

NUIPaint

Exploring and developing new multitouch interactions by creating new widgets for a concrete graphics application

A GSOC 09 Proposal for NUI Group

Sharath Patali | sharath.patali@gmail.com | Mangalore | India | 26th March 2009

Abstract

Digital Photo Editing and Drawing is one of those multimedia applications which require a lot of user interactions to achieve a given task. For example a basic task like selecting a color requires 3 different mouse clicks, whereas using the Multi-touch it can be achieved through two touches. This proposal is for a multi-touch based photo manipulator and drawing application that would take advantage of the speedier, fewer and more intuitive inputs of a multi-touch surface thus enhancing the productivity of the user.

One major goal is to explore new interactions & widgets. This is a really good method because trying the widgets and interactions with an actual application is the only way to really test & put these into context. The application will be as a collection of reusable PyMT widgets thus adding value to the PyMT core.

NUIPaint is an application which provides a collaborative environment for painting and photo editing. It will also provide advanced photo editing and drawing tools in a single user environment through a natural user interface with gesture based manipulation.

NUIPaint would include following features.

- *Several new UI and non-UI PyMT (Python Multi-touch Framework) widgets*
- *Multi-user drawing and photo editing Sandbox*
- *Various kinds of brushes*
- *A gesture based UI which will reduced the number of steps to achieve a task*
- *Various GLSL (OpenGL Shading Language) based filters like brightness, contrast, hue, sharpness, blur etc*
- *Save the work to hard disk or directly upload to website such as Flickr.com*

Detailed Description

The aim of this project is to develop a multi-user photo editing and drawing sandbox with each user being able to manipulate an image or a canvas independently in separate windows. The focus is on developing a set of core UI widgets for PyMT framework and using these widgets build powerful photo editing software with user friendly interface using multi-touch and gestures. The PyMT framework is chosen as it provides the necessary platform to develop such applications.

Limitations of the existing applications

1. Existing applications are built for single user input and hence they use single user specific UI (GUI). There is no true support for Zoom-able User Interfaces (ZUI).
2. Camera based Multi-touch setup have very low resolution which hinders working at detailed levels.
3. Traditional GUI system relies on multiple mouse clicks to accomplish simple tasks. Using multi-touch there is scope for improvement.

NUIPaint application addresses the above limitation through various features as follows.

- **PyMT Widgets:** As part of the proposed application development several new UI and non-UI widgets will be added to the core PyMT framework. This would enhance the widget inventory of PyMT enabling reuse and increase adoption of the PyMT in the Open Source community. Some of the new widgets are listed below.
 - Icon Menu (with Horizontal or vertical Orientation): Animated, Customizable, using XML for specifying image and action for each icon. In the lines of OS X taskbar
 - Alert/Confirmation Widgets
 - Enhance Inner window widget with option for customizations; options for adding own button and skins: This widget will be the collaboration windows.
 - File Save/Open Dialogue Boxes: These are file system browser like widget.
 - Natural Color Mixing Widget
 - Dynamic GLSL widget: Using which a user can write his own GLSL code and apply it in real-time.
 - Enhancement to Kinetic List Widget so that it supports multiple layout objects like Icon with Title and Description
 - Image selection widget
 - Color Selection Widget: This widget can be used to select a color from palette, with support for storing favorite colors into swatch. Multiple fingers on the palette blends multiple colors.
 - Help Description Popup Widget: This is a animated widget which pops up information regarding any tool you touch on the screen
 - A Draw Brush Widget: This widget will provide a way to draw a brush head either using a tiny pixel editor or from an existing images.
 - And several others which I'll discover during the course of development.
- **Multi-user interaction:** Provide multiple windows for different users for interaction. Each window will have a reduced toolset with the most commonly used ones available in the screen. This toolset will be within the scope of the window providing a way for multiple users to work on the same surface
- **Full Screen Mode:** In the full screen mode of the window the number of tools are increased with advanced features. This utilizes the concept of Zoom-able User Interface.
- **Gesture based interface**
 - **Tool Selection Gesture:** Touching and holding on a tool widget would popup a sub widget with several sub-tools and by touching one of icons the sub-tool would get selected.
 - **Area selection through gestures:** The selection tool will be a PyMT widget which acts like an area mapping widget over an image. When the selection tool is activated and when a user touches two different points on the image like the two opposite corners of a rectangle box, a rectangle

selection box is created there, user can resize it rotate/scale/translate the selection box, and on double tapping it, the image gets cropped to that area.(Figure below Interaction 3)

- **Zoom-in/Zoom-out gestures:** A two finger gesture like Pinch/Expand will provide a good way to focus on a particular area of interest, if a user wants to do detailing on a particular area he can zoom in and then work on it, obviously the brush size decreases with regards to the size of the image and hence detailed work. This will help avoid the problem of lower resolution of the camera used in the multi-touch system
- **Graphical Tools:** Support for Standard tools like Textured Brushes, Selection tool, Zoom tool
- **Photo Filters:** A new core class for PyMT called Filters will be developed to supporting GLSL image processing filters similar to Google's Picasa software. These filters can be used by frame buffer objects or image data. Each filter will be modular and would provide scope for development of newer filters. GLSL will provide Hardware Acceleration to implement image processing filters like Brightness Adjustment, Contrast, Hue Correction, Sharpness, Invert, Blur
- **A Natural Color Mixing Palette widget:** Users can choose two colors, put the patch of one color on the palette and select the other color and then mix it with the existing color to get the desired result. This provides a completely natural way of user interaction.
- **UNDO/Rollback Feature:** Maintaining a history of modifications to the original image in a history file, and even save last five modifications to the computer memory so that undo can be faster.
- **Layer system:** Layer based editing system in which each layer can be a widget with transparent background. Finally they can be merged into a single image. The layers can be Frame Buffer Objects(FBO) providing hardware acceleration
- **Open & Save Files To Hard disk:** With Support for large variety of image formats. Save the edited image/canvas to the hard disk by dumping the frame buffer onto the hard drive.
- **Upload the image:** Ability to upload the work/image to a external website such as Flickr using the APIs.

Stretch Goals of the project:

- **Eye-Fi based Photo Transfer:** Transfer photos from and Eye-Fi card directly to the multi-touch table. The following existing Python based solution will be enhanced to fit in to multi-touch environment. Reference: <http://returnbooleantue.blogspot.com/2009/01/eye-fi-standalone-server.html>
- **Advanced integration with online photo website like Flickr:** Further development to support features such as searching images, selecting an image for manipulating etc.

Justification for the proposed application

- Develop and increase the widget inventory of the PyMT core by creating new widgets, improving existing widgets to be more customizable and feature rich, acting as the basis for further spawning of new widgets.

- Existing Drawing sandbox application in NUI group is only a proof of concept. The proposed application NUIPaint would be a powerful application with standard feature sets.
- The option of developing plug-ins for existing application like GIMP is not pursued because traditionally it is build for WIMP (Windows, Icons, Menus, Pointer) Architecture. There is little scope to use full potential of a multi-touch interface without considerable changes to the underlying software.
- The aim is to build an application with standard features in the three month period and later evolve it as the first full featured software application from the NUI Group
- Personnel experience of being part of the Open Source PyMT project and being well versed with PyMT widget architecture increases the chances of success of the project. Also this would showcase the power of the PyMT framework as platform for rapid prototyping as well as advanced application development.

Timeline

Phase 0 (20th April – April 27th)

Discussions with Mentors regarding the possibilities of various Application framework designs, Interaction designs

Phase 1 (28th April - May 12th)

Build Base Application Framework, with support for Layering, Undo History Maintenance mechanism, Frame Buffers

Phase 2 (May 13th – May 27th)

Develop Base UI widgets, like Icon Menu, Enhancement to Kinetic Scrollers, Open File Dialog Boxes, Alert/Confirmation Popups, Help Widgets and other UI elements

Phase 3 (May 28th - June 23rd)

Develop Application Specific widgets, like Natural Color Mixer, Selection Tools, Different Kinds of Brushes, Draw Brush Widgets, Color Selection Widgets etc.

Phase 4 (June 24th - July 8th)

Develop the GLSL Filter classes for PyMT core. This will include all modular filters like brightness, contrast, color invert, sharpness etc

Phase 5 (July 9th - July 23rd)

Development of windowing widgets and modification to existing innerWindow classes to support customization, this windowing class will be the base for Multi-user support.

Phase 6 (July 24th - August 9th)

Development of saving work area/canvas into secondary Storage and Flickr Support, Testing the widgets and fixing bugs

Phase 7 (August 10th - August 17th)

Performance Optimization, Full Testing, Betterment of Documentation

Personal Profile

Sharath Madhava Patali

Email: sharath.patali@gmail.com

Location: Mangalore, Karnataka, India

Time zone: GMT +530

Blog: sharathpatali.wordpress.com

Age: 22

Academic Background: Undergraduate in Information Science at Canara Engineering College, Affiliated to Visveswaraiah Technological University, Belgaum, Karnataka, India.

Open Source development experience: I am the contributing developer for the Open Source PyMT framework. I am part of PyMT from Jan 5th of 2009 and am well versed with the framework. My main contribution for the project are developing of example applications that showcases the framework capabilities

I have also modified the FLOSC gateway tool so that it can be configured to avoid the IR hotspot from the projector. The details of this development can be found at <http://nuigroup.com/forums/viewthread/3762/>

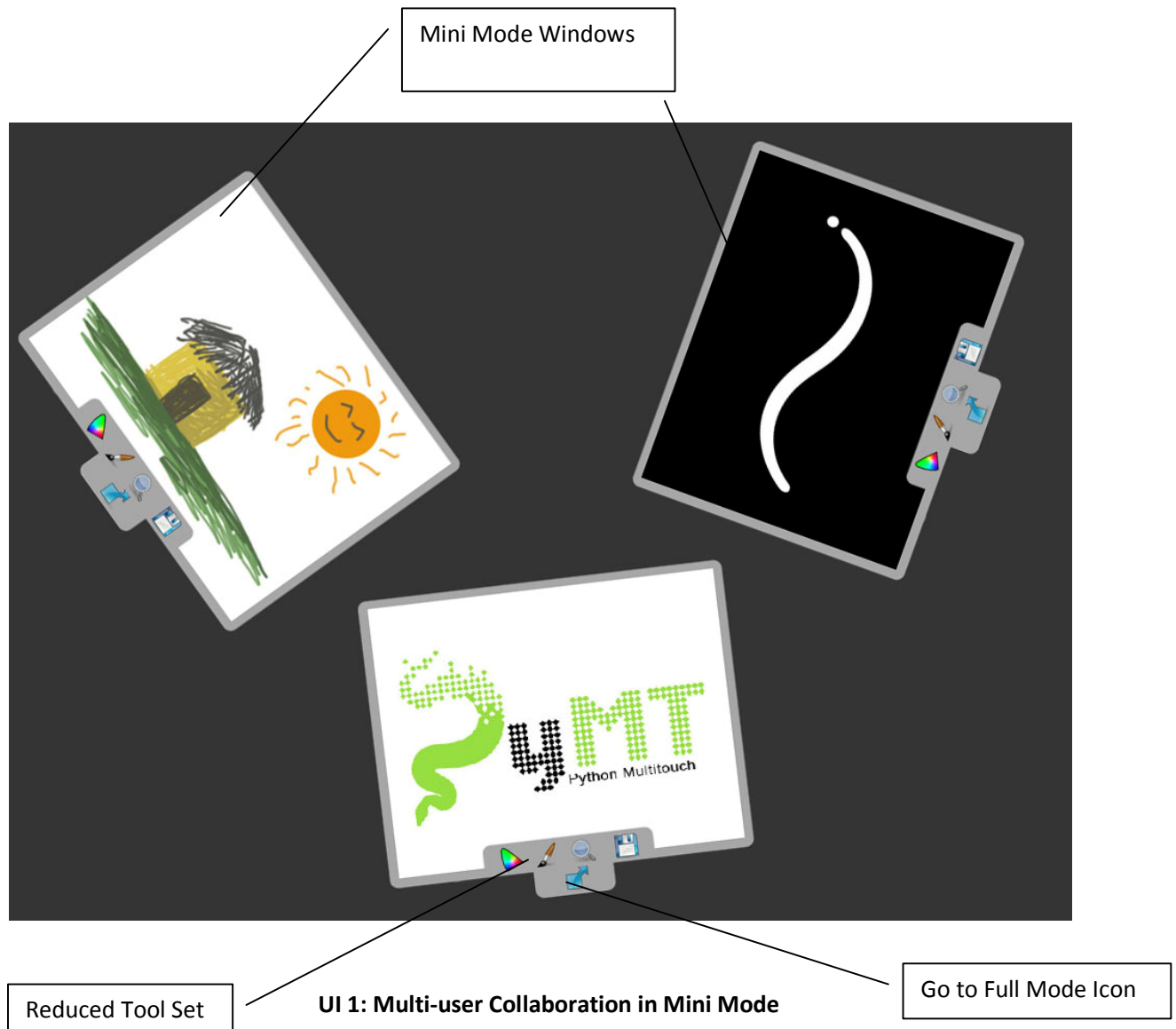
Activity level within the NUI Group Community: I am a NUI group member since April 22, 2008 and has been contributing on several aspects of Multi-touch development to the community by participating in discussions. In addition I have contributed modFlosc tool to the community.

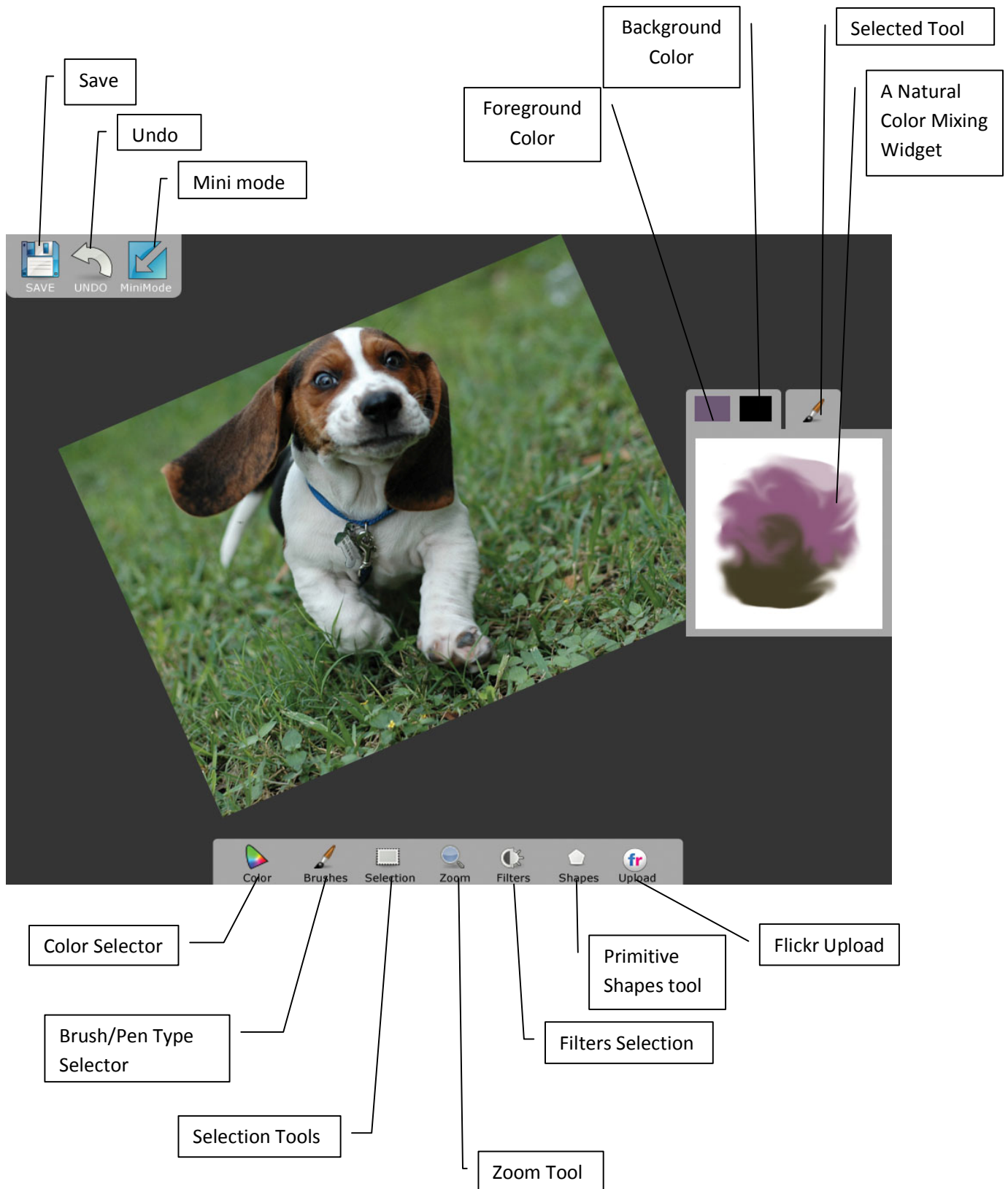
Experience with Multi-touch platform: I have more than a year of experience in the area of multi-touch surfaces and Natural User Interface. As part of my final year Bachelor of Engineering requirement I have successfully built my own FTIR few months back and presently improving on the design by building a Laser based setup. The software applications would be developed using PyMT.

References

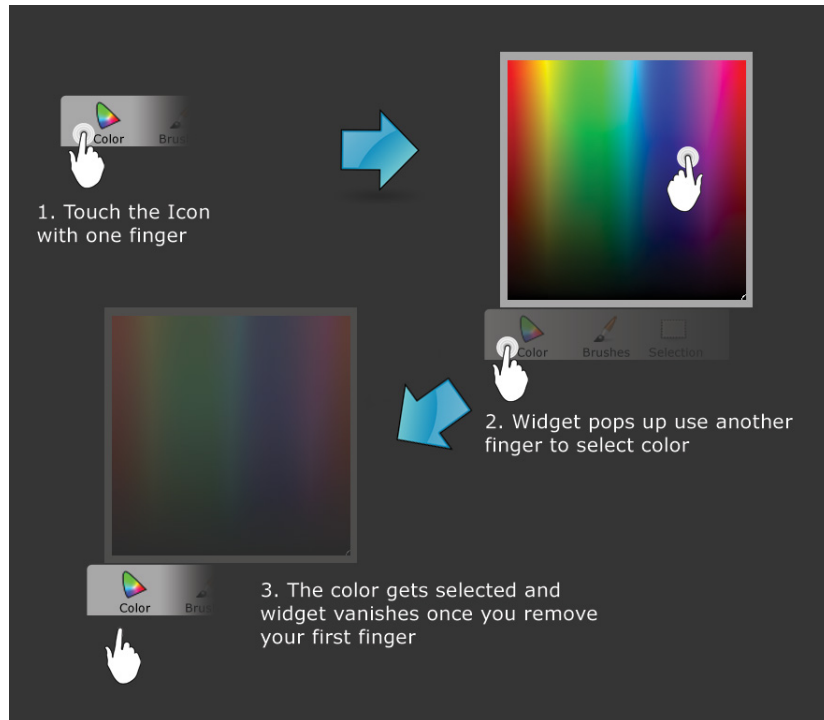
- **PyMT Framework**, a multi-touch application development framework written in python
[<http://pymt.txzone.net/>]
- **Python Imaging Library(PIL)**
[<http://www.pythonware.com/products/pil/>]
- **GLSL Filters tutorial**
[<http://student.kuleuven.be/~m0216922/CG/filtering.html>]
- **Python Flickr Api**
[<http://stuvel.eu/projects/flickrapi>]

UI mockups and Interactions

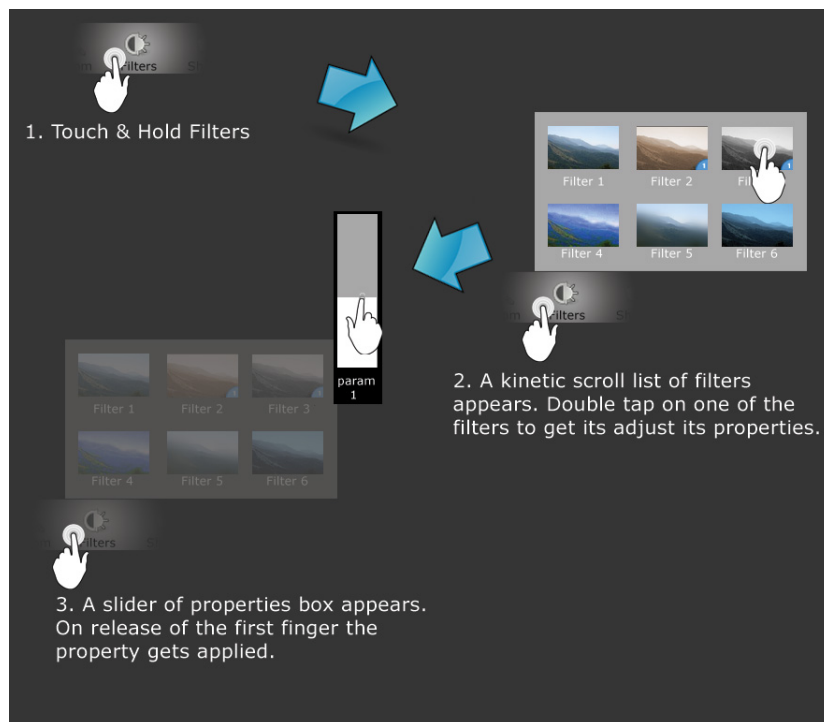




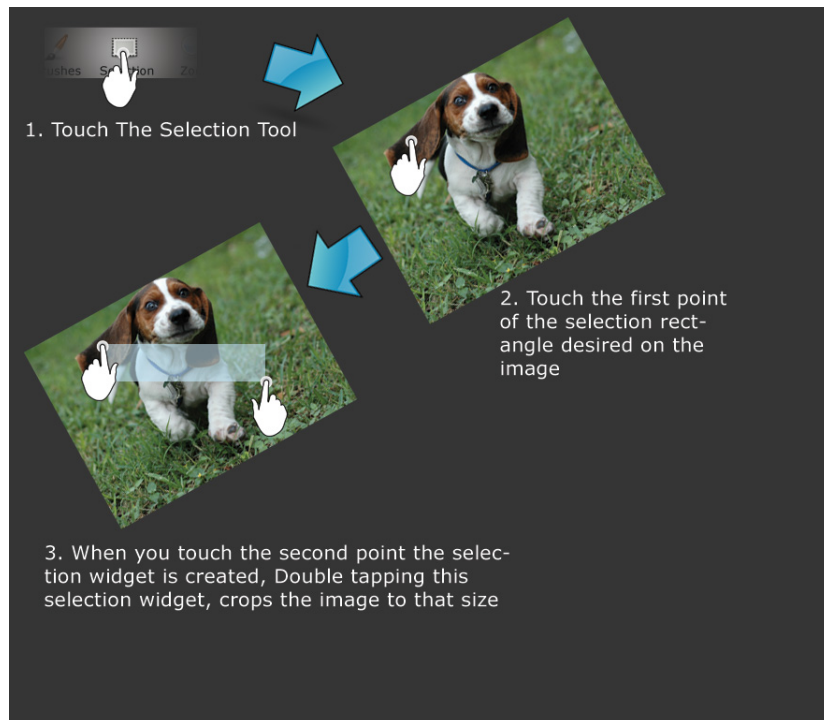
UI 2: A Full screen editing of a photo with Several Tools



Interaction 1: Color Selection Using MT



Interaction 2: Filter Selection Using MT



Interaction 3: Using Selection Tool