

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CD - Common Definition file

PAGE 25

```
; Stack Area  
1000    PDL    EQU      $1000      ;where the push-down-list starts  
00FB    PD.LEN  EQU      $1000-MM.END ;length of push-down list  
        END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ONCE - Once only Initialization

PAGE 26

```
INCLUD B:ONCE.ASM
!!!!!! ONCE - ONLY INITIALIZATION !!!!!!
!!!!!! ONCE: Once-only Initialization !!!!!!

C000      ORG      $C000
C000      ONCE    EQU      $          ;in the beginning...
                                ; DEMO/GAME Start-Up
C000 CEC0B1  DEMO    LDU      #DEMO10   ;initialize RAM
C003 2003    BRA     COMINI
                                ;+
C005 CEC124  GAME    LDU      #GAME10   ;initialize RAM/fall into COMINI
```

; COMINI: Common Initialization Routine

Inputs:  
U - return addr

```
C008 10CE1000    COMINI EQU    $PDL      ;--- start of procedure COMINI
C008 10CE1000    LDS      #PDL      ;set stack pointer

; Program PIA$0
C00C BEFF00      LDX      #PIA$0      ;PIA$0 addr
C00F CC34FA      LDD      #$34FA     ;bit patterns
C012 A703        STA      P.PICRE,X   ;SEL2=0, Disable IRQ to CPU
C014 A701        STA      P.PICRA,X   ;SEL1=0, Disable IRQ to CPU

; Program PIA$1
C016 BEFF20      LDX      #PIA$1      ;PIA$1 addr
C019 A701        STA      P.PICRA,X   ;Cassette Motor OFF, Disable FIRQ to CPU
C01B 6F03        CLR      P.PICRE,X   ;program inputs/outputs
C01D E702        STB      P.PIIOB,X   ;Outputs: VDG(b7-b3), 1-Bit Sound(b1)
C01F 863C        LDA      #$3C       ;Six Bit Sounds ON, Disable FIRQ to CPU
C021 A703        STA      P.PICRB,X   ;.

; Set SAM and VDG modes
C023 CC2046      LDD      #B0.SAM    ;magic SAM bits!
C026 BDC266      JSR      SAM       ;turn on graphics screen
C029 86F8        LDA      #Z11111000  ;insure VDG and SAM are in sync
C02B A702        STA      P.PIIOB,X   ;.

; Zero All Of RAM
C02D 8E0200      LDX      #$0200    ;clear all of our RAM
C030 6F80        CLR      ,X+       ;zero next location
C032 BC4000      CMPX    #$4000    ;done yet?
C035 25F9        BLO      CINI10    ;nope

; Initialize RAM Locations
C03D 108ED7E8    LDY      #RAMDAT  ;RAM initialization start addr
C041 A6A0        CINI20   LDA      Y+       ;load byte count
C043 2741        BEQ      CINI40    ;we are done
C045 AEA1        LDX      ,Y++     ;load addr
C047 8D02        BSR      COPY     ;invoke copy function
C049 20F6        BRA      CINI20    ;loop
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ONCE - Once only Initialization

PAGE 28

## COPY: Byte Copy Routine

## Inputs:

A - Byte Count  
 X - Destination Addr  
 Y - Source Addr

## Returns:

A - Zero  
 X - Destination Addr + Byte Count  
 Y - Source Addr + Byte Count

C04B E6A0	COPY	LDB	,Y+	;load source byte
C04D E780		STB	,X+	;store into destination
C04F 4A		DECA		;decrement byte count
C050 26F9		BNE	COPY	;round and round
C052 39		RTS		;bye

## SYSTCB: Create System TCBs

## Returns:

Y - ptr to RAM initialization data

C053 3477	SYSTCB	PSHS	CC,A,B,X,Y,U	;save regs
C055 1A10		ORCC	#CC,I	;disable interrupts
C057 BE029F		LDX	#QUEBEG	;reset all queue ptrs
C05A 6F80	STCB10	CLR	,X+	:
C05C BC02AD		CMPX	#QUEEND	:
C05F 25F9		BLO	,STCB10	:
C061 BE09FD	;	LDX	#TCBLND	;reinitialize TCB area
C064 9FB9		STX	TCBPTR	:
C066 6F80	STCB20	CLR	,X+	:
C068 BC0B07		CMPX	#TCREND	:
C06B 25F9		BLO	STCB20	:
C06D 108ED7DC	;	LDY	#TCBDAT	;where TCB data resides
C071 0ABB		DEC	RSTART	;flag SCHEdULER restart
C073 CC000C		LDD	#Q,SCD	;TCBs start in SCHEduler queue
C076 AEA1	STCB30	LDX	,Y++	;load routine name
C078 270A		BEQ	STCB99	;done
C07A BDC25C		JSR	GETTCB	;allocate a new TCB
C07D AF43		STX	P,TCRDN,U	;store in TCB
C07F BDC21D		JSR	QUEADD	;add TCB to the correct queue
C082 20F2		BRA	STCB30	;loop for next TCB
C084 35F7	STCB99	PULS	CC,A,B,X,Y,U,PC	;restore regs/exit

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ONCE - Once only Initialization

PAGE 29

; Create All Objects

C086 8DCB	CINI40	BSR	SYSTCB	;create system TCBs
C088 CEDA91		LDU	#OMXTAB	;distribution table base addr
C08B 4F		CLRA		;reset object type
C08C E6C4	CINI42	LDB	,U	;load entry
C08E C40F		ANDB	#\$0F	;separate object count
C090 D78C		STB	OBJCNT	;+
C092 E6C0	;	LDB	,U+	;re-load entry/advance ptr
C094 54		LSRB		;extract maximum object level
C095 54		LSRB		;
C096 54		LSRB		;
C097 54		LSRB		;
C098 D78D		STB	OBJLVL	;
C09A 3F	CINI44	SWI		;create the object
C09B 17		FCB	OBIRTH	;
C09C 6A05		DEC	F,DCOWN,X	mark as creature owned
C09E 5C		INCB		;go down a level
C09F C105		CMPB	#5	;at bottom?
COA1 2F02		BLE	CINI46	;not yet
COA3 D68B		LDB	OBJLVL	;at bottom - start again
COA5 0ABC	CINI46	DEC	OBJCNT	;one less object to create
COA7 26F1		BNE	CINI44	;
;				not done yet
COA9 4C	INCA			;advance to the next object
COAA 1183DAA3	CMPU		#OMXEND	;
COAE 25DC	BLD		CINI42	done yet?
				nope

## ; Display COPYRIGHT Notice

COR0 CE0388	LDU	#TXTSTS	; string i/o to status line
COR3 0AB7	DEC	TXBFLG	
COR5 3F	SWI		; clear the status line
COR6 0A	FCB	CLRSTS	,
;			
COR7 3F	SWI		; from on the high...
COR8 02	FCR	DUTSTI	
COR9 F8	FCR	Z11111000	;COPYRIGHT DYNA MICRO MCMLXXXII
CORA DF	FCR	Z11011111	;COPYRIGHT DYNA MICRO MCMLXXXII
CORB 0C	FCR	Z00001100	;COPYRIGHT DYNA MICRO MCMLXXXII
CORC C9	FCR	Z11001001	;COPYRIGHT DYNA MICRO MCMLXXXII
CORD 22	FCR	Z00100111	;COPYRIGHT DYNA MICRO MCMLXXXII
CORE 45	FCR	Z01000101	;COPYRIGHT DYNA MICRO MCMLXXXII
CORF 00	FCR	Z00000000	;COPYRIGHT DYNA MICRO MCMLXXXII
COC0 02	FCR	Z00000010	;COPYRIGHT DYNA MICRO MCMLXXXII
COC1 65	FCR	Z01100101	;COPYRIGHT DYNA MICRO MCMLXXXII
COC2 C1	FCR	Z11000001	;COPYRIGHT DYNA MICRO MCMLXXXII
COC3 03	FCR	Z00000011	;COPYRIGHT DYNA MICRO MCMLXXXII
COC4 52	FCR	Z01010010	;COPYRIGHT DYNA MICRO MCMLXXXII
COC5 39	FCR	Z00111001	;COPYRIGHT DYNA MICRO MCMLXXXII
COC6 3C	FCR	Z00111100	;COPYRIGHT DYNA MICRO MCMLXXXII
COC7 00	FCR	Z00000000	;COPYRIGHT DYNA MICRO MCMLXXXII
COC8 68	FCR	Z01101000	;COPYRIGHT DYNA MICRO MCMLXXXII
COC9 DA	FCR	Z110101010	;COPYRIGHT DYNA MICRO MCMLXXXII
COCA CC	FCR	Z11001100	;COPYRIGHT DYNA MICRO MCMLXXXII
COCB 63	FCR	Z01100011	;COPYRIGHT DYNA MICRO MCMLXXXII
COCC 09	FCR	Z00001001	;COPYRIGHT DYNA MICRO MCMLXXXII
COCB 48	FCB	Z01001000	;COPYRIGHT DYNA MICRO MCMLXXXII
;			
COCE 0FB7	PSTS90	CLR	; restore standard i/o
CODO 39		RTS	; go home

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ONCE - Once only Initialization

PAGE 31

Arm the ABORT function

COD1 0A77 DEMO10 DEC AUTFLG indicate AUTOPLAY mode  
COD3 8D3F BSR IRQSYN enable/sync on IRQ

### Coming Attractions . . .

COD5 8E1F10 LDX #WIZ1 ;use the crescent wizzard  
COD8 0A9E DEC FADFLG ;fade  
CODA 3F SWI ;(don't erase status line!)  
CODB 14 FCB WIZTNO

CODEC	3F	SWI	OUTSTI	welcome message (part I)
CODB	03	FCB	Z100111111	^I DARE YE ENTER...^
CODEF	9FF	FCB	Z110100010	^I DARE YE ENTER...^
CODF	B2	FCB	Z000000010	^I DARE YE ENTER...^
COEO	03	FCB	Z000000110	^I DARE YE ENTER...^
COE1	06	FCB	Z010000101	^I DARE YE ENTER...^
COE2	45	FCB	Z000000110	^I DARE YE ENTER...^
COE3	06	FCB	Z010000101	^I DARE YE ENTER...^
COE4	4A	FCB	Z000000110	^I DARE YE ENTER...^
COE5	02	FCB	Z010010101	^I DARE YE ENTER...^
COE6	EDN	FCB	Z000000010	^I DARE YE ENTER...^
COE7	65	FCB	Z101110101	^I DARE YE ENTER...^
COE8	97	FCB	Z100000101	^I DARE YE ENTER...^
COE9	BB	FCB	Z100101111	^I DARE YE ENTER...^
COEA	FF	FCB	Z101111101	^I DARE YE ENTER...^
COEB	80	FCB	Z111011111	^I DARE YE ENTER...^
		FCB	Z100000000	^I DARE YE ENTER...^

COEC	3F	SWI		welcome message (part III)
COED	02	FCB	OUTSTI	... .THE DUNGEONS OF DAGGORATH
COEE	F7	FCB	X11110111	... .THE DUNGEONS OF DAGGORATH
COEF	BD	FCB	X10111101	... .THE DUNGEONS OF DAGGORATH
COFO	FA	FCB	X11101010	... .THE DUNGEONS OF DAGGORATH
COF1	20	FCB	X00100000	... .THE DUNGEONS OF DAGGORATH
COF2	A0	FCB	X10100000	... .THE DUNGEONS OF DAGGORATH
COF3	25	FCB	X00100101	... .THE DUNGEONS OF DAGGORATH
COF4	5C	FCB	X01011100	... .THE DUNGEONS OF DAGGORATH
COF5	72	FCB	X01110010	... .THE DUNGEONS OF DAGGORATH
COF6	BD	FCB	X10111101	... .THE DUNGEONS OF DAGGORATH
COF7	D3	FCB	X11010011	... .THE DUNGEONS OF DAGGORATH
COF8	03	FCB	X00000011	... .THE DUNGEONS OF DAGGORATH
COF9	CC	FCB	X11001100	... .THE DUNGEONS OF DAGGORATH
COFA	02	FCB	X00000010	... .THE DUNGEONS OF DAGGORATH
COFB	04	FCB	X00000100	... .THE DUNGEONS OF DAGGORATH
COFD	E7	FCB	X11100111	... .THE DUNGEONS OF DAGGORATH
COFE	7C	FCB	X01111100	... .THE DUNGEONS OF DAGGORATH
COFF	83	FCB	X10000011	... .THE DUNGEONS OF DAGGORATH
C100	44	FCB	X01000100	... .THE DUNGEONS OF DAGGORATH
C101	6F	FCB	X01101111	... .THE DUNGEONS OF DAGGORATH
	7B	FCB	X01111011	... .THE DUNGEONS OF DAGGORATH

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ONCE - Once only Initialization

PAGE 32

```
C102 3F      SWI          ;display message for a while
C103 10      FCB  WAIT    ;      :
C104 3F      SWI          ;      :
C105 10      FCB  WAIT    ;      :
;                                ;
C106 3F      SWI          ;now fade the wizard out
C107 15      FCB  WIZOUT  ;      :
C108 3F      SWI          ;blank the screen
C109 09      FCB  ZFLOP   ;      :
C10A 0AB4     DEC  UPDATE  ;      :
C10C 13      SYNC         ;      :
;                                ;
; Create Autoplay Dungeon/Drop Player into it
;                                ;
C10D 8602     LDA  #2      ;level three
C10F CED7D5   LDU  #DEMOPAT ;use demo initialization data
C112 201B     BRA  GAME20  ;      :
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ONCE - Once only Initialization

PAGE 3

```

; IRQSYN: Enable and Synchronize on IRQ

C114 CC343C    IRQSYN   LDD      #$343C      ;bit patterns
C117 B7FF21     STA       P.PICRA+PIA$1  ;turn off cassette motor
C11A F7FF23     STB       P.PICRB+PIA$1  ;enable Six BIT Sounds
C11D 4C          INCA
C11E B7FF03     STA       P.PICRB+PIA$0  ;ACCA <- $35
C121 3CEF        CWAI      #(CC,I XOR $FF) ;SEL2=0, IRQ on falling edge FS
C123 39          RTS

```

Game Initialization				
C124 8DEE	GAME10	BSR	IRQSYN	;enable/sync on IRQ
C126 CC100B		LDD	#\$100B	;set initial position
C129 DB13		STD	PROW	,
C12B OF17		CLR	PPOW	;correct power settings
C12D 4F		CLRA		;start on level 0
C12E CED7D9		LDU	#GAMDAT	;game initialization data

```

; Common DEMO/GAME Code

C131 3F      GAME20 SWI      ;display filler message
C132 16      FCB      PREFAR
C133 3F      SWI      ;create the new level
C134 1A      FCB      NEWLVL

; Create and Stow Objects into Bag

C135 108E0229  LDY      #BAGPTR    ;load ptr to bag
C139 A6C0      GAME30 LDA      ,U+        ;load next object type
C13B 2B12      BMI      GAME40     ;done
C13D 3F      SWI      ;create the object
C13E 12      FCB      ORBIRTH    ;mark as player owned
C13F 6C05      INC      P,OCOWN,X
C141 1E13      EXG      X,U
C143 3F      SWI      ;
C144 18      FCB      OCBFIL    ;
C145 1E13      EXG      X,U
C147 6F08      CLR      P,OCREV,X    ;reveal the object
C149 AFA4      STX      P,OCPTR,Y    ;append to bag list
C14B 1F12      TFR      X,Y        ;advance to next object
C14D 20EA      BRA      GAME30     ;loop until done

C14F 0D77      GAME40 TST      AUTFLG    ;autoplay?
C151 2713      BEQ      GAME50     ;nope - normal startup

C153 0A9B      DEC      SLEEP     ;turn off scheduler
C155 8ECB82    LDX      #MAPPER   ;throw up map display
C158 9FB2      STX      DSPMOD    ;show everything
C15A 0A94      DEC      MAPFLG
C15C 3F      SWI      ;
C15D 0E      FCB      PUPDAT   ;
C15E 3F      SWI      ;
C15F 10      FCB      WAIT     ;
C160 3F      SWI      ;
C161 10      FCB      WAIT     ;
C162 0F9B      CLR      SLEEP    ;wake up

C164 13      SYNC    ;delay for AUTOPLAY synchronization
C165 13      SYNC    ;

C166 3F      GAME50 SWI      ;initial view
C167 19      FCB      INIVU    ;fake the prompt
C168 3F      SWI      ;
C169 0F      FCB      PROMPT  ;fall into the scheduler...
C16A 7EC1F5    JMP      SCHED

; END

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 35

INCLUD B:COMMON.ASM

!!!!!!  
THE SCHEDULER  
!!!!!!

The scheduler utilizes the "Round-Robin" algorithm.  
Tasks are executed one-at-a-time, with each running  
to completion. Tasks should be as short as possible  
to insure all tasks run within their time windows.

SCDQUE is a 16-bit address pointing at a linked  
list of TCB's which make up the scheduling queue.

Upon task completion, the registers should have the  
following values:

A - New Countdown Time (Optional)

B - Queue Pointer or Zero (destroy the TCB)

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGERATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 36

; GETBUF: Get the next buffer from the cassette

Inputs:

X - buffer addr

Returns:

A - zero if no errors

B - block type

X - buffer addr updated (if no errors)

Z - condition code set if no errors

C16B BF007E	GETBUF	STX >\$007E	;set the buffer addr
C170 103F		SWI2	;read the next block
C172 06		FCB	
C173 40		TSTA	;set condition codes
C174 1026DEAF		LBNE	\$A027 ;restart the world if error
C178 F6007C		LDB	>\$007C ;load the block type
C17B 39		RTS	;bye

; PIATAP: Initialize PIA's for Tape Operations

C17C CEFF00	PIATAP	LDU #PIA\$0	;addr of PIA #0 (\$FF00)
C17F CC343C		LDD #\$343C	;magic PIA bit patterns
C182 A743		STA P.PICRB,U	;SEL2=0, Disable IRQ to CPU
C184 B7FF23		STA P.PICRB+PIA\$1	;Six Bit Sounds OFF, Disable FIRQ to CPU
C187 F7FF21		STB P.PICRA+PIA\$1	;Cassette Motor ON, Disable FIRQ to CPU
C18A 39		RTS	

; TAPDLY: Delay to insure motor has started

C18B 9E00	TAPDLY	LDX BIGZER	;insure motor is running
C18D 301F	TAPDLY10	LEAX -1,X	;
C18F 26FC		BNE TAPDLY10	;
C191 39		RTS	;bye

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 37

**SAVE:** Process the Game Save Function

C192 8DE8 SAVE BSR PIATAP ;init PIA's for tape operations  
C194 8DF5 BSR TAPDLY ;delay for a while  
C196 8DF3 BSR TAPDLY ;  
;

C198 103F	SWI2	WRTLDR-\$A000	;write the leader
C19A 0C	FCB		
C19B 103F	SWI2	BLKOUT-\$A000	;write the name file block
C19D 08	FCB		

```
        Write DATA Blocks  
C19E BDEB      BSR      TAPDLY      ;delay for a while  
C1A0 103F      SWI2      UNTILH #4000  ;write second leader  
C1A2 0C      ECE
```

```

C1A3 BE0200      ;SAVE10    LDIX    #DP,BEG      ;start of save area
C1A6 CD0180      ;SAVE10    LDD     #$0180      128-byte DATA blocks
C1A9 FD007C      ;SAVE10    STD     >$007C      ;buffer addr
C1AC BF007E      ;SAVE10    STX     >$007E      ;write the next block
C1AF 103F        SWI2
C1B1 08          FCB     BLKOUT-$A000
C1B2 8C0F05      CMPX    #MM,END      done yet?
C1B5 25EF        BLO     SAVE10      nope

```

```
        Write EOF Block  
C1B7 FF007C      STU      >$007C      end of file block ($FF00)  
C1BA 103F      SWI2      ;  
C1BC 08      FCB      BLKOUT-$A000      ;  
C1BD 80CC      BSR      TAPDLY      some blank tape  
C1BF 202B      BRA      LOAD90      drop into common code
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 38

```

; LOAD: Process Game Restore Function

C1C1 8DB9    LOAD   BSR    PIATAP      ; init PIA's for tape operations
C1C3 103F    SWI2
C1C5 04      FCB    CSRDON-$A000  ; get into bit sync
; ,.

; Search For Correct File

C1C6 DE0B    LOAD10 LDU    FLOP       ; use alternate screen as buffer
C1C8 AEC4    LDX    P.VDBAS,U
C1CA 8DA1    BSR    GETBUF     ; get the next bufferful
C1CC 26F8    BNE    LOAD10     ; not a file header
; ,.

; Compare filenames

C1CE AEC4    LDX    P.VDBAS,U  ; cassette buffer addr
C1D0 CE0313    LDU    #TOKEN    ; TOKEN buffer addr
C1D3 C608    LDB    #8         ; eight characters in filename
C1D5 A680    LDA    ,X+        ; load next char
C1D7 A1C0    CMPA   ,U+        ; did we match
C1D9 26E6    BNE    LOAD       ; nope - get next cassette buffer
C1DB 5A      DECB
C1DC 26F7    BNE    LOAD12    ; one less to match
; ,. until we have a complete match

; We found our file...

C1DE 103F    SWI2
C1E0 04      FCB    CSRDON-$A000  ; re-synchronize
C1E1 BE0200    LOX    #$0200    ; start of restore area
C1E4 8D87    BSR    GETBUF     ; get the next buffer
C1E6 2AF0    BPL    LOAD12    ; not done yet
; ,.

; Restart the Game

C1E8 10CE1000 LDS    #PDL      ; reset the stack ptr
C1EC BDC114    LOAD90 JSR    IRQSYN   ; enable/sync on IRQ
C1EF 0FB8    CLR    ZFLAG     ; indicate tape operation complete
C1F1 3F      SWI
C1F2 19      FCB    INIVU     ; show what is going on
C1F3 3F      SWI
C1F4 0F      FCB    PROMPT   ; prompt the human
; ,.

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 39

```

; SCED1: The "Merry-Go-Round"

C1F5 CE02AB    SCED   LDU    #SCDQUE      ;start at queue head
C1F8 0FBB      CLR    RSTART      ;reset "restart" flag
C1FA 1F32      SCEDO  TFR    U,Y      ;curr TCB becomes prev TCB

; Check for TAPE functions

C1FC 0DB8      SCED1  TST    ZFLAG      ;check for load/save (-1/1)
C1FF 2E92      BGT    SAVE       ;
C200 2BBF      BMI    LOAD       ;

; Process next entry in SCDQUE

C202 EECA      LDU    P,TCPTB,U    ;advance curr TCB ptr
C204 27EF      BEQ    SCED      ; start again if tail reached

; Execute the selected task

C206 3460      PSHS   Y,U      ;save TCB pointers
C208 ADD803    JSR    [P,TCRTN,U]  ;dispatch to routine
C20B 3560      PULS   Y,U      ;restore TCB pointers

; Do we need to restart?

C20D 0DBB      TST    RSTART      ;check restart flag?
C20F 26E4      BNE    SCED      ; we need to start again

; Check disposition upon execution completion

C211 C10C      CMPB   #Q,SCD      ;leave in the SCED queue?
C213 27E5      BEQ    SCEDO      ; yes - process next TCB

; Remove TCB from SCED and add to appropriate queue

C215 BD21      BSR    QUERMV     ;de-link current TCB from chain
C217 BD04      BSR    QUEADD     ;reschedule TCB to appropriate queue
C219 1F23      TFR    Y,U      ;fake previous TCB run
C21B 20DF      BRA    SCED1      ;process next TCB

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines.

PAGE 40

```
!!!!!!QUEUE MANAGER!!!!!!
!!!!!!QUEUE MANAGER!!!!!!
```

All queues are composed of forward linked lists of TCB's.  
 Associated with each queue is an initial queue pointer.  
 The linked lists are terminated with a null.

\*\*\*\*\*

QUEADD: Add a TCB to the end of a queue (linked list)

Inputs:

- A - Countdown Time (Optional)
- B - Queue Code
- U - pointer to TCB

C21D 3417	QUEADD	PSHS CC,A,B,X	;save some regs
C21F 1A10		ORCC #CC,I	;non-interruptible code
C221 A742		STA P.TCTIM,U	;store countdown time in TCB
C223 BE029F		LDX #QUEBEG	;where the queue ptrs reside
C226 3A		ABX	;determine addr of queue ptr
C227 4F		CLRA	;zap link ptr
C228 5F		CLRB	
C229 EDC4		STD P.TCPTR,U	; ;
C22B 10A384	QUEAD1	CMPD X	;search for end of linked list
C22E 2704		BEQ QUEAD2	;found it!
C230 AE84		LDX P.TCPTR,X	;advance to next TCB
C232 20F7		BRA QUEAD1	;loop until we find it
C234 EF84	QUEAD2	STU X	;link last TCB to new TCB
C236 3597		PULS CC,A,B,X,PC	;restore regs/go home

QUERMV: Remove a TCB from a Linked List

Inputs:

- U - Ptr to TCB to be removed
- Y - Ptr to previous TCB

C238 3411	QUERMV	PSHS CC,X	;save condition codes
C23A 1A10		ORCC #CC,I	;non-interruptible code
C23C AEC4		LDX P.TCPTR,U	;get curr TCB link
C23E AFA4		STX P.TCPTR,Y	;leapfrog prev TCB past this TCB
C240 3591		PULS CC,X,PC	;restore condition codes/go home

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 41

|| QUESCN: Process TCB's in the CLOCK queues

|| Inputs:  
|| U - Queue Pointer

C242 3474  
C244 0D9B  
C246 2612  
|| QUESCN PSHS R,X,Y,U ;save some regs  
|| TST SLEEP ;is the system SLEEPing?  
|| BNE QUESC9 ; yes - don't process queues

|| Traverse the Linked List

C248 1F32  
C24A EEC4  
C24C 270C  
C24E 6A42  
C250 26F6  
|| QUESCO TFR U,Y ;curr TCB becomes prev TCB  
|| LDU P,TCPTR,U ;advance curr TCB ptr  
|| BEQ QUESC9 ;null ptr means tail reached  
|| DEC P,TCTIM,U ;decrement countdown timer  
|| BNE QUESCO ; time left - process next TCB

|| Reschedule TCB into SCDQUE

C252 8DE4  
C254 C60C  
C256 8DC5  
C258 20EE  
|| BSR QUERMV ;delink curr TCB  
|| LDB #Q,SCD ;load queue code  
|| BSR QUEADD ;add TCB to the scheduler queue  
|| BRA QUESC9 ;process next TCB  
|| QUESC9 PULS R,X,Y,U,PC ;restore regs/go home

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

BAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 42

|||||||||  
TCB MANAGER  
|||||||||

GETTCB: Allocates a TCB

Returns:  
U - address of TCB

C25C 3410	GETTCB	PSHS X	; save regs
C25E DEB9		LDU TCBPTR	; ptr to next available TCB
C260 3047		LEAX TC,LEN,U	; advance ptr
C262 9FB9		STX TCBPTR	; update RAM
C264 3590		PULS X,PC	; restore regs/exit

!!!!!!  
SAM PROGRAMMING ROUTINES  
!!!!!!

The Motorola MC6883 Synchronous Address Multiplexer (SAM) registers are mapped from \$FFC0 to \$FFDF. Each register consists of two consecutive bytes. Writing (any value) to the even byte will clear the register. Writing to the odd byte will set the register.

The SAM bits are defined as:

TY	Map Type (must be zero)
M1,M0	Map Size: 00 - 4K Dynamic 01 - 16K Dynamic 10 - 64K Dynamic 11 - 64K Static
R1,R0	MPU Rate: 00 - 0.9MHz Only (Transparent Refresh) 01 - 0.9/1.8MHz (Transparent Refresh) 10 - 1.8MHz Only 11 - 1.8MHz Only
P1	Paging (must be zero)
F6-F0	Display Offset Page Number (512 bytes/page)
V2-V0	VDG Mode: 000 - Alpha Internal/External, Semi-Graphics 4/6 001 - Color/Resolution Graphics-1 (G2C/G2R) 010 - Color Graphics-2 (G2C) 011 - Resolution Graphics-2 (G2R) 100 - Color Graphics-3 (G3C) 101 - Resolution Graphics-3 (G3R) 110 - Color/Resolution Graphics-6 (G6C/G6R) 111 - DMA

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 44

SAM: Reprogram the SAM

Inputs:

A - Bit:	7 6 5 4 3 2 1 0
SAM:	TY M1 M0 R1 R0 F1 F6 F5
B - Bit:	7 6 5 4 3 2 1 0
SAM:	F4 F3 F2 F1 F0 V2 V1 V0

C266 3416	SAM	PSHS	A,B,X	; save regs
C268 BEFFC0		LDX	#\$FFC0	; SAM base addr
C26B 44	SAM10	LSRA		; 16-bit logical shift right
C26C 56		RORB		
C26D 2503		BCS	SAM12	; former LSBit was set
C26F A784		STA	0,X	; clear the SAM reg
C271 8C		FCB	SKIP2	
C272 A701	SAM12	STA	1,X	; set the SAM reg
C274 3002		LEAX	2,X	; advance to next reg
C276 8CFFD4		CMPX	#\$FFD4	; done yet?
C279 25F0		BLD	SAM10	; nope
C27B 3596		PULS	A,B,X,PC	; restore regs/exit

!!!!!!  
C L O C K   R O U T I N E S  
!!!!!!

CLOCK is the IRQ interrupt service routine. This routine provides a number of functions:

- \* Swaps Graphics-6 Resolution (G6R) Screens
- \* Produces 30Hz Buzzing Noise for Opening
- \* Maintains Heartbeat/flash
- \* Maintains the Time-Of-Day (TOD) Counters
- \* Maintains the CLOCK Queues
- \* Performs Keyboard Input

IRQ is a hardware generated interrupt which occurs at 1/60th of a second intervals, called a "JIFFY".

If location UPDATE is non-zero, CLOCK will swap the display screens (by reprogramming the SAM), and will reprogram the VDG to the correct color set. The SAM requires buffers to be on page (512-bytes) boundaries.

FLIP is a byte containing the SAM page number of the buffer currently being displayed. FLOP points at the buffer being updated.

The TOD timers will keep time for 256 days before rolling over.

The POLCAT routine is a part of the BASIC ROM routines. If a character is seen it will be added to the Keyboard buffer, KDBUF.

The CLOCK queues are linked lists of TCB's. Each TCB in a list has its "countdown" timer decremented. If a timer has expired, the TCB is transferred from the CLOCK queue to the SCHED queue for execution.

If location SLEEP is non-zero, CLOCK will not scan the queues. Instead it will decrement SLEEP once per second and will only process the queues when SLEEP becomes zero. In this way, the whole scheduling process can be suspended.

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 46

```

; CLOCK: IRQ Interrupt Service Routine
C27D 8EFF20    CLOCK   LDX    #PIA$1      ;load ptr to PIA #1
C280 A688E3    LDA    P,PICRB-$20,X  ;check for JIFFY interrupt
C283 102A0099  LBPL    CLK90      ; ignore all but jiffy interrupts

; Insure Correct DP
C287 8602      LDA    #$02      ;org the DP at $0200
C289 1F8B      TFR    A,DP      ;           .

; Swap the Display Screens
C28B 0DB4      TST    UPDATE     ;do we need to update the screen?
C28D 270E      BEQ    CLK20     ;nope
C28F DC09      LDD    FLIP      ;exchange display pointers
C291 DE0B      LDU    FLOP      ;
C293 DD0B      STD    FLOP      ;
C295 DF09      STU    FLIP      ;
C297 EC44      LDI    P,VIDSAM,U ;SAM/VID page number
C299 8DCB      BSR    SAM       ;change the SAM address
C29B OFB4      CLR    UPDATE     ;reset flag

; 30-Hz. Buzzing Sound (used in opening)
C29D 0D9C      CLK20   TST    NOISEF    ;buzz flag on?
C29F 2708      BEQ    CLK30     ;nope
C2A1 0390      COM    NOISEV    ;complement noise value
C2A3 969D      LDA    NOISEV    ;load noise value
C2A5 48        LSLA   NOISEV    ;adjust position
C2A6 48        LSLA   NOISEV    ;
C2A7 A784      STA    P,PIIODA,X ;output to DAC (bit7-bit2)

```

```

        ; Heartbeat Sound
C2A9 0DB1    CLK30   TST    HBEATF   ;should we consider making a sound?
C2AB 272F    BEQ     CLK40    ;nope
C2AD 0AAE    DEC     HEARTC   ;should we make a sound?
C2AF 262B    BNE     CLK40    ;nope
C2B1 96AF    LDA     HEARTR   ;reset heart beat counter
C2B3 97AE    STA     HEARTC   ;
;
C2B5 E602    LDB     P.PIIOB,X  ;flip single bit sound output bit
C2B7 C802    EDORB   #BIT1   ;
C2B9 E702    STB     P.PIIOB,X  ;
;
        ; Heartbeat Status Line Flash (can be disabled)
C2BB 0DAD    TST    HEARTF   ;process heart beat?
C2BD 271D    BEQ     CLK40    ;nope
;
C2BF CE0388    LDU    #TXTSTS  ;status TXB
C2C2 AE44    LDX     P.TXCUR,U ;save current cursor
C2C4 CC000F    LDD     #15     ;position for HEART
C2C7 ED44    STD     P.TXCUR,U ;
;
C2C9 8620    LDA     #I.SHL   ;small heart left
C2CB 03B0    COM     HEARTS   ;small or large heart?
C2CD 2702    BEQ     CLK32    ;small
C2CF 8622    LDA     #I.LHL   ;large heart left
;
C2D1 BDCA17    CLK32   JSR     TXTDPB   ;deposit char
C2D4 6C45    INC     F.TXCUR+1,U ;advance cursor position
C2D6 4C      INCA    ;now process right side
C2D7 BDCA17    JSR     TXTDPB   ;deposit char
C2D9 AF44    STX     F.TXCUR,U ;restore original cursor
;
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 48

```

; Update Time-Of-Day (TOD) Counters
C2IC CEO2A1 CLK40 LDU #JIFQUE      ;always process the jiffy queue
C2DF BDC242    JSR QUESCN      ;
;                                ;
C2E2 BE0295    LDX #JIFFY       ;beginning of timers
C2E5 108EC324  LDY #ROLTAB     ;table of rollover values
C2E9 6C84      CLK42 INC X      ;bump the timer
C2EB 8C029A    CMPX #DAY       ;are we done?
C2EE 270F      BEQ CLK50      ;yes
C2FO A684      LDA ,X         ;reach rollover value?
C2F2 A1A0      CMPA ,Y+       ;
C2F4 2D09      BLT CLK50      ;nope - done with timers
;                                ;
C2F6 6FB0      CLR ,X+       ;clear timer/advance to next timer
C2F8 3342      LEAU ,2,U      ;advance to next queue
C2FA BDC242    JSR QUESCN     ;process the queue
C2FD 20EA      BRA CLK42      ;loop

; Perform Keyboard Input
C2FF 0D28      CLK50 TST FAINT   ;have we fainted?
C301 261D      BNE CLK90      ;yes - don't scan keyboard

; Determine Scanning Method
C303 0D77      TST AUTFLG     ;are we in autoplay?
C305 2711      BEQ CLK60      ;nope - use regular i/o rtn

; AUTOPLAY - Abort upon Key depression
C307 7FFF02    CLR P.PIIOB+PIAS0  ;strobe the Keyboard
C30A B6FF00    LDA P.PIIOA+PIAS0  ;collect PIA info
C30D 847F      ANDA #$7F       ;mask out MSBit
C30F 817F      CMPA #$7F       ;any keys depressed?
C311 270D      BEQ CLK90      ;nope
;                                ;
C313 8EC005    LDX #GAME      ;force transfer upon keystroke
C316 AF6A      STX 10,S       ;
;                                ;
; GAME - Use POLCAT Routine/Buffer Chars
C318 103F      CLK60 SWI2      ;scan the keyboard
C31A 00        FCB POLCAT-$A000  ;check for nulls
C31B 4D        TSTA             ;ignore nulls
C31C 2702      BEQ CLK90      ;deposit char in buffer
C31E 8D20      BSR KBDPUT     ;
;                                ;
C320 B6FF02    CLK90 LDA P.PIIOB+PIAS0  ;reset PIA for next interrupt
C323 38        RTI             ;dismiss the interrupt
;                                ;
; ROLTAB: Rollover Values for Timers
C324 06        ROLTAB FCB 6      ;jiffy
C325 CA        FCB 10         ;tenth

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 49

C326	3C	FCB	60	;second
C327	3C	FCB	60	;minute
C328	18	FCB	24	;hour

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMMON - Common Operating System Routines

PAGE 50

!!!!!!  
KEYBOARD SERVICE ROUTINES  
!!!!!!

The Keyboard service routines use a 32-byte circular buffer. There is no check for buffer overflow/overrun.

\*\*\*\*\*

KBDGET: Get a character from the buffer

Returns:

A - char

C329 3415	KBDGET	PSHS	B,X,CC	; save regs/condition codes
C32B 1A10		ORCC	#CC,I	; non-interruptible code
C32D 4F		CLRA		; default char if none found
C32E 8E02D1		LDX	#KBDBUF	; load buffer base addr
C331 D6BC		LDB	KBDHDR	; load GET offset
C333 D1BD		CMFB	KBDTAL	; empty buffer?
C335 2707		BEQ	KBDGT9	; yes
C337 A685		LDA	B,X	; get the next char
C339 5C		INCB		; bump offset
C33A C41F		ANDB	#Z00011111	; wrap at 32 bytes
C33C D7BC		STR	KBDHDR	; update offset
C33E 3595		PULS	B,X,CC,PC	; restore regs/condition codes/go home

KBDPUT: Add a char to the buffer

Inputs:

A - char

C340 3415	KBDPUT	PSHS	B,X,CC	; save regs/condition codes
C342 1A10		ORCC	#CC,I	; non-interruptible code
C344 8E02D1		LDX	#KBDBUF	; load buffer base address
C347 D6BD		LDB	KBDTAL	; load PUT offset
C349 A785		STA	B,X	; deposit byte in buffer
C34B 5C		INCB		; bump offset
C34C C41F		ANDB	#Z00011111	; wrap at 32 chars
C34E D7BD		STB	KBDTAL	; update offset
C350 3595		PULS	B,X,CC,PC	; restore regs/condition codes/go home

;

END

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMSWI - Common SWI Dispatch

PAGE 51

INCLUD B:COMSWI.ASM

!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!!  
C O M M O N S W I D I S P A T C H  
!!!!!! !!!!!!! !!!!!!! !!!!!!! !!!!!!!

```

NAM    COMSWI
LIBRY  CD.TXT
XDEF   SWISER,SW2SER,
XDEF   PUPSUB,SNOBJ,OBJNAM,PLOOK

```

```

XREF  ASRD4,ASRD5,ASRD6,RITMSK
XREF  STSVDB,PRIVDB,VFTTAB,EXAMIO,FNOBJ
XREF  SYSTCB,DGNGEN,CBIRTH,VIEWER,TXTXXX,TXTSCR
XREF  ADJTAB,GENTAB,FNOCEL,ODBTAB,COPY,XXXTAB
XREF  WIZ1,T,SHI4,T,SWO3,T,TOR4
PSCT

```

SWISER: SWI Interrupt Handler

C352 1CEF	SWISER	ANIICC	<code> #(CC.I XOR \$FF)</code> ;restore IRQ interrupt	
C354 AE6A		LDX	10,S	;load return addr
C356 A680		LDA	1+,S	;load parameter byte
C358 AF6A		STX	10,S	;update return addr
C35A BEC384		LDX	#SWIBEG	first entry in dispatch table
C35D CEC995		LDU	#SWITAB	offset table addr
C360 E6C0	SSER10	LDB	,U+	load next offset
C362 3A		ABX		add to base addr
C363 4A		DECA		decrement counter
C364 2AFA		BPL	SSER10	not done yet
C366 AFE3		STX	,--S	stack dispatch addr
C368 EC63		LDD	3,S	restore regs before dispatching
C36A AE66		LDX	6,S	
C36C EE6A		LDU	10,S	
C36E ADF1		JSR	[,S+1]	dispatch to routine/exit
C370 3B		RTI		dismiss the interrupt

SW2SER: SWI2 Interrupt Handler

C371 5F	SW2SER	CLRB		;switch to BASIC dp
C372 1F9B		TFR	B,DP	,
C374 EE6A		LDU	10,S	;load return addr
C376 E6C0		LDB	,U+	;load BASIC parameter byte
C378 EF6A		STU	10,S	;update return addr
C37A CEA000		LDU	##A000	;table base addr
C37B ADD5		JSR	[B,U]	dispatch to routine
C37F A761		STA	1,S	;return parameters
C381 AF64		STX	4,S	
C383 3B		RTI		;bye

## NUM: Define SWI Codes

0000	FOO	SET	0	
0000	SETFAD	EQU	F00	;set amount of fade based on RANGE
0001	VCTLST	EQU	F00+1	;vector list processor
0002	OUTSTI	EQU	F00+2	;string output (string follows)
0003	OUTSTR	EQU	F00+3	;string output
0004	OUTCHR	EQU	F00+4	;character output
0005	EXPAND	EQU	F00+5	;string decompression routine
0006	EXPANO	EQU	F00+6	; secondary entry point
0007	RANDOM	EQU	F00+7	;random number generator
0008	ZFLIP	EQU	F00+8	;current screen clear
0009	ZFLOP	EQU	F00+9	;alternate screen clear
000A	CLRSTS	EQU	F00+10	;status line clear
000B	CLRPRI	EQU	F00+11	;primary text area clear
000C	HUPDAT	EQU	F00+12	;update the heartrate
000D	STATUS	EQU	F00+13	;update the status line
000E	PUPDAT	EQU	F00+14	;update the main screen
000F	PROMPT	EQU	F00+15	;prompt the human for input
0010	WAIT	EQU	F00+16	;wait for a number of jiffies
0011	ZERO	EQU	F00+17	;clear RAM
0012	NEGRAM	EQU	F00+18	;set RAM to negative one
0013	WIZIN	EQU	F00+19	;fade-in
0014	WIZINO	EQU	F00+20	; secondary entry point
0015	WIZOUT	EQU	F00+21	;fade-out
0016	PREPAR	EQU	F00+22	;"PREPARE!" msg on EXAMINE screen
0017	OBIRTH	EQU	F00+23	;object creation routine
0018	OCBFIL	EQU	F00+24	;OCB filling routine

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMSWI - Common SWI Dispatch

PAGE 53

```
0019      ;INIVU EQU    F00+25      ;create initial view
001A      ;NEWLVL EQU    F00+26     ;create a new level
001B      ;ISOUND EQU    F00+27     ; secondary entry point
001C      ;SOUNDS EQU    F00+28     ;sound routines
C384      ;SWIBEG EQU    $          ;where the code starts
;          END
```

INCLUD B:VCTLST.ASM

;-----  
DISPLAY LIST PROCESSOR  
-----;

## Inputs:

X - vector list pointer  
 UXSCAL - radix-7 scaling constant for Y  
 VYSCAL - radix-7 scaling constant for Y

## Notes:

B B B B B B B B


 Radix-7 Binary Point

Coordinate list is stored in memory as: y,x y,x y,x.....

Since there are no y values greater than 191, values from 192 to 255 are available for use as flags and indicators.

## SETFAX: Set the Lighting Levels Based on RANGE

C384 966E	SETFAX	LDA	RLIGHT	;assume regular light
C386 0D75		TST	MAGFLG	;regular light
C388 2704		BEQ	SFAD10	;magic light
C38A 966F		LDA	MLIGHT	;reset flag
C38C 0F75		CLR	MAGFLG	
C38E 5F	SFAD10	CLRB		;assume maximum brightness
C38F 8007		SUBA	#7	;offset brightness level
C391 908B		SUBA	RANGE	;attenuate brightness with distance
C393 2C0A		BGE	SFAD30	;use maximum brightness
C395 5A		DEC B		;assume total darkness
C396 81F9		CMPA	#-7	
C398 2F05		BLE	SFAD30	;total darkness
C39A 8ECB96		LDX	#BITMSK+8	
C39D E686		LDB	A,X	;load lighting value
C39F D72D	SFAD30	STB	VCTFAD	;set fade value
C3A1 39	COMRTS	RTS		;bye

```

        ; VCTLSX: Vector List Processing Routine
C3A2    VCTLSX EQU      $          ; where it all begins
C3A2    0F51     CLR      DRWFLG   ; flag - no starting (Y,X)

        ; Test if VCTFAD says to block out this vector list
C3A4    962D     LDA      VCTFAD   ; load fade-out value
C3A6    4C        INCA     VCTAB   ; is it $FF?
C3A7    274B     BEQ      VCTEND  ; draw nothing/exit

        ; Special Vector List Control Code Processing
C3A9    E684     LDR      X          ; load y-coordinate
C3AB    COFA     SUBB    #V$RTS   ; special control code?
C3AD    2520     BCS     VCTABS  ; no - absolute mode processing
C3AF    3001     LEAX    1,X      ; advance byte ptr
C3B1    108EC3B9  LDY     #VCTDIS ; dispatch table addr
C3B5    E6A5     LDB     B,Y      ; load offset value
C3B7    6EAS     JMP     B,Y      ; , ,
C3B9    10        VCTDIS EQU      $          ; dispatch table
C3BA    06        FCB     VCTRTS-VCTDIS ; ($FA) - V$RTS
C3BB    5E        FCB     VCTJSR-VCTDIS ; ($FB) - V$JSR
C3BC    0H        FCB     VCTREL-VCTDIS ; ($FC) - V$REL
C3BD    3D        FCB     VCTJMP-VCTDIS ; ($FD) - V$JMP
C3BE    12        FCB     VCTEND-VCTDIS ; ($FE) - V$END
C3BF    00        FCB     VCTNEW-VCTDIS ; ($FF) - V$NEW

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VCTLST - Display List Processor

PAGE 56

; VCTJSR: Jump to Vector List Subroutine  
C3BF EC81 VCTJSR LDD ,X++ ;load subroutine addr  
C3C1 AFE3 STX ,--S ;store return addr  
C3C3 1F01 TFR D,X ;process subroutine  
C3C5 8C FCB SKIP2 ;drop into V\$NEW...  
  
; VCTJMP: Jump to a New Vector List  
C3C6 AE84 VCTJMP LDX ,X ;load new vector list addr  
C3C8 8C FCB SKIP2 ;drop into V\$NEW...  
  
; VCTRSTS: Return from Vector List Subroutine  
C3C9 AEE1 VCTRSTS LDX ,S++ ;load return addr/drop into V\$NEW...  
  
; VCTNEW: Start new vector list  
C3CB OF51 VCTNEW CLR DRWFLG ;indicate we should get new start  
C3CD 20DA BRA VCTL10 ;loop for next coordinates

## ; VCTABS: Absolute Mode Processing

C3CF 0D51	VCTABS	TST	DRWFLG	; have we set starting coordinates?
C3D1 2606		BNE	VABSOO	; yes
C3D3 8D0D	;	BSR	VABSO20	; set starting coordinates
C3D5 0A51		DEC	DRWFLG	; say we have started
C3D7 20D0		BRA	VCTL10	; loop for next coordinates
C3D9 8D05	VABSOO	BSR	VABSO10	; set end point
C3DB FDCAB7		JSR	VECTOR	; draw the vector/exit
C3DE 20C9		BRA	VCTL10	; loop for next coordinates
C3E0 8D15	VABSO10	BSR	NEWOLD	; end point becomes beginning
C3E2 E680	VABSO20	LDB	,X+	; load Y-coordinate
C3E4 D754		STB	TY	; store absolute value
C3E6 8D18		BSR	ASCALY	; scale Y-coordinate
C3E8 D307		ADDD	VCNTRY	; add Y-centroid
C3EA DD33		STD	Y1	; store new end point
C3EC E680	;	LDB	,X+	; load X-coordinate
C3EE D752		STB	TX	; store absolute value
C3F0 8D14		BSR	ASCALX	; scale X-coordinate
C3F2 D305		ADDD	VCNTRX	; add X-centroid
C3F4 DD35		STD	X1	; store new end point
C3F6 39	VCTEND	RTS		; done

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VCTLST - Display list Processor

PAGE 58

NEWOLD: Copy Ending (Y,X) to Starting (Y,X)

C3F7 DC33	NEWOLD	LDD	Y1	;copy Y-end points
C3F9 DD2F		STD	Y0	,
C3FB DC35		LDD	X1	;copy X-end points
C3FD DD31		STD	X0	,
C3FF 39		RTS		;done

mSCALC: Scale Coordinates

Where:

m - (R)eative or (Absolute  
c - (Y)-coordinate or (X)-coordinate

Inputs:

A - Y/X coordinate

Returns:

D - Y/X coordinate scaled

C400 9650	ASCALY	LDA	VYSCAL	;load Y-scaling value
C402 D008		SUBB	VCNTRY+1	;distance from centroid
C404 2004		BRA	ASCA10	,
C406 964F	ASCALX	LDA	VXSCAL	;load X-scaling value
C408 D006		SUBB	VCNTRX+1	;distance from centroid
C40A 2503	ASCA10	BCS	SCAL10	, negative
C40C 3D	SCAL00	MUL		;perform scaling
C40D 2005		BRA	SCAL20	,
C40F 50	SCAL10	NEGB		;absolute value for scaling
C410 3D		MUL		;unsigned multiply
C411 B0CA99		JSR	NEGD	,fix sign
C414 7ED377	SCAL20	JMP	ASRD07	,RADIX-7/exit

## ; VCTREL: Relative Mode Processing

```
C417 A680    VCTREL LDA      ,X+          ;check for V$ABS
C419 27B0      BEQ      VCTNEW           ; saw V$ABS/simulate V$NEW/done
C41B 8DDA      BSR      NEWOLD           ;copy end points to start points
```

## ; Extract Y-coordinate nibble

```
C41D E61F      LDB      -1,X          ;recover original data byte
C41E 52        ASRB                ;MSNibble into LSNibble (signed)
C420 52        ASRB
C421 57        ASRB
C422 57        ASRB
C423 58        ASRB
C424 DB54      ADDB      TY          ;range (-16:14)
C426 D754      STB      TY          ;effect relative change in absolute
C428 8DD6      BSR      ASCALY         ;scale y-coordinate
C42A D307      ADDB      VCNTRY         ;add Y-centroid
C42C DD33      STD      Y1          ;store as new end point
```

## ; Extract and process X-coordinate nibble.

```
C42E E61F      LDB      -1,X          ;recover original data byte
C430 C40F      ANDB      #Z00001111   ;extract LSNibble
C432 C508      BITR      #Z00001000   ;check "sign bit"
C434 2702      BEQ      VREL20         ;positive
C436 CAFO      ORB      #Z11110000   ;sign extend
C438 58        LSLB      VREL20         ;range (-16:14)
C439 DB52      ADDB      TX          ;effect relative change
C43B D752      STB      TX
C43D 8DC7      BSR      ASCALX         ;scale x-coordinate
C43F D305      ADDB      VCNTRX         ;add X-centroid
C441 DD35      STD      X1          ;store new end point
```

```
C443 BDCA87      JSR      VECTOR         ;draw the vector
C446 20CF      BRA      VCTREL         ;loop until V$ABS is seen
```

; END

```
INCLUD B:TXTSER.ASM
TTL TXTSER - Text Service Routines
```

TXTSTI: Compressed String Output Immediate  
Called via COMSWI.

Inputs: PC - points to compressed

Returns: PC <- PC + (length of string)

C448 AE6C	TXTSTI	LDX 12,S	;load string addr
C44A 3F		SWI	;expand the string
C44B 05		FCB EXPAND	
C44C AF6C		STX 12,S	;update the return addr
C44E 8E0335		LDX #STRING+1	;expanded string addr
C451 8C		FCB SKIP2	;drop into TXTSTR

TXTSTR: Output a character string to the display  
Called via COMSWI.

Input: X - pointer to string

C452 3F	TSTR10	SWI	;print the char
C453 04		FCR OUTCHR	
C454 A680	TXTSTR	LDA ,X+	;get next char
C456 2AFA		BPL TSTR10	;loop until terminator (\$FF)
C458 39		RTS	;exit

TXTCHR: Output a character to the screen  
Called via COMSWI.

Input: A - Char to write

C459 0DB7	TXTCHR	IST TXBFLG	;do standard i/o?
C45B 2603		BNE TCHR10	;nope
C45D CE0390		LDD #TXTPRI	;i/o to primary text area

C460 AE44	TCHR10	LDX P,TXCUR,U	;load cursor position
C462 BDC9B2		JSR TXTXXX	;dispatch to routine
C465 AC42		CMPX P,TXCONT,U	;need to scroll?
C467 2503		BLO TCHR99	;nope
C469 BDC9D4		JSR TXTSCR	;scroll one line
C46C AF44	TCHR99	STX P,TXCUR,U	;update cursor position
C46E 39		RTS	;go home

;  
END

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
EXPAND - Byte Expansion Routine

PAGE 61

```

INCLUD B:EXPAND.ASM
EXPAND: 5-Bit Character String Expansion Routine

Inputs:
    X - pointer to 5-bit string
    U - expansion buffer addr (EXPANO only)
Returns:
    X - points past 5-bit string
    STRING - expanded 5-bit character string

C46F CE0335 EXPANX LDU    #STRING+1      ;default expansion buffer addr
C472          EXPANO EQU    $                 ;secondary entry point

; Expand 5-Bit Character String
C472 315F      LEAY   -1,U             ;IY <- 5-bit offset counter address
C474 6FA4      CLR    0,Y             ;reset 5-bit char offset
C476 8D14      BSR    GETFI4         ;get first 5-bit char
C478 1F98      TFR    B,A             ;use as byte count
C47A 8D10      BSR    GETFI4         ;get the next 5-bit character
C47C E7C0      STB    ,U+             ;Buffer bytes...
C47E 4A        DECA              ;decrement byte count
C47F 2AF9      BPL    EXPANO          ;loop until done
C481 A7C4      STA    ,U              ;terminate buffer with $FF

; Strings Terminate on Byte Boundaries
C483 6DA4      TST    0,Y             ;are we already on byte boundary?
C485 2702      BEQ    EXPANO          ;yes
C487 3001      LEAX   1,X             ;advance to next byte

; Found our man...
C489 AF66      EXPANO STX    6,S             ;return parameter
C48B 39        RTS                  ;that's all folks!!!

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
EXPAND - Byte Expansion Routine

PAGE 62

GETFIV: Get the next 5-bit Char

Inputs:

X - pointer to compressed string (byte pointer)  
0,Y - 5-bit offset counter

Returns:

B - 5-bit character  
X - updated  
0,Y - updated

C48C 3442      GETFIV PSHS A,U                ;save regs

Dispatch to correct routine

C48E A6A4      LDA 0,Y                        ;use 5-bit char offset as index  
C490 CEC4A2      LDU #FIVDSP                  ;table base addr  
C493 A6C6      LDA A,U                        ;load offset value  
C495 ADC6      JSR A,U                        ;dispatch

Update 5-bit Offset Counter

C497 A6A4      LDA 0,Y                        ;advance 5-bit offset counter  
C499 4C      INCA                                ;  
C49A 8407      ANDA #7                        ;MOD 8  
C49C A7A4      STA 0,Y                        ;  
Return ByteC49E C41F      ANDB #\$1F                        ;use only lower 5-bits  
C4A0 35C2      FULS A,U,PC                    ;restore regs/exit

## ; 5-bit Character Extraction Routine Dispatch Table

C4A2	08	FIVDSP	FCB	CHAR00-FIVDSP
C4A3	0E		FCB	CHAR10-FIVDSP
C4A4	13		FCB	CHAR20-FIVDSP
C4A5	17		FCB	CHAR30-FIVDSP
C4A6	1C		FCB	CHAR40-FIVDSP
C4A7	21		FCB	CHAR50-FIVDSP
C4A8	25		FCB	CHAR60-FIVDSP
C4A9	2A		FCB	CHAR70-FIVDSP

## ; 5-bit Char Extraction Routines

C4AA	E684	CHAR00	LDB	,X	;load byte
C4AC	54	CHAR02	LSRB		;shift into position
C4AD	54	CHAR04	LSRB		;:
C4AE	54	CHAR06	LSRB		;:
C4AF	39		RTS		;done
C4B0	EC80	CHAR10	LDD	,X+	;load byte (across boundary)
C4B2	7ED379		JMP	ASRD6	;shift into position/exit
C4B5	E684	CHAR20	LDB	,X	;load byte
C4B7	20F5	BRA	CHAR06		;shift into position/exit
C4B9	EC80	CHAR30	LDD	,X+	;load byte (across boundary)
C4BB	7ED37D		JMP	ASRD4	;shift into position/exit
C4BE	EC80	CHAR40	LDD	,X+	;load byte (across boundary)
C4C0	7ED377		JMP	ASRD7	;shift into position/exit
C4C3	E684	CHAR50	LDB	,X	;load byte
C4C5	20E6	BRA	CHAR04		;shift into position/exit
C4C7	EC80	CHAR60	LDD	,X+	;load byte (across boundary)
C4C9	7ED37B		JMP	ASRD5	;shift into position/exit
C4CC	E680	CHAR70	LDB	,X+	;load byte
C4CE	39		RTS		;already in position/exit
			END		

INCLUD B:RANDOM.ASM

!!!!!! RANDOM NUMBER GENERATOR !!!!!!!

General Purpose Random Number Generator  
by the Polynomial Method.

Returns an 8-bit random number in ACCA.

C4CF BE0008	RANDOX	LDX	\$0008	;initialize shift counter
C4D2 5F	RND1	CLRB		;clear feedback byte
C4D3 108E0008		LDY	\$0008	
C4D7 966D		LDA	SEED+2	;load byte containing feedback bits
C4D9 84E1		ANDA	#\$E1	;mask all except feedback bits
C4DB 48	RND2	LSLA		;shift feedback bits into carry
C4DC 2401		BCC	RND3	
C4DE 5C		INCB		;count number of ones
C4DF 313F	RND3	LEAY	-1,Y	;loop eight times
C4E1 26F8		BNE	RND2	
C4E3 54		LSRB		;shift LSR of feedback into carry
C4E4 096B		ROL	SEED	;24-bit shift
C4E6 096C		ROL	SEED+1	
C4E8 096D		ROL	SEED+2	
C4EA 301F		LEAX	-1,X	;shift eight times
C4EC 26E4		BNE	RND1	
C4EE 966B		LDA	SEED	;grab output value
C4F0 A763		STA	3,S	;modify A on the way out...
C4F2 39		RTS		;exit
		END		

```

        INCLUDE B:CLEAR.ASM
;----- DISPLAY CLEARING ROUTINE -----
C4F3 DE09    ZFLIPX   LDU     FLIP      ; current VDB ptr
C4F5 8C       FCB      SKIP2    ; 
C4F6 DE0B    ZFLOPX   LDU     FLOP      ; alternate VDB ptr
C4F8 D62C    LDB      VGINV    ; load inverse mode value
C4FA 8D1B    BSR      CLEAR    ; clear the screen
C4FC EF6A    STU      10,S    ; return VDB ptr
C4FE 39       RTS      ; bye

C4FF BE0388    CLRSTX   LDX     #TXTSTS ; home cursor
C502 CED87C    LDU      #STSVD8 ; clear both screens
C505 2006    BRA      CLRSUB   ;
;----- 
C507 BE0390    CLRPRX   LDX     #TXTPRI ; home cursor
C50A CED888    LDU      #PRIVDB ; clear both screens
;----- 
C50B 6F04    CLRSUB   CLR      P.TXCUR,X ; home cursor
C50F 6F05    CLR      P.TXCUR+1,X ;
C511 E606    LDR      P.TXINV,X ; load inverse flag value
C513 8D02    BSR      CLEAR    ; clear the screen
C515 3346    LEAU     VD.LEN,U ; advance to next VDB

C517 3476    CLEAR    PSHS    A,B,X,Y,U ; save regs
C519 1D       SEX      ; propagate ACCB across ACCA
C51A 1F02    TFR    B,Y    ; copy pattern to IY
C51C 30C4    LEAX   P.VDBAS,U ; load ptr to base addr
C51E EE42    LDU    P.VDEND,U ; load last addr
;----- 
C520 3626    CLEAR1   PSHU    B,Y    ; clear 4 bytes at a time
C522 11A384    CMPU    ,X    ;
C525 26F9    BNE    CLEAR1  ; loop until done
C527 30F6    PULS   A,B,X,Y,U,PC ; restore regs/exit
;----- 
;----- END -----

```

INCLUD B:HUPDAT.ASM

HUPDAX: Update the heart rate

Note:

$$J = \frac{P * 64}{F + (D * 2)} - 19$$

Where:

J = jiffy delay count (HEARTR)  
 P = power (PPOW)  
 D = damage (PDAM)

All calculations are done with 24-bit arithmetic.

Temp Storage:

T0 - denominator (24-bit)  
 T1 -  
 T2 -  
 T3 - numerator (24-bit)  
 T4 -  
 T5 -  
 T6 - dividend (8-bit)

C529

HUPDAX EQU \$ ;the beginning

Calculate numerator

C529	OFC1	HUPD00	CLR	T0	;((p*64)
C52B	DC17		LDD	PPOW	;
C52D	DDC2		STD	T1	;
C52F	8606		LDA	#6	;64=2**6
C531	08C3	HUPD10	LSL	T2	least significant byte
C533	09C2		ROL	T1	middle byte
C535	09C1		ROL	T0	most significant byte
C537	4A		DECA		;
C538	26F7		BNE	HUPD10	;

Calculate denominator

C53A	OFC4		CLR	T3	; (d*2)
C53C	DC21		LDD	PDAM	;
C53E	DDC5		STD	T4	;
C540	08C6		LSL	T5	least significant byte
C542	09C5		ROL	T4	middle byte
C544	09C4		ROL	T3	most significant byte
C546	DC17		LDD	PPOW	(pt(d*2))
C548	D3C5		ADDD	T4	;
C54A	DDC5		STD	T4	;
C54C	D6C4		LDR	T3	;
C54E	C900		ADC8	#0	;
C550	D7C4		STB	T3	;

```

; Calculate dividend
C552 0FC7    CLR    T6      ;reset dividend
C554 DCC2    HUPD20   LDD    T1      ;16 LSBits of numerator
C556 93C5    SUBD   T4      ;subtract 16 LSBits of denominator
C558 DDC2    STD    T1      ;store back into numerator
C55A 96C1    LDA    T0      ;8 MSBBits of numerator
C55C 92C4    SBCA   T3      ;sub 8 LSBits of denominator + carry
C55E 97C1    STA    T0      ;store back into numerator
C560 0CC7    INC    T6      ;bump dividend
C562 24F0    BCC    HUPD20   ;loop until numerator goes negative

; Update the heartrate
C564 96C7    LDA    T6      ;load dividend
C566 8013    SUBA   #19     ;subtract nineteen
C568 97AF    STA    HEARTR  ;store as new heartrate

; Are we unconscious (fainted)?
C56A 0II28    TST    FAINT   ;check the flag
C56C 2627    BNE    HUPD40   ;    we are out cold!
C56E 8103    CMPA   #3      ;should we faint?
C570 2E3C    BGT    HUPD90   ;    nope

; Special Effects - Fainting
C572 3F      SWI    ;clear primary text area
C573 0B      FCB    CLRPPRI
C574 966E    LDA    RLIGHT  ;save RLIGHT for later
C576 9770    STA    GLIGHT
C578 0A6F    HUPD30   DEC    MLIGHT  ;reduce lighting
C57A AD9F02B2 JSR    CDSMPMOD  ;show the new view
C57E 0AB4    DEC    UPDATE   ;request and wait for screen update
C580 13      SYNC
C581 0A6E    DEC    RLIGHT  ;lower lighting
C583 966E    LDA    RLIGHT  ;done?
C585 81F8    CMPA   #-8      ;    nope
C587 2EEF    BGT    HUPD30
C589 3F      HUPD32   SWI    ;insure screen is blank
C58A 09      FCB    ZFLOP
C58B 0AB4    DEC    UPDATE
C58D 0A28    DEC    FAINT   ;indicate we have fainted
C58F 0FBC    CLR    KBDHDR  ;insure empty buffer
C591 0FB0    CLR    KBDTAL
C593 2019    BRA    HUPD90

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
HUPDAT - Heart Update Routine

PAGE 69

```
C5D3 0F28      CLR    FAINT      ;force GAME restart on char
C5D5 0A77      DEC    AUTFLG
C5D7 20FE      BRA    $
END
```

;then loop forever

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
STATUS - Status Line Update Routine

PAGE 70

INCLUD B:STATUS.ASM

; STATUSX: Status line Update Routine

C5D9 CE0388	STATUX	LDU #TXTSTS	; i/o to status line
C5DC 0AB2		DEC TXBFLG	; ;
C5DE 962C		LDA VDGINV	; set status line inverse byte
C5E0 43		COMA	; ;
C5E1 A746		STA P.TXINV,U	; ;
; Left hand			
C5E3 4F		CLRA	; clear left hand text area
C5E4 5F		CLRR	; ;
C5E5 BD22		BSR SPACES	; ;
C5E7 ED44		STD P.TXCUR,U	; home the cursor
C5E9 9E1D		LDX PLHAND	; display what left hand is holding
C5EB BD2A		BSR OBJNAM	; ;
C5ED 3F		SWI	; ;
C5EE 03		FCB OUTSTR	; print the string
; Right hand			
C5EF CC0011		LDD #17	; clear right hand text area
C5F2 BD15		BSR SPACES	; ;
C5F4 9E1F		LDX PRHAND	; determine right hand object
C5F6 BD1F		BSR OBJNAM	; ;
C5F8 1F12		TFR X,Y	; make copy of string ptr
; Right justify			
C5FA CC0021		LDD #33	; count backwards
C5FD 5A	STAT10	DEC B	; decrement
C5FE 6DAO		TST Y+	; look for end of string
C600 2AFB		BPL STAT10	; not found
C602 ED44		STD P.TXCUR,U	; position cursor
C604 3F		SWI	; ;
C605 03		FCB OUTSTR	; print the string
C606 0FB7		CLR TXBFLG	; resume standard i/o
C608 39		RTS	; done

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
STATUS - Status Line Update Routine

PAGE 70

INCLUD B:STATUS.ASM

; STATUX: Status line Update Routine

C5D9 CE0388	STATUX	LDD	#TXTSTS	;i/o to status line
C5DC 0A87		DEC	TXBFLG	;
C5DE 962C		LDA	VDGINV	;set status line inverse byte
C5E0 43		COMA		;
C5E1 A746		STA	P.TXINV,U	;
; Left hand				
C5E3 4F		CLRA		;clear left hand text area
C5E4 5F		CLRB		;
C5E5 8D22		BSR	SPACES	;
C5E7 ED44		STD	P.TXCUR,U	;home the cursor
C5E9 9E1D		LDX	PLHAND	;display what left hand is holding
C5EB 8D2A		BSR	OBJNAM	;
C5ED 3F		SWI		;
C5EE 03		FCB	OUTSTR	;print the string
; Right hand				
C5EF CC0011		LDD	#17	;clear right hand text area
C5F2 8D15		BSR	SPACES	;
C5F4 9E1F		LDX	PRHAND	;determine right hand object
C5F6 8D1F		BSR	OBJNAM	;
C5F8 1F12		TFR	X,Y	;make copy of string ptr
; Right justify				
C5FA CC0021		LDD	#33	;count backwards
C5FD 5A	STAT10	DEC8		;decrement
C5FE 6DA0		TST	,Y+	;look for end of string
C600 2AFB		BPL	STAT10	;not found
C602 ED44		STD	P.TXCUR,U	;position cursor
C604 3F		SWI		;
C605 03		FCB	OUTSTR	;print the string
C606 0FB7		CLR	TXBFLG	;resume standard i/o
C608 39		RTS		;done

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
STATUS - Status Line Update Routine

PAGE 71

SPACES: Print 15 spaces

Inputs:

I = cursor position  
U = TXB pointer

C609 3406	SPACES	PSHS	A,B	;save regs
C60B ED44		STD	P,TXCUR,U	;position cursor
C60D CC000F		LDD	*I,SP*256) DR 15	;space,,15
C610 3F	SPACE1	SWI		
C611 04		FCR	OUTCHR	;output the char
C612 5A		DEC8		
C613 26FB		BNE	SPACE1	;LOOP
C615 3586		PULS	A,B,PC	;restore regs

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
STATUS - Status Line Update Routine

PAGE 72

```

; OBJNAM: Return string pointer for given object
; Inputs:   X = OCB pointer
; Returns:  X = object name string pointer

C617 3466    OBJNAM  PSHS    A,B,Y,U
C619 3184    LEAY    0,X      ;copy OCB ptr (setting CC)
C61B 2605    BNE     ONAM10   ;valid pointer
;
C61D BEC650  ; LDX     #M$EMPT  ;empty string
C620 201A    BRA     ONAM99   ;
;
; Decide what form to use
;
C622 CE0313  ONAM10  LDW     #TOKEN   ;buffer addr
C625 602B    TST     P,OCTREV,Y ;has the object been revealed?
C627 2609    BNE     ONAM20   ;  nope - use generic only
;
; Copy <adjective> to buffer
;
C629 A629    LDA     P,OCTYP,Y ;load type
C62B 8ED8F4  LDX     #ADJTAB+1 ;packed strings
C62E 8D0E    BSR     COPY$    ;copy to buffer
;
C630 6F5F    CLR     -1,U    ;use space as separator
;
; Copy <generic> to buffer
;
C632 A62A    ONAM20  LDA     P,OCCLS,Y ;class determines generic name
C634 8ED96B  LDX     #GENTAB+1 ;packed strings
C637 8D05    BSR     COPY$    ;copy to buffer
;
C639 8E0313  ; LDX     #TOKEN   ;return Buffer addr
C63C 35E6    ONAM99  PULS   A,B,Y,U,PC ;EXIT
;
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
STATUS - Status Line Update Routine

PAGE 73

```
; COPY$: Copy String from Packed Array to Buffer
C63E 3412    COPY$   PSHS   A,X      ;save regs
C640 3F        COPY10  SWI
C641 05        FCB    EXPAND   ;expand the 5-bit string
C642 4A        DECA
C643 2AFB      BPL    COPY10   ;is this the string we want?
C645 BE0336      LDX    #STRING+2 ;nope
;                               ;load string addr

; Copy string to buffer
C648 A680      COPY20  LDA    ,X+      ;load next char
C64A A7C0      STA    ,U+      ;copy to destination
C64C 2AFA      BPL    COPY20   ;loop
;                               ;exit
C64E 3592      PULS   A,X,PC
C650 050D101419M$EMPT  FCB    $05,$0D,$10,$14,$19,$FF ;'EMPTY'
;                               ;END
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PUPDAT - Update the Screen

PAGE 74

```

        INCLUDE B:PUPDAT.ASM
; PUPDAX: Update the screen

C656 0D28    PUPDAX  TST     FAINT
C658 2605      BNE     PUPD99   ;are we unconscious?
;                                ;yes - don't do anything

C65A BD04    ;
C65C 0AB4    BSR     PUPSUB  ;generate next display in FLOP
C65E 13       DEC     UPDATE   ;request screen switch
C65F 39       SYNC    ;wait for it to happen
;                                ;exit

C660 3476    PUPSUB  PSHS    A,B,X,Y,U ;save some regs
C662 DC26      LDD     PRLITE  ;load player regular/magic light values
C664 DE24      LDU     PTORCH  ;load torch OCB ptr
C666 2704      BEQ     PSUB10 ;no torch!
C668 AB47      ADDA    P.OCXXX+1,U ;add regular light component
C66A EB48      ADDB    P.OCXXX+2,U ;add magic light component
C66C DD6E      PSUB10 STD     RLIGHT ;store player values
;                                ;create new screen in FLOP
C66E AD9F02B2  JSR     [DSPMOD]
C672 35F6      PULS    A,B,X,Y,U,PC ;restore regs/go home
END

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.0B

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MISC - Misc Service Routines

PAGE 75

INCLUD B:MISC.ASM

TTL MISC - Miscellaneous SWI Routines

PROMPT: Prompt the Human for Input

C674 8E C67A	PROMPX	LDX	#M\$PROM1	;y'ungry?
C677 3F		SWI		;m'ungry...
C678 03		FCB		
C679 39		RTS		;t'seet...

C67A 1E 1E	M\$PROM1	FCB	I.CR,I.DOT	
C67C 1C 24 FF	M\$CURS	FCB	I.BAR,I.BS,\$FF	

WAITX: Delay Processing 1.5 Second

C67F C651	WAITX	LDB	#81	;delay for 1.35 seconds
C681 13	WAIT10	SYNC		;wait for a jiffy to occur
C682 5A		DEC8		
C683 26 FC		BNE	WAIT10	;decrement and loop
C685 39		RTS		

ZERO: Zero a block of RAM (Called via COMSWI)  
NEGРАM: Set a Block of RAM to negative ones (\$FF)

Inputs:

X - starting addr  
U - ending addr

C686 4F	ZEROX	CLRA		;zero RAM
C687 8C		FCB	SKIP2	
C688 86 FF	NEGРАX	LDA	#\$FF	;set RAM to negative ones
C68A A7 80	SETRAX	STA	,X+	
C68C AC 6A		CMPX	10,S	;set the next byte
C68E 26 FA		BNE	SETRAX	;are we done?
C690 39		RTS		;nope
				;done

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MISC - Misc Service Routines

PAGE 76

; BASICX: Transfer to BASIC ROM Routine

C691 5F	BASICX	CLRB		;switch to BASIC DP
C692 1F9B		TFR	B,DP	
	;			,
C694 EE6C		LDU	12,S	;load return addr
C696 E6C0		LDB	,04	;load BASIC parameter
C698 EF6C		STU	12,S	;update return addr
	;			,
C69A CEA000		LDU	##A000	;BASIC ROM Dispatch Table
C69D ADDS		JSR	[B,UI]	;dispatch to basic
	;			,
C69F A763		STA	3,S	;return ACCA
C6A1 AF66		STX	6,S	;return IX
C6A3 39		RTS		

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MISC - Misc Service Routines

PAGE 77

## ; WIZIX: Fade In the Wizard

## Inputs:

X - vector list addr

C6A4 0FB1	WIZIX	CLR	HBEATF	;turn off the heart
C6A6 3F		SWI		;zap the status line
C6A7 0A		FCB	CLRSTS	;clear the primary text area
C6A8 3F		SWI		
C6A9 0B	WIZIXO	FCB	CLRPRI	
C6AA CC8080		LDD	#\$8080	;manually set range to one
C6AB DD4F		STD	VXSCAL	

## ; Immediate or Fade in?

C6AF B69E		LDB	FADFLG	;check flag
C6B1 2704		BEQ	WIZI10	;show up now!
C6B3 C620		LDB	#32	;fade-in
C6B5 0A9C		DEC	NOISEF	;buzzing noise
C6B7 BD1E	WIZI10	BSR	WIZZES	;display the wizard
C6B9 5A		DECB		;decrease the amount of fade
C6BA 5A		DEC8		
C6BB 2AFA		BPL	WIZI10	;loop until done
C6BD 0F9C		CLR	NOISEF	;turn off noise
C6BF 0F9E		CLR	FADFLG	;reset flag
C6C1 3F	WIZI20	SWI		;loud explosion
C6C2 1B		FCB	ISOUND	
C6C3 18		FCB	A\$EXP1	
C6C4 39		RTS		;done

## ; WIZOX: Fade Out the Wizard

## Inputs:

X - vector list addr

C6C5 3F	WIZOX	SWI		;clear the primary text area
C6C6 0B		FCB	CLRPRI	
C6C7 BDF8		BSR	WIZI20	;loud explosion
C6C9 5F		CLRB		;loud to soft, fade out
C6CA 0A9C		DEC	NOISEF	;more noise
C6CC BD09	WIZO10	BSR	WIZZES	;display the wizard
C6CE 5C		INCB		;increase amount of fade
C6CF 5C		INCB		
C6D0 C120		CMPB	#32	;until done
C6D2 24F8		BNE	WIZO10	
C6D4 0F9C		CLR	NOISEF	;turn off noise
C6D6 39		RTS		;done

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MISC - Misc Service Routines

PAGE 78

WIZZES: Draw the Wizard

Inputs:

B - amount of fade/noise  
X - vector list addr

C6D7 3450	WIZZES	PSHS X,U	; save regs
C6D9 D720		STB VCTFAD	; set fade value
C6DB D790		STB NOISEV	; set buzz volume
C6DD 3F		SWI	; blank the alternate screen
C6DE 09		FCB ZFLOP	.
C6DF 3F		SWI	.
C6E0 01		FCB	.
C6E1 0AB4		VCTLST	.
C6E3 13		DEC UPDATE	; flag for update
C6E4 35D0		SYNC	; wait for flip
		PULS X,U,PC	; restore regs/exit

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MISC - Misc Service Routines

PAGE 79

| PREPAR: Display "PREPARE!" on EXAMINE Screen

C6E6 B0D4B9	PREFAX	JSR	EXAMIO	prepare to do i/o to EXAMINE screen
C6E9 CC012C		LBN	#32*9+12	set cursor position
C6EC ED44		STD	P.TXCUR,U	
C6EE 3F		SWI		deliver the message
C6EF 02		FCR	OUTSTI	
C6F0 3C		FCB	Z00111100	PREPARE!
C6F1 24		FCB	Z00100100	PREPARE!
C6F2 58		FCB	Z01011000	PREPARE!
C6F3 06		FCB	Z00000110	PREPARE!
C6F4 45		FCB	Z01000101	PREPARE!
C6F5 D8		FCB	Z11011000	PREPARE!
C6F6 0FB7	CLR	TXBFLG		restore standard i/o
C6F8 0AB4	DEC	UPDATE		request screen update
C6FA 39	RTS			,bye.
	END			

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
OBIRTH - Object Birth Routine

PAGE 80

```

        INCLUDE B:OBIRTH.ASM
        ; OBIRTX: Create an Object

        Inputs:
          A - Object type
          B - Level number
        Returns:
          X - Pointer to OCB
          - Create Object Control Block (OCB)

C6FB      OBIRTX EQU    $
        ; Find the next available OCB

C6FB BE0F
C6FD EF66
C6FF 304E
C701 9F0F      LDU    OCBPTR      ;load current OCB ptr
                  STU    6,S       ;return as parm
                  LEAX   OC,LEN,U  ;advance OCB ptr
                  STX    OCBPTR

        ; Fill in the new OCB

C703 A749
C705 E744
C707 3F
C708 18      STA    P,OCTYP,U  ;store the object type
                  STB    P,OCLVL,U  ;store the level number
                  SWI    ,          ;fill-in regular parameters
                  FCB    OCBFIL    ;           :
                  ;           :

        ; Implement Generic Values (as necessary)

C709 E64A
C70B BEC719
C70E A685
C710 2B06      LDR    P,OCCLS,U  ;use class as index into GENVAL
                  LDX    #GENVAL    ;translation table base addr
                  LDA    B,X       ;load translation value
                  BMI    OBIR99    ;minus means don't change
                  ;           :

C712 E64B
C714 3F
C715 18
C716 E74B      LDR    P,OCREV,U  ;preserve relevation requirements
                  SWI    ,          ;overwrite with generic values
                  FCB    OCBFIL    ;           :
                  STB    P,OCREV,U ;           :

C718 39      OBIR99 RTS      ;bye

        ; GENVAL: Generic/Specific Translation Table

C719
C719 FF      GENVAL EQU    $      ;table begins here
                  FCB    -1       ;(0) K,FLAS
C71A FF      FCB    -1       ;(1) K,RING
C71B FF      FCB    -1       ;(2) K,SCRD
C71C 10      FCB    T,SHI4   ;(3) K,SHIE
C71D 11      FCB    T,SWO3   ;(4) K,SWOR
C71E OF      FCB    T,TOR4   ;(5) K,TORC

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
OBIRTH - Object Birth Routine

PAGE 81

```

; OCBFIX: Fill in OCB information
; Inputs:
;   A = object type
;   U = ptr to OCB

C71F      OCBFIX EQU    $          ;where everything starts
; Copy Regular Parameters
C71F 48      LSLA              ;calculate index (times OD.LEN)
C720 48      LSLA              ;(OD.LEN = 4)
C721 8EDA00   LDX   #ODBTAB    ;table base addr
C724 3186     LEAY   A,X       ;IY - source addr
C726 304A     LEAX   P,OCCLS,U ;IX - destination addr
C728 8604     LDA   #OD.LEN   ;ACCA - byte count
C72A BBC04B   JSR    COPY      ;copy it

; Process Special Parameters
C72B BE11A60   LDX   #XXXTAB-4 ;special parameter table
C730 3004     OFIL10  LEAX   4,X   ;advance ptr
C732 A684     LDA   0,X       ;load special object type
C734 2B0C     BMI   OFIL99    ;minus means we are done
C736 A163     CMPA   3,S     ;match current object type?
C738 26F6     BNE   OFIL10    ;nope - loop until match
;                               ;copy special parameters
C73A EC01     LDD   1,X       ;
C73C ED46     STD   P,OCXXX,U ;      ;
C73E A603     LDA   3,X       ;      ;
C740 A748     STA   P,OCXXX+2,U ;      ;

C742 39      OFIL99  RTS      ;restore regs/exit
END

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PLOOK - Look Command

PAGE 82

```
INCLUD B:PLOOK.ASM
;!!!!!! L O O K C O M M A N D !!!!!!
```

; PLOOK: Forward Looking View

C743 3F	INIVUX	SWI	clear the status line
C744 0A		FCB CLRSTS	,
C745 3F		SWI	clear the primary text area
C746 0B		FCB CLRPRI	,
C747 3F		SWI	calculate the heartrate
C748 0C		FCB HUPDAT	,
C749 0CAE		INC HEARTC	insure immediate flash
C74B 0AAD		DEC HEARTF	,
C74D 0AB1		DEC HBEATF	turn on visual flash
C74F 3F		SWI	,
C750 0D		FCB STATUS	turn on audio thump
			,
C751 8ECE66	PLOOK	LDX #VIEWER	update the status line
C754 9FB2		STX DSPMOD	,
C756 3F		SWI	forward looking display
C757 0E		FCB PUPDAT	,
C758 39		RTS	update the screen
	;		,
		END	exit

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
NEWLVL - Create a New Dungeon Level

PAGE 83

INCLUD B:NEWLVL.ASM

TTL NEWLVL - Create a New Dungeon Level

NEWLVX: Create a New Dungeon Level

Inputs:

A - level to be created

C759

NEWLVX EQU \$

Calculate new creature matrix index

C759 9781  
C75B C60C  
C75D 30  
C75E C30398  
C761 DD82

```

    STA      LEVEL      ;store the level number
    LDB      #CTYPES   ;number of creatures
    MUL
    ADDD   #CMXLND   ;:
    STB      CMXPTR   ;:

```

Calculate Vertical Features Index

C763 D681  
C765 SECFFD  
C768 9F86  
C76A A680  
C76C 2AF0  
C76E 5A  
C76F 2AF7

```

    LDB      LEVEL      ;load the level number
    LDX      #VFTTAB   ;table base addr
    NLVL10 STX      VFTPTR   ;assume this is correct
    NLVL12 LDA      ,X+      ;search for negative
    BPL      NLVL12   ;:
    DECB
    BPL      NLVL10   ;done?
                                ;nope

```

Armagedon!

C771 8E03D4  
C774 CE05F4  
C777 3F  
C778 11

```

    LDX      #CCBLND   ;table base addr
    LDU      #CCBEND   ;ending addr
    SWI      ZERO      ;clear the entire CCB table

```

Create a new world...

C779 BD0533  
C77C BDCC9C

```

    JSR      SYSTCB   ;create new system TCBs
    JSR      DGNGEN   ;create the new level

```

Give this new world LIFE...

C77F DE82  
C781 860B  
C783 E4C6  
C785 2706  
C787 BD0FA5  
C78A 5A  
C78B 26FA  
C78D 4A  
C78E 2AF3

```

    LDW      CMXPTR   ;load CMXLND index
    LDA      #CTYPES-1 ;start with most ferocious creatures
    NLVL30 LDB      A,U      ;number of this type of creature
    NLVL32 BEQ      NLVL34   ;done
    NLVL32 JSR      CBIRTH   ;give birth to a creature
    NLVL32 DECB
    NLVL32 BNE      NLVL32   ;one less creature to create
    NLVL34 DECA
    NLVL34 BPL      NLVL30   ;until we are done
                                ;advance to next creature type
                                ;until done

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
NEWLVL - Create a New Dungeon Level

PAGE 84

```

; Randomly Attach Creature-Owned Objects
C790 C03C3    LDU    #CCBLND-CC,LEN   ;load of creatures
C793 0F91     CLR    #FINDEF
C795 B1CF63    NLVL40 JSR    FNDOBJ   ;start at top of object list
C798 271C     BEQ    NLVL50      ;find the next object on our level
C79A 6D05     TST    P.OCOWN,X   ;exhausted objects
C79C 2AF7     BPL    NLVL40      ;is this object creature owned
                                ;nope

; Find the next creature
C79E 33C811   NLVL42 LEAU   CC,LEN,U   ;find a creature
C7A1 118305F4 CMPU   #CCBEND
C7A5 2E03     BLO    NLVL44      ;exhausted table?
C7A7 C03D4     LDU    #CCBLND
                                ;nope
                                ;start at top

C7AA 6D4C     NLVL44 TST    P.CCUSE,U   ;this creature alive?
C7AC 27F0     BEQ    NLVL42      ;nope

; Attach Object to Creature
C7AE EC48     LDD    P.CCOBJ,U   ;load creature object ptr
C7B0 AF48     STX    P.CCOBJ,U   ;insert object
C7B2 ED84     STD    P.OCPTR,X
C7B4 20DF     BRA    NLVL40      ;up top for another

; Determine VIDGINV Setting
C7B6 9681     NLVL50 LDA    LEVEL
C7B8 8401     ANDA   #BIT0
C7BA 40       NEGA
C7BB 972C     STA    VIDGINV
C7BD B70396   STA    P.TXINV+TXTPRI
C7C0 B70386   STA    P.TXINV+TXTEXA
C7C3 43       COMA
C7C4 B7038E   STA    P.TXINV+TXTSTS
                                ;load the level number
                                ;determine even/odd (0/1)
                                ;convert to mask ($00/$FF)
                                ;set the inverse byte
                                ;update primary TXB inverse flag
                                ;update examine TXB inverse flag
                                ;invert the "inverse" flag
                                ;update status TXB inverse flag

C7C7 39       RTS
END
                                ;done

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 85

```
INCLUD B:SOUNDS.ASM
      TTL SOUNDS - Sound Generator
```

```
!!!!!!SOUND GENERATOR!!!!!!
!!!!!!SOUND GENERATOR!!!!!
```

```
SOUNDS: Sound Generation Routine
Called via COMSWI.
```

```
Inputs:
```

```
A = sound identification code
B = volume level
```

```
"Actual" sounds entry point
```

C7C8 AE6C	SOUNDI	LDX	12,S	;load return addr
C7CA A680		LDA	,X+	;load SOUND parameter byte
C7CC AF6C		STX	12,S	;update return addr
C7CE C6FF		LDB	#\$FF	;use full volume
 ;				
C7D0	SOUNDX	EQU	\$	
C7D0 D761		STB	SNVOL	;store volume parameter
C7D2 8EC7DC		LDX	#SNITAB	;table base addr
C7D5 48		LSLA		;*2 for word offsets
C7D6 AD96		JSR	[A,X1]	;dispatch to correct routine
C7D8 2FFF20		CLR	P.PIIOA+PIAS\$1	;leave DAC at zero
C7D8 39		RTS		;bye

## Sound Dispatch Macros

```

SND      MACR
A$`0    EQU     FOO
FOO      SET     FOO+1
        XDEF    A$`0
        FIB     \1
ENDM

```

```
FOO      SET     0
```

## Sound Dispatch Table

C7DC	SNDTAB	EQU	\$	;table starts here
------	--------	-----	----	--------------------

## Creature Sounds

0000	A\$SQKO	SND	0	SQKO,SQUEAK	;spider
C7DC C82B		EQU		SQUEAK	;spider
		FDB		,	*
0001	A\$RTLO	SND	1	RTLO,RATTLE	;viper
C7DE C850		EQU		RATTLE	;
		FDB		,	:
0002	A\$RORO	SND	2	RORO,GROWL	;stone giant 1
C7E0 C951		EQU		GROWL	;
		FDB		,	:
0003	A\$BEPO	SND	3	BEPO,BEOOP	;blob
C7E2 C83C		EQU		BEOOP	;
		FDB		,	:
0004	A\$KLKO	SND	4	KLKO,KLANK	;Knight 1
C7E4 C8E2		EQU		KLANK	;
		FDB		,	:
0005	A\$ROR1	SND	5	ROR1,GRAWL	;stone giant 2
C7E6 C955		EQU		GRAWL	;
		FDB		,	:
0006	A\$RTL1	SND	6	RTL1,PSSST	;scorpion
C7E8 C84A		EQU		PSSST	;
		FDB		,	:
0007	A\$KLK1	SND	7	KLK1,KKLANK	;Knight 2
C7EA C8DE		EQU		KKLANK	;
		FDB		,	:
0008	A\$PSHT	SND	8	PSHT,PSSHT	;wraith
C7EC C84D		EQU		PSSHT	;
		FDB		,	:
0009	A\$ROR2	SND	9	ROR2,SNARL	;balrog
		EQU		,	:

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 87

C7EE C959		FDB	SNARL	; .
000A C7F0 C877	A\$SQK1	SND EQU FDB	SQK1,BDLBDL 10 BDLBDL	;wizzard 1 ; ;
000B C7F2 C877	A\$SQK2	SND EQU FDB	SQK2,BDLBDL 11 BDLEDL	;wizzard 2 ; ;

## Object Sounds

000C	SNDOBJ	EQU	12	
000C C7F4 C80A	A\$FLAS	SND EQU FDB	FLAS,GLUGLG 12 GLUGLG	;flask ; ;
000D C7F6 C811	A\$RING	SND EQU FDB	RING,PHASER 13 PHASER	;ring ; ;
000E C7F8 C827	A\$SCRO	SND EQU FDB	SCRO,WHOOP 14 WHOOP	;scroll ; ;
000F C7FA C81A	A\$SHIE	SND EQU FDB	SHIE,CLANG 15 CLANG	;shield ; ;
0010 C7FC C8A6	A\$SWOR	SND EQU FDB	SWOR,WHOOSH 16 WHOOSH	;sword ; ;
0011 C7FE C8B2	A\$TORC	SND EQU FDB	TORC,CHUCK 17 CHUCK	;torch ; ;

## Miscellaneous Sounds

0012 C800 C93F	A\$KLK2	SND EQU FDB	KLK2,KLINK 18 KLINK	;hit creature ; ;
0013 C802 C8E6	A\$KLK3	SND EQU FDB	KLK3,CLANK 19 CLANK	;hit player ; ;
0014 C804 C8F4	A\$THUD	SND EQU FDB	THUD,THUD 20 THUD	;hit wall ; ;

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 88

0015	C86D	A\$EXP0	SND	EXPO,BANG	;explosion 0
			EQU	21	
			FDB	BANG	:
0016	C88A	A\$EXP1	SND	EXP1,KABOOM	;explosion 1
			EQU	22	
			FDB	KABOOM	:

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 89

```

;!!!!!!SOUND GENERATORS!!!!!!
;GLUGLG: Flask drinking sound
;PHASER: Ring attack sound
;MSQUEK:
;MSQUEQ:
;WHOOP:
;SQUEAK: Subroutine to produce a short tone sweep of
;         rising pitch.
;         Usable for small animal squeaks or dripping water
;* * * * *

```

C80A	SNBAS1	EDU	\$	;sound base address 1
C80A CEC823	GLUGLG	LDU	#MSQUEQ	;address of sound subroutine
C80D 8604		LDA	#4	;iteration count
C80F 2005		BRA	PHAS2	,
C811 CEC81F	PHASER	LDU	#MSQUEK	,
C814 860A		LDA	#10	;iteration count
C816 975F	PHAS2	STA	SNSAVA	;scratch iteration counter
C818 ADC4	PHAS1	JSR	U	call indicated subroutine
C81A 0A5F		DEC	SNSAVA	;iteration counter
C81C 26FA		BNE	PHAS1	,loop until done
C81E 39		RTS		
C81F 8E0040	MSQUEK	LDX	#\$0040	,
C822 10		FCB	\$10	;skip next 3 bytes (LDY)
C823 8E0080	MSQUEQ	LDX	#\$0080	,
C826 10		FCB	\$10	;skip next 3 bytes (LDY)
C827 8E0100	WHOOP	LDX	#\$0100	;pitch and duration
C82A 10		FCB	\$10	,skip next 3 bytes (LDY)
C82B 8E0020	SQUEAK	LDX	#\$0020	,
C82E 8D05	SNSQK1	BSR	SNSQK2	;pitch and duration
C830 301F		LEAX	-1,X	,
C832 26FA		BNE	SNSQK1	;increase frequency
C834 39		RTS		,continue
C835 86FF				
C837 8D30	SNSQK2	LDA	#\$FF	
C839 4F		BSR	SNSUB2	,
C83A 202D		CLRA		,
		BRA	SNSUB2	,
C83C 8E0500	REOOP	LDX	#\$0500	,
C83F 8D44	REOOP1	BSR	SNSQK2	;decrease frequency
C841 X0RR70		LFAX	48,X	

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 90

CB44 BC0800	CMPX	#\$0800	;
CB47 25F6	BLD	BE00P1	;
CB49 39	RTS		;
			;

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 91

; \* \* \* \* \*

RATTLE: Subroutine to generate a short train of noise pulses. Each pulse is separated from the next by a period of silence.

Usable for rattlesnake like sounds

; \* \* \* \* \*

C84A 8602	PSSST	LDA	#2	
C84C 8C		FCB	SKIP2	
C84D 8601	PSSHT	LDA	#1	rattle count +1
C84F 8C		FCB	SKIP2	[ sets CCR ]
C850 860A	RATTLE	LDA	#10	rattle count + 1
C852 9762		STA	SNDLAY	use as scratch counter in this routine
C854 108E00C0	SNRAT2	LDY	#\$00C0	noise duration
C858 8D74	SNRAT1	BSR	SNOISE	get a random noise value
C85A 8D69		BSR	SNOUT	
C85C 313F		LEAY	-1,Y	
C85E 26F8		BNE	SNRAT1	
C860 8D58		BSR	SNWT1K	;wait! X-Reg. = \$1000
C862 0A62		DEC	SNDLAY	;decrement counter
C864 26EE		BNE	SNRAT2	
C866 39		RTS		;end of this sound

Local subroutine for code compression

C867 8D65	SNSUB3	BSR	SNOISE	;get some noise
C869 8D5A	SNSUB2	BSR	SNOUT	;out to the DAC
C86B 2050		BRA	SNWAIT	;wait around for a while

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 92

```

* * * * * * * * *
KABOOM: Double explosion sounds that sounds like "Kaboom".
BANG
THUD

* * * * * * * * *

BANG    LDU    #BANGD      ;bang data
        BRA    BOOMER     ;common code

THUD    LDU    #THUDD      ;thud data
        BRA    BOOMER     ;common code

; BDLBBL  LDA    #8          ;whoop count
        STA    SNGAVA     ;make some noise
BDLB10   BSR    SNOISE     ;16-bit random (0-127)
        CLRA
        LSRA
        BNE    BDLB12     ;insure non-zero!
        INCB
        TFR    D,X         ;copy random
        BSR    SNSQK1     ;squeak!
        DEC    SNSAVA     ;decrement and loop
        BNE    BDLB10     ;fall into KABOOM when thru

; KABOOM  LDU    #THUDD      ;point to data table
        BSR    BOOMER     ;wait: X-Reg. = $1000
        BSR    SNWT1K     ;advance to next set of data
        LEAU   4,U

; BOOMER  LDX    0,U         ;set pitch
BOOM1   LDY    2,U
BOOM2   BSR    SNSUB3
        LEAY   -1,Y
        BNE    BOOM2
        LEAX   2,X
        CMPX   #$0150
        BNE    BOOM1

RTS

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 93

\* \* \* \* \*

CHUCK:  
Sound of torch lighting

WHOOSH: Subroutine to simulate the sound of something passing  
rapidly by. This sound is created by passing white noise  
through an envelope generator with a short attack time  
followed by a slower decay time.

Useful for the sound of an arrow in flight or the swing of  
a sword, axe, mace, or other such device.

\* \* \* \* \*

C8A6	BDC931	WHOOSH	JSR	SETNVA	;env. gen. setup routine
C8A9	80		FCB	\$80	;increment
C8AA	BD76	SNWSH2	BSR	SNZNVA	;get some noise
C8AC	2504		BCS	CHUCK	;goto decay if attack done
C8AE	BD15		BSR	SNOUT	;out to DAC
C8B0	20F8		BRA	SNWSH2	;continue
C8B2	BDC92E	CHUCK	JSR	SETNVD	;envelope generator setup
C8B5	A0		FCB	\$A0	;decrement
C8B6	BD6E	SNWSH1	BSR	SNENVN	;compression subroutine
C8B8	20FC		BRA	SNWSH1	;continue

;

```
=====
| SOUND GENERATOR SUPPORT SUBROUTINES
=====
```

```
* * * * *
```

```
; Short period programmable timer subroutine
```

C8BA BE1000	SNWT1K	LDX	#\$1000	;default period entry point
C8BD 3410	SNWAIT	PSHS	X	
C8BF 301F	SNWATI	LEAX	-1,X	;decrement counter
C8C1 26FC		BNE	SNWATI	;done?
C8C3 3590	SNCLK4	PULS	X,PC	

```
* * * * *
```

```
; Sounds - output routine
```

C8C5 D661	SNOUT	LDB	SNVOL	;load desired volume level
C8C7 3D		MUL		
C8C8 84FC		ANDA	#\$FC	
C8CA B7FF20		STA	\$FFF20	;output to DAC
C8CD 39		RTS		

```
* * * * *
```

```
; Noise generator
```

C8CE DC56	SNOISE	LDD	SNDRND	
C8D0 58		LSLB		;multiply by five
C8D1 49		ROLA		
C8D2 58		LSLB		
C8D3 49		ROLA		
C8D4 D356		ADDD	SNDRND	
C8D6 5C		INCB		
C8D7 DD56		STD	SNDRND	;save new value
C8D9 39		RTS		

```
;
```

\* \* \* \* \*

CLANK:  
KLANK:

Subroutines to generate the sounds of objects striking other objects. This sound is created by playing two tones of a slightly de-tuned 6th harmonic relationship.

\* Useful for impacts with metal objects, falling objects, etc.

\* \* \* \* \*

C8DA 8D39	CLANG	BSR CSEUP	
C8DC 64		FCB \$64	
C8DD 24		FCB \$24	
C8DE 8D35	KKLANK	BSR CSEUP	"KKLank" setup
C8E0 32		FCB \$32	
C8E1 12		FCB \$12	
C8E2 8D31	KLANK	BSR CSEUP	"Klank" setup routine
C8E4 AF		FCB \$AF	;frequency 1
C8E5 36		FCB \$36	;frequency 2
C8E6 8D2D	CLANK	BSR CSEUP	"clank" setup routine
C8E8 19		FCB \$19	;frequency 1
C8E9 09		FCB \$09	;frequency 2
C8EA 8D42	SNCLK5	SETNVD	common env. gen. set-up
C8EC 60		FCB \$60	decrement constant
C8ED 9E63		LDX FREQ1	load frequency 1
C8EF 109E65		LDY FREQ2	load frequency 2
C8F2 4F		CLRA	clear sound sample value
C8F3 301F	SNCLK2	LEAX -1,X	
C8F5 2606		BNE SNCLK1	
C8F7 9E63		LDX FREQ1	load frequency 1
C8F9 887F		EORA #\$7F	flip low order 7 bits of sound sample
C8FB 800D		BSR SNCLK3	call local subroutine
C8FD 313F	SNCLK1	LEAY -1,Y	
C8FF 26F2		BNE SNCLK2	
C901 109E65		LDY FREQ2	load frequency 2
C904 8880		EORA #\$80	flip hi order bit
C906 8D02		BSR SNCLK3	
C908 20E9		BRA SNCLK2	
			Local Subroutine
C90A 9759	SNCLK3	STA SNSAU	save sound sample
C90C 8D70		BSR SNENV	
C90E 23B3		BLS SNCLK4	
C910 8DB3		BSR SNOUT	
C912 9659		LDA SNSAV	restore sound sample
C914 39		RTS	

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 96

```

; Local subroutine
C915 AEE1    CSETUP   LDX      ,S++      ;load the stacked PC
C917 E680    LDB      ,X+      ;load frequency 1
C919 4F      CLRA     ;16-bit value
C91A DD63    STD      FREQ1    ;save it
C91C E680    LDB      ,X+      ;load frequency 2
C91E DD65    STD      FREQ2    ;save it
C920 20CB    ; BRA     SNCLK5    ;force a branch

; Compression subroutine
C922 8DA4    SNZNVA   BSR      SNOISE
C924 2067    BRA     SNENVVA  ;
; Compression subroutine
C926 8DA6    SNENVN   BSR      SNOISE
C928 8D54    SNENVVT  BSR      SNENV
C92A 2397    BLS      SNCLK4
C92C 2097    BRA      SNOUT    ;
; Local subroutine
C92E 9E03    SETNVD   LDX      NEGONE  ;decay mode setup
C930 10      FCB      $10      ;skip next 3 bytes (LDY)
C931 9E00    SETNVA   LDX      BIGZER  ;attack mode setup
C933 9F5B    STX      SNENVVS ;store starting volume
C935 AEE4    LDX      ,S      ;get the PC
C937 E680    LDB      ,X+      ;load the increment/decrement
C939 4F      CLRA     ;16-bit value
C93A DD5D    STD      SNENVD  ;store it
C93C AFE4    STX      ,S      ;pointer becomes new PC
C93E 39      SNLXT1   RTS      ;return to caller
;
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 97

\* \* \* \* \*

KLINK: Subroutine to simulate the metal/metal impact sound of swordplay. This sound is generated by mixing a high pitched tone with white noise. The resultant signal is passed through an envelope generator with an instantaneous rise time, and a short decay time.

Useful for swordplay, Knife combat sounds, or the sound of light metal objects striking one another such as the opening of a door lock.

\* \* \* \* \*

C93F 8D8D	KLINK	BSR	SETNVD	common env. gen. set-up
C941 60		FCB	\$60	decrement constant
C942 BDC8CE	SNKLK1	JSR	SNOISE	get some noise
C945 44		LSRA		(some noise)
C946 8DE0		BSR	SNENVT	compression subroutine
C948 BDC8CE		JSR	SNOISE	get some noise
C94B 8A80		ORA	#\$80	add tone component to sample
C94D 8D99		BSR	SNENVT	compression subroutine
C94F 20F1		BRA	SNKLK1	continue

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 98

; \* \* \* \* \*

; Snarl/Growl Sound

; \* \* \* \* \*

C951 BE0300	GROWL	LDX	\$#0300	; parameter
C954 10		FCB	\$10	; skip next 3 bytes (LDY)
C955 BE0200	GRAWL	LDX	\$#0200	; parameter
C958 10		FCB	\$10	; skip next three bytes (LDY)
C959 BE0100	SNARL	LDX	\$#0100	; parameter
C95C 9F5D	SNARL1	STX	SNENV0	;
C95E 4F		CLRA		;
C95F 5F		CLRB		;
C960 DD5B		STI	SNENV5	;
C962 8DBE	SNGRL1	BSR	SNZNVA	;
C964 250B		BCS	SNGRLE2	;
C966 RDC8C5		JSR	SNOUT	;
C969 BE00F0		LDX	\$#00F0	;
C96C RDC8FD		JSR	SNWAIT	;
C96F 20F1		BRA	SNGRL1	;
C971 8DBB	SNGRL2	BSR	SETNVD	; env. gen. setup
C973 40		FCB	\$40	; decay rate
C974 8DB0	SNGRL3	BSR	SNENVN	; compression subroutine
C976 BE0060		LDX	\$#0060	;
C979 BDC8FD		JSR	SNWAIT	;
C97C 20F6		BRA	SNGRL3	;

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SOUNDS - Sound Generator

PAGE 99

\* \* \* \* \*

; Decay envelope generator.  
; BLS after return tests if done

C97E 3402	SNENV	PSHS	A	; save sound sample
C980 DC5B		LDD	SNENV\$	
C982 935D		SUBD	SNENV\$	
C984 3401	SNENV1	PSHS	CC	
C986 DD5B		STD	SNENV\$	; new volume level
C988 E661		LDB	1,S	; restore sound sample
C98A 3D		MUL		; times volume level
C98B 3585		PULS	CC,B,PC	; restore condition codes, etc. return

\* \* \* \* \*

; Attack Envelope Generator  
; BCS after return tests if done

C98D 3402	SNENVA	PSHS	A	
C98F DC5B		LDD	SNENV\$	
C991 D35D		ADDI	SNENV\$	
C993 20EF		BRA	SNENV1	

;

## SWITAB: Define SWI Offsets

C384  
C995

FOO SWITAB SET EQU \$ ;offset initialization

;define offset table

C995 00	FOO	FCB SET	(SETFAX-FOO) AND \$FF	;set amount of fade based on RANGE
C384			SETFAX	;
C996 1E	FOO	FCB SET	(VCTLSX-FOO) AND \$FF	;vector list processor
C3A2			VCTLSX	
C997 A6	FOO	FCB SET	(TXTSTI-FOO) AND \$FF	;string output (string follows)
C448			TXTSTI	
C998 0C	FOO	FCB SET	(TXTSTR-FOO) AND \$FF	;string output
C454			TXTSTR	
C999 05	FOO	FCB SET	(TXTCHR-FOO) AND \$FF	;character output
C459			TXTCHR	
C99A 16	FOO	FCB SET	(EXPANX-FOO) AND \$FF	;string decompression routine
C46F			EXPANX	
C99B 03	FOO	FCB SET	(EXPA00-FOO) AND \$FF	; secondary entry point
C472			EXPA00	
C99C 5D	FOO	FCB SET	(RANDOX-FOO) AND \$FF	;random number generator
C4CF			RANDOX	
C99D 24	FOO	FCB SET	(ZFLIPX-FOO) AND \$FF	;current screen clear
C4F3			ZFLIPX	
C99E 03	FOO	FCB SET	(ZFLOPX-FOO) AND \$FF	;alternate screen clear
C4F6			ZFLOPX	
C99F 09	FOO	FCB SET	(CLRSTX-FOO) AND \$FF	;status line clear
C4FF			CLRSTX	
C9A0 08	FOO	FCB SET	(CLRPXR-FOO) AND \$FF	;primary text area clear
C507			CLRPXR	
C9A1 22	FOO	FCB SET	(HUPDAX-FOO) AND \$FF	;update the heartrate
C529			HUPDAX	
C9A2 B0	FOO	FCB SET	(STATUX-FOO) AND \$FF	;update the status line
C5D9			STATUX	
POAT 70	FOO	FCB	(PUPTRX-FOO) AND \$FF	;update the main screen

```
C656      FOO      SET      PUPIDAX
C9A4 1E    FOO      SET      (PROMPX-FOO) AND $FF      ;prompt the human for input
C674      FOO      SET      PROMPX
C9A5 0B    FOO      SET      (WAITX-FOO) AND $FF      ;wait for a number of jiffies
C67F      FOO      SET      WAITX
C9A6 07    FOO      SET      (ZEROX-FOO) AND $FF      ;clear RAM
C686      FOO      SET      ZEROX
C9A7 02    FOO      SET      (NEGRAX-FOO) AND $FF      ;set RAM to negative one
C688      FOO      SET      NEGRAX
C9A8 1C    FOO      SET      (WIZIX-FOO) AND $FF      ;fade-in
C6A4      FOO      SET      WIZIX
C9A9 04    FOO      SET      (WIZIX0-FOO) AND $FF      ;secondary entry point
C6A8      FOO      SET      WIZIX0
C9AA 1D    FOO      SET      (WIZOX-FOO) AND $FF      ;fade-out
C6C5      FOO      SET      WIZOX
C9AB 21    FOO      SET      (PREPAX-FOO) AND $FF      ;"PREPARE!" msg on EXAMINE screen
C6E6      FOO      SET      PREPAX
C9AC 15    FOO      SET      (OBIRTX-FOO) AND $FF      ;object creation routine
C6FB      FOO      SET      OBIRTX
C9AD 24    FOO      SET      (OCBFIX-FOO) AND $FF      ;OCB filling routine
C71F      FOO      SET      OCBFIX
C9AE 24    FOO      SET      (INIVUX-FOO) AND $FF      ;create initial view
C743      FOO      SET      INIVUX
C9AF 16    FOO      SET      (NEWLUX-FOO) AND $FF      ;create a new level
C759      FOO      SET      NEWLUX
C9B0 6F    FOO      SET      (SOUNDI-FOO) AND $FF      ;secondary entry point
C7C8      FOO      SET      SOUNDI
C9B1 08    FOO      SET      (SOUNDX-FOO) AND $FF      ;sound routines
C7D0      FOO      SET      SOUNDX
                                END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.09

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMTXT - Text Display Service Routines

PAGE 2

INCLUDE B:\COMTXT.ASM

EXT DISPLAY SERVICE ROUTINES

These routines maintain a scrolling text display in Graphic Resolution-6 (G6R) mode. Included are character/string output routines and a software character generator.

Depending on a flag (P,TXSGL), the routines will generate text information in a single or both display planes.

NAM COMTXT  
LIBRY CD  
XDEF TXTXXX, TXTSCR, TXTDPB  
XDEF LSLD05, LSLD4, LSLD03, LSLD02, LSLD01  
XREF SWCTAB, SPCTAB  
PSCT

## TXTXXX: Character Handling Routines

C9B2 8124 TXTXXX CMPA #I.BS  
 C9B4 2709 BEQ TXTBS  
 C9B6 811F CMPA #I.CR  
 C9B8 2710 BEQ TXTCR

## Printable Characters

C9BA BD5B BSR TXTDPB ;deposit byte into display  
C9BC 3001 LEAX 1,X ;advance cursor  
C9BE 39 RTS ;bye

Backspace

```

C9BF 301F    TXTBS   LEAX    -1,X      ; cursor back one position
C9C1 9C03    CMPX    NEGONE   ; wraparound?
C9C3 2604    BNE     TXTBS1  ; hope
C9C5 AE42    LDX     P,TXCNT,U ; wraparound to end
C9C7 301F    LEAX    -1,X      ;
C9C9 39      TXTBS1  RTS     ; bye

```

## Carriage Return

```

C9CA 308820    TXTCR   LEAX    32,X      down one line
C9CD 1E01       EXG     D,X      perform logical operation on IX
C9CF C4E0       ANDB    #X11100000 back to beginning of line
C9D1 1E01       EXG     D,X      restore regs
C9D3 39         RTS

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMTXT - Text Display Service Routines

PAGE 3

```

; TXTSCR: Screen Scrolling Routine
;
; Inputs:
;   U - TXB pointer
;
; Returns:
;   X - new cursor position (beginning of last line)
;       - screen scrolled up one line

C9D4 3436    TXTSCR  PSHS    A,B,X,Y      ; save regs
C9D6 AEC4      LDX     P,TXBAS,U      ; pointer to first line
C9D8 EC42      LDD     P,TXCONT,U    ; form byte count
C9DA 830020    SUBD    #32          ; minus one line
C9DD ED62      STD     2,S          ; return new cursor position
C9DF 8D2F      BSR     LSLD3        ; byte count = chars * 8
C9E1 1F02      TFR     D,Y          ; copy byte count

C9E3 EC890100  TSCR10   LDD     32*8,X      ; copy two bytes from next line
C9E7 6D47      TST     P,TXSQL,U    ; processing single/double screen?
C9E9 2604      BNE     TSCR12        ; single
C9EB ED891800  STD     G6,LEN,X    ; deposit two bytes into second screen
C9EF ED81      TSCR12   STD     ,X++        ; deposit two bytes into first screen
C9F1 313E      LEAY    -2,Y         ; decrement loop count
C9F3 26EE      BNE     TSCR10        ; loop

C9F5 E646      ;           LDB     P,TXINV,U    ; process inverse-mode
C9F7 1D        SEX     ,            ; (magic cuz' P,TXINV is 0 or -1)
C9F8 108E0100  LDY     #32*8        ; 'clear last line' counter

C9FC 6D47      TSCR20   TST     P,TXSQL,U    ; processing single/double screen?
C9FE 2604      BNE     TSCR22        ; single
CA00 ED891800  STD     G6,LEN,X    ; deposit two bytes into second screen
CA04 ED81      TSCR22   STD     ,X++        ; deposit two bytes into first screen
CA06 313E      LEAY    -2,Y         ; decrement loop count
CA08 26F2      BNE     TSCR20        ; loop

CA0A 35B6      ;           PULS   A,B,X,Y,PC  ; restore regs/exit

; LSLDn: 16-bit Logical Shift Left
;
; CA0C 58      LSLD5   LSLB      MSB into carry
CA0D 49      ROLA      ; carry into LSB
CA0E 58      LSLD4   LSLB      ; and so on...
CA0F 49      ROLA      ;
CA10 58      LSLD3   LSLB      ;
CA11 49      ROLA      ;
CA12 58      LSLD2   LSLB      ;
CA13 49      ROLA      ;
CA14 58      LSLD    LSLB      ;
CA15 49      ROLA      ;
CA16 39      RTS      ; go home

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMTXT - Text Display Service Routines

PAGE 4

```

        ; TXTDPB: Software Character Generator
        ; Inputs:
        ;   A - Display Code Character
        ;   U - Pointer to TXB

CA17 3476    TXTDPB  PSHS    A,B,X,Y,U      ;save regs
        ; Select Proper Character Set
CA19 8120    CMPA    #$20      ;normal character?
CA1B 250C    BLO     NORDPB    ; yes

        ; Special Characters
CA1D 8020    SUBA    #$20      ;offset = (char value - $20) * 7
CA1F C607    LDB     #7
CA21 30      MUL
CA22 C3DBB6    ADDD    #SPCTAB    ;special character table
CA25 1F01    TFR     D,X      ;software character addr
CA27 201B    BRA     DPB00      ;drop into common code

        ; Normal Characters
CA29 C605    NORDPB  LDB     #5      ;offset = char value * 5
CA2B 3D      MUL
CA2C C3DBB1B  ADDD    #SWCTAB    ;software char table base addr
CA2F 1F01    TFR     D,X      ;5-bit string addr
CA31 CE0357    LDU     #SWCHAR+1  ;s/w char expansion buffer
CA34 3F      SWI
CA35 06      FCB     EXPANO    ;expand the string

        ; Center 5-bit data in each byte
CA36 BE035E    LDX     #SWCHAR+8  ;starting point
CA39 6882    NDPB10  LSL     ,-X    ;shift previous char
CA3B 6884    LSL     X
CA3D 8C0357    CMPX    #SWCHAR+1  ;done?
CA40 22F7    BHI     NDPB10    ;nope
CA42 EE66    LDU     6,S      ;restore TXB ptr
;
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMTXT - Text Display Service Routines

PAGE 5

Calculate Display Screen Address

Note: P,TXCUR,U contains the following bit pattern

LLLL LLLL LLLC CCCC

Where: L - forms the LINE count  
C - forms the COLUMN count

display offset = (LINE\*32\*8) + COLUMN

CA44 EC44	DPB00	LDD	P,TXCUR,U	;load cursor position
CA46 BDC8		ESR	LSL#3	;magic bit manipulation!!!
CA48 54		LSRB		;
CA49 54		LSRB		;
CA4A 54		LSRB		;
CA4B E3C4		ADD#	P,TXBAS,U	;add base addr to get pointer
CA4D 1F02		TFR	D,Y	;transfer to index register

At this point:

X - ptr to software character (7-bytes)  
Y - display screen addr  
U - ptr to TXB

CA4F C607	DPB10	LDB	#7	;loop count
CA51 A680		LDA	,X+	;load next byte
CA53 AB46		EORA	P,TXINV,U	;process inverse mode
CA55 A7A4		STA	,Y	;deposit into display
CA57 6047		TST	P,TXSGL,U	;write to a single screen?
CA59 2604		BNE	DPB20	;yes
CA5B A7A91800		STA	G6,LEN,Y	;write into both displays
CA5F 31A820	DPB20	LEAY	32,Y	;advance to next scan line
CA62 5A		DEC#		;decrement loop count
CA63 26EC		BNE	DPB10	;loop until done
CA65 35F6		PULS	A,B,X,Y,U,PC	;restore regs/exit
			END	

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VECTOR - Vector Generator

PAGE 6

```
INCLUD B:VECTOR.ASM
;-----;
; V E C T O R   G E N E R A T O R
;-----;
NAM VECTOR
LIBR CD
XDEF VECTOR,NEGD,DIVIDE,BITMSK
XREF ASRD6,ASRD3,LSL05
PSCT
PAGE
```

## DIVIDE: Software Division Routine

Inputs:  
 D - 16-bit unsigned dividend  
 X - divisor

Returns:  
 D - 16-bit unsigned result (RADIX-8: bbbbbbbb.bbbbbbbb)

Destroys:  
 T0,T1,T2

```
CA67 3416 DIVIDE PSHS A,B,X      ; save regs
CA69 6FE4 CLR 0,S                ; reset quotient
CA6B 6F61 CLR 1,S                ; store dividend (T0:T1:T2)
CA6D OFC1 STD  T0
CA6F DDC2 STD  T1
```

## Special Test for Zero/One

```
CA71 2724 BEQ DIV99      ; dividend was zero
CA73 10A362 CMPD 2,S      ; dividend = divisor?
CA76 2604 BNE DIV00      ; nope
CA78 6CE4 INC  0,S      ; set result
CA7A 201B BRA DIV99      ; done
```

## Shift and Subtract

```
CA7C 8E0010 DIV00 LDX #16      ; number of times to loop
CA7F 08C3 DIV10 LSL T2      ; dividend * 2
CA81 09C2 ROL  T1
CA83 09C1 ROL  T0
CA85 6861 LSL 1,S      ; quotient * 2
CA87 69E4 ROL  0,S
```

## Trial Subtraction

```
CA89 DCC1 LDD  T0      ; 16-MSBits of dividend
CA8B A362 SUBD 2,S      ; trial subtraction successful?
CA8D 2504 BCS DIV20      ; nope
CA8F DDC1 STD  T0      ; update 16-MSBits of dividend
CA91 6C61 INC  1,S      ; quotient = quotient + 1
CA93 301F DIV20 LEAX -1,X     ; decrement and loop
CA95 26E8 BNE DIV10      ; until remainder < divisor
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VECTOR - Vector Generator

PAGE 7

; NEGDI: 16-bit Negation Routine

CA99 43	NEGD	COMA	;16-bit complement
CA9A 53		COMB	
CA9B C30001		ADDD #1	;plus one
CA9E 39		RTS	

; INCRE: Calculate Incremental Values

Inputs:

D - 16-bit signed value  
LENGTH - 16-bit divisor

Returns:

D - result of (D/LENGTH)

CA9F 3416	INCRE	PSHS A,B,X	;save regs
CAA1 9E43		LDX LENGTH	;load length into X
CAA3 ECE4		LDD O,S	;check sign
CAA5 2A07		BPL INCR10	; D is already positive

; D is negative

CAA7 8DFO		BSR NEGD	;use absolute value
CAA9 8DBC		BSR DIVIDE	;use software division routine
CAA9 8DEC		BSR NEGD	;make negative again
CAA9 8C		FCB SKIP2	
CAAE 8DB7	INCR10	BSR DIVIDE	
CAE0 EDE4	INCR90	STD O,S	;use software division routine
CAE2 3596		PULS A,B,X,PC	;store return parameter
			;return parm/exit

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VECTOR - Vector Generator

PAGE 8

```

; VECTOR: Simple Digital Differential Analyzer

CAB4 ZECB8A    VECT00 JMP     VECT99      ;local jump
CAB7 3476      VECTOR PSHS   A,B,X,Y,U  ;save regs

; Set Fade Parameters
CAB9 0C2D      INC     VCTFAD    ;invisible line?
CABB 27F7      BEQ     VECT00    ; yes - don't draw anything
CACD 962D      LDA     VCTFAD    ;set initial fade count
CABF 972E      STA     FADECNT ;.

; Calculate Length
CAC1 DC35      LDD     X1        ;dx = (x1-x0)
CAC3 9331      SUBD   X0        ;
CAC5 DD3E      STD     DX+1    ; already absolute value
CAC7 2A02      BPL    VECT12    ;make absolute
CAC9 8DCE      BSR    NEGD    ;assume this will be our length
CAC8 DD43      VECT12 STD     LENGTH

CACD DC33      LDD     Y1        ;dy = (y1-y0)
CACF 932F      SUBD   Y0        ;
CAC1 DD41      STD     DY+1    ; already absolute value
CAC3 2A02      BPL    VECT14    ;make absolute
CAC5 8DCE      BSR    NEGD    ;length = max (abs(dx),abs(dy))
CAC7 109343    VECT14 CMPD   LENGTH
CACD 2D04      BLT    VECT20    ;
CAC8 DD43      STD     LENGTH
CACF 27D4      BEQ    VECT00    ;zero length means skip processing

; Calculate X,Y Increments
CAE0 DC3E      VECT20 LDD     DX+1    ;dx = (x1-x0)/length
CAE2 8DBB      BSR    INCRE   ;
CAE4 DD3E      STD     DX+1    ;extend sign
CAE6 1F89      TFR    A,B     ;sex
CAE8 1D       LDB    #1      ;byte move delta
CAE9 C601      STA    DX      ;
CAEB 973D      BPL    VECT22    ;positive delta
CAED 2A01      NEGB   DELTAX ;negative delta
CAEF 50       VECT22 STB     DELTAX ;store byte move delta

CAF2 DC41      LDD     DY+1    ;dy = (y1-y0)/length
CAF4 8DA9      BSR    INCRE   ;
CAF6 DD41      STD     DY+1    ;extend sign
CAF8 1F89      TFR    A,B     ;sex
CAF9 1D       LDB    #32    ;scan line delta
CAF0 C620      STA    DY      ;
CAF2 9740      BPL    VECT24    ;positive delta
CAF4 2A01      NEGB   DELTAY ;negative delta
CAF6 50       VECT24 STB     DELTAY ;store scan line delta
CAF8 D746

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

BAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VECTOR - Vector Generator

PAGE 9

```

; Set (XX = X0 + 0.5) (YY = Y0 + 0.5)
CB04 DC31      LDD    X0          ; set integer portion of XX,YY
CB06 DD37      STD    XX
CB08 DC2F      LDD    Y0          ;
CB0A DD3A      STD    YY
CB0C B680      LDA    #BIT7     ; set fraction portion of XX,YY
CB0E 9739      STA    XX+2     ;
CB10 973C      STA    YY+2     ;

; Calculate Initial Screen Coordinate
CB12 AE42      LDX    P,VIDEND,U   ; store display ending, base addr
CB14 9F49      STX    VIDEND      ; (frees U-reg)
CB16 AEC4      LDX    P,VDBAS,U   ;
CB18 9F47      STX    VDBAS      ;

CB1A DC3A      LDD    YY          ; load Y-coordinate
CB1C B0CA0C    JSR    LS1D5      ; multiply by 32 to get scan line addr
CB1F 308B      LEAX   I,X         ; add to base addr

CB21 DC37      LDD    XX          ; load X-coordinate
CB23 B0D37F    JSR    ASRD3      ; divide by 8 to get byte addr
CB26 308B      LEAX   I,X         ; we now have exact display byte addr

CB28 CECB8E    LDW    #BITMSK   ; load bit mask
CB2B 109E43    LDY    LENGTH     ; loop counter

; Process Fade Function
CB2E 0A2E      VECT30 DEC    FADCNT   ; should we plot this dot?
CB30 2622      BNE    VECT40     ; nope
CB32 962D      LDA    VCTFAD     ; reset counter
CB34 972E      STA    FADCNT     ;

; Clip all offscreen points
CB36 0D37      TST    XX          ; clip X-coordinate
CB38 261A      BNE    VECT40     ;
CB3A 9C47      CMPX   VIDBAS     ; clip Y-coordinate
CB3C 2516      BLO    VECT40     ;
CB3E 9C49      CMPX   VIDEND     ;
CB40 2412      BHS    VECT40     ;

; Plot Next Point
CB42 D638      LDD    XX+1      ; load low order byte
CB44 C407      ANDB   #7         ; MOD 8 to get bit position
CB46 A6C5      LDA    B,U        ; load positioned bit
CB48 0D2C      TST    VBGINV    ; normal or inverse mode?
CB4A 2704      BEQ    VECT32    ; normal mode
CB4C 43       COMA   ,X         ; inverse mask
CB4D A484      ANDA   ,X         ; clear bit
CB4F 8C       FCB    SKIP2     ;
CB50 AA84      VECT32 ORA    ,X         ; mark bit
CB52 A784      STA    VDLYHYTE  ; update display byte

```

```

; Increment X,Y

CB54 9638    VECT40 LDA    XX+1      ;modify and save middle byte
CB56 84F8    ANDA   #Z11111000   ;
CB58 97C1    STA    TO          ;
;                                         ;
CB5A DC38    LDD    XX+1      ;increment middle, low bytes
CB5C D33E    ADDD   DX+1      ;
CB5E DD38    STD    XX+1      ;
CB60 D637    LDB    XX          ;
CB62 D93D    ADCB   DX          ;
CB64 D737    STB    XX          ;
;                                         ;
CB66 84F8    ANDA   #Z11111000   ;time for byte move?
CB68 91C1    CMPA   TO          ;
CB6A 2704    BEQ    VECT50     ;nope
CB6C D645    LDR    DELTAX    ;move byte
CB6E 3085    LEAX   B,X        ;
;                                         ;
CB70 DC38    VECT50 LDD    YY+1      ;increment middle, low bytes
CB72 97C1    STA    TO          ;save middle byte for later
CB74 D341    ADDD   YY+1      ;
CB76 DD38    STD    YY+1      ;update middle, low bytes
CB78 D63A    LDB    YY          ;
CB7A D940    ADCB   DY          ;
CB7C D73A    STB    YY          ;
;                                         ;
CB7E 91C1    CMPA   TO          ;time for scan line change?
CB80 2704    BEQ    VECT60     ;nope
CB82 D646    LDR    DELTAY    ;change scan lines
CB84 3085    LEAX   B,X        ;
;                                         ;

; Decrement Loop Counter

CB86 313F    VECT60 LEAY  -1,Y      ;decrement and loop
CB88 26A4    BNE    VECT30     ;
;                                         ;
CB8A 0A2D    VECT99 DEC    VCTFAD    ;restore VCTFAD to original value
CB8C 35F6    PULS   A,B,X,Y,U,PC  ;restore regs/exit
;                                         ;
CB8E 80      BITMSK FCB   BIT7
CB8F 40      FCB   BIT6
CB90 20      FCB   BITS
CB91 10      FCB   BIT4
CB92 08      FCB   BIT3
CB93 04      FCB   BIT2
CB94 02      FCB   BIT1
CB95 01      FCB   BIT0
;                                         ;
END

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PARSER - Line Parsing Routines

PAGE 11

INCLUD B:PARSER.ASM

;!!!!!! P A R S I N G R O U T I N E S !!!!!!!

GETTOK: Return the next token in LINBUF

Inputs:

LINPTR - where to start search

Returns:

LINPTR - pointing past token

TOKEN - token terminated by a (\$FF)

N - condition code set if no token found

CB96 3452  
CB98 9E11  
CB9A CE0313GETTOK PSHS A,X,U ;save some regs  
LOX LINPTR ;load line buffer pointer  
LDU #TOKEN ;load token buffer base addr

; Eat leading spaces

CB9D A680  
CB9F 27FC  
CBA1 2002GTOK10 LDA ,X+ ;get next input char  
BEQ GTOK10 ;eat leading spaces  
BRA GTOK22 ;drop into loop

; Copy non-spaces to TOKEN

CBA3 A680  
CBA5 2F08  
CBA7 A7C0  
CBA9 11830333  
CBAD 25F4GTOK20 LDA ,X+ ;get input string char  
GTOK22 BLE GTOK30 ;(\$FF) or space means we are done  
STA ,U+ ;otherwise copy char to token buffer  
CMPU #TOKEND ;insure token buffer is not overrun  
BLO GTOK20 ;

; Terminate token with (\$FF)

CBAF 86FF  
CBB1 A7C0  
CBB3 9F11  
CBB5 2D0313  
CBB8 3502GTOK30 LDA #\$FF ;terminator  
STA ,U+  
STX LINPTR ;update pointer  
TST TOKEN ;return N set if no token  
PULS A,X,U,PC ;restore regs/go home

```

PAROBJ: Parse an object from the line buffer
Inputs:
    LINBUF - line buffer
    LINPTR - line buffer pointer
Returns:
    A - object class
    B - object type
    OBJCLS - object class
    OBJTYP - object type
    LINPTR - updated past object tokens
Format:
    1) <generic>
    2) <adjective> <generic>
Examples:
    1) "SCROLL"
    2) "GUIDANCE SCROLL"

```

CBEA 0F90	PAROBJ	CLR	SPEFLG	;assume generic form
CBEF 8ED96A		LDX	#GENTAB	;generic token search
CBBF 8D2B		BSR	PARSER	
CBC1 2B05		BMI	PDBJ10	;search failure
CBC3 271A		BEQ	PARERR	;null tokens are illegal
CBC5 DB8E		STD	OBJTYP	;store parms
CBC7 39		RTS		;go home
; Check for an adjective				
CBC8 0A90	PDBJ10	DEC	SPEFLG	;must be specific form
CBCA 8ED8F3		LDX	#ADJTAB	;search for an adjective
CBCD 8D18		BSR	PARSEO	
CBCF 2F0E		BLE	PARERR	;null/illegal tokens not allowed
CBD1 DD8E		STD	OBJTYP	;save adjective parms
CBD3 8ED96A		LDX	#GENTAB	;search for generic tokens
CBD6 8D14		BSR	PARSER	
CBD8 2F05		BLE	PARERR	;null/illegal tokens
CBD9 D18F		CMPB	OBJCLS	;matching class?
CBD0 2601		BNE	PARERR	;nope - loser
CBOE 39	PDBJ99	RTS		;return as a winner!

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PARSER - Line Parsing Routines

PAGE 13

```
; PARERR: Tell user the bad news
CBDF 3262    PARERR  LEAS    2,S          ;remove return addr
CRE1 3F        CMDERR  SWI
CRE2 02        FCB     OUTSTI
CRE3 17        FCB     Z00010111      ;???
CRE4 7B        FCB     Z01111011      ;???
CRE5 D0        FCB     Z11010000      ;???
CBE6 39        ;           RTS
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PARSER - Line Parsing Routines

PAGE 14

; PARSER: Get and Classify the next TOKEN

Inputs:

X - pointer to token table  
LINPTR - Line Buffer Pointer

Returns:

A - Token type  
B - Token class  
TOKEN - Token  
LINPTR - updated past last token  
N - condition code set if search failure

CBE7 3476	PARSE0	PSHS	A,B,X,Y,U	;secondary entry point
CBE9 4F		CLRA		
CBEA 2008		BRA	PARS02	; ;
CBEC 3476	PARSER	PSHS	A,B,X,Y,U	; save some regs
CBEE 4F		CLRA		; assume next token is null
CBEF 5F		CLB		
CBF0 BDA4		BSR	GETTOK	; get 'the next token'
CBF2 2B39		BMI	PARS92	; token is null
CBF4 0F78	PARS02	CLR	PARFLG	; reset "seen match" flag
CBF6 0F7B		CLR	FULFLG	; reset "full word" match flag
CBF8 E680		LDB	,X+	; load the number of tokens
CBFA D779		STB	PARCNT	; store as loop count

; Compare TOKEN against next command

CBFF CE0313	PARS10	LDU	*TOKEN	; reset token byte pointer
CBFF 3F		SWI		; expand 5-bit string
CC00 05		FCB	EXPAND	
CC01 108E0336		LDY	*STRING+2	; load command byte pointer
CC05 E6C0	PARS12	LDB	,U+	; load next token byte
CC07 2B0E		BMI	PARS20	; winner - end of token
CC09 E1A0		CMPB	,Y+	; token match command?
CC0B 2615		BNE	PARS30	; nope - loser
CC0D 6DA4		TST	,Y	; reach end of command?
CC0F 2AF4		BPL	PARS12	; nope - continue comparisons
CC11 6DC4		TST	,U	; end of token?
CC13 2A0D		BPL	PARS30	; nope - loser
CC15 0A7B		DEC	FULFLG	; end token/command - full match

; We have a Winner!

CC17 0D78	PARS20	TST	PARFLG	; have we already seen a match?
CC19 2610		BNE	PARS90	; can't have two matches!
CC1B 0C78		INC	PARFLG	; flag successful command match
CC1D F60335		LDB	STRING+1	; load token class
CC20 EDE4		STD	0,S	; store return values

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PARSER - Line Parsing Routines

PAGE 15

```
; Advance to next command
CC22 4C    PARS30 INCA      ;bump command counter
CC23 0A79    DEC    PARCNT   ;decrement loop counter
CC25 2605    BNE    PARS10   ;loop until table exhausted

; Finish up and go home
CC27 0D78    TST    PARFLG   ;did we match a command?
CC29 2604    BNE    PARS99   ;yes - go home a winner
CC2B DC03    PARS90 LDD    NEGONE  ;loser...
CC2D EDE4    PARS92 STD    O,S
CC2F 35F6    PARS99 PULS   A,B,X,Y,U,FC ;restore regs/go home
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PARSER - Line Parsing Routines

PAGE 16

PARHND: Parse LEFT or RIGHT hand

Inputs:

LINPTR

Returns:

A - Token Type

B - Token Class

U - #PLHAND or #PRHAND

TOKEN = token

LINPTR = updated past last token

N = condition code set if parse fails

CC31	8ED8D9	PARHND	LDX	#DIRTAB	;directions
CC34	8DB6		BSR	PARSER	;get the next token
CC36	2FA7		BLE	PARERR	;null/illegal token
CC38	CE021F		LDU	#PRHAND	;right hand
CC3B	8101		CMPA	#T,RT	;
CC3D	2707		BEQ	PHND90	;
CC3F	CE021D		LDU	#PLHAND	;left hand
CC42	8100		CMPA	#T,LT	;
CC44	2699		BNE	PARERR	;
CC46	AEC4	PHND90	LDX	,U	;load OCB ptr
CC48	39		RTS		;exit
	;				
			END		

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DGNGEN - Dungeon Maze Generator

PAGE 17

INCLUDE B:DGNGEN.ASM

!!!!!!  
D U N G E O N   G E N E R A T O R  
!!!!!!

DGNGEN: Subroutine to generate the dungeon.

Resulting dungeon is created in a row/column matrix  
based at DGNMAP.

Architecture and contents of the dungeon are based  
on predetermined "seeds" passed to the random number  
generator in COMMON.TXT

Dungeon matrix format is as follows:

B B B B B B B B

-----

| | | | +-----

North Features

| | | +-----

East Features

| | +-----

South Features

| +-----

West Features

Bit Patterns denote:

00 = Passage  
01 = Regular Door  
10 = Secret Door  
11 = Wall

0003      N.WALL    EQU    Z00000011  
000C      E.WALL    EQU    Z00001100  
0030      S.WALL    EQU    Z00110000  
00C0      W.WALL    EQU    Z11000000

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DNGEN - Dungeon Maze Generator

PAGE 19

RNDCEL: Pick A Random Cell in DGNMAP

Returns:

A - Row  
B - ColumnX - DGNMAP addr  
Calls:  
MAP32

CC71 3F	RNDCEL	SWT	RANDOM	; a pair of random numbers
CC72 07		FCB	#31	;MOD 32
CC73 841F		ANDA		
CC75 1F89		TFR	A,B	
CC77 3F		SWT		; (then fall into MAP32)
CC78 07		FCB	RANDOM	
CC79 841F		ANDA	#31	;MOD 32

MAP32: Calculate Map Address Based on ROW/COL

Inputs:

A - row  
B - colReturns:  
X - DGNMAP addr

CC7B 3406	MAP32	PSHS	A,B	; save regs
CC7D 841F		ANDA	#31	; insure row/col are MOD 32
CC7F C41F		ANDB	#31	
CC81 1F01		TFR	D,X	; make a copy of row/col
CC83 C620		LDB	#32	; offset = row*32 + col
CC85 30		MUL		
CC86 C305F4		ADD	#MAZLND	; add base addr
CC89 1E01		EXG	D,X	; D<--row/col, X<--DGNMAP+(row*32)
CC8B 30		ABX		; add col to addr
CC8C 3586		PULS	A,B,PC	; restore regs/exit

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DGNGEN - Dungeon Maze Generator

PAGE 18

FRIEND: Copy Neighbors to Storage

Inputs:

A - row

B - column

U - addr of storage

Returns:

0,U 1,U 2,U

3,U 4,U 5,U

6,U 7,U 8,U

&gt; 3x3 matrix centered on ROW,COL

CC49 3456	FRIEND	PSHS A,B,X,U	; save regs
CC4B 4A		DECA	; upper row
CC4C 8D08		BSR FSUB10	
CC4E 4C		INCA	; middle row
CC4F 8D05		BSR FSUB10	
CC51 4C		INCA	; lower row
CC52 8D02		BSR FSUB10	
CC54 3506		PULS A,B,X,U,PC	; restore regs/exit
CC56 3406	FSUB10	PSHS A,B	; save regs
CC58 5A		DEC8	; left column
CC59 8D05		BSR FSUB20	
CC5B 5C		TINC8	; center column
CC5C 8D02		BSR FSUB20	
CC5E 5C		INC8	; right column
CC5F 8C		FCR SKIP2	; drop into FSUB20 (hack!)
CC60 3406	FSUB20	PSHS A,B	; save position
CC62 8D2A		BSR BORDER	; are we out-of-bounds?
CC64 2605		BNE FSUB22	yes
CC66 8D13		BSR MAP32	
CC68 A684		LDA X	; calculate map addr
CC6A 8C		FCB SKIP2	; load cell contents
CC6B 86FF	FSUB22	LDA #\$FF	; solid wall pattern
CC6D A7C0		STA ,U	; store into storage
CC6F 3584		PULS A,B,PC	; restore regs/exit

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DGNGEN - Dungeon Maze Generator

PAGE 20

BORDER: Are we out-of-bounds?

Inputs:

A - row  
B - col

Returns:

Z - Condition code set if position is legal

CC8E 3406	BORDER	PSHS	A,B	---- start of procedure BORDER
CC90 841F		ANDA	#31	MOD 32
CC92 A1E4		CMPA	0,S	still same after MOD?
CC94 2604		BNE	BORD99	nope
CC96 C41F		ANDB	#31	MOD 32
CC98 E161		CMPB	1,S	still same?
CC9A 3586	BORD99	PULS	A,B,PC	---- end of procedure BORDER

```

; DGNGEN: Create the Dungeon Maze
CC9C      DGNGEN EQU    $           ;in the beginning...
; Phase I: Create Maze
CC9C  BE05F4      LDX      #MAZLND      ;all of "maze" to $FF
CC9F  CE09F4      LDU      #MAZEND      ;
CCA2  3F          SWI      '           ;
CCA3  12          FCB      NEGRAM      ;
; Initialize RANDOM number generator
CCA4  8EC0DF      LDX      #LVLTAB      ;point to random seed table
CCA7  D681          LDB      LEVEL       ;load dungeon level #
CCA9  3A          ABX      '           ;point to 3-byte seed
CCAA  EC81          LDD      ,X++        ;first two bytes
CCAC  DD6B          STD      SEED       ;
CCAЕ  A684          LDA      ,X          ;third byte
CCB0  976D          STA      SEED+2      ;
;                                         ;
CCR2  108E01F4      LDY      #500        ;number of cells to create
CCB6  RDCC71      JSR      RNDCEL      ;pick a random starting cell
CCB9  DD7C          STD      DROW       ;
;
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DGNGEN - Dungeon Maze Generator

PAGE 22

```

; Select a Random Direction/Distance
CCR8 3F      DGEN10 SWI      ;random direction
CCBC 07      FCB     RANDOM
CCR9 8403    ANDA    #3      ;MOD 4
CCBF 978A    STA     DIR
CCC1 3E      SWI      ;random distance
CCC2 07      FCB     RANDOM
CCC3 8407    ANDA    #7      ;MOD 8
CCC5 4C      INCA      ;range (1:8)
CCC6 977E    STA     DST
CCC8 2008    BRA     DGEN30

; Perform Distance Check
CCC4 DC88    DGEN20 LDD      ;update position
CCC5 DD7C    STD     DROW
CCC6 0A7E    DEC     DST
CCD0 27E9    BEQ     DGEN10  ;done - select a new dir/dst

; Take a "tentative" step
CCD2 DC7C    DGEN30 LDD      ;parms to STEP
CCD4 B0D11B   JSR     STEP
CCD7 8BB5    BSR     BORDER
CCD9 26E0    BNE     DGEN10  ;out-of-bounds?
                                ;yes!

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DGNGEN - Dungeon Maze Generator

PAGE 23

; Have we been here before?

```
CCDB DD88      STD    R0W      ;store position for later
CCDD 6D84      TST    ,X      ;have we been here before?
CCDF 27E9      BEQ    DGEN20   ; non-virgin/loop
```

; Check for "cleared" corner patterns

```
CCE1 CE09F4      LDU    #NEIBOR  ;addr of "neighbors" storage
CCE4 BDCC49      JSR    FRIEND  ;copy our "friends"
CCE7 A643      LDA    3,U      ;upper left corner
CCE9 ABC4      ADDA   0,U      ;
CCEB AB41      ADDA   1,U      ;
CCED 27C0      BEQ    DGEN10   ; loser
CCEF A641      LDA    1,U      ;upper right corner
CCF1 AB42      ADDA   2,U      ;
CCF3 AB45      ADDA   3,U      ;
CCF5 27C4      BEQ    DGEN10   ; loser
CCF7 A645      LDA    5,U      ;lower right corner
CCF9 AB48      ADDA   6,U      ;
CCFB AB47      ADDA   7,U      ;
CCFD 27C0      BEQ    DGEN10   ; loser
CCFF A647      LDA    7,U      ;lower left corner
CD01 AB46      ADDA   8,U      ;
CD03 AB43      ADDA   9,U      ;
CD05 27B4      BEQ    DGEN10   ; loser
```

; Count new cells and loop

```
CD07 6F84      CLR    ,X      ;mark a new cell
CD09 313F      LEAY   -1,Y    ;decrement cell count
CD0B 26BD      BNE    DGEN20   ;loop until done
```

## ; Phase III: Create Walls

CD0D 0F7C	CLR	BROW	; reset indices	
CD0F 0F7D	CLR	DCOL	; ;	
CD11 DC7C	DGEN50	LDD	BROW	; calculate cell addr
CD13 BDCC7B		JSR	MAP32	
CD16 A684		LDA	,X	; load cell contents
CD18 4C		INCA		; check for completely walled cell
CD19 2726		BEQ	DGEN70	; skip processing

## ; Determine Neighbors

CD1B DC7C	LDD	BROW	; load position
CD1D CE09F4	LDD	#NEIBOR	; where to put neighbors
CD20 BDCC49	JSR	FRIEND	; find our friends
CD23 A684	LDA	,X	; load cell contents
CD25 C6FF	LDB	#\$FF	; solid wall pattern

## ; Create Walls

CD27 E141	CMPB	1,U	; wall to the north?
CD29 2602	BNE	DGEN60	; nope
CD2B 8A03	ORA	#N.WALL	; create northern wall
CD2D E143	DGEN60	CMPB	; wall to the west?
CD2F 2602	BNE	DGEN62	; nope
CD31 8AC0	ORA	#W.WALL	; create western wall
CD33 E145	DGEN62	CMPB	; wall to the east?
CD35 2602	BNE	DGEN64	; nope
CD37 8A0C	ORA	#E.WALL	; create eastern wall
CD39 E147	DGEN64	CMPB	; wall to the south?
CD3B 2602	BNE	DGEN66	; nope
CD3D 8A30	ORA	#S.WALL	; create southern wall
CD3F A784	DGEN66	STA	; store "walled" cell

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DGNGEN - Dungeon Maze Generator

PAGE 25

## ; Bump Indexes and Loop

CD41 C620	DGEN70	LDB	#32	;rollover value
CD43 0C7D		INC	DCOL	;bump column counter
CD45 D17D		CMPB	DCOL	
CD47 26C8		BNE	DGEN50	;loop
CD49 0F7D		CLR	DCOL	;reset column counter
;				
CD4B 0C7C		INC	DROW	;bump row counter
CD4D D17C		CMPB	DROW	
CD4F 26C0		BNE	DGEN50	;loop

## ; Create Doors/Secret Doors

CD51 C646		LDB	#70	;number of regular doors
CD53 CECDAAA		LDD	#DORTAB	
CD55 8D15	DGEN80	BSR	MAKDOR	;put in a door
CD58 5A		DEC8		;decrement count
CD59 26FB		BNE	DGEN80	;
;				
CD5B C62D		LDB	#45	;number of secret doors
CD5D CECDAE		LDD	#SDRTAB	
CD60 8D0B	DGEN82	BSR	MAKDOR	;put in a secret door
CD62 5A		DEC8		;decrement count
CD63 26FB		BNE	DGEN82	;

## ; Spin the random number generator

CD65 B697		LDB	SECOND	;use the jiffy counter
CD67 3F	DGEN90	SWI		;another random number
CD68 07		FCB	RANDOM	
CD69 5A		DEC8		;decrement counter
CD6A 26FB		BNE	DGEN90	;
;				
CD6C 39		RTS		;bye

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DONGEN - Dungeon Maze Generator

PAGE 26

; MAKDOR: Create Doors

Inputs:

A - door pattern  
U - table of doorsCD6D 3476      MAKDOR PSHS A,B,X,Y,U      ;save regs  
CD6F 108ECDA6      LDY #MSKTAB      ;table of bit masks

; Where can we put a door in?

CD73 BDCC71      MDOR10 JSR RNDCEL      ;get a random cell  
CD76 DD88      STD ROW      ;store for later use  
CD78 E684      LDR X      ;can we put a door here?  
CD7A C1FF      CMPB #\$FF  
CD7C 22F5      BEQ MDOR10      ; nope  
  
CD7E 3F      SWI RANDOM (0:3)  
CD7F 07      FCB      ;  
CD80 8403      ANDA #3      ;  
CD82 978A      STA DIR      ;this will be our direction  
CD84 E5A6      BITB A,Y      ;can we put a door in?  
CD86 26FB      BNE MDOR10      ; nope

; Found a suitable place for a door...

CD88 EAC6      ORB A,U      ;put the door in  
CD8A E784      STB ,X      ;  
  
CD8C DC88      LDD ROW      ;calculate coordinates  
CD8E BBD11B      JSR STEP      ;remember to fix adjoining cell  
CD91 D68A      LDB DIR      ;load original direction  
CD93 CB02      ADDB #2      ;opposite wall  
CD95 C403      ANDB #3      ;MOD 4  
CD97 A684      LDA ,X      ;load adjoining cell features  
CD99 AAC5      ORB B,U      ;mark new door  
CD9B A784      STA ,X      ;update cell  
  
CD9D 35F6      PULS A,B,X,Y,U,PC      ;restore regs/exit

## ; Dungeon Generator Seeds

CD9F	LVTAB	EQU	\$	
CD9F 73	;DEBUG	FCB	\$25	;level-0
CDA0 C7		FCB	\$73	;level-1
CDA1 5D		FCB	\$C7	;level-2
CDA2 97		FCB	\$5D	;level-3
CDA3 F3		FCB	\$97	;level-4
CDA4 13		FCB	\$F3	;level-5
CDA5 87		FCB	\$13	
		FCB	\$87	

## ; Bit Mask Table

CDA6 03	MSKTAB	FCB	Z00000011	;north
CDA7 0C		FCB	Z00001100	;east
CDA8 30		FCB	Z00110000	;south
CDA9 C0		FCB	Z11000000	;west

## ; Regular Door Table

CDAA 01	DORTAB	FCB	HF.DOR	;north
CDAB 04		FCB	HF.DOR*4	;east
CDAC 10		FCB	HF.DOR*16	;south
CDAD 40		FCB	HF.DOR*64	;west

## ; Secret Door Table

CDAE 02	SDRTAB	FCB	HF.SDR	;north
CDAF 08		FCB	HF.SDR*4	;east
CDB0 20		FCB	HF.SDR*16	;south
CDB1 80		FCB	HF.SDR*64	;west

; END

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MAPPER - Map Display Generator

PAGE 28

```

INCLUD B:MAPPER.ASM
!!!!!! MAP DISPLAY GENERATOR !!!!!!
NAM MAPPER
LIBRY CD
XDEF MAPPER
XREF FNOOBJ,MAP32
PSCT

MAPPER: Generate a Top-View Display of the Dungeon
This routines creates a top-view map of the current dungeon
level in the alternate screen (FLOP).

Legend:
Black - Clear path
Blue - Door
Red - Secret Door
White - Wall

CDB2 DE0B MAPPER LDU FLOP ;use alternate VDB
CDB4 CC1F1F LDD #$1F1F ;set intial row,col
CDB7 DD7C STD DROW ;.

CDB9 DC7C MAPP10 LDD DROW ;load current position
CDBB 8D54 BSR DSP32 ;calculate display cell addr
CDC0 BCCCC7B JSR MAP32 ;calculate cell addr
CDC0 5F CLR.B ;assume empty cell (solid black)
CDC1 A684 LDA ,X ;walled in cell?
CDC3 4C INCA ;nope
CDC4 2601 BNE MAPP20 ;.
CDC6 5A DECB ;unoccupiable cell (solid white)

; Store Empty/Full Cell Pattern into Display

CDC7 8606 MAPP20 LDA #6 ;loop count
CDC9 E7A4 MAPP22 STB 0,Y ;store byte
CDCB 31A820 LEAY 32,Y ;advance to next scan line
CDCE 4A DECA ;decrement and loop
CDCF 26F8 RNE MAPP22 ;.

; Bump Counters

CDD1 0A7D DEC DCOL ;inner loop
CDD3 2AE4 BPL MAPP10 ;.

CDD5 861F LDA #31 ;outer loop
CDD7 977D STA DCOL ;(reset inner loop counter)
CDD9 0A7C DEC DROW ;.
CDDB 2ADC BPL MAPP10 ;.

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MAPPER - Map Display Generator

PAGE 29

## ; Display Other Features?

CDD0 0D94	TST	MAPFLG	; check the flag
CDDF 274A	BEQ	MAPP50	; just show our position

## ; Mark Objects

CDE1 0F91	CLR	OFINDF	; start at top of list
CDE3 BDCF63	MAPP30	JSR FNDOBJ	; find the next object on this level
CDE6 270F	BEQ	MAPP40	; done

CDE8 6D05	TST	P.DCOWN,X	; owned by anyone?
CDEA 26F7	BNE	MAPP30	; yes

CDEC ECO2	LDD	P.DCROW,X	; load position
CDEE BD21	BSR	DSP32	; calculate display addr
CDF0 CC0008	LDD	#\$0008	; mark object
CDF3 BD28	BSR	MARK4	
CDF5 20EC	BRA	MAPP30	; loop until done

## ; Mark Creatures

CDF7 BE03C3	MAPP40	LDX #CCBLND-CC.LEN	; starting point
C DFA 308811	MAPP42	LEAX CC.LEN,X	; advance ptr
CDFD BC05F4	CMPX	#CCBEND	; done yet?
CE00 2729	BEQ	MAPP50	; yes

CE02 6DOC	TST	P.CCUSE,X	; live creature?
CE04 27F4	BEQ	MAPP42	; nope

CE06 ECOF	LDD	P.CCROW,X	; load position
CE08 BD07	BSR	DSP32	; calculate display addr
CE0A CC1054	LDD	#\$1054	; mark creature
CE0D BD0E	BSR	MARK4	
CE0F 20E9	BRA	MAPP42	; loop until done

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MAPPER - Map Display Generator

PAGE 30

;; DSP32: Local Subroutine to Calculate Display Offset

Inputs:

A = Row  
B = Col

U = Ptr to VDB

Returns:  
Y = Display Addr

CE11 1F02	DSP32	TFR D,Y	
CE13 C6C0		LDB #32*6	;duplicate row,col
CE15 3D		MUL	;multiply by row byte count
CE16 E3C4		ADDI P,VDBAS,U	;add display offset
CE18 1E02		EXG D,Y	;swap
CE1A 31A5		LEAY B,Y	
CE1C 39		RTS	;restore regs/exit

;; MARK4: Local Subroutine to Mark the Inner Cells

CE1D A7A820	MARK4	STA 32*1,Y	
CE20 E7A840		STB 32*2,Y	
CE23 E7A860		STB 32*3,Y	
CE26 A7A90080		STA 32*4,Y	
CE2A 39	COMRTS1	RTS	

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
MAPPER - Map Display Generator

PAGE 31

```

; Mark Player
CE2B BC13    MAPP50  LDD     PROW      ;our position
CE2D 8DE2    BSR     DSP32      ;calculate display addr
CE2F CC2418   LDD     #$2418    ;mark our position
CE32 8DE9    BSR     MARK4     ;"X" marks the spot

; Mark Vertical Features
CE34 9E86    LBX     VFTPTR    ;vertical features index
CE36 8D00    BSR     MAPP60    ;up vertical features

; Common Subroutine to mark Vertical Features
CE38 A680    MAPP60  LDA     X+      ;end of table?
CE3A 2BEE    BMI     COMRTS1  ;yes - we are done
CE3C EC81    LDD     X++      ;load position
CE3E 8DD1    BSR     DSP32      ;calculate display addr
CE40 CC3C24   LDD     #$3C24    ;mark the cell
CE43 8DD8    BSR     MARK4     ;+
CE45 20F1    BRA     MAPP60    ;loop until done

; END

```

```

INCLUD B:VIEWER.ASM
;!!!!!! D U N G E O N V I E W E R !!!!!!
; SETSCL: Set Scaling Based on RANGE
CE47 3412    SETSCL   PSHS    A,X      ;save regs
CE49 8ECF48    LDX      #HLFSCL   ;assume half-step forward
CE4C 0073      TST      HLFSTP   ;
CE4E 260C      BNE      SSCL10   ;
CE50 30890001  ;       LEAX     (BAKSCL-HLFSCL),X ;assume half-step backward
CE54 0074      TST      BAKSTP   ;
CE56 2604      BNE      SSCL10   ;
CE58 3089FFF5  ;       LEAX     (NORSCL-BAKSCL),X ;normal scaling
CE5C 968B      SSCL10  LDA      RANGE    ;load range
CE5E A686      LDA      A,X      ;load table entry
CE60 974F      STA      VXSCAL   ;set scaling parameters
CE62 9750      STA      VYSCAL   ;
CE64 3592      PULS    A,X,PC  ;restore regs/exit

```

VIEWER: Subroutine to display, in 3-D perspective view,  
the player's view of the inside of the maze.

INPUTS:

PROW Row occupied by player (Y-coordinate)

PCOL Column occupied by player (X-coordinate)

PDIR Direction player is facing

0 = North

1 = East

2 = South

3 = West

RLIGHT Ambient light level in the dungeon. Values can range  
from 0 to 15 where zero equals total ambient darkness  
and 5 to 8 is the typical ambient light level at  
dungeon levels 1 or 2, and 3 to 5 is the ambient  
light level at dungeon levels 3 or 4.

MLIGHT Magical light level. This value is analogous to  
TLIGHT, but only affects the visibility of magical  
doors, objects and magical creatures.  
The ambient light level, RLIGHT does not affect  
the visibility of magical things.

\* \* \* \* \*

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VIEWER - Dungeon Viewer

PAGE 34

```

; VIEWER: Initialization
CE66 3F      VIEWER SWI
CE67 09      FCB      ZFLOP    ;clear alternate screen
CE68 0F8B      CLR      RANGE   ;
CE6A DC13      LDD      PROW    ;reset range
CE6C DD7C      STD      DROW    ;load player position
;                                ;make copy

; Main Loop
CE6E 8DD7      VIEW10  BSR      SETSCL  ;set scaling parameters
CE70 DC7C      LDD      DROW    ;load our position
CE72 BDCC7B      JSR      MAP32   ;calc cell addr
CE75 A684      LDA      ,X      ;load cell contents
CE77 CE09F4      LDU      #NEIBOR ;extracted data goes here
CE7A BE0004      LDX      #4      ;loop counter
CE7D 1F89      TFR      A,B    ;make copy of cell contents
CE7F C403      ANDB     #3      ;extract 2 LSBits
CE81 E744      STB      4,U    ;make two copies
CE83 E7C0      STB      ,U+    ;
CE85 44      LSRA      ;shift over
CE86 44      LSRA      ;decrement and loop
CE87 301F      LEAX     -1,X
CE89 26F2      BNE      VIEW12
;                                ;
CE8B B623      LDR      PDIR    ;set base addr for features
CE8D CE09F4      LDU      #NEIBOR ;;
CE90 33C5      LEAU     B,U    ;;
;
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VIEWER - Dungeon Viewer

PAGE 35

| Forward-Looking Architectural Features Drawing Routine

```
CE92 108EDBDE VIEW20 LDY    #FLATAB      ;table base addr
CE96 A6A0   VIEW22 LDA    ,Y+          ;load relative direction change
CE98 2B3E   BMI    VIEW30      ; we are done
;                                ;load cell wall features
CE9A E6C6   LDR    A,U          ;times 2 for word entries
CE9C 58     LSLB   CMPB   #HF.SDR*2  ;secret door?
CE9D C104   BNE    VIEW24      ;nope
CE9F 2608   ;
;                                ;load secret door vector list addr
CEA1 AEA5   LDX    B,Y          ;draw with magic light
CEA3 0A75   DEC    MAGFLG
CEA5 8D27   BSR    DRAWIT
CEA7 C606   LDB    #HF.WAL*2  ;then draw a wall (regular light)
;
CEA9 AEA5   VIEW24 LDX    B,Y          ;load vector list addr
CEAB 8D21   BSR    DRAWIT
;                                ;draw with regular light
CEAD 3128   LEAY   BRA    VIEW22      ;advance to next table entry
CEAF 20E5   ;loop
CEB1 39     COMRTS2 RTS
```

## ; PDRAW: Peek-A-Boo Drawing Routine

CEB2 1F12	PDRAW	TFR	X,Y	; save vector list ptr
CEB4 6DC5		TST	B,U	; passage to our left?
CEB6 26F9		BNE	COMRTS2	nope
CEB8 DB23		ADD	PDIR	; effect relative change in direction
CEBA D78A		STB	DIR	
CEBC DC7C		LDD	DROW	; our position
CEBE BBD11B		JSR	STEP	
CEC1 BDCF82		JSR	CFIND	
CEC4 27EB		BEQ	COMRTS2	; anything here?
CEC6 1E12		EXG	X,Y	nope
				; swap pointers/fall into CMRDRW

## ; CMRDRW: Draw Magical/Non-Magical Creatures

CEC8 6D22	CMRDRW	TST	P,CCMGO,Y	; magic creature?
CECA 2702		BEQ	DRAWIT	non-magical creature
CECC 0A75		DEC	MAGFLG	; draw with magic light/fall into DRAWIT

## ; DRAWIT: Draw Vector List in Alternate Screen

CECE 3440	DRAWIT	PSHS	U	; save regs
CED0 3F		SWI		; set the lighting levels
CED1 00		FCR	SETFAD	
CED2 DE0B		LDD	FLOP	; alternate screen VDR
CED4 3F		SWI		; draw it
CED5 01		FCR	VCTLST	
CED6 35C0		PULS	U,PC	; restore regs/exit

## ; Forward-Looking Creature Drawing Routine

CED8 DC7C VIEW30 LDD DROW ;our position  
CEDA BDCFB2 JSR CFIND ;creature here?  
CEDD 270C BEQ VIEW32 ; nope

CEDF 1F12 TFR X,Y ;copy CCR ptr  
CEE1 E62D LDB P:CCTYP,Y ;load creature type  
CEE3 58 LSLB ;\*2 for word offsets  
CEE4 8EDAA3 LDX #FWDCRE ;forward looking creature lists  
CEE7 AE85 LDX B,X ;load vector list addr  
CEE9 8DD0 BSR CMRIRW ;draw the creature

CEEB C603 VIEW32 LDB #3 ;peek-a-boo to our left  
CEEF 8EDCB0 LDX #LPEEK ;  
CEF0 8DC0 BSR PDRAW ;

CEF2 C601 ;  
CEF4 8EDCB9 LDB #1 ;peek-a-boo to our right  
CEF7 8DE9 LDX #RPEEK ;  
BSR PDRAW ;

## ; Draw vertical features

CEF9 8EED3C LDX #CELINE ;assume ceiling line  
CEFC DC7C LDD DROW ;our position  
CEFE BDCFE1 JSR VFIND ;any vertical features?  
CF01 2B06 BMI VIEW40 ; nope - draw ceiling line

CF03 8EIDCC2 LDX #FWDVER ;forward looking vertical features  
CF06 48 LSLA ;\*2 for word offset  
CF07 AE86 LDX A,X ;load vector list addr  
CF09 8DC3 BSR DRAWIT ;draw with regular light

## ; Draw objects

CF0B 0F91 VIEW52 CLR OFINDF ;start at top of object lists  
CF0D DC7C ! DD DROW ;load position  
CF0F BDCF53 JSR OFIND ;any objects?  
CF12 2710 BEQ VIEW60 ; nope

CF14 A60A LDA P:OCCLS,X ;load object class  
CF16 48 LSLA ;\*2 for word offsets  
CF17 8ED9EE LDX #FWDOBJ ;forward-looking object vector lists  
CF1A AE86 LDX A,X ;load vector list addr  
CF1C 0A75 DEC MAGFLG ;draw with both magic/regular light  
CF1E 8DAE BSR DRAWIT ;  
CF20 8DAC BSR DRAWIT ;  
CF22 20E9 BRA VIEW52 ;loop until there are no more objects

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VIEWER - Dungeon Viewer

PAGE 38

```
; Check "Line-of-Sight"
CF24 6BC4    VIEW60  TST    0,U      ;passage in front of us?
CF26 2615    BNE    VIEW99      ; nope - we are done

; Take a step and increase range
CF28 9623    LDA    PDIR      ;restore original direction
CF2A 978A    STA    DIR       ;
CF2C DC7C    LDD    DROW      ;our position
CF2E BDD11B   JSR    STEP      ;take a step
CF31 DD7C    STD    DROW      ;update our position
;
CF33 0C8B    INC    RANGE      ;have we gone far enough?
CF35 968B    LIA    RANGE      ;
CF37 8109    CMPA   #9        ;
CF39 102FFF31 LBLE   VIEW10    ;nope
;
CF3D 39      VIEW99  RTS      ;common RETURN
```

Scaling constants - look-up table  
 Values in this table are indexed on RANGE (the distance of an object or feature from the player).  
 Values in the table are fixed point, radix-2, binary.

B B B B B B B B  
 ^  
 +-----< Binary Point

CF3E	NORSCL	EQU	\$	;base
				Range = 0: (cell occupied by player)
CF3E C8		FCB	%11001000	; 1 72/128ths size
				Range = 1: (cell immediately in front of player)
CF3F 80		FCB	%10000000	; 128/128ths size (1:1)
				Range = 2: (next cell out)
CF40 50		FCB	%01010000	; 80/128ths size
				Range = 3:
CF41 32		FCB	%00110010	; 50/128ths size
				Range = 4:
CF42 1F		FCB	%00011111	; 31/128ths size
				Range = 5:
CF43 14		FCB	%00010100	; 20/128ths size
				Range = 6:
CF44 0C		FCB	%00001100	; 12/128ths size
				Range = 7:
CF45 08		FCB	%00001000	; 8/128ths size
				Range = 8:
CF46 04		FCB	%00000100	; 4/128ths size
				Range = 9:
CF47 02		FCB	%00000010	; 2/128ths size

```
; "Half-Step" scaling constant table
CF48    HLFSC1 EQU    $           ;
; Range = 0 plus a half step
CF49 FF   FCB     255          ; 1 127/128ths size
; Range = 1 plus a half step
CF49 9C   BAKSCL EQU    $           ; *DEBUG
CF49 9C   FCB     156          ; 1 28/128ths size
; Range = 2 plus a half step
CF4A 64   FCB     100          ; 100/128ths size
; Range = 3 plus a half step
CF4B 41   FCB     65           ; 65/128ths size
; Range = 4 plus a half step
CF4C 28   FCB     40           ; 40/128ths size
; Range = 5 plus a half step
CF4D 1A   FCB     26           ; 26/128ths size
; Range = 6 plus a half step
CF4E 10   FCB     16           ; 16/128ths size
; Range = 7 plus a half step
CF4F 0A   FCB     10           ; 10/128ths size
; Range = 8 plus a half step
CF50 06   FCB     6            ; 6/128ths size
; Range = 9 plus a half step
CF51 03   FCB     3            ; 3/128ths size
; End of scaling table 2
CF52 01   FCB     1           ; *DEBUG
;
END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMCRE - Creature Movement Routines

PAGE 41

INCLUDE B1 CONCRETE ASME

## CREATURE MOVEMENT ROUTINES

**OFIND:** Find an unowned object based on position

Inputs:

A - Row  
E - Col

R = Column  
RETENTION

**OPTINDF** - flag to indicate ptr "rewind" (0 - start again)

### Returns

*X* = pointer to Object Control Block (OCB)  
*Z* = condition code set if search fails

2 = condition code set if search failed

```

CF53 8D0E      OFIND    BSR      FNOBJ      find next object on our level
CF55 270B      BEQ      OFIND9     search failure
CF57 10A302    CMPD     P,OCROW,X position match?
CF5A 26F7      BNE      OFIND     nope - try again
CF5C 6B05      TST      P,OCOWN,X owned by anyone?
CF5E 26F3      BNE      OFIND     yes - try again
CF60 1CFB      ANDCC    #(CC.Z XOR $FF) winner!
CF62 39        RTS      OFIND9     bye

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMCRE - Creature Movement Routines

PAGE 42

FNDOBJ: Find the next object on our level  
 Inputs:  
 OFINIDF - flag to indicate ptr "rewind" (0 - start again)  
 Returns:  
 X - ptr to Object Control Block (OCB)  
 Z - condition code set if search fails

CF63 3402	FNDOBJ	PSHS	A	; save regs
CF65 9681		LDA	LEVEL	; load current level number
CF67 9E92	;	LDX	OFINIDP	; assume old ptr
CF69 0B91		TST	OFINIDF	; initialize ptr?
CF6B 2605		BNE	FIND10	; nope
CF6D 8E0B07		LDX	#OCLND-OC.LEN	; top of table
CF70 0A91		DEC	OFINIDF	; say we have started
CF72 300E	FIND10	LEAX	OC.LEN,X	; advance to next entry
CF74 9F92		STX	OFINIP	; update ptr
CF76 9C0F		CMPX	OCBPTR	; reach end of table?
CF78 2706		BEQ	FIND99	; yes - loser
CF7A A104	;	CMPA	P,OCLVL,X	; is this object on our level?
CF7C 26F4		BNE	FIND10	; nope
CF7E 1CFB		ANDCC	#\$CC.Z XOR \$FF	; winner!
CF80 3582	FIND99	PULS	A,PC	; restore regs/exit

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMCRE - Creature Movement Routines

PAGE 43

```
CFIND: Find a creature based on position
```

Inputs:

A - Row  
B - Column

Returns:

X - Pointer to CCB (invalid if search fails)  
Z - condition code set if search fails

CF82 8E03C3	CFIND	LDX	#CCBLND-CC.LEN	;load pointer to CCB table
CF85 308811	CFND10	LEAX	CC.LEN,X	;advance to next CCB
CF88 8C05F4		CMPX	#CCREND	;check for overrun
CF8B 2709		BEQ	CFND99	;loser!
CF8D 10A30F		CMPD	P.CCRROW,X	;is this the creature we want?
CF90 26F3		BNE	CFND10	;nope
CF92 6D0C	CFND30	TST	P.CCUSE,X	;insure that this is a live creature
CF94 27EF		BEQ	CFND10	;oops - dead (invalid) creature
CF96 39	CFND99	RTS		;winner!

```
FNDCEL: Find an occupiable cell
```

Returns:

A - Row  
B - Col

CF97 3416	FNDCEL	PSHS	A,B,X	;save regs
CF99 BDCC71	FCEL10	JSR	FNDCEL	;pick a cell, any cell
CF9C EDE4		STD	0,S	;assume row/col is ok
CF9E A684		LDA	,X	;can we live here?
CFA0 4C		INCA		(solid wall = \$FF)
CFA1 27F6		BEQ	FCEL10	;nope
CFA3 3596		PULS	A,B,X,PC	;restore regs/exit

```

;!!!!!! CREATOR POPULATOR !!!!!!
;CBIRTH: Give Birth to a Creature
;Inputs:
;    A - Creature Type
;Outputs:
;    - Creates a Creature Control Block (CCB)
;    - Creates and schedules CMOVE TCB

CFA5 3476 CBIRTH PSHS A,B,X,Y,U      ;save some regs
; Find the next available CCB
CFA7 CE03C3 CBIR00 LDU #CCBLND-CC.LEN ;CCB table base addr
CFAA 33C811 LEAU CC.LEN,U           ;advance to next CCB
CFAD 6A4C TST P.CCUSE,U           ;is this CCB free?
CFAF 26F9 BNE CBIR00             ;no
CFB1 6A4C DEC P.CCUSE,U           ;mark CCB in-use
; Fill in the new CCB
CFR3 A74D STA P.CCTYP,U          ;store the creature type
CFB5 C608 LDB #CD.LEN            ;calculate offset into CDBTAB
CFB7 30 MUL                   ;add
CFB8 C3DABB ADDI #CDBTAB         ;add CDB table base addr
CFBB 1F02 TFR D,Y               ;source ptr
CFBD 1F31 TFR U,X               ;destination ptr
CFBF 8608 LDA #CD.LEN            ;number of bytes to copy
CFC1 BDC04B JSR COPY              ;copy ROM info
; Place the new creature in the maze
CFC4 8DD1 CBIR20 BSR FNDCEL       ;find an occupiable cell
CFC6 8DBA BSR CFIND             ;does somebody else already live here?
CFC8 26FA BNE CBIR20             ;oops, somebody here already
CFC9 ED4F STD P.CCRROW,U           ;save position in CCB
; Create the Creature Movement TCB
CFCC 1F31 TFR U,X               ;copy CCB ptr
CFCE BDC25C JSR GETTCB           ;allocate core for TCB
CFD1 AF45 STX P.TCDDTA,U          ;store CCB ptr
CFD3 CC0041 LDD #CMOVE            ;store routine name
CFD6 ED43 STD P.TCRTN,U           ;P.
CFD8 A606 LDA P.CCTMV,X           ;load the delay time
CFDA C604 LDB #Q.TEN              ;into the TENTH queue
CFDC BDC21D JSR QUEADD           ;add to the correct queue
CFDF 35F6 PULS A,B,X,Y,U,PC        ;restore regs/go home

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMCRE - Creature Movement Routines

PAGE 45

## VFIND: Vertical Feature Search Routine

## Inputs:

A - row  
B - column

## Returns:

A - vertical feature code  
N - condition code set if search fails

CFE1 3456	VFIND	PSHS	A,B,X,U	; save regs
CFE3 DE86		LDD	VFTPTR	; base addr of vertical features table
CFE5 8D0B		BSR	VSUB00	; check for vertical features up
CFE7 4B		TSTA		; ; winner
CFE8 2A04		BPL	VFIND9	
CFEA BD06		BSR	VSUB00	; check for vertical features down
CFEC BB02		ADDA	#2	
CFEE A7E4	VFIND9	STA	0,S	; store return code
CFF0 35D6		PULS	A,B,X,U,PC	; restore regs/exit
CFF2 A6C0	VSUB00	LDA	,U+	; load vertical feature code
CFF4 2B06		BMI	VSUB99	; end of table
CFF6 AEC1		LDX	,U++	; load vertical feature position
CFF8 AC62		CMFX	2,S	; did we get a match?
CFFA 26F6		BNE	VSUB00	; nope
CFFC 39	VSUB99	RTS		; bye

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.0B

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMCRE - Creature Movement Routines

PAGE 46

; Vertical Features Table

CFFD	OFTTAB	EQU	\$	level	0	1	2	3	4
CFFD 80		FCB	-128						
CFFE 010017		FCB	1,00,23	ladder	DN	UP			
D001 000F04		FCB	0,15,04	hole	DN	UP			
D004 001411		FCB	0,20,17	hole	DN	UP			
D007 011C1E		FCB	1,28,30	ladder	DN	UP			
D00A 80		FCB	-128						
D00B 010203		FCB	1,02,03	ladder	DN	UP			
D00E 00031F		FCB	0,03,31	hole	DN	UP			
D011 001314		FCB	0,19,20	hole	DN	UP			
D014 001F00		FCB	0,31,00	hole	DN	UP			
D017 80		FCB	-128						
D018 80		FCB	-128						
D019 00001F		FCB	0,00,31	hole			DN	UP	
D01C 000500		FCB	0,05,00	hole			DN	UP	
D01F 00161C		FCB	0,22,28	hole			DN	UP	
D022 001F10		FCB	0,31,16	hole			DN	UP	
D025 80		FCB	-128						
D026 80		FCB	-128						

## ; CREGEN: Creature Regeneration Task

```

D027 9E82    CREGEN   LDX     CMXPTR      ;load current matrix level ptr
D029 C60B    LDB     #CTYPES-1   ;number of creatures - 1
D02B 4F       CLRA
;
D02C AB85    CGEN10   ADDA    B,X      ;figure number of creatures on level
D02E 5A       DECB
D02F 2AFB    BPL     CGEN10      ;are we done?
;
D031 8120    CMPA    #32      ;max number of creatures on this level?
D033 2408    BHS     CGEN99      ;yes
;
D035 3F       SWI
D036 07       FCB     RANDOM     ;create a random creature
D037 8407    ANDA    #7        ;(0-7)
D039 8B02    ADDA    #2        ;(2-9)
D03B 6C86    INC     A,X      ;allow an extra creature of this type
;
D03D CC0508    CGEN99   LDD     #(5*256)+Q,MIN ;re-sched every 5 minutes
D040 39       RTS
;
END

```

INCLUD B:CRETUR.ASM

!!!!!! CREATURE MOVEMENT ROUTINES !!!!!!!

The actions of a creature are governed primarily by  
what is in the cell with him/her/it:

1) Treasures/Magical Objects

A creature will pick up whatever is in the cell.  
If there is more than one object, the creature will  
pick them up one at a time. Since the highest priority  
action for a creature is to gather objects, the human  
can delay creature attacks by dropping objects.

2) Attacking Unlucky Adventurers

If a human should happen to be in the same cell as  
creature, the creature will attack. Creature attacks will  
be repeated at "attack" speed (shorter delay times).

3) Move toward Tasty Humans

If there is nothing in the cell, the creature will  
"look" for the human. If it can see the human, the  
creature will attempt to close. The creature will move  
randomly if it cannot see the human.

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CRETUR - Creature Movement Routines

PAGE 49

## ; CMOVE: Creature Movement Task

D041 10AE45	CMOVE	LDY	P.TCDTA,U	;load CCR pointer
D044 002B		TST	FRZFLG	;are we "frozen"?
D046 2622		BNE	CMOV12	; yes!

## ; Kill Off Dead Creatures

D048 E62C	LDB	P.CCUSE,Y	;are we still alive?	
D04A 2601		BNE	CMOV10	; yes
D04C 39		RTS		;dead creature - zap the TCB

## ; Should I pick up objects?

D04D A62D	CMOV10	LDA	P.CCTYP,Y	;what am I?
D04F 8106		CMPA	#6	
D051 271A		BEQ	CMOV20	; i am a SCORPION!
D053 810A		CMPA	#10	
D055 2C16		BGE	CMOV20	; i am a WIZZARD!

## ; Look for Treasures/Objects

D057 EC2F	LDD	P.CCRROW,Y	;our position
D059 0F91	CLR	OFINDF	;look for an object
D05B BDCF53	JSR	OFIND	
D05E 270D	BEQ	CMOV20	; no objects here

## ; Pick up an object

D060 EC28	LDD	P.CCOBJ,Y	;load current object ptr	
D062 AF28	STX	P.CCOBJ,Y	;new object becomes our first object	
D064 ED84	STD	P.OCPTR,X	;link new object with rest of list	
D066 6A05	DEC	P.OCOWN,X	;mark object as creature owned	
D068 3F	SWI			
D069 0E	FCB	PUPDAT	;request immediate update	
D06A 7ED103	CMOV12	JMP	CMOV90	;done for this turn

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CRETUR - Creature Movement Routines

PAGE 50

## Attack Any Unlucky Adventurers

D06D EC2F	CMOV20	LDD	P.CCROW,Y	;is the player here?
D06F 109313		CMPD	PROW	,
D072 263E		BNE	CMOV50	;nope
D074 A62D		LDA	P.CCTYP,Y	;creature type
D076 C6FF		LDB	#\$FF	;loud
D078 3F		SWI		,
D079 1C		FCB	SOUNDS	,

## Set Player Shielding Parameters

D07A CC8080	LDD	#\$8080	load minimal shielding values
D07D 9E1D	LDX	PLHAND	process left hand
D07F 8D1D	BSR	SHIELD	,
D081 9E1F	LDX	PRHAND	process right hand
D083 8D19	BSR	SHIELD	,
D085 971A	STA	PMGH	store shielding parameters
D087 D71C	STB	PPHD	,

## Process Actual Attack

D089 1F21	TFR	Y,X	creature is the attacker
D08B CE0217	LDU	#PLRBLK	the player is the defender
D08E BDD3D7	JSR	ATTACK	drop the hammer!
D091 2B06	BMI	CMOV30	,missed...

## SOUND\$: A\$KLK3

D093 3F	SWI		;CLANK!!!
D094 1B	FCB	ISOUND	,
D095 13	FCB	A\$KLK3	,
D096 BDD40C	JSR	DAMAGE	we took a chunk out of him

CMOV30 SWI  
FCB HUPDAT  
JMP CMOV92

D099 3F	CMOV30	SWI	update the heart rate
D09A 0C		FCB	,
D09B 7ED10F		HUPDAT	prepare for next attack

## SHIELD: Local routine to process Shield

D09E 3416	SHIELD	PSHS	A,B,X	;save regs
D0A0 270E		BEQ	SHLD99	,empty hand!
D0A2 A60A		LDA	P.OCCLS,X	;load class
D0A4 8103		CMPA	#K.SHIE	is it a shield?
D0A6 2608		BNE	SHLD99	,nope
D0A8 AE06		LDX	P.OCXXX,X	store greater value as return parms
D0AA ACE4		CMPX	0,S	,
D0AC 2402		BHS	SHLD99	current shielding is superior
D0AE AFE4		STX	0,S	upgrade shielding
D0B0 3596	SHLD99	PULS	A,B,X,PC	restore regs/exit

; See if we are vertically/horizontally in line with player

```
D0B2 9113    CMOV50  CMFA   PROW
D0B4 2600    BNE     CMOV52  ;, nope
              ;
D0B6 A6A810  LDA     P.CCCOL,Y ;are we west/east of player?
D0B9 C601    LDB     #1      ;assume we are west of player
D0B8 9014    SUBA   PCOL
D0BD 2B11    BMI    CMOV60 ;west of player
D0BF C603    LDB     #3      ;east of player
D0C1 2000    BRA    CMOV60 ;
              ;
D0C3 EC2F    CMOV52  LDD    P.CCROW,Y ;same column as player?
D0C5 B114    CMFB   PCOL
D0C7 261B    BNE    CMOV70 ;, nope
              ;
D0C9 C602    LDB    #2      ;assume north
D0CB 9013    SUBA   PROW
D0CD 2B01    BMI    CMOV60 ;north of player
D0CF 5F      CLRB
              ;south of player
```

; In-Line with Player - can we see him?

```
D0D0 D78A    CMOV60  STB    DIR   ;store the direction
D0D2 EC2F    LDD    P.CCROW,Y ;load current position
              ;
D0D4 8D60    CMOV62  BSR    STEPOK ;take a tentative step
D0D6 260C    BNE    CMOV70 ;loser - we hit a wall!
D0D8 109313  CMPD   PROW ;player here?
D0D8 26F7    BNE    CMOV62 ;not here - keep looking
```

; Saw the player!

```
D0D9 D68A    LDB    DIR   ;load direction of travel
D0DF E72E    STB    P.CCDIR,Y ;face creature toward player
D0E1 5F      CLRB  ;no relative change in direction
D0E2 201D    BRA    CMOV78 ;move toward player if possible
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CRETUR - Creature Movement Routines

PAGE 52

## ; Select Random Movement Sequence

D0E4 8E0114	CMOV70	LDX #MOVTAB	;assume fwd/ltr sequence
D0E7 3F	SWI		;use other sequence?
D0E8 07	FCB RANDOM		;
D0E9 4D	TSTA		;
D0EA 2802	BMI CMOV72		nope
D0EC 3003	LEAX 3,X		;use fwd/rtr/ltr sequence

## ; Attempt to change directions 25% of the time

D0EE 8403	CMOV72	ANDA #3	;use 2 LSBits of random for that 25%
D0F0 2602	BNE CMOV74		;not this time
D0F2 3001	LEAX 1,X		;LT/RT or (RT/LT) before FWD (magic!)

## ; Try the preferential directions

D0F4 8603	CMOV74	LDA #3	;loop count
D0F6 E680	CMOV76	LDB ,X+	;load next relative change
D0F8 8055	BSR CWALK		;attempt to move
D0FA 2707	BEQ CMOV90		winner!
D0FC 4A	DECA		try again
D0FD 26F7	BNE CMOV76		;

D0FF C602	LDB #2		;finally try to back out
D101 804C	BSR CWALK		;don't care if we can't move

D103 A626	CMOV90	LDA P,CCTMV,Y	;assume movement delay time
-----------	--------	---------------	-----------------------------

D105 AE2F	LDX P,CCROW,Y		;same square as player?
D107 9C13	CMPX PROW		;
D109 2604	BNE CMOV98		nope
D10B 3F	SWI FCB		force update
D10C 0E	PUPDAT		;
D10D OFB5	CLR NEWLUK		no need for delayed update

D10F A627	CMOV92	LDA P,CCTAT,Y	;use attack delay time
D111 C604	CMOV98	LDB #Q,TEN	into the TENTH queue
D113 39	CMOV99	RTS	;that's all folks!!!

## ; Preferential Random Movement Table

D114 00	MOV TAB	FCB 0	;forward
D115 03		FCB 3	;left
D116 01		FCB 1	;right
D117 00		FCB 0	;forward
D118 01		FCB 1	;right
D119 03		FCB 3	;left
D11A 00		FCB 0	;forward

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CRETUR - Creature Movement Routines

PAGE 53

STEP: Take a step in the current direction

Inputs:

A - row position  
 B - column position  
 DIR - direction to move

Returns:

A - new position  
 B - new position  
 X - cell address

D11B 3406	STEP	PSHS A,B	;stack row/col
D11D D68A		LDB DIR	;load current direction
D11F C403		ANDB #Z000000011	;MOD 4
D121 58		LSLB	#\$2 for word offset
D122 8ED12E		LDX #STPTAB	;load base addr
D125 EC85		LDI B,X	;load row/col offsets
D127 ABE0		ADD A, <sup>,S+</sup>	;offset row
D129 EBE0		ADD B, <sup>,S+</sup>	;offset column
D12B 7ECC7B		JMP MAP32	;calculate cell addr

Row/Column Modification Table

D12E FF00	STPTAB	FCB -1,0	;north
D130 0001		FCB 0,1	;east
D132 0100		FCB 1,0	;south
D134 00FF		FCB 0,-1	;west

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CRETUR - Creature Movement Routines

PAGE 54

; STEPOK: Attempt to Take a Step

Inputs:

A - Row  
B - Column

DIR - direction

Returns:

A - Possibly Updated Row

B - Possibly Updated Column

X - Possible Updated to Cell Addr

Z - Condition code set if step is legal

D136 3476	STEPOK	PSHS A,B,X,Y,U	; save regs
D138 8DE1		BSR STEP	; take a tentative step
D13A BDCC8E		JSR BORDER	; perform border check
D13B 260E		BNE STOK99	; loser - o.b.
D13F 1F03		TFR D,U	; save updated row/col
D141 A684		LDA ,X	; load cell features
D143 4C		INCA	; solid walls (\$FF)?
D144 2706		BEQ STOK98	; yes - loser
D146 EFE4		STU 0,S	; update position
D148 AF82		STX 2,S	; update cell addr
D14A 8601		LDA #1	; insure Z-condition code will be set
D14C 4A	STOK98	DECA	; set/reset Z-condition code
D14D 35F6	STOK99	PULS A,B,X,Y,U,PC	; restore regs/exit

CWALK: Attempt to move creature in desired direction

**Inputs:**

B - relative change in direction  
Y - pointer to CCB

**Returns:**

P.CCDIR,Y - modified if CWALK wins  
P.CCROW,Y - modified if CWALK wins  
P.CCCOL,Y - modified if CWALK wins  
Z - Condition code set if CWALK wins

**Destroys:**

DIR,ROW,COL

D14F 3416      CWALK      PSHS      A,B,X      ; save some regs  
D151 E82E      ADDB      P.CCDIR,Y      ; effect relative change in direction  
D153 C403      ANDB      #3      ; MOD 4  
D155 D78A      STB      DIR      ;  
;

Is the next cell occupiable?

D157 EC2F      LDD      P.CCROW,Y      ; load current position  
D159 80DB      BSR      STEPOK      ; take a tentative step  
D15B 263C      BNE      CWLK99      ; loser  
;

Somebody here already?

D15D BDCF82      JSR      CFIND      ; use updated coordinates  
D160 2637      BNE      CWLK99      ; occupied cell - loser  
;

D162 ED2F      STD      P.CCROW,Y      ; update creature position  
D164 D68A      LDB      DIR      ; update direction  
D166 E72E      STB      P.CCDIR,Y      ;  
;

Need to update screen?

D168 EC2F      LDD      P.CCROW,Y      ; load position  
D16A 9013      SUBA      PROW      ; absolute row difference  
D16C 2A01      BPL      CWLK10      ;  
D16E 40      NEGA      ;  
D16F D014      SUBB      PCOL      ; absolute column difference  
D171 2A01      BPL      CWLK12      ;  
D173 50      NEGB      ;  
;

D174 B7C1      CWLK12      STB      TO      ; store column difference  
D176 91C1      CMPA      TO      ; bigger difference into ACCA  
D178 2C02      BGE      CWLK20      ;  
D17A 1E89      EXG      A,B      ;  
;

D17C 97C1      CWLK20      STA      TO      ; store big difference  
D17E 8108      CMPA      #8      ; greater than 8?  
D180 2E16      BGT      CWLK92      ; no update/no sounds  
;

D182 C102      CMPPR      #2      ; check little difference  
D184 5E19      BGT      CWLK99      ; no update/no sounds  
;

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
CRETUR - Creature Movement Routines

PAGE 56

; Should we make a sound?

```
D186 3F      SWI
D187 07      FCB    RANDOM      ;make sounds 50% of the time
D188 8501     BITA   #BIT0
D18A 270A     BEQ    CWLK90    ;silence...
```

; Creatures are louder as they approach

```
D18C 96C1     LDA    T0      ;use range to attenuate volume
D18E C61F     LDB    #31
D190 3D      MUL
D191 53      COMB
D192 A62D     LDA    P,CCTYP,Y ;creature type as index
D194 3F      SWI
D195 1C      FCB    SOUNDS
D196 0AB5     CWLK90    DEC    NEWLUK ;request screen update
D198 4F      CWLK92    CLRA   ;winner - set Z-condition code
D199 3596     CWLK99    PULS   A,B,X,PC ;restore regs/go home
;END
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMPLR - Common Player Routines

PAGE 57

```

INCLUD B:COMPLR.ASM
!!!!!! COMMON PLAYER ROUTINES !!!!!!
NAM    COMPLR
LIBRY  CD
XDEF   BURNER,LUKNEW,HSLOW
XREF   ASRID6,MAPPER,ADJTAB,GENTAB,T+TORS
PSCT

; BURNER† Torch Burn-Time Task

D19B DE24    BURNER LDD    PTORCH      ; load ptr to torch OCB
D19D 271D    BEQ    BURN99     ; no torch!

; Decrement timer

D19F A646    LDA    P.OCXXX,U    ; load timer value
D1A1 2719    BEQ    BURN99     ; dead torch
D1A3 4A      DECA   P.OCXXX,U    ; decrement timer
D1A4 A746    STA    P.OCXXX,U    ; update timer
D1A6 8105    CMPA   #5        ; dead torch at FIVE minutes to go
D1A8 2E06    BGT    BURN10    ; still ok

; Torch just died...

D1AA C618    LDB    #T.TORS    ; mark as dead
D1AC E749    STB    P.OCTYP,U    ; insure it shows up as dead
D1AE 6F4B    CLR    P.OCREV,U

; Adjust regular/magic light as necessary

D1B0 A147    BURN10 CMPA   P.OCXXX+1,U    ; adjust regular light?
D1B2 2C02    BGE    BURN20    ; nope
D1B4 A747    STA    P.OCXXX+1,U    ; update regular light

D1B6 A148    BURN20 CMPA   P.OCXXX+2,U    ; adjust magic light?
D1B8 2C02    BGE    BURN99    ; nope
D1BA A748    STA    P.OCXXX+2,U    ; update magic light
D1BC 0A89    BURN99 DEC    NEWLUR    ; request delayed screen update

; SCHEI$ 1,Q,MIN    ; once per minute

D1BE CC0108    LDD    #((1*256)+Q.MIN) ; once per minute
D1C1 39      RTS

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMPLR - Common Player Routines.

PAGE 58

```
; LUKNEW: Task to Update the Screen if necessary
D1C2 0DB5    LUKNEW   TST     NEWLUK      ;did a creature request an update?
D1C4 2607          BNE     LNEW10    ; yes
D1C6 BECDB2          LDX     #MAPPER   ;is the map display up?
D1C9 9CB2          CMPX    DSFMOD0
D1CB 2604          BNE     LNEW99    ; nope
D1CD 0FB5          LNEW10   CLR     NEWLUK    ;reset flag
D1CF 3F           SWI     PUPDAT   ;update the screen
D1D0 0E           FCB     *(3*256)+Q,TEN
D1D1 CC0304          LDD     RTS
D1D4 39
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMPLR - Common Player Routines

PAGE 59

; HSLOW: Player Damage Recovery

D1D5 4F HSLOW CLRA ;reset accumulator  
D1D6 5F CLR B  
D1D7 9321 SUBD PDAM ;recover 1/64th of damage  
D1D9 B00379 JSR ASRD6  
D1DC D321 ADDD PDAM ;-(pdimage/64) + pdimage  
D1DE 2E02 BGT HSLOW2 ;still damage left

D1E0 4F HSLOW1 CLRA ;can't have negative damage  
D1E1 5F CLR B  
D1E2 BD21 HSLOW2 STD PDAM ;update damage counter  
D1E4 3F SWI HUPDAT ;update the heart rate  
D1E5 0C FCB ;  
;

; Heartrate determines execution frequency

D1E6 96AF LDA HEARTB ;delay time  
D1E8 C602 LDB #0.JIF ;jiffy queue  
D1EA 39 RTS ;exit  
;

END

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
HUMAN - Human Interface Module

PAGE 60

INCLUD B:HUMAN.ASM

; HUMAN INTERFACE MODULE

; PLAYER: Keyboard Input/Autoplay Task

This task takes its input from the Keyboard  
 or from the AUTOPLAY tables. A character at  
 a time is fed to the subroutine HUMAN. This  
 task runs as often as possible (delay: 1 Jiffy).

D1EB 0D77	PLAYER	TST	AUTFLG	;autoplay?
D1ED 262C		BNE	, PLAY20	; yes
; Process Keyboard Buffer				
D1EF BDC329	PLAY10	JSR	KBDGET	;get the next char
D1F2 4B		TSTA		;check for nulls
D1F3 2753		BEQ	PLAY99	; NULL means we are done
D1F5 0D28		TST	FAINT	;did we faint?
D1F7 26F6		BNE	PLAY10	; yes - just eat chars
; Convert from ASCII to Internal Codes				
D1F9 8120		CMPA	#C,SP	;space?
D1FB 2718		BEQ	PLAY14	; yes
D1FD C61F		LDB	#I,CR	;assume carriage return
D1FF 810D		CMPA	#C,CR	
D201 270F		BEQ	PLAY12	; was a carriage return
D203 C624		LDB	#I,BS	;assume backspace
D205 8108		CMPA	#C,BS	
D207 2709		BEQ	PLAY12	; was a backspace
D209 5F		CLRB		
D20A 8141		CMPA	#'A'	;non-alpha becomes a space
D20C 2504		BLO	PLAY12	;uppercase-A
D20E 815A		CMPA	#'Z'	;non-alpha
D210 2303		BLS	PLAY14	;uppercase-Z
D212 1F98		PLAY12	TFR	;was alpha
D214 8C			FCB	
D215 841F		PLAY14	SKIP2	;replace original character
D217 8D33			ANDA	(use full 8-bits)
D219 2004			BSR	;use lower 5-bits
			HUMAN	;process the char
			BRA	;loop

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
HUMAN - Human Interface Module

PAGE 61

## ; Process Autoplay Commands

D21B 109E0D	PLAY20	LDY	AUTPTR	;load ptr to next command
D21E E6A0		LDB	,Y+	;load token count
D220 2A07		BPL	PLAY30	; commands still left
D222 3F		SWI		
D223 10		FCB	WAIT	;otherwise hang around for awhile
D224 3F		SWI		
D225 10		FCB	WAIT	
D226 7EC000		JMP	DEMO	;then restart the whole shebang!!!

## ; Feed next autoplay command to HUMAN

D229 AEA1	PLAY30	LDX	,Y++	;load next token string ptr
D22B CE0361		LDU	#OBJSTR+1	;buffer addr
D22E 3F		SWI		;expand the 5-bit string
D22F 06		FCB	EXPANO	
D230 3341		LEAU	I,U	;advance past class byte
D232 3F		SWI		;delay between words
D233 10		FCB	WAIT	
D234 8C		FCB	SKIP2	
D235 8D15	PLAY40	BSR	HUMAN	;process the char
D237 A6C0		LDA	,U+	;load next char
D239 2AFA		BPL	PLAY40	; valid chars goto HUMAN
D23B 4F		CLRA		
D23C 8D0E		BSR	HUMAN	;terminate tokens with spaces
D23E 5A		DECW		
D23F 26E8		BNE	PLAY30	;decrement token count
D241 861F	PLAY50	LDA	#I,CR	
D243 8D07		BSR	HUMAN	;terminate commands with return
D245 109F0D		STY	AUTPTR	
D248 CC0102	PLAY99	LDD	*(1*256)+Q,JIF	;update ptr
D24B 39		RTS		;once per jiffy

## HUMAN: Process Character Input/Command Dispatch

## Inputs:

A - Char to be processed

## Returns:

LINBUF - Char buffered

LINPTR - updated

## Notes:

This routine is responsible for maintaining the primary text display. It handles character echoing, erasing deleted characters, and prompting the player. As characters are received, they are buffered until a line terminator (carriage return) is seen. This routine will attempt to dispatch to the proper command handler based on the first TOKEN in the line buffer.

D24C 3476

HUMAN PSHS A,B,X,Y,U ;save regs

; Check if we are displaying the map

D24E 0DAB

HMAN10	TST	HEARTF	;are we in map mode?
D250 2604	BNE	HMAN20	;nope
D252 3F	SWI		;forward-looking display
D253 19	FCB	INIVU	
D254 3F	SWI		;fake the prompt
D255 OF	FCB	PROMPT	

; Process Characters

D256 DE11

HMAN20	LDD	LINPTR	;load line buffer ptr
D258 811F	CMPA	#I.CR	;check for line terminator
D25A 2713	BEQ	HMAN30	
D25C 8124	CMPA	#I.BS	;check for character delete
D25E 271D	BEQ	HMAN40	

; Buffer Normal Chars

D260 3F

D261 04	SWI		;echo char to the screen
D262 A7C0	FCB	OUTCHR	
D264 8EC67C	STA	UT	;buffer char and advance ptr
D267 3F	LDX	#M\$CURS	;print the cursor
D268 03	SWI		
D269 11830311	FCB	OUTSTR	
D26B 2645	CMPU	#LINEND	;insure against buffer overruns
	BNE	HMAN99	;buffer still has room

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
HUMAN - Human Interface Module

PAGE 63

## ; Process Carriage Returns

D26F 4F	HMAN30	CLRA	;erase old cursor
D270 3F		SWI	;
D271 04		FCB	OUTCHR
D272 DC03		LDD	NEGONE
D274 EDC1		STD	,U++
D276 CE02F1		LDU	#LINBUF
D279 DF11		STU	LINPTR
D27B 2015		BRA	HMAN50

## ; Process Character Delete

D27D 118302F1	HMAN40	CMPU	#LINBUF	;check buffer boundary
D281 2731		BEQ	HMAN99	;
D283 335F		LEAU	-1,U	;decrement ptr
D285 BED28C		LDX	#M\$ERAS	;print <BS><SP><BS>
D288 3F		SWI	;	;
D289 03		FCB	OUTSTR	;
D28A 2028		BRA	HMAN99	;loop for more
D28C 0024241C24	M\$ERAS	FCB	I,SP,I,BS,I,BS,I,BAR,I,BS,\$FF	

## ; Dispatch to Proper Routine

D292 BED894	HMAN50	LDX	#CMDTAB	;primary commands
D295 BDCBEC		JSR	PARSER	;get the next token
D298 270D		BEQ	HMAN70	;null token
D29A 2A05		BPL	HMAN60	;legal command
D29C BDCRE1		JSR	CMDERR	;indicate error
D29F 2006		BRA	HMAN70	;flush rest of line
D2A1 48	HMAN60	LSLA	;	;
D2A2 BED9D0		LDX	#DISPAT	;code#2 for offset
D2A5 AD96		JSR	[A,X]	;dispatch table base addr
				;dispatch to correct routine
D2A7 CE02F1	HMAN70	LDU	#LINBUF	;reset line buffer ptr
D2AA 0DAD		TST	HEARTF	;are we in map mode?
D2AC 2706		BEQ	HMAN99	;yes - no prompt
D2AE 0D28		TST	FAINT	;are we unconscious?
D2B0 2602		BNE	HMAN99	;yes - no prompt
D2B2 3F		SWI	;	;
D2B3 0F		FCB	PROMPT	;prompt the human
D2B4 DF11	HMAN99	STU	LINPTR	;update the line ptr
D2B6 35F6		PULS	A,B,X,Y,U,PC	;restore regs/exit
		END		

```

INCLUD B:PATTK.ASM
!!!!!!ATTACK COMMAND!!!!!!
PATTK: Player attack routine
Command Syntax:
    ATTACK LEFT           ;attack with object in left hand
    ATTACK RIGHT          ;attack with object in right hand
D2B8 BDCC31  PATTK   JSR      PARHND
D2BB EEC4     LDU      ,U        ;select a hand
D2BD 2603     BNE      PATT10
D2BF CE0B07   LDU      #EMPHND ;load held object
                           ;something in hand
                           ;empty hand

Set Up ATTACK Parameter Blocks
D2C2 1F32  PATT10  TFR      U,Y      ;make copy of OCB ptr
D2C4 A64C  LDA      P,OCHMGO,U ;copy offensive parms to PLRBLK
D2C6 9719  STA      PMGO
D2C8 A64D  LDA      P,OCPHO,U
D2CA 971B  STA      PPPO      ;;
                           ;;

Calculate amount of energy expended
D2CC 9B19  ADDA    PMGO      ;physical + magic offense
D2CE 46    RORA    ;1/8th of total offensive pct
D2CF 44    LSRA
D2D0 44    LSRA
D2D1 9E17  LDX      PPOW      ;load current power rating
D2D3 BDD436  JSR      SCAL16
D2D6 D321  ADDB    PDAM      ;increase damage
D2D8 DD21  STD      PDAM      ;;

Attack Sound Effect Goes Here
D2DA A64A  LDA      P,OCCLS,U ;make object sound
D2DC 8B0C  ADDA    #SNDOBJ
D2DE C6FF  LDR    #$FF      ;loud
D2E0 3F    SWI
D2E1 1C    FCB    SOUNDS    ;;

Decrement Ring Usage Counters
D2E2 A649  LDA      P,OCTYP,U ;player using incontrollable ring?
D2E4 8113  CMPA    #T,RN11
D2E6 2D0F  BLT     PATT20    ;loser
D2E8 8115  CMPA    #T,RN12
D2EA 2E0B  BGT     PATT20    ;loser

D2EC 6A46  DEC      P,OCXXX,U ;decrement usage counter
D2EE 2607  BNE     PATT20    ;charges still left
D2F0 8616  LDA      #T,RN20
D2F2 A749  STA      P,OCTYP,U ;gold ring (useless)
D2F4 BDD638  JSR      PREU000 ;change used rings into gold

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 65

; Is there something to attack?

```
D2F7 BC13    PATT20 LDD PROW      ;is there a creature here?
D2F9 BDCFB2    JSR CFIND
D2FC 2777    BEQ PATT99      ;  nope
```

; Attempt to hit creature

```
D2FE CE0217    LDU #PLRBLK   ;player is attacker
D301 1E13    EXG X,U       ;creature is the defender
D303 A62A    LDA P.OCCLS,Y  ;check for a ring
D305 8101    CMPA #K,RING
D307 2716    BEQ PATT24    ;  rings are guaranteed to hit
D309 BDD3D7    JSR ATTACK   ;swing at the creature!
D30C 2B67    BMI PATT99    ;  big miss
```

; Potentially hit creature - how dark is it?

```
D30E 109E24    LDY PTORCH   ;using a torch?
D311 2706    BEQ PATT22   ;  nope!
D313 A629    LDA P.OCTYP,Y ;what kind of torch...
D315 8118    CMPA #T,TORS
D317 2606    BNE PATT24   ;dead?
D319 3F        SWI         ;only 25% of normal percentage
D31A 07        FCB         ;RANDOM
D31B 8403    ANDA #3       ;  (use the two LSBits)
D31D 2656    FNE PATT99   ;  sorry - darkness prevails
```

; We hit the creature!

```
PATT24
D31F 3F        SWI
D320 18        FCB ISOUND
D321 12        FCB A$KLK2  ;KLINK!!!
D322 3F        SWI
D323 02        FCB OUTSTI
D324 16        FCB Z00010110
D325 F7        FCB Z11110111
D326 B0        FCB Z10110000
D327 BDD40C    JSR DAMAGE
D32A 2249    BHI PATT99  ;take a chunk out of him
                           ;  still alive
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 66

```

; DEAD Creature! Drop all objects to the floor
D32C 3048    PATT30 LEAX    P.CCOBJ,U      ;load initial object ptr
D32E AE84    PATT32 LDX     P.OCPTR,X      ;advance to next object
D330 2708    BEQ     PATT40      ;done
D332 6F05    CLR     P.OCOWN,X      ;mark object unowned
D334 EC4F    LDD     P.CCROW,U      ;drop objects where creature stands
D336 ED02    STD     P.OCROW,X      ;mark new position
D338 20F4    BRA     PATT32      ;do next object

; Decrement proper entry in CMXLND
D33A 9E82    PATT40 LDX     CMXPTR      ;creature matrix pointer
D33C F64D    LDB     P.CCTYP,U      ;load creature type
D33E 6A85    DEC     B,X          ;decrement table entry
D340 6F4C    CLR     P.CCUSE,U      ;mark creature as dead
D342 3F      SWI     PUPDAT      ;update the screen
D343 0E      FCB     PUPDAT      ;
D344 3F      SOUND$ A$EXPO      ;loud explosion!
D345 18      SWI     PUPDAT      ;
D346 15      FCB     ISOUND       ;;
D347 ECC4    LDD     P.CCPOW,U      ;increase PPOW by 1/8th of creature
D349 8D34    BSR     ASRD3       ;add
D34B D317    ADDD    PPOW        ;add to our power level
D34D 2A02    BPL     PATT42      ;power within limits
D34F 867F    LDA     #$7F        ;peg power at approximately at 32K
D351 DD17    PATT42 STD     PPOW        ;update player power

; Check for Fake Wizard (Plain)
D353 A64D    LDA     P.CCTYP,U      ;plain wizard?
D355 810A    CMPA    #10        ;;
D357 272D    BEQ     ENDGAM      ;nope

; Check for Evil Wizard (Crescent)
D359 810B    CMPA    #11        ;evil wizard?
D35B 2618    BNE     PATT99      ;nope

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 68

; ASRDn: 16-bit Arithmetic Shift Right

D377 47	ASRD7 ASRA
D378 56	RORB
D379 47	ASRD6 ASRA
D37A 56	RORB
D37B 47	ASRD5 ASRA
D37C 56	RORB
D37D 47	ASRD4 ASRA
D37E 56	RORB
D37F 47	ASRD3 ASRA
D380 56	RORB
D381 47	ASRD2 ASRA
D382 56	RORB
D383 47	ASRD1 ASRA
D384 56	RORB
D385 39	RTS

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 67

| Perform Special Ring Riddle Code

D35D 0A2B	DEC	FRZFLG	;stop all creature activity
D35F CC0713	LDD	#\$0713	;set lighting levels
D362 DD26	STD	PRLITE	
D364 8E0B23	LDX	\$OCBLND+OC.LEN	;single object (OMEGA ring)
D367 9FOF	STX	OCEPTR	
D369 DC00	LDD	BIGZER	;zero all ptrs
D36B DD29	STD	BAGPTR	;nothing in bag
D36D DD24	STD	PTORCH	;no torch
D36F DD1F	STD	PRHAND	;nothing in hands
D371 DD1D	STD	PLHAND	
D373 3F	SWI		;update the screen
D374 19	FCR	INIVU	
D375 3F	PATT99 SWI	HUPDAT	;update the heartrate
D376 0C	FCB		

## ; Perform End Game Transition

```

D386 8EDF10 ENDGAM LDX #WIZ1      ; fade in crescent wizzard
D389 3F SWI                   ; (clear the status line)
D38A 13 FCB WIZIN             ; .

D38B 3F SWI                   ; a message from our sponsor
D38C 02 FCB OUTSTI            ; .
D38D FF FCB Z11111111          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D38E C0 FCB Z11000000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D38F 57 FCB Z01010111          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D390 3E FCB Z00111110          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D391 A7 FCB Z10100111          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D392 46 FCB Z01000110          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D393 C0 FCB Z11000000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D394 90 FCB Z10010000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D395 51 FCB Z01010001          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D396 32 FCB Z00110010          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D397 28 FCB Z00101000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D398 1E FCB Z00011110          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D399 60 FCB Z01100000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D39A 51 FCB Z01010001          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D39B 09 FCB Z00001001          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D39C 98 FCB Z10011000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D39D 20 FCB Z00100000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D39E C0 FCB Z11000000          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D39F E7 FCB Z11100111          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D3A0 DE FCB Z11011110          ; ^ ENOUGH! I TIRE OF THIS PLAY...
D3A1 F0 FCB Z11110000          ; ^ ENOUGH! I TIRE OF THIS PLAY...

D3A2 3F SWI                   ; more commercials
D3A3 02 FCB OUTSTI            ; .
D3A4 E8 FCB Z11101000          ; PREPARE TO MEET THY DOOM
D3A5 00 FCB Z00000000          ; PREPARE TO MEET THY DOOM
D3A6 08 FCB Z00001000          ; PREPARE TO MEET THY DOOM
D3A7 48 FCB Z01001000          ; PREPARE TO MEET THY DOOM
D3A8 B0 FCB Z10110000          ; PREPARE TO MEET THY DOOM
D3A9 OC FCB Z00001100          ; PREPARE TO MEET THY DOOM
D3AA 8A FCB Z10001010          ; PREPARE TO MEET THY DOOM
D3AB 0A FCB Z00001010          ; PREPARE TO MEET THY DOOM
D3AC 3C FCB Z00111100          ; PREPARE TO MEET THY DOOM
D3AD 0D FCB Z00001101          ; PREPARE TO MEET THY DOOM
D3AE 29 FCB Z00101001          ; PREPARE TO MEET THY DOOM
D3AF 68 FCB Z01101000          ; PREPARE TO MEET THY DOOM
D3B0 0A FCB Z00001010          ; PREPARE TO MEET THY DOOM
D3B1 23 FCB Z00100011          ; PREPARE TO MEET THY DOOM
D3B2 20 FCB Z00100000          ; PREPARE TO MEET THY DOOM
D3B3 23 FCB Z00100011          ; PREPARE TO MEET THY DOOM
D3B4 DE FCB Z11011110          ; PREPARE TO MEET THY DOOM
D3B5 DD FCB Z11011101          ; PREPARE TO MEET THY DOOM
D3B6 EF FCB Z11101111          ; PREPARE TO MEET THY DOOM
D3B7 60 FCB Z01100000          ; PREPARE TO MEET THY DOOM
D3B8 3F SWI                   ; take your time...
D3B9 10 FCB WAIT              ;

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 70

```
; Strip player of everything/save PLHAND, PRHAND, FTORCH
D3BA DE24      LDU    PTORCH      ;torch is the only object left in bag
D3BC DF29      STU    BAGPTR
D3BE 2704      BEQ    EGAM10     ;no torch even!
D3C0 4F        CLR.A   CLRA       ;terminate object list
D3C1 5F        CLR.B   CLR.B
D3C2 E1C4      STD    P,OCPTR,U  ;
D3C4 CC00C8      EGAM10    LDD    $200    ;set penalty weight
D3C7 DD15      STD    POBJWT
;
; Create Endgame Upper Level
D3C9 B603      LDA    #3          ;endgame upper level
D3CB 3F        SWI
D3CC 1A        FCB    NEWLVL
D3CD BDCF97      JSR    FNICEL    ;a new position for the player
D3D0 DD13      STD    PROW
;
D3D2 3F        SWI
D3D3 15        FCB    WIZOUT    ;now fade the wizzard out
D3D4 3F        SWI
D3D5 19        FCB    INIVU     ;display initial screen
D3D6 39        RTS
;
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 71

; ATTACK: Combat Resolution Routine

Inputs:

X - Attacker Attack Block

U - Defender Attack Block

Returns:

N - condition code set if attack fails

D3D7 3456	ATTACK	PSHS	A,B,X,U	; save regs
D3D9 860F		LDA	#15	; set percentage index
D3DB 97C1		STA	T0	,

; Calculate Defender/Attacker Percentage Index (steps of 25%)

D3D0 ECC4		LDD	P.ATPOW,U	;defender power
D3DF A34A		SUBD	P.ATDAM,U	;minus defender damage
D3E1 BDCA12		JSR	LSLD2	;times 4
D3E4 A384	ATTK10	SUBD	P.ATPOW,X	;multiple subtractions
D3E6 2504		BCS	ATTK20	finally minus
D3E8 0AC1		INC	T0	decrement index
D3EA 26F8		BNE	ATTK10	, loop until zero

; Use Index to Calculate Reward/Penalty Percentage

D3EC B6C1	ATTK20	LDB	T0	;load percentage index
D3EE C003		SUBB	#3	;offset index
D3F0 2A09		BPL	ATTK22	; reward anything above 100%
D3F2 50		NEGB		
D3F3 8619		LDA	#25	penalize anything below 75%
D3F5 30		MUL		,
D3F6 BDCA99		JSR	NEGD	,
D3F9 2003		BRA	ATTK30	,
D3FB 860A	ATTK22	LDA	#10	;reap the rewards
D3FD 30		MUL		,
D3FE EDE3	ATTK30	STD	--S	;stack reward/penalty
D400 3F		SWI		;create 16-bit random number (0:255)
D401 07		FCB	RANDOM	,
D402 1F89		TFR	A,B	,
D404 4F		CLRA		,
D405 E3E1		ADDI	,S++	calculate weighted percentage
D407 83007F		SUBD	#127	;did we hit him?
D40A 35D6	ATTK99	PULS	A,B,X,U,PC	;restore regs/exit

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 72

; DAMAGE: Process Damage Done

Inputs:

X = Attacker ATB ptr

U = Defender ATB ptr

Returns:

N = condition code set if defender died

D40C 3476  
D40E 1F12

DAMAGE PSHS A,B,X,Y,U ;save regs  
TFR X,Y ;copy attacker ATB ptr

; Calculate Magical Damage

D410 AEAA  
D412 A622  
D414 8D20  
D416 1F01  
D418 A643  
D41A 8D1A  
D41C E34A  
D41E ED4A

LDX P.ATPOW,Y ;channel power thru magic offense pct  
LDA P.ATMGO,Y  
BSR SCAL16  
TFR D,X ;filter damage thru magic defense pct  
LDA P.ATMGD,U  
BSR SCAL16  
ADDU P.ATDAM,U ;accumulate defender damage  
STD P.ATDAM,U

; Calculate Magical Damage

D420 AEAA  
D422 A624  
D424 8D10  
D426 1F01  
D428 A645  
D42A 8D0A  
D42C E34A  
D42E ED4A

LDX P.ATPOW,Y ;channel power thru physical offense pct  
LDA P.ATPHO,Y  
BSR SCAL16  
TFR D,X ;filter damage thru physical defense pct  
LDA P.ATPHD,U  
BSR SCAL16  
ADDU P.ATDAM,U ;accumulate defender damage  
STD P.ATDAM,U

D430 AEC4  
D432 AC4A  
D434 35F6

LDX P.ATPOW,U ;compare power vs. damage  
CMPX P.ATDAM,U  
PULS A,B,X,Y,U,PC ;restore regs/exit

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PATTK - Attack Command

PAGE 73

SCAL16: Multiply 16-bit number by RADIX-7 value

Inputs:

X - 16-bit unsigned number  
A - RADIX-7 value

Returns:

D - 16-bit unsigned result

D436 3416	SCAL16	PSHS A,B,X	; save regs
D438 0FC1		CLR T0	; reset MSByte of result
D43A E663		LDB 3,S	; LSBByte of 16-bit number
D43C 3D		MUL	; multiply by RADIX-7 value
D43D BDC2		STD T1	; T1:carry, T2:LSByte of result
D43E ;			
D43F A6E4		LDA 0,S	; load RADIX-7 value
D441 E662		LDB 2,S	; load MSByte of 16-bit number
D443 3D		MUL	; multiply
D444 B3C1		ADDD T0	; add carry portion
D446 0BC3			
D448 59		LSL T2	; MSBit of LSBByte into carry
D449 49		ROLB	; shift carry into final result
D44A EDE4		ROLA	
D44C 3596		STD 0,S	; return as parm
		PULS A,B,X,PC	; restore regs/exit
		END	

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PCLIMB - Climb Command

PAGE 74

```

INCLUDE B:PCLIMB.ASM
!!!!!! CLIMB COMMAND !!!!!

PCLIMB: Climb Up/Down Ladders

Command Syntax:
    CLIMB           ;climb up a ladder
    CLIMB UP        ;
    CLIMB DOWN      ;climb down

D44E DC13
D450 BDCFE1
D453 2B1A
D455 97C1
;
;          PCLIMB  LDD    PROW
;          JSR    VFIND
;          BMT    PCLI00
;          STA    TO
;          ;our position
;          ;what kind of vertical feature?
;          ;no features
;          ;save vertical features on stack

D457 8ED8D9
D45A BDCBEC
D45D 2F10
;
;          LDX    #DIRTAB
;          JSR    PARSER
;          BLE    PCLI00
;          ;directions
;          ;get the next token
;          ;null/illegal token

D45F D6C1
D461 8104
D463 2700
D465 8105
D467 2606
;
;          LDB    TO
;          CMPA   #T,UP
;          BEQ    PCLI10
;          CMPA   #T,DN
;          BNE    PCLI00
;          ;retrieve vertical feature code
;          ;climb up
;          ;climb down
;          ;.

Go Down a level
;
;          LDA    #1
;          BITB   #BIT1
;          BNE    PCLI20
;          ;assume we are going down
;          ;downward vertical feature?
;          ;yes
;          JMP    CMDERR
;          ;can't go down!

Go Up a Level
;
;          PCLI10 LDA    #-1
;          CMPB   #VF,LUP
;          BNE    PCLI00
;          ;assume we are going up
;          ;ladder up?
;          ;nope - no way up

Process Level Change
;
;          PCLI20 SWI    PREPAR
;          FCB    LEVEL
;          ADDA   SWI
;          NEWLVL
;          SWI
;          INIVU
;          RTS
;          ;display prepare message
;          ;change the level
;          ;create a new dungeon level
;          ;print current screen
;          ;bye

END
;
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PEXAM - Examine Command

PAGE 75

INCLUD B:PEXAM.ASM

!!!!!!  
EXAMINE COMMAND  
!!!!!!

PEXAM: Examine our Backpack and the Floor

Command Syntax:  
EXAMINE

D481	BED495	PEXAM	LDX	#EXAMIN	set the examine display mode
D484	9FB2		STX	DSPMOD	
D486	3F		SWI		update the display
D487	OE		FCB	PUFDAT	
D488	39		RTS		,bye

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PEXAM - Examine Command

PAGE 76

## ; EXAMIN: "EXAMINE" Command Display Generator

D489 3F	EXAMIO	SWI	;clear alternate screen
D48A 09		FCB	ZFLOP
D48B AEC4		LDX	P.VDBAS,U
D48D CE0380		LDU	#TXTEXA
D490 AFC4		STX	P.TXBAS,U
D492 0AB7		DEC	TXBFLG
D494 39		RTS	;bye
D495 8DF2	EXAMIN	BSR	EXAMIO
D497 0FB6		CLR	NEWLIN

; prepare for i/o to EXAMINE screen  
; indicate we are on a new line

## ; Print Header

D499 CC000A		LDD	#10	;set cursor position
D49C ED44		STP	P.TXCUR,U	;header
D49E 3F		SWI		
D49F 02		FCB	OUTSTI	
D4A0 62		FCB	Z01100010	;IN THIS ROOM^
D4A1 6C		FCB	Z01011100	;IN THIS ROOM^
D4A2 0A		FCB	Z00001010	;IN THIS ROOM^
D4A3 21		FCB	Z00100001	;IN THIS ROOM^
D4A4 33		FCB	Z00110011	;IN THIS ROOM^
D4A5 04		FCB	Z00000100	;IN THIS ROOM^
D4A6 9E		FCB	Z10011110	;IN THIS ROOM^
D4A7 F6		FCB	Z11110110	;IN THIS ROOM^
D4A8 FC		FCB	Z11111100	;IN THIS ROOM^

## ; Check for Creatures

D4A9 DC13		LDD	PROW	;check for a creature
D4AB BDCAF82		JSR	CFIND	
D4AE 2710		BEQ	EXAMIO	;nothing here
D4B0 AE44		LDX	P.TXCUR,U	;position cursor
D4B2 300B		LEAX	11,X	
D4B4 AF44		STX	P.TXCUR,U	;indicate bad guy present
D4B6 3F		SWI		
D4B7 02		FCB	OUTSTI	
D4B8 56		FCB	Z01010110	;CREATURE ^A
D4B9 C7		FCB	Z11000111	;CREATURE ^A
D4BA 22		FCB	Z00100010	;CREATURE ^A
D4BB 86		FCB	Z10000110	;CREATURE ^A
D4BC 95		FCB	Z10010101	;CREATURE ^A
D4BD 91		FCB	Z10010001	;CREATURE ^A
D4BE 77		FCB	Z01110111	;CREATURE ^A
D4BF F0		FCB	Z11110000	;CREATURE ^A

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PEXAM - Examine Command

PAGE 77

; Check for Objects on the Floor

D4C0 0F91	EXAM10	CLR	OFINDF	;start at beginning
D4C2 DC13	EXAM12	LDD	PROW	;find an object
D4C4 BDCF53		JSR	OFIND	
D4C7 2704		BEQ	EXAM20	; no more objects
D4C9 8D3A		BSR	PRTOBJ	;print object name/new line
D4CB 20F5		BRA	EXAM12	;loop

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PEXAM - Examine Command

PAGE 78

```

; Draw Separation Line
D4CD 0BB6    EXAM20  TST     NEWLIN      ;check new line flag
D4CF 2702    BEQ     EXAM24      ;
D4D1 8B2B    BSR     PCRLF      ;force new line
D4D3 CC1B20   EXAM24  LDD     #(I,EXCL*256)+32 ;I,EXCL,,32
D4D6 3F       EXAM26  SWI     ;
D4D7 04       FCB     OUTCHR     ;
D4D8 5A       DECB    ;
D4D9 26FB    BNE     EXAM26      ;

; Print Header
D4DB AE44    LDX     P,TXCUR,U   ;position cursor
D4DD 300C    LEAX    12,X      ;
D4DF AF44    STX     P,TXCUR,U   ;
D4E1 3F       SWI     ;
D4E2 02       FCB     OUTSTI     ;header
D4E3 40       FCB     Z01000000  ;BACKPACK^
D4E4 82       FCB     Z10000010  ;BACKPACK^
D4E5 35       FCB     Z00110101  ;BACKPACK^
D4E6 C0       FCB     Z11000000  ;BACKPACK^
D4E7 23       FCB     Z00100011  ;BACKPACK^
D4E8 5F       FCB     Z01011111  ;BACKPACK^
D4E9 C0       FCB     Z11000000  ;BACKPACK^

; Display Bag Contents
D4EA BE0229   EXAM30  LDX     #BAGPTR  ;initial ptr
D4EB AE84    LDX     P,OCPTR,X   ;advance to next object
D4EF 270A    BEQ     EXAM99      ;exhausted list

; Check for the TORCH
D4F1 9C24    CMPX    PTORCH     ;lighted torch?
D4F3 2602    BNE     EXAM32      ;nope
D4F5 6346    COM     P,TXINV,U  ;highlight burning torch
D4F7 8DOC    EXAM32  BSR     PRTOBJ   ;print object name/new line
D4F9 20F2    BRA     EXAM30      ;round and round
D4FB 0FB7    EXAM99  CLR     TXBFLG    ;restore standard i/o
D4FD 39       RTS

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PEXAM - Examine Command

PAGE 79

```

; PCRLF: Print Carriage Return/Linefeed
D4FE 861F    PCRLF   LDA     #I.CR      ;print carriage return
D500 3F       SWI
D501 04       FCB     DUTCHR
D502 0FB6     CLR     NEWLIN    ;reset flag
D504 39       RTS

; PTOBJ: Print the Object Name in Regular Mode
D505 3416    PTOBJ   PSHS    A,B,X    ;save regs
D507 BDC617   JSR     OBJNAM   ;determine object name
D50A 3F       SWI
D50B 03       FCB     OUTSTR   ;print object name
;
D50C 962C    LDA     VGINVU   ;insure highlighting is off
D50E A746    STA     P.TXINV,U
;

; Determine TAB/CRLF
D510 03B6    COM     NEWLIN   ;flip flag
D512 270A    BEQ     POBJ101  ;perform new line function
;
D514 EC44    LDD     P.TXCUR,U ;load cursor position
D516 C30010   ADDB    #16      ;adjust cursor
D519 C4F0    ANDB    #$FO
D51B ED44    STD     P.TXCUR,U ;update cursor
D51D 8C       FCB     SKIP2
;
D51E 8DDE    POBJ101  BSR     PCRLF   ;output CRLF
D520 3596    POBJ990  PULS    A,B,X,PC ;restore regs/exit
;

END

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PGET - Get Command

PAGE 80

```

INCLUDE B:PGET.ASM
!!!!!! GET COMMAND !!!!!!
NAM PGET
LIBRY CD
XDEF PGET,PDROP,PSTOW,PSTOWO,PPULL
XREF PARHND,PAROBJ,CMDERR,OFIND,OBJWGT
PSCT

```

PGET: Grab an object with either hand

Command Syntax:

```

GET LEFT <generic>
GET LEFT <specific>
GET RIGHT <generic>
GET RIGHT <specific>

```

D522 8D52	PGET	BSR	HNDPAR	;determine left/right hand
D524 2640		BNE	ERRCMD1	; something already in hand!
; Search the cell for the object				
D526 BDCBBA	PGET10	JSR	PAROBJ	;determine generic/specific object
D529 0F91		CLR	OFINDF	;start search from the beginning
D52B DC13	PGET20	LDD	PROW	;position
D52D BDCF53		JSR	OFIND	;get the next object
D530 2741		BEQ	ERRCMD1	; can't find the object
D532 0D90		TST	SPEFLG	;generic/specific?
D534 2606		BNE	PGET22	; specific
D536 A60A		LDA	P.OCCLS,X	;perform generic compare
D538 918F		CMPA	OBJCLS	;
D53A 2004		BRA	PGET24	;
D53C A609	PGET22	LDA	P.OCTYP,X	;perform specific compare
D53E 918E		CMPA	OBJTYP	;
D540 26E9	PGET24	BNE	PGET20	; loop if failure
; Place the object into the correct hand				
D542 AFC4	PGET30	STX	,U	;store object ptr into correct hand
D544 6C05		INC	P.OCOWN,X	;mark object as owned
D546 E60A		LDB	P.OCCLS,X	;accumulate weight
D548 8ED9FA		LDX	#OBJWGT	;
D54B E685		LDR	B,X	;
D54D 4F		CLRA		16-bit value/fall into WUPDAT
D54E 201B		BRA	WUPDAT	;

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PGET - Get Command

PAGE 81

```

!!!!!! D R O P C O M M A N D !!!!!

PDROP: Drop an object

Command Syntax:
DROP LEFT           ;drop object in left hand
DROP RIGHT          ;drop object in right hand

D550 8D24    PDROP   BSR    HNDPAR
D552 271F    BEQ    ERRCMD1
;               ;left/right hand
;               ;empty hand

D554 4F      CLRA
D555 5F      CLRB
D556 EDC4    STD    ,U
D558 6F05    CLR    P.OCOWN,X
D55A DC13    LDD    PROW
D55C E002    STD    P.OCROW,X
D55E 9681    LDA    LEVEL
D560 A704    STA    P.OCLVL,X
;               ;mark hand empty
;               ;
;               ;mark object unowned
;               ;load our position
;               ;store as object position
;               ;store the current level

Decrease Player Weight

D562 E60A    LDB    P.OCCCLS,X
D564 8ED9FA  LDX    #OBJWGT
D567 E685    LDR    B,X
D569 50      NEGB
D56A 10      SEX
;               ;load generic class
;               ;table base addr
;               ;load weight value
;               ;negate weight (16-bit value)

WUPDAT: Common Player Weight Modification Routine

Used by PGET,PDROP

D56B D315    WUPDAT ADDD  POBJWT
D56D DD15    STD   POBJWT
D56F 3F      SWI
D570 OC      FCB   HUPDAT
D571 2044    BRA   COMUPD
;               ;effect relative change
;               ;update player weight
;               ;update the heartrate
;               ;update status line/display

Local Jumps

D573 7ECB81  ERRCMD1 JMP   CMDERR
D576 7ECC31  HNDPAR JMP   PARHND
;               ;flag errors
;               ;dispatch to hand parse routine

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PGET - Get Command

PAGE 82

!!!!!!  
S T O W C O M M A N D  
!!!!!!

PSTOW: Put an object into our bag

Command Syntax:

STOW LEFT  
STOW RIGHT

D579 8DFB	PSTOW	BSR	HNDPAR	; determine left/right hand
D57B 27F6		BEQ	ERRCMD1	; empty hand!
D57D DC29	PSTOWO	LDD	BAGPTR	; insert object into bag list
D57F ED84		STD	P_OOPTR,X	; ;
D581 9F29		STX	BAGPTR	; ;
D583 4F		CLRA		; zap hand pointer
D584 5F		CLRB		; ;
D585 EDC4		STB	U	; ;
D587 202E		BRA	COMUPD	; update status line/display
				;

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PGET - Get Command

PAGE 83

```

!!!!!! PULL COMMAND !!!!!

;PPULL: Take an object from our bag

Command Syntax:
    FULL LEFT <generic>
    FULL LEFT <specific>
    FULL RIGHT <generic>
    FULL RIGHT <specific>

D589 8DEB    PPULL   BSR     HNDPAR      ;determine left/right hand
D58B 26E6    BNE     ERRCMD1    ;non-empty hand
D58D BUCBBA  JSR     PAROBJ     ;determine generic/specific name

; Attempt to find object in BAG

D590 BE0229  PULL10 LDX     #BAGPTR    ;starting point of the loop
D593 1F12    TFR     X,Y      ;save current link ptr
D595 AE84    LDX     ,X       ;advance link ptr
D597 27DA    BEQ     ERRCMD1    ;search failure
D599 0D90    TST     SPEFLG    ;generic/specific?
D59B 2606    BNE     PULL12    ;specific

D59D A60A    LDA     P,OCCCLS,X ;perform generic compare
D59F 918F    CMPA    OBJCLS    ;
D601 2004    BRA     PULL14    ;:

D5A3 A609    PULL12 LDA     P,OCTYP,X ;perform specific compare
D5A5 918E    CMPA    OBJTYP    ;
D5A7 26EA    BNE     PULL10    ;loop if failure

; Place the Object into the Correct Hand

D5A9 EC84    LDD     P,OCPTR,X ;remove object from bag list
D5AB EDA4    STD     ,Y       ;store object into correct hand
D5AD AFC4    STX     ,U       ;

D5AF 4F      CLRA    CLRAB    ;clear ACCA, ACCB
D5B0 5F      CLRB    CLRB    ;
D5B1 9C24    CMPX    PTORCH   ;using this object as torch?
D5B3 2602    BNE     COMUPD   ;nope
D5B5 DD24    STD     PTORCH   ;no longer using this torch

D5B7 3F      COMUPD  SWI     ;update status line
D5B8 0D      FCB     STATUS   ;
D5B9 3F      SWI     ;update main display
D5B0 0E      FCB     FUPIAT  ;bye
D5B2 39      RTS     ;END

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PINCAN - Incant Command

PAGE 84

```

;-----+
;      INCLUD B:PINCAN.ASM
;-----+
;      I N C A N T   C O M M A N D
;-----+



; PINCAN: Activate a magic object via incantation

D5BC 8E08F3 PINCAN LDX    #ADJTAB      ;find a magic word?
D5BF BDCBEC     JSR    PARSER
D5C2 2F2B       BLE    PINC99      ;full/illegal token
D5C4 0D7B       TST    FULFLG      ;spell out the entire word?
D5C6 2727       BEQ    PINC99      ;nope
D5C8 DD8E       STD    OBJTYP      ;store parms
;
D5CA DE1D       LDU    PLHAND      ;process left hand object
D5CC 8D02       BSR    PINC10      ;process right hand object
D5CE DE1F       LDU    PRHAND      ;process right hand object

; Are we holding a ring?

D5D0 271D       BEQ    PINC99      ;empty hand!
D5D2 A64A       LDA    P,OCCLS,U   ;do we have a ring?
D5D4 8101       CMPA   #K.RING    ;nope
D5D6 2617       BNE    PINC99      ;nope

; Did he say the secret word? (and win $50...)
;
D5D8 A647       LDA    P,OCXXX+1,U ;load special object type
D5DA 2713       BEQ    PINC99      ;must have already been INCANTed
D5DC 918E       CMPA   OBJTYP     ;does it match the token code?
D5DE 260F       BNE    PINC99      ;nope

; Transform the object...
;
D5E0 A749       STA    P,OCTYP,U   ;set the new object type
D5E2 3F          SWI    ,          ;fill in new parameters
D5E3 18          FCB    OCBFIL     ;
;
D5E4 3F          SWI    ,          ;make the ring sound
D5E5 1B          FCB    ISOUND     ;
D5E6 0D          FCB    A$RING     ;
D5E7 3F          SWI    ,          ;show the new ring
D5E8 0D          FCB    STATUS     ;
D5E9 6F47       CLR    P,OCXXX+1,U ;insure he can't do this again

; Terminate the game?
;
D5EB 8112       CMPA   #T,RN15    ;is this the OMEGA ring?
D5ED 2701       BEQ    WINNER    ;yes - that's all folks!!!
;
D5EF 39          PINC99 RTS      ;bye
;
```

```
; WINNER: Tell Player the Good News...
```

D5F0 8EDF39	WINNER	LDX	#WIZZ	; star wizzard
D5F3 0A9E		DEC	FADEFLG	; fade in
D5F5 3F		SWI		(clear the status line)
D5F6 13		FCB	WIZIN	
D5F7 3F		SWI		; we have a winner
D5F8 02		FCB	OUTSTI	
D5F9 FF		FCB	Z11111111	^BEHOLD! DESTINY AWAITS THE HAND
D5FA C4		FCB	Z11000100	^BEHOLD! DESTINY AWAITS THE HAND
D5FB 54		FCB	Z01010100	^BEHOLD! DESTINY AWAITS THE HAND
D5FC 3D		FCB	Z00111101	^BEHOLD! DESTINY AWAITS THE HAND
D5FD 84		FCB	Z10000100	^BEHOLD! DESTINY AWAITS THE HAND
D5FE D8		FCB	Z11011000	^BEHOLD! DESTINY AWAITS THE HAND
D5FF 08		FCB	Z00001000	^BEHOLD! DESTINY AWAITS THE HAND
D600 59		FCB	Z01011001	^BEHOLD! DESTINY AWAITS THE HAND
D601 D1		FCB	Z11010001	^BEHOLD! DESTINY AWAITS THE HAND
D602 2E		FCB	Z00101110	^BEHOLD! DESTINY AWAITS THE HAND
D603 C8		FCB	Z11001000	^BEHOLD! DESTINY AWAITS THE HAND
D604 03		FCB	Z00000011	^BEHOLD! DESTINY AWAITS THE HAND
D605 70		FCB	Z01110000	^BEHOLD! DESTINY AWAITS THE HAND
D606 A6		FCB	Z10100110	^BEHOLD! DESTINY AWAITS THE HAND
D607 93		FCB	Z10010011	^BEHOLD! DESTINY AWAITS THE HAND
D608 05		FCB	Z00000101	^BEHOLD! DESTINY AWAITS THE HAND
D609 10		FCB	Z00010000	^BEHOLD! DESTINY AWAITS THE HAND
D60A 50		FCB	Z01010000	^BEHOLD! DESTINY AWAITS THE HAND
D60B 20		FCB	Z00100000	^BEHOLD! DESTINY AWAITS THE HAND
D60C 2E		FCB	Z00101110	^BEHOLD! DESTINY AWAITS THE HAND
D60D 20		FCB	Z00100000	^BEHOLD! DESTINY AWAITS THE HAND
D60E 3F		SWI		; and so on...
D60F 02		FCB	OUTSTI	
D610 C8		FCB	Z11001000	OF A NEW WIZARD...
D611 00		FCB	Z00000000	OF A NEW WIZARD...
D612 00		FCB	Z00000000	OF A NEW WIZARD...
D613 00		FCB	Z00000000	OF A NEW WIZARD...
D614 00		FCB	Z00000000	OF A NEW WIZARD...
D615 03		FCB	Z00000011	OF A NEW WIZARD...
D616 CC		FCB	Z11001100	OF A NEW WIZARD...
D617 00		FCB	Z00000000	OF A NEW WIZARD...
D618 81		FCB	Z10000001	OF A NEW WIZARD...
D619 C5		FCB	Z11000101	OF A NEW WIZARD...
D61A B8		FCB	Z10111000	OF A NEW WIZARD...
D61B 2E		FCB	Z00101110	OF A NEW WIZARD...
D61C 9D		FCB	Z10011101	OF A NEW WIZARD...
D61D 06		FCB	Z00000110	OF A NEW WIZARD...
D61E 44		FCB	Z01000100	OF A NEW WIZARD...
D61F F7		FCB	Z11110111	OF A NEW WIZARD...
D620 BC		FCB	Z10111100	OF A NEW WIZARD...
D621 20FE		BRA	\$	; round and round (forever)...
		END		

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PREVEA - Reveal Command

PAGE 86

```
INCLUDE B:PREVEA.ASM
; REVEAL COMMAND
; NAM PREVEA
; LIBRy CD
; XDEF PREVEA,PREV00
; XREF PARHND
; PSCT

PREVEA: Reveal a given magical object
Command Syntax:
REVEAL LEFT
REVEAL RIGHT

D623 B0CC31    PREVEA JSR      PARHND      ;determine left/right hand
D626 EEC4        LDU      ,U          ;load object ptr
D628 2714        BEQ      PREV99     ; empty hand

; Confirm Power Requirements have been met
D62A A64B        LDA      P.OCREV,U   ;do we have enough power to reveal?
D62C 2710        BEQ      PREV99     ; already revealed
D62E C619        LDB      #25         ;
D630 3B          MUL      ;           ;
D631 109317       CMPD     PPOW        ;           ;
D634 2E08        BGT      PREV99     ; nope

; Transform and Reveal the Object
D636 A649        LDA      P.OCTYP,U   ;copy new parameters
D638 3F          SWI      ;           ;
D639 18          FCB      OCBFIL     ;           ;
D63A 6F4B        CLR      P.OCREV,U   ;mark as revealed
D63C 3F          SWI      ;           ;
D63D 0D          FCB      STATUS      ;update the status line
D63E 39          PREV99    RTS        ;exit
; END
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PTURN - Turn Command

PAGE 87

```

    INCLUDE B:PTURN.ASM
!!!!!! T U R N   C O M M A N D !!!!!

;PTURN: Change the Player Direction

Command Syntax:
    TURN LEFT           ;left-hand 90-degree turn
    TURN RIGHT          ;right-hand 90-degree turn
    TURN AROUND         ;180-degree turn

D63F 8ED8D9 PTURN    LDX      #DIRTAB      ;direction names
D642 BBCBEC          JSR      PARSER       ;get the second token
D645 2F4C            BLE      ERRCMD2     ;null/illegal token
D647 D623            LDB      FDIR        ;load current facing direction

; Check for left turn

D649 8100            CMPA    #T.LT        ;left turn?
D64B 2607            BNE    PTUR10      ;nope
D64D 5A              DECB
D64E 8D10            BSR    PREVU        ;change directions
D650 8D22            BSR    LRTURN      ;create next display
D652 2015            BRA    PTUR90      ;display left-to-right turn
;               ,

; Check for right turn

D654 8101            PTUR10  CMPA    #T.RT        ;right turn?
D656 2605            BNE    PTUR20      ;nope
D658 5C              INCB
D659 8D12            BSR    PREVU        ;change directions
D65B 200A            BRA    PTUR22      ;create next display
;               , display right-to-left turn

; Check for about-face

D65D 8103            PTUR20  CMPA    #T.AROU     ;turn around?
D65F 2632            BNE    ERRCMD2     ;loser2!
D661 CB02            ADDB    #2
D663 8D08            BSR    PREVU        ;change directions
D665 8D10            BSR    RLTURN      ;create next display
D667 8D1B            PTUR22  BSR    RLTURN      ;round and round
;               ,

; Show the next display

D669 0AB4            PTUR90  DEC    SYNC        ;request screen update
D66B 13              RTS
D66C 39

```

```

; PREVU: Generate the next display in FLOP
D66D C403    PREVU   ANDB   #3      ;directions are MOD 4
D66F D723    STB     PDIR
D671 7EC660   JMP     PUPSUB  ;generate display without update

; LRTURN: Display Left-to-Right Turn Sequence
D674 8020    LRTURN  EQU    $       ;--- start of procedure LRTURN
D674 8030    BSR     TURN00 ;do initialization
D676 260B    BNE     LRTU99 ;not looking forward

; LRTU10: Display Left-to-Right Turn Sequence
D678 CC0008    LDD     #8      ;start from left side
D678 8030    BSR     TURN10 ;draw/erase line
D67D C30020   ADDD    #32     ;move toward the right
D680 40      TSTA
D681 27F8    BEQ     LRTU10 ;done yet?
D683 39      RTS     LRTU99 ;nope
;--- end of procedure LRTURN

; RLTURN: Display Right-to-Left Sequence
D684 8D10    RLTURN  EQU    $       ;--- start of procedure RLTURN
D684 8D20    BSR     TURN00 ;do initialization
D686 260A    BNE     RLTU99 ;not looking forward

; RLTU10: Display Right-to-Left Sequence
D688 CC00F8    LDD     #248    ;start from right side
D688 8D20    BSR     TURN10 ;draw/erase the line
D68D 830020   SUBD    #32     ;move toward left
D690 2AF9    BPL     RLTU10
D692 39      RTS     RLTU99 ;--- end of procedure RLTURN

; Local Jumps
D693 7ECBE1   ERRCMD2 JMP     CMDERR ;flag errors

; TURN00: Initialization
D696 DEB2    TURN00  LDW     DSPMOD ;are we in forward viewing mode?
D698 1183CE66  CMPU   #VIEWER
D69C 261B    BNE     TURN02 ;nope - don't do anything

; TURN02: Initialization
D6A0 8E8080   LDX     #$8080 ;manually set scaling
D6A1 9F4F    STX     VXSCAL
D6A3 0F8B    CLR     RANGE
D6A5 3F      SWI     SETFAD ;set the range
D6A6 00      FCB     SETFAD ;set the amount of fading

; TURN04: Initialization
D6A7 3F      SWI     ZFLIP
D6A8 08      FCB     #LINES ;erase the CURRENT screen
D6A9 8ED6C8   LBX     SWI
D6AC 3F      SWI     VCTLST ;draw horizontal lines
D6AD 01      FCB     #Y0
; TURN06: Initialization
D6AE BE0011   LDX     #17    ;set Y-coordinates
D6B1 9F2F    STX     Y0

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PTURN - Turn Command

PAGE 89

D6B3 8E0087 LDX #135  
D6B4 9F33 STX Y1  
  
D6B8 4F CLRA  
D6B9 39 RTS \*successful return  
; BYE

```

; TURN10: Draw/Erase a line

D6BA DD31    TURN10  STB     X0      ; store x-coordinates
D6BC DD35          STD     X1
D6BE 8D00          BSR     TURN12 ; draw the line
D6C0 BDCAB7    TURN12  JSR     VECTOR
D6C3 032C          COM     VDGINV ; then erase it
D6C5 39          RTS

```

#### LINES: Horizontal Lines

B6C6 1000	LINES	FCB	16,0
B6C8 10FF		FCB	16,255
B6CA FF		FCB	V\$NEW
;			
B6CB 8800		FCB	136,0
B6CD 88FF		FCB	136,255
B6CF FE		FCB	V\$END

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PTURN - Turn Command

PAGE 90

```

!!!!!! MOVE COMMAND !!!!!!
MOVE: Player Movement

Command Syntax:
MOVE           ;move forward
MOVE BACKWARD ;take a step backward
MOVE LEFT     ;sidestep left
MOVE RIGHT    ;sidestep right

D6D0 8E08D9   PMOVE  LDX    #DIRTAB
D6D3 B0CBEC   JSR    PARSER
D6D6 2DBB     BLT    ERRCMD2
D6D8 2E09     BGT    PMOV20
; direction names
; get the second token
; illegal token
; legal token

D6DA 0A73     Move Forward
; DEC      HLFSTP ;request half-step view
; SWI
; FCB      PUPDAT
; CLR     HLFSTP
; BRA     PMOV22
; step forward

D6E3 8102     Move Backward
PMOV20  CMPA   #T:BAK
D6E5 260C     BNE    PMOV30
D6E7 0A74     DEC    BAKSTP
D6E9 3F       SWI
D6EA 0E       FCB   PUPDAT
D6EB C602     LDB   #2
D6EE 0F74     CLR    BAKSTP
D6EF 8D2F     BSR    PSTEP
D6F1 201B     BRA    PMOV90
; backward
; flag for back half-step
; update the screen
; step backward

D6F3 8101     Move Left/Right
PMOV30  CMPA   #T,RT
D6F5 260A     BNE    PMOV40
D6F7 C601     LDB   #1
D6F9 8D25     BSR    PSTEP
D6FB 2611     BNE    PMOV90
D6FD 8D85     BSR    RLTURN
D6FF 200D     BRA    PMOV90
; sidestep right?
; nope
; take a step right
; we couldn't move!
; right-to-left line

D701 8100     PMOV40  CMPA   #T,LT
D703 268E     BNE    ERRCMD2
D705 C603     LDB   #3
D707 8D17     BSR    PSTEP
D709 2603     BNE    PMOV90
D70B B00674   JSR
; sidestep left?
; nope - error
; take a step left
; we couldn't move!
; left-to-right line

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PTURN - Turn Command

PAGE 91

; Determine energy requirements

D70E DC15	PMOV90 LDD	POBJWT	; (weight / 8) + 3
D710 B0D37F	JSE	ASRD3	; divide by 8
D713 C30003	ADDD	#3	
D716 D321	ADDD	PDAM	; accumulate as damage
D718 DU21	STD	PDAM	
D71A 3F	SWI		
D71B OC	FCB	HUPDAT	
D71C 0AB4	DEC	UPDATE	; request screen switch
D71E 13	SYNC		; wait for it to happen
D71F 39	RTS		

```
; PSTEP: Move the Player
```

```
; Inputs:
```

```
; B - relative change in direction
```

```
; Returns:
```

```
; PROW = updated
```

```
; PCOL = updated
```

D720 3406	PSTEP	PSHS	A,B	{save some regs
D722 6FE2		CLR	,S	;assume we will win
D724 DB23		ADDB	FDIR	;effect relative change
D726 C403		ANDB	#3	;MOD 4
D728 D78A		STR	FIR	;save direction
D72A DC13		LDD	PROW	;player position
D72C BDD136		JSR	STEPOK	;can we legally move?
D72F 2707		BEQ	FSTP10	; yes

```
; We can't move!
```

```
; SOUND$: A$THUD
```

```
$THUD!!!
```

D731 3F		SWI		
D732 1B		FCB	ISOUND	; ;
D733 14		FCB	A\$THUD	; ;
D734 6AE4		DEC	O,S	; loser
D736 DC13		LDD	PROW	; restore original position
D738 DD13	PSTP10	STD	PROW	; update position
D73A BDC660		JSR	PUPSUB	; create image in backplane
D73D 6DE0	PSTP99	TST	,S	; set condition codes
D73F 3586		PULS	A,B,PC	; restore regs/exit
		END		

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PUSE - Use Command

PAGE 93

```

;-----+
;      INCLUDE B:PUSE.ASM
;-----+
;      U S E   C O M M A N D
;-----+



;-----+
; PUSE: Activate on Object
;-----+
; Command Syntax:
; USE LEFT
; USE RIGHT

D741 BBCC31    PUSE    JSR     PARHND      ;parse for an object
D744 2721       BEQ     PUSE24     ;empty hand

;-----+
; Dispatch to proper routine
;-----+
D746 EC09       LDD     P.OCTYP,X    ;load the object type/class
D748 C105       PUSE10   CMPB    #K.TORC   ;torch?
D74A 260B       BNE     PUSE20     ;nope
D74C 9F24       STX     PTORCH    ;light torch
D74E RDD57D     PUSE12   JSR     PSTOWO   ;stow object

;-----+
; SOUND$: A$TORC
;-----+
;make loud torch sound
D751 3F          SWI     I$OUND
D752 1B          FCB     A$TORC
D753 11          FCB     ISOUND
D754 3F          SWI     A$TORC
D755 0E          FCB     PUPDAT
D756 39          RTS
;-----+
; Check for specific objects
;-----+
D757 1F13       PUSE20   TFR    X,U      ;copy OCP ptr
D759 8ED768     LDX     #PUSETAB  ;dispatch table addr
D75C A184       PUSE22   CMPA    ,X      ;match?
D75E 2708       BEQ     PUSE30     ;yes
D760 3003       LEAX    3,X      ;advance to next entry
D762 6CD77A     CMPX    #PUSEEND  ;exhausted table?
D765 25F5       BLO     PUSE22     ;nope
D767 39          PUSE24   RTS
;-----+
D768 6E9801~     PUSE30   JMP     [1,X]    ;dispatch to proper routine/exit

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PUSE - Use Command

PAGE 94

; Macro To Define Specific Object Dispatch Table Entries

```
USM    MACR
      FCB  \0
      FDB  \1
ENDM
```

```
D76B    USETAB EQU    $          ;beginning of table
D76B 05   FCB    T,FLA1    ;avatari flask (strength)
D76C D77A   FDB    UFL100   ;soma flask (healing)
D76E 09   FCB    T,FLA2    ;gift flask (poison)
D76F D783   FDB    UFL200   ;
D771 08   FCB    T,FLA3    ;
D772 D787   FDB    UFL300   ;
D774 04   ;           FCB    T,SCR1    ;seer scroll (map w/oll)
D775 D7A2   FDB    USC200   ;
D777 07   FCB    T,SCR2    ;vision scroll (plain)
D778 D7A0   FDB    USC100   ;
D77A    USEEND EQU    $
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PULSE - Use Command

PAGE 95

## ; Avatar Flask (Strength)

D77A CC03E8	UFL100	LDD	\$1000	;increase power by 1000 pts
D77D D317		ADD	PPOW	; ;
D77F DD17		STD	PPOW	; ;
D781 200F		BRA	UFL900	; ;

## ; Soma Flask (Healing)

D783 4F	UFL200	CLRA		;heal all damage
D784 5F		CLRB		; ;
D785 2009		BRA	UFL310	; ;

## ; Solonium Flask (Poison)

D787 9E17	UFL300	LDX	PPOW	;cause 80% damage
D789 8666		LDA	\$102	; ;
D78B BDD436		JSR	SCAL16	; ;
D78E D321		ADD	PDAM	; ;
D790 DD21	UFL310	STD	PDAM	; ;
D792 C617	UFL900	LDR	#T,FLA4	;now empty the flask
D794 E749		STB	P,OCTYP,U	;empty flasks are always revealed
D796 6F4B		CLR	P,OCREV,U	; ;
D798 3F		SWI		; ;
D799 1B		FCB	ISOUND	;make flask sound
D79A 0C		FCB	A\$FLAS	;update the status line
D79B 3F		SWI		; ;
D79C 0D		FCB	STATUS	;update the heartrate
D79D 3F		SWI		; ;
D79E 0C		FCB	HUPDAT	;done
D79F 39		RTS		; ;

## ; Vision Scroll

D7A0 4F	USC100	CLRA		;map w/o features
D7A1 8C		FCB	SKIP2	; ;

## ; Seer Scroll

D7A2	DEBMAP	EQU	\$	;debugging command
D7A2 86FF	USC200	LDA	##FF	;map with the features
D7A4 9794		STA	MAPFLG	;set the flag
D7A6 604B		TST	P,OCREV,U	;has this object been revealed?
D7A8 260C		BNE	USC199	;nope!

## ; SOUND\$. A\$SCRO

D7AA 3F		SWI		; ;
D7AB 1B		FCB	ISOUND	; ;
D7AC 0E		FCB	A\$SCRO	; ;
D7AD 0FAD	USC210	CLR	HEARTF	;turn off the heart
D7AF 8ECDB2		LDX	#MAPPER	;set map display mode
D7B2 9FB2		STX	DSPMOD	; ;
D7B4 7F		SMT		; ;

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
PUSE - Use Command

PAGE 96

```
D7B5 0E      FCB      PUPDAT      ;done  
D7B6 39      USC199   RTS  
;  
END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
FZTAFE - Cassette Tape Routines

PAGE 97

```

INCLUDE B:PTAPE.ASM
;CASSSETTE TAPE ROUTINES

PZLOAD: Game Restore Function
PZLOAD BSR      FILNAM      ;get the file name
DEC      ZFLAG       ;flag for load
RTS
;
FILNAM LDX      #TOKEN     ;initialize TOKEN buffer
LEAU    32,X        ;negative ones.
SWI
FCB      NEGRAM     ;grab the next token/exit
JMP      GETTOK

PZSAVE: Game Restore Function
PZSAVE BSR      FILNAM      ;get the file name
STX      >$007E      ;set buffer addr
LDI      #$000F      ;BLKTYP,,BLKLEN (0=namefile,,15=length)
STD      >$007C      ;
;

Set up NAMEFILE Block:
Bytes  Description
0-7   Program Name
8      File Type (0=BASIC, 1=DATA, 2=Machine Language)
9      ASCII Flag (0=Binary, -1=ASCII)
10     Gap Flag (1=continuous, -1=gaps)
11-12  Start Addr of Machine Language Program
13-14  Load Addr of Machine Language Program

LDA      #2           ;file type = machine language
STA      8,X          ;ASCII flag = Binary
CLR      9,X          ;
INC      ZFLAG       ;flag for save
RTS
;
END

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMDAT - Initialization Data

PAGE 98

```
INCLUD B:COMDAT.ASM
!!!!!!INITIALIZATION DATA!!!!!!
NAM COMDAT
LIBRy CD
XDEF TCBDAT,RAMDAT,DEMDAT,GAMDAT
XDEF DSPO,DSP1,STSVDB,FRIVDB
XDEF CMTTAB,CMTEND
XREF T,SW02,T,SW03,T,TOR4,T,SHI4
XREF PLAYER,LUKNEW,HSLOW,BURNER,CREGEN
XREF CLOCK,SWISER,GAME,AUTTAB
XREF SW2SER
PSCT

DEMDAT: Demonstration Mode Initial Player Objects
D7D5 0B DEMDAT EQU $ ;where data starts
D7D6 0F FCB T,SW02 ;iron sword
D7D7 10 FCB T,TOR4 ;pine torch
D7D8 FF FCB T,SHI4 ;leather shield
          FCB -1 ;end of data

GAMDAT: Game Mode Initial Player Objects
D7D9 11 GAMDAT EQU $ ;where data starts
D7D9 11 FCB T,SW03 ;wooden sword
D7D9 11 FCB T,TOR4 ;pine torch
D7D9 11 FCB -1 ;end of data

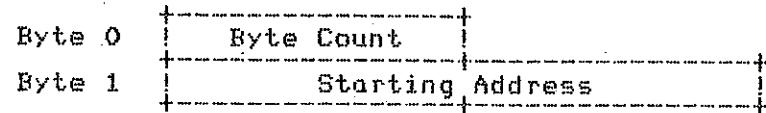
System Task Control Blocks
D7DC D1EB TCBDAT EQU $ ;where the data starts
D7DC D1EB FDB PLAYER ;Human Input Module
D7DE D1C2 FDB LUKNEW ;Delayed Update Task
D7E0 D1E5 FDB HSLOW ;Heartrate Hysteresis Task
D7E2 D19B FDB BURNER ;Torch Burn Module
D7E4 D027 FDB CREGEN ;Creature Regeneration
D7E6 0000 FDB 0 ;end of initial system TCBs
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMDAT - Initialization Data

PAGE 99

Initialization Data Format:



```
INI      MACR      \1          ;byte count  
        FCB       \0          ;addr  
        FDR  
        ENDM
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMDAT - Initialization Data

PAGE 0

```

D7E8      RAMDAT EQU    $           ; Interrupt Vectors
D7E8  0C      INI     U$SWI2,3*4   ; four interrupt vectors
D7E9 0103      FCB     3*4
D7EB 7EC371      FDB     U$SWI2
D7EE 7EC352      JMP     SW2SER
D7F1 7EC27D      JMP     SWISER
D7F4 7EC27D      JMP     CLOCK
                                ; SWI2
                                ; SWI
                                ; NMI
                                ; IRQ

```

ONCE-only Initialization Parameters			
	INI	BIGONE+1,23	where to start
D7F7 17	FCB	23	.
D7F8 0202	FDB	BIGONE+1	.
D7FA 01	FCB	1	BIGONE (RH)
D7FB FFFF	FDB	\$FFFF	NEGONE
D7FD 0080	FDB	128	VCNTRX
D7FF 004C	FDB	76	VCNTRY
D801 D870	FDB	DSP0	FLIP
D803 D876	FDB	DSP1	FLOP
D805 D988	FDB	AUTTAB	AUTPTR
D807 0B15	FDB	OCBLND	OCBPTR
D809 02F1	FDB	LINBUF	LINPTR
D80B 0C	FCB	12	PROW
D80C 16	FCB	22	PCOL
D80D 0023	FDB	30+5	POBJWT
D80F 17A0	FCB	\$17,160	PPOW

Text Control Blocks

D811 54	INI	TXTExA,(TX,LEN*3)+(CTYPES*5)	
D812 0380	FCB	(TX,LEN*3)+(CTYPES*5)	
	FDB	TXTExA	
D814 1000	FDB	DO\$BAS	:TXTExA,TXBAS
D816 0260	FDB	32*19	:TXTExA,TXCHR
D818 0000	FDB	0	:TXTExA,TXCUR
D81A 00FF	FCB	0,-1	,TXTExA,TXINV,,TXTExA,TXSGL
D81C 2300	FDB	DO\$BAST+(256*19)	:TXTSTS,TXBAS
D81E 0040	FDB	32*2	:TXTSTS,TXCHR
D820 0000	FDB	0	:TXTSTS,TXCUR
D822 FF00	FCB	-1,0	,TXTSTS,TXINV,,TXTSTS,TXSGL
D824 2400	FDB	DO\$BAST+(256*20)	:TXTPRI,TXBAS
D826 0080	FDB	32*4	:TXTPRI,TXCHR
D828 0000	FDB	0	:TXTPRI,TXCUR
D82A 0000	FCB	0,0	,TXTPRI,TXINV,,TXTPRI,TXSGL

### Creature/Level Matrix

182C 0909040200CMTTAB FCB 9,9,4,2,0,0,0,0,0,0,0,0  
0909040200CMTTAB FCB 9,9,4,2,0,0,0,0,0,0,0,0

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMDAT - Initialization Data

PAGE 1

D844 0000000400 FCB 0,0,0,4,0,6,8,4,0,0,1,0  
D850 0000000000 FCB 0,0,0,0,0,0,8,6,6,4,0,0  
D85C 0202020202 FCB 2,2,2,2,2,2,2,4,4,8,0,1  
D868 CMTEND EQU \$

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMDAT - Initialization Data

PAGE 2

Empty Hand Parameters

```
    INI    P.OCCLS+EMPHND,4 ;two bytes
D868 04    FCB    4
D869 0E11    FDR    P.OCCLS+EMPHND
D86B 0400    FCB    4,0      ;sword noises
D86D 0005    FCB    0,5      ;magic/physical offense
```

End of Direct Page RAM Initial Values

```
D86F 00    FCB    0
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
COMDAT - Initialization Data

PAGE 3

!!!!!!  
PURE DATA BLOCKS  
!!!!!!

; Vector Display Control Blocks

D870 1000	DSPO	FDB	D0\$BAS	;display base addr
D872 2300		FDB	D0\$END	;display ending addr
D874 2046		FDB	D0.SAM	;magic number for SAM
D876 2800	DSP1	FDB	D1\$BAS	;display base addr
D878 3B00		FDB	D1\$END	;display ending addr
D87A 20A6		FDB	D1.SAM	;magic number for SAM
D87C 2300	STSVDB	FDB	D0\$BAS+152*32	;152nd scan line
D87E 2400		FDB	D0\$BAS+160*32	;160th scan line
D880 0000		FDB	0	;filler
D882 3B00		FDB	D1\$BAS+152*32	; ;
D884 3C00		FDB	D1\$BAS+160*32	; ;
D886 0000		FDB	0	; ;
D888 2400	PRIVDB	FDB	D0\$BAS+160*32	;160th scan line
D88A 2800		FDB	D0\$BAS+192*32	;192nd scan line
D88C 0000		FDB	0	;filler
D88E 3C00		FDB	D1\$BAS+160*32	; ;
D890 4000		FDB	D1\$BAS+192*32	; ;
D892 0000		FDB	0	; ;
		END		

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - ToKen Tables

PAGE 4

```
INCLUD B:TOKEN.ASM
!!!!!! TOKEN TABLES !!!!!!
NAM    TOKEN
XDEF   CMDTAB,DIRTAB,GENTAB,ADJTAB,AUTTAB,AUTEND
PSCT

Are we debugging?
IFUD DEBFLG
DEBFLG EQU 0
ENDC
```

## ; CMDTAB: Primary Command Tokens

D894 OF 0000	CMDTAB	FCB	CMDINUM	Number of entries
	FOO	SET	0	
0000	T.ATTK	XDEF EQU	T.ATTK,M\$ATTK	
0001	FOO	SET	FOO	
D895	M\$ATTK	EQU	\$	
D895 30		FCB	Z00110000	;ATTACK
D896 03		FCB	Z00000011	;ATTACK
D897 4A		FCB	Z01001010	;ATTACK
D898 04		FCB	Z00000100	;ATTACK
D899 6B		FCB	Z01101011	;ATTACK
0001	T.CLIM	XDEF EQU	T.CLIM,M\$CLIM	
0002	FOO	SET	FOO	
D89A	M\$CLIM	EQU	\$	
D89A 28		FCB	Z00101000	;CLIMB
D89B 06		FCB	Z00000110	;CLIMB
D89C C4		FCB	Z11000100	;CLIMB
D89D B4		FCB	Z10110100	;CLIMB
D89E 40		FCB	Z01000000	;CLIMB
0002	T.DROP	XDEF EQU	T.DROP,M\$DROP	
0003	FOO	SET	FOO	
D89F	M\$DROP	EQU	\$	
D89F 20		FCB	Z00100000	;DROP
D8A0 09		FCB	Z00001001	;DROP
D8A1 27		FCB	Z00100111	;DROP
D8A2 C0		FCB	Z11000000	;DROP
0003	T.EXAM	XDEF EQU	T.EXAM,M\$EXAM	
0004	FOO	SET	FOO	
D8A3	M\$EXAM	EQU	\$	
D8A3 38		FCB	Z00111000	;EXAMINE
D8A4 0B		FCB	Z00001011	;EXAMINE
D8A5 80		FCB	Z10000000	;EXAMINE
D8A6 B5		FCB	Z10110101	;EXAMINE
D8A7 2F		FCB	Z00101110	;EXAMINE
D8A8 28		FCB	Z00101000	;EXAMINE
0004	T.GET	XDEF EQU	T.GET,M\$GET	
0005	FOO	SET	FOO	
D8A9	M\$GET	EQU	\$	
D8A9 18		FCB	Z00011000	;GET
D8AA 0E		FCB	Z00001110	;GET
D8AB 5A		FCB	Z01011010	;GET
D8AC 00		FCB	Z00000000	;GET

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 6

		XDEF	T.INCN,M\$INCN
0005	T.INCN	EQU	FOO
0006	FOO	SET	FOO+1
D8AD	M\$INCN	EQU	\$
D8AD 30		FCB	Z00110000 ;INCANT
D8AE 12		FCB	Z00010010 ;INCANT
D8AF E1		FCB	Z11100001 ;INCANT
D8B0 85		FCB	Z10000101 ;INCANT
D8B1 D4		FCB	Z11010100 ;INCANT
		XDEF	T.LOOK,M\$LOOK
0006	T.LOOK	EQU	FOO
0007	FOO	SET	FOO+1
D8B2	M\$LOOK	EQU	\$
D8B3 20		FCB	Z00100000 ;LOOK
D8B3 18		FCB	Z00011000 ;LOOK
D8B4 F7		FCB	Z11101011 ;LOOK
D8B5 AC		FCB	Z10101100 ;LOOK
		XDEF	T.MOVE,M\$MOVE
0007	T.MOVE	EQU	FOO
0008	FOO	SET	FOO+1
D8B6	M\$MOVE	EQU	\$
D8B6 20		FCB	Z00100000 ;MOVE
D8B7 1A		FCB	Z00011010 ;MOVE
D8B8 FB		FCB	Z11110101 ;MOVE
D8B9 14		FCB	Z00010100 ;MOVE
		XDEF	T.PULL,M\$PULL
0008	T.PULL	EQU	FOO
0009	FOO	SET	FOO+1
D8B9	M\$PULL	EQU	\$
D8B9 20		FCB	Z00100000 ;PULL
D8BB 21		FCB	Z00100001 ;PULL
D8BC 56		FCB	Z010101010 ;PULL
D8BD 30		FCB	Z00110000 ;PULL
		XDEF	T.REVE,M\$REVE
0009	T.REVE	EQU	FOO
000A	FOO	SET	FOO+1
D8BE	M\$REVE	EQU	\$
D8BE 30		FCB	Z00110000 ;REVEAL
D8BF 24		FCB	Z00100100 ;REVEAL
D8C0 5B		FCB	Z01011011 ;REVEAL
D8C1 14		FCB	Z00010100 ;REVEAL
D8C2 2C		FCB	Z00101100 ;REVEAL
		XDEF	T.STOW,M\$STOW
000A	T.STOW	EQU	FOO
000B	FOO	SET	FOO+1
D8C3	M\$STOW	EQU	\$
D8C3 20		FCB	Z00100000 ;STOW
D8C4 27		FCB	Z00100111 ;STOW
D8C5 47		FCB	Z01000111 ;STOW
D8C6 DC		FCB	Z11011100 ;STOW

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 7

		XDEF	T.TURN,M\$TURN
000B	T.TURN	EQU	FOO
000C	FOO	SET	FOO+1
D8C7	M\$TURN	EQU	\$
D8C7 20		FCB	Z00100000
D8C8 29		FCB	Z00101001
D8C9 30		FCB	Z01011001
D8CA 38		FCB	Z00111000
			,TURN
		XDEF	T.USC,M\$USC
000C	T.USC	EQU	FOO
000D	FOO	SET	FOO+1
D8CB	M\$USC	EQU	\$
D8CB 18		FCB	Z00011000
D8CC 28		FCB	Z00101011
D8CD 32		FCB	Z00110010
D8CE 80		FCB	Z10000000
			,USC
		XDEF	T.ZLOA,M\$ZLOA
000D	T.ZLOA	EQU	FOO
000E	FOO	SET	FOO+1
D8CF	M\$ZLOA	EQU	\$
D8CF 28		FCB	Z00101000
D8D0 34		FCB	Z00110100
D8D1 07		FCB	Z11000111
D8D2 84		FCB	Z10000100
D8D3 80		FCB	Z10000000
			,ZLOAD
		XDEF	T.ZSAV,M\$ZSAV
000E	T.ZSAV	EQU	FOO
000F	FOO	SET	FOO+1
D8D4	M\$ZSAV	EQU	\$
D8D4 28		FCB	Z00101000
D8D5 35		FCB	Z00110101
D8D6 30		FCB	Z00110000
D8D7 08		FCB	Z11011000
D8D8 A0		FCB	Z10100000
			,ZSAVE

## ; Debugging Commands

D8D9 ; IF DEBFLG

```

D8D9 ; XDEF T.QMAP,M$QMAP
D8D9 T.QMAP EQU FOO
D8D9     SET FOO+1
D8D9 M$QMAP FCB Z00100000 ;QMAP
D8D9     FCB Z00100010 ;QMAP
D8D9     FCB Z11010000 ;QMAP
D8D9     FCB Z11000000 ;QMAP

```

```

D8D9 ; XDEF T.QUP,M$QUP
D8D9 T.QUP EQU FOO
D8D9     SET FOO+1
D8D9 FOO FCB Z00011000 ;QUP
D8D9     FCB Z00100011 ;QUP
D8D9     FCB Z01011000 ;QUP
D8D9     FCB Z00000000 ;QUP

```

```

D8D9 ; XDEF T.QDN,M$QDN
D8D9 T.QDN EQU FOO
D8D9     SET FOO+1
D8D9 FOO FCB Z00011000 ;QDN
D8D9     FCB Z00100010 ;QDN
D8D9     FCB Z01000111 ;QDN
D8D9     FCB Z00000000 ;QDN

```

```

D8D9 ; XDEF T.QRUP,M$QRUP
D8D9 T.QRUP EQU FOO
D8D9     SET FOO+1
D8D9 FOO FCB Z00100000 ;QRUP
D8D9     FCB Z00100011 ;QRUP
D8D9     FCB Z00101010 ;QRUP
D8D9     FCB Z11000000 ;QRUP

```

```

D8D9 ; XDEF T.QRDN,M$QRDN
D8D9 T.QRDN EQU FOO
D8D9     SET FOO+1
D8D9 FOO FCB Z00100000 ;QRDN
D8D9     FCB Z00100011 ;QRDN
D8D9     FCB Z00100010 ;QRDN
D8D9     FCB Z00111000 ;QRDN

```

```

D8D9 ; XDEF T.QMUP,M$QMUP
D8D9 T.QMUP EQU FOO
D8D9     SET FOO+1
D8D9 FOO FCB Z00100000 ;QMUP
D8D9     FCB Z00100010 ;QMUP
D8D9     FCB Z11010100 ;QMUP
D8D9     FCB Z11000000 ;QMUP

```

```

D8D9 ; XDEF T.QMDN,M$QMDN
D8D9 T.QMDN EQU FOO
D8D9     SET FOO+1

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 9

```
D8D9      M$QMDN  FCB    Z00100000  ;QMDN
D8D9      FCB    Z00100010  ;QMDN
D8D9      FCB    Z11010010  ;QMDN
D8D9      FCB    Z00111000  ;QMDN
D8D9      ;      ENDFIF   ; DEBFLG (end of debug command table.)
D8D9      ;      ENDFIF
000F      CMDNUM EQU     FOO      ;number of entries
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - ToKen Tables

PAGE 10

```
; DIRTAB: Direction ToKen Tables
D8D9 06      DIRTAB   FCB     DIRNUM
0000          FOO      SET     0
;
;           XDEF    T,LT,M$LT
0000          T,LT    EQU    FOO
0001          FOO    SET    FOO+1
D8DA          M$LT    EQU    $
D8DA 20        FCB    Z00100000 ;LEFT
D8DB 18        FCB    Z00011000 ;LEFT
D8DC 53        FCB    Z01010011 ;LEFT
D8DD 50        FCB    Z01010000 ;LEFT
;
;           XDEF    T,RT,M$RT
0001          T,RT    EQU    FOO
0002          FOO    SET    FOO+1
D8DE          M$RT    EQU    $
D8DE 28        FCB    Z00101000 ;RIGHT
D8DF 24        FCB    Z00100100 ;RIGHT
D8E0 93        FCB    Z10010011 ;RIGHT
D8E1 A2        FCB    Z10100010 ;RIGHT
D8E2 80        FCB    Z10000000 ;RIGHT
;
;           XDEF    T,BAK,M$BAK
0002          T,BAK   EQU    FOO
0003          FOO    SET    FOO+1
D8E3          M$BAK   EQU    $
D8E3 20        FCB    Z00100000 ;BACK
D8E4 04        FCB    Z00000100 ;BACK
D8E5 11        FCB    Z00010001 ;BACK
D8E6 AC        FCB    Z10101100 ;BACK
;
```

UNIFIELD TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08  
DAGGERATH U3.2 - Copyright (c) 1983 Unifield Technologies Inc.  
TORNEN - Token Tables  
PAGE 11

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 12

; ADJTAB: Adjective Token Tables

D8F3 19  
0000 ADJTAB FCB ADJNUM  
FOO SET 0

; Level-4

0000	XDEF	T.RN05,M\$RN05
0001	EQU	FOO
D8F4	SET	FOO+1
D8F4 38	EQU	\$
D8F5 67	FCB	Z00111000 ;SUPREME
D8F6 58	FCB	Z01100111 ;SUPREME
D8F7 48	FCB	Z01011000 ;SUPREME
D8F8 AD	FCB	Z10101101 ;SUPREME
D8F9 28	FCB	Z00101000 ;SUPREME

; Level-3 Objects

0001	XDEF	T.RN01,M\$RN01
0002	EQU	FOO
D8FA	SET	FOO+1
D8FA 28	EQU	\$
D8FB 54	FCB	Z00101000 ;JOULE
D8FC FA	FCB	Z01010100 ;JOULE
D8FD B0	FCB	Z11111010 ;JOULE
D8FE AO	FCB	Z10110000 ;JOULE

; XDEF T.SW01,M\$SW01

0002	EQU	FOO
0003	SET	FOO+1
D8FF	EQU	\$
D8FF 31	FCB	Z00110001 ;ELVISH
D900 0A	FCB	Z00001010 ;ELVISH
D901 CB	FCB	Z11001011 ;ELVISH
D902 26	FCB	Z00100110 ;ELVISH
D903 68	FCB	Z01101000 ;ELVISH

; XDEF T.SHI1,M\$SHI1

0003	EQU	FOO
0004	SET	FOO+1
D904	EQU	\$
D904 38	FCB	Z00111000 ;MITHRIL
D905 DA	FCB	Z11011010 ;MITHRIL
D906 9A	FCB	Z10011010 ;MITHRIL
D907 22	FCB	Z00100010 ;MITHRIL
D908 49	FCB	Z01001001 ;MITHRIL
D909 60	FCB	Z01100000 ;MITHRIL

## ; Level-2 Objects

0004	XDEF	T.SCR1,M\$SCR1
	T.SCR1	EQU FOO
0005	FOO	SET FOO+1
D90A	M\$SCR1	EQU \$
D90A 20		FCB Z00100000 ;SEER
D90B A6		FCB Z10100110 ;SEER
D90C 52		FCB Z01010010 ;SEER
D90D C8		FCB Z11001000 ;SEER

## ; XDEF T.FLA1,M\$FLA1

0005	T.FLA1	EQU FOO
0006	FOO	SET FOO+1
D90E	M\$FLA1	EQU \$
D90E 28		FCB Z00101000 ;THEWS
D90F 28		FCB Z00101000 ;THEWS
D910 82		FCB Z10000010 ;THEWS
D911 DE		FCB Z11011110 ;THEWS
D912 60		FCB Z01100000 ;THEWS

## ; Level-1 Objects

0006	XDEF	T.RN03,M\$RN03
	T.RN03	EQU FOO
0007	FOO	SET FOO+1
D913	M\$RN03	EQU \$
D913 20		FCB Z00100000 ;RIME
D914 64		FCB Z01100100 ;RIME
D915 96		FCB Z10010110 ;RIME
D916 94		FCB Z10010100 ;RIME

## ; XDEF T.SCR2,M\$SCR2

0007	T.SCR2	EQU FOO
0008	FOO	SET FOO+1
D917	M\$SCR2	EQU \$
D917 30		FCB Z00110000 ;VISION
D918 AC		FCB Z10101100 ;VISION
D919 99		FCB Z10011001 ;VISION
D91A A5		FCB Z10100101 ;VISION
D91B EE		FCB Z11101110 ;VISION

## ; XDEF T.FLA3,M\$FLA3

0008	T.FLA3	EQU FOO
0009	FOO	SET FOO+1
D91C	M\$FLA3	EQU \$
D91C 20		FCB Z00100000 ;ABYE
D91D 02		FCB Z00000010 ;ABYE
D91E 2C		FCB Z00101100 ;ABYE
D91F 94		FCB Z10010100 ;ABYE

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 14

```

0009      ; XDEF T.FLA2,M$FLA2
          T.FLA2 EQU FOO
000A      FOO SET FOO+1
D920      M$FLA2 EQU $
D920 20   FCB Z00100000 ;HALE
D921 10   FCB Z00010000 ;HALE
D922 16   FCB Z00010110 ;HALE
D923 14   FCB Z00010100 ;HALE

          ; XDEF T.TOR1,M$TOR1
000A      T.TOR1 EQU FOO
000B      FOO SET FOO+1
D924      M$TOR1 EQU $
D924 29   FCB Z00101001 ;SOLAR
D925 66   FCB Z01100110 ;SOLAR
D926 F6   FCB Z11110110 ;SOLAR
D927 06   FCB Z00000110 ;SOLAR
D928 40   FCB Z01000000 ;SOLAR

          ; XDEF T.SHI3,M$SHI3
000B      T.SHI3 EQU FOO
000C      FOO SET FOO+1
D929      M$SHI3 EQU $
D929 30   FCB Z00110000 ;BRONZE
D92A C5   FCB Z11000101 ;BRONZE
D92B 27   FCB Z00100111 ;BRONZE
D92C BB   FCB Z10111011 ;BRONZE
D92D 45   FCB Z01000101 ;BRONZE

          Level-0

          ; XDEF T.RN02,M$RN02
000C      T.RN02 EQU FOO
000D      FOO SET FOO+1
D92E      M$RN02 EQU $
D92E 30   FCB Z00110000 ;VULCAN
D92F 60   FCB Z01101101 ;VULCAN
D930 56   FCB Z01010110 ;VULCAN
D931 OC   FCB Z00001100 ;VULCAN
D932 2E   FCB Z00101110 ;VULCAN

          ; XDEF T.SW02,M$SW02
000D      T.SW02 EQU FOO
000E      FOO SET FOO+1
D933      M$SW02 EQU $
D933 21   FCB Z00100001 ;IRON
D934 13   FCB Z00010011 ;IRON
D935 27   FCB Z00100111 ;IRON
D936 B8   FCB Z10111000 ;IRON

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 15

```

;          XDEF    T.TOR2,M$TOR2
000E      T.TOR2  EQU    FOO
000F      FOO    SET    FOO+1
D937      M$TOR2 EQU    $
D937 29   FCB    Z00101001 ;LUNAR
D938 59   FCB    Z01011001 ;LUNAR
D939 57   FCB    Z01010111 ;LUNAR
D93A 06   FCB    Z00000010 ;LUNAR
D93B 40   FCB    Z01000000 ;LUNAR

;          XDEF    T.TOR4,M$TOR4
000F      T.TOR4  EQU    FOO
0010      FOO    SET    FOO+1
D93C      M$TOR4 EQU    $
D93C 21   FCB    Z00100001 ;PINE
D93D 60   FCB    Z01100000 ;PINE
D93E 97   FCB    Z10010111 ;PINE
D93F 14   FCB    Z00010100 ;PINE

;          XDEF    T.SHI4,M$SHI4
0010      T.SHI4  EQU    FOO
0011      FOO    SET    FOO+1
D940      M$SHI4 EQU    $
D940 38   FCB    Z00111000 ;LEATHER
D941 D8   FCB    Z11011000 ;LEATHER
D942 50   FCB    Z01010000 ;LEATHER
D943 D1   FCB    Z11010001 ;LEATHER
D944 05   FCB    Z00000101 ;LEATHER
D945 90   FCB    Z10010000 ;LEATHER

;          XDEF    T.SW03,M$SW03
0011      T.SW03  EQU    FOO
0012      FOO    SET    FOO+1
D946      M$SW03 EQU    $
D946 31   FCB    Z00110001 ;WOODEN
D947 2E   FCB    Z00101110 ;WOODEN
D948 F7   FCB    Z11110111 ;WOODEN
D949 90   FCB    Z10010000 ;WOODEN
D94A AE   FCB    Z10101110 ;WOODEN
;
```

```

; Special Objects

0012      XDEF    T,RN15,M$RN15
          T,RN15 EQU    FOO
0013      FOO     SET    FOO+1
          M$RN15 EQU    $
D94B 28   FCB     Z00101000 ;FINAL
D94C 4C   FCB     Z01001100 ;FINAL
D94D 97   FCB     Z10010111 ;FINAL
D94E 05   FCB     Z00000101 ;FINAL
D94F 80   FCB     Z10000000 ;FINAL

; XREF T,RN11,M$RN11
0013      T,RN11 EQU    FOO
          FOO     SET    FOO+1
          M$RN11 EQU    $
D950 30   FCB     Z00110000 ;ENERGY
D951 4A   FCB     Z01001010 ;ENERGY
D952 E2   FCB     Z11100010 ;ENERGY
D953 C8   FCB     Z11001000 ;ENERGY
D954 F9   FCB     Z11111001 ;ENERGY

; XREF T,RN13,M$RN13
0014      T,RN13 EQU    FOO
          FOO     SET    FOO+1
          M$RN13 EQU    $
D955 18   FCB     Z00011000 ;ICE
D956 52   FCB     Z01010010 ;ICE
D957 32   FCB     Z00110010 ;ICE
D958 80   FCB     Z10000000 ;ICE

; XREF T,RN12,M$RN12
0015      T,RN12 EQU    FOO
          FOO     SET    FOO+1
          M$RN12 EQU    $
D959 20   FCB     Z00100000 ;FIRE
D95A 4C   FCB     Z01001100 ;FIRE
D95B 99   FCB     Z10011001 ;FIRE
D95C 14   FCB     Z00010100 ;FIRE
;
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - ToKen Tables

PAGE 17

```
; XDEF T.RN20,M$RN20
0016    T.RN20 EQU FOO
0017    FOO SET FOO+1
D95D    M$RN20 EQU $
D95D 20   FCB Z00100000 ;GOLD
D95E 4E   FCB Z01001110 ;GOLD
D95F F6   FCB Z11110110 ;GOLD
D960 10   FCB Z00010000 ;GOLD

; XDEF T.FLA4,M$FLA4
0017    T.FLA4 EQU FOO
0018    FOO SET FOO+1
D961    M$FLA4 EQU $
D961 28   FCB Z00101000 ;EMPTY
D962 0A   FCB Z00001010 ;EMPTY
D963 08   FCB Z11011000 ;EMPTY
D964 53   FCB Z01010011 ;EMPTY
D965 20   FCB Z00100000 ;EMPTY

; XREF T.TORS,M$TORS
0018    T.TORS EQU FOO
0019    FOO SET FOO+1
D966    M$TORS EQU $
D966 21   FCB Z00100001 ;DEAD
D967 48   FCB Z01001000 ;DEAD
D968 50   FCB Z01010000 ;DEAD
D969 90   FCB Z10010000 ;DEAD

0019    ADJNUM EQU FOO
```

## ; GENTAB: Generic Token Tables

D96A 06	GENTAB	FCB	GENNUM
0000	FOO	SET	0
0000	T,FLAO	XDEF EQU	T,FLAO,M\$FLAO
0001	FOO	SET	FOO+1
D96B	M\$FLAO	EQU	\$
D96B 28	FCB	Z00101000	;FLASK
D96C 0C	FCB	Z00001100	;FLASK
D96D C0	FCB	Z11000000	;FLASK
D96E CD	FCB	Z11001101	;FLASK
D96F 60	FCB	Z01100000	;FLASK
0001	T,RINO	XDEF EQU	T,RINO,M\$RINO
0002	FOO	SET	FOO+1
D970	M\$RINO	EQU	\$
D970 20	FCB	Z00100000	;RING
D971 64	FCB	Z01100100	;RING
D972 97	FCB	Z10010111	;RING
D973 1C	FCB	Z00011100	;RING
0002	T,SCRO	XDEF EQU	T,SCRO,M\$SCRO
0003	FOO	SET	FOO+1
D974	M\$SCRO	EQU	\$
D974 30	FCB	Z00110000	;SCROLL
D975 06	FCB	Z10100110	;SCROLL
D976 39	FCB	Z00111001	;SCROLL
D977 3D	FCB	Z00111101	;SCROLL
D978 8C	FCB	Z10001100	;SCROLL
0003	T,SHIO	XDEF EQU	T,SHIO,M\$SHIO
0004	FOO	SET	FOO+1
D979	M\$SHIO	EQU	\$
D979 30	FCB	Z00110000	;SHIELD
D97A E6	FCB	Z11100110	;SHIELD
D97B 04	FCB	Z10000100	;SHIELD
D97C 95	FCB	Z10010101	;SHIELD
D97D 84	FCB	Z10000100	;SHIELD
0004	T,SWOO	XDEF EQU	T,SWOO,M\$SWOO
0005	FOO	SET	FOO+1
D97E	M\$SWOO	EQU	\$
D97E 29	FCB	Z00101001	;SWORD
D97F 27	FCB	Z00100111	;SWORD
D980 77	FCB	Z01110111	;SWORD
D981 C8	FCB	Z11001000	;SWORD
D982 80	FCB	Z10000000	;SWORD

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - Token Tables

PAGE 19

	XDEF	T,TORO,M\$TORO
0005	T,TORO	EQU FOO
0006	FOO	SET FOO+1
D983	M\$TORO	EQU \$
D983 29		FCB Z00101001 TORCH
D984 68		FCB Z01101000 TORCH
D985 F9		FCB Z11111001 TORCH
D986 0B		FCB Z00001101 TORCH
D987 00		FCB Z00000000 TORCH
0006	GENNUM	EQU FOO

## Auto-Play Commands

```

ATM    MACR
      FCB    NARG
      FDB    \0
      IFGT NARG-1
      FDB    \1
      ENIC
      IFGT NARG-2
      FDB    \2
      ENDI
      ENDM

D988   AUTTAB EQU   $
D988 01 , ATM   M$EXAM
D989 D8A3 , FCB   1
             FDB   M$EXAM

D98B 03 , ATM   M$PULL,M$RT,M$TORO
D98C D8BA , FCB   3
D98E D8DE , FDB   M$PULL
D990 D983 , FDB   M$RT
             FDB   M$TORO

D992 02 , ATM   M$USE,M$RT
D993 D8CB , FCB   2
D995 D8DE , FDB   M$USE
             FDB   M$RT

D997 01 , ATM   M$LOOK
D998 D8B2 , FCB   1
             FDB   M$LOOK

D99A 01 , ATM   M$MOVE
D99B D8B6 , FCB   1
             FDB   M$MOVE

D99D 03 , ATM   M$PULL,M$LT,M$SHIO
D99E D8BA , FCB   3
D9A0 D8DA , FDB   M$PULL
D9A2 D979 , FDB   M$LT
             FDB   M$SHIO

D9A4 03 , ATM   M$PULL,M$RT,M$SWOO
D9A5 D8BA , FCB   3
D9A7 D8DE , FDB   M$PULL
D9A9 D97E , FDB   M$RT
             FDB   M$SWOO

D9AB 01 , ATM   M$MOVE
D9AC D8B6 , FCB   1
             FDB   M$MOVE

D9AD 01 , ATM   M$MOVE
              FCB   1

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
TOKEN - ToKen Tables

PAGE 21

```
D9AF D8B6      FDB    M$MOVE  
;  
D9B1 02      ATM    M$ATTK,M$RT  
D9B2 D895      FCB    2  
D9B4 D8DE      FDB    M$ATTK  
;  
D9B6 02      ATM    M$TURN,M$RT  
D9B7 D8C7      FCB    2  
D9B9 D8DE      FDB    M$TURN  
;  
D9B8 01      ATM    M$MOVE  
D9BC D8B6      FCB    1  
;  
D9BE 01      ATM    M$MOVE  
D9BF D8B6      FCB    1  
;  
D9C1 01      ATM    M$MOVE  
D9C2 D8B6      FCB    1  
;  
D9C4 02      ATM    M$TURN,M$RT  
D9C5 D8C7      FCB    2  
D9C7 D8DE      FDB    M$TURN  
;  
D9C9 01      ATM    M$MOVE  
D9CA D8B6      FCB    1  
;  
D9CC 01      ATM    M$MOVE  
D9CD D8B6      FCB    1  
D9CF EQU      EQU    $  
D9CF FF      FCB    -1  
;  
END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 22

```
INCLUD B:DTABAS.ASM
!!!!!!DATABASE DEFINITIONS!!!!!!
NAM DTABAS
LIBR CD
XDEF DISPAT
XDEF CDBTAB,ODBTAB,FWDCRE,XXXTAB
XDEF OMXTAB,OMXEND,FWDOBJ,OBJWGT
*XREF T.RN11,T.RN12,T.RN13,T.RN15
*PSCT
*IFUD DEBFLG
DEBFLG EQU 0
ENDC
```

```
/*
 * CMDXXX: Command Definition Macro
 *
 * Arguments:
 *   0) Command String
 *   1) 4 letter abbreviation
 *   2) Zero
 *   3) Service Routine
 *
CMDXXX MACR
  \0 ATTACK,ATTK,0,PATTK
  \0 CLIMB,CLIM,0,PCLIMB
  \0 DROP,DROP,0,PDROP
  \0 EXAMINE,EXAM,0,PEXAM
  \0 GET,GET,0,PGET
  \0 INCANT,INCN,0,PINCN
  \0 LOOK,LOOK,0,PLLOOK
  \0 MOVE,MOVE,0,PMOVE
  \0 PULL,PULL,0,PPULL
  \0 REVEAL,REVE,0,PREVEA
  \0 STOW,STOW,0,PSSTOW
  \0 TURN,TURN,0,PTURN
  \0 USE,USE,0,PUSE
  \0 ZLOAD,ZLOA,0,PZLOAD
  \0 ZSAVE,ZSAV,0,PZSAVE
ENDM
*
 * Debugging Commands Macro
 *
 * Arguments:
 *   0) Command String
 *   1) 4-letter abbreviation
 *   2) Zero
 *   3) Service Routine
 *
DEBXXX MACR
  \0 QMAP,QMAP,0,DEBMAP
ENDM
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 24

```
/*
** Direction Token Macro
**
** Arguments:
**   0) Command String
**   1) 4 letter abbreviation
**   2) Zero
**
DIRXXX MACR
  \0    LEFT,LT,0
  \0    RIGHT,RT,0
  \0    BACKWARD,BAK,0
  \0    AROUND,AROU,0
  \0    UP,UP,0
  \0    DOWN,IN,0
ENDM
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

BAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
ITABAS - Database Definitions

PAGE 25

```
/*
 * GENXXX: General Object Definition Macro
 *
 * Arguments:
 *   0) Object Name
 *   1) 4-letter abbreviation
 *   2) Class
 *   3) Forward Looking Vector List
 *   4) Look Down Vector List
 *   5) Object Weight
 *
GENXXX MACR
  \0     FLASK,FLAO,K,FLAS,FFLASK,DFLASK,5
  \0     RING,RINO,K,RING,FRING,DRING,1
  \0     SCROLL,SCRO,K,SCRO,FSCROL,DSCROL,10
  \0     SHIELD,SHIO,K,SHIE,FSHIEL,DSHIEL,25
  \0     SWORD,SWOO,K,SWOR,FSWORD,DSWORD,25
  \0     TORCH,TORO,K,TORC,FTORCH,DTORCH,10
ENDM
```

```

/*
* OBJXXX: Object Definition Macro
*
* Arguments:
*   0) Object Name
*   1) 4-letter Abbreviation
*   2) Generic Class
*   3) Reveled Power Requirements
*   4) Magic Offense
*   5) Physical Offense
*   6) Initial Level where object shows up
*   7) Number of objects
*
* Arguments (8-A) if present will generate an entry
* in XXXTAB (special object parameters).
0000    LVL0    EQU    0
0001    LVL1    EQU    1
0002    LVL2    EQU    2
0003    LVL3    EQU    3
0004    LVL4    EQU    4
0005    LVL5    EQU    5
*
OBJXXX MACR
*
* Level-4 Objects
*   \0      SUPREME,RN05,K.RING,255,0,5,LVL4,1,3,T.RN15,0
*
* Level-3 Objects
*   \0      JOULE,RN01,K.RING,170,0,5,LVL3,1,3,T.RN11,0
*   \0      ELVISH,SW01,K.SWOR,150,64,64,LVL3,1
*   \0      MITHRIL,SHI1,K.SHIE,140,13,26,LVL3,2,64,64,0
*
* Level-2 Objects
*   \0      SEER,SCR1,K.SCRO,130,0,5,LVL2,3
*   \0      THEWS,FLA1,K.FLAG,70,0,5,LVL2,3
*
* Level-1 Objects
*   \0      HOTH,RN03,K.RING,52,0,5,LVL1,1,3,T.RN13,0
*   \0      VISION,SCR2,K.SCRO,50,0,5,LVL1,3
*   \0      ABYE,FLA3,K.FLAG,48,0,5,LVL1,6
*   \0      HALE,FLA2,K.FLAG,40,0,5,LVL1,4
*   \0      SOLAR,TOR1,K.TORC,70,0,5,LVL1,4,60,13,11
*   \0      BRONZE,SHI3,K.SHIE,25,0,26,LVL1,6,96,128,0
*
* Level-0 Objects
*   \0      VULCAN,RN02,K.RING,13,0,5,LVL0,1,3,T.RN12,0
*   \0      IRON,SW02,K.SWOR,13,0,40,LVL0,4
*   \0      LUNAR,TOR2,K.TORC,25,0,5,LVL0,8,30,10,4
*   \0      FINE,TOR4,K.TORC,5,0,5,LVL0,8,15,7,0

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 27

```
;*      \0      WOODEN,SW03,K,SW0R,5,0,16,LVL0,4  
;      ENDM
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 28

```
/*
 * SPCXXX: Special Object Definition Macro
 *
 * Arguments:
 *   0) Object Name
 *   1) 4-letter Abbreviation
 *   2) Generic Class
 *   3) Reveal Power Requirements (zero)
 *   4) Magic Offense
 *   5) Physical Offense
 *
 * SPCXXX MACR
 *   \0     FINAL,RN15,K,RING,0,0,0
 *   \0     ENERGY,RN11,K,RING,0,255,255
 *   \0     ICE,RN13,K,RING,0,255,255
 *   \0     FIRE,RN12,K,RING,0,255,255
 *   \0     GOLD,RN20,K,RING,0,0,5
 *   \0     EMPTY,FLA4,K,FLAS,0,0,5
 *   \0     DEAD,TOR5,K,TORC,5,0,5,0,0,0,0,0
 ENDM
```

```
/*
** Creature Definition Macro
**
** Arguments:
**   0) Vector List
**   1) Movement Delay Time
**   2) Attack Delay Time
**   3) Magic Offense
**   4) Magic Defense
**   5) Physical Offense
**   6) Physical Defense
**   7) Hit Points
**
CREXXX MACR
  \0     SPIDER,23,11,0,255,128,255,32
  \0     VIPER,15,7,0,255,80,128,56
  \0     SGINT1,29,23,0,255,52,192,200
  \0     BLOB,31,31,0,255,96,167,304
  \0     KNIGHT1,13,7,0,128,96,60,504
  \0     SGINT2,17,13,0,128,128,48,704
  \0     SCORP,5,4,255,128,255,128,400
  \0     KNIGHT2,13,7,0,64,255,8,800
  \0     WRAITH,3,3,192,16,192,8,800
  \0     BALROG,4,3,255,5,255,3,1000
  \0     WIZO,13,7,255,6,255,0,1000
  \0     WIZI,13,7,255,6,255,0,8000
ENDM
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 30

```
*  
* DISPAT: Command Dispatch Macro  
*  
CDM    MACR  
XREF  \3  
FIRB  \3  
ENDM  
  
D9D0      DISPAT EQU $  
          CMDXXX CDM  
D9D0  D2B8  FDB  PATTK  
D9D2  D44E  FDB  PCCLIMB  
D9D4  D550  FDB  PDROP  
D9D6  D481  FDB  PEXAM  
D9D8  D522  FDB  PGET  
D9DA  D5BC  FDB  PINCAN  
D9DC  C751  FDB  PLLOOK  
D9DE  D6D0  FDB  PMOVE  
D9E0  D589  FDB  PPULL  
D9E2  D623  FDB  PREVEA  
D9E4  D579  FDB  PSTOW  
D9E6  D63F  FDB  PTURN  
D9E8  D741  FDB  PUSE  
D9EA  D7B7  FDB  PZLOAD  
D9EC  D7C7  FDB  PZSAVE  
  
D9EE      IF      DEBFLG  
          FDB  DEBMAP  
ENDIF
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 31

```
** FWDOBJ: Object Forward Looking Vector Lists
** MACR XREF \3
FLVL   MACR XREF \3
      FIR   \3
      ENDM

D9EE    FWDOBJ EQU   $ GENXXX FLVL
        ; FDB   FFLASK
D9F0    DC19  FDB   FRING
D9F1    DC21  FDB   FSCROL
D9F2    DC2A  FDB   FSHIEL
D9F4    DBFA  FDB   FSWORD
D9F6    DC0F  FDB   FTORCH
D9F8    DC07  FDB   FTORCH
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 32

```
;;
;; OBJWGT: Object Weight Table
;;
;WGT    MACR
;      FCB    \5
;      ENDM
;
;D9FA    OBJWGT EQU   $    GENXXX WGT
;          FCB    5      flask
;D9FB    05      FCB    1      ring
;D9FC    01      FCB    10     scroll
;D9FD    19      FCB    25     shield
;D9FE    19      FCB    25     sword
;D9FF    0A      FCB    10     torch
```

```

/*
 * ODBTAB: Object Definition Blocks
 */
ODB    MACR
      FCB    \2,\3
      FCB    \4,\5
      ENDM

DA00   * ODBTAB EQU   $
      OBJXXX ODB
DA00 01FF0005 ; FCB    K.RING,255,0,5      ;Supreme Ring
DA04 01AA0005 ; FCB    K.RING,170,0,5      ;Joule Ring
DA08 04964040 ; FCB    K.SWOR,150,64,64    ;Elvish Sword
DA0C 038C001A ; FCB    K.SHIE,140,13,26    ;Mithril Shield
DA10 02820005 ;
DA14 00460005 ; FCB    K.SCRO,130,0,5      ;Seer Scroll
                  FCB    K.FLAS,70,0,5      ;Thews Flask
DA18 01340005 ;
DA1C 02320005 ; FCB    K.RING,52,0,5      ;Hoth Ring
DA20 00300005 ; FCB    K.SCRO,50,0,5      ;Vision Scroll
DA24 00280005 ; FCB    K.FLAS,48,0,5      ;Abye Flask
DA28 05460005 ; FCB    K.FLAS,40,0,5      ;Hale Flask
DA2C 0319001A ; FCB    K.TORC,70,0,5      ;Solar Torch
                  FCB    K.SHIE,25,0,5,6    ;Bronze Shield
DA30 010D0005 ;
DA34 04000028 ; FCB    K.RING,13,0,5      ;Vulcan Ring
                  FCB    K.SWOR,13,0,40    ;Iron Sword
DA38 05190005 ; FCB    K.TORC,25,0,5      ;Lunar Torch
DA3C 05050005 ; FCB    K.TORC,5,0,5       ;Pine Torch
DA40 0305000A ; FCB    K.SHIE,5,0,10     ;Leather Shield
DA44 04050010 ; FCB    K.SWOR,5,0,16     ;Wooden Sword
                  SPCXXX ODR
DA48 01000000 ; FCB    K.RING,0,0,0      ;Final Ring
DA4C 0100FFFF ; FCB    K.RING,0,255,255    ;Energy Ring
DA50 0100FFFF ; FCB    K.RING,0,255,255    ;Ice Ring
DA54 0100FFFF ; FCB    K.RING,0,255,255    ;Fire Ring
DA58 01000005 ; FCB    K.RING,0,0,0,5     ;Gold Ring
DA5C 00000005 ; FCB    K.FLAS,0,0,0,5     ;Empty Flask
DA60 05050005 ; FCB    K.TORC,5,0,5      ;Dead Torch

```

```

/*
** XXXTAB: Special Parameter Tables
**
** For Torches:
**   .OCXXX+0 ==> timer value
**   .OCXXX+1 ==> regular lighting value
**   .OCXXX+2 ==> magical lighting value
**
** For Shields:
**   .OCXXX+0 ==> Magical Defense Filter
**   .OCXXX+1 ==> Physical Defense Filter
**   .OCXXX+2 ==> unused
**
*** XXX MACR
IFGT (NARG=8)
IFUD T.\1
  XREF T.\1
ENDC
  FCB    T.\1
  FCB    \8,\9,\A
ENDC
ENDM

DA64  XXXTAB EQU   $
OBJXXX XXX
      FCB    T.RN05,3,T.RN15,0 ;Supreme/Final Ring
      FCB    T.RN01,3,T.RN11,0 ;Joule/Energy Ring
      FCB    T.SHI1,64,64,0  ;Mithril Shield
      FCB    T.RN03,3,T.RN13,0 ;Hoth/Ice Ring
      FCB    T.TOR1,60,13,11 ;Solar Torch
      FCB    T.SHI3,96,128,0 ;Bronze Shield
      FCB    T.RN02,3,T.RN12,0 ;Vulcan/Fire Ring
      FCB    T.TOR2,30,10,4  ;Lunar Torch
      FCB    T.TOR4,15,7,0  ;Pine Torch
      FCB    T.SHI4,108,128,0 ;Leather Shield

DABC  SPCXXX XXX
      FCB    T.TOR5,0,0,0 ;Dead Torch
      FCB    -1           ;table terminator
DA90 FF

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 35

```

/*
** OMXTAB: Object Distribution Table
**
** Bit7-4: Object Level Number
** Bit3-0: Number of objects
*/
OMX MACR
LH SET (\$61,\$0F)!<4
RH SET (\$71,\$0F)
FCB LHI+RH
ENDM

DA91 OMXTAB EQU $OBJXXX
DA91 41 FCB ((LVL4 AND \$0F)*16) OR (1 AND \$0F)
DA92 31 FCB ((LVL3 AND \$0F)*16) OR (1 AND \$0F)
DA93 31 FCB ((LVL3 AND \$0F)*16) OR (1 AND \$0F)
DA94 32 FCB ((LVL3 AND \$0F)*16) OR (2 AND \$0F)
DA95 23 FCB ((LVL2 AND \$0F)*16) OR (3 AND \$0F)
DA96 23 FCB ((LVL2 AND \$0F)*16) OR (3 AND \$0F)
DA97 11 FCB ((LVL1 AND \$0F)*16) OR (1 AND \$0F)
DA98 13 FCB ((LVL1 AND \$0F)*16) OR (3 AND \$0F)
DA99 16 FCB ((LVL1 AND \$0F)*16) OR (6 AND \$0F)
DA9A 14 FCB ((LVL1 AND \$0F)*16) OR (4 AND \$0F)
DA9B 14 FCB ((LVL1 AND \$0F)*16) OR (4 AND \$0F)
DA9C 16 FCB ((LVL1 AND \$0F)*16) OR (6 AND \$0F)
DA9D 01 FCB ((LVLO AND \$0F)*16) OR (1 AND \$0F)
DA9E 04 FCB ((LVLO AND \$0F)*16) OR (4 AND \$0F)
DA9F 08 FCB ((LVLO AND \$0F)*16) OR (8 AND \$0F)
DAA0 08 FCB ((LVLO AND \$0F)*16) OR (8 AND \$0F)
DAA1 03 FCB ((LVLO AND \$0F)*16) OR (3 AND \$0F)
DAA2 04 FCB ((LVLO AND \$0F)*16) OR (4 AND \$0F)

DAA3 OMXEND EQU $

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 36

```
/*  
 * FWDCRE: Forward Looking Creature Vector Lists
```

```
CVL    MACR  
      XREF  \0  
      FDB   \0  
      ENDM
```

DAA3	*	FWDCRE	EQU	\$
			CREXXX	CVL
DAA3	DE26		FDB	SPIDER
DAA5	DFCA		FDB	VIPER
DAA7	DD41		FDB	SGINT1
DAA9	DE59		FDB	BLOB
DAA8	DE82		FDB	KNIGT1
DAA0	DD51		FDB	SGINT2
DAAF	DE3F		FDB	SCORP
DAB1	DE9D		FDB	KNIGT2
DAB3	DE07		FDB	WRAITH
DAB5	DOA3		FDB	BALROG
DAB7	DF65		FDB	WIZO
DAB9	DF10		FDB	WIZI

```

/*
** CDBTAB: Creature Definition Blocks
*/
CIB    MACR
      FDB    \7          ;power
      FCB    \3,\4        ;magical offense/defense
      FCB    \5,\6        ;physical offense/defense
      FCB    \1,\2        ;movement/attack delay times
      ENDM

DABB   * CDBTAB EQU $CREXXX CDB
DABB 0020 ; FDB 32           ;spider
DABB 00FF ; FCB 0,255
DABF 80FF ; FCB 128,255
DAC1 170B ; FCB 23,11

DAC3 0038 ; FDB 56           ;viper
DAC5 00FF ; FCB 0,255
DAC7 5080 ; FCB 80,128
DAC9 0F07 ; FCB 15,7

DACE 00C8 ; FDB 200          ;stone giant 1
DACE 00FF ; FCB 0,255
DACE 34C0 ; FCB 52,192
DAD1 1D17 ; FCB 29,23

DAD3 0130 ; FDB 304          ;blob
DAD5 00FF ; FCB 0,255
DAD7 60A7 ; FCB 96,167
DAD9 1F1F ; FCB 31,31

DADB 01F8 ; FDB 504          ;Knight 1
DADD 0080 ; FCB 0,128
DADE 603C ; FCB 96,60
DAE1 0D07 ; FCB 13,7

DAE3 02C0 ; FDB 704          ;stone giant 2
DAE5 0080 ; FCB 0,128
DAE7 8030 ; FCB 128,48
DAE9 110D ; FCB 17,13

DAEB 0190 ; FDB 400          ;scorpion
DAED FF80 ; FCB 255,128
DAEF FF80 ; FCB 255,128
DMF1 0504 ; FCB 5,4

DAF3 0320 ; FDB 800          ;Knight 2
DAF5 0040 ; FCB 0,64
DAF7 FF08 ; FCB 255,8
DAF9 0D07 ; FCB 13,7

DAFB 0320 ; FDB 800          ;wraith
DAFD C010 ; FCB 192,16
DAFF C008 ; FCB 192,8
DAB1 07A7 ; FCB 7,7

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
DTABAS - Database Definitions

PAGE 38

```
DB03 03E8      ;          FDB    1000          ;balrog
DB05 FF05      ;          FCB    255,5
DB07 FF03      ;          FCB    255,3
DB09 0403      ;          FCB    4,3
;
DB0B 03E8      ;          FDB    1000          ;wiz0
DB0D FF06      ;          FCB    255,6
DB0F FF00      ;          FCB    255,0
DB11 0B07      ;          FCB    13,7
;
DB13 1F40      ;          FDB    8000          ;wiz1
DB15 FF06      ;          FCB    255,6
DB17 FF00      ;          FCB    255,0
DB19 0B07      ;          FCB    13,7
;
END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 39

INCLUD B:SWCHAR.ASM

!!!!!!!!!!!!!!  
S O F T W A R E C H A R A C T E R T A B L E  
!!!!!!!!!!!!!!

Entries in the software character tables are 5 bits  
wide and 7 bytes long:

Byte 0 ...XXXXX  
Byte 1 ...XXXXX  
Byte 2 ...XXXXX  
Byte 3 ...XXXXX  
Byte 4 ...XXXXX  
Byte 5 ...XXXXX  
Byte 6 ...XXXXX

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 40

DB1B	SWCTAB	EQU	\$	; where the tables begin
DB1B 30	;	FCB	Z00110000	;* Space = \$00
DB1C 00	;	FCB	Z00000000	;* Space = \$00
DB1D 00	;	FCB	Z00000000	;* Space = \$00
DB1E 00	;	FCB	Z00000000	;* Space = \$00
DB1F 00	;	FCB	Z00000000	;* Space = \$00
DB20 31	;	FCB	Z00110001	;* Letter A = \$01
DB21 15	;	FCB	Z00010101	;* Letter A = \$01
DB22 18	;	FCB	Z00011000	;* Letter A = \$01
DB23 FE	;	FCB	Z11111110	;* Letter A = \$01
DB24 31	;	FCB	Z00110001	;* Letter A = \$01
DB25 37	;	FCB	Z00110111	;* Letter B = \$02
DB26 A3	;	FCB	Z10100011	;* Letter B = \$02
DB27 1F	;	FCB	Z00011111	;* Letter B = \$02
DB28 46	;	FCB	Z01000010	;* Letter B = \$02
DB29 3E	;	FCB	Z00111110	;* Letter B = \$02
DB2A 33	;	FCB	Z00110011	;* Letter C = \$03
DB2B A3	;	FCB	Z10100011	;* Letter C = \$03
DB2C 08	;	FCB	Z00001000	;* Letter C = \$03
DB2D 42	;	FCB	Z01000010	;* Letter C = \$03
DB2E 2E	;	FCB	Z00101110	;* Letter C = \$03
DB2F 37	;	FCB	Z00110111	;* Letter D = \$04
DB30 A3	;	FCB	Z10100011	;* Letter D = \$04
DB31 18	;	FCB	Z00011000	;* Letter D = \$04
DB32 C6	;	FCB	Z11000010	;* Letter D = \$04
DB33 3E	;	FCB	Z00111110	;* Letter D = \$04
DB34 37	;	FCB	Z00110111	;* Letter E = \$05
DB35 E1	;	FCB	Z11100001	;* Letter E = \$05
DB36 0F	;	FCB	Z00000111	;* Letter E = \$05
DB37 42	;	FCB	Z01000010	;* Letter E = \$05
DB38 1F	;	FCB	Z00011111	;* Letter E = \$05
DB39 37	;	FCB	Z00110111	;* Letter F = \$06
DB3A E1	;	FCB	Z11100001	;* Letter F = \$06
DB3B 0F	;	FCB	Z00000111	;* Letter F = \$06
DB3C 42	;	FCB	Z01000010	;* Letter F = \$06
DB3D 10	;	FCB	Z00010000	;* Letter F = \$06
DB3E 33	;	FCB	Z00110011	;* Letter G = \$07
DB3F E3	;	FCB	Z11100011	;* Letter G = \$07
DB40 08	;	FCB	Z00001000	;* Letter G = \$07
DB41 4E	;	FCB	Z01001110	;* Letter G = \$07
DB42 2F	;	FCB	Z00101111	;* Letter G = \$07
DB43 34	;	FCB	Z00110100	;* Letter H = \$08
DB44 63	;	FCB	Z01100011	;* Letter H = \$08
DB45 1F	;	FCB	Z00011111	;* Letter H = \$08
DB46 C6	;	FCB	Z11000010	;* Letter H = \$08
DB47 31	;	FCB	Z00110001	;* Letter H = \$08

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 41

DB48 33		FCB	Z00110011	;* Letter I	= \$09
DB49 88		FCB	Z10001000	;* Letter I	= \$09
DB4A 42		FCB	Z01000010	;* Letter I	= \$09
DB4B 10		FCB	Z00010000	;* Letter I	= \$09
DB4C 8E		FCB	Z10001110	;* Letter I	= \$09
;					
DB4D 30		FCB	Z00110000	;* Letter J	= \$0A
DB4E 42		FCB	Z01000010	;* Letter J	= \$0A
DB4F 10		FCB	Z00010000	;* Letter J	= \$0A
DB50 86		FCB	Z10000110	;* Letter J	= \$0A
DB51 2E		FCB	Z00101110	;* Letter J	= \$0A
;					
DB52 34		FCB	Z00110100	;* Letter K	= \$0B
DB53 65		FCB	Z01100101	;* Letter K	= \$0B
DB54 4C		FCB	Z01001100	;* Letter K	= \$0B
DB55 52		FCB	Z01010010	;* Letter K	= \$0B
DB56 51		FCB	Z01010001	;* Letter K	= \$0B
;					
DB57 34		FCB	Z00110100	;* Letter L	= \$0C
DB58 21		FCB	Z00100001	;* Letter L	= \$0C
DB59 08		FCB	Z00001000	;* Letter L	= \$0C
DB5A 42		FCB	Z01000010	;* Letter L	= \$0C
DB5B 1F		FCB	Z00011111	;* Letter L	= \$0C
;					
DB5C 34		FCB	Z00110100	;* Letter M	= \$0D
DB5D 77		FCB	Z01101011	;* Letter M	= \$0D
DB5E 5A		FCB	Z01011010	;* Letter M	= \$0D
DB5F D6		FCB	Z11010110	;* Letter M	= \$0D
DB60 31		FCB	Z00110001	;* Letter M	= \$0D
;					
DB61 34		FCB	Z00110100	;* Letter N	= \$0E
DB62 63		FCB	Z01100011	;* Letter N	= \$0E
DB63 9A		FCB	Z10011010	;* Letter N	= \$0E
DB64 CE		FCB	Z11001110	;* Letter N	= \$0E
DB65 31		FCB	Z00110001	;* Letter N	= \$0E
;					
DB66 33		FCB	Z00110011	;* Letter O	= \$0F
DB67 A3		FCB	Z10100011	;* Letter O	= \$0F
DB68 18		FCB	Z00001100	;* Letter O	= \$0F
DB69 C6		FCB	Z11000010	;* Letter O	= \$0F
DB6A 2E		FCB	Z00101110	;* Letter O	= \$0F
;					
DB6B 37		FCB	Z00110111	;* Letter P	= \$10
DB6C A3		FCB	Z10100011	;* Letter P	= \$10
DB6D 1F		FCB	Z00011111	;* Letter P	= \$10
DB6E 42		FCB	Z01000010	;* Letter P	= \$10
DB6F 10		FCB	Z00010000	;* Letter P	= \$10
;					
DB70 33		FCB	Z00110011	;* Letter Q	= \$11
DB71 A3		FCB	Z10100011	;* Letter Q	= \$11
DB72 18		FCB	Z00001100	;* Letter Q	= \$11
DB73 D6		FCB	Z11010110	;* Letter Q	= \$11
DB74 4D		FCB	Z01001101	;* Letter Q	= \$11
;					
DB75 37		FCB	Z00110111	;* Letter R	= \$12
DB76 A7		FCB	Z10100011	;* Letter R	= \$12

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 42

DB77	1F	FCB	Z00011111	* Letter R	= \$12
DB78	52	FCB	Z01010010	* Letter R	= \$12
DB79	51	FCB	Z01010001	* Letter R	= \$12
DB7A	33	FCB	Z00110011	* Letter S	= \$13
DB7B	A3	FCB	Z10100011	* Letter S	= \$13
DB7C	07	FCB	Z00000111	* Letter S	= \$13
DB7D	06	FCB	Z00000110	* Letter S	= \$13
DB7E	2E	FCB	Z00101110	* Letter S	= \$13
DB7F	37	FCB	Z00110111	* Letter T	= \$14
DB80	EA	FCB	Z11101010	* Letter T	= \$14
DB81	42	FCB	Z01000010	* Letter T	= \$14
DB82	10	FCB	Z00010000	* Letter T	= \$14
DB83	84	FCB	Z10000100	* Letter T	= \$14
DB84	34	FCB	Z00110100	* Letter U	= \$15
DB85	63	FCB	Z01100011	* Letter U	= \$15
DB86	18	FCB	Z00011000	* Letter U	= \$15
DB87	C6	FCB	Z11000110	* Letter U	= \$15
DB88	2E	FCB	Z00101110	* Letter U	= \$15
DB89	34	FCB	Z00110100	* Letter V	= \$16
DB8A	63	FCB	Z01100011	* Letter V	= \$16
DB8B	15	FCB	Z00010101	* Letter V	= \$16
DB8C	28	FCB	Z00101000	* Letter V	= \$16
DB8D	84	FCB	Z10000100	* Letter V	= \$16
DB8E	34	FCB	Z00110100	* Letter W	= \$17
DB8F	63	FCB	Z01100011	* Letter W	= \$17
DB8G	1A	FCB	Z00011010	* Letter W	= \$17
DB8H	D7	FCB	Z11010111	* Letter W	= \$17
DB8I	71	FCB	Z01110001	* Letter W	= \$17
DB8J	34	FCB	Z00110100	* Letter X	= \$18
DB8K	62	FCB	Z01100010	* Letter X	= \$18
DB8L	A2	FCB	Z10100010	* Letter X	= \$18
DB8M	2A	FCB	Z00101010	* Letter X	= \$18
DB8N	31	FCB	Z00110001	* Letter X	= \$18
DB8O	34	FCB	Z00110100	* Letter Y	= \$19
DB8P	62	FCB	Z01100010	* Letter Y	= \$19
DB8Q	A2	FCB	Z10100010	* Letter Y	= \$19
DB8R	10	FCB	Z00010000	* Letter Y	= \$19
DB8S	84	FCB	Z100000100	* Letter Y	= \$19
DB8T	37	FCB	Z00110111	* Letter Z	= \$1A
DB8U	C2	FCB	Z11000010	* Letter Z	= \$1A
DB8V	22	FCB	Z00100010	* Letter Z	= \$1A
DB8W	22	FCB	Z00100010	* Letter Z	= \$1A
DB8X	1F	FCB	Z00011111	* Letter Z	= \$1A
DB8Y	31	FCB	Z00110001	* Exclamation point	Code = \$1B
DB8Z	08	FCB	Z00001000	* Exclamation point	Code = \$1B
DB8A	42	FCB	Z01000010	* Exclamation point	Code = \$1B
DB8B	10	FCB	Z00010000	* Exclamation point	Code = \$1B

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 43

DBA6	04	;	FCB	Z00000100	;* Exclamation point	Code = \$1B
DBA7	30	;	FCB	Z00110000	;* Underline	= \$1C
DBA8	00	;	FCB	Z00000000	;* Underline	= \$1C
DBA9	00	;	FCB	Z00000000	;* Underline	= \$1C
DBAA	00	;	FCB	Z00000000	;* Underline	= \$1C
DBAB	1F	;	FCB	Z00011111	;* Underline	= \$1C
DBAC	33	;	FCB	Z00110011	;* Question Mark	= \$1D
DBAD	A2	;	FCB	Z10100010	;* Question Mark	= \$1D
DBAE	13	;	FCB	Z00010011	;* Question Mark	= \$1D
DBAF	10	;	FCB	Z00010000	;* Question Mark	= \$1D
DBB0	04	;	FCB	Z00000100	;* Question Mark	= \$1D
DBB1	30	;	FCB	Z00110000	;* Period	= \$1E
DBB2	00	;	FCB	Z00000000	;* Period	= \$1E
DBB3	00	;	FCB	Z00000000	;* Period	= \$1E
DBB4	00	;	FCB	Z00000000	;* Period	= \$1E
DBB5	04	;	FCB	Z00000100	;* Period	= \$1E

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 44

!!!!!!  
S P E C I A L   C H A R A C T E R   T A B L E  
!!!!!!

The following table entries contain bit patterns  
used in making up special figures.

DBB6      SPCTAB EQU    \$  
              Small Heart Left    Code = \$20

DBB6 00      FCB    Z00000000  
DBB7 00      FCB    Z00000000  
DBB8 01      FCB    Z00000001  
DBB9 01      FCB    Z00000001  
DBBA 00      FCB    Z00000000  
DBBB 00      FCB    Z00000000  
DBBC 00      FCB    Z00000000

              Small Heart Right    Code = \$21

DBBD 00      FCB    Z00000000  
DBBE A0      FCB    Z10100000  
DBBF F0      FCB    Z11110000  
DBC0 F0      FCB    Z11110000  
DBC1 E0      FCB    Z11100000  
DBC2 40      FCB    Z01000000  
DBC3 00      FCB    Z00000000

              Large Heart Left    Code = \$22

DBC4 00      FCB    Z00000000  
DBC5 01      FCB    Z00000001  
DBC6 03      FCB    Z00000011  
DBC7 03      FCB    Z00000011  
DBC8 01      FCB    Z00000001  
DBC9 00      FCB    Z00000000  
DBCA 00      FCB    Z00000000

              Large Heart Right    Code = \$23

DBCB 00      FCB    Z00000000  
DBCC B0      FCB    Z10110000  
DBCD F8      FCB    Z11111000  
DBCE F8      FCB    Z11111000  
DBCF E0      FCB    Z11100000  
DBD0 E0      FCB    Z01000000  
DBD1 40      FCB    Z01000000

              END

DBD2 0080      THUDD    FDB    \$0080  
DBD4 0001      FDB    \$0001

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
SWCHAR - Software Character Table

PAGE 45

DBE6 0050		FDB	\$0050	
DBE8 0004		FDB	\$0004	
DBE4 0050	BANGD	FDB	\$0050	
DBE5 0005		FDB	\$0005	
DBDE	FLATAB	EQU	\$	;beginning of table
DBE6 03		FCB	3	
DBDF DC4F		FDB	LPASAG	;left passage
DBE4 DC6B		FDB	LDOOR	;door
DBE3 DC9B		FDB	LSDOOR	;secret door
DBE5 DC33		FDB	LWALL	;wall
DBE7 00		FCB	0	
DBE8 DC6A		FDB	FPASAG	;forward passage
DBEA DC8B		FDB	FDOOR	;door
DBEC DCA9		FDB	FSDOOR	;secret door
DBEE DC45		FDB	FWALL	;wall
DBF0 01		FCB	1	
DBF1 DC5D		FDB	RPASAG	;right passage
DBF3 DC7B		FDB	RD00R	;door
DBF5 DCA2		FDB	RSDOOR	;secret door
DBF7 DC3C		FDB	RWALL	;wall
DBF9 FF		FCB	-1	;end of table

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VOBJ - Object Vector Lists

PAGE 46

INCLUD B:VOBJ.ASM

\*\*\*\*\*  
OBJECT VECTOR LISTS  
\*\*\*\*\*NAM VOBJ  
LIBRY CDXDEF FSHIEL,FFLASK,FRING  
XDEF FSCROL,FWORD,FTORCH  
OPT NOMEX,NOCEN  
DSCT

Looking Forward - Shield

DBFA 86AC FSHIEL FCB 134,172 ;bottom left of shield  
DBFC 80C0 FCB 128,192  
DBFE 7ABA FCB 122,186DC00 80A8FC SVORG 128,168  
DC03 3E SVECT 134,164  
FCB 128,168,V\$REL  
FCB (((134-128)/2) AND \$0F)\*16) OR (((164-168)/2) AND \$0F)DC04 04 SVECT 134,172  
FCB (((((134-134)/2) AND \$0F)\*16) OR (((172-164)/2) AND \$0F))DC05 00FE SVEND ;end of vector list  
FCB V\$ABS,V\$END ;end of vector list

\*\*\*\*\*

Looking Forward - Torch

DC07 FTORCH SVORG 118,60 ;tip of torch  
SVECT 116,74DC07 763CFC FCB 118,60,V\$REL  
DC0A F7 FCB (((((116-118)/2) AND \$0F)\*16) OR (((74-60)/2) AND \$0F))DC0B FF SVECT 114,72  
FCB (((((114-116)/2) AND \$0F)\*16) OR (((72-74)/2) AND \$0F))DC0C 2A SVECT 118,60  
FCB (((((118-114)/2) AND \$0F)\*16) OR (((74-72)/2) AND \$0F))

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VOBJ - Object Vector Lists

PAGE 47

DC0D 00FE

SVEND ;end of vector list

FCB V\$ABS,V\$END

\* \* \* \* \*

Looking Forward - Sword

DC0F 7250

FSWORD FCB 114,80 ;base of handle  
DC11 7C64 FCB 124,100 ;tip of blade  
DC13 FF FCB V\$NEW

DC14 7452

DC16 7256

DC18 FE

FCB 118,82 ;hand guard  
FCB 114,86  
FCB V\$END ;end of vector list

\* \* \* \* \*

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VOBJ - Object Vector Lists

PAGE 48

\* \* \* \* \*

Looking Forward - Flask

DC19 FFLASK  
    SVORG 110,162 ;top of flask  
    SVECT 120,164  
  
DC19 6EA2FC  
DC1C 51     FCB 110,162,V\$REL  
          FCB (((((120-110)/2) AND \$0F)\*16) OR (((164-162)/2) AND \$0F)  
          SVECT 120,160  
  
DC1D 0E     FCB (((((120-120)/2) AND \$0F)\*16) OR (((160-164)/2) AND \$0F)  
          SVECT 110,162  
  
DC1E B1     FCB (((((110-120)/2) AND \$0F)\*16) OR (((162-160)/2) AND \$0F)  
          SVEND ;end of vector list  
  
DC1F 00FE    FCB V\$ABS,V\$END

\* \* \* \* \*

Looking Forward - Ring

DC21 FRING  
    SVORG 122,60 ;top of ring  
    SVECT 124,62  
  
DC21 7A3CFC  
DC24 11     FCB 122,60,V\$REL  
          FCB (((((124-122)/2) AND \$0F)\*16) OR (((62-60)/2) AND \$0F)  
          SVECT 126,60 ;  
  
DC25 1F     FCB (((((126-124)/2) AND \$0F)\*16) OR (((60-62)/2) AND \$0F)  
          SVECT 124,58 ;  
  
DC26 FF     FCB (((((124-126)/2) AND \$0F)\*16) OR (((58-60)/2) AND \$0F)  
          SVECT 122,60 ;  
  
DC27 F1     FCB (((((122-124)/2) AND \$0F)\*16) OR (((60-58)/2) AND \$0F)  
          SVEND ;

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VOBJ - Object Vector Lists

PAGE 49

```
DC28 00FE      FCB      V$ABS,V$END

* * * * * * * *
Looking Forward - Scroll
DC2A      FSCROL
          SVORG   118,194      ;top of scroll
          SVECT    120,192
DC2A 76C2FC      FCB      118,194,V$REL
DC2D 1F      FCB      (((((120-118)/2) AND $0F)*16) OR (((192-194)/2) AND $0F)

          SVECT    126,200
DC2E 34      FCB      (((((126-120)/2) AND $0F)*16) OR (((200-192)/2) AND $0F)

          SVECT    124,202
DC2F F1      FCB      (((((124-126)/2) AND $0F)*16) OR (((202-200)/2) AND $0F)

          SVECT    118,194
DC30 DC      FCB      (((((118-124)/2) AND $0F)*16) OR (((194-202)/2) AND $0F)

          SVEND      ;end of vector list
DC31 00FE      FCB      V$ABS,V$END

* * * * * * * *
END
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VARC - Architectural Vector Lists

PAGE 50

```
INCLUD B:VARC.ASM
!!!!!! CH I T E C T U R A L V E C T O R L I S T S !!!!!!!
NAM      VARC - Architectural Vector Lists
LIBRY    CD

XDEF    LPASAG,RPASAG,FPASAG
XDEF    LDOOR,RDOOR,FDOOR
XDEF    LSDOOR,RSDOOR,FSDOOR
XDEF    LWALL,RWALL,FWALL
XDEF    LPEEK,RPEEK
DSCT
```

Vector lists are stored as point pairs, with the Y coordinate appearing first: y<sub>1</sub>, x<sub>1</sub>, y<sub>2</sub>, x<sub>2</sub>, y<sub>3</sub>, x<sub>3</sub>, etc.

Special command codes can appear in place of the Y-coordinate (see COMDEF definitions).

When describing creature parts, left and right are from the creature's point of view.

When describing inanimate objects, left and right are from the player's point of view. (screen left and right).

When in offset-mode, each data byte is divided into 2 4-bit nybbles, each of which are 4-bit two's complement offsets which are each multiplied by 2 and added to the preceding point's coordinates to obtain the next y,x coordinate. The hi-order nybble is the y offset and the lo-order nybble is the x offset.

```
!!!!!! ARCHITECTURAL FEATURES !!!!!!!
!!!!!!
```

## Left Wall

```
DC33 101B LWALL FCB 16,27 ;top horizontal line
DC35 2640 FCB 38,64
DC37 7240 FCB 114,64 ;far vertical line
DC39 881B FCB 136,27 ;bottom horizontal line
DC3B FE FCB V$END ;end of this list
```

```
*****
```

## Right Wall

```
DC3C 10E5 RWALL FCB 16,229 ;top horizontal line
DC3E 26C0 FCB 38,192
DC40 72C0 FCB 114,192 ;far vertical line
DC42 88E5 FCB 136,229 ;bottom horizontal line
DC44 FE FCB V$END ;end of this list
```

```
*****
```

## Forward Wall

```
DC45 2640 FWALL FCB 38,64 ;top horizontal line
DC47 26C0 FCB 38,192
DC49 FF FCB V$NEW ;start new sequence of vectors
DC4A 7240 FCB 114,64 ;bottom horizontal line
DC4C 72C0 FCB 114,192
DC4E FE FCB V$END ;end of this list
```

```
*****
```

## Left Passage

```
DC4F 261B LPASAG FCB 38,29 ;top horizontal line
DC51 2640 FCB 38,64
DC53 7240 FCB 114,64 ;vertical line
DC55 721B FCB 114,27 ;bottom horizontal line
DC57 FF FCB V$NEW ;start new vector sequence
DC58 101B FCB 16,27 ;continuation of top wall line
DC5A 2640 FCB 38,64
DC5C FE FCB V$END ;end of this list
```

```
*****
```

## Right Passage

```
DC5D 26E5 RPASAG FCB 38,229 ;top horizontal line
DC5F 26C0 FCB 38,192
DC61 72C0 FCB 114,192 ;vertical line
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VARC - Architectural Vector Lists

PAGE 52

```

DC65 FF      FCB    V$NEW      ;start new vector sequence
DC66 10E5    FCB    16,229     ;continuation of top wall line
DC68 26C0    FCB    38,192     ;end of this list
DC6A FE      FPASAG   FCB    V$END
; * * * * * * * *
; Left Door
DC6B 8028    LDOOR   FCB    128,40     ;nearer vertical line
DC6D 4128    FCB    65,40
DC6F 4438    FCB    68,56
DC71 7738    FCB    119,56
DC73 FF      FCB    V$NEW      ;start new vector sequence
DC74 5C30    FCB    92,48
DC76 5D34    FCB    93,52     ;doorknob
DC78 FD      FCB    V$JMP      ;chain to another list
DC79 DC33    FDB    LWALL     ;superimpose left wall
; * * * * * * * *
; Right Door
DC7B 80D8    RDOOR   FCB    128,216    ;nearer vertical line
DC7D 41D8    FCB    65,216
DC7F 44C8    FCB    68,200
DC81 77C8    FCB    119,200
DC83 FF      FCB    V$NEW      ;start new vector sequence
DC84 5C00    FCB    92,208
DC86 5DCC    FCB    93,204
DC88 FD      FCB    V$JMP      ;chain to another list
DC89 DC3C    FDB    RWALL     ;superimpose right wall
; * * * * * * * *
; Forward Door
DC8B 726C    FDOOR   FCB    114,108    ;left side vertical line
DC8D 436C    FCB    67,108
DC8F 4394    FCB    67,148
DC91 7294    FCB    114,148
DC93 FF      FCB    V$NEW      ;start new vector sequence
DC94 5E7E    FCB    94,126
DC96 5E82    FCB    94,130     ;doorknob
DC98 FI      FCB    V$JMP      ;chain to another list
DC99 DC45    FDB    FWALL     ;superimpose forward wall
; * * * * * * * *

```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VARC - Architectural Vector Lists

PAGE 53

; Left Secret-Door

DC9B 8028	LSDOOR FCB 128,40	; nearer line
DC9D 4232	FCB 66,50	; apex of triangle
DC9F 753A	FCB 117,58	; farther line
DCA1 FE	FCB V\$END	; end of this list

; \* \* \* \* \*

; Right Secret-Door

DCA2 80D8	RSDOOR FCB 128,216	; nearer line
DCA4 42CE	FCB 66,206	; apex of triangle
DCA6 75C6	FCB 117,198	; farther line
DCA8 FE	FCB V\$END	; end of this list

; \* \* \* \* \*

; Forward Secret-Door

DCA9 716C	FSDOOR FCB 113,108	; left line
DCAE 4380	FCB 67,128	; apex of triangle
DCAD 7294	FCB 114,148	; right line
DCAF FE	FCB V\$END	; end of this list

;

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VARC - Architectural Vector Lists

PAGE 54

\* \* \* \* \*

Left Peek-a-boo shape vector list

DCB0            LPEEK  
               SVORG 100,28  
               SVECT 108,36  
  
DCB0 641CFC      FCB 100,28,V\$REL  
DCB3 44           FCB (((108-100)/2) AND \$0F)\*16) OR (((36-28)/2) AND \$0F)  
               SVECT 112,32  
  
DCB4 2E           FCB (((112-108)/2) AND \$0F)\*16) OR (((32-36)/2) AND \$0F)  
               SVECT 120,36  
  
DCB5 42           FCB (((120-112)/2) AND \$0F)\*16) OR (((36-32)/2) AND \$0F)  
               SVECT 128,28  
  
DCB6 4C           FCB (((128-120)/2) AND \$0F)\*16) OR (((28-36)/2) AND \$0F)  
               SVEND                          ;end of vector list  
  
DCB7 00FE          FCB V\$ABS,V\$END

\* \* \* \* \*

Right Peek-a-boo shape vector list

DCB9            RPEEK  
               SVORG 100,228  
               SVECT 108,220  
  
DCB9 64E4FC      FCB 100,228,V\$REL  
DCBC 4C           FCB (((108-100)/2) AND \$0F)\*16) OR (((220-228)/2) AND \$0F)  
               SVECT 112,224  
  
DCBD 22           FCB (((112-108)/2) AND \$0F)\*16) OR (((224-220)/2) AND \$0F)  
               SVECT 120,220  
  
DCBE 4E           FCB (((120-112)/2) AND \$0F)\*16) OR (((220-224)/2) AND \$0F)  
               SVECT 128,228

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VARC - Architectural Vector Lists

PAGE 55

DCEBF 44 : FCB (((((128-120)/2) AND \$0F)\*16) OR (((228-220)/2) AND \$0F)

: SVEND ;end of this vector list

DCC0 00FE : FCB V\$ABS,V\$END

, \* \* \* \* \* \* \*  
END

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VERT - Vertical Feature Vector Lists

PAGE 56

```
INCLUD B:VERT.ASM
;-----;
; V E R T I C A L   F E A T U R E   V E C T O R   L I S T S ;
;-----;
; NAM    VERT
; LIBRY   CD
; XDEF    FWDOVER,CELINE
; OPT     NOMEX,NOGEN
; DSCT.
```

; Forward Looking Features Table

DCC2 DDOE	FWDOVER	FDB	HOLEUP	; (0) VF.HUP
DCC4 DCCA		FDB	FLUP	; (1) VF.LUP
DCC6 DD2A		FDB	HOLEDN	; (2) VF.HDN
DCC8 DCDO		FDB	FLDN	; (3) VF.LDN

; Forward Looking Ladder Up Sequence

DCCA FB	FLUP	FCB	V\$JSR	; subroutine
DCCB DCD6		FDB	LADDER	; ladder
DCCD FD		FCB	V\$JMP	; jump to
DCCE DDOE		FDB	HOLEUP	; hole-up

; Forward Looking Ladder Down Sequence

DCDO FB	FLDN	FCB	V\$JSR	; subroutine
DCD1 DCD6		FDB	LADDER	; ladder
DCD3 FD		FCB	V\$JMP	; jump to
DCD4 DD2A		FDB	HOLEDN	; hole-down

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
VERT - Vertical Feature Vector Lists

PAGE 57

; Forward Looking Ladder Vector List Subroutine

DCD6 1874	LADDER	FCB	24,116	;upper left corner
DCD8 8074		FCB	128,116	;lower left corner
DCDA FF		FCB	V\$NEW	
DCDB 188C	;	FCB	24,140	;upper right corner
DCDD 808C		FCB	128,140	;lower right corner
DCDF FF		FCB	V\$NEW	
DCE0 1C74	;	FCB	28,116	;1st rung
DCE2 1C8C		FCB	28,140	
DCE4 FF		FCB	V\$NEW	
DCE5 2874	;	FCB	40,116	;2nd rung
DCE7 288C		FCB	40,140	
DCE9 FF		FCB	V\$NEW	
DCEA 3474	;	FCB	52,116	;3rd rung
DCEC 348C		FCB	52,140	
DCEE FF		FCB	V\$NEW	
DCEF 4074	;	FCB	64,116	;4th rung
DCF1 408C		FCB	64,140	
DCF3 FF		FCB	V\$NEW	
DCF4 4C74	;	FCB	76,116	;5th rung
DCF6 4C8C		FCB	76,140	
DCF8 FF		FCB	V\$NEW	
DCF9 5874	;	FCB	88,116	;6th rung
DCFB 588C		FCB	88,140	
DCFD FF		FCB	V\$NEW	
DCFE 6474	;	FCB	100,116	;7th rung
DD00 648C		FCB	100,140	
DD02 FF		FCB	V\$NEW	
DD03 7074	;	FCB	112,116	;8th rung
DD05 708C		FCB	112,140	
DD07 FF		FCB	V\$NEW	
DD08 7B74	;	FCB	123,116	;9th rung
DD0A 7B8C		FCB	123,140	
DD0C FF		FCB	V\$NEW	
DD0D FA	;	FCB	V\$RTS	;return

## ; Forward Looking Ceiling Hole Vector List

DD0E 2264	ROLEUP	FCB	34,100	;back left corner
DD10 185C		FCB	24,92	;front left corner
DD12 18A4		FCB	24,164	;nearer horizontal edge
DD14 229C		FCB	34,156	;right edge
DD16 2264		FCB	34,100	;farther horizontal edge
DD18 1864		FCB	24,100	;line up on left
DD1A FF		FCB	V\$NEW	;start new sequence of vectors
DD1B 229C		FCB	34,156	;farther right corner
DD1D 189C		FCB	24,156	;line up on right
DD1F FF		FCB	V\$NEW	;start new sequence of vectors
DD20 1C2F		FCB	28,47	;left end of left ceiling line
DD22 1C60		FCB	28,96	;right end of left ceiling line
DD24 FF		FCB	V\$NEW	;start new sequence of vectors
DD25 1CA1		FCB	28,161	;left end of right ceiling line
DD27 1CD2		FCB	28,210	;right end of right ceiling line
DD29 FE		FCB	V\$END	;done

## ; Forward Looking Floor Hole Vector List

DD2A 7664	HOLEDN	FCB	118,100	;back left corner
DD2C 805C		FCB	128,92	;front left corner
DD2E 80A4		FCB	128,164	;nearer horizontal edge
DD30 769C		FCB	118,156	;right edge
DD32 7664		FCB	118,100	;farther edge

DD34 8064		FCB	128,100	;line down on left
DD36 FF		FCB	V\$NEW	;start new sequence of vectors

DD37 769C		FCB	118,156	;farther right corner
DD39 809C		FCB	128,156	;line down on right
DD3B FF		FCB	V\$NEW	;fall into CELINE

## ; Forward "Passage" (line across ceiling)

DD3C 1C2F	CELINE	FCB	28,47	;line across ceiling
DD3E 1CD2		FCB	28,210	
DD40 FE	VNULL	FCB	V\$END	;end of vector list

;

END

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.

B3 - Creature Vector List (I)

PAGE 60

```

INCLUD E:B3.ASM
!!!!!! CREATURE VECTORS LISTS (I) !!!!!!
NAM B3 - Creature Vector Lists (I)
LIBRY CD
XDEF SGINT1,SGINT2,BALROG,BLOB
XDEF WRAITH,SPIDER,SCORP
OPT NOMEX,NUGEN
DSCT

Stone Giant 1 - Club

DD41 SGINT1
    SVORG 104,98      ;top of club handle at hand
    SVECT 98,112

DD41 6862FC DD44 D7
    FCB 104,98,V$REL
    FCB (((98-104)/2) AND $0F)*16) OR (((112-98)/2) AND $0F)

    SVECT 92,120
DD45 D4
    FCB (((92-98)/2) AND $0F)*16) OR (((120-112)/2) AND $0F)

    SVECT 94,128
DD46 14
    FCB (((94-92)/2) AND $0F)*16) OR (((128-120)/2) AND $0F)

    SVECT 96,132
DD47 12
    FCB (((96-94)/2) AND $0F)*16) OR (((132-128)/2) AND $0F)

    SVECT 102,132      ;meet leg
DD48 30
    FCB (((102-96)/2) AND $0F)*16) OR (((132-132)/2) AND $0F)

    SVECT 104,126
DD49 1D
    FCB (((104-102)/2) AND $0F)*16) OR (((126-132)/2) AND $0F)

    SVECT 104,120
DD4A 0D
    FCB (((104-104)/2) AND $0F)*16) OR (((120-126)/2) AND $0F)

    SVECT 102,114
DD4B FD
    FCB (((102-104)/2) AND $0F)*16) OR (((114-120)/2) AND $0F)

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D3 - Creature Vector List (I)

PAGE 61

```

        SVECT  106,100      ;bottom of club handle at hand
DD4C 29      FCB    (((((106-102)/2) AND $0F)*16) OR (((100-114)/2) AND $0F)

        SVNEW  ;
DD4D 00      FCB    V$ABS

        FCB    V$JMP
DD4E FD      FDB    SGIANT      ;chain to sgiant
DD4F DD62    FCB    V$NEW      ;remainder of Stone Giant

        Stone Giant 2 - Axe
DD51 6862    SGINT2 FCB    104,98      ;top upper axe handle at hand
DD53 5E7C    FCB    94,124
DD55 607E    FCB    96,126
DD57 6A64    FCB    106,100      ;lower upper axe handle
DD59 FF      FCB    V$NEW

DD5A 6684    FCB    102,132      ;top of axe blade
DD5C 5C72    FCB    92,114      ;back tip
DD5E 6676    FCB    102,118      ;lower bend
DD60 6E72    FCB    110,114      ;bottom of blade

        ;SGINT2 falls thru to SGIANT

        SGIANT  SVORG
DD62          SVECT  102,132      ;left leg (at club or axe)
                  102,136

        FCB    102,132,V$REL
DD65 02      FCB    (((((102-102)/2) AND $0F)*16) OR (((136-132)/2) AND $0F)

        SVECT  112,148
DD66 56      FCB    (((((112-102)/2) AND $0F)*16) OR (((148-136)/2) AND $0F)

        SVECT  122,160      ;left heel
DD67 56      FCB    (((((122-112)/2) AND $0F)*16) OR (((160-148)/2) AND $0F)

        SVECT  124,174
DD68 17      FCB    (((((124-122)/2) AND $0F)*16) OR (((174-160)/2) AND $0F)

        SVECT  120,170
DD69 EE      FCB    (((((120-124)/2) AND $0F)*16) OR (((170-174)/2) AND $0F)

        SVECT  120,174
DD7A 02      FCB    (((((120-120)/2) AND $0F)*16) OR (((174-170)/2) AND $0F)

```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D3 - Creature Vector List (I)

PAGE 62

DD6B EA	SVECT FCB	116,162 ;left upper ankle (((116-120)/2) AND \$0F)*16) OR (((162-174)/2) AND \$0F)
DD6C BB	SVECT FCB	106,152 (((106-116)/2) AND \$0F)*16) OR (((152-162)/2) AND \$0F)
DD6D BB	SVECT FCB	96,142 ;left knee (((96-106)/2) AND \$0F)*16) OR (((142-152)/2) AND \$0F)
DD6E EA	SVECT FCB	92,130 (((92-96)/2) AND \$0F)*16) OR (((130-142)/2) AND \$0F)
DD6F EA	SVECT FCB	88,118 ;left thigh at arm (((88-92)/2) AND \$0F)*16) OR (((118-130)/2) AND \$0F)
DD70 00	SVNEW FCB	V\$ABS
 ; face		
DD71 4E5CFC	FCB	78,92,V\$REL
DD74 C2	FCB	(((70-78)/2) AND \$0F)*16) OR (((96-92)/2) AND \$0F)
DD75 S1	FCB	(((80-70)/2) AND \$0F)*16) OR (((98-96)/2) AND \$0F)
DD76 S3	FCB	(((86-80)/2) AND \$0F)*16) OR (((94-98)/2) AND \$0F)
DD77 CF	FCB	(((78-86)/2) AND \$0F)*16) OR (((92-94)/2) AND \$0F)
DD78 FC	FCB	(((76-78)/2) AND \$0F)*16) OR (((84-92)/2) AND \$0F)
DD79 42	FCB	(((84-76)/2) AND \$0F)*16) OR (((88-84)/2) AND \$0F)
DD7A 13	FCB	(((86-84)/2) AND \$0F)*16) OR (((94-88)/2) AND \$0F)
DD7B 00	SVNEW FCB	V\$ABS
 ; bottom of club/axe handle and right leg		
DD7C 6A5AFC	FCB	106,90,V\$REL
DD7F 1E	FCB	(((108-106)/2) AND \$0F)*16) OR (((86-90)/2) AND \$0F)
DD80 11	FCB	(((110-108)/2) AND \$0F)*16) OR (((88-86)/2) AND \$0F)
DD81 F3	FCB	(((108-110)/2) AND \$0F)*16) OR (((94-88)/2) AND \$0F)
DD82 62	FCB	(((120-108)/2) AND \$0F)*16) OR (((98-94)/2) AND \$0F)
DD83 39	FCB	(((126-120)/2) AND \$0F)*16) OR (((84-98)/2) AND \$0F)
DD84 E2	FCB	(((122-122)/2) AND \$0F)*16) OR (((88-84)/2) AND \$0F)
DD85 0C	FCB	(((122-122)/2) AND \$0F)*16) OR (((80-88)/2) AND \$0F)
DD86 E4	FCB	(((118-122)/2) AND \$0F)*16) OR (((88-80)/2) AND \$0F)
DD87 8A	FCB	(((102-118)/2) AND \$0F)*16) OR (((76-88)/2) AND \$0F)
DD88 E2	FCB	(((98-102)/2) AND \$0F)*16) OR (((80-76)/2) AND \$0F)
DD89 00	SVNEW FCB	V\$ABS
 ; arm and head		
DD8A 5654FC	FCB	86,84,V\$REL
DD8B 54	FCB	(((96-86)/2) AND \$0F)*16) OR (((92-84)/2) AND \$0F)
DD8C 65	FCB	(((108-96)/2) AND \$0F)*16) OR (((102-92)/2) AND \$0F)
DD8F 2E	FCB	(((112-108)/2) AND \$0F)*16) OR (((98-102)/2) AND \$0F)

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D3 - Creature Vector List (I)

PAGE 63

```
DD90 CA      FCB    (((((104-112)/2) AND $0F)*16) OR (((86-98)/2) AND $0F)  
DD91 BA      FCB    (((((94-104)/2) AND $0F)*16) OR (((74-86)/2) AND $0F)  
DD92 A1      FCB    (((((82-94)/2) AND $0F)*16) OR (((76-74)/2) AND $0F)  
DD93 D4      FCB    (((((76-82)/2) AND $0F)*16) OR (((84-76)/2) AND $0F)  
DD94 EE      FCB    (((((72-76)/2) AND $0F)*16) OR (((80-84)/2) AND $0F)  
DD95 12      FCB    (((((74-72)/2) AND $0F)*16) OR (((84-80)/2) AND $0F)  
DD96 D2      FCB    (((((68-74)/2) AND $0F)*16) OR (((88-84)/2) AND $0F)  
DD97 13      FCB    (((((70-68)/2) AND $0F)*16) OR (((94-88)/2) AND $0F)  
DD98 E1      FCB    (((((66-70)/2) AND $0F)*16) OR (((96-94)/2) AND $0F)  
DD99 20      FCB    (((((70-66)/2) AND $0F)*16) OR (((96-96)/2) AND $0F)  
DD9A F6      FCB    (((((68-70)/2) AND $0F)*16) OR (((108-96)/2) AND $0F)  
DD9B 24      FCB    (((((72-68)/2) AND $0F)*16) OR (((116-108)/2) AND $0F)  
DD9C 72      FCB    (((((86-72)/2) AND $0F)*16) OR (((120-116)/2) AND $0F)  
DD9D 58      FCB    (((((96-84)/2) AND $0F)*16) OR (((104-120)/2) AND $0F)  
DD9E EE      FCB    (((((92-96)/2) AND $0F)*16) OR (((100-104)/2) AND $0F)  
DD9F C5      FCB    (((((84-92)/2) AND $0F)*16) OR (((110-100)/2) AND $0F)  
DDAO BE     FCB    (((((74-84)/2) AND $0F)*16) OR (((106-110)/2) AND $0F)  
          SVEND  
D0A1 00FE    FCB    V$ABS,V$END
```

; End of Stone-Giant vector list

; \* \* \* \* \*

## ; Balrog vector list

; Outside legs, wings, underside of arms,  
and bottom "V" of torso

DDA3 507C	BALROG	FCB	80,124	;top of right leg at body
DDA5 5E72		FCB	94,114	;right knee
DDA7 6E78		FCB	110,120	;right ankle
DDA9 8470		FCB	132,112	;right toe
DDAB 684E		FCB	104,78	;right wing
DDAD 8430		FCB	132,48	
DDAF 4448		FCB	68,72	
DBB1 5420		FCB	84,32	
DBE3 1558		FCB	22,88	;bottom tip of right hand
DBE5 2472		FCB	52,114	;right armpit
DBE7 5C80		FCB	92,128	;crotch
DBE9 348E		FCB	52,142	;left armpit
DBRB 16A8		FCB	22,168	;bottom tip of left hand
DBRD 58E0		FCB	88,224	;left wing
DBRF 44B8		FCB	68,184	
DDC1 84D0		FCB	132,208	
DDC2 70B2		FCB	112,178	
DDC5 8490		FCB	132,144	;left toe
DDC7 6E89		FCB	110,136	
DDC9 5E8E		FCB	94,142	
DDCB 5084		FCB	80,132	;top of left leg at body
DDCD FF		FCB	V\$NEW	

## ; Inside of legs

DDCE 8470FC		FCB	132,112,V\$REL	
DDD1 C5		FCB	((((124-132)/2) AND \$OF)*16) OR (((122-112)/2) AND \$OF)	
DDD2 92		FCB	((((110-124)/2) AND \$OF)*16) OR (((126-122)/2) AND \$OF)	
DDD3 BE		FCB	((((100-110)/2) AND \$OF)*16) OR (((122-126)/2) AND \$OF)	
DDD4 C3		FCB	((((92-100)/2) AND \$OF)*16) OR (((128-122)/2) AND \$OF)	
DDD5 43		FCB	((((100-92)/2) AND \$OF)*16) OR (((134-128)/2) AND \$OF)	
DDD6 5E		FCB	((((110-100)/2) AND \$OF)*16) OR (((130-134)/2) AND \$OF)	
DDD7 72		FCB	((((124-110)/2) AND \$OF)*16) OR (((134-130)/2) AND \$OF)	
DDD8 45		FCB	((((132-124)/2) AND \$OF)*16) OR (((144-134)/2) AND \$OF)	
;		FCB	V\$NEW	
DDD9 00		FCB	V\$ABS	

## ; Tail

DDEA 527AFC		FCB	82,122,V\$REL	
DDED 78		FCB	((((96-82)/2) AND \$OF)*16) OR (((106-122)/2) AND \$OF)	
DDEE E9		FCB	((((92-96)/2) AND \$OF)*16) OR (((92-106)/2) AND \$OF)	
DDEF 80		FCB	((((76-92)/2) AND \$OF)*16) OR (((86-92)/2) AND \$OF)	
DDE0 EC		FCB	((((72-76)/2) AND \$OF)*16) OR (((78-86)/2) AND \$OF)	
DDE1 33		FCB	((((78-72)/2) AND \$OF)*16) OR (((84-78)/2) AND \$OF)	
DDE2 0C		FCB	((((78-78)/2) AND \$OF)*16) OR (((76-84)/2) AND \$OF)	
DDE3 24		FCB	((((82-78)/2) AND \$OF)*16) OR (((84-76)/2) AND \$OF)	
DDE4 72		FCB	((((96-82)/2) AND \$OF)*16) OR (((88-84)/2) AND \$OF)	
DDE5 47		FCB	((((104-96)/2) AND \$OF)*16) OR (((102-88)/2) AND \$OF)	
DDE6 57		FCB	((((100-104)/2) AND \$OF)*16) OR (((114-102)/2) AND \$OF)	

```

        SVNEW
DDE7 00      FCB    V$ABS

        Arms and head

DDE8 16A8FC  FCB    22,168,V$REL
DDEB 2D      FCB    (((((26-22)/2)) AND $0F)*16) OR (((162-168)/2)) AND $0F)
DDEC C2      FCB    (((((18-26)/2)) AND $0F)*16) OR (((166-162)/2)) AND $0F)
DDED 3D      FCB    (((((24-18)/2)) AND $0F)*16) OR (((160-166)/2)) AND $0F)
DDEE 30      FCB    (((((30-24)/2)) AND $0F)*16) OR (((160-160)/2)) AND $0F)
DDEF 4B      FCB    (((((38-30)/2)) AND $0F)*16) OR (((150-150)/2)) AND $0F)
DDFO 4B      FCB    (((((46-38)/2)) AND $0F)*16) OR (((140-150)/2)) AND $0F)
DDF1 ED      FCB    (((((42-46)/2)) AND $0F)*16) OR (((134-140)/2)) AND $0F)
DDF2 B2      FCB    (((((32-42)/2)) AND $0F)*16) OR (((138-134)/2)) AND $0F)
DDF3 9D      FCB    (((((18-32)/2)) AND $0F)*16) OR (((132-138)/2)) AND $0F)
DDF4 21      FCB    (((((32-18)/2)) AND $0F)*16) OR (((134-132)/2)) AND $0F)
DDF5 3D      FCB    (((((38-32)/2)) AND $0F)*16) OR (((128-134)/2)) AND $0F)
DDF6 DD      FCB    (((((32-38)/2)) AND $0F)*16) OR (((122-128)/2)) AND $0F)
DDF7 91      FCB    (((((18-32)/2)) AND $0F)*16) OR (((124-122)/2)) AND $0F)
DDF8 7D      FCB    (((((32-10)/2)) AND $0F)*16) OR (((118-124)/2)) AND $0F)
DDF9 52      FCB    (((((42-32)/2)) AND $0F)*16) OR (((122-118)/2)) AND $0F)
DDFA 63      FCB    (((((54-42)/2)) AND $0F)*16) OR (((128-122)/2)) AND $0F)
DDFB A3      FCB    (((((42-54)/2)) AND $0F)*16) OR (((134-128)/2)) AND $0F)
DDFC 2D      FCB    (((((46-42)/2)) AND $0F)*16) OR (((128-134)/2)) AND $0F)
DDFD ED      FCB    (((((42-46)/2)) AND $0F)*16) OR (((122-128)/2)) AND $0F)
DDFE 2D      FCB    (((((46-42)/2)) AND $0F)*16) OR (((116-122)/2)) AND $0F)
DDFF CB      FCB    (((((38-46)/2)) AND $0F)*16) OR (((106-116)/2)) AND $0F)
DE00 CB      FCB    (((((30-38)/2)) AND $0F)*16) OR (((96-106)/2)) AND $0F)
DE01 DO      FCB    (((((24-30)/2)) AND $0F)*16) OR (((96-96)/2)) AND $0F)
DE02 DD      FCB    (((((18-24)/2)) AND $0F)*16) OR (((90-96)/2)) AND $0F)
DE03 42      FCB    (((((26-18)/2)) AND $0F)*16) OR (((94-90)/2)) AND $0F)
DE04 ED      FCB    (((((22-26)/2)) AND $0F)*16) OR (((88-94)/2)) AND $0F)

        SVEND
DE05 00FE    FCB    V$ABS,V$END

```

End of Balrog vector list

\* \* \* \* \*

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D3 - Creature Vector List (I)

PAGE 46

; Wraith vector list

; Eyebrows

DE07 3E44	WRAITH	FCB	62,68
DE09 4458		FCB	68,88
DE0B 3864		FCB	56,100
DE0D FF		FCB	V\$NEW

; Eyes

DE0E 4A5A		FCB	74,90	\$bridge of nose
DE10 464AFC		FCB	70,74,V\$REL	
DE13 33		FCB	((((76-70)/2) AND \$0F)*16)	OR (((80-74)/2) AND \$0F)
DE14 F5		FCB	((((74-76)/2) AND \$0F)*16)	OR (((90-80)/2) AND \$0F)
DE15 F5		FCB	((((72-74)/2) AND \$0F)*16)	OR (((100-90)/2) AND \$0F)
DE16 C1		FCB	((((64-72)/2) AND \$0F)*16)	OR (((102-100)/2) AND \$0F)
DE17 5A		FCB	((((74-64)/2) AND \$0F)*16)	OR (((90-102)/2) AND \$0F)
DE18 62		FCB	((((86-74)/2) AND \$0F)*16)	OR (((94-90)/2) AND \$0F)
DE19 0E		FCB	((((86-86)/2) AND \$0F)*16)	OR (((90-94)/2) AND \$0F)
DE1A 00		SVNEW		
		FCB	V\$ABS	

; Mouth

DE1B 6450FC		FCB	100,80,V\$REL	
DE1E B3		FCB	((((90-100)/2) AND \$0F)*16)	OR (((86-80)/2) AND \$0F)
DE1F 17		FCB	((((92-90)/2) AND \$0F)*16)	OR (((100-86)/2) AND \$0F)
DE20 34		FCB	((((98-92)/2) AND \$0F)*16)	OR (((108-100)/2) AND \$0F)
DE21 EB		FCB	((((94-98)/2) AND \$0F)*16)	OR (((98-108)/2) AND \$0F)
DE22 0A		FCB	((((94-94)/2) AND \$0F)*16)	OR (((86-98)/2) AND \$0F)
DE23 3D		FCB	((((100-94)/2) AND \$0F)*16)	OR (((80-86)/2) AND \$0F)
DE24 00FE		SVEND		
		FCB	V\$ABS,V\$END	

; End of Wraith vector list

; \* \* \* \* \*

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D3 - Creature Vector List (I)

PAGE 67

; Spider vector list

; Outside legs and body

DE26	SPIDER	FCB	124,160,V\$REL
DE26	C2	FCB	((((116-124)/2) AND \$0F)*16) OR (((164-160)/2) AND \$0F)
DE2A	22	FCB	((((120-116)/2) AND \$0F)*16) OR (((168-164)/2) AND \$0F)
DE2B	E4	FCB	((((116-120)/2) AND \$0F)*16) OR (((176-168)/2) AND \$0F)
DE2C	24	FCB	((((120-116)/2) AND \$0F)*16) OR (((184-176)/2) AND \$0F)
DE2D	2C	FCB	((((124-120)/2) AND \$0F)*16) OR (((176-184)/2) AND \$0F)
DE2E	EC	FCB	((((120-124)/2) AND \$0F)*16) OR (((168-176)/2) AND \$0F)
DE2F	04	FCB	((((120-120)/2) AND \$0F)*16) OR (((176-168)/2) AND \$0F)
DE30	04	FCB	((((120-120)/2) AND \$0F)*16) OR (((184-176)/2) AND \$0F)
DE31	E2	FCB	((((116-120)/2) AND \$0F)*16) OR (((188-184)/2) AND \$0F)
DE32	42	FCB	((((124-116)/2) AND \$0F)*16) OR (((192-188)/2) AND \$0F)
DE33	00	SVNEW	FCB V\$ABS

; Inner legs and mouth

DE34	7CA8FC	FCB	124,168,V\$REL
DE37	C1	FCB	((((116-124)/2) AND \$0F)*16) OR (((170-168)/2) AND \$0F)
DE38	21	FCB	((((120-116)/2) AND \$0F)*16) OR (((172-170)/2) AND \$0F)
DE39	12	FCB	((((122-120)/2) AND \$0F)*16) OR (((176-172)/2) AND \$0F)
DE3A	E2	FCB	((((120-122)/2) AND \$0F)*16) OR (((180-176)/2) AND \$0F)
DE3B	E1	FCB	((((116-120)/2) AND \$0F)*16) OR (((182-180)/2) AND \$0F)
DE3C	41	FCB	((((124-116)/2) AND \$0F)*16) OR (((184-182)/2) AND \$0F)
DE3D	00FE	SVEND	FCB V\$ABS,V\$END

; End of Spider vector list

; \* \* \* \* \*

## ; Scorpion vector list

## ; Tail, body, and back leg

DE3F	704AFC	SCORP	FCB 112,74,V\$REL
DE42	E0		FCB (((((108-112)/2) AND \$0F)*16) OR (((74-74)/2) AND \$0F))
DE43	EE		FCB (((((104-108)/2) AND \$0F)*16) OR (((70-74)/2) AND \$0F))
DE44	2C		FCB (((((108-104)/2) AND \$0F)*16) OR (((62-70)/2) AND \$0F))
DE45	42		FCB (((((116-108)/2) AND \$0F)*16) OR (((66-62)/2) AND \$0F))
DE46	14		FCB (((((118-116)/2) AND \$0F)*16) OR (((74-86)/2) AND \$0F))
DE47	14		FCB (((((120-118)/2) AND \$0F)*16) OR (((82-74)/2) AND \$0F))
DE48	20		FCB (((((124-120)/2) AND \$0F)*16) OR (((82-82)/2) AND \$0F))
DE49	0C		FCB (((((124-124)/2) AND \$0F)*16) OR (((74-82)/2) AND \$0F))
DE4A	CC		FCB (((((118-124)/2) AND \$0F)*16) OR (((66-74)/2) AND \$0F))
DE4B	22		FCB (((((120-116)/2) AND \$0F)*16) OR (((70-66)/2) AND \$0F))
DE4C	0C		FCB (((((120-120)/2) AND \$0F)*16) OR (((62-70)/2) AND \$0F))
DE4D	22		FCB (((((124-120)/2) AND \$0F)*16) OR (((66-62)/2) AND \$0F))
DE4E	00	; SVNEW	FCB V\$MRS

## ; Front legs

DE4F	7C5AFC		FCB 124,90,V\$REL
DE52	E0		FCB (((((120-124)/2) AND \$0F)*16) OR (((90-90)/2) AND \$0F))
DE53	0C		FCB (((((120-120)/2) AND \$0F)*16) OR (((82-90)/2) AND \$0F))
DE54	2C		FCB (((((124-120)/2) AND \$0F)*16) OR (((74-82)/2) AND \$0F))
DE55	20		FCB (((((128-124)/2) AND \$0F)*16) OR (((74-74)/2) AND \$0F))
DE56	04		FCB (((((128-128)/2) AND \$0F)*16) OR (((82-74)/2) AND \$0F))
DE57	00FE	; SVEND	FCB V\$ABS,V\$END

## ; End of Scorpion vector list

\*\*\*\*\*

## ; Blob vector list

DE59	52B2FC	BLOB	FCB 82,130,V\$REL
DE5C	28		FCB (((((86-82)/2) AND \$0F)*16) OR (((114-130)/2) AND \$0F))
DE5D	70		FCB (((((100-86)/2) AND \$0F)*16) OR (((108-114)/2) AND \$0F))
DE5E	5F		FCB (((((110-100)/2) AND \$0F)*16) OR (((106-108)/2) AND \$0F))
DE5F	50		FCB (((((120-110)/2) AND \$0F)*16) OR (((106-106)/2) AND \$0F))
DE60	5B		FCB (((((130-120)/2) AND \$0F)*16) OR (((96-106)/2) AND \$0F))
DE61	F5		FCB (((((128-130)/2) AND \$0F)*16) OR (((106-96)/2) AND \$0F))
DE62	2F		FCB (((((132-128)/2) AND \$0F)*16) OR (((104-106)/2) AND \$0F))
DE63	D5		FCB (((((126-132)/2) AND \$0F)*16) OR (((114-104)/2) AND \$0F))
DE64	17		FCB (((((128-126)/2) AND \$0F)*16) OR (((128-114)/2) AND \$0F))
DE65	17		FCB (((((130-128)/2) AND \$0F)*16) OR (((142-128)/2) AND \$0F))
DE66	F3		FCB (((((128-130)/2) AND \$0F)*16) OR (((148-142)/2) AND \$0F))
DE67	22		FCB (((((132-128)/2) AND \$0F)*16) OR (((152-148)/2) AND \$0F))
DE68	E1		FCB (((((126-132)/2) AND \$0F)*16) OR (((154-152)/2) AND \$0F))
DE69	14		FCB (((((130-128)/2) AND \$0F)*16) OR (((162-154)/2) AND \$0F))
DEAA	TB		FCB (((((124-130)/2) AND \$0F)*16) OR (((158-142)/2) AND \$0F))

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D3 - Creature Vector List (I)

PAGE 69

DE6B 8F        FCB     (((((108-124)/2) AND \$0F)\*16) OR (((154-156)/2) AND \$0F)  
DE6C 8D        FCB     (((((92-108)/2) AND \$0F)\*16) OR (((148-154)/2) AND \$0F)  
DE6D DB        FCB     (((((86-92)/2) AND \$0F)\*16) OR (((138-148)/2) AND \$0F)  
DE6E EC        FCB     (((((82-86)/2) AND \$0F)\*16) OR (((130-138)/2) AND \$0F)  
DE6F 00        ;  
                ;      SYNEW  
                ;      FCB     V\$ABS.

; Eyes

DE70 5682FC    FCB     86,130,V\$REL  
DE73 53        FCB     (((((92-86)/2) AND \$0F)\*16) OR (((136-130)/2) AND \$0F)  
DE74 31        FCB     (((((98-92)/2) AND \$0F)\*16) OR (((138-136)/2) AND \$0F)  
DE75 18        FCB     (((((100-98)/2) AND \$0F)\*16) OR (((128-138)/2) AND \$0F)  
DE76 91        FCB     (((((86-100)/2) AND \$0F)\*16) OR (((130-128)/2) AND \$0F)  
DE77 3B        FCB     (((((92-86)/2) AND \$0F)\*16) OR (((120-130)/2) AND \$0F)  
DE78 5F        FCB     (((((102-92)/2) AND \$0F)\*16) OR (((118-120)/2) AND \$0F)  
DE79 F5        FCB     (((((100-102)/2) AND \$0F)\*16) OR (((128-118)/2) AND \$0F)  
DE7A 00        ;  
                ;      SYNEW  
                ;      FCB     V\$ABS

; Mouth

DE7B 6C74      FCB     108,116  
DE7D 7276      FCB     114,118  
DE7F 7890      FCB     120,144  
DE81 FE        FCB     V\$END

; End of Blob vector list

; \* \* \* \* \* \* \* \* \* \*  
END

INCLUD B:D4.ASM

||||||||| C R E A T U R E   V E C T O R   L I S T S   ( I I ) |||||

NAM D4 - Creature Vector Lists (II)

LIBRY CD

XDEF KNIGT1,KNIGT2,VIPER  
XDEF WIZ0,WIZ1,WIZ2  
OPT NOMEX,NOGEN  
DSCT

; Knight Type 1 - shield entry point (orange coding)

DE82	KNIGT1	FCB	34,124,V\$REL
DE82	227CFC	FCB	((((34-34)/2) AND \$0F)*16) OR (((132-124)/2) AND \$0F)
DE85	04	FCB	((((36-34)/2) AND \$0F)*16) OR (((130-132)/2) AND \$0F)
DE86	1F	FCB	((((36-36)/2) AND \$0F)*16) OR (((126-130)/2) AND \$0F)
DE87	0E	FCB	((((34-36)/2) AND \$0F)*16) OR (((124-126)/2) AND \$0F)
DE88	FF	SVNEW	
DE89	00	FCB	V\$ABS
DE8A	508E	FCB	80,142 ;left arm
DE8C	4088	FCB	64,136
DE8E	2E92	FCB	46,146
DE90	409C	FCB	64,156
DE92	528C	FCB	82,140
DE94	4C88	FCB	76,136
DE96	4092	FCB	64,146
DE98	3A8C	FCB	58,140
DE9A	FD	FCB	V\$JMP
DE9B	DEB3	FDB	KNIGHT

; Knight Type 2 - shield entry point (red coding)

DE9D	KNIGT2	FCB	30,126,V\$REL
DE9D	1E7EFC	FCB	((((40-30)/2) AND \$0F)*16) OR (((126-126)/2) AND \$0F)
DEA0	50	FCB	((((40-40)/2) AND \$0F)*16) OR (((124-126)/2) AND \$0F)
DEA1	0F	FCB	((((36-40)/2) AND \$0F)*16) OR (((124-124)/2) AND \$0F)
DEA2	E0	SVNEW	
DEA3	00	FCB	V\$ABS
DEA4	2C96	FCB	44,150
DEA5	34A6	FCB	52,166
DEA6	4CA4	FCB	76,164
DEA7	5C96	FCB	92,150
DEA8	4C88	FCB	76,136
DEA9	3486	FCB	52,134
DEB0	2C96	FCB	44,150
DEB2	FF	FDB	V\$NEW

; falls thru to KNIGHT

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies, Inc.  
D4 - Creature Vector List (II)

PAGE 71

\* \* \* \* \*

; \* \* \* \* \*

## ; Knight's Main Body

## ; Left leg

DEB3 508C	KNIGHT FCB	80,140	; outside of left leg
DEB5 8098	FCB	128,152	; outside of left ankle
DEB7 84A0	FCB	132,160	; toe of left foot
DEB9 8490	FCB	132,144	; heel of left foot
DEBB 7E90	FCB	126,144	; ankle of left foot
DER0 5482	FCB	84,130	; to crotch
DEBF FF	V\$NEW		; start new sequence

## ; Right Leg

DEC0 547E	FCB	84,126	; crotch
DEC2 7E6E	FCB	126,110	; ankle
DEC4 846E	FCB	132,110	; heel
DEC6 845C	FCB	132,92	; toe
DEC8 8066	FCB	128,102	; outside of ankle
DECA 5074	FCB	80,116	; to hip
DECC FF	V\$NEW		; start new sequence

## ; Torso

DED1 508CFD	FCB	80,140,V\$REL	
DED0 3A	FCB	((((86-80)/2) AND \$0F)*16)	OR (((128-140)/2) AND \$0F)
DED1 D9	FCB	((((80-86)/2) AND \$0F)*16)	OR (((114-128)/2) AND \$0F)
DED2 83	FCB	((((64-80)/2) AND \$0F)*16)	OR (((120-114)/2) AND \$0F)
DED3 DE	FCB	((((58-64)/2) AND \$0F)*16)	OR (((116-120)/2) AND \$0F)
DED4 AD	FCB	((((46-58)/2) AND \$0F)*16)	OR (((110-116)/2) AND \$0F)
DED5 E6	FCB	((((42-46)/2) AND \$0F)*16)	OR (((122-110)/2) AND \$0F)
DED6 A1	FCB	((((30-42)/2) AND \$0F)*16)	OR (((124-122)/2) AND \$0F)
DED7 E2	FCB	((((26-30)/2) AND \$0F)*16)	OR (((128-124)/2) AND \$0F)
DED8 22	FCB	((((30-26)/2) AND \$0F)*16)	OR (((132-128)/2) AND \$0F)
DED9 61	FCB	((((42-30)/2) AND \$0F)*16)	OR (((134-132)/2) AND \$0F)
DEDA 26	FCB	((((46-42)/2) AND \$0F)*16)	OR (((146-134)/2) AND \$0F)
DEDB EA	FCB	((((42-46)/2) AND \$0F)*16)	OR (((134-146)/2) AND \$0F)
DEDC 20	FCB	((((46-42)/2) AND \$0F)*16)	OR (((134-134)/2) AND \$0F)
DED0 30	FCB	((((52-46)/2) AND \$0F)*16)	OR (((128-134)/2) AND \$0F)
DEDE DD	FCB	((((46-52)/2) AND \$0F)*16)	OR (((122-128)/2) AND \$0F)
DEDF EO	FCB	((((42-46)/2) AND \$0F)*16)	OR (((122-122)/2) AND \$0F)
DEEO 00	V\$NEW		; start new sequence
	FCB	V\$ABS	

## ; Plume

DEE1 3480	FCB	52,128	; chin of helm
DEE3 1480FC	FCB	20,128,V\$REL	
DEE6 0E	FCB	((((20-20)/2) AND \$0F)*16)	OR (((124-128)/2) AND \$0F)
DEE7 21	FCB	((((24-20)/2) AND \$0F)*16)	OR (((126-124)/2) AND \$0F)
DEE8 02	FCB	((((24-24)/2) AND \$0F)*16)	OR (((130-126)/2) AND \$0F)
DEE9 E1	FCB	((((20-24)/2) AND \$0F)*16)	OR (((132-130)/2) AND \$0F)
DEEA OE	FCB	((((20-20)/2) AND \$0F)*16)	OR (((128-132)/2) AND \$0F)

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D4 - Creature Vector List (II)

PAGE 73

DEEB 00

FCB V\$ABS

; Sword Handle and Lower Arm

DEEC 4A66FC	FCB 74,102,V\$REL
DEEF EO	FCB (((((70-74)/2) AND \$0F)*16) OR (((102-102)/2) AND \$0F))
DEF0 02	FCB (((((70-70)/2) AND \$0F)*16) OR (((106-102)/2) AND \$0F))
DEF1 D0	FCB (((((64-70)/2) AND \$0F)*16) OR (((106-106)/2) AND \$0F))
DEF2 08	FCB (((((64-64)/2) AND \$0F)*16) OR (((90-106)/2) AND \$0F))
DEF3 30	FCB (((((70-64)/2) AND \$0F)*16) OR (((90-90)/2) AND \$0F))
DEF4 02	FCB (((((70-70)/2) AND \$0F)*16) OR (((94-90)/2) AND \$0F))
DEF5 20	FCB (((((74-70)/2) AND \$0F)*16) OR (((94-94)/2) AND \$0F))
DEF6 01	FCB (((((74-74)/2) AND \$0F)*16) OR (((96-94)/2) AND \$0F))
DEF7 30	FCB (((((80-74)/2) AND \$0F)*16) OR (((96-96)/2) AND \$0F))
DEF8 02	FCB (((((80-80)/2) AND \$0F)*16) OR (((100-96)/2) AND \$0F))
DEF9 D0	FCB (((((74-80)/2) AND \$0F)*16) OR (((100-100)/2) AND \$0F))
DEFA 01	FCB (((((74-74)/2) AND \$0F)*16) OR (((102-100)/2) AND \$0F))
DEFB 87	FCB (((((58-74)/2) AND \$0F)*16) OR (((116-102)/2) AND \$0F))

DEFC 00

FCB V\$ABS

; Upper Part of Arm and Sword Blade

DEFD 2E6E	FCB 46,110
DEFF 4066	FCB 64,102
DEF01 4064	FCB 64,100
DEF03 1E66	FCB 30,102
DEF05 1462	FCB 20,98
DEF07 1E5E	FCB 30,94
DEF09 4060	FCB 64,96
DEF0B 4062	FCB 64,98
DEF0D 1462	FCB 20,98
DEF0F FE	FCB V\$END

;end of Knight vector list

; End of Knight vector list

; \* \* \* \* \*

\* \* \* \* \*

## Wizard 1

with crescent shaped sceptre point

WIZ1

```

DF10 2E62FC      FCB    46,98,V$REL
DF13 21          FCB    (((((50-46)/2) AND $0F)*16) OR (((100-98)/2) AND $0F)
DF14 2F          FCB    (((((54-50)/2) AND $0F)*16) OR (((98-100)/2) AND $0F)
DF15 2D          FCB    (((((58-54)/2) AND $0F)*16) OR (((92-98)/2) AND $0F)
DF16 FD          FCB    (((((56-58)/2) AND $0F)*16) OR (((86-92)/2) AND $0F)
DF17 CE          FCB    (((((48-56)/2) AND $0F)*16) OR (((82-86)/2) AND $0F)
DF18 C2          FCB    (((((40-48)/2) AND $0F)*16) OR (((86-82)/2) AND $0F)
DF19 F5          FCB    (((((38-40)/2) AND $0F)*16) OR (((90-86)/2) AND $0F)
DF1A 12          FCB    (((((40-38)/2) AND $0F)*16) OR (((94-90)/2) AND $0F)
DF1B OF          FCB    (((((40-40)/2) AND $0F)*16) OR (((92-94)/2) AND $0F)
DF1C 1E          FCB    (((((42-40)/2) AND $0F)*16) OR (((88-92)/2) AND $0F)
DF1D 3F          FCB    (((((48-42)/2) AND $0F)*16) OR (((86-88)/2) AND $0F)
DF1E 21          FCB    (((((52-48)/2) AND $0F)*16) OR (((88-86)/2) AND $0F)
DF1F 12          FCB    (((((54-52)/2) AND $0F)*16) OR (((92-88)/2) AND $0F)
DF20 E3          FCB    (((((50-54)/2) AND $0F)*16) OR (((98-92)/2) AND $0F)
DF21 EO          FCB    (((((46-50)/2) AND $0F)*16) OR (((98-98)/2) AND $0F)

SVNEW           ; start new sequence
DF22 00          FCB    V$ABS

```

## Crescent on cape

```

DF23 689AFC      FCB    104,154,V$REL
DF26 21          FCB    (((((108-104)/2) AND $0F)*16) OR (((156-154)/2) AND $0F)
DF27 2F          FCB    (((((112-108)/2) AND $0F)*16) OR (((154-156)/2) AND $0F)
DF28 2D          FCB    (((((116-112)/2) AND $0F)*16) OR (((148-154)/2) AND $0F)
DF29 FD          FCB    (((((114-116)/2) AND $0F)*16) OR (((142-148)/2) AND $0F)
DF2A CE          FCB    (((((106-114)/2) AND $0F)*16) OR (((138-142)/2) AND $0F)
DF2B C2          FCB    (((((98-106)/2) AND $0F)*16) OR (((142-138)/2) AND $0F)
DF2C F3          FCB    (((((96-98)/2) AND $0F)*16) OR (((146-142)/2) AND $0F)
DF2D 12          FCB    (((((98-96)/2) AND $0F)*16) OR (((150-146)/2) AND $0F)
DF2E OF          FCB    (((((98-98)/2) AND $0F)*16) OR (((148-150)/2) AND $0F)
DF2F 1E          FCB    (((((100-98)/2) AND $0F)*16) OR (((144-148)/2) AND $0F)
DF30 3F          FCB    (((((106-100)/2) AND $0F)*16) OR (((142-144)/2) AND $0F)
DF31 22          FCB    (((((110-106)/2) AND $0F)*16) OR (((146-142)/2) AND $0F)
DF32 12          FCB    (((((112-110)/2) AND $0F)*16) OR (((150-146)/2) AND $0F)
DF33 E2          FCB    (((((108-112)/2) AND $0F)*16) OR (((154-150)/2) AND $0F)
DF34 EO          FCB    (((((104-108)/2) AND $0F)*16) OR (((154-154)/2) AND $0F)

SVNEW           ;
DF35 00          FCB    V$ABS

```

```

DF36 FD          FCB    V$JMP      ; transfer to another list
DF37 DF45        FCB    WIZO      ; pointer to wizard main vector list

```

\* \* \* \* \*

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D4 - Creature Vector List (II)

PAGE 75

; \* \* \* \* \*

Wizard 2

with Star shaped sceptre point

DF39 2856      WIZZ      FCB      40,86  
DF3B 405C      FCB      64,92  
DF3D 2A64      FCB      42,100  
DF3F 3652      FCB      54,82  
DF41 3868      FCB      56,104  
DF43 2856      FCB      40,86  
DF45 FF      FCB      V\$NEW ;start new sequence

; Stars on cape

DF46 428CFC      FCB      66,140,V\$REL  
DF49 70      FCB      (((((80-66)/2) AND \$0F)\*16) OR (((140-140)/2) AND \$0F)  
DF4A A0      FCB      (((((68-80)/2) AND \$0F)\*16) OR (((134-140)/2) AND \$0F)  
DF4B 35      FCB      (((((74-68)/2) AND \$0F)\*16) OR (((144-134)/2) AND \$0F)  
DF4C 1B      FCB      (((((76-74)/2) AND \$0F)\*16) OR (((134-144)/2) AND \$0F)  
DF4D B3      FCB      (((((66-76)/2) AND \$0F)\*16) OR (((140-134)/2) AND \$0F)  
DF4E 00      FCB      V\$NEW ;start new sequence

; Lower right start

DF4F 6092      FCB      96,146  
DF51 7894      FCB      120,148  
DF53 6488      FCB      100,136  
DF55 6A90      FCB      106,154  
DF57 748A      FCB      116,138  
DF59 6092      FCB      96,146  
DF5B FF      FCB      V\$NEW ;start new sequence

; Upper left star

DF5C 5074FC      FCB      80,116,V\$REL  
DF5F 53      FCB      (((((90-80)/2) AND \$0F)\*16) OR (((122-116)/2) AND \$0F)  
DF60 EC      FCB      (((((86-90)/2) AND \$0F)\*16) OR (((114-122)/2) AND \$0F)  
DF61 E4      FCB      (((((82-86)/2) AND \$0F)\*16) OR (((122-114)/2) AND \$0F)  
DF62 4D      FCB      (((((90-82)/2) AND \$0F)\*16) OR (((116-122)/2) AND \$0F)  
DF63 B0      FCB      (((((80-90)/2) AND \$0F)\*16) OR (((116-116)/2) AND \$0F)  
DF64 00      FCB      V\$NEW ;  
DF65 00      FCB      V\$ABS ;fall thru to WIZO

; \* \* \* \* \*

\*\*\*\*\*

## Main Wizard vector list

DF65	WIZO	FCB 64,124,V\$REL
DF65	407CFC	FCB (((((72-64)/2) AND \$0F)*16) OR (((120-124)/2) AND \$0F)
DF68	4E	FCB (((((64-72)/2) AND \$0F)*16) OR (((120-120)/2) AND \$0F)
DF69	C0	FCB (((((78-64)/2) AND \$0F)*16) OR (((110-120)/2) AND \$0F)
DF6A	7B	FCB (((((64-78)/2) AND \$0F)*16) OR (((102-110)/2) AND \$0F)
DF6B	9C	FCB (((((58-64)/2) AND \$0F)*16) OR (((110-102)/2) AND \$0F)
DF6C	D4	FCB (((((54-58)/2) AND \$0F)*16) OR (((118-110)/2) AND \$0F)
DF6D	E4	FCB (((((50-54)/2) AND \$0F)*16) OR (((120-118)/2) AND \$0F)
DF6E	E1	FCB (((((46-50)/2) AND \$0F)*16) OR (((122-120)/2) AND \$0F)
DF6F	E1	FCB (((((40-46)/2) AND \$0F)*16) OR (((116-122)/2) AND \$0F)
DF70	DD	FCB (((((42-40)/2) AND \$0F)*16) OR (((108-116)/2) AND \$0F)
DF71	1C	FCB (((((28-42)/2) AND \$0F)*16) OR (((120-108)/2) AND \$0F)
DF72	96	FCB (((((28-28)/2) AND \$0F)*16) OR (((126-120)/2) AND \$0F)
DF73	03	
DF74	00	SYNEW ;start new sequence
		FCB V\$ABS
DF75	1C82FC	FCB 28,130,V\$REL
DF78	03	FCB (((((28-28)/2) AND \$0F)*16) OR (((136-130)/2) AND \$0F)
DF79	45	FCB (((((36-28)/2) AND \$0F)*16) OR (((146-136)/2) AND \$0F)
DF7A	71	FCB (((((50-36)/2) AND \$0F)*16) OR (((148-146)/2) AND \$0F)
DF7E	DA	FCB (((((44-50)/2) AND \$0F)*16) OR (((136-148)/2) AND \$0F)
DF7C	1E	FCB (((((46-44)/2) AND \$0F)*16) OR (((132-136)/2) AND \$0F)
DF7D	11	FCB (((((48-46)/2) AND \$0F)*16) OR (((134-132)/2) AND \$0F)
DF7E	E1	FCB (((((44-48)/2) AND \$0F)*16) OR (((136-134)/2) AND \$0F)
DF7F	00	SYNEW ;start new sequence
		FCB V\$ABS
DF80	3086	FCB 48,134
DF82	368E	FCB 54,142
DF84	74A4	FCB 116,164 ;down left side of cape
DF86	8484	FCB 132,132 ;to tip of sceptre
DF88	8276	FCB 130,118 ;across bottom of cape
DF8A	785E	FCB 120,94
DF8C	5A6E	FCB 90,110 ;to shaft of sceptre
DF8E	8484	FCB 132,132 ;down shaft of sceptre
DF90	486A	FCB 72,106 ;up inside of shaft to hand
DF92	FF	FCB V\$NEW ; , ,
DF93	4066FC	FCB 64,102,V\$REL
DF96	1F	FCB (((((66-64)/2) AND \$0F)*16) OR (((100-102)/2) AND \$0F)
DF97	BD	FCB (((((56-66)/2) AND \$0F)*16) OR (((94-100)/2) AND \$0F)
DF98	F1	FCB (((((54-56)/2) AND \$0F)*16) OR (((96-94)/2) AND \$0F)
DF99	53	FCB (((((64-54)/2) AND \$0F)*16) OR (((102-96)/2) AND \$0F)
DF9A	00	SYNEW ;start new sequence
		FCB V\$ABS
DF9B	4266FC	FCB 66,102,V\$REL
DF9E	1E	FCB (((((68-66)/2) AND \$0F)*16) OR (((98-102)/2) AND \$0F)
DF9F	32	FCB (((((74-68)/2) AND \$0F)*16) OR (((102-98)/2) AND \$0F)
DFA0	11	FCB (((((76-74)/2) AND \$0F)*16) OR (((104-102)/2) AND \$0F)
DFA1	73	FCB (((((78-74)/2) AND \$0F)*16) OR (((116-104)/2) AND \$0F))

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D4 - Creature Vector List (II)

PAGE 77

```

; SVNEW           ;start new sequence
DFA2 00          FCB    V$ABS

DFA3 5870        FCB    88,112      ;cape from sceptre shaft
DFA5 4878        FCB    72,120      ;to armpit
DFA7 FF          FCB    V$NEW       ;start new sequence

; Hat from left side of beard
DFA8 3E84        FCB    62,132      ;left of beard below head
DFAA 1480        FCB    20,128      ;to top of hat
DFAC 347A        FCB    52,122      ;right of beard below head
DFAE 407A        FCB    64,122      ;right beard detail
DFB0 3C7C        FCB    60,124
DFB2 7280        FCB    114,128     ;to bottom tip of beard
DFB4 5082        FCB    80,130
DFB6 4482        FCB    48,130
DFB8 3E84        FCB    62,132
DFBA FF          FCB    V$NEW

; Face detail
DEBB 2882FC      FCB    40,130,V$REL
DFBF FF          FCB    (((((38-40)/2) AND $0F)*16) OR (((128-130)/2) AND $0F))
DFC0 1E          FCB    (((((40-38)/2) AND $0F)*16) OR (((124-128)/2) AND $0F))
DFC1 11          FCB    (((((42-40)/2) AND $0F)*16) OR (((126-124)/2) AND $0F))
DFC2 F2          FCB    (((((40-42)/2) AND $0F)*16) OR (((130-126)/2) AND $0F))
DFC3 20          FCB    (((((46-40)/2) AND $0F)*16) OR (((128-130)/2) AND $0F))
DFC4 0F          FCB    (((((50-46)/2) AND $0F)*16) OR (((128-128)/2) AND $0F))
DFC5 C0          FCB    (((((50-50)/2) AND $0F)*16) OR (((126-128)/2) AND $0F))
DFC6 FF          FCB    (((((42-50)/2) AND $0F)*16) OR (((126-126)/2) AND $0F))
DFC7 31          FCB    (((((46-40)/2) AND $0F)*16) OR (((124-126)/2) AND $0F))

DFC8 00FE        SVEND
                  FCB    V$ABS,V$END      ;end of wizard vector list

; End of Wizard vector list
; * * * * * * * *
```

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
D4 - Creature Vector List (II)

PAGE 78

\* \* \* \* \*

Viper vector list

Body and neck

DFCA 8482	VIPER	FCB	132,130	right neck at ground
DFCC 707A		FCB	112,122	up neck
DFCE 5C7C		FCB	92,124	
DFD0 5E7E		FCB	94,126	right chin
DFD2 5E82		FCB	94,130	left chin
DFD4 5C84		FCB	92,132	
DFD6 7082		FCB	112,130	
DFD8 808C		FCB	128,140	
DFIA 8488		FCB	132,136	left neck at ground
DFID 6472		FCB	132,114	right body at ground
DFIE 786C		FCB	120,108	
DFEO 6A76		FCB	106,118	
DFE2 7870		FCB	120,112	tip of tail
DFE4 7C74		FCB	124,116	
DFE6 7C7E		FCB	124,126	meet neck
DFEB FF		FCB	V\$NEW	

Head

DFE9 6478FC		FCB	100,120, V\$REL	
DFEC E0		FCB	((((96-100)/2) AND \$0F)*16) OR (((120-120)/2) AND \$0F)	
DFED E2		FCB	((((92-96)/2) AND \$0F)*16) OR (((124-120)/2) AND \$0F)	
DFEE EE		FCB	((((88-92)/2) AND \$0F)*16) OR (((120-124)/2) AND \$0F)	
DFEF EO		FCB	((((84-88)/2) AND \$0F)*16) OR (((120-120)/2) AND \$0F)	
DFF0 F1		FCB	((((82-84)/2) AND \$0F)*16) OR (((122-120)/2) AND \$0F)	
DFF1 F2		FCB	((((86-82)/2) AND \$0F)*16) OR (((126-122)/2) AND \$0F)	
DFF2 EE		FCB	((((82-86)/2) AND \$0F)*16) OR (((122-126)/2) AND \$0F)	
DFF3 06		FCB	((((92-82)/2) AND \$0F)*16) OR (((134-122)/2) AND \$0F)	
DFF4 2E		FCB	((((86-82)/2) AND \$0F)*16) OR (((130-134)/2) AND \$0F)	
DFF5 E2		FCB	((((82-86)/2) AND \$0F)*16) OR (((124-130)/2) AND \$0F)	
DFF6 11		FCB	((((84-82)/2) AND \$0F)*16) OR (((136-134)/2) AND \$0F)	
DFF7 20		FCB	((((88-84)/2) AND \$0F)*16) OR (((136-136)/2) AND \$0F)	
DFF8 2E		FCB	((((92-88)/2) AND \$0F)*16) OR (((132-136)/2) AND \$0F)	
DFF9 22		FCB	((((96-92)/2) AND \$0F)*16) OR (((136-132)/2) AND \$0F)	
DFFA 20		FCB	((((100-96)/2) AND \$0F)*16) OR (((136-136)/2) AND \$0F)	

DFFF 00FE      SVEND      V\$ABS,V\$END

End of Viper vector list

\* \* \* \* \*

END

UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
KSK - Good Use of leftover Bytes

PAGE 79

```
DFFD 4B534B      INCLUD B:KSK.ASM
                  FCC    'KSK'      ;three more extra bytes!!!
                  END
;
; End of program
0000      END
```

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
---- SYMBOL TABLE ----

PAGE 80

A\$BEPO	0003	BEOOP	C83C	C.SUB	001A	CLK50	C2FF	CWLK99	D199
A\$EXP0	0015	BEOOP1	C83F	C.SYN	0016	CLK60	C318	DO	02C9
A\$EXP1	0016	BIGONE	0201	C.US	001F	CLK90	C320	DO\$BAS	1000
A\$FLAS	000C	HIGZER	0200	C.VT	0008	CLOCK	C27D	DO\$END	2300
A\$KLKO	0004	BIT0	0001	CBIR00	CFAA	CLRPR1	000E	DO.LEN	1300
A\$KLK1	0007	BIT1	0002	CBIR20	CFC4	CLRSTS	C507	DO.SAM	2046
A\$KLK2	0012	BIT2	0004	CHIRTH	CFA5	CLRSTX	C4FF	D1	02CA
A\$KLN3	0013	BIT3	0008	CC.C	0001	CLRSUB	C50D	D1\$BAS	2800
A\$FSHT	0008	BIT4	0010	CC.E	0080	CMDERR	CBE1	D1\$END	3800
A\$RING	000D	BIT5	0020	CC.F	0040	CMDNUM	000F	D1.LEN	1300
A\$ROR0	0002	BIT6	0040	CC.H	0020	CMDTAB	DB94	D1.SAM	2046
A\$ROR1	0005	BIT7	0080	CC.I	0010	CMOV10	DO4D	D2	02CB
A\$ROR2	0009	BITMSK	CB8E	CC.LEN	0011	CMOV12	DO6A	D3	02CD
A\$RTLO	0001	BLKIN	A006	CC.N	0008	CMOV20	DO6D	D4	02CD
A\$RTL1	0006	BLKOUT	A008	CC.V	0002	CMOV30	DO99	D5	02CE
A\$SCRO	000E	BLLOB	DE59	CC.Z	0004	CMOV50	DOB2	D6	02CF
A\$SHIE	000F	BOOM1	C895	CCBEND	05F4	CMOV52	DOC3	D7	02D0
A\$SQNO	0000	BOOM2	C898	CCBLND	03D4	CMOV60	DO0D	DAMAGE	040C
A\$SQN1	000A	BOOMER	C893	CD.LEN	0008	CMOV62	DO4D	DAY	029A
A\$SQN2	000B	BORD99	CC9A	CDTAB	DABB	CMOV70	DOE4	DCOL	027D
A\$SWOR	0010	BORDER	CC8E	CELCCNT	027F	CMOV72	DOE4	DEATH	C5B5
A\$THUD	0014	BUFFLG	02BE	CELINE	DD3C	CMOV74	DOF4	DEBFLG	0000
A\$TORC	0011	BURN10	D1B0	CFIND	CF82	CMOV76	DOF6	DEBMAP	D7A2
ADJNUM	0019	BURN20	D1B6	CFND10	CF85	CMOV78	D101	DELTAX	0245
ADJTAB	I8F3	BURN99	D1BC	CFND30	CF92	CMOV90	D103	DELTAY	0246
ASCA10	C40A	BURNER	D19B	CFND99	CF96	CMOV92	D10F	DEMDAT	D7D5
ASCALX	C406	CACK	0006	CGEN10	D02C	CMOV98	D111	DEMO	C000
ASCALY	C400	CC.BEL	0007	CGEN99	D03D	CMOV99	D113	DEMO10	C011
ASRD	I383	CC.BS	0008	CHAR00	C4AA	CMOVE	D041	DGEN10	CCRB
ASRD2	I381	CC.CAN	0018	CHAR02	C4AC	CMRDWR	CECB	DGEN20	CCC8
ASRD3	I37F	CC.CR	000D	CHAR04	C4AD	CMTEND	D868	DGEN30	CCB2
ASRD4	I37H	CC.DC1	0011	CHAR06	C4AE	CMTTAB	D82C	DGEN50	C011
ASRD5	I37B	CC.DC2	0012	CHAR10	C4B0	CMXEND	03D4	DGEN60	C02D
ASRD6	I379	CC.DC3	0013	CHAR20	C4B5	CMXLND	0398	DGEN62	C033
ASRD7	I377	CC.DC4	0014	CHAR30	C4B9	CMXPTR	0282	DGEN64	C039
AT.LEN	000C	CC.DEL	007F	CHAR40	C4BE	COL	0289	DGEN66	C03F
ATTACK	D3D7	CC.DLE	0010	CHAR50	C4C3	COMINI	C008	DGEN70	CD41
ATTK10	D3E4	CC.EM	0019	CHAR60	C4C7	COMRTS	C3A1	DGEN80	C056
ATTK20	D3EC	CC.ENQ	0005	CHAR70	C4C0	COMRTS1	CE2A	DGEN82	C060
ATTK22	D3FB	CC.EOT	0004	CHRROUT	A002	COMRTS2	CEB1	DGEN90	C067
ATTK30	D3FE	CC.ESC	0018	CHUCK	C8B2	COMUPD	D5B7	DNGEN	CC9C
ATTK99	D40A	CC.ETB	0017	CINI10	C030	COPY	C04B	DIR	028A
AUTEND	D9CF	CC.ETX	0003	CINI20	C041	COPY\$	C63E	DIRNUM	0004
AUTFLG	0277	CC.FF	000C	CINI40	C086	COPY10	C640	DIRTAB	D8D9
AUTPTR	0200	CC.FS	001C	CINI42	C08C	COPY20	C648	DISPAT	D9D0
AUTTAB	D988	CC.GS	001D	CINI44	C09A	CREGEN	D027	DIV00	CA7C
BAGPTR	0229	CC.HT	0009	CINI46	C0A5	CSETUP	C915	DIV10	CA7F
BAKSCL	CF49	CC.LF	000A	CLANG	C8DA	CSRIDON	A004	DIV20	CA93
BAKSTP	0274	CC.NAK	0015	CLANK	C8E6	CTYPES	000C	DIV99	CA97
BALROG	DIA3	CC.NUL	0000	CLEAR	C517	CWALK	D14F	DIVIDE	CA67
BANG	C86D	CC.NUL	000E	CLEAR1	C520	CWLK10	D16F	DORTAB	CDAA
BANGO	DRDA	CC.ST	000F	CLK20	C59D	CWLK12	D174	DP.BEG	0200
BASICX	C691	CC.SO	000E	CLK30	C2A9	CWLK20	D17C	DP.END	02D1
BDLB10	C87B	CC.SOH	0001	CLK32	C2D1	CWLK90	D196	DP.LEN	0001
BDLB12	C882	CC.SP	0020	CLK40	C2DC	DPR00	C44	DP.R00	C44

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
--- SYMBOL TABLE ---

PAGE 81

DPB20	CA5F	FREQ2	0265	HMAN70	D2A7	KLANK	C8E2	M\$GET	D8A9
DRAWIT	CECE	FRIEND	CC49	HMAN99	D2B4	KLINK	C93F	M\$INCN	D8AD
DROW	0270	FRING	DC21	HNDPAR	D576	KNIGHT	DEB3	M\$LOOK	D8B2
DRWFLG	0251	FRZFLG	022B	HOLEIN	DD2A	KNIGT1	DE82	M\$LT	D8DA
DSFO	D870	FSCROL	DC26	HOLEUP	DD0E	KNIGT2	DE9D	M\$MOVE	D8B6
DSP1	D876	FSDOOR	DCA9	HODQUE	D2A9	LADDER	DC16	M\$PROM1	C67A
DSP32	CE11	FSHIEL	DBFA	HOUR	D299	LODOOR	DC4B	M\$PULL	D8BA
DSPMOD	02B2	FSUR10	CC56	HSLOW	D105	LENGTH	D243	M\$REVE	D8BE
DST	027E	FSUB20	CC60	HSLOW1	D1E0	LEVEL	D281	M\$RINO	D970
DX	0230	FSUB22	CC6B	HSLOW2	D1E2	LINBUF	02F1	M\$RN01	D8FA
DY	0240	FSWORD	DC0F	HUMAN	D24C	LINEND	0311	M\$RN02	D92E
E.WALL	000C	FTORCH	DC07	HUPD00	D529	LINES	D6C6	M\$RN03	D913
EGAM10	D3C4	FULFLG	D27B	HUPD10	D531	LINFTR	0211	M\$RN05	D8F4
EMPHND	0E07	FWALL	DC45	HUPD20	D554	LNEW10	D1CD	M\$RN11	D950
ENDGAM	D386	FWDCRE	DA63	HUPD30	D578	LOAD	C1C1	M\$RN12	D959
ERRCMD1	D573	FWDOBJ	D9EF	HUPD32	D589	LOAD10	C1C6	M\$RN13	D955
ERRCMD2	D693	FWDVER	DCC2	HUPD40	D595	LOAD12	C1D5	M\$RN15	D94B
EXAM10	D4C0	G6.LEN	1800	HUPD42	D599	LOAD20	C1E4	M\$RN20	D950
EXAM12	D4C2	GAMBAT	D7D9	HUPD90	D5AE	LOAD90	C1EC	M\$RT	D8DE
EXAM20	D4CD	GAME	C005	HUPDAT	000C	LPASAG	DC4F	M\$SCR0	D924
EXAM24	D4D3	GAME10	C124	HUPDAX	D529	LPEEK	DCB0	M\$SCR1	D90A
EXAM26	D4D6	GAME20	C131	I.BAR	001C	LRTU10	D67B	M\$SCR2	D917
EXAM30	D4ED	GAME30	C139	I.BS	0024	LRTU99	D6B3	M\$SHI0	D979
EXAM32	D4F7	GAME40	C14F	I.CR	001F	LRTURN	D674	M\$SHI1	D904
EXAM39	D4FB	GAME50	C166	I.DOT	001E	LSDOOR	DC9B	M\$SHI3	D929
EXAMIN	D495	GENNUM	0004	I.EXCL	001B	LSLD1	CA14	M\$SHI4	D940
EXAMIO	D489	GENTAB	D96A	I.LHL	0022	LSLD2	CA12	M\$STOW	D8C3
EXP400	C472	GENVAL	C719	I.LHR	0023	LSLD3	CA10	M\$SW00	D97E
EXP410	C47A	GETBUF	C16D	I.GUES	0010	LSLD4	CA0E	M\$SW01	D8FF
EXP490	C489	GETFI4	C48C	I.SHL	0020	LSLIS5	CA0C	M\$SW02	D933
EXPANO	0006	GETTCB	C25C	I.SHR	0021	LUKNEW	D1C2	M\$SW03	D946
EXPAND	0005	GETTOK	CB96	I.SP	0000	LVL0	0000	M\$T080	D983
EXPANX	C46F	GLUGLG	C80A	INCR10	CAAE	LVL1	0001	M\$TOR1	D924
FADCNT	022E	GRAWL	C955	INCR90	DAB0	LVL2	0002	M\$TOR2	D937
FADFLG	029E	GROWL	C951	INCRE	CA9F	LVL3	0003	M\$TOR4	D93C
FAINT	0228	GTOK10	CB9D	INIVU	0019	LVL4	0004	M\$TORS	D966
FCB	0003	GTOK20	CBA3	INIVUX	C743	LVL5	0005	M\$TURN	D8C7
FCC	0003	GTOK22	CBA5	IRQSYN	C114	LVLPTB	0284	M\$UP	D8EC
FCEL10	CF99	GTOK30	CBAF	ISOUND	001B	LVLTAB	CD9F	M\$USE	D8CB
FDB	0004	HBEATF	02B1	JIFFY	0295	LWALL	DC33	M\$ZLOA	D8CF
FLOOR	DC8B	HEARTC	02AE	JIFQUE	02A1	M\$AROU	D8E7	M\$ZSAV	D804
FFFLASK	DC19	HEARTF	02AB	JOYIN	A00A	M\$ATTK	D895	MAGFLG	0275
FILNAM	D7BC	HEARTR	02AF	K.FLAS	0000	M\$BAK	D8E3	MAKIOR	CD6D
FIND10	CF72	HEARTS	02B0	K.RING	0001	M\$CLIM	D89A	MAP32	CC7B
FIND99	CF80	HF.DOR	0001	K.SCRO	0002	M\$CURS	C67C	MAPFLG	0294
FIVDSP	C4A2	HF.PAS	0000	K.SHIE	0003	M\$DN	D8EF	MAPF10	CDB9
FLATAB	DBDE	HF.SDR	0002	K.SWDR	0004	M\$DROP	D89F	MAPF20	CDC7
FLBN	DCB0	HF.WAL	0003	K.TORC	0005	M\$EMPT	C650	MAPF22	CDC9
FLIP	0209	HLFSCL	CF48	KABOOM	C88A	M\$ERAS	D28C	MAPF30	CDE3
FLOP	020B	HLFSTP	D273	KBDUF	02D1	M\$EXAM	D8A3	MAPF40	CDF7
FLUP	DCCA	HMAN10	D24E	KBDGET	C329	M\$FLAO	D96R	MAPF42	CDF8
FNDCEL	CF97	HMAN20	D256	KBDHDR	02BC	M\$FLA1	D90E	MAPF50	CE2B
FNDOBJ	CF63	HMAN30	D26F	KBDPUT	C340	M\$FLA2	D920	MAPPER	CDE38
FOO	0006	HMAN40	D270	KBDTAL	D2BD	M\$FLA3	D91C	MARK4	CE1D
FPASAG	DC60	HMAN50	D292						

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
---- SYMBOL TABLE ----

PAGE 82

MAZLNID	05F4	OFF	0000	P.OCXXX	0006	PDL	1000	PREVU	D66D
MDOR10	C073	OFIL10	C730	P.ODCLS	0000	PDRAW	CEB2	PRHAND	021F
MINQUE	02A7	OFIL99	C742	P.ODMGO	0002	PDRP	D550	PRIVDR	D888
MINUTE	0298	OFIND	CF53	P.ODPHO	0003	PEXAM	D481	PRLITE	0226
MLIGHT	026F	OFIND9	CF82	P.ODREV	0001	PGET10	D526	PROMPT	000F
MM.BEG	0201	OFINDF	0291	P.PICRA	0001	PGET20	D528	PROMPX	C674
MM.END	0F05	OFINDP	0292	P.PICRB	0003	PGET22	D530	PROW	0213
MOVTAB	D114	OLIGHT	0270	P.PIJOA	0000	PGET24	D540	PRTOBJ	D505
MSKTAB	CDA6	OMXEND	DA93	P.PIJOB	0002	PGET30	D542	PSSHT	C84D
MSQUEQ	C81F	OMXTAB	DA91	P.TCITA	0005	PHAS1	C818	PSTEP	D720
N.WALL	0003	ONAM10	C622	P.TCPTR	0000	PHAS2	C816	PSTOW	D579
NDPB10	CA39	ONAM20	C632	P.TCRTN	0003	PHASER	C811	PSTOWO	D570
NEGD	CA99	ONCE	C000	P.TCTIM	0002	PHND90	CC46	PSTP10	D738
NEGONE	0203	OROW	0271	P.TXBCS	0000	PIA\$0	FF00	PSTP99	D730
NEGRAM	C608	OUTCHR	0004	P.TXCNT	0002	PIA\$1	FF20	PSTS90	COCE
NEGRAX	0212	OUTSTI	0002	P.TXCUR	0004	PIATAP	C17C	PSUB10	C66C
NEIBOR	0FF4	OUTSTR	0003	P.TXINV	0006	PINC10	D510	PTORCH	0224
NEWLIN	02B6	P.ATDAM	000A	P.TXSGL	0007	PINC99	D5EF	PTUR10	D654
NEWLUK	02B5	P.ATMGO	0003	P.UDBAS	0000	PINCAN	D5BC	PTUR20	D650
NEWLVL	001A	P.ATMGO	0002	P.UDEND	0002	PLAY10	D1EF	PTUR22	D667
NEWLVLX	C759	P.ATPHD	0005	P.UDSAM	0004	PLAY12	D212	PTUR90	D669
NEWOLD	C3F7	P.ATPHO	0004	PARCNT	0279	PLAY14	D215	PTURN	D63F
NLVL10	C768	P.ATPOW	0000	PARRERR	CBDF	PLAY20	D21B	PULL10	D593
NLVL12	C76A	P.ATXXX	0006	PARFLG	0278	PLAY30	D229	PULL12	D5A3
NLVL30	C783	P.CCCOL	0010	PARNHD	CC31	PLAY40	D235	PULL14	D5A7
NLVL32	C787	P.CCDAM	000A	PAROBJ	CRBA	PLAY50	D241	PUDF99	C65F
NLVL34	C78D	P.CCDIR	000E	PARS02	DBF4	PLAY99	D248	PUPDAT	000E
NLVL40	C795	P.CCMGD	0003	PARS10	CBFC	PLAYER	D1EE	PUPDAX	C656
NLVL42	C79E	P.CCMGO	0002	PARS12	CC05	FLHAND	D21D	PUPSUB	C660
NLVL44	C7AA	P.CCOBJ	0008	PARS20	CC17	FLLOOK	C751	PUSE	D741
NLVL50	C7B6	P.CCPHD	0005	PARS30	CC22	FLRBLK	0217	PUSE10	D748
NOISEF	029C	P.CCPHO	0004	PARS90	CC2B	PMGD	021A	PUSE12	D74E
NOISEU	029D	P.CCPOW	0000	PARS92	CC2D	PMGO	0219	PUSE20	D757
NORDPP	CA29	P.CCROW	000F	PARS99	CC2F	PMLITE	0227	PUSE22	D75C
NORSCL	CF3E	P.CCTAT	0007	PARSEO	CBE7	PMOV20	D6E3	PUSE24	D767
NULQUE	029F	P.CCTMV	0006	PARSER	CBEC	PMOV22	D6EF	PUSE30	D768
OBIR99	C718	P.CCTYP	000D	PASFGL	0276	PMOV30	D6F3	PZLOAD	D7B7
OBIRTH	0017	P.CCUSE	000C	PATT10	D2C2	PMOV40	D701	PZSAVE	D7C7
OBIRTX	C4FB	P.COMGD	0003	PATT20	D2F7	PMOV90	D70E	Q.HOU	000A
OBJCLS	028F	P.COMGO	0002	PATT22	D319	PMOVE	D6D0	Q.JIF	0002
OBJCNT	028C	P.COPHD	0005	PATT24	D31F	POBJ10	CBC8	Q.MIN	0008
OBJJLV	028D	P.COPHO	0004	PATT30	D32C	POBJ101	D51E	Q.NUL	0000
OBJNAM	C617	P.COPOW	0000	PATT32	D32E	POBJ99	CBDE	Q.SCD	000C
OBJSTR	0360	P.CDTAT	0007	PATT40	D33A	POBJ990	D520	Q.SEC	0006
OBJTYP	028E	P.CDTMV	0006	PATT42	D351	POBJWT	0215	Q.TEN	0004
OBJWGT	B9FA	P.CCLS	000A	PATT99	D375	POLCAT	A000	QUEAD1	C22B
OC.LEN	000E	P.CCOL	0003	PATTK	D2B8	PPHD	021C	QUEAD2	C234
OCBEND	0F05	P.CCLVL	0004	PCLI00	D46F	PPHO	021B	QUEADO	C21D
OCBFIL	0018	P.CGMGO	000C	PCLI10	D472	PPOW	0217	QUEBEG	D29F
OCBFIIX	C71F	P.CCOWN	0005	PCLI20	D478	PPULL	D589	QUEEND	D2A8
OCBLND	0B15	P.CCPHO	000B	PCLIMB	D44E	PREPAR	0016	QUERMV	C238
OCBPTR	020F	P.CCPTR	0000	PCOL	0214	PREPAX	C6E6	QUESCO	C248
OCOL	0272	P.CCREV	000B	PCRLF	D4FE	PREV00	B638	QUESC9	C25A
OD.LEN	0004	P.CCROW	0002	PD.LEN	00FB	PREV99	D63E	QUESCN	C242
ODRTAR	DA00	P.CCTYP	0000	PDAM	0221	PREVFA	D677	PAMPA7	D7E8

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
---- SYMBOL TABLE ----

PAGE 83

RANDOM	0007	SNARL1	C95C	STATUS	C5D9	T.SH14	0010	TXTCR	C9CA
RANDOX	C4CF	SNBAS1	C80A	STCB10	C05A	T.STOW	000A	TXTDPB	CA17
RANGE	C28B	SNCLK1	C8FD	STCB20	C066	T.SWO0	0004	TXTEXA	0380
RATTLE	C850	SNCLK2	CBF3	STCB30	C076	T.SWO1	0002	TXTFRI	0390
RDOOR	DC7B	SNCLK3	C90A	STCB99	C084	T.SWO2	000D	TXTSCR	C9D4
RLIGHT	026E	SNCLK4	C8C3	STEP	D11B	T.TOR0	0005	TXTSTI	C448
RLTU10	D68B	SNCLK5	C8EA	STEPOK	D136	T.TOR1	000A	TXTSTR	C454
RLTU99	D692	SNDLAY	0262	STOK98	D14C	T.TOR2	000E	TXTSTS	0388
RLTURN	D684	SNDOBJ	000C	STOK99	D14D	T.TOR4	000F	TXTXXX	C9E2
RMB	0002	SNDRAM	0256	STPTAB	D12E	T.TOR5	0018	TY	0254
RND1	C4D2	SNDRND	0256	STRING	0334	T.TURN	000B	UFL100	D77A
RND2	C4DB	SNDTAB	C7DC	STSVD8	D87C	T.UP	0004	UFL200	D783
RND3	C4DF	SNENV	C97E	SW2SER	C371	T.USE	000C	UFL300	D787
RNDCEL	CC71	SNENV1	C984	SWCHAR	0356	T.ZLOA	000D	UFL310	D790
ROLTAB	C324	SNENV2	C98D	SWCTAB	DB1B	T.ZSAV	000E	UFL900	D792
ROW	0288	SNENV3	025D	SWIBEG	C384	TO	02C1	UPDATE	02B4
RPASAG	DC5D	SNENVN	C926	SWIJMP	02BF	T1	02C2	USC100	D7A0
RPEEK	DCB9	SNENVS	025B	SWISER	C352	T2	02C3	USC199	D7B6
RSOOCR	DCA2	SNENVT	C928	SWITAB	C995	T3	02C4	USC200	D7A2
RSTART	02BB	SNGRL1	C962	SYSTCB	C053	T4	02C5	USC210	D7AD
RWALL	DC3C	SNGRL2	C971	T.AROU	0003	T5	02C6	USEEND	D77A
S.WALL	0030	SNGRL3	C974	T.ATTR	0000	T6	02C7	USETAB	D76B
SAM	C266	SNKLK1	C942	T.BAK	0002	T7	02C8	V\$ABS	0000
SAM10	C26B	SNLXT1	C93E	T.CLIM	0001	TAPDLY	C18B	V\$END	00FE
SAM12	C272	SNOISE	C8CE	T.DN	0005	TC.LEN	0007	V\$IRQ	010C
SAVE	C192	SNOTE1	0267	T.DROP	0002	TCBDA	D7DC	V\$JMP	00FD
SAVE10	C1A6	SNOTE2	0269	T.EXAM	0003	TCREND	0B07	V\$JSR	00FB
SCAL00	C40C	SNOUT	C8C5	T.FLA0	0000	TCBLND	09FD	V\$NEW	00FF
SCAL10	C40F	SNRAT1	C858	T.FLA1	0005	TCBPTR	02B9	V\$NMI	0109
SCAL16	D436	SNRAT2	C854	T.FLA2	0009	TCHR10	C460	V\$REL	00FC
SCAL20	C414	SNSAV	0259	T.FLA3	0008	TCHR99	C46C	V\$RTS	00FA
SCDQUE	02AB	SNSAVA	025F	T.FLA4	0017	TDLY10	C18D	V\$SWI	0106
SCHEP	C1F5	SNSQK1	C82E	T.GET	0004	TENQUE	02A3	V\$SWI2	0103
SCHEPD	C1FA	SNSQK2	C835	T.INCN	0005	TENTH	0294	V\$SWI3	0100
SCHEP1	C1FC	ONSUB2	C869	T.LOOK	0006	THUD	C872	VABSO0	C3B9
SCORF	DE3F	ONSUB3	C867	T.LT	0000	THUDD	DB02	VABS10	C3E0
SDRTAB	CDAE	SNVOL	0261	T.MOVE	0007	TIMBEG	0295	VABSO20	C3E2
SECOND	0297	SNWAIT	C8BD	T.FULL	0008	TIMEND	029B	VCNTRX	0205
SECQUE	02A5	SNWAT1	C8BF	T.REVE	0009	TOKEN	0313	VCNTBY	0207
SEED	026B	SNWSH1	C8B6	T.RINO	0001	TOKEND	0333	VCTABS	C3CF
SETFAD	0000	SNWSH2	C8AA	T.RN01	0001	TSCR10	C9E3	VCTDIS	C3B9
SETFAX	C384	SNWT1K	C8BA	T.RN02	000C	TSCR12	C9EF	VCTEND	C3F6
SETNVA	C931	SNZNVA	C922	T.RN03	0006	TSCR20	C9FC	VCTFAD	022D
SETNVD	C92E	OUNDI	C7C8	T.RN05	0000	TSCR22	CA04	VCTJMP	C3C6
SETRAX	C68A	SONUDS	001C	T.RN11	0013	TSTR10	C452	VCTJSR	C3BF
SETSCL	CE47	SONUNDX	C7D0	T.RN12	0015	TURN00	D696	VCTL10	C3A9
SFAD10	C38E	SPACE1	C610	T.RN13	0014	TURN02	D6B9	VCTLST	0001
SFAD30	C39F	SPACES	C609	T.RN15	0012	TURN10	D6BA	VCTLSX	C3A2
SGIANT	DD62	SPCTAB	DBB6	T.RN20	0016	TURN12	D6C0	VCTNEW	C3CB
SGINT1	DD41	SPEFLG	0290	T.RT	0001	TX	0252	VCTREL	C417
SGINT2	DB51	SPIDER	BE26	T.SCR0	0002	TX.LEN	0008	VCTRTS	C3C9
SHIELD	DD9E	SQUEAK	C82B	T.SCR1	0004	TXBFLG	02B7	VD.LEN	0006
SHLD99	DOB0	SSCL10	CE5C	T.SCR2	0007	TXTBS	C9BF	VDIAS	0247
SKIP2	008C	SSER10	C360	T.SHI0	0003	TXTBS1	C9C9	VDEND	0249
SLEEP	029B	STAT10	CSFD	T.SHI1	0003	TYTCH	C9C9	WTCTM1	022C
SNARI	CEF0	STAT10	0000	T.SHTZ	0000				

## UNIFIED TECHNOLOGIES 6809 ASSEMBLER - VERSION 1.08

DAGGORATH V3.2 - Copyright (c) 1983 Unified Technologies Inc.  
---- SYMBOL TABLE ----

PAGE 84

VECT00	CAB4	VF.LDN	0003	VIEW99	CF3D	WHOOSH	C8A6	WRTLDR	A00C
VECT12	CACB	VF.LUP	0001	VIEWER	CE66	WINNER	D5F0	WUPDAT	D56B
VECT14	CAE7	VFIND	CFE1	VIFER	DFCA	WIZO	DF65	XO	0231
VECT20	CAE0	VFIND9	CFEE	VNULL	DD40	WIZI	DF10	X1	0235
VECT22	CAF0	VFTPTR	0286	VREL20	C438	WIZZ	INF39	XX	0237
VECT24	CB02	VFTTAB	CFFD	VSUB00	CFF2	WIZI10	C4B7	XXXTAB	DA64
VECT30	CB2E	VIEW10	CE6E	VSUB99	CFFC	WIZI20	C6C1	YO	022F
VECT32	CB50	VIEW12	CE7B	VXOFF	024B	WIZIN	0013	Y1	0233
VECT40	CB54	VIEW20	CE92	VXSCAL	024F	WIZINO	0014	YY	023A
VECT50	CB70	VIEW22	CE96	VYOFF	024D	WIZIX	C6A4	ZERO	0011
VECT60	CB96	VIEW24	CEA9	VYSCAL	0250	WIZIXO	C6A8	ZEROX	C6B6
VECT99	CBBA	VIEW30	CED8	W.WALL	00C0	WIZO10	C6CC	ZFLAG	02B8
VECTOR	CAB7	VIEW32	CEE8	WAIT	0010	WIZOUT	0015	ZFLIP	0008
VERIFY	027A	VIEW40	CF09	WAIT10	C6B1	WIZOX	C6C5	ZFLIPX	C4F3
VF.HDN	0002	VIEW52	CF0D	WAITX	C67F	WIZZES	C6D7	ZFLOP	0009
VF.HUP	0000	VIEW60	CF24	WHOOP	C827	WRAITH	DE07	ZFLOPX	C4F6