Greed#

General Information

Author: Carsten Strotmann, Winfried Piegsda
Language: C
Compiler/Interpreter: CC65
Published: ABBUC 2004

Download: Greed/agreed2.atr

Title Tune: Greed/greed.mp3 Greed/greed.mid.mov

Atari Greed#

Atari Greed scored 6th place at the 2004 ABBUC Software contest

Instructions #

ATARI Greed

based on the UNIX game "Greed" by Matthew T. Day and Eric S. Raymond.

Written for the 2004 ABBUC Software contest by Winfried Piegsda and Carsten Strotmann.

Goal: Numbers from 1 t 9 will be shown in an area of 38 x 18 fields. The round spot is the pawn in the game. The player must try to collect as many numbers as possible before running out of moves. For every field moved the score will increase by one.

The pawn can be moved by Joystick or Keyboard in six directions. The range of the move will be calculated by the first number to be collected. All possible moves can be displayed by pressing the 'P' key.

There is a time running out for each level. To enter the next level, a predefinded amount (percentage) of the area must be cleaned. The remaining percentage will be displayed in the lower right corner near the score and highscore.

Will the level treshold overshoot a bonus will be added to the score.

The menu can be displayed with the 'ESC' Key.

have fun with Atari Greed

Winfried Piegsda and Carsten Strotmann

Anleitung #

ATARI Greed

basierend auf dem UNIX Spiel "Greed" von Matthew T. Day und Eric S. Raymond.

Programmiert fuer den ABBUC Programmier- wettbewerb 2004 von Winfried Piegsda und Carsten Strotmann.

Ziel des Spiels: Auf einer Flaeche von maximal 38 x 18 Zellen werden Zahlen von 1 bis 9 dargestellt. Ein runder Ball in dieser Flaeche stellt die Spielfigur da. Der Spieler muss versuchen moeglichst viel Punkte zu sammeln, den dem er Felder ueberquert und loescht. Fuer jedes geloeschte Feld wird die Punktzahl erhoeht.

Mit dem Joystick oder der Tastatur kann die Spielfigur in sechs Richtungen bewegt werden. Die Schreittweite in jede Richtung wird durch die direkt an die Spielfigur angrenzende Zahl bestimmt. Die moeglichen Zuege koennen mit der Vorschaufunktion (p - Preview) angezeigt werden.

Fuer jeden Level stehen eine bestimmte Zeit in Minuten zur Verfuegung. Um in einen neuen Level zu wechseln, muss die Mindestanzahl an Feldern abgeraumt werden. Die Prozentzahl der noch zu loeschenden Felder wird in der Statuszeile rechts neben den Punkten und der Hoechstpunktzahl angezeigt.

Wir diese Prozentzahl untersschritten so wird ein Bonus am Ende des Levels auf die Punktzahl addiert.

Mit der 'ESC' Taste kann das Menue aufgerufen werden.

Viel Spass mit Atari Greed wuenschen Winfried Piegsda Carsten Strotmann

1.1 Make Scripts

```
export CC65_LIB=~/develop/cc65/libsrc
export CC65_INC=~/develop/cc65/include

cl65 -l -O --add-source -m agreed.map -t atari agreed.c dli1.s
cp agreed ~/atari/Atari800MacX/Harddrive1/agreed.com


export CC65_LIB=~/develop/cc65/libsrc
export CC65_INC=~/develop/cc65/include

ca65 -t none level00.s
cl65 -t none level00.o -o ~/atari/Atari800MacX/Harddrive1/level00.agl
ca65 -t none level01.s
cl65 -t none level01.o -o ~/atari/Atari800MacX/Harddrive1/level01.agl
ca65 -t none level02.s
cl65 -t none level02.o -o ~/atari/Atari800MacX/Harddrive1/level02.agl
ca65 -t none level03.s
cl65 -t none level03.o -o ~/atari/Atari800MacX/Harddrive1/level03.agl
ca65 -t none level04.s
cl65 -t none level04.o -o ~/atari/Atari800MacX/Harddrive1/level04.agl
ca65 -t none level05.s
cl65 -t none level05.o -o ~/atari/Atari800MacX/Harddrive1/level05.agl
ca65 -t none level06.s
cl65 -t none level06.o -o ~/atari/Atari800MacX/Harddrive1/level06.agl
ca65 -t none level07.s
cl65 -t none level07.o -o ~/atari/Atari800MacX/Harddrive1/level07.agl
ca65 -t none level08.s
cl65 -t none level08.o -o ~/atari/Atari800MacX/Harddrive1/level08.agl
ca65 -t none level09.s
cl65 -t none level09.o -o ~/atari/Atari800MacX/Harddrive1/level09.agl

Modules list:
-------------
atari.o:
    CODE              Offs = 000000   Size = 00008F
```

```
    BSS               Offs = 000000   Size = 000004
    DATA              Offs = 000000   Size = 00001A
    EXEHDR            Offs = 000000   Size = 000006
    AUTOSTRT          Offs = 000000   Size = 000006
agreed.o:
    CODE              Offs = 00008F   Size = 002534
    RODATA            Offs = 000000   Size = 0004DD
    BSS               Offs = 000004   Size = 0003A8
    DATA              Offs = 00001A   Size = 00010D
dli1.o:
    CODE              Offs = 0025C3   Size = 000C1F
/Users/cas/develop/cc65/libsrc/atari.lib(cgetc.o):
    CODE              Offs = 0031E2   Size = 000051
/Users/cas/develop/cc65/libsrc/atari.lib(clock.o):
    CODE              Offs = 003233   Size = 000023
/Users/cas/develop/cc65/libsrc/atari.lib(close.o):
    CODE              Offs = 003256   Size = 00001E
/Users/cas/develop/cc65/libsrc/atari.lib(ctype.o):
    RODATA            Offs = 0004DD   Size = 000100
/Users/cas/develop/cc65/libsrc/atari.lib(fdtable.o):
    CODE              Offs = 003274   Size = 0001BA
    BSS               Offs = 0003AC   Size = 000006
/Users/cas/develop/cc65/libsrc/atari.lib(getfd.o):
    CODE              Offs = 00342E   Size = 000029
    DATA              Offs = 000127   Size = 00002C
/Users/cas/develop/cc65/libsrc/atari.lib(graphics.o):
    CODE              Offs = 003457   Size = 000081
    RODATA            Offs = 0005DD   Size = 000043
/Users/cas/develop/cc65/libsrc/atari.lib(graphuse.o):
    DATA              Offs = 000153   Size = 000001
/Users/cas/develop/cc65/libsrc/atari.lib(mul40.o):
    CODE              Offs = 0034D8   Size = 000020
    BSS               Offs = 0003B2   Size = 000001
/Users/cas/develop/cc65/libsrc/atari.lib(open.o):
    CODE              Offs = 0034F8   Size = 0000B7
/Users/cas/develop/cc65/libsrc/atari.lib(read.o):
    CODE              Offs = 0035AF   Size = 00002C
/Users/cas/develop/cc65/libsrc/atari.lib(rwcommon.o):
    CODE              Offs = 0035DB   Size = 000056
/Users/cas/develop/cc65/libsrc/atari.lib(ucase_fn.o):
    CODE              Offs = 003631   Size = 000061
/Users/cas/develop/cc65/libsrc/atari.lib(write.o):
    CODE              Offs = 003692   Size = 000028
/Users/cas/develop/cc65/libsrc/atari.lib(_fdesc.o):
    CODE              Offs = 0036BA   Size = 00001C
/Users/cas/develop/cc65/libsrc/atari.lib(_file.o):
    DATA              Offs = 000154   Size = 000016
/Users/cas/develop/cc65/libsrc/atari.lib(_fopen.o):
    CODE              Offs = 0036D6   Size = 000080
    BSS               Offs = 0003B3   Size = 000002
/Users/cas/develop/cc65/libsrc/atari.lib(_hextab.o):
    RODATA            Offs = 000620   Size = 000010
/Users/cas/develop/cc65/libsrc/atari.lib(_oserror.o):
    BSS               Offs = 0003B5   Size = 000001
/Users/cas/develop/cc65/libsrc/atari.lib(_printf.o):
    CODE              Offs = 003756   Size = 0003A1
    BSS               Offs = 0003B6   Size = 00002B
    DATA              Offs = 00016A   Size = 000003
/Users/cas/develop/cc65/libsrc/atari.lib(errno.o):
```

```
    BSS               Offs = 0003E1   Size = 000002
/Users/cas/develop/cc65/libsrc/atari.lib(fclose.o):
    CODE              Offs = 003AF7   Size = 000026
/Users/cas/develop/cc65/libsrc/atari.lib(fopen.o):
    CODE              Offs = 003B1D   Size = 00001D
/Users/cas/develop/cc65/libsrc/atari.lib(fread.o):
    CODE              Offs = 003B3A   Size = 000092
    BSS               Offs = 0003E3   Size = 000002
/Users/cas/develop/cc65/libsrc/atari.lib(fwrite.o):
    CODE              Offs = 003BCC   Size = 000082
    BSS               Offs = 0003E5   Size = 000002
/Users/cas/develop/cc65/libsrc/atari.lib(ltoa.o):
    CODE              Offs = 003C4E   Size = 0000AF
    RODATA            Offs = 000630   Size = 00000C
/Users/cas/develop/cc65/libsrc/atari.lib(memcpy.o):
    CODE              Offs = 003CFD   Size = 00003A
/Users/cas/develop/cc65/libsrc/atari.lib(memmove.o):
    CODE              Offs = 003D37   Size = 00003C
/Users/cas/develop/cc65/libsrc/atari.lib(memset.o):
    CODE              Offs = 003D73   Size = 00003F
/Users/cas/develop/cc65/libsrc/atari.lib(rand.o):
    CODE              Offs = 003DB2   Size = 00004C
    DATA              Offs = 00016D   Size = 000004
/Users/cas/develop/cc65/libsrc/atari.lib(sprintf.o):
    CODE              Offs = 003DFE   Size = 00002B
    BSS               Offs = 0003E7   Size = 000001
/Users/cas/develop/cc65/libsrc/atari.lib(strlen.o):
    CODE              Offs = 003E29   Size = 000016
/Users/cas/develop/cc65/libsrc/atari.lib(strlower.o):
    CODE              Offs = 003E3F   Size = 000028
/Users/cas/develop/cc65/libsrc/atari.lib(strupper.o):
    CODE              Offs = 003E67   Size = 000028
/Users/cas/develop/cc65/libsrc/atari.lib(vsprintf.o):
    CODE              Offs = 003E8F   Size = 000076
    DATA              Offs = 000171   Size = 000008
/Users/cas/develop/cc65/libsrc/atari.lib(zerobss.o):
    CODE              Offs = 003F05   Size = 000023
/Users/cas/develop/cc65/libsrc/atari.lib(add.o):
    CODE              Offs = 003F28   Size = 00001A
/Users/cas/develop/cc65/libsrc/atari.lib(addeqsp.o):
    CODE              Offs = 003F42   Size = 000011
/Users/cas/develop/cc65/libsrc/atari.lib(addysp.o):
    CODE              Offs = 003F53   Size = 00000E
/Users/cas/develop/cc65/libsrc/atari.lib(and.o):
    CODE              Offs = 003F61   Size = 000010
/Users/cas/develop/cc65/libsrc/atari.lib(aslax3.o):
    CODE              Offs = 003F71   Size = 00000E
/Users/cas/develop/cc65/libsrc/atari.lib(axlong.o):
    CODE              Offs = 003F7F   Size = 000012
/Users/cas/develop/cc65/libsrc/atari.lib(bneg.o):
    CODE              Offs = 003F91   Size = 00000E
/Users/cas/develop/cc65/libsrc/atari.lib(callmain.o):
    CODE              Offs = 003F9F   Size = 000017
    BSS               Offs = 0003E8   Size = 000004
/Users/cas/develop/cc65/libsrc/atari.lib(condes.o):
    CODE              Offs = 003FB6   Size = 000038
    BSS               Offs = 0003EC   Size = 000001
    DATA              Offs = 000179   Size = 000007
/Users/cas/develop/cc65/libsrc/atari.lib(decax1.o):
```

```
    CODE              Offs = 003FEE   Size = 000007
/Users/cas/develop/cc65/libsrc/atari.lib(decsp1.o):
    CODE              Offs = 003FF5   Size = 000009
/Users/cas/develop/cc65/libsrc/atari.lib(decsp2.o):
    CODE              Offs = 003FFE   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(decsp3.o):
    CODE              Offs = 00400B   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(decsp4.o):
    CODE              Offs = 004018   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(decsp5.o):
    CODE              Offs = 004025   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(decsp6.o):
    CODE              Offs = 004032   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(decsp8.o):
    CODE              Offs = 00403F   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(div.o):
    CODE              Offs = 00404C   Size = 000018
/Users/cas/develop/cc65/libsrc/atari.lib(enter.o):
    CODE              Offs = 004064   Size = 00000E
/Users/cas/develop/cc65/libsrc/atari.lib(ge.o):
    CODE              Offs = 004072   Size = 00000A
/Users/cas/develop/cc65/libsrc/atari.lib(icmp.o):
    CODE              Offs = 00407C   Size = 00002C
/Users/cas/develop/cc65/libsrc/atari.lib(incax1.o):
    CODE              Offs = 0040A8   Size = 000007
/Users/cas/develop/cc65/libsrc/atari.lib(incax2.o):
    CODE              Offs = 0040AF   Size = 000007
/Users/cas/develop/cc65/libsrc/atari.lib(incax3.o):
    CODE              Offs = 0040B6   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incax5.o):
    CODE              Offs = 0040BB   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incax6.o):
    CODE              Offs = 0040C0   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incax7.o):
    CODE              Offs = 0040C5   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incaxy.o):
    CODE              Offs = 0040CA   Size = 00000B
/Users/cas/develop/cc65/libsrc/atari.lib(incsp1.o):
    CODE              Offs = 0040D5   Size = 000007
/Users/cas/develop/cc65/libsrc/atari.lib(incsp2.o):
    CODE              Offs = 0040DC   Size = 000016
/Users/cas/develop/cc65/libsrc/atari.lib(incsp3.o):
    CODE              Offs = 0040F2   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incsp4.o):
    CODE              Offs = 0040F7   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incsp5.o):
    CODE              Offs = 0040FC   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incsp6.o):
    CODE              Offs = 004101   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incsp7.o):
    CODE              Offs = 004106   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(incsp8.o):
    CODE              Offs = 00410B   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(ldaxi.o):
    CODE              Offs = 004110   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(ldaxsp.o):
    CODE              Offs = 00411D   Size = 000009
/Users/cas/develop/cc65/libsrc/atari.lib(leasp.o):
    CODE              Offs = 004126   Size = 000009
```

```
/Users/cas/develop/cc65/libsrc/atari.lib(leave.o):
    CODE              Offs = 00412F   Size = 00001D
/Users/cas/develop/cc65/libsrc/atari.lib(lpush.o):
    CODE              Offs = 00414C   Size = 00001E
/Users/cas/develop/cc65/libsrc/atari.lib(lsub.o):
    CODE              Offs = 00416A   Size = 000021
/Users/cas/develop/cc65/libsrc/atari.lib(lt.o):
    CODE              Offs = 00418B   Size = 00000A
/Users/cas/develop/cc65/libsrc/atari.lib(ludiv.o):
    CODE              Offs = 004195   Size = 000074
/Users/cas/develop/cc65/libsrc/atari.lib(makebool.o):
    CODE              Offs = 004209   Size = 000031
/Users/cas/develop/cc65/libsrc/atari.lib(mod.o):
    CODE              Offs = 00423A   Size = 000014
/Users/cas/develop/cc65/libsrc/atari.lib(mul.o):
    CODE              Offs = 00424E   Size = 00002C
/Users/cas/develop/cc65/libsrc/atari.lib(mulax5.o):
    CODE              Offs = 00427A   Size = 000014
/Users/cas/develop/cc65/libsrc/atari.lib(mulax7.o):
    CODE              Offs = 00428E   Size = 000019
/Users/cas/develop/cc65/libsrc/atari.lib(neg.o):
    CODE              Offs = 0042A7   Size = 00000E
/Users/cas/develop/cc65/libsrc/atari.lib(popsreg.o):
    CODE              Offs = 0042B5   Size = 000010
/Users/cas/develop/cc65/libsrc/atari.lib(push1.o):
    CODE              Offs = 0042C5   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(push2.o):
    CODE              Offs = 0042CA   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(push3.o):
    CODE              Offs = 0042CF   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(push6.o):
    CODE              Offs = 0042D4   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(push7.o):
    CODE              Offs = 0042D9   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(pusha.o):
    CODE              Offs = 0042DE   Size = 000016
/Users/cas/develop/cc65/libsrc/atari.lib(pushaff.o):
    CODE              Offs = 0042F4   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(pushax.o):
    CODE              Offs = 0042F9   Size = 00001A
/Users/cas/develop/cc65/libsrc/atari.lib(pushc0.o):
    CODE              Offs = 004313   Size = 000005
/Users/cas/develop/cc65/libsrc/atari.lib(pushw.o):
    CODE              Offs = 004318   Size = 00000F
/Users/cas/develop/cc65/libsrc/atari.lib(pushwsp.o):
    CODE              Offs = 004327   Size = 00001C
/Users/cas/develop/cc65/libsrc/atari.lib(return0.o):
    CODE              Offs = 004343   Size = 000004
/Users/cas/develop/cc65/libsrc/atari.lib(shelp.o):
    CODE              Offs = 004347   Size = 00001E
/Users/cas/develop/cc65/libsrc/atari.lib(shl.o):
    CODE              Offs = 004365   Size = 000036
/Users/cas/develop/cc65/libsrc/atari.lib(shrax1.o):
    CODE              Offs = 00439B   Size = 000008
/Users/cas/develop/cc65/libsrc/atari.lib(staspidx.o):
    CODE              Offs = 0043A3   Size = 000016
/Users/cas/develop/cc65/libsrc/atari.lib(staxsp.o):
    CODE              Offs = 0043B9   Size = 00000B
/Users/cas/develop/cc65/libsrc/atari.lib(staxspi.o):
```

```
    CODE              Offs = 0043C4   Size = 00001B
/Users/cas/develop/cc65/libsrc/atari.lib(sub.o):
    CODE              Offs = 0043DF   Size = 000015
/Users/cas/develop/cc65/libsrc/atari.lib(subeqsp.o):
    CODE              Offs = 0043F4   Size = 000015
/Users/cas/develop/cc65/libsrc/atari.lib(subysp.o):
    CODE              Offs = 004409   Size = 00000D
/Users/cas/develop/cc65/libsrc/atari.lib(toslong.o):
    CODE              Offs = 004416   Size = 000038
/Users/cas/develop/cc65/libsrc/atari.lib(udiv.o):
    CODE              Offs = 00444E   Size = 000036
/Users/cas/develop/cc65/libsrc/atari.lib(umod.o):
    CODE              Offs = 004484   Size = 000011
/Users/cas/develop/cc65/libsrc/atari.lib(zeropage.o):
    ZEROPAGE          Offs = 000000   Size = 00001A
/Users/cas/develop/cc65/libsrc/atari.lib(_cursor.o):
    BSS               Offs = 0003ED   Size = 000001


Segment list:
-------------
Name                   Start   End     Size
---------------------------------------------
EXEHDR                 000000  000005  000006
ZEROPAGE               000082  00009B  00001A
CODE                   002E00  007294  004495
RODATA                 007295  0078D6  000642
DATA                   0078D7  007A56  000180
BSS                    007A57  007E44  0003EE
AUTOSTRT               007E45  007E4A  000006


Exports list:
-------------
__BSS_LOAD__               007A57 RLA    __BSS_RUN__                007A57 RLA
__BSS_SIZE__               0003EE REA    __CODE_LOAD__              002E00 RLA
__CONSTRUCTOR_COUNT__      000002 REA    __CONSTRUCTOR_TABLE__      0078D1 RLA
__DESTRUCTOR_COUNT__       000001 REA    __DESTRUCTOR_TABLE__       0078D5 RLA
__clocks_per_sec           006046 RLA    __ctype                    007772 RLA
__do_oserror               006419 RLA    __errno                    007E38 RLA
__fdesc                    0064BA RLA    __filetab                  007A2B RLA
__fopen                    0064D6 RLA    __graphmode_used           007A2A RLA
__hextab                   0078B5 RLA    __inviocb                  006427 RLA
__oserror                  007E0C RLA    __printf                   006649 RLA
__rwsetup                  0063DB RLA    __seterrno                 006420 RLA
_cgetc                     005FE2 RLA    _clock                     006033 RLA
_close                     006056 RLA    _dli01                     005E44 RLA
_exit                      002E5B RLA    _fclose                    0068F7 RLA
_fnt14                     005743 RLA    _fnt7                      0053C3 RLA
_fopen                     00691D RLA    _fread                     00693A RLA
_fwrite                    0069CC RLA    _graphics                  006257 RLA
_ltoa                      006A6A RLA    _main                      0051C8 RLA
_memcpy                    006AFD RLA    _memmove                   006B37 RLA
_memset                    006B7B RLA    _menuflg                   005E43 RLA
_open                      0062F8 RLA    _rand                      006BB2 RLA
_read                      0063AF RLA    _sprintf                   006BFE RLA
_srand                     006BEF RLA    _strlen                    006C29 RLA
_strlower                  006C3F RLA    _strupr                    006C67 RLA
_ultoa                     006AC1 RLA    _vsprintf                  006CC6 RLA
```

_write	006492 RLA	addeq0sp	006D42 RLA
addeqysp	006D44 RLA	addysp	006D54 RLA
addysp1	006D53 RLA	aslax3	006D71 RLA
axlong	006D86 RLA	axulong	006D7F RLA
bnegax	006D91 RLA	boolge	00701F RLA
boollt	007017 RLA	booluge	00702F RLA
callmain	006D9F RLA	closeallfiles	0063A0 LAI
clriocb	0060BB RLA	cursor	007E44 RLA
decax1	006DEE RLA	decsp1	006DF5 RLA
decsp2	006DFE RLA	decsp3	006E0B RLA
decsp4	006E18 RLA	decsp5	006E25 RLA
decsp6	006E32 RLA	decsp8	006E3F RLA
donelib	006DBF RLA	enter	006E64 RLA
fd_index	0079FE RLA	fd_table	007A0A RLA
fddecusage	0060FC RLA	fdt_to_fdi	00622E RLA
fdtoiocb	0060C8 RLA	fdtoiocb_down	006074 RLA
findfreeiocb	0060E7 RLA	getfd	006245 RLA
incax1	006EA8 RLA	incax2	006EAF RLA
incax3	006EB6 RLA	incax4	006ECA RLA
incax5	006EBB RLA	incax6	006EC0 RLA
incax7	006EC5 RLA	incaxy	006ECC RLA
incsp1	006ED5 RLA	incsp2	006EE4 RLA
incsp3	006EF2 RLA	incsp4	006EF7 RLA
incsp5	006EFC RLA	incsp6	006F01 RLA
incsp7	006F06 RLA	incsp8	006F0B RLA
initlib	006DB6 RLA	initscrmem	0062BD LAI
initsp	002E86 LAI	ldax0sp	006F1D RLA
ldaxi	006F10 RLA	ldaxysp	006F1F RLA
leaasp	006F26 RLA	leave	006F3C RLA
leavey	006F39 RLA	memcpy_getparams	006B1D RLA
memcpy_upwards	006B00 RLA	mul40	0062D8 RLA
mulax5	00707A RLA	mulax7	00708E RLA
negax	0070A7 RLA	newfd	00612A RLA
popax	006EDC RLA	popsargs	007147 RLA
popsreg	0070B5 RLA	ptr1	00008A RLZ
ptr2	00008C RLZ	ptr3	00008E RLZ
ptr4	000090 RLZ	push0	0070F9 RLA
push1	0070C5 RLA	push2	0070CA RLA
push3	0070CF RLA	push6	0070D4 RLA
push7	0070D9 RLA	pusha	0070E2 RLA
pusha0	0070FB RLA	pushaFF	0070F4 RLA
pushax	0070FD RLA	pushc0	007113 RLA
pusheax	006F52 RLA	pushw	007118 RLA
pushw0sp	007127 RLA	pushwysp	007129 RLA
regbank	000096 RLZ	regsave	000086 RLZ
return0	007143 RLA	shlax3	006D71 RLA
shrax1	00719B RLA	sp	000082 RLZ
sreg	000084 RLZ	staspidx	0071A3 RLA
stax0sp	0071B9 RLA	staxspidx	0071C4 RLA
staxysp	0071BB RLA	subeqysp	0071F6 RLA
subysp	007209 RLA	tmp1	000092 RLZ
tmp2	000093 RLZ	tmp3	000094 RLZ
tmp4	000095 RLZ	tosadda0	006D28 RLA
tosaddax	006D2A RLA	tosandax	006D63 RLA
tosaslax	007167 RLA	tosdiva0	006E4C RLA
tosgea0	006E74 RLA	tosicmp	006E7C RLA
toslong	007234 RLA	toslta0	006F8D RLA
tosmoda0	00703A RLA	tosmula0	00704E RLA
tosmulax	007050 RLA	tossuba0	0071DF RLA

```
tossubax                    0071E1 RLA    tossubeax                   006F6A RLA
tosudivax                   007250 RLA    tosudiveax                  006F95 RLA
tosumoda0                   007284 RLA    tosumula0                   00704E RLA
tosumulax                   007050 RLA    ucase_fn                    006431 RLA
udiv16                      00725F RLA    zerobss                     006D05 RLA


Imports list:
-------------
__BSS_LOAD__ ([linker generated]):
    atari.o                   crt0.s(18)
__BSS_RUN__ ([linker generated]):
    zerobss.o                 zerobss.s(8)
__BSS_SIZE__ ([linker generated]):
    zerobss.o                 zerobss.s(8)
__CODE_LOAD__ ([linker generated]):
    atari.o                   crt0.s(18)
__CONSTRUCTOR_COUNT__ ([linker generated]):
    condes.o                  condes.s(18)
__CONSTRUCTOR_TABLE__ ([linker generated]):
    condes.o                  condes.s(18)
__DESTRUCTOR_COUNT__ ([linker generated]):
    condes.o                  condes.s(19)
__DESTRUCTOR_TABLE__ ([linker generated]):
    condes.o                  condes.s(19)
__clocks_per_sec (clock.o):
    agreed.o                  agreed.s(26)
__ctype (ctype.o):
    strupper.o                strupper.s(12)
    strlower.o                strlower.s(12)
__do_oserror (rwcommon.o):
    open.o                    open.s(19)
    write.o                   write.s(5)
    read.o                    read.s(8)
    close.o                   close.s(9)
    graphics.o                graphics.s(14)
__errno (errno.o):
    _fopen.o                  errno.inc(7)
    fwrite.o                  errno.inc(7)
    fread.o                   errno.inc(7)
    fopen.o                   errno.inc(7)
    fclose.o                  errno.inc(7)
    rwcommon.o                errno.inc(7)
__fdesc (_fdesc.o):
    fopen.o                   fopen.s(10)
__filetab (_file.o):
    _fdesc.o                  _file.inc(24)
    atari.o                   crt0.s(17)
__fopen (_fopen.o):
    fopen.o                   fopen.s(10)
__graphmode_used (graphuse.o):
    graphics.o                graphics.s(19)
__hextab (_hextab.o):
    ltoa.o                    ltoa.s(10)
__inviocb (rwcommon.o):
    write.o                   write.s(5)
    read.o                    read.s(8)
    close.o                   close.s(10)
```

```
__oserror (_oserror.o):
    open.o                  errno.inc(7)
    write.o                 write.s(5)
    read.o                  read.s(8)
    close.o                 close.s(9)
    rwcommon.o              errno.inc(7)
    graphics.o              graphics.s(14)
__printf (_printf.o):
    vsprintf.o              vsprintf.s(9)
__rwsetup (rwcommon.o):
    write.o                 write.s(5)
    read.o                  read.s(8)
__seterrno (rwcommon.o):
    open.o                  open.s(19)
    graphics.o              graphics.s(14)
_cgetc (cgetc.o):
    agreed.o                agreed.s(13)
_clock (clock.o):
    agreed.o                agreed.s(27)
_close (close.o):
    open.o                  open.s(15)
    fclose.o                fclose.s(10)
_dli01 (dli1.o):
    agreed.o                agreed.s(28)
_exit (atari.o):
    agreed.o                agreed.s(16)
_fclose (fclose.o):
    agreed.o                agreed.s(17)
_fnt14 (dli1.o):
    agreed.o                agreed.s(31)
_fnt7 (dli1.o):
    agreed.o                agreed.s(30)
_fopen (fopen.o):
    agreed.o                agreed.s(18)
_fread (fread.o):
    agreed.o                agreed.s(19)
_fwrite (fwrite.o):
    agreed.o                agreed.s(20)
_graphics (graphics.o):
    agreed.o                agreed.s(12)
_ltoa (ltoa.o):
    _printf.o               _printf.s(11)
_main (agreed.o):
    callmain.o              callmain.s(11)
_memcpy (memcpy.o):
    vsprintf.o              vsprintf.s(9)
    agreed.o                agreed.s(23)
_memmove (memmove.o):
    agreed.o                agreed.s(24)
_memset (memset.o):
    agreed.o                agreed.s(25)
_menuflg (dli1.o):
    agreed.o                agreed.s(29)
_open (open.o):
    _fopen.o                _fopen.s(10)
_rand (rand.o):
    agreed.o                agreed.s(14)
_read (read.o):
    fread.o                 fread.s(10)
```

```
_sprintf (sprintf.o):
    agreed.o                agreed.s(21)
_srand (rand.o):
    agreed.o                agreed.s(15)
_strlen (strlen.o):
    _printf.o               _printf.s(12)
    agreed.o                agreed.s(22)
_strlower (strlower.o):
    _printf.o               _printf.s(12)
_strupr (strupper.o):
    ucase_fn.o              ucase_fn.s(25)
_ultoa (ltoa.o):
    _printf.o               _printf.s(11)
_vsprintf (vsprintf.o):
    sprintf.o               sprintf.s(8)
_write (write.o):
    fwrite.o                fwrite.s(10)
addeq0sp (addeqsp.o):
    agreed.o                agreed.s(768)
    agreed.o                agreed.s(904)
    agreed.o                agreed.s(1082)
    agreed.o                agreed.s(1280)
    agreed.o                agreed.s(3514)
    agreed.o                agreed.s(4211)
    agreed.o                agreed.s(4778)
    agreed.o                agreed.s(6132)
addeqysp (addeqsp.o):
    agreed.o                agreed.s(891)
    agreed.o                agreed.s(1069)
    agreed.o                agreed.s(1380)
    agreed.o                agreed.s(1627)
    agreed.o                agreed.s(1914)
    agreed.o                agreed.s(2158)
    agreed.o                agreed.s(2583)
    agreed.o                agreed.s(3988)
    agreed.o                agreed.s(4442)
    agreed.o                agreed.s(4688)
    agreed.o                agreed.s(5650)
    agreed.o                agreed.s(6126)
addysp (addysp.o):
    open.o                  open.s(20)
    leave.o                 leave.s(13)
    incsp8.o                incsp8.s(8)
    incsp7.o                incsp7.s(8)
    incsp6.o                incsp6.s(8)
    incsp5.o                incsp5.s(8)
    incsp4.o                incsp4.s(8)
    incsp3.o                incsp3.s(8)
    sprintf.o               sprintf.s(8)
    fwrite.o                fwrite.s(11)
    fread.o                 fread.s(11)
    agreed.o                agreed.s(1417)
    agreed.o                agreed.s(1806)
    agreed.o                agreed.s(2900)
    agreed.o                agreed.s(3839)
addysp1 (addysp.o):
    sub.o                   sub.s(8)
    ludiv.o                 ludiv.s(8)
    lsub.o                  lsub.s(11)
```

```
    and.o                   and.s(8)
aslax3 (aslax3.o):
    agreed.o                agreed.s(1668)
    agreed.o                agreed.s(2452)
axlong (axlong.o):
    _printf.o               _printf.s(9)
axulong (axlong.o):
    _printf.o               _printf.s(9)
    agreed.o                agreed.s(3045)
bnegax (bneg.o):
    agreed.o                agreed.s(5450)
    agreed.o                agreed.s(6372)
boolge (makebool.o):
    ge.o                    ge.s(8)
boollt (makebool.o):
    lt.o                    lt.s(8)
booluge (makebool.o):
    agreed.o                agreed.s(1325)
    agreed.o                agreed.s(1652)
callmain (callmain.o):
    atari.o                 crt0.s(15)
clriocb (fdtable.o):
    open.o                  open.s(16)
    graphics.o              graphics.s(16)
cursor (_cursor.o):
    cgetc.o                 cgetc.s(10)
decax1 (decax1.o):
    agreed.o                agreed.s(3572)
    agreed.o                agreed.s(6362)
    agreed.o                agreed.s(6699)
decsp1 (decsp1.o):
    agreed.o                agreed.s(642)
    agreed.o                agreed.s(2268)
    agreed.o                agreed.s(3139)
    agreed.o                agreed.s(3326)
decsp2 (decsp2.o):
    toslong.o               toslong.s(8)
    agreed.o                agreed.s(450)
    agreed.o                agreed.s(1146)
    agreed.o                agreed.s(1823)
    agreed.o                agreed.s(1932)
    agreed.o                agreed.s(1967)
    agreed.o                agreed.s(3605)
    agreed.o                agreed.s(3866)
    agreed.o                agreed.s(4506)
    agreed.o                agreed.s(6490)
decsp3 (decsp3.o):
    agreed.o                agreed.s(705)
    agreed.o                agreed.s(1882)
    agreed.o                agreed.s(2045)
    agreed.o                agreed.s(3661)
    agreed.o                agreed.s(6393)
decsp4 (decsp4.o):
    lpush.o                 lpush.s(11)
    sprintf.o               sprintf.s(8)
    agreed.o                agreed.s(802)
    agreed.o                agreed.s(951)
    agreed.o                agreed.s(2921)
decsp5 (decsp5.o):
```

```
    agreed.o                agreed.s(2643)
decsp6 (decsp6.o):
    _printf.o               _printf.s(9)
    agreed.o                agreed.s(1319)
    agreed.o                agreed.s(3038)
    agreed.o                agreed.s(4823)
    agreed.o                agreed.s(6066)
decsp8 (decsp8.o):
    agreed.o                agreed.s(1623)
donelib (condes.o):
    atari.o                 crt0.s(15)
enter (enter.o):
    agreed.o                agreed.s(701)
    agreed.o                agreed.s(798)
    agreed.o                agreed.s(947)
    agreed.o                agreed.s(1142)
    agreed.o                agreed.s(2245)
    agreed.o                agreed.s(2639)
    agreed.o                agreed.s(2917)
    agreed.o                agreed.s(3565)
    agreed.o                agreed.s(3601)
    agreed.o                agreed.s(3856)
    agreed.o                agreed.s(4287)
    agreed.o                agreed.s(4496)
    agreed.o                agreed.s(5127)
    agreed.o                agreed.s(6355)
fd_index (getfd.o):
    fdtable.o               fdtable.s(10)
fd_table (getfd.o):
    fdtable.o               fdtable.s(10)
fddecusage (fdtable.o):
    open.o                  open.s(17)
    graphics.o              graphics.s(15)
fdt_to_fdi (getfd.o):
    fdtable.o               fdtable.s(11)
fdtoiocb (fdtable.o):
    rwcommon.o              rwcommon.s(10)
    graphics.o              graphics.s(17)
fdtoiocb_down (fdtable.o):
    close.o                 close.s(10)
findfreeiocb (fdtable.o):
    open.o                  open.s(18)
    graphics.o              graphics.s(13)
getfd (getfd.o):
    atari.o                 crt0.s(17)
incax1 (incax1.o):
    agreed.o                agreed.s(744)
    agreed.o                agreed.s(1180)
    agreed.o                agreed.s(1386)
    agreed.o                agreed.s(1532)
    agreed.o                agreed.s(1584)
    agreed.o                agreed.s(1748)
    agreed.o                agreed.s(1843)
    agreed.o                agreed.s(2510)
    agreed.o                agreed.s(3695)
    agreed.o                agreed.s(6097)
    agreed.o                agreed.s(6368)
    agreed.o                agreed.s(6408)
    agreed.o                agreed.s(6698)
```

```
incax2 (incax2.o):
    agreed.o                agreed.s(865)
    agreed.o                agreed.s(1014)
    agreed.o                agreed.s(1207)
incax3 (incax3.o):
    agreed.o                agreed.s(847)
    agreed.o                agreed.s(996)
    agreed.o                agreed.s(4550)
incax4 (incaxy.o):
    agreed.o                agreed.s(1167)
incax5 (incax5.o):
    agreed.o                agreed.s(726)
    agreed.o                agreed.s(1155)
    agreed.o                agreed.s(4542)
incax6 (incax6.o):
    agreed.o                agreed.s(714)
    agreed.o                agreed.s(823)
    agreed.o                agreed.s(972)
incax7 (incax7.o):
    agreed.o                agreed.s(811)
    agreed.o                agreed.s(960)
incaxy (incaxy.o):
    incax7.o                incax7.s(8)
    incax6.o                incax6.s(8)
    incax5.o                incax5.s(8)
    incax3.o                incax3.s(8)
    agreed.o                agreed.s(456)
    agreed.o                agreed.s(2393)
    agreed.o                agreed.s(2724)
    agreed.o                agreed.s(4174)
    agreed.o                agreed.s(4433)
    agreed.o                agreed.s(5164)
incsp1 (incsp1.o):
    agreed.o                agreed.s(433)
    agreed.o                agreed.s(2330)
    agreed.o                agreed.s(3309)
    agreed.o                agreed.s(3486)
incsp2 (incsp2.o):
    staxspi.o               staxspi.s(8)
    staspidx.o              staspidx.s(8)
    popsreg.o               popsreg.s(8)
    agreed.o                agreed.s(684)
    agreed.o                agreed.s(2028)
    agreed.o                agreed.s(3548)
    agreed.o                agreed.s(6338)
incsp3 (incsp3.o):
    agreed.o                agreed.s(475)
    agreed.o                agreed.s(3718)
incsp4 (incsp4.o):
    open.o                  open.s(19)
    _fopen.o                _fopen.s(11)
    rwcommon.o              rwcommon.s(8)
    agreed.o                agreed.s(6806)
incsp5 (incsp5.o):
    agreed.o                agreed.s(1459)
    agreed.o                agreed.s(1502)
    agreed.o                agreed.s(1892)
    agreed.o                agreed.s(6465)
incsp6 (incsp6.o):
```

```
    fwrite.o                fwrite.s(11)
    fread.o                 fread.s(11)
    agreed.o                agreed.s(1524)
    agreed.o                agreed.s(1576)
    agreed.o                agreed.s(3058)
    agreed.o                agreed.s(4201)
    agreed.o                agreed.s(4767)
    agreed.o                agreed.s(6174)
incsp7 (incsp7.o):
    agreed.o                agreed.s(2228)
incsp8 (incsp8.o):
    agreed.o                agreed.s(1835)
    agreed.o                agreed.s(1948)
    agreed.o                agreed.s(1983)
    agreed.o                agreed.s(5110)
initlib (condes.o):
    atari.o                 crt0.s(15)
ldax0sp (ldaxsp.o):
    rwcommon.o              rwcommon.s(8)
    agreed.o                agreed.s(466)
    agreed.o                agreed.s(1236)
    agreed.o                agreed.s(1519)
    agreed.o                agreed.s(1571)
    agreed.o                agreed.s(1830)
    agreed.o                agreed.s(1949)
    agreed.o                agreed.s(1984)
    agreed.o                agreed.s(3507)
    agreed.o                agreed.s(3709)
    agreed.o                agreed.s(3910)
    agreed.o                agreed.s(4519)
ldaxi (ldaxi.o):
    agreed.o                agreed.s(745)
    agreed.o                agreed.s(866)
    agreed.o                agreed.s(1015)
    agreed.o                agreed.s(1198)
    agreed.o                agreed.s(2394)
    agreed.o                agreed.s(4183)
    agreed.o                agreed.s(4543)
    agreed.o                agreed.s(6369)
ldaxysp (ldaxsp.o):
    open.o                  open.s(20)
    fwrite.o                fwrite.s(11)
    fread.o                 fread.s(11)
    rwcommon.o              rwcommon.s(8)
    agreed.o                agreed.s(1342)
    agreed.o                agreed.s(1529)
    agreed.o                agreed.s(1581)
    agreed.o                agreed.s(1667)
    agreed.o                agreed.s(1825)
    agreed.o                agreed.s(1890)
    agreed.o                agreed.s(1939)
    agreed.o                agreed.s(1974)
    agreed.o                agreed.s(2015)
    agreed.o                agreed.s(2047)
    agreed.o                agreed.s(2376)
    agreed.o                agreed.s(2677)
    agreed.o                agreed.s(2851)
    agreed.o                agreed.s(2975)
    agreed.o                agreed.s(3095)
```

```
    agreed.o                  agreed.s(3868)
    agreed.o                  agreed.s(4348)
    agreed.o                  agreed.s(4508)
    agreed.o                  agreed.s(5518)
    agreed.o                  agreed.s(6402)
leaasp (leasp.o):
    agreed.o                  agreed.s(713)
    agreed.o                  agreed.s(810)
    agreed.o                  agreed.s(959)
    agreed.o                  agreed.s(1154)
    agreed.o                  agreed.s(2391)
    agreed.o                  agreed.s(2805)
    agreed.o                  agreed.s(3082)
    agreed.o                  agreed.s(3571)
    agreed.o                  agreed.s(3738)
    agreed.o                  agreed.s(4180)
    agreed.o                  agreed.s(4333)
    agreed.o                  agreed.s(4541)
    agreed.o                  agreed.s(4900)
    agreed.o                  agreed.s(5163)
    agreed.o                  agreed.s(6361)
leave (leave.o):
    agreed.o                  agreed.s(3584)
    agreed.o                  agreed.s(6376)
leavey (leave.o):
    agreed.o                  agreed.s(781)
    agreed.o                  agreed.s(930)
    agreed.o                  agreed.s(1125)
    agreed.o                  agreed.s(1302)
    agreed.o                  agreed.s(2622)
    agreed.o                  agreed.s(2760)
    agreed.o                  agreed.s(2998)
    agreed.o                  agreed.s(3644)
    agreed.o                  agreed.s(4225)
    agreed.o                  agreed.s(4479)
    agreed.o                  agreed.s(4797)
    agreed.o                  agreed.s(6049)
memcpy_getparams (memcpy.o):
    memmove.o                 memmove.s(10)
memcpy_upwards (memcpy.o):
    memmove.o                 memmove.s(10)
mul40 (mul40.o):
    cgetc.o                   cgetc.s(10)
mulax5 (mulax5.o):
    agreed.o                  agreed.s(6426)
mulax7 (mulax7.o):
    agreed.o                  agreed.s(1444)
negax (neg.o):
    shelp.o                   shelp.s(11)
    mod.o                     mod.s(11)
    div.o                     div.s(11)
newfd (fdtable.o):
    open.o                    open.s(17)
    graphics.o                graphics.s(18)
popax (incsp2.o):
    ltoa.o                    ltoa.s(9)
    _printf.o                 _printf.s(9)
    shelp.o                   shelp.s(11)
    vsprintf.o                vsprintf.s(8)
```

```
    memset.o                memset.s(17)
    memcpy.o                memcpy.s(12)
popsargs (shelp.o):
    mod.o                   mod.s(11)
    div.o                   div.s(11)
popsreg (popsreg.o):
    umod.o                  umod.s(8)
    udiv.o                  udiv.s(8)
    shl.o                   shl.s(8)
    mul.o                   mul.s(8)
ptr1 (zeropage.o):
    strupper.o              strupper.s(13)
    strlower.o              strlower.s(13)
    ltoa.o                  ltoa.s(11)
    _printf.o               _printf.s(13)
    _fopen.o                _fopen.s(12)
    umod.o                  umod.s(9)
    udiv.o                  udiv.s(9)
    staxspi.o               staxspi.s(9)
    staspidx.o              staspidx.s(9)
    pushw.o                 pushw.s(9)
    mulax7.o                mulax7.s(9)
    mulax5.o                mulax5.s(8)
    mod.o                   mod.s(12)
    ludiv.o                 ludiv.s(9)
    ldaxi.o                 ldaxi.s(8)
    zerobss.o               zerobss.s(9)
    vsprintf.o              vsprintf.s(10)
    strlen.o                strlen.s(8)
    sprintf.o               sprintf.s(9)
    memset.o                memset.s(18)
    memmove.o               memmove.s(11)
    memcpy.o                memcpy.s(13)
    fwrite.o                fwrite.s(14)
    fread.o                 fread.s(14)
    fclose.o                fclose.s(11)
    agreed.o                agreed.s(10)
ptr2 (zeropage.o):
    strupper.o              strupper.s(13)
    strlower.o              strlower.s(13)
    ltoa.o                  ltoa.s(11)
    _printf.o               _printf.s(13)
    ludiv.o                 ludiv.s(9)
    memmove.o               memmove.s(11)
    memcpy.o                memcpy.s(13)
    agreed.o                agreed.s(10)
ptr3 (zeropage.o):
    ltoa.o                  ltoa.s(11)
    ludiv.o                 ludiv.s(9)
    memset.o                memset.s(18)
    memmove.o               memmove.s(11)
    memcpy.o                memcpy.s(13)
ptr4 (zeropage.o):
    ucase_fn.o              ucase_fn.s(24)
    fdtable.o               fdtable.s(9)
    umod.o                  umod.s(9)
    udiv.o                  udiv.s(9)
    shelp.o                 shelp.s(12)
    mul.o                   mul.s(9)
```

```
    ludiv.o                 ludiv.s(9)
    memmove.o               memmove.s(11)
push0 (pushax.o):
    agreed.o                agreed.s(752)
    agreed.o                agreed.s(2002)
    agreed.o                agreed.s(2061)
    agreed.o                agreed.s(3503)
    agreed.o                agreed.s(3609)
    agreed.o                agreed.s(4242)
    agreed.o                agreed.s(4330)
    agreed.o                agreed.s(5878)
    agreed.o                agreed.s(6070)
push1 (push1.o):
    _printf.o               _printf.s(9)
    agreed.o                agreed.s(1935)
    agreed.o                agreed.s(1970)
    agreed.o                agreed.s(3080)
    agreed.o                agreed.s(4331)
    agreed.o                agreed.s(4875)
    agreed.o                agreed.s(5856)
    agreed.o                agreed.s(6364)
    agreed.o                agreed.s(6486)
push2 (push2.o):
    agreed.o                agreed.s(2799)
    agreed.o                agreed.s(2931)
    agreed.o                agreed.s(3519)
    agreed.o                agreed.s(4885)
push3 (push3.o):
    agreed.o                agreed.s(3166)
    agreed.o                agreed.s(3353)
push6 (push6.o):
    agreed.o                agreed.s(3751)
push7 (push7.o):
    agreed.o                agreed.s(6452)
pusha (pusha.o):
    pushc0.o                pushc0.s(8)
    agreed.o                agreed.s(453)
    agreed.o                agreed.s(1440)
    agreed.o                agreed.s(1482)
    agreed.o                agreed.s(1549)
    agreed.o                agreed.s(1601)
    agreed.o                agreed.s(1640)
    agreed.o                agreed.s(1860)
    agreed.o                agreed.s(2213)
    agreed.o                agreed.s(2335)
    agreed.o                agreed.s(2709)
    agreed.o                agreed.s(3141)
    agreed.o                agreed.s(3328)
    agreed.o                agreed.s(3531)
    agreed.o                agreed.s(4186)
    agreed.o                agreed.s(4302)
    agreed.o                agreed.s(4851)
    agreed.o                agreed.s(5745)
    agreed.o                agreed.s(6554)
pusha0 (pushax.o):
    push7.o                 push7.s(8)
    push6.o                 push6.s(8)
    push3.o                 push3.s(8)
    push2.o                 push2.s(8)
```

```
    push1.o                 push1.s(8)
    agreed.o                agreed.s(420)
    agreed.o                agreed.s(710)
    agreed.o                agreed.s(807)
    agreed.o                agreed.s(956)
    agreed.o                agreed.s(1151)
    agreed.o                agreed.s(1321)
    agreed.o                agreed.s(1485)
    agreed.o                agreed.s(1648)
    agreed.o                agreed.s(1904)
    agreed.o                agreed.s(2004)
    agreed.o                agreed.s(2075)
    agreed.o                agreed.s(2252)
    agreed.o                agreed.s(2669)
    agreed.o                agreed.s(2812)
    agreed.o                agreed.s(2968)
    agreed.o                agreed.s(3079)
    agreed.o                agreed.s(3147)
    agreed.o                agreed.s(3334)
    agreed.o                agreed.s(3753)
    agreed.o                agreed.s(3973)
    agreed.o                agreed.s(4244)
    agreed.o                agreed.s(4306)
    agreed.o                agreed.s(4673)
    agreed.o                agreed.s(4855)
    agreed.o                agreed.s(6156)
    agreed.o                agreed.s(6580)
pushaFF (pushaff.o):
    agreed.o                agreed.s(3862)
    agreed.o                agreed.s(4502)
pushax (pushax.o):
    _printf.o               _printf.s(9)
    pushw.o                 pushw.s(8)
    pushaff.o               pushaff.s(8)
    callmain.o              callmain.s(11)
    fwrite.o                fwrite.s(11)
    fread.o                 fread.s(11)
    fopen.o                 fopen.s(11)
    agreed.o                agreed.s(708)
    agreed.o                agreed.s(805)
    agreed.o                agreed.s(954)
    agreed.o                agreed.s(1149)
    agreed.o                agreed.s(1340)
    agreed.o                agreed.s(1452)
    agreed.o                agreed.s(1495)
    agreed.o                agreed.s(1537)
    agreed.o                agreed.s(1589)
    agreed.o                agreed.s(1676)
    agreed.o                agreed.s(1848)
    agreed.o                agreed.s(2007)
    agreed.o                agreed.s(2054)
    agreed.o                agreed.s(2255)
    agreed.o                agreed.s(2646)
    agreed.o                agreed.s(2806)
    agreed.o                agreed.s(2938)
    agreed.o                agreed.s(3041)
    agreed.o                agreed.s(3171)
    agreed.o                agreed.s(3358)
    agreed.o                agreed.s(3508)
```

```
    agreed.o                  agreed.s(3608)
    agreed.o                  agreed.s(3677)
    agreed.o                  agreed.s(3739)
    agreed.o                  agreed.s(3869)
    agreed.o                  agreed.s(4247)
    agreed.o                  agreed.s(4295)
    agreed.o                  agreed.s(4509)
    agreed.o                  agreed.s(4863)
    agreed.o                  agreed.s(5135)
    agreed.o                  agreed.s(6069)
    agreed.o                  agreed.s(6319)
    agreed.o                  agreed.s(6405)
    agreed.o                  agreed.s(6519)
pushc0 (pushc0.o):
    agreed.o                  agreed.s(2682)
    agreed.o                  agreed.s(4300)
    agreed.o                  agreed.s(4832)
    agreed.o                  agreed.s(6191)
pusheax (lpush.o):
    _printf.o                 _printf.s(9)
    agreed.o                  agreed.s(3043)
pushw (pushw.o):
    agreed.o                  agreed.s(3573)
    agreed.o                  agreed.s(6363)
    agreed.o                  agreed.s(6403)
pushw0sp (pushwsp.o):
    agreed.o                  agreed.s(751)
    agreed.o                  agreed.s(872)
    agreed.o                  agreed.s(3675)
    agreed.o                  agreed.s(4546)
    agreed.o                  agreed.s(6114)
pushwysp (pushwsp.o):
    fwrite.o                  fwrite.s(11)
    fread.o                   fread.s(11)
    agreed.o                  agreed.s(874)
    agreed.o                  agreed.s(1050)
    agreed.o                  agreed.s(1368)
    agreed.o                  agreed.s(1435)
    agreed.o                  agreed.s(1477)
    agreed.o                  agreed.s(1539)
    agreed.o                  agreed.s(1591)
    agreed.o                  agreed.s(1850)
    agreed.o                  agreed.s(1888)
    agreed.o                  agreed.s(1934)
    agreed.o                  agreed.s(1969)
    agreed.o                  agreed.s(2023)
    agreed.o                  agreed.s(2060)
    agreed.o                  agreed.s(2414)
    agreed.o                  agreed.s(2664)
    agreed.o                  agreed.s(2838)
    agreed.o                  agreed.s(2963)
    agreed.o                  agreed.s(3069)
    agreed.o                  agreed.s(4142)
    agreed.o                  agreed.s(4401)
    agreed.o                  agreed.s(4538)
    agreed.o                  agreed.s(5713)
    agreed.o                  agreed.s(6116)
regbank (zeropage.o):
    _printf.o                 _printf.s(13)
```

```
regsave (zeropage.o):
    agreed.o                agreed.s(10)
return0 (return0.o):
    _fopen.o                _fopen.s(11)
    _fdesc.o                _fdesc.s(9)
    shl.o                   shl.s(8)
    fwrite.o                fwrite.s(11)
    fread.o                 fread.s(11)
shlax3 (aslax3.o):
    agreed.o                agreed.s(1707)
shrax1 (shrax1.o):
    agreed.o                agreed.s(2017)
sp (zeropage.o):
    ucase_fn.o              ucase_fn.s(24)
    _printf.o               _printf.s(13)
    _fopen.o                _fopen.s(12)
    fdtable.o               fdtable.s(9)
    toslong.o               toslong.s(9)
    subysp.o                subysp.s(9)
    subeqsp.o               subeqsp.s(8)
    sub.o                   sub.s(9)
    staxspi.o               staxspi.s(9)
    staxsp.o                staxsp.s(8)
    staspidx.o              staspidx.s(9)
    pushwsp.o               pushwsp.s(8)
    pushax.o                pushax.s(8)
    pusha.o                 pusha.s(8)
    popsreg.o               popsreg.s(9)
    ludiv.o                 ludiv.s(9)
    lsub.o                  lsub.s(12)
    lpush.o                 lpush.s(12)
    leave.o                 leave.s(14)
    leasp.o                 leasp.s(8)
    ldaxsp.o                ldaxsp.s(8)
    incsp2.o                incsp2.s(8)
    incsp1.o                incsp1.s(8)
    icmp.o                  icmp.s(9)
    enter.o                 enter.s(8)
    decsp8.o                decsp8.s(8)
    decsp6.o                decsp6.s(8)
    decsp5.o                decsp5.s(8)
    decsp4.o                decsp4.s(8)
    decsp3.o                decsp3.s(8)
    decsp2.o                decsp2.s(8)
    decsp1.o                decsp1.s(8)
    and.o                   and.s(9)
    addysp.o                addysp.s(8)
    addeqsp.o               addeqsp.s(8)
    add.o                   add.s(11)
    vsprintf.o              vsprintf.s(10)
    sprintf.o               sprintf.s(9)
    memset.o                memset.s(18)
    agreed.o                agreed.s(10)
    atari.o                 zeropage.inc(11)
sreg (zeropage.o):
    ltoa.o                  ltoa.s(11)
    _printf.o               _printf.s(13)
    udiv.o                  udiv.s(9)
    shl.o                   shl.s(9)
```

```
    shelp.o                 shelp.s(12)
    popsreg.o               popsreg.s(9)
    mul.o                   mul.s(9)
    ludiv.o                 ludiv.s(9)
    lsub.o                  lsub.s(12)
    lpush.o                 lpush.s(12)
    icmp.o                  icmp.s(9)
    div.o                   div.s(12)
    axlong.o                axlong.s(8)
    clock.o                 clock.s(10)
    agreed.o                agreed.s(10)
staspidx (staspidx.o):
    agreed.o                agreed.s(2090)
    agreed.o                agreed.s(3687)
    agreed.o                agreed.s(6668)
stax0sp (staxsp.o):
    agreed.o                agreed.s(457)
    agreed.o                agreed.s(732)
    agreed.o                agreed.s(836)
    agreed.o                agreed.s(985)
    agreed.o                agreed.s(1173)
    agreed.o                agreed.s(1327)
    agreed.o                agreed.s(1826)
    agreed.o                agreed.s(1941)
    agreed.o                agreed.s(1976)
    agreed.o                agreed.s(2048)
    agreed.o                agreed.s(2471)
    agreed.o                agreed.s(2970)
    agreed.o                agreed.s(3616)
    agreed.o                agreed.s(3664)
    agreed.o                agreed.s(3909)
    agreed.o                agreed.s(4518)
    agreed.o                agreed.s(5643)
    agreed.o                agreed.s(6098)
    agreed.o                agreed.s(6513)
staxspidx (staxspi.o):
    agreed.o                agreed.s(2483)
staxysp (staxsp.o):
    agreed.o                agreed.s(830)
    agreed.o                agreed.s(979)
    agreed.o                agreed.s(1359)
    agreed.o                agreed.s(1534)
    agreed.o                agreed.s(1586)
    agreed.o                agreed.s(1670)
    agreed.o                agreed.s(1845)
    agreed.o                agreed.s(1886)
    agreed.o                agreed.s(2374)
    agreed.o                agreed.s(2651)
    agreed.o                agreed.s(2825)
    agreed.o                agreed.s(2950)
    agreed.o                agreed.s(3050)
    agreed.o                agreed.s(4346)
    agreed.o                agreed.s(5151)
    agreed.o                agreed.s(6080)
    agreed.o                agreed.s(6397)
    agreed.o                agreed.s(6761)
subeqysp (subeqsp.o):
    agreed.o                agreed.s(2116)
    agreed.o                agreed.s(2443)
```

```
    agreed.o                agreed.s(4053)
    agreed.o                agreed.s(4753)
    agreed.o                agreed.s(5727)
subysp (subysp.o):
    ucase_fn.o              ucase_fn.s(25)
    agreed.o                agreed.s(2250)
    agreed.o                agreed.s(2778)
    agreed.o                agreed.s(3736)
    agreed.o                agreed.s(4292)
    agreed.o                agreed.s(5132)
tmp1 (zeropage.o):
    ltoa.o                  ltoa.s(11)
    _printf.o               _printf.s(13)
    fdtable.o               fdtable.s(9)
    staxspi.o               staxspi.s(9)
    staspidx.o              staspidx.s(9)
    shrax1.o                shrax1.s(8)
    shelp.o                 shelp.s(12)
    mul.o                   mul.s(9)
    mod.o                   mod.s(12)
    incaxy.o                incaxy.s(8)
    div.o                   div.s(12)
    aslax3.o                aslax3.s(8)
    memset.o                memset.s(18)
    memmove.o               memmove.s(11)
    memcpy.o                memcpy.s(13)
    fread.o                 fread.s(14)
    graphics.o              graphics.s(20)
    getfd.o                 getfd.s(10)
    agreed.o                agreed.s(10)
tmp2 (zeropage.o):
    ucase_fn.o              ucase_fn.s(22)
    open.o                  open.s(22)
    fdtable.o               fdtable.s(9)
    shelp.o                 shelp.s(12)
    div.o                   div.s(12)
    rwcommon.o              rwcommon.s(7)
    graphics.o              graphics.s(20)
tmp3 (zeropage.o):
    ucase_fn.o              ucase_fn.s(24)
    open.o                  open.s(24)
    fdtable.o               fdtable.s(9)
    ludiv.o                 ludiv.s(9)
    rwcommon.o              rwcommon.s(7)
    graphics.o              graphics.s(20)
tmp4 (zeropage.o):
    open.o                  open.s(22)
    ludiv.o                 ludiv.s(9)
    mul40.o                 mul40.s(8)
tosadda0 (add.o):
    agreed.o                agreed.s(731)
    agreed.o                agreed.s(828)
    agreed.o                agreed.s(977)
    agreed.o                agreed.s(1172)
tosaddax (add.o):
    agreed.o                agreed.s(721)
    agreed.o                agreed.s(818)
    agreed.o                agreed.s(967)
    agreed.o                agreed.s(1162)
```

```
    agreed.o                agreed.s(1346)
    agreed.o                agreed.s(1682)
    agreed.o                agreed.s(4185)
tosandax (and.o):
    agreed.o                agreed.s(6371)
tosaslax (shl.o):
    agreed.o                agreed.s(6370)
tosdiva0 (div.o):
    agreed.o                agreed.s(3071)
tosgea0 (ge.o):
    agreed.o                agreed.s(3902)
    agreed.o                agreed.s(4590)
    agreed.o                agreed.s(6764)
tosicmp (icmp.o):
    lt.o                    lt.s(8)
    ge.o                    ge.s(8)
    agreed.o                agreed.s(746)
    agreed.o                agreed.s(867)
    agreed.o                agreed.s(1016)
    agreed.o                agreed.s(1199)
    agreed.o                agreed.s(1372)
    agreed.o                agreed.s(1891)
    agreed.o                agreed.s(2262)
    agreed.o                agreed.s(3057)
    agreed.o                agreed.s(4544)
    agreed.o                agreed.s(5755)
    agreed.o                agreed.s(6582)
toslong (toslong.o):
    agreed.o                agreed.s(3047)
toslta0 (lt.o):
    agreed.o                agreed.s(2379)
    agreed.o                agreed.s(3510)
    agreed.o                agreed.s(3871)
    agreed.o                agreed.s(4351)
    agreed.o                agreed.s(4511)
    agreed.o                agreed.s(5137)
    agreed.o                agreed.s(6321)
    agreed.o                agreed.s(6625)
tosmoda0 (mod.o):
    agreed.o                agreed.s(3098)
    agreed.o                agreed.s(3679)
    agreed.o                agreed.s(6407)
    agreed.o                agreed.s(6697)
tosmula0 (mul.o):
    agreed.o                agreed.s(1326)
    agreed.o                agreed.s(1653)
    agreed.o                agreed.s(2257)
    agreed.o                agreed.s(3792)
    agreed.o                agreed.s(3939)
    agreed.o                agreed.s(4373)
    agreed.o                agreed.s(4639)
    agreed.o                agreed.s(5584)
    agreed.o                agreed.s(6646)
tosmulax (mul.o):
    agreed.o                agreed.s(4184)
tossuba0 (sub.o):
    agreed.o                agreed.s(1279)
    agreed.o                agreed.s(6456)
tossubax (sub.o):
```

```
    agreed.o                  agreed.s(2018)
    agreed.o                  agreed.s(3797)
tossubeax (lsub.o):
    agreed.o                  agreed.s(3048)
tosudivax (udiv.o):
    fwrite.o                  fwrite.s(12)
    fread.o                   fread.s(12)
    agreed.o                  agreed.s(2261)
    agreed.o                  agreed.s(3796)
tosudiveax (ludiv.o):
    agreed.o                  agreed.s(3046)
tosumoda0 (umod.o):
    agreed.o                  agreed.s(2319)
tosumula0 (mul.o):
    agreed.o                  agreed.s(422)
    agreed.o                  agreed.s(720)
    agreed.o                  agreed.s(817)
    agreed.o                  agreed.s(966)
    agreed.o                  agreed.s(1161)
    agreed.o                  agreed.s(1487)
    agreed.o                  agreed.s(4888)
    agreed.o                  agreed.s(6606)
tosumulax (mul.o):
    fwrite.o                  fwrite.s(12)
    fread.o                   fread.s(12)
ucase_fn (ucase_fn.o):
    open.o                    open.s(25)
udiv16 (udiv.o):
    umod.o                    umod.s(8)
    mod.o                     mod.s(11)
    div.o                     div.s(11)
zerobss (zerobss.o):
    atari.o                   crt0.s(16)
```

CC65 Sourcecode#

```
/* -*- C -*- ************************************************************
 *
 *        ATARI Greed
 *                Copyright 2003-2004 Carsten Strotmann, Winfried Piegsda.
 *                based on greed for Unix,
 *                written by Matthew T. Day and Eric S. Raymond
 *
 *
 *
 *
 *  System        : Atari 800XL/130XL/800XE
 *  Module        :
 *  Object Name : $RCSfile: GameAtariGreed.txt,v $
 *  Revision        : $Revision: 1.3 $
 *  Date             : Fri Oct 30 12:00:00 2004
 *  Author        : Carsten Strotmann, Winfried Piegsda
 *  Created By   : <unknown>
 *  Created      : Sun Dec 7 15:50:35 2003
 *  Last Modified : <040425.2127>
 *
 *  Description A version of the game "greed" for the cc65 6502 C-Compiler,
```

```
 *                for ATARI 8-bit Microcomputer
 *
 *  Notes
 *          greed homepage: http://catb.org/~esr/greed/
 *
 *
 ************************************************************************
 */

#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>

#define MAXSCORE 10 /* max number of highscore entries */
#define rnd(x) (int) ((rand() % (x))+1) /* rnd() returns random num between 1 and x */
#define MK 20         /* marker of current screen location */
#define MAXX 38
#define MAXY 18
#define LEVELLINE 1
#define MENULINE 161
#define STATUSLINE 152
#define XOFFSET 1
#define YOFFSET 1
#define SOUNDBASE 8A00
#define TITLEBASE 0xD800
#define SAVEBASE  0xED60
#define SCORENAME 8
#define SCOREFILESIZE (MAXSCORE * (SCORENAME + 1 + sizeof(int)))    /* total byte size
#define LEVELFILESIZE 130
#define MAXMINUTES 8     // minutes for level
#define GET_WORD(p) (*(p) + ((unsigned) (p)[1] << 8))

extern void dli01(void);  /* is provided in dli.s */
extern void dli02(void);  /* is provided in dli.s */
extern void dli03(void);  /* is provided in dli.s */
extern void dli04(void);  /* is provided in dli.s */
extern void dli05(void);  /* is provided in dli.s */
extern char menuflg;
extern char fnt7;
extern char fnt14;

static unsigned char dllist[] = {        0x70, 0x70,                        // 2 * 8 S
                                        0x80,                              // one scanlin
                                        0x00,                              // 1 black Sca
                                        0x4f, 80, 161,                     // Gr.8 + load
                                        0x0f, 0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f,                        // seven Lines
                                        0xA0,                              // two empty s
                                        0x70, 0x00,                        // 9 Scanlines
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,            // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,            // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,            // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,            // 8 Gr.8 Line
```

```
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f,                     // 6 Gr.8 Line
                                        0x4f ,0, 176,                   // 1 Gr.8+Load
                                        0x0f,                           // 1 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x0f, 0x0f, 0x0f, 0x8f,
                                        0x0f, 0x0f, 0x0f, 0x0f,         // 8 Gr.8 Line
                                        0x70,                           // 8 Scanlines
                                        0x00,                           // 1 Scanline
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f,               // 7 Gr.8 Line
                                        0x70, 0x20,                     // 11 Scanline
                                        0x80,                           // 1 Scanline
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x0f, 0x0f, 0x0f, 0x0f,
                                        0x8f, 0x0f, 0x0f, 0x0f,         // 14 Gr.8 Lin
                                        0x00,                           // 1 Scanline
                                                    0x70,                           //
                                        0x41, 0x00, 0x28                     // Jump t
                  };
static char highscorefile[]= "D:AGREED.HIG";
static char continuemsg[] = "press key to continue...";
static char helpmsg[] = "Hit '?' for help";
static char fourdigitform[] = " %04d ";
static char levelbuf[130];

static int allmoves = 0;
static int score = 0;
static int levelscore;;
static int highscore = 0;
static unsigned int maxvalue;
static char level = 0, oldlevel = 0;
static char grid[MAXY][MAXX];
static int x,y;
static char havebotmsg = 0;
```

```
static char soundflg = 0;
static char exitflg = 0;
static int oldtime, maxtime;
static char scorelist[SCOREFILESIZE];

char* getscorename(char pos)
{
     return (&scorelist[pos *  (SCORENAME + 1 + sizeof(int))]);
}

int getscorevalue(char pos)
{
     char* p;
     p = getscorename(pos) + 9;
     return (*(p) + ((unsigned) (p)[1] << 8));
}


/* ATARI specific stuff */

void waitvbi(void)  // sync with vertical blank interrupt
{
     while (!*(char*) 0xD40B)
          continue;
}

void enable_os(void)
{
     if (*(char*) 0x2A00 == 0x4c) // OS Switch Routine loaded??
     {
          __asm__( "jsr $2a00" );
     }
}

void disable_os(void)
{
     if (*(char*) 0x2A00 == 0x4c) // OS Switch Routine loaded??
     {
          __asm__( "jsr $2a04" );
     }
}

void startmusic(void)
{
     if (*(char*) 0x8A00 == 0x68) // sound loaded??
     {
          __asm__( "jsr $8A01" );
     }
     soundflg = 1;
}

void stopmusic(void)
{
     if (*(char*) 0x8A00 == 0x68) // sound loaded??
     {
          __asm__( "jsr $8A08" );
     }
     soundflg = 0;
}
```

```
static char Asc2Int(unsigned char c)
{
     unsigned char x;
    x = c & 0x7f;

     if (x < 0x20  && x >= 1) c += 0x40;
    else if (x > 0x1f && x < 0x60) c -= 0x20;

    return c;
}

void clearblock(x, y, xl, yl)
char x;
char y;
char xl;
char xl;
{
     char yy;
     unsigned char *mptr;

     mptr = (char*) ((*(unsigned int*)0x58) + (40 * y) + x);

     for (yy = 0; yy < yl; yy++)
     {
          memset(mptr, 0, xl);
          mptr += 40;
     }
}

void saveblock(x, y, xl, yl)
char x;
char y;
char xl;
char xl;
{
     unsigned char *mptr, *dptr;

     mptr = (char*) ((*(unsigned int*)0x58) + (40 * y) + x);
     dptr = (char*) SAVEBASE;

     disable_os();
     for (y = 0; y < yl; y++)
     {
          memcpy(dptr, mptr, xl);
          mptr += 40;
          dptr += xl;
     }
     enable_os();
}

void restoreblock(x, y, xl, yl)
char x;
char y;
char xl;
char xl;
{
     unsigned char *mptr, *dptr;
```

```
     mptr = (char*) ((*(unsigned int*)0x58) + (40 * y) + x);
     dptr = (char*) SAVEBASE;

     disable_os();
     for (y = 0; y < yl; y++)
     {
          for (y = 0; y < yl; y++)
          {
               memcpy(mptr, dptr, xl);
               mptr += 40;
               dptr += xl;
          }
     }
     enable_os();
}

void invertblock(x, y, xl, yl)
char x;
char y;
char xl;
char xl;
{
     unsigned char *mptr;

     mptr = (char*) ((*(unsigned int*)0x58) + (40 * y) + x);

     for (y = 0; y < yl; y++)
     {
          for (x = 0; x < xl; x++)
          {
               *mptr ^= 0xFF;
               mptr++;
          }
          mptr += (40 - xl);
     }
}

static void gputcxyvar(int x, int y, unsigned char c, unsigned char* chptr, char maxli
{
     unsigned char *mptr;
     int ch;
     int inv;

     inv = (0xff * (c > 0x7f));
     c = c & 0x7f;

     mptr  = (char*) ((*(unsigned int*)0x58) + y * 40 + x);

     for (ch = 0; ch < maxlines; ch++)
     {
          mptr += 40;
          chptr++;
          *mptr = (*chptr ^ inv);
     }
}

static void gputcxy7(int x, int y, unsigned char c)
{
  gputcxyvar(x,y,c, (char*) (&fnt7 + c * 7), 6);
```

```
}

static void gputcxy14(int x, int y, unsigned char c)
{
     gputcxyvar(x,y,c, (char*) (&fnt14 + c * 14), 13);
}

void gprintxy14 (int x, int y, char* str)
{
     while (*str != '\0')
     {
            gputcxy14(x++,y,*str++);
     }
}

void gprintxy7 (int x, int y, char* str)
{
     while (*str != '\0')
     {
        gputcxy7(x++,y,*str++);
     }
}

static void gputcxy(int x, int y, unsigned char c)
{
     unsigned char *chptr, *mptr, *sptr;
     char ch, inv;

    x += XOFFSET;
    y += YOFFSET;

    c = Asc2Int(c);
     inv = (0xff * (c > 0x7f));
     c = c & 0x7f;

     y = y * 8;
     sptr = mptr = (char*) ((*(unsigned int*)0x58) + y * 40 + x);
    chptr = (char*) ((*(unsigned char*)0x2f4) * 0x100 + (c * 8));

     for (ch = 0; ch < 8; ch++)
     {
          *mptr = (*chptr ^ inv);
          mptr += 40;
          chptr++;
     }
     if (inv) // round edges
     {
          *sptr = (*sptr ^ 0x81);
          sptr += 40*7;
          *sptr = (*sptr ^ 0x81);
     }
}

void gprintxy (int x, int y, char* str)
{
     char* s;
     s = str;
     while (*s != '\0')
     {
```

```
          gputcxy(x++,y,*s++);
     }
}

void pause(int ticks)
{
     int i;
     char rtclk3;

     for (i = 0; i < ticks; ++i)
     {
          rtclk3 = *(char*) 0x14;
          while (rtclk3 == *(char*) 0x14) {}
     }
}


/* File IO Functions */

int Fread (FILE* F, void* Buf, unsigned Size)
{
     size_t Res;
     Res = fread (Buf, 1, Size, F);
     return Res > 0? Res : 0;
}

int Fwrite (FILE* F, const void* Buf, unsigned Size)
{
     size_t Res;
     Res = fwrite (Buf, 1, Size, F);
     return Res > 0? Res : 0;
}

void statusline(char* str)
{
    gprintxy7(0,STATUSLINE,"                                                    ");
    gprintxy7(20 - strlen(str) / 2,STATUSLINE,str);
}

void getname(char x, char y, char* nameptr)
{
     char i;
     char* str;
     str = nameptr;

     statusline("please enter your name:");
     memset(nameptr,0,8);
     i = 0;
     gprintxy7(x,y,"_");
     while ((*nameptr = cgetc()) != 155)
     {
          if (++i > 7)
          {
                --i;
                --nameptr;
          }
          if ((*nameptr) == 126)
          {
                if (i)
```

```
                {
                  --i;
                  --nameptr;
                }
          }
          else
                ++nameptr;

         *nameptr = '_';
         gprintxy7(x,y,"         ");
         gprintxy7(x,y,str);
     }
     gputcxy7(x+i,y,' ');
     *nameptr = '\0';
}

void topscores(newscore)
int newscore;
{
     int i, j;
     char buf[8];
     int* p;

     if (levelbuf[2] <= (levelscore * 100 / maxvalue))
     {
          char c;
          c = (levelscore * 100 / maxvalue);
          sprintf(levelbuf,"next level! %d bonus!", c);
          statusline(levelbuf);
          score += c;
          level = (++level % 9); // next level
          c = cgetc();
     }

     saveblock(3,12,33,116);
     clearblock(3,12,33,116);

     gprintxy7(13,16,"highscores");

     for (i = 0; i < MAXSCORE; ++i)
     {
          if (getscorevalue(i) < newscore)
          {
                if (i < MAXSCORE)
                {
                     for (j = MAXSCORE-1; j >= i; --j)
                     {
                          memcpy(getscorename(j+1),getscorename(j), 11);
                     }
                }

                getname(13,28+i*8,getscorename(i));
                p = (int*) (getscorename(i) + 9);
                *p = newscore;
                newscore = 0;
          }

          sprintf(buf,"%2d.",i + 1);
          gprintxy7(8,28+i*8,buf);
```

```
          gprintxy7(13,28+i*8,getscorename(i));
          sprintf(buf,fourdigitform,getscorevalue(i));
          gprintxy7(25,28+i*8,buf);
      }
      statusline("Press a key...");
      i = cgetc();
      restoreblock(3,12,33,116);
      statusline(helpmsg);
}

void loadscore()
{
    FILE* file;
    int rc;
    char i;

    file = fopen(highscorefile,"r");
     if (file)
     {
          rc = Fread (file, scorelist, SCOREFILESIZE);
          fclose(file);
          highscore = getscorevalue(0);
     }
     else
     {
          statusline("No Highscorefile, creating new file!");
          for (i = 0; i <= MAXSCORE; ++i)
          {
                memset(getscorename(i), ' ',8);
                *(getscorename(i) + 9) = 0;
                *(getscorename(i) + 10) = 0;
          }
          i = cgetc();
     }
}

void loadlevel(char level)
{
    FILE* file;
     char filename[14];
    int rc;
    char i, s;

     s = soundflg;
     if (s)
          stopmusic();
     *(char*) 0x22f = 0;
     pause(2);

     sprintf(filename,"LEVEL%02d.AGL", level);

    file = fopen(filename,"r");
     if (file)
     {
      rc = Fread (file, levelbuf, LEVELFILESIZE);
      fclose(file);
        *(char*) 0x22f = 0x22;
        oldlevel = level;
```

```
      }
     else
     {
          *(char*) 0x22f = 0x22;
          statusline("Could not load level!");
          i = cgetc();
     }
     if (s)
        startmusic();
}

void savescore()
{
    FILE* file;
    int rc;

     stopmusic();
     *(char*) 0x22f = 0;
     pause(2);

     statusline("Saving Highscorefile...");

    file = fopen(highscorefile,"w");
     if (file)
     {
          rc = Fwrite (file, scorelist, SCOREFILESIZE);
          fclose(file);
     }
     else
          statusline("Error saving Highscorefile!");
     *(char*) 0x22f = 0x22;
}

void resettime(void)
{
     *(int*) 0x13 = 0; // reset RTCLOK to zero
}

void showtime(void)
{
     int time;
     char buf[4];

     time = maxtime - (clock() / _clocks_per_sec()); // time in seconds
     if (time != oldtime)
     {
          sprintf(buf," %02d", time / 60); // minutes
          gprintxy7(32,LEVELLINE,buf);
          sprintf(buf,":%02d", time % 60); // seconds
          gprintxy7(35,LEVELLINE,buf);
          oldtime = time;
     }
}

void help(void)
{

     char c;
```

```
     saveblock(3,12,33,116);
     clearblock(3,12,33,116);

     gprintxy7(3,20," ATARI greed help                  ");
     gprintxy7(3,28," 'M' = toggle music                 ");
     gprintxy7(3,36," 'Q' = quit game                    ");
     gprintxy7(3,44," 'P' = show possible moves          ");
     gprintxy7(3,52," 'ESC' = toggle menu                ");
     gprintxy7(3,64," use joystick or keys to move  ");
     gprintxy7(3,72,"        W  E   R                ");
     gprintxy7(3,80,"           \\  |  /                   ");
     gprintxy7(3,88,"        S -  +  - D                 ");
     gprintxy7(3,96,"           /  |  \\                   ");
     gprintxy7(3,104,"       Z  X   C                ");
     statusline(continuemsg);
     c = cgetc();
     restoreblock(3,12,33,116);
     statusline(helpmsg);
}

void info(void)
{

     char c;

     saveblock(3,12,33,116);
     clearblock(3,12,33,116);

     gprintxy7(3,20," ATARI greed                           ");
     gprintxy7(3,28," based on the UNIX game 'greed'  ");
     gprintxy7(3,36," written by matthew t. day          ");
     gprintxy7(3,44," and eric s. raymond                ");
     gprintxy7(3,60," programmed on an apple mac        ");
     gprintxy7(3,68," by carsten strotmann              ");
     gprintxy7(3,76," using the cc65 crosscompiler  ");
     gprintxy7(3,92," sound made by winfried piegsda  ");
     gprintxy7(3,100," using the pegasus soundmonitor  ");
     gprintxy7(3,108," graphic design by w. piegsda     ");
     statusline(continuemsg);
     c = cgetc();
     restoreblock(3,12,33,116);
     statusline(helpmsg);
}

void showmarker(void)
{
     int i = 0;

     for (; i < 10; ++i);
     {
          pause(2);
          gputcxy(x,y,MK + 0x80);
          gputcxy(x,y,MK);
     }
}

void botmsg(msg)
char *msg;
{
```

```
     statusline(msg);
     havebotmsg = 1;
}

void quit() {
     int ch;

     botmsg("Really quit?",0);


     if ((ch = cgetc()) != 'y' && ch != 'Y') {
          return;
     }
     exitflg = 1;
}

void earthquake(void)
{
  char i;
  char *p;

  p = (char*) 0x2800;

  for (i=0; i < 20; ++i)
  {
     *p = rnd(7) * 0x10;
     pause(rnd(3));
    }
  *p = 0x70;
}


void showscore(void)
{
     char buf[8];
     char perc;
     sprintf(buf,fourdigitform, score);
     gprintxy14(6,MENULINE,buf);
     sprintf(buf,fourdigitform, highscore);
     gprintxy14(22,MENULINE,buf);
     perc = levelbuf[2] - ((levelscore * 100) / maxvalue);
     if (perc > 100)
          perc = 0;
     sprintf(buf,"%3d%%",perc);
     gprintxy14(36,MENULINE,buf);
}

void showmoves(on)
int on;
{
     int dy = -1;
     int dx;

     for (; dy <= 1; ++dy) {
          if (y+dy < 0 || y+dy >= MAXY) continue;
          for (dx = -1; dx <= 1; ++dx) {

                int j=y, i=x, d=grid[y+dy][x+dx];
```

```
                if (!d) continue;
                do {
                     j += dy;
                     i += dx;
                     if (j < 0 || i < 0 || j >= MAXY || i >= MAXX || !grid[j][i]) brea
                } while (--d);

                if (!d) {
                     int j=y, i=x, d=grid[y+dy][x+dx];

                     /* The next section chooses inverse-video   *
                      * or not, and then "walks" chosen valid   *
                      * move, reprinting characters with new mode */

                     do {
                          j += dy;
                          i += dx;
                          gputcxy(i, j,  grid[j][i] + '0' + (0x80 * on)); /* print pos
                     } while (--d);

                }
           }
      }

}

void printscoreline(void)
{
     gprintxy14(0,MENULINE,"Score:");
     gprintxy14(12,MENULINE,"Highscore:");
     gprintxy14(29,MENULINE,"Finish:");
}

void refresh()
{
     int y,x;
     char levelname[33];

     statusline("refreshing screen...");
     clearblock(0,8,39,MAXY*8);

     memcpy(levelname,(char*) (levelbuf + 3),32);
     levelname[32] = '\0';
     gprintxy7(0,LEVELLINE,levelname);
     printscoreline();

     for (y=0; y < MAXY; ++y)
          for (x=0; x < MAXX; ++x)
                if (grid[y][x])
                     gputcxy(x,y,grid[y][x] + '0');

     showmarker();
     showmoves(allmoves);
     showscore();
     statusline("hit esc for menu");
}
```

```
int othermove(bady, badx)
int bady, badx;
{
     int dy = -1;
     int dx;

     for (; dy <= 1; ++dy)
          for (dx = -1; dx <= 1; ++dx)
               if ((!dy && !dx) || (dy == bady && dx == badx) || y+dy < 0 && x+dx < 0
                    continue;
               else
               {
                    int j = y;
                    int i = x;
                    int d = grid[y+dy][x+dx];

                    if (!d) continue;

                    do {
                         j += dy;
                         i += dx;
                         if (j < 0 || i < 0 || j >= MAXY || i >= MAXX || !grid[j][i])
                    } while (--d);
                    if (!d) return 1;
               }
     return 0;
}

void menu(void)
{
     char c;
     char mlen[] = {8,9,4,4,4};
     char mchoice = 0;
     char moffset = 0;

     // set and enable DLI

     waitvbi();
     menuflg = 1;    // switch DLI colors
     clearblock(0, MENULINE, 40, 14);
     statusline("Choose Menu...");

     while (c != 27)
     {
          gprintxy14(1,MENULINE,"Continue  Highscore  Info  Help  Quit");
          invertblock((2 * mchoice) + moffset + 1, MENULINE + 1, mlen[mchoice], 14);
          c = cgetc();
          switch(c)
          {
               case 30: // cursor left
                    if (mchoice > 0)
                    {
                         --mchoice;
                         moffset -= mlen[mchoice];
                    }
                    break;
               case 31: // cursor right
                    if (mchoice < 4)
                    {
```

```
                        moffset += mlen[mchoice];
                        ++mchoice;
                    }
                    break;
                case 155: // enter
                    switch (mchoice)
                    {
                        case 0:
                            c = 27; // exit menu
                            break;
                        case 1:
                            topscores(score);
                            break;
                        case 2:
                            info();
                            break;
                        case 3:
                            help();
                            break;
                        case 4:
                            quit();
                            c = 27;
                            break;
                    }
                    break;
            }
        }

        waitvbi();
        menuflg = 0;

        clearblock(0, MENULINE, 40, 14);
        printscoreline();
        statusline(helpmsg);
       showscore();
}

int tunnel(cmd)
char cmd;
{
        int dy, dx, distance;
        int i,j,d;

        if (oldtime <= 0)
        {
            statusline ("T I M E  O U T !!");
            i = cgetc();
            return(0); // timeout
        }

        switch(cmd)
        {
           case 27: /* ESC */
               menu();
                   if (exitflg)
                       return(0);
                   else
                       return(1);
               break;
```

```
     case 't':
     case 'T': /* top scores */
           topscores(score);
           return(1);
           break;
     case 'm': /* sound off */
     case 'M':
           if (soundflg)
           {
                stopmusic();
                statusline("music off");
           }
           else
           {
                startmusic();
                statusline("music on");
           }
           return(1);
           break;
case 's': /* key left */
case 'S':
case '4':
     dy = 0;
     dx = -1;
     break;
case 'x': /* key down */
case 'X':
case '2':
     dy = 1;
     dx = 0;
     break;
case 'e': /* key up */
case 'E':
case '8':
     dy = -1;
     dx = 0;
     break;
case 'd': /* key right */
case 'D':
case '6':
     dy = 0;
     dx = 1;
     break;
case 'z': /* key left/down */
case 'Z':
case '1':
     dy = 1;
     dx = -1;
     break;
case 'c': /* key right/down */
case 'C':
case '3':
     dy = dx = 1;
     break;
case 'w': /* key left/up */
case 'W':
case '7':
     dy = dx = -1;
     break;
```

```
 case 'r': /* key right/up */
 case 'R':
 case '9':
      dy = -1;

      dx = 1;
      break;
 case 'p':
 case 'P':
      allmoves = !allmoves;
      showmoves(allmoves);
      return(1);
 case 'q':
 case 'Q':
      quit();
      if (exitflg)
            return(0);
      else
            return(1);
case 'o':
case 'O':
    earthquake();
    return(1);
 case '?':
      help();
      return(1);
 case 'i':
 case 'I':
      info();
      return(1);
 case 'a':
 case 'A':
      refresh();

      /* refresh; falls through to return */
 default:
      return(1);
 }

 distance = (y + dy >= 0 && x + dx >= 0 && y + dy < MAXY && x + dx < MAXX) ? grid[

 j = y;
 i = x;
 d = distance;

 do {
      j += dy;
      i += dx;

      if (j >= 0 && i >= 0 && j < MAXY && i < MAXX && grid[j][i])
            ;
      else if (!othermove(dy, dx)) { /* no other good move */
            j -= dy;
            i -= dx;
            gputcxy(x,y,' ');
            while (y != j || x != i) {
                 y += dy;
                 x += dx;
                 ++score;
```

```
                    ++levelscore;
                    if (score > highscore)
                         highscore = score;
                    gputcxy(x,y,' ');
               }
               gputcxy(x,y,'*');
               showscore();
               topscores(score);
               return(0);
          }
          else
          {
               botmsg("Bad move!",1);
               return(1);
          }

     } while (--d);

     if (allmoves) showmoves(0);

     if (havebotmsg) {          /* if old bottom msg exists */
          printscoreline();
          statusline(helpmsg);
          havebotmsg = 0;
     }

     gputcxy(x,y,' ');
     do {
          y += dy;
          x += dx;
          ++score;
          ++levelscore;
          if (score > highscore)
               highscore = score;
          grid[y][x] = 0;
          gputcxy(x,y,' ');
     } while (--distance);
     gputcxy(x,y,MK);
     if (allmoves) showmoves(1);
     showscore();
     return(1);
}

void intro(void)
{
     char y;
     char x;
     unsigned char *chptr;
     unsigned char *mptr;

     memset((char*) (*(unsigned int*)0x58), 0, 40 * 8); // clear levelname
     chptr = (char*) TITLEBASE;
     mptr  = (char*) ((*(unsigned int*)0x58) + (40*8)+1);

     disable_os();
     for (y = 0; y < 144; ++y)
     {
          memcpy(mptr,chptr,MAXX);
          chptr += MAXX;
```

```
          mptr += 40;
     }
     enable_os();
     statusline("ATARI greed version 0.91");
     pause(90);
     statusline("press key to start game...");
     x = cgetc();
}

char getcommand(void)
{
     char c = 0;
     char j = 0;

     while (c == 0)
     {
          if ((*(char*) 0x2fc) != 0xFF) // key pressed?
          {
                c = cgetc();
          }
          j = *(char*) 0x278;
          if (j != 0xf) // Joystick?
          {
                switch(j)
                {
                     case 14:
                          c = 'e'; // up
                          break;
                     case 7:
                          c = 'd'; // right
                          break;
                     case 13:
                          c = 'x'; // down
                          break;
                     case 11:
                          c = 's'; // left
                          break;
                     case 9:
                          c = 'z'; // left/down
                          break;
                     case 5:
                          c = 'c'; // right/down
                          break;
                     case 10:
                          c = 'w'; // left/up
                          break;
                     case 6:
                          c = 'r'; // right/up
                          break;
                }
          }
          showtime();
          if (oldtime <= 0) c = ' ';
     }
     return(c);
}

int bittest(val,bit)
int val, bit;
```

```
{
     return !(val & (1 << bit));
}

char getplayfield(char x, char y)
{
     int* p;
     char c;
     p = (int*) (levelbuf + 35);
     while (bittest(*p, c = rnd(9)))
               ;
     p = (int*) (levelbuf + 40 + (y * 5 ) + (x / 8));

     return(bittest(*p, 7 - (x % 8)) ? 0 : c);
}

int main(void) {
     int val = 1;
     int dllist_old;

     graphics(8);
    highscore = 0;

    *(char*) 0x02c6 = 0;
    *(char*) 0x02c5 = 0xF;

    dllist_old = *(int*) 0x230;
     memmove((char*) 0x2800, &dllist, sizeof(dllist));
     *(int*) 0x230 = (int) 0x2800;

    // set and enable DLI

    *(int*) 0x200 = (int) &dli01;
    *(char*) 0xd40e = 0xc0;

     loadscore();
     loadlevel(level);
     startmusic();

     while(!exitflg)
     {
          intro();
          statusline("starting new game...");
          if (level != oldlevel)
             loadlevel(level);
          score = levelscore = maxvalue = 0;
          maxtime = levelbuf[39] * 60; // timeout minutes

          srand(*(int*) 0xD20A); /* initalize seed with random number, ATARI specific

          for (y=0; y < MAXY; ++y)
                for (x=0; x < MAXX; ++x)
                     if (grid[y][x] = getplayfield(x,y))
                          ++maxvalue;

          while (!getplayfield(y = rnd(MAXY)-1, x = rnd(MAXX)-1))
                    ;       /* random initial location */

          grid[y][x] = 0; /* eat initial square */
```

```
          refresh();
          resettime();

          while((val = tunnel(getcommand())) > 0)
                continue;

     }

     topscores(score);
     stopmusic();
     savescore();

    // disable DLI

    *(char*) 0xd40e = 0x60; // NMIEN VBI and RESET on
    graphics(0);

     exit(0);
}
```

Assembler Code (ca65)#

Display List Interrupts#

```
    .include "/Users/cas/develop/cc65/asminc/atari.inc"
    .export _dli01
    .export _dli02
    .export _dli03
    .export _dli04
    .export _dli05
     .export _menuflg
     .export _fnt7
     .export _fnt14

_fnt7:       .incbin "seven.fnt"
_fnt14:     .incbin "fourteen.fnt"

_menuflg:    .byte 0

.proc   _dli01

    pha
    txa
    pha

     nop
     nop
    ldx DLI02cnt2

L1:
    lda DLI02fade2-1,x
    sta WSYNC
    sta COLBK
    sta COLPF2
     nop
    dex
```

```
    bne L1

    lda #<_dli02
    ldx #>_dli02
    sta VDSLST
    stx VDSLST+1

    pla
    tax
    pla

    rti

.endproc

DLI02fade:  .byte $9E, $9C, $9A, $98, $96, $94, $92, $90
DLI02fadem: .byte $0E, $0C, $0A, $08, $06, $04, $02, $00
DLI02cnt:   .byte 8
DLI02fade2: .byte $00, $02, $04, $06, $0E
DLI02fade2m:    .byte $90, $92, $94, $96, $9E
DLI02fadepm:     .byte $98, $98, $9A, $9C, $9E
DLI02cnt2:  .byte 5

.proc   _dli02

    pha
    txa
    pha

    ldx DLI02cnt
    dex
L1:
    lda _menuflg
     beq X1
    lda DLI02fadem-1,x
     bne X2
X1:
    lda DLI02fade-1,x
X2:
    sta WSYNC
    sta COLBK
    sta COLPF2
    dex
    bne L1

    sta WSYNC
     sta COLPM0
     sta COLPM1
     lda #$2C
     sta HPOSP0
     lda #$CC
     sta HPOSP1
     lda #$FF
     sta GRAFP0
     sta GRAFP1
     lda %00000010
     sta GRACTL
    ldx DLI02cnt2
L2:
```

```
    lda DLI02fade2-1,x
    sta WSYNC
    sta COLPF2
     lda DLI02fadepm-1,x
     sta COLPM0
     sta COLPM1
    dex
    bne L2

    lda #<_dli03
    ldx #>_dli03
    sta VDSLST
    stx VDSLST+1

    pla

    tax
    pla

    rti

.endproc

DLI03fade:  .byte $90, $92, $94, $96, $98, $9A, $9C, $9E
DLI03fadem: .byte $00, $02, $04, $06, $08, $0A, $0C, $0E
DLI03cnt:   .byte 8
DLI03fade2m:    .byte $9E, $96, $94, $92, $90
DLI03fadepm:     .byte $9E, $9C, $9A, $98, $98
DLI03fade2: .byte $0E, $06, $04, $02, $01
DLI03cnt2:  .byte 5

.proc   _dli03

    pha
    txa
    pha

    ldx DLI03cnt2
L1:
    lda DLI03fade2-1,x
    sta WSYNC
    sta COLPF2
     lda DLI03fadepm-1,x
     sta COLPM0
     sta COLPM1
    dex
    bne L1

    sta WSYNC
     lda #0
     sta HPOSP0
     sta HPOSP0
     lda #$0
     sta GRAFP0
     sta GRAFP1


    ldx DLI03cnt
    dex
```

```
L2:
     lda _menuflg
     beq X1
    lda DLI03fadem-1,x
     bne X2
X1:
    lda DLI03fade-1,x
X2:
    sta WSYNC
    sta COLBK
    sta COLPF2
    dex
    bne L2

    sta WSYNC
    stx COLBK
    stx COLPF2

    lda #<_dli04
    sta VDSLST
    lda #>_dli04
    sta VDSLST+1

    pla
    tax
    pla

    rti

.endproc

.proc   _dli04

    pha
    txa
    pha

    ldx DLI02cnt2
L4:
     lda _menuflg
     beq X1
    lda DLI02fade2m-1,x
     bne X2
X1:
    lda DLI02fade2-1,x
X2:
    sta WSYNC
    sta COLBK
    sta COLPF2
    dex
    bne L4

    stx WSYNC
    stx COLBK
    stx COLPF2

    lda #<_dli05
    sta VDSLST
    lda #>_dli05
```

```
    sta VDSLST+1

    pla
    tax
    pla

    rti

.endproc

.proc   _dli05

    pha
    txa
    pha

    ldx DLI03cnt2
L4:
     lda _menuflg
     bne X1
    lda DLI03fade2-1,x
     bne X2
X1:
     nop
     nop
    lda DLI03fade2m-1,x
X2:
    sta WSYNC
    sta COLBK
    sta COLPF2
    dex
    bne L4

    sta WSYNC
    stx COLBK
    stx COLPF2

    lda #<_dli01
    sta VDSLST
    lda #>_dli01
    sta VDSLST+1

    pla
    tax
    pla

    rti

.endproc
```

Access RAM under OS#

```
; RAMXL
; routines to access RAM under OS-ROM

     .word $FFFF
     .word $2a00
     .word end-1
```

```
     .org $2a00

intv  = $FFF0
nmiv  = $FFFA
resv  = $FFFC
irqv  = $FFFE

portb   =   $D301
nmien   =   $D40E

on:         jmp os_on
x_save:   .byte $00
off:          jmp os_off
;
doirq:      stx x_save
                tax         ; a - irq #
                jsr os_on
                lda intv,x
                sta jmpvec+1
                lda intv+1,x
                sta jmpvec+2

                lda #>return
                pha
                lda #<return
                pha
                cli
                php
;
jmpvec:   jmp $FFFF ; -- will be overwritten
;
return:   jsr os_off
                ldx x_save
                pla
                rti
;
nmi_han:     pha
                lda #$0A
                jmp doirq
;
irq_han:     pha
                lda #$0E
                jmp doirq
;
os_on:      lda portb
                ora #$01  ; toggle OS bit on
                sta portb
                rts
;
os_off:   lda portb
                and #$FE    ; toggle OS bit off
                sta portb
                rts
;
install:     lda #0
                sta nmien
                sei
                jsr os_off
```

```
                lda #<nmi_han
                sta nmiv
                lda #>nmi_han
                sta nmiv+1

                lda #<irq_han
                sta irqv
                lda #>irq_han
                sta irqv+1

                jsr os_on
                cli
                lda #$40
                sta nmien
                rts

memcpy:
                src = $f0
                dst = $f2
                cnt = $f4
; src = $F0-$F1
; dst = $F2-$F3
; cnt = $F4-$F5
                ldy #0
L1:
                lda (src),y
                sta (dst),y

                inc src
                bne L2
                inc src+1
L2:
                inc dst
                bne L3
                inc dst+1
L3:
                dec cnt
                bne L1
                dec cnt+1
                bpl L1
                rts

movetitle:
                lda #<titlestart
                sta src
                lda #>titlestart
                sta src+1

                lda #<$D800 ; $D800
                sta dst
                lda #>$D800
                sta dst+1

                lda #<5472
                sta cnt
                lda #>5472
                sta cnt+1

                jsr install
```

```
                jsr os_off
                jsr memcpy
                jsr os_on
                rts
end:

;--------------------------

     .word $FFFF
     .word $3000
     .word end2-1
     .org $3000

titlestart:
     .incbin "titelbild.raw"
end2:

     .word $FFFF
     .word $02e2
     .word end3-1
     .org $02e2
     .word movetitle
```

Level Files#

Level 1 (Example)#

```
; level 0 for ATARI Greed
; Version 1.0
; total length 130 bytes

; Magic Code, 'AG'
magic:  .byte   "AG"

; percent needed to complete level
percent: .byte 65

; Level Title 32 Chars
title:  .byte   "aller anfang ist einfach..."
          .res $20 - (* - title)
; Possible values (bitfield), 10 bits
;                   FEDCBA9876543210
values: .word %0000001111111111 ; 0-9

; Possible goodies (bitfield), 16 Bits
;                   FEDCBA9876543210
goodies: .word %0000000000000000 ; no goodies

; Time in minutes
time:   .byte 8 ; 8 Minutes time

; levelmask 5 x 18 bytes (bitfield)
levelmask: .incbin "level00.raw"
```