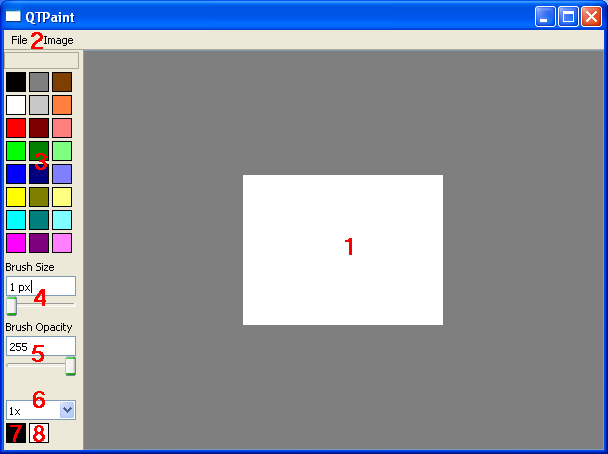
QtPaint

**Introduction**

QtPaint is an open-source raster graphics software being developed by Omid Davoodi, Iran University of Science and Technology. The aim of the project is to create a powerful but simple to use and learn painting software. QtPaint is projected to cover the distance between the software bundled by default with the operating systems, like MS-Paint, and professional raster graphics programs like Adobe Photoshop. It is intended to be used by ordinary people whose needs are sophisticated enough not to be addressed by the simpler programs, but not enough to use even a small portion of the abilities of the professional software.

**How to use the program**

Bellow is the image of QtPaint's main user interface.

**1 : The Canvas**

The canvas is the area where the image is drawn. You can draw on the canvas by bringing the mouse pointer on the desired location, pressing and holding the left mouse button, and moving the mouse pointer. To stop the drawing, simply release the left mouse button. The default brush color is black.

You can also use the right mouse button in that procedure, which would paint using the secondary color of the program. By default, the secondary color is white. For more information on how to change the color of the brush, see the sections 2 and 3.

**2: The Menus**

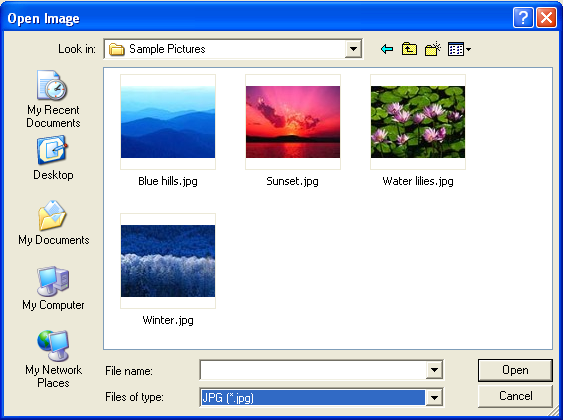
There are two menus on the program's user interface. File, and Image.

**File**

File menu contains the following actions:

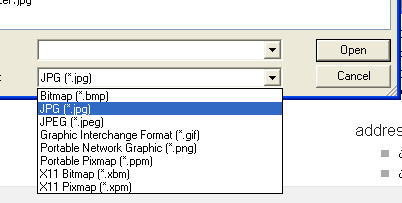
**New**: Clears the canvas. Any unsaved images will be lost by this action.

**Open**: Loads an image from your computer into QtPaint. Any unsaved images will be lost by this action.



Navigate through your computer to find the image you want to edit. Click on the image and then click "Open".

QtPaint supports eight image formats that can be loaded. To select a different format, click on the drop box on the bottom of the dialog and then, choose a format from the list.



It is important to remember that your edited image should be saved if you do not want the changes you made be lost.

**Save**: Saves your image. A very important step is to include the extension in your file name. For example if you want to save your picture by the name "home" and by format "jpg", you should name the file as "home.jpg". The supported formats for saving are: bmp, jpg, jpeg, png, ppm, xbm and xpm. Note that QtPaint doesn’t support saving as gif.

If you want to save on an existing file, there is no need to include the extension.

**Exit**: Closes the application. Any unsaved work will be lost.

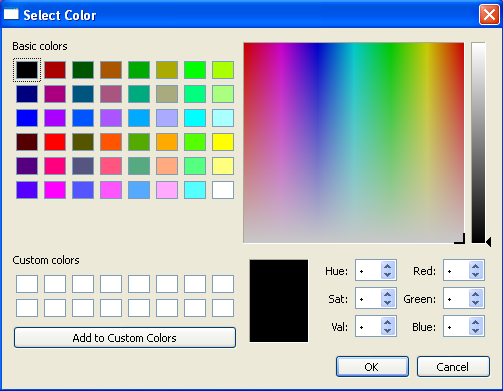
**Image**

Image menu contains the following actions:

**Resize**: Changes the size of the current image. In the dialog shown, determine the new width and height of the image and click OK to change the size. Be warned that changing the size into a smaller one will lose the data on the areas removed.

**Invert**: Inverts the image.

**Color**: Shows a dialog in which you can change the primary color of your brush. Pick a color from the table in the left or use the more advanced picker in the right, then click OK.



**3: Color table**

Color table contains a selection of 24 colors by default which you can use to paint your image. To change the color of your primary brush, left click on the one you desire. To change the color of your secondary brush, right click on the color.

You can change the color of these 24 palettes by double clicking on the boxes. This will bring the same color dialog that was mentioned above. The difference is that in addition to changing the primary color of your brush, it will also change the color of the palette box you double clicked on.

**4: Brush size controllers**

The size of your brush can be changed using the controllers in this section. You can either change the size using the text box, by inserting your desired size, or by using the slider. The size can be 100 px at maximum. Brush size affects both the primary and secondary colors of your brush,

**5: Opacity controllers**

The opacity of your brush can be changed using the controllers in this section. You can either change the opacity using the text box, by inserting your desired size, or by using the slider. The minimum opacity is 0, which is completely transparent. The maximum is 255, which is completely opaque. Brush opacity affects both the primary and secondary colors of your brush,

**6: Zoom level**

This option determines the level of your zoom. To zoom, click on the drop box and pick the magnification you desire. To go back into your normal zoom, simply pick the first magnification "1x".

**7: Primary brush color**

This box shows your current primary brush color.

**8: Secondary brush color**

This box shows your current secondary brush color.

**The DrawTable widget**

The core of QtPaint is the "drawtable" widget. This widget should work on Qt framework version 5 and older. DrawTable handles the "painting" action, applies the supported effects on the image and executes the file actions such as saving and loading. It also adds scrollbars if they are necessary for completely displaying the image.

Here we will review the exclusive properties and methods of DrawTable:

**myDrawWidget(QWidget \*parent = 0)**

The constructor, like many other widgets, has a "parent" argument that can be set to any other widget. The behavior is similar to all other Qt widgets.

**void changePen(QPen npen)**

Changes the primary pen of the widget. It gets a standard QPen as its sole argument. The painting using the left-mouse-button will be done with the new pen afterwards.

**void changeSecondaryPen(QPen npen)**

Changes the secondary pen of the widget. It gets a standard QPen as its sole argument. The painting using the right-mouse-button will be done with the new pen afterwards.

**QPen getPen()**

Returns a QPen with properties similar to that of the widget's primary pen.

**QPen getSecondaryPen()**

Returns a QPen with properties similar to that of the widget's secondary pen.

**void resizePicture(int width, int height)**

Changes the size of the image. Pixels which are not shown after the operation will be lost.

**void clearImage()**

Clears the contents of the canvas. Retains the size.

**void loadImageFromFile(QString filename, const char \* format = 0)**

Loads an image from the computer into the canvas. Supports these formats: bmp, jpg, jpeg, png, gif, ppm, xbm and xpm

You can specify the format by changing the format argument. If not, the format will be determined using the file's own properties.

**void saveImageFile(QString filename, const char \* format = 0)**

Saves the contents of the canvas into your computer. Supports these formats: bmp, jpg, jpeg, png, ppm, xbm and xpm

You can specify the format by changing the format argument. If not, the format will be determined using the file's name.

If the file doesn't have an extension, it will not load anything.

**void invertImage()**

Inverts the current image.

**void changeHue(int hue)**

Changes the hue of the whole image by the value "hue". "hue" can range from 0 to 360

**void changeSaturation(int sat)**

Changes the saturation of the whole image by the value "sat". "sat" can range from -100 to 100

**void changeLightness(int lig)**

Changes the lightness of the whole image by the value "lig". "lig" can range from -100 to 100

**void setZoomLevel(int level)**

Changes the zoom level. "level" must be a natural number. Very high levels of zoom (more than 16) are not advised. Level 1 is the "normal" zoom of the widget.