

USER GUIDE

Char-EDITOR V.0.0.7 X86-WIN32 TOOL FOR

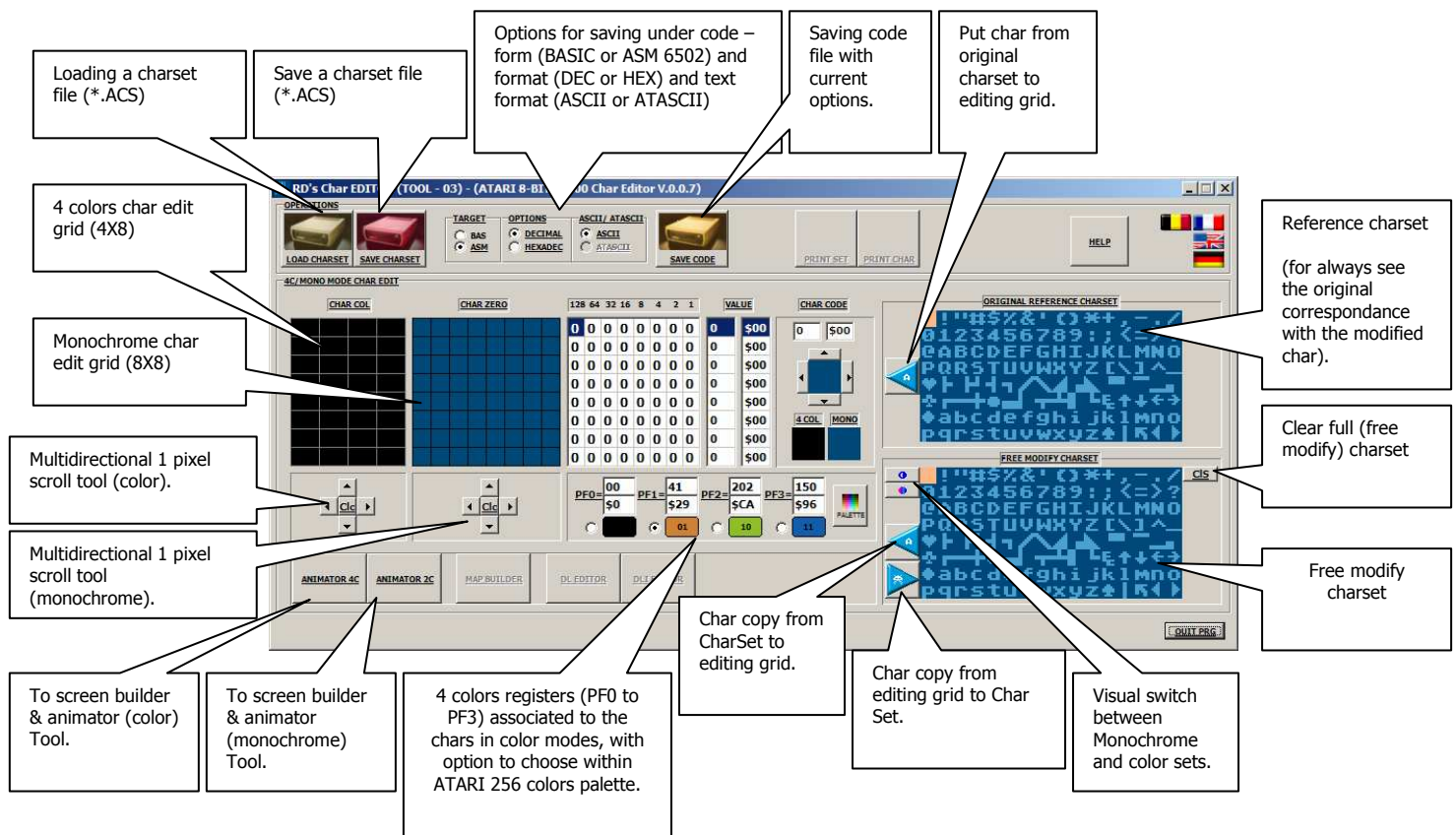


1. Create a character set

1.1. Displays

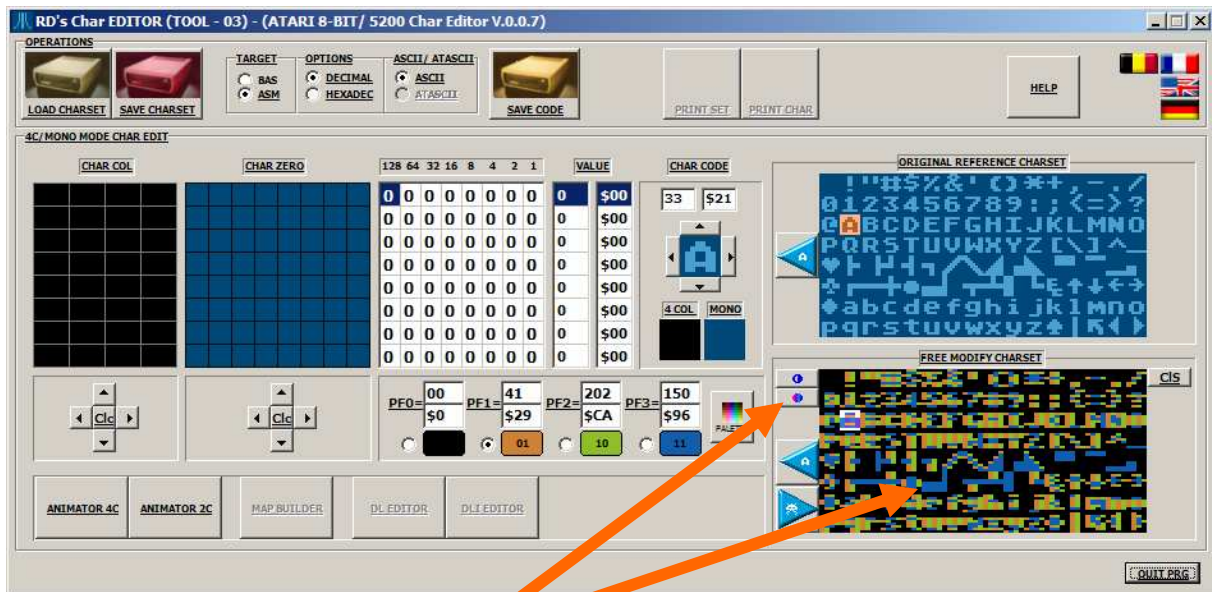
The main screen of program is organized and the central area of window is the one to edit a char set of 128 items.

On ATARI 8-BIT, the last bit (7) only say to put chars in inverted-video to the system, so we just need the first 128 chars.

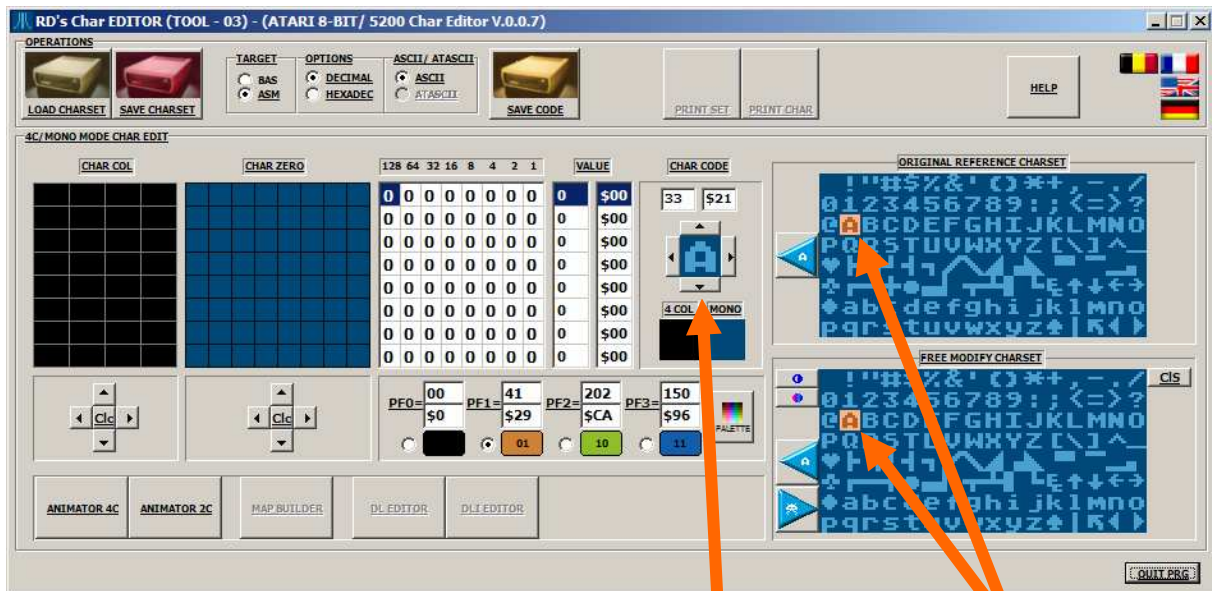


1.2. Modify the original character set.

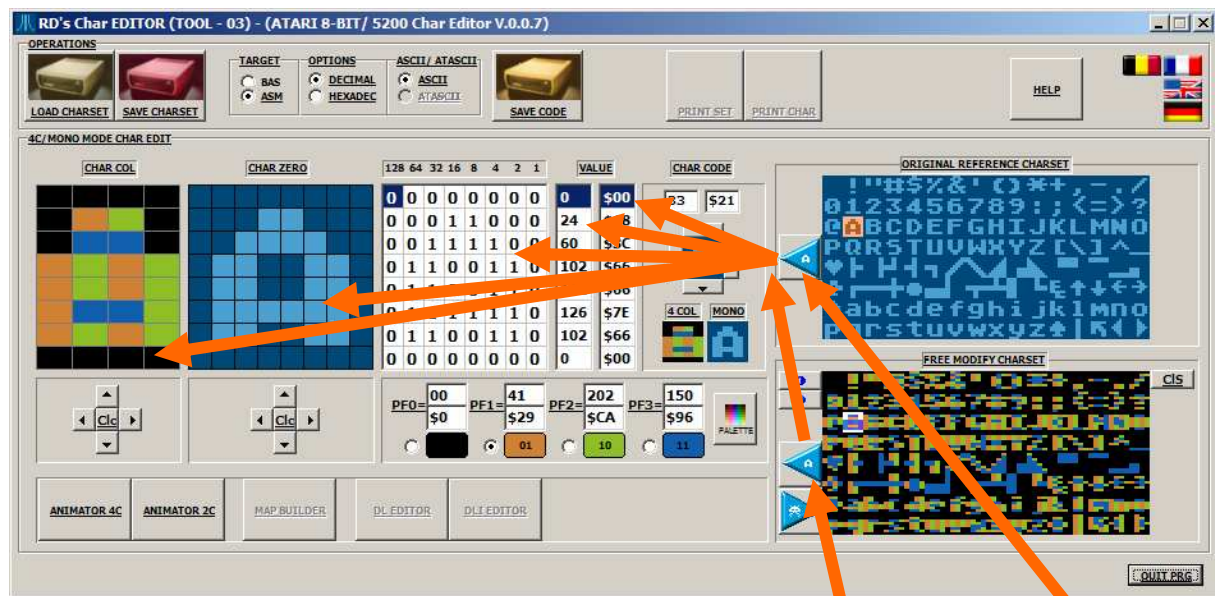
1.2.1. First : Switch to the color mode.



1.2.2. Second : choose a character to modify within the 128 ones.



1.2.3. Third : Character transfer to edit.

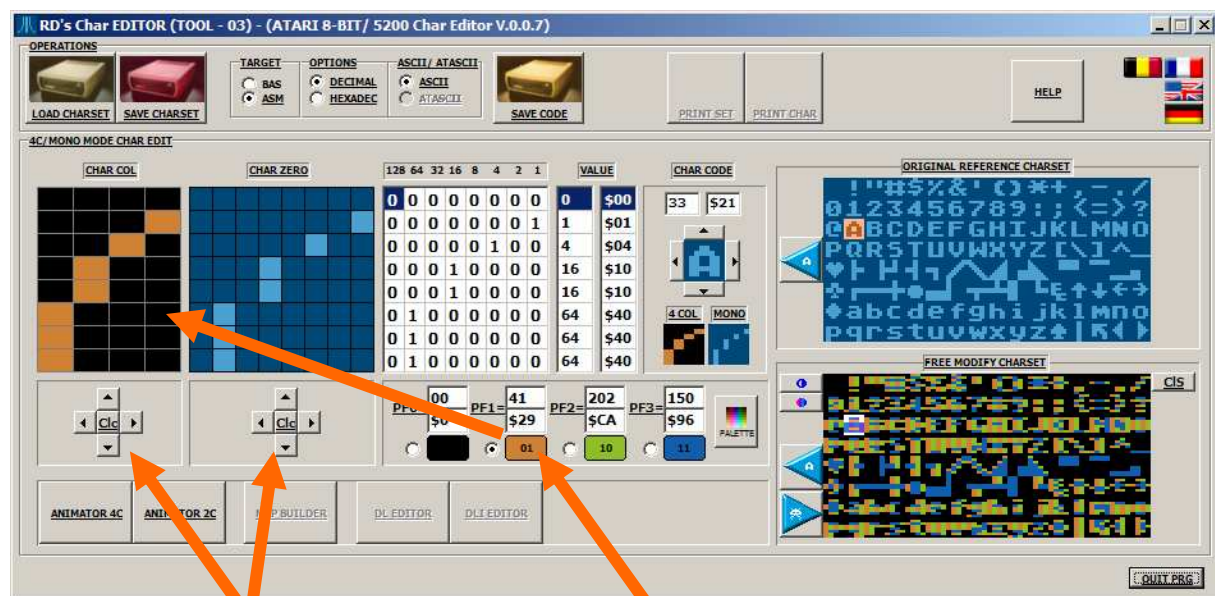


1 > We can copy an original char to grid edit (Char from the ROM of ATARI 8-BIT).

2 > Another way, we can copy an original OR already modified char to grid edit.

3/ At this time the selected char « A » is copied in the edit grid, both color and monochrome char in same time.
In addition we have a visualisation of the binary-decimal-hexadecimal grid for each graphic line.

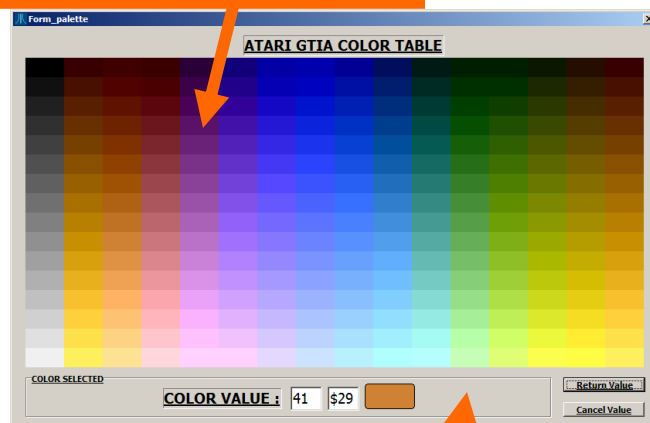
1.2.4. Fourth : Edit the choosen character



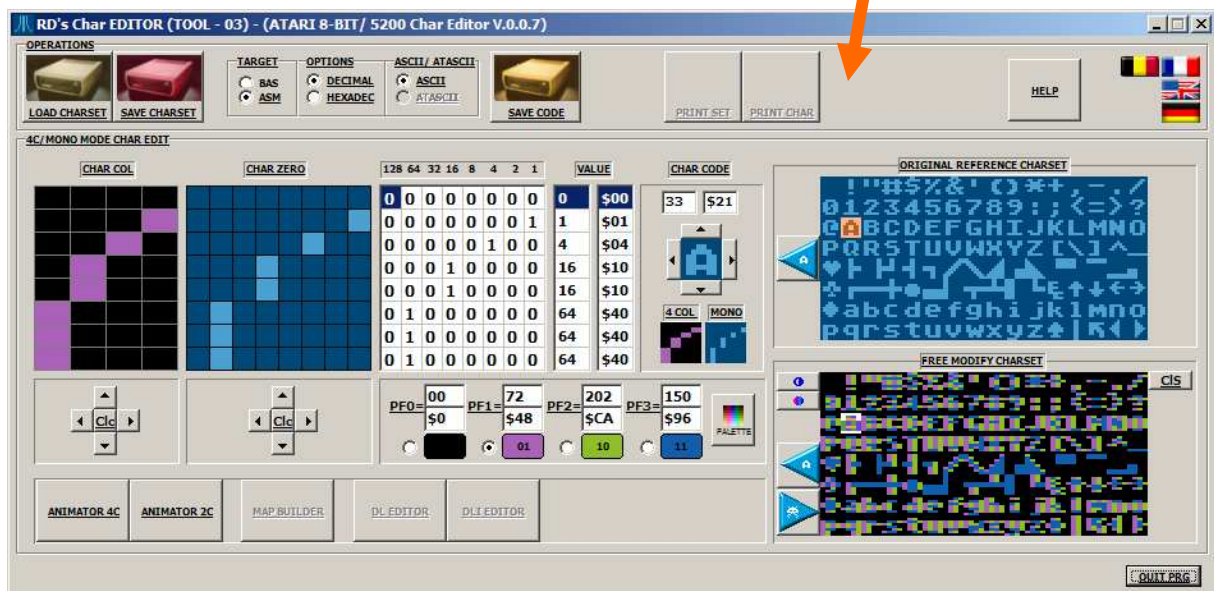
Few little tools are available here, as multidirectional scrolling (1 pixel on the grid) and option to clear the full grids.

With the select of one color within the 4, we can drawing directly pixel by pixel on the color/ monochrome grid.
Warning : As a color pixel is 2 bit large, the monochrome grid show a different draw, this is normal when we use color modes !!!

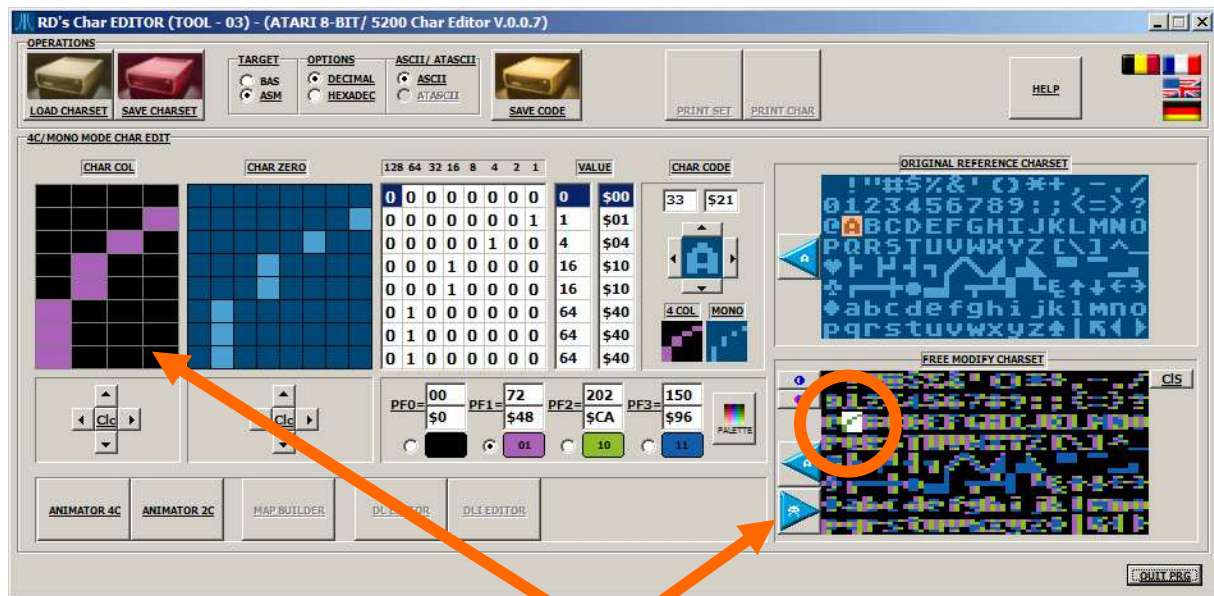
1.2.5. Fifth :How to change one of the 4 colors to another ?



Change PF1 color (41) to violet (72)



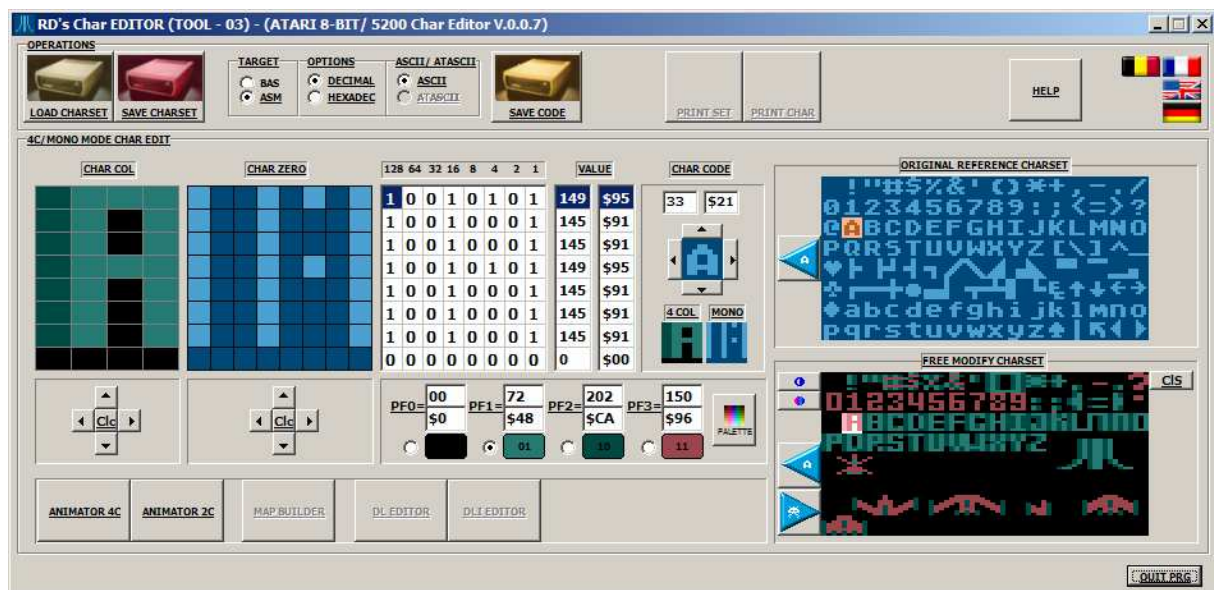
1.2.6. Sixth : Replace the modified char in the character set



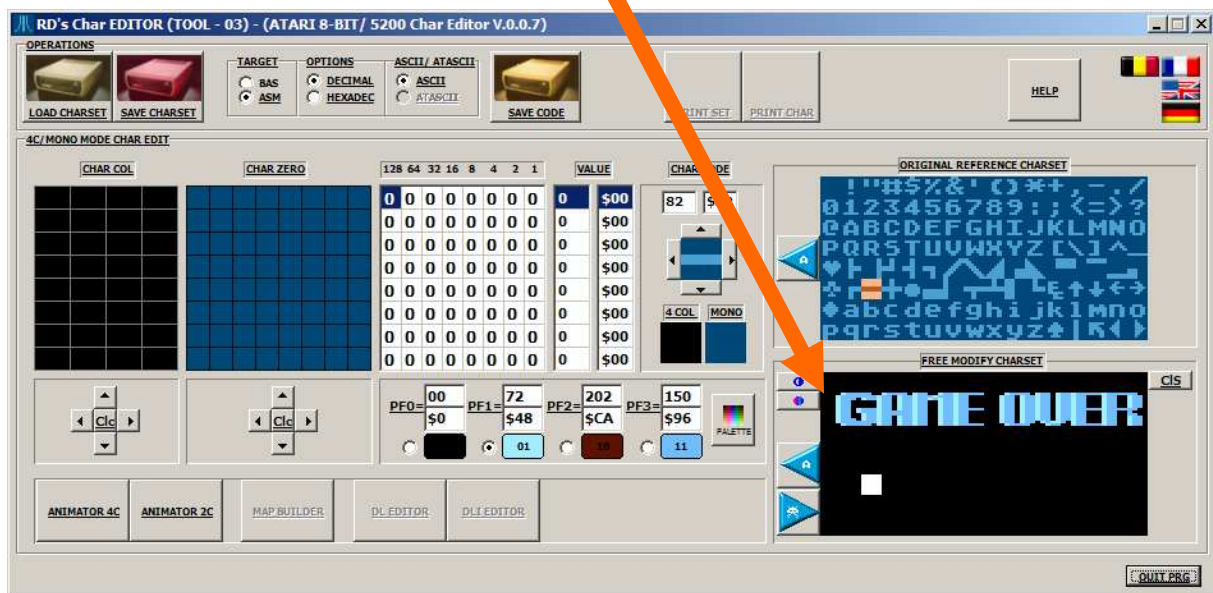
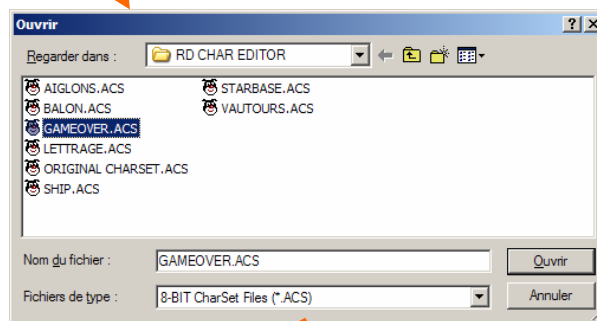
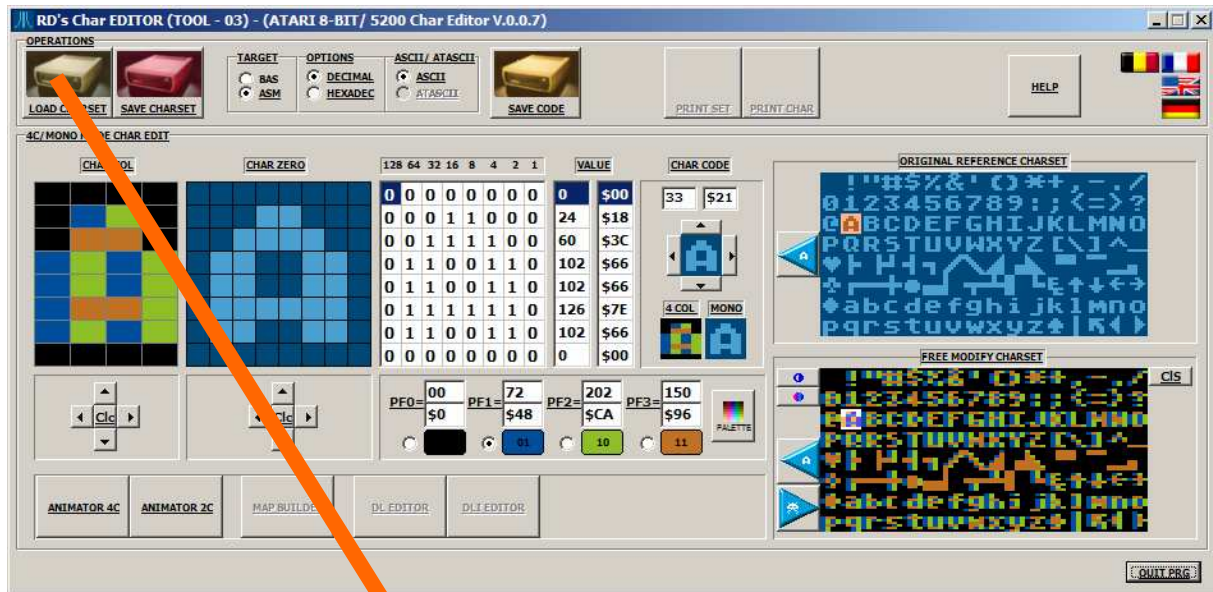
Once the char is finished in the edit grid, we must replace it in the charset with the little blue bolt icon (invader motif ;-)). Then, we can see the modification on the « free modify » charset. Repeat operation for all chars needed.

1.2.7. Example of a complete character set :

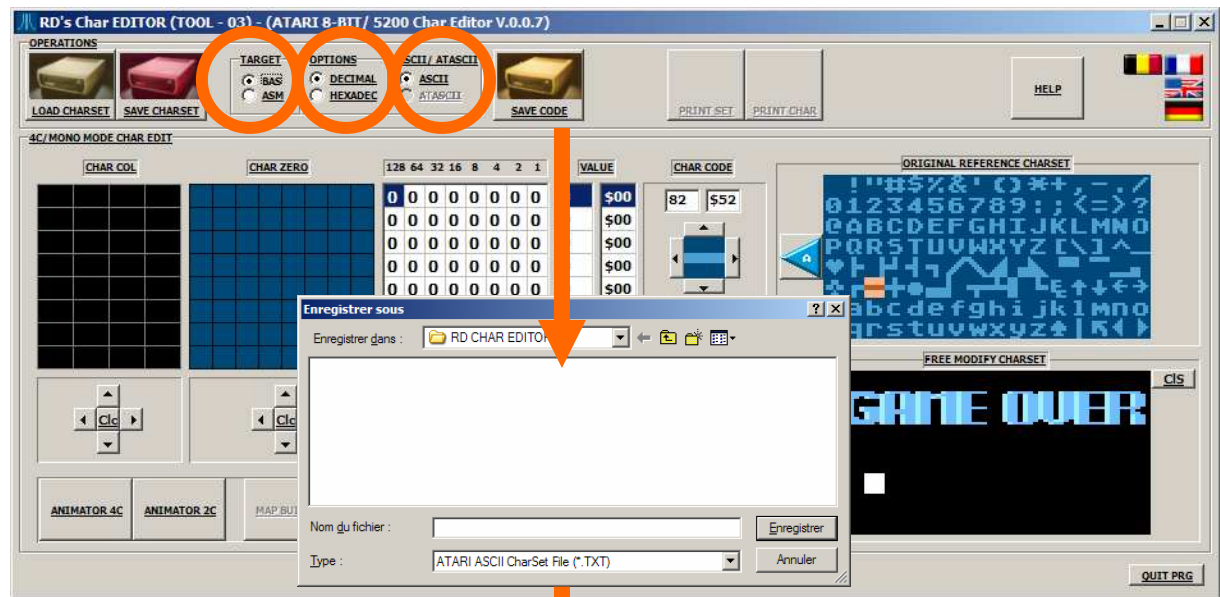
Here we have the example of a charset made with some « blank » chars. Don't forget that the ATARI 8-Bit can generate an interrupt witch provide the possibility to have a different charset on each line of screen in text-mode and many different color by line. A single pointer to change, a Display List to adapt.



1.2.8. Example of loading/ saving a charset :



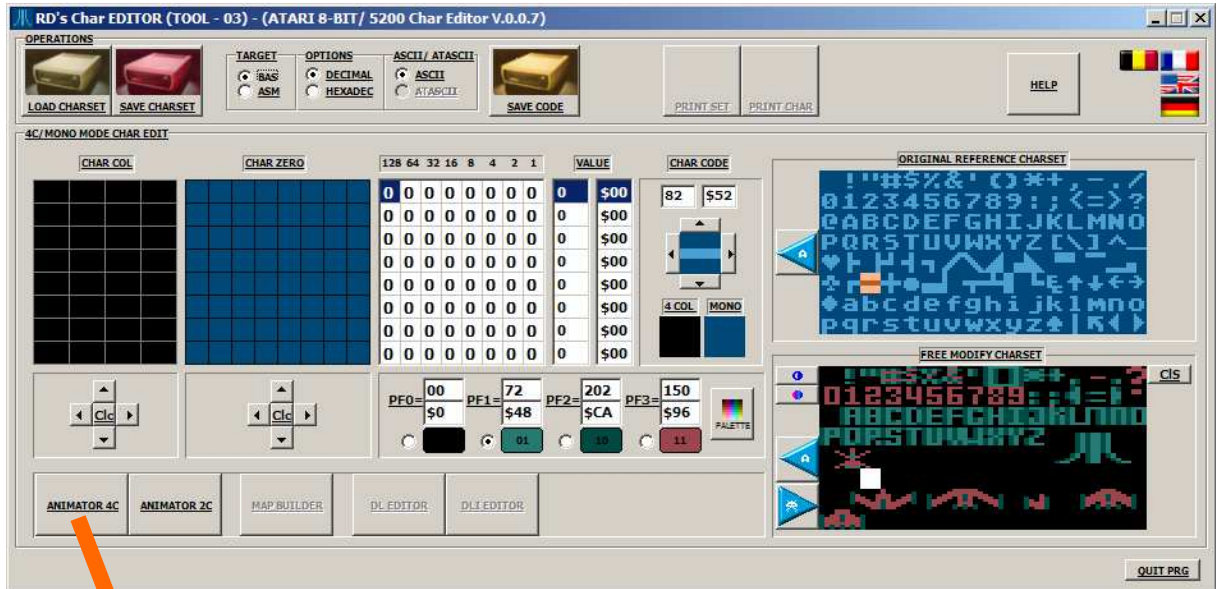
1.2.9. Example of saving charset on a CODE form (only for ATARI BASIC & ASM6502) :



1010 REM	10350 DATA	11120 DATA
1020 REM X:\XL-XE\RD PHOENIX PROJECT\RD CHAR	10360 DATA	11130 DATA
EDITOR\GAMEOVER.TXT.LST	10370 DATA	11140 DATA
1030 REM	10380 DATA	11150 DATA
1040 REM *****	10390 DATA	11160 DATA
1050 REM * REDEFINITION DE 128 CARACTERES *	10400 DATA	11170 DATA
1060 REM *****	10410 DATA	11180 DATA
1070 REM	10420 DATA	11190 DATA
1080 REM *****	10430 DATA	11200 DATA
1090 REM RESERVATION DE 1Kb (4 PAGES DE 256 BYTES)	10440 DATA	11210 DATA
1100 REM *****	10450 DATA	11220 DATA
1110 REM	10460 DATA	11230 DATA
1120 TOP=PEEK(106)-4	10470 DATA	11240 DATA
1130 POKE 106,TOP	10480 DATA	11250 DATA
1140 REM	10490 DATA	11260 DATA
1150 REM *****	10500 DATA	11270 DATA
1160 REM RECOPIE DU SET EN ROM VERS LA RAM	10510 DATA	
1170 REM *****	10520 DATA	
1180 REM	10530 DATA	
1190 ROM=PEEK(756)*256	10540 DATA	
1200 RAM=TOP*256	10550 DATA	
1210 REM	10560 DATA	
1220 FOR I=0 TO 1023	10570 DATA	
1230 POKE RAM+I,PEEK(ROM+I)	10580 DATA	
1240 NEXT I	10590 DATA	
1250 REM	10600 DATA	
1260 REM *****	10610 DATA	
1270 REM COPIE DU NOUVEAU SET VERS LA RAM	10620 DATA	
1280 REM *****	10630 DATA	
1290 REM	10640 DATA	
1300 ST=(TOP*256)	10650 DATA	
1310 FOR I=0 TO 1023	10660 DATA	
1320 READ A	10670 DATA	
1330 POKE ST+I,A	10680 DATA	
1340 NEXT I	10690 DATA	
1350 REM	10700 DATA	
1360 REM *****	10710 DATA	
1370 REM ACTIVATION DU NOUVEAU SET	10720 DATA	
1380 REM *****	10730 DATA	
1390 REM	10740 DATA	
1400 POKE 756,TOP	10750 DATA	
1410 REM	10760 DATA	
10000 DATA 000,000,000,000,000,000,000,000,000	10770 DATA	
10010 DATA 000,000,000,000,000,000,000,000,000	10780 DATA	
10020 DATA 000,000,000,000,000,000,000,000,000	10790 DATA	
10030 DATA 000,000,000,000,000,000,000,000,000	10800 DATA	
10040 DATA 000,000,000,000,000,000,000,000,000	10810 DATA	
10050 DATA 000,000,000,000,000,000,000,000,000	10820 DATA	
10060 DATA 000,000,000,000,000,000,000,000,000	10830 DATA	
10070 DATA 000,000,000,000,000,000,000,000,000	10840 DATA	
10080 DATA 000,000,000,000,000,000,000,000,000	10850 DATA	
10090 DATA 000,000,000,000,000,000,000,000,000	10860 DATA	
10100 DATA 000,000,000,000,000,000,000,000,000	10870 DATA	
10110 DATA 000,000,000,000,000,000,000,000,000	10880 DATA	
10120 DATA 000,000,000,000,000,000,000,000,000	10890 DATA	
10130 DATA 000,000,000,000,000,000,000,000,000	10900 DATA	
10140 DATA 000,000,000,000,000,000,000,000,000	10910 DATA	
10150 DATA 000,000,000,000,000,000,000,000,000	10920 DATA	
10160 DATA 000,001,001,001,001,001,001,001,001	10930 DATA	
10170 DATA 127,255,252,240,240,240,241,241	10940 DATA	
10180 DATA 240,253,125,001,001,001,253,253	10950 DATA	
10190 DATA 127,241,241,241,241,241,241,255	10960 DATA	
10200 DATA 199,247,247,247,247,247,247,247	10970 DATA	
10210 DATA 193,247,221,193,193,193,193,193	10980 DATA	
10220 DATA 247,247,247,247,247,247,247,247	10990 DATA	
10230 DATA 255,192,192,192,192,192,255,192	11000 DATA	
10240 DATA 000,001,001,001,001,001,001,001	11010 DATA	
10250 DATA 127,241,241,241,241,241,241,241	11020 DATA	
10260 DATA 199,247,247,247,247,247,247,247	11030 DATA	
10270 DATA 199,199,199,199,199,199,199,199	11040 DATA	
10280 DATA 223,223,223,223,223,223,223,223	11050 DATA	
10290 DATA 253,001,001,001,001,001,253,001	11060 DATA	
10300 DATA 255,240,240,240,240,240,255,240	11070 DATA	
10310 DATA 240,124,124,124,124,124,240,192	11080 DATA	
10320 DATA 001,001,001,001,000,000,000,000	11090 DATA	
10330 DATA 240,240,252,255,127,000,000,000	11100 DATA	
10340 DATA 125,125,125,253,249,000,000,000	11110 DATA	

2. Screen builder with a charater set

2.1. The screen builder



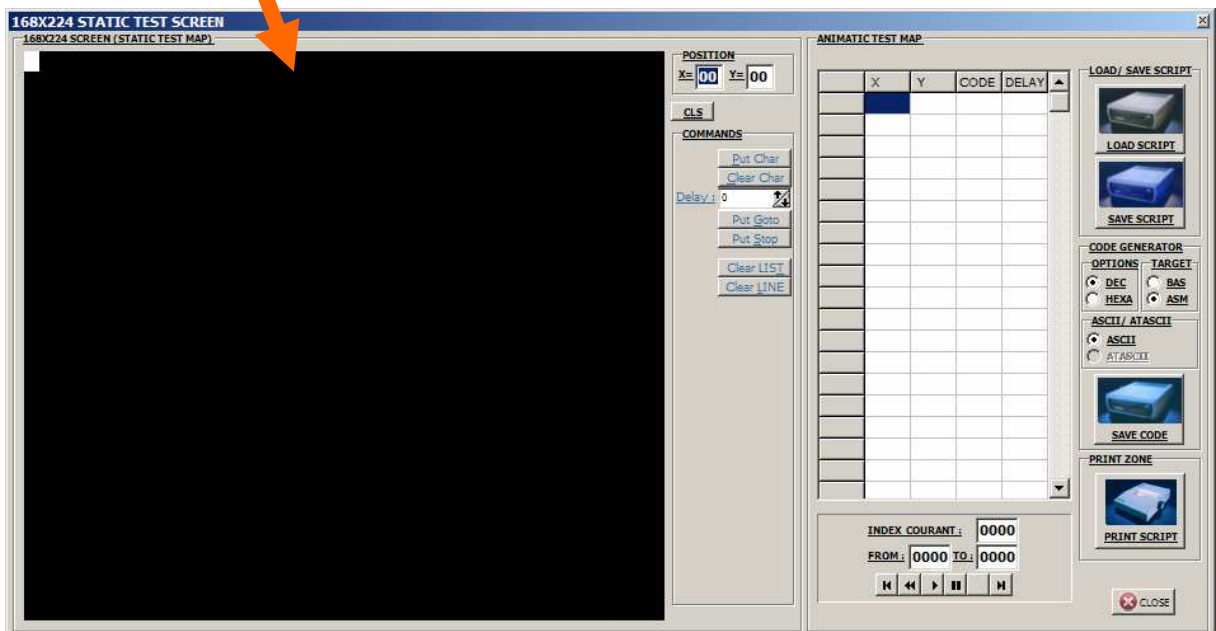
The screen builder/ animator is made with a screen-zone witch represent a 4-color text-screen (40X24 without borders) of an ATARI 800. Note that by DLI modify it's easily possible to extend theses numbers.

What can we do ?

With the charset from main screen, we can put chars on this screen-zone with the mouse and commands.

Some informations are available as X,Y char position. A data-grid witch shows where we put the chars and possibility to set a delay before the next operation.

Beyond just building a screen, there is a micro-language to execute automatic animations with 3 instructions : the 1st is put the char at X,Y, with code C and delay D (in milliseconds), the 2nd is a GOTO XXXX instruction where XXXX is the ligne number to jump and finally instruction STOP indicate the end of sequence. This was made for simulations before switching on real 8-bit machine.



2.2. Example of construction

