Multipaint

Multipaint 2018b (1.8.2018, 5th release, 18.10.2018 fixed version)

Multipaint allows you to draw pictures with the color limitations of some typical 8-bit computer platforms. The display formats fully supported are Commodore 64 high resolution, Commodore 64 multicolor, ZX Spectrum, MSX Screenmode 2, Amstrad CPC mode 0, Commodore Plus/4 high resolution and Commodore Plus/4 multicolor. Multipaint now includes partially supported new target platforms, with limited file import/export facilities.

Multipaint is meant to encourage a way of creating low resolution pixel images in direct dialogue with the color limitations of the target platform. Especially 8-bit computer visuals are often defined by a presence of a color grid which is coarser than the actual pixel resolution which is quite low to begin with. Working with and around these limits often means trying out a variety of approaches as you go along.

With Multipaint, any changes the current tool would inflict on the color grid are always visible as the action is done, just as it would be on the real hardware. All tools are intended to work directly with as few parameters as possible and to be relevant for 8-bit drawing.

The program is heavily influenced by Daniel Silva's original Deluxe Paint on the Amiga. This stems from the belief that the DPaint model is good for low resolution (320x200x16) work. If you prefer Photoshop/GIMP style editing I can recommend *Pixel Polizei*, an 8-bit image creation tool for your preferred interface.

Acknowledgments:

The Multipaint model and newest features

If you have stumbled here with no familiarity with pixel art, the C64 multicolor mode might be the best starting point as the color limitations are not as severe. If you need advice for 8-bit image-making there are many examples, tutorials and information on the internet.

Multipaint attempts to resolve the color foreground/background internally as you go along. For the technically minded, the program appears to change or even lose internal picture information. Take a ZX Spectrum mode for example: If a character area is drawn full with color red, it will be further on treated as an empty character of that background color. In other words, Multipaint is not an editor, but a paint program.

When editing complex color graphics I cannot really remember whether a particular color on screen is technically "background" or "ink". I wanted a program to solve this for me, and this is the main reason why I made Multipaint. I have done my best to make the drawing experience as simple and intuitive, but the platform characteristics cannot simply be ignored when drawing.

If you hover a drawing tool (pixel) over a color area where a new color does no longer fit, the color underneath will be changed, if possible. See platform-specific behaviors below.

Multipaint now supports aspect ratios (in a limited way). There is a square aspect ratio and a flat aspect ratio, which can be closer to the true target platform aspect ratio. This is currently only relevant for the MSX mode.

Multipaint now has a 1:1 preview. Press *backspace* to switch it on/off, and use mouse to drag the preview around. If you need more room for the window, you may want to check the new preference options for forcing the window size.

Multipaint C64 modes support alternative palettes. As there is really no "one true C64 palette", you can use the normal image loader to load an .act format palette. The current image won't be overwritten, only the current palette will change.

You can also default to an existing palette file in the prefences. At www.colodore.com you can export colodore c64 palette variants in .act format.

I have included a colodore palette file and also the old pepto, in case you want to revert to the default. The .act file has 256 colors in 24-bit R,G,B byte format, only the 16 first colors are used. A hex editor is enough for changing the values.

Supported target platforms

Commodore 64 hires

The resolution is 320 x 200, with 40 x 25 color resolution, two colors for each 8 x 8 pixel area. There are 16 colors to choose from. The border color can be selected from the 16, but there is no overall background color.

Color behavior: If you attempt to draw with a new color in an 8x8 area that already has two colors, the color underneath the pixel(s) will be changed to the current color. You may also force the single-color 8x8 area under the pointer with left-ctrl/command.

Commodore 64 multicolor

Here we have a lower 160 x 200 pixel resolution with a 40 x 25 color resolution. Each 4x8 pixel area can hold three different colors plus the overall background color, which is uniform for the whole picture. The colors can be any of the 16.

Color behavior: If you attempt to draw with a new color in a 4x8 area that already has the maximum of three colors, the color underneath the new pixel(s) will be changed to the new color. You may force colors (left-ctrl) under the pointer, but screen background cannot be forced if the area has all the three other colors. The background color can be drawn freely everywhere.

ZX Spectrum

Pretty similar to the C64 hires, but the screen area is 256 x 192 pixels. The color grid is 32 x 24, with 8 primary video colors in two brightness variations. The black color does not have a bright variant, giving a total of 15 different colors.

A subtler limitation is that the brightness variations cannot be mixed inside the same 8 x 8 area. The exception is the color black, giving rise to the primaries-on-black aesthetic that commanded many ZX Spectrum games.

The ZX Spectrum has a border color which can be selected from the 8 low-brightness colors. There is no overall background color. The FLASH attribute has not been implemented in Multipaint.

Color behavior: As in C64 hires, but the brightness brings additional complexity. For convenience, the corresponding color in different brightness is usually treated as the same color.

MSX1

Just as with the ZX Spectrum, the screen area is 256 x 192 pixels. But here the color grid is 32x192, which means 2 colors can appear in an 8 x 1 pixel area. The MSX has an overall background zero-color, selectable from the 15. This also determines the border color. Therefore there are two black color slots in the palette to begin with.

Although the situation is technically better than with the Spectrum, the MSX is a bit infamous for having a contrastless and reduntant color palette.

MSX1 mode now supports the flat aspect ratio mode, which is closer to the real aspect ratio.

Color behavior: As in C64 hires, but bear in mind the color area is 8x1.

Commodore Plus/4 hires

The hires mode is very similar to Commodore 64 hires, a 320 x 200 pixel area with a 40 x 25 color resolution. There are a staggering 121 colors to choose from. However, instead of a free palette there are 16 colors with 8 luminosities, giving a palette dominated with pastel colors.

Color behavior: As in C64 hires.

Commodore Plus/4 multicolor

Again, this is very similar to Commodore 64 multicolor, with a 160 x 200 pixel area divided in 40 x 25 color resolution. The colors can be chosen from the 121 color palette, but now only two unique colors are allowed per 4x8 pixel area. There are two separate overall "background" colors to choose from. Some very impressive results can be achieved with this mode but it is tricky to work with.

In theory the Plus/4 modes also cover the Commodore 16, but the export prg files will not work on a C16 at the moment.

Color behavior: As in C64 multicolor, but note that as there are two background colors, the 4x8 areas are more easily "suffocated". The areas with either of the two overall screen background colors cannot be forced to a new color if the area already holds the two non-background colors. The two background colors can be drawn freely everywhere.

Amstrad CPC mode 0

A 160x200 bitmap with 16 colors, from today's perspective the CPC is the most "normal" graphics mode in Multipaint. The colors can be chosen from a maximum of 27, using a palette slider with two steps for red, green and blue each. The color selection is a bit bright and lacking

in greys. Currently there is no export/import format for CPC, but a binary executable can be exported.

Color behavior: The force color (left-ctrl/command) key can be used to change the color inside a 4x8 area under the pointer. This is arbitrary from the CPC point of view but consistent with the other modes.

Instructions

The screenshot below shows the general Multipaint layout:
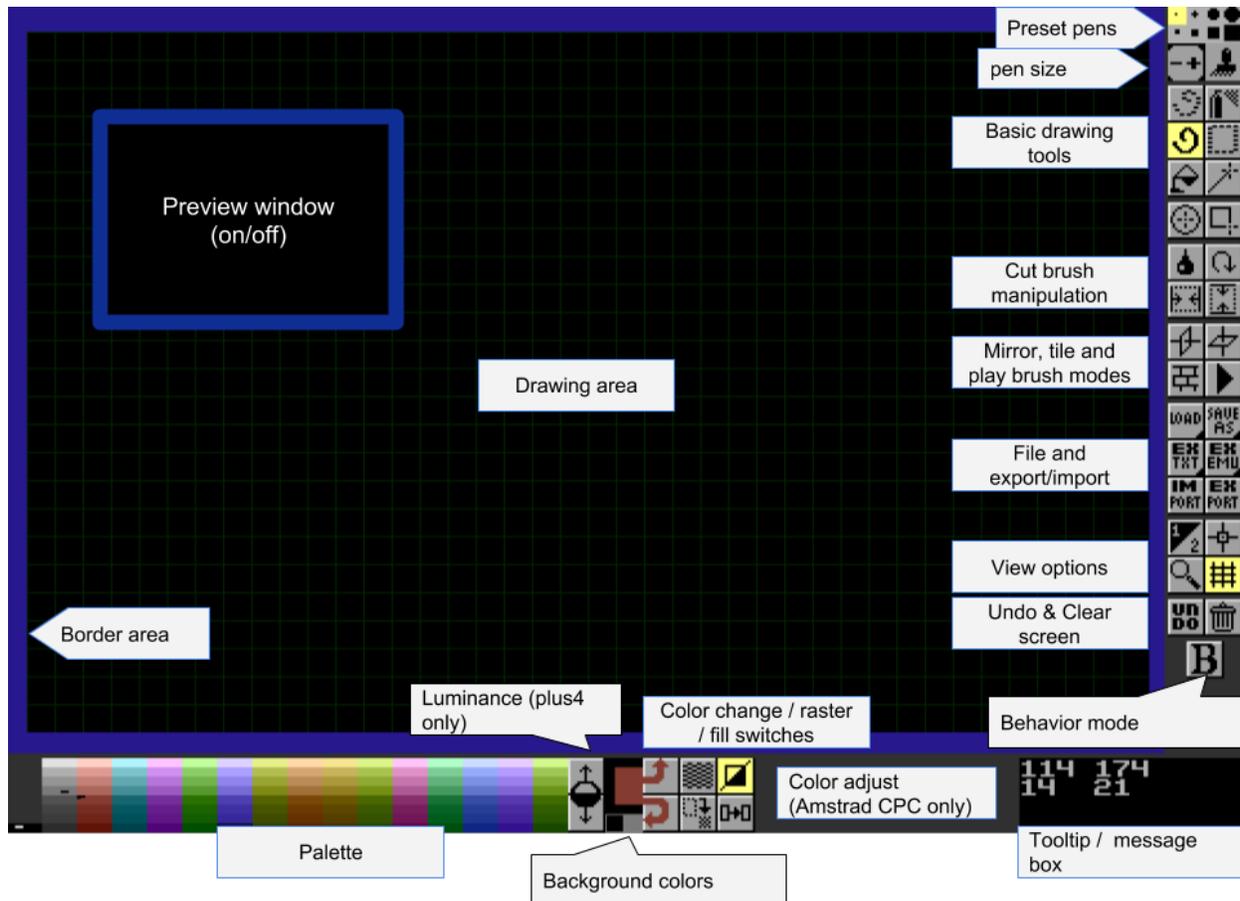
All drawing tools have been collected to the right side of the drawing area. The bottom of the window is mostly reserved for color-related options. This section is opened up in detail below.

At the bottom right there is the help/tooltip/message window, which displays coordinates and other helpful information. It will show the keyboard shortcuts for each tool if you hover the mouse over the icons.
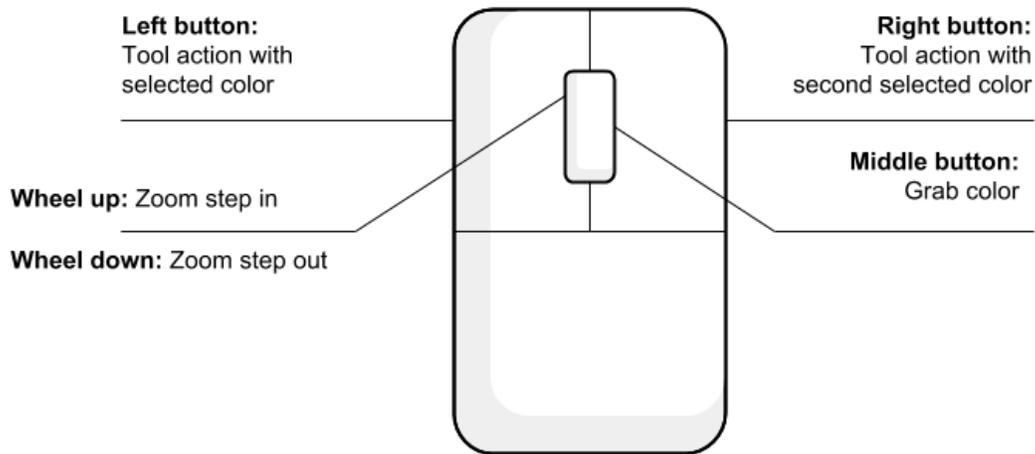
The *preview window* can only be turned on and off from the keyboard, using *backspace.*

There are options for the window size, adjustable from the prefs.txt file. (see further below)

Mouse control

A mouse with at least two buttons is required for this program to work fully.

In the drawing area:

Alternatively, use key , (comma) for the grab color function.

(Middle mouse button+shift)	**Grab a grid-sized brush from under the point** **Alternatively, use key v**
Hold Shift	Keep grid constraint on ("snap to grid")
Hold Control/Cmd	Force color under pointer

Over the palette:

Mouse Left button	Choose left button color
Mouse Right button	Choose right button (second) color
	Key < or clicking the color plate inverts the colors.

Over icons:

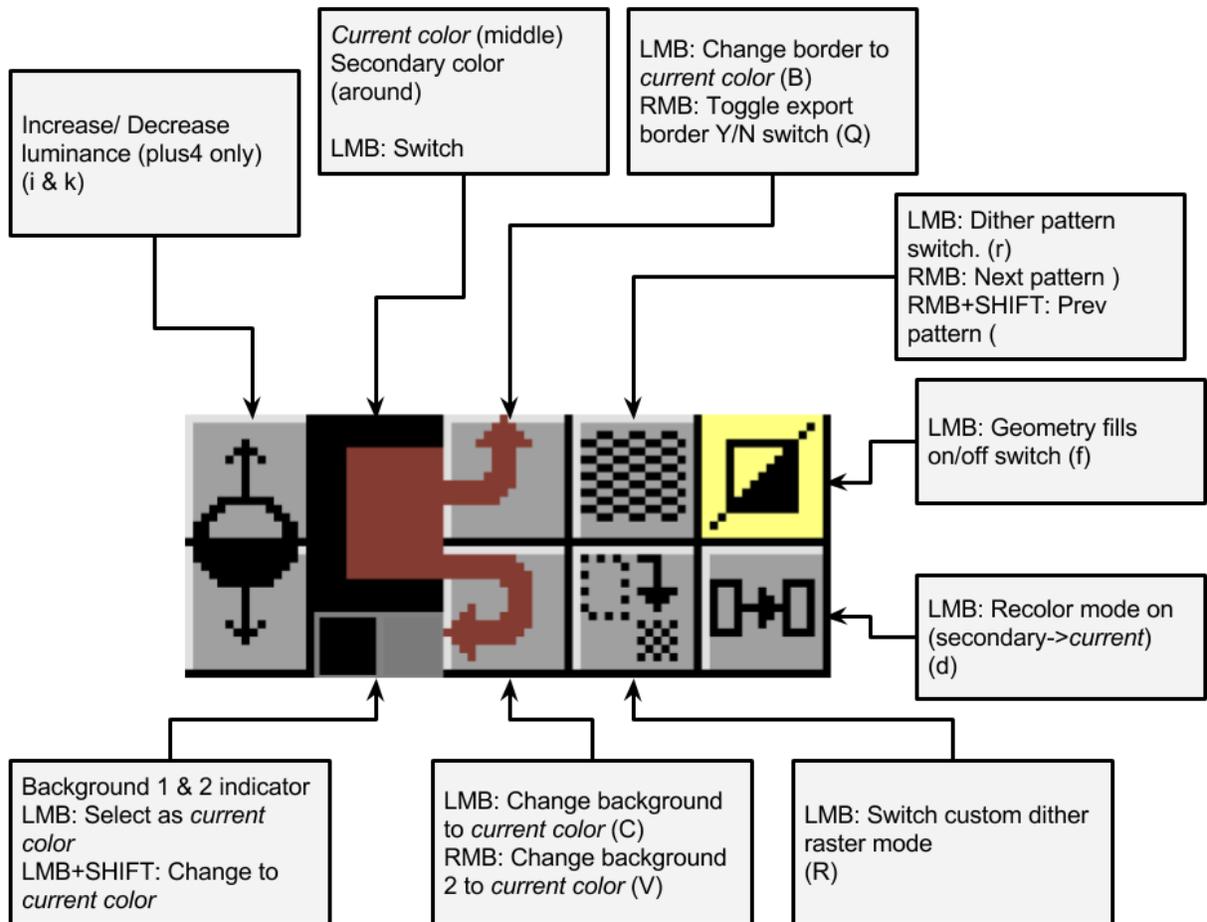|	
Mouse Left button	Command
Mouse Right button	Alternative Command (when appropriate)
Hover	View tooltip/key shortcut

Following instructions list all the key shortcuts for the tools and actions in the program.

Note that for capital letter keys you need to press SHIFT+key to achieve the effect. For example "t" is simply the key t, whereast T is SHIFT+t. Care has been taken that a laptop keyboard and other limited keyboards would be sufficient for using this program. The numeric keypad and function keys generally do nothing. Multipaint has not been made with touch screens or pen/tablets in mind.

Color options layout (bottom of the screen)

Increase/ Decrease luminance (plus4 only) (i & k)

Current color (middle)
Secondary color (around)

LMB: Switch

LMB: Change border to current color (B)
RMB: Toggle export border Y/N switch (Q)

LMB: Dither pattern switch. (r)
RMB: Next pattern)
RMB+SHIFT: Prev pattern (

LMB: Geometry fills on/off switch (f)

LMB: Recolor mode on (secondary->current) (d)

Background 1 & 2 indicator
LMB: Select as current color
LMB+SHIFT: Change to current color

LMB: Change background to current color (C)
RMB: Change background 2 to current color (V)

LMB: Switch custom dither raster mode (R)

LMB=Left Mouse Button click
RMB=Right Mouse Button click

The current color is the color with which all drawing operations are made, when clicking left mouse button. The secondary color is generally the second mouse button color. The secondary color also acts as the brush transparency. Changing the secondary color after cutting a brush changes the transparent color.

The key shortcuts are explained in mode detail below.

This group has colors and geometry fills. Not all platforms have the same set. **Clicking the tiny boxes below the color indicator selects either of the background colors, if available. Click+shift to change them to current color.**
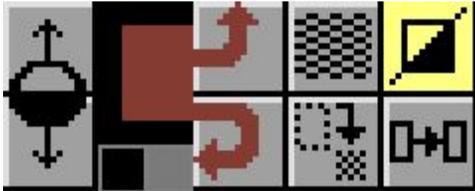

<	Invert colors	Swap left mousebutton/right mousebutton color
B	Set Border color	Make current color Border color (not in MSX1)
C	Set Background color	Make current color Background in C64 multicolor/MSX1
V	Set 2nd Background	Make current color Background 2 in Plus/4 multicolor.
r	Simple raster on/off	Draw with the current preset dither pattern.
(	Select previous dither	Select previous from a set of preset dithers.
)	Select next dither	Select next from a set of preset dithers.
]	Switch dither offset x	Change dither point of origin x
[	Switch dither offset y	Change dither point of origin y
R	Brush pattern on/off	Draw with a brush-derived pattern.
f	Fill geometry on/off	Solid rectangle and ellipse commands. Default:on.
d	Recolor mode on/off	Change instances of secondary color to primary color
,	Grab color (pipette)	Switch color to one underneath the pointer
TAB	next color	Change color to the next one in the palette (SHIFT -)
i	increase brightness	Select next in brightness cycle="lighter"
k	decrease brightness	Select previous in brightness cycle="darker"
Q	Export border switch	Select whether to export border with PNGs

In recolor (d)-mode, the selected right button color will be used as a "from"-color, whereas the selected left button color will work as a "to"-color. The recolor tool works somewhat differently in other modes, it will act as a pixel-level stencil for the selected color. Luminance/brightness is for Commodore Plus/4. The C64 and MSX modes have an experimental arranged order from darkest color to the lightest.

Switch between preset dither patterns by either with the right mouse button over the raster icon, or with the keys (and). [and] keys alter the coordinate zero point for the dither pattern. The (R) custom pattern mode requires that a brush is created with the grab brush (key 4). This works best if the pattern is simple and monochromatic. The secondary (background) color acts as transparency, which needs to be checked if the 'R' mode does not appear to work.

Preset pens / Brush group

These show what kind of pen or brush is currently being used. The pixel is a safe starting point:
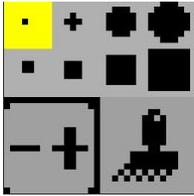
-	Reduce pen size	Alternatively, h
+	Increase pen size	Alternatively, H
.	Pixel-sized pen	Revert to 1-pixel size drawing at any point.
9	Paint with cut brush	The brush has to be first cut with key 4.

Generally, all the drawing tools will work with the currently selected pen / brush. For example, if you cut a brush and start drawing geometric lines, these lines will follow the shape of the brush from start to finish.

Basic tools group

These are the basic workhorses of Multipaint. Select a tool and start messing around:

|			
1	Draw	Draws pixels
2	Spraycan	Draws a squiqqle of random pixels
3	Continuous Line	Draws a continuous line
4	Grab Brush	Grab an area of the page as a brush
5	Flood Fill	Fill an enclosed area
6	Line	Draw a single line between two points
7	Ellipse	Draw an ellipse
8	Rectangle	Draw a rectangle

As mentioned above the tools work differently with Left and Right Mousebuttons. With the Grab Brush (4) command this means the right button clears the grabbed area after the selection. After grabbing a brush, brush-related switches will be reset for clarity.

Geometry tools 4,6,7 and 8 require you to press mousebutton, drag mouse and release to finish the action. Given the nature of the 8-bit graphic modes it often makes sense to use the grid constraint (key c or hold shift) when grabbing and drawing with brushes.

The key v can now be used to quick-grab a grid-sized brush from under the pointer. If your grid constraint 'c' is on, this can help simulate a tile-editor approach to picture making, using on-screen elements as your tile palette.
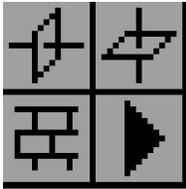
Brush manipulation group

These affect the currently active cut brush:
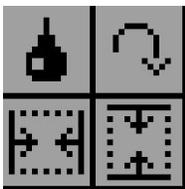
|		
p	Recolor brush on/off	Draw brush with selected color instead of brush colors
z	Rotate brush clockwise	One 90-degree step at a time.
x	Flip brush horizontal	
y	Flip brush vertical	

These are playful tools for experimentation and emergent graphics, or for animation/games:

X	Page X mirror on/off	Every action will be duplicated along central axis
Y	Page Y mirror on/off	Every action will be duplicated along central axis
t	Tile draw on/off	Every action is duplicated across the grid
n	Playbrush on/off	Enter/exit the Playbrush mode
N	Playbrush speed	Switch between three different animation speeds

The mirror switches are a simple way to work with symmetry or bring out some unexpected results. The tile mode helps with drawing continuous tiles for games, laying out patterns for certain types of fonts etc. Currently it works with the alterable grid sizes 4,8 and 16. Note that rectangle, circle and brush tools have size limits when in tile mode.

The Playbrush mode is used for mocking up tiny sprite-style animations or for creating extraordinary live brushes. Press 'n' to activate the mode. The size and amount of frames are interpreted from the current image. A row of sprites need to be bookended by two equally sized, solid rectangles. The sprites and the rectangles need to be of the same width. It is easiest to load the included example file *playbrush256.png* and press 'n' to see how this function works. The Playbrush is kept in memory as long as you do not grab other brushes.
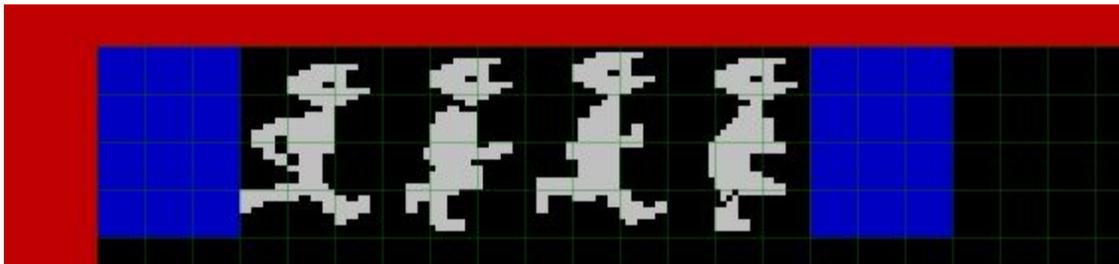
Example of four bookended 24x32 frames prepared for the Playbrush (n) mode

The included file *playbrush256.png* can be loaded to 256-wide and 320-wide resolutions.

File and Import/Export group

All file commands are collected here. Note that the import/export functions are not always fully implemented for every platform, and with the partially supported target platforms these may not work at all.

l	Load page	Load file into the current page
s	Save As	Save current page with a new filename
S	Save	Overwrite existing file with current filename
A	Export text file	Export the page as source friendly data (experimental)
E	Export emulator	Export the page as an emulator file
W	Import from 8-bit paint	Import a file from an 8-bit paint program (not all platforms)
w	Export to 8-bit paint	Export a file to an 8-bit paint program (not all platforms)

View options group

Following options and commands affect how the page is viewed:
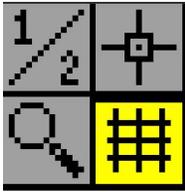
j	Spare page	Switch between back spare and front page
J	Copy to other	Copies current page to spare/front
c	Snap to grid on/off	Constrain actions to grid. Useful for brush actions.
g	Show grid on/off	Show the grid. Useful for color checking
G	Switch grid size 4,8,16	Changes grid size between 4,8 and 16 pixel**s**
m	Magnify on/off	3 x magnify on current pointer position
M	Extreme magnify on/off	8 x magnify on current pointer position
Up	Scroll up	Scroll magnified screen up
Down	Scroll down	Scroll magnified screen down
Right	Scroll right	Scroll magnified screen right
Left	Scroll left	Scroll magnified screen left

An alternative for (c) is to hold down the SHIFT key during actions.

The spare page can hold brushes, or switch between two alternate versions of your image, etc. Note that the spare page has its own filename. Be careful not to overwrite one with the other when saving.

Page commands group

These commands affect the whole screen:

|	
u	Undo	Undo last action
U	Redo	Redo action (20 steps)
o	Clear screen	

Affects color adaptation:

b   behavior mode on/off   Switches the intelligent color behavior on and off

The first versions of Multipaint used a thing I called "intelligent color adaptation". Using ZX Spectrum mode as an example, this meant that whatever color was seen as dominant within an 8x8 area was considered the background. However, it tended to be problematic at times. The *force color* (ctrl) was added as a remedy for changing a color if Multipaint refused to suggest it.

From the *Metal Edition* (2017) onwards, Multipaint processes color in a more straightforward fashion. It is still "intelligent" - an 8x8 area filled with one color is always a background color. But now, if there is no room for a new color in the area, Multipaint simply changes the color underneath to the pen color.

The *force color* (ctrl) has a new role: It allows you to change the color below the pen for the whole 8x8 area. (Bearing in mind the 8x8 can also be 4x8 or 8x1 in certain modes.)

There are situations where the old intelligent mode is useful, especially when freehand drawing over existing shapes. So I have let the old mode stay as an alternative 'b' mode.
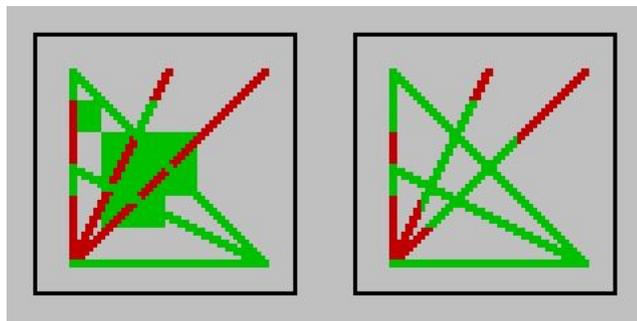
Left: ZX Spectrum without the 'b' mode. Right: using the 'b' intelligent adaptation. In both cases, the red lines have been drawn first, and the green lines on top of them.

About file import/export

The files can be saved in the .bin format, or alternatively as a .png image file. **You need to type in the png file extension in order to export a PNG.** You can also load .png or .jpeg files, which will be *roughly* converted to the current platform color and resolution limitations.

Multipaint can also load and save in a few 8-bit formats. These are *Art studio* for C64 hires, *Advanced Art Studio* for C64 Multicolor, *scr* (screen$) for ZX Spectrum, *sc2* (Screenmode2) for MSX, *Botticelli* for Plus/4 hires and *Multi Botticelli* for Plus/4 multicolor. The Amstrad CPC mode does not support this feature yet, but an executable binary can be exported.

The spare page key switches between two different pages, so you can rapidly edit and move elements between two different images in the same session. With Multipaint, you cannot hold more than two files "open". The two pages have their own undo buffers and filenames, so saving the front page does not overwrite the other file. Press shift+j to copy the current page to the other page. This action can be undo-ed at the target page.

The *E - Export emulator* key outputs a different kind of file depending on the chosen platform, whereas the *w/W* import and export files in native paint program formats.

C64: Multipaint outputs a PRG file, which can be loaded into VICE C64 emulator (x64). If you want to transfer the file to a real C64 it should run happily there. Multipaint also understands the Art Studio and Advanced Art Studio formats. The FLI and AFLI are not exported.

ZX Spectrum: Multipaint outputs a TAP file which can be loaded in the Fuse Spectrum Emulator for example. If you use the TAP2WAV utility, you can play the output audio to a real Spectrum. Some hardware allows you to load TAPs directly into a real Spectrum, too. Multipaint understands the SCR (SCREEN$) file format. FLASH is not supported.

MSX: Multipaint outputs a COM file, which is an executable for MSX-DOS. For certain emulators, the COM file has to be placed inside a suitable disk image before running. Multipaint understands the SC2 (Screenmode2) file format.

Plus/4: As with C64, the output is a PRG file and can be loaded for example into VICE. (xplus4). Multipaint loads and saves the Botticelli and Multi Botticelli file formats.

CPC: The output is a binary file that can be transferred to a CPC disk image with a suitable software, such as iDSK. Multipaint does not currently support any CPC paint program file formats.

The preferences file

You can alter some of the Multipaint options through the prefs.txt file. It is located at the same folder as Multipaint. Edit it with a text editor.

ZOOM=

1,2,3. Leave empty for default 2. The tiny window may fit better and be speedier on some older computers/displays.

MACHINE=

Start automatically with the specified platform. Leave empty to ask each time. See the prefs.txt file contents for valid entries.

PATH=

The default path to your image folder. Leave empty for default.

PALETTE64=

Full filepath and filename to your alternative C64 palette file in .act format. Leave empty for the default "pepto" palette.

FORCEX=
FORCEY=

ADJUSTH=
ADJUSTV=

The Multipaint program window can forced into a certain size regardless of ZOOM, for example for better preview window positioning. The target screen positioning can be adjusted as the amount of pixels from the toolboxes. The repositioning ONLY applies for forced windows.

Do not confuse the ZOOM and FORCEX/Y with target image size, these can not be adjusted.

FILELIST=

Default mode for displaying files. 0=normal, 1=list.

PNGSCALE=

PNG export image scale. Leave empty for 1.

PNGHBORD=
PNGVBORD=

PNG export border width and height. Whether the border is exported or not, is controlled from inside Multipaint, so here 0 defaults to 1.

NOASPECT=

Defaults to 0. If you select 1, the flat aspect ratio modes (MSX) will have 1:1 aspect ratio. Note that 256-high modes have a larger window size that may not fit your sscreen.

WHEELUP=
WHEELDOWN=

Multipaint now has simple mousewheel support. The mousewheel defaults to zoom in/zoom out, but it is customizable in case you want to invert or remove this. See the prefs file for a few examples.

MIDDLEBUTTON=

The middlebutton, which defaults to pipette/grab color, is also now customizable.

The mouse customization support is still very primitive. Note that although the command values correspond with the ASCII values of the keypresses that activate the commands, not everything works well.

Additional key commands

The following keyboard commands are mostly added to provide some comfort for those who expect typical key shortcuts. There are also some actions that are not directly doable otherwise.

I have not ensured these work very well across all platforms. In Mac OS X, you may have to use the CMD key instead of CTRL.

CTRL+Z	Undo
CTRL+Y or CTRL+SHIFT+Z	Redo
CTRL+A	Select whole screen as a brush
CTRL+C	"Copy" Set tool to Grab Brush
CTRL+V	"Paste" draw with a cut brush
CTRL+S	Save current page

CTRL+N   Clear page

Some tips

In the included *keysheet.png* file I have highlighted the shortcuts that I think would be most useful to know first for effective use of Multipaint.

Use the middle mouse button (or comma-key) for picking colors from the screen rather than from the palette.

The ctrl/cmd key is used for changing color areas inside the color grid.

The draw model is hierarchical: You can draw a squiggle, grab that squiggle as a brush and then draw lines and boxes with that brush. If you get lost, press . (comma) to get to the single pixel mode.

Large brushes are likely to be very slow with continuous line and the geometry tools.

With geometry tools, note that many switches can be updated as long as the mousebutton has not been released. For example, fill geometry and the raster pattern can be turned on/off while drawing lines, rectangles and ellipses.

If you want to load target machine images as png files, this is possible. For example, this could be a 320x200 image file depicting C64 screen contents, from an emulator or internet. The best results are achieved if the image colors are really close to the target platform colors. With C64 this means the source image would have to follow the Pepto palette quite closely. Using alternative palettes will change the results of the conversion process.

Changelog

18.10.2018 version (Multipaint 2018b, minor fixes)
-Current color indicator bug fixed
-Preview window update more consistent
-Moved the tool icons slightly down, not so near the "close window" button in some systems
-Fixed some of the undocumented modes, e.g. 256 height works without flat aspect

1.8.2018 version (Multipaint 2018)

-Fixed the mac version bug where the icon graphics could not be fully loaded. Other problems with running Multipaint may still persist depending on your mac system and system security settings.
-Added alternative palette support for C64 modes
-Added an alternate aspect ratio support for MSX

-Added a preview window (backspace)
-Added window and mousebutton/wheel adjustment functions in the prefs
-Fixed filename bugs, importing png/jpg no longer changes the edited file name
-Added undo levels to 20

Changes from 22.5.2016 version

-Changed the color behavior to a more straightforward model. Old behavior retained as 'b' mode.
-Overall color behavior is more uniform between formats, multicolor and otherwise
-Changes to mouse event handling should make the program a bit more useable across platforms and computer speeds
-Added preset dither (raster) patterns and offset adjustment
-A bit more visible grid (not in plus4 and CPC modes)
-Metal User Interface. Why? I wanted some program changes to be visual. Tiny adjustments to icon graphics and visual behavior. Visible dither on icon, visible spare page on icon.
-Bug fix: UI elements overlapped in CPC mode when using ZOOM=3
-Bug fix: Machine selection through prefs.txt did not really work
-Bug fix: In CPC mode palette changes could not be undoed (in loading pngs for example)

Changes from 28.3.2016 version

-Added Commodore Plus/4 modes with relevant changes to palette display
-Added Amstrad CPC mode with palette selector
-Added native paint program Import/Export options for most platforms.
-Grid size variants 4,8,16 accessible from 'G'
-Tile draw 't' switch for drawing continuous tiles
-Playbrush 'n' switch for live brushes and testing small sprite animations
-A bit experimental brightness keys 'i' and 'k' for working with Plus/4
-Brush rotate is now step-by-step, brush orientation generally improved
-Mac OS X bug: ctrl key was a bad choice for force color. Thanks to Marq for pointing this out!
-Bug fix: cutting a brush resulted in an unnecessary step in the undo buffer
-Bug fix: some problems with ZX Spectrum screen clearing in certain conditions
-Internal work to improve brush functions

Changes from 15.2.2016 version

-Visual border: previously the border selection was not visible except in the palette.
-Introduced the "force color" (hold ctrl) key to help pixel editing in low-color modes.
-Fixed ZX Spectrum brightness selection problem: previously it was difficult to alter brightness with the recolor tool.
-Introduced shift+j key command for "copy to other page".
-Changed the way lines behave when using grid constrained brush drawing.
-The magnify window tool is more accurate and centres more accurately.
-The brush handle is more consistent when rotating and flipping.
-The flood fill origin point is now never grid constrained.
-The ZX Spectrum palette is now similar to the Fuse palette.
-Bug fix: MSX png output could be incorrect depending on background color.