

File: CCVehShEdit-Manual.PDF

Format: PDF

First release: Sept. 16th, 2006

Last updated: June 13th, 2011

Author: Mafi; closecombat2@claranet.de; <http://closecombat2.fortunecity.com/>

Actual version: 1.01

Close Combat series of games

CCVehShEdit - The Manual

(a CC-vehicle shadow editor
for PC- & Mac)



Credits and all my thanks to the following programming gods:

Gerry Shaw (aka TinTin) for his file format descriptions posted on the Internet since 1998. And to Cpl_Filth for all his great tools for PC, first of all "SprTool.exe", which taught me how to scan the CC2 sprite files for valid datas.

Special thanks for beta-testing to Francisco Arias "Nembo"!

The CCVehShEdit sprite and texture file editor is not endorsed or promoted by the original game manufacturers. You will use this program entirely at your own risk. I recommend that you work only on backups of your datas and that you save your work often!

Installation and System Requirements

To install CCVehShEdit just copy the program file to your HD.

PC version of CCVehShEdit: should work under W2K, WinXP and W7. No W98-support anymore since v1.01.

MacOS-68k version of CCVehShEdit is a carbonized application and should work under MacOS 8.5 – 9.2.2 (requires CarbonLib 1.6 or newer) and MacOS-X 10.2.8 or newer (PPC processor and Classic environment needed). **You must ensure to give CCVehShEdit enough (more than 400 MB) RAM in MacOS Classic's Finder! Otherwise a system breakdown with loss of data might occur!**

MacOS-X-PPC version of CCVehShEditX is a carbonized application and should work under MacOS-X 10.2.8 up to 10.6.7 (PPC processor or Rosetta environment needed) and under MacOS 8.5 – 9.2.2 (requires CarbonLib 1.6 or newer).

MacOS-X-i386 version of CCVehShEditX is an Intel-processor-only application and should work under MacOS-X 10.4.11 or newer (Tiger, Leo, SnowLeo and Lion).

1. Introduction

CCVehShEdit is intended to modify vehicle shadow datas of the CloseCombat series of computer games, ranging from CC2 to the latest CCM version. "Close Combat - A Bridge Too Far" (abbreviated CC2, ABTF, CC2-ABTF) was the second game of the CloseCombat-series created by Atomic and presented by Microsoft to the Mac- and PC-community in 1997. CC2 was the last game of this series for the MacOS. The series was then continued by SSI, UbiSoft and Destineer / MatrixGames for PCs only (up to day CC3, CC4, CC5, CCM, The Road to Baghdad released in January 2004, abbreviated: RtB, publicly published CloseCombat Marines 3.1 in late 2004, CCM v4 released to the USMC in summer 2005 and the CCMRAFReg released in Feb. 2006, CCMT, CoI, CC:WaR, CC:TLD and CC:LSA). The last release to the public was up to now (summer 2011) "CC: Last Stand Arnhem" in 2010.

All games of the Close Combat series store their vehicle shadow datas and vehicle graphical datas (= vehicle texture datas) in several files. File formats differ between CC2 and the later releases, also the file names and the storing location differs. First of all you must know what is stored in what kind of file:

- **CC2 vehicle graphical texture file:** CC2 stores the vehicle graphics in "txtf"-files. The file format is identical to CC2's map graphics file format except for two entries in the file header (that means Big Endian encoding). CC2 stores for every vehicle up to 4 different "txtf"-files (one file for hull zoomed view, turret zoomed view, hull normal view and turret normal view). Non-turreted vehicles are stored in 2 files (one for zoomed view and the other one for normal view). The header of these files does not contain any hotspot informations. CC2 requires naming of these files sequentially from "Txtr000" to "Txtr153".
- **CC2 vehicle shadow file:** the shadows are stored in "SPRI"-files (= sprite files encoded in Big Endian). Each vehicle needs two shadow sprite files: one for zoomed and the other one for normal view. These files are named "VehB####" (B = big) or "VehS####" (S = small) respectively, where '###' stands for three digits representing the class number of the team which is using this vehicle. This numbering is not identical to the one of the "txtf"-files. The team class number can be read in CC2's base file "Teams" (plain ASCII TAB-separated text file). CC2 vehicle shadow files contain sprite animations consisting of 32 pictures. If the vehicle is turreted, then there will be two sprite animations each of 32 pictures. There is only one exception from this rule: the 88-FlaK "rotates on base", showing the gun's legs always in the same position. Therefore the shadow sprite file of the 88-FlaK gun contains one picture only in the sprite animation of the "hull". The animation sequence is 32 entries long, pointing always to the same one and only "hull shadow" picture. The animation sequence of the gun contains 32 different graphics, which are not shown in the game, perhaps due to an internal CC2-bug.
- **CC3 tank texture file:** CC3 is a totally PC-born game, introducing a new file format for the graphical vehicle textures encoded in Little Endian. The tank texture files are storing now 17 graphics and coordinates of 2 hotspots. The first picture is showing the vehicle hull or turret in the usual bird eye's view, the following 16 pictures are showing the vehicle texture rolling to the sides (rolling slightly and rolling drastically). The meaning of the two hotspots for each picture is not totally revealed yet.
CC3 stores the tank textures in 8 files if it is a turreted vehicle and in 4 files if the vehicle is turretless. The file name is not longer than 8 chars followed by a DOS-like filename extension. The different files contain an identification header ranging from 0 to 7 for
 - 0 = *.NNH (normal view, normal camo, hull),
 - 1 = *.NNT (normal view, normal camo, turret),
 - 2 = *.NSH (normal view, special (summer ?) camo, hull)
 - 3 = *.NST (normal view, special (summer ?) camo, turret)

4 = *.NNH (zoomed view, normal camo, hull),

5 = *.NNT (zoomed view, normal camo, turret),

6 = *.NSH (zoomed view, special (summer ?) camo, hull)

7 = *.NST (zoomed view, special (summer ?) camo, turret)

All these files are bundled together into the ZFX-archive "Tanks.zfx". File format infos on "ZFX" archives are following below.

- **CC3 gun texture file:** because a gun will not move around in CC3, the rolling effects are not needed. Therefore for each gun only one graphic is stored in a CC3-"txtf"-files, encoded in Little Endian and containing coordinates of one hotspot. The file format is similiar to the "txtf"-file format of CC2 except for these hotspot coordinates after the width/height entries in the header. File name rules are the same as for tanks. CC3's gun texture files are bundled together in the "Guns.zfx" archive.
- **CC3 vehicle shadow file:** like in CC2 the shadows are stored in "SPRI"-files (= sprite files encoded in Big Endian, although it is a PC-born game). Again each vehicle needs two shadow sprite files: one for zoomed and the other one for normal view. These files are named identical to the corresponding vehicle, followed by a filename extension "*.NSD" (normal view shadow) or "*.ZSD" (zoomed view shadow) respectively. CC3 vehicle shadow files contain only one sprite animation for the hull consisting of 64 pictures. No seperate animation for the turret's gun anymore. CC3 vehicle shadow files are bundled together in the archive "Shadows.zfx".
- **CC4 vehicle texture file:** CC4 and all newer games are using the CC3 tank texture file format for tank and gun texture graphics, giving them all the filename extension "*.TEX". Again 17 pictures in each TEX-file. But there are no longer zoomed views of the vehicles. Guns can move, therefore the rolling effect graphics are included for them. The identification of hull or turret graphics or special camo graphics can be done by parts of the Windows-like file name (including extension up to 256 characters long). CC4 and CC5 use the character strings "hull" and "turr" inside the filename to indicate that. The TEX-files are bundled together in AZP-archives "Tanks.azp" and "Guns.azp". File format infos on "AZP" archives are following below.
- **CC4 vehicle shadow file:** the shadows are stored in "IRPS"-files (= sprite files encoded in Little Endian). Now each vehicle needs only one shadow sprite files: for normal view only. These files are named identical to a corresponding vehicle, followed by a filename extension "*.SPR" (sprite). CC4 vehicle shadow files contain only one sprite animation for the hull consisting of 64 pictures. No seperate animation for the turret's gun anymore. It is now possible that different vehicles are sharing common shadow sprite files. CC4 vehicle shadow files are bundled together in the archive "Shadows.azp".
- **Newer CC versions** should follow the CC4 formats and rules.

2. How to use the program

After launching the program you will see a tool window on the left side of your computer screen. Here you can select different graphical editing tools like in other graphics editors. The main difference is that CCVehShEdit will avoid anti-aliasing. The vehicle shadow files are storing solely informations of shadow ON/OFF (= 1 / 0). Although they can contain solid colors as well, they normally did not! So the tool functionality of CCVehShEdit is concentrated around the sole purpose to paint/draw shadow or no-shadow (it can do more, of course, but you will not often use it ???). The drawing canvas does contain the actual shadow graphics, depending on the rotation angle you have set. You will see the hull's and/or turret's texture graphic placed over the drawing canvas as an uneditable layer (rotated correctly to the rotation angle you have selected). During painting/drawing

you will see where you paint, and after the painting action the hull's and/or turret's texture graphic layer will be restored. You can toggle the visibility of the hull's and/or turret's texture graphic layer as well as you can toggle the visibility of the hotpoint marking indicator (a little red cross) and the visibility of the shadow graphic itself.

Cpl_Filth's tool "SprTool.exe" introduced the light green color representation of the shadow encoding inside the sprite files. I followed this way again with CCVehShEdit. White will be "no-shadow" = "transparency". For other purposes there are two more special encodings for sprite files possible, but these are not used by original shadow sprite files (I only found the "FillArea" encoding in CCM's modern tank shadows).

To make the changing of the vehicle texture graphics possible, you can export and import graphics via the clipboard. Use an external graphics editor for painting them. CCVehShEdit will only allow you "painting" shadow (a necessary note for those who believe that my tools are no real editors).

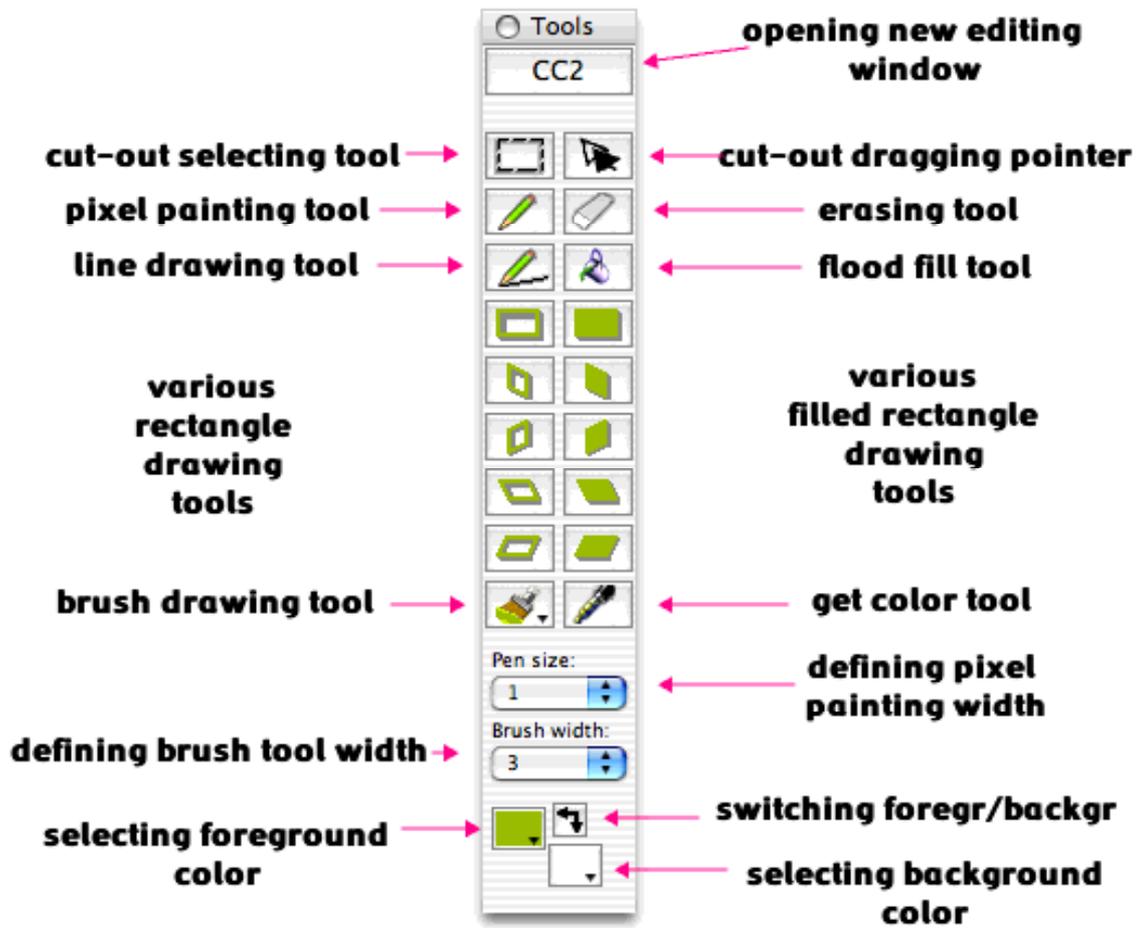
But CCVehShEdit will allow you to edit all hotpoint entries of vehicle texture files. So I think it is a sufficient way for modders to patch nearly all the vehicle related graphics. Nearly? Yes, wreck files must be modified using other tools (CC2Tools, RtBTool). A real editor can create files from scratch. Okay, CCVehShEdit will not do this! It is only capable to edit existing files. This will avoid file format conversion troubles. Duplicate the original file you want to edit and go on modding.

Once the program is launched and you have seen the Tools window on the left, the Win-user will see a multiple documents interface with a menu bar. Mac-users will have access to their familiar menu bar as usual.

The Tools window

The available tools are nothing spectacular. What you must know is that settings here will have effect to all editing windows. The behavior of the cut-out handling might be something strange: you must use the cut-out selecting tool to define a cut-out rectangle. Clicking with the mouse outside or inside of the cut-out rectangle will deselect the cut-out definition. If you want to drag the cut-out around you must use the cut-out dragging pointer. You cannot drag the cut-out outside the editing canvas. The brush drawing tool is a specialized version of the pixel painting tool with a selectable shape. It is the tool I had used most.

The selecting of the fore- and background color lets you select all the four special colors (some CCM vehicle shadows are using the "FillArea" color for unknown reason) as well as any opaque color. I recommend to use only the "Transparency" and "Shadow" representing colors for vehicle shadow files.



Common menubar commands

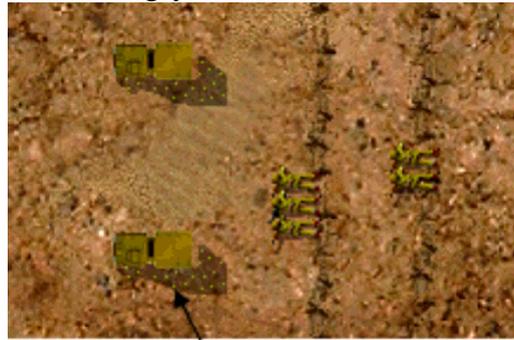
There are two ways of using this program: I will now describe the way for the main part of the CC community. The specials for the CC2 fans will follow soon.

The menubar will give you access to the following operations (from left to right):

- **File>New editing window:** use this command to make an editing window visible (you can have more than one).
- **File>Load single CC2 Txtr/shadow combination:** select this command to load into the frontmost editing window vehicle texture and shadow graphics. You will be asked if you want to load a turret graphics, too. The graphics loaded will be treated as "normal view" graphics even if you have loaded zoomed view graphics. You must ensure to load the correct texture/shadow combination.
- **File>Load single CC3 NNH/NSD/ZSD combination:** select this command to load CC3 vehicle files. Select the *.nnh (hull) file first. CCVehShEdit will automatically load the corresponding turret file if it is found in the same folder where the hull file is located. That means when you are selecting a *.nsh (hull, summer camo) file, the program will look for a *.nst file instead. If no turret file can be found, you will be asked if you want to search for one. The corresponding zoomed vehicle texture files will be also loaded if they are present inside the same folder. Then you will be asked for the shadow sprite file *.nsd. The corresponding *.zsd file will be loaded automatically if it is present inside the

- same folder. If you accidentally select the *.zsd file first, the program will look for the corresponding *.nsd file.
- **File>Load single CC4/5/M/6 TEX/SPR combination:** select this command to load CC4 or newer vehicle files. Select the TEX file of the hull first. If the corresponding turret file can be detected automatically, then it will be loaded, otherwise you will be asked for it. Then you will be asked for the SPR shadow file.
 - **File>Save:** saving the contents of the frontmost editing window. The program will remember the absolute paths of all loaded files unless you are using the "Save as..." command. That means if you are using the program on original files and then issuing the "Save" command the original datas will be overwritten. Please make backups of your files prior to use this program!
 - **File>Save as:** you will be asked where to store the files of the frontmost editing window.
 - **File>Export...:** you can export the actual visible shadow, hull and turret texture and a screenshot of the drawing canvas into a 16-bit TARGA graphic file.
 - **File>Close:** closing the frontmost window. If it is an editing window with unsaved datas present you will be asked if you want to save them prior to closing the window.
 - **File>Preferences:** go there to make some settings. Mainly to stop the program from showing the CC2 vehicle team table on startup (disable "prefer CC2 editing"). The program's preferences will be stored in a file named "CCVehShEdit.ini" in a special hidden folder. The location of this INI file depends on your operating system and can be controlled in the "About" section of the "Preferences window".
 - **File>Quit:** if unsaved work is present, you will be asked if you want to save it. For Windows-version: pressing the close button in the title bar of the multiple documents interface window will also quit the program.
 - **Edit>Undo:** all graphical operations and all transforming actions can be made undone. There is a 10-step deep undo circle for graphical operations, which will be interrupted when a transforming action took place.
 - **Edit>Cut:** requires to select some graphical area (= cut-out) of the actual shadow graphic. This area will be deleted, copied to the clipboard and replaced by the actual background color.
 - **Edit>Copy:** requires to select some graphical area of the actual shadow graphic. This area will be copied to the clipboard.
 - **Edit>Paste:** if there is a graphical contents present at the clipboard, then it will be pasted into the actual shadow graphic.
 - **Edit>Trim:** requires to select some graphical area of the actual shadow graphic. The actual shadow graphic will be resized and trimmed to the size and contents of this area.
 - **Edit>Clear:** requires to select some graphical area of the actual shadow graphic. This area will be deleted and replaced by the actual background color.
 - **Edit>Select all:** the entire actual shadow graphic will be selected as cut-out.
 - **Edit>Unselect:** the actual cut-out selection will be removed, pasting in the cut-out at it's present place.
 - **Edit>Edit hull texture>Copy hull:** copies the entire graphic of the actual hull texture to the clipboard.
 - **Edit>Edit hull texture>Paste hull:** if there is a graphical contents present at the clipboard, then it will be pasted in as the actual hull graphic, including changing of the hull graphic's size.
 - **Edit>Edit hull texture>Paste to all hull textures:** intended for CC3-or-newer graphical textures consisting of 17 pictures: if there is a graphical contents present at the clipboard, then it will be pasted in for all the 17 hull graphics, including changing of the hull graphics' size.
 - **Edit>Edit turret texture:** same operations as above but for the turret textures.

- **Copy screenshot:** gives you the ability to copy a screenshot of the actual editing canvas (respecting zoom scale and hotpoint visibility) to the clipboard.
- **View>Information:** will show up a window where the characteristics and filenames will be shown of the files loaded by the frontmost editing window.
- **View>View at 100%:** set zoom scale to 100%.
- **View>Zoom in / View>Zoom out:** change the zoom scale of your editing canvas (max. is 800%).
- **View>Set rotation and rolling to 0:** will set rotation to north-up and rolling to bird eye's view for the frontmost editing window.
- **View>Take this window's position as default:** will set the frontmost editing window's coordinates and width/height as default position for the first editing window after startup. These values will be saved in the INI file.
- **Actions>Convert all :** for the frontmost editing window: depending on the format of your shadow files (CC2, CC3, CC4-or-newer) you can convert them into another CC-format. This action converts both normal view and zoomed view shadows if needed by the target format and available by the source format.
- **Actions>Set all solid colors to shadow:** for the frontmost editing window: depending on which view is selected (normal view or zoomed view) all shadow graphics will lose their (accidentally) solid colors, getting them converted into shadow pixels. This action is necessary because the program's internal drawing operating might cause solid pixels under certain screen resolutions on certain computers (example: Macintosh). If you encounter graphical effects like below when testing your vehicle, then use this action.



irregular solid colors

Solid colors inside the vehicle's shadow:

- **Actions>Clear all shadows:** for the frontmost editing window: depending on which view is selected (normal view or zoomed view) all shadow graphics will be filled with the actual transparency color (that means that this command does not use the selected background color but always the transparency color. The transparency color is usually represented by "white" and can be set in the Preferences window).
- **Actions>Enlarge all ... shadows to same size:** for the frontmost editing window: depending on which view is selected (normal view or zoomed view) all shadow graphics will be expanded to identical width and height and common hotpoint (the expansion will be done by adding pixel columns/lines on the pictures' sides).
- **Actions>Minimize all ... shadows:** for the frontmost editing window: depending on which view is selected (normal view or zoomed view) all shadow graphics will be cut on their sides to minimize them. The result will be shadow graphics fitting best to original CC4-or-newer shadow sprite files without surrounding transparency space.
- **Actions>Resize shadow graphics:** for the frontmost editing window: you can select if you want to resize (adding/deleting pixel columns/lines) the actual shadow graphic only or the entire set of pictures of this view.
- **Actions>Shrink/blow-up shadow graphics:** for the frontmost editing window: you can select if you want to rescale (in %) the actual shadow graphic only or the entire set of

pictures of this view. To avoid anti-aliasing results, only few rescaling operations are possible.

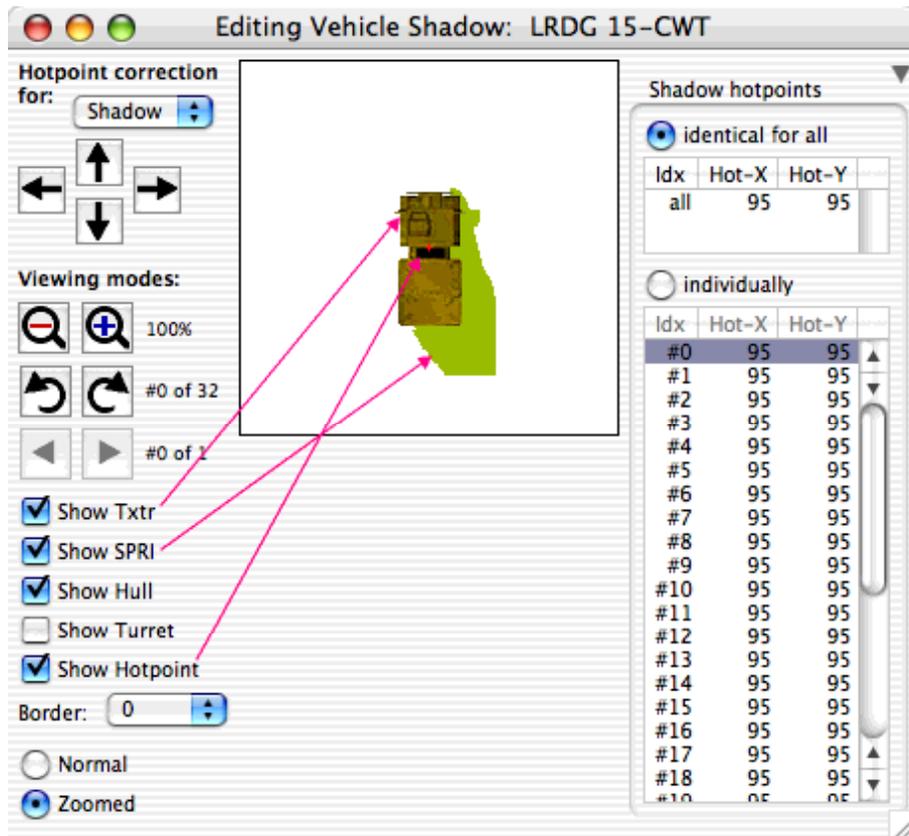
- **Actions>Rotate/flip shadow graphics:** for the frontmost editing window: you can select if you want to rotate and/or flip the actual shadow graphic only or the entire set of pictures of this view.
- **Archives:** the upper four commands can be used to extract the contents out of AZP and ZFX archives or to repack them. Repacking of AZP archives will be done by taking all files out of a selected folder in their alphabetically order. ZFX archives will be repacked with files in their "natural" order. There are also two converting commands to translate CC3 shadow sprite files to CC4/5/M/RtB format and back without loading them into an Editing window. A similiar conversion for NNH/NNT - TEX files is not necessary, because their file format is identical from CC3 throughout the latest CC release (exception: CC3 guns).
- **Windows:** will give you access to make the tools window and the two special CC2 windows visible in case you have closed them.

The Editing window

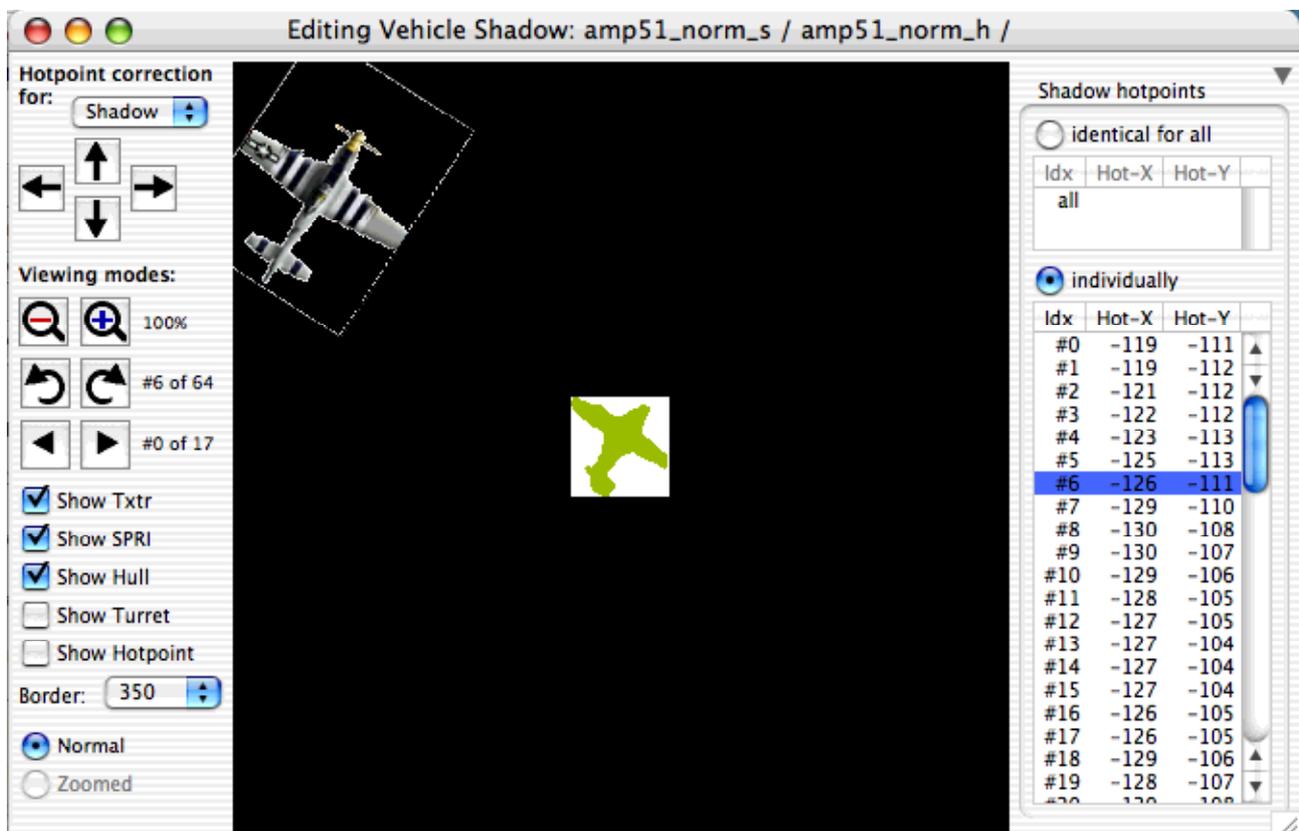
The Editing window is divided into 4 sections. From left to right:

- **Viewing tools:**
 - Hotpoint correction arrows: there is a popupmenu above these arrows to select which hotpoint you want to move around. The resulting numerical values will be entered automatically into the corresponding lists of this editing window. In case of a completely loaded CC2 installation texture hotpoint corrections will be also written to the "CC2 vehicle hotpoint table" window. If you select the radio button "identical for all" of the corresponding listbox, then the hotpoint correction will have effect for all those hotpoints
 - Viewing modes: zoom in/out, rotating, rolling.
 - Checkboxes for selecting what you want to see: if shadow is disabled, then the background will be black. For CC2 only: the shadow graphics of the turret's gun will be only visible if "Hull" is deselected and "Turret" and "SPRI" are selected!
 - Border: a black painted border around the shadow graphic. Select here the width of this border. You will need this for CC4-or-newer or other minimized shadows (or textures with negative hotpoint coordinates) to make the texture graphic fully visible. It is a workaround, a compromise...
 - Select zoomed or normal view for CC2 and CC3 only.
- **Editing canvas:** select the editing tool from the Tools window and paint/erase your shadow. The mouse cursor will show a custom shape until you press the mouse button.
- **Shadow hotpoint lists:** here you can edit the numerical values of the shadow graphics' hotpoints. If you select the radio button "identical for all" then you will have access only to the upper list. Entering values here will have effect for all pictures of the given view. If you select "individually", then you have access to the lower listbox where you can change the hotpoint entries individually. Warning: when entering values here the program will check for identical values. If you change the individual values in the way to get (accidentally) common values for all hotpoints, then the radio button "individually" will be disabled and the "identical for all button" will be enabled.
- **Hull and turret hotpoint lists:** for CC2 and CC3-guns only one hotpoint is listed here. All other CC vehicle texture graphics will show 17 entries. If you want to see the second hotpoint, scroll the lists horizontally to the left. Except for this, their behavior is identical

to the lists of the shadow hotspots. To make them visible, toggle the little symbol in the upper right corner of the Editing window.



Picture: an example of an Editing window loaded with a CC2 vehicle



Picture: an example of an Editing window loaded with a CC5 plane, showing the need of a large border.

CC2 specials

Another way to access the vehicle files is to load all vehicle related files into RAM. This is implemented for CC2 only. For this task you have additional menu bar commands and some commands work slightly differently:

- **File>Save:** the saving command will not only write the changes back to disk but will also store the new values in RAM. Same to say for the "Save as" command.
- **Edit>Select all:** let you select all entries to prepare copying their texts to the clipboard.
- **Edit>Copy:** will copy the selected text(s) to the clipboard.
- **File>Select the CC2 installation:** use this command to decide in which folder your CC2 installation resides. For accessing Windows-installations you must select a folder where the CC2 "Data" and "Graphics" folders are located. For accessing Macintosh-installations you must select the volume (CD or CD-image) where the folder "Data" is located (this folder must again contain the subfolders "Data" and "Graphics"). Usually this folder is located on a volume named "Close Combat".
- **File>Load all CC2 vehicle related files:** will load the entire vehicle stuff including the base files "Teams" and "Vehicles". The result will be presented in a separate window, the "CC2 Vehicle Teams" window. You can determine in the Preferences window if you want to have the hotpoint table from inside the application to be loaded automatically or not.
- **Windows>Show CC2 vehicle team list:** will bring this window to front. From this window you can select the desired vehicle by double-clicking on an entry to show it in a new Editing window.
- **Windows>Show CC2 vehicle hotpoint table:** will bring a window to front where the CC2 vehicle hotpoints are listed. This hotpoint table is located inside the executable CC2 game application ("CC2.exe" for Windows / "A Bridge Too Far" for MacOS). At startup CCVehShEdit contains the original default hotpoint table, so you must not load it each time again. But if you want to use a modified table, I recommend that you load this table from the patched EXE file / patched application file using the following commands:
- **File>Load CC2 hotpoint table from EXE/app:** this command and the following two commands are only available within the window "CC2 vehicle hotpoint table". Using the loading command you can load the hotpoint datas out of the CC2 EXE/application file. CCVehShEdit will automatically determine the version and operation system ID of the selected EXE/application file. CCVehShEdit can identify 6 different versions: PC demo, PC v1, PC v2, Mac demo, Mac v1 and Mac v2. If no such version is detected, CCVehShEdit will try to load datas from top of input file. CCVehShEdit will check for a valid hotpoint table by looking for the table's terminating character string "Ttxt".
- **File>Save CC2 hotpoint table to EXE/app:** will write the contents of the "CC2 vehicle hotpoint table" window back to the application file. This will patch a executable program file without warning. Please be sure that you have a backup copy of the original file before doing this.
- **File>Save CC2 hotpoint table as:** will write the contents of the "CC2 vehicle hotpoint table" window back to a file the user must select. If you select a non-existing file, then the table will be written to a new created file where the table is starting at top of file. The terminating "Ttxt" will also be written.

The CC2 hotpoints table itself let you edit the hotpoint coordinates only. Double click on the entry to make it editable. All other texts are for information only. Changed values will be displayed in bold and italics. On closing the window or at application quitting you will be asked if you want to save your changes.

Not implemented

The program will not change the animation sequence logic. Please respect this when editing the CC2 88-FlaK vehicle. You can change the animation sequence logic for CC2 vehicle shadows using CC2Sprinter. I don't recommend to change the sequence logic for shadow files of CC3-or-newer. CCVehShEdit can handle animation sequences of a length of 128 entries. Actual CC files are limited to a length of 64 entries. But no one knows what future will bring. Up today I don't know the real meaning of the two hotpoints of CC3/CC4 TEX file entries. That's the reason why the turret placement of the rolling graphics is wrong. But this has no effect for shadow designers because the turret gun has no separate shadow animation sequence.

3. Appendix

Hotpoints for CC2 "txtf" Files

In 2006 I located the hotpoints table inside the CC2.exe / ABTF application. To place the vehicle / gun graphics (of the texture files "Ttxtf####") correct over the shadow graphics coming from the "VehS####" / "VehB####" files CC2 must know where the center of these graphics is. The same to say for the relation between hull and turret: the turret will rotate around its center and this turret-center will be placed at the hull's center. The shadow graphics are stored inside sprite files. These sprite files contain not only the shadow graphics but also the x/y-coordinate of the graphic's center (where 0,0 is the upper left corner).

The shadow sprites contain usually 32 graphics (one graphic for the 32 possible directions) for the hull and if needed 32 graphics for the turret's gun. The shadow graphics of the 8.8-FlaK are showing how CC2 is really respecting the entries in these shadow sprites: CC2 defines the 8.8-gun's base as fixed, therefore it is only one single shadow graphics implemented, and it's animation sequence contains 32 times the reference number to this single graphic (in this case: number 0). For the barrel of the gun there are 32 different graphics implemented. All other shadow sprite files are organized as described before (32 hull + 32 turret's gun graphics).

In CC3 or newer the TXTF-files contain hotpoint-coordinates, too. But in CC2 the hotpoints are hard coded inside the executable application (ABTF for Mac / CC2.exe for PC). These are the offsets of the table containing the hotpoints:

version	Mac	PC
demo	Offset = 150EBAhex	Offset = 147A90hex
first version after install from CD	Offset = 155A7Bhex	Offset = 147F40hex
last update from the MS-Net (Microsoft Support Center)	Offset = 15663Bhex	Offset = 148950hex
Format of the table	174 entries 3 bytes per entry each value 1 Byte seperating zero byte after each entry (exceptions below!)	174 entries 4 bytes per entry each value ShortInt (2 bytes) byte format LITTLE ENDIAN Hot-X followed by Hot-Y

	Hot-X followed by Hot-Y	
Table's size	522 bytes	696 bytes

The format of the table of the PC version is simple: for every "Ttr###" file there are two Short-Int (2 bytes) values (Hot-X and Hot-Y) stored. The format of the Mac version is compressed and therefore much more complicated: each entry is three bytes long. The first byte contains the Hot-X value (exception: first entry: here you must subtract 26 from the Hot-X value to get the correct value). The second byte contains the Hot-Y value (exception: last entry, here you must take the third byte to get the Hot-Y value). The third byte is zero (exception: last entry, here the third byte contains the Hot-Y value).

For all versions and for all executables the table is followed by the string sequence "Ttr%" which helped me to identify the structure.

The original datas of this table:

File	Hot-X	Hot-Y
Txtr000	19	44
Txtr001	14	44
Txtr002	10	22
Txtr003	7	22
Txtr004	22	41
Txtr005	14	54
Txtr006	10	21
Txtr007	7	27
Txtr008	22	55
Txtr009	11	28
Txtr010	38	37
Txtr011	18	54
Txtr012	19	19
Txtr013	9	27
Txtr014	22	36
Txtr015	13	49
Txtr016	11	18
Txtr017	7	25
Txtr018	16	45
Txtr019	8	23
Txtr020	20	12
Txtr021	10	6
Txtr022	20	41
Txtr023	16	60
Txtr024	10	21
Txtr025	8	30
Txtr026	18	31
Txtr027	11	28
Txtr028	9	16
Txtr029	6	14
Txtr030	25	51
Txtr031	19	78
Txtr032	12	26
Txtr033	9	40
Txtr034	23	41
Txtr035	15	79
Txtr036	12	21
Txtr037	8	40
Txtr038	20	41
Txtr039	16	52
Txtr040	10	21
Txtr041	8	26
Txtr042	29	49
Txtr043	15	25
Txtr044	21	36
Txtr045	11	18
Txtr046	29	49
Txtr047	15	25
Txtr048	22	37
Txtr049	11	19
Txtr050	22	40
Txtr051	11	20
Txtr052	23	44

File	Hot-X	Hot-Y
Txtr053	17	58
Txtr054	12	22
Txtr055	9	29
Txtr056	23	44
Txtr057	17	72
Txtr058	12	22
Txtr059	9	36
Txtr060	15	28
Txtr061	12	26
Txtr062	8	14
Txtr063	6	12
Txtr064	25	79
Txtr065	12	40
Txtr066	16	34
Txtr067	8	17
Txtr068	25	51
Txtr069	19	78
Txtr070	12	26
Txtr071	9	40
Txtr072	16	39
Txtr073	8	20
Txtr074	29	49
Txtr075	15	25
Txtr076	27	63
Txtr077	14	32
Txtr078	16	34
Txtr079	8	17
Txtr080	15	34
Txtr081	8	17
Txtr082	16	34
Txtr083	8	17
Txtr084	20	41
Txtr085	16	69
Txtr086	10	21
Txtr087	8	35
Txtr088	21	43
Txtr089	16	52
Txtr090	11	22
Txtr091	8	26
Txtr092	16	27
Txtr093	11	28
Txtr094	8	14
Txtr095	6	14
Txtr096	18	39
Txtr097	9	20
Txtr098	16	39
Txtr099	8	20
Txtr100	26	49
Txtr101	15	27
Txtr102	13	25
Txtr103	8	14
Txtr104	26	49
Txtr105	16	41

File	Hot-X	Hot-Y
Txtr106	13	25
Txtr107	8	21
Txtr108	15	34
Txtr109	8	17
Txtr110	15	44
Txtr111	8	22
Txtr112	16	44
Txtr113	8	22
Txtr114	16	44
Txtr115	8	22
Txtr116	21	44
Txtr117	11	22
Txtr118	26	50
Txtr119	18	92
Txtr120	13	25
Txtr121	9	46
Txtr122	29	94
Txtr123	15	47
Txtr124	23	78
Txtr125	12	39
Txtr126	20	60
Txtr127	9	30
Txtr128	17	45
Txtr129	9	23
Txtr130	29	61
Txtr131	15	31
Txtr132	15	44

File	Hot-X	Hot-Y	
Txtr133	8	22	
Txtr134	17	34	
Txtr135	12	25	
Txtr136	7	17	
Txtr137	6	12	
Txtr138	17	34	
Txtr139	14	49	
Txtr140	8	17	
Txtr141	7	25	
Txtr142	22	37	
Txtr143	11	19	
Txtr144	27	63	
Txtr145	14	32	
Txtr146	22	37	
Txtr147	11	19	
Txtr148	29	49	
Txtr149	15	25	
Txtr150	22	41	
Txtr151	14	54	
Txtr152	10	21	
Txtr153	7	27	
Txtr154 – Txtr173	15	131	Flamethrower animation

File formats

For an in-depth description of all sprite files please read my guide "CC2Guide-SpriteFiles-v6.pdf". For analyzing them or for editing the sprite animation entries please use my tool "CC2Sprinter v2.8" or newer (available for Mac & PC). CC4/5/M/RtB AZP and TEX files can be edited also with my "RtBTool v2.1" or newer. And CC3's and CC2's "txtf"-files can be converted into TGA files and back by using my tool "CC2Tools v4.12".

ZFX file format (CC3)

This file format is used by all the *.zfx files of CC3.

Whole file is encoded in Little Endian (PC/Intel like). First description of this file format was made by Gerry Shaw aka "Tin Tin", he wrote: "... the *.zfx file contain a bunch of other files concatenated into a single file. This is done to decrease seek times when reading files from disk or CD." With the help of Zonbie I can give here a complete description of the directory:

Intel byte-style, Little Endian

```
// header 4 bytes
long   NumberOfEntries    // 4 bytes containing number of files in this ZFX-archive

// directory with 600 file definition entries, fixed size
// 1.: file name table with 600 entries, fixed size of 7800 bytes
// each entry 13 bytes ASCIIZ: 8.3 DOS like filenames (up to 8 chars, dot, 3 chars extension)
//          terminated by a zero byte.
array(600) of char(13)    // ASCIIZ, padded with zeros to a fixed length of 13 bytes.
// 2.: file offset table with 600 entries, fixed size of 2400 bytes
array(600) of long       // 4 bytes per entry, offsets counted from top of file.
//3.: file size table with 600 entries, fixed size of 2400 bytes
array(600) of long       // 4 bytes per entry.

// file datas to follow, concatenated without gap
data
```

The original ZFX archives are filled with garbage datas in the higher unused areas of the file offset and file size table.

AZP file format (CC4 and newer)

Used by the files Tanks.azp, Guns.azp and Shadows.azp

Intel byte-style, Little Endian

```
// header 4 bytes
long   NumberOfEntries    // 4 bytes

// directory
// each entry is 264 bytes long
char(256)  filename      // 256 bytes, windows-style filename, padded with zero bytes
long   Offset // where the file will start, offset from top of file
long   Length// length of file (in bytes)

// entries
data
```

TEX file format

CC4/CC5/CCM/RtB files: *.tex inside the files 'Tanks.azp' and 'Guns.azp' identical with CC3's *.nnh/*.nnt-files

Actualized documentation of Sept. 10th, 2006. Contents of these *.tex files: 17 small graphical textures for vehicles, guns and airplanes.

Byte order: Little Endian (Intel style)

```
// header, 8 bytes
long           // 4 bytes headerID, usually 00000000h, but also possible 00000001h - 00000003h
               // in CC3: ranging from 00000000h - 00000007h
long           // size of image data area behind the last directory
// because the directories are of fixed size, you can calculate:
// 'file size' = 484 + 'image data area size'

// directory #1, containing hotspot entries, 1 hotspot per image, probably the rotation-center:
for entry = 0 to 16
    long        // hotspot-value #0 (x)
    long        // hotspot-value #1 (y)
next

// directory #2, containing hotspot entries, again 1 hotspot per image,
// probably the turret-fixation (?):
for entry = 0 to 16
    long        // hotspot-value #2 (x)
    long        // hotspot-value #3 (y)
next

// directory #3, containing width and height for every image:
for entry = 0 to 16
```

```

        long          // width
        long          // height
next

// directory #4, containing image data offsets (counted from top of image data area)
for entry = 0 to 16
        long          // offset; first offset is always 00000000h
next

// the graphical datas will follow immediately
data (16-bit encoded RGB-graphics)

```

"txtf" files with hotpoint coordinates (CC3 guns/wrecks) identical with CC4/CC5/CCM/RtB special txtf-files inside the file 'FT.azp'

Their header is 8 bytes longer than a regular CC2-txtf . They contain one picture.

Byte order: Little Endian (Intel style)

```

// header, 24 bytes
char(4)      // 4 bytes headerID, always 'txtf'
short        // 2 bytes, unknown, always 0000h
short        // 2 bytes, versionID, always 0002h
long         // 4 bytes, image width
long         // 4 bytes, image height
long         // 4 bytes, hotpoint-X
long         // 4 bytes, hotpoint-Y

// the graphical datas will follow immediately (one image only)
data (16-bit encoded RGB-graphics)

```

"txtf" files of CC2

They contain one picture. Their header differ from a CC2 map graphics file in having a special header-ID ("txtf") and a different version-ID (1 instead of 2).

Byte order: Big Endian (Macintosh/Motorola style)

```

// header, 16 bytes
char(4)      // 4 bytes headerID, always 'txtf'
short        // 2 bytes, unknown, always 0000h
short        // 2 bytes, versionID, always 0001h
long         // 4 bytes, image width
long         // 4 bytes, image height

// the graphical datas will follow immediately (one image only)
data (16-bit encoded RGB-graphics)

```

Some actual forums for CC-users:

CloseCombat HQ: <http://www.ryanross.net/cc/>

CSO: <http://www.closecombat.org/>

CCS: <http://www.closecombatseries.net/>

MatrixGames: <http://www.matrixgames.com/forums/>

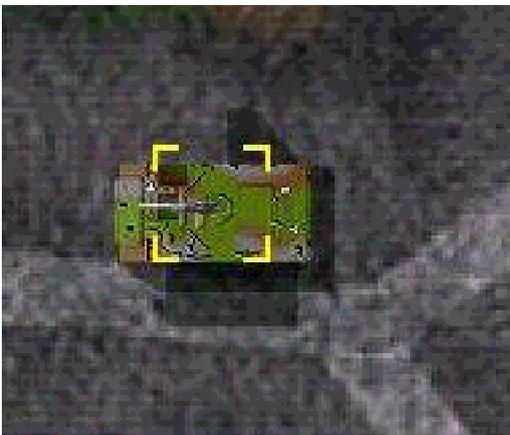
Vehicle shadow problems 2009

In 2009 NEMBO located a vehicle shadow modding related problem and it's solution. He gave me the permission to describe it here together with his screenshots:

Problem: turreted vehicle, the vehicle has both a turret and a hull shadow defined in it's VehB### and VehS### files. Under certain circumstances you will see this vehicle with turret to use the hull-shadow for both the turret and the hull.

Reason: the teams-slot is not prepared for storing a turreted vehicle.

Solution: Nembo found that the vehicle must be turned into a turreted vehicle by modding the base file "Vehicles" this way: the turret weapon type was set to fixed and the rotation value to 0. Changing these values fixed the problem.



Error: doubled hull shadow



Solved: turret has own shadow.

MAFI

closecombat2@claranet.de

<http://closecombat2.fortunecity.com/>

<http://cc2revival.npage.de/>

<http://www.ftf.claranet.de/>

<http://www.afrika.claranet.de/>

<http://www.closecombat2.claranet.de/>

<http://www.dieppe.claranet.de/>

<http://www.cc2.claranet.de/>

<http://www.mappa.claranet.de/>

<http://www.geocities.com/cc2revival/>