

Multipaint 2020

<http://multipaint.kameli.net/>

The Multipaint model	3
Supported target platforms	4
General instructions and the Multipaint layout	7
Color options layout (bottom of the screen)	9
Pull-down menus	10
File handling	10
Mouse	11
Tool groupings and shortcuts	12
Color options group, continued	12
Preset pens / Brush group	14
Basic tools group	15
Brush manipulation group	15
Play tools group	16
File and Import / Export group (Icons removed in 2019 version)	17
View options group	18
Page commands group	18
Pipette	19
FX paintmodes (Beta, Atari ST)	20
More information	21
Settings menu item	21
About file import / export	21
The preferences file	24
Command line arguments (Probably Linux only?)	26
Additional key commands	27
Some tips	28
Changelog	28
2020 (2020) Processing 3 w/ resizable appwindow. Added CPC overscan, C64 Free	28
2019.9 (2020) Added CPC1 mode & fixes	29
2019.8 (29.3.2020) Palettebox & improvements	29
2019.6 (14.3.2020) added Atari ST mode	29
2019.5 (4.1.2020) added command line options (Linux)	29
2019.3 fixed version	29
2019.2 fixed version	29
2019 version (Multipaint 2019)	30

Multipaint 2020 manual

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 and multicolor, ZX Spectrum and extended ULAplus palette mode, MSX Screenmode 2, Amstrad CPC mode 0/1, Commodore Plus/4 high resolution and multicolor, Sinclair QL mode 8 and Atari ST low resolution mode. Multipaint also partially supports other 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 low resolution drawing.

The program is heavily influenced by Daniel Silva's original *Deluxe Paint* on the Amiga. This choice 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 Processing code was written by Tero Heikkinen (Dr. TerrorZ)

The source makes use of some of Markku Reunanen's (Marq) Java code adapted from his excellent PETSCII editor. Marq also wrote the Amstrad CPC template files for MODE 0 and MODE 1 and the overscan mode.

Thanks to KiCHY for sending information about the Commodore Plus/4 and sending the Botticelli files.

Thanks to Isildur for suggestions about the dithering tool presets and behavior.

A bin2tap file output from the Fuse package was the basis for the ZX Spectrum TAP export template.

The pepto and colodore C64 palettes are created by Philip "Pepto" Timmermann.

ULAplus by Andrew Owen 2009. Thanks for Andrew Owen, Andy Green, Jonathan de Vaal and Mike van der Lee for discussions, testing and insights. See <https://sites.google.com/site/ulaplus/>

Thanks to Andy Zeidler (Shine) and Compyx for comments and testing.

Thanks to Stefan Vogt for testing the Atari ST mode.

The Multipaint model

This is not a manual for drawing low resolution and low color images. If you need advice for making 8-bit or 16-bit low pixel art, there are many examples, tutorials and information on the internet.

Old graphics modes were not always about lower resolution and less colors. Some modes have very peculiar color limitations, which Multipaint simulates.

In color-challenged modes, 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.*

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.

Commodore 64 free

This is a duplicate of the Commodore 64 hires, but the attribute limitations do not apply. You can therefore for example work on sprite combination graphics. However Multipaint does not do these type of conversions or exports.

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.

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.

ZX Spectrum with ULAplus

The ULAplus palette enhancement is developed by Andrew Owen and implemented in various modern ZX Spectrum computers and emulators. ULAplus can be combined with different video modes. Multipaint only supports 256 x 192 with 32 x 24 color grid for now.

Color behavior: In this mode, you can use 64 different colors from a total of 256 colors. The 64 colors cannot be directly combined with each other, but Multipaint tries its best to make it easier by duplicating colors when needed. The experience should be quite close to the ZX Spectrum mode.

Any colors are translated into legal combinations when possible, adding duplicates of the colors to the palette parts where it is necessary. You may find that using the mode carelessly results in a lot less than 64 unique colors, as the number of needed duplicates keeps growing.

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.

MSX1 mode can be viewed in a flat aspect ratio mode, although it is very approximate.

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.

Amstrad CPC mode 0 overscan

The same as mode 0, except the resolution is 192x270. In Amstrad this is a software-enabled mode.

Amstrad CPC mode 1

A 320x200 bitmap with 4 colors. Just as with mode 0, the colors can be chosen from a maximum of 27, using a palette slider with two steps for red, green and blue each.

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.

Sinclair QL mode 8

The Sinclair QL is a 256 x 256 flat aspect ratio mode. This is a bitmap mode with a fixed palette of 8 colors and an atrocious FLASH bit which is not simulated here. The aspect ratio is not exactly correct either. There's no executable export as a QL can use the format output directly. In a vanilla QL you can LBYTES mdv1_filename,131072 to get the binary on the screen.

Color behavior: The force color (left-ctrl/command) key can be used to change the color inside a 8x8 area under the pointer. This is arbitrary from the QL point of view but consistent with the other modes.

Atari ST low resolution

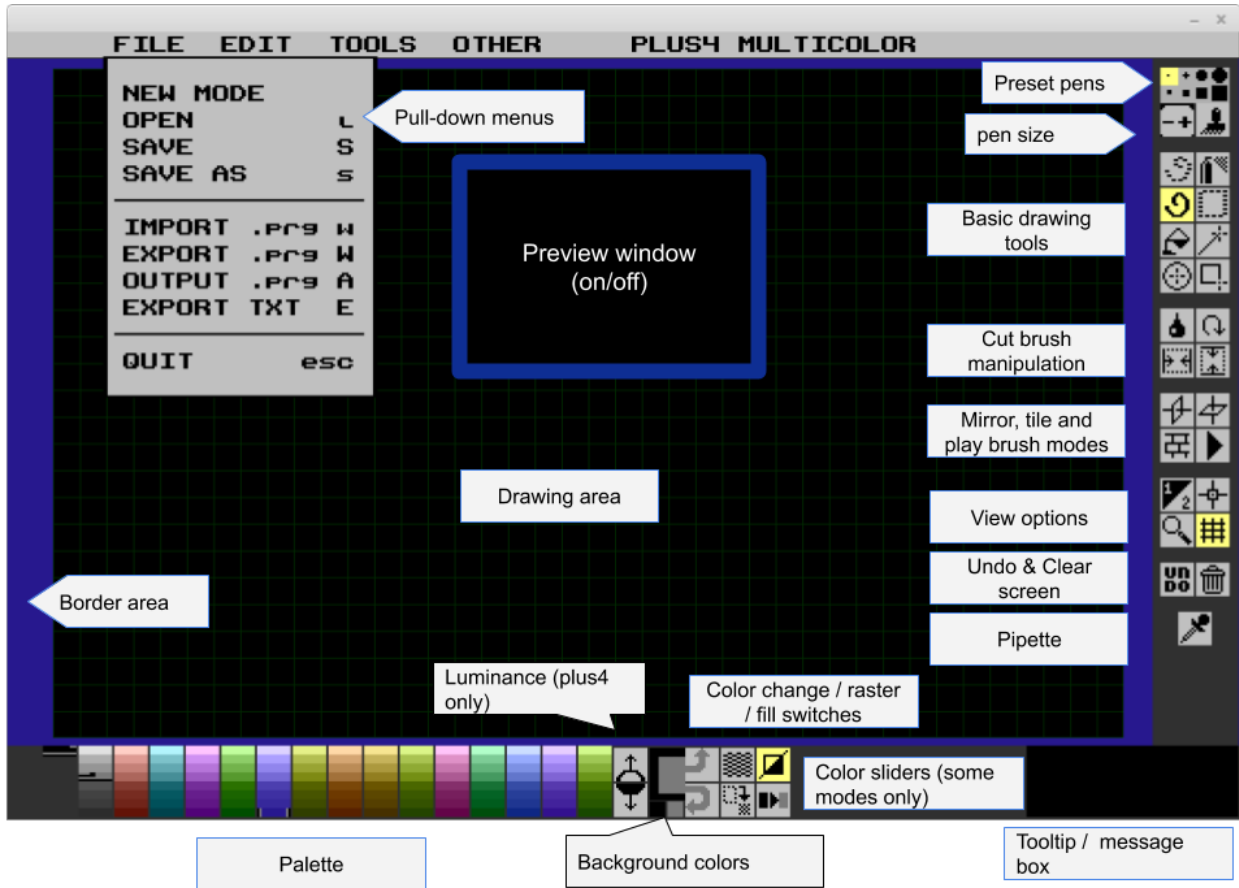
Atari ST 320 x 200 with 16 colors. This is a bitmap mode with adjustable palette, 8 steps for Red, Green and Blue. The Atari STE colors (with 16 steps) are not as yet supported.

The export/import modes are Degas Elite, or "pi1".

Color behavior: The force color (left-ctrl/command) key can be used to change the color inside a 8x8 area under the pointer. This is arbitrary from the Atari ST point of view but consistent with the other modes.

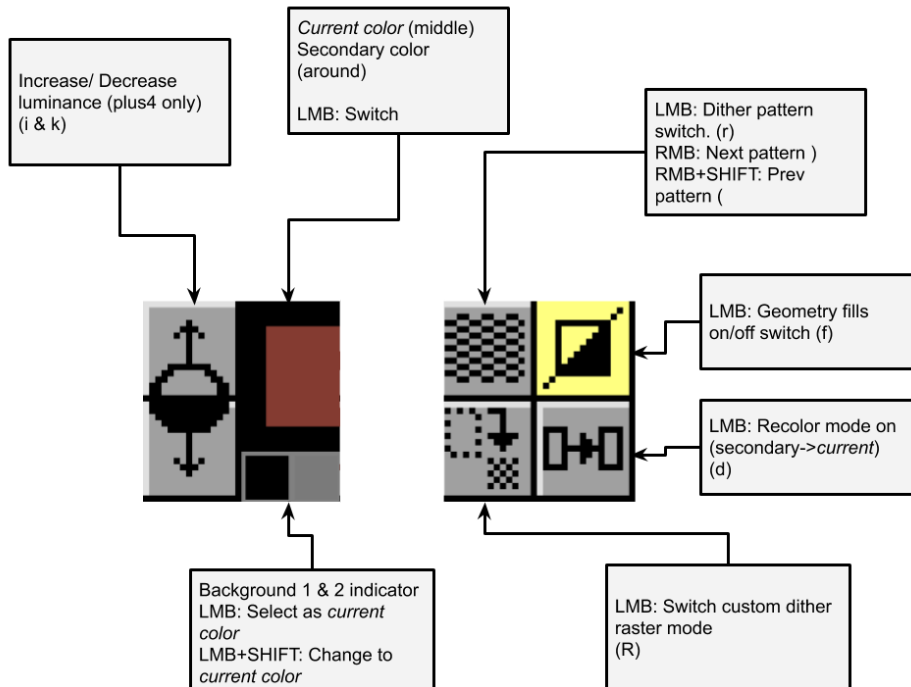
General instructions and the Multipaint layout

The screenshot below shows the general Multipaint layout:



All drawing tools have been collected to one side of the drawing area (the side can be changed). The bottom of the window is mostly reserved for color-related options. This section is opened up in detail below.

At the bottom there is a 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.



Color options layout (bottom of the screen)

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.

Border and Background change options are now found from the menus or the PALETTE window. The key shortcuts are explained in mode detail further below.

From Multipaint 2020 onwards, the application window can be resized by dragging the window edges. The window contents are scaled depending on whether the full screen mode fits together with the toolabrs. If you want to disable this auto-scaling, you can use LOCK SCALE from the OTHER menu. Maximize/Minimize may work depending on the system.

Pull-down menus

The top of the screen has pull-down menus for accessing tools and commands. For the most part the commands itself are not new. Significantly, NEW MODE allows you to change the platform without having to restart Multipaint.

File operations and import/export has now been moved to these menus to reduce the clutter in the toolbars.

FILE	EDIT	TOOLS	OTHER	FX (beta) *
NEW MODE OPEN SAVE SAVE AS IMPORT * EXPORT * OUTPUT * EXPORT TXT SETTINGS QUIT	CLEAR UNDO REDO SPARE TO SPARE COPY ALL FLIP	RESET PALETTE SWAP []/[] SET BORDER * MAKE [] = [] SET BACK1 * SET BACK2 * SET GRID GRID 8X0 GRID 8X8 GRID 16X16 GRID 24X21 HIDE/SHOW GRID	TOOL SIDE LOCK SCALE DARK THEME PREVIEW PEPTO * COLODORE * ASPECT * ABOUT	NORMAL SHADE SMOOTH DARK/LIGHT MIX TINT SMEAR BLEND

Options marked with * may be context/platform specific. Bolded options do not have a key shortcut.

File handling

Multipaint 2019 comes with revised file handling for increased safety.

- File overwrites and mode changes will have to be confirmed.
- Multipaint reminds you of unsaved work when loading.
- Unsaved pages are pointed out when quitting
- You cannot UNDO past a point that would require a filename change.
- Using “clear” will reset the filename.

A page is “unsaved” if a tool has been used on the page or you have changed colors..

Multipaint is written with Processing. It is notoriously difficult to make bypass the window close button on the exported application with this language. The 2020 version does attempt this, though, and it should work.

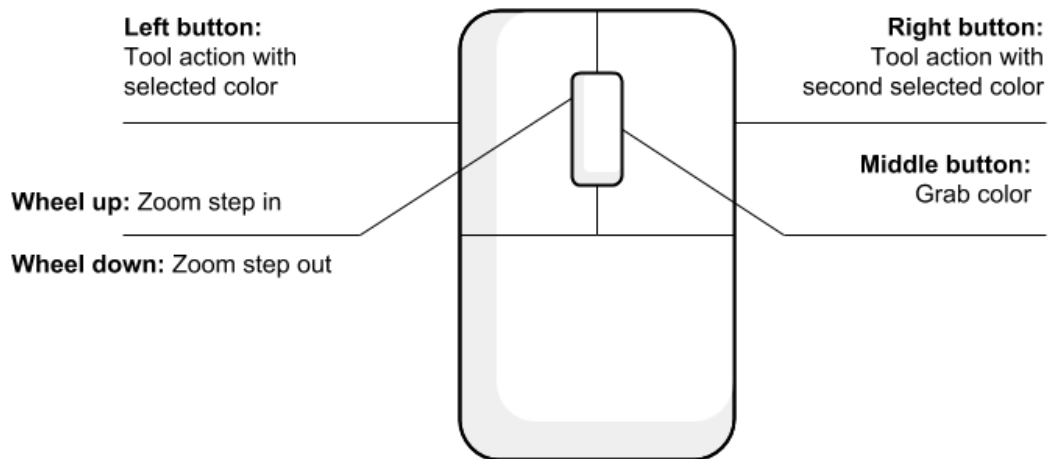
Multipaint tries to save the pages as backups on any exit, the files are called **multipaint_auto_page1.bin** and **multipaint_auto_page2.bin** and should be found from the Multipaint folder. If you feel you may have accidentally quit Multipaint, there is a chance you can recover work from the backup files.

For now, you cannot UNDO a mode change directly. In the meantime, you can RECALL the old mode contents immediately after changing the mode, using the pop-up dialogue.

Mouse

A mouse with at least two buttons is required for Multipaint to work fully. Three-button mouse with a wheel is recommended. Multipaint has not been designed for touch screens or pen/tablets, but it should work. From 2019 onwards the interface is bit more pen-friendly.

In the drawing area:



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

~~(Middle mouse button+shift)~~

~~Grab a grid-sized brush from under the point~~

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 / Select tool

Mouse Right button

Alternative Command (when appropriate)

Hover

View tooltip / key shortcut

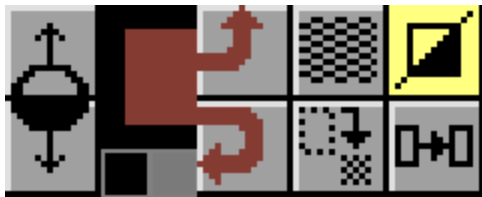
Tool groupings and shortcuts

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, whereas 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.

Color options group, continued

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.**



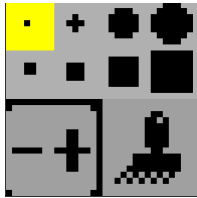
<	Invert colors	Swap left mousebutton/right mousebutton color
B	Set Border color	Make current color Border color (not in MSX1)
C	Set Background color	Make current color Background in C64 multicolor/MSX1
V	Set 2nd Background	Make current color Background 2 in Plus/4 multicolor.
r	Simple raster on/off	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 squiggly 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

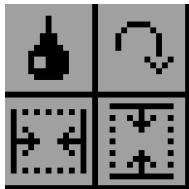
As mentioned above the tools work differently with Left and Right Mousebuttons. With the Grab Brush (4) command this means the right button clears the grabbed area after the selection. After grabbing a brush, brush-related switches will be reset for clarity.

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

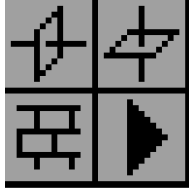
These affect the currently active cut brush:



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	

Play tools group

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. The Playbrush is kept in memory as long as you do not grab other brushes. You can do the Playbrush in spare page and use it in the other page.



Example of four bookended 24x32 frames prepared for the Playbrush (n) mode

File and Import / Export group (Icons removed in 2019 version)

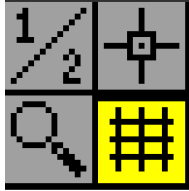
The file options have been collected at the FILE drop-down menu and no longer have corresponding icons. The key shortcuts are still the same.

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. See the section further below in More Information/About file import and export

I	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 pixels
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	

Pipette

A pipette tool icon has been added, mostly because people thought Multipaint does not have pipette/color picker.

Select the tool, click on the image to select the color underneath.



e
, or Middle mousebutton

Activate pipette tool
Pick up color directly (whichever tool is active)

FX paintmodes (Beta, Atari ST only)

I have added a number of 'effects' paintmodes for Atari ST to help create 16-bit graphics.

Using these mechanically probably won't result in great pictures. It's more for that one time when you would need a large area to be darker or lighter to everything else.

NORMAL	The normal color mode
SHADE	If underlying colors are in palette range, they are stepped up or down depending on mousebutton.
SMOOTH	A kind of rough blur.
DARKEN/LIGHT	Infers the color that is the step darker from the underlying color, using the palette steps as a starting point.
MIX	Mixes the current color with underlying color and uses that color.
TINT	Tint the underlying colors with current color. Note the current color darkness affects the result. Right mousebutton: Desaturate
SMEAR	Moves pixels about. Does not work well with geometry tools.
BLEND	Moves pixels about and then mixes them. Does not work well with geometry tools.

Much of these depend on how the palette has been arranged.

If you have set the palette range for a color slide, then SHADE can be used for darkening areas. In some cases DARKEN/LIGHT might already achieve the desired effect.

The **DRAW** and **CONTINUOUS DRAW** work differently with FX. Plain DRAW will repeatedly affect the canvas with the FX brush. CONTINUOUS DRAW will only perform the FX once to the area you are going through, until you release the mouse button.

Note that the FX work together with custom brushes and dither too. SMEAR and BLEND are unique in that they accept the dither pattern as a source, but the outgoing pixels do not adhere to the dither pattern. Sometimes, using a loose dither with BLEND can give better results.

More information

The following topics may be of interest if you want to get deeper into Multipaint.

Settings menu item

A new menu item has been added to the file menu. It will open a small settings menu, currently in beta. These 'settings' are not the same as the prefs.txt discussed below.

These settings will be saved together with the BIN format (Multipaint format) image files. So, if you export or save your work as PNG or platform-specific formats, these settings are not stored.

Add borders to .PNG? If set, any PNG/JPG exported images will have a border with them. This may not be relevant or accurate for the chosen platform.

Hex coordinates? If set, multipaint will display coordinates in hex instead of decimal.

Pixelgrid in magnify? If set, the full magnify mode will display grid on a pixel-level, if the grid is on.

About file import and 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.

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 into a Spectrum Emulator. If you use the TAP2WAV utility, you can play the TAP file as 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. The FLASH attribute is not supported.

ZX Spectrum ULAplus: Multipaint outputs a TAP file which can be loaded in an emulator. If the emulator does not support ULAplus (Fuse does not), the colors will not display correctly.

The SCR (SCREEN\$) file format is the same order as with Spectrum, but adds 64 bytes of palette information. Multipaint recognizes the files from their length.

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. Somewhat unfortunately the extension for an Amstrad binary is “bin”, the same as Multipaint files.

To create a new disk image in iDSK:

```
iDSK mydisk.dsk -n
```

To add the exported Amstrad binary “amstrad.bin” to the disk image, use this:

```
iDSK mydisk.dsk -i amstrad.bin -t 1 -e 6000 -c 6000
```

Multipaint does not currently support any CPC paint program file formats.

Sinclair QL: The output is a binary file that corresponds with the screen structure of the original unexpanded Sinclair QL, typically found at address 131072.

After you have managed to get the file on a media such as floppy disk, you could use the following SuperBASIC command to load it directly to the screen memory:

```
LBYTES flp0_filename,131702
```

Atari ST: The output is a Degas Elite format image, which contains the bitmap in the screen structure of a 320x200x16 low resolution mode, together with the palette information.

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. Macintosh users need to dig it from the “package contents”. Change it with a text editor. In the future, the options are hopefully accessible from within Multipaint itself.

A few settings can be adjusted from the menu item of the same name. This is not the same as the prefs, and are not save alongside the prefs.txt.

ZOOM=

2,3. Leave empty for default 2. I have removed the support for zoom level 1 for the time being. Tell me if you absolutely need it. Zoom level 2 should be comfortable on a vertical resolution of 768 but not lower.

INVERTGUI=

Use 0 to keep Toolbar at the right, 1 for left.

DARKTHEME=

Leave empty or 0 for light GUI theme. Use 1 to make the dark theme default.

GRIDBRIGHTNESS=192

Adjust the grid brightness from 8 to 248.

SCALELOCK=

If you want the GUI to have constant scale (despite resizing app window) use 1

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 be 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, this can not be adjusted.

FILELIST=

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

"FILELIST= 1" no longer works with 2019.3

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 screen.

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.

This 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.

Command line arguments (Probably Linux only?)

Multipaint allows certain options to be passed by the command line (e.g from terminal or a launcher). These are partly the same as in the prefs.txt above. It is possible to pass multiple parameters by separating them with space.

For the casual user, these might not be needed. I added this mostly because I needed them. For setting up a development environment for a particular platform, it might be useful to connect multipaint to a particular folder and files.

This way it is also possible to launch Multipaint when in another folder, by passing the explicit path using MPPATH. If another program or a launcher attempts to run Multipaint, this parameter is mandatory.

For example, the following command line would both set the MPPATH and the default image file path:

```
./multipaint MPPATH=/home/user/Multipaint/ PATH=/home/user/mpimages/
```

Following parameters are allowed. They are executed in the order given, except the high priority MPPATH which will always be processed first.

```
MPPATH=fullpath/to/multipaint/folder/
```

Define Multipaint application folder path explicitly. Can be used for running the app from another folder or via a launcher icon. Prefs.txt file will be fetched from here.

```
PATH=fullpath/to/folder/
```

As in prefs, sets the default folder for image files.

```
LOAD=fullpath/to/filename
```

Loads the specified image file: bin, jpg, png.

SAVE=fullpath/to/filename

Saves a specific file in the defined format: bin, jpg, png.

PNGSCALE=value

Set png export size multiplication. (1-10)

PNGHBORDER=value

Set png export horizontal border size. Relative to PNGSCALE.

PNGVBORDER=value

Set png export vertical border size. Relative to PNGSCALE.

EXPBORDER=value

Define whether to export border at all when exporting images. (0=off, 1=on)

MACHINE=platform_id

Set the platform. C64, C64M, SPECTRUM, MSX, CPC, PLUS4, PLUS4M, QLLOW, ULAPLUS, ATARIST, AMIGA, C64NOLIMIT, COCO3

EXIT

Exit Multipaint (without warning). Only makes sense when scripting from outside. Multipaint itself does not do batches or handle wildcards in file names.

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 . (period) to get to the single pixel mode. SPACE can be now used to reset all the tool properties.

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 dither pattern can be turned on/off while drawing lines, rectangles and ellipses. It may be educational to alter the dither pattern coordinates while drawing using the [and] keys.

If you want to load target machine images as png files, this is possible. For example, this could be a 320x200 image file with 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.

If you load in a png image with a border, Multipaint will check if it could be cropped. This is done if the image inside the borders corresponds exactly with the platform resolution. For example, a png with 384x272 pixels could be identified as a 320x200 image. However, if the border is in any way ambiguous, Multipaint will simply load it all in an scale the image.

Changelog

2020.1 (24.11.2020 inevitable post-release fix version)

- The window resize non-working on some Linux systems has been hopefully fixed.
- The window close 'x' button has been caught.
- Magnifying glass tool had a glitch when using 2x wide pixel modes
- Adjusted the 'unsaved' status to be more logical
- A partial fix to viewport drawing problems emerging when the viewport overlaps the toolbar (shouldn't be allowed in the first place but...)

2020 (2020) Processing 3 w/ resizable appwindow. Added CPC overscan, C64 Free

- Processing 3 exported application
- Application window resizable, interface is scaled (optional)
- Dark GUI theme option
- Amstrad CPC0 overscan mode is complete with executable export
- New SET GRID window that allows non-prescribed grids and grid offset
- Grid brightness adjustable (as its again been wrecked)
- Better ->ULAplus conversions
- JPG/PNG conversion results more informative
- JPG/PNG border deduction (if possible)
- Internal conversions are preferably cropped
- C64 Palette selection preserved between C64-type modes
- Used colors are marked in the palette
- Change a color into another (globally)

2019.9 (2020) Added CPC1 mode & fixes

- Amstrad CPC mode is complete with executable export

2019.8 (29.3.2020) Palettebox & improvements

- New PALETTE window, with manipulation tools such as Spread & Cycle
- Windows can be moved around. The image can be drawn on if the palette window is open
- Palette auto-color adding can be locked in ZX ULAplus mode
- Dither/raster etc. switches don't affect image import any more, for clarity
- Fixed PNG/JPG bordered output bug in 1x1 resolution (last line was duplicated)
- Bunch of FX paintmodes for Atari ST mode in beta:
SHADE,SMOOTH,DARKEN,MIX,TINT,BLEND,SMEAR

2019.6 (14.3.2020) added Atari ST mode

- The Atari ST mode has been made into a fully supported mode (pi1 export/import)

- A “change color into another” quick menu command (for paletted modes)
- A “switch palette index” quick menu command (for paletted modes)
- A settings menu has been added, in ‘beta’ state
- Pixel-level grid when zoomed in (from settings menu)
- Display coordinates in hex (from settings menu)

2019.5 (4.1.2020) added command line options (Linux)

-Arguments can now be passed with the command line. Couldn’t find a way to do this on a Mac, though.

2019.3 fixed version

-Fileselector style change in the hope it will work more universally (with Macs). This sadly removes sorting features from the selector.

2019.2 fixed version

- Koala import/export bug fixed in C64 multicolor mode
- Koala made default in that mode
- Plus4 import/export bug fixed
- New mode is auto-activated when loading a BIN from another mode
- Fix to the “colour underneath” indicator

2019 version (Multipaint 2019)

- Added ZX Spectrum ULaplus mode
- Promoted Sinclair QL mode 8
- Pull down menus and dialogues
- Safer file handling, autosave pages on exit
- Unsaved file indicator *
- New machine select dialogue and on-the-fly mode change
- GUI changes, can be flipped left/right handed
- Smoother mousewheel zoom (5 levels) tha centers on cursor
- Improved txt output / parameters
- Exported files can be re-written with ‘save’
- Direct pepto/colodore access from menu
- Direct aspect ratio toggle for MSX (not for QL)
- C64 sprite-sized grid, MSX 8x0 grid
- Altered palette adjustment sliders, added slight ‘snap’
- Various MSX-related bugs fixed (hopefully)

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 undone (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.

2013 fall: Work started on Multipaint