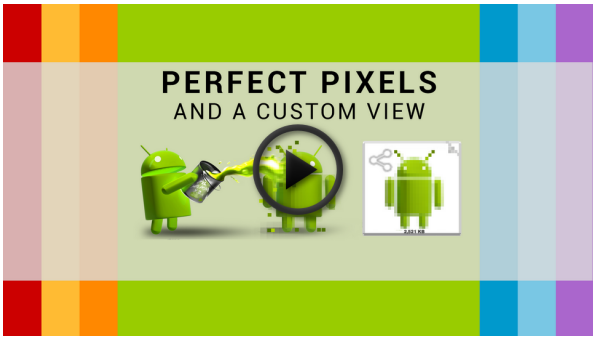
Week 8 Overview

Help

An App for Moving Pixels

On this page:

Video Lectures

Video Lecture	Key Topics	Transcript	Video Download	S Cap F
8.1. Perfect Pixels and a Custom View				
 (00:18:06)	• Programmatic views • Inner classes • Pixel perfect experiments • Working with floats • Exploring Android source with grepcode • Invalidating views		(31.3 MB)	

Next Steps for 8.1

1. Create your own project! If you need to peek at the 'solution' there is a snapshot of the project available:
2. In your onCreate method create a small 10 x 10 Bitmap and Canvas and custom View:

```
mBitmap = Bitmap.createBitmap(10, 10, Bitmap.Config.ARGB_8888);
// Then write some code to create a Canvas, some Paint and experiment with Canvas here!
// e.g. thick, thin lines and hairlines, edge-only and solid fill styles, paint and colo
// e.g. When will drawing a line or rectangle to (9,9) paint the corner pixel?
// e.g. Can you make semi-transparent paint?
// Create your own View to display your:
```

Video Lecture	Key Topics	Transcript	Video Download	S Cap F

```
mView = new View(this) {
            protected void onDraw(Canvas canvas) {
                canvas.drawColor(0xffff9900); // Orange
                float scaleX = this.getWidth() / ((float) mBitmap.getWidth());
                float scaleY = this.getHeight() / ((float) mBitmap.getHeight());
                Log.d("MainActivity","Scale:"+scaleX+","+scaleY);
                // Use canvas.save(); if you want to restore it later
                canvas.scale(scaleX, scaleY);
                // You can draw your bitmap here ...
                Paint paint = new Paint(); //
                paint.setFilterBitmap(false);
                canvas.drawBitmap(mBitmap, 0, 0, paint);

             }
          };
setContentView( mView );
```

3. Experiment with drawing lines, edge-only, solid-only, and both fill styles, paint and color—see the Java c
4. (Optional) Recommended Readings:
   - Bitmap
   - Canvas
   - Paint

8.2. Canvas Transforms to Animate My Penguin

 (00:28:56)	• Newtonian penguin • Canvas transformations • Single-touch actions		(57.9 MB)	

Next Steps for 8.2

1. Choose your own icons or graphics to use in your app. Lawrence used the Penguins with passions collec
2. Experiment with setAntiAlias(...), rotate(angle), rotate(angle,x,y), translate(tx,ty), scale(sx,sy), save(), anc

Video Lecture	Key Topics	Transcript	Video Download	S Cap F

3. Play with int and float variables (see below). Try multiplying and dividing values and passing values from When do you need "(float)"?
4. Use the current SystemClock.uptimeMillis() to animate your view. Some things to try:

```
int count = 1;
float amt = 3.5;
float test1 = count * amt; // Does this work? How do you fix it?
int test2 = count * amt; // Does this work? How do you fix it?
int total3 = 3.8 * count; // Does this work? How do you fix it?
int total = 35;
int students = 50;
float average = total/students; // Why is average 0 !?

float angle = SystemClock.uptimeMillis() / 10.0f; // Full rotation in 3.600 seconds
postInvalidateDelayed(20); // post a message on the UI message queue that will l
20 ms
```

5. At the end of your onDraw method, invalidate your view to ask it to redraw again after 20 milliseconds.
6. (Optional) Recommended Readings:
   - SystemClock
   - Canvas
   - Math public methods
   - Primitive Data Types
   - StackOverflow: Why f is placed after float values

8.3. A Trail of Transparent Pixels

 (00:12:55)	• Semi-transparent bitmap trails • Reading xml dimensions • Rescaling bitmaps		(24.6 MB)	

Next Steps for 8.3

You can download and play with Lawrence's project (MovingPixels-v3.zip). Then play with and add interactiv

Video Lecture	Key Topics	Transcript	Video Download	S Cap F

1. Add an OnTouchListener to your app:

```
        mOnTouch = new OnTouchListener() {

            @Override
            public boolean onTouch(View v, MotionEvent event) {

                // FYI For simplicity, this app is not multi-touch aware:
                // When the user performs a multi-touch event the app will get
                // large action values (because the 'action' parameter
                // also encodes additional multi-touch information)
                // So they are ignored by the app if you compare action to the ACTION_ c
                int action = event.getAction();
                Log.d(TAG,"Event:"+action+" at "+ event.getX() +"," +  event.getY());
               // Do something with action, getX and getY values!
                 return true;
           }
        };
// Don't forget to  your_view.setOnTouchListener( mOnTouch );
```

2. Experienced Java programmers users may want to learn about gestures and multi-touch handling (see li experiment with the examples in Tracking Movement
3. Add finger painting to your app. Your mutable bitmap (from createBitmap) might be a different size, so sc values:

```
                float scaleX = mBitmap.getWidth() / v.getWidth(); // Warning - fix this
                float scaleY = mBitmap.getHeight() / v.getHeight(); // Warning - fix thi
                float pointX = event.getX() * scaleX;
                float pointY = event.getY() * scaleY;
```

4. Have fun and be creative! Don't forget to share your experiences, screenshots and cool bits of code on tl
5. (Optional) Recommended Readings:
   - MotionEvent
   - View.OnTouchListener
   - Tracking Movement
   - Using Touch Gestures (Advanced)

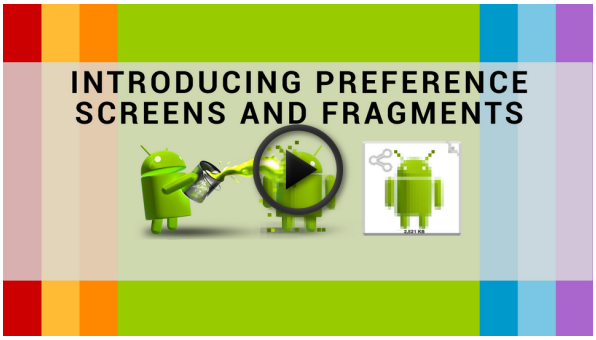
8.4. Refactoring and Custom Views in XML

Video Lecture	Key Topics	Transcript	Video Download	S Cap F
 (00:17:53)	• Refactoring towards readable code • Refactoring to nested and a separate View class • Custom views in xml		(38.6 MB)	

Next Steps for 8.4

1. Try some refactoring surgery on your own app. **We suggest you export your project source code firs start over!**
2. Refactor your onDraw method into small methods that achieve one thing.
3. Ready to be a surgeon? Time to play at code surgery! Can you do the same on your app? You will likely along the way—that's OK—it's just a hands-on way of learning what you can and cannot do with Java. S Lawrence's code changes from an anonymous inner View class to:
   - An inner class with a name ("public class ..."
   - A nested class ("public static class")
   - A completely separate Java class in its own file.
4. Unlike real surgery you can always delete your project, re-import it and start again. Pro tip: If you right-cli also use Eclipse's "Replace With" option to go back in time and replace your file(s) with an earlier versior
5. (Optional) Recommended Readings:
   - Java Methods
   - Constructor fundamentals
   - Guide to Custom Views

8.5. Introducing Preference Screens and Fragments

 (00:27:14)	• Preference settings • Developing for New and Old • Preferences for pre-Honeycomb (<API 11) • API 11+: Using fragments and fragment		(51.8 MB)	

Video Lecture	Key Topics	Transcript	Video Download	S Cap F
	transactions • PreferenceManager • Responding to the action bar/menu bar events • Listening for changes in preferences			

Next Steps for 8.5

Okay—we admit we crammed a lot of awesome content into the last video! Download Lawrence's project (M

For your own project:

1. Create your own PreferenceScreen xml file. Add a checkbox and edit text. You can also group your item PreferenceCategory.
2. Set the key, title, and summary. An example preferences file is shown below. Remember the key will be code.

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
    <CheckBoxPreference android:key="gravity" android:title="Gravity" android:summary="P
loor"/>
    <EditTextPreference android:key="name" android:title="Penguin's name" android:summar
ame?"/>
</PreferenceScreen>
```

3. Create a new class (and include it in your ApplicationManifest) that extends PreferenceActivity:

```
public class SettingsActivity extends PreferenceActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.your_prefs); // Still works on current 4.x devi
    }
```

4. If you'd like to support Fragments see additional Lawrence's SettingsActivity.java
5. Read your preference values from Java:

```
mPrefs = PreferenceManager.getDefaultSharedPreferences(getContext());
mEnableGravity = mPrefs.getBoolean("gravity", true);
```

Video Lecture	Key Topics	Transcript	Video Download	S Cap F

6. Listen for changes in preferences:

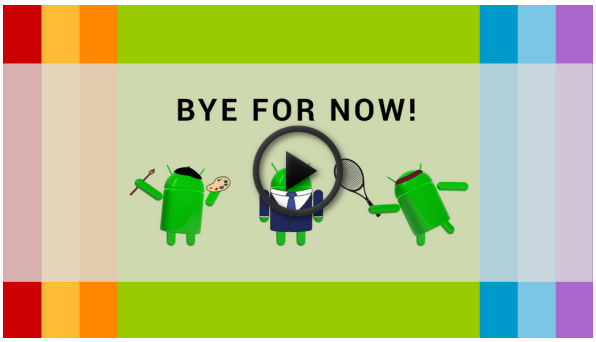
```
mPrefs.registerOnSharedPreferenceChangeListener( this );

// You will need to add "implements OnSharedPreferenceChangeListener" to the right class
public void onSharedPreferenceChanged(SharedPreferences p, String key) { Log.d(TAG,"Key
}
```

7. (Optional) In your main activity you can respond to settings item in the ActionBar/Menu to start your Setti

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if(id == R.id.action_settings) {
            Intent i = new Intent(this,SettingsActivity.class);
            startActivity(i);
            return true;
        }
        return super.onOptionsItemSelected(item);
}
```

8. Have fun playing with preferences. We look forward to seeing your creations in the forums!
9. (Optional) Recommended Readings:
   - An in-depth guide to using Preferences
   - Fragments in Android (Advanced)

Bye For Now!

 (00:01:15)	Lawrence wishes you all the best for your future endeavors. Visit the optional **Week 9: Bonus Materials and Beyond!** page for links to more resources to help you continue to expand your knowledge of application development for		(2.4 MB)	

Video Lecture	Key Topics	Transcript	Video Download	S Cap F
	Android.			

Assignments

Once you have finished watching the videos for this week, complete the quiz on the information you learned.

To begin, access the quiz page below and click the **Start Quiz Now** button at the bottom of that page. You have 5 attempts to complete this quiz.

Go to Week 8 Quiz

This quiz is due by Sunday, February 16 at 11:55 PM Central Time (time zone conversion).

This week you will complete the evaluation phase of Assignment 3. To find out more about this process, access the Assignment 3 Evaluation page below.

Go to Assignment 3 Evaluation

This assignment is due by Sunday, February 16 at 11:55 PM Central Time (time zone conversion).

Time

This module will last **7 days** and should take **approximately 4-8 hours** of dedicated time to complete, including the videos and assignments.

Tips for Success

To do well this week, I recommend that you do the following:

- Review the video lectures a number of times to gain a solid understanding of the key questions and concepts introduced this week.
- When possible, provide tips and suggestions to your peers in this class. As a learning community, we can help each other learn and grow. One way of doing this is by helping to address the questions that your peers pose. By engaging with each other, we'll all learn better.
- It's always a good idea to refer to the video lectures in your responses. When appropriate, critique the information presented.
- Take notes while you watch the lectures for this week. By taking notes, you are interacting with the material and will find that it is easier to remember and to understand. With your notes, you'll also find that it's easier to complete your assignments. So, go ahead, do yourself a favor; take some notes!

Getting and Giving Help

We strongly encourage you to join the culture of the application development community. This means not struggling with problems in isolation! Rather, when you encounter a problem, please try the following:

- Turn to your favorite search engine and search the Internet for help. Often, you will be most successful in finding the help you need by searching for the exact text of an error message you might be encountering. Sometimes, adding the term RESOLVED to your search query will help you hone-in on Discussion forum posts where someone else has received advice that ultimately resolved the problem they were encountering.
- Form groups of friends, both here in this class and perhaps locally in your geographic area. You can explore the Getting to Know Your Classmates forum, reach out via the course's social media venues, or join a Meetup.
- Use the forums dedicated to each week's topics for help solving technical problems on your computer or Android device. Please use the forum that most closely matches your problem. Explore the forum to see if others have encountered the same problem and received helpful advice that may be useful in your situation. If your problems persist, please do post in the forums to ask for help.

If you encounter a problem with the course itself, you have options! You can get help via any of the following means:

- You can report a specific problem by clicking on the **Help** link at the top right of any course page.
- Use the Course Materials Errors forum for problems with course materials such as typos, factual errors, or grading errors.
- Use the Technical Issues forum for problems related to the Coursera platform such as broken links, error messages, and other technical issues.

Due to the very large number of students enrolled in this course, the instructor is not able to answer emails sent directly to his account. Rather, all questions should be posted to one of the above forums. You are encouraged to help your fellow students by responding to posts made in these forums with solutions and by "voting up" the most important posts. University of Illinois staff will monitor these forums and will focus their attention on those that have been voted up the most.

Created Wed 12 Dec 2012 7:12 AM PST

Last Modified Tue 11 Feb 2014 12:38 PM PST