

# Android App Programming Directed Study ~ DrawingFun

Jenny Huang

November 17, 2014

## Contents

<b>1</b>	<b>first checkin 10/27/2014</b>	<b>1</b>
1.1	Goal . . . . .	1
1.2	Course Introduction . . . . .	1
1.3	Project Introduction . . . . .	1
1.4	References . . . . .	1
<b>2</b>	<b>Checkin for 11/3/2014</b>	<b>3</b>
2.1	Buttons I have worked on . . . . .	3
2.1.1	Color <sub>Picker</sub> : . . . . .	3
2.1.2	Undo/Redo: . . . . .	3
2.2	Functionalities and References . . . . .	3
2.2.1	Color <sub>Picker</sub> : . . . . .	3
2.2.2	Undo/Redo Buttons: . . . . .	3
2.3	Snapshot . . . . .	5
2.4	Todo . . . . .	6
2.4.1	Drawing shapes with Finger for primitives . . . . .	6
2.4.2	Load image file button . . . . .	6
2.4.3	Erase Rectangle . . . . .	6
2.4.4	Undo/Redo . . . . .	6
<b>3</b>	<b>Checkin for 11/10/2014</b>	<b>6</b>
3.1	Buttons I have worked on . . . . .	6
3.1.1	shapeBtn for primitives . . . . .	6
3.2	Functionalities and References . . . . .	6
3.2.1	shapeBtn for primitives: Drawing shapes with Finger for primitives . . . . .	6
3.3	Snapshot . . . . .	8
3.4	Special Situation . . . . .	9
3.5	Todo . . . . .	9
<b>4</b>	<b>Checkin for 11/17/2014</b>	<b>9</b>
4.1	Buttons I have worked on . . . . .	9
4.2	Functionalities and References . . . . .	9
4.2.1	openBtn for loading an image file as an ImageView . . . . .	9
4.2.2	Undo/Redo . . . . .	9
4.2.3	References: all about Android . . . . .	10
4.3	Snapshot . . . . .	12
4.4	Todo . . . . .	13

## 1 first checkin 10/27/2014

### 1.1 Goal

- According to the instructor’s requirements that we are going to implement an simple window’s Paint like Android app for later on integrating Unicon’s 2D graphics to Android app.

## 1.2 Course Introduction

- We have only two students, the other one is an undergraduate exchange student with solid Java programming background and relatively slightly weak problem-solving skills. For the first more than half semester, we used Sudoku as the starting point and tried several different topics to get our hands wet.

## 1.3 Project Introduction

- It's after middle term already, the way we were currently trying on to make it work may just work perfectly for the other classmate, but for me, I feel like it takes forever for me to be able to make any significant progress. So about half a month ago, I was motivated and thought instead of surfacing around and having fun learning by trial and error, maybe I should start from a simple GUI app as a starting point and try my best to expand/extend the APP functionality from there. And also we would be able to work to our final project slightly earlier.
- This GUI will be my very second GUI interface that I have ever created for my Computer Science major, (this first one was a Python Tkinter GUI one week short project for plotting graphics with data abstracted from a backend database during an internship;). And I guess it may still be slightly difficult for me to start writing Android App code of my own line by line, so I simply searched the internet, and trying a tutorial to make a working starting point Android Paint GUI. I integrated the codes from the reference link all together, fixed minor compile errors, and it worked!
- This "Copied" GUI will serve as the starting point, and my functionality updates start from here, and I will update my progress for this project later on by week according to the instructor's requirements and suggestions.

## 1.4 References

- <http://code.tutsplus.com/tutorials/android-sdk-create-a-drawing-app-interface-creation--mo>



# Drawing App



## 2 Checkin for 11/3/2014

### 2.1 Buttons I have worked on

#### 2.1.1 `ColorPicker`:

#### 2.1.2 Undo/Redo:

### 2.2 Functionalities and References

#### 2.2.1 `ColorPicker`:

- Motivated by the Picasso Android app, seeing their multiple color choices, our starting point **12** fixed colors were too limited.

#### 2.2.2 Undo/Redo Buttons:

- Also motivated by the Picasso app, intended to work on **Undo** button, and ended up found **Redo** button could be very convenient as well.
- needs to update these Undo/Redo methods later on, this is just the starting point most basic implementation for this button set.





## 2.4 Todo

### 2.4.1 Drawing shapes with Finger for primitives

refer to the reference below:

- <http://gmariotti.blogspot.com/2014/01/drawing-shapes-with-fingers.html>
- This button will be first priority to finish

### 2.4.2 Load image file button

### 2.4.3 Erase Rectangle

### 2.4.4 Undo/Redo

## 3 Checkin for 11/10/2014

### 3.1 Buttons I have worked on

#### 3.1.1 shapeBtn for primitives

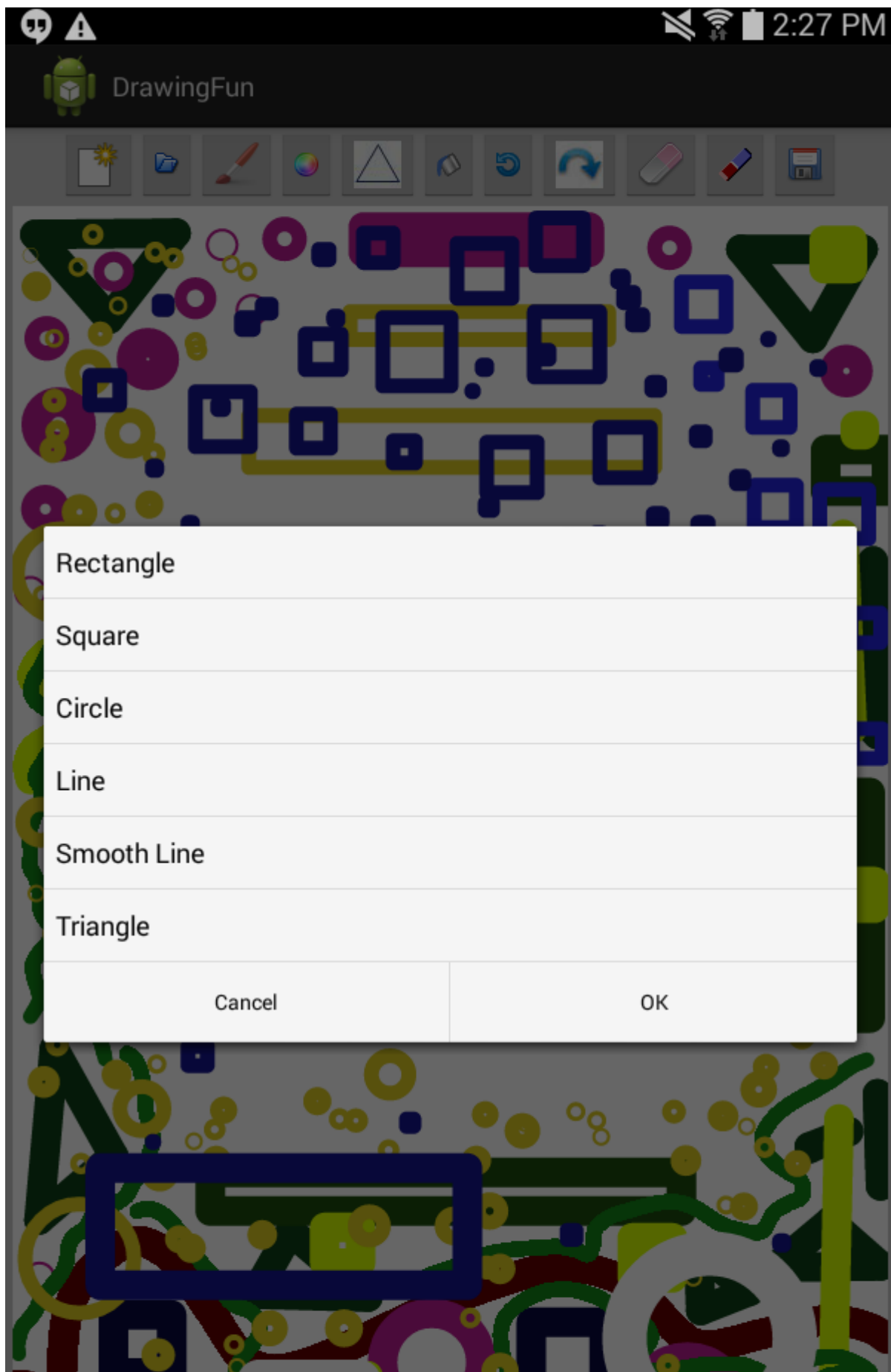
### 3.2 Functionalities and References

#### 3.2.1 shapeBtn for primitives: Drawing shapes with Finger for primitives

- refer to the reference below for some basic shapes: line, smooth line, circle, triangle, Rectangle, square
- <http://gmariotti.blogspot.com/2014/01/drawing-shapes-with-fingers.html>
- **ListView** in **Alert Dialog** is searched from online without direct reference.
- Since the erase was using draw smooth line. This button works also means that I could erase a "**Rectangle**" shape, or "**Circle**" shape.
- I have other course priority for the passed week, so I just have enough time to finish this course's priority, but I will try to work harder in order to finish all the functionalities for this course.
- It's not a good looking ListView, but yet it's a fully functional button.
- This button right now is fully functional, but to finish this project first, I have not spent any quality time to expand any primitives yet, rather than the existing six ones from the reference listed below.







### 3.4 Special Situation

- There were too many students piled/lined up in front of Dr. Jeffery's door, and he didn't break the line by stating that it's our direct study time. So the other classmate and I just stepped away from his office, and we didn't really meet during last week.
- The other classmate and I have talked, and we happened to have worked on the same shapeBtn, I applied ListView in a dialog box with all six drawing shapes applied, and he created a (ListView? not sure) with a clickable button as one element with four shapes applied. And he agreed my ListView looked way prettier than his buttons did.
- But I am willing to and more than happy to think that he could have worked on something else important for him that I actually didn't have time to work on during the passed week.

### 3.5 Todo

- Load image file button
- Erase Rectangle
- Undo/Redo
- Fill paint

## 4 Checkin for 11/17/2014

### 4.1 Buttons I have worked on

- openBtn for loading an image file as an ImageView
- Undo/Redo

### 4.2 Functionalities and References

#### 4.2.1 openBtn for loading an image file as an ImageView

- The method I applied is memory saving for AsyncTask, which is better than load images directly, which could potentially block UI for couple of seconds;
- Loaded an image from online, but would like to try load internal images from device later on, like a drawing which I saved earlier onto my internal device;
- Potentially apply layer oncepts to produce multiple layer drawing, needs suggestions to organize my idea how to implement this feature.
- **Question:** Right now, my image is an ImageView in my layout, what ideas that I could use to set/change/-transfer my ImageView to be my draw view background?
- References:
  - <http://www.learn2crack.com/2014/06/android-load-image-from-internet.html>
  - <http://stackoverflow.com/questions/5776851/load-image-from-url>
  - <https://github.com/koush/UrlImageViewHelper>

#### 4.2.2 Undo/Redo

- After implemented subclass SuperActivity class which extends Activity on week checkin for 11/10/2014 for my ListView implementation, subclass of Path() was very difficult for me to think about implement before, but after my trial on ListView, super/sub class in Java all made sense to me now. It's a piece of cake, and I know I can wrap whatever material I need in order to paint nice and neat.
- Implemented by developing a subclass myPath to wrap the super Path(), drawPaint color, and drawPaint stroksize together as an object.

- Based on previous progress that I can undo/redo only with all the drawCanvas with the same paint color, now my updo/redo paths could be colorful and with various strokesizes.

- References:

Path() library:

[http://greppcode.com/file/repository.greppcode.com/java/ext/com.google.android/android/2.3.1\\_r1/android/graphics/Path.java](http://greppcode.com/file/repository.greppcode.com/java/ext/com.google.android/android/2.3.1_r1/android/graphics/Path.java)

Bitmap cacheing:

<http://stackoverflow.com/questions/3406910/efficient-2d-drawing-in-android/3408641#3408641>

- **Questions:**

1. Undo/Redo for simple path seem to behavior fairly ok, but instead of lineTo wired line, how do I implement smooth line? How could I differentiate different strokesizes more clear with lines I have so far?

2. One little detail though, I dras after touch up, my paint color change delayed, how do I implement **real time**?

**Answered:** drawPath.reset() produced all the trouble.

3. About previous ListView six different shapes, with Undo/Redo properly functionaing, I realize I just lost my siz shapes again cause I need to rewrite/implement methods in order for them to be able to Undo/Redo ~? (My subclass works perfectly for this propose, just that I lost my internal link to primitives, which means I probably should rewrite my primitives draw methods according to undo/redo prerequisites. I don't think it will be difficult, but I don't have enough time for this for the pass week, and I need to organize my ideas about these implementation clear. )

4. I prioritize undo/redo to be more important than any other buttons cause I know they would give me great practise together with primitives implementation methods rewrite. So I have not touch "Erase Rectangle" button and "paint fill" button yet. According to these idea, I would prioritize Rectangle rewrite with the highest priority, so that later on I can follow up with erase Rectangle (which means draw Rectangle first, fill with background color, and undo could remove this erase step). Correct me if I am wrong.

#### 4.2.3 References: all about Android

- <https://github.com/keseenhoo/android-training-course-in-chinese>





## 4.4 Todo

Only two button left untouched, could do the following or anything I am interested to implement.

- Erase Rectangle
- Fill paint

Potential interests: may implement depends on how I spend thanksgiving ~

- touch ImageView Activities: zoomin, zoomout, rotate, fading, etc
- SurfaceView rotate images through new thread
- canvas save() and restore()
- OpenGL spinning circle
- widely used draw methods
- Easy draw operations