

# **digiKam: ImageEditor [Port Image Editor Canvas Classes to Qt Model/View]**

Name : **Pravesh Gupta**  
Email Id : [praveshgupta1993@gmail.com](mailto:praveshgupta1993@gmail.com)  
Freenode IRC nick : **praveshgupta1993**  
Nationality : **Indian**  
Location(City,Country,TimeZone) : **Chandigarh, Panjab, India,UTC +5:30**  
Spoken Languages : **English, Hindi**

## **Proposal Title :-** **Image Editor Canvas Classes ReImplementation**

### **Short Description:-**

Image Editor Canvas classes are Qt3Support Classes based on Q3ScrollView which will disappear with next major Qt release(Qt 5 and onwards). This Q3Support class is used by digiKam Image Editor to render lead editor canvas and some canvas tools as Colorcorrection, CurvesAdjust, LevelAdjust, etc. Since Qt 3 support classes will be deprecated in the coming releases while the current implementation of Image Editor Canvas Classes is on Qt 3 support, so it is necessary to port this interface to new Qt 4 model/view.

### **Motivation / Use Cases :-**

digiKam is an awesome professional open source photo management software. It does photo management, metadata management, rendering images to look more beautiful and appealing to our needs and above all this professional software is open source. digiKam is based on Qt language and uses qt3 support classes which will be deprecated with the next major qt release. So for running digiKam, there will be an requirements of older version of Qt(<4.9). Technologies come, Technologies improve and Technologies improve other Technologies. It becomes better. So to adapt to Qt(>5) technology improvement, project aim would be to port digiKam:Image Editor->Canvas Classes to Qt 4 Model/View programming structure.

Qt is a C++ based cross platform application development language with a graphical user interface(GUI). It implements Model/View Programming Architecture which manages the relationship between data models and the way it is represented to the user. Advantage of this architecture is that it provides greater flexibility to customize the presentation of items and provides a standard model interface to allow a wide range of data sources to be used with existing item views. Change in the underlying data model changes all the related views so that consistency and security of data remains. To allow the interaction between the models and views, delegates take the user inputs and actions from the views and performs changes in the underlying data model and rendering individual items in some views. It made me feel good that how with such a delecacy, this technique be implemented.

I am a C++ fan(although I have also done Java), because as long as I think all good programming languages uses C++ concept of Object Oriented Programming. Qt is also based on C++ and in addition, also provides a nice GUI framework. And with all structured model/view

coding, I was attracted to it and at ease to begin coding on this project.

## **Project Goals :-**

- Port digiKam:ImageEditor Canvas classes to model/view programming.
- Repair effected Image Editor classes to adapt Canvas classes.
- Create a new context menu for Undo & Redo functioning in User Filter Tab of Versioning Sidebar.
- Provide functioning of Pressing 'Enter' on the Selected Image Area makes 'Crop to Selection'.

## **New Features to be Added/Proposed :-**

1. Undo & Redo functioning on clicking of Filters in User Filter Tab of Versioning Sidebar.
2. Provide functioning that Pressing 'Enter' on the Selected Image Area makes 'Crop to Selection'.
3. Add new buttons in the ToolBar Area for FullScreenMode and SlideShow like in Digikam MainWindow.

## **Implementation Details :-**

### **• Current Implementation :-**

#### **| -Image Editor API :-**

---core	The core implementation including all code non relevant of GUI
---dialogs	dialog of IE, as color management settings used when image is imported to editor.
---editor	GUI classes of main windows and derivated classes shared with showfoto (stand alone version of IE)
---main	main KDE XML gui class and other common files.
---plugin	IE image editor plugins interface. This will permit tools to IE core methods.
---printiface	IE printing interface.
---q3support	All Qt3 support classes are hosted here, including the famous canvas
---rawimport	RAW import tool. Note that i already ported this tool from Qt3support to Qt4 model view.
---widgets	General widgets used in IE

Q3ScrollView is Qt3 Compatibility class. preview

Functions of Q3ScrollView that are currently being used in ImageEditor q3Support are :-

#### **Public Functions :-**

- contentsHeight
- contentsWidth
- contentsToViewPort
- contentsX
- contentY
- hScrollBarMode
- horizontalScrollBar
- repaintContents
- setCornerWidget

- setHScrollBarMode
- setVScrollBarMode
- updateContents
- verticalScrollBar
- viewport
- 
- viewportToContents
- visibleHeight
- visibleWidth

Public Slots:-

- center
- ensureVisible
- resizeContents
- scrollBy
- setContentsPos
- updateScrollBar

Public Signals:-

- horizontalSliderPressed
- horizontalSliderReleased
- verticalSliderPressed
- verticalSliderReleased

Protected Functions :-

- contentsMouseDoubleClickEvent
- contentsMouseClickedEvent
- contentsMouseReleaseEvent
- contentsMouseMoveEvent
- contentsWheelEvent
- viewportPaintEvent

Reimplemented Protected Functions :-

- resizeEvent

CornerWidget is used for panning the viewport implemented in PreviewWidget Class.

The classes in ImageEditor that are using Q3ScrollView as Base Class are :-

```

|---q3Support
|   - canvas.cpp & canvas.h      - For canvas management
|   - imageregionwidget.cpp & imageregionwidget.h - To draw an image clip region
|   - previewwidget.cpp & previewwidget.h      - To provide Image Preview on Canvas.
|   - drubberband.cpp & drubberband.h          - To select a region of image.

```

ImageRegionWidget class is used by digiKam Image editor to render canvas of tools as ColorCorrection, CurvesAdjust, LevelAdjust, etc. It uses PreviewWidget class to provide preview implementation.

PreviewWidget class inherits Q3ScrollView class to render image (zooming, panning, resizing, painting the preview).

Canvas Class inherits Q3ScrollView for painting reimplementation on the canvas.

DrubberBand takes Q3ScrollView viewport as argument to make an selection.

## • Execution of qt3Support classes in Digikam and ShowFoto :-

### Digikam :-

Here when the DigikamApp class calls ImageEditor constructor and then it constructs the main canvas using slotOpenUrl() function. In this step it creates ImageInfoModel database model and ImageFileModel for the image as a Database Model. This class processes information of image in ImageFileModel and loads the image in the Image Thumbnail for the LBN and calls Canvas class "load(const QString& filePath, IOFileSettings\* const ioFileSettings)" function which further calls EditorCore class load(const QString& filePath, IOFileSettings\* const ioFileSettings) function to dig out ImageDescription(LoadingImageDescription) and using only the filePath and thereon DImg class processes the image according to the needs. Thus in a way Canvas class works on filePath or url of the file to preview Image while the original Image information is dig out from the digiKam using ImageInfoModel database to show in the thumbbar or to show the properties.

### ShowFoto :-

Here when ShowFoto class function slotOpenUrl(KUrl url) calls Canvas class load(const QString& filePath, IOFileSettings\* const ioFileSettings) function. ShowFoto provides filePath QString using KUrl's function toLocalPath(). Canvas here again calls the EditorCore class function load(const QString& filePath, IOFileSettings\* const ioFileSettings) function to dig out ImageDescription(LoadingImageDescription) and thereon DImg class processes the image according to the needs.

So, here also ShowFoto supplies the filePath to Canvas class load() function which further supplies it to the EditorCore class load() function to handle.

### To be Implemented :-

Q3ScrollView is now replaced by new QAbstractScrollArea and QScrollArea classes. The methods of Q3ScrollView can be implemented using QAbstractScrollArea and QScrollArea.

Common Classes that define Model/View digiKam Canvas are :-

- GraphicsDImgView : This class is inherited from Model/View core implementation QGraphicsView to display zoomable view. QGraphicsView is further inherited from QAbstractScrollArea. GraphicsDImgView also includes QScrollBar.
- ImagePreviewItem : This class is derived from GraphicsDImgItem, which defines canvas painting reimplementation. This can be set as the item for GraphicsDImgView class using setItem() function of GraphicsDImgView.

For Porting, for the first, I will have to find out the common functionalities of Canvas and ImageRegionWidget classes such as resize(), panIconSelectionMoved(), slotCornerButtonPressed() etc. which uses Q3ScrollView to work and then, derive a new class(may be named CanvasView) that will have the common functionalities of Canvas, ImageRegionWidget class since both have their API's quite similar in some functions. This will reduce and simplify the implementation. This new class will be derived by GraphicDImgView for ScrollArea implementation and will be based upon ImagePreviewItem to make canvas painting reimplementation in Model/View. This is necessary so that I can have a firm base-ground to implement Q3ScrollView dependent functions through GraphicsDImgView.

Next will be to port DRubberBand class to Model/View and use CanvasView as argument in the constructor so that it would be ready when to be included in Canvas class.

Next task will be to port ImageRegionWidget class to Qt Model/View and then Canvas class to Qt Model/View both including the newly created CanvasView class so that the common functionalities can be accessed.

EditorStackView class is used to make a stack of the two Canvases, editor canvas & tool canvas and switching between the two at context requirement.

EditorStackView class shows canvas in two modes :-

- CanvasMode
- ToolViewMode

EditorStackView class uses PreviewWidget class to render the images in of Canvas which is in ToolViewMode i.e. which are using ImageRegionWidget class to access image to render. Since it is dependent on PreviewWidget, the next task will comprise of removing all the dependency of PreviewWidget class on EditorStackView class by simply including the CanvasView class.

Now Different Color Tools use ImageRegionWidget to show preview. So modifications in all the color files will have to implemented next to comply with the new coding structure.

### **Tentative TimeLine :-**

//To be added

### **Bugs Solved/To be Solved :-**

1. **Bug 318834** : Reported bug in digiKam/ImageEditor(ImageWindow.cpp) and then patched and committed with the help of Gilles Caulier.[1]
2. **Bug 220739** : Reported bug and tested the patch committed by Gilles Caulier.[2]
3. **Bug 307885** : Suggested an approach to solve this bug. To be Solved.[3]

### **Why I am apt for the project ? :-**

- 1) For the prerequisite, I have decent experience in coding in C,C++,Java and Qt framework. I have also done database handling using MySql. I also acquired sufficient knowledge on the working of Image Editor .
- 2) I am familiar with version controls systems, and their working, use git when required.
- 3) I have discussed the project and its technical outline of implementation with Islam Wazery <[wazery@ubuntu.com](mailto:wazery@ubuntu.com)>, mentor for this project.
- 4) I am comfortable with working under a mentor who is not in physical contact with me.
- 5) I have already downloaded the code from Git, compiled it, and collected relevant information about the execution of program(chain of command) and have read the coding of my part to understand the working of classes and relevant classes that affect my part or the relevant classes that are being affected by my part. It will surely give me a go ahead when the real coding will start.
- 6) I am quite adapt with Linux(had been a course subject to me) and KDE environment and can compile software from Tarballs and from Git.
- 7) I am a coding fan, and derives pleasure from the way coding is done.
- 8) I know how to trick Google in searching the item of my interest and does not feel shame in asking questions. I use IRC's and forums whenever I feel the sense of asking a question.
- 9) I am hardworking, clever and dedicated to my work. Generally when I don't get something right, there is constant eagerness from inside to find the right way.
- 10) Fluent with English so conversation won't be a problem(also speaks Hindi).
- 11) I am in contact with past indian GSOC(2012) aspirant, Abhinav Badola<[mail.abu@gmail.com](mailto:mail.abu@gmail.com)>.
- 12) I have well read the timeline of GSOC.

**Do you have other obligations from late May to early August (school, work, vacation, etc.)? Please note that we expect the Summer of code to be a full time, 40-hr a week occupation. It is important to be clear and upfront about other commitments that you may have during that**

## **time.**

I will be having my 6th semester exams from 20<sup>th</sup>May-31<sup>st</sup>May, so I will be indulged in them till 31<sup>st</sup>May. After that, I will be free during the entire summer coding period and will be available approximately 9 hours a day, 6 days a week. My college will start from the first week of August, at that time I will be able to commit myself 6 hours everyday, 7 days a week.

## **About Me :-**

My name is Pravesh Gupta. I am a middler studying Information Technology at University Institute of Engineering and Technology(UIET), Panjab University, Chandigarh, one of the oldest universities of India. Actually I am from Delhi, India.

I like programming and hacking. Apart from the regular college courses, I do hacking now and then. I know how to break Windows Login Password, how to stop wifi access to an wifi connected ip address, how to find the person who has stopped mine. I have gone till level 7 of [www.hackthissite.com](http://www.hackthissite.com).

I am also drawn to logical problems and find them attracting and interesting. For C++, I have also solved some mathematical/computer programming problems on [www.projecteuler.net](http://www.projecteuler.net) in my freshman year. I have also participated in the Infosys India Logical Programming Competition- Aspirations. I have also made a console based soduku & puzzle game in my freshman year while learning C & C++ language. I have also made a core java and mysql based project Driving Licence Registration System while learning core java.

I have been using Linux/Gnu for about 2 years since I came to know about open source and KDE for about 6-7 months. I have been guiding many students to use open source environments and also have attended the Software Freedom Day, North India's first open source conference.

I am a computer geek, and feel attracted to find the root cause of the problems I am facing and analyzing how the things can be done in the shortest ways. I also find tweaking the environment to my needs and make it more simple, fast and attractive as attractive. I love to read about new gadgets and new technologies. I have a great typing speed however also sometimes finds myself typing in the air.

I am a fan of music. I love soft music but I have versatile taste of music ranging from alternative, blues, pop, rap and trance. I love to watch tv show like friends, prison break, supernatural, vampire diaries....list goes on. I love to play table tennis and football.

I am open source enthusiast and wish to contribute to this great community. If given a chance to work on the project, I assure to dedication of at least 40 hours per week to the work and that I do not have other obligations during the period of the program. Also, I would continue to work on this project even after GSOC, and give effort to push the code to the stable version as soon as possible. I plan to actively maintain my code and do bug-gixing in digiKam even after my GSOC time period ends.

## **References :-**

- [1]. [https://bugs.kde.org/show\\_bug.cgi?id=318834](https://bugs.kde.org/show_bug.cgi?id=318834)
- [2]. [https://bugs.kde.org/show\\_bug.cgi?id=220739](https://bugs.kde.org/show_bug.cgi?id=220739)
- [3]. [https://bugs.kde.org/show\\_bug.cgi?id=307885](https://bugs.kde.org/show_bug.cgi?id=307885)
- [4]. <https://github.com/praveshgupta1993/Games>
- [5]. <https://github.com/praveshgupta1993/Java/tree/master/Driving-Licence-Registration-Project>
- [6]. <https://github.com/praveshgupta1993>